

Edícia výskumných textov
Informatiky a informačných technológií

Umelá inteligencia a kognitívna veda II

Vydané s podporou firmy Gratex International, a.s.

Táto publikácia vznikla za podpory grantovej agentúry VEGA SR v rámci grantových úloh VG1/0804/08 a VG 1/0243/10

Vladimír Kvasnička
Jiří Pospíchal
Pavol Návrat
Peter Lacko
Peter Trebatický
(editori)

Umelá inteligencia a kognitívna veda II

Slovenská technická univerzita v Bratislave, 2010

Redakčná rada série *Umelá inteligencia a kognitívna veda*

prof. RNDr. Jozef Kelemen, DrSc.
prof. Ing. Štefan Kozák, CSc.
prof. Ing. Vladimír Kvasnička, DrSc. (predseda)
prof. Ing. Pavol Návrat, CSc.
prof. RNDr. Jiří Pospíchal, DrSc.
prof. Ing. Peter Sinčák, DrSc.

PUBLIKÁCIU PODPORILO ZDRUŽENIE

GRATEX IT INŠTITÚT

v rámci fondu GraFIIT

www.gratex.com

Všetky práva vyhradené. Nijaká časť textu nesmie byť použitá na ďalšie šírenie akoukoľvek formou bez predchádzajúceho súhlasu autorov alebo vydavateľstva.

Príspevky boli recenzované redakčnou radou edície *Umelá inteligencia a kognitívna veda*.

Schválilo vedenie Fakulty informatiky a informačných technológií STU v Bratislave dňa 15.12.2009, uznesenie č. 2009.28.1

© Fakulta informatiky a informačných technológií STU v Bratislave

ISBN XXX-XX-XXX-XXXX-X

AUTORSKÝ KOLEKTÍV KNIHY

Ing. Štefan Babinec, PhD.

E-mail: stefan.babinec@stuba.sk

Ústav informatizácie, automatizácie a matematiky, Fakulta chemickej a potravinárskej technológie, Slovenská technická univerzita, Radlinského 9, 812 37 Bratislava

Mgr. Michal Bída

E-mail: michal.bida@gmail.com

Kabinet software a výuky informatiky, Matematicko-fyzikální fakulta, Univerzita Karlova, Malostranské nám. 25, 118 00 Praha 1

RNDr. Anna Bou Ezzeddine

E-mail: anna.bou.ezzeddine@stuba.sk

Ústav informatiky a softvérového inžinierstva, Fakulta informatiky a informačných technológií Slovenská technická univerzita Ilkovičova 3, 842 16 Bratislava 4

Mgr. Cyril Brom, PhD.

E-mail: brom@ksvi.mff.cuni.cz

Kabinet software a výuky informatiky, Matematicko-fyzikální fakulta, Univerzita Karlova, Malostranské nám. 25, 118 00 Praha 1

Mgr. Vladimír Dziuban

E-mail: dziuban@fmph.uniba.sk

Katedra aplikovanej informatiky, Fakulta matematiky, fyziky a informatiky Univerzita Komenského, Mlynská dolina, 848 48 Bratislava

Mgr. Juraj Frank

E-mail: frank@fmph.uniba.sk

Katedra aplikovanej informatiky, Fakulta matematiky, fyziky a informatiky Univerzita Komenského, Mlynská dolina, 848 48 Bratislava

Doc. PhDr. Dušan Gálik, CSc.

E-mail: Dusan.Galik@savba.sk

Filozofický ústav Slovenskej akadémie vied, Klemensova 19, 813 64 Bratislava

RNDr. Martin Homola

E-mail: martin.homola@fmph.uniba.sk

Katedra aplikovanej informatiky, Fakulta matematiky, fyziky a informatiky Univerzita Komenského, Mlynská dolina, 848 48 Bratislava

Mgr. Martin Klíma

ext. spolupracovník Mgr. Broma
Kabinet software a výuky informatiky, Matematicko-fyzikální fakulta, Univerzita Karlova, Malostranské nám. 25, 118 00 Praha 1

Mgr. Jakub Gemrot

E-mail: jakub.gemrot@gmail.com

Mgr. Rudolf Kadlec

E-mail: rudolf.kadlec@gmail.com

Ing. Matej Košík

E-mail: matej.kosik@fiit.stuba.sk

prof. Ing. Štefan Kozák, CSc.

E-mail: kozak@stuba.sk

prof. Ing. Vladimír Kvasnička, DrSc.

E-mail: kvasnicka@fiit.stuba.sk

doc. RNDr. Ladislav Kvasz, PhD.

E-mail: ladislav.kvasz@ku.sk

doc. Ing. Marián Mach, CSc.

E-mail: Marian.Mach@tuke.sk

doc. RNDr. Mária Markošová, PhD.

E-mail: markosova@fmph.uniba.sk

Kabinet software a výuky informatiky,
Matematicko-fyzikální fakulta,
Univerzita Karlova, Malostranské nám.
25, 118 00 Praha 1

Kabinet software a výuky informatiky,
Matematicko-fyzikální fakulta,
Univerzita Karlova, Malostranské nám.
25, 118 00 Praha 1

Ústav aplikovanej informatiky, Fakulta
informatiky a informačných technológií
Slovenská technická univerzita
Ilkovičova 3, 842 16 Bratislava 4

Ústav aplikovanej informatiky, Fakulta
informatiky a informačných technológií
Slovenská technická univerzita
Ilkovičova 3, 842 16 Bratislava 4 a

Ústav riadenia a priemyselnej
informatiky, Fakulta elektrotechniky a
informatiky, Slovenská technická
univerzita, Ilkovičova 3, 812 19
Bratislava

Ústav aplikovanej informatiky, Fakulta
informatiky a informačných technológií
Slovenská technická univerzita
Ilkovičova 3, 842 16 Bratislava 4

Katedra matematiky a didaktiky
matematiky, Ped. Fakulta Univerzity
Karlovy, 116 39 Praha a

Katedra filozofie, Filozofická Fakulta
Katolíckej Univerzity, 03 401
Ružomberok

Katedra kybernetiky a umelej
inteligencie, Fakulta elektrotechniky a
informatiky, Technická univerzita,
Letná 9, 042 00 Košice

Katedra aplikovanej informatiky,
Fakulta matematiky, fyziky a
informatiky Univerzita Komenského,
Mlynská dolina, 848 48 Bratislava

prof. Ing. Pavol Návrat, PhD.

E-mail: navrat@fiit.stuba.sk

Ústav informatiky a softvérového
inžinierstva, Fakulta informatiky a
informačných technológií Slovenská
technická univerzita Ilkovičova 3, 842
16 Bratislava 4

Bc. Radek Píbil

E-mail: black_hand@centrum.cz

Kabinet software a výuky informatiky,
Matematicko-fyzikální fakulta,
Univerzita Karlova, Malostranské nám.
25, 118 00 Praha 1

Mgr. Tomáš Plch

E-mail: tomas.plch@gmail.com

Kabinet software a výuky informatiky,
Matematicko-fyzikální fakulta,
Univerzita Karlova, Malostranské nám.
25, 118 00 Praha 1

prof. RNDr. Jiří Pospíchal, DrSc.

E-mail: pospichal@fiit.stuba.sk

Ústav aplikovanej informatiky, Fakulta
informatiky a informačných technológií
Slovenská technická univerzita
Ilkovičova 3, 842 16 Bratislava 4

Mgr. Kristína Rebrová

E-mail: kristina.rebrova@fmph.uniba.sk

Katedra aplikovanej informatiky,
Fakulta matematiky, fyziky
a informatiky Univerzita Komenského,
Mlynská dolina, 848 48 Bratislava

Doc. Ing. Lukáš Sekanina, PhD.

E-mail: sekanina@fit.vutbr.cz

Ústav počítačových systémů,
Fakulta informačních technologií,
Vysoké učení technické v Brně, 612
66 Brno

Mgr. Radovan Šikl, PhD.

E-mail: sikl@psu-cas.cz

Psychologický ústav Akademie věd ČR,
Veveří 97, 602 00 Brno

Mgr. Michal Šimeček

E-mail: michal.simecek@volny.cz

Psychologický ústav Akademie věd ČR,
Veveří 97, 602 00 Brno

Mgr. Daniel Svozil, PhD.

E-mail: daniel.svozil@gmail.com

Laboratoř informatiky a chemie,
Vysoká škola chemicko-technologická,
166 28 Praha



*Táto kniha je venovaná pamiatke zosnulého
Prof. Ing. Vladimíra Vojteka, CSc. (*1954 – † 2009)*

Predhovor

Umelá inteligencia a kognitívna veda sú už dobre inštitucionalizované vedné odbory, ktoré majú interdisciplinárne až transdisciplinárne zameranie. Dávajú rigorózne zameraným informatikom krídla fantázie, umožňujúce presah ich technickej/ prírodovednej disciplíny smerom k humanitným a behaviorálnym vedám.

Na Slovensku tieto dva odbory majú už svoje študijné programy: umelá inteligencia v Košiciach na FEI TU a kognitívna veda v Bratislave na FMFI UK. Už desať rokov sa na česko-slovenskej, alebo slovensko-českej konferencii *Kognícia a umelý život* schádzajú odborníci z rôznych vedných oblastí (informatici, matematici, filozofi, psychológovia, lekári,...), ktorých zjednocuje záujem o poznanie kognitívnych procesov, prebiehajúcich v ľudskom mozgu a o ich simuláciu metódami modernej umelej inteligencie: neurónových sietí, evolučných algoritmov a multiagentových systémov.

Edícia *Umelá inteligencia a kognitívna veda* sa zameriava na prezentáciu výsledkov tejto komunity, dosiahnutých v umelej inteligencii a v kognitívnej vede, s cieľom pomôcť študentom najmä bakalárskeho, ale aj pokročilejšieho štúdia.

Obsah predkladanej knihy *Umelá inteligencia a kognitívna veda II* dokumentuje, že tieto vedné odbory našli pevné zakotvenie tak vo filozoficky, sociálnovedne, ako aj prírodovedne a informaticky orientovanej komunite pracovníkov, v oblastiach filozofie mysle, kognitívnej psychológie, umelej inteligencie, kognitívnej vedy a umelého života.

Ďakujeme všetkým autorom, ktorí prispeli svojimi príspevkami do tejto knihy, spoločnosti Gratex International, a.s., reprezentovanej Ing. Ivanom Polášekom, PhD., za finančnú podporu vydania tejto knihy a vedeniu Fakulty informatiky a informačných technológií STU v Bratislave, menovite dekanovi fakulty prof. RNDr. Ľudovítovi Molnárovi, DrSc., za podporu vydávania tejto edície a jej zaradenie do fakultného edičného radu „Edícia výskumných textov“.

Túto knihu venujeme pamiatke zosnulého kolegu a priateľa prof. Ing. Vladimíra Vojteka, CSc. (*1954 - †2009), riaditeľa Ústavu aplikovanej informatiky FIIT STU, ktorý väčšine z nás veľmi priateľsky „šéfoval“ a mal veľké pochopenie pre naše aktivity v umelej inteligencii a kognitívnej vede.

V Bratislave, marec 2010.

Vladimír Kvasnička
Jiří Pospíchal
Pavol Návrat
Peter Lacko
Peter Trebatický
(editori)

Obsah

Predhovor

| | |
|--|-----------|
| (1) Rekurentné neurónové siete s echo stavmi | 1 |
| <i>Š. Babinec</i> | |
| 1 Úvod do Echo State neurónových sietí | 1 |
| 2 Architektúra a formálny popis..... | 4 |
| 3 Vzťah medzi ESN a neznámym systémom | 5 |
| 4 Matematický náhľad | 6 |
| 5 Trénovanie Echo State neurónových sietí | 9 |
| 6 Krátkodobá pamäť v ESN | 12 |
| 7 Podrobný popis algoritmu | 14 |
| 8 Pokročilá verzia učiaceho algoritmu | 19 |
| 9 Záver | 20 |
| Literatúra | 20 |
| | |
| (2) Virtuální agenti | 21 |
| <i>C. Brom, M. Bída, M. Klíma, J. Gemrot, R. Píbil, T. Plch, R. Kadlec</i> | |
| 1 Byl jednou jeden virtuální agent | 21 |
| 2 Virtuální agent je (skoro) z masa a kostí | 22 |
| 3 Kde bydlí virtuální agent? | 30 |
| 4 Virtuální agent chodí a ohmatává svět | 33 |
| 5 Agentova kniha Předpisů | 41 |
| 6 Spolupráce agentů | 46 |
| 7 Závěr | 47 |
| Literatura | 48 |
| | |
| (3) Sémantický web: Niektoré aktuálne výzvy | 53 |
| <i>J. Frank, V. Dzuiban a M. Homola</i> | |
| 1 Úvod | 53 |
| 2 Technológie sémantického webu | 56 |
| 3 Niektoré výzvy sémantického webu | 69 |
| 4 Záver | 84 |
| Literatúra | 85 |
| | |
| (4) Teória evolúcie a filozofia | 89 |
| <i>D. Gálik</i> | |
| 1 Úvod | 89 |
| 2 Význam teórie evolúcie, jej postavenie v systéme vied | 90 |
| 3 Stručný náčrt histórie teórie evolúcie | 97 |

| | | |
|------------|---|------------|
| 4 | Úrovne evolučného procesu | 106 |
| 5 | Univerzálny darvinizmus a univerzálna teória evolúcie | 110 |
| 6 | Záver | 118 |
| | Literatúra | 119 |
| (5) | Prírodná dedukcia | 125 |
| | <i>M. Košík</i> | |
| 1 | Úvod | 125 |
| 2 | Výroková logika | 126 |
| 3 | Predikátová logika | 137 |
| 4 | Záver | 145 |
| | Literatúra | 145 |
| (6) | Aplikovaná fuzzy logika | 147 |
| | <i>Š. Kozák</i> | |
| 1 | Charakteristika metód výpočtovej inteligencie využívajúcich fuzzy logiku | 147 |
| 2 | Oblasti aplikácie fuzzy logiky | 148 |
| 3 | Matematické modely procesov na báze fuzzy logiky | 160 |
| 4 | Zovšeobecnenie použitia jednoduchých fuzzy algoritmov riadenia | 170 |
| 5 | Aplikácia fuzzy logiky pre modelovanie a riadenie nelineárnych systémov – demonštračné príklady | 172 |
| 6 | Aplikácia fuzzy logiky a genetických algoritmov | 192 |
| 7 | Pokročilé formy riadenia na báze fuzzy logiky | 199 |
| 8 | Záver | 206 |
| | Literatúra | 207 |
| (7) | Technika sémantických tabiel v logike | 209 |
| | <i>V. Kvasnička, J. Pospíchal</i> | |
| 1 | Úvodné poznámky | 209 |
| 2 | Boolove funkcie | 215 |
| 3 | Metóda sémantických tabiel vo výrokovej logike | 223 |
| 4 | Sémantické tablá v predikátovej logike | 239 |
| 5 | Sémantické tablá v modálnej logike | 246 |
| 6 | Záver | 260 |
| | Literatúra | 260 |

| | |
|--|------------|
| (8) Náčrt teórie potencialít jazyka matematiky | 263 |
| <i>L. Kvasz</i> | |
| 1 Úvod | 263 |
| 2 Doplnenie potencialít re-prezentácií | 265 |
| 3 Náčrt potencialít objektácií | 272 |
| 4 Náčrt potencialít re-formulácií | 286 |
| Literatúra | 289 |
| (9) Riešenie problémov s ohraničeniami | 291 |
| <i>M. Mach</i> | |
| 1 Úvod | 291 |
| 2 Problémy s ohraničeniami | 292 |
| 3 Algoritmické spĺňanie ohraničení | 295 |
| 4 Stochastické prehľadávanie v akcii | 306 |
| 5 Sila znalostného prístupu | 313 |
| 6 Záver | 318 |
| Literatúra | 318 |
| (10) Dynamika sietí | 321 |
| <i>M. Markošová</i> | |
| 1 Úvodné poznámky | 321 |
| 2 Krátky pohľad do teórie grafov | 323 |
| 3 Reálne siete a ich vlastnosti | 330 |
| 4 Modely rastúcich sietí | 337 |
| 5 Hierarchické siete | 354 |
| 6 Rôzne aplikácie teórie dynamických sietí | 360 |
| 7 Perspektívy ďalšieho výskumu | 376 |
| Literatúra | 377 |
| (11) Včelí úľ ako metafora riešenia problémov v umelej inteligencii | 381 |
| <i>P. Návrat, A. Bou Ezzeddine</i> | |
| 1 Úvod | 381 |
| 2 Samoorganizácia v biologických systémoch | 383 |
| 3 Modely inšpirované správaním sa sociálneho hmyzu | 385 |
| 4 Návrh modelu včelieho úľa | 392 |
| 5 Vyhľadávanie na webe | 398 |
| 6 Stochastická optimalizácia funkcií | 403 |
| 7 Hierarchia včelieho úľa | 406 |
| 8 Záver | 407 |
| Literatúra | 407 |

| | |
|--|------------|
| (12) Vnímanie a pomenovávanie farieb a farebných kategórií | 411 |
| <i>K. Rebrová, M. Takáč</i> | |
| 1 Úvod | 411 |
| 2 Vnímanie farieb | 412 |
| 3 Základné farebné kategórie a World Color Survey | 423 |
| 4 Simulácia kategorizácie farieb | 429 |
| 5 Záver | 433 |
| Literatúra | 434 |
| (13) Evoluční návrh hardware | 437 |
| <i>L. Sekanina</i> | |
| 1 Úvod | 437 |
| 2 Evoluční návrh | 439 |
| 3 Příklady evolučního návrhu | 442 |
| 4 Evoluce přímo v hardware | 449 |
| 5 Problém implementace a evoluční návrh | 456 |
| 6 Patentová ochrana výsledků získaných evolučním návrhem .. | 459 |
| 7 Závěr | 462 |
| Literatura | 463 |
| (14) Jak a proč vznikají zrakové klamy | 467 |
| <i>M. Šimeček, R. Šíkl</i> | |
| 1 Úvod | 467 |
| 2 Kognitivní teorie zrakových klamů | 472 |
| 3 Fyziologické teorie zrakových klamů | 491 |
| 4 Od fyziologie zpět ke kognici | 501 |
| 5 Realita, iluze nebo podvod | 505 |
| 6 Závěr | 518 |
| Literatura | 519 |
| (15) Použití vrstevnatých neuronových sítí v bioinformatice | 523 |
| <i>D. Svozil</i> | |
| 1 Úvod do bioinformatické problematiky | 523 |
| 2 Rozpoznávání vzorů | 533 |
| 3 Neuronové sítě | 540 |
| 4 Aplikace neuronových sítí v bioinformatice | 550 |
| 5 Závěr | 567 |
| Literatura | 568 |
| Index | 571 |

Rekurentné neurónové siete s echo stavmi

Štefan BABINEC¹

Abstrakt. Adaptácia rekurentných neurónových sietí je zložitý proces, pričom nie je garantované, že dynamika dosiahne požadované správanie. Na druhej strane aj náhodne inicializovaná rekurentná sieť má už pred začiatkom tréningu zmysluplnú dynamiku a je schopná „predspracovať“ vstupné postupnosti a signály. Netrénovanú rekurentnú sieť možno použiť v rámci komplexnejšej architektúry, a tak využiť predspracovanie vstupu na dosiahnutie požadovaného výsledku. Práve tieto poznatky viedli k vytvoreniu špecifického druhu rekurentných neurónových sietí, označovaných ako Echo State neurónové siete, alebo neurónové siete s echo stavmi.

1 Úvod do Echo State neurónových sietí

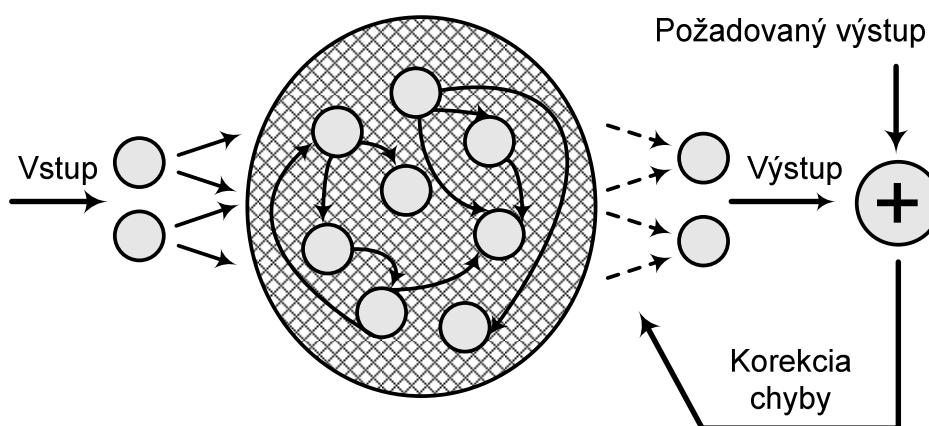
Echo State neurónová sieť (ESN) je z architektonického hľadiska rovnaká ako klasická rekurentná sieť. Obsahuje vstupné, skryté a výstupné neuróny organizované do vstupnej, skrytej a výstupnej vrstvy. Hlavný rozdiel spočíva v rekurentnej časti siete, ktorá je v prípade ESN tvorená veľkým počtom zvyčajne riedko poprepájaných neurónov s netrénovanými spojeniami (dynamický rezervoár – DR). K adaptácii váh dochádza iba vo výstupnej vrstve, ktorej úlohou je využiť predspracovanie vstupu náhodne inicializovaným dynamickým rezervoárom k produkovaniu požadovaného výstupu (obr. 1).

Pri klasických rekurentných neurónových sieťach (RC ANN – Recurrent Artificial Neural Network, obr. 2) je z hľadiska časovej a výpočtovej náročnosti tréningu najproblematickejšou časťou práve rekurentná vrstva siete. Keďže v prípade ESN táto časť nie je trénovaná, môže obsahovať veľký počet neurónov bez toho, aby sa výrazne zvýšila časová náročnosť tréningu. Dynamický rezervoár teda obsahuje stovky neurónov, zatiaľ čo pri klasických sieťach je veľkosť rekurentnej časti siete z uvedeného dôvodu obmedzená rádovo na desiatky neurónov. Ako bude ukázané neskôr, pri tréningu váh výstupnej vrstvy nie je potrebné šíriť chybový signál späť v čase, a preto sa na ich tréning využíva algoritmus lineárnej regresie. Prínosom je podstatne nižšia výpočtová zložitosť, ako aj fakt, že nevzniká problém lokálnych

¹ Fakulta Chemickej a Potravinárskej Technológie, Slovenská Technická Univerzita, Oddelenie Matematiky, Radlinského 9, Bratislava 812 37, E-mail: stefan.babinec@stuba.sk

miním, ktoré sú známym obmedzením algoritmov založených na minimalizácii chyby metódou najprudšieho spádu [5].

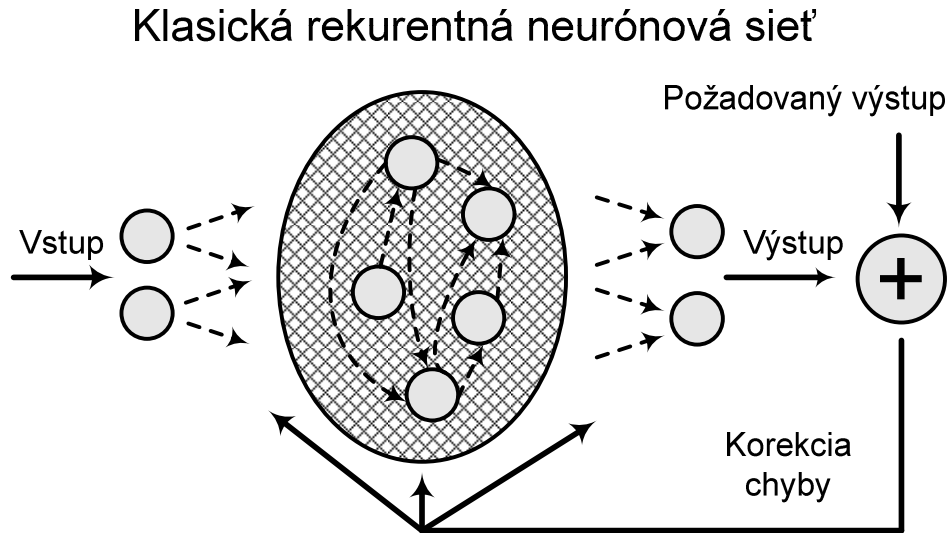
Echo State neurónová sieť



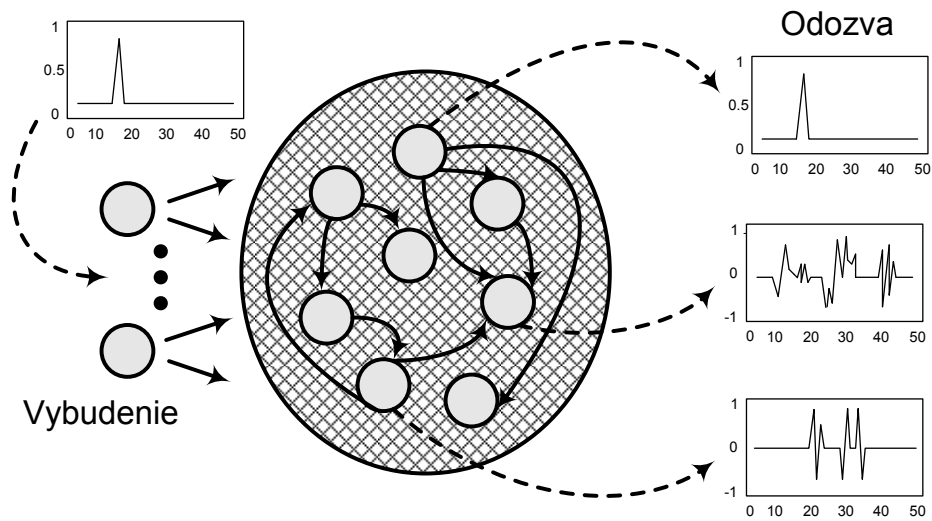
Obr. 1. Echo State neurónová sieť s netrénovanou náhodne inicializovanou rekurentnou vrstvou, takzvaným dynamickým rezervoárom. K adaptácii váh dochádza iba medzi skrytou a výstupnou vrstvou.

Echo State neurónové siete sa teda na rozdiel od klasických sietí nepokúšajú dosiahnuť požadované správanie (dynamiku) adaptáciou váh relatívne malého počtu neurónov, ale využívajú dynamiku mohutnej (náhodne inicializovanej) skrytej vrstvy na extrahovanie zaujímavých vlastností zo vstupnej postupnosti. Úlohou výstupných neurónov je transformovať tieto vlastnosti (aktivity neurónov v dynamickom rezervoári) do požadovaného výstupu.

Proces extrahovania zaujímavých vlastností dynamickým rezervoárom je znázornený na obr. 3. Vstupný signál sa privedie prostredníctvom vstupných neurónov do skrytej vrstvy, kde následne dôjde k interakcii veľkého počtu náhodne poprepájaných neurónov. Takto je výsledná odozva dynamického rezervoára v porovnaní so vstupným signálom podstatne informačne „bohatšia“. V dynamickom rezervoári sa typicky nachádzajú stovky neurónov a ich aktivity reprezentujú obraz vstupného signálu v mnohorozmernom stavovom priestore, ktorý zároveň tvorí priestor zaujímavých vlastností (feature space). Aj v prípade klasických prístupov k tréningu dopredných či rekurentných sietí môžeme hovoriť o priestore zaujímavých vlastností, ktorý tvoria aktivity skrytej, resp. rekurentnej vrstvy. Ten má však, ako už bolo spomenuté, zvyčajne menší počet dimenzií (typicky desiatky neurónov). Klasické prístupy sa snažia tréningom zmeniť rozloženie stavov (aktivity skrytých neurónov) v tomto priestore takým spôsobom, aby boli výstupné neuróny schopné produkovať požadované výstupy [5]. Existuje aj prístup, veľmi podobný Echo State neurónovým sieťam, ktorý sa volá „Liquid State Machine“ a bol nezávisle vynájdený a skúmaný Wolfgangom Maassom (viac v [6, 1]).



Obr. 2. Klasická rekurentná neurónová sieť obsahujúca vstupnú, skrytú a výstupnú vrstvu. Všetky váhy v tejto sieti sú tréňované.



Obr. 3. Úloha dynamického rezervoára spočíva vo vytváraní potenciálne zaujímavých vlastností zo vstupného signálu. Riedko poprepájané neuróny skrytej vrstvy produkujú odozvu, ktorá je „informačne bohatším obrazom” vstupnej postupnosti [2].

2 Architektúra a formálny popis

V tejto časti si zjednotíme označovanie jednotlivých vektorov a matíc, ktoré budeme používať v nasledujúcich kapitolách.

Sieť sa bude skladať z K vstupných neurónov, pričom vstupný vektor si označíme ako:

$$\vec{u}(n) = (u_1(n), \dots, u_K(n))^t, \quad (1)$$

z N skrytých (vnútorných) neurónov, pričom stavový vektor si označíme ako:

$$\vec{x}(n) = (x_1(n), \dots, x_N(n))^t, \quad (2)$$

a L výstupných neurónov, pričom výstupný vektor označíme ako:

$$\vec{y}(n) = (y_1(n), \dots, y_L(n))^t, \quad (3)$$

kde index t označuje transponovanie a n predstavuje časovú jednotku. Zavedieme si ešte vektor požadovaných výstupov, ktorý si označíme ako:

$$\vec{d}(n) = (d_1(n), \dots, d_L(n))^t, \quad (4)$$

Všetky synaptické váhy v ESN budeme ukladať do takzvaných váhových matíc. Matica vstupných váh bude mať rozmery $N \times K$ a označíme si ju

$$\vec{W}^{in} = (w_{ij}^{in}). \quad (5)$$

Matica skrytých váh bude mať rozmery $N \times N$ a označíme si ju

$$\vec{W} = (w_{ij}). \quad (6)$$

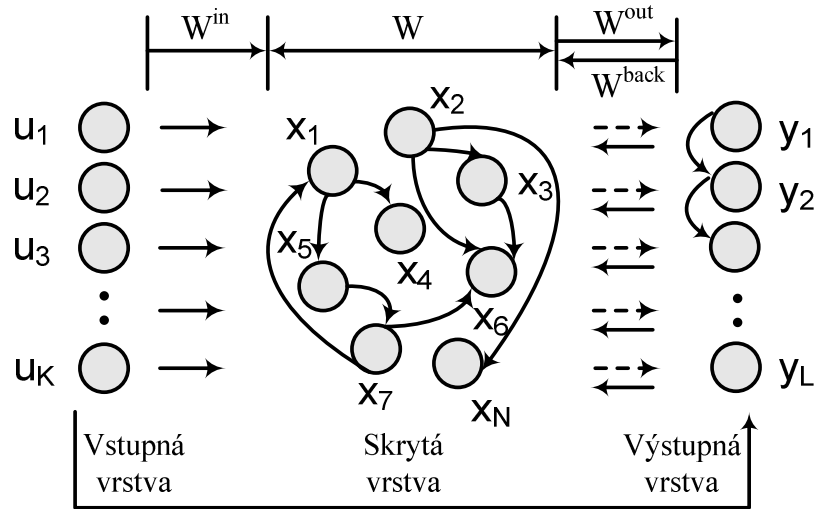
a maticu výstupných váh s rozmermi $L \times (K + N + L)$ označíme

$$\vec{W}^{out} = (w_{ij}^{out}). \quad (7)$$

V niektorých úlohách riešených pomocou ESN môže byť nutné použiť aj spätné synaptické váhy z výstupných neurónov do dynamického rezervoára. Matica týchto váh bude mať rozmery $N \times L$ a označíme ju ako

$$\vec{W}^{back} = (w_{ij}^{back}). \quad (8)$$

Nulovú hodnotu v ktorejkoľvek váhovej matici môžeme interpretovať ako neexistenciu spojenia medzi danými neurónmi. Treba si všimnúť, že matica \vec{W}^{out} nemusí obsahovať iba váhy z vnútorných neurónov. Môžu existovať spojenia aj priamo z vstupných neurónov na výstupné a takisto vzájomné spojenia medzi výstupnými neurónmi.



Obr. 4. Základná architektúra Echo State neurónovej siete. Medzi voliteľné spojenia patria spojenia priamo medzi vstupnou a výstupnou vrstvou a spojenia navzájom medzi jednotlivými výstupnými neurónmi. V niektorých typoch úloh je nutné použiť aj spätné synaptické spojenia z výstupných neurónov do skrytej vrstvy. Šípky v poslednom rade, smerujúce zo skrytej vrstvy (dynamického rezervoára) do výstupnej vrstvy, označujú spojenia, ktoré sa ako jediné trénujú v Echo State neurónových sieťach.

3 Vzťah medzi ESN a neznámym systémom

Z pohľadu systémového inžinierstva, dynamika (neznámeho) systému závisí na nasledujúcom formalizme :

$$\vec{d}(n) = e(\vec{u}(n), \vec{u}(n-1), \dots, \vec{d}(n-1), \vec{d}(n-2), \dots), \quad (9)$$

kde e je (pravdepodobne vysoko komplexná a nelineárna) funkcia predchádzajúcich vstupov a výstupov nášho systému. Vzťah (9) ukazuje najvšeobecnejší možný spôsob, ako sa dá popísať deterministicky stacionárny systém. V inžinierskych problémoch dochádza často k zjednodušovaniu uvedenej rovnice. Napríklad sa predpokladá, že funkcia e je lineárna a má iba konečnú množinu argumentov (čiže systém má konečnú pamäť). My sa ale budeme držať našej všeobecnej verzie.

Úloha nájdenia vhodného modelu pre neznámy systém (9) nás zavedie k hľadaniu vhodnej aproximácie už spomínanej systémovej funkcie e . Budeme predpokladať ESN s lineárnymi výstupnými neurónmi za účelom zjednodušenia zápisu. Potom výstup trénovanej neurónovej siete je lineárnou kombináciou jej jednotlivých stavov.

$$\begin{aligned}
e(\vec{u}(n), \vec{u}(n-1), \dots, \vec{d}(n-1), \vec{d}(n-2), \dots) &= \\
&= \vec{d}(n) \\
&\approx \vec{y}(n) \\
&= \sum w_i^{out} x_i(n) \\
&= \sum w_i^{out} e_i(\vec{u}(n), \vec{u}(n-1), \dots, \vec{y}(n-1), \vec{y}(n-2), \dots).
\end{aligned} \tag{10}$$

Zo zápisu (10) je jasne viditeľné, že požadovaná aproximácia systémovej funkcie e môže byť interpretovaná ako lineárna kombinácia echo funkcií e_i . Argumenty funkcií e a e_i sú v princípe identické: v oboch prípadoch je to kolekcia predchádzajúcich vstupov a systémových výstupov (respektíve výstupov neurónovej siete). Bez existencie echo stavov by sme na jednej strane nemohli matematicky porozumieť vzťahu medzi výstupom neurónovej siete a skutočným výstupom systému a na druhej strane by ani algoritmus tréningovania Echo State neurónových sietí nebol funkčný.

4 Matematický náhľad

Hlavným kľúčom k pochopeniu ESN je koncept echo stavov. Existencia (alebo neexistencia) echo stavov v neurónovej sieti vyjadruje schopnosť tejto siete učiť sa (alebo sa neučiť). Pri detailnejšom pohľade zistíme, že táto vlastnosť závisí od váhových matic \vec{W}^{in} , \vec{W} a prípadne, ak existuje, tak aj matice \vec{W}^{back} . Táto schopnosť je takisto relatívna vzhľadom k tréningovým dátam. Tá istá netréningovaná sieť môže mať echo stavy pre určité dáta, ale v prípade použitia iných dát už nie. Z tohto dôvodu požadujeme, aby tréningový vstupný vektor $\vec{u}(n)$ pochádzal z kompaktného intervalu U a tréningový výstupný vektor $\vec{d}(n)$ z kompaktného intervalu D . Najskôr si zavedieme presnú matematickú definíciu echo stavov a následne sa pokúsime o intuitívnu interpretáciu tejto definície.

Definícia 1. (Echo Stav) Predpokladajme netréningovanú neurónovú sieť s váhovými maticami \vec{W}^{in} , \vec{W} a \vec{W}^{back} . Táto sieť nech je budená učiacim vstupom $\vec{u}(n)$ a učiacim (požadovaným) výstupom $\vec{d}(n)$, ktoré nech sú z kompaktných intervalov U a D . Neurónová sieť (\vec{W}^{in} , \vec{W} , \vec{W}^{back}) má echo stavy vzhľadom na U a D , ak pre každú vstupno/výstupnú sekvenciu $(\vec{u}(n), \vec{d}(n-1))$, zľava idúcu od nekonečna, kde $n = \dots, -2, -1, 0$, a pre všetky sekvencie stavov $\vec{x}(n), \vec{x}'(n)$, ktoré dostaneme pomocou vzťahov

$$\vec{x}(n+1) = \vec{f}(\vec{W}^{in}\vec{u}(n+1) + \vec{W}\vec{x}(n) + \vec{W}^{back}\vec{d}(n)) \tag{11}$$

$$\vec{x}'(n+1) = \vec{f}(\vec{W}^{in}\vec{u}(n+1) + \vec{W}\vec{x}'(n) + \vec{W}^{back}\vec{d}(n)) \tag{12}$$

platí, že $\vec{x}(n) = \vec{x}'(n)$ pre všetky $n \leq 0$.

Intuitívne môžeme povedať, že ak neurónová sieť „bežala“ veľmi dlhý čas (od mínus nekonečna), aktuálny stav neurónovej siete je jedinečne určený históriou predkladaných vstupov a požadovaných výstupov. Ekvivalentne môžeme povedať, že pre každý vnútorný signál $x_i(n)$ existuje echo funkcia e_i , ktorá mapuje históriu vstupov a výstupov do aktuálneho stavu:

$$e_i : (U \times D)^{-N} \rightarrow R \quad (13)$$

$$(\dots, (\vec{u}(-1), \vec{d}(-2)), (\vec{u}(0), \vec{d}(-1))) \rightarrow x_i(0).$$

Zjednodušene môžeme povedať, že (trénovaná) neurónová sieť $(\vec{W}^{in}, \vec{W}, \vec{W}^{out}, \vec{W}^{back})$ je Echo State neurónovou sieťou ak jej netrénované „jadro“ $(\vec{W}^{in}, \vec{W}, \vec{W}^{back})$ má schopnosť dosiahnuť echo stavy vzhľadom na vstupno-výstupnú sekvenciu z ľubovoľného kompaktného intervalu $U \times D$.

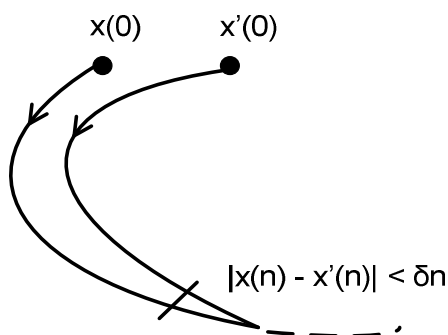
Existuje mnoho iných podmienok, ktoré sú ekvivalentné s podmienkami potrebnými pre dosiahnutie echo stavov. Uvedieme ešte jednu definíciu ekvivalentnú s definíciou 1. Podrobne v [2].

Definícia 2. (Kontraktívnosť). S tými istými predpokladmi ako v definícii 1, neurónová sieť $(\vec{W}^{in}, \vec{W}, \vec{W}^{back})$ je kontraktívna vzhľadom na U a D , ak pre všetky vstupno/výstupné sekvencie $(\vec{u}(n), \vec{d}(n-1)) \in U \times D$ sprava idúce do nekonečna, kde $n = 0, 1, 2, \dots$ existuje $(\delta_n)_{n \geq 0}$, také, že pre všetky počiatočné stavy $\vec{x}(0), \vec{x}'(0)$ a pre všetky $n \geq 0$ platí, že $|\vec{x}(n) - \vec{x}'(n)| < \delta_n$, kde $\vec{x}(n)$ (resp. $\vec{x}'(n)$) je stav neurónovej siete v čase n .

Intuitívne môžeme povedať, že časom dôjde k takzvanému „vymytiu“ vplyvu počiatočnej inicializácie neurónovej siete (obr. 5).

Schopnosť dosiahnuť echo stavy úzko súvisí s algebraickými vlastnosťami váhovej matice \vec{W} . Žiaľ, zatiaľ nie sú známe žiadne nutné a postačujúce podmienky, na základe ktorých by sa dalo jednoznačne rozhodnúť, či pri zadaných $\vec{W}^{in}, \vec{W}, \vec{W}^{back}$ má daná neurónová sieť schopnosť dosiahnuť echo stavy alebo nie.

Veta 1. Predpokladajme netrénovanú sieť $(\vec{W}^{in}, \vec{W}, \vec{W}^{back})$. Stav tejto siete v nasledujúcom kroku vypočítame na základe vzťahu (11) a ako prechodovú funkciu použijeme tanh (hyperbolický tangens). Nech \vec{W} má spektrálny polomer $|\lambda_{\max}| > 1$, kde $|\lambda_{\max}|$ je najväčšia absolútna hodnota vlastného vektora matice \vec{W} . Potom táto neurónová sieť nemá schopnosť dosiahnuť echo stavy vzhľadom na žiadnu vstupno/výstupnú sekvenciu z intervalu $U \times D$, obsahujúcu nulovú vstupno/výstupnú sekvenciu $(0, 0)$.



Obr. 5. Grafické znázornenie efektu nazývaného „vymytie“ vplyvu počiatkovej inicializácie Echo State neurónovej siete.

Na prvý pohľad sa nám môže zdať, že toto tvrdenie nám nijako nepomôže pri hľadaní Echo State neurónových sietí. Ale praktickými experimentmi bolo zistené, že ak nedodržíme podmienky uvedené vo vete 1, čiže ak spektrálny polomer váhovej matice je menší ako jedna, tak neurónová sieť, ktorú získame týmto spôsobom, má echo stavy.

Hypotéza 1. Nech δ a ε sú dve malé kladné čísla. Potom existuje veľkosť neurónovej siete N taká, že ak dynamický rezervoár o veľkosti N skonštruujeme náhodným vygenerovaním matice \vec{W}_0 za použitia uniformného rozdelenia a z rozsahu $[-1,1]$, ďalej túto maticu \vec{W}_0 normalizujeme na \vec{W}_1 , $\vec{W}_1 = \frac{1}{|\lambda_{\max}|} \vec{W}_0$, kde $|\lambda_{\max}|$ je spektrálny polomer matice \vec{W}_0 a nakoniec upravíme maticu \vec{W}_1 na maticu \vec{W}_2 , $\vec{W}_2 = (1 - \delta)\vec{W}_1$, čím \vec{W}_2 získa spektrálny polomer $(1 - \delta)$, tak potom neurónová sieť $\vec{W}^{in}, \vec{W}_2, \vec{W}^{back}$ má echo stavy s pravdepodobnosťou $1 - \varepsilon$.

Všimnime si, že vo vete 1 a aj v hypotéze 1 sa vôbec nespomína obmedzenie vzťahujúce sa na váhové matice \vec{W}^{in} , a \vec{W}^{back} . Zdá sa, že hodnoty týchto váh sú nepodstatné pre dosiahnutie echo stavov. V praxi sa ukázalo, že tieto váhy môžu byť ľubovoľne zvolené, bez toho aby to nejakým spôsobom ovplyvnilo schopnosť neurónových sietí dosiahnuť echo stavy.

Starostlivá voľba spektrálneho polomeru váhovej matice dynamického rezervoára \vec{W} má kľúčový dosah na eventuality úspech tréningu Echo State neurónových sietí. Dôvodom je fakt, že spektrálny polomer je v úzkom kontakte s vlastnou dynamikou skrytej vrstvy Echo State neurónovej siete. Malé hodnoty spektrálneho polomeru vytvoria takzvaný „rýchly“ dynamický rezervoár, veľké hodnoty naopak „pomalý“ dynamický rezervoár. Vlastná dynamika úlohy by mala zodpovedať dynamike Echo State neurónovej siete. Treba poznamenať, že dynamika Echo State neurónovej siete závisí exponenciálne na hodnote $1 - (\text{spektrálny polomer DR})$. Takže ak budeme voliť

hodnoty spektrálneho polomeru dynamického rezervoára v poradí: 0.99, 0.98, 0.97, bude to viesť k exponenciálnemu a nie k lineárnemu nárastu dynamiky neurónovej siete.

Ako už bolo povedané, podmienkou pre existenciu echo stavov sú hodnoty váh v dynamickom rezervoári – prvky matice \vec{W} . Ak je norma tejto matice menšia ako 1, odozva dynamického rezervoára môže byť použitá v ESN sieti. Základná úvaha pre existenciu echo stavov je znázornená v rovnici (14). Stav siete v nasledujúcom časovom kroku $\vec{x}(n+1)$ je určený nasledujúcim vstupom $\vec{u}(n+1)$, stavom siete v aktuálnom kroku $\vec{x}(n)$ a aktuálnym požadovaným výstupom $\vec{d}(n)$. Avšak aj aktuálny stav $\vec{x}(n)$ je možné vyjadriť ako funkciu vstupu $\vec{u}(n)$, stavu $\vec{x}(n-1)$ a požadovaného výstupu $\vec{d}(n-1)$.

$$\vec{x}(n+1) = \vec{f}(\vec{W}^{in}\vec{u}(n+1) + \vec{W}\vec{x}(n) + \vec{W}^{back}\vec{d}(n)) \quad (14)$$

kde

$$\vec{x}(n) = \vec{f}(\vec{W}^{in}\vec{u}(n) + \vec{W}\vec{x}(n-1) + \vec{W}^{back}\vec{d}(n-1))$$

kde

$$\vec{x}(n-1) = \vec{f}(\vec{W}^{in}\vec{u}(n-1) + \vec{W}\vec{x}(n-2) + \vec{W}^{back}\vec{d}(n-2))$$

atď.

Pri tomto rozvoji je názornejšia úloha matice \vec{W} v dynamike siete. Ak je norma matice \vec{W} menšia ako 1, tak sieť má tendenciu postupne „zabúdať“ informáciu z predchádzajúcich časových krokov a „najväčší“ vplyv na stav má aktuálny vstup. Ak je teda sieť prezentovaná najprv rozdielna a následne od určitého času identická postupnosť, sieť postupne zabudne na počiatkové rozdiely a s narastajúcim časom bude odozva dynamického rezervoára pre obe postupnosti rovnaká. Inými slovami, sieť má v sebe zahrnutú pamäť s exponenciálnym poklesom informácie [5].

5 Trénovanie Echo State neurónových sietí

Ako už bolo zmieňované, pri trénovaní Echo State neurónových sietí dochádza iba k adaptácii výstupných váh. Získanie Echo State neurónovej siete s požadovanými vstupno-výstupnými vlastnosťami je ale dvojstupňová úloha. Najprv si musíme zabezpečiť RC ANN so schopnosťou mať echo stavy. V druhej časti je našou úlohou vypočítanie výstupných synoptických váh. Teraz si popíšeme základnú myšlienku tejto stratégie.

V tejto kapitole budeme pracovať so systémom, ktorý je budený iba vstupným signálom. Prevezmeme si štandardný pohľad z teórie systémov a predstavíme si dynamický systém (deterministický a v diskretnom čase) ako funkciu \vec{G} , ktorá produkuje výstup systému na základe histórie jeho vstupov.

$$\bar{y}(n+1) = \bar{G}(\dots, \bar{u}(n), \bar{u}(n+1)). \quad (15)$$

Pri bližšom pohľade na rovnicu (15) zistíme, že je určitým spôsobom obmedzujúca. Nepripúšťa tú možnosť, že výstup systému $\bar{y}(n+1)$ môže závisieť aj na predchádzajúcich výstupoch $\bar{y}(n), \bar{y}(n-1), \dots$. Echo State neurónové siete dokážu pracovať aj s autoregresívnymi systémami, ale v tejto časti sa obmedzíme iba na čisto neautoregresívny prípad.

Teraz predpokladajme, že máme Echo State neurónovú sieť. Ďalej si predstavme, že sieti bola predkladaná dlhá vstupná sekvencia. Vzhľadom na to, že naša RC ANN má vlastnosť echo stavov a nachádza sa už v stave po počiatočnom prechodnom procese, môžu byť stavy vnútorných neurónov zapísané nasledujúcou rovnicou (s určitou voľnosťou k notácii):

$$x_i(n) \approx e_i(\dots, \bar{u}(n), \bar{u}(n+1)), \quad (16)$$

kde e_i je echo funkcia i-teho neurónu. Ak je sieť nehomogénna, rôzne echo funkcie sa budú výrazne navzájom odlišovať, a tak budú vytvárať potrebnú dynamiku neurónovej siete.

Naším cieľom je teda v zmysle rovnice (15) aproximovať systém $\bar{y}_{teach}(n+1) = \bar{G}_{teach}(\dots, \bar{u}(n), \bar{u}(n+1))$ pomocou výstupu neurónovej siete. Základnou myšlienkou je aproximácia \bar{G}_{teach} pomocou kombinácie jednotlivých echo funkcií $\bar{E} = (e_1, \dots, e_N)$, v zmysle minimalizácie strednej kvadratickej chyby. Výstup neurónovej siete vypočítame na základe vzťahu

$$\bar{y}(n+1) = \bar{f}^{out}(\bar{W}^{out}(\bar{u}(n+1), \bar{x}(n+1))). \quad (17)$$

Ako aktivačná funkcia výstupných neurónov je v tomto prípade použitá funkcia tanh (hyperbolický tangens). Keďže táto funkcia je invertovateľná, môžeme prepísať rovnicu (17) nasledujúcim spôsobom

$$(\bar{f}^{out})^{-1}(\bar{y}(n+1)) = \bar{W}^{out}(\bar{u}(n+1), \bar{x}(n+1)). \quad (18)$$

Vložením echo funkcií \bar{E} vytvoríme nový zápis

$$(\bar{f}^{out})^{-1}(\bar{y}(n+1)) = \bar{W}^{out}(\bar{u}(n+1), \bar{E}(\dots, \bar{u}(n-1), \bar{u}(n), \bar{u}(n+1))). \quad (19)$$

Rozdiel medzi skutočným a požadovaným výstupom (invertovaným vzhľadom na aktivačnú funkciu výstupných neurónov) si označíme $\mathcal{E}_{train}(n)$.

$$\begin{aligned} \mathcal{E}_{train}(n) &= (\bar{f}^{out})^{-1}(\bar{y}_{teach}(n)) - (\bar{f}^{out})^{-1}(\bar{y}(n)) \\ &= (\bar{f}^{out})^{-1}(\bar{y}_{teach}(n)) - \bar{W}^{out}(\bar{u}(n), \bar{E}(\dots, \bar{u}(n-1), \bar{u}(n))). \end{aligned} \quad (20)$$

Posledným krokom je vypočítanie váhových koeficientov nachádzajúcich sa v matici \bar{W}^{out} takým spôsobom, aby sme minimalizovali strednú kvadratickú chybu (MSE)

$$MSE_{train} = \frac{1}{(n_{max} - n_{min} + 1)} \sum_{n=n_{min}}^{n_{max}} \varepsilon_{train}^2(n). \quad (21)$$

MSE_{train} je chyba na trénovacej množine, n_{min} je index trénovacej vzorky, pri ktorej už došlo k „vymytiu“ vplyvu počiatocnej inicializácie a n_{max} je index poslednej trénovacej vzorky.

Túto sekciu si uzavrieme krátkym zhrnutím všeobecnej verzie trénovacieho procesu.

- Úloha

Dané: trénovacia vstupno-výstupná sekvencia $(\vec{u}_{teach}(n), \vec{y}_{teach}(n))_{n=0, \dots, n_{max}}$.

Požadované: RC ANN, ktorej výstup $\vec{y}(n)$ aproximuje $\vec{y}_{teach}(n)$.

- Vytvorenie RC ANN s echo stavmi

Najprv skonštruujeme neurónovú sieť s echo stavmi a následne zvolíme takú veľkosť dynamického rezervoára, aby odrážala potreby či už komplexnosti úlohy alebo požadovanej presnosti.

- Vzorkovacia fáza

Začneme z ľubovoľného počiatocného stavu neurónovej siete $\vec{x}(0)$ a vypočítame nový stav siete pre každú vstupnú vzorku $n = 0, \dots, n_{max}$ na základe vzorca:

$$\vec{x}(n+1) = \vec{f}(\vec{W}^{in} \vec{u}_{teach}(n+1) + \vec{W} \vec{x}(n)) \quad (22)$$

Následne je dôležité zvoliť si vhodný časový okamih n_{min} , po ktorom predpokladáme, že už došlo k vymytiu vplyvu počiatocnej inicializácie siete. Po tomto okamihu je už stav neurónovej siete určený iba predchádzajúcimi stavmi a vstupmi neurónovej siete.

- Vypočítanie výstupných váh

Nech $\vec{y}_{teach}(n) = (y_{teach,1}(n), \dots, y_{teach,L}(n))$ a $\vec{G}'_{teach}(n) = (\vec{f}^{out})^{-1}(\vec{y}_{teach}(n))$.

Výstupné váhy \vec{W}^{out} vypočítame takým spôsobom, aby sme minimalizovali MSE:

$$\frac{\sum_{n=n_{min}}^{n_{max}} (\vec{G}'_{teach}(n) - \vec{W}^{out}(\vec{u}_{teach}(n), \vec{E}(\dots, \vec{u}_{teach}(n-1), \vec{u}_{teach}(n))))^2}{(n_{max} - n_{min} + 1)}. \quad (23)$$

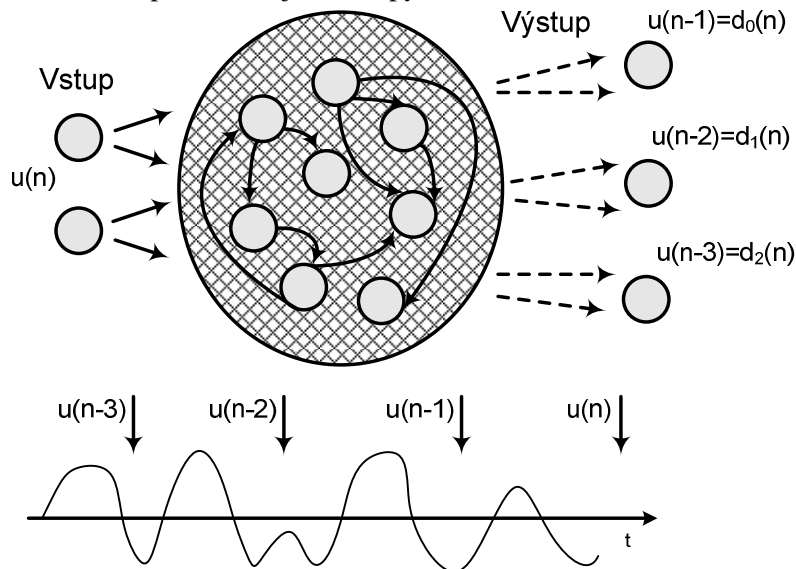
K vypočítaniu týchto váh môžeme použiť ľubovoľný algoritmus lineárnej regresie. S takto vypočítanými váhami je naša neurónová sieť pripravená na použitie.

6 Krátkodobá pamät' v ESN

Pod pojmom krátkodobá pamät' (STM – Short Term Memory) budeme rozumieť vlastnosť nejakého vstupno-výstupného systému, kde aktuálny výstup $\bar{y}(n)$ závisí na predchádzajúcich hodnotách vstupu $\bar{u}(n-k)$ a/alebo na predchádzajúcich hodnotách výstupu $\bar{y}(n-k)$. Táto vlastnosť úzko súvisí s prechodnými dynamickými procesmi v neurónovej sieti. Ako náprotivok ku krátkodobej pamäti existuje dlhodobá pamät' (LTM – Long Term Memory), za ktorú zodpovedá zmena synaptických váh počas učiaceho procesu neurónovej siete.

Klasickým prístupom pri potrebe krátkodobej pamäte (hlavne v dopredných neurónových sieťach) je vytvorenie takzvaného časového okna na vstupe neurónovej siete. V tomto prípade nebudeme na vstup neurónovej siete privádzať iba sekvenciu v aktuálnom časovom kroku $\bar{u}(n)$, ale aj jej predchádzajúce hodnoty $\bar{u}(n), \bar{u}(n-1), \dots, \bar{u}(n-k)$. Na rozdiel od tohto prístupu rekurentné neurónové siete ponúkajú alternatívne riešenie potrieb krátkodobej pamäti. Z dôvodu existencie rekurentných prepojení aktuálny stav neurónovej siete $\bar{x}(n)$ obsahuje aj informáciu o predchádzajúcich vstupoch. Z tohto dôvodu, nie je nutné vytvárať časové okno na vstupe neurónovej siete.

Lepšie poznatky o krátkodobej pamäti v Echo State neurónových sieťach získame, ak budeme trénovať Echo State neurónovú sieť na úlohe zameranej priamo na krátkodobú pamät' (obr. 6). Predstavme si Echo State neurónovú sieť s dvoma vstupnými neurónmi a mnohými výstupnými neurónmi. Na vstup $u(n)$ privádzame iba biely šum a na výstupné neuróny privádzame jeho časovo oneskorené hodnoty. Naším cieľom je zistiť, ako dlho a s akou presnosťou je Echo State neurónová sieť schopná zapamätávať si predchádzajúce vstupy.



Obr. 6. Echo State neurónová sieť trénovaná ako prediktor histórie vlastných vstupov.

Najprv si ale musíme uviesť presnú definíciu intuitívneho chápania takzvanej „úspešnosti“ neurónovej siete na úlohách s časovým oneskorením o k krokov. Nech $r(u(n-k), y_k(n))$ je korelačný koeficient medzi skutočným oneskoreným signálom $u(n-k)$ a hodnotou $y_k(n)$ z výstupného neurónu, ktorý sme trénovali na oneskorenie k . Tento koeficient sa pohybuje v rozpätí hodnôt -1 a 1 . Jeho umocnením získame veličinu nazývanú v štatistike determinančný koeficient $r^2(u(n-k), y_k(n))$. Hodnoty tohto koeficientu sa už pohybujú v rozpätí 0 a 1 . Hodnota 1 indikuje perfektnú koreláciu medzi skutočným signálom a výstupom neurónovej siete a hodnota 0 indikuje totálnu stratu korelácie. Perfektné obnovenie signálu oneskoreného o k krokov teda bude znamenať, že $r^2(u(n-k), y_k(n)) = 1$, kompletná strata zase $r^2(u(n-k), y_k(n)) = 0$.

Teraz si zadefinujme kapacitu pamäte (MC – Memory Capacity) neurónovej siete ako sumu všetkých týchto koeficientov cez všetky oneskorenia na výstupe neurónovej siete:

$$MC = \sum_{k=1}^{\infty} r^2(u(n-k), y_k(n)). \quad (24)$$

Keďže už máme zadefinovanú MC, môžeme uviesť najpodstatnejšie závery týkajúce sa kapacity pamäte Echo State neurónových sietí.

Veta 2. V Echo State neurónovej sieti, ktorej dynamický rezervoár sa skladá z N neurónov, kapacita pamäte MC je menšia, nanajvýš rovná N ($MC \leq N$).

To znamená, že maximálna možná kapacita pamäte ESN je ohraničená veľkosťou dynamického rezervoára.

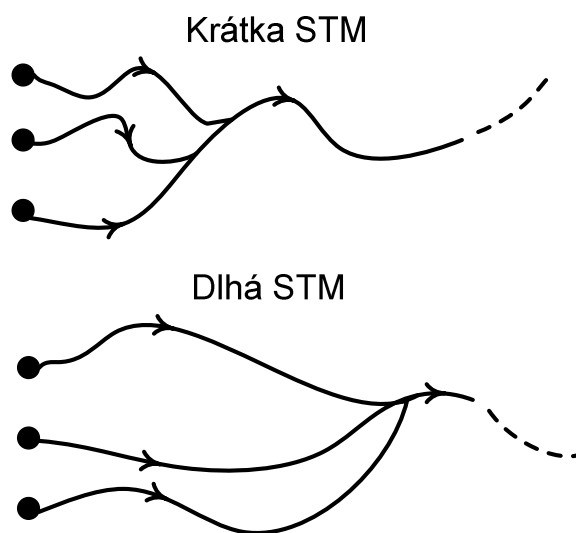
Veta 3. V lineárnej Echo State neurónovej sieti, ktorej dynamický rezervoár sa skladá z N neurónov, kapacita pamäte MC je rovná N ($MC = N$).

To znamená, že pomocou lineárnej Echo State neurónovej siete vieme dosiahnuť maximálnu možnú kapacitu pamäte. Za lineárnu neurónovú sieť považujeme takú sieť, ktorej vnútorné neuróny majú lineárnu aktivačnú funkciu, napríklad funkciu identity.

Veta 4. V lineárnej Echo State neurónovej sieti nemôžu byť dlhé oneskorenia nikdy naučené lepšie ako krátke oneskorenia (monotónne zabúdanie).

Ak potrebujeme Echo State neurónovú sieť s čo možno najdlhšou krátkodobou pamäťou (obr. 7), môžeme v praktických úlohách vyskúšať niektorý z nasledujúcich krokov (alebo ich kombináciu):

- Použijeme čo možno najväčší dynamický rezervoár. Toto je najefektívnejšia a všeobecne aplikovateľná metóda, ale vyžaduje dostatočne veľkú tréningovú množinu.
- Použijeme malé hodnoty vstupných váh. Tým zabezpečíme, že neurónová sieť bude pracovať takmer v lineárnom rozsahu. Tu sa ale môžeme dostať do konfliktu s nelineárnou charakteristikou úlohy.
- Použijeme lineárne aktivačné funkcie neurónov v dynamickom rezervoári. Opäť nám hrozí ten istý problém ako v predchádzajúcom bode.
- Použijeme špeciálne pripravený dynamický rezervoár s takmer jednotkovou váhovou maticou (bližšie informácie v [3]).
- Použijeme spektrálny polomer matice blízky hodnote 1. Takto vytvorená Echo State neurónová sieť bude fungovať úspešne iba pri úlohách s pomalou dynamikou.



Obr. 7. Porovnanie takzvanej krátkej a dlhej krátkodobej pamäte.

7 Podrobný popis algoritmu

V tejto kapitole bude podaný podrobný a kompletný popis tréningovania ESN na rozdiel od všeobecnej a zjednodušenej verzie uvedenej v kapitole 5. V tomto popise budeme predpokladať, že výstupné neuróny majú sigmoidálnu prechodovú funkciu a že existujú spätné spojenia z výstupných neurónov do dynamického rezervoára. Tým pádom sme zobrali do úvahy tú najvšeobecnejšiu verziu algoritmu. V mnohých prípadoch sa používajú jednoduchšie verzie. Napríklad výstupné neuróny nie sú sigmoidálne ale lineárne, nemusia existovať spätné spojenia z výstupných neurónov do dynamického rezervoára, alebo dokonca môžeme mať aj neurónovú sieť bez vstupných neurónov. V takýchto prípadoch nie je problém upraviť všeobecnú verziu učiaceho algoritmu tak, aby vyhovoval použitej architektúre.

- *Dané:* Trénovacia vstupno/výstupná sekvencia $(\vec{u}(1), \vec{d}(1)), \dots, (\vec{u}(T), \vec{d}(T))$.
- *Požadované:* Natrénovaná ESN $(\vec{W}^{in}, \vec{W}, \vec{W}^{out}, \vec{W}^{back})$. Od tejto natrénovanej siete vyžadujeme, aby v testovacej fáze, keď je jej predkladaný už len vstup $\vec{u}(n)$, správne aproximovala svojím výstupom $\vec{y}(n)$ požadovaný výstup $\vec{d}(n)$.
 1. V skutočnosti môžeme očakávať od ESN správnu aproximáciu požadovaného výstupu, až keď dôjde k „vymytiu“ počiatočnej prechodnej dynamiky siete. Tá je spôsobená tým, že na začiatku trénovacieho procesu sú stavy ESN náhodne (ľubovoľne) zvolené. Z tohto dôvodu vlastne požadujeme natrénovanú ESN, ktorá správne aproximuje požadovaný výstup v časových krokoch $n = T_0, \dots, T$, kde $T_0 > 1$. Hodnota parametra T_0 závisí na veľkosti dynamického rezervoára Echo State neurónovej siete. Pohybuje sa od 10 (pre malé siete) až po 500 (pre veľké siete).
 2. Čo vlastne prioritne chceme, nie je dobrá aproximácia učiacich dát, ale oveľa dôležitejšia je dobrá aproximácia testovacích dát. Tieto dáta sú samozrejme vygenerované tým istým systémom ako údaje použité na trénovanie neurónovej siete.
- *Krok 1.* V tomto kroku získame nenatrénovanú ESN $(\vec{W}^{in}, \vec{W}, \vec{W}^{back})$, ale už so schopnosťou mať echo stavy. Ak v tejto fáze privedieme na vstup nejakú sekvenciu, tak môžeme pozorovať veľmi zaujímavé dynamické správanie sa neurónov v DR.

Tento krok zahŕňa veľa heuristík. Ukážeme si najčastejšie používaný postup [2].

1. Náhodne vygenerujeme maticu vnútorných váh \vec{W}_0 (matica váh DR).
2. Vypočítame spektrálny polomer matice \vec{W}_0 , ktorý označíme $|\lambda_{\max}|$. Následne normalizujeme maticu \vec{W}_0 pomocou $|\lambda_{\max}|$, čím získame novú maticu \vec{W}_1 ,

$$\vec{W}_1 = \frac{1}{|\lambda_{\max}|} \vec{W}_0.$$
3. V tomto kroku vynásobíme maticu \vec{W}_1 parametrom α , čím získame novú maticu \vec{W} so spektrálnym polomerom α , pričom $\alpha < 1$.
4. Náhodne vygenerujeme maticu vstupných váh \vec{W}^{in} a maticu spätných váh \vec{W}^{back} . Po tomto kroku môžeme skonštatovať, že naša neurónová sieť $(\vec{W}^{in}, \vec{W}, \vec{W}^{back})$, nezávisle na tom ako sú zvolené $\vec{W}^{in}, \vec{W}^{back}$, je Echo State neurónovou sieťou.

Poznámky:

1. Našou snahou je zabezpečiť bohatú rozmanitosť dynamiky skrytých neurónov v DR. Z tohto dôvodu by mala matica \vec{W}_0 obsahovať dostatočné množstvo nulových (alebo nule blízky) hodnôt – vytvoríme takzvanú „riedku“ maticu. Okrem toho, váhy by mali byť zhruba vyvážené, t.j. priemerná hodnota by sa mala pohybovať okolo nuly. Jednou z možností je vygenerovanie nenulových váh z uniformného rozdelenia a rozsahu $[-1, 1]$.
 2. Veľkosť N matice \vec{W}_0 by mala odrážať aj dĺžku T tréningových dát a zároveň aj zložitosť úlohy. V praxi bolo zistené, že N by sa malo držať v rozmedzí od $T/10$ do $T/2$ v závislosti na zložitosti dát (čím pravidelnejšie dáta, tým bližšie k $T/10$ môže byť zvolená hodnota N). Týmto jednoduchým opatrením zabránime preučeniu neurónovej siete. Samozrejme, čím zložitejšie dáta, tým je potrebný väčší dynamický rezervoár.
 3. Voľba parametra α je rozhodujúca pre celkovú funkčnosť ESN. Prax je taká, že tento parameter je potrebné doladiť pre danú úlohu vyskúšaním rôznych hodnôt.
 4. Dôležitú úlohu zohráva aj veľkosť vstupných váh \vec{W}^{in} . Veľké hodnoty váh matice \vec{W}^{in} spôsobujú, že neurónová sieť je silne riadená vstupom. Naopak malé hodnoty spôsobujú, že stavy jednotlivých neurónov sú len jemne vybudené okolo kludového (nulového) stavu DR. Postupom času začnú neuróny operovať okolo lineárnej – centrálnej časti sigmoidy. Čiže získame sieť s takmer lineárnou dynamikou. Veľké hodnoty váh v matici \vec{W}^{in} vedú neuróny bližšie k saturačnej oblasti sigmoidy, čo vedie k nelineárnejšiemu správaniu výsledného modelu. V extrémnych prípadoch by mohlo dôjsť až k takzvanej binárnej dynamike siete. V tomto prípade by jednotlivé neuróny produkovali len hodnoty $-1/+1$.
 5. Obdobné pravidlá platia aj pre hodnoty váh v matici \vec{W}^{back} .
- *Krok 2.* V tomto kroku vypočítame maticu \vec{M} (matica vnútorných stavov neurónovej siete) a maticu \vec{T} (matica požadovaných výstupov neurónovej siete) – je to takzvaná vzorkovacia fáza.

Tento krok je len mechanický a nezaŕňa žiadnu heuristiku:

1. Náhodne nainicializujeme počiatočné stavy neurónovej siete, napríklad nulové stavy $\vec{x}(0) = 0$.
2. Následne začneme sieti predkladať jednotlivé kombinácie vstupov $\vec{u}(n)$ a k nim odpovedajúcich požadovaných výstupov $\vec{d}(n-1)$. To opakujeme pre

všetky učiace vzorky, $n = 0, \dots, T$. Počas toho, čo je sieť budená jednotlivými vstupno/výstupnými sekvenciami, vypočítavame stavy na jednotlivých skrytých neurónoch:

$$\vec{x}(n+1) = \vec{f}(\vec{W}^{in}\vec{u}(n+1) + \vec{W}\vec{x}(n) + \vec{W}^{back}\vec{d}(n)) \quad (25)$$

3. V čase $n = 0$, kde $\vec{d}(n)$ nie je definovaná, použijeme $\vec{d}(n) = 0$.
4. Teraz si musíme zvoliť hodnotu časového kroku T_0 , po ktorom predpokladáme, že dôjde už k „vymytiu“ vplyvu počiatočnej inicializácie neurónovej siete. Od tohto časového kroku T_0 ukladáme každý stav neurónovej siete $\vec{x}(n)$ v ďalšom kroku ako nový riadok v stavovej matici \vec{M} . Na konci tohto procesu získame maticu vnútorných stavov neurónovej siete s rozmermi $(T - T_0 + 1) \times (K + N + L)$.
5. Podobným spôsobom, pre čas väčší alebo rovný T_0 , zozbierame do matice \vec{Z} sigmoidálne-invertované učiace výstupy $\tanh^{-1}\vec{d}(n)$. Tým dostaneme maticu \vec{Z} o rozmeroch $(T - T_0 + 1) \times L$.

Poznámka: Tu si treba dať pozor, aby sme do matíc \vec{M} a \vec{Z} ukladali vektory $\vec{x}(n)$ a $\tanh^{-1}\vec{d}(n)$ a nie $\vec{x}(n)$ a $\tanh^{-1}\vec{d}(n-1)$!

- **Krok 3.** Vypočítanie výstupnej váhovej matice.

1. Najprv vypočítame pseudoinverznú maticu \vec{M}^{-1} z matice \vec{M} .
2. Následne vynásobíme maticu \vec{M}^{-1} a maticu \vec{Z} . Tým získame maticu $(\vec{W}^{out})^t$ o rozmeroch $(K + N + L) \times L$, ktorej i -ty stĺpec obsahuje výstupné váhy zo všetkých skrytých neurónov do i -teho výstupného neurónu:

$$(\vec{W}^{out})^t = \vec{M}^{-1}\vec{Z}. \quad (26)$$

3. Stačí už len transponovať maticu $(\vec{W}^{out})^t$ na maticu \vec{W}^{out} a získame nami požadovanú maticu výstupných váh.

- **Krok 4.** Fáza použitia neurónovej siete.

Echo State neurónová sieť $(\vec{W}^{in}, \vec{W}, \vec{W}^{out}, \vec{W}^{back})$ je v tejto chvíli natrénovaná a pripravená na použitie. V tejto fáze sú siete predkladané už len vstupné sekvencie a jej úlohou je čo najlepšie aproximovať požadovaný výstup.

$$\bar{x}(n+1) = \vec{f}(\vec{W}^{in}\bar{u}(n+1) + \vec{W}\bar{x}(n) + \vec{W}^{back}\bar{y}(n)), \quad (27)$$

$$\bar{y}(n+1) = \vec{f}^{out}(\vec{W}^{out}(\bar{u}(n+1), \bar{x}(n+1), \bar{y}(n))), \quad (28)$$

kde \vec{f}^{out} je aktivačná funkcia na výstupných neurónoch.

Možné varianty:

1. Pri niektorých dátach môže dochádzať k problémom so stabilitou vo fáze použitia natrénovanej neurónovej siete. V takýchto prípadoch veľmi často pomôže pridanie šumu vo vzorkovacej fáze a následne upravenie vzťahu (25):

$$\bar{x}(n+1) = \vec{f}(\vec{W}^{in}\bar{u}(n+1) + \vec{W}\bar{x}(n) + \vec{W}^{back}\bar{d}(n) + v(n)), \quad (29)$$

kde $v(n)$ je biely uniformný šum (typický rozsah od 0,0001 do 0,001). Bližšie informácie sú v [2].

2. V niektorých prípadoch nemusíme použiť synaptické spojenie smerujúce z výstupných neurónov naspäť na výstupné. V takýchto prípadoch nám vznikne matica výstupných synaptických váh \vec{W}^{out} o rozmeroch $(K+N) \times L$ a namiesto rovnice (28) použijeme nasledujúci zápis:

$$\bar{y}(n+1) = \vec{f}^{out}(\vec{W}^{out}(\bar{u}(n+1), \bar{x}(n+1))). \quad (30)$$

3. Niekedy môže byť modelovaný systém vysoko nelineárny. Pre jeho spracovanie sa dobré výsledky dosiahli použitím takzvaných „rozšírených“ stavov neurónov vo vzorkovacej fáze a následne aj vo fáze použitia. To znamená, že ako doplnok k originálnemu stavu $\bar{x}(n)$ je ešte pridaná určitá nelineárna transformácia tohto stavu. Najjednoduchšou transformáciou je druhá mocnina pôvodného stavu. Počas vzorkovacej fázy neukladáme do matice \vec{M} len stavy $\bar{x}(n) = (x_1(n), \dots, x_{K+N+L}(n))$, ale aj ich rozšírené verzie $(\bar{x}(n), \bar{x}^2(n)) = (x_1(n), \dots, x_{K+N+L}(n), x_1^2(n), \dots, x_{K+N+L}^2(n))$. Tým pádom nám vznikne matica výstupných váh \vec{W}^{out} o veľkosti $L \times 2(K+N+L)$ a namiesto vzorca (28) použijeme jeho upravenú verziu:

$$\bar{y}(n+1) = \vec{f}^{out}(\vec{W}^{out}(\bar{u}(n+1), \bar{x}(n+1), \bar{y}(n), \bar{u}^2(n+1), \bar{x}^2(n+1), \bar{y}^2(n))). \quad (31)$$

8 Pokročilá verzia učiaceho algoritmu

Túto modifikáciu učenia navrhli H. Jaeger a H. Haas [4] pre Echo State neurónovú sieť so spätnou väzbou (s váhami smerujúcimi z výstupných neurónov späť do dynamického rezervoára \vec{W}^{back}). Jej podstata spočíva v znovu využití výstupného signálu z dynamického rezervoára pre opätovné preučenie siete.

Keď natrénovaná sieť v procese „života“ autonómne generuje výstupný signál, prijíma svoj vlastný výstupný signál cez spätnoväzobné váhy. Tento signál sa líši od pôvodne použitého v procese vzorkovania, čo spôsobí, že stavy v dynamickom rezervoári budú odlišné ako v procese učenia. Ak by sa počas učenia siete privádzal na výstupné neuróny signál podobný tomu, ktorý bude vystupovať z dynamického rezervoára v procese života siete, presnosť modelu sa zvýši. Túto myšlienku možno zahrnúť do procesu učenia siete v nasledujúcich etapách:

1. Echo State neurónovú sieť najprv naučíme klasickou verziou učiaceho algoritmu, čím dostaneme predbežné výstupné váhy, ktoré si označíme ako \vec{W}_{out}^0 .
2. Z pôvodných vzoriek použitých na učenie siete a predbežných výstupných váh \vec{W}_{out}^0 vygenerujeme novú sériu tréningových vzoriek $\vec{d}'(n)$ pre $n > 1$, a to nasledovne:
 - položíme $\vec{d}'(1) = \vec{d}(1)$ a odštartujeme sieť z náhodného stavu $\vec{x}(1)$. Na výstup siete privedieme $\vec{d}(1)$ a vypočítame $\vec{x}(2)$ a $\vec{y}(2)$ pomocou týchto vzťahov:

$$\vec{x}(2) = \tanh(\vec{W}\vec{x}(1) + \vec{W}_{back}\vec{d}(1)), \quad (32)$$

$$\vec{y}(2) = \tanh(\vec{W}_{out}^0(\vec{x}(2))), \quad (33)$$

a položíme $\vec{d}'(2) = \vec{y}(2)$.

- pre ďalšie vzorky $\vec{d}'(3), \dots, \vec{d}'(n)$ použijeme nasledovné zovšeobecnené vzorce:

$$\vec{x}(n) = \tanh(\vec{W}\vec{x}(n-1) + \vec{W}_{back}\vec{d}(n-1)), \quad (34)$$

$$\vec{y}(n) = \tanh(\vec{W}_{out}^0(\vec{x}(n))), \quad (35)$$

a položíme $\vec{d}'(n) = \vec{y}(n)$.

3. Tento proces opakujeme n-krát, čím dostaneme novú sériu vzoriek, ktorá predstavuje jedнокrokovú predikciu z pôvodne natrénovanej siete.

4. Novo získané vzorky $\vec{d}'(n)$ použijeme ako nové vzorky na opätovné preučenie siete, čím dostaneme konečné výstupné váhy \vec{W}_{out} .

Bližšie informácie k tejto metóde, experimenty ako aj dosiahnuté výsledky možno nájsť v [4].

9 Záver

Echo State neurónové siete patria k relatívne novým prístupom v oblasti neurónových sietí. Ich výhodou je väčšia príbuznosť k biologickým modelom kvôli ich rekurentnému charakteru a využitiu rezervoáru dynamického správania sa bez nastavovania váh. V tomto dynamickom rezervoári sa vytvára informačne „bohatšia“ odozva vstupného signálu, ktorá je použitá na produkovanie požadovaného výstupu. Z hľadiska využitia sú vhodné na naučenie sa časovej rady, čo je nevyhnutný predpoklad napríklad pre riadenie motoriky, nech už u človeka, alebo u robota, alebo napr. na spracovanie reči. Oproti ostatným typom rekurentných sietí majú Echo State siete podstatnú výhodu v schopnosti učiť sa "jednokrokovu", keďže adaptácia výstupnej vrstvy je realizovaná metódou lineárnej regresie.

Podakovanie: Táto kapitola vznikla za podpory grantovej agentúry VEGA SR v rámci grantových úloh 1/0804/08 a 1/0141/10.

Literatúra

- [1] Goldenholz, D.: *Liquid Computing: A Real Effect*. Technical Report, Boston University Department of Biomedical Engineering, 2002.
- [2] Jaeger, H.: *The „Echo State“ Approach to Analysing and Training Recurrent Neural Networks*. German National Research Center for Information Technology, GMD report 148, 2001.
- [3] Jaeger, H.: *Short Term Memory in Echo State Networks*. German National Research Center for Information Technology, GMD report 152, 2002.
- [4] Jaeger, H., Haas, H.: *Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Telecommunication*. Science, 304, 78-80, 2004.
- [5] Makula, M., Čerňanský, M.: *Neurónové siete založené na echo stavoch*. Kognice a umělý živo IV, Opava, pp. 363-371, 2004.
- [6] Natschlag, T., Maass, W., Markram, H.: *The Liquid Computer: A Novel Strategy for Real-time Computing on Time Series*. Special Issue on Foundations of Information Processing of TELEMATIK, 8(1), 39-43, 2002.

Virtuální agenti

Cyril BROM, Michal BÍDA, Martin KLÍMA, Jakub GEMROT, Radek PÍBIL, Tomáš PLCH,
Rudolf KADLEC¹

Abstrakt. V posledných štyridsiatich rokoch došlo k nebývalému rozvoju číslicových informačných, komunikačných a riadiacich technológií a systémov, ktoré priam revolučným spôsobom menia a ovplyvňujú kvalitu a spôsob života na našej planéte. Celosvetové a celoeurópske inovatívne trendy v oblasti komplexnej informatizácie procesov a procesných činností smerujú k potrebe výskumu, vývoja a implementácie nových systémových riešení a zmien, ktoré si vyžadujú nové inteligentné, vysoko autonómne vnorené systémy riadenia, diagnostiky a komunikácie. Cieľom príspevku je návrh metodológie, metód a algoritmov inteligentnej integrovanej analýzy a syntézy informačných, komunikačných a riadiacich systémov pre nové evolučné štruktúry riadenia využívajúce vnorenosť, autonómnosť a inteligenciu.

1 Byl jednou jeden virtuální agent...

Snad odnepaměti fascinuje lidi představa, že by mohli vytvořit bytosti podobné jim samým, obdařené inteligencí. Kdo by neznal golema, Frankensteina a roboty? Některé z těchto bytostí přestávají existovat jen v našich představách a jejich zjednodušeniny začínají ožít pod rukama vědců. Zřejmě posledním přírůstkem do rodiny těchto bytostí jsou *virtuální agenti* (“virtual agents”). Jde o počítačové programy, které imitují chování inteligentních bytostí ve fiktivním světě modelovaném na počítači pomocí virtuální reality. Často jsou imitovanými bytostmi lidé, a proto se virtuálními agentům někdy také říká *virtuální lidé* (“virtual humans”). Není to ale označení přesné – opomíjí virtuální ne-lidi; zvířata, fantaskní monstra a – ano – i virtuální roboty.

Kdo nikdy neviděl virtuální agenty v akci, může pochybovat o jejich schopnostech – *ono* to opravdu něco umí? Kdo je někdy viděl krátce – například na popularizační přednášce nějakého nadšence – možná se začal bát – neovládnu nás? Kdo čte hodně sci-fi, věří futurologickým předpovědím kdejakých šarlatánů a už delší dobu hraje počítačové hry, diví se – ono to pořád *nefunguje*?

Existuje ještě čtvrtá kategorie. Byli bychom rádi, aby se do ní zařadil čtenář, který si přečte tuto kapitolu. Aby ohledně schopností virtuálních agentů (dnešních

¹ Kabinet software a výuky informatiky, Matematicko-fyzikální fakulta Univerzity Karlovy v Praze, E-mail: brom@ksvi.mff.cuni.cz

i budoucích) zachovával zdravou skepsi a uvědomoval si, že naprogramovat takového agenta je hodně těžké a že to koneckonců musí udělat člověk. Zdravá skepse je ovšem převlečený opatrný optimismus.

Oslí můstek na závěr úvodu: víte, jak poznáte *dobrého* úředníka? Když za dobrým úředníkem přijdete s problémem, snaží se vám pomoci ho vyřešit. Špatný úředník naproti tomu nalistuje Velké Pravidlo ve své Knize Předpisů a udělá to, co mu Pravidlo řekne. Jde typicky o něco, co ani trochu nesouvisí s tím, proč jste za ním přišli.

Proč o tom mluvíme? Pravda je taková, že dnešní virtuální agenti jsou jako špatní úředníci. Špatní úředníci ještě budou nějakou dobu mezi námi a stejně tak i virtuální agenti budou ještě nějakou dobu listovat ve svých Knihách Předpisů. Pojďme se na to podívat blíže... Začněme ale pěkně po pořádku. I špatný úředník je především bytost z masa a kostí.

2 Virtuální agent je (skoro) z masa a kostí

Každý virtuální agent má *tělo*. Z čeho takové tělo je? Předně si musíme uvědomit, že tato otázka má trochu jiný smysl než tatáž otázka v kontextu živého organismu.

Každý špatný úředník má tělo. Z čeho takové tělo je?

No ovšem, z masa a kostí. Trochu by to mohl kazit biolog, který by mohl namítnout – nene, z buněk. Rýpavý chemik by zase mohl začít vyjmenovávat chemické sloučeniny. Fyzik atomy. Všichni by se ale nakonec shodli, že jejich odpovědi vždy odkazují k jednomu – k matérii, z níž je tělo utkané – k hmotě. Pouze každý z nich popisuje hmotu jinou řečí.

Virtuální agent v tomto smyslu žádné tělo nemá. V jistém – jiném – smyslu je agent z pixelů na monitoru. Pixely ovšem virtuálnímu agentu *nepatří*. Když se virtuální agent pohne, zobrazí ho jiná sada pixelů: pixely jsou mu propůjčovány, zatímco atomy člověka utváří jaksi kontinuálně (pravda, sem tam nějaký atom přibude a jiný ubude, ale to sem teď nepleťme).

V dalším – opět jiném – smyslu je virtuální agent z „nul“ a „jedniček“, které běhají po základní desce počítače, procesoru, paměti, harddisku a tak dále – ale ani tyto „nuly“ a „jedničky“ se nemají k virtuálnímu agentovi tak, jako k člověku atomy.

Přesto – na obrazovce vidíme tělo virtuálního agenta (obr. 1). Dává tedy snad alespoň *nějaký* smysl ptát se, „z čeho to tělo je...“? Ano, dává. Ten smysl je určen těmi, kteří tu otázku přirozeně kladou a odpovídají na ni; grafickými designéry a programátory. Z jejich pohledu je panák na obrazovce *z kostí, z trojúhelníků, z rigu a z textury*. Tyto čtyři pojmy jsou myšlenkové konstrukty, které se ukázaly jako užitečné při rozhovorech o návrhu a tvorbě těla virtuálního agenta. Jsou natolik užitečné, že se jimi pojmenovávají konkrétní obrazce, které se používají jako meziprodukty při návrhu takového virtuálního těla. A také se jimi pojmenovávají konkrétní datové struktury, které při použití jiného měřítka vidíme jako „nuly“ a „jedničky“. Na obrazovce ovšem žádné kosti, trojúhelníky ani rig nevidíme; když už, tak snad jakousi pokřivenou texturu.

Když stavíme tělo agenta, jako jednu z prvních věcí musíme vytvarovat jeho povrch. Použijete na to *trojúhelníky* (obr. 2), obecně *polygony* (konvexní n-úhelníky; tyto dva termíny budeme dále používat jako synonyma a budeme mít namysli

trojúhelníky; pojem *trigon* se nepoužívá). Čím více jich je, tím je povrch „zaoblenější“. Panáci z dnešních počítačových her mají tělo z tisíců, ba i desetitisíců polygonů. Tělo z polygonů je jako loutka z drátů, proto se mu někdy říká *drátový model* nebo častěji (zvláště v cizině) anglickým slovem *mesh*, které původně znamená „síťovina“.

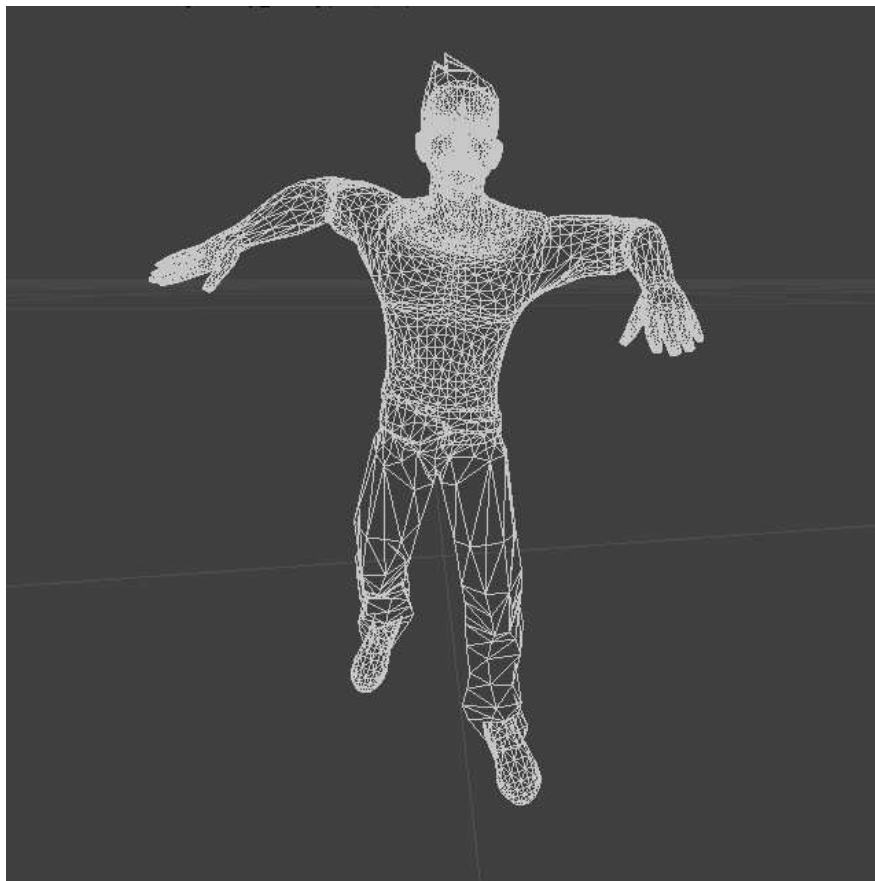
Dráty musíme obalit „omalovanou gázou“ – *texturou*. Textura určuje barvu kůže a šatů. Tomu, jak je textura na trojúhelníky připevněna, tedy který kousek textury obarví který trojúhelník, se říká *wrap*. Dvě věci jsou důležité. Zaprvé, agent většinou nemá pod šaty kůži: má šaty „přilepené“ rovnou na dráty – jako ta loutka. Zadruhé, „výčnělky“ na těle panáka musíme udělat pomocí polygonů. Textura je „plochá“. Chcete náramek? Přidejte padesát polygonů. Je nicméně pravda, že některé malé „výčnělky“ lze, ba i velmi dobře, „nakamuflovat“ vhodným stínováním textury, popř. aplikací dalších efektů.



Obr. 1. Ukázka virtuálního agenta.

Trojúhelníky a textura tvoří dohromady cosi jako kůži a šaty. Co je „pod“ nimi? A potřebujeme vůbec něco? Potřebujeme, jinak by měl náš agent problém se hýbat. Pokud chcete udělat s panákem krok, znamená to rozpohybovat (v grafickém softwaru) všechny ty tisícovky polygonů. Textura se navíc musí hýbat s nimi – jako ta gáza nalepená na drátovou loutku.

Možností, jak panákem hýbat, je více, ale nejčastěji se používá jeden trik: panákovi se přidělá *kostra* (obr. 3), ta se rozpohybuje a její pohyb se automaticky přenesou na vrcholy trojúhelníků. Protože kostra je z *kostí*, jichž jsou desítky, zatímco polygonů je o několik řádů více, vyjde to jednodušeji. Samotná kostra se rozhýbe (většinou) buď ručně nebo pomocí technologie *motion capture* (nepřekládáme do češtiny, české překlady se nepoužívají). Standardní motion capture funguje tak, že na živého herce navěsíme několik referenčních značek, typicky alespoň dvanáct, jejichž pozici v 3D prostoru budeme zaznamenávat pomocí speciálních kamer (často infračervených), a necháme herce provést několik pohybů – například chůzi, rozhlížení se, nadávání apod. Pomocí těchto záznamů potom rozhýbeme kostru.

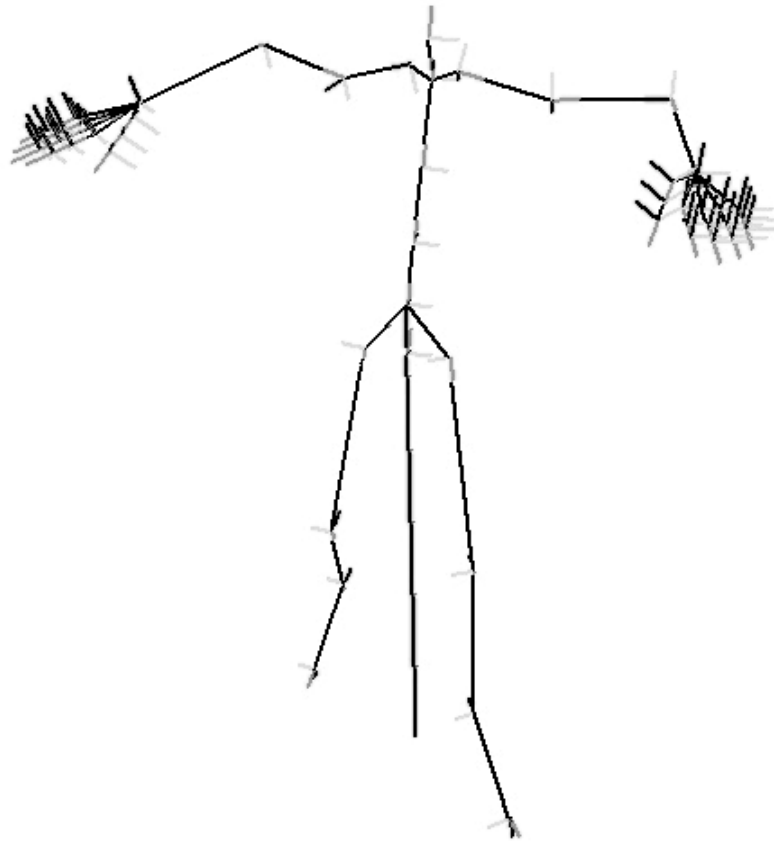


Obr. 2. Drátový model těla virtuálního agenta.

Aby se s kostrou hýbaly i trojúhelníky (a potažmo textura), musíme každému trojúhelníku, přesněji každému jeho vrcholu, přiřadit kost, k níž patří. Vrchol se pak bude hýbat s touto kostí. Zvedneme-li vřetení kost, algoritmus pro nás dopočítá, jak to bude vypadat, když se zvednou vrcholy trojúhelníků na předloktí. A protože s vřetení kostí se zvedne i „dlaňová“ kost, pohnou se i vrcholy dlaní a prstů.

U kloubů je to složitější – trojúhelníkům u kloubů musíme většinou přiřadit obě kosti. To proto, že pokud pohnete například s pažní kostí, měly by se posunout nejen trojúhelníky paže, ale trochu i trojúhelníky, které tvoří rameno a část zad.

Tomuto přiřazení se říká *rig* – zjednodušeně řečeno jde o analogii svalů, pokud budeme svaly chápat pouze jako pojivo mezi kostmi a kůží. Práce je to nad hlavu. Pokud uděláte chybu, při určitých pohybech panáka to může vypadat dost špatně (obr. 4). Ale jak jednu kostru, *rig* a drátový model máte, můžete vytvořit mnoho variací téže postavy jen výměnou textury.



Obr. 3. Kostra virtuálního agenta. Krátké úsečky u kloubů znázorňují orientaci kloubů.

Tělem virtuálního agenta tedy pohybujeme tak, že na kostře „přehráváme“ jednotlivé pohyby, jež jsme získali pomocí technologie motion capture nebo vytvořili ručně, a určitý algoritmus díky rigu přenáší pohyb kostry na drátový model a zase jiný algoritmus na rozpočítávanou trojúhelníky nanáší texturu. Jednotlivým pohybům se říká *animace*.

Z výše popsaného plynou dvě důležité věci. Zaprvé, aby na sebe animace navazovaly, musí začínat a končit v neutrální pozici, jinak při přechodu mezi dvěma animacemi uvidíme „skok“ (leďa bychom přišli s nějakým dalším trikem, jak se „skoku“ vyhnout – viz dále). Zadruhé, pokud si nepřipravíme nějakou animaci předem, máme většinou smůlu. Panák umí sebrat předmět ze stolu, ale zapoměli jsme na animaci sebrání předmětu ze země? Pozdě.

Tedy přesněji řečeno, mnoho lidí z oblasti počítačové grafiky pracuje na tom, aby ty dva body nebyly pravda a někdy se jim to i docela daří. Máme animace chůze a běhu a potřebovali bychom *automaticky* spočítat přechod mezi nimi? Dá se s tím něco udělat. Máme panáka, který nedýchá? (Když panák stojí a *vůbec* se nehýbe, vypadá to velmi nepřírozeně – člověk, který stojí na místě, není latexová figurína a ve skutečnosti se trochu pohybuje.) I s tím se dá něco dělat.



Obr. 4. Špatný „rig“ agenta. Všimněme si, že ramena přiléhají špatně k tělu agenta. V podpaží je tělo jakoby deformované. Toto je důsledek špatného nastavení rigu agenta.

Říká se tomu *procedurální* generování animací; myslí se tím to, že určitý program za nás sám rozpohybuje tělo agenta: dáme mu například na vstup dvě animace, které nekončí v neutrální póze, a on sám dopočítá přechod mezi nimi. Zlí jazykové tvrdí, že lidé vyvíjející počítačové hry ve skutečnosti netouží po lepších algoritmech umělé inteligence, nýbrž po lepších algoritmech pro procedurální generování animací. Chtějí magické tlačítko HUP, které bude jako ubrousek Prostří se: zmáčkneme ho a panáci budou tančit, jak pán poroučí! A desítky grafiků půjdou k vodě.

Nakonec se ještě aspoň stručně vraťme k animacím, které nepoužívají kostru. V některých případech, zejména při animaci obličejů (mimika a řeč) není přístup kostry a rigu vhodný, protože věrohodná animace by vyžadovala více kostí než celý zbytek postavy. Ve hrách se používají různé metody, buď tzv. *morph targety* nebo *deformační klece*. V obou případech je trik v tom, že se model hlavy rozdělí na několik částí (více či méně pečlivě, tím se právě ty metody liší) a tyto části se pak animují pomocí jednoduchých „globálních“ operací jako je roztážení, pootočení a podobně.

2.1 K dalšímu čtení a zamyšlení

V této kapitole blíže popíšeme některé pokročilé technické problémy, které při animování agentů mohou nastat, a způsoby jejich řešení. Podíváme se na to, co všechno se dá spočítat přímo za běhu programu a jaká jsou omezení takovýchto animací. Dále se podíváme na to, jak se dá simulovat např. pád ze schodů a jestli je možné spojit dvě animace do jedné. Čtenář, který spěchá, může tuto kapitolu přeskocit. Čtenář, který hledá souhrnný text o počítačové grafice, se může s důvěrou obrátit na knihu [59]. O animacích obecně se lze dočíst v [45].

Lineární algebra

Počítačová animace (a to nejen ve hrách) je hezkou aplikací základů lineární algebry, jinak jen středně populárního předmětu. Pokud se jí chcete zabývat jinak než jako uživatelé, uděláte dobře, když zbystríte, až se bude mluvit o „homogenních souřadnicích“, „kvaternionech“, o vektorech a maticích ani nemluvě.

Pro animace jsou zvláště důležité *kvaterniony* (jsou to vlastně zobecněná komplexní čísla ne s jednou, ne se dvěma, ale se třemi imaginárními složkami), které umožňují elegantně „sčítat“ rotace, základní stavební blok kosterní animace.

Matematika potřebná pro animace je pěkně popsána například v [3].

Procedurální animace

Uvažme následující příklad: Máme agenta, strážce určitého objektu, a chceme, aby sledoval pohledem kolemjdoucí postavy – tedy aby směrem k nim natáčel hlavu. Takovou animaci si nelze připravit předem, protože nevíme, jak se bude hráč v prostředí pohybovat. Můžeme ale použít tzv. procedurální animace, ty jsou vytvářené přímo za běhu našeho programu. Spočítat totiž, jak má náš agent natočit hlavu, není nijak těžké. Na druhou stranu jsou s tím spojeny určité problémy. Zatím si nijak nedáváme pozor na omezení pohyblivosti krku a očí (agent schopný otočit hlavou

kolem dokola by působil opravdu divně). Tato omezení je opět nutné ručně předem specifikovat. Více se o tomto problému dozvíme níže.

Napadají vás další situace, kde bychom upotřebili procedurální animace? Poznamenejme, že jednoduché procedurální animace, jako je otáčení hlavy, často nestačí. Pro vytvoření opravdu kvalitního dojmu ze scény je třeba je zkombinovat se složitějšími přístupy, z nichž některé jsou vysvětleny dále.

Fyzika

Představme si situaci, ve které chceme, aby náš agent spadl ze schodů. Aby to vypadalo věrohodně, měl by náš agent dynamicky interagovat s prostředím. Měl by se odrážet od zdí, nebo si třeba vykloubit ruku, která se mu zasekla v zábradlí, a podobně. Pokud bychom se drželi pouze standardních animací, tak bychom pravděpodobně měli jenom kácející se postavu, která by ignorovala svoje okolí: mohla by třeba skončit s rukou či nohou ve zdi nebo by jí visely nohy ve vzduchu nad schody.

V takovéto situaci musíme animaci počítat přímo za běhu programu. Co budeme všechno potřebovat? Bude nám stačit obyčejná kostra? O každém kloubu těla víme, že má určitou tuhost, která není pro všechny klouby stejná. To samé se týká třeba i pevnosti (po jejím překročení se kloub přelomí vejpůl) a samozřejmě nesmíme opomenout, že ne všechny klouby mají stejný tvar (strukturu), což ovlivňuje jejich způsob ohýbání (uvažme např. kyčelní a kolenní kloub). Tyto vlastnosti obyčejná kostra nepokrývá, takže o ně budeme muset kostru rozšířit. Tato data je opět nutné přidat ke kostře ručně.

Ve výsledku pak pomocí těchto rozšířených parametrů necháme prostředí působit na našeho agenta – docílíme tím poměrně realistického efektu – agent se skácí ze schodů. Někdy se navíc tento postup kombinuje se standardními animacemi. Výsledek je pak ještě o stupeň lepší, i když náročnější na výpočet. Více se o této problematice lze dozvědět v textech [31], [32].

Inverzní kinematika

Máme vůbec něco takového, co by se podobalo výše zmíněnému ubrousku? Něco, co by nám řeklo, jaké kvaterniony budeme potřebovat, například při onom sbírání věcí ze stolu a ze země?

Jednou možností je to vypočítat. Fyzikové se pohybem zabývají už docela dlouhou dobu. Určitě si mnozí z vás vzpomenou na fyzikální disciplínu zvanou „kinematika“. Pokud máme ocelové rameno s alespoň jedním kloubem, upevníme jeden jeho konec a pak za druhý zatáhneme, fyzik je schopen s velkou přesností říci, jak bude výsledný pohyb *celého* ramene vypadat. Představa, že panák obdobně natáhne k předmětu konec ruky (prsty a dlaň) a zbytek těla se automaticky přizpůsobí, je docela lákavá. Sice ten zbytek těla se v reálném světě na tomto pohybu podílí velmi aktivně svými svaly, ale potřebujeme to takto precizně? Vždyť chování jako ve výše zmíněném příkladu s ocelovým ramenem by nám úplně stačilo! A skutečně – tento přístup se v počítačové grafice používá pod názvem *inverzní kinematika* („inverzní“ proto, že je zbytek těla jenom tahán, a sám netlačí jako v reálném světě) a dává celkem dobré výsledky.

Neradujeme se ale předčasně, není to tak jednoduché, jak to může vypadat na první pohled. Za vším jsou celkem složité rovnice a navíc je zapotřebí ke kostem přidat i „klouby“ s danou (rozumnou) tuhostí a rozsahem pohybu, což je opět práce pro člověka, nikoli pro počítač. Navíc je důležité si uvědomit, že tyto rovnice nemusí mít jednoznačné řešení: když zatáhnete za konec ruky nebo ocelového ramene, klouby se mohou do výsledné pozice dostat více způsoby.

Ve hrách se takto řešená inverzní kinematika používá často u jeřábů nebo jiných strojů, kde nevadí (ba je přímo žádoucí) strojený a strojový vzhled těchto animací. O inverzní kinematice se píše ve většině základních učebnic o animacích, například [23], [45]. Doporučit lze také text [20].

Blendování animací

Typickým problémem, který se v rámci animování agentů řeší, je *blendování* (česky někdy spojování, mixování) animací. Obvykle animace fungují tak, že tělo agenta ovládá právě jedna animace, agent například buď jde, nebo běží. Ale co v případě, že agent přechází z chůze do běhu? Mohli bychom si připravit přechodovou animaci a použít ji po skončení probíhajícího cyklu chůze (přechodová animace by začínala v základní pozici chůze a končila v základní pozici běhu). To má tu nevýhodu – kromě toho, že je to práce navíc –, že to může dost dlouho trvat, což je problém, zvláště když se jedná o hráčem ovládanou postavu. Také můžeme problém ignorovat a prostě například z 10. snímku animace chůze přejít na 11. snímek animace běhu. To je řešení, které bylo dřív celkem běžné, ale tehdy byly také postavy hrubší a monitory měly menší rozlišení. Dnes je preferovaný způsob blendování, neboli prolínání, animací. Pro každou kost se vezmou hodnoty z obou animací a vhodným způsobem se zprůměrují (zde se opět s výhodou využijí zmíněné kvaterniony). Jedná se o vážený průměr, kde se váhy mění tak, že jejich součet je stále 1; na začátku je váha animace, do které přecházíme (v našem případě běhu), rovná 0, na konci je rovná nule váha výchozí animace (v našem případě chůze). Výsledkem je plynulý a většinou celkem věrohodný přechod mezi oběma animacemi.

Blendování animací se dá též využít k inverzní kinematice. V takovém případě se postupuje tak, že grafici-animátoři vytvoří několik verzí stejné animace, například krok 30 cm dlouhý a krok 60 cm dlouhý. Pokud pak budeme chtít udělat krok 40 cm dlouhý, zprůměrujeme obě animace s váhami 0,67 a 0,33.

Obdobný problém je skloubení dvou různých animací. Řekněme, že máme předpřipravenou animaci chůze a otočení hlavy. Mohli bychom je automaticky spojit? Animace chůze se stará o nohy a zbytek těla ji už tolik nezajímá. Animace otočení hlavy zase hýbe pouze hlavou. Kostí, které tyto animace rozhybávají, se tedy nepřekrývají a spojení animací je poměrně jednoduché: pustíme animaci chůze na nohy a animaci otočení hlavy na hlavu a máme agenta, co po nás pokukuje, když prochází kolem. Někdy jde obdobu tohoto postupu aplikovat i tehdy, když se některé kosti překrývají: použijeme na tyto kosti vhodně zvolený vážený průměr.

Více se o blendování animací lze dočíst například v [27], [19].

3 Kde bydlí virtuální agent?

Otázka má mnoho rovin, podobně jako otázka, z čeho virtuální agent sestává. My se zde ptáme na virtuální svět a myšlenkové konstrukty, které designéři používají k jeho návrhu.

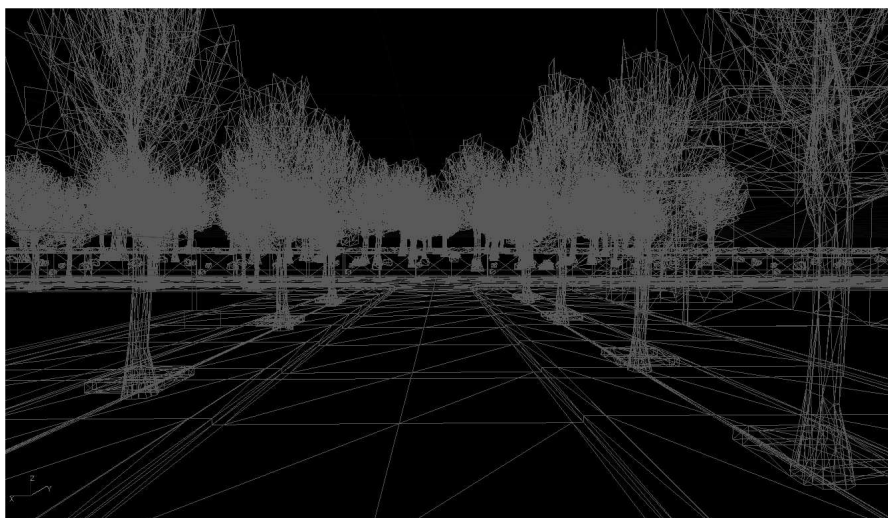
Virtuální svět (obr. 5) je tvořen podobně jako tělo panáků: z polygonů (obr. 6) potažených texturami. Drátovému modelu se často říká *geometrie světa*. O konkrétním zobrazení světa rozhoduje ale i mnoho „nehmotných“ prvků, například pozice světel či přítomnost mlhy. V dnešních počítačových hrách mohou virtuální agenti hru světel a stínů také vnímat, například ve hře Thief [35]. Někdy ovšem o těchto prvcích agent vůbec neví: tyto prvky jsou zde jenom z „okrasných“ důvodů: pro lidského diváka. To znamená, že když se schováte do příšeří, agent vás uvidí normálně.

Některé předměty mohou být součástí geometrie, tzn. nelze s nimi pohybovat; například stromy. Jiné předměty může jít přenášet; například batoh. Možná tušíte, že může být docela problém, aby virtuální agent *viditelně* nosil v ruce předměty. Když živý člověk drží předmět, jeho končetiny se kvůli tomu všelijak deformují: to znamená, že podobně bychom museli deformovat i kostru agenta. To je v lepším případě spousta práce, v horším to vůbec nejde: například agent může mít dlaňovou kost, ale už ne kosti pro jednotlivé prsty: a pak mu prsty jednoduše neohnete. V některých hrách, kde se často nosí předměty jenom jednoho typu (ano, například zbraně), se to řeší tak, že se kostra a drátový model vytvoří v pozici, kdy agent něco nosí. Pokud agent nic v ruce nemá, stejně vypadá, jako by něco držel. Jindy se zkrátka přenášený předmět nezobrazí a umístí se do agentova pomyslného batohu – *inventáře*.



Obr. 5. Ulice města. Na konci ulice vidíme stromy za zdí. Zeď svět ukončuje.

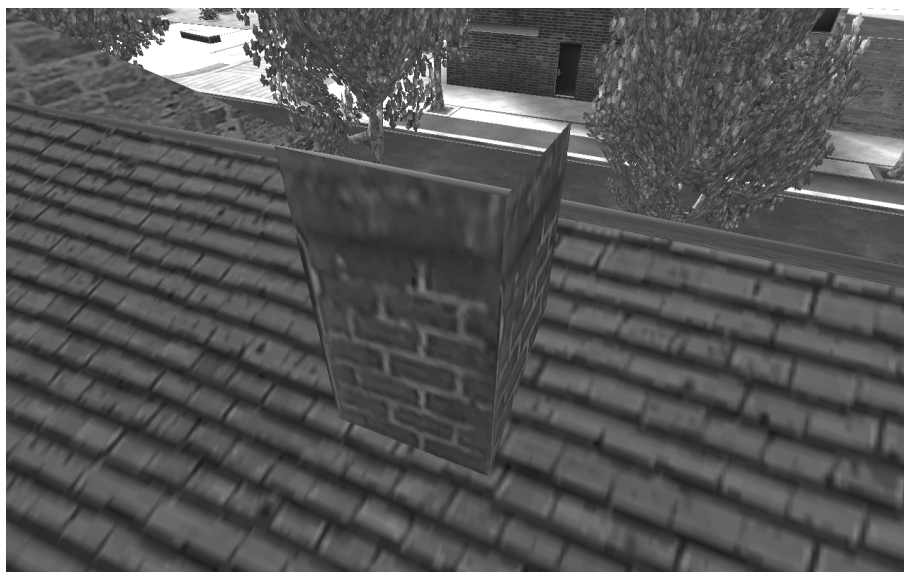
Jiným typickým problémem je, jak svět „ukončit“. V mnoha aplikacích končí svět skálou nebo stěnou, případně se „konec“ utápí v mlze. To, že virtuální svět je jako filmová kašírka, která má pouze vytvářet iluzi reality, se dobře projevuje v místech, na která byste se „normálně“ neměli dostat, například právě za koncem světa (obr. 7, 8).



Obr. 6. Stejná ulice jako na obrázku 5, tentokrát z drátového modelu.



Obr. 7. Stromy za zdí (z obr. 5) z pohledu zvenku. Všimněme si, že svět končí pár metrů za zdí a některé ze stromů se vznášejí ve vzduchu – klasický trik.



Obr. 8. Detail komínu shora. Všimněme si, že zadní část stěny komínu chybí. Komín není tvořen vlastním modelem, ale pouze texturami, které jsou z jedné strany průhledné. Při pohledu zdola toto není patrné.

3.1 Širší kontext

Pro úplnost dodejme dvě věci. Zprvce, mohlo by se zdát, že virtuální agenti obývají jenom světy počítačových her. Je sice pravda, že počítačové hry se stávají plnohodnotným médiem – podle ESA [21] například roční tržby herního průmyslu překonaly v USA tržby z kin, hráčům je v USA v průměru 35 let a 40% z nich jsou ženy – a není tudíž důvod, proč bychom se měli za počítačové hry stydět, nicméně virtuální agenti nalezneme i jinde. Snad nejzajímavějšími oblastmi použití jsou výukové aplikace, terapeutické aplikace a různé druhy simulací [6], [10].

Zadruhé, pojem *agent* se používá i pro entity jiného typu než *virtuální* bytosti. Obecně se požaduje, aby šlo o entitu v nějakém smyslu autonomní či inteligentní. Takováto entita musí být schopna se sama rozhodovat (nepotřebuje k tomu pokyn od člověka) a tato rozhodnutí musí dávat smysl (musí sledovat uživatelský cíl). O jemnosti definice agenta se vedou spory (viz např. [25], [58]) – není totiž příliš jasné, co to znamená „inteligentní“ a „autonomní“. Wooldridge [58] uvádí k zamýšlení příklad s topením a termostatem. „Hloupé“ topení, které můžete pouze zapnout a vypnout, nelze rozhodně nazvat autonomním – jeho stav řídí člověk. Jelikož takové topení nepodniká vlastní akce, nemůže se jevit ani jako inteligentní – je člověku pouze užitečné, dokáže mu vyhrát obydlí. Ale co když takové topení hřeje přespříliš? Člověk jej musí sám vypnout a až začne být zase zima, musí jej opět zapnout. S tím však může být potíž v šestipokojovém rodinném domě, kde udržovat člověku příjemnou teplotu 22 stupňů znamená neustále zapínat a vypínat různá „hloupá“ topení. Proto člověk vymyslel termostat. Z hlediska funkčnosti se můžeme na termostat dívat jako na

způsob, kterým člověk dokáže sdělit topení, jakou teplotu má ve svém okolí udržovat. Topení se tak stává „chytrým“, je schopné autonomně reagovat na změny teploty ve svém okolí a podle potřeby se samo zapnout nebo vypnout. Nasadíme-li si „agentí“ brýle, můžeme takovéto topení s termostatem vnímat jako autonomního agenta, jehož svět je tvořen jedinou veličinou (teplotou okolí). Člověk takovému agentu řekne, jakou teplotu má udržovat v domě, a agent již sám podniká potřebné akce k dosažení tohoto cíle. Když však brýle sundáme, vidíme místo agenta kus železa, co hřeje. Lze jej nazvat inteligentním? A o to se, zjednodušeně řečeno, právě vedou spory.

Do definice „být inteligentní“ nakonec často spadnou živé organismy, roboti² a mnoho programů – například prohlížečů webu. Pojem lze ovšem nejen upřesňovat, ale také zjemňovat a vytvářet všelijaké podskupiny, např. *softwarové agenty* nebo *inteligentní softwarové agenty* a pod. Konkrétně ti pyšníci se adjektivem „inteligentní“ tvoří samostatnou, poměrně rozsáhlou výzkumnou oblast (blíže viz [58]). Pro náš typ agentů je klíčové adjektivum „virtuální“. Odráží se v něm dvě věci: fakt, že mají virtuální tělo, a fakt, že bydlí ve virtuálním světě. Například světem mnoha inteligentních softwarových agentů je Internet a žádné tělo v našem smyslu nemají.

O vztahu virtuálních agentů k jiným agentům se více píše v [10], [7].

V kontextu virtuálních agentů lze o autonomii říci následující. Při návrhu a tvorbě virtuálního agenta, ať už mluvíme o jeho vizuální podobě nebo o jeho chování, je třeba mít na paměti, k jakému účelu ho konstruujeme. Většinou není naším cílem simulace reality ale napodobení reality, vytvoření virtuálního světa, který zprostředkuje divákovi *iluzi*, že se nachází ve funkčním a fungujícím světě. Toho je zpravidla jednodušší dosáhnout vhodným zjednodušením problému než jeho plnou simulací.

Vezměme si jako příklad hru, kde hráč může chodit nebo jezdit po městě, v zásadě libovolným způsobem (jako je např. GTA nebo Mafie). Bylo by možné simulovat automobilový provoz v celém městě, s tisíci nebo deseti tisíci agenty, kteří by měli svoje autonomní cíle a jejich interakcí při dosahování těchto cílů by emergentně vznikala automobilový provoz. Nebo je možné simulovat provoz jenom v okolí hráče a celý ho bude řídit jeden „loutkář“, pohybuující jednotlivými auty jako loutkami (a který bude mít nějaká pravidla, jak má hustota provozu a typ aut záviset na místě a denní době). Druhý způsob je zatím stále mnohem schůdnější a používanější. K tomuto problému se ještě vrátíme na konci kapitoly.

4 Virtuální agent chodí a ohmatává svět

Virtuální agent se může ve virtuálním světě pohybovat – od toho koneckonců celá simulace je. Z pozice programátora to vypadá tak, že agentovi přikáže přehrát animaci chůze, nebo jiného typu pohybu, z pozice $\langle x1, y1 \rangle$, kde agent právě stojí, na $\langle x2, y2 \rangle$, kde má stát po ukončení animace. Přitom ovšem programátor narazí na několik problémů:

² Další nejasností je, jestli by se měli chápat agenti a roboti jako životní nebo neživotní (tj. jsou to ty roboty a ty agenty nebo ti roboti a ti agenti?) Zde se kloníme k druhé variantě, protože je nám sympatičtější.

1. Jak daleko může být $\langle x_2, y_2 \rangle$ od $\langle x_1, y_1 \rangle$?
2. Co když je mezi $\langle x_1, y_1 \rangle$ a $\langle x_2, y_2 \rangle$ překážka?
3. Co když se mezi $\langle x_1, y_1 \rangle$ a $\langle x_2, y_2 \rangle$ objeví překážka poté, co animace začne?
4. Jak agent vlastně ví, kudy má jít?

Odpovíme postupně. V navazující kapitole se pak budeme ptát, proč by vlastně měl agent někam chodit: jaký je účel jeho pohybu?

1. Jak daleko může být $\langle x_2, y_2 \rangle$ od $\langle x_1, y_1 \rangle$?

Odpověď je jednoduchá: tak daleko, jak daleko „jde animace“. Pokud máme animaci dvojkroku, kterou jsme získali pomocí motion capture, a herec přitom ušel metr a půl, i virtuální panák, pokud je stejně veliký jako herec, by měl ve virtuálním světě ujít metr a půl. Animaci lze pustit i na delší nebo kratší vzdálenost, například „natáhnout“ ji na tři metry, ale panák potom bude nepřirozeně „klouzat“.

2. Co když je mezi $\langle x_1, y_1 \rangle$ a $\langle x_2, y_2 \rangle$ překážka?

Představme si, že náš agent hraje špatného úředníka, který z nějakého důvodu „chce“ opustit svou kancelář. Na pozici $\langle x_1, y_1 \rangle$ bude jeho stůl, na pozici $\langle x_2, y_2 \rangle$ budou dveře. Na první pohled triviální: natočíme panáka směrem ke dveřím a přehrajeme za sebou tolik dvojkroků, aby to tak akorát vyšlo až k východu (pokud nebude vzdálenost v metrech dělitelná 1.5, poslední dvojkrok nějak „zašudláme“). Ale co když bude stát v cestě stůl?

Řeší se to takto: představte si, že úředníkovi kouká z pupku několik provázků, za jeden z nich ho někdo, říkejme mu dveřník, tahá směrem ke dveřím. Jakmile si úředník všimne, že mu v cestě stojí stůl, hodí jeden ze zbývajících provázků kolegyni, která stojí bokem – v rohu u okna (obr. 9). Tahají teď oba: dveřník i kolegyně. Výsledek bude, že úředník půjde trochu bokem – mezi roh a dveře. Pokud si stolu všiml včas, podaří se mu ho bez úhony obejít. Jakmile bude na straně stolu, požádá kolegyni, aby přestala tahat a hodila mu provázek zpátky.

Proč ve chvíli, kdy ouřada obchází stůl, nepřestane úplně tahat dveřník? Kolegyně by přeci odtáhla úředníka na stranu stolu rychleji; dveřník jí to jenom „kazí“. Důvod je ten, že by úředník zatočil ve chvíli, kdy by jeho kolegyně začala tahat, příliš prudce; takhle normální lidé nechodí, ti většinou zatáčí plynule. Díky plynulosti to bude ve skutečnosti ještě trochu složitější: úřednice bude napřed tahat málo, potom bude postupně zabírat více a nakonec bude postupně přestávat.

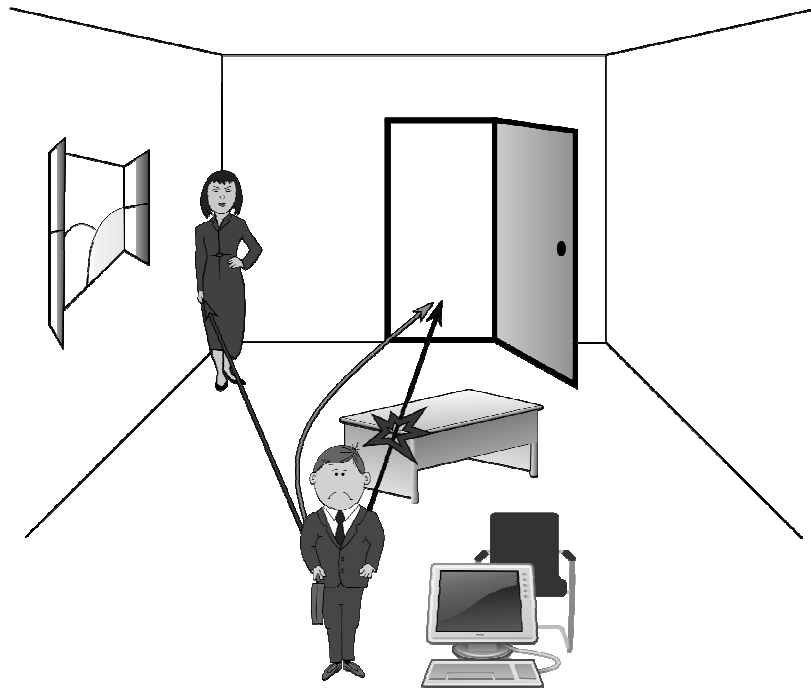
Formálně si ovšem představujeme úředníka (agenta) jako hmotný bod a provázky jako silové vektory. Úředník jde směrem daným složením těchto vektorů. Výpočetně je to velmi efektivní; v jedné počítačové hře takhle dokážete „tahat“ tisíce agentů. Trikem s provázky (silovými vektory) dokážeme udělat i složitější věci: aby se dva agenti navzájem vyhnuli, aby sledoval jeden druhého, aby se houfovali v davu, aby šli podél stěny a podobně. Těmto trikům se říká *steerovací techniky* či *pravidla* („steering rules“). Mimochodem, hodí se i na řízení (virtuálního) auta či houfu zvířat: třeba netopýrů z filmu Batman.

Ne vždy to ale vypadá dobře (obr. 10). Občas potřebujeme složitější mechanismus; vrátíme se k němu za chvíli.

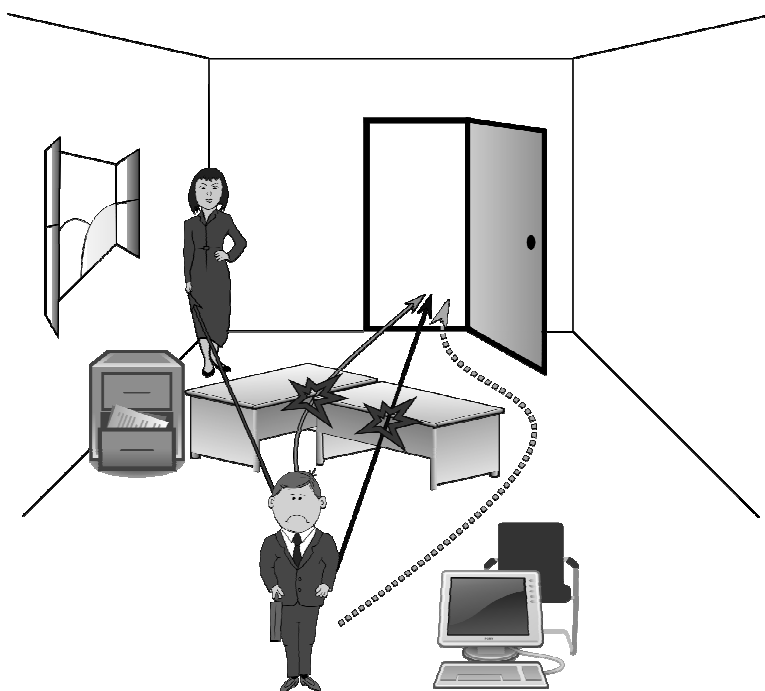
3. Co když se mezi $\langle x1, y1 \rangle$ a $\langle x2, y2 \rangle$ objeví překážka poté, co animace začne?

První část odpovědi je jednoduchá: pokud si agent překážky všimne, použije steerovací techniku pro vyhnutí se překážce (tj. použije „kolegyni“ vpravo či vlevo od překážky). Druhá část odpovědi je těžší: jak si agent překážky všimne? To nás přivádí k myšlence, že agent potřebuje nějakým způsobem vnímat svět.

Vzpomeňte si na pojem geometrie prostoru. Geometrii prostoru potáhneme texturou, uděláme její projekci do roviny a tu zobrazíme na monitoru (obr. 5). Vidí náš agent to, co člověk: 2D projekci geometrie potažené texturou? Rozhodně ne. Rozpoznávání toho, co je na 2D obrázcích (nebo videích) se věnuje oblast počítačového vidění. Nikdo na světě ale zatím neumí udělat „umělý zrak“, který by viděl tak dobře, jako vidí člověk: rozpoznával stovky různých objektů, v různém natočení a různě velikých, třeba i částečně zakrytých; dokázal poznat, kdo co dělá, a podobně. Musíme na to jít jinak. Klíčové je uvědomit si, že virtuální agent, který se má vyhýbat překážkám, toho zas tolik vidět nemusí: stačí mu vidět právě ty překážky.



Obr. 9. Úředník jde od počítače ke dveřím a přitom se vyhne stolu. Při obcházení stolu je dočasně „přitahován“ do rohu místnosti.



Obr. 10. Jednoduchý steering dovedl úředníka do pasti mezi uskupení objektů. Chytřejší by bylo vyhnout se celému uskupení najednou.

Jednoduchý trik je následující: vybavíme agenta dlouhými vousy, kterými bude ochmatávat terén před sebou, podobně jako slepec zkoumá prostor holí nebo potkan hmatovými vousy (obr. 11). Vousy informují agenta o tom, že narazil na překážku a v jakém je směru. Přesněji se tomu říká technika *vrhání paprsku* („ray-casting“; paprsek = vous). Kolik má takový agent vousů? Čím více, tím je větší šance, že detekuje překážku. Na druhou stranu každý vous navíc představuje výpočetní zátěž pro procesor. Takový agent z počítačové hry Unreal Tournament 2004 má vousy jen tři. Má je u kolen; do této výšky zasahuje většina překážek (obr. 12). Co když je překážka menší nebo je ve výšce očí? Inu, tak do ní agent vrazí... nebo projde skrz.

Kdyby tedy náš úředník chodil jako panák z Unreal Tournamentu, měl by být úplně slepý a mít tři vousy u kolen (trochu jako Děd Vševěd). Vidíme, že i mozek špatných úředníků toho zvládne víc než mechanismus panáků pro vyhýbání se překážkám.

4. Jak agent vlastně ví, kudy má jít?

I špatný úředník si po několika dnech v práci zapamatuje, že chce-li odejít z kanceláře, musí nejprve obejít stůl. A že obejít stůl znamená jít nejprve k oknu a pak až ke dveřím kanceláře. Tuto cestu si vybere ze dvou důvodů. Jednak je to nejkratší cesta ven z kanceláře a jednak na její trase nejsou žádné překážky.



Obr. 11. Hmatové fousy potkana.



Obr. 12. „Fousy“ virtuálního agenta. „Fous“ vlevo detekuje překážku.

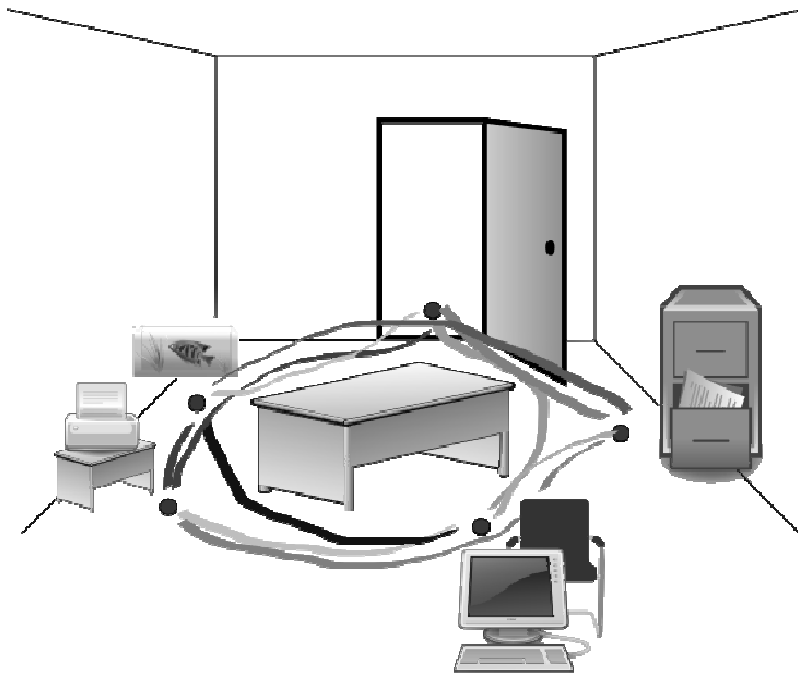
Pokud je náš špatný úředník svědomitý, může se stát, že jej povýší a přestěhuje se do velké kanceláře s velkým stolem, počítačem, tiskárnou, registraturou na šanony a akváriem. Zde již úředník potřebuje znát více cest než jen od své židle ke dveřím. Musí umět zajít si pro vytisknuté dokumenty, odtud je zaneš do registratury a čas od času nakrmit rybičky. V hlavě tedy nosí jakousi mapu kanceláře, do které má zaznačeno, jak přecházet mezi jednotlivými místy (obr. 13). Všimněme si, že pro samotný pohyb jsou důležitá pouze místa a cesty – z obrázku by klidně mohl zmizet stůl či podstavec s akváriem. K samotnému pohybu nepotřebujeme vědět, že obcházíme stůl nebo že se vyhýbáme podstavci s akváriem, důležitá je pro nás pouze informace cest, po kterých se můžeme bezpečně pohybovat. Jinými slovy náš úředník nepotřebuje vidět, stačí mu chodit po paměti (představte si, že jdete v noci na záchod a praskla žárovka).

Problém s mapou na obr. 13 je, že počet cest je kvadratický vzhledem k počtu vrcholů – musíme se umět dostat odkudkoli kamkoli (snad až na triviální cesty typu akvárium – akvárium). Pokud bychom chtěli do takového grafu zahrnout i kanceláře podřízených, neřku-li celou budovu úřadu, dostaneme se rychle do problémů. Určitě vás napadá jednoduchý trik. Bližším pohledem na cesty v kanceláři zjistíme, že cesta od židle úředníka k tiskárně je velice podobná cestě k akváriu, liší se jen dodatečným krokem k akváriu. Můžeme tedy zrušit cestu židle – akvárium a zapamatujeme si, že cesta od židle k akváriu vede přes tiskárnu. Podobným způsobem upravíme všechny cesty, které mají některé části společné (obr. 14). Vznikne nám tak graf, kterému se říká *navigační*, a jeho vrcholy se označují jako *navigační body* („way points“ nebo také „navigation points“ či zkráceně „navpoints“). Odpověď na otázku „kudy má agent jít?“ se tak mění na úkol nalezení cesty v grafu z výchozího místa do požadovaného cíle. Obecně se jedná o úlohu nalezení nejkratší cesty v grafu s kladným ohodnocením hran. Úlohu můžeme řešit například pomocí Dijkstrova algoritmu [18], A* [52] či Floyd-Warshallovým algoritmem [17].

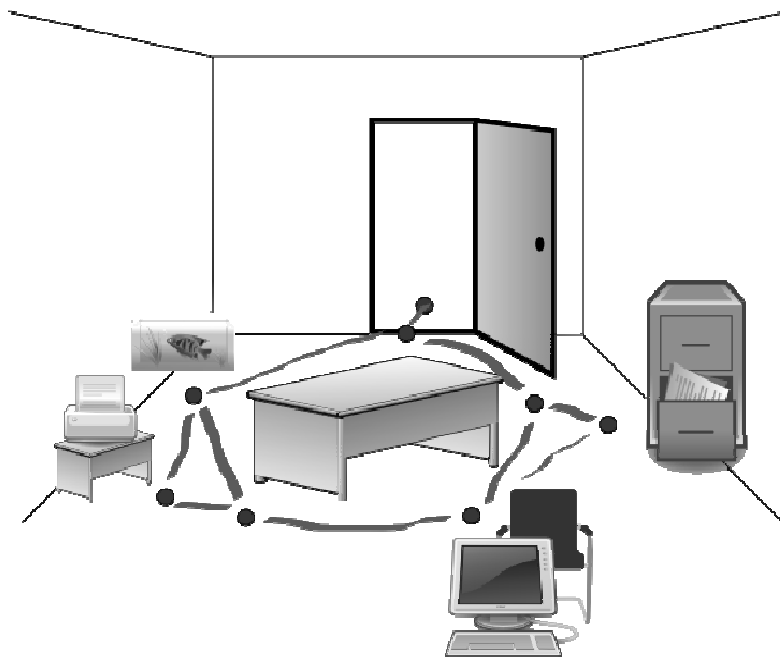
Mohlo by se zdát, že takovýto graf je vše, co náš virtuální agent potřebuje k bezpečnému pohybu po úřadě. Avšak opak je pravdou. Náš graf je totiž statický; je vytvořen s představou, že zdi kanceláře se nezmění přes noc, nábytkem nikdo nebude hýbat a po cestičkách se nebudou pohybovat jiní úředníci. Co když ale stejný graf využívá vícero úředníků a střetnou se na chodbě? Zde je ještě pomoc jednoduchá: zkombinujeme hledání cesty v grafu s navigačními pravidly popsány výše. Ale co když nějaký zlomyslník schválně šoupne v noci se stolem uprostřed kanceláře a přehradí cestu v grafu? Co když dojde k přestavbě úřadu a do kanceláře povedou nové dveře nebo staré budou zazděny? Ukazuje se, že náš špatný úředník bude opět chytřejší než virtuální agent. Úředník si jednoduše přestaví svou mapu v hlavě a přizpůsobí se novým podmínkám. Toto „přizpůsobení“ agentům již jde hůře, ale i je dokážeme naučit, aby svůj graf uměli revidovat (viz dále).

4.1 K dalšímu čtení a zamyšlení

Techniku steeringu důkladně rozpracoval už v 80. letech Craig Reynolds [49]. Existují samozřejmě i různá rozšíření, některé práce jsou publikované ve známé sérii knih o umělé inteligenci v počítačových hrách AI Game Programming Wisdom [48].

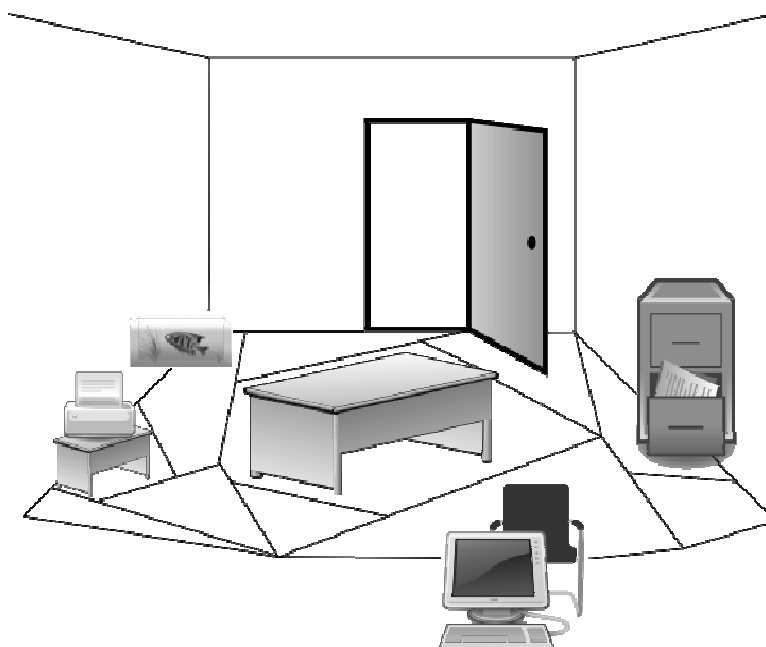


Obr. 13. Všechny možné cesty v úředníkově nové kanceláři.



Obr. 14. Navigační graf.

Způsobů, jak tvořit „mapu“ prostředí po agenty, je více. Pokud jste se zarazili nad přílišnou jednoduchostí navigačních grafů, zarazili jste se správně. Jednoduchý navigační graf popisuje prostředí velice hrubě – zkuste si představit, že nevnímáte nic víc než pár bezrozměrných čar spojující pár důležitých míst ve vašem bytě. Co budete dělat, když po stejné cestě již někdo půjde? Jak daleko se od ní můžete odchýlit, aniž byste do něčeho narazili? Jedním ze způsobů, jak tento problém vyřešit, je použití techniky *navigační mřížky* („navigation mesh“ nebo též „navmesh“) ([54]). Technika vychází z myšlenky využít známou geometrii virtuálního prostředí a automaticky ji transformovat na mapu složenou z trojúhelníků nebo obecně polygonů (obr. 15). Tyto trojúhelníky (polygony) potom použijeme podobně jako vrcholy navigačního grafu. Sousednost dvou trojúhelníků je analogická hraně grafu. Trojúhelníky ale navíc obsahují informaci o tom, kde se může úředník bezpečně pohybovat.



Obr. 15. Navigační mřížka.

S pohybem agentů se pojí také další problémy než jen reprezentace prostředí. Nové problémy přináší například velké virtuální světy, ve kterých jsou klasické algoritmy na hledání nejkratší cesty v grafu pomalé. Problémy způsobují výtahy, dveře či schody, při kterých se musí klást zvýšená opatrnost při spouštění animací, a v neposlední řadě problémy přináší také koordinace pohybu velkého počtu agentů či pohyb skupinek agentů ve formacích. Pokud prostředí začne být příliš velké, řeší se vyhledávání cesty v grafu pomocí hierarchického plánování cest [4]. Výtahy, dveře a schody lze řešit například speciálními navigačními body, ve kterých se animace synchronizují a pohyb velkého počtu agentů lze elegantně řešit již zmíněným steeringem.

Hledání cesty je problém, který musí vyřešit skoro každá hra a tak není divu, že na tento problém existují programy, zvané *middleware*, které mohou vývojáři integrovat do svého projektu. Příkladem *middlewareu*, který využívá *navmeshe* je třeba *PathEngine* [46] nebo *Kynapse* [1] (tento *middleware* toho ovšem umí o hodně víc, než jenom hledání cesty).

4.2 Následuj toho, koho vidíš!

Na závěr si popíšme chování agenta – *stopaře* –, který dokáže následovat jiného agenta. Takové chování se často hodí v počítačových hrách.

Vytvoření *stopaře* je poměrně jednoduché. Jediné informace z prostředí, jež bude potřebovat, jsou pozice ostatních agentů, které momentálně vidí. Máme-li tyto informace k dispozici, můžeme vytvořit *stopaře* následovně: V momentě, kdy se v jeho zorném poli objeví cizí agent, vyšleme příkaz „jdi na pozici tohoto agenta“ („jdi“ pomocí navigačních pravidel; tzn. pokud se například po cestě objeví překážka, obejdi ji). V případě, že je agentů ve *stopařově* zorném poli více, bude *stopař* následovat nejbližšího z nich. Jak se lze zbavit *stopaře*, pokud nás sleduje? Zprvů můžeme zmizet z jeho zorného pole. Zadruhé můžeme přiběhnout ke skupince agentů a doufat, že *stopař* začne sledovat někoho z nich. Nebo ho můžeme zkusit vlákat do pasti jako na obrázku 10.

Zbývá vyřešit jeden problém: co to znamená, že *stopař* někoho „vidí“? Obvykle to znamená, že se cizí agent (nebo objekt) nachází v jeho *zorném poli* („field of view“) a mezi *stopařem* a agentem není žádná neprůhledná překážka. Přítomnost překážky zjistíme již známou technikou vrhání paprsků: vrhneme paprsek z „očí“ agenta směrem k těžišti objektu: pokud paprsek narazí do překážky, řekneme, že objekt agent nevidí (dobrá, pokud je objekt zakrytý částečně, tak to nemusí fungovat, ale vrhat víc paprsků zase stojí výpočetní zdroje...). Šířka zorného pole agenta bývá obvykle zhruba 90°.

Existují i složitější mechanismy vnímání, ale používají se jen tam, kde je to opravdu nutné: například ve hře *Thief*, kde se agenti často pohybují v příšerí a kde jejich vjemy závisí na kdejakém šustnutí [35]. Výše jsme říkali, že agenti často nevidí zdroje světla, nevnímají stíny či mlhu. S vylepšenými mechanismy vnímání to nemusí být pravda. Na akademické půdě se také věnuje několik skupin modelování pozornosti (např. [33]). Jde o to, že když je kolem agenta hodně viditelných předmětů, neměl by si jich stihnout všimnout všech. Pomocí mechanismu pozornosti lze také modelovat oční pohyby: agenti vypadají věrohodněji, pokud se jejich oči natáčejí podle toho, na který objekt se dívají.

K zamyšlení: dokázali byste agenta *stopaře* vylepšit tak, aby se vymotal z pasti (obr. 10), pokud je do ní naveden? Nápověda: zkuste využít navigačního grafu.

5 Agentova kniha předpisů

Náš agent umí chodit, ale tím to zatím končí. Jak ho donutit k tomu, aby udělal něco složitějšího, třeba přenesl někam předmět? Pamatujete na knihu předpisů špatného úředníka? Je to kniha, do které úředník nahlédne, aby zjistil, jak má zareagovat, když něco chcete. Náš agent potřebuje také takovou knihu. Této knize někteří lidé

programující virtuální agenti říkají *reaktivní plán*, jiní *procedurální paměť*, ale to teď není příliš důležité.

V Knize mohou být různé magické formule. Snad nejjednoduššímu typu takových formulí se říká *reaktivní pravidla*. Jsou to pravidla jestliže – pak:

if podmínka **then** příkaz

V Knize jich je několik. Takový virtuální prodavač z počítačové hry, který nikdy neopouští svůj krámk, by jich mohl mít třeba pět:

1. **if** hori **then** utikej pryč
2. **if** utok **then** bran se
3. **if** hrac se pta **and** neni utocnik **then** odpovez
4. **if** hrac je blizko **then** predstirej praci
5. **else** nic nedelej

Už vidíte paralelu se špatným úředníkem?

Pravidla bereme shora dolů, pokud platí více podmínek („if“ část), vezmeme to vyšší. Důležité je, že pravidla kontrolujeme neustále (přesněji: několikrát za vteřinu). Jde o to, aby agent včas dokázal zareagovat na změnu kontextu (baví se s hráčem, začne hořet).

Co to přesně znamená „utíkej pryč“ nebo „braň se“? V zásadě máme dvě možnosti. Zaprvé, může to znamenat udělat něco, co už umíme: pustit nějakou animaci, sekvenci animací (to bude případ „braň se“) nebo někam poodejít (to bude případ „utíkej pryč“). Může to také znamenat, že během animace zmizí z virtuálního světa nějaký předmět (agent ho „sebral“) nebo se tam objeví (agent ho „položil“) nebo se změní *stav* nějakého objektu (agent během operace „braň se“ rozbil vázu na polici). O předmětech řekneme více za chvíli.

Zadruhé, pokud jde o něco opravdu složitějšího, může to znamenat, že máme nalistovat v Knize Předpisů další stránku, kde jsou další pravidla. Představme si, že pravidlo 4 bude příkazovat, aby úředník – pardon, virtuální prodavač – opravdu pracoval:

4. if hrac je blizko **then** opravdu pracuj (viz str. 42)

To ale znamená spoustu věcí! Prodávat zboží, objednávat zboží, dělat inventuru a podobně. Na straně 42 náš prodavač najde příslušná pravidla. Tomuto „listování“ se říká *hierarchický přístup*. Důležité je, že až si prodavač prohlédne, co je na straně 42, vrátí se zase na předchozí stranu – ne jako v „gameboocích“.

Nyní se dostáváme ke klíčové otázce: kdo Knihu Předpisů připraví? Knihu musí od začátku až do konce napsat programátor nebo (v kontextu počítačových her) herní designér. Od začátku až do konce. A pak ji musí ladit a ladit, aby agent nedělal něco jiného, než co programátor chce. Je to moře práce.

Nedá se Kniha nějak... vygenerovat sama? Jako že by se panák sám něco naučil?

Nedá. Přesněji: nikdo to neumí. Říká se tomu „million dollar question“. Pokud přijdete na to, jak Knihu generovat automaticky, možná nedostanete Nobelovu cenu, ale rozhodně nikdo z vaší rodiny až do třetího kolena nebude muset do smrti pracovat. Lidé, co vyvíjejí počítačové hry, vás zasypou penězi. Vlastně jde o obdobu magického tlačítka HUP. Vzpomínáte? Generovalo samo animace. Zde máme jeho bratříčka HOPLA, který generuje automaticky chování.

Uvažte: člověku to trvá desítky let, než se naučí svou Knihu Předpisů. Navíc není jasné, jestli lidé (vyjma špatných úředníků) opravdu nějaké Knihy Předpisů v hlavách nosí. Virtuálními agenty se lidé zabývají tak 20 let. Opravdu je realistické očekávat, že by za tak krátkou dobu někdo přišel na algoritmus, kterým naučit virtuálního panáka to, co dělá živý člověk?

5.1 Agent nosí předměty a něco si při tom pamatuje

Zůstaňme ještě chvíli u našeho virtuálního prodavače. Představme si, že prodavači právě dorazila zásilka zboží do skladu a on by ji rád přenesl do svého obchodu. Aby to prodavač zvládl, přidáme mu následující pravidla:

```
if mam_vec and jsem_v_obchode then poloz_vec
if mam_vec then jdi_do_obchodu
if jsem_ve_skladu and vidim_vec then vezmi_vec
else jdi_do_skladu
```

Tato sada pravidel zajistí, že prodavač přenese všechno zboží ze skladu do obchodu. Všimněme si jedné věci. Náš prodavač už nereaguje jenom na prostředí, ale také na svůj vnitřní stav – podmínka `mam_vec`. V tomto případě se jedná o jednoduchou formu paměti – prodavač si pamatuje, že něco nese. Jednoduchá forma paměti v našem příkladu se nazývá *krátkodobá paměť*. Zjednodušeně řečeno je to datová struktura, do které ukládáme vše, o čem si myslíme, že se nám v brzké budoucnosti hodí. Časem to z ní musíme také vymazat – v našem případě ve chvíli, kdy prodavač věc položí.

Kdysi někteří namítali, že paměť nepotřebujeme [11]. Skutečnost je ovšem taková, že pokud bychom krátkodobou paměť k dispozici neměli, Kniha Předpisů by se nám psala mnohem hůře. V příkladu s nošením bedny to zas tak dramatické není: agent bez paměti by se mohl jednoduše podívat, jestli v rukách něco drží nebo ne – jako paměť by mu v jistém smyslu posloužilo externí prostředí. Někdy to ale takhle jednoduše nejde: uvažme telefonování. Zvednete sluchátko a nic se neozývá: pokud si nepamatujete, že jste právě sluchátko zvedli a že hovor má právě začít, nemáte jak se jenom na základě stavu vnějšího světa rozhodnout, jestli říct „Haló, je tam někdo?“ (začátek hovoru) nebo sluchátko položit (konec hovoru). Jinými slovy v jednom stavu světa můžete udělat dvě různé akce a dokážete se rozhodnout jenom podle toho, co jste dělali předtím. Tato situace se nazývá „perceptual aliasing problem“.

Nabízí se také otázka, jestli by si agent neměl pamatovat něco dlouhodobě. Psychologové nechápou lidskou paměť jako monolitickou strukturu, ale jako sadu systémů [56]. Zjednodušeně řečeno: jedna složka systému je *dlouhodobá episodická paměť* – to je paměť na minulé zážitky („včera jsem byl na Karlštejně“). Jiná složka je *sémantická paměť* – to je paměť na faktické informace („Karlštejn je v Čechách.“). Další složka je *procedurální paměť* – paměť na motorické schopnosti. Obdobou dlouhodobé procedurální paměti je u virtuálních agentů *Knihy Předpisů*. Zatím se příliš nevytvářeli virtuální agenti, kteří by upotřebili sémantickou a episodickou paměť, proto výzkum v této oblasti teprve začíná. Není to tak jednoduché, jak to může na první pohled vypadat. Například episodická paměť nemůže fungovat jako prosté video. Tak jako si skuteční prodavači nepamatují všechno zboží, co jim kdy prošlo rukama, ani virtuální prodavač by neměl mít kompletní seznam všech beden zboží, co přenesl. Navíc by to stálo příliš mnoho výpočetních zdrojů. Jako úvodní čtení na toto téma lze doporučit například [9], [8].

5.2 K dalšímu čtení

Reaktivní pravidla jsou jenom jedním ze způsobů, jak psát do *Knihy Předpisů*. Mezi další způsoby patří například *konečné automaty* („finite state machines“) nebo *stromy chování* („behavioral trees“). Úvod k těmto a dalším technikám můžete nalézt například v [10], [13], [14]) nebo opět v knihách ze série *AI Game Programming Wisdom* [48]. Komplexní text, který uvádí čtenáře do mnoha technik používaných pro tvorbu agentů v počítačových hrách je [30]. Pojednává dokonce i o několika nesmělých pokusech týkajících se učení. Jiný článek pojednávající o učení agentů v počítačových hrách je [24]. Nejde ovšem o učení celé *Knihy Předpisů*, ale pouze o drobné modifikace (například: umím už jíst a naučím se, že kráva chutná lépe než kamení). Problému učení se věnuje také open source hra *NERO* [41]. Agenti z této hry jsou namísto *Knihy Předpisů* řízení umělou neuronovou sítí, pro jejíž učení se používají genetické algoritmy [55]. Pokud bylo agentovo chování uživatelem kladně ohodnoceno, budou jeho geny, kódující výsledné chování, s velkou pravděpodobností přeneseny i na další generaci agentů. Naopak geny špatně hodnocených agentů z populace vymizí. Může být zajímavé sledovat, jak se agenti postupně učí jednotlivá chování jako například chůzi podél stěn, výsledné chování ale zdaleka nedosahuje komplexnosti chování napsaného člověkem. Podobné tvrzení platí i o počítačové hře *Creatures*, která uvádí virtuální agenty, jejichž chování se rovněž částečně učí neuronovými sítěmi [28] (tentokrát bez použití genetických algoritmů).

Knihy Předpisů je ve skutečnosti často naprogramovaná přímo v programovacím jazyce, jímž je psaná počítačová hra (např. C++), nebo v „pomocném“, tzv. *skriptovacím* jazyce, který základní jazyk doplňuje (např. Lua). Speciálně pro rozhodovací pravidla ale také existuje několik programů, které umožňují takové *Knihy* psát a pak v nich vyhledávat – například reaktivní plánovač *POSH* [12], nástroj *Jess* [26] nebo nástroj *Soar* [57] (který toho ve skutečnosti umí mnohem víc).

Jak ví každý, kdo někdy potkal špatného úředníka, pomocí *Knihy Předpisů* se mnoho problémů vyřešit nedá. O omezení tohoto přístupu se píše v [5]. O alternativě se dočtete dále.

5.3 Umí agent také něco sám vyřešit?

Už víme, že Kniha Předpisů je návodem, jak má úředník reagovat na předem popsané situace. To ovšem nemusí stačit na to, aby se agent z našeho pohledu choval inteligentně. Představme si, že bychom chtěli, aby náš úředník vyřešil určitý problém. Pokud by měl pouze Knihu Předpisů a v té by řešení problému nenašel, nevěděl by, co má dělat a jen by stál a díval se před sebe, případně reagoval na jiné podněty z okolí. My bychom rádi, aby se úředník zamyslel a pokusil se *sám* najít řešení – potřebovali bychom chytrého úředníka.

Na takový úkol už Kniha Předpisů nestačí a je třeba naučit našeho úředníka přemýšlet. Nikdo neví, jak skutečný člověk myslí, ale existují techniky umělé inteligence, které některé aspekty myšlení alespoň částečně postihují – například plánování. Plánování je činnost, kterou se lidé zabývají každý den, ať už v malém (jak se ráno obléct) nebo ve velkém (jak postavit most). Samotné plánování se dá charakterizovat jako proces, jehož výsledkem je *plán* – posloupnost akcí (např. najdi tričko, obleč si tričko, najdi kalhoty, obleč si kalhoty), které je třeba provést, abychom se z určitého *počátečního stavu* (stojím v pokoji nahý) dostali do *cílového stavu* (stojím v pokoji oblečený).

Abyste mohl úředník plánovat, musí vědět, jaké akce může provést. Akci lze popsat seznamem jejích *předpokladů* a *efektů*. K provedení určité akce potřebujeme, aby její předpoklady byly splněné. Například, pokud si chceme obléct tričko, musíme ho nejprve držet v ruce. Držení trička představuje předpoklad pro akci oblékání. Kromě předpokladů musíme vědět, jaké budou mít akce dopad na svět kolem nás, potažmo na svět kolem našeho úředníka. Jestliže si obleču tričko, dopadem (efektem) bude, že tričko už nedržím v ruce (toto se nazývá *negativní efekt*, protože už není pravda, že držím tričko v ruce), ale mám ho oblečené na těle (tzv. *pozitivní efekt*, protože se stalo pravdou, že tričko je na těle).

Ovšem ani popis akcí našemu úředníkovi na vyřešení problému – naplánování postupu – nestačí. Nejprve potřebuje znát výchozí stav, ze kterého plánuje – čili poznat stav světa, ve kterém se úředník nachází před tím, než začne provádět plán. Zde s výhodou využijeme systém vnímání, který jsme popsali výše. Problémem ale může být, že agent nezná stav světa kompletně: nevidí (nebo by neměl vidět) za zdi. Plánovat tak můžeme jen s částečnou informací.

Jak by ale náš úředník mohl vytvořit plán, kdyby nevěděl, k čemu se má dopracovat? Potřebuje tedy znát i cíl svého úsilí – cílový stav světa, ve kterém se má nacházet poté, co svůj plán uskuteční.

Samotné plánování – neboli schopnost úředníka vytvořit posloupnost takových akcí, kterými se dostane z výchozího stavu do cílového stavu – představuje velmi složitý a vědecky zajímavý námět. Nebudeme zde rozebírat jednotlivé algoritmy, ale existují dva základní postupy, jak k plánování přistupovat – *dopředný* a *zpětný*. *Dopředný* („forward“) způsob se pokouší budovat plán od počátečního stavu ke koncovému. *Zpětný* („backward“) způsob se snaží vycházet z konečného stavu a postupně se dopracovat až k počátečnímu.

Proč se vlastně používají Knihy Předpisů a ne plánování? Protože s Knihami Předpisů je to jednodušší a hlavně rychlejší. Knihy představují jakýsi spolehlivý

automat, který dobře funguje pro často se opakující situace. Uvažte: když si ráno čistíte zuby, také „neplánujete“ každý pohyb kartáčkem. Praktické omezení klasického plánování je rovněž v tom, že jeho výsledkem je pevně daná sekvence akcí: pokud se svět změní předtím, než se dostanete do cíle (což by se například v akčních hrách mohlo stávat dost často), původně naplánované akce přestanou dávat smysl. Agent by musel plán změnit, a protože samotné plánování nějakou dobu trvá, mohlo by se stát, že by agent nedělal nic jiného, než plánoval a plánoval. Je třeba najít způsob, jak Knihy Předpisů s plánováním kombinovat – to je ovšem otázka pro budoucí výzkum.

O plánování se více můžete dočíst například v [34], [38] nebo souhrnné knize [29]. V kontextu počítačových her opět odkazujeme na sérii AI Game Programming Wisdom [48] nebo článek o hře F.E.A.R. [44].

6 Spolupráce agentů

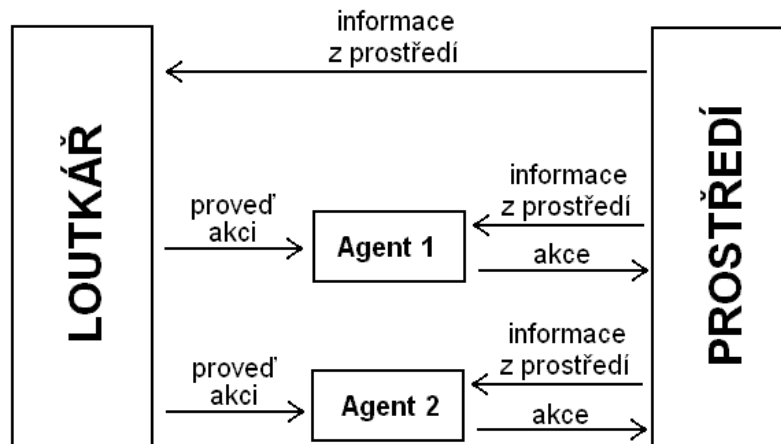
Představte si, že náš špatný úředník se pohádá se svou kolegyní o to, kdo má lepší sešívačku. V reálném světě vznikne taková žárlivá scéna sama od sebe (člověku se až dere na jazyk odborný termín *emergentně*). Jak je to ve virtuálním světě? Zde si musíme uvědomit jednu důležitou věc: ve většině aplikací nám nepůjde o to, aby se virtuální úředník s virtuální kolegyní opravdu hádali. Půjde o to, aby si to divák *myslel*. Ano, jako v loutkovém divadle chceme vytvořit *iluzi* hádky, chceme diváka *podvést*. V oblasti virtuálních agentů se v této souvislosti hovoří o *uvěřitelnosti* agentů („believability“) [36].

Za tím účelem použijeme spíše než poznatků z umělé inteligence triků z oblasti kreslených filmů. Takovým trikem například je, že by divák měl pochopit motivaci postav. Vědec studující umělou inteligenci nebo inteligentní softwarové agenty by mohl trvat na tom, že oba agenti by měli být autonomní. To ovšem trochu připomíná situaci, kdy loutkové divadlo hraje dva loutkáři naráz a nemají předem daný scénář. Jak jsme již naznačili výše, z technického hlediska bývá často jednodušší, když agenti pomyslně spojí síly, „domluví se“ na tom, jak má celá scénka vypadat, a pak ji přehrají. Jak vypadá ono „domluvení se“? Jsou dvě možnosti. Za prvé, buď si tajně vymění informace, aniž by o tom divák věděl („Hele, já se s tebou teď jako pohádám o to, kdo má lepší sešívačku.“ – „O kterou? Já mám dvě.“ – „O tu červenou.“ – „A kdo vyhraje?“ – „Já.“ – „Ne, já.“ – „Ok, tak ty. A v půlce hádky po tobě hodím jablko.“ – „Ok, a já uhnu a jablko rozbije okno.“). Často ovšem může být jednodušší druhý způsob: oba agenti se odevzdají do rukou centrálního *loutkáře* („drama manager“, „story director“), který je chvíli bude řídit oba (obr. 16).

Důležité je ovšem rozlišovat, jestli je scénka interaktivní nebo ne. Může vstoupit divák do děje a začít uklidňovat? Bez interakce je to jednoduché: v podstatě generujeme scénku z animovaného filmu. S interakcí ovšem musíme vymyslet spoustu reaktivních pravidel, které panákům říkají, jak mají zareagovat na tu kterou situaci, a běda, jak na nějakou situaci zapomeneme!

Centrální způsob řízení se ovšem nehodí ve chvílích, kdy nám opravdu záleží na tom, aby agenti jednali samostatně. Uvažte simulaci chování davu při krizových situacích: máme například (virtuální) stadion s tisícovkou lidí, náhle začne hořet, lidé začnou v panice utíkat, a my se ptáme, jestli jsou východy ze stadionu postaveny dost

dobře na to, aby všichni lidé ve zdraví utekli (podobný problém se řeší např. v [53]). Zde je samozřejmě důležité, aby každý agent jednal sám za sebe; nejde nám totiž o vytvoření iluze prchajícího davu, ale o model reálné situace. A v reálném světě se lidé většinou nedomlouvají přes centrálního arbitra o tom, jak utéct z hořící budovy.



Obr. 16. Architektura programu s centrálním „loutkářem“.

6.1 K dalšímu čtení

V akademické komunitě se problém spolupráce virtuálních agentů na přehrávání scének řeší zejména v oblasti disciplíny, která se jmenuje *virtuální vypravěčství* („virtual storytelling“). Zatím však, pokud můžeme soudit, k žádnému významnému pokroku nedošlo – je zkrátka potřeba správně napsat Knihu Předpisů. Někteří se snaží průběh scénky naplánovat. K dalšímu čtení lze doporučit články [15], [37], [51].

Z jiného úhlu se řeší problém spolupráce virtuálních agentů v týmových akčních hrách: zde je o to, jak řídit několik agentů, aby dosáhli společného cíle (např. získat vlajku). Agenti se také mohou domlouvat, ale většinou v rámci zachování věrohodnosti bez centrálního arbitra. Narozdíl od řízení jednoho agenta zde musíme dbát na to, aby si agenti vzájemně nepřekáželi, spolupracovali (např. krytí) a podobně. V kontextu počítačových her se tomuto problému věnují články [40], [42], [43], [50].

7 Závěr

Dospěli jsme na konec našeho příběhu. Co si z něj odnést? Především dvě věci. Zaprvé, pokud se chceme věnovat studiu virtuálních agentů, musíme si uvědomit, že jde o oblast z pomezí umělé inteligence a počítačové grafiky. Znalosti z těchto disciplín musíme vhodně skloubit. Zadruhé, chování virtuálního agenta musí někdo naprogramovat: nepadne samo z nebe. Hlavní technikou, která se dnes používá a ještě

dlouho bude používat je kniha Předpisů (nebo její analogie), doplněná o navigační pravidla a mechanismy pro hledání cesty v grafu. Pokročilejší techniky umělé inteligence, jako například neuronové sítě nebo plánování, se zatím příliš nepoužívají, a to zejména díky komplexnosti a dynamice virtuálních světů. Obecně platí, že tyto techniky jsou vyzkoušeny jen v malých a statických světech (výjimky samozřejmě existují).

Koho zajímá, jak bude příběh virtuálních agentů pokračovat, necht' sleduje vhodné konference a fóra. Zejména lze doporučit konference Intelligent Virtual Agents, Artificial Intelligence for Interactive Digital Entertainment, Computational Intelligence and Games, Computer Animation and Social Agents, Autonomous Agents and Multiagent Systems a Interactive Digital Storytelling. Zajímavé fórum je aigamedev.com a relevantní žurnál IEEE Transactions on Computational Intelligence and AI in Games. Kdo by si chtěl vyzkoušet sám programovat virtuální agenty, může začít s projektem Pogamut [16].

Pro úplnost dodejme, že některým problémům jsme se zde nevěnovali, například problematice sociálního chování nebo modelování emocí. Některé z těchto témat jsou rozvinuty v [2], [10]. Další komplexní úvodní texty k problematice virtuálních agentů jsou [7], [47].

Poděkování: Tato kapitola vznikla za podpory projektu CZ.2.17/3.1.00/31162, který je financován Evropským Sociálním Fondem, rozpočtem České Republiky a rozpočtem hl. m. Prahy. Náš vlastní výzkum týkající se virtuálních agentů je podpořen projektem P103/10/1287 (GA ČR), výzkumným záměrem MSM0021620838 MŠMT ČR, grantem 201/09/H057 a GAUK 21809. Autorem obrázků 1-8 je Zbyněk Krulich, jemuž patří velký dík.

Literatura

- [1] Autodesk: Kynapse project. URL: <http://www.autodesk.com/kynapse> [29.1.2010]
- [2] Bída, M., Kadlec, R., Brom, C.: Význam emocí pro umělé bytosti. In: *Mysel, inteligencia a život*. Vydavatelství STU, Bratislava (2007) 158-172.
- [3] Bourgh, D.M.: *Physics for Game Developers*. O'Reilly Media, 2001.
- [4] Botea, A., Müller, M. and Schaeffer, J.: Near optimal hierarchical path-finding. *Journal of Game Development* 1 (2004) 7-28.
- [5] Brom, C.: Hierarchical Reactive Planning: Where is its limit? In (Bryson J. J., Prescott T. J., Seth A. K., eds): *Modelling Natural Action Selection: Proceedings of an International Workshop Edinburgh, Scotland, 2005*, pp. 235 – 242.
- [6] Brom, C.: Virtuální učení. In: *Vesmír* 87/10 (2008) 706-711.
- [7] Brom, C.: *Řízení virtuálních lidí ve velkých virtuálních světech*. Disertační práce. MFF UK, 2007.
- [8] Brom, C., Lukavský, J.: Towards Virtual Characters with a Full Episodic Memory II: The Episodic Memory Strikes Back. In: *Proceedings of Empathic Agents*, AAMAS workshop, 2009, pp. 1 – 9.

-
- [9] Brom, C., Pešková, K., Lukavský, J.: What does your actor remember. Towards characters with a full episodic memory. In: *Proceedings of 4th ICVS*, LNCS 4871. Berlin, Springer-Verlag, 2007, pp. 89 – 101.
- [10] Brom, C., Šisler, V., Hoffmann, M.: Virtuální lidé. In (Mařík V., Stěpánková O., Lažanský J., eds): *Umělá inteligence 5*. Academia, Praha, 2007.
- [11] Brooks, R.A.: Intelligence without representation. In: *Artificial Intelligence* 47, (1991) 139-159.
- [12] Bryson, J.J.: Intelligence by design: Principles of Modularity and Coordination for Engineering Complex Adaptive Agent. PhD Thesis, MIT, Department of EECS, Cambridge, MA, 2001.
- [13] Bryson, J.J.: Action Selection and Individuation in Agent Based Modelling. In (Sallach D. L., Macal Ch., eds): *The Proceedings of Agent 2003: Challenges of Social Simulation*, 2003.
- [14] Bryson, J.J.: How to Make a Monkey Do Something Smart. URL: <http://www.cs.bath.ac.uk/~jjb/web/how-to-monkey.pdf> [29.1.2010]
- [15] Cavazza, M., Charles, F., Mead, S.J.: Character-based Interactive Storytelling. In: *IEEE Intelligent Systems* 17(4) (2002) 17-24.
- [16] Gemrot, J., Kadlec, R., Bida, M., Burkert, O., Pibil, R., Havlicek, J., Zemcak, L., Simlovic, J., Vansa, R., Stolba, M., Plch, T., Brom C.: Pogamut 3 Can Assist Developers in Building AI (Not Only) for Their Videogame Agents. In: *Agents for Games and Simulations*, LNCS 5920, Springer, 2009, pp. 1 – 15. URL: <http://artemis.ms.mff.cuni.cz/pogamut/tiki-index.php> [29.1.2010]
- [17] Cormen, T.H., Leiserson, C.E., Rivest, R.L.: *Introduction to Algorithms* (1st ed.). MIT Press and McGraw-Hill, 1990, pp. 558 – 565.
- [18] Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C. Section 24.3: Dijkstra's algorithm. In: *Introduction to Algorithms* (Second ed.). MIT Press and McGraw-Hill, 2001, pp. 595 – 601.
- [19] Edsall, J.: Animation Blending: Achieving Inverse Kinematics and More. Gamasutra Online, July 4, 2003. URL: http://www.gamasutra.com/features/20030704/edsall_01.shtml [29.1.2010]
- [20] Elias, H.: Inverse Kinematics – Improved Methods. Online. 2004. URL: http://freespace.virgin.net/hugo.elias/models/m_ik2.htm [29.1.2010]
- [21] The Entertainment Software Association (ESA). URL: <http://www.theesa.com/> [29.1.2010]
- [22] Epic Games: UnrealTournament 2004. URL: <http://www.unrealtournament.com> [29.1.2010]
- [23] Erleben, K., Sporryng, J., Henriksen, K., Dohlmann, H.: *Physics Based Animation*. Charles River Media, 2005.
- [24] Evans, R.: Varieties of Learning. In: *AI Game Programming Wisdom I* (Rabin S., ed.), Charles River Media, Inc., Hingham, Massachusetts, 2002, pp. 567 – 579.

-
- [25] Franklin, S., Graesser, A.: Is it an agent, or just a program?: A taxonomy for autonomous agents. In: *Intelligent Agents III Agent Theories, Architectures, and Languages*, LNAI Volume 1193, Springer, Berlin, 1997, pp. 21 – 35.
- [26] Friedman-Hill, E.: *Jess in action: java rule-based systems*. Manning Publications, 2003.
- [27] GAME ANiM Articles: Basics: Animation Blending. In: *GAME ANiM Online*. June 19, 2005, URL: <http://www.gameanim.com/2005/06/19/blending-the-future-of-non-linear-animation/> [29.1.2010]
- [28] Grand, S., Cliff, D., Malhotra, A.: Creatures: Artificial life autonomous software-agents for home entertainment. In (Johnson W. L., ed.): *Proceedings of the First International Conference on Autonomous Agents*, ACM press, 1997, pp. 22 – 29.
- [29] Ghallab, M., Nau, D., Traverso P.: *Automated Planning: Theory and Practice*. Morgan Kaufmann Publishers, San Francisco, Ca, USA, 2004.
- [30] Champandard, A.J.: *AI Game Development: Synthetic Creatures with learning and Reactive Behaviors*. USA, New Riders, 2003.
- [31] Jakobsen, T.: Advanced Character Physics. In: *Proceedings of Game Developer Conference (GDC'01)*, San Diego, USA, 2001. URL: <http://www.gpgstudy.com/gpgiki/GDC%202001%3A%20Advanced%20Character%20Physics> [29.1.2010]
- [32] Kennedy, R.: Contact Physics. Gamasutra Online. January 21, 2003. URL: http://www.gamasutra.com/view/feature/2903/contact_physics.php [29.1.2010]
- [33] Kim, Y., Hill, R.W. Jr., Traum, D.R.: A Computational Model of Dynamic Perceptual Attention for Virtual Humans. In: *Conference on Behavior Representation in Modeling and Simulation (BRIMS)*. Universal City, CA, 2005.
- [34] Lažanský, J.: Plánování. In: *Umělá inteligence I*. Academia, 2000, 184-216.
- [35] Leonard, T.: Building an AI Sensory System: Examining The Design of Thief: The Dark Project. In: *Gamasutra Online*, March 7, 2003.
- [36] Loyall, B.A.: Believable Agents: Building Interactive Personalities. Ph.D. diss. Carnegie Mellon University, 1997.
- [37] Magerko, B.: Intelligent Story Direction in the Interactive Drama Architecture. In: *AI Game Programming Wisdom 3*, Charles River Media, 2006.
- [38] Mařík, V.: Řešení úloh a využívání znalostí. In: *Umělá inteligence I*. Academia, 1993, pp. 33 – 66.
- [39] Mulley, G., Bittarelli, M.: Ragdoll Physics. Essay. Rensselaer Polytechnic Institute, Computer Science Department, 2007. URL: http://www.cs.rpi.edu/~cutler/classes/advancedgraphics/S07/final_projects/mulley_bittarelli.pdf [29.1.2010]
- [40] Munoz-Avila, H., Hoang, H.: Coordinating Teams of Bots withing Hierarchical Task Network Planning. In (Steve Rabin, ed.): *AI Game Programming Wisdom 3*, Charles River Media, 2006, pp. 417 – 427.
- [41] NERO homepage. URL: <http://code.google.com/p/opennero/> [26.1.2010]

-
- [42] Orkin, J.: Simple Techniques for Coordinated Behavior. In (Steve Rabin, ed.): *AI Game Programming Wisdom 2*, Charles River Media, 2004, pp. 199 – 206.
- [43] Orkin, J.: Applying Goal-Oriented Action Planning to Games. In (Steve Rabin, ed.): *AI Game Programming Wisdom 2*, Charles River Media, 2004, pp. 217 – 227.
- [44] Orkin J.: 3 States & a Plan: The AI of F.E.A.R. In: *Proceedings of Game Developer's Conference*, 2006.
- [45] Parent, R.: *Computer Animation, Second Edition: Algorithms and Techniques*. Morgan Kaufmann, 2007.
- [46] PathEngine homepage. URL: <http://www.pathengine.com/> [29.1.2010]
- [47] Prendinger, H., Ishizuka, M.: *Life-like characters*. Springer, Berlin, 2004.
- [48] Rabin, S. (ed.): *AI Game Programming Wisdom 1-4*. Charles River Media, 2002-2008.
- [49] Reynolds, C.: A web page on agents and steering behaviour. URL: <http://www.red3d.com/cwr/> [29.1.2010]
- [50] Reynolds, J.: Team Member AI in an FPS. In (Steve Rabin, ed.): *AI Game Programming Wisdom 2*, Charles River Media, 2004, pp. 207 – 215.
- [51] Riedl, M.O., Young, R.M.: An Intent-Driven Planner for Multi-Agent Story Generation. In: *Proc. Third ACM Joint Conference on Autonomous Agents and Multi-Agent Systems*, New York, USA, 2004, pp. 186 – 193.
- [52] Russell, S.J., Norvig, P.: A*Search. In: *Artificial Intelligence: A Modern Approach, 2nd Ed.* Upper Saddle River, N.J.: Prentice Hall, 2003, pp. 97 – 104.
- [53] Shia, J., Renb, A., Chenb, Ch.: Agent-based evacuation model of large public buildings under fire conditions. In: *Automation in Construction* 18(3) (2009) 338-347.
- [54] Snook, G.: Simplified 3D Movement and Pathfinding Using Navigation Meshees. In (Mark DeLoura, ed.): *Game Programming Gems*, Charles River Media, 2000, pp. 288 – 304.
- [55] Stanley, K.O., Bryant, B.D., Miikkulainen R.: Real-time Neuroevolution in the NERO Video Game. In: *IEEE Transactions on Evolutionary Computation*, 9(6) (2005) 653-668.
- [56] Sternberg, R.J.: *Kognivní psychologie*. Portál, 2002.
- [57] University of Michigan: SOAR. URL: <http://sitemaker.umich.edu/soar/home> [24. 4. 2007]
- [58] Wooldridge, M.: *An Introduction to MultiAgent Systems*. John Wiley & Sons, 2002.
- [59] Žára, J., Beneš, B., Sochor, J., Felkel, P.: *Moderní počítačová grafika*. 2. vydání. Computer Press, 2005.

Sémantický web: Niektoré aktuálne výzvy

Juraj FRANK, Vladimír DZIUBAN a Martin HOMOLA¹

Abstrakt. Vízia sémantického webu, ktorú predstavil v roku 2001 v časopise Science Tim Bernes-Lee, opisuje inteligentný web budúcnosti, na ktorom budú informácie s dobre zadaným významom, a budú tak zmysluplné nielen pre ľudí, ale budú otvorené aj pre efektívne strojové spracovanie. Takmer desať rokov výskumu v oblasti SW nás výrazne posunulo k naplneniu tejto vízie, aj keď nemožno tvrdiť, že by došlo k masovému presadeniu sémantických technológií. Predstavíme úlohu ontológií a ontologické reprezentatívne jazyky RDF a OWL. Pomocou týchto technológií je možné už dnes vytvárať aplikácie blízke vízii SW, spomenieme niektoré existujúce a sústredíme sa na aktuálne výzvy v tejto problematike.

1 Úvod

Víziu sémantického webu predstavili svetu v roku 2001 Berners-Lee a kol. [6]. Základnou motiváciou pre sémantický web je obrovské množstvo informácií nachádzajúcich sa na webe, a fakt, že usporiadanie týchto informácií je usporobené predovšetkým pre ľudských používateľov webu, na čítanie, nie sú však publikované vo formáte, ktorý by bol dobre prístupný pre strojové spracovanie. Toto konštatovanie bolo pravdivé v roku 2001, keď vízia sémantického webu vznikla, ale je rovnako pravdivé aj dnes. Hoci v súčasnosti už disponujeme celou rodinou sémantických technológií, s ktorými sa zoznámime v tejto kapitole, väčšina informácií publikovaných na webe je stále vo formáte HTML alebo XHTML dokumentov, v ktorých je gro informácií zakódovaných v texte napísanom prirodzeným jazykom.

Takto reprezentované znalosti sú veľmi náročné na akékoľvek strojové spracovanie. Pokiaľ by sme chceli lepšie strojovo uchopiť význam, zakódovaný v súčasných webových dokumentoch, narážame na všetky problémy týkajúce sa spracovania prirodzeného jazyka. Tieto sú spojené predovšetkým s obrovskou komplexnosťou prirodzených jazykov, ale tiež s výraznou vágnosťou a viacznačnosťou bežných formulácií, s ktorými si ľudský čitateľ vie poradiť bez toho, aby si to uvedomoval, ale pri strojovom spracovaní sú prekážkou.

Vízia sémantického webu opisuje web budúcnosti, na ktorom informácie majú dobre podložený a jednoznačne reprezentovaný význam, zachytený v jazyku oveľa

¹ Fakulta matematiky, fyziky a informatiky, Univerzita Komenského v Bratislave, E-mail: frank@fmph.uniba.sk, dziuba@fmph.uniba.sk, homola@fmph.uniba.sk

prístupnejšom pre automatické spracovávanie. Takýto web budeme vedieť využívať novými, predtým nepredstaviteľnými spôsobmi, za pomoci nových automatizovaných nástrojov pracujúcich so znalosťami, nazývaných často agenty sémantického webu. Tieto agenty v budúcnosti dokážu do veľkej miery samostatne nachádzať zdroje informácií a tieto spracovávať, prevezmú za nás mnohé rutinné úlohy a uľahčia nám tak prácu.

Rozoberme si teraz jednotlivé aspekty sémantického webu podrobnejšie:

- **reprezentačný jazyk:** od prirodzeného jazyka musíme prejsť k jednoznačnému reprezentačnému jazyku s logickou sémantikou. Výskum v oblasti reprezentácie znalostí ponúka nepreberné množstvo takýchto jazykov [32]. Pri výbere vhodného jazyka však musíme prihliadať na niektoré špecifiká webu a jeho distribuovanú podstatu. Nemožno napríklad prijať predpoklad uzavretého sveta, na ktorom často riešenia v oblasti reprezentácie znalostí stavajú. Tiež je nutné jednotlivé reprezentované entity nejakým spôsobom pomenúvať a referencovať. Už v roku 2001 boli za tri základné reprezentačné technológie vybrané XML, ktorá prináša jednotnú syntax, RDF s logickou sémantikou a URI zabezpečujúca menný priestor. Cieľom reprezentačných jazykov nie je nahradiť, ale doplniť, teda anotovať tradičné dáta určené pre ľudského čitateľa.
- **agenty sémantického webu:** informácie zakódované v prístupných reprezentačných jazykoch budú môcť využívať a spracovávať agenty sémantického webu. Pod týmto súhrnným názvom chápeme jednak budúce inteligentné webové klienty, ktoré používatelia prevádzkujú na osobných počítačoch, a ktoré, podobne ako dnešné webové prehliadače, budú pre nich akousi „bránou do sveta sémantického webu“. Zároveň sem však patria aj rôzne inteligentné služby, prevádzkované na webových serveroch, s ktorými osobné agenty sémantického webu budú komunikovať. Pre túto druhú skupinu sa ujal názov webové služby. Agenty sémantického webu budú schopné medzi sebou komunikovať, teda vymieňať si informácie. Komunikáciu možno veľmi jednoducho realizovať na báze dnešných webových technológií. Keďže informácie, s ktorými agenty pracujú budú mať logickú sémantiku, agenty budú môcť s využitím automatického usudzovania nad týmito informáciami vykonávať aj mnohé inteligentné funkcie a aplikácie.
- **ontológia:** ani fakt, že informácie budú reprezentované v jazyku s logickou sémantikou nezabezpečí, že agenty si budú schopné „porozumieť“ a samostatne spolupracovať bez toho, aby táto spolupráca bola nejakým spôsobom predprogramovaná. Obojstranné uchopenie významu dvoma autonómnymi agentami zabezpečia ontológie. Ontológia je „terminologická databáza“ v ktorej sú reprezentované koncepty, ktoré môžu byť predmetom komunikácie a vzťahy medzi nimi. Koncepty v ontológii možno referencovať pri manipulácii so znalosťami a pri komunikácii. Pokiaľ dva rôzne, nezávislé agenty referencujú tú istú ontológiu, dokážu samostatne rozpoznať, aký je význam poznatkov komunikovaných druhou stranou.

Pre lepšiu názornú ilustráciu predpokladanej sily sémantického webu si požičiame príklad z pôvodného opisu tejto vízie [6]. Predstavme si situáciu, že vyhľadáme lekára, ktorý nám pri našej osobnej návšteve v ambulancii predpíše rehabilitačnú terapiu. Následne je potrebné vyhľadať a vybrať si rehabilitačné pracovisko, ktoré poskytuje všetky potrebné služby pokrývajúce predpísanú terapiu, ktoré nám vyhovuje (t.j. napríklad nachádza sa vo vhodnej vzdialenosti od nášho bydliska, či pracoviska a pod.) a naplánovať a dohodnúť s vybraným rehabilitačným pracoviskom potrebný počet návštev, ich čas a dĺžku trvania. Túto úlohu budeme môcť vďaka budúcemu sémantickému webu plne zveriť nášmu osobnému agentovi, ktorý ju vyplní v súčinnosti s inými agentami, ktoré za týmto účelom prevádzkuje ambulancia nášho lekára, či rôzni iní poskytovatelia zdravotnej starostlivosti, podobne ako dnes prevádzkujú webové stránky.

V prvom kroku prevezme náš osobný agent podrobný opis terapie od lekárovho agenta. Následne pomocou vyhľadávacích služieb sémantického webu vyhľadá všetkých relevantných poskytovateľov zdravotnej starostlivosti pre požadovanú terapiu. Môže to urobiť vďaka tomu, že lekárov agent poskytol podrobný opis úkonov terapie anotovaný vo vhodnom reprezentačnom jazyku. Za pomoci ontológií dostupných na sémantickom webe dokáže náš osobný agent zistiť, či daný poskytovateľ poskytuje všetky potrebné úkony, ktoré predpísal lekár. V ďalšom kroku náš osobný agent vyfiltruje pracoviská vyhovujúce naším preferenciám. Spojí sa s agentom (webovou službou) našej zdravotnej poisťovne a zistí, ktorých poskytovateľov poisťovňa pokrýva. Ďalej porovná vzdialenosť poskytovateľa od miesta nášho bydliska a pracoviska. Potrebné informácie o našich preferenciách má buď predkonfigurované, alebo si vyžiada vstupy od používateľa. Následne sa spojí s agentom (webovou službou) vybraných poskytovateľov a porovná zoznam voľných termínov s našim pracovným kalendárom, ktorý tiež spravuje, a navrhne niekoľko možných plánov návštev v niekoľkých rôznych pracoviskách. Opäť je to možné vďaka dostupnej ontológii pokrývajúcej doménu plánovania úloh a stretnutí, ktorú oba komunikujúce agenty referencujú. Môžeme predpokladať, že agent následne plány predloží svojmu majiteľovi, ktorý si z nich vyberie, alebo zmení niektoré preferencie a nechá agenta plán preplánovať.

Takéto ambiciózne aplikácie budúceho sémantického webu sú aj po takmer desaťročí intenzívneho výskumu stále víziou budúcnosti. Napriek tomu sú plne plauzibilné a stále predstavujú výzvu ďalšiemu vedeckému bádaniu v tejto oblasti. V nasledujúcej kapitole ponúkame prehľad niekoľkých najdôležitejších technológií, ktoré v rámci bádania v oblasti vznikli. Sústredíme sa predovšetkým na reprezentačné jazyky RDF, RDF Schema a OWL, ktoré sú základným stavebným kameňom sémantického webu, pretože umožňujú sémantické anotácie webových dát v jazyku s logickou sémantikou (RDF) a umožňujú tiež tvorbu, webové publikovanie a zdieľanie ontológií (OWL), ktoré sú kritickým prvkom sémantického webu, zabezpečujúcim sémantickú interoperabilitu agentov. V druhej časti kapitoly sa budeme zaoberať najaktuálnejšími problémami a výzvami, ktoré sémantický web prináša. Výskum distribuovaných ontológií má výnimočné postavenie z hľadiska možností prepájania ontológií rozličných doménových oblastí, spomenieme si špecifiká spracovávania distribuovaných ontologických štruktúr a spôsoby, akými nad

nimi možno usudzovať. Budeme hovoriť aj o tom, ako aktuálne technológie Webu 2.0 ovplyvňujú získavanie, spracovávanie a poskytovanie sémantických informácií a ako môžu tieto technológie efektívne spolupracovať na skvalitňovaní sémantického webu. Nakoniec si v krátkosti predstavíme možnosti aplikácie sémantických technológií v desktopovom prostredí vo forme sémantického desktopu a ukážeme si, ako môže takéto riešenie fungovať v praktických podmienkach.

2 Technológie sémantického webu

2.1 URI

Jedným z cieľov webu pri jeho vzniku bolo vytvoriť globálnu komunitu, v ktorej môže ktokoľvek zdieľať informácie s hocikým iným. Na dosiahnutie tohto cieľa sa na webe používa jednotný identifikačný mechanizmus URI alebo Uniform Resource Identifier, teda jednotný identifikátor zdrojov. URI je teda objekt, najčastejšie v podobe textového reťazca, ktorý identifikuje nejaký zdroj. Zdrojom môže byť čokoľvek, obvykle hovoríme o zdrojoch ako o webovských stránkach, obrázkoch, dokumentoch, no vo všeobecnosti môže ísť aj o ľubovoľný objekt z reálneho sveta (kniha či hrnec).

Príklad URI:

```
http://www.example.com/education/universities.html?where=england#ratings
```

URI sa skladá z viacerých častí. Na začiatku reťazca až po dvojbodku je špecifikovaná tzv. URI schéma, v našom príklade `http`. Každá URI schéma má definovaný systém pridelovania identifikátorov a ich asociácie so zdrojmi. Za dvojčiarou lomkou nasleduje názov autority, v tomto prípade `www.example.com`. Tento názov väčšinou prideluje lokálna registračná autorita vo forme domény, ďalšie časti URI sú už pod kontrolou správcov daného servera. Cesta `/education/universities.html` špecifikuje umiestnenie zdroja v rámci autority a jej komponenty sú zvyčajne organizované hierarchicky, hoci to nie je pravidlom. Požiadavka `?where=england` spolu s cestou slúži pre bližšiu identifikáciu zdroja v rámci autority. Časť `#reviews` sa nazýva fragmentový identifikátor a označuje zdroj viazaný na iný, primárny zdroj a jeho využitie sa rôzni podľa aplikácie.

V prípade získavania HTML dokumentov identifikuje fragmentový identifikátor element, ktorého atribút `id` má hodnotu identifikátora. V našom príklade teda môže dokument `universities.html` obsahovať rôzne informácie o univerzitách v Anglicku, napríklad ich prehľad, delenie podľa zamerania či lokality, spôsob prijímania študentov, hodnotenia. Každéj súčasti tohto dokumentu možno priradiť atribút `id`, pomocou ktorého ich neskôr môžeme v URI priamo referencovať. Ak chceme napríklad získať hodnotenia univerzít označené v HTML dokumente atribútom `id="ratings"`, musíme použiť fragmentový identifikátor `#ratings`. V prípade RDF identifikuje fragmentový identifikátor nie časť dokumentu, ale prakticky ľubovoľnú vec, abstraktnú alebo konkrétnu, živú alebo neživú, ktorej

sémantika je deklarovaná v tomto dokumente. Dôležitou vlastnosťou URI je to, že umožňuje každému používateľovi webu vytvárať neobmedzený počet mien pre najrôznejšie zdroje, objekty, súbory a dokumenty. Tým sa tvorí základný adresný priestor webu a teda aj budúceho sémantického webu.

2.2 XML

XML, teda eXtensible Markup Language je značkovací jazyk, ktorý bol po prvý krát definovaný a zverejnený konzorciom W3C v roku 1998. Tento jazyk umožňuje anotovať textové dáta značkami nazývanými tagy. Ide o doplnkové informácie, ktoré môžu zohrávať významnú úlohu v budovaní sémanticky obohatených štruktúr. Jeho najdôležitejšou vlastnosťou je jednoduchá rozšíriteľnosť a možnosť vytvárania vlastných značiek, čo nám umožňuje definovať syntax akéhokoľvek reprezentačného jazyka. Práve táto kľúčová vlastnosť je príčinou jeho mimoriadneho rozšírenia v širokom spektre aplikácií. Jazyk XML sa preukázal ako mimoriadne vhodný pre použitie na webe, preto aj všetky technológie sémantického webu stavajú práve na XML.

Formát XML bol navrhnutý tak, aby bol čitateľný nielen strojovo, ale aj ľuďmi, z toho vyplývajú viaceré nároky, ktoré sú naň kladené. XML dokument obsahuje elementy, a to buď párové, ktoré majú začiatočnú aj koncovú značku, alebo nepárové, ktoré sú tvorené iba jednou značkou. Element môže ďalej obsahovať atribúty, ktoré však nezasahujú do pôvodnej textovej informácie, ale dopĺňajú a rozširujú ju. XML dokumenty je možné reprezentovať prostredníctvom stromovej štruktúry. Z toho dôvodu je nutné, aby boli XML dokumenty správne formované tzv. well-formed, teda každý začatý element musí byť aj ukončený a to aj v prípade, že ide o prázdny element. Zároveň sa elementy nesmú prekrývať. Well-formed dokumenty uľahčujú spracovávanie dokumentov nielen ľudskými, ale aj strojovými metódami.

Pri strojovom spracovaní XML je vhodné vopred poznať štruktúru dokumentov, s ktorými sa chystáme pracovať. Práve na to slúži definícia štruktúry dokumentu prostredníctvom tzv. DTD alebo Document Type Definition. Pomocou DTD možno presne definovať, ktoré elementy a atribúty sú v danom dokumente prípustné, ako ich možno používať a aký obsah môžu mať. Dokumenty spĺňajúce určité DTD nazývame anglickým slovom valid, to znamená platné. Na účely ich overovania existuje množstvo nástrojov tzv. validátorov, ktoré kontrolujú platnosť XML voči nejakému DTD. Vhodne štruktúrované XML dokumenty teda možno spracovávať pomocou rôznych nástrojov, pretože týmto nástrojom okrem nášho dokumentu poskytneme aj akúsi definíciu gramatiky, ktorou je náš dokument písaný. Pre popis štruktúry XML dokumentu sa dá použiť aj alternatívny popis pomocou XML Schema. Významná je najmä možnosť definovania dátových typov elementov a ich atribútov, môžeme napríklad obmedziť nejaký atribút spôsobom, že bude do neho možné zapisovať len dátum. Ďalej je možné definovať štandardné hodnoty elementov a atribútov, poradie a počty elementov, alebo to či môžu byť elementy prázdne alebo musia obsahovať text. XML dokumenty, ktoré majú špecifikovanú štruktúru pomocou XML Schema, možno podobne ako pri DTD validovať a overiť tak korektnú štruktúru dokumentu.

XML DOM, teda Document Object Model predstavuje univerzálny spôsob, ktorým je možné pracovať s XML dokumentmi a ktorý reprezentuje vnútorné aspekty týchto dokumentov, teda elementy a ich atribúty vo forme objektov a ich vlastností stromovej štruktúry. Vďaka tomu je možné s týmito objektmi pracovať akoby šlo natívne objekty príslušného prostredia s väzbou na DOM. XML dokument je pre DOM použitie potrebné načítať pomocou tzv. parsera, pričom bežné webovské prehliadače obsahujú zabudovaný XML DOM parser, aby umožnili skriptovacím jazykom prístup k elementom a atribútom v dokumente vo forme objektov.

Príklad jednoduchého XML dokumentu:

```
<?xml version="1.0"?>
<!DOCTYPE car SYSTEM "car.dtd">
<car yearofproduction="2009">
  <engine volume="2200" cylinders="8">
    <make>Ferrari</make>
  </engine>
  <body metallic="yes" color="red" />
  <equipment>
    <radio>Ferrari Generic Car Radio</radio>
    <airconditioning>Ferrari Standard AC</airconditioning>
    <firstaidkit>EU compliant FAK</firstaidkit>
    <tires>Summer tires for sport driving</tires>
    <tires>Winter tires for driving on snow</tires>
  </equipment>
  <price>5850 credits</price>
</car>
```

XML dokumenty sú uložené obvykle niekde na súborovom systéme počítača, resp. servera, no ak chceme pracovať s väčším množstvom dát kódovaných, je potrebné uložiť ich do databázy. Keďže v praxi sa čoraz viac vyskytovali konverzie dát z databáz do XML dokumentov a naopak, začali vznikať databázy schopné ukladať dáta priamo vo formáte XML, pri ktorých odpadla nutnosť konverzie. V praxi môže ísť o dva druhy softvérových riešení: buď ide o bežne používané relačné databázové systémy s podporou na ukladanie a čítanie XML dokumentov, alebo ide o tzv. natívne XML databázy (NXD), ktorých základnou úložnou jednotkou je XML dokument, podobne ako pri bežných relačných databázach je základnou jednotkou riadok tabuľky. Viaceré XML dokumenty uložené v NXD potom možno zoskupiť do kolekcí. Databáza pritom môže naraz obhospodarovať viacero dokumentov vo viacerých kolekciami.

Prístup k dátam v XML databáze je možný pomocou jazyka XPath, pomocou ktorého možno vytvárať jednoduché požiadavky na dokumenty alebo kolekcie dokumentov. Pre selekciu nejakého elementu z XML databázy zapíšeme v jazyku XPath jednoducho len cestu v XML strome po jeho uzloch. Rozšírené možnosti požiadaviek na XML databázu poskytuje jazyk XQuery, ktorý vďaka svojej syntaxi FLWOR, teda príkazom 'for', 'let', 'where', 'order' a 'return' v kombinácii s XPath ako

selekčným mechanizmom poskytuje jednoduché možnosti na úpravu a transformáciu dát priamo v požiadavke na databázu.

Dôraz je kladený aj na medzinárodnú použiteľnosť, často sa preto využíva univerzálne kódovanie znakov Unicode, ktorého miera používania sa zvýšila aj vďaka širšiemu využívaniu formátu XML. Štandardy založené na XML sa okrem webu a sémantických technológií rozšírili aj do rôznych oblastí praxe, či už ide o bankovníctvo, priemysel, uplatnili sa v kancelárskom použití -- vypovedajú o tom štandardné formáty dokumentov ako sú XHTML, Open Document Format alebo Office Open XML.

2.3 RDF

Resource Description Framework (RDF) je prvou z rodiny sémantických technológií z dielne W3C. Ako názov napovedá, ide o jazyk určený na zapisovanie jednoduchých tvrdení o rôznych zdrojoch. Zdrojmi sa chápu predovšetkým rôzne webové zdroje, ale v prenesenom význame môže byť v RDF zdrojom ľubovoľný objekt, o ktorom chceme zaznačiť nejaké údaje (viď nižšie). Hlavným prínosom RDF je predovšetkým presne stanovená logická sémantika týchto tvrdení. Jednou z motivácií pre vznik RDF bol fakt, že kým XML, ktoré bolo predstavené už skôr, prinieslo na web jednotnú syntax, o význame zaznamenaných údajov nenesie žiadnu informáciu. Uvažujme príklad jednoduchého XML dokumentu popisujúceho autá:

```
<?xml version="1.0"?>
<car id="BA716ER...
<make>Volkswagen</make>
<model>Lupo</model>
<color>red</color>
</car>
```

Kým človek, majúci základné znalosti angličtiny, ktorý si súbor prečíta hneď pochopí, že sa popisuje auto, pre softvérového agenta, ktorý nemá žiadnu ďalšiu informáciu o tomto dokumente, nemajú tieto dáta väčšiu hodnotu ako XML dokument:

```
<?xml version="1.0"?>
<wqr fg="BA716ER...
<fghj>Volkswagen</fghj>
<rFXws>Lupo</rFXws>
<DERii>red</DERii>
</wqr>
```

Samotný jazyk XML význam jednotlivých elementov a ich vzťah nijako nepopisuje. Cieľom sémantického webu je zaviesť do praxe také reprezentačné formáty, ktoré by umožnili aspoň základnú interpretáciu zaznamenaných dát. Jazyk RDF nás v tejto snahe posúva vpred. Umožňuje nám zapisovať jednoduché výroky tvaru:

Subjekt Predikát Objekt

Takéto výroky reprezentujú jednoduché asertívne priradenie: Predikátom sa Subjektu priraduje Objekt. Teda v predikátovej logike prvého rádu by sme mohli toto priradenie vyjadriť formulou Predikát(Subjekt,Objekt). Pokúsme sa teraz, aj keď zatiaľ nepresne, popísať tieto dáta z príkladu vyššie pomocou RDF, pričom využijeme výrokový tvar:

```
BA716ER type car .
BA716ER make Volkswagen .
BA716ER model Lupo .
BA716ER color red .
```

RDF stavia na dvoch predchádzajúcich webových štandardoch: XML a URI. Písali sme síce, že RDF vzniklo predovšetkým kvôli prekonaniu problémov s nedostatočnou sémantikou XML zápisov, XML je však využité ako základná kánonická syntax pre RDF. Výhodou takejto syntaxe je, že sa dá veľmi ľahko spracovávať pomocou štandardných nástrojov pre prácu s XML, ktoré sme si popísali vyššie. Nevýhodou je, že je zle čitateľná pre ľudí, preto vznikli rôzne iné syntaxe, s niektorými z nich sa zoznámime nižšie, je však dôležité upozorniť, že všetky sú ekvivalentné. URI slúži v jazyku RDF na identifikáciu zdrojov. Voľba použiť URI vychádza z myšlienky, že tieto identifikátory možno použiť na identifikáciu ľubovoľných vecí, nie len webových dokumentov, ako sme sa už dozvedeli vyššie. Zdrojom rozumieme čokoľvek, čo je možné pomocou tohto štandardu jednoznačne identifikovať. Môžu to teda byť webové dokumenty, obrázky a iné zdroje dostupné online, no zároveň môže ísť aj o reálny objekt, či osobu, ktorú chceme popísať, resp. o ktorej chceme vyjadriť nejaké tvrdenie.

Ako sme si už povedali, vzťahy medzi entitami RDF modelu môžu byť vyjadrené pomocou rôznej syntaxe. Uvažujme príklad, že by sme chceli zapísať tvrdenie "Auto má červenú farbu." Základným kameňom RDF dokumentu je triplet, teda trojica tvorená subjektom (Car), predikátom (hasColor) a objektom (Red), pričom v RDF dokumente sa môže nachádzať ľubovoľný počet tripletov. V jazyku RDF by sme teda takéto a podobné tvrdenia zapísali nasledovne:

```
ex:Car ex:hasColor ex:Red .
ex:Car ex:yearProduced "2009" .
```

Na zápis tohto tvrdenia sme použili tzv. Turtle syntax, ktorá je ľahko čitateľná pri bežnom čítaní. Pri strojovom spracovaní sa však obvykle používa RDF/XML syntax:

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:ex="http://www.example.com/terms/">
  <rdf:Description
rdf:about="http://example.com/terms/objects/Car">
    <ex:hasColor
rdf:resource="http://example.com/terms/colors/Red"/>
    <ex:yearProduced>2009</ex:yearProduced>
  </rdf:Description>
</rdf:RDF>
```

Z príkladu je jasné, že subjekt, predikát aj objekt v našej ukážke identifikujeme pomocou URI. V praxi to teda znamená, že môžeme vytvárať tvrdenia o ľubovoľnom zdroji, ktorému je možné priradiť URI. Ďalšou možnosťou, ako vyjadriť takéto tvrdenie je pomocou grafu. Subjekt a objekt budú tvoriť vrcholy grafu, predikát bude reprezentovaný jednosmernou hranou smerujúcou zo subjektu smerom k objektu (obr.1).



Obr. 1. Vyjadrenie jednoduchého RDF tripletu pomocou grafu.

Samotné RDF obsahuje niekoľko preddefinovaných kľúčových slov, ktoré sú v sémantike fixne interpretované. Súborne ich nazývame tiež RDF slovník z angl. RDF vocabulary. Sú to:

```

rdf:type rdf:Property rdf:XMLLiteral rdf:nil rdf:List
rdf:Statement rdf:subject rdf:predicate rdf:object rdf:first
rdf:rest rdf:Seq rdf:Bag rdf:Alt rdf:_1 rdf:_2 ... rdf:value
  
```

Vysvetlime si teraz význam niektorých týchto kľúčových slov. Slovo `rdf:type` je inštanciou vlastnosti a pomocou konštruktu `Volkswagen rdf:type Auto` vyjadrujeme tvrdenie, že Volkswagen je inštanciou triedy Auto, inými slovami, že Volkswagen je typ auta. Každá vlastnosť musí byť typu `rdf:Property`. Výraz `rdf:List` sa používa pre tvorbu jednoduchých RDF kolekcí v podobe štruktúry podobnej spájanému zoznamu. V RDF však existuje viacero rôznych typov kolekcí, `rdf:Bag` sa používa pre neusporiadané kolekcie, `rdf:Seq` sa používa pre usporiadané kolekcie a `rdf:Alt` pre kolekcie, kde obvykle vyberáme na spracovanie len jeden prvok z kolekcie.

Sémantika RDF vychádza z modelovej sémantiky používanej v predikátovej logike, či v iných logikách. RDF sa však do značnej miery vymyká predikátovej logike prvého rádu, ide o formalizmus vyšších rádo. Každý objekt môže byť interpretovaný ako zdroj (t.j. v jazyku predikátovej logiky individuál), ale zároveň aj ako vlastnosť (t.j. predikát). Toto nám umožňuje vytvárať tvrdenia aj o vlastnostiach. Ako príklad nám poslúži jedna z axiém RDF sémantiky:

```

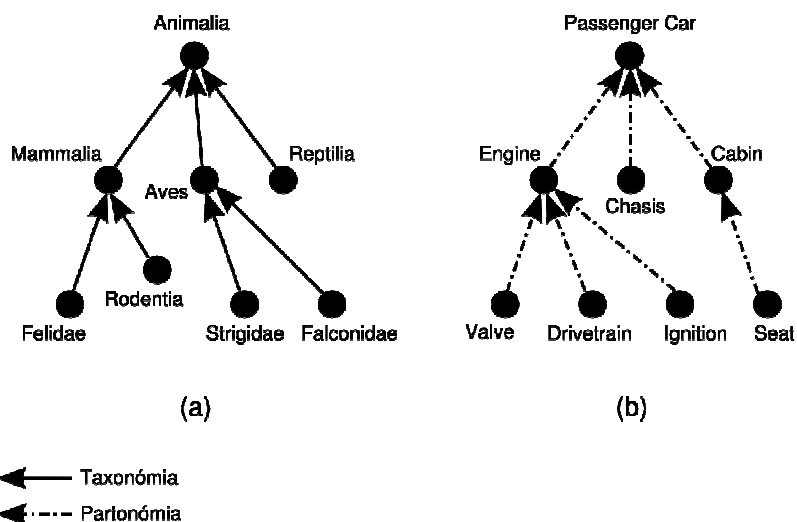
rdf:type    rdf:type    rdf:Property .
  
```

Tento triplet, ktorý musí byť v každej interpretácii pravdivý, definuje, že typom vlastnosti `rdf:type` je `rdf:Property`. Inými slovami, hovorí, že objekt `rdf:type` je vlastnosť. Podrobnú špecifikáciu sémantiky RDF nájde čitateľ v [16]. Problémom mnohých logík vyšších rádov sú ich zlé výpočtové vlastnosti. Kľúčové usudzovacie problémy môžu mať veľmi ťažkú výpočtovú zložitosť, alebo dokonca môžu byť nerozhodnuteľné. Našťastie, usudzovanie v RDF je v triede P, pokiaľ RDF grafy neobsahujú nepomenované vrcholy (t.j. premenné) a v triede NP, ak nepomenované vrcholy obsahujú [31]. Tento, na formalizmus logickej reprezentácie poznatkov pomerne dobrý výsledok, je možný vďaka tomu, že RDF je vo svojej podstate vlastne veľmi jednoduchý jazyk, ak nerátame sémantiku vyšších rádov: umožňuje nám používať len unárne a binárne predikáty a vôbec neobsahuje negáciu. Vyjadrovacie prostriedky, ktoré nám poskytuje sú teda pomerne obmedzené. RDF sa preto hodí predovšetkým na popis dát, nie je však vhodné na konštrukciu zložitejších ontológií. Za týmto účelom musíme siahnuť po silnejšom jazyku.

2.4 RDF Schema

Ako sme zdôraznili už v úvode, kritickou súčasťou sémantického webu sú ontológie. Ontológia je akási terminologická databáza s logickou štruktúrou. Sú v nej uložené údaje o terminológii, teda o rôznych pojmoch, ktoré používame pri zapisovaní údajov a pri komunikácii. Rozpoznávame tri základné typy pojmov. *Individuály* reprezentujú fyzické, či abstraktné objekty. Napríklad krajina Slovensko, či konkrétna osoba Isaac Newton môžu byť reprezentované inštanciami. Priradíme individuálom identifikátory `isaac` a `slovakia`. *Triedy*, ktoré sú tiež konceptmi zodpovedajú množinám, či skupinám individuálov. Príkladom môžu byť triedy všetkých ľudí (`Person`), žien (`Woman`), mužov (`Man`), alebo vedcov (`Scientist`), alebo napríklad obyvateľov Slovenska (`InhabitantOfSlovakia`). Individuály môžu patriť do jednej alebo viacerých tried, hovoríme, že sú *inštanciou* triedy, do ktorej patria. Napr. Isaaca Newtona by sme zaradili do tried všetkých ľudí, mužov a vedcov, ale nie do tried všetkých žien, či obyvateľov Slovenska. Posledným typom ontologickej entity je *vlastnosť* (tiež rola). Rola je vlastne pomenovaná relácia medzi dvojicami objektov. Napr. by sme mohli vytvoriť rolu `livesIn` a použiť ju na modelovanie faktu, že nejaký individuál z triedy `Person` býva na mieste reprezentovanom inštanciou triedy `Location`.

Dôležitou vlastnosťou ontológií je, že nám umožňujú okrem samotných pojmov presne definovať aj vzťahy medzi nimi. Ontologické reprezentačné jazyky sa tu vo veľkom opierajú o výskum v logickej reprezentácii poznatkov a využívajú na zachytenie týchto vzťahov logiku a logickú sémantiku. Napr. triedu všetkých žien môžeme poľahky zdefinovať ako triedu takých objektov, ktoré sú osobou, ale nie mužom. Využívajúc fakt, že v jazyku predikátovej logiky zodpovedajú triedy unárnym predikátom, môžeme takýto vzťah vyjadriť nasledovným axiómom: $Woman(x) \Leftrightarrow Person(x) \wedge \neg Man(x)$. Prípadne triedu všetkých obyvateľov Slovenska by sme mohli zdefinovať ako množinu takých individuálov, u ktorých evidujeme že na Slovensku bývajú: $InhabitantOfSlovakia(x) \Leftrightarrow livesIn(x, slovakia)$.



Obr. 2. (a) Vďaka vlastnostiam môžu v ontológii byť aj iné hierarchie. Na obrázku (b) vidíme príklad tzv. partonómie, teda hierarchie generovanej vlastnosťou "byť súčasťou" (partOf).

Dôležité postavenie majú v ontológiách vzťahy byť *podkonceptom* a byť *podrolou*. Hovoríme im tiež vzťahy subsumpcie medzi rolami a konceptami. Vďaka subsumpcii existuje v ontológii tzv. subsumpčná hierarchia konceptov a rolí, ktorú si môžeme predstaviť ako strom, ktorého koreňom je najvšeobecnejší koncept (rola) a listami sú koncepty (role) veľmi špecifické. Príklad jednoduchej subsumpčnej hierarchie vidíme na obrázku 2.

Isté črty ontologických jazykov nesie už RDF, ako vidno z opisu v predchádzajúcej časti. RDF umožňuje predovšetkým prácu s individuálmi a vlastnosťami. Jednoduchým rozšírením RDF, vhodnejším pre reprezentáciu ontológií je RDF Schema. Ide o nadstavbu nad RDF, ktorá vznikla rozšírením syntaxe a sémantiky RDF. RDF Schema dopĺňa nasledovné kľúčové slová:

```
rdfs:domain rdfs:range rdfs:Resource rdfs:Literal rdfs:Datatype
rdfs:Class rdfs:subClassOf rdfs:subPropertyOf rdfs:member
rdfs:Container rdfs:ContainerMembershipProperty rdfs:comment
rdfs:seeAlso rdfs:isDefinedBy rdfs:label
```

Pomocou kľúčových slov `rdfs:Class`, `rdfs:subClassOf` umožňuje slovník RDFs vytvárať triedy a podtriedy.

```
ex:Car rdf:type rdfs:Class .
ex:Vehicle rdf:type rdfs:Class .
ex:Car rdfs:subClassOf ex:Vehicle .
```

Kľúčové slová `rdfs:domain` a `rdfs:range` ďalej umožňujú nastaviť vlastnostiam (predikátom) obmedzenia na subjekt a objekt.

```
ex:hasPart    rdfs:domain    ex:Car .
ex:hasEngine rdfs:range     ex:Engine .
```

Slovník RDF Schema umožňuje tvorbu jednoduchých hierarchií tried, ich vlastností, ďalej umožňuje zdefinovať doménu a obor hodnôt vlastnosti. Umožňuje teda tvorbu veľmi jednoduchých ontológií. Pre komplexnejšie ontológie môžeme využiť jazyk OWL. Výhodou RDFs je predovšetkým nižšia výpočtová zložitosť. Usudzovanie v RDF Schema nemá vyššiu zložitosť, než usudzovanie v RDF [31]. Jedným z nástrojov na uchovávanie dát v RDF a RDFs je Sesame RDF data store. Sesame môže byť použitý ako databáza pre ukladanie dát v RDF a RDF Schema, alebo ako knižnica, ktorá poskytuje aplikáciám možnosť interne narábať s dátami v RDF. Obsahuje nástroje na spracovávanie, interpretovanie, dopytovanie a ukladanie informácií v RDF, či už ako interná súčasť vlastnej aplikácie alebo ako samostatná databáza, ktorá môže bežať na samostatnom serveri.

Jednou z aplikácií RDF je využívanie slovníka FOAF (z angl. Friend Of A Friend) na popis osôb a vzťahov medzi nimi. FOAF informácia môže ďalej obsahovať inštitúcie a projekty, pre ktoré daná osoba pracuje, aké dokumenty daná osoba vytvorila, či obrázky týchto osôb. Informácia o jednej osobe môže byť distribuovaná na viacerých miestach, keďže architektúra RDF umožňuje takúto decentralizáciu, avšak jedna osoba by mala byť identifikovaná jedným URI. V našom príklade by si mohol anglický fyzik Isaac Newton umiestniť na svoju web-stránku takýto FOAF kód.

```
<foaf:PersonalProfileDocument rdf:about="">
  <foaf:maker rdf:resource="#me"/>
  <foaf:primaryTopic rdf:resource="#me"/>>
</foaf:PersonalProfileDocument>
<foaf:Person rdf:ID="me">
  <foaf:name>Isaac Newton</foaf:name>
  <foaf:mbox rdf:resource="mailto:newton@cam.ac.uk"/>
  <foaf:knows>
    <foaf:Person>
      <foaf:name>Gottfried Leibniz</foaf:name>
      <foaf:mbox
rdf:resource="mailto:gottfriedwl@royalsociety.org"/>
    </foaf:Person>
  </foaf:knows>
</foaf:Person>
```

V zjednodušenej Turtle syntaxi sa táto informácia dá zapísať nasledovne:

```
<#me>
  a foaf:Person ;
  foaf:name "Isaac Newton" ;
  foaf:mbox <mailto:newton@cam.ac.uk> ;
```

```
foaf:knows [  
  a foaf:Person ;  
  foaf:name "Gottfried Leibniz" ;  
  foaf:mbox <mailto:gottfriedwl@royalsociety.org>  
] .
```

Tak ako Isaac Newton, tak aj ktokoľvek kto má pocit, že sa s určitou osobou pozná si môže na web umiestniť svoju časť sociálnej siete. Môžu tak spraviť aj inštitúcie a zverejniť v tomto formáte zoznam svojich zamestnancov a projektov, na ktorých pracujú. Takýmto spôsobom je informácia distribuovaná na viacero miest.

RDF sa často využíva aj na vyjadrenie metadát, teda dát o dátach. Príkladom môže byť RDF slovník Dublin Core, ktorý obsahuje elementy na popis metadát dokumentov podľa štandardov knižničnej a informačnej vedy. Takýmito metadátami sú napríklad názov, meno autora, bližší popis, dátum, autorské práva, jazyk, zdroj a mnohé ďalšie. Príklad zápisu takýchto metadát:

```
<rdf:RDF  
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"  
  xmlns:dc="http://purl.org/dc/elements/1.1/">  
  <rdf:Description  
rdf:about="http://books.google.com/books?id=Tm0FAAAAQAAJ">  
    <dc:title>Principia</dc:title>  
    <dc:creator>Isaac Newton</dc:creator>  
    <dc:subject>gravity, weight, motion, physics</dc:subject>  
    <dc:description>  
      In this work Newton stated the three universal laws of  
      motion. He used the Latin word gravitas (weight) for the  
      effect that would become known as gravity, and defined the  
      law of universal gravitation.  
    </dc:description>  
    <dc:date>1687-7-5</dc:date>  
  </rdf:Description>  
</rdf:RDF>
```

2.5 OWL

V predchádzajúcej časti sme si ukázali, že pomocou RDFs je možné vytvárať jednoduché hierarchie tried a vlastností. Ak však chceme nejakú doménu popísať komplexne pomocou tried, vlastností a individuálov a zároveň tieto objekty detailne charakterizovať, ako aj vzťahy medzi nimi, je nutné siahnuť po nejakom silnejšom jazyku. Konzorcium W3C za týmto účelom v roku 2002 predstavilo Web Ontology Language (OWL), ktorý sa stal štandardom vo februári 2004 [24]. V októbri 2009 bola štandardizovaná nová verzia OWL 2 [15]. Kým RDFs nám umožnilo vytvárať jednoduché hierarchie tried a vlastností, OWL značne rozširuje tieto výrazové možnosti. OWL prináša možnosť popísať vzťahy medzi triedami, dopĺňa do jazyka tzv. konštruktory zložitých tried, pomocou ktorých môžeme zadefinovať komplexné triedy za pomoci jednoduchších tried. Môžeme skonštruovať napríklad prienik, zjednotenie, či komplement triedy.

```
:Woman owl:equivalentClass [
  rdf:type owl:Class ;
  owl:intersectionOf ( :Person [ owl:complementOf :Man ] )
] .
```

Príklad vyššie jednoduchým spôsobom definuje koncept Woman (žena) ako koncept Person (osoba), ktorá je zároveň komplementom konceptu Man (muž).

```
:Parent owl:equivalentClass [
  rdf:type owl:Class ;
  owl:unionOf ( :Mother :Father )
] .
```

V tomto príklade sme zadefinovali komplexnú triedu Parent (rodič) pomocou zjednotenia tried Mother a Father (mama a otec). Každý individuál, ktorý patrí do jednej z jednoduchých tried Mother alebo Father je zároveň aj individuálom triedy Parent.

Pomocou tzv. reštrikcií môžeme tiež vymedziť triedu ako množinu objektov majúcich vzťahy určitého typu s inými objektmi. Môžeme napr. zadefinovať auto (Car) ako vozidlo (Vehicle), ktoré má motor (Engine) a štyri kolesá (Wheel), využívajúc pri tom rolu hasPart.

```
:Car owl:equivalentClass [
  rdf:type owl:Class ;
  owl:intersectionOf ( :Vehicle
    [ rdf:type owl:Restriction ;
      owl:onProperty :hasPart ;
      owl:someValuesFrom :Engine
    ]
    [ rdf:type owl:Restriction ;
      owl:onProperty :hasPart ;
      owl:onClass :Wheel ;
      owl:qualifiedCardinality "4"^^xsd:nonNegativeInteger
    ]
  )
] .
```

Existujú tiež vymenované koncepty a ďalšie konštruktory.

```
:VysegradFour owl:equivalentClass [
  rdf:type owl:Class ;
  owl:oneOf ( :CzechRepublic :Hungary :Slovakia :Poland )
] .
```

OWL pozná dva typy vlastností, tzv. objektové vlastnosti, ktoré spájajú 2 inštancie a dátové vlastnosti, pomocou ktorých možno k objektom priradiť hodnoty v rôznych dátových typoch. Možno napr. zaznamenať vek konkrétnej osoby:

```
:Jane :hasAge 17 .
```

Alebo možno zdefinovať triedu mládež (Youth) ako ľudí s ohraničeným vekom:

```
:Youth rdfs:subClassOf
  [ rdf:type owl:Restriction ;
    owl:onProperty :hasAge ;
    owl:someValuesFrom
      [ rdf:type rdfs:Datatype ;
        owl:onDatatype xsd:integer ;
        owl:withRestrictions (
          [ xsd:minExclusive "12"^^xsd:integer ]
          [ xsd:maxInclusive "18"^^xsd:integer ]
        )
      ]
  ] .
```

Technicky vzaté, je jazyk OWL implementovaný ako syntaktické a sémantické rozšírenie RDF a RDFs. OWL pridáva nový menný priestor owl:, v ktorom sa nachádza veľké množstvo "kľúčových slov" (v zmysle RDF ide o vlastnosti), ktoré majú v sémantike fixne stanovenú interpretáciu. Ide predovšetkým o tieto vlastnosti:

```
owl:AllDisjointClasses owl:AllDisjointProperties
owl:allValuesFrom owl:Annotation owl:AsymmetricProperty
owl:Axiom owl:cardinality owl:Class owl:complementOf
owl:DatatypeProperty owl:differentFrom owl:disjointUnionOf
owl:disjointWith owl:distinctMembers owl:equivalentClass
owl:equivalentProperty owl:FunctionalProperty owl:hasValue
owl:imports owl:intersectionOf owl:InverseFunctionalProperty
owl:inverseOf owl:IrreflexiveProperty owl:maxCardinality
owl:members owl:minCardinality owl:NamedIndividual owl:Nothing
owl:object owl:ObjectProperty owl:oneOf owl:Ontology
owl:OntologyProperty owl:predicate owl:propertyDisjointWith
owl:ReflexiveProperty owl:Restriction owl:sameAs
owl:someValuesFrom owl:subject owl:SymmetricProperty owl:Thing
owl:TransitiveProperty owl:unionOf owl:withRestrictions
```

OWL dokumenty možno teda zapisovať v RDF/XML syntaxi, s použitím nového OWL slovníka. Táto syntax sa používa ako kánonická pri zaznamenávaní, uchovávaní a strojovom spracovaní ontológií. Pre ľudského čitateľa je však priveľmi neprehľadná a zle čitateľná. Našťastie si môžeme opäť pomôcť Turtle syntaxou, či niektorou inou syntaxou, pričom špecifikácia OWL 2 nám ich ponúka hneď päť. Príklad na deklaráciu komplexnej triedy pomocou komplementu, ktorý sme uviedli vyššie v Turtle syntaxi, môžeme zapísať v RDF/XML syntaxi takto:


```

<owl:Class rdf:about="Woman">
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <owl:Class rdf:about="Person"/>
        <owl:Class>
          <owl:complementOf rdf:resource="Man"/>
        </owl:Class>
      </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>

```

Sémantika OWL, ktorá je rozšírením sémantiky RDF, je silnejšia a zaručuje OWL isté vhodné výpočtové vlastnosti. Táto sémantika vychádza z deskriptívnych logík [4]. OWL vo verzii 2 zodpovedá expresívnej deskriptívnej logike SROIQ(D) [18]. Vďaka tomu je možné využiť na inferenciu nad OWL ontológiami existujúce usudzovače pracujúce s deskriptívnymi logikami Pellet², Racer³, alebo Fact++⁴, čo výrazne pridáva na praktickej využiteľnosti jazyka OWL.

V ontológii môžeme mať definovaný napríklad koncept auto (Car), ktorý má motor a 4 kolesá. Následne vytvoríme podtriedu, do ktorej budú patriť všetky individuály konceptu Car s nejakou špecifickou vlastnosťou (UnusualCar). Vložíme koncept nezvyčajného trojkolesového auta (ThreeWheeledUnusualCar), ktoré by malo len 3 kolesá. Následne spustíme usudzovač, aby overil konzistenciu ontológie. Usudzovač vyhlási, že takáto ontológia je nekonzistentná a dokonca nás navedie na miesto, kde je chyba. V našom prípade označí nekonzistentné koncepty Car a ThreeWheeledUnusualCar.

Význam jazyka OWL dokumentuje aj fakt, že na webe je v súčasnosti voľne dostupných viac než 20 000 ontológií v tomto jazyku. Väčšinou ide o doménovo špecifické ontológie vytvorené pre konkrétnu oblasť (medicína, jedlo, počasie, ...), no dostupné sú aj veľmi všeobecné ontológie určené na prepájanie doménovo špecifických ontológií, tzv. upper ontológie. Na vyhľadanie vhodnej ontológie pre konkrétny účel je možné použiť vyhľadávače Swoogle⁵ alebo Watson⁶. Veľký počet voľne dostupných ontológií v jazyku OWL napovedá, že sme sa opäť viac priblížili k vízii sémantického webu. O niektorých výzvach sémantického webu budeme hovoriť v nasledovnej časti.

² <http://pellet.owldl.org/>

³ <http://www.racer-systems.com/>

⁴ <http://code.google.com/p/factplusplus/>

⁵ <http://swoogle.umbc.edu/>

⁶ <http://kmi-web05.open.ac.uk/WatsonWUI/>

3 Niektoré výzvy sémantického webu

3.1 Distribuované ontológie

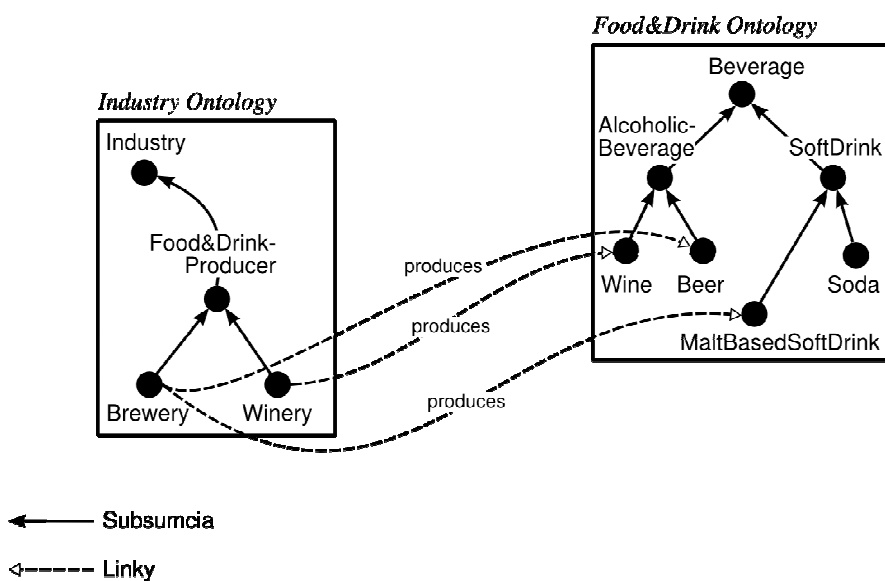
Ontológie stoja od začiatku v centre pozornosti vízie sémantického webu. Už v roku 2001 Berners-Lee a kol. [6] označili ontológie za hlavný nosný prvok pre význam na sémantickom webe. Zároveň však v tej istej publikácii volajú po ďalšom výskume v tejto oblasti, ktorý by umožnil lepšie integrovať ontologické dáta s inherentne distribuovaným prostredím webu. Distribuovaný charakter webových zdrojov je jednou z najsilnejších stránok webu. Fakt, že dáta sú zverejňované nezávisle od seba z rôznych zdrojov, ktoré sa navzájom dopĺňajú, referencujú a prelínajú výrazne prispel k rozmachu webu do jeho dnešnej podoby. Je preto viac než žiaduce, aby aj ontologické dáta mali tento charakter.

Problémom a výskumnou výzvou, na ktorú pri takomto uvažovaní narazíme, je fakt, že historicky väčšina výskumu v ontologickej reprezentácii znalostí distribuovaný scenár vôbec neuvažuje [4,30]. Ontológia je takmer vždy vnímaná ako monolitická štruktúra bez ďalšieho delenia. Z pohľadu sémantického webu je takýto prístup nepostačujúci. Len veľmi ťažko si možno predstaviť, že by existovala len jedna centrálna ontológia, obsahujúca všetky ontologické znalosti modelované jednotným spôsobom. Takúto ontológiu by bol jednak problém skonštruovať, teda dohodnúť sa na jedinom pohľade na svet, a tiež udržiavať, keďže zásahy do takej obrovskej ontológie by nutne boli opatrné a zdĺhavé. Tiež by bol problém s jej využitím a usudzovaním nad jej štruktúrami, vďaka zložitosti usudzovacích algoritmov by sme sa prakticky nemuseli dočkať odpovede.

V kontraste, distribuovaná ontologická reprezentácia musí umožniť existenciu nezávislých zdrojov ontologických dát, z ktorých každý pokrýva určitú znalostnú doménu. Tieto zdroje sa môžu čiastočne aj úplne prekrývať. Môžu sa navzájom referencovať a využívať tak znalosti zachytené v inom zdroji. Môže dôjsť k dočasnému výpadku niektorého z nich. V neposlednom rade, keďže zdroje sú úplne nezávislé, musíme sa vysporiadať aj s prípadom, ak je niektorý zo zdrojov lokálne nekonzistentný, alebo ak niektoré zdroje obsahujú nekompatibilné, konfliktné informácie. Aj za takýchto okolností chceme s distribuovanou ontológiou pracovať, identifikovať relevantné znalosti v nej uložené a usudzovať s nimi. Problém distribuovanej ontologickej reprezentácie je teda zaujímavou a netriviálnou výzvou vedeckého bádania.

V literatúre nájdeme hneď niekoľko prístupov k distribuovaným ontológiám. Väčšinou sú predstavené ako rozšírenie deskriptívnych logík [4], ktoré sú overeným a dobre prebádaným reprezentačným jazykom pre klasické ontológie. Za účelom kombinácie a prepojenia niekoľkých ontologických zdrojov využívajú tieto prístupy nové konštrukty, ktoré sa pre každý z nich líšia. Rozlišujeme tak medzi tromi paradigmami v distribuovaných ontológiách: linkovanie, mapovanie a importovanie ontológií.

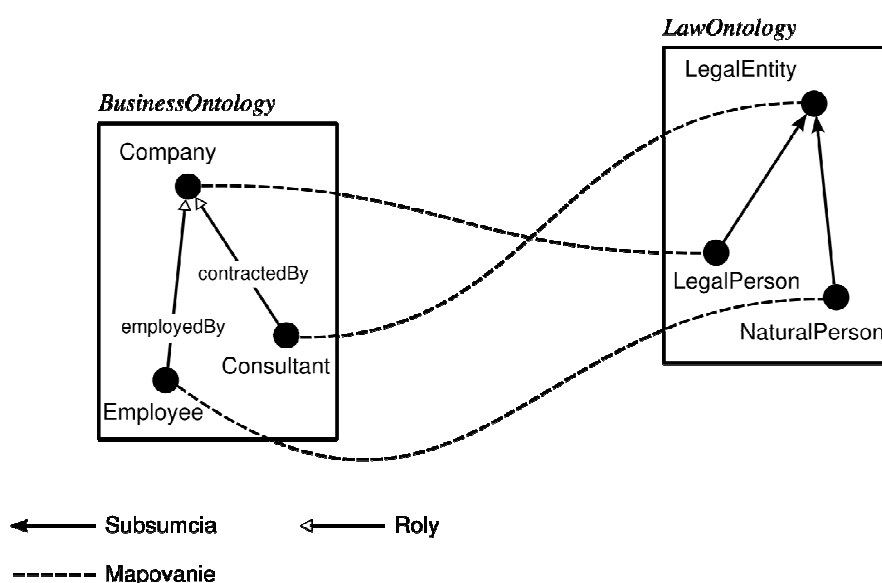
Linkovanie ontológií umožňuje nový druh vlastností, nazývaných *linky*, ktoré spájajú inštalácie medzi dvoma ontológiami. V RDF aj v OWL využívame vlastnosti (role) na spojenie dvoch inštalácií, či na definíciu zloženého konceptu. Podobne aj link spája dve inštalácie, ale každú z inej ontológie. Môžeme ho tiež využiť na definíciu zloženého konceptu, ktorý je vystavaný z podkonceptov iných ontológií. Ak máme napríklad jednu ontológiu rôznych druhov podnikov v priemysle a druhú ontológiu rôznych potravinárskych produktov, môžeme v prvej z nich zdefinovať koncept Winery, tj. vinohradnícky závod, ako triedu všetkých producentov produkujúcich víno, pričom využijeme koncept Wine, nachádzajúci sa v ontológii produktov a link produces. Vid' obrázok 3. Ontológia podnikov tak využíva znalosti predtým zachytené v ontológii potravinárskych produktov. Ontologické linkovanie vyžaduje, aby sme znalostnú doménu striktno rozdelili medzi jednotlivé ontológie, ktoré sú poprepájané linkami. Tak je to aj v našom príklade, kde inštalácie v jednej ontológii sú priemyselné podniky a v druhej potravinárske produkty, a majú teda vždy prázdny prienik. Táto požiadavka môže byť niekedy obmedzujúca vzhľadom na scenár sémantického webu, pretože nepripúšťa duplicitu znalostí.



Obr. 3. Linkovanie ontológií.

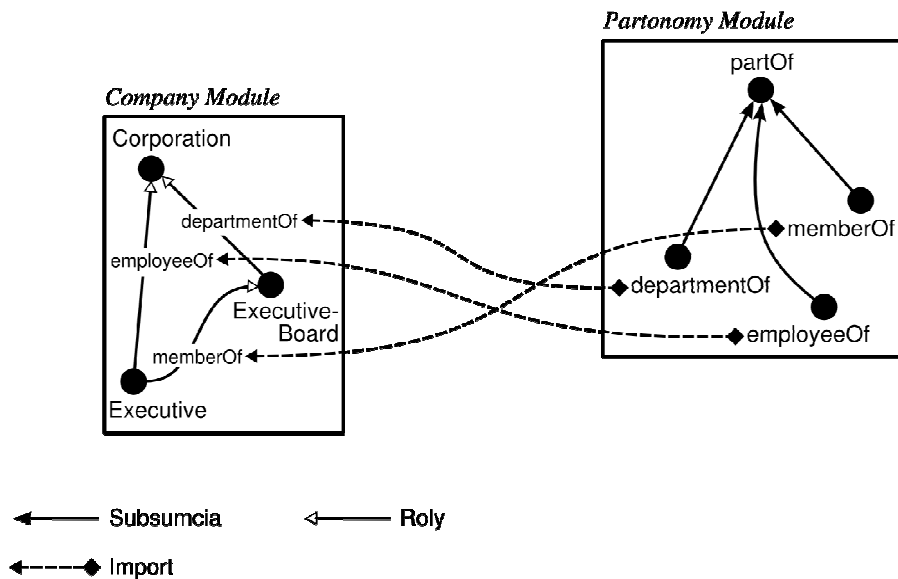
Mapovanie ontológií umožňuje vyznačiť vzťahy medzi entitami v rôznych ontológiách pomocou *sémantického mapovania*. Mapovanie je nový konštrukt, pridaný do jazyka. Mapovať je možné napríklad medzi dvoma konceptami, medzi dvoma vlastnosťami (rolami), či individuálmi. Rozpoznávame viacero druhov mapovaní. Mapovanie môže byť jednosmerné, alebo obojsmerné. V závislosti od ich druhu môžeme mapovaním reprezentovať ekvivalenciu, disjunkciu, či subsumpciu medzi

dvoma entitami z dvoch rôznych ontológií. Uvažujme príklad dvoch ontológií, pozri obr. 4. Prvá zachytáva obchodné vzťahy medzi inštanciami, druhá zachytáva právne vzťahy. Mapovaním sme vyznačili vzťahy medzi konceptami z týchto dvoch ontológií, napríklad fakt, že konzultant (Consultant) môže byť rovnako fyzická aj právnická osoba (mapujeme s nadtriedou LegalEntity), ale koncept zamestnanec (Employee) vždy zodpovedá konceptu fyzická osoba (NaturalPerson). V tomto prípade sme využili mapovanie reprezentujúce ekvivalenciu medzi konceptami. Sémantika ontologického mapovania umožňuje prenos znalostí z jednej ontológie do druhej, a tým aj ich opätovné využitie. Navyše, ako vidno aj z nášho príkladu, možno mapovať aj medzi ontológiami, ktoré sa čiastočne prelínajú.



Obr. 4. Mapovanie ontológií.

Importovanie ontológií umožňuje vymedziť podmnožinu entít v jednej ontológii (tzv. zdrojová ontológia) a *importovať* ich do inej ontológie (tzv. cieľová ontológia). Tieto entity následne môžeme v cieľovej ontológii ľubovoľne využiť pri modelovaní. Sémantika ontologických importov zabezpečí, že spolu s importovanými entitami sa prenesú aj sémantické vzťahy medzi nimi, ktoré vyplývajú zo zdrojovej ontológie. Dochádza tak k efektívnemu opätovnému využitiu znalostí zachytených v zdrojovej ontológii. Pozrime si príklad na obrázku 5. Jedna z ontológií na obrázku modeluje opäť obchodné vzťahy. Využíva pri tom niekoľko vlastností, resp. rolí ako *memberOf*, *employeeOf* a *departmentOf*. Tieto role sú všetky importované zo samostatného ontologického modulu, ktorý zachytáva vzťahy medzi rôznymi typmi partonómii.



Obr. 5. Importovanie ontológií.

Medzi známe formalizmy nasledujúce paradigmu ontologického mapovania patria Distributed Description Logics (DDL) [8] a Integrated Distributed Description Logics (P-DL) [34]. Kým DDL pracuje s jednosmerným mapovaním, IDDL využíva obojsmerné mapovanie. Pod ontologické linkovanie radíme formalizmus E-connections [22], a pod paradigmu importovania radíme Package-based Description Logics (P-DL) [5]. Medzi silné výsledky v tejto oblasti patria usudzovacie algoritmy, ktoré poznáme, aspoň do istej miery pre každý z uvedených formalizmov. Najlepšie vyvinuté sú algoritmy pre E-connections, ktorý bol prakticky implementovaný v usudzovači Pellet a pre DDL, prakticky implementovaný v usudzovači DRAGO⁷. Zaujímavosťou systému DRAGO je, že umožňuje plne distribuované usudzovanie, kedy je každá ontológia spravovaná svojím vlastným usudzovacím agentom, pričom agenti medzi sebou komunikujú. Takýto scenár je v súlade s ďalším predpokladom vízie sémantického webu, a síce, že jednotlivé ontologické zdroje nemusia nutne poskytnúť všetky svoje dáta, ale miesto nich len rozhranie, pomocou ktorého je možné nad dátami robiť požiadavky. Čitateľovi s hlbším záujmom o túto problematiku odporúčame prehľadovú prácu [29].

⁷ <http://drago.itc.it/>

3.2 Sémantický web a Web 2.0

Pojmom Web 2.0 sú označované webovské aplikácie, ktoré umožňujú zdieľanie informácií, podporujú interoperabilitu s inými aplikáciami a vďaka dizajnu, ktorý je zameraný na používateľa je ich používanie jednoduché a umožňuje kolaboráciu medzi viacerými používateľmi. Často sa týmto spôsobom interaktívne mení samotný obsah umiestnený na týchto webovských službách, čo nebolo vždy bežné, používatelia boli zvyknutí pasívne prijímať informácie, ktoré im boli na webovských stránkach poskytnuté. Aj keď názov Web 2.0 môže evokovať, že ide o nejakú novú verziu pôvodného webu, nejde o žiadnu aktualizáciu technických špecifikácií alebo štandardov, ide tu skôr o zmenu spôsobu, akým koncoví používatelia a vývojári používajú web. Web 2.0 priniesol mnoho nových pojmov a aplikácií, s ktorými sa dnes už bežne stretávame: blogy, wiki stránky, folksonómie, sociálne siete, komunity a podobne. Web 2.0 je často stavaný do kontrastu so sémantickým webom, v skutočnosti však nejde o alternatívne prístupy. Zatiaľ čo Web 2.0 má vplyv najmä na interakciu používateľov s webom, sémantický web otvára nové technologické možnosti pre kombináciu dát a služieb z rôznych zdrojov.

Jednou z obáv vedeckej komunity sémantického webu je otázka, či budú používatelia ochotní poskytovať k obsahu aj doplnkové informácie, resp. metadáta. Vďaka rozmachu Web 2.0 však už vieme, že používatelia okrem obsahu radi publikujú aj metadáta, poznáme ich napríklad vo forme tabuliek a kategórií na Wikipedii, fotiek organizovaných do galérií na Flickr, či v podobe štruktúrovaných informácií vložených na webstránky pomocou tzv. mikroformátov. Mikroformáty sú jednoduché dátové štruktúry resp. konvencie, ktoré využívajú už existujúce široko rozšírené štandardy HTML resp. XHTML. Dáta poskytnuté v mikroformáte sú následne jednoducho spracovateľné a prístupné strojovými metódami. Nemôžeme však predpokladať, že bežný používateľ sa naučí zapisovať metadáta v jazyku RDF, XML, alebo nejakom mikroformáte. Toto tvrdenie je posilnené faktom, že mnoho používateľov ani v súčasnosti nie je schopných tvoriť svoje webstránky v korektnom HTML kóde. Vieme teda, že ochota poskytovať takéto štruktúrované dáta u používateľov existuje, avšak spôsob akým ich zadávajú musí byť jasne orientovaný na nejakú konkrétnu úlohu. Zároveň je potrebné pri tom využívať priateľské používateľské rozhranie, ktoré ukryje zložitosť skutočnej reprezentácie dát na pozadí.

Úspešne motivovať používateľov sa podarilo službe Google Image Labeler (obr.6), pomocou ktorej používatelia poskytujú anotácie obrázkov vo forme jednoduchej hry, pôvodne pomenovanej ako ESP Game. Tieto anotácie sa potom využívajú na zdokonaľovanie vyhľadávania obrázkov na webe. Hru vždy hrajú spoločne dvaja náhodní hráči, ktorí sa nepoznajú a môžu sa nachádzať kdekoľvek na svete. Po začatí hry sa obidvom hráčom zobrazí ten istý obrázok. Ich jediným spôsobom komunikácie a zároveň jediným možným úkonom je poskytnúť k tomuto obrázku slovný opis. V prípade, že sa obaja hráči zhodnú na rovnakom popise, obidvaja získajú body. Hráči sú teda motivovaní poskytovať zmysluplné anotácie, navyše musia pritom myslieť aj na to, akú anotáciu by obrázku priradil niekto iný. Vďaka jednoduchému používateľskému rozhraniu a orientácii používateľov na

konkrétnu úlohu resp. hru sa podarilo tvorcom tohto nástroja získať od používateľov cenné dáta.

Mnohé webstránky sú už teraz technicky pripravené na publikovanie sémantických dát bez toho, aby o tom autor stránky niečo tušil. Spomeňme tzv. blogy (z angl. "web logs"), ktoré sa dnes stali obvyklou, či dokonca v niektorých prípadoch hlavnou súčasťou webovských stránok. Na svojom blogu môžu autori publikovať jednoduchou formou svoje názory, hodnotenia či pocity vo forme malých, ohraničených príspevkov zoradených podľa dátumu zverejnenia. Samotná jednoduchosť zverejňovania príspevkov pomohla prelomiť istú bariéru a mnohí pôvodne pasívni používatelia webu sa vďaka blogovaniu dostali k aktívnemu publikovaniu vlastného obsahu. Nástroje na blogovanie možno doplniť o existujúce rozšírenia, ktoré poskytujú informácie o autoroch v mikroformáte, napríklad v podobe vizitky autora blogu, ktorú si môže používateľ stiahnuť do svojho elektronického adresára.

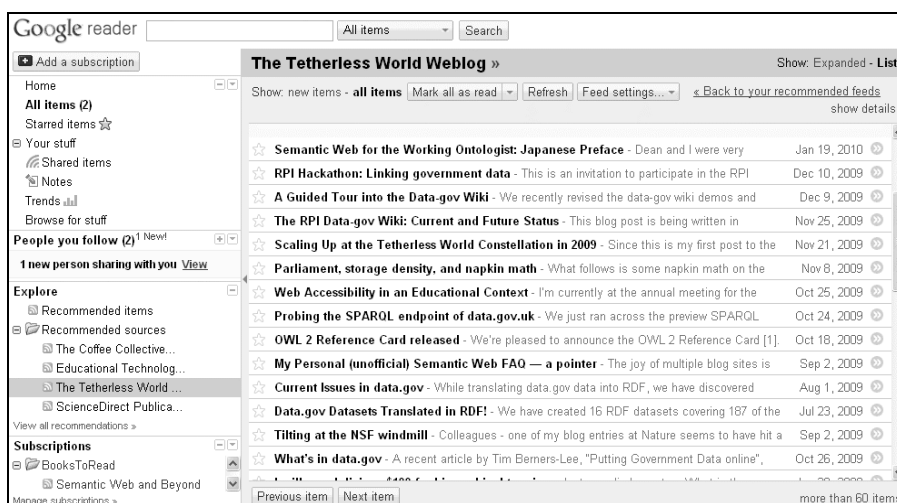
Tak ako blogovanie prinieslo viac aktívnych používateľov na webe, existuje istá šanca ako by táto forma publikácie mohla pomôcť rozšíreniu sémantického webu. Prečo? Blogovanie a sémantický web majú na prvý pohľad málo spoločného, no keďže obidva prístupy vedú k spoločnému cieľu -- web dneška obohatený o metadáta -- odpoveďou by mohol byť koncept tzv. *sémantického blogovania*, teda publikovania blogov kódovaných v RDF formáte [20].



Obr. 6. Google Image Labeler: hra, pri ktorej dvaja používatelia popisujú obrázok slovami. Ak sa zhodnú, získavajú body.

Blogy umožňujú autorom vyjadrovať sa, teda anotovať zdroje na webe a publikovať tieto anotácie pre ostatných. Príspevky na blogoch sú veľmi často hodnoteniami iných stránok, recenziami produktov, odpoveďami na otázky položené inde na webe, teda reakciami na príspevky ostatných autorov. Sú vlastne popismi iných zdrojov, čo nám napovedá, že tu by mohol nájsť využitie jazyk RDF, ktorý je práve na tieto účely vhodný. Príspevky bývajú zoskupené podľa spoločného autora, avšak zoskupenie podľa spoločnej témy by mohlo uľahčiť orientáciu v konverzácii prebiehajúcej na blogoch viacerých autorov.

Charakteristickou črtou blogovania je publikovanie v podobe krátkych ohraničených príspevkov. Ďalším kľúčovým konceptom je decentralizované publikovanie. Každý používateľ si spravuje a vydáva svoj vlastný blog, existujú však aj spoločné blogy projektov, tímov alebo firiem. Pocit "vlastníctva", teda plnej kontroly nad obsahom a vzhľadom blogu prispieva k atraktivnosti blogovania ako takého. Decentralizovaný charakter blogov hrá úlohu aj z pohľadu technickej realizácie. Blogy používateľov nemusia byť umiestnené na jednom centrálnom serveri, ale bývajú roztrúsené na mnohých miestach na webe. Sledovanie viacerých blogov môže byť preto náročné, no je to možné vďaka publikovaniu metadát týkajúcich sa príspevkov.



Obr. 7. RSS čítačka Google Reader. Blogy sledované pomocou RSS môžeme síce chronologicky čítať a usporiadať do vlastných kategórií, no nemožno sledovať tematickú príbuznosť blogov alebo priebeh jednej konverzácie na viacerých blogoch.

Metadáta o príspevkoch sú už dnes štandardnou súčasťou blogov, konkrétne sa vyskytujú v podobe zoznamu príspevkov, ktorý je strojovo spracovateľný. Využíva sa pritom formát RDF Site Summary, skrátene RSS, založený na XML. (V praxi sa vyskytujú rôzne verzie a mutácie RSS, nie všetky založené na RDF, okrem toho je bežný aj ďalší formát Atom, ale keďže všetky vychádzajú z XML a obsahujú rovnaké

informácie, je možné ich transformovať do RDF pomocou technológie XSL transformácií.) RSS umožňuje zoskupovanie príspevkov z rôznych blogov od viacerých autorov na jednom mieste pomocou nástrojov na čítanie takýchto formátov, tzv. RSS agregátorov alebo čítačiek. Stále však neumožňuje napríklad sledovanie jednej témy alebo konverzácie (obr.7).

Z pohľadu sémantického webu je blogovanie priateľskou formou vytvárania sémantických anotácií. Blogy v súčasnosti bežne obsahujú recenzie produktov, komentáre a iné informácie v prirodzenom ľudskom jazyku. Anotácie sémantických blogov však môžu obsahovať aj strojovo spracovateľné dáta, napríklad kvantitatívne hodnotenia produktov. S týmito potom môže pracovať sémantický nástroj vo forme agenta, ktorý je schopný zoskupovať hodnotenia konkrétneho produktu a pomáhať tak zákazníkom napríklad pri nákupe elektroniky, či výbere hudby. Iným príkladom by mohli byť príspevky zamerané na šport. Hodnotenia športových zápasov by mohli automaticky obsahovať strojovo spracovateľné informácie o výsledku zápasu, dlhodobé štatistiky jednotlivých súperov by následne generoval sémantický agent agregovaním a spracovávaním týchto čiastkových informácií.

Keďže autori zvyknú na svojich blogoch písať o viacerých témach, obvykle rozdeľujú príspevky do viacerých kategórií podľa tematického zamerania. Sémantický klient Haystack napríklad umožňuje definovať kolekcie, ktorých úlohou je slúžiť ako kategória, do ktorej možno priradiť príspevky. Kolekcia môže byť identifikovaná pomocou URI, ktorým pomenujeme nejakú kategóriu. Takéto kategórie môžu byť vďaka tomu používané nielen na blogoch ľudí zameraných na rovnakú oblasť záujmu, ale vo všeobecnosti na rôznych blogoch. Vďaka univerzálnosti URI sa navyše vyhneme mylnému spájaniu kategórií s rovnakým názvom (napr. kategória "Jaguar" na blogu zoológa a kategória "Jaguar" na blogu výskumníka operačných systémov pravdepodobne obsahujú príspevky rôznych tém), majú totiž odlišné URI. Zároveň sa tým uľahčí vytváranie mapovaní medzi rôznymi kategorizačnými schémami. Kolekcie v sémantickom klientovi možno okrem toho využiť pri pohľade na vzťahy medzi jednotlivými členmi kolekcie. Užitočné môže byť skúmanie relácie *inReplyTo*, sémantický klient zobrazí príspevky a šípkami označí, ktorý príspevok bol publikovaný ako odpoveď na niektorý iný. Informáciu o takejto relácii však musíme najskôr nejakým spôsobom získať. V tomto prípade sa tieto metadáta zapisujú pri používaní sémantického klienta, ktorý pri čítaní blogov ponúka akciu "Odpovedaj na tento blog". Táto akcia spôsobí vytvorenie nového príspevku, ktorý v sebe bude obsahovať reláciu *inReplyTo* spojenú s pôvodným príspevkom.

Využitie dodatočných metadát teda môže v budúcnosti slúžiť novým spôsobom nazerania na príspevky na blogoch. Okrem v súčasnosti bežného agregovania príspevkov z rôznych zdrojov zoradených podľa času publikovania ich tak budeme môcť sledovať či už vo forme vizualizovanej konverzácie prebiehajúcej na viacerých blogoch, alebo podľa spoločnej kategórie resp. témy identifikovanej pomocou rovnakého URI.

V duchu rozširovania existujúcich používateľských rozhraní za účelom získania štruktúrovaných sémantických informácií od používateľov sa nesú aj projekty Semantic MediaWiki [21] alebo MoKi [28]. Oba nástroje sú rozšíreniami softvéru MediaWiki, ktorý je známy najmä svojím použitím v projekte slobodnej online

encyklopédie Wikipedia. Využitie rozšírenia Semantic MediaWiki práve na Wikipedii by mohlo pomôcť odstrániť niektoré jej zreteľné problémy. Prvým problémom je nekonzistentnosť obsahu: rovnaké informácie sa nachádzajú na viacerých stránkach, ako však zaručiť ich konzistenciu, ak encyklopédiu upravuje veľké množstvo editorov, pričom kolaboratívne upravovanie viacerými autormi je jednou zo základných črt Wikipédie a wiki systémov ako takých? Príkladom problematickej stránky môže byť napríklad "Zoznam 10 najobývanejších európskych hlavných miest". Ak sa napríklad zmení populácia nejakého hlavného mesta z tohto zoznamu, treba v tom istom kroku aktualizovať množstvo iných stránok. Druhým problémom Wikipédie je náročný prístup k znalostiam: ak by sme napríklad chceli získať zoznam všetkých európskych hlavných miest s nadmorskou výškou vyššou ako 200 metrov, neuspeli by sme. Napriek tomu, že tieto údaje sa vo Wikipedii nachádzajú, nevieme sa k nim dopracovať jednoduchou cestou. Tretím problémom je využitie informácií obsiahnutých vo Wikipedii na iných miestach a iným spôsobom, než v okne webovského prehliadača. Príkladom môže byť hudobný prehrávač Amarok, ktorý vo svojom okne zobrazuje dopĺňujúce informácie o práve pustenej skladbe. Môže však poskytnúť len presne taký obsah, aký nájdeme pri prezeraní Wikipédie cez prehliadač, čiže pre tento účel zbytočne rozsiahly a zložitý. Neexistuje možnosť ako použiť len niektoré údaje, napr. zobrazíť fotografiu hudobnej skupiny, popísať hudobný štýl a obdobie a zoznam albumov.

Riešenie spomenutých problémov teda prináša nástroj Semantic MediaWiki, ktorý umožňuje editorom vkladať do stránok pomocou špeciálnych značiek sémantické anotácie. Tieto sú následne namapované na formálny popis pomocou jazyka OWL DL. Editorom wiki obsahu stačí poznať niekoľko typov anotácií. Jednoduchou formou anotácií sú kategórie, ktoré slúžia na klasifikáciu príspevkov do nejakej skupiny. Ak chceme wiki stránke o Isaacovi Newtonovi priradiť kategóriu Physics, vložíme na túto stránku značku `[[Category:Physics]]`. Vzťah medzi dvoma wiki stránkami možno vyjadriť pomocou relácií. Ak chceme uviesť na stránke o Isaacovi Newtonovi informáciu o krajine, v ktorej sa narodil, vložíme na jeho wiki stránku značku `[[bornIn::England]]`, čím zaznačíme sémantickú reláciu `bornIn` medzi stránkami `Isaac Newton` a `England`. Pre zaznačenie Newtonovho dátumu narodenia stačí vložiť na stránku o ňom značku `[[bornOn:=Jan 4 1643]]` a využiť pri tom tretí typ sémantickej anotácie wiki textu a to špecifikáciu atribútov.

Okamžité použitie sémantických dát je možné prostredníctvom dopytovacieho jazyka SPARQL. Krátke požiadavky v tomto jazyku možno vkladať priamo do wiki textov. Ak takýmto spôsobom vytvoríme stránku Zoznam 10 najobývanejších európskych miest a vložíme do stránky SPARQL požiadavku, tento zoznam bude vždy aktuálny a konzistentný s údajmi na ostatných stránkach, to všetko za predpokladu, ak sme pri ich tvorbe použili sémantické anotácie, z ktorých je pri vyhodnocovaní tejto požiadavky čerpané. V našom prípade to znamená, že zoradiť mestá podľa populácie môžeme len za predpokladu, že veľkosť populácie bola sémanticky zaznačená pre každé z týchto miest. Znovupoužitie údajov zo sémantických anotácií na inom mieste, napr. v hudobnom prehrávači, je možné vďaka jednoduchému exportu zachytenej ontológie v štandardnom formáte OWL/RDF. Existuje aj opačná cesta, importovať

dáta z externých OWL ontológií alebo FOAF dát a tieto následne popísať v ľudskej, zrozumiteľnej reči.

Napriek všetkým výhodám sa doposiaľ nepodarilo presadiť široké nasadenie rozšírenia Semantic MediaWiki na Wikipedii, no používa sa čoraz viac v menších príbuzných projektoch s víziou budúcnosti, v ktorej bude Semantic MediaWiki štandardnou súčasťou Wikipedie. Ak však chceme využívať, kombinovať, prepájať či overovať sémantické znalosti z Wikipedie už teraz, je potrebné tieto údaje získať z Wikipedie priamo. Mnohé faktografické informácie sú vo Wikipedii zapísané vo forme rámkov po pravej strane textu, tzv. infoboxov, kde sú v štruktúrovanej forme zhrnuté základné dáta ohľadom danej témy. Projekt DBpedia [2] tieto informácie z príspevkov získava a ukladá si ich vo formáte RDF, pre zaujímavosť v novembri roku 2009 obsahovala DBpedia 479 miliónov RDF tripletov. Podobný prístup bol použitý pri tvorbe ontológie YAGO, z angl. Yet Another Great Ontology [12]. Znalosti do nej boli získavané extrakciou údajov z Wikipedie a ich následným prepojením na lexikálnu databázu anglického jazyka WordNet [26]. V porovnaní s projektom DBpedia obsahuje síce YAGO len 2 milióny entít a 20 miliónov informácií, no presnosť zachytených znalostí je vyššia. Vhodnou kombináciou týchto a ďalších projektov zameraných na získavanie sémantických dát do ontológií môžeme strojovo spracovávať, kombinovať a využívať znalosti obsiahnuté vo Wikipedii a iných databázach už teraz.

Z vyššie uvedených príkladov je skutočne zrejmé, že koncepty Webu 2.0 a sémantického webu nie sú protichodné, ale sa dopĺňajú. Zatiaľ čo prepracované používateľské rozhrania Web 2.0 služieb ako sú blogy, či wiki stránky umožnili získanie väčšieho množstva štruktúrovaných dát od používateľov, sémantické technológie umožňujú tieto dáta spracovávať a ponúkať ich na ďalšie použitie v štandardných a otvorených formátoch, najčastejšie v podobe RDF/OWL ontológií, ktoré môžu byť následne využité na ďalšie účely v najrôznejších situáciách.

3.3 Sémantický desktop

Pojmom desktop, teda pracovná plocha, alebo desktopové prostredie zvykne označovať grafické používateľské rozhrania moderných osobných počítačov. Služi nám ako východiskové prostredie pre vykonávanie všetkých bežných činností akými sú tvorba dokumentov, správa súborov, prehliadanie multimédií, komunikácia alebo práca s organizérom a plánovanie. Charakteristické vzory interakcie medzi používateľom a desktopom využívajú pre zobrazovanie a manipuláciu s informáciami na ploche systém ikon, odkazov, menu a okien aplikácií ovládaných nejakým typom polohovacieho zariadenia, najčastejšie myši. K jej používaniu sa viaže sada špecifických konceptov, akými sú rôzne formy klikania, či funkcia drag'n'drop, ktoré majú zefektívniť, alebo priamo umožniť vykonávanie rozličných aktivít na desktope. Desktop vznikol začiatkom 70. rokov 20. storočia využitím metafory, teda prenesením aspektov skutočného pracovného stola a kancelárie do virtuálneho prostredia na monitore [7]. Počas 80. rokov sa podarilo tieto riešenia komerčne presadiť, vďaka čomu sa s nimi zoznámili milióny používateľov a v súčasnej dobe

patria k najrozšírenejším rozhraniam operačných systémov počítačov, ktoré máme v domácnostiach a na pracoviskách.

Obvykle používané desktopové prostredia však trpia nedostatkami, ktoré kladú prekážky prirodzenému spôsobu práce s informáciami. Ilustrujme si ich na príklade bežnej aktivity na desktope, akou je kategorizácia dát. Ak chce používateľ vytvoriť kategóriu pre určitú skupinu súborov, väčšinou ich uloží do výstižne pomenovaného adresára, ktorý umiesti na vhodné miesto v hierarchickej adresárovej štruktúre. Problém však nastáva, ak vznikne potreba prelínania kategórií, alebo konflikt v ich hierarchii. Stromové štruktúry diskových súborových systémov, ktoré sú neoddeliteľnou súčasťou desktopov poskytujú v takomto prípade riešenie v podobe tvorby symbolických odkazov na súbory, čo je však prácne a pri väčšom množstve kategórií a triedených súborov aj značne neprehľadné. Súbory, ktoré nie sú uložené na lokálnom disku nemá používateľ vôbec možnosť kategorizovať. Ďalší problém nastáva, ak jednotlivé kvantá dát, s ktorými pracujeme nie sú uložené v samostatných súboroch, ale sú napríklad agregované vo forme archívu elektronickej pošty. V rámci desktopu sú totiž určité kategórie implicitne vytýčené na základe formátu uloženia, ktorý definuje spôsob práce s dátami, pričom každý typ je spravovaný oddelene pomocou špecifického softvéru. Používateľ nemá k dispozícii efektívny nástroj, pomocou ktorého by mohol s informáciami pracovať v konkrétnych kontextoch ich významu, napríklad zaradiť zdrojové materiály, tímovú komunikáciu, kontakty a projektový plán do spoločnej kategórie a v rámci nej k nim jednotne pristupovať.

Problémom je aj tvorba doplnkových informácií k existujúcim dátam, teda anotácia. V bežnom desktopovom prostredí je takáto funkcia zväčša umožnená prostredníctvom špecifickej anotačnej aplikácie, týmto riešením však trpí následná dostupnosť, keďže všetky informácie sú uložené v špecializovanom formáte, nečitateľnom pre iné aplikácie. Desktop je navyše sociálne veľmi uzavretý svet. Poskytuje nám síce viacero možností komunikácie, no neposkytuje žiadny nástroj, ktorý by umožnil dvom, alebo viacerým používateľom dostatočne jednoducho spolupracovať na kolaboratívnej kategorizácii, alebo inej činnosti vykonávanej na desktope prostredníctvom internetu.

Sémantický desktop je jedným z pokusov o riešenie týchto problémov súčasných desktopových rozhraní. Označuje všeobecne pracovnú plochu počítača, alebo jej časť, obohatenú o technológie využívané v oblasti sémantického webu [10]. Jeho ambíciou je integrovať nielen dáta uložené na lokálnom počítači, ale pripájať aj informácie zo vzdialených zdrojov, dostupných prostredníctvom internetu. Vďaka sémantickým technológiám by mal byť používateľ schopný vytvárať ľubovoľné typy kategórií zo všetkých želaných druhov a formátov dát, ktoré sa navyše môžu voľne prekrývať. Oproti prostrediu, ktoré existuje na webe sa teda musia desktopové sémantické riešenia vysporiadať so širším rozpätím dátových formátov a typových kategórií. Sémantický systém však môže získavať informácie o vzťahoch medzi dátami aj samostatne, zaznamenávaním aktivít používateľa, ukladaním prídavných informácií ku všetkým dátam, ktoré používateľ uloží na lokálne záznamové médiá, alebo ku ktorým vzdialene pristupoval. Používanie sémantického desktopu ovplyvňuje aj informovanosť o sociálnych vzťahoch medzi používateľom a jeho okolím. Sémantické technológie umožňujú efektívne zaznamenávať komunikáciu, jej kontext a dátovú výmenu. Do

veľkej miery zjednodušujú špecifikovanie dát, ktoré máme záujem zdieľať so vzdialeným používateľom, či riešenie tímových úloh, pri ktorých je dôležitá čo najdokonalejšia synchronizácia aktivít a možnosť efektívne kooperovať.

Pre fungovanie takéhoto systému je nevyhnutné, aby sémantické dáta boli dostupné v každej chvíli všetkým aplikáciám, predpokladom čoho je aj ich jednotný formát. Táto sémantická databáza by mala obsahovať informácie o všetkých typoch zdrojov, mala by byť teda schopná pokryť aj atomicitu na subsúborovej úrovni t.j. na úrovni bajtov. Úspešnejšie projekty sémantického desktopu využívajú taxonomické štruktúry a ontológie vytvorené pomocou RDF resp. RDFs, pričom ako dopytovací jazyk sa využíva SPARQL. Použitie pôvodne webových technológií môže v budúcnosti uľahčiť integráciu s týmto prostredím a jeho špecifickými sémantickými aplikáciami.

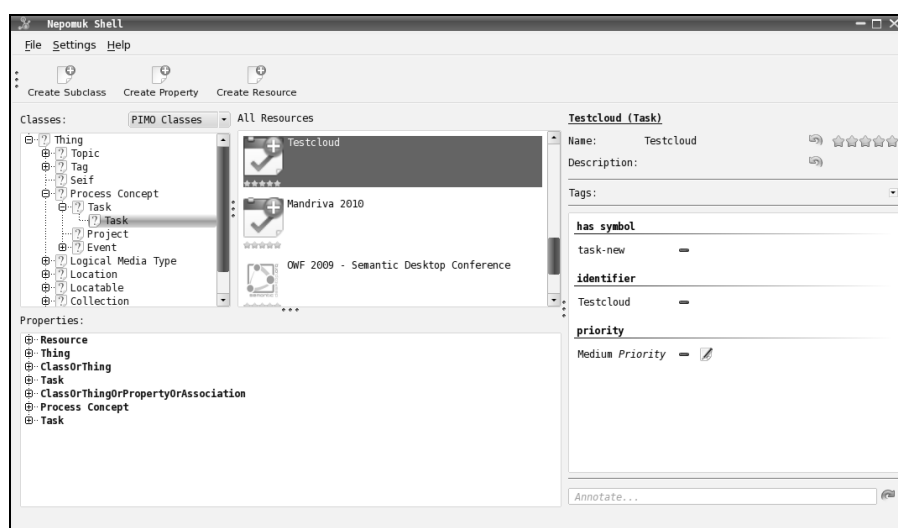
Ukážme si detailnejšie, akým spôsobom sa sémantické technológie v desktopovom prostredí môžu uplatniť. Základným kameňom týchto technológií je koncepcia metadát, teda dát, ktoré vypovedajú o iných dátach. Metadáta využívame pri práci s desktopom už aj dnes, zvyčajne ide o doplňujúce informácie k dátam, ktoré môžu špecifikovať ich pôvod, autora, čas a dátum vzniku a iné. Viaceré riešenia, ktoré sa pokúšajú o zjednodušenie a zníženie množstva úloh nutných k správe dokumentov, využívajú práve údaj o čase a dátume modifikácie v spojení so systémom pre správu verzií dokumentov. Takéto riešenia organizujú akoby tok dokumentov v čase [14], pričom je zachovaná perzistencia želaných verzií dát bez ohľadu na to, o aký typ dokumentu ide.

Predstavme si bežnú situáciu, pri ktorej používateľ chce preniesť svoju obľúbenú hudobnú nahrávku z CD média do počítača. Metadáta s menom interpreta, názvami piesní, žánrom, či inými špecifikáciami môžu vzniknúť dvoma spôsobmi. Používateľ sa môže rozhodnúť, že polia s údajmi vyplní ručne, ale môže využiť službu automatického rozpoznania nahrávky a údaje budú získané z verejne dostupnej databázy na internete. V prípade iných typov dát to môže byť podobné. Používateľ označí dokument značkou, teda tagom a vytvára navzájom sa prelínajúce kategórie, keďže jeden súbor môže byť označený ľubovoľným množstvom tagov. Takýto prístup nám však umožní iba triediť obsah podľa kľúčových slov. Na to, aby sme mohli informácie prepájať podľa významu potrebujeme zo značiek vytvoriť štruktúry, teda ontológie, ktoré umožnia usudzovať o kontextoch. Keby napríklad používateľ dostal elektronickou poštou dokumenty, ktoré potrebuje k aktuálnej úlohe, uloží ich na lokálny disk a sémantický agent súbory prepojí s doplnkovými informáciami, ako emailovou adresou autora, či samotným textom správy. V ideálnom prípade by sa používateľ vôbec nemusel zaujímať o kategorizáciu a správu dát, všetko by sa dialo na pozadí bežných činností. Hĺbka integrácie existujúcich riešení sémantického desktopu však zatiaľ takýto komfort neumožňuje.

V nasledujúcej časti si ukážeme, ako môže vyzeráť sémantické desktopové riešenie v praxi. Jedným z nich je NEPOMUK⁸, open source framework pre personalizovanú správu informácií a znalostí vytvorený podľa špecifikácie Social Semantic Desktop (obr. 8). NEPOMUK je pokusom o vytvorenie unifikovaného

⁸ <http://nepomuk.semanticdesktop.org/>

riešenia, teda metód, dátových štruktúr a nástrojov pre vytvorenie kolaboratívneho prostredia, ktoré okrem služieb sémantického desktopu prináša aj sociálny aspekt, výmenu informácií a sémantických dát cez sieť. Tieto vlastnosti umožňujú spoločnú tvorbu znalostí a ich štruktúr v zdieľaných ontológiách. NEPOMUK nie je samostatnou desktopovou aplikáciou, ale skôr komplexne štruktúrovanou sémantickou databázou, pomocou ktorej medzi sebou ostatné aplikácie komunikujú. Základom tohto systému je NRL teda NEPOMUK's Representational Language. Je to metajazyk, ktorý zabezpečuje nástroje pre správu a interpretáciu zdrojov a vytvára vhodné podmienky pre inferenciu. Pomocou neho je možné špecifikovať vlastnosti objektov a štruktúr v ontológiách, ako napríklad požiadavky na kardinalitu, doménu, alebo rozsah pôsobenia. NRL je jazyk vytvorený na báze RDF. Základnými konceptmi v NRL sú RDF triplet, pomenovaný graf, rola grafu a grafové náhľady, teda graph views. Pomenovaný graf je RDF graf označený menom, pričom v NRL musia byť všetky RDF triplety priradené k nejakému pomenovanému grafu. Rola grafu charakterizuje konkrétny pomenovaný graf z hľadiska jeho účelu a obsahu, zároveň je tiež pomenovaným grafom. Keďže interpretácia konkrétneho grafu by mala ostávať vždy rovnaká kvôli zachovávaniu integrity ontológie, náhľady slúžia pre rôzne interpretácie v odlišných situáciách. Vytvárajú sa prostredníctvom pridávania, alebo odoberania RDF tripletov vzhľadom k pôvodnému pomenovanému grafu. Aby sa zabránilo možnému zaneseniu nepresných, alebo kontradiktívnych informácií do ontológií, usudzovanie nad štruktúrami NRL prebieha s predpokladom uzavretého sveta.

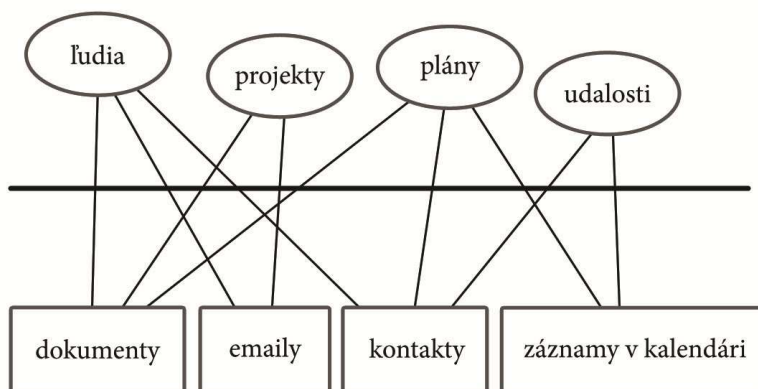


Obr. 8. Prehľadanie obsahu ontológií pomocou Nepomuk Shell.

NRL nám umožňuje definovať konkrétne ontológie a ich špecifické vlastnosti. V rámci NEPOMUK desktopu existuje niekoľko samostatných ontológií. Časť z nich je určená na reprezentáciu abstraktných entít, akými sú projekty, plány a osobné informácie všeobecne. Druhá časť sústreďuje pozornosť na natívne štruktúry, teda

používateľské dáta, dokumenty, alebo ich časti, ktoré je možné fyzicky vymedziť v bajtoch. Ontológia NIE, teda NEMPOMUK Information Element, zabezpečuje správu a spracovávanie týchto fyzických zdrojov. Jej základnou myšlienkou je oddelenie konceptov reprezentácie dátových entít od ich interpretácie. Interpretáciou rozumieme určitú informáciu, napríklad kontaktné údaje Isaaca Newtona. Táto informácia je uložená v nejakom dátovom objekte, ktorý môže mať rôzne vlastnosti a nie vždy je priamo súborom vhodne uloženým v štruktúre adresárov na disku. NIE preto umožňuje z používateľského hľadiska zjednotiť rôzne dátové objekty a okrem tradičných lokálnych, či webových dokumentov reprezentovať aj iné typy dát, akými sú položky v databáze, alebo prílohy ukryté v archíve elektronickej pošty. Každá z týchto entít môže byť interpretovaná ako kontakt a túto informáciu je možné bližšie charakterizovať kľúčovými slovami, alebo inými doplnkovými informáciami o jej význame, napríklad že ide o kontaktné údaje pána Newtona do jeho kancelárie na Univerzite v Cambridge. NIE sa skladá z jadra a čiastkových ontológií, ktoré sú určené pre konkrétne dátové zdroje, ako napríklad ontológia NFO, ktorá pracuje so súborami, NCO pre spracovávanie kontaktov, alebo NCAL spravujúca záznamy v kalendári.

osobné informácie



natívne štruktúry

Obr. 9. Informácie spracovávané desktopom.

PIMO je skratkou pre Personal Information Modeling, ide teda o ontológiu popisujúcu informácie s ktorými bežne pracujeme v rámci organizérov a príbuzných aplikácií. Umožňuje definovať agentov, stretnutia, udalosti, projekty, témy a vytvárať medzi nimi vzťahy ako účastník stretnutia, alebo organizátor. Princíp fungovania tejto ontológie je veľmi podobný ako v prípade NIE. Máme k dispozícii určité preddefinované entity, teda napríklad triedu pre agenta reprezentujúceho nejakého

človeka a tejto entite môžeme priradiť vlastnosti reprezentované vzťahom medzi našou pôvodnou a nejakou ďalšou entitou. Hoci nepatrí priamo do dvoch hlavných podskupín, NAO, teda NEPOMUK Annotation Ontology je veľmi dôležitou súčasťou celého systému. Jej hlavnou úlohou je umožniť používateľom a aplikáciám anotovať všetky druhy zdrojov v ostatných ontológiách. Anotácia môže mať rôznu podobu, jednak môže ísť o jednoduché označovanie zdrojov kľúčovými slovami v podobe tagov a ohodnocovanie ich kvality podľa rozličných stupníc, avšak prichádza do úvahy aj možnosť využiť špecifickejšie triedy a vlastnosti, napríklad definovať priamo autora nejakého zdroja, tému, či označiť vzťah medzi dvoma zdrojmi. Táto ontológia je rovnako ako všetky v rámci NEPOMUK je vytvorená pomocou NRL, funguje preto veľmi podobne ako ontológie popísané vyššie.

Skúsme si teraz uviesť príklad, ako by mohlo vyzerat' praktické použitie takejto štruktúry ontológií. Nech `pimo:IsaacNewton` je inštanciou triedy `pimo:Person`, ktorá je podtriedou triedy `pimo:Agent`. Ďalej nech `pimo:RoyalSocietyAnnual` je inštanciou triedy `pimo:SocialEvent`. Pomocou definície vzťahu `pimo:attends`, ktorého rozsahom pôsobenia sú inštancie `pimo:SocialEvent` a doménou sú inštancie `pimo:Person` môžeme vyjadriť informáciu, že Sir Newton je tento rok medzi účastníkmi každoročnej konferencie Kráľovskej spoločnosti. Predstavme si ďalej, že niekde v štruktúre adresárov na disku máme uložený súbor `Principia.pdf`. Tento súbor je označený ako `nfo:PaginatedTextDocument`, čo znamená, že ide o textový dokument s jasne odlíšenými stranami definovaný ontológiou NFO. Vďaka tranzitívnosti vzťahov v hierarchii tried zistíme, že táto trieda je podtriedou `nie:InformationElement`, teda základného elementu interpretácie dát. V skutočnosti tieto entity so súborom spája vlastnosť `nie:isStoredAs` vzťahujúca sa k `nie:DataObject`, reprezentujúcim daný súbor. Tento dokument si môžeme ďalej anotovať a vytvoriť referenciu pomocou vlastnosti `nao:hasReference`, ktorá sa môže vzťahovať k ľubovoľnému zdroju, v tomto prípade k `pimo:IsaacNewton`.

Projekt NEPOMUK je veľmi dôležitým krokom v oblasti sémantického desktopu nielen kvôli dosiaľ uvedeným vlastnostiam. Je totiž súčasťou softvérového balíka KDE, ktorý tvorí jedno z najvýznamnejších desktopových prostredí pre Linux a unixové systémy čím výrazne napomáha použiteľnosti celého systému v praxi. NEPOMUK sa v KDE aktivuje veľmi jednoducho pomocou tlačidla Desktop search. Praktické využívanie možností, ktoré táto funkcia ponúka je však len v začiatkoch, keďže aplikácií, ktoré sú schopné využívať jeho potenciál je zatiaľ málo. V aktuálnej verzii KDE je NEPOMUK schopný spolupráce so správcom súborov Dolphin, vyhľadávacou a indexovaciou aplikáciou Strigi a správcom úloh Scribo. Integrácia s Dolphin umožňuje používateľovi hodnotiť, označovať, či komentovať súbory a následne podľa týchto metadát aj vyhľadávať, teda pomocou dopytov zostavovať akési dočasné virtuálne adresáre. Pomocou Strigi je možné automatizovane získavať rôzne metadáta k súborom, Scribo zasa umožňuje vytvárať kontexty aktuálnych úloh a v rámci nich spravovať zdroje a dokumenty. K dispozícii je ešte aj aplikácia Nepomuk Shell, ktorá slúži ako klientská aplikácia nad databázami ontológií.

Alternatívou ku KDE implementácií môže byť Gnowsis⁹, avšak táto implementácia zďaleka nedisponuje takou podporou v rámci nejakého desktopového prostredia. Iným príkladom rozvíjajúceho sa projektu v rámci širokého záberu sémantického desktopu je aj vyššie spomenutý Haystack [27]. Ide o jeden z pôvodných projektov tohto druhu, na rozdiel od NEPOMUK však nejde o konkrétny softvérový balík, ale súbor rozličných sémanticky obohatených aplikácií sústredených okolo špecifického organizéru, Haystack klienta. Využíva vlastný programovací jazyk Adenine na báze RDF a Java, ktorý je uspôsobený pre prácu s RDF štruktúrami a v istom zmysle reprezentuje tendenciu v zjednocovaní používania tradične oddelených webových a desktopových technológií. Súčasné trendy vo vývoji hardvéru a mobilných komunikačných platforiem ešte viac zvyrazňujú tendenciu desktopu stať sa skutočne jednotným multifunkčným a centralizovaným informačným rozhraním, uspôsobeným presne na mieru konkrétnemu jednotlivcovi. Zdieľanie týchto používateľských profilov môže postupne vytvoriť skutočnú sémantickú sociálnu sieť a zároveň do nej pootvorí dvere aj umelým entitám a agentom schopným porozumieť dokonale našim individuálnym potrebám.

4 Záver

Víziu sémantického webu [6] je potrebné chápať predovšetkým ako obrovskú výskumnú výzvu. Cieľom tejto výzvy je premeniť web, ktorý dnes vnímame predovšetkým ako obrovskú množinu informácií, prístupných pre konzumáciu ľudským používateľom, ktorí sa v takejto obrovskej množine dát neraz strácajú, na prostredie nového typu, v ktorom sú informácie v oveľa väčšej miere prístupné aj na strojové spracovanie. Takáto zmena umožní ľudským používateľom pristupovať k webu novým, oveľa efektívnejším spôsobom, práve vďaka novým softvérovým agentom a webovým službám, ktoré budú pracovať s informáciami oveľa efektívnejšie.

K plnej realizácii celej vízie sémantického webu je ešte ďaleko, a sémantický web nás zatiaľ nezasiahol a neovplyvnil naše životy, tak ako to bolo v prípade webu v deväťdesiatych rokoch minulého storočia. Napriek tomu možno konštatovať, že v tejto oblasti došlo k výraznému pokroku. Podarilo sa vytvoriť pevnú nosnú konštrukciu v podobe technologickej infraštruktúry, ktorú predstavujú sémantické technológie ako RDF, RDFs a OWL, s ktorými sme sa zoznámili v tejto kapitole. Pomocou týchto technológií je možné publikovať dáta a vytvárať ontológie v jazyku s presnou logickou sémantikou, či vytvárať anotácie k existujúcim webovým dokumentom. Bol tiež položený základ pre technologickú infraštruktúru v oblasti webových služieb [13]. Na týchto technológiách možno už dnes stavať najrôznejšie aplikácie spracúvajúce sémantické dáta spôsobom, ktorý je veľmi blízky tomu, ako to bolo opísané v pôvodnej vízii sémantického webu.

V druhej časti sme sa venovali trom vybraným aktuálnym výskumným výzvam v tejto oblasti, ktoré považujeme za mimoriadne zaujímavé. Ide o vybudovanie infraštruktúry pre reprezentáciu skutočne distribuovaných ontológií, ktoré sú oveľa lepšie uspôsobené potrebám sémantického webu. Ďalšou výzvou je konfrontácia

⁹ <http://www.gnowsis.org/>

prístupu sémantického webu a zdanlivo konkurenčnej idey označovanej ako Web 2.0, kde sa ukazuje, že obe tieto idey majú zaujímavé styčné body a môžu jedna druhú výrazne obohatiť. Okrem toho sme sa pozreli na problém sémantického desktopu, teda aplikácie sémantických technológií v desktopových pracovných prostrediach. Takýto prístup môže napomôcť efektívnejšej správe používateľských dát, a tiež postupne zmazať rozdiely, ktoré vznikli rokmi konceptuálneho oddelovania priestoru lokálneho počítača a vzdialených uzlov na sieti.

Výskum v oblasti sémantického webu je však oveľa širší, než ako bolo možné zmapovať v tejto kapitole. Medzi oblasti, ktorých sme sa nedotkli patria: webové služby, dotazovanie sa nad sémantickými dátami, manažment dôvery, vysporiadanie sa s vágnosťou a neurčitnosťou, aplikovanie pravidlových systémov na sémantickom webe, či idea ambientnej inteligencie, ale aj mnohé iné. Viac sa čitateľ môže dozvedieť z nespočetnej literatúry, ktorá v tejto oblasti existuje (napr. [1, 3, 9, 11, 13, 17, 30]).

Podakovanie: Tento príspevok vznikol v Centre kognitívnych vied KAI, FMFI UK v Bratislave a bol podporený KEGA MŠ SR, projekt č. 3/7300/09.

Literatúra

- [1] Antoniou, G., van Harmelen, F. A Semantic Web Primer. MIT Press, 2nd edition, 2008.
- [2] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z. DBpedia: A Nucleus for a Web of Open Data. In: The Semantic Web. LNCS 4825, pp. 722-735, Springer; 2008
- [3] Augusto, J.C., Shapiro, D. Advances in Ambient Intelligence. IOS Press, 2007.
- [4] Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. The Description Logic Handbook. CambridgeUniversity Press, 2003.
- [5] Bao, J., Voutsadakis, G., Slutzki, G., Honavar, V. Package-Based Description Logics. In: Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization. LNCS 5445, 349-371. Springer, 2009.
- [6] Berners-Lee, T., Hendler, J., Lassila, O. The Semantic Web. Scientific American 284(5), 34-43. 2001.
- [7] Blackwell, A. F. The reification of metaphor as a design tool. ACM Trans. Comput.-Hum. Interact. 13, 4 (Dec. 2006), 490-530, 2006.
- [8] Borgida, A., Serafini, L. Distributed Description Logics: Assimilating Information from Peer Sources. Journal of Data Semantics 1, 153-184. 2003
- [9] Cardoso, J., Sheth, A.P. Semantic Web Services, Processes and Applications. Springer, 2006.
- [10] Decker, S., Frank, M.: The social semantic desktop. WWW2004 Workshop Application Design, Development and Implementation Issues in the Semantic Web. 2004.

- [11] De Virgilio, R., Giunchiglia, F., Tanca, L. *Semantic Web Information Management: A Model-Based Perspective*. Springer, 2010.
- [12] Fabian M. Suchanek, Gjergji Kasneci and Gerhard Weikum, "Yago – A Core of Semantic Knowledge", 16th international World Wide Web conference, WWW 2007.
- [13] Fensel, D., Lausen, H., Polleres, A., Bruijn, J., Stollberg, M., Roman, D., Domingue, J. *Enabling Semantic Web Services*. Springer, 2007.
- [14] Freeman, E., Gelernter, D. *Lifestreams: a storage model for personal data*. SIGMOD Rec., 25(1), 80-86, 1996.
- [15] Golbreich, C., Wallace, E.K., *OWL 2 Web Ontology Language New Features and Rationale*. W3C Recommendation, 2009.
- [16] Hayes, P. *RDF Semantics*, W3C Recommendation, 2004.
- [17] Hitzler, P., Rudolph, S., Kroetzsch, M. *Foundations of Semantic Web Technologies*. Taylor & Francis, 2009.
- [18] Horrocks, I., Kutz, O., Sattler, *The even more irresistible SROIQ*. Procs. of KR 2006.
- [19] Jacobs I. and Walsh N. (eds.). *Architecture of the World Wide Web, Volume One*. 2004. <http://www.w3.org/TR/webarch/>
- [20] Karger, D.R., Quan, D. *What Would It Mean to Blog on the Semantic Web? The Semantic Web – ISWC 2004*, LNCS 3298, pp. 214-228, Springer, 2004.
- [21] Krötzsch, M., Vrandečić, D., Völkel, M. *Semantic MediaWiki*. In: *The Semantic Web - ISWC 2006*, LNCS 4273, pp. 935-942, Springer; 2006.
- [22] Kutz, O., Lutz, C., Wolter, F., Zakharyashev, M. *E-connections of abstract description systems*. *Artificial Intelligence*, 156(1), 1-73. 2004.
- [23] Ling Liu, M. Tamer Özsu (eds.). *Encyclopedia of Database Systems*. Springer, 2009
- [24] McGuinness, D.L., van Harmelen, F. *OWL Web Ontology Language Overview*. W3C Recommendation, 2004.
- [25] Mika, P. *Social Networks and the Semantic Web*. Springer, 2007.
- [26] Miller, G. A., Beckwith, R., C. D. Fellbaum, D. Gross, K. Miller. *WordNet: An online lexical database*. *Int. J. Lexicograph.* 3, 4, pp. 235-244, 1990.
- [27] Quan, D., Huynh, D., Karger, D. R.: *Haystack: A Platform for Authoring End User Semantic Web Applications*. *International Semantic Web Conference 2003*, pp 738-753, 2003.
- [28] Rospocher, M., Ghidini Ch., Pammer, V., Serafini, L., Lindstaedt, S., *MoKi: the Modelling wiKi*, In *Proceedings of the SemWiki2009 - Fourth Workshop on Semantic Wikis*, Hersonissos, Heraklion, Crete, Greece, 2009
- [29] Serafini, L., Homola, M., *Modular Knowledge Representation and Reasoning in the Semantic Web*. In: *Semantic Web Information Management: A Model-Based Perspective*. Springer, 2010.

-
- [30] Staab, S., Studer, R. Handbook on Ontologies. Springer, 2004.
 - [31] ter Horst, H.J. Completeness, decidability and complexity of entailment for RDF Schema and a semantic extension involving the OWL vocabulary. Journal of Web Semantics 3(2-3), 79-115, 2005.
 - [32] van Harmelen, F., Lifschitz, V., Porter, B. Handbook of Knowledge Representation. Elsevier Science, San Diego, USA, 2007.
 - [33] von Ahn, L., Dabbish, L. Labeling Images with a Computer Game. ACM Conference on Human Factors in Computing Systems, pp 319-326. CHI 2004.
 - [34] Zimmermann, A. Integrated Distributed Description Logics. Procs. of DL-2007. CEUR-WS.org 250. 2007.

Teória evolúcie a filozofia

Dušan GÁLIK¹

Abstrakt. Teória evolúcie je jednou kľúčových teórií súčasnej vedy. Cieľom kapitoly je oboznámiť čitateľa so stručným vývojom evolučného myslenia s dôrazom na vývoj teórie evolúcie ako vedeckej teórie, s jej štruktúrou a postavením v systéme vied, ako aj s teoretickými problémami, ktoré svojím vývojom spôsobila a spôsobuje, a so spôsobmi ich riešenia. Dôraz sa bude klásť na tie problémy, ktoré súvisia s prekračovaním čisto biologického či prírodovedného skúmania a s prerastaním teórie vysvetľujúcej pôvodne len vznik biologických druhov do univerzálnej teórie evolúcie, predovšetkým na problém vzťahu prírodných vied na jednej, sociálnych a humanitných na druhej strane.

1 Úvod

Teória evolúcie sa v dvadsiatom storočí stala kľúčovou teóriou súčasnej vedy. Tvorí teoretický základ nielen pre vysvetľovanie javov v prírodnom svete, ale v poslednej tretine dvadsiateho storočia a na začiatku 21. storočia začala výrazne prenikať na pôdu spoločenských a humanitných vied. To prirodzene vyvolalo búrlivé reakcie z ich strany. Hoci sa niektoré spoločenské vedy svojou metodológiou približujú prírodným vedám (a toto približovanie bude v budúcnosti nevyhnutnosťou), spoločenské a humanitné vedy sa v zásade opierajú o iný teoretický a metodologický základ ako prírodné vedy. Nesmierny význam teórie evolúcie spočíva v odhaľovaní „mechanizmov“ evolučných procesov na rozličných úrovniach organizácie, vrátane ľudskej spoločnosti a takých vlastností, ktoré pokladáme za výlučné schopnosti človeka (sociálne správanie, kognitívne schopnosti, ľudská myseľ, vedomie, „duch“), čo vedie k radikálnej zmene v chápaní postavenia človeka v prírodnom (prirodzenom) dianí, ale malo by viesť (a v rámci mnohých prúdov evolučného myslenia aj vedie) k prehodnoteniu antagonistického vzťahu spoločenských a humanitných vied na jednej a prírodných vied na druhej strane. Jedným z predpokladov je odstránenie existujúcich nedorozumení, ktoré vyplývajú z nepochopenia, ale aj z nedostatočného a vágneho vymedzenia základných termínov, ako aj uvedomenie si ich podmienenosti určitým širším kultúrnym (filozofickým) prostredím.

¹ Filozofický ústav Slovenskej akadémie vied, Klemensova 19, 813 64 Bratislava, Dusan.Galik@savba.sk

2 Význam teórie evolúcie, jej postavenie v systéme vied

Slová (výrazy) *evolúcia* a *teória evolúcie* sa v priebehu minulého storočia, najmä v jeho druhej polovici, stali bežnou súčasťou jazyka vedy. Používajú sa s akousi samozrejmosťou, s intuitívnym presvedčením o ich jednoznačnosti, v ktorom sa odráža samozrejmosť faktu, že vesmír, ktorého sme súčasťou, náš svet, nejakým spôsobom vznikol, prešiel a prechádza postupnosťou zmien. Spôsob ich používania, predovšetkým používania výrazu *evolúcia*, sa však vyznačuje prekvapujúcou rozmanitosťou a nejednoznačnosťou, v ktorej sa odráža rozmanitosť toho, čo pod *evolúciou* chápeme.

2.1 Čo je to evolúcia

Termín *evolúcia* sa vo vede používa v dvoch základných významoch. V užšom význame ako označenie procesov vzniku, pretrvávania, premien a zániku druhov živých organizmov, a v širšom význame na označenie vzniku, pretrvávania, premien a zániku rozličných druhov súcien, od neživých súcien cez druhy živých organizmov až po rozmanité druhy „nadbiologických“ (sociálnych, ekonomických, kultúrnych atď.) štruktúr. Kým v druhom, širšom význame, sa termín *evolúcia* týka všetkých druhov súcien, všetkého, čo „obsahuje“ známy vesmír, vrátane vesmíru (sveta, bytia) ako takého, v prvom význame sa týka len určitého vymedzeného druhu súcien, živých organizmov, ktoré možno charakterizovať pomocou niekoľkých základných vlastností, akými sú reprodukcia, autoregulácia, premena látok a energií, výmena látok, energií a informácií s prostredím. Termín *evolúcia* sa do vedy dostala najmä prostredníctvom teórie evolúcie živých organizmov. Hoci evolúcia živých organizmov predstavuje len jeden z možných prípadov evolúcie sebareprodukujúcich sa štruktúr (o ktorej sa hovorí aj ako o *evolúcii s pamäťou* [30], [93]), je pochopiteľné, že *evolúcia* sa veľmi často stotožňuje s *evolúciou živých organizmov*, s *biologickou evolúciou*.

Hoci intuitívne by sa mohlo zdať, že ak hovoríme o evolúcii, máme na mysli približne to isté (napokon, *evolúcia* nie je nič iné ako *vývoj* označený cudzím slovom), že teda používame výraz *evolúcia* viac-menej v rovnakom význame, v skutočnosti sa v súčasnej vede používa niekoľko významovo odlišných pojmov evolúcie. Akýkoľvek pokus o odstránenie tejto viacznačnosti a o presné vymedzenie v zmysle požiadaviek vedeckej exaktnosti pritom naráža na viac otvorených otázok ako jednoznačných riešení.

Nejednoznačnosť významu termínu *evolúcia* má viacero príčin. Jednou z nich môže byť spôsob, akým bolo latinské *evolutio* zavedené do vied o živej prírode. *Evolutio* je odvodené z latinského *evolvere*, čo znamená *rozbaľiť*, *rozvinúť*, *rozvíjať*. Už pri interpretácii pôvodného významu narazíme na nejednoznačnosť – kým niektoré zdroje uvádzajú, že *evolutio* sa pôvodne používalo vo význame *rozbaľiť*, *rozvinúť* *zvitok (knihu)* [48], [105] a v tomto zmysle sa niekedy pripisuje Cicerovi [27], iné zdroje uvádzajú jeho používanie vo všeobecnejšom význame ako *rozbalenie*, *rozvinutie*, *rast častí už existujúcich v kompaktnej forme* [8], [9], [42], teda ako rozvinutie vopred danej a hotovej kompaktnej štruktúry, jej smerovanie k vopred

danému cieľu. V [93] sa ako prvý pravdepodobný výskyt uvádza až dielo *De docta ignorantia* významného učenca prelomu stredoveku a renesancie Mikuláša Kuzánskeho.² V anglickom jazyku, z ktorého sa toto slovo dostalo do ostatných jazykov, sa prvý výskyt slova *evolution* kladie do prvej polovice 17. storočia (Harper [48] pritom uvádza rok 1622), pričom je zaujímavé, že sa používa v užšom význame *rozbaľiť zvitok, knihu*.

Vo vedách o živej prírode sa výraz *evolúcia* v podobe latinského *evolutio* objavuje v 18. storočí, nie však v anglickej, ale v kontinentálnej prírodovednej literatúre. Ani tu však nepanuje zhoda – kým podľa niektorých zdrojov bol prvým autorom francúzsky prírodovedec Charlesa Bonnet, ktorý ho použil v roku 1762 vo svojom diele *Considérations sur les corps organisés* (napr. [25]), podľa iných to bolo ešte v roku 1744, pričom autorom má byť švajčiarsky prírodovedec, zakladateľ experimentálnej fyziológie Albrecht von Haller [42]. V čom sa však zhodujú, je spôsob, akým obaja prírodovedci latinské *evolutio* používajú – nie v súvislosti s vývojom druhov živých organizmov, ale pri opise embryonálneho vývoja – v jeho pôvodnom latinskom význame *rozvinutie, rast už existujúcich hotových štruktúr*. Tak Haller ako aj Bonnet patrili medzi zástancov tzv. preformacionizmu (hoci Haller sa neskôr priklonil k epigenéze [89], z ktorej sa vyvinula súčasná teória embryogenézy), názoru, že vývoj embrya spočíva v raste malého homunkula obsiahnutého vo vajíčku, pričom každý homunkulus predstavuje zmenšeninu dospelého človeka, obsahuje všetky orgány síce v hotovej, ale v nerozvinutej forme. Každý homunkulus pritom nesie vajíčko s ďalším homunkulom.

Po úpadku preformacionizmu na sklonku 18. storočia sa výraz *evolúcia* ešte používal na opis individuálneho vývoja organizmov (zrejme vďaka tomu, že *evolúcia* v klasickom zmysle slova znamená aj smerovanie k vopred danému cieľu). V 19. storočí sa však už začína používať aj vo význame, ktorý je blízky modernému, teda ako vývoj zložitejších štruktúr z jednoduchších, vznik, pretrvávajúce a zánik kvalitatívne odlišných druhov súcién vrátane druhov živých organizmov. O jeho zavedenie a rozšírenie v tomto zmysle sa pričínil anglický filozof Herbert Spencer [9].

Vo rodiacej sa teórii evolúcie sa termín *evolúcia* nepresadzoval ľahko. Zástancovia myšlienky premenlivosti biologických druhov vrátane Darwina sa termínu *evolúcia* spočiatku vyhýbali³, čiastočne vzhľadom na jeho používanie v embryológii (a to tak vzhľadom na preformacionizmus ako aj na smerovanie vývinu k vopred danému cieľu), čiastočne preto, lebo Darwin pokladal Spencerovu teóriu za špekulatívnu a odmietal aj spájanie vývoja s nevyhnutnosťou progresívneho vývoja. Darwin používal na označovanie procesov premeny druhov v tom čase obvyklé výrazy *transmutácia*, *transformácia*, o svojej teórii hovoril ako o *teórii pôvodu (vzniku) s modifikáciou*. V druhej polovici 19. storočia však boli Spencerove myšlienky medzi prírodovedcami

2 Kuzánsky používa tento termín v desiatej kapitole druhej knihy, keď hovorí o názoroch iných autorov na pohyb planét, ktorý je rozvinutím prvého pohybu a na pohyb pominuteľných a pozemských vecí, ktorý je rozvinutím pohybu planét. Ide o rozvinutie pohybu, ktorý už je obsiahnutý v svetovom duchu [59]. Aj tu ide o *evolúciu* v zmysle rozbalenia už hotovej, preexistujúcej štruktúry.

3 Výraz *evolution* sa v prvom vydaní *Pôvodu druhov* nevyskytuje vôbec, po prvý raz objavuje až v šiestom vydaní, aj to len na jednom mieste.

natol'ko populárne, že sa Darwin neskorších dielach, napríklad v *Pôvode človeka* [darwin 1871], priklonil k používaniu výrazu *evolúcia*.

Azda ešte viac ako pôvodný význam determinujú význam a spôsob používania termínu *evolúcia* v súčasnej vede filozofické teórie vstupujúce do vytvárania a interpretácie vedeckej teórie, teda to, čo sa označuje ako jej filozofické či metafyzické „pozadie“. *Evolúcia* vstrebala myšlienku nevyhnutnosti pokroku, ktorej základy sa objavujú vo filozofii francúzskeho osvietenstva, aby svojím pozitívnym duchom, vierou v pozitívne dôsledky progresívneho vývoja prenikla do dominujúcich smerov európskeho filozofického myslenia a do európskej vedy. V Británii sa určujúcim interpretačným rámcom vedeckého poznania stala Spencerova teória univerzálnej evolúcie, svojou myšlienkou o nevyhnutnosti smerovania evolúcie k čoraz vyšším, dokonalejším formám korešpondujúca s britským duchom pokroku zrodenom v priemyselnej revolúcii 18. storočia. V nemecky hovoriacich krajinách sa pretavila do nemeckej naturfilozofie a klasickej nemeckej filozofie vrcholiacej v Heglovom systéme, v ktorom je – podobne ako neskôr u Spencera – nevyhnutnosť progresívneho charakteru evolúcie (vývoja) sformulovaná v podobe imanentného univerzálneho zákona vývoja [49].

V našom vedeckom a kultúrnom prostredí, výrazne determinovanom štyridsaťročnou neprirodzenou, ideologicky nanútenou a deformujúcou hegemoniou derivátu heglovstva – marxistickej filozofie, možno silný vplyv tejto tradície ilustrovať na slovníkových definíciách *evolúcie*. Pri nahliadnutí do niekoľkých slovníkov neprekvapí, že evolúcia sa definuje ako postupný vývoj smerujúci k vyšším, vyvinutejším, zložitejším formám, štruktúram. *Krátky slovník slovenského jazyka* [58] vymedzuje evolúciu ako *postupnú kvalitatívnu zmenu, vývin, vývoj*⁴. *Slovník súčasného slovenského jazyka* [11] ako *proces postupnej zmeny živých a neživých systémov (jednotlivca, druhu, celej prírody či vesmíru), pomalý kontinuálny vývoj*. Vývoj sa charakterizuje ako *postupný rad zmien smerujúci obyčajne k zložitejšej a dokonalejšej forme* [92], prípadne *proces, ktorý vedie obyčajne k zložitejším a dokonalejším formám* [97].

Stotožňovanie evolúcie s nevyhnutnosťou progresívneho vývoja je ešte jednoznačnejšie pri užšom chápaní evolúcie, vo význame biologickej evolúcie. Podľa [91] je biologická evolúcia *vývoj ako proces postupného zdokonaľovania živej hmoty a foriem života od jednoduchých organizmov k čoraz zložitejším, od nižších k vyšším*. V encyklopédii *Beliana* sa biologická evolúcia charakterizuje ako *vývojový proces postupného zdokonaľovania foriem života, postupná premena štruktúrne a funkčne jednoduchších organizmov na štruktúrne a funkčne zložitejšie organizmy, začínajúc najprimitívnejšími prokaryontnými praorganizmami (archeónmi) a vrcholiac vývojom človeka* [5].

Hoci by sa mohlo zdať, že progresivistické chápanie evolúcie (podľa ktorého evolúcia **je** progresívny vývoj, pričom ide o proces, ktorý má podobu nevyhnutnosti, zákona, navyše ide o proces smerujúci k určitému završeniu, cieľu) je prirodzeným

4 Pokusy vymedziť *evolúciu* prostredníctvom *vývoja* nie sú práve najšťastnejšie. Tieto slová sú skôr synonymá, definícia jedného prostredníctvom druhého je definícia kruhom, ako také je prázdna.

dôsledkov faktov, že priamo vyplýva zo získaných poznatkov o svete, v skutočnosti je determinované spomínaným filozofickým pozadím, interpretačným prostredím vedeckej teórie. Existuje však aj iné chápanie, ktoré nepopiera skutočnosť, že v evolúcii dochádza k procesom, ktoré možno označiť za progresívne, popiera však nevyhnutnosť progresívneho charakteru evolúcie. Podľa týchto predstáv je evolúcia len realizáciou niektorých z možností, pričom kľúčovú úlohu v ďalšom smerovaní evolučného procesu hrá vplyv náhodných faktorov.

V súvislosti s chápaním významu termínu *evolúcia* treba upozorniť ešte na jeden problém, o ktorom hovorí napr. Fajkus [28], totiž na otázku, čo vlastne chápeme pod evolúciou v širokom slova zmysle – máme na mysli jeden proces, ktorý je základom procesov vytvárania kvalitatívne nových štruktúr na rozličných úrovniach evolúcie, či dokonca vytvárania kvalitatívne rozličných úrovní evolučného procesu, alebo hovoríme o niekoľkých procesoch, navzájom od seba nezávislých, neopísateľných jedným či niekoľkými univerzálnymi zákonmi? Ak však evolúcia nie je univerzálnym procesom, môžeme hovoriť o evolúcii v najširšom slova zmysle? Nejde v prípade pokusov o univerzalizáciu *evolúcie*, pričom by malo ísť najmä o expanziu, či skôr o extrapoláciu tohto termínu z biologických vied do vied o neživej prírode a do spoločenských a humanitných vied (o extrapoláciu, ktorá vychádza z darvinistického chápania evolúcie v živej prírode), len o neodôvodnenú infláciu tohto termínu, a teda o jeho faktické vyprázdnenie?

Domnievam sa, že takýto názor sa opiera o nesprávny argument. Napriek tomu, že evolúcia v neživej prírode a evolúcia v živej prírode (vrátane evolúcie človeka, ľudskej spoločnosti) predstavujú zásadne odlišný typ evolučného procesu, neexistuje dôvod, prečo by sme procesy na rozličných úrovniach organizácie nemohli označovať rovnakým termínom. Dostaneme sa síce do rozporu s už spomínanou filozofickou tradíciou, a to hneď dvomi spôsobmi – jednak v otázke jedného univerzálného zákona evolúcie (vývoja), jednak v otázke jedného pojmu evolúcie, tento rozpor však môže viesť k plodnému rozchodu s tradíciou, podľa ktorej je pojem nejakého predmetu vymedzením jeho podstaty (momentom stávania pojmu sú bytie a podstata [51], pojem je živlom existencie pravdy [49]; mať pojem predmetu znamená preniknúť k jeho podstate [29], [14], [15]), a k nadviazaniu na tradíciu, podľa ktorej pojem je význam nejakého jazykového výrazu [32], [13], [67]. Môžeme mať rozličné *pojmy evolúcie*, našou úlohou je spresňovať (explikovať, ako by povedal Carnap) ich význam, prehľbovať naše poznanie o evolúcii tak, aby sme vedeli, o čom hovoríme.

2.2 Teória evolúcie – status a štruktúra

Cieľom vedeckého poznania je dosiahnutie pravdivého poznania (opisu a vysvetlenia) javov sveta (bytia, diania). Cieľom vedeckého poznania však nie je opis jednotlivých udalostí, ale hľadanie pravidielností, opakujúcich sa vlastností, procesov a vzťahov, pričom osobitným cieľom vedy je hľadanie tých, ktoré sú všeobecné a nevyhnutné. O všeobecných a nevyhnutných vlastnostiach, procesoch, vzťahoch hovoríme ako o *zákonitostiach*, *zákonoch*. Systematické vyjadrenie zákona či zákonov určitej oblasti diania sa dosahuje prostredníctvom formulovania *vedeckej teórie*. Formulácia vedeckej

teórie na báze rozsiahleho skúmania oblasti sveta, ktorá je predmetom skúmania, predstavuje zavŕšenie úsilia o poznanie danej oblasti sveta, základ pre porozumenie svetu, pre jeho interpretáciu a pochopenie. Vedecké teórie ako poznanie zákonov sveta sú preto kľúčovými bodmi, o ktoré sa opiera vedecké poznanie sveta.

V súčasnej filozofii vedy panujú rozličné názory na to, čo je to vedecká teória. V zásade však môžeme povedať, že vedecká teória je systém výrokov, ktorý v špecifickom jazyku charakteristickom pre danú vednú oblasť vyjadruje poznanie zákonov sveta. Nie každý systém výrokov však predstavuje vedeckú teóriu. Systém ašpirujúci na to, aby bol vedeckou teóriou musí splniť základné kritériá vedeckosti.

Medzi tieto kritériá patria formálne vlastnosti teórie – každá vedecká teória má určitú vonkajšiu podobu, formu, štruktúru, ktorá je daná jej špecifickým jazykom – formálnym aparátom teórie, definovaním základných termínov, ich vzájomnými vzťahmi, vzťahmi medzi základnými termínmi a ostatnými termínmi teórie. Štruktúra teórie musí spĺňať základné logické pravidlá: definície základných termínov nesmú byť tautologické, nesmú obsahovať neznáme výrazy. Dôsledky musia vyplývať z predpokladov a musí sa dať dokázať, že z daných predpokladov skutočne vyplývajú. Systém nesmie obsahovať výroky, ktoré si navzájom odporujú, pretože v systéme, ktorý obsahuje navzájom si protirečiacie výroky, je možné tvrdiť a dokázať akýkoľvek výrok.

Nemenej dôležité sú obsahové vlastnosti teórie. Medzi tieto vlastnosti patria: spôsob definovania základných termínov a ostatných termínov teórie a zmysluplnosť tvrdení obsiahnutých vo výrokoch teórie. Zmysluplnosť znamená schopnosť teórie opisovať a vysvetľovať prírodné procesy formulovaním všeobecných zákonov. O výrokoch teórie môžeme tvrdiť, že sú zmysluplné vtedy, ak im aspoň principiálne dokážeme priradiť pravdivostnú hodnotu. Ak má byť systém výrokov považovaný za vedeckú teóriu, musí byť podrobený kritickému skúmaniu a prinajmenšom principiálne overiteľný, testovateľný. Ak systém výrokov spĺňa všetky tieto vlastnosti, môžeme ho pokladať za vedeckú teóriu.

Teória evolúcie je vedecká teória, ktorej cieľom je poznanie zákonitostí evolučného procesu na jeho jednotlivých úrovniach, formulovanie zákonov evolúcie. Ak ale máme problém s exaktným vymedzením evolúcie, ak nepracujeme s jedným, ale s niekoľkými pojmami evolúcie, v akom zmysle potom hovoríme o teórii evolúcie?

Je zrejme, že môžeme hovoriť minimálne o dvoch rozličných významoch, v akých sa používa termín *teória evolúcie* – v užšom význame ako teória evolúcie živých organizmov (teória biologickej evolúcie) a v širšom význame ako teória (teórie) opisujúca a vysvetľujúca celú škálu evolučných procesov od vzniku vesmíru až po sociálnu a kultúrnu evolúciu. Odhliadnuc od tohto rozdielu treba vziať do úvahy, že máme do činenia prinajmenšom s dvomi zásadne odlišnými druhmi štruktúr, útvarov, objektov. Prvý z nich vzniká spájaním, syntézou, reakciou jednotlivých zložiek, častí, prvkov, čím vzniká štruktúra kvalitatívne odlišná od svojich konštitutívnych častí. Druhý druh štruktúr je obdarený unikátnou schopnosťou *samoorganizácie* a *sebareprodukcie*, schopnosťou na základe premeny látok a energií zachovávať

vnútornú organizáciu svojich konštitutívnych častí a vytvárať kópie seba samej.⁵ A nielen to, v dostatočne dlhom časovom období a za vhodných podmienok vedie tento proces k rozrôžňovaniu sebareprodukujúcich sa štruktúr, k postupnému objavovaniu sa rozmanitosti foriem, druhov. Až vznikom týchto štruktúr, ktoré zvykneme označovať ako živé, začína evolúcia v pravom zmysle slova. Evolúcia sebareprodukujúcich sa štruktúr schopných samoorganizácie tvorí hlavný predmet skúmania teórie evolúcie.

Teória evolúcie vo všeobecnom zmysle sa musí vysporiadať s niekoľkými kľúčovými problémami. Po prvé musí objasniť, vďaka akým vlastnostiam, akými procesmi a za akých podmienok môžu vznikať samoorganizujúce a sebareprodukujúce sa štruktúry. Po druhé, musí vysvetliť spôsoby ich organizácie a reprodukcie, ktorý je procesom uchovávanía, pretrvávania a zániku daných štruktúr (druhov) v čase, ale zároveň aj to, prečo, ako a za akých podmienok dochádza k vzniku a rastu rozmanitosti, alebo inak a jednoduchšie, musí odhaliť zákony premien druhov živých organizmov. Po tretie, musí objasniť vznik, pretrvávajúce a zánik rozličných druhov reprodukcie, najmä tých, ktoré postupne opúšťajú čisto biologický substrát a vedú k vzniku kvalitatívne nových druhov štruktúr, k tomu, čo sa označuje ako sociokultúrna evolúcia.

Úlohou vedeckej teórie je hľadanie všeobecných a nevyhnutných vlastností, procesov, vzťahov. Pri skúmaní evolúcie (tak v širšom ako aj v užšom význame) však narazíme na problém, ako chápať evolúciu z hľadiska všeobecnej povahy jej zákonov. Či budeme hovoriť o evolúcii vesmíru alebo len o evolúcii života, evolúcia predstavuje jedinečný, neopakovateľný proces vytvárania usporiadaných štruktúr, jedinečnú historickú udalosť. Ak však ide o jedinečný a neopakovateľný proces, môžeme o teórii evolúcie hovoriť ako o vedeckej teórii v rovnakom zmysle ako v prípade iných vedeckých teórií? Je úlohou teórie evolúcie opis a vysvetlenie *zákonov* evolučného procesu alebo môže byť nanajvýš opisom sledu po sebe nasledujúcich historických udalostí?

Na túto otázku možno odpovedať trojakým spôsobom, pričom odpoveď súvisí s chápaním evolúcie ako historicity. Po prvé, evolúciu možno chápať ako postupnosť (následnosť) neopakovateľných udalostí. Teória evolúcie tak môže byť nanajvýš viac či menej podrobným opisom sledu historických udalostí. Z tohto hľadiska je akékoľvek úsilie o vytvorenie teórie evolúcie v zmysle teórie hľadajúcej zákony vývoja, teda teórie v zmysle empirických vied zbytočné a nevyhnutne stroskotá.

Po druhé, evolúciu možno chápať ako postupnosť štádií, ktorá je usmernená nevyhnutným procesom, vnútorným, imanentným zákonom, pričom úlohou teórie evolúcie je odhalenie tohto zákona, ako aj zákonov riadiacich čiastkové evolučné procesy na jednotlivých úrovniach.

Po tretie, evolúcia sa nechápe ako proces, ktorý je usmernený nejakým vnútorným zákonom, ale ako proces ktorý je výsledkom pôsobenia opakujúcich sa dejov, síl, zákonov, ktoré pôsobia vždy a všade tam, kde sú dané primerané podmienky. Evolúcia je historický proces, ktorý však nie je usmernený žiadnym vnútorným zákonom, ale

5 Niekedy sa termín *evolúcia* obmedzuje len reprodukujúce sa systémy. V tomto zmysle by tento termín pokrýval javy biologickej a sociokultúrnej evolúcie, javy neživej prírody by doň nespadať; pozri napr. [30], [28].

realizáciou niektorých možností, procesom, ktorý prebieha na báze prírodných zákonov. Úlohou teórie evolúcie je odhaliť procesy, vzťahy, zákony a podmienky, ktoré vedú k vzniku kvalitatívne nových štruktúr, k ich pretrvávaniu a zániku, o odhalení zákonov vzniku, pretrvávania a zániku štruktúr. Teória evolúcie potom nebude odhalením základného zákona vývoja, ale skôr teóriou teórií – systémom, ktorý bude zahŕňať poznanie evolučných zákonitostí na všetkých úrovniach.

Evolúcia predstavuje komplexný problém, ku ktorému sa pristupuje z rozličných hľadísk a prístupov. Vzhľadom na rozmanitosť vysvetlení evolučných procesov by sme skôr ako o *teórii* mali hovoriť o *teóriách* evolúcie. Nemáme tu na mysli len podstatný rozdiel medzi vedeckými teóriami na jednej strane a filozofickými teóriami na druhej strane. Aj z hľadiska vedy existujú rôzne vysvetlenia, rôzne teórie usilujúce sa vyriešiť ten istý problém (stačí spomenúť len spory o vysvetlenie premeny druhov v rámci darvinistického prístupu alebo spor darvinistické-nedarvinistické vysvetlenia biologickej evolúcie). Dôležité je, aby každý takýto prístup spĺňal základné kritériá vedeckosti, aby bolo možné objektívne sa presvedčiť, či daná teória môže byť pravdivá alebo nepravdivá.

2.3 Postavenie teórie evolúcie v systéme vedeckého poznania

Ak budeme evolúciu chápať v širokom zmysle slova, potom má teória evolúcie v systéme vied kľúčové postavenie. Je završením, syntézou všetkého poznania, ktorý o svete máme. Táto teória ukazuje, ako všetky procesy, ktoré poznáme z vied, vstupujú do existencie, vzniku, pretrvávania a zániku rozličných štruktúr. Tým, že odhaľuje prirodzené príčiny javov presahuje hranice jednotlivých vied, ale aj prírodných, spoločenských a humanitných vied. Neexistuje veda ani vedná disciplína, ktorá by mala pri budovaní teórie evolúcie výsostné postavenie. Na vysvetlení takých komplexných javov je nevyhnutná široká interdisciplinárna spolupráca často z veľmi vzdialených oblastí, čo kladie na výskum aj na vedcov vysoké nároky. Dôležitá je povaha interdisciplinarita – treba dosiahnuť stav, v ktorom sa od opisu určitého problému z hľadiska rozličných disciplín prejde k jeho syntetickému *multidisciplinárnemu* opisu.

To sa, samozrejme, v plnej miere týka aj teórie biologickej evolúcie. Tá má v systéme vedeckého poznania špecifické postavenie. Nielen preto, že stála na začiatku odhaľovania procesov vývoja sebareprodukujúcich sa štruktúr, ale najmä preto, že do svojho vysvetlenia zahrnula človeka, ľudský rod a zásadným spôsobom vstúpila na pôdu, ktorá sa zdala (a pre mnohých sa ešte stále zdá) byť dominantnou, či skôr výhradnou oblasťou spoločenských a humanitných vied. Teória evolúcie sa vďaka tomu stala jednou z fundamentálnych teórií vedy, ktorá podstatným spôsobom ovplyvňuje nielen naše videnie a chápanie prírodného diania, ale predovšetkým odhaľuje náš vlastný pôvod, čím ovplyvňuje videnie a chápanie postavenia človeka a ľudskej spoločnosti v prírodnom dianí. Dotýka sa chápania zmyslu ľudskej existencie, najzásadnejších filozofických, ideologických, svetonázorových a náboženských otázok

Je pochopiteľné, že teória, ktorá si kladie nárok na vysvetlenie pôvodu takých výsostne „človečenských“ vlastností akými sú ľudské poznávanie sveta, správanie človeka, ľudský „duch“, ktorá sa usiluje ukázať, že procesy vedúce v živej prírode k vzniku nových druhov, v ľudskej spoločnosti vedú k vzniku rozličných kultúr, sa stane objektom nemilosrdnej kritiky, rozličných interpretácií a dezinterpretácií. Žiaľ, veľmi často a unáhle sa pritom prehliada jedno z posolstiev teórie evolúcie – možnosť vytvorenia mostu ponad priepasť medzi prírodnými vedami na jednej strane a spoločenskými a humanitnými vedami na druhej strane. Priepasť, ktorá spočíva v rozdielnosti teórií a metodológií, ktorá sa prejavuje aj v rozdielnosti nazerania na povahu vedeckého skúmania, na jeho ciele a postupy. Na je prekvenutie je nevyhnutný široký dialóg, otvorený a nezaujatý prístup zo všetkých strán. Zdá sa však, že v tomto procese sme ešte len na začiatku.

3 Stručný náčrt histórie teórie evolúcie

Historický opis zrodu teórie evolúcie, prvých evolučných myšlienok obvykle začína v 18. storočí s odvolaním sa na práce Buffona, Erasma Darwina, Lamarcka, Saint-Hilaira a ďalších. Možno je to dôsledok skutočnosti, že teóriu evolúcie pokladáme za súčasť modernej vedy a základy vedy v modernom zmysle slova sa obvykle kladú do obdobia renesancie a najmä novoveku. Možno je to dôsledok Butlerovho pokusu ukázať, že Charles Darwin len v novej podobe opakuje myšlienky týchto mysliteľov [10]. Možno je to aj dôsledok toho, že pod evolúciou, vývojom sa v prevažnej miere chápe postupný vznik čoraz zložitejších, dokonalejších, vyšších štruktúr. Ak však pod evolúciou budeme rozumieť vznik kvalitatívne odlišných štruktúr bez toho, aby sme do tohto procesu kládli progres (v akomkoľvek zmysle slova) ako univerzálny a nevyhnutný princíp, potom musíme siahnuť hlbšie do histórie a vrátiť sa ku koreňom európskej vedy.

3.1 Od počiatkov k transformácii druhov

Prvé teórie usilujúce o prirodzené vysvetlenie vzniku, premeny a zániku vecí nájdeme v antickej prírodnej filozofii. Tu sa rodí európska veda, pričom pri jej zrode stáli tie isté otázky, ktoré si kladie teória evolúcie – čo je to svet, aký je jeho pôvod, ako vznikla rozmanitosť druhov vecí, prečo veci vznikajú a zanikajú atď. Pozoruhodnou črtou antickej filozofie je rôznorodosť koncepcií, vytváranie teórií vzniku a zániku, ktoré sa niekde dopĺňajú, ale zväčša si konkurujú, súperia o najlepšie vysvetlenie.

Spoločným základom koncepcií antických filozofov je úsilie odhaliť za rozmanitosťou sveta spoločný jednotiaci princíp. Antická prírodná filozofia je hľadaním základných prvkov (živlov, počiatkov, princípov), ktorých činnosťou, zlučovaním a rozlučovaním, miešaním, miesením vznikajú všetky veci, vrátane živých organizmov a človeka. Rozliční myslitelia sa líšia v tom, čo pokladajú za počiatky – podľa Táleta všetko jestvujúce pochádza z vody, podľa Anaximena je prvkom, počiatkom vzduch, podľa Herakleita oheň. Anaximandros svojím uvažovaním pokročil za konkrétne materiálne prvky, za počiatok vyhlásil *neobmedzené (apeiron)*, ktoré

predchádza materiálnym živlom. Podľa Anaxagora nemôžeme hovoriť o počte počiatkov, pretože ich je nekonečné množstvo.

Za vrchol antickej prírodnej filozofie môžeme pokladať teóriu štyroch živlov poeleatského filozofa Empedokla z Akragantu. Empedokles oddelil večné a nemenné prvky od hybných princípov. Existujú štyri základné prvky alebo živly: *ohneň, voda, zem a vzduch*. Každá zmena, vznik a zánik všetkých vecí, je dôsledkom miešania a oddeľovania prvkov, ktoré sa uskutočňuje pôsobením hybných princípov (pohybových príčin) – *Lásky a Sváru*. Miešanie prvkov, vznik a zánik vecí prebieha v cykloch. Najvyšším cyklom, ktorý zahŕňa celý svet, vznik a zánik všetkých vecí, je *Kozmický cyklus*. Jeho súčasťou je *Veľký cyklus*, v ktorom vznikajú živé tvory vrátane človeka.

Živé tvory však nevznikajú naraz. Najprv zo základných elementov vznikajú končatiny, hlavy bez tiel, oči, ktoré nie sú pripojené k hlavám atď. V druhom štádiu sa jednotlivito existujúce orgány začínajú náhodne spájať, pričom vznikajú rozličné monštrá: býky s ľudskou hlavou, ľudia s býčou hlavou, tvory, ktoré nie sú ani muž, ani žena. Do procesu vstupuje akási protopodoba prírodného výberu – nie všetky tvory majú také usporiadanie tela, ktoré im umožňuje prežiť súperenie s ostatnými tvormi. Dochádza k postupnému oddeľovaniu pohlaví. V poslednom štádiu, ktoré mala predstavovať súčasnosť, už nové formy nevznikajú priamo zo základných elementov, ale len reprodukciou existujúcich organizmov. Druhy, ktoré sa zachovali až do tohto štádia, sú vybavené mimoriadnou odvahou a šikovnosťou, ktorá im umožnila prežiť a rozmnožovať sa [53].

Empedoklovo učenie významne ovplyvnilo ďalší vývoj antickej filozofie. Teória štyroch živlov sa premietla aj do tých koncepcií, podľa ktorých *voda, ohneň, zem a vzduch* nie sú skutočnými počiatkami. Za tieto sa pokladajú atómy, čísla alebo idey, do každej však v určitej fáze vstupuje činnosť štyroch živlov. Pozoruhodná je Platónova teória, v ktorej sa nemateriálne počiatky, Idey, zložitým spôsobom spájania formy a telesnosti vtlačajú do priestoru, vytvárajú sa geometrické útvary, ktorých spájaním vznikajú štyri živly. Ich miešaním, miesením, zlučovaním a rozlučovaním vzniká postupne celý vesmír – hviezdy, Zem a všetko jestvujúce.

Platón bol presvedčený, že vznikanie vecí, na ktorom sa pod vedením Demiurga zúčastňujú božstvá, je procesom úpadku pôvodne dokonalej Idey. Degenerácia sa prejavuje aj pri vzniku živých organizmov: ako prvý vzniká najdokonalejší zo živých tvorov – muž. Ak však svoj život neprežil riadne, pri ďalšom návrate z hviezd vojde jeho duša do prirodzenosti ženy. Postupným úpadkom mužského rodu vznikli všetky ostatné druhy živočíchov [Tim. 49A-92C; 78].⁶ Vznikom živočíchov úpadok, samozrejme, nekončí, pokračuje v postupnej degenerácii aténskeho štátu, zaplňaní sveta čoraz nedokonalejšími bytosťami. Napriek tejto chmúrnej teórii je Platón optimista, domnieva sa, že tento nepriaznivý vývoj možno zvrátiť. Je však potrebné, aby štát riadili tí, ktorí sú schopní preniknúť až k Ideám, aby vytvorili štát podľa ich obrazu [79].

6 Za pozornosť stojí, že „štvernohé pozemské zvieratá vznikli z ľudí, ktorí nemali nijakú lásku k filozofii ani vedecký záujem o prírodu, pretože prestali používať kruhovitý pohyb v hlave a ako vodcov nasledujú tie časti duše, ktoré sú umiestnené v prsiach“ [Tim. 91D; 78].

Teória štyroch živlov ovplyvnila nielen antickú filozofiu, ale aj formovanie teoretického myslenia v rodiaciach sa vedách. Vďaka Hippokratovi, Platónovi, Aristotelovi a Galénovi, pre ktorých bolo základným vysvetlením fungovania živých tiel, ostalo fundamentálnou teóriou vied o živých organizmoch v antickom období, v stredoveku, aj v rodiacej sa renesančnej a novovekej vede a filozofii. Niektoré črty tohto učenia možno v biomedicínskych vedách nájsť ešte na konci 18. storočia.

Teórie vzniku a zániku sa v európskej filozofii a vede opäť objavujú až v druhej polovici 18. storočia. Objavujú sa myšlienky, ktoré kliesnia cestu vedeckej teórii evolúcie živých organizmov, vo filozofii vznikajú univerzálne teórie vývoja, ktoré sa – podobne ako v antickej filozofii – pokúšajú obsiahnuť celý svet, vesmír, bytie. Kým prírodovedné teórie sú inšpirované pokrokom poznania v rozličných oblastiach vedeckého skúmania (geológia, paleontológia, spor vitalizmu a preformácie v rodiacej sa embryológii, problém spontánnej generácie, vzniku života, problém vymedzenia druhov a ich nemennosti ako dôsledku ich božského stvorenia prebiehajúci na pozadí sporu o povahu univerzálií), filozofické teórie opierajú svoje inšpirácie o myšlienky osvietencov smerujúce k zdôvodneniu potreby spoločenskej zmeny.

Jedným z prvých mysliteľov, ktorý sa zaoberal premenou živých organizmov, bol Georges Buffon. Buffon vo svojich úvahách dospieva k názoru, že linnéovské druhy nie sú druhy, ale len variety, ktoré sa môžu meniť. Bol presvedčený, že skutočné druhy vznikajú z pôvodnej ancestrálnej formy pôsobením vonkajšieho prostredia priamo na organizmy prostredníctvom organických častíc (povahu tohto procesu však bližšie neobjasňuje). Tento proces vedie k postupnej degenerácii ancestrálneho typu, ktorá môže pokračovať počas niekoľkých generácií, až kým nedosiahne hranicu danú fundamentálnou podstatou danej formy (typu). Pozoruhodné je, že tento proces je vratný – ak sa zmenia podmienky vyvolávajúce degeneráciu, organizmy sa môžu vrátiť k pôvodnej forme [9].

Pravdepodobne prvým prírodovedcom, ktorý publikoval teóriu transmutácie druhov s detailným výkladom procesu vzniku nových druhov, bol Darwinov starý otec Erasmus Darwin, lekár, básnik a vynálezca. Podobnosti medzi myšlienkami Erasma Darwina a Lamarcka viedli Charlesa Darwina (ale napríklad aj Samuela Butlera) k tomu, že obviňoval Lamarcka z vykrádania jeho starého otca. Je pravda, že obaja myslitelia sa odvolávajú na potreby organizmu ako hlavnú príčinu ich zmien. Zásadne sa však líšia vo výklade, ako sa tieto potreby prejavujú do zmien v stavbe organizmov. Je preto pravdepodobnejšie, že podobnosti v ich koncepciách sú dôsledkom inšpirovania sa tým istým zdrojom, G. Buffonom, ako aj zhodnotenia poznatkov farmárov a šľachtiteľov hospodárskych zvierat a plodín a v neposlednom rade výsledkom vlastných pozorovaní, ktoré ich priviedli k podobným záverom.

Podľa Erasma Darwina všetky druhy živočíchov, tie, ktoré žili na Zemi v minulosti, ale aj tie, ktoré ju obývajú v súčasnosti, sú potomkami jedného prvotného živého *filamentu*, ktorý sa postupne mení. Jeho transformácie sú spôsobené *Veľkou Prvou Príčinou*, ktorá ho vybavila schopnosťou neustále sa zdokonaľovať, získavať nové orgány a vlastnosti a prenášať ich do ďalších generácií. Proces vytvárania nových orgánov a vlastností, proces transmutácie druhov, je riadený, podobne ako neskôr u Lamarcka, potrebami organizmu, najmä potrebou rozkoše, hladu a bezpečia. Napríklad rohy, ktorými sa vyznačujú niektoré druhy živočíchov, rastú samcom

z potreby získať samice a ubrániť si svoj hárem pred ostatnými samcami, chobot sa slonom predlžuje pôsobením potreby dosiahnuť lístie vo vysokých korunách stromov, krídla, plutvy alebo tvrdé ulity vznikajú z potreby utiecť alebo ochrániť sa pred nepriateľom atď. [16]

Skutočnosť, že všetky živé organizmy, sú potomkami prvého živého filamentu znamená, že sú pospájané príbuzenskými vzťahmi. Všetko živé je jedna rodina jedného rodiča. Súčasťou tejto rodiny je aj človek, ktorý je rovnako potomkom prvého živého filamentu, ako všetky ostatné organizmy. [16]

Je pozoruhodné, že už Erasmus Darwin si všimol, že každá generácia plodí viac potomstva, ako môže prežiť, čo vyvoláva súťaž a selekciu, ktorá by mohla byť riadiacou silou evolučnej zmeny. Sám však tieto myšlienky ďalej nerozpracoval a ostali nepovšimnuté. V tejto súvislosti sa občas objavuje myšlienka o novej inšpirácii Charlesa Darwina dielom svojho starého otca. Je pravda, že v mladom veku Darwin obdivoval práce svojho starého otca. Neskôr sa však o nich vyjadroval veľmi kriticky, odsudzoval ich špekulatívny charakter a nedostatočné zdôvodňovanie teoretických záverov empirickými faktami. [20]

Ďalším z radu tých, čo stáli pri zrode teórie evolúcie, bol Lamarckov súčasník a mladší kolega Geoffroy Saint-Hilaire. Saint-Hilaire rozpracoval Cuvierovu myšlienku, že všetky živočíchy sú vytvorené na základe jedného zo štyroch základných plánov. Podľa Saint-Hilaira všetky živočíchy predstavujú jeden základný stavebný plán. Jednotlivé živočíšne druhy sú variantmi jedného fundamentálneho archetypu, pričom medzi nimi jestvujú konštantné vzťahy. Hoci spočiatku neuvažoval o transmutácii druhov, koncom 18. a začiatkom 19. storočia sa začal zaoberať myšlienkou, že všetky súčasné živočíšne druhy môžu byť potomkami tvorov žijúcich pred potopou sveta. V dvadsiatych rokoch 19. storočia sa priklonil k Lamarckovým myšlienkam o premene druhov, ktorej príčinou je vytváranie nových návykov, zmeny v používaní a nepoužívaní orgánov. Lamarckova *Zoologická filozofia* pre neho predstavovala nový základ, na ktorom treba rozvíjať úvahy o prírode. Na rozdiel od Lamarcka kládol väčší dôraz na vplyv prostredia. Podľa jeho názoru je formatívna sila, ktorá modifikuje tvar organizmov, výsledkom pôsobenia vonkajšieho prostredia, najmä veľkých zmien, o ktorých hovorí geológia. Na druhej strane prílišné sústredenie na hľadanie podobností medzi rozličnými druhmi živočíchov, jednotnej fundamentálnej formy, mu zabránilo študovať rozmanitosť v živočíšnej ríši, škálu rozličných adaptácií na rôzne podmienky prostredia, ktorá sa pre evolucionistov stala vlastným základom evolučného procesu.

Približne v tom istom období, ako sa objavujú evolučné úvahy v prírodných vedách, začínajú sa predovšetkým v nemeckej filozofii objavovať úvahy, ktoré vedú k pokusom o vytvorenie všeobecnej teórie vývoja, evolúcie. Cieľom je odhalenie všeobecných princípov a podmienok evolučného procesu ako celku, jeho vnútornej podstaty, zákona, o ktorý sa opiera každý parciálny evolučný proces. Odhalenie podstaty evolúcie ako celku, totality je cestou k správne pochopeniu sveta (kozmu, bytia), k odkrytiu jeho skutočného zmyslu.

Najmä v nemeckom prostredí vznikajú filozofické teórie ako reakcia proti „racionálnemu empirizmu“ prírodných vied, proti ich materializmu, reprezentovanému mechanistickým názorom a mechanistickou metodológiou. Formujú sa

prostredníctvom myšlienok tzv. prírodnej filozofie (*naturfilozofie*), ale aj vrcholných predstaviteľov klasickej nemeckej filozofie. Podľa predstaviteľov naturfilozofie je mechanizmus v zásade nesprávny a neúplný spôsob poznávania sveta, pretože neberie do úvahy spirituálny aspekt človeka a prírody. Naturfilozofia nadväzuje na mystické tradície, predovšetkým na Jacoba Böhma, podľa ktorého je všetko živé i neživé obdarené duchom, ktorého možno poznať skúmaním aktivity živých tel. Príroda je manifestáciou Boha, ako taká všetko označuje znakmi, ktoré sú kľúčom k odhaleniu vnútorných síl, kľúčom k Jazyku Prírody. Naturfilozofia preto kladie dôraz na spirituálne vciťovanie sa, na špekulatívne myslenie, ktoré nemá byť brzdené empirickými predsudkami. Viacerých predstaviteľov inšpiroval Immanuel Kant rozlíšením noumenálneho a fenomenálneho. Domnievali sa, že transcendentálne poznanie vecí o sebe, podstat, možno dosiahnuť nie rozumom, racionálnou rekonštrukciou, ale vciťovaním sa do podstaty vecí. Na rozdiel od nich vrcholný predstaviteľ klasickej nemeckej filozofie Georg Wilhelm Friedrich Hegel, ktorého myšlienky mali veľký vplyv na vývoj naturfilozofie, kladol dôraz na rozumové preniknutie do skutočnej podstaty procesov, na skúmanie vecí nie zvonka, čo je podľa neho charakteristické pre prístup vied, ale zvnútra tak, aby racionálna rekonštrukcia viedla k odhaleniu vnútornej logiky vývoja ich podstat.

Pre vrcholných predstaviteľov naturfilozofie, akými boli Goethe, Schelling či Oken, predstavuje príroda, svet ako celok spirituálnu jednotu. Úlohou filozofie je odhaliť principiálny vzťah medzi človekom a svetom, medzi mikrokozmosom a makrokozmosom. Evolúcia je procesom vývoja Absolútna (ako pojmu), procesom, ktorý vnútornou aktivitou absolútna vedie z pôvodného stavu nediferencovanosti k stále vyššiemu poriadku (Schelling), dialektickým procesom vývoja ducha, jeho fenomenológie, postupného vyjavovania v rôznych štádiách od najnižšieho po najvyššie, ktoré je výrazom jeho vnútornej aktivity (Hegel). Lorenz Oken, lekár a prírodovedec, ktorý sa na začiatku 19. storočia stal vedúcim predstaviteľom naturfilozofie, tvrdil, že zákony ducha sa neodlišujú od zákonov prírody, zákony prírody sú vyjadrením zákonov ducha. Človek ako univerzálny duch disponuje jazykom, ktorý je duchom sveta. Ako taký je schopný poznať, odhaliť, vyjadriť zákony ducha i zákony prírody. Všetko jestvujúce predstavuje reťaz bytia, všetko živé pochádza z toho istého základného materiálu [76].

Na rozdiel od naturfilozofov sa britský filozof Herbert Spencer usiloval postupovať v súlade s poznatkami, teóriami a metódami prírodných vied. Jeho cieľom bolo vybudovať filozofiu ako univerzálnu vedu na báze jestvujúceho vedeckého poznania, ako vedu, ktorá odhaľuje univerzálne zákony všetkého jestvujúceho striktno vedeckým spôsobom. Spencer vytvoril nielen jednu z univerzálnych teórií vývoja (pričom tento vývoj vyjadroval termínom *evolúcia*), v ktorej je evolúcia živých organizmov len jednou z jeho častí, ale zároveň zdôraznil význam *evolúcie* ako nevyhnutne progresívneho vývoja. Bol presvedčený, že vývoj, evolúcia znamená nevyhnutný pokrok k zložitejším, vyšším a dokonalejším formám, pričom tento proces sa riadi jedným univerzálnym zákonom – procesom rastúcej integrácie diferencovaných častí, ktorým z pôvodnej jednoduchej nediferencovanej homogenity vzniká zložitá diferencovaná heterogenita. Tento zákon sa od ostatných prírodných zákonov líši len svojou všeobecnosťou, ostatné prírodné zákony sú jeho špeciálnymi

prípadi. Ako taký sa týka všetkého vznikajúceho, od hviezd a galaxií až po vznik človeka, jeho mysle, ľudskej spoločnosti a jej vývoja. [95]

Spencerove myšlienky sa vzhľadom na blízkosť vedeckému poznaniu aj vzhľadom na encyklopedický rozsah (Spencer sa v duchu pozitivistickej filozofie pokúsil obsiahnuť všetko vedecké poznanie) tešilo v druhej polovici 19. storočia veľkej popularite. Napokon výraz *prežívanie zdatných*, ktorý Darwin neskôr prevzal do svojej teórie a ktorý sa stal predmetom kontroverzných diskusií, pochádza pôvodne od Spencera, ktorý si dokonca uplatňoval nárok na teóriu prírodného výberu. Neskôr však začal prírodovedcom prekážať špekulatívny charakter jeho teórie, v 20. storočí sa už jeho prínos pre rozvoj teórie evolúcie pokladal za zanedbateľný (pozri napr. [71]).

3.2 Základy teórie evolúcie živých organizmov – Lamarck, Darwin

Na prelome 18. a 19. storočia sa objavuje vysvetlenie procesu premeny druhov, ktoré sa pokladá za prvú ucelenú teóriu evolúcie živej prírody. Jej autorom je francúzsky zoológ Jean-Baptiste Antoine de Monet, chevalier de Lamarck.

Podľa Lamarcka je premena biologických druhov, evolúcia, procesom progresívnych zmien, ktoré vedú k rastu zložitosti v stavbe tela a jeho fungovaní. Hlavnou hybnou silou evolučného procesu je vnútorná tendencia, potreba dokonalejšie sa prispôsobiť podmienkam prostredia, čo sa prejavuje v raste zložitosti. Najjednoduchšie živé organizmy, ako sú nálevníky, vznikajú spontánnou generáciou, samovoľným vznikom z neživej hmoty, ako produkt pôsobenia veľmi jemných tekutín, kalorických a elektrických, ktoré v látkach slizovitej povahy vytvárajú vakuoly a bunky, čím vznikajú primitívne živé organizmy. Od počiatku sú vybavené vnútornou potrebou vyvíjať sa, nadobúdať čoraz zložitejšie formy, prispôbovať sa čoraz dokonalejšie podmienkam prostredia. Táto vnútorná potreba sa uskutočňuje vytváraním nových návykov, používaním, resp. nepoužívaním orgánov. Vlastný mechanizmus pôsobenia vnútorných potrieb predstavuje prúdenie tekutín (kalorických a elektrických) v smere používania orgánov, čo vyvoláva prerážanie nových kanálikov, vedúcich k zmenám v stavbe tela. Vplyvom meniacich sa podmienok prostredia sa menia nároky na potrebu organizmov, na používanie či nepoužívanie orgánov. Meniace sa prostredie nie je bezprostrednou príčinou evolučných zmien. K premene druhov by dochádzalo aj v stabilnom, nemeniacom sa prostredí, pretože hlavnou hybnou silou evolúcie je neustála potreba lepšie sa prispôsobiť podmienkam prostredia, ktorá sa uskutočňuje opakovaným používaním orgánov (podľa [88]).

Súčasťou Lamarckovej teórie evolúcie je dedičnosť vlastností získaných používaním či nepoužívaním orgánov. Je iróniou dejín vedeckého myslenia, že hoci tento princíp nehrá v Lamarckovej teórii hlavnú úlohu a Lamarck ani nie je jeho autorom (napokon aj myšlienka o zmene stavby orgánov pod vplyvom ich používania či nepoužívania nepochádza od Lamarcka; Lamarck bol však prvý, kto tieto myšlienky spojil do konzistentnej teórie), stal sa ústrednou myšlienkou *lamarckistických* koncepcií evolúcie, ktoré sa kladú do protikladu s *darwinistickými* teóriami evolúcie. Darwin však neodmieta dedičnosť získaných vlastností, ale Lamarckov princíp vnútorných potrieb organizmu, ktorý, zrejme pod vplyvom Lyellovho podania, interpretuje ako *vôľu* alebo *túžbu* organizmov po zdokonaľovaní. Hoci podľa Lamarcka

ide o čisto fyzikálny princíp (podľa Lamarcka potreba pôsobí prostredníctvom fyzikálnych – mechanických a hydraulických – síl), jeho kritici (vrátane Darwina), ale aj zástancovia (pozri napr. [54] mu pripisujú vnášanie vôľového, a teda nemateriálneho, spirituálneho princípu do vysvetľovania evolúcie, čo je v prírodnej vede nielen neprípustné, ale aj v úplnom rozpore s pozorovanými faktami.

Lamarck sa počas svojho života nedočkal nielen uznania, ale ani širšej popularity. V tomto smere rozhodne nemohol súperiť so svojím hlavným odporcom, Georgesom Cuvierom. Podľa Cuviera Lamarckovo tvrdenie o postupnom, graduálnom charaktere procesu premeny druhov nezodpovedá paleontologickým údajom. Nenašli sa žiadne pozostatky druhov, ktoré tvoria chýbajúce prechodné články v evolúcii, hoci existencia takýchto druhov z teórie evolúcie vyplýva. Navyše zmeny, ktoré možno pozorovať u jednotlivcov rôznych druhov, sa odohrávajú v úzkych hraniciach a týkajú sa len nepodstatných vlastností. A napokon pozostatky druhov, ktoré žili v dávnych dobách, napríklad múmie níljských krokodílov z obdobia starého Egypta, staré okolo tritisíc rokov, sa vôbec nelíšia od dnes žijúcich krokodílov. Žiadna významná zmena nenastala, to znamená, že *teória transmutácie* je nezmysel. Paleontologické nálezy svedčia skôr o opakujúcom sa náhlom vyhynutí všetkých žijúcich organizmov vplyvom globálnej katastrofy, napríklad potopy, a rovnako náhlom vzniku nových druhov, bez akýchkoľvek genetických väzieb na vyhynuté druhy. Podľa Cuviera sú všetky časti organizmov vzájomne úzko koordinované a späté s prostredím, v ktorom príslušný organizmus žije. Každá významnejšia zmena by viedla k rozpadu vnútornej organizácie a k narušeniu jemnej rovnováhy medzi organizmom a prostredím, ktorá bola stanovená Stvoriteľom. Cuvier venoval veľkú pozornosť kritike princípu používania a nepoužívania orgánov, ktorý interpretoval ako zmeny vyvolávané túžbami a želaniami hlúpych zvierat. Cuvier bol nielen brilantný vedec. Svoje vedecké argumenty dokázal okoreniť dávkou sarkazmu. Lamarck napríklad v súlade so svojou teóriou tvrdil, že druhy slepých rýb, ktoré sa našli v tmavých jaskyniach, stratili schopnosť vidieť nepoužívaním zrakových orgánov, tým, že pre život v jaskynných podmienkach nepotrebujú používať oči. Keď Lamarck oslepol, Cuvier počas jednej verejnej prednášky na jeho adresu uviedol, že jeho slepota je spôsobená skutočnosťou, že nepoužíva svoj zrak správnym spôsobom, odmieta sa správne pozeráť na prírodu [100].

V dvadsiatom storočí prišlo k oživeniu záujmu o Lamarckovu teóriu, presnejšie o tú časť, ktorá nebola ani jeho objavom, ani v jeho koncepcii nehrala významnejšiu úlohu – o dedičnosť získaných vlastností. Tento záujem súvisí na jednej strane s presadením sa darvinistického výkladu premeny druhov, na druhej strane s problémami, ktoré darvinizmus nedokázal uspokojivo vyriešiť. Tzv. *lamarckistické teórie* ([54] a i.), ktoré zdôrazňujú evolučnú úlohu zmien získaných počas individuálneho života, patria medzi výrazných oponentov darvinistických teórií.

Prvé vysvetlenie procesu premien druhov živých organizmov, ktoré môžeme označiť za skutočne vedecké prišlo až v polovici 19. storočia. Jeho autorom je vnuk Erasma Darwina, Charles Robert Darwin.

Darwin bol výrazne ovplyvnený tradíciou britského empirizmu opierajúceho sa o metódu indukcie. Osvojil si názor, že „veda spočíva v zhromažďovaní faktov takým spôsobom, aby z nich bolo možné odvodiť všeobecné zákony alebo pravidlá“ ([20],

32). Bol presvedčený – a pokúsil sa to ukázať aj vo svojich spisoch – , že svoju teóriu, všeobecné zákony premeny druhov živých organizmov, odvodil na základe spracovania nesmierneho množstva dát, ktoré nazbieral počas svojej päťročnej cesty okolo sveta na palube lodi *Beagle*. Jeho vysvetlenie je pritom udivujúco jednoduché. Nezavádza žiadny neznámy, zatiaľ nepostihnutý princíp, ani sa neusiluje o aplikáciu známych fyzikálnych (mechanických) zákonov ako jeho predchodcovia. Vychádza len z pozorovaných javov, z takých vlastností, s ktorými sa stretávame u všetkých druhov živých organizmov:

Po prvé, všetky živé organizmy sa rozmnožujú.

Po druhé, potomkovia dedia vlastnosti od svojich rodičov.

Po tretie, napriek veľkej podobnosti vlastností medzi príslušníkmi jedného druhu existuje medzi nimi celý rad odchýliek, variácií.

Po štvrté, v prírode existuje obrovský nepomer medzi počtom vyprodukovaných zárodkov a počtom jedincov, ktorí sa dožívajú reprodukčného štádia.

Podľa Darwina z týchto skutočností, z dedičnosti, premenlivosti, nevyhnutnosti súperenia o zdroje potravy a možnosti reprodukcie (z nevyhnutnosti „boja o život“) vyplýva, že v prírode pôsobí *prírodný výber* – v interakcii s prostredím sa z množstva variet vyberajú tie, ktoré sú najlepšie prispôbené na dané podmienky prostredia. V postupnosti mnohých generácií počas nesmierne dlhých geologických období vedie tento proces k postupnej premene pôvodnej rodičovskej formy, k vzniku nových foriem, nových druhov. Princíp, ktorý mal Darwinovi pôvodne slúžiť na vysvetlenie vzniku adaptácií, účelného prispôsobenia sa živých organizmov na podmienky prostredia, sa stal kľúčovým princípom pre vysvetlenie premeny druhov živých organizmov.

Darwin sa pri svojich úvahách opieral o niekoľko teoretických princípov. Okrem indukčnej metódy (ktorej uplatnenie pri tvorbe Darwinovej teórie je vďačným predmetom skúmania historikov a filozofov vedy) Darwin vychádzal z *princípu uniformity*, ktorý v silnej podobe tzv. *uniformitarianizmu* sformuloval britský geológ James Hutton v 18. storočí a ktorý si Darwin osvojil vďaka encyklopedickému dielu Charlesa Lyella *Principles of Geology*⁷. Základom *princípu uniformity* je tvrdenie, že prírodné procesy pôsobia v súčasnosti rovnako (rovnakou silou, rýchlosťou a intenzitou) ako pôsobili v minulosti. V podobe *uniformitarianizmu* sa bezprostredne spája s gradualizmom, s myšlienkou postupného pomalého a kontinuálneho hromadenia drobných zmien, ktoré v dostatočne dlhom časovom horizonte môžu vyústiť do podstatných zmien. Darwin bol presvedčený, že vznik druhov je výsledkom dlhého procesu postupných zmien, viackrát opakoval, že príroda nerobí skoky.

Druhým významným teoretickým podnetom sa Darwinovi stala práca Thomasa Malthusa *On Population*, v ktorej rozpracoval myšlienku o vzťahu rastu populácie a rastu zdrojov potrebných na jej udržanie, ako aj dôsledkov, k akým tento rozporný

⁷ Základné tézy *uniformitarianizmu* sformuloval Hutton, samotný výraz *uniformitarianizmus* po prvý použil až významný britský učenec William Whewell v roku 1832 pri recenzii Lyellovho diela

vzťah vedie. Zdá sa, že Malthusova práca inšpirovala Darwina, aby si uvedomil dôsledky nepomeru medzi počtom zárodkov a počtom dospelých jedincov.⁸

Darwinova teória sa stala jedným zo základných teoretických pilierov, ktorý na konci 19. storočia umožnil zjednotenie rozmanitých „prírodospytných“ disciplín do vedy označovanej ako biológia. Štruktúra jeho teórie a spôsob vysvetlenia v mnohom pripomína Newtonovu fyziku – rozmanitosť vlastností živých organizmov redukuje na niekoľko základných vlastností spoločných pre všetky živé organizmy, z ktorých odvodzuje jednoduchý princíp premeny druhov. Za prevratné (a to ešte aj v súčasnosti) treba pokladať fundamentálnu úlohu náhody a zmenu postavenia človeka. Darwin zásadným spôsobom mení chápanie zákonitosti a usmernenia prírodného diania. Náhoda nie je niečo, čo len modifikuje podmienky, za akých prebiehajú prírodné procesy, čo len mení „formy prejavu podstatných vzťahov“. Náhoda je niečo, čo je podstatnou súčasťou evolučného procesu, čo zásadným spôsobom determinuje jeho neurčitost' a nepredpovedateľnosť. Človek nie je zavŕšením vývoja, ani centrom diania, ale len jedným z mnohých produktov evolučného procesu.

Darwinova teória sa stala predmetom rozsiahlych diskusií a sporov, ktoré čoskoro prekročili rámec biologických a aj prírodných vied. Darwin si zrejme už v štyridsiatych rokoch 19. storočia, keď jeho teória existovala len vo forme náčrtov, uvedomoval prevratný dosah svojich myšlienok. Obava o ich prijatie bola napokon jedným z dôvodov, prečo tak dlho otáľal s ich uverejnením.

Ako ukazuje história i súčasnosť, jeho obavy boli opodstatnené. Jeho teória vyvolala kontroverzné ohlasy, široká diskusia, polemiky, interpretácie a dezinterpretácie siahajúce až k ideologickému zneužívaniu pokračujú až do súčasnosti. V 20. storočí sa stala jednou zo základných teórií modernej vedy a svoju silu ukazuje aj na začiatku 21. storočia.

3.3 Teória evolúcie a revolúcia v biológii v 20. storočí

Darwinova teória nedala odpoveď na všetky otázky. Naopak, skôr by sa zdalo, že otvorila viac problémov ako riešení. Plodnosť vedeckých teórií sa však neodráža len v tom, koľko problémov dokáže vyriešiť, ale aj v tom, koľko nových problémov nastoľuje, koľko možných smerov ďalšieho vývoja poznania otvára. [81]

Ako zásadný problém, ktorého riešenie viedlo v 20. storočí k revolúcii v biologických vedách, sa ukázalo vysvetlenie mechanizmov dedičnosti a premenlivosti. Hoci po krátkom úspechu *neodarvinizmu*, ktorý sa na prelome 19. a 20. storočia pokúsil o syntézu prvých poznatkov o dedičnosti s teóriou prírodného výberu, došlo v prvých desaťročiach k výraznému ochladeniu vzťahu prírodovedcov k darvinovskej teórii evolúcie, v tridsiatych a štyridsiatych rokoch došlo k syntéze

⁸ Je zaujímavé, že tá istá Malthusova práca inšpirovala aj Alfreda Russella Wallacea k sformulovaniu rovnakého princípu, aký objavil Darwin. List mladého Wallacea a rukopis jeho knihy sa stali pre Darwina podnetom pre urýchlené vyhotovenie náčrtu teórie, ktorý vyšiel v roku 1859 pod názvom *O pôvode druhov prírodným výberom alebo zachovávanie zvýhodnených rás v boji o život*.

populačnej genetiky a teórie prírodného výberu, k vytvoreniu *syntetickej teórie evolúcie*.⁹

Zásadným prelomom v biologických vedách a aj v teórii evolúcie bolo poznanie procesov dedičnosti na molekulárnej úrovni. Odhalenie materiálneho nositeľa dedičnosti a jeho pozoruhodných vlastností bolo základom revolúcie v biologických vedách, ktorá ukázala cestu k poznaniu a pochopeniu života na jeho elementárnej úrovni.

Samozrejme, že poznatky získané v rozličných disciplínach, od genetiky cez embryológiu až po paleontológiu viedli k sporom o povahu evolučného procesu, a teda aj o adekvátnu podobu teórie evolúcie. Naplno sa prejavil spor medzi darvinistami a nedarvinistami. Spory sa však viedli aj v rámci darvinizmu, takže nemôžeme hovoriť o teórii evolúcie, ale skôr o teóriách evolúcie. Centrálnym bodom sváru sa stalo postavenie a úloha prírodného výberu, jeho vzťah k procesom dedičnosti a premenlivosti. Povaha diskusií často naberala podobu ideologických stretov ako vedeckej diskusie (napokon samo označovanie *darwinizmus*, *lamarckizmus* atď. vyvoláva skôr ideologické ako vedecké konotácie), pričom sa strácal a zamlžoval problém, ktorý bol pôvodne nastolený. Napriek tomu došlo k rozpracovaniu a presadeniu sa teórie evolúcie, v ktorej hrá dôležitú úlohu proces selekcie reprodukujúcich sa štruktúr.

V poslednej tretine dvadsiateho storočia došlo k významnému prieniku teórie biologickej evolúcie do oblasti sociálnej evolúcie, vrátane sociálnej evolúcie človeka. Hamiltonova teória príbuzenského výberu sa vďaka Dawkinsovej popularizácii v teórii sebeckého génu stala východiskom pre skúmanie evolučných (biologických) základov ľudskej sociability a kultúry. Dawkinsova abstrakcia na úroveň replikátorov ako všeobecného vyjadrenia sebareprodukujúcich sa štruktúr vytvorila predpoklady pre vytvorenie *univerzálnej teórie evolúcie* na báze selekcie, diferencovaného prežívania reprodukujúcich sa štruktúr, *univerzálneho darvinizmu*. Tým sa vytvorili teoretické podmienky pre nástup matematických metód, vytvárania matematických modelov evolučných procesov na báze algoritmov, ktoré sú cestou k pochopeniu evolúcie na jej rozličných úrovniach.

4 Úrovne evolučného procesu

Charakteristickou črtou evolúcie je vytváranie kvalitatívne rozličných druhov štruktúr. Každý z nich sa vyznačuje sebe vlastnými zákonmi, vzťahmi usporiadanosti a usporadúvania. Spôsob organizovania štruktúr vytvára úrovne evolučného procesu. Vysvetlenie vzniku jednotlivých úrovní a vzniku a fungovania vlastností a vzťahov charakteristických pre každú úroveň je jednou zo základných úloh teórie evolúcie. Ďalšou nemenej dôležitou otázkou je vysvetlenie vzťahov medzi rozličnými úrovňami, ako aj povaha týchto vzťahov.

9 Termín *neodarvinizmus* zaviedol Georg Romanes v roku 1905, aby odlíšil teóriu prírodného výberu opierajúcu sa o poznatky rodiacej sa genetiky od teórie dedičnosti získaných vlastností. Neskôr sa začal používať a dodnes sa rozšírene používa ako označenie syntetickej teórie evolúcie.

4.1 Evolúcia ako proces produkovania kvalitatívne odlišných úrovní organizácie

Vymedzenie úrovni evolučného procesu by zdanlivo nemal byť problém. Ani tu však nemôžeme hovoriť o zhode. Rozdiely panujú nielen v tom, že sa niekedy hovorí o *úrovních*, inokedy o *stupňoch*, *fázach* alebo o *štádiách* evolučného procesu, ale aj v tom, aká je povaha rozdielov medzi úrovňami a ako chápať povahu ich vzájomných vzťahov. Podľa prevládajúceho názoru tvorí evolúcia *hierarchický systém úrovní*, v ktorom z „nižších“ úrovní vznikajú „vyššie“. Platí to tak pre evolúciu v širokom zmysle slova, ako aj pre evolúciu živých organizmov. Evolúciu by sme si mali predstaviť ako pyramídu či špirálu stúpajúcu k čoraz vyšším úrovňam, pričom hlavnou ambíciou teórie evolúcie by malo byť vysvetlenie prechodu z jednej úrovne do druhej, zákony vzniku kvalitatívne nových úrovní.

Postupnosť úrovní evolučného procesu, keď z jednoduchších, menej dokonalých, nižších úrovní vznikajú zložitejšie, dokonalejšie, vyššie úrovne, má charakter nevyhnutnosti – procesy na nižšej úrovni v určitom bode jej vývoja dospejú do štádia, kedy nevyhnutne dochádza k prechodu na vyššiu úroveň. Táto postupnosť je navyše usmernenou nevyhnutnosťou, ktorú riadi vnútorný zákon evolúcie, imanentný zákon pohybu vyvíjajúcich sa podstát, vnútorná, objektívna *logika vývoja* atď. Podľa týchto predstáv je evolúcia proces, ktorý z určitého východiskového bodu, z počiatočného stavu, postupnosťou štádií nevyhnutne smeruje k svojmu završeniu, k cieľu. Pod úrovňami, štádiami atď. evolúcie sa chápu nové stupne vo vývoji idey, nové formy pohybu hmoty, nové úrovne kozmogenézy, stupne zložitosti atď. Každá nová úroveň predstavuje kvalitatívne nové, vyššie štádium vývoja. Od predchádzajúcich sa odlišuje novými vlastnosťami, ktoré nie sú vysvetliteľné z vlastností starej, nižšej úrovne, je charakteristická nárastom zložitosti v štruktúre súcien, ktoré tvoria jej základné konštitutívne zložky, a v spôsoboch ich existencie, ale aj nárastom zložitosti v štruktúre úrovne samej, ktorá je daná vzájomnými vzťahmi medzi súciami, zákonitosťami, ktoré sú jej vlastné, miestom, ktoré zaujíma v evolučnom procese a vzťahmi k jeho ostatným úrovňam.

Z hľadiska tohto prístupu je evolúcia procesom nevyhnutného progresívneho vývoja, ktorý smeruje od vzniku vesmíru cez vznik a vývoj života k vzniku a vývoju človeka, ľudského vedomia a spoločnosti. Pre koncepcie opierajúce sa o takéto chápanie evolučného procesu je biologická evolúcia len jednou z jeho súčastí, ktorá však zohráva zásadnú úlohu. Završením biologickej evolúcie je vznik človeka, ľudského správania, vedomia, jazyka, ducha, čo predstavuje zlomový bod v evolučnom procese. Evolúcia prechádza do novej fázy, do štádia, ktorým prekračuje rámec biologickej evolúcie, ale pre mnohých autorov aj hranice prírodnej, prirodzenej evolúcie. Evolúcia ľudskej spoločnosti, vedomia, ľudského ducha nie je len završením biologickej evolúcie: znamená završenie evolúcie ako celku, dosiahnutie jej vnútorného zmyslu (ktorým môže byť stotožnenie sa s absolútnou ideou, vznik komunistického ľudskej spoločnosti, vytvorenie noosféry, sféry vývoja vedomia až k bodu splynutia s kozmickým Kristom, vytvorenie globálnej civilizácie atď.).

Medzi progresivistické koncepcie evolúcie opierajúcimi sa o základný vnútorný (imanentný) zákon vývoja patria tak filozofické teórie, ktoré vytvárajú univerzalizujúce

a totalizujúce (nazerajúce evolúciu, bytie, stávanie v jeho celostnosti) teórie vývoja (nájdeme tu teórie univerzálnej vychádzajúce z dialektickej povahy zákonov vývoja – Schelling [90], Hegel [49], Marx a Engels [26] a ich nasledovníci; opierajúce sa o nejakú vede neznámu a vedou nepostihnuteľnú vnútornú silu, napr. Bergsonov *élan vital* [6], prípadne o zákon rastu zložitosti, komplexikácie – Hartmannovu *novú ontológiu* [46], Teilharda de Chardin [98], ktorého koncepciu na Slovensku rozvíja Ján Letz a ďalšie), ale aj koncepcie usilujúce sa o vytvorenie univerzálnej teórie evolúcie ako vedeckej teórie. V druhom prípade ide buď o pokus vytvoriť vedeckú teóriu vývoja zovšeobecnením niektorého zo známych fyzikálnych zákonov (Spencerova teória [94], [95]), v ktorej bol univerzálnym zákonom evolúcie zovšeobecnený termodynamický zákon; rozličné verzie antropického princípu odvolávajúce sa na základné fyzikálne konštanty a ich vzťah k existencii a k štruktúre vesmíru [4], [21]; Kauffmanova teória zákonitého rastu zložitosti, ktorú odvodzuje z teórie chaosu [55] a i.). Napriek proklamovanej snahe o vedeckosť však tieto teórie prekračujú rámec overiteľných výpovedí a pohybujú sa na neistej pôde filozofických úvah, ktoré do veľkej miery predchádzajú a zároveň výrazne determinujú výber a spôsob formulácie „vedeckého“ univerzálneho zákona evolúcie.

S rozmanitosťou úrovní reprezentujúcich rozličné podoby zložitosti sa nestretávame len v rámci evolúcie ako univerzálneho procesu. Každá úroveň evolučného procesu predstavuje zložitú organizáciu, stavbu, funkcionálnych vzťahov. Osobitne sa to prejavuje v živej prírode, ktorá predstavuje „úroveň úrovní“. Nemožno sa čudovať, že aj medzi evolucionistami nájdeme zástancov nevyhnutnosti progresivity evolúcie, jej smerovania od nižších, menej dokonalých foriem, k čoraz zložitejším, vyšším formám, pričom jej završením je vznik človeka. Silou, ktorá má usmerňovať evolúciu smerom k vyššej progresivite je podľa niektorých darvinistov prírodný (prirodzený) výber. Svoje úvahy opierajú o vytvorenie vedeckého kritéria rastu progresivity, ktorým by mohol byť vznik vyššej úrovne biologickej efektivity definovanej ako *vzrast kontroly a nezávislosti na prostredí* alebo ako *schopnosť získavať a spracovať informáciu o prostredí* [2].

Chápanie evolúcie ako procesu nevyhnutne usmerneného rastu progresivity naráža na niekoľko základných problémov. Prvým je vymedzenie progresivity – doteraz nejestvuje uspokojivá definícia progresívneho vývoja. Problémom nie je len vymedzenie kritérií progresivity, ktoré by nevychádzali z metafyzických predpokladov postulujúcich progresivitu ešte pred vymedzením kritérií, na základe ktorých sa bude posudzovať, a opierali sa o vedecké (overiteľné) poznanie objektívnych vlastností, ale aj vágnosť definícií progresu ako rastu zložitosti, „vyššieho“ či „nižšieho“ stupňa vývoja atď. Nevyhnutnosť progresívneho charakteru evolúcie pritom nevyplýva z teórií ani z poznatkov, ktoré zatiaľ máme k dispozícii. Ak by progresívny charakter bol nevyhnutnou črtou evolučného procesu, musel by byť odvoditeľný z daných teórií a poznatkov. Ani to by však nestačilo. Museli by sme vedieť, že predpoklady, o ktoré sa opierame, teda príslušné teórie a poznatky, sú pravdivé. Z filozofie vedy však vieme, na aké problémy narazíme pri pokusoch o zdôvodnenie, potvrdenie či overenie pravdivosti vedeckých teórií (pozri napr. [81], [82]). Progresivistické chápanie evolúcie je skôr vyjadrením osobného postoja opierajúceho sa o určité metafyzické predpoklady podložené silným antropocentrizmom.

Tieto metafyzické predpoklady pritom nie sú ničím novým. V zásade sa opierajú o novoplatónsku predstavu sveta ako ontologického schodiska, rebríka či reťaze bytia. Podľa tejto predstavy je svet systémom hierarchicky usporiadaných štruktúr, od najjednoduchších po najzložitejšie, od najnižších po najvyššie, od najmenej dokonalých po najdokonalejšie. Kým v tých teoretických systémoch, ktoré niektorí zástancovia progresívnej evolúcie označujú ako *metafyzické*, je táto usporiadanosť daná stvoriteľom alebo povahou základných princípov, počiatkov, pričom všeobecný základ súcien, ich podstata je nemenná, v koncepciách zástancov progresívneho vývoja je základnou charakteristikou bytia zmena, pohyb, vývoj, vrátane zmeny, pohybu, vývoja podstaty – premena podstaty určitého druhu na iný, vyšší druh (na podstatu vyššieho rádu). Postulovanie týchto princípov je však v rozpore s odhaľovaním reálnych procesov premien v živom i neživom svete.

Aká je možná alternatíva? Ak sa vrátíme k Darwinovej téze o kľúčovej úlohe náhody v biologickej evolúcii, potom môžeme evolučný proces nazerať ako realizáciu niektorých možností v závislosti od interakcie s podmienkami prostredia. Náhoda pritom do evolúcie vstupuje na všetkých úrovniach – na úrovni reprodukcie, ktorá je vďaka náhodným nepresnostiam, chybám, procesom prekríženia atď. zdrojom variability, na úrovni konfrontácie s prostredím, v ktorom sa vlohy získané reprodukciou majú možnosť prejaviť a musia sa osvedčiť. Zmeny prostredia sú pritom náhle, neovplyvniteľné činnosťou organizmov a ich dôsledky mávajú apokalyptické dôsledky. Evolúcia sebareprodukujúcich sa štruktúr nie je primárne procesom generujúcim čoraz zložitejšie a dokonalejšie štruktúry, ale dosahovaním reprodukčnej úspešnosti vzhľadom na meniace sa podmienky prostredia. Evolúcia nie je absolútnym, ale relatívnym procesom, v ktorom možno zložitost' posudzovať len vo vzťahu určitej štruktúry k inej štruktúre, aj to len z určitého hľadiska. Vznik zložitých štruktúr je teda skôr vedľajším produktom rozmanitosti a náhodných zmien ako výsledkom vnútornej nevyhnutnosti. Evolúcia nie je stromom, ktorého vrcholom je vznik človeka, ale skôr krajinou, v ktorej sa objavujú rozličné vrcholy progresivity, rastu zložitosti štruktúrnej a funkčnej organizácie, ktoré na seba nemusia nevyhnutne nadväzovať, prechádzať jeden do druhého. Evolúcia je procesom, ktorý vytvára podmienky na realizáciu rozmanitých možností, sama tieto možnosti vytvára. Či sa však uskutočnia, ako dlho budú úspešné, to je otvorená otázka.

4.2 Hierarchia úrovní a problém redukcie

Hierarchickú organizovanosť možno nazerať dvojakým spôsobom. Prvý vychádza z progresivistického stanoviska, v ktorom má hierarchia úrovní podobu pyramídy. Každá vyššia úroveň vyrastá z procesov na nižšej úrovni. V určitom bode evolúcie dochádza ku kvalitatívnej zmene, k vzniku nových procesov, zložitejších štruktúr a vzťahov medzi nimi. Vzniká nová úroveň, ktorá prekonáva, prerastá pôvodnú úroveň, je „nad“ ňou, obsahuje, zahŕňa ju ako svoj moment, svoju súčasť. Tak úroveň sociálnej evolúcie obsahuje biologický základ, vznik človeka s vlastnosťami, ktorými prekračuje biologický svet, prírodu ako bezprostredné prostredie vývoja sociálnosti. Biologická evolúcia zasa vyrastá z chemických procesov, chemická evolúcia z fyzikálnych procesov. Na každej úrovni sa prejavujú procesy nižších úrovní, každá nová úroveň

však predstavuje vyššiu zložitosť vzťahov, ktorá sa nevyčerpáva, neredukuje na procesy nižších úrovní. Z toho je zrejماً obmedzenosť prírodných (empirických) vied – keďže tieto postupujú predovšetkým metódou redukcie (rozkladu celku na jeho časti a ich vzájomné vzťahy), nemôžu postihnúť zložitosť vzťahov na vyšších úrovniach, nie sú schopné postihnúť úroveň ako celok, celosnosť, jednotu atď. Redukciu ako metódu prírodných vied treba doplniť o holistický prístup, ktorý odhalí zákonitosti na úrovni celku.

Druhý prístup sa opiera o neprogresivistické stanovisko. Hierarchia úrovní nemá podobu pyramídy, hovorí sa skôr o *zahniezdení*, *vnorení* nových úrovní evolučného procesu. Nové úrovne neobsahujú staré ako svoju súčasť, naopak, sú v nich obsiahnuté. Evolúcia „vyšších“ úrovní nemôže prekračovať nižšie úrovne, musí prebiehať v ich rámci. Žiadny proces evolúcie živých organizmov nemôže prebiehať „mimo a nad“ fyzikálnymi či chemickými procesmi (zákonmi). Živé organizmy nejestvujú „mimo“ fyzikálneho sveta, sú *zahniezdené* vo fyzikálnom svete. Všetky procesy musia byť vysvetliteľné ako fyzikálne procesy, treba odhaliť ich fyzikálne pozadie. To neznamenaá redukovať zložitosť vzťahov na úrovni celku na vzťahy medzi časťami danej úrovne. Poznanie skladby nejakej štruktúry a vysvetlenie vzťahov jej konštitutívnych častí je však jedným z predpokladov jej poznania. Opis procesov z holistického stanoviska, nech už sa pod ním rozumie čokoľvek, bude neúplný, pokiaľ sa nebude opierať o poznanie procesov na úrovni častí. Bunku ako celok nepochopíme, pokiaľ nepochopíme, čo sa deje na úrovni jej molekulárnej štruktúry.

Rovnako nevyhnutné je aj poznanie nejakej štruktúry ako celku (bunky, organizmu, spoločenstva). Ale aj k celku môžeme a musíme pristupovať z rozličných hľadísk. Neexistuje disciplína, ktorá by z hľadiska daného celku mala privilegovaný prístup k vysvetleniu príslušných procesov. Na dosiahnutie úplného opisu a pochopenia je nevyhnutný široký multidisciplinárny prístup, spolupráca rozmanitých disciplín, ktoré odhalia rôznorodé stránky, charakteristiky príslušného celku. Môžeme sa síce domnievať, že napríklad procesy sociálnej evolúcie sú výhradnou doménou spoločenských vied (prirodzene, ak pokladáme sociálnu úroveň za „najvyššiu“, ťažko pristúpime na myšlienku, že veda opisujúca procesy na nižšej úrovni nám môže byť v niečom nápomocná). Ak však sociálnu evolúciu nebudeme pokladať za niečo, čo je „nad“ biologickou evolúciou, nehovoriac o ešte „nižších“ úrovniach evolučného procesu, ale za niečo, čo prebieha v konečnom dôsledku v hraniciach vymedzených fyzikálnymi zákonmi, skôr alebo neskôr dospejeme k presvedčeniu, že aj procesy na úrovni sociálnej evolúcie ako určitého celku majú svoju fyzikálnu stránku. Odhalenie fyziky týchto procesov je preto neoddeliteľnou súčasťou úplného vysvetlenia príslušných procesov evolúcie na každej úrovni.

5 Univerzálny darvinizmus a univerzálna teória evolúcie

Rozvoj evolučnej teórie v biologických vedách viedol v poslednej tretine 20. storočia k intenzívnemu skúmaniu tých ľudských vlastností, ktoré sa pokladajú za doménu spoločenských a humanitných vied – ľudského správania, vrátane sociálneho správania, vedomia, poznávania, morálky. Objavil sa celý rad disciplín s prívlastkom „evolučná“ (evolučná epistemológia, evolučná psychológia, evolučná medicína, ale aj

evolučná ekonomika a evolučná estetika), ktoré sa opierajú o darvinistický výklad evolúcie vlastností, procesov, vzťahov, ktoré sa stali predmetom ich skúmania. Tieto disciplíny sa opierajú o dva predpoklady. Prvým predpokladom je človek ako produkt biologickej evolúcie – ak je výsledkom evolúcie stavba a fungovanie ľudského tela, ak je výsledkom evolúcie mozog, potom sú výsledkom evolúcie aj ľudské vedomie (mysel), ľudské správanie, ľudské kognitívne schopnosti. Druhým predpokladom je analogickosť procesov evolúcie reproduktívnych sa štruktúr – aj tie prebiehajú na rôznych úrovniach na základe analogických alebo dokonca rovnakých princípov.

5.1 Evolúcia a poznanie: evolučná epistemológia

Hoci viaceré východiská pre skúmanie kognitívnych vlastností človeka načrtol už Darwin a v priebehu 20. storočia sa nimi zaoberali viacerí autori, o systematickom výskume a jeho skutočnom rozmachu môžeme hovoriť až v jeho poslednej tretine. Jednou z prvých „aplikovaných“ evolučných disciplín sa stala *evolučná epistemológia* (evolučná teória poznania), ktorá sa v sedemdesiatych a osemdesiatych rokoch vyprofilovala do pomerne silného smeru skúmania.

Evolučná epistemológia, sa od počiatkov rozvíjala v dvoch hlavných smeroch – v smere, ktorý má bližšie k vedeckému skúmaniu evolúcie poznania (K. Lorenz, D. T. Campbell, R. Riedl, G. Vollmer, F. M. Wuketits a ďalší¹⁰) a v smere, ktorý predstavuje skôr filozofickú aplikáciu evolučných myšlienok na skúmanie vývoja poznania, predovšetkým vedeckého poznania (K. R. Popper, S. T. Toulmin, D. L. Hull, R. J. Richards).

Evolučná epistemológia vychádza z tézy, že človek je produktom biologickej evolúcie, to znamená, že aj jeho vlastnosti, najmä správanie a kognitívne schopnosti, sú produktom evolúcie. Evolučná epistemológia ide však výrazne za tézu o ľudskom poznaní ako výsledku evolúcie. Podľa Lorenza, Riedla a ďalších autorov poznanie nie je výlučnou charakteristikou človeka, naopak, poznanie pokladajú za jeden zo základných prejavov života. Kľúčové postavenie poznania (poznávania) sa prejavuje v zmene v chápaní teórie evolúcie – predstavitelia vedeckého smeru evolučnej epistemológie tvrdia, že by sme nemali hovoriť o skúmaní evolúcie poznania v rámci teórie evolúcie, ale o evolúcii ako o procese získavania poznania. Jedným z hlavných cieľov je preformulovať pôvodne filozofický pojem poznania ako vedecký pojem, ako schopnosť prijímať, spracúvať a odovzdávať informácie. Vysvetlenie poznania, jeho evolúcie je potom cestou, ako vybudovať epistemológiu na vedeckých základoch, čo bude viesť k vytvoreniu mosta na prekľenie priepasti medzi prírodnými vedami na jednej strane, a spoločenskými a humanitnými vedami na druhej strane. Dôsledkom bude nielen odstránenie zbytočnej a zavádzajúcej metafyzickej špekulatívnosti, ktorou sa vo väčšej či menšej miere vyznačuje každá filozofia, ale zároveň vytvorenie predpokladov na odstránenie tradičných filozofických dichotómií typu hmota –

10 Nezávisle od týchto autorov na Slovensku rozvíjal myšlienky veľmi blízke evolučnej epistemológii Ladislav Kováč [56]. Svoje myšlienky publikoval v absurdnej dobe potláčania občianskych slobôd v mene pokroku daného nevyhnutnosťou „neúprosnej logiky dejín“. Hoci sa napokon podarilo článok publikovať v časopise *Biologické listy*, celý náklad bol zlikvidovaný.

vedomie (duša, duch), subjekt – objekt, subjektivismus – objektivismus, racionalizmus – empirizmus, deduktivismus – induktivismus, ľudská spoločnosť – príroda atď. (pozri napr. [87]).

Evolučná epistemológia vyrástla predovšetkým na etológii, porovnávacom výskume správania živých organizmov. To však neznamená, že poznanie sa obmedzuje na správanie živých organizmov. Poznanie sa chápe v značne širokom význame, keď všetky vlastnosti, ktoré živé organizmy získavajú v procese evolúcie vrátane stavby tela, spôsobu fungovania orgánov, správania atď. sú výsledkom poznávania zákonitostí prostredia, v ktorom príslušný organizmus žije. Napríklad plutva je poznaním hydrodynamických vlastností vody, krídlo poznaním zákonitostí prúdenia vzduchu, oko je poznaním optických vlastností svetla atď. Riedl [86] v tejto súvislosti poznamenáva, že fyzici len znovuobjavili zákonitosti, ktoré už dávno pred nimi objavili a z úspechom využívali viaceré druhy organizmov. Rôzne variácie v stavbe a fungovaní živých organizmov sú dokladom určitého druhovošpecifického poznania daného prostredia. Evolučná epistemológia reformuluje a rieši Kantov problém apriórnych základov poznania – z pohľadu vývoja druhu, teda z pohľadu fylogénzy, je poznanie, ktoré si daný druh osvojí, aposteriórne, vytvára sa v procese jeho fylogénzy. Na druhej strane každé individuum daného druhu prichádza na svet zo všetkými vlastnosťami, ktoré určujú jeho druhovú príslušnosť, sú zakódované v genetickej informácii, ktorú zdedilo od svojich rodičov. Z pohľadu individua je toto poznanie apriórny, predchádza každú jeho možnú skúsenosť, na jeho základe rozvíja svoje individuálne poznávanie prostredia, v ktorom žije.

Predstavitelia evolučnej epistemológie sa domnievajú, že postavenie epistemológie na bázu empirickej vedy vedie k vyriešeniu aj takého problému, ktorý je základným problémom filozofie a na ktorý nedokázala filozofia dať za 2 000 rokov uspokojivú odpoveď – problém pravdivosti poznania. Toto riešenie je veľmi jednoduché a priamo vyplýva z chápania evolúcie živých organizmov ako evolúcie poznania. V procese evolúcie dochádza k odstraňovaniu nevhodných odchýlok a k zachovávaní vhodných odchýlok. To znamená, že evolúcia je procesom odstraňovania nesprávneho, neadekvátneho, nepravdivého poznania a zachovávaní správneho, adekvátneho, pravdivého poznania. Stavba tela, fungovanie orgánov spôsob správania, to všetko predstavuje poznanie prostredia, v ktorom organizmus žije. Pokiaľ je toto poznanie adekvátne prostrediu, pokiaľ mu zodpovedá, potiaľ umožňuje organizmu v danom prostredí prežiť. Pokiaľ však nezvyšuje pravdepodobnosť prežitia, je neadekvátnym, nesprávnym, nepravdivým poznaním prostredia a bude v procese selekcie eliminované. Problém pravdivosti poznania je tak podľa predstaviteľov vedeckého smeru evolučnej teórie poznania riešiteľný ako vedecký problém v termínoch prírodného výberu, adaptácie (adaptovanosti, adaptovateľnosti) a prežívania zdatných.

Vedecký smer evolučnej epistemológie otvoril celý rad konkrétnych empirických problémov, ktoré sú predmetom intenzívneho výskumu. Evolúcia kognitívnych schopností a ich výskum sa však medzitým stal predmetom záujmu zo strany *kognitívnej vedy*, filozofie mysle a ďalších disciplín [38], [39]. Keďže vo všetkých prípadoch ide o výskum založený na širokej interdisciplinárnej báze, dá sa očakávať ich postupné splývanie. Problémom vedeckého smeru však ostáva vágnosť pojmu

poznania, ako aj vzťah vedy a filozofie. Vzhľadom na osobitosť kladenia filozofických otázok a povahu filozofických problémov je diskutabilné, či sa postulovaním poznania ako predmetu vedeckého skúmania celkom zbavíme filozofického pozadia daného problému, či preformulovaním filozofického problému v jazyku vedy neriešime iný problém. Filozofia môže prísť o časť svojej kompetencie, filozofický problém vo vlastnom zmysle slova ostáva. Treba povedať, že v prípade apriórnosti poznania a pravdivosti poznania evolučná epistemológia neponúkla skutočné vyriešenie daných problémov, ale skôr odbočenie od vlastného problému [33], [34], [35]. Skutočne môžeme tvrdiť, že „pravdivosť“ vytvorenia nejakej štruktúry na základe „poznania“ zákonov je to isté ako pravdivosť jazykových výrazov? Ako v prípade jazykových výrazov uplatníme kritérium reprodukčného úspechu? O reprodukcii čoho hovoríme? O reprodukcii výrazu, alebo o reprodukcii jeho nositeľa? V dejinách nájdeme dostatok príkladov, ktoré nás učia, že ľži bývajú oveľa úspešnejšie, reprodukovujú sa oveľa jednoduchšie a v oveľa väčších objemoch ako pravda.

Druhý, filozofický smer evolučnej teórie poznania pristupuje takpovediac „zhora“. Podľa autorov, ako sú napr. K. R. Popper, S. T. Toulmin, D. L. Hull, R. J. Richards je biologická evolúcia špecifickým prípadom všeobecného vzoru evolučnej zmeny. Problémom je vytvorenie všeobecnej teórie, ktorá by opisovala všeobecný vzor evolučnej zmeny a bola by aplikovateľná (špecifikovateľná) na jednotlivé prípady (úrovne) evolučného procesu [99], [52]. Je pravda, že jednotliví autori sa líšia tak v otázke potreby takejto teórie, ako aj jej statusu i spôsobu vytvárania. Poppera zaujíma predovšetkým problém rastu vedeckého poznania, vychádza však z podobne všeobecnej tézy ako predstaviteľia vedeckého smeru, že poznanie je charakteristickou črtou života – život je riešenie problémov. Každý organizmus, či je to améba alebo Einstein, musí riešiť problémy, na ich riešenie vytvára rôzne hypotézy, vstupujúce do konfrontácie s prostredím, ktoré odstraňuje nepravdivé hypotézy. Evolúcia je procesom eliminácie omylov, nezávisle od toho, akým spôsobom sa generujú hypotetické riešenia problémov alebo aké dôsledky má eliminácia omylov (pre amébu môže znamenať smrť, pre Einsteina opravu vo výpočte) [82], [80].

Autori ako S. T. Toulmin, R. J. Richards či D. L. Hull sa neobmedzujú len na skúmanie vedeckého poznania, hoci u každého z nich hrá toto skúmanie dôležitú úlohu pri vytváraní všeobecnej teórie evolúcie. Na rozdiel od Poppera skúmajú rast vedeckého poznania aj z hľadiska vplyvu vonkajších faktorov, medzi ktoré patrí napr. súťaženie medzi rôznymi vedeckými skupinami, podmienenosť vedeckého výskumu jeho sociálnymi väzbami: Cieľom je skúmanie evolúcie vedy nielen ako určitého spôsobu poznania, ale aj ako inštitúcie, ktorá je produktom sociálnej evolúcie a zároveň faktorom, ktorý podstatným spôsobom túto evolúciu ďalej ovplyvňuje. Pojmy, s ktorými pracuje biologická teória, dostávajú v týchto teóriách iný obsah [99], [88], alebo slúžia ako východisko pre vytvorenie nových, všeobecných pojmov [52].

5.2 Teória evolúcie a problém sociálnej evolúcie

Pochopiteľne, každý zásah prírodných vied do sfér, ktoré pokladajú za svoje spoločenské a humanitné vedy, vyvoláva kontroverznú reakciu z ich strany. Naplno sa to prejavilo vo chvíli, keď evolučná teória vytvorila predpoklady pre skúmanie

sociálneho správania. Výskum rozličných spoločenstiev v živočíšnej ríši ukázal, že ľudské sociálne správanie, vytváranie zložitých nadindividuálnych sietí vzťahov, je len jedným z mnohých prípadov sociálneho správania v živej prírode. A navyše sociálne správanie živých organizmov vrátane človeka je produktom tých istých procesov, ktoré vedu k vzniku biologických druhov. Podľa Wilsona [102] sú základné prejavy sociálneho správania človeka, vrátane tých, ktoré sa pokladajú za výlučný dôsledok jeho vedomých (duševných, duchovných) schopností, teda to, čo často pokladáme za výnimočnú vlastnosť človeka, niečo, čo tvorí novú, vyššiu vrstvu evolučného procesu, v skutočnosti výsledkom pôsobenia tých istých príčin a je determinované tými istými procesmi ako napríklad „inštinktívne“ správanie hmyzu.

Vysvetlenie sociálneho správania predstavovalo pre darvinistické teórie evolúcie nemalý problém. Sociálne správanie znamená vzájomnú kooperáciu indivíduí, schopnosť obetovať sa v prospech iných členov spoločenstva, čo je v rozpore s tézou o súperení indivíduí v boji o prežitie. Ak je však súperenie (súťaž, boj) indivíduí a prežívanie zdatných (t. j. reprodukčný úspech) jednou z hybných síl evolúcie, ako potom vysvetlíme existenciu organizmov, ktoré sa vzdávajú vlastného rozmnožovania v prospech rozmnožovania iných jedincov, ako je to napríklad v prípade sterilných kást hmyzu? Možno vôbec kooperáciu a altruizmus vysvetliť v termínoch prírodného výberu?

Niektorí autori (medzi typických predstaviteľov patrí Lorenz a ostatní zástancovia vedeckého smeru evolučnej epistemológie) sa usilovali tieto problémy vyriešiť zavedením tzv. skupinovej selekcie. Základnou jednotkou evolučného procesu, jednotkou, ktorá je objektom pôsobenia prírodného výberu, nie je jednotlivý organizmus, ale druh (populácia). To, o čo v evolúcii ide, nie je prežívanie indivíduí, ale prežívanie druhov. Vznik kooperácie, altruizmu alebo existenciu sterilných kást sa vysvetľuje záujmom zachovania druhu.

William Hamilton [45] navrhol odlišné riešenie. Na základe svojich pozorovaní vytvoril teóriu príbuzenského výberu, ktorú podoprel matematickým aparátom. O čosi viac ako desať rokov sa jeho teória stala nesmierne populárna v podobe Dawkinsovej *teórie sebeckého génu* [dawkins 123].

Ústrednou myšlienkou Hamiltonovej teórie je tvrdenie, že základnou jednotkou evolúcie nie je jednotlivý organizmus, ani druh (populácia organizmov), ale gén. Inými slovami, v evolúcii nejde o prežívanie indivíduí alebo druhov (populácií), ale o reprodukciu a prežívanie génov. Prežívanie indivíduí alebo skupín indivíduí je v podstate spôsob, akým gény bojujú o svoje prežitie, akým sa rozširujú a prenášajú do ďalších generácií. Hamilton ukázal, že rozličné formy kooperácie a altruizmu, s ktorými sa stretávame v prírode a ktoré sú základom sociálneho správania tohto živočíšneho druhu, v podstate znamenajú nepriamu podporu reprodukcie svojich génov prostredníctvom podpory reprodukcie svojich príbuzných. O tento teoretický základ sa opiera Wilson pri vysvetľovaní evolúcie sociálneho správania človeka v *Sociobiológii*, opiera sa oň aj súčasná *evolučná psychológia*, ktorá sa niekedy pokladá za priameho nasledovníka sociobiológie.

Výsledkom skúmania v týchto disciplínach je poznanie, že prejavy sociálneho správania človeka, rozmanitosť foriem ľudskej kultúry sú podmienené biologickými základmi v oveľa väčšej miere, ako sme ochotní pripustiť. Zároveň však v procese

biologickej evolúcie, v procese, ktorý sa zakladá na replikácii a selektívnemu prežívaniu *biologickej informácie*¹¹, dochádza k vzniku a k reprodukcii informácie, ktorá sa relatívne osamostatňuje od reprodukcie biologickej informácie a ktorú môžeme označiť za *sociálnu informáciu*. O sociálnej informácii hovoríme na úrovni správania živých organizmov, komunikácie medzi individuami toho istého biologického druhu.

Sociálna informácia má viacero foriem – má podobu rozličných chemických a fyzikálnych signálov, na ktoré reagujú ostatní členovia príslušného spoločenstva. Vo väčšine prípadov sa bezprostredne prekrýva s biologickou informáciou, čo okrem iného znamená aj to, že sa jej obsah, spôsob jej kódovania a dekódovania odovzdáva z generácie na generáciu prostredníctvom genetickej informácie. Na určitej úrovni evolúcie, pravdepodobne na úrovni cicavcov, najmä u primátov, dochádza k dôležitému posunu – obsah sociálnej informácie, spôsob jej kódovania a dekódovania a jej odovzdávanie sa začína oddeľovať od genetickej informácie. Tento proces súvisí so zmenami v nervovej sústave, ktorá umožňuje kódovanie, uchovávanie a dekódovanie informácie získanej procesom učenia od ostatných príslušníkov druhu.

Postupné oddeľovanie sociálnej a biologickej informácie má pre evolúciu závažné dôsledky. Znamená naštartovanie *sociokultúrnej evolúcie*, keď jeden a ten istý biologický druh vytvára rozmanité formy sociálnej informácie, spôsoby jej odovzdávania z generácie na generáciu, teda rozmanité tradície v rámci jednej kultúry, rozmanité kultúry. V zárodočnej forme možno tento proces pozorovať u primátov. Je napríklad známe, že u šimpanzov toho istého druhu rodiny žijúce v rozličných teritóriách vytvárajú rozličné tradície – odlišnosti v sociálnom správaní, v druhoch a spôsoboch používania jednoduchých nástrojov pri zaobstarávaní potravy atď. –, teda akési *protokultúry*, ktoré sa uchovávajú a prenášajú z generácie na generáciu nie prostredníctvom genetickej informácie, ale výhradne procesom učenia. Tento proces dostáva nové dimenzie u človeka vďaka artikulovanej reči. Vytvára sa vhodné médium, ktoré je nositeľom kódovanej sociálnej (sociokultúrnej) informácie. Hoci sociálna (sociokultúrna) evolúcia ostáva závislá na biologickej evolúcii (je viazaná na vytvorenie a primerané fungovanie určitých telesných orgánov – mozgu, ústnej dutiny, perí, jazyka, zubov atď.), jej obsah, podoba, forma, spôsob jej odovzdávania atď. sú od tohto biologického základu nezávislé.

Dawkins bol zrejme prvý, kto si uvedomil, vznik informácie nového typu, ako skutočnosť, že jej reprodukcia nie je bezprostredne naviazaná na reprodukciu biologickej informácie. Popri géne zavádza nový hypotetický druh replikátora – *mém*¹², ktorý má byť nositeľom a základnou jednotkou sociokultúrnej evolúcie. Postulácia *mémov* ako replikátorov odlišných od génov znamená uvedomenie si a uznanie, že sociokultúrna evolúcia sa v niečom líši od biologickej evolúcie, že dochádza k relatívnemu oddeleniu replikácie biologickej a sociálnej informácie.

11 Pod *biologickou informáciou* rozumiem nielen genetickú informáciu, ale aj spôsob jej dekódovania, „interpretácie“ do stavby a fungovania organizmu, informačný tok vo vnútornom prostredí organizmu a medzi organizmom a jeho prostredím.

12 Výraz *mém* má vyjadrovať jednak zvukovú podobnosť s výrazom *gén*, jednak vzťah k ľudskej mysli (*mind*), ale má aj evokovať spôsob odovzdávania sociálnej informácie *napodobňovaním* (*mimézis*).

Dawkinsova práca vyvolala tvrdú kritiku¹³, ale aj široký ohlas, ktorý viedol k pokusu vytvoriť novú vednú disciplínu – *memetiku*, ktorej cieľom malo byť skúmanie sociokultúrnej evolúcie ako procesu reprodukčného úspechu mémov. Problémom však ostáva vymedzenie *mému* ako základnej jednotky sociálnej informácie. Hoci rozličné vymedzenia *mému* vychádzajú z predpokladu, že štruktúra a spôsob fungovania sociálnej informácie sa v zásade nelíši od podobných vlastností biologickej informácie, zásadným problémom memetického prístupu ostáva nejasnosť toho, čo by vlastne *mém* mal byť. Preto v literatúre nachádzame často výrazne odlišné definície tohto termínu (stačí porovnať napr. Dawkinsovo chápanie s Wilsonovým; pozri napr. [22], [7], [104], [1]).

5.3 Univerzálny darvinizmus

Je možné, že postulovanie *mému* ako základnej jednotky sociálnej (sociokultúrnej) evolúcie sa napokon ukáže ako omyl, ako slepá cesta vývoja teórie evolúcie. Dawkinsovo vymedzenie *mému* ako replikátora nového typu má pre ďalšie smerovanie teórie evolúcie nesmierny význam. Nie svojím obsahom, ale spôsobom nastolenia otvára cestu k vytvoreniu univerzálnej teórie evolúcie sebareprodukujúcich sa štruktúr, k *univerzálnemu darvinizmu*.

Termín *univerzálny darvinizmus* pravdepodobne ako prvý použil Richard Dawkins [23], pričom ho používa v dvoch významoch – vo význame univerzálnej teórie biologickej evolúcie a vo význame evolúcie akýchkoľvek sebareprodukujúcich sa štruktúr, vrátane sociokultúrnej evolúcie. Dawkins urobil dôležitý metodologický krok – oddelil evolúciu od konkrétneho materiálneho nositeľa, ktorý poznáme z evolúcie života na Zemi. Na to, aby došlo k evolúcii, potrebujeme ľubovoľnú entitu, ktorá bude nositeľkou informácie. Táto entita sa musí vyznačovať jednou dôležitosťou vlastnosťou – musí byť schopná vytvárať kópie informácie napríklad prostredníctvom kopírovania seba samej. Inými slovami, musí byť schopná replikovať sa, preto túto entitu nazval *replikátor*. Okrem toho potrebujeme entitu, ktorá bude schopná replikátory prenášať. Prenášače replikátorov nazval *vehikuly*. Nositeľ informácie, ktorá je základom biologickej evolúcie na Zemi, gén, je len jedným z možných prípadov replikátorov, ktorý si na svoje prežítie a šírenie vytvára vehikuly v podobe živých organizmov, ich rozmanitých tiel. Dawkinsova abstrakcia od konkrétneho nositeľa informácie a redukcia jeho vlastností na schopnosť replikácie vytvára základné podmienky pre formuláciu všeobecnej teórie evolučnej zmeny, pre vytvorenie všeobecnej, univerzálnej teórie evolúcie. Základom tejto teórie je replikácia entít, ktoré sú nositeľmi informácie – *replikátorov*, a ich diferenčné prežítie prostredníctvom

13 Nesmieme zabúdať na historickú skúsenosť s tzv. *sociáldarvinizmom*. *Sociáldarvinizmus* sa zakladal skôr na Spencerovej ako Darwinovej teórii evolúcie, interpretoval prežítie zdatných ako prežítie najlepších členov spoločnosti s kritériami zodpovedajúcimi skôr predstave malomeštiaka o spoločenskom úspechu ako vedeckej teórii. Uplatňovanie *sociáldarvinistických* myšlienok malo katastrofálne dôsledky a viedlo k rezervovanému postoju spoločenských a humanitných vied vo vzťahu k darvinistickým explanáciám procesov sociálnej evolúcie. Neskôr však viedlo k neoprávneným výčitkám napr. smerom k sociobiológii ako pokusu o znovuoživenie sociáldarvinizmu.

súperenia entít, v ktorých sa replikátory „vozia“, *vehikulov*. Dawkins je presvedčený, že táto teória by mala byť schopná opísať evolúciu života kdekoľvek vo vesmíre. Ak došlo alebo dôjde evolúcii života kdekoľvek vo vesmíre, tento proces bude prebiehať rovnako ako na Zemi – ako diferenčné prežívanie replikátorov prostredníctvom súperenia *vehikulov*. Keďže ide o všeobecné vyjadrenie Darwinovho princípu prírodného výberu, Dawkins túto teóriu označuje *univerzálny darvinizmus*.

Univerzálny darvinizmus nie je len teóriou biologickej evolúcie, nech by sa táto evolúcia odohrávala kdekoľvek vo vesmíre. Presnejšie povedané nie je len teóriou evolúcie, ktorá sa zakladá na replikácii a odovzdávaní biologickej informácie. Univerzálny darvinizmus má byť teóriou, ktorá bude schopná opísať a vysvetliť každú evolúciu systémov spočívajúcu na replikácii a odovzdávaní informácie, nezávisle od povahy tejto informácie. To predovšetkým znamená opis evolúcie systémov vyznačujúcich sa prenosom *sociokultúrnej* informácie, teda na opis sociokultúrnej evolúcie.

Hamiltonova teória príbuzenského výberu a následná Dawkinsova abstrakcia základného mechanizmu evolúcie na úroveň *replikátor-vehikul* mali niekoľko závažných dôsledkov pre ďalší vývoj teórie evolúcie. Hamiltonova teória umožňuje vysvetliť vznik a existenciu rozličných druhov sociálneho správania vrátane sociálneho správania človeka. Umožňuje *redukovať* rozličné prejavy sociálneho správania človeka na ich spoločný základ. Oveľa významnejším dôsledkom tejto abstrakcie je pochopenie algoritmického charakteru procesov evolúcie sebareprodukujúcich sa štruktúr. To znamená vytvorenie predpokladov pre uplatnenie matematických metód pri skúmaní evolučných procesov, hľadanie *evolučných algoritmov* a postupné vytváranie matematických modelov evolučných procesov ([68], [60], [61], [83], [77], [85] a i.), do ktorého vstupuje široké uplatnenie počítačových metód. Na začiatku 21. storočia sa tak otvárajú nové dimenzie teórie evolúcie. Nielenže sa vytvárajú možnosti pre skúmanie a vysvetlenie zdanlivo veľmi odlišných procesov (môžeme porovnať napríklad vysvetlenie evolučných zmien opierajúcich sa o zmeny vo frekvenciách génov versus vysvetlenie zmien v ekonomických procesoch), ale zásadným spôsobom sa mení charakter teórie evolúcie, ktorá sa z formy „rozprávania o udalostiach“ mení na štandardnú formalizovanú vedeckú teóriu. Čo je však nemierné dôležité, vytvárajú sa predpoklady pre užšiu spoluprácu prírodných vied na jednej strane a spoločenských a humanitných vied na druhej strane.

Postupná univerzalizácia a matematizácia teórie evolúcie vyvoláva pochopiteľne celý rad otázok. Ak pominieme neustále sa vynárajúci problém redukcie, kľúčovou otázkou bude konfrontácia matematických modelov s dátami získanými v rozličných vedách, ako aj postupné spresňovanie používaných termínov a predovšetkým zmena v ich obsahu pri skúmaní procesov na rozličných úrovniach evolúcie. Predovšetkým na úrovni sociokultúrnej evolúcie, ktorá predstavuje multiúrovňový proces, bude treba precízne formulovať reprodukujuce sa štruktúry na jeho jednotlivých úrovniach, ich podobnosti, rozdiely, procesy ich selekcie a *prežívania zdatných*. Možno nám to umožní pochopiť aj také nuansy našej existencie, akou je zvláštnosť posudzovania kvality ľudského života a kvality ľudskej osobnosti na základe úspešnej reprodukcie finančných prostriedkov na osobnom konte v úspešne sa reprodukujujúcich bulvárnych

médiách. Pravda, oveľa dôležitejšie je pochopenie evolučných procesov v záujme zachovania existencie ľudského rodu.

6 Záver

Jazyk každej vedeckej disciplíny, každá vedecká teória vrátane teórie evolúcie obsahuje výrazy, ktoré nie sú v tomto jazyku definované (pričom mnohé z nich v ňom nie sú ani definovateľné). Keďže tieto výrazy často vstupujú do definovania základných termínov danej disciplíny či súboru disciplín alebo vied, výsledkom sú vágne alebo nejednoznačné definície týchto termínov, ktoré sú jednou z príčin možných rozdielnych interpretácií. [rozličné interpretácie opierajúce sa o rozličné filozofické východiská, ktoré môžu spätne ovplyvňovať rozličné ontológie.

Ontológie ako teórie bytia si robia nárok byť univerzálnymi teóriami, teóriami všetkého. Ich cieľom je sformulovať ani nie tak teóriu sveta, ale skôr východisko pre pochopenie sveta, a teda aj pre interpretáciu všetkého (nielen vedeckého) poznania o svete (hoci vedecké poznanie tvorí jadro takéhoto interpretačného rámca). Ich cieľom nie je vysvetliť, čo je to svet, ale pochopiť ho, nájsť jeho zmysel a následne miesto a zmysel ľudskej existencie (individuálnej, teda aj svojej vlastnej, ale aj existencie ľudského rodu) v ňom. Do takejto interpretácie nevyhnutne vstupujú hodnotové aspekty, buď v skrytej alebo explicitne vyjadrenej podobe. Ontológie však predstavujú problém z hľadiska statusu svojich tvrdení. Hoci sa totiž mnohé ontologické systémy pokladajú za vyvrcholenie vedeckého poznania a za jeho zavŕšenie (mnohé z nich postulujú univerzálne zákony bytia platné pre všetky jeho úrovne), ich tvrdenia (a teda ani nimi sformulované zákony) nemajú povahu vedeckých tvrdení, nie sú overiteľné, testovateľné, vyvrátiteľné. Nie je možné určitú filozofickú pozíciu podporiť (a už vôbec nie zdôvodniť) nejakou vedeckou teóriou alebo poznatkom. Filozofická pozícia je len cestou k novej interpretácii. Takmer každá ontológia vidí samu seba ako tú správnu, postuluje nevyhnutnosť svojho osvojenia, pretože jej pochopenie má byť východiskom zo súčasnej krízy (pozri napr. [93]).

Problém však nie je v rozmanitosti ontológií a už vôbec nie v určovaní tej správnej ontológie. Oveľa dôležitejšie ako hľadať správnu ontológiu je prehĺbovať vedecké poznanie sveta, pochopenie evolučných procesov na základe overiteľných poznatkov a teórií a analýza spôsobov, akým používame pojmy, aby sme si navzájom rozumeli. Základom riešenia problémov je porozumenie – porozumenie problémom, ale aj vzájomné porozumenie pri hľadaní rozličných ciest k ich riešeniu.

Podakovanie: Tento príspevok vznikol vo Filozofickom ústave SAV za podpory grantovej agentúry VEGA v rámci riešenia grantovej úlohy č. 2/0207/09 „Metodologické a filozofické problémy prírodných, sociálnych a humanitných vied“.

Literatúra

- [1] Aunger, R. (ed.): *Darwinizing Culture*. Oxford University Press, Oxford, 2000.
- [2] Barlow, C. (ed.): *Evolution Extended*. The MIT Press, Cambridge (Mass.), 1994.
- [3] Barret, L. - Dunbar, R. - Lycett, J.: *Evoluční psychologie člověka*. Portál, Praha, 2007.
- [4] Barrow, J. - Tipler, F.: *Anthropological Cosmological Principle*. Oxford University Press, Oxford, 1996.
- [5] *Encyclopaedia Beliana, Eh-Gala*. IV. zv. Veda, Bratislava, 2005.
- [6] Bergson, H.: *Creative Evolution*. Henry Holt & Co, New York, 1911.
- [7] Blackmore, S.: *The Meme Machine*. Oxford University Press, Oxford, 1999.
- [8] Bowler, P. J.: The changing meaning of „evolution“. *Journal of the History of Ideas*, **36** (1975), s. 95-114.
- [9] Bowler, P. J.: *Evolution. The History of an Idea*. [1983], University of California Press, Berkeley, Revised edition, 1989.
- [10] Butler, S.: *Evolution, Old and New: Or the Theories of Buffon, Dr. Erasmus Darwin, and Lamarck, as compared with that of Mr. Charles Darwin*. E. P. Dutton&Co, New York, 1911.
- [11] Buzássyová, K. - Jarošová, A. a kol.: *Slovník súčasného slovenského jazyka A-G*. Veda, Bratislava, 2006.
- [12] Campbell, D. T.: *Evolutionary Epistemology*. In Radnitzky, G. - Bartley, W. W. (eds.): *Evolutionary Epistemology, Rationality, and the Sociology of Knowledge*. Open Court, La Salle, 1987.
- [13] Cmorej, P.: *Úvod do logickej syntaxe a sémantiky*. Iris, Bratislava, 2001.
- [14] Černík, V.: *Systém kategórií materialistickej dialektiky*. Pravda, Bratislava, 1986.
- [15] Černík, V. - Farkašová, E. - Viceník, J.: *Teória poznania*. 2. opravené vydanie. Pravda, Bratislava, 1987.
- [16] Darwin, E.: *Zoonomia or the Laws of Organic Life*. Vol. I, II. J. Johnson, London, 1796.
- [17] Darwin, E.: *The Temple of Nature*. J. Johnson, London, 1803.
- [18] Darwin, Ch.: *On the Origin of Species*. Murray, London, 1859.
- [19] Darwin, Ch.: *The Descent of Man and Selection in Relation to Sex*. Murray, London, 1871.
- [20] Darwin, Ch.: *Autobiography*. Watt&Co., London, 1931.
- [21] Davies, P.: *Jsme sami? O důsledcích případného objevu mimozemského života*. Archa, Bratislava, 1996.
- [22] Dawkins, R.: *The Selfish Gene*. Oxford University Press, Oxford, 1976.
- [23] Dawkins, R.: *Universal Darwinism*. In D.S. Bendall (ed.): *Evolution: From Molecules to Men*, Cambridge University Press, Cambridge, 1983, s. 403-425.

- [24] Dawkins, R.: *The Extended Phenotype*. Oxford University Press, Oxford, 1992.
- [25] *Encyclopedia Britannica*.
<<http://www.britannica.com/EBchecked/topic/73167/Charles-Bonnet>>
- [26] Engels, F.: *Dialektika prírody*. Pravda, Bratislava, 1976.
- [27] Evolution. <<http://www.geocities.com/levensverhaal/evolution.htm>>
- [28] Fajkus, B.: Změna – vývoj – evoluce. In Nosek, J.-Havlík, V. (eds.): *Evoluce a věda*. OPS, Nymburk, 2008, s. 115-120..
- [29] Filkorn, V.: *Úvod do metodológie vied*. Vydavateľstvo SAV, Bratislava, 1960.
- [30] Flegr, J.: *Zamrzlá evoluce*. Academia, Praha, 2006.
- [31] Flegr, J.: *Úvod do evoluční biologie*. Academia, Praha, 2007.
- [32] Frege, G.: *Funktion, Begriff, Bedeutung*. Vandenhoeck&Ruprecht, Göttingen, 1994.
- [33] Gálik, D.: K niektorým problémom evolučnej teórie poznania. *Filozofia* **47** (1992), s. 458-466.
- [34] Gálik, D.: Dva smery evolučnej teórie poznania. *Filozofia*, **48** (1993), s. 786-791.
- [35] Gálik, D.: Evolučná teória poznania a problém apriórnych základov poznania. *Filozofia*, **50** (1995), s. 353-360.
- [36] Gálik, D.: Univerzálny darvinizmus. In *Kognitívne vedy III. Zborník prednášok*. ChtF STU, Bratislava, 2000, s. 1-7.
- [37] Gálik, D.: Univerzálna teória evolúcie a problém mémov. In Nosek, J. (ed.): *Memy ve vědě a filosofii*. Filosofia, Praha, 2004, s. 133-146.
- [38] Gáliková, S.: *Úvod do filozofie mysle*. Honner, Martin, 2001.
- [39] Gáliková, S.: *Psyché*. Veda, Bratislava, 2007.
- [40] Goethe, J. W.: *O prírode a umení*. Pravda, Bratislava, 1982.
- [41] Gould, S. J.: *Ontogeny and Phylogeny*, Belknap Press, Cambridge, Mass., 1977.
- [42] Gould, S. J.: *Ever Since Darwin*. Penguin, London, 1991.
- [43] Gould, S. J.: On Replacing the Idea of Progress with an Operational Notion of Directionality. In Hull, D. - Ruse, M.: *The Philosophy of Biology*. Oxford University Press, Oxford, 1998, s. 650-668.
- [44] Gould, S. J.-Lewontin, R. C.: The Spandrels of San Marco and the Panglossian Paradigm. *Proceedings of the Royal Society*, **B 205** (1979), s. 581-598.
- [45] Hamilton, W. D.: The Genetical Theory of Social Behavior. *Journal of Theoretical Biology* **7** (1964), s. 1-52.
- [46] Hartmann, N.: *Nové cesty ontológie*. Pravda, Bratislava, 1976.
- [47] Havlík, V.: Ontologie evoluční perspektivy. In Nosek, J. - Havlík, V. (eds.): *Evoluce a věda*. OPS, Nymburk, 2008, s. 121-142.
- [48] Harper, D.: Evolve. *Online Etymology Dictionary*, 2001. <[Dictionary.com http://dictionary.reference.com/browse/evolve](http://dictionary.reference.com/browse/evolve)>
- [49] Hegel, G. W. F.: *Fenomenologie ducha*. Nakladatelství ČSAV, Praha, 1960.

- [50] Hegel, G. W. F.: *Logika ako veda*. Zv. I. Pravda, Bratislava, 1985.
- [51] Hegel, G. W. F.: *Logika ako veda*. Zv. II. Pravda, Bratislava, 1985.
- [52] Hull, D.: *Science as a Process*. University of Chicago Press, Chicago, 1988.
- [53] Hrušovský I. a kol.: *Predsokratici a Platón. Antológia z diel filozofov*. Zv. I. Epoque, Bratislava, 1970.
- [54] Jablonka, E. - Lamb, M. J.: *Epigenetic Inheritance and Evolution: the Lamarckian Dimension*, Oxford University Press, Oxford, 1995.
- [55] Kauffman, S.: *At Home in the Universe*. Penguin Books, London, 1996.
- [56] Kováč, L.: Úvod do kognitívnej biológie. *Biologické listy* **51** (1986), č. 3, s. 172-190.
- [57] Kováč, L.: Bioenergetika: kľúč k pochopeniu mozgu a mysle. In Kvasnička, V. - Kelemen, J. - Pospíchal, J. (eds.): *Modely mysle*. Európa, Bratislava, 2008, s. 100-126.
- [58] Kačala, J. - Pisárčiková, M. - Považaj, M.: *Krátky slovník slovenského jazyka*. Veda, Bratislava, 2003.
- [59] Kuzánsky, M.: *O učenej nevedomosti*. Pravda, Bratislava, 1979.
- [60] Kvasnička, V. - Pospíchal, J. - Tiňo, P.: *Evolučné algoritmy*. Vydavateľstvo STU, Bratislava, 2000.
- [61] Kvasnička, V. - Pospíchal, J.: Darwinovská evolúcia ako algoritmus. In Kvasnička, V. a kol.: *Umelá inteligencia a kognitívna veda I*. STU, Bratislava, 2009, s. 231-280.
- [62] Lorenz, K.: *Die Rückseite des Spiegels*. Piper, München, 1973.
- [63] Lorenz, K.: *Der Abbau des Menschlichen*. Piper, München, 1983.
- [64] Lorenz, K.: *Základy etologie*. Academia, Praha, 1993.
- [65] Markoš, A.: *Povstávání živého tvaru*. Vesmír, Praha, 1997.
- [66] Markoš, A.: *Readers of the Digest of Life*. Oxford University Press, Oxford, 2003.
- [67] Materna, P.: *Svět pojmů a logika*. Filosofia, Praha, 2000.
- [68] Maynard Smith, J.: *Evolution and the Theory of Games*. Cambridge University Press, Oxford, 1982.
- [69] Maynard Smith, J. - Szatmáry, E.: *The Major Transitions in Evolution*. W. H. Freeman, Oxford, 1995.
- [70] Mayr, E.: *Evolution and the Diversity of Life*. Belknap Press, Cambridge (Mass.), 1977.
- [71] Mayr, E.: *Growth of Biological Thought*. Harvard University press, Cambridge (Mass.), 1982.
- [72] Mayr, E.: *Toward a New Philosophy of Biology*. Belknap Press, Cambridge (Mass.), 1988.
- [73] Mayr, E. - Provine, W. B. (eds.): *The Evolutionary Synthesis*. Harvard University Press, Cambridge (Mass.), 1998.

- [74] McShea, D. W.: Complexity and Evolution: What Everybody Knows. In Hull, D. - Ruse, M.: *The Philosophy of Biology*. Oxford University Press, Oxford, 1998, s. 625-649.
- [75] Nosek, J.: Evoluční vs. transcendentální povaha poznání. In Nosek, J.-Havlík, V. (eds.): *Evoluce a věda*. OPS, Nymburk, 2008, s. 105-114.
- [76] Oken, L.: *Elements of Physiophilosophy*. Ray Society, London, 1847.
- [77] Pelikán, M.: Evolučné algoritmy. In Kvasnička, V. a kol: *Umelá inteligencia a kognitívna veda I*. STU, Bratislava, 2009, s. 335-354.
- [78] Platón: *Dialógy*. Zv. 3. Tatran, Bratislava, 1990.
- [79] Platón: *Ústava*. Pravda, Bratislava, 1980.
- [80] Popper, K. R.: *Objective Knowledge*. Revised Edition. Clarendon Press, Oxford, 1979.
- [81] Popper, K. R.: *The Logic of Scientific Discovery*. Routledge, London, 1995.
- [82] Popper, K. R.: *Conjectures and Refutations*. Routledge, London, 1996.
- [83] Pospíchal, J. - Kvasnička, V.: In silico simulácia etnických konfliktov. In Kvasnička, V. - Kelemen, J. - Pospíchal, J. (eds.): *Modely mysle*. Európa, Bratislava, 2008, s. 160-178.
- [84] Raup, D.: *O zániku druhů*. Nakladatelství Lidové noviny, Praha, 1991.
- [85] Riddley, M.: *Původ cnosti*. Portál, Praha, 2000.
- [86] Riedl, R.: *Biology of Knowledge*. John Wiley, Chichester, 1984.
- [87] Riedl, R.: *Die Spaltung des Weltbildes*. Paul Parey, Berlin, 1985.
- [88] Richards, R. J.: *Darwin and the Emergence of Evolutionary Theories of Mind and Behavior*. University of Chicago Press, Chicago, 1987.
- [89] Roe, S. A.: The development of Albrecht von Haller's views on embryology. *Journal of the History of Biology*, **2** (1975), s.167-190.
- [90] Schelling, F. W. J.: *Výbor z díla. Rané spisy*. Svoboda, Praha, 1990.
- [91] *Slovník cudzích slov (akademický)*. SPN, Praha, 2005.
- [92] *Slovník slovenského jazyka*. Vydavateľstvo SAV, Bratislava, 1959-1968.
- [93] Šmajš, J.-Krob. J.: *Evoluční ontologie*. Masarykova univerzita, Brno, 2003.
- [94] Spencer, H.: Progress: Its Law and Cause. *Westminster and Foreign Quarterly Review*, **67** (1857), s. 445-485.
- [95] Spencer, H.: *First Principles*. Williams&Norgate, London, 1862.
- [96] Strobachová, I.: Evoluce jako pokrok? In Nosek, J.-Havlík, V. (eds.): *Evoluce a věda*. OPS, Nymburk, 2008, s. 365-372.
- [97] *Synonymický slovník slovenčiny*. Veda, Bratislava, 2004.
- [98] Teilhard de Chardin, P.: *Vesmír a lidstvo*. Vyšehrad, Praha, 1990.
- [99] Toulmin, S.: *Human Understanding*. Oxford University Press, Oxford, 1972.
- [100] Trusted, J.: *Beliefs and Biology: Theories of Life and Living*. Macmillan Press, London, 1996.

-
- [101] Vollmer, G.: *Evolutionäre Erkenntnistheorie*. S. Hirzel, Stuttgart, 1987.
- [102] Wilson, E. O.: *Sociobiology: The New Synthesis*. Harvard University Press, Cambridge (Mass.), 1975.
- [103] Wilson, E. O.: *O lidské přirozenosti*. Nakladatelství Lidové noviny, Praha, 1993.
- [104] Wilson, E. O.: *Konsilience*. Nakladatelství Lidové noviny, Praha, 1999.
- [105] Wilton, D.: Evolution. Wordorigins.org, 2006. <<http://www.wordorigins.org/index.php/more/282>>
- [106] Wuketits, F.M. (ed.): *Concepts and Approaches in Evolutionary Epistemology*. Reidel, Dordrecht, 1984.
- [107] Wuketits, F. M.: *Evolution, Erkenntnis, Ethik*. Wissenschaftliche Buchgesellschaft, Darmstadt, 1984.
- [108] Wuketits, F. M.: *Evolutionary Epistemology and Its Implications for Humankind*. State University of New York, New York, 1990.

Prirodzená dedukcia

Matej KOŠÍK¹

Abstrakt. Formálne logické systémy umožňujú matematikom zapisovať dôkazy teorémov v tvare, v ktorom rozhodovanie, či je daný dôkaz správny alebo nie, už nestojí a nepadá na omylnosti človeka, ale toto rozhodnutie vie efektívne vykonať aj stroj. Prirodzená dedukcia je jedným z mnohých formálnych logických systémov. Jeho autori sa ho snažili definovať tak, aby bol najbližšie tomu, ako ľudia usudzujú. Text obsahuje úvod do problematiky, vysvetľuje význam jednotlivých odvodzovacích pravidiel a ukazuje ich použitie pri dokazovaní jednoduchých teorémov.

1 Úvod

Či je dôkaz nejakého tvrdenia napísaný formálne alebo neformálne, matematik vie rozhodnúť, či daný dôkaz má alebo nemá chybu. Na druhú stranu, človeku len prospeje, ak sa zoznámí s nejakým formálnym systémom. Ľahšie sa argumentuje v prípade, že sa stretne s chybnými dôkazmi. Je dobré uvedomiť si, aký význam majú pojmy: výrok, predikát, funkčný symbol, teória, model [1]. Je dobré vedieť, čo znamená, ak sa dokáže nejaké tvrdenie z nedefinovaných pojmov. Je dobré vedieť, čo znamená, keď sa nejaké tvrdenie dokáže z *nekonzistentnej teórie*. Aj keď je taký dôkaz bezchybný, nemá žiadnu váhu.

Výroková a predikátová logika sa učí na gymnáziu. Výroková logika je jazyk, ktorý pozostáva z výrokových atómov (p, q, r, \dots) a logických spojok ($\wedge, \vee, \Rightarrow, \Leftrightarrow, \neg$). Predikátová logika predstavuje rozšírenie výrokovkej logiky o univerzálny a existenčný kvantifikátor (\forall, \exists). V tomto dokumente zapisujeme predikátové formuly s kvantifikátorom v tvare $\forall x : a \bullet p$ alebo $\exists x : a \bullet p$, kde x je meno premennej, a je typ premennej a p je predikátová formula. S dokazovaním správnosti alebo nesprávnosti výrokov nie je výrazný problém. Pomocou tabuľkovej metódy je možné ukázať, či alebo daný výrok (formula výrokovkej logiky) tautológiou. Tabuľka 1 ukazuje príklad takeého dôkazu. Pravdivosť výrokov (formúl výrokovkej logiky) je možné dokázať takzvanou *tabuľkovou metódou*, tak ako v tab. 1. Podobne triviálna metóda už ale pre formuly predikátovej logiky nezaberá. Pri dôkaze pravdivosti formuly $(\forall x : a \bullet p \wedge q) \Rightarrow ((\forall x : a \bullet p) \wedge (\forall x : a \bullet q))$ je potrebné postupovať celkom inak.

¹ Fakulta informatiky a informačných technológií, STU Bratislava, Ilkovičova 3, 812 19 Bratislava, E-mail: kosik@fiit.stuba.sk

Musíme si zvoliť množinu axiémov a množinu odvodzovacích pravidiel. Axiómy sú také formuly, ktoré považujeme za pravdivé a ich platnosť ďalej už nedokazujeme. Odvodzovacie pravidlá nám pomôžu odvodiť nové platné formuly z iných platných formúl. Pravidlá môžeme zapisovať nasledovným spôsobom:

$$\frac{\text{predpoklad}_1 \quad \dots \quad \text{predpoklad}_n}{\text{záver}} \text{ meno}$$

Pravidlo s uvedeným *menom* nám dovoľuje tvrdiť, že platí uvedený *záver* ak sa nám podarí dokázať platnosť uvedených *predpokladov*. V sekcii 2 uvádzame pravidlá na dokazovanie platnosti formúl výrokovkej logiky. V sekcii 3 sú uvedené pravidlá na dokazovanie platnosti formúl predikátovej logiky. Pravidlá sme čerpali z prvej časti knihy, ktorá s venuje tzv. Z kalkulu [2].

Tabuľka 1. Formula $(p \wedge q \Rightarrow r) \Rightarrow (p \Rightarrow (q \Rightarrow r))$ je tautológiou, pretože je pravdivá pre každé pravdivostné ohodnotenie výrokových atómov.

| p | q | r | $(p \wedge q \Rightarrow r) \Rightarrow (p \Rightarrow (q \Rightarrow r))$ | | | | |
|-----|-----|-----|--|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

2 Výroková logika

Výroková logika, ako každý jazyk, má svoju syntax a svoju sémantiku. Syntax je daná definíciou 1. Sémantika jednotlivých logických spojok je v tomto texte definovaná množinou základných odvodzovacích pravidiel:

- pre konjunkciu: $\wedge I, \wedge E_1, \wedge E_2$
- pre disjunkciu: $\vee I_1, \vee I_2, \vee E^{[1]}$
- pre implikáciu: $\Rightarrow E, \Rightarrow I^{[1]}$
- pre ekvivalenciu: $\Leftrightarrow I, \Leftrightarrow E_1, \Leftrightarrow E_2$
- pre negáciu: $\neg I^{[1]}, \text{false } E^{[1]}, \neg E$

ktoré uvádzame v sekciiach 2.1-2.6. Dokázateľné formuly sú pravdivé. Nedokázateľné formuly sú nepravdivé.

Definícia 1 (Výroková formula [1]) Predpokladajme, že sme si zvolili nejakú množinu atómov At , ktorá neobsahuje žiadny zo siedmich symbolov

$(,), \wedge, \vee, \Rightarrow, \Leftrightarrow, \neg$. Prvkom množiny At hovoríme *výrokové atómy* alebo *atómy*. Nasledujúce pravidlá definujú, ktoré z výrazov zostavených z prvkov množiny $At \cup \{ (,), \wedge, \vee, \Rightarrow, \Leftrightarrow, \neg \}$ sú tzv. *výrokovými formulami*.

- každý atóm je výroková formula,
- ak p je výroková formula, tak $\neg p$ je výroková formula,
- ak p a q sú výrokové formuly, tak $(p \wedge q)$, $(p \vee q)$, $(p \Rightarrow q)$, $(p \Leftrightarrow q)$ sú výrokové formuly.

Obvyklá dohoda je, že:

- vonkajší pár zátvoriek je možné vynechať,
- logické spojky majú vzostupnú prioritu v nasledujúcom poradí:
 $\Leftrightarrow, \Rightarrow, \vee, \wedge, \neg$,
- vzhľadom na to, že konjunkcia je asociatívna, z formúl ako $p \wedge (q \wedge r)$ a $(p \wedge q) \wedge r$ vynechávame zátvorky,
- vzhľadom na to, že disjunkcia je asociatívna, z formúl ako $p \vee (q \vee r)$ a $(p \vee q) \vee r$ vynechávame zátvorky,
- implikácia má asociativitu sprava-doľava, t.j. formulu $p \Rightarrow q \Rightarrow r$ treba chápať ako $p \Rightarrow (q \Rightarrow r)$,
- ak za jednou negáciou nasleduje rovno ďalšia, tak ten pár zátvoriek medzi nimi môžeme vynechať, t.j. formulu $\neg\neg p$ je treba chápať ako $\neg(\neg p)$.

Ak sme na tieto pravidlá zvyknutí, umožňuje to zo zápisu formúl vynechať značné množstvo zátvoriek.

2.1 Konjunkcia

Pravidlá na zavedenie a eliminovanie konjunkcie:

$$\frac{p \quad q}{p \wedge q} \wedge I$$

$$\frac{p \wedge q}{p} \wedge E_1$$

$$\frac{p \wedge q}{q} \wedge E_2$$

Pravidlo $\wedge I$ hovorí, že ak sa nám podarí dokázať platnosť formuly p a q , tak môžeme tvrdiť, že platí formula $p \wedge q$.

Pravidlá $\wedge E_1$ a $\wedge E_2$ hovoria, že ak sa nám podarí dokázať platnosť formuly $p \wedge q$, tak môžeme tvrdiť, že platí formula p resp. q .

Príklad 1 Ukážme si, ako je možné tieto pravidlá použiť na dokázanie komutatívnosti disjunkcie. Ide o to dokázať, že z platnosti $p \wedge q$ vyplýva platnosť $q \wedge p$. T.j.:

$$\frac{p \wedge q}{q \wedge p} \wedge \text{comm}$$

Dôkaz je pomerne jednoduchý:

$$\frac{\frac{p \wedge q}{q} \wedge E_2 \quad \frac{p \wedge q}{p} \wedge E_1}{q \wedge p} \wedge I$$

Dôkaz vyzerá ako strom. V koreni je cieľová formula, ktorú chceme dokázať: $q \wedge p$. V listoch sú predpoklady, ktoré môžeme použiť na dôkaz cieľa. V našom prípade je len jeden jediný predpoklad: $p \wedge q$. Vnútorne uzly obsahujú jednotlivé medzikroky dôkazu:

- Z $p \wedge q$ sme pomocou pravidla $\wedge E_2$ odvodili, že platí q .
- Z $p \wedge q$ sme pomocou pravidla $\wedge E_1$ odvodili, že platí p .
- Z q a p sme pomocou pravidla $\wedge I$ odvodili, že platí $q \wedge p$.

2.2 Disjunkcia

Pravidlá pre zavedenie a eliminovanie disjunkcie:

$$\frac{p}{p \vee q} \vee I_1 \quad \frac{q}{p \vee q} \vee I_2 \quad \frac{p \vee q \quad \frac{[p]^{[i]}}{r} \quad \frac{[q]^{[i]}}{r}}{r} \vee E^{[i]}$$

Pravidlo $\vee I_1$ hovorí, že ak sa nám podarí dokázať p , tak môžeme tvrdiť, že platí $p \vee q$. Podobne, pravidlo $\vee I_2$ hovorí, že ak sa nám podarí dokázať q , tak môžeme tvrdiť, že platí $p \vee q$. Predstavme si, že máme nejakú množinu Γ , z ktorej sa nám podarilo dokázať platnosť $p \vee q$. Ak sa nám ešte podarí dokázať,

- že nejaká formula r vyplýva z Γ, p
- že tá istá formula r vyplýva aj z Γ, q

tak podľa pravidla $\vee E^{[i]}$ môžeme tvrdiť, že r vyplýva z Γ .

Γ, p v tomto prípade predstavuje množinu predpokladov Γ obohatenú o formulu p . Podobne, Γ, q predstavuje množinu Γ obohatenú o q .

Príklad 2 Uvedené odvodzovacie pravidlá je možné použiť na dôkaz komutatívnosti disjunkcie:

$$\frac{p \vee q}{q \vee p} \vee \text{comm}$$

T.j. že ak sa nám podarí dokázať $p \vee q$, môžeme tiež tvrdiť, že platí $q \vee p$. Dôkaz:

$$\frac{p \vee q \quad \frac{[p]^{[1]}}{q \vee p} \vee I_2 \quad \frac{[q]^{[1]}}{q \vee p} \vee I_1}{q \vee p} \vee E^{[1]}$$

Tie dodatočne zavedené predpoklady $[p]$ a $[q]$ sú vizuálne odlišené od originálnych originálneho predpokladu $p \vee q$.

To malé nenápadné číslo v hranatých zátvorkách nám umožňuje spárovať jednotlivé dodatočné predpoklady s miestom, kde sme ich zaviedli. Je to užitočné najmä vtedy, keď je takých miest viac ako jedno.

2.3 Implikácia

Pravidlá pre zavedenie a eliminovanie implikácie:

$$\frac{p \quad p \Rightarrow q}{q} \Rightarrow E \qquad \frac{[p]^{[i]} \quad q}{p \Rightarrow q} \Rightarrow I^{[i]}$$

Pravidlo $\Rightarrow E$ predstavuje klasický *modus ponens*. Ak máme dokázané, že platí $p \Rightarrow q$ a súčasne máme dokázané, že platí p , toto pravidlo hovorí, že súčasne môžeme tvrdiť, že platí q .

Predstavme si, že máme nejakú množinu predpokladov Γ . Ak sa nám podarí dokázať, že z Γ, p vyplýva q , tak podľa pravidla $\Rightarrow I^{[i]}$ môžeme tvrdiť, že z Γ vyplýva $p \Rightarrow q$.

Príklad 3 Tranzitívnosť implikácie:

$$\frac{p \Rightarrow q \quad q \Rightarrow r}{p \Rightarrow r} \Rightarrow \text{trans}$$

sa pomocou uvedených pravidiel dá dokázať nasledovným spôsobom:

$$\frac{\frac{[p]^{[1]} \quad p \Rightarrow q}{q} \Rightarrow E \quad q \Rightarrow r}{\frac{r}{p \Rightarrow r} \Rightarrow I^{[1]}} \Rightarrow E$$

2.4 Ekvivalencia

Pravidlá pre zavedenie a eliminovanie ekvivalencie sú:

$$\frac{p \Rightarrow q \quad q \Rightarrow p}{p \Leftrightarrow q} \Leftrightarrow I \qquad \frac{p \Leftrightarrow q}{p \Rightarrow q} \Leftrightarrow E_1 \qquad \frac{p \Leftrightarrow q}{q \Rightarrow p} \Leftrightarrow E_2$$

Príklad 4 Tranzitívnosť ekvivalencie:

$$\frac{p \Leftrightarrow q \quad q \Leftrightarrow r}{p \Leftrightarrow r} \Leftrightarrow \text{trans}$$

je možné dokázať pomocou uvedených odvodzovacích pravidiel a s využitím už dokázaného pravidla \Rightarrow trans :

$$\frac{\frac{\frac{p \Leftrightarrow q}{p \Rightarrow q} \Leftrightarrow E_1 \quad \frac{q \Leftrightarrow r}{q \Rightarrow r} \Leftrightarrow E_1}{p \Rightarrow r} \Rightarrow \text{trans} \quad \frac{\frac{q \Leftrightarrow r}{r \Rightarrow q} \Leftrightarrow E_2 \quad \frac{p \Leftrightarrow q}{q \Rightarrow p} \Leftrightarrow E_2}{r \Rightarrow p} \Rightarrow \text{trans}}{p \Leftrightarrow r} \Leftrightarrow I$$

2.5 Kongruencia

V prípade ekvivalencie vieme s výhodou využiť to, že sa jedná o tzv. *kongruenciu*. Znamená to, že ak platí $p \Leftrightarrow q$, tak v ľubovoľnej formule, ktorá obsahuje p (označme ju ako $C[p]$) môžeme p nahradiť formulou q . Novú formulu označme $C[q]$.

$$\frac{C[p] \quad p \Leftrightarrow q}{C[q]} \text{congr}$$

Toto pravidlo nám umožňuje vykonať v dôkaze taký krok, kde jednu časť nejakej formuly nahradíme inou (ekvivalentnou) formulou.

2.6 Negácia

Pravidlá pre zavedenie a eliminovanie negácie:

$$\frac{[p]^{[i]}}{false} \neg I^{[i]} \qquad \frac{[\neg p]^{[j]}}{p} false E^{[j]} \qquad \frac{p \quad \neg p}{false} \neg E$$

Ak máme nejakú množinu predpokladov Γ a podarí sa nám ukázať, že množina Γ, p je sporná (že z nej vieme odvodiť nepravdu), tak pravidlo $\neg I^{[i]}$ hovorí, že z Γ vyplýva $\neg p$.

Ak z $\Gamma, \neg p$ vieme odvodiť nepravdu, pravidlo $false E^{[j]}$ hovorí, že z Γ vyplýva p .

Ak sa nám z Γ podarí dokázať p a $\neg p$, tak pravidlo $\neg E$ hovorí, že Γ je sporná.

Príklad 5 Pravidlo na eliminovanie dvojitej negácie:

$$\frac{\neg \neg p}{p} \neg \neg E$$

je možné pomocou uvedených pravidiel dokázať takto:

$$\frac{\neg \neg p \quad [\neg p]^{[1]}}{false} \neg E \qquad \frac{false}{p} false E^{[1]}$$

Príklad 6 Pravidlo na zavedenie dvojitej negácie:

$$\frac{p}{\neg \neg p} \neg \neg I$$

je možné pomocou uvedených pravidiel dokázať takto:

$$\frac{p \quad [\neg p]^{[1]}}{false} \neg E \qquad \frac{false}{\neg \neg p} \neg I^{[1]}$$

2.7 Ďalšie príklady

Rôzne ďalšie odvodená pravidlá:

$$\frac{p}{p \wedge p} \wedge \text{idempot}$$

$$\frac{p}{p \vee p} \vee \text{idempot}$$

$$\frac{p_1 \wedge p_2 \Rightarrow q}{p_1 \Rightarrow p_2 \Rightarrow q} \text{prop}_1$$

$$\frac{p_1 \wedge p_2 \wedge p_3 \Rightarrow q}{p_1 \Rightarrow p_2 \Rightarrow p_3 \Rightarrow q} \text{prop}_2$$

$$\frac{\neg(p \vee q)}{\neg p \wedge \neg q} \text{de Morgan}_1$$

$$\frac{\neg(p \wedge q)}{\neg p \vee \neg q} \text{de Morgan}_2$$

$$\frac{(p \wedge q) \wedge r}{p \wedge (q \wedge r)} \wedge \text{assoc}$$

$$\frac{(p \vee q) \vee r}{p \vee (q \vee r)} \vee \text{assoc}$$

$$\frac{}{p \vee \neg p} \text{tertia non datur}$$

$$\frac{p \Leftrightarrow q}{q \Leftrightarrow p} \Leftrightarrow \text{comm}$$

Čitateľ má možnosť vyskúšať si ich dokázať vlastnými silami a porovnať svoj dôkaz s doleuvedenými uvedenými vzorovými riešeniami.

Príklad 7 Dá sa dokázať, že konjunkcia je idempotentná operácia.

$$\frac{p}{p \wedge p} \wedge \text{idempot}$$

Dvojitá čiara namiesto jednoduchej znamená, že pravidlo je možné použiť v oboch smeroch.

Dôkaz toho, že pravidlo funguje jedným smerom (zhora-dolu).

$$\frac{p \quad p}{p \wedge p} \wedge \text{I}$$

Dôkaz toho, že pravidlo funguje aj druhým smerom (zdola-hore).

$$\frac{p \wedge p}{p} \wedge \text{E}_1$$

Príklad 8 Dá sa dokázať, že disjunkcia je idempotentná operácia:

$$\frac{p}{p \vee p} \vee \text{idempot}$$

Dôkaz pravidla v jednom smere:

$$\frac{p}{p \vee p} \vee I_1$$

Dôkaz pravidla druhým smerom:

$$\frac{p \vee p \quad [p]^{[1]} \quad [p]^{[2]}}{p} \vee E^{[1]}$$

Príklad 9 Konjunkciu v antecedente implikácie $p_1 \wedge p_2 \Rightarrow q$ je možné nahradiť samostatnými implikáciami $p_1 \Rightarrow p_2 \Rightarrow q$:

Dôkaz:

$$\frac{p_1 \wedge p_2 \Rightarrow q}{p_1 \Rightarrow p_2 \Rightarrow q} \text{prop}_1$$

$$\frac{\frac{\frac{[p_1]^{[1]} \quad [p_2]^{[2]}}{p_1 \wedge p_2} \wedge I \quad p_1 \wedge p_2 \Rightarrow q}{q} \Rightarrow I^{[2]}}{p_2 \Rightarrow q} \Rightarrow I^{[1]}}{p_1 \Rightarrow p_2 \Rightarrow q} \Rightarrow E$$

Príklad 10 Konjunkciu v antecedente implikácie $(p_1 \wedge p_2 \wedge p_3) \Rightarrow q$ je možné nahradiť samostatnými implikáciami $p_1 \Rightarrow (p_2 \Rightarrow (p_3 \Rightarrow q))$:

$$\frac{p_1 \wedge p_2 \wedge p_3 \Rightarrow q}{p_1 \Rightarrow p_2 \Rightarrow p_3 \Rightarrow q} \text{prop}_2$$

Dôkaz:

$$\frac{\frac{\frac{\frac{[p_1]^{[1]} \quad [p_2]^{[2]}}{p_1 \wedge p_2} \wedge I \quad [p_3]^{[3]}}{p_1 \wedge p_2 \wedge p_3} \wedge I \quad p_1 \wedge p_2 \wedge p_3 \Rightarrow q}{q} \Rightarrow I^{[3]}}{p_3 \Rightarrow q} \Rightarrow I^{[2]}}{p_2 \Rightarrow p_3 \Rightarrow q} \Rightarrow I^{[1]}}{p_1 \Rightarrow p_2 \Rightarrow p_3 \Rightarrow q} \Rightarrow E$$

Príklad 11 Jedno z de Morganových pravidiel hovorí, že negácia disjunkcie je konjunkcia negácií:

$$\frac{\neg(p \vee q)}{\neg p \wedge \neg q} \text{ de Morgan}_1$$

Dôkaz tvrdenia jedným smerom:

$$\frac{\frac{\frac{[p]^{[1]}}{p \vee q} \vee I_1 \quad \neg(p \vee q)}{\text{false}} \neg E \quad \frac{\frac{[q]^{[2]}}{p \vee q} \vee I_2 \quad \neg(p \vee q)}{\text{false}} \neg E}{\frac{\text{false}}{\neg p} \neg I^{[1]} \quad \frac{\text{false}}{\neg q} \neg I^{[2]}}{\neg p \wedge \neg q} \wedge I$$

Dôkaz tvrdenia druhým smerom:

$$\frac{\frac{\neg p \wedge \neg q}{\neg p} \wedge E_1 \quad \frac{\frac{[\neg \neg(p \wedge q)]^{[1]}}{p \wedge q} \neg \neg E}{p} \neg E}{\frac{\text{false}}{\neg(p \wedge q)} \text{false E}^{[1]}}$$

Príklad 12 Druhé z de Morganových pravidiel hovorí, že negácia konjunkcie je disjunkcia negácií:

$$\frac{\neg(p \wedge q)}{\neg p \vee \neg q} \text{ de Morgan}_2$$

Dôkaz jedným smerom:

$$\frac{\frac{\frac{[\neg(\neg p \vee \neg q)]^{[1]}}{\neg \neg p \wedge \neg \neg q} \text{ de Morgan}_1 \quad \frac{\frac{[\neg(\neg p \vee \neg q)]^{[1]}}{\neg \neg p \wedge \neg \neg q} \text{ de Morgan}_1}{\frac{\neg \neg p}{p} \neg \neg E \quad \frac{\neg \neg q}{q} \neg \neg E}}{\frac{p \wedge q}{\neg(p \wedge q)} \neg E} \quad \frac{\text{false}}{\neg p \vee \neg q} \text{false E}^{[1]}$$

Dôkaz druhým smerom:

$$\frac{\neg p \vee \neg q \quad \frac{\frac{[\neg p]^{[2]} \quad \frac{[p \wedge q]^{[1]}}{p} \wedge E_1}{false} \neg E \quad \frac{[\neg q]^{[2]} \quad \frac{[p \wedge q]^{[1]}}{q} \wedge E_2}{false} \neg E}{false} \vee E^{[2]}}{false} \neg I^{[1]}}{\neg(p \wedge q)}$$

Príklad 13 Konjunkcia je asociatívna operácia:

$$\frac{(p \wedge q) \wedge r}{p \wedge (q \wedge r)} \wedge \text{assoc}$$

Dôkaz platnosti pravidla jedným smerom:

$$\frac{\frac{\frac{(p \wedge q) \wedge r}{p \wedge q} \wedge E_1 \quad \frac{(p \wedge q) \wedge r}{p} \wedge E_1}{p} \wedge E_1 \quad \frac{\frac{(p \wedge q) \wedge r}{p \wedge q} \wedge E_1 \quad \frac{(p \wedge q) \wedge r}{q} \wedge E_2}{q} \wedge E_2 \quad \frac{(p \wedge q) \wedge r}{r} \wedge E_2}{q \wedge r} \wedge I}{p \wedge (q \wedge r)} \wedge I$$

Dôkaz platnosti pravidla v opačnom smere:

$$\frac{\frac{\frac{p \wedge (q \wedge r)}{p} \wedge E_1 \quad \frac{p \wedge (q \wedge r)}{q \wedge r} \wedge E_2}{q \wedge r} \wedge E_1 \quad \frac{\frac{p \wedge (q \wedge r)}{q \wedge r} \wedge E_1 \quad \frac{p \wedge (q \wedge r)}{q} \wedge E_2}{q} \wedge E_2 \quad \frac{p \wedge (q \wedge r)}{r} \wedge E_2}{1p \wedge q} \wedge I}{(p \wedge q) \wedge r} \wedge I$$

Príklad 14 Disjunkcia je asociatívna operácia:

$$\frac{(p \vee q) \vee r}{p \vee (q \vee r)} \vee \text{assoc}$$

Príklad 15 Princíp vylúčenia tretieho (tertia non datur) hovorí, že ak zoberieme ľubovoľnú formulu p , tak buď platí p alebo $\neg p$:

$$\frac{}{p \vee \neg p} \text{ tertia non datur}$$

Dôkaz:

$$\frac{\frac{\frac{[\neg(p \vee \neg q)]^{[1]}}{\neg p \wedge \neg \neg p} \wedge E_1}{\neg p} \text{ de Morgan}_1}{\frac{\frac{[\neg(p \vee \neg q)]^{[1]}}{\neg p \wedge \neg \neg p} \wedge E_2}{\neg \neg p} \neg E}{\frac{false}{p \vee \neg p} \text{ false E}^{[1]}}{\neg E}}$$

Príklad 16 Ekvivalencia je komutatívna operácia:

$$\frac{p \Leftrightarrow q}{q \Leftrightarrow p} \Leftrightarrow \text{comm}$$

Dôkaz:

$$\frac{\frac{p \Leftrightarrow q}{q \Rightarrow p} \Leftrightarrow E_2}{\frac{p \Leftrightarrow q}{p \Rightarrow q} \Leftrightarrow E_1}{q \Leftrightarrow p} \Leftrightarrow I$$

3 Predikátová logika

Syntax predikátovej logiky je daná definíciou 2. Sémantika tohto jazyka je daná odvodzovacími pravidlami:

- tými, ktoré platia pre výrokovú logiku
- pre univerzálny kvantifikátor: $\forall I^{[i]}$, $\forall E$
- pre existenčný kvantifikátor: $\exists I$, $\exists E^{[j]}$

Definícia 2 (Predikátová formula [1]) V predikátových formulách (presnejšie formulách predikátovej logiky prvého rádu) sa môžu vyskytovať symboly viacerých druhov:

- *Logické spojky*: $\wedge, \vee, \Rightarrow, \Leftrightarrow, \neg$
- *Kvantifikátory*: univerzálny kvantifikátor \forall a kvantifikátor \exists .
- *Zátvorky*: $()$.
- *Premenné*: $\{x, y, u, v, \dots, x_0, x_1, \dots\}$
- *Funkčné symboly* pre označenie operácií s objektami. Ich návratovou hodnotou sú opäť objekty. Každému funkčnému symbolu F je priradené prirodzené číslo $0 \leq n$ nazývané *árnosť* symbolu F . Napríklad „+“ spravidla označuje binárny funkčný symbol, t.j. funkčný symbol s arnosťou 2. Funkčné symboly s arnosťou 0 sa nazývajú *konštanty*.
- *Predikátové symboly* (tiež *relačné symboly*) pre označenie vzťahov medzi objektami. Každému predikátovému symbolu P je priradené prirodzené číslo $1 \leq n$ zvané *árnosť* symbolu P . Napríklad „ \in “ spravidla označuje binárny predikátový symbol, t.j. predikátový symbol s arnosťou 2.

Množinu všetkých premenných označme Var a predpokladajme o nej, že je nekonečná spočítateľná. Logickým spojkám a kvantifikátorom sa hovorí *logické symboly*. Funkčné a predikátové symboly spolu určujú *jazyk* (alebo *teóriu*) L .

Množina všetkých *termov jazyka* L je najmenšia množina výrazov spĺňajúcich podmienky:

- každá premenná je term (jazyka L)
- ak sú t_1, \dots, t_n termy a $F \in L$ je funkčný symbol s arnosťou n , tak $F(t_1, \dots, t_n)$ je term jazyka L .

Atomická formula jazyka L je každý výraz tvaru $P(t_1, \dots, t_n)$, kde t_1, \dots, t_n sú termy jazyka L a $P \in L$ je predikátový symbol s arnosťou n . Množina všetkých (predikátových) formulí jazyka L je najmenšia množina výrazov spĺňajúcich podmienky:

- každá atomická formula je formula jazyka L ,
- ak p a q sú formuly jazyka L , x je premenná a a je nejaká množina objektov, potom aj výrazy $(p \wedge q)$, $(p \vee q)$, $(p \Rightarrow q)$, $(p \Leftrightarrow q)$, $\neg p$, $\forall x : a \bullet p$, $\exists x : a \bullet p$ sú formuly jazyka L .

3.1 Kvantifikátory a deklarácie premenných

Podobne ako aj v [2], oba druhy formulí s kvantifikátorom majú tvar:

$$Qx : a \bullet p$$

kde

- Q je kvantifikátor (\forall alebo \exists),
- x je viazaná premenná,
- a je typ premennej x a
- p je predikát.

Oba kvantifikátory teda zavádzajú *viazanú premennú*, ktorá je analogická ku lokálnej premennej v bloku nejakého programu. V kvantifikovanej formule $Qx : a \bullet p$ je premenná x viazaná v p . Kvantifikátory viažu premenné veľmi voľne, takže rozsah platnosti premennej siaha až po najbližšiu ukončovaciu zátvorku.

Príklad 17 V nasledujúcej formule je rozsah platnosti premennej x označený zloženou zátvorkou:

$$(\underbrace{\forall x : a \bullet p \Rightarrow q \wedge r}_{\text{rozsah platnosti } x}) \vee s \Rightarrow t$$

Ak formula obsahuje viac ako jeden kvantifikátor, rozsahy platností sa môžu prekryvať. Ak má každá viazaná premenná jedinečné meno, nie je to žiadny problém. V opačnom prípade tam je v niektorom rozsahu platnosti diera.

Príklad 18 V nasledujúcej formule

$$\forall y : a \bullet p \Rightarrow q \wedge (\forall y : b \bullet r \Rightarrow s \Rightarrow t) \wedge u \vee v$$

má rozsah platnosti prvej premennej y diery, ktorá zodpovedá rozsahu platnosti druhej premennej y .

$$\forall y : a \bullet \underbrace{p \Rightarrow q}_{\text{rozsah platnosti prvého } y} \wedge (\underbrace{\forall y : b \bullet r \Rightarrow s \Rightarrow t}_{\text{rozsah platnosti druhého } y}) \wedge u \vee v$$

Konkrétny výskyt nejakej premennej v nejakej formule môže byť buď *voľný* alebo *viazaný*.

Príklad 19 V tejto formule:

$$\forall x : \mathbf{N} \bullet x > y$$

sa premenná y sa vyskytuje voľne. Premenná x je tam viazaná:

$$\underbrace{\forall x : \mathbf{N}}_{\text{miesto väzby}} \bullet \overbrace{x > y}^{\text{rozsah platnosti } x}$$

3.2 Substitúcia

Pod substitúciou rozumieme nahradenie všetkých voľných výskytov danej premennej daným termom. Pri tom je potrebné dodržať určité pravidlá.

Pomocou ekvivalencie budeme vysvetľovať efekt substitúcie vo formulách s kvantifikátorom.

V najjednoduchšom prípade má substituovaná premenná rovnaké meno ak kvantifikovaná premenná:

$$(\forall x : a \bullet p)[t/x] \Leftrightarrow (\forall x : a \bullet p)$$

$$(\exists x : a \bullet p)[t/x] \Leftrightarrow (\exists x : a \bullet p)$$

V tom prípade substitúcia nemá žiadny efekt. Substitúcia sa totiž týka len voľných výskytov danej premennej a v horeuvedených formulách sú všetky výskyty premennej x viazané.

Príklad 20 Uvažujme nasledujúcu substitúciu²

$$(\exists x : Osoba \bullet \neg(x \text{ Pozná } y))[t/x]$$

Keď vykonáme naznačenú substitúciu, dostaneme:

$$\exists x : Osoba \bullet \neg(x \text{ Pozná } y)$$

Je to logické, lebo v pôvodnej formule sa nenachádzajú žiadne voľné výskyty premennej x , ktoré by sme mali šancu nahradiť zadaným termom t .

Ak substituovaná premenná má iné meno ako kvantifikovaná premenná, potom sa substitúcia vykonáva nasledujúcim spôsobom:

$$(\forall y : a \bullet p)[t/x] \Leftrightarrow (\forall y : a \bullet p[t/x])$$

$$(\exists y : a \bullet p)[t/x] \Leftrightarrow (\exists y : a \bullet p[t/x])$$

Príklad 21 Uvažujme nasledujúcu substitúciu:

$$(\exists x : Osoba \bullet \neg(x \text{ Pozná } y))[t/y]$$

Výsledkom je formula:

$$(\exists x : Osoba \bullet \neg(x \text{ Pozná } t))$$

Pravda je ale ešte trochu zložitejšia. Horeuvedené pravidlo je použiteľné len v prípade, že keď samotný term t neobsahuje žiadnu takú voľnú premennú, ktorá by sa (po vykonaní substitúcie) mohla stať viazanou premennou. V takomto prípade je potrebné najprv vykonať alfa-konverziu v pôvodnej formule (zmeniť meno problematickej viazanej premennej) a až potom vykonať substitúciu. Príslušná voľná premenná potom ostane, aj po substitúcii, voľnou premennou:

² Nech *Pozná* je binárny predikátový symbol, ktorý vyjadruje fakt, že prvá osoba pozná druhú osobu.

$$(\forall y : a \bullet p)[t/x] \Leftrightarrow (\forall y : a \bullet p[z/y])[t/x]$$

$$(\exists y : a \bullet p)[t/x] \Leftrightarrow (\exists y : a \bullet p[z/y])[t/x]$$

Hore z predstavuje takú premennú, ktorá sa v pôvodnej formule nevyskytuje a ani nie je voľnou premennou v t .

Príklad 22 Ak napríklad chceme vykonať nasledovnú substitúciu:

$$(\exists y : Osoba \bullet \neg(y \text{ Pozná } x))[y/x]$$

je najprv potrebné vykonať alfa-konverziu:

$$(\exists z : Osoba \bullet \neg(z \text{ Pozná } x))[y/x]$$

a až potom je možné vykonať samotnú substitúciu:

$$\exists z : Osoba \bullet \neg(z \text{ Pozná } y)$$

Ak hlavným operátorom vo formule nie je kvantifikátor, potom je možné efekt substitúcie vysvetliť jednoducho:

$$\begin{aligned} (\neg p)[t/x] &\Leftrightarrow \neg p[t/x] \\ (p \wedge q)[t/x] &\Leftrightarrow p[t/x] \wedge q[t/x] \\ (p \vee q)[t/x] &\Leftrightarrow p[t/x] \vee q[t/x] \\ (p \Rightarrow q)[t/x] &\Leftrightarrow p[t/x] \Rightarrow q[t/x] \\ (p \Leftrightarrow q)[t/x] &\Leftrightarrow p[t/x] \Leftrightarrow q[t/x] \end{aligned}$$

3.3 Univerzálny kvantifikátor

V prípade výrokovej logiky sme mali možnosť dokázať platnosť formúl dvoma spôsobmi: klasickou tabuľkovou metódou a použitím dedukčných pravidiel. V prípade predikátovej logiky je tabuľková metóda vo všeobecnosti nepoužiteľná, pretože množina hodnôt, ktorú môžu jednotlivé viazané premenné nadobúdať, je jednoducho príliš veľká. Môžeme ale nadviazať na dedukčné pravidlá pre výrokovú logiku tak, že pridáme pravidlá na zavedenie a eliminovanie kvantifikátorov.

Ak by sme sa pozerali na univerzálny kvantifikátor ako na zovšeobecnenú konjunkciu, tak by sme mohli dostať pravidlá pre univerzálny kvantifikátor zovšeobecnením pravidiel pre konjunkciu. Uvažujme najprv pravidlo na zavedenie konjunkcie. Aby sme dokázali, že platí $p \wedge q$, je potrebné dokázať aj p a aj q . Analogicky, aby sme dokázali platnosť $\forall x : a \bullet p$, je potrebné dokázať, že p platí pre každú hodnotu x z množiny a . Toto neznie príliš nádejne, pretože to môže viesť k nekonečnému počtu premís a preto nekonečnému počtu dôkazov.

Lepším prístupom by mohlo byť, keď sa nám podarí ukázať platnosť p výhradne z predpokladu, že $x \in a$ bez toho, aby sme nejako obmedzovali to, ktoré konkrétne x z a berieme. Takýto dôkaz p potom platí pre všetky $x \in a$.

$$\frac{[x \in a]^{[i]} \quad p}{\forall x : a \bullet p} \forall I^{[i]} \quad \begin{array}{l} x \text{ sa nevyskytuje ako voľná premenná} \\ \text{v ostatných predpokladoch formuly} \\ \forall x : a \bullet p \end{array}$$

Použitie tohto pravidla je korektné len v tom prípade, že žiadny z ostatných predpokladov dokazovanej formuly $\forall x : a \bullet p$ neobsahuje voľnú premennú x . Aby mohlo platiť $\forall x : a \bullet p$, musíme dokázať, že platí p platí pre ľubovoľné $x \in a$.

Ak platí $p \wedge q$ tak je možné tvrdiť, že platí aj p aj q . Ak platí $\forall x : a \bullet p$, tak je možné tvrdiť, že platí $p[t/x]$ pre ľubovoľné $t \in a$.

$$\frac{t \in a \quad \forall x : a \bullet p}{p[t/x]} \forall E$$

V špeciálnom prípade, keď na mieste voľnej premennej t používame práve premennú x , pravidlo vyzerá takto:

$$\frac{x \in a \quad \forall x : a \bullet p}{p} \forall E$$

3.4 Existenčný kvantifikátor

Existenčný kvantifikátor sa dá chápať ako zovšeobecnená disjunkcia. Pravidlá pre zavedenie a eliminovanie existenčného kvantifikátora sú analogické pravidlám pre zavedenie a eliminovanie disjunkcie.

Ak sme chceli dokázať $p \vee q$, stačilo nám dokázať buď p alebo q (pravidlá $\vee I_1$ a $\vee I_2$). Pri existenčnom kvantifikátore je to podobné. Ak chceme dokázať $\exists x : a \bullet p$, stačí nám dokázať $p[t/x]$ pre niektoré $t \in a$.

$$\frac{t \in a \quad p[t/x]}{\exists x : a \bullet p} \exists I$$

Na elimináciu disjunkcie slúži pravidlo $\vee E^{[i]}$. Eliminovanie existenčného kvantifikátora je podobné. Ak sa nám z nejakej množiny predpokladov Γ podarí dokázať, že platí $\exists x : a \bullet p$, tak ak sa podarí z predpokladov Γ , $x \in a \wedge p$ dokázať nejaký predikát r , tak môžeme tvrdiť, že r vyplýva z predpokladov Γ .

Toto pravidlo je možné korektne použiť len v prípade, ak sa x nevyskytuje ako

$$\frac{\exists x : a \bullet p \quad \frac{[x \in a \wedge p]^{[i]} \quad r}{r} \exists E^{[i]}}{r} \exists E^{[i]} \quad \begin{array}{l} x \text{ sa nevyskytuje ako voľná premenná} \\ \text{v ostatných predpokladoch formuly} \\ \exists x : a \bullet p \end{array}$$

Príklad 24 Platnosť pravidla:

$$\frac{\neg (\exists x : a \bullet p)}{\forall x : a \bullet \neg p} \text{ de Morgan}_3$$

sa dá dokázať aj jedným:

$$\frac{\frac{\frac{[x \in a]^{[1]} \quad [p]^{[2]}}{\exists x : a \bullet p} \exists I \quad \neg (\exists x : a \bullet p)}{\neg I[2]} \neg E}{\frac{false}{\neg p} \neg I[2]} \neg E}{\forall x : a \bullet \neg p} \forall I[1]$$

aj druhým smerom:

Príklad 25 Pravidlo:

$$\frac{\neg (\forall x : a \bullet p)}{\exists x : a \bullet \neg p} \text{ de Morgan}_4$$

sa dá dokázať jedným:

$$\frac{\frac{\frac{[\exists x : a \bullet \neg p]^{[1]}}{\forall x : a \bullet \neg \neg p} \text{ de Morgan}_3 \quad [x \in a]^{[2]}}{\forall E} \quad \frac{\neg \neg p}{\neg \neg E} \quad \frac{p}{\forall I[2]}}{\forall x : a \bullet p} \quad \neg (\forall x : a \bullet p)}{\frac{false}{\exists x : a \bullet \neg p} \text{ false E}^{[1]}} \neg E$$

aj druhým smerom:

$$\frac{\frac{\frac{\exists x : a \bullet \neg p}{\neg \neg (\exists x : a \bullet \neg p)} \neg \neg I \quad \frac{\frac{[\neg \neg p]^{[1]}}{p} \neg \neg E \quad \frac{[p]^{[2]}}{\neg \neg p} \neg \neg I}{\neg \neg p \Rightarrow p} \Rightarrow I^{[1]} \quad \frac{\neg \neg p}{p \Rightarrow \neg \neg p} \Rightarrow I^{[2]}}{\neg \neg p \Leftrightarrow p} \Leftrightarrow I}{\neg (\forall x : a \bullet \neg \neg p)} \text{ de Morgan}_3}{\neg (\forall x : a \bullet p)} \text{ congr}$$

4 Záver

V súvislosti s formálnymi logickými systémami sú zaujímavé dva druhy programov:

- tzv. *proof checkers*
- a tzv. *theorem provers*

Programy v prvej skupine (proof checkers) sú schopné overiť, či človekom predložený dôkaz danej vety je správny. Programy v druhej skupine (theorem provers) sú schopné nielen skontrolovať správnosť predloženého dôkazu, ale príslušný dôkaz aj (v niektorých prípadoch) nájsť (ak existuje). Druhá skupina programov už v podstate zasahuje do umelej inteligencie. Prvá skupina programov ponecháva kreativitu človeku. Najznámejšie z týchto programov sú vymenované v publikácii [3]. Jeden z uvedených systémov (Mizar) podporuje prirodzenú dedukciu.

Zástupcom neklasickej logiky je systém Coq [4]. Taktiež uľahčuje tvorbu a overenie dôkazov matematických viet. Je ale zameraný na generovanie certifikovaných programov. Postup je tu trochu obrátený naruby. Namiesto dokazovania správnosti nejakého hotového programu sa postupuje inak. Pre danú špecifikáciu sa konštruktívne dokáže, že existuje program, ktorý danú špecifikáciu spĺňa. Tento dôkaz musí vykonať človek v interakcii s Coq-om. Extrahovanie skompilovateľného programu z uvedeného konštruktívneho dôkazu je potom už možné zautomatizovať a toto vie na požiadanie urobiť systém Coq.

Literatúra

- [1] Švejdar V.: *Logika, neúplnosť, složitost a nutnost*. Academia, nakladatelství České republiky, 2002.
- [2] Woodcock J., Davies J.: *Using Z: specification, refinement, and proof*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1996.
- [3] Wiedijk F.: *The seventeen provers of the world*.
- [4] Bertot Y., Casteran P.: *Interactive Theorem Proving and Program Development*. Texts in Theoretical Computer Science. An EATCS Series. Springer-Verlag, 2004.

Aplikovaná fuzzy logika

Štefan KOZÁK¹

Abstrakt. Metódy a algoritmy výpočtovej inteligencie, založené na fuzzy množinách a fuzzy logike nachádzajú čoraz väčšie uplatnenie v mnohých oblastiach ľudskej činnosti. Sú využívané najmä v úlohách pre podporu a optimalizáciu rozhodovania, riadenia a diagnostiky procesov, tak v priemysle, ako aj v oblasti ekonomických a socio-technických procesov. Množstvo i oblastí aplikácií fuzzy logiky neustále pribúda. Stalo sa už dnes štandardom, že bežný užívateľ spotrebnej elektroniky začína dôverovať a dáva prednosť tým výrobkom, ktoré využívajú pre svoju funkcionálnosť fuzzy logiku. Jedným zo významných faktorov, ktoré umožňujú úspešne aplikovať metódy fuzzy logiky najmä vo výrobkoch spotrebnej elektroniky, sú inteligentné algoritmy riadenia a vnorené vysoko výkonné mikropočítače, s vysokou integritou prvkov a funkcií, čím sa zvyšuje funkcionálnosť a komfort výrobkov. Algoritmy fuzzy logiky využívajú skúsenosti a expertné hodnotenia javov a situácií, ktoré sú sekundárne využívané tak pri tvorbe matematických modelov zložitých nelineárnych procesov, ako aj pri návrhu samotných algoritmov riadenia. Ukazuje sa, že najmä zložité nelineárne procesy v priemysle, pri ktorých nie je možné či jednoduché získať matematický model opisujúci chovanie sa takéhoto procesu štandardnými identifikačnými metódami, sú typickou oblasťou, kde je možno demonštrovať rad výhod zavedenia fuzzy algoritmov riadenia. Táto kapitola je zameraná na opis a charakteristiku takých procesov, ktoré efektívne využívajú a aplikujú algoritmy fuzzy logiky. Výsledky aplikácie fuzzy logiky, simulačných riešení a overenia fuzzy algoritmov modelovania a riadenia sú demonštrované na vybraných príkladoch v numerickej a grafickej forme pre posúdenie vhodnosti a kvality riešenia v porovnaní s konvenčnými postupmi.

1 Charakteristika metód výpočtovej inteligencie využívajúcich fuzzy logiku

Život každého z nás je dnes výrazne ovplyvňovaný zavádzaním nových progresívnych a inteligentných technológií, ktoré významne ovplyvňujú náš život prostredníctvom

¹ Fakulta elektrotechniky a informatiky, STU Bratislava, Ilkovičova 3, 812 19 Bratislava, E-mail: stefan.kozak@stuba.sk

výrobkov spotrebnej elektroniky, nových priemyselných technológií, výrobných systémov, sociálnych a obchodných činností, moderných diagnostických a prieskumných systémov. Metódy výpočtovej inteligencie využívajúce fuzzy množiny, umelé neurónové siete, genetické algoritmy a genetické programovanie sú významným a silným faktorom podieľajúcim sa na kvalite výrob, technológií, bezpečnosti systémov a minimalizácie energií.

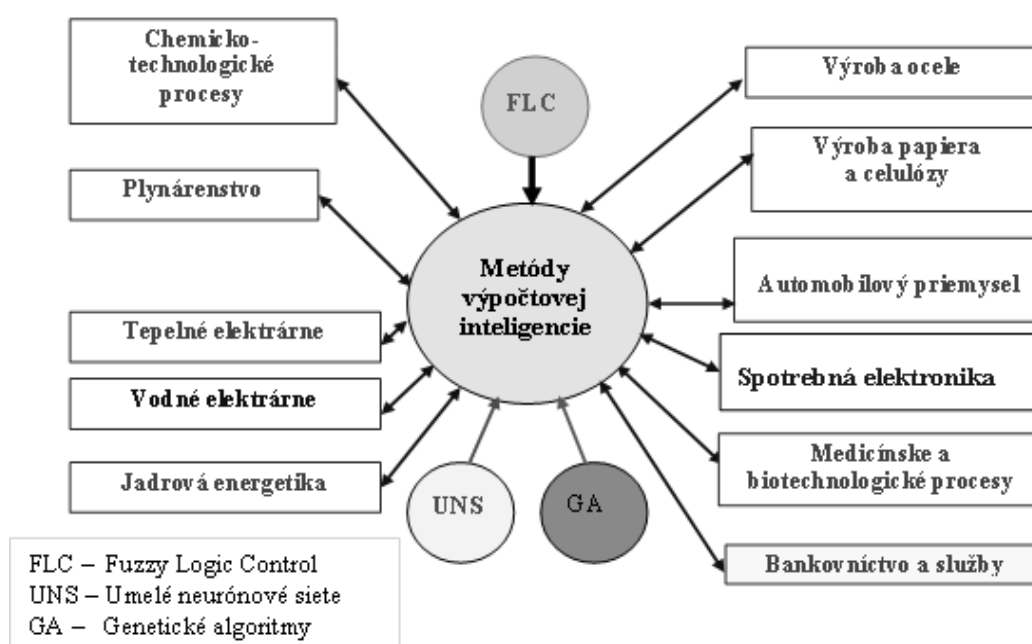
Fuzzy logika [5,16,8] predstavuje dnes jednu z najväčších oblastí využívania metód výpočtovej inteligencie v rozhodovacích a riadiacich činnostiach vo všetkých oblastiach ľudskej činnosti. V súvislosti s rozvojom moderných prostriedkov výpočtovej a riadiacej techniky zabudovanej do výrobkov spotrebnej elektroniky, hračiek, v zdravotníctve, bankovníctve a do priemyselných výrob v mnohých odvetviach je nevyhnutné naďalej rozvíjať a implementovať aj inteligentné komunikačné a inteligentné riadiace algoritmy [12] zabezpečujúce vysokú kvalitu, komfort, spoľahlivosť a bezpečnosť (obr. 1).

Fuzzy množiny tvoria v súčasnosti neoddeliteľnú súčasť moderných vedeckých disciplín a výrazne prispeli k riešeniu mnohých technických a netechnických problémov, ktoré sa v praxi nedali konvenčnými postupmi modelovania a riadenia efektívne zvládnuť. Fuzzy teória, metódy a algoritmy na báze fuzzy logiky prenikli do množstva aplikácií v oblasti riadenia, diagnostiky, spracovania obrazov a prognostike chovania sa procesov v kritických situáciách. Teória fuzzy množín tvorí významnú časť metód umelej inteligencie a patrí medzi najlepšie matematicky spracovanú časť umelej inteligencie. Moderné priemyselné procesy v chémii, biotechnológiách, energetike sú dnes ovládané a riadené prostredníctvom inteligentných a optimálnych algoritmov s minimalizáciou energií, optimálnou komunikáciou a s adaptívnymi resp. samo nastavujúcimi sa parametrami riadenia.

2 Oblasti aplikácie fuzzy logiky

Fuzzy logika predstavuje paradigmu pre efektívny a alternatívny návrh algoritmov modelovania a riadenia, ktoré môžu byť použité v pokročilých technológiách pre buď vnorené autonómne systémy alebo ako zložité distribuované sieťové aplikácie. Použitím fuzzy logiky môžu dizajnéri dosiahnuť nižších nákladov na vývoj, lepšie vlastnosti a výkonnosť produktu. Fuzzy množiny sa stali jedným z dominantných faktorov priamo využívaných metód výpočtovej inteligencie v praxi. Dnes už mnoho projektantov a dodávateľov riadiacich systémov ponúka vo svojich štandardných SW balíkoch fuzzy logiku ako komplexný systém pre ladenie parametrov a priamu implementáciu týchto metód. Taký systém umožní minimalizovať náklady na prevádzku, zabezpečuje vysokú kvalitu a stabilitu riadenia, vynikajúcu funkcionálnu a komfort výrobných a zariadení, ktoré tieto algoritmy majú zabudované. Systémy riadenia a algoritmy na báze fuzzy logiky a zariadenia disponujúce fuzzy logikou môžu byť ľahko modifikovateľné a zovšeobecniteľné, rýchlo uvedené na trh s minimalizáciou vývojových a implementačných prác. Čo teda fuzzy logika pre dnešného človeka prináša:

- Fuzzy logika znižuje dobu projektovania a vývoj algoritmov riadenia.
- Fuzzy logika zjednodušuje návrh riadiacích a rozhodovacích algoritmov.
- Fuzzy logika skracuje dobu uvedenia nových smart výrobkov na trh.
- Fuzzy logika nahrádza ťažkopádne a náročne algoritmy riadenia nelineárnych procesov.
- Fuzzy logika zvyšuje výkon zariadení, prevádzkyschopnosť, spoľahlivosť a aktívnu bezpečnosť prvkov a zariadení.
- Fuzzy logika zjednodušuje implementáciu.
- Fuzzy logika znižuje celkové náklady na riadiaci SW a HW.



Obr. 1. Prienik metód výpočtovej inteligencie do aplikačných oblastí

V súčasnosti oblasť aplikácie fuzzy algoritmov sa neustále rozširuje a preniká aj do takých systémov, ktoré sú náročné a vyžadujú dokonalé a presné matematické modely a precízne rozhodovacie a riadiace algoritmy. Fuzzy algoritmy modelovania a riadenia sú založené na troch základných pilieroch ako je fuzifikácia, inferencia a defuzifikácia. V ďalšej časti uvedieme vybrané oblasti aplikácie fuzzy množín a fuzzy logiky. Fuzzy aplikácie riadiacich systémov: firmy Yamakava prvé fuzzy VLSI obvody Omron, Motorola, Allen Bradley, Siemens (fuzzy regulátory), fuzzy procesory, fuzzy obvody.

2.1 Fuzzy logika v spotrebnej elektronike

Využitie fuzzy logiky v spotrebnej elektronike má stále stúpajúci trend. Jednou z najväčších aplikácií fuzzy logiky sú spotrebiče v domácnosti. Komplexné fuzzy

algoritmy riadenia práčky a pružná adaptácia režimov pracieho cyklu je realizovaná mikropočítačovými riadiacimi algoritmi na báze fuzzy logiky. Algoritmy sú realizované tak, aby používateľ mal možnosť interagovať s počítačom a práčkou a nastavovať vstupné ako aj priebežné hodnoty vstupných parametrov, napr. druh tkaniny, druh znečistenia a pod. a zároveň v priebehu činnosti pracieho cyklu kontrolovať stav a zisťovať či proces prebieha v poriadku na základe analýzy stavu snímaných veličín (teploty, hladina, znečistenie vody a pod.).

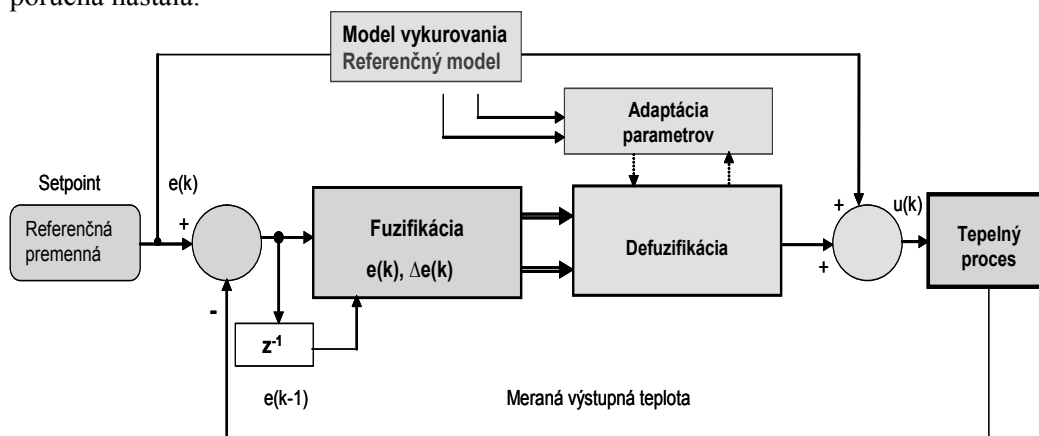
| | | |
|--|---|--|
| | | |
| Vybrané oblasti aplikácie fuzzy logiky | | |
| Automatická prevodovka, ABS systémy, optimalizácia spaľovania – Honda, Nissan, Subaru fuzzy logika automaticky zabezpečuje radenie prevodového pomeru podľa zaťaženia motora, štýlu jazdy, stavu a tvaru cesty | Inteligentné kuchynské systémy varne Mikrovlnné rúry – Sharp, Hitachi, Sanyo, Toshiba fuzzy logika riadenie a optimalizácia rovnomernosti ohrevu a prípravy teplých jedál | Tepelné procesy , hriankovače, káfovary, bojler, fuzzy logika optimalizuje procesy prípravy jedla a nápojov, opekania podľa typu chleba, ohrevov vody a tekutín, tlaku pary podľa typu kávy a receptúr |
| Umývačka riadu – Matsushita, Sanyo, fuzzy logika riadi umývací cyklus a stratégiu umývania v závislosti na množstve riadu, type potraviny a jeho množstve | Chladiace a klimatizačné systémy , Chladičky Elektrolux, Gorenje, Samsung fuzzy logika aplikovaná na optimalizáciu chladenia a odmrazovania s optimalizáciou príkonu | Automatická práčka Daewoo, Samsung, Electrolux, Gorenje, Sanyo, Sharp, fuzzy logika nastavuje optimalizuje pracie režimy, minimalizuje spotrebu energie |
| Riadenie výtahov – Fujitec, Toshiba, Mitsubishi fuzzy logika riadi plynulosť a bezpečnosť prevádzky, optimalizuje čakaciu dobu a zastavovanie výtahov | Televízory Samsung, Sony, Panasonic samočinné prispôsobovanie obrazu fuzzy logikou podľa osvetlenia miestnosti, viacnásobné prekryvanie obrazov | Makroekonomické a mikroekonomické modely. Fuzzy logika – fuzzy modely predikcie, viackriteriálneho rozhodovania. |
| Ždravotníctvo Omron, Simatic, Schneider Electric fuzzy logika optimalizuje dávkovanie liekov a infúzií, optimalizácia tepu arytmií, adaptácia načúvacích systémov | Vysávače Elektrolux, Eta, Hitachi, Toshiba fuzzy logika zabezpečuje adaptáciu sily sania podľa typu podlahy a stavu jej zašpinenia. | Priemyselné procesy, biotechnológie Simatic, Allen Bradley, Fisher Rossemount, Foxboro, Yokogawa fuzzy logika v riadení procesov, distribuované systémy riadenia, sieťové technológie, PLC systémy. |
| Fotoaparáty Minolta, Canon, Panasonic fuzzy logika riadi optimalizáciu tienenia a zaostrovania | Videosystémy , Videokamera, Canon, Panasonic, Sanyo stabilizácia obrazu pri snímaní, pohybu a oscilácií pomocou fuzzy logiky | Energetika kotly, čističky, elektrárne Schneider Electric, Siemens Teleperm fuzzy logika v riadení a predikcii kritických stavov |

Obr. 2. Vybrané oblasti aplikácie fuzzy logiky

Na základe množstva prádla v práčke je možno v automatickom režime optimalizovať potrebné množstvo vody a podľa stavu vody a obsahu nečistôt vo vode je algoritmovaná a optimalizovaná doba prania a doba plákania. Uvedené automatizované činnosti u moderných pračiek sú realizované na základe fuzzy algoritmov, ktoré zabezpečujú minimálnu spotrebu vody, minimalizáciu elektrickej energie a úsporu pracích prostriedkov.

Ďalšou aplikáciou fuzzy algoritmov sú *umývačky riadu*. Na základe voľby fuzzy programu používateľ má možnosť využívať niekoľko variantných režimov mytia pre rôzne prevádzkové režimy napr. nízke a vysoké teploty umývania. Umývačka riadu samočinne rozpoznáva tak množstvo ako aj stav zašpinenia nádob a na základe fuzzy algoritmov rozhodovania nastavuje tak množstvo vody, ako aj jej optimálnu teplotu, čas umývania a plákania. Prieběžná analýza stavov v umývačke na základe snímania prevádzkových veličín umožňuje zase adaptovať umývacie režimy tak aby bola zabezpečená minimálna spotreba vody a elektrickej energie pri vysokej kvalite umývania.

Fuzzy algoritmy prania a umývania predstavujú otvorený systém a je možné ich v prípade vývoja nových lepších algoritmov modifikovať a preprogramovať. Okrem riadiacích činností umožňuje fuzzy logika realizovať aj aktívnu diagnostiku týkajúcu sa napr. napájania vodou, elektrinou, otvorenia dvierok, stavu čerpadla a pod. a tak zistiť a vizualizovať typ poruchy pre ktorú zariadenie nemôže riadne fungovať. Diagnostika celého zariadenia môže byť detailne realizovaná na základe špeciálneho diagnostického režimu a algoritmov a umožňuje pre opravu zariadení presnejšie špecifikovať typ poruchy a miesto poruchy resp. sekvenciu činností po ktorých porucha nastala.



Obr. 3. Bloková schéma aplikácie fuzzy algoritmov do procesov vykurovania

Riadenie kotla s plynule riaditeľným fuzzy regulátorom ponúka dnes mnoho výrobcov. Namiesto bežného ekvitermálneho riadenia teploty kotlovej vody podľa vonkajšej teploty a s korekciami na osvit, vietor a s priradenou krivkou ohrevu na základe tepelných strát a akumulačných vlastností budovy) je pre úlohy fuzzy riadenia využívaná a snímaná len výstupná teplota vody z kotla, ktorá reprezentuje reálny stav výkonu vyhrievacieho telesa (riadenie záťaže). Fuzzy regulátor pracuje ako adaptívny

regulátor (obr. 3), ktorý generuje riadiaci zásah na základe porovnania a vyhodnotenia ďalších veličín využívaných pre úlohy adaptívneho riadenia. Sú to predovšetkým nasledovné informácie:

- Informácie o minulých hodnotách tepla s jednodňovým oneskorením v rovnakom čase alebo priemerná spotreba tepla, poukazujúca na tepelný stav vykurovaného objektu.
- Informácia o trendoch spotreby tepla, ktorá reaguje na rozdiel medzi žiadanou teplotou vo vnútri a aktuálnou hodnotou vonkajšej teploty.
- Informácie o krátkodobých trendoch spotreby tepla, ktorá je funkciou osvetlenia, tepelnej produkcie ľudí vo vnútri objektu alebo závisí na otvorení okien.
- Informácie o dennom/týždennom/mesačnom profile tepelných strát na základe štatistických meraní meteorologických údajov.

2.2 Fuzzy logika v automobilovom priemysle

Fuzzy logika patrí medzi tie metódy výpočtovej inteligencie, ktoré v priebehu posledných tridsiatich rokov od svojho vzniku rýchlo našli uplatnenie takmer vo všetkých oblastiach ľudskej činnosti. Metódy fuzzy logiky výrazne prispievajú k zvýšeniu kvality výrobkov, bezpečnosti, prevádzkyschopnosti, kvality života. Medzi najväčšie oblasti využitia fuzzy logiky patria výrobky spotrebnej elektroniky. V tejto časti sú demonštrované možnosti aplikácie fuzzy logiky v automobilovom priemysle.

Jednou z oblastí, kam fuzzy logika v posledných rokoch prenikla, je automobilový priemysel. V priemyselne vyspelých štátoch ako je Japonsko, USA, Nemecko, Veľká Británia a Francúzsko sa auta s fuzzy logikou úspešne predávajú, spoločným motívom je nová inteligentná technológia pre automobilový priemysel. Dôvody záujmu sú jednoduché, používatelia v súčasnosti vyžadujú vysoký komfort vozidla, nenáročnosť obsluhy a vysokú bezpečnosť vozidla, čo im fuzzy logika a fuzzy systémy riadenia a ovládania dokážu oveľa lepšie splniť v porovnaní s konvenčnými riešeniami

2.3 Anti-Lock brzdových systémov s Fuzzy Logic

Prvé aplikácie protišmykových brzdových systémov sa datujú od roku 1947. Boeing Corporation vyvinula prvý protišmykový brzdový systém pre lietadlá ako elektronicko-mechanický systém. V súčasnosti je systémom ABS štandardne vybavená väčšina vozidiel. Používajú sa pritom elektronické senzory, ktoré merajú rýchlosť každého kolesa a pomocou vnorených mikrokontrolérov riadia tlak tekutín pre brzdové valce. Fuzzy systém ABS je založený na integrácii skúseností a vedomostí z rokov testovania na rôznych cestách v odlišnom podnebí.

ABS systém sleduje počas brzdenia pohyb kolies a v prípade, že zistí, že niektoré z kolies sa na vozovke začína šmykať, zmenší alebo úplne vypne brzdú silu. Tým sa snaží medzi vozovkou a pneumatikou udržať valivé trenie a zabrániť kľzavému treniu a teda zaisťuje lepšiu ovládateľnosť automobilu. Nevýhodou konvenčných ABS systémov je vo väčšine prípadov dlhšia brzdová dráha (čo vyplýva zo samotného princípu ABS). Problém s použitím konvenčných systémov ABS nastáva aj na tzv.

roletách na ceste, kedy kolesá nadskakujú na nerovnostiach vozovky a tým odovzdajú senzorom nesprávnu informáciu o nepomere rýchlosti otáčania, ktorú systém ABS vyhodnotí ako nebezpečenstvo šmyku. Systém ABS sa stal bežnou súčasťou dnešných moderných vozidiel, jeho kvalita sa výrazne zlepšila aplikáciou fuzzy logiky. Dynamika systému a celkový brzdny systém vozidla je silno nelineárny dynamický systém, čo môže spôsobiť pri aplikácii konvenčného riešenia už uvedené problémy so stabilitou a neriaditeľnosťou vozidla. Využitie fuzzy logiky, ako inteligentného, znalostného systému je vhodné vtedy, ak je systém nelineárny, zložitý a často matematicky neopísateľný. Účelom vylepšeného systému ABS je zabrániť zablokovaniu kolies a tým vzniku šmyku. Umožňuje riadiť auto aj počas intenzívneho brzdzenia bez ohľadu na kvalitu povrchu vozovky. Hardvérová implementácia fuzzy logiky zabezpečuje, že dĺžka jedného cyklu riadenia fuzzy logikou klesla na 0.5 ms. Takéto inteligentné riešenie viedlo k výborným výsledkom správania sa brzdových systémov testovaných vozidiel.

2.4 Fuzzy ABS systémy

Fuzzy logika je veľmi efektívna metodológia pre zavádzanie fuzzy algoritmov priamo do technického riešenia nových vozidiel. Vzhľadom na túto skutočnosť väčšina aplikácií ABS systémov, ktoré sú úspešne implementované do moderných vozidiel, využíva fuzzy logiku. V súčasnosti sú to predovšetkým automobily značky Nissan a Mitsubishi, ktoré majú nasadené fuzzy ABS systémy, ale aj niektoré vozidlá firiem Honda, Mazda, Hyundai, BMW, Mercedes-Benz, Bosch a Peugeot obsahujú moderné algoritmy riadenia ABS systémov na báze fuzzy logiky.

Ďalším dôvodom, prečo fuzzy ABS systémy našli uplatnenie v automobilovom priemysle je vysoká kvalita a účinnosť brzdneho systému, čas regulácie pre všetky slučky je do 5 milisekúnd. Za tento čas musia príslušné mikroprocesory snímať a spracovať údaje, riadiť samotné ventily a realizovať riadiaci zásah. Pre systémy ABS sa používajú 16 bitové mikroprocesory, ktoré dokážu za pol milisekundy spracovať a vyhodnotiť snímanú udalosť s minimálnymi nárokmi na pamäť – 2kB ROM pamäte.

Systém fuzzy logiky využíva analýzu stavov a hodnôt vstupných premenných, ktoré reprezentujú zmenu rýchlosti vozidla (spomalenie, zrýchlenie), zmenu rýchlosti otáčanie kolies, hydraulický tlak brzdovej kvapaliny a pod. Hodnoty týchto premenných predstavujú a reprezentujú nepriamy ukazovateľ aktuálnej situácie brzdzenia a jeho správanie sa v čase. Aplikácie už prvých vyvíjaných algoritmov ukázali, že prvý prototyp algoritmu iba so šiestimi fuzzy pravidlami výrazne zlepši kvalitu brzdzenia aj v prípade zasneženej vozovky.

2.5 Ďalšie aplikácie fuzzy logiky v ABS

Vzhľadom na vysokú konkurenciu výrobcov v automobilovej technike, väčšina výrobcov sa zdráha zverejniť detailnejšie podrobnosti o inteligentných technológiách a algoritmoch, ktoré používajú vo vozidlách. Niektoré spoločnosti a výrobcovia vozidiel sa tiež obávajú negatívneho významu slova "fuzzy" vo svojich výrobkoch a systémoch pretože fuzzy v ich ponímaní predstavuje zvlášť v Európe "nepresnosť",

a preto sa obávajú, že používatelia vozidiel by si mohli myslieť, že fuzzy ABS systémy majú nižšiu kvalitu. V Japonsku je tento pojem silne etablovaný vo všetkých oblastiach ľudskej činnosti a preto ich obyvatelia sú hrdí na to, že predstavujú svetovú špičku v predmetnej oblasti. V Nemecku je však ešte množstvo firiem a výrobcov automobilovej techniky, ktoré aplikáciu fuzzy logiky príliš nekomentujú a nezdôrazňujú napriek tomu, že ju aplikujú do algoritmov realizovaných v mikro-počítačoch v C-kóde alebo v assembleri. Niektorí používatelia, ktorí sú znalí v oblasti fuzzy logiky, majú obavy zo stability fuzzy algoritmov, avšak obavy z nezaručenia stability sú neopodstatnené vzhľadom na to, že fuzzy algoritmy prispievajú len k doladovaniu už stabilného globálneho systému.

2.6 Riadenie spaľovania pomocou Fuzzy logiky

Ovládanie a riadenie spaľovania motorov osobných a nákladných automobilov sa stáva čoraz zložitejším v dôsledku vyššej emisnej normy a náročných požiadaviek na vyššiu palivovú efektívnosť. Pred dvadsiatimi rokmi boli systémy riadenia aplikované iba na niektoré prvky a mechanické systémy (karburátor, rozdeľovač a ističe kontaktov). V súčasnosti sú vnorené mikro-počítačové systémy aplikované na vstrekovanie paliva a optimalizáciu zapalovacieho bodu. Vzhľadom na to, že stratégie riadenia pre spaľovanie motora veľmi silno závisia na aktuálnej hodnote pracovného bodu (nelineárneho systému ktorý je funkciou viacerých premenných napr. otáčok, momentu a pod.), lineárne modely riadenia a použitie konvenčných metód riadenia typu PID nie sú vhodné. Jednou z možností efektívneho riešenia je dopredu pripraviť tabuľku riadenia (look-up table control) umožňujúcu riadiť proces spaľovania pre jednotlivé pracovné body. *Look-up*-tabuľka je generovaná na základe výsledkov rozsiahlych testov a skúseností technika.

Aplikácia *look-up tabuľky* je však vhodná len pre tri dimenzie (2 vstupy, 1 výstup). Tiež vytváranie a interpretácia týchto look-up-tabuliek je komplikovaná a je vedou aj umením ju optimálne navrhnuť. Nahradenie týchto podporných prehľadových tabuliek sa dá realizovať oveľa lepšie fuzzy logikou (pravidlami). Žiaľbohu, väčšina výrobcov nie je ochotná zverejňovať žiadne podrobnosti o ich aplikácii riadenia procesu spaľovania fuzzy logikou. To je spôsobené tým, že pravidlá pre fuzzy logiku systému obsahujú kompletne riadenie motora realizované na základe vývojových prác v spoločnosti a riešitelia nie sú ochotní túto algoritmizáciu ďalej šíriť. Výrobcovia automobilov sa obávajú, že sa ich konkurenti môžu naučiť nimi vyvinuté riešenie využiť vo svojich aplikáciách. Všetky tieto fuzzy algoritmy riadenia spaľovania sú založené na analýze stavov a obsahujú niekoľko modulov fuzzy logiky (štartovanie, normálny prevádzkový režim, cestný profil, premenlivá záťaž a pod.). Riadiaci systém spaľovania automaticky identifikuje stav motora a pomocou pravidiel a lingvistických premenných popisujúcich situáciu generuje optimálny prísun paliva. Podobne ako aj u ABS systémov aj riadenie motora vyžaduje vypočítať a realizovať riadiaci zásah veľmi rýchlo. Niektoré systémy riadenia spaľovania sú schopné výpočet riadiaceho zásahu realizovať do jednej milisekundy. Mnohí výrobcovia automobilov kombinujú fuzzy logiku a *look-up* tabuľku a často transformujú súbor pravidiel do tejto tabuľky.

Aj keď výpočet sa realizuje veľmi rýchlo, nároky na pamäť v tomto prípade enormne stúpajú, napr. pre 2 vstupy a jeden výstup je potrebné mať pamäť 64 kb ROM.

Jednou z ďalších aplikácií fuzzy logiky v automobilovej technike je riadenie automatickej prevodovky a automatické prepínanie a radenie prevodových stupňov. Firma Nissan aplikovala fuzzy logiku do automatických prevodoviek už v roku 1991 a postupne aj ďalšie svetové automobilky využívajú fuzzy logiku pre zvýšenie komfortu vodiča a zabezpečenie plynulosti jazdy. Fuzzy logika vyhodnocuje nielen aktuálnu rýchlosť vozidla, ale zohľadňuje aj situácie ako vodič zrýchľuje a brzdí a rýchlosť zmeny stláčania plynového pedálu a adaptuje svoju činnosť na základe reakcií vodiča.

2.7 Fuzzy modely v stavebníctve

Zabezpečenie požadovaných pevnostných charakteristík betónových konštrukcií patrí medzi dôležité a sledované vlastnosti stavieb, mostov priehrad a pod. Uvedené pevnostné charakteristiky sú závislé od kvality a zloženia betónu a jeho prísad. V súčasnosti sa pre zlepšenie a modifikáciu pevnostných charakteristík využíva veľké množstvo plastifikačných prísad ovplyvňujúcich cementovú suspenziu. Ich pôsobenie a pridanie do betónových zmesí je závislé na type a požadovaných vlastnostiach betónových konštrukcií, ktoré sú primárne závislé od kvality cementu.

Výpočet odhadov pevnostných charakteristík pomocou fuzzy modelov môže slúžiť predovšetkým ako vhodný modelovací nástroj v procese návrhu nových betónových zmesí a v procese výberu optimálnych zmesí pre konkrétny problém.

V procese návrhu nových stavebných hmôt na báze cementu a taktiež v procese výberu optimálnych stavebných hmôt pre danú aplikáciu je nutné realizovať množstvo laboratórnych skúšok pre dosiahnutie požadovaných vlastností. Fuzzy logika je jednou z tých možností a nástrojov, ktoré umožňujú znížiť finančné náklady spojené s veľkým počtom laboratórnych analýz. Práca [7] sa zaoberá využívaním fuzzy logiky pre riešenie uvedeného problému a poukazuje na novú metodológiu, ktorá na základe fuzzy modelov umožňuje efektívne problém riešiť.

Pre zostavenie a odladenie fuzzy modelov pre odhad pevnostných charakteristík betónu je nutné najskôr realizovať laboratórne merania pre získanie tzv. *trénovacích dát*.

Na základe expertných požiadaviek na výsledné fuzzy systémy a podľa charakteru aplikácie je zrejmé, že vstupné premenné by mali plne charakterizovať fyzikálne a chemické vlastnosti cementu. Teda voľba počtu a charakteru vstupných premenných by mala byť realizovaná tak, aby pomocou nich bolo možné zohľadniť vplyv na pevnostné charakteristiky betónu. Voľba typu fuzzy modelu musí zohľadňovať charakter nameraných údajov a zabezpečiť výpovednosť modelu. Pre uvedenú aplikáciu sa najlepšie hodí Takagi-Sugeno typ fuzzy modelu. Tento typ modelu sa javí ako najvhodnejší predovšetkým z dôvodu možnosti kvantitatívneho charakteru opisu, čo umožňuje zohľadniť tak expertné vedomosti ako aj výsledky laboratórnych meraní.

Pri využití programového systému Matlab-Simulink [2] je vhodné použiť dva typy fuzzy modelov.

- Prvý z nich využíva pre určenie optimálneho počtu a charakteru fuzzy pravidiel *fuzzy zhlukovacie metódy* (Substraktive clustering). Tieto metódy umožňujú výrazné zníženie počtu pravidiel. Tým môžeme dosiahnuť väčšiu prehľadnosť a rýchlejšie spracovanie výstupov z modelu. Každému zhluku je potom ANFIS metódou priradené pravidlo typu *if - then*.
- Druhým typom modelu typu Sugeno v systéme MATLAB je využitie tzv. mriežkového delenia (Grid partition) priestoru tvoreného kartézskym súčinom univerzí prislúchajúcich vstupným premenným. Pri tomto prístupe je nutné najprv zadať počet fuzzy množín pokrývajúcich jednotlivé univerzá a tvar funkcií príslušnosti pre každú premennú modelu. Počet generovaných pravidiel závisí len na počte vstupných a výstupných premenných a na počte použitých fuzzy množín.

Testovanie a voľba optimálneho nastavenia parametrov modelu sa realizuje po návrhu optimálneho počtu a typu funkcií príslušnosti. Za týmto účelom bola funkčnosť modelu overená pomocou dvoch skupín údajov a to trénovanej a testovanej množiny. Odhadované pevnostné charakteristiky betónu sú pevnosť v ťahu a za ohybu a boli testované po 7 a 28 dňoch. Boli vytvorené štyri fuzzy modely pre každú z piatich skúmaných plastifikačných prísad, pričom prostredníctvom modelov sa odhaduje jedna z uvedených pevnostných charakteristík. Vstupné premenné boli volené tak aby charakterizovali najdôležitejšie vlastnosti cementu a plastifikačné prísady, teda merný povrch, pevnosť cementu v tlaku, objemovú stálosť a percentuálny podiel plastifikačných prísad vzhľadom k hmotnosti cementu.

2.8 Riadenie systému dopravy v tuneloch fuzzy logikou

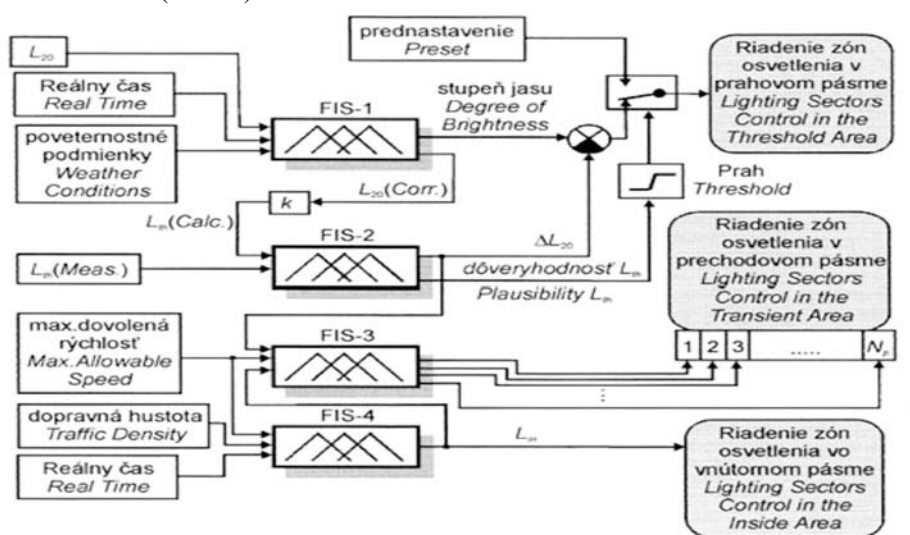
Pri riešení problému dopravy v rámci mesta môžu nastať rôzne situácie, o ktorých nevieme kedy presne nastanú. Premávku snímajú dopravné snímače ako sú par. indukčné slučky, optické snímače, video detektory a pod.



Obr. 4. Aplikácia fuzzy logiky do riadenia procesov v tuneli Branisko

Vysoko náročným problémom je riadenie premávky v tuneli [13]. Kapacita cestných tunelov býva obmedzená stavebnými podmienkami i sklonovými pomermi. Ako príklad možno uviesť údaje charakterizujúce priepustnosť 4975 m dlhého slovenského tunela Branisko v jednom smere pri 40% podiele nákladných vozidiel a štvorsekundových priemerných odstupoch. Plynulý dopravný tok bol charakterizovaný kapacitou 8978 osobných a 2992 nákladných vozidiel denne, odhady pre dopravnú špičku hovoria o 898 osobných a 300 nákladných vozidlách za hodinu. Včasné rozpoznanie blížiacej sa kolóny vozidiel a jej rýchlosti je nevyhnutné pre adaptívne riadenie technológií: *vetranie, osvetlenie, dopravné značenie a signalizáciu v tuneli*.

Naopak, schopnosť systému detegovať ojedinelé vozidlo zvyšuje hospodárnosť prevádzky tunela. Cieľom fuzzy riadenia je obmedziť potrebu korekčných zásahov dopravného dispečera do procesu riadenia a v mimoriadnych situáciách mu poskytnúť variantné riešenia (Obr. 5).



Obr. 5. Aplikácia fuzzy logiky do riadenia procesov v tuneli Branisko

Výhody riešenia riadenia procesov dopravy na báze fuzzy logiky:

- vyhodnotenie informácií od rôznych typov zariadení pracujúcich s rôznou presnosťou;
- dvojstupňovou kontrolou sa zvyšuje vierohodnosť signálov od snímačov;
- pomocou snímačov susedných detekčných stanovišť umožňuje vypočítať substitučné;
- hodnoty chýbajúcich informácií;
- dostatočne presná kategorizácia dopravného prúdu a predikcia jeho parametrov;
- optimalizácia riadenia v kritických situáciách na báze expertných znalostí.

2.9 Fuzzy logika v modelovaní logistiky a dopravných uzlov

Jednou z ďalších efektívnych aplikácií využitia fuzzy logiky je modelovanie, riadenie a simulácia dopravných a logistických uzlov. V práci [17] je demonštrovaný príklad určenia nástupišťa pomocou fuzzy logiky v kritických prípadoch, ak vlak je zmeškaný a je potrebné tomuto vlaku v danom čase priradiť voľnú koľaj v železničnej stanici. Pre riešenie tejto úlohy bol na základe reálnych prevádzkových údajov aplikovaný fuzzy regulátor typu Mamdani. Pri určení vhodnej koľaje pre prichádzajúci oneskorený vlak je potrebné vo fuzzy modeli zohľadniť aktuálne obsadenie koľají, plánované obsadenie koľají v ďalšom časovom období a rozmiestnenie prípojných vlakov. Ako ďalšie doplňujúce kritérium možno vybrať vzdialenosť náhradnej koľaje od koľaje na ktorú mal pôvodne vlak prísť.

Pre výber náhradnej koľaje je možné aplikovať niektorú z nasledujúcich stratégií:

- preferovať a snažiť sa o maximálne dodržanie pôvodného plánu obsadenia koľají,
- minimalizácia doby prestupu,
- minimalizácia oneskorenia ďalších vlakov.

Na základe reálnych údajov získaných z klasických dopravných poriadkov a denníkov bola navrhnutá štruktúra hodnôt (ostrých) vstupných premenných a ich rozsahy. Zároveň bol navrhnutý spôsob ohodnotenia koľají z hľadiska vhodnosti ich priradenia oneskorenému vlaku a tak získaná hodnota výstupnej premennej.

V práci [17] je ďalej deklarovaný spôsob fuzifikácie t.j. získanie pravidiel (súboru pravidiel) typu *if-then* z ostrých hodnôt vstupov a výstupov. Dekompozíciou súboru pravidiel sa získali dve bázy pravidiel. Prvá báza pravidiel (117) má sedem vstupov a jeden výstup. Druhá báza vznikla z prvého súboru pravidiel, obsahuje 64 pravidiel pre päť vstupov a jeden výstup. Overenie aplikácie bolo realizované na základe reálnych prevádzkových údajov. Navrhnuté fuzzy modely a regulátory dokázali vybrať skupinu koľají vhodných pre priradovanie oneskorených vlakov s návrhom viacerých variantných riešení.

2.10 Fuzzy logika v geotechnike

Ďalšou oblasťou využitia fuzzy množín je oblasť geotechniky. Je veľmi náročné a dokonca aj nemožné zostaviť matematický model horniny, ktorej vlastnosti sa menia napr. v trase razeného banského diela-tunela (pevnosť, smer diskontinuit, porušenosť, typ horniny) [4]. Mení sa tiež režim razenia (prítlak, otáčky, krútiaci moment), uhol ostria rozpojovacích nástrojov vplyvom opotrebovania, pracovná schopnosť rozpojovacej hlavy raziaceho stroja v dôsledku postupnej výmeny opotrebovaných diskov.

Riešením uvedených problémov sa dlhodobo zoberá Ústav geotechniky SAV v Košiciach. Výsledkom niekoľkoročného snaženia bolo vytvorenie optimalizačného systému (*w*-optimalizátor raziaceho stroja), ktorý umožňuje z monitorovaných údajov vypočítať optimálny režim razenia. Počas razenia systém zaznamenával v cca 2-sekundových intervaloch prítlak hlavy na čelbu F [kN], otáčky rozpojovacej hlavy n [s⁻¹], krútiaci moment hlavy raziaceho stroja Mk [kN.m] a hĺbku zatlačenia diskov do

horniny – h [mm]. Z týchto veličín sa v reálnom čase vypočítala merná objemová práca rozpojovania w [MJ.m-3], rozpojovací výkon Q [m3.hod-1] a doporučený optimálny prítlak F_{dop} [kN].

Dlhoročné skúsenosti zo štúdia procesu rozpojovania hornín, poznatky z analýzy údajov získaných z monitorovania procesu razenia, ako aj zložitost' systému hornina – nástroj, priviedli riešiteľov k myšlienke použiť fuzzy expertný systém, ako ďalšiu metódu pre spracovanie databázy údajov získanej z razenia. Najzávažnejšou časťou expertného systému je báza znalostí, ktorá rozhoduje o celkovej kvalite expertného systému a predovšetkým o kvalite odpovede. Stupeň naplnenia bázy znalostí je závislý na dostupných znalostiach a skúsenostiach. Vytvorená báza obsahuje 4 závislé a jednu nezávislú premennú veličinu. Nezávislými premennými sú prítlak hlavy raziaceho stroja – F [kN], hĺbka zatlačenia disku do horniny – h [mm], krútiaci moment hlavy raziaceho stroja a tzv. RQD koeficient. Závislou premennou, ktorá je zároveň aj výstupnou premennou, je pevnosť horniny v prostom tlaku. Prítlak, krútiaci moment a hĺbka zatlačenia predstavujú režimové parametre. Parameter prítlaku bol rozdelený do piatich lingvistických kategórií. Na základe dosiahnutých prevádzkových hodnôt vyplývajúcich z konštrukcie raziaceho stroja krútiaci moment nadobúda 4 lingvistické hodnoty. Pri voľbe lingvistických názvov sledovaných premenných sa vychádzalo z veľmi presných údajov, získaných z monitorovania procesu strojného razenia diaľničného tunela Branisko. Pre každú lingvistickú hodnotu bola ďalej definovaná príslušná fuzzy množina štvoricou čísiel a, b, c, d . Intervaly $a-b, c-d$ sú intervaly miery fuzzy príslušnosti prvku k množine. Interval $b-c$ vyjadruje interval istej príslušnosti prvku k množine. Ďalším vstupným parametrom, ktorý charakterizuje vlastnosť horniny je RQD koeficient. Podľa dosiahnutej porušenia horninového prostredia môže byť RQD veľmi zlá, zlá, stredná, dobrá alebo excelentná. Po definovaní premenných na základe poznatkov získaných z monitorovania procesov strojného razenia sa vytvorili pravidlá typu *IF-THEN*, pričom každé pravidlo bolo ohodnotené váhou. Vytvorená báza znalostí obsahuje 792 pravidiel. Výstupom riešenia fuzzy systému je pevnosť horniny v tlaku.

2.11 Aplikácia fuzzy logiky v hodnotení genotypovej variability

V súčasnosti so zvyšujúcim sa záujmom o zachovávanie existujúcich rastlinných genetických zdrojov narastá aj potreba skvalitnenia informácií o už existujúcich kolekciiach rastlín [14]. Jednou z ciest je postupný opis kolekciiach za použitia rozličných typov klasifikátorov. Rovnako dôležitá je tiež otázka určenia hraníc pre deväť bodovú stupnicu. K vytvoreniu novej metódy aplikáciou teórie fuzzy množín viedli riešiteľov dva dôvody. Väčšina klasifikátorov je zostavená tak, že veľkosť tried deskriptora pre jednotlivé body stupnice je rovnaká. Je to však v rozpore s intuitívnou predstavou. Pre mnohé znaky existujú medzinárodne resp. celoslovensky používané klasifikátory. Existujú však v súčasnosti znaky, pre ktoré je nutné vytvoriť akceptovateľné klasifikátory. Na biologický materiál boli aplikované aj ďalšie metódy založené na fuzzy metódach. Zložité biologické vzťahy často nie je možné dostatočne presne opísať klasickými regresnými vzťahmi. Je to však možné pomocou transformácie údajov na systém fuzzy rozhodnutí. Systém fuzzy rozhodnutí je univerzálny

prostriedok regresie pre opísanie závislostí biologických javov. Fuzzy c zhluková analýza umožňuje nájsť taký rozklad, ktorý lepšie reprezentuje štruktúru údajov o biologickom materiáli, ako klasická zhluková analýza. Umožňuje nájsť typických predstaviteľov takéhoto rozkladu. Preto je vhodná pre typologizáciu biologického materiálu. *Fuzzy c zhluková analýza* umožňuje navyše hodnotiť biologický materiál aj v časovom kontexte.

Realizačným výstupom výskumného riešenia boli navrhnuté nové postupy a aplikácie fuzzy metód v biológii (pre selekciu deskriptorov, určenie hraníc deskriptorov, výber perspektívnych jedincov v procese šľachtenia, popis zložitých biologických vzťahov pomocou systému fuzzy rozhodnutí, typologizácia pomocou *fuzzy c zhlukovej analýzy*, aplikácia fuzzy databázového systému, umožňujúceho zdokonalený výber z databáz biologických materiálov).

2.12 Fuzzy logika v anestéziológii a resuscitácii

Na Klinike anestéziológie a resuscitácie Lekárskej fakulty UP a Fakultnej nemocnice v Olomouci je už niekoľko rokov v prevádzke unikátny systém Relax 2005 pre dávkovanie infúzií riadenia svalových relaxancií – liekov pre svalové uvoľnenie – počas celkovej anestézie [1]. Jadrom systému Relax 2005 je počítačový program, ktorý vyhodnocuje stav pacienta a riadi dávkovanie pomocou infúznej pumpy. Pacient je pomocou snímačov napojený na monitor, ktorý vyhodnocuje stupeň svalovej relaxácie pacienta. Údaje sú odovzdávané počítačovému programu, ktorý na základe algoritmu na báze fuzzy logiky rozhoduje, čo má infúzna pumpa ďalej robiť na základe aktuálneho stavu relaxancie. Na základe vygenerovaného výstupu z algoritmu sa vyšle signál na dodanie určitého množstva infúzie do tela pacienta. Dávkovaná infúzia spôsobí zmenu svalovej relaxácie pacienta. To je zase indikované monitorom, ktorý umožňuje posielanie aktuálnej informácie o novom stave svalovej relaxácie. Táto činnosť je aktivovaná dovtedy, kým nie je dosiahnutý požadovaný stav svalovej relaxácie. Základom celého systému riadenia je automatický spätnoväzbový systém založený na fuzzy logike. Fuzzy logika využíva vytvorený matematický model systému, pomocou súboru pravidiel a zavedených lingvistických premenných. Postup anestéziológa sa dá opísať súborom pravidiel. Fuzzy logika v tomto prípade nahrádza a aproximuje činnosť lekára, pomocou súboru pravidiel, napr. *“ak je stupeň relaxácie dostatočne vysoký a bol dostatočne vysoký aj v čase predtým, potom pumpa nemá nič dávkovať”*. Systém bol aplikovaný pre viac ako 200 pacientov s priemernou dĺžkou automatického riadenia takmer štyri hodiny pre každého pacienta. Presnosť dávkovania s použitím fuzzy logiky je najmenej o jeden rád lepšia, ako je možné dosiahnuť pomocou manuálneho riadenia infúznej pumpy. Ďalšou prednosťou navrhovanej metodiky je možnosť zastavenia systému a ukončenia regulácie dávkovania rýchlo a bezpečne do 15 minút.

3 Matematické modely procesov na báze fuzzy logiky

Pri aplikácii fuzzy algoritmov riadenia tvorí neoddeliteľnú súčasť matematický model na základe fuzzy množín a fuzzy logiky. Identifikácia štruktúr a parametrov

matematických modelov reálnych procesov sa v praxi stretáva s dvoma veľmi nepríjemnými problémami. Na jednej strane je to prílišná zložitosť reality, ktorá spôsobuje, že buď nie sme schopní matematický model vôbec určiť a zostaviť, alebo odvodený analytický matematický model je tak zložitý, že sa nedá efektívne použiť pre procesy rozhodovania, diagnostiky a riadenia. Na druhej strane je to neurčitosť, ktorá je spôsobená našou neschopnosťou presne diferencovať realitu a presnejšie definovať inštrumentálne pojmy. Táto neurčitosť nepredstavuje však prekážku pri použití prirodzeného jazyka. Jeho najdôležitejšou vlastnosťou je totiž vágnosť jeho sémantiky a schopnosť s vágnymi pojmami pracovať. Teória fuzzy množín sa ponúka a dá využiť na opis procesov (reálnych systémov) dvojakého druhu:

- Jedny sa dajú označiť ako „fuzzy“, t.j. ich presný opis teoreticky neexistuje.
- Druhé svojou podstatou „fuzzy“ nie sú, avšak môžu byť natoľko zložené, že nie sme schopní presne ich opísať. Fuzzy premenné vytvárajú most medzi verbálnym a matematickým modelom reálneho systému. Fuzzy systém je behaviorálny kauzálny systém, ktorý charakterizuje a opisuje skúmaný proces iba prostredníctvom jeho vonkajších prejavov, t.j. niektorých merateľných, nemerateľných a pozorovateľných parametrov a javov, ktorých hodnoty sa kauzálny menia. Tieto parametre modelu sa spravidla delia na vstupy, výstupy, popr. stavy, resp. poruchové veličiny.

Existuje niekoľko dôvodov, prečo sa navrhujú a využívajú fuzzy modely pre praktické problémy už uvedených aplikácií:

- Modelovaný proces je príliš zložitý a jeho opis sa nedá získať konvenčnými identifikačnými a modelovacími technikami (cementárnska pec, tunelové pece a pod.)
- Rad dominantných faktorov charakterizujúcich proces nie je možné v modeli zohľadniť a kvantifikovať a je často charakterizovaný iba verbálne.
- Presnejšiu špecifikáciu chovania sa procesu a vplyvu určitých vonkajších parametrov nevieme do modelu zohľadniť
- Získanie a presné vyjadrenie matematického modelu je časovo príliš náročné a vyžadovalo by zložité experimenty.
- Použitie slov prirodzeného jazyka pre opis chovania sa procesu a požadovanej činnosti je výstižné a jednoduché, pretože je človeku najbližšie.
- Čas investovaný do návrhu fuzzy modelu je oveľa menší ako pri konvenčných technikách.
- Pre vytvorenie presného modelu je nutné experimentovať, čo v mnohých prípadoch nie je možné buď z technických dôvodov, ekonomických dôvodov alebo to vyplýva z povahy samotného procesu. Fuzzy model je v každom prípade objektívnejší ako verbálny opis postavený na čisto intuitívnom základe.

3.1 Všeobecný metodologický postup pri tvorbe fuzzy modelu

- Identifikácia a definícia maxima faktorov (veličín), ktoré budú v modeli použité a ich roztriedenie na závislé a nezávislé premenné, popr. konštanty.
- Preskúmanie všetkých premenných z hľadiska nezávislosti, rozhodnutie, či dané závislé faktory ponechať a určenie, ktoré závislé premenné sú determinované ktorými skupinami nezávisle premenných, t.j., koľko fuzzy modelov je treba vypracovať.
- Určenie tých faktorov, ktoré budú spracované ako jazykové premenné a určenie škál jazykových hodnôt.
- Určenie univerz jazykových premenných, t.j. z akých prvkov sa budú skladať (čísla, intervaly a i.) a aký bude ich rozsah.
- Transformácia jazykových hodnôt na konkrétne fuzzy množiny. To sa dá spraviť dobre len na základe dobrej znalosti študovaného procesu najlepšie metódou expertných odhadov jedného alebo niekoľkých expertov.
- Sformulovanie vlastného fuzzy modelu najskôr vo forme podmienených výrokov a potom jeho matematické spracovanie.

Fuzzy modely patria medzi tie matematické modely, ktoré môžu byť využité tak pre statické opisy procesov ako aj pre opis a aproximáciu dynamiky nelineárnych dynamických systémov. Fuzzy matematické modely spolu s matematickými modelmi pomocou umelých neurónových sietí môžeme začleniť medzi modely ktoré môžeme opísať metódami výpočtovej inteligencie.

Fuzzy modely tvoria základ pre aplikáciu moderných fuzzy algoritmov riadenia. Samotná štruktúra návrhu fuzzy modelov a algoritmov fuzzy riadenia sa od prvotných koncepcií a od prvého návrhu vypracovaného *Mamdanim* pre modelovanie a riadenie tlaku pary a otáčok parného stroja veľmi nezmenila.

Všeobecná forma opisu nelineárneho dynamického modelu

- spojité forma (stavový opis nelineárnymi diferenciálnymi rovnicami)

$$\frac{dx}{dt} = f[t, x(t), u(t), v(t)], \quad x(t_0) = x_0$$

$$y(t) = g[t, x(t), u(t)]$$

- diskretná forma (stavový opis) $x(k+1) = f[k, x(k), u(k), v(k)]$

$$y(k) = g[k, x(k), u(k)]$$

- diskretná forma (I/O model, nelineárna diferenčná rovnica)

$$y(k) = f(y(k-1), \dots, y(k-n_y), u(k-1-d), \dots, u(k-n_{u-k}-d) + v(k))$$

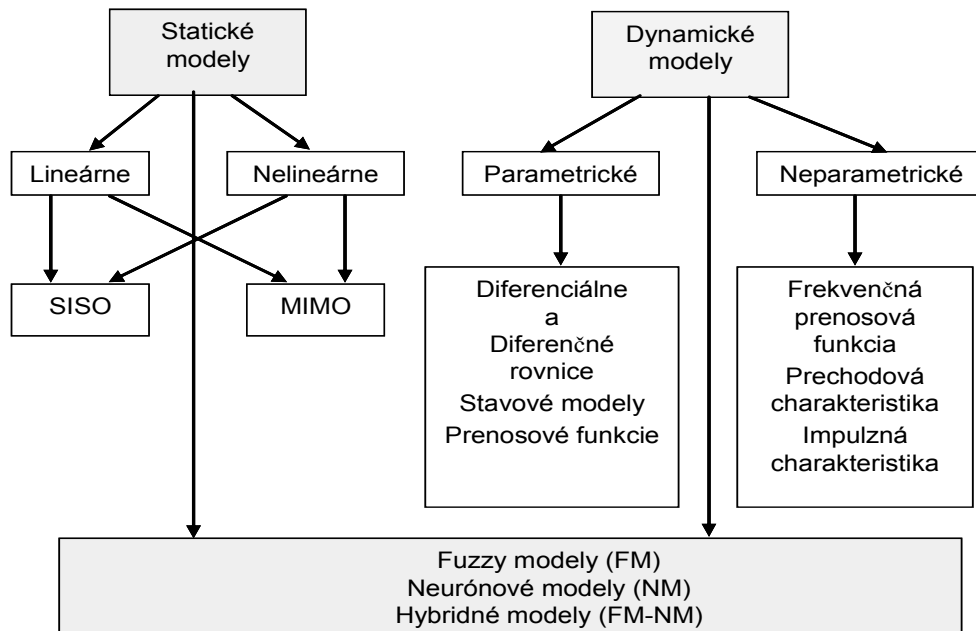
Neparametrické nelineárne fuzzy modely a opisy

- typ **Mamdani** (pre i-té pravidlo, A^i_1, \dots, B^i_2 sú fuzzy množiny)

Rⁱ: IF x_1 is A^i_1 and x_2 is A^i_2 and ... x_n is A^i_n THEN y_1 is B^i_1 and y_2 is B^i_2 and ... y_m is B^i_m

- typ Takagi Sugeno (pre i -té pravidlo)

R^i : *IF* x_1 is A_1^i and x_2 is A_2^i and ... x_n is A_n^i
THEN y_1 is $y_i = b_{i0} + b_{i1}x_1 + b_{i2}x_2 + \dots + b_{in}x_n, 1 \leq i \leq m$



Obr. 6. Všeobecná klasifikácia matematických modelov pre úlohy riadenia

Ak nemáme k dispozícii jazykový opis systému, ale môžeme použiť vstupno – výstupné numerické údaje z experimentu, fuzzy model systému identifikujeme pomocou nich.

3.2 Využitie metód zhlukovej analýzy (Cluster analysis) pri modelovaní procesov fuzzy množinami

Zhluková analýza sa zaoberá metódami zaradenia objektov do skupín tak, aby bola čo najväčšia podobnosť v rámci skupiny a čo najväčšia rozdielnosť medzi skupinami. Zhluková analýza v skutočnosti zahŕňa množstvo metód pre generovanie optimálnej štruktúry fuzzy modelov.

Pre praktické aplikácie sa využívajú najmä nasledovné metódy:

1. Hierarchické zhlukovacie metódy
2. Nehierarchické zhlukovacie metódy

Hierarchické zhlukovacie metódy vychádzajú zo spájania jednotlivých objektov, ktoré reprezentujú zhluky. Ich spájaním sa v každom kroku počet zhlukov postupne

zmenšuje, až sa nakoniec všetky zhluky spoja do jedného celku. Hierarchické metódy vedú k hierarchickej (stromovej) štruktúre, ktorá sa graficky zobrazuje ako stromový diagram (dendrogram). Stromové zhlukovacie metódy začínajú výpočtom vzdialeností medzi objektmi. Využíva sa buď Euklidovská vzdialenosť alebo vzdialenosť Manhattan (City-block). Euklidovská vzdialenosť medzi objektmi i a j s n charakteristikami (premennými) sa vypočíta podľa vzťahu:

$$d_{i,j} = \sqrt{\sum_{k=1}^n (x_{ik} - x_{jk})^2} \quad (1)$$

Vzdialenosť Manhattan (City-block):

$$d_{i,j} = \sum_{k=1}^n |x_{ik} - x_{jk}| \quad (2)$$

Keď už sú vypočítané vzdialenosti medzi všetkými dvojicami objektov, musíme určiť pravidlo, podľa ktorého sa budú objekty spájať do zhlukov, teda ako sa bude určovať vzdialenosť medzi zhlukmi. Existujú viaceré pravidlá spájania: jednoduché spájanie (najbližší sused), kompletné spájanie (najvzdialenejší sused), nevážený párový priemer (priemer skupín), vážený párový priemer (jednoduchý priemer), nevážený centroid (centroid), vážený centroid (medián) a Wardova metóda. Táto metóda sa zreteľne odlišuje od všetkých ostatných pretože na určenie vzdialenosti medzi zhlukmi využíva štatistický prístup.

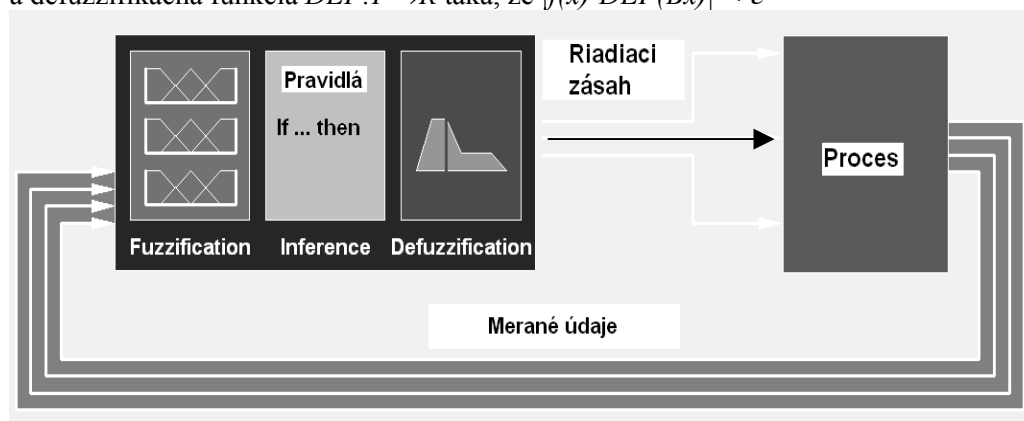
Nehierarchické zhlukovacie metódy nevytvárajú stromovú štruktúru. Najznámejšia nehierarchická zhlukovacia metóda je metóda *k-priemerov* (k-means). Táto metóda sa vyznačuje tým, že vyprodukuje presne k-zhlukov, tak aby bol vnútro skupinový súčet štvorcov minimálny. Je to najvhodnejšia metóda na formovanie malého počtu zhlukov z veľkého počtu pozorovaní.

Aplikácia metód výpočtovej inteligencie nachádza uplatnenie vo všetkých oblastiach ľudskej činnosti. Fuzzy logika v riadení (Fuzzy Logic Control) je aplikovaná pre svoju jednoduchosť a kvalitu riadenia tak do výrobkov spotrebnej elektroniky ako aj v priemyselných oblastiach, bankovníctve a zdravotníctve.

Fuzzy logika predstavuje teda univerzálny aproximátor umožňujúci pomocou základných operácií ako je fuzifikácia, defuzifikácia pomocou lingvistických premenných aproximovať chovanie sa procesu. Pôvodná myšlienka využitia fuzzy logiky len v rozhodovacích procesoch v ekonómii, filozofii a psychológii postupne prerástla do technických disciplín a oblasti a stala sa evolučným prvkom v aplikácii moderných algoritmov modelovania a riadenia silno-nelineárnych a neurčitých procesov, ktoré nedokážeme opísať bežnými prostriedkami (diferenciálne rovnice, diferenčné rovnice, I/O opisy pomocou prenosových funkcií v Laplaceovej a Z-transformácii). Aproximácia chovania procesu použitím fuzzy logiky sa dá opísať a vyjadriť pomocou *IF-THEN* pravidiel.

Fuzzy množiny predstavujú univerzálny aproximátor (fuzzy aproximácia je schopná aproximovať s ľubovoľnou presnosťou každú spojitú ohraničenú funkciu.

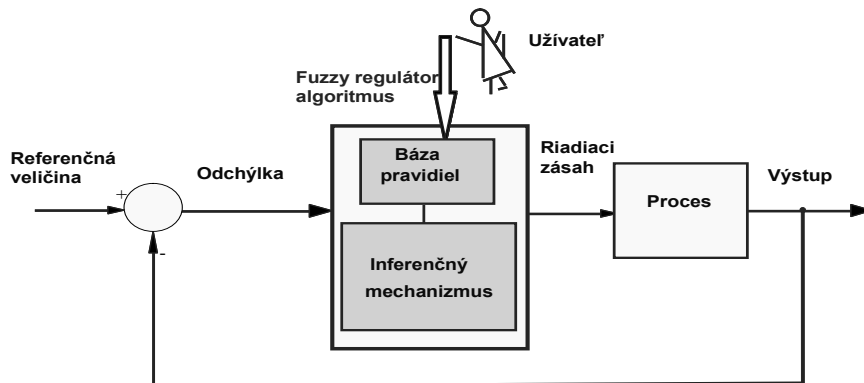
Veta (bez dôkazu): Ak $U \subseteq R$ je kompaktná množina, potom ku každej ohraničenej spojitej funkcii $f: U \rightarrow R$ a pre $\varepsilon > 0$ existujú fuzzy množiny $A_j \subseteq U$, $B_j \subseteq R$, $j=1, \dots, m$ a defuzzifikačná funkcia $DEF: F \rightarrow R$ taká, že $|f(x) - DEF(Bx)| < \varepsilon$



Obr. 7. Bloková schéma zapojenia fuzzy regulátora do riadenia nelineárnych procesov

V priebehu niekoľkých posledných rokov sa ukázalo že pomocou týchto pravidiel môžeme navrhnúť a aplikovať algoritmy ovládania a riadenia procesov úspešnejšie a predovšetkým jednoduchšie ako je to možné klasickými riadiacimi algoritmiami založenými na I/O regulátoroch typu PID alebo stavových lineárno-kvadratických (LQ a LQG) regulátoroch. Pretože aj v súčasnosti je najviac algoritmov riadenia realizovaných v priemysle typu PID, najväčšiu množinu aplikácií v priemysle (až 89%) predstavujú digitálne realizácie týchto algoritmov, ktoré sa zväčša úspešne implementujú do jednoduchých spätnoväzbových štruktúr (obr. 7.)

Samotné nastavovanie parametrov PID riadiacich algoritmov je ešte aj dnes vedou aj umením a vyžaduje mnohokrát poznať matematický model riadeného procesu a prvotne realizovať identifikáciu matematických modelov a odhadu ich parametrov a až potom sekundárne nastavovať a počítať optimálne hodnoty koeficientov riadiaceho zásahu. Z hľadiska zatriedenia fuzzy modelov medzi zaužívanú klasifikáciu matematických modelov ich možno umiestniť tak ako je to znázornené na obr. 6.



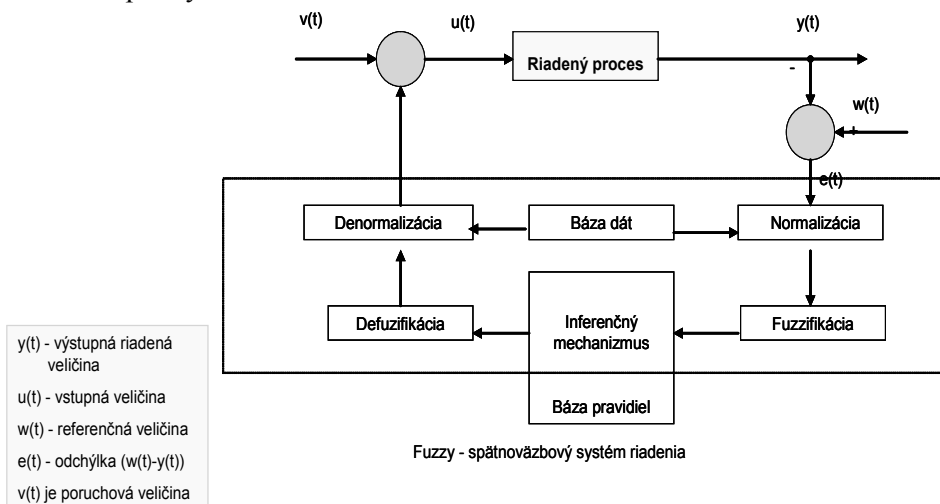
Obr. 8. Spätnoväzbová štruktúra riadenia s fuzzy regulátorom

Štruktúra fuzzy riadenia (obr. 7 a obr. 8) (fuzzy regulátora) pozostáva z troch základných blokov :

- Blok fuzifikácie
- Blok defuzifikácie
- Blok riadenia

Pri praktickej aplikácii fuzzy algoritmov riadenia je potrebné však uvažovať rozdielnú štruktúru riadenia oproti konvenčným štruktúram. Pri aplikácii fuzzy algoritmov spätnoväzbového riadenia musíme pôvodnú spätnoväzbovú štruktúru ešte rozšíriť o bloky:

- Blok normalizácie
- Blok denormalizácie
- Blok pre výber inferenčného mechanizmu



Obr. 9. Základná spätnoväzbová štruktúra fuzzy riadenia

Blok fuzifikácie transformuje namerané, snímané (ostré) údaje na fuzzy množiny. Blok defuzzifikácie má opačnú úlohu a to transformovať výstupné údaje na reálne skutočné (ostré) čísla umožňujúce realizovať riadiaci zásah.

Prvým krokom k realizácii *fuzifikačných* a *defuzzifikačných* procedúr je potreba upraviť (normalizovať) merané údaje do hodnôt v intervale $\langle 0,1 \rangle$, resp. $\langle -1,1 \rangle$, ako aj spätne transformovať (denormalizovať) výstupné fuzzy množiny na reálne ostré čísla odpovedajúce skutočným reálnym hodnotám. Samotný proces normalizácie v spätnoväzbovom obvode riadenia sa týka predovšetkým troch veličín a to výstupnej riadenej veličiny, vstupnej veličiny - riadiaci zásah, regulačnej odchýlky, ale aj diferencie riadiaceho zásahu a regulačnej odchýlky. Pri *normalizácii* ako aj pri *denormalizácii* je potrebné poznať jednotlivé rozsahy minimálnych a maximálnych veličín. Interval do ktorého transformujeme snímané údaje predstavuje *normalizované univerzum*. Pre regulačnú odchýlku je to napr. interval

$$\begin{aligned} e &\in \langle e_{\min}, e_{\max} \rangle \\ e_{\min} &= w_{\min} - y_{\max} = -\varepsilon_1 \\ e_{\max} &= w_{\max} - y_{\min} = \varepsilon_1 \end{aligned}$$

Ak meraný rozsah regulačnej odchýlky je $\langle -\varepsilon_1, \varepsilon_1 \rangle$ a chceme ho zobrazit' na fuzzy množinu v intervale $\langle \varepsilon_2, \varepsilon_2 \rangle$, potom musíme regulačnú odchýlku prenásobiť mierkou $k_e = \varepsilon_2 / \varepsilon_1$.

Pre vytvorenie celkového fuzzy algoritmu riadenia musíme podobným postupom prepočítať aj ostatné veličiny v riadiacom obvode. Defuzzifikované normované hodnoty riadiaceho zásahu je potrebné tak isto pretransformovať na výsledný interval, ktorý bude predstavovať skutočnú hodnotu riadiaceho zásahu.

$$\begin{aligned} u &\in \langle u_{\min}, u_{\max} \rangle \\ u_{\min} &= -\delta_1 \\ u_{\max} &= \delta_1 \end{aligned}$$

Teda ak pôvodný interval riadiaceho zásahu (po defuzzifikácii) je $\langle -\delta_1, \delta_1 \rangle$ a výsledný interval je $\langle -\delta_2, \delta_2 \rangle$, potom koeficient pre prepočet riadiaceho zásahu je $k_u = \delta_2 / \delta_1$.

Táto úprava sa realizuje v denormalizačnom bloku. Dominantnou časťou komplexnej štruktúry fuzzy riadenia je tzv. ústredný modul riadenia, ktorý je zložený zo submodulu bázy pravidiel a submodulu interferenčného mechanizmu. S bázou pravidiel a s ďalšími modulmi komunikuje báza dát. *Báza dát* a *báza pravidiel* tvoria modul nazývaný *báza znalostí*. *Báza dát* zabezpečuje a poskytuje nevyhnutné informácie pre správne fungovanie a súčinnosť fuzifikačného a defuzzyfikačného bloku a bázy pravidiel. Báza dát obsahuje:

- funkcie príslušnosti všetkých vstupných a výstupných fuzzy množín
- intervaly, v ktorých sú vymedzené hodnoty vstupných a výstupných veličín

- koeficienty prepočtu (merítka), ktoré boli použité pri normalizácii a denormalizácii

Báza pravidiel obsahuje všetky typy pravidiel, ktoré sú potrebné pre lingvistickú aproximáciu vyjadrujúcu závislosť medzi vstupmi a výstupmi riadiaceho systému. Jednotlivé pravidlá sú v tvare *IF* (merané premenné procesu - stav procesu) *THEN* (riadiaci zásah). Pravidlá sú medzi sebou prepojené spojku *OR* (t.j. alebo), ktorá je interpretovaná ako logický súčet. *Inferenčný mechanizmus* umožňuje stanoviť výstupnú fuzzy množinu pre danú vstupnú fuzzy množinu, alebo ostrú vstupnú hodnotu pri zvolenom spôsobe interpretácie implikácií v pravidlách.

Samotný jednoduchý fuzzy algoritmus riadenia (regulátor) sa chová ako všeobecný nelineárny stavový regulátor, ktorý realizuje statickú transformáciu vstupných a stavových premenných. Pre zohľadnenie dynamiky jednoduchého spätnoväzbového regulátora je potrebné poznať nasledujúce premenné (vzorkované diskkrétne hodnoty v diskrétnych okamžikoch $t=kT$, kde T je perióda vzorkovania)

- hodnotu odchýlky $e(k) = w(k) - y(k)$
- diferenciu odchýlky: $\Delta e(k) = e(k) - e(k-1)$
- diferenciu riadiaceho zásahu: $\Delta u(k) = u(k) - u(k-1)$

kde $w(k)$ označuje referenčnú premennú, $y(k)$ výstupnú riadenú (regulovanú) veličinu a $u(k)$ riadiaci zásah v k -tom diskrétnom kroku.

Všeobecný opis takéhoto regulátora je možné vyjadriť diferencnou rovnicou:

$$u(k) = f(e(k), e(k-1), \dots, e(k-n_e), u(k-1), u(k-2), \dots, u(k-n_u))$$

kde f je nelineárna funkcia pre aproximáciu chovania sa regulátora.

Pre praktické aplikácie fuzzy algoritmov riadenia sú v súčasnosti využívané analógie konvenčných diskrétnych regulátorov, tzv. PID algoritmov riadenia (Proporcionálne Integračno - Derivačný algoritmus riadenia). Na rozdiel od konvenčných foriem PID algoritmov, ktoré sú lineárne, fuzzy analógie týchto regulátorov sú nelineárne. V praxi sa využívajú štyri základné formy opisov a vyjadrení a to:

$$u(k) = K_p e(k)$$

(riadiaci zásah vstupujúci do procesu $u(k)$ v k -tom diskrétnom kroku je priamo úmerný súčinu aktuálnej hodnoty regulačnej odchýlky $e(k)$ a koeficienta zosilnenia K_p (proporcionálny algoritmus riadenia).

Pravidlá pre zápis takéhoto zákona riadenia sú veľmi jednoduché a sú v tvare:

$$\text{if } e(k)=FM1 \rightarrow \text{THEN } u(k)=FM2 ,$$

kde $FM1$ a $FM2$ sú príslušné fuzzy množiny regulačnej odchýlky, resp., riadiaceho zásahu.

Najčastejším fuzzy riadiacim algoritmom v spätnoväzbovom obvode je tzv. F-PD regulátor (riadiaci zásah je lineárnou kombináciou súčtu proporcionálnej časti

a diferenciálnej (derivačnej časti) časti. Riadiaci zásah F-PD riadenia je vyjadrený diferenciálnou rovnicou:

$$u(k) = K_p e(k) + K_D \Delta e(k)$$

(koeficienty tohto regulátora predstavujú hodnoty zosilnení proporcionálnej resp. diferenciálnej časti). Podobne ako pri konvenčných tzv. PD algoritmoch aj fuzzy F-PD algoritmus dokáže stabilizovať proces. Pre fuzzy PD algoritmus (F-PD) je riadiaci zásah generovaný na základe pravidiel:

$$\text{if } e(k)=\text{FM1 and } \Delta e(k)=\text{FM then } u(k)=\text{FM2}$$

(FM1 a FM2 sú príslušné fuzzy množiny).

Súbor pravidiel, podľa ktorých je nastavovaná hodnota riadiaceho zásahu z F-PD algoritmu môžeme graficky znázorniť v dvojrozmernom priestore (viď. Tab. 1). Počet pravidiel je voliteľný, štandardne pre F-PD regulátor je postačujúca rozmer (7 x 7)

Tab. 1.: Báza pravidiel pre fuzzy PD regulátor môže byť interpretovaná maticou pravidiel (7x7)

| Δe e | <i>NB</i> | <i>NM</i> | <i>NS</i> | <i>ZE</i> | <i>PS</i> | <i>PM</i> | <i>PB</i> |
|-----------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| <i>NB</i> | NB | NB | NB | NB | NM | NS | ZE |
| <i>NM</i> | NB | NB | NB | NM | NS | ZE | PS |
| <i>NS</i> | NB | NB | NM | NS | ZE | PS | PM |
| <i>ZE</i> | NB | NM | NS | ZE | PS | PM | PB |
| <i>PS</i> | NM | NS | ZE | PS | PM | PB | PB |
| <i>PM</i> | NS | ZE | PS | PM | PB | PB | PB |
| <i>PB</i> | ZE | PS | PM | PB | PB | PB | PB |

Trošku zložitejšia aproximácia odvodu riadiaceho zásahu je pre PI (proportionálno – integračného riadenia). Fuzzy PI (F-PI) algoritmus pracuje podobne ako PD algoritmus s tým rozdielom že sa snažíme aproximovať deriváciu riadiaceho zásahu, výstupom je teda diferenciálna riadiaceho zásahu, ktorú musíme potom integrovať aby sme dostali hodnotu riadiaceho zásahu:

$$\Delta u(k) = K_p \Delta e(k) + K_I e(k)$$

Pravidlá pre F-PI algoritmus riadiaceho zásahu majú tvar:

$$\text{if } e(k)=\text{FM and } \Delta e(k)=\text{FM} \rightarrow \text{then } \Delta u(k)=\text{FM},$$

F-PID algoritmus riadenia môžeme potom najjednoduchšie získať kombináciou PD a PI algoritmov (ako súčet PD a PI riadiaceho zásahu). Nerekurentná forma výpočtu riadiaceho zásahu je potom v tvare

$$u(k) = K_p e(k) + K_I \sum e(k) + K_D \Delta e(k)$$

Jedná sa teda o aproximáciu funkčnej závislosti (nerekurentná forma)

$$u(k) = f\left(e(k), \sum e(k), \Delta e(k)\right)$$

alebo rekurentná forma :

$$u(k) - u(k-1) = u(k) = f(e(k), e(k-1), e(k-2))$$

4 Zovšeobecnenie použitia jednoduchých fuzzy algoritmov riadenia

Hlavnou a základnou požiadavkou pre návrh kvalitných stabilizujúcich fuzzy (PI, PD a PID) algoritmov riadenia je návrh a aplikácia pravidiel. V súčasnosti bolo vypracovaných množstvo „šablón“ vhodných pre určité typy procesov (spotrebná elektronika – práčky, fotoaparáty, lekárske prístroje, pohony, automobily ABS, riadenie kotlov a pod.), ktoré sú aplikované a využívané v praxi. Pre určité typy typických aplikácií fuzzy regulátorov typu P, PI, PD a PID sú určité pravidlá už odskúšané a odporúčané. Známe sú určité všeobecné pravidlá – metapravidlá spracované v roku 1977 [6]. Báza pravidiel odvodená na základe troch základných metapravidiel, ktoré vystihujú vlastnosti jednoduchého spätnoväzbového riadenia pozostáva z týchto základných metapravidiel:

Metapravidlo 1: ak je regulačná odchýlka $e(k)$ a jej diferenciacia $\Delta e(k)$ nulová, potom riadiaci zásah nie je potrebné meniť a nemalo by dôjsť k žiadnemu regulačnému zásahu $u(k)$ (akčná veličina i jej derivácia (diferencia) je nulová).

Metapravidlo 2: ak regulačná odchýlka $e(k)$ postupne klesá, potom nie je potrebné meniť aktuálnu hodnotu riadiaceho zásahu $u(k)$.

Metapravidlo 3: ak regulačná odchýlka $e(k)$ sa nemení, potom je riadiaci zásah $u(k)$ je nenulový a je závislý na hodnotách a znamienku odchýlky $e(k)$ a jej diferenciacii $\Delta e(k)$.

Tab. 2. Zohľadnenie metapravidiel pre fuzzy regulátor – stabilizácia procesu

| Δe e | NB | NM | NS | ZE | PS | PM | PB |
|-----------------|----|----|----|----|----|----|----|
| NB | NB | NB | NB | NB | NM | NS | ZE |
| NM | NB | NB | NB | NM | NS | ZE | PS |
| NS | NB | NB | NM | NS | ZE | PS | PM |
| ZE | NB | NM | NS | ZE | PS | PM | PB |
| PS | NM | NS | ZE | PS | PM | PB | PB |
| PM | NS | ZE | PS | PM | PB | PB | PB |
| PB | ZE | PS | PM | PB | PB | PB | PB |

Tab. 2 prezentuje 5 rôznych skupín, ktoré sú funkciou reálnej situácie riadenia.

1. *Prvá skupina pravidiel* - jadro tabuľky (označená bielou farbou) – zohľadňuje situáciu ak sú hodnoty regulačnej odchýlky a jej diferencie malé, resp. nulové (ustálený stav). V tomto prípade to ale znamená, že regulovaná veličina sa málo líši od ustáleného stavu. Riadiaci zásah generovaný na základe týchto pravidiel v jadre tabuľky bude malý alebo nulový.
2. *Druhá skupina pravidiel* (tučne vyznačený obdĺžnik vpravo hore) – zohľadňuje tie stavy, pri ktorých hodnoty regulačnej odchýlky nadobúdajú stredné až veľké

hodnoty so záporným znamienkom. To znamená, že hodnoty regulovanej veličiny sa v značnej miere líšia od hodnôt ustáleného stavu. Ak je zmena odchýlky kladná potom tento stav naznačuje, že regulačná odchýlka sa znižuje. Riadiaci zásah musí byť potom taký aby zabezpečil cieľavedomé približovanie k ustálenej hodnote (napr. „ak hodnota regulačnej odchýlky je veľká a jej zmena je stredná potom riadiaci zásah a jej zmena mala byť malá (záporná)“).

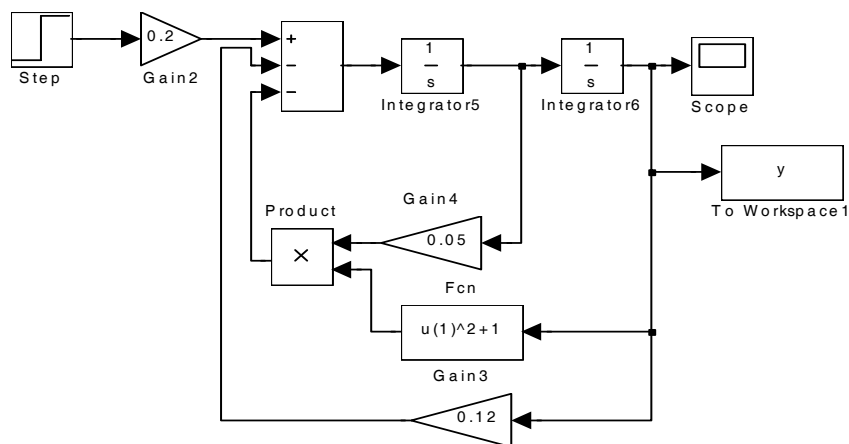
3. *Tretia skupina pravidiel* (napravo dole) – zohľadňuje situáciu ak regulovaná veličina je buď blízko ustálenému stavu alebo je pod ním. Diferencia regulačnej odchýlky je však kladná, čo ale znamená, že sa regulovaná veličina vzdďaluje od ustáleného stavu. Pre zmenu tejto situácie a stabilizáciu procesu je nutné aby zmena nasledujúceho riadiaceho zásahu bola kladná.
4. *Štvrtá skupina pravidiel* (vľavo dole) – zohľadňuje stav ak hodnoty regulačnej odchýlky sú kladné, stredné alebo veľké. Tento stav odpovedá situácii, kedy regulovaná veličina je ďaleko pod hodnotou ustáleného stavu. Pretože zmena regulačnej odchýlky je záporná, hodnoty regulovanej veličiny sa pohybujú smerom k ustálenej hodnote. Riadiaci zásah by mal taký – podobne (ako v skupine pravidiel 2) aby zabezpečil konvergenciu regulovanej veličiny k ustálenej hodnote.
5. *Piata skupina pravidiel* (vľavo hore) – obsahuje skupinu pravidiel, ktoré sú charakterizované tým, že regulovaná veličina je buď blízko ustálenému stavu, alebo je veľmi vzdialená od neho (nad ním). Diferencia odchýlky je záporná, čo ale znamená, že sa regulovaná veličina vzdďaluje od svojho ustáleného stavu. Cieľavedomú zmenu tohto stavu a stabilizáciu procesu riadenia dosiahneme realizáciou záporného riadiaceho zásahu.

5 Aplikácia fuzzy logiky pre modelovanie a riadenie nelineárnych systémov – demonštračné príklady

Prvoradou úlohou pri aplikácii fuzzy logiky v riadení je vytvorenie matematického modelu procesu opísaného fuzzy množinami, overenie kvality modelovania a jeho aplikácia.

Aplikácia fuzzy logiky pre úlohy riadenia nelineárnych systémov je demonštrovaná na modeli nelineárneho systému. Uvažujme nelineárny model riadeného procesu opísaného nelineárnou diferenciálnou rovnicou v tvare:

$$a \frac{d^2 y}{dt^2} + b \frac{dy}{dt} f(y) + cy = Ku(t)$$



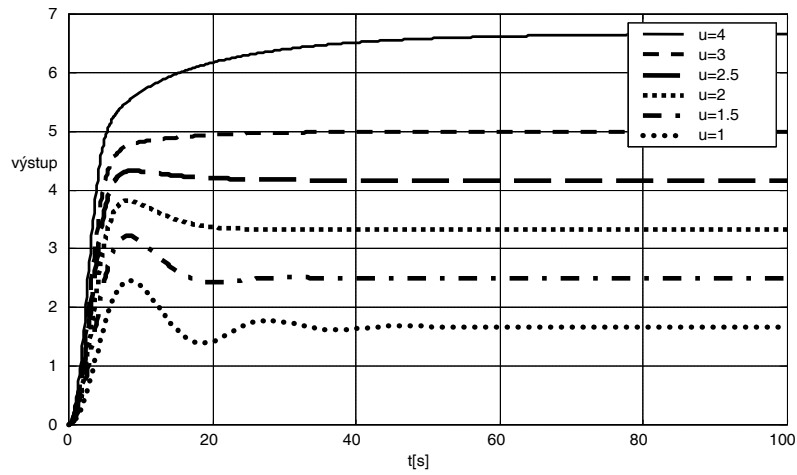
Obr. 10. Schéma simulačného modelu nelineárneho systému v prostredí Matlab-Simulink

Cieľom demonštračného príkladu je návrh postupov pre modelovanie nelineárneho systému konvenčnými riešeniami a návrh algoritmov riadenia na báze fuzzy logiky. Pre simuláciu boli zvolené nasledovné konštanty $a=1$, $b=0.05$, $f(y)=1+y^2$, $c=-0.12$, $K=0.2$.

Nelineárny systém bol simulovaný za účelom modelovania a simulácie v prostredí programového systému Matlab-Simulink [2] (viď. obr. 10). Pre zadanú nelineárnu funkciu bolo prvoradé vytvoriť zodpovedajúcu simulačnú schému v prostredí Simulink za účelom zistenia dynamiky nelineárneho systému.

5.1 Vlastnosti modelu nelineárneho systému

Charaktery prechodových dejov pôvodného nelineárneho systému (NS) sú pre rôzne hodnoty vstupnej veličiny u zobrazené na obr. 11.



Obr. 11. Prechodové deje (NS) pre rôzne hodnoty vstupnej veličiny u

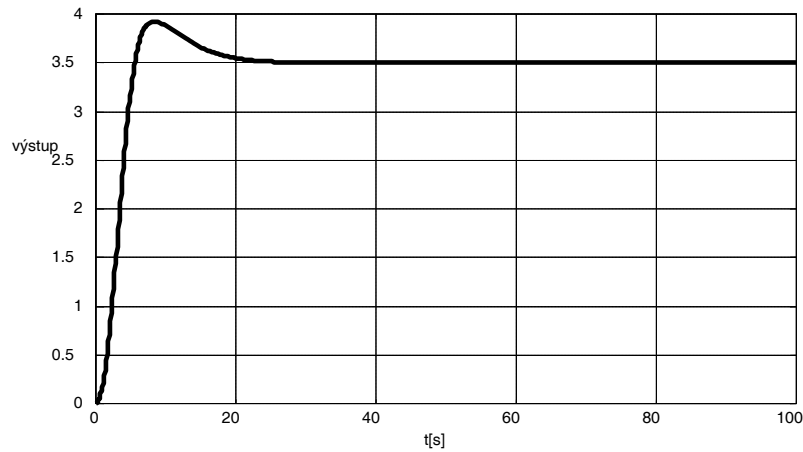
5.2 Linearizácia v pracovnom bode systému

Pre návrh konvenčných algoritmov riadenia typu PID a výpočet jeho parametrov bolo potrebné linearizovať pôvodný nelineárny systém. V ustálenom stave je vstupno – výstupná závislosť (hodnoty) vyjadrená v Tab. č. 3.

Tab. 3. Výber pracovného bodu nelineárneho systému

| | | | | | |
|-----|-----|-----|-----|-----|---|
| u | 1.2 | 1.8 | 2.1 | 2.4 | 3 |
| y | 2 | 3 | 3.5 | 4 | 5 |

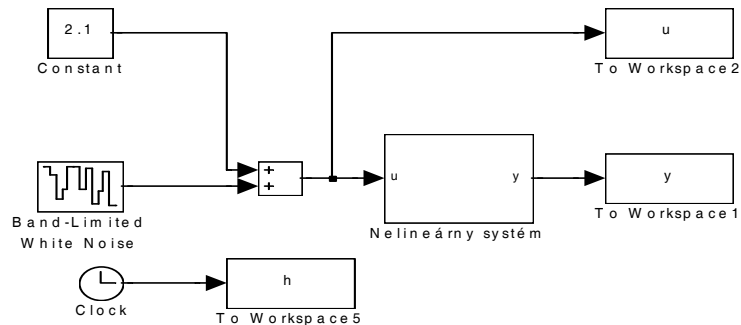
Z hodnôt uvedených v tabuľke Tab. 3 bol vybraný pracovný bod $u = 2.1$ a $y = 3.5$, ktorý sme následne overili i simulačne.



Obr. 12. Časový priebeh dynamiky procesu pre zvolený pracovný bod

5.3 Modelovanie a identifikácia parametrov linearizovaného systému v okolí pracovného bodu.

Úlohou identifikácie bolo nájsť optimálne hodnoty a štruktúru linearizovaného systému. Pre generovanie vstupného signálu potrebného pre identifikáciu bol použitý blok - „Band – Limited White Noise“ (obr. 13)



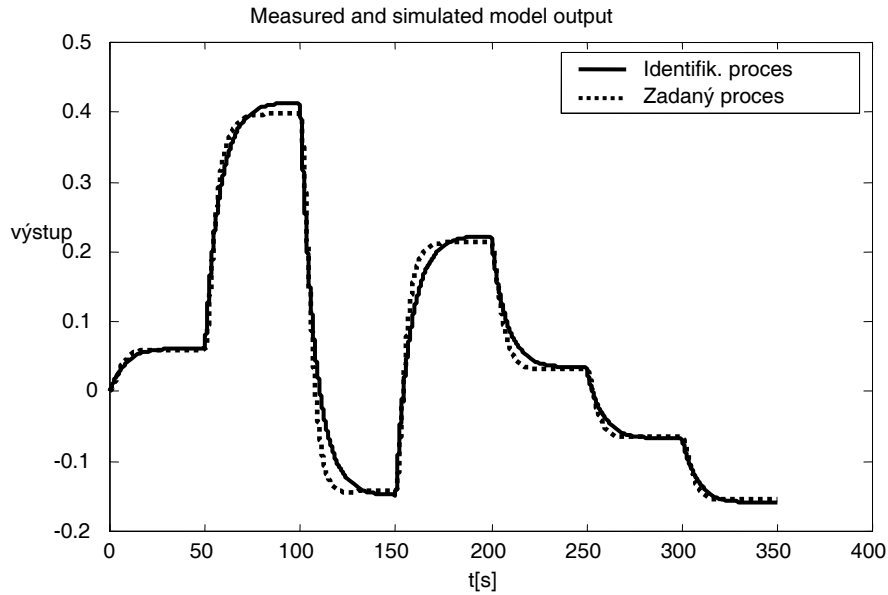
Obr. 13. Simulačná schéma v prostredí Matlab-Simulink použitá pre identifikáciu parametrov

Pre nájdenie optimálnej štruktúry bol aplikovaný TOOLBOX IDENT pomocou ktorého bol získaný model v tvare diferenciálnej rovnice pre rôzne pracovné body, ktorý bol následne prepočítaný na model vyjadrený prenosovou funkciou v s-oblasti.

$$PB [2.1 \ 3.5]$$

$$ARX 220 \quad G_2(s) = \frac{0.0046s + 0.1864}{s^2 + 0.6607s + 0.112}$$

Priebehy výstupnej veličiny pre identifikované modely opísané prenosovou funkciou $G_2(s)$ v porovnaní s reálnymi výstupmi z pôvodného modelu sú zobrazené na obr. 14.



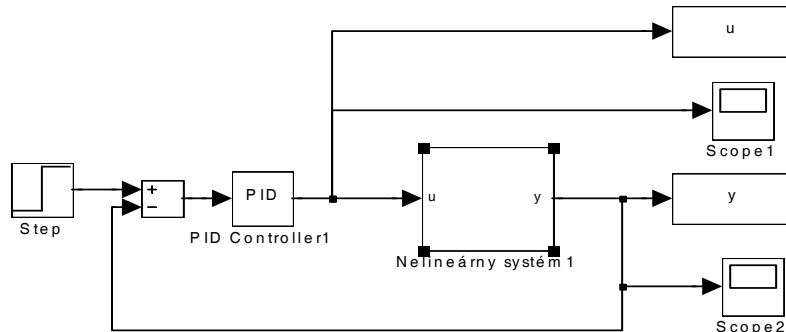
Obr. 14. Časový priebeh pôvodného NS a identifikovaného linearizovaného modelu

5.4 Výpočet optimálnych koeficientov PID regulátora

Pre návrh optimálnej štruktúry a výpočet parametrov PID regulátora bola aplikovaná metóda inverznej dynamiky [15]. Optimálne parametre sú:

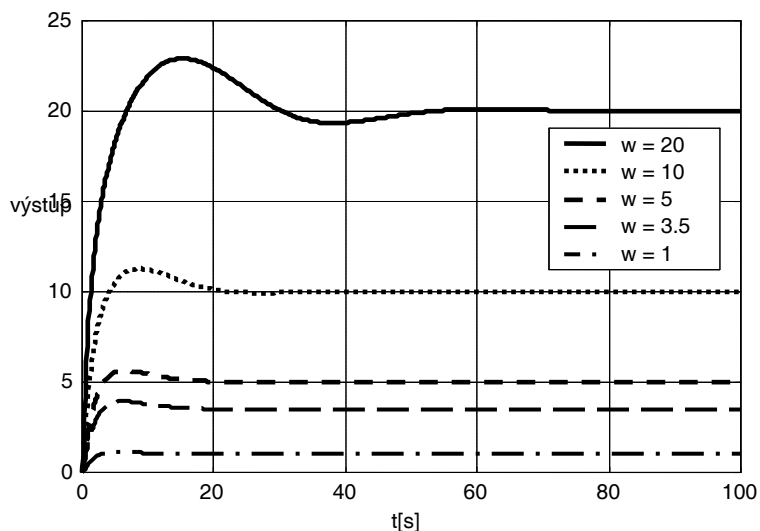
$$K_p = \frac{2T_i}{k_1(2T_w + T)} ; T_i = 2\xi_0 T_0 - T ; T_d = \frac{T_0}{2\xi_0} - \frac{T}{4} ; T_w = 0.2 ; T = 0$$

$$K_p = 17.725, T_i = 5.899, T_d = 1.5135$$



Obr. 15. Simulačná schéma spätnoväzbový obvod pre klasický PID regulátor

Priebeh výstupnej regulovanej veličiny s konvenčným PID regulátorom je na obr. 16. Simulácia bola realizovaná v prostredí Matlab-Simulink (obr. 15).



Obr. 16. Časový priebeh výstupnej regulovanej veličiny pre rôzne hodnoty referenčnej premennej s konvenčným PID regulátorom

5.5 Návrh FUZZY PID regulátora pre nelineárny systém

Pri návrhu Fuzzy PID regulátora vychádzame z dosiahnutých priebehov a hodnôt jednotlivých premenných pri aplikácii konvenčného PID regulátora. Pre každú zo simulovaných veličín (obr. 17) určíme pre návrh fuzzy regulátorov rozsahy, v ktorých sa veličiny pohybujú pri regulácii nelineárnej sústavy. Vstupné a výstupné veličiny fuzzy regulátora a ich rozsahy:

Vstupné veličiny: *Regulačná odchýlka e*, rozsah (-1,1)
 Derivácia regulačnej odchýlky de, rozsah (-2,2)

Výstupná veličina : Riadiaci zásah *u*, rozsah (-5 , 5)

Navrhovaný fuzzy regulátor pracuje s pravidlami typu:

if (e is Ae) and (de is Ade) then (u is Bu)

Všeobecná báza pravidiel je rozmeru (5 x 5) a je uvedená v Tab. č. 4.

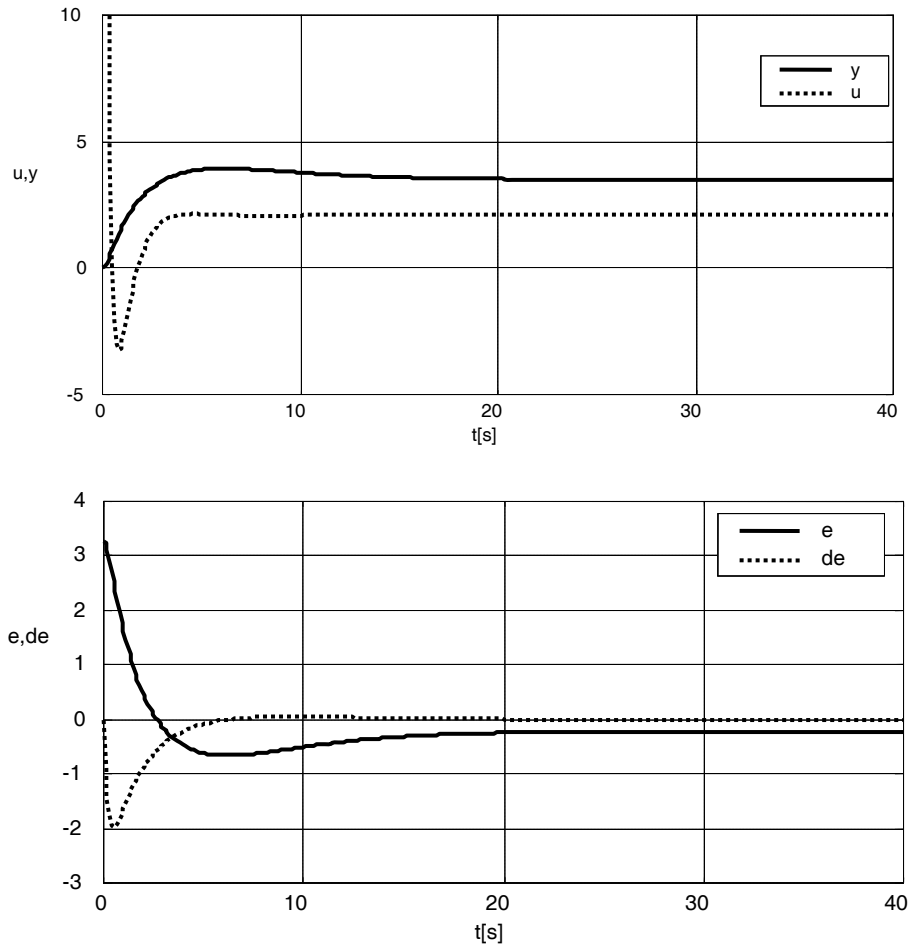
Tab. 4. Súbor pravidiel pre fuzzy regulátor

| de/e | NB | NM | ZE | PM | PB |
|------|----|----|----|----|----|
| NB | NB | NB | NM | NM | ZE |
| NM | NB | NM | NM | ZE | PM |
| ZE | NM | NM | ZE | PM | PM |
| PM | NM | ZE | PM | PM | PB |
| PB | ZE | PM | PM | PB | PB |

kde hodnoty odchýlky sú vyjadrené ako:

| |
|---|
| NB – záporná veľká NM – záporná stredná ZE – nulová PM – kladná stredná PB – kladná veľká |
|---|

Pre dosiahnutie kvality regulácie boli realizované rôzne kombinácie metód fuzifikácie, defuzifikácie a aplikované rôzne druhy funkcií príslušnosti.



Obr. 17. Časový priebeh veličín (u , y , e , de) získaných z aplikácie klasického PID regulátora

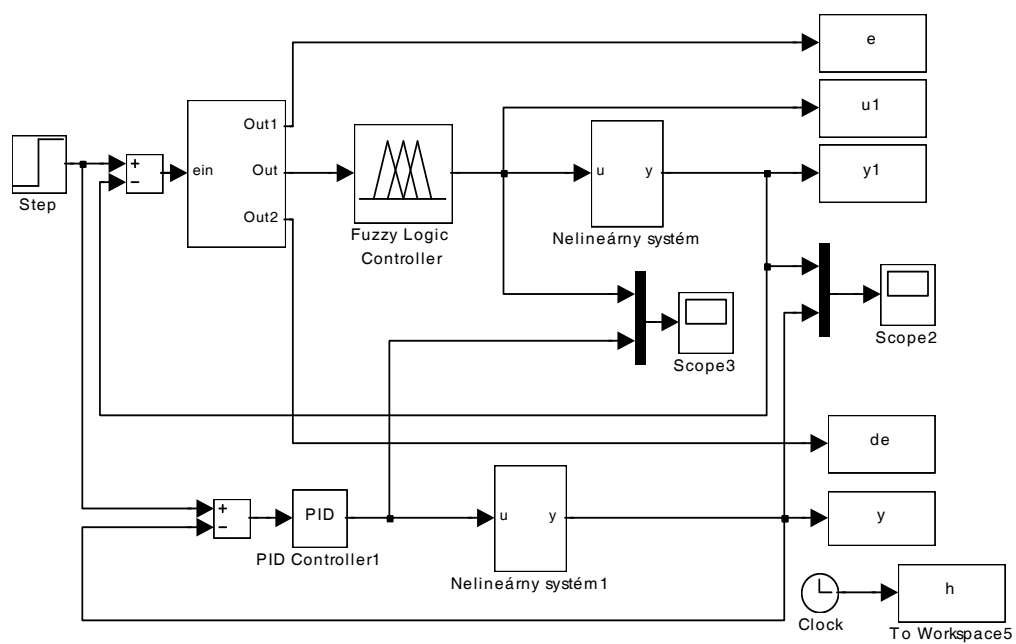
Vybrané metódy fuzifikácie:

| | |
|------------|-----|
| And metóda | min |
| Or metóda | max |
| Implikácia | min |
| Agregácia | max |

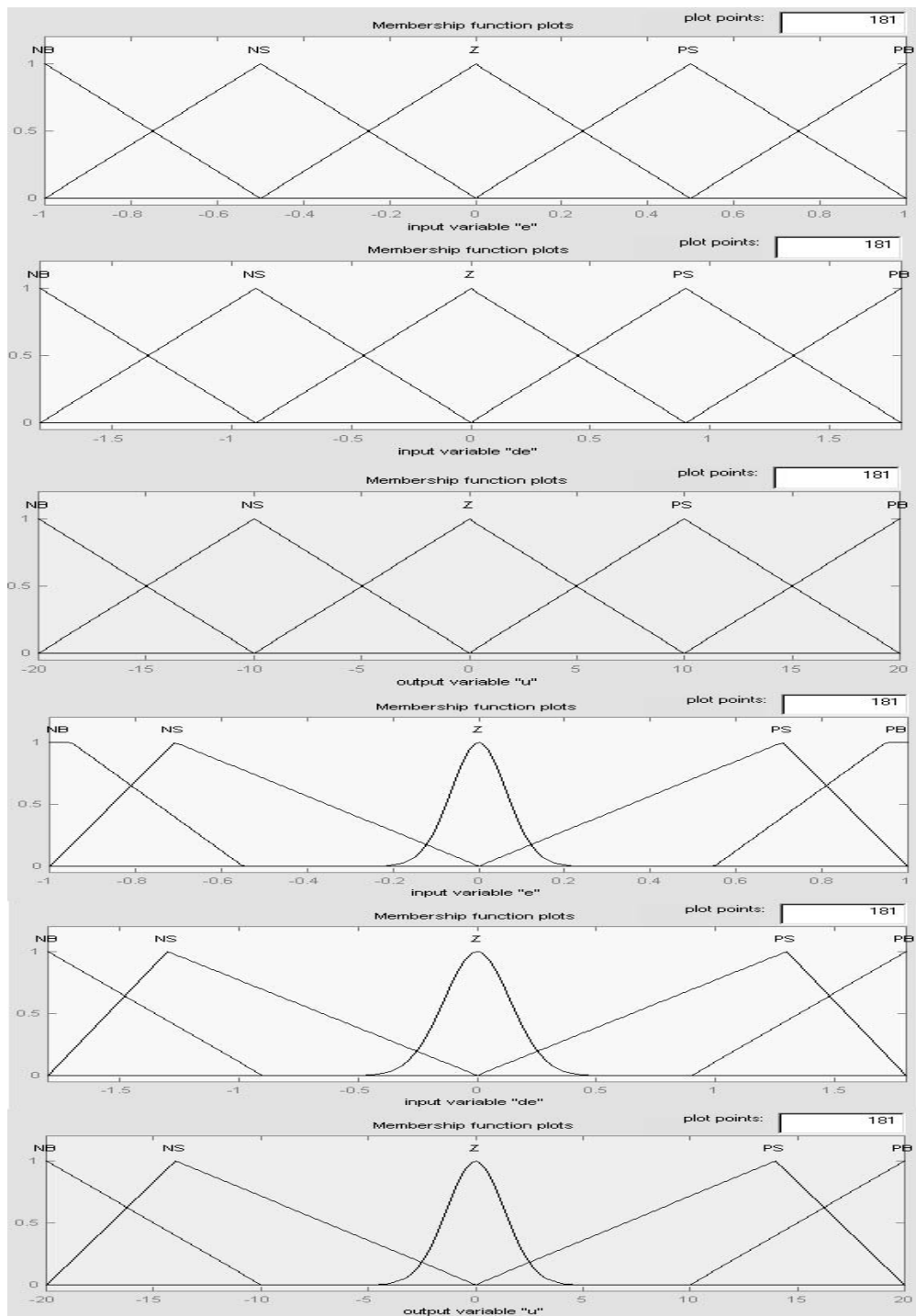
Vybrané a aplikované metódy defuzifikácie:

- metóda centroidu (COA)

Funkcie príslušnosti typ: trojuholníkové (obr. 19)

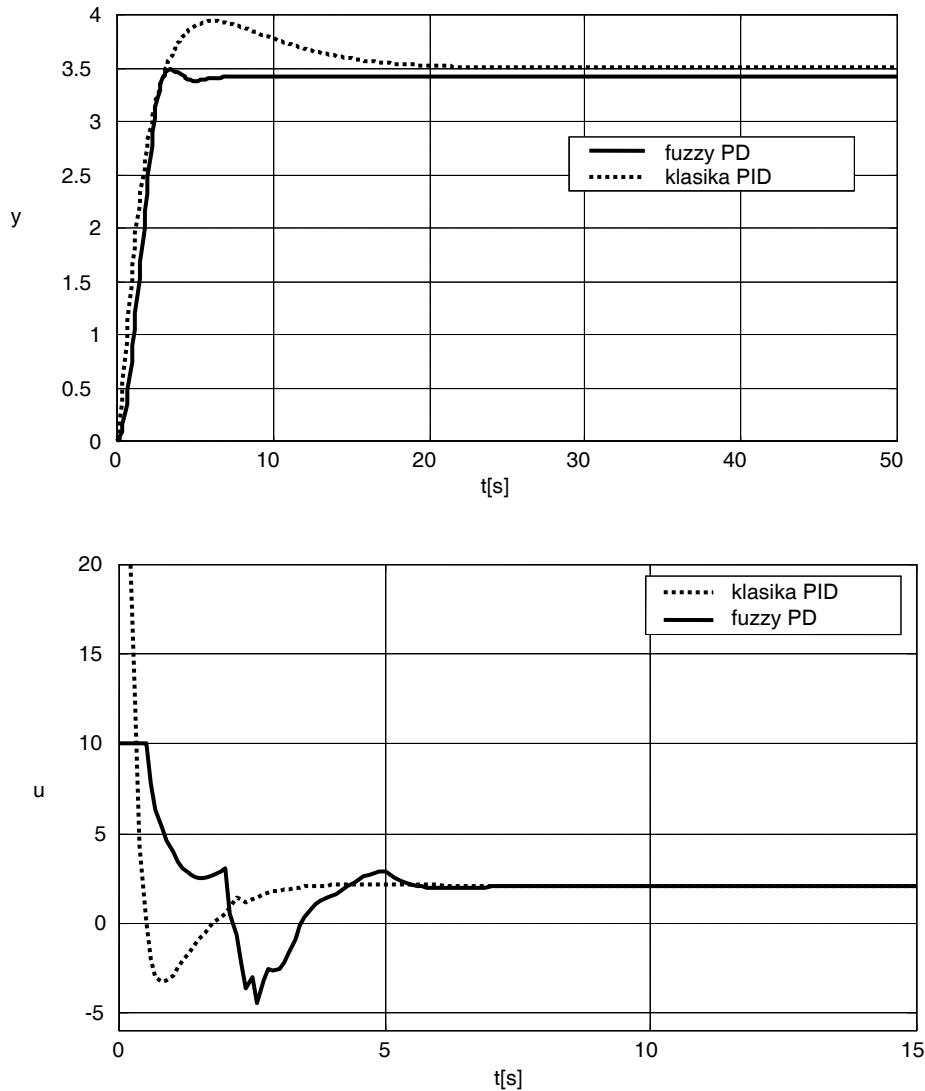


Obr. 18. Simulačná schéma pre fuzzy PD regulátor+klasický PID regulátor



Obr. 19. Typy funkcií príslušnosti a ich rozloženie pre Fuzzy PD regulátor

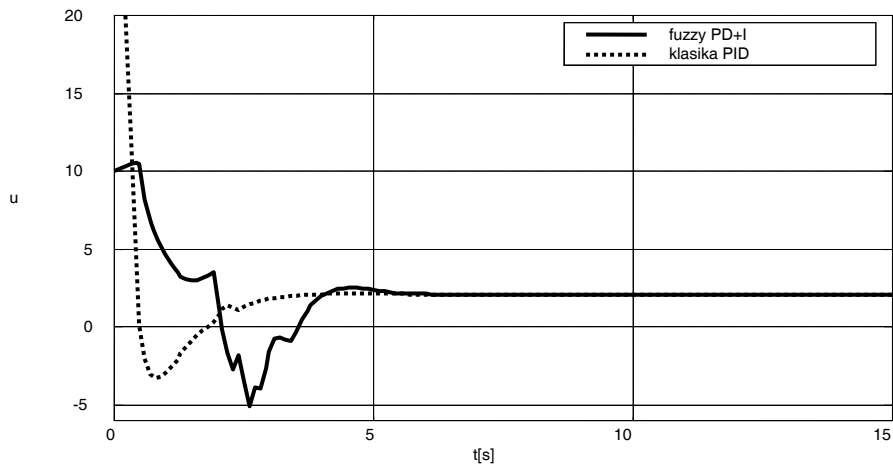
Pri aplikácii iba PD regulátora vznikla malá trvalá regulačná odchýlka, ktorá sa dala odstrániť pridaním I zložky. Časový priebeh regulácie je zobrazený na obr. 20.



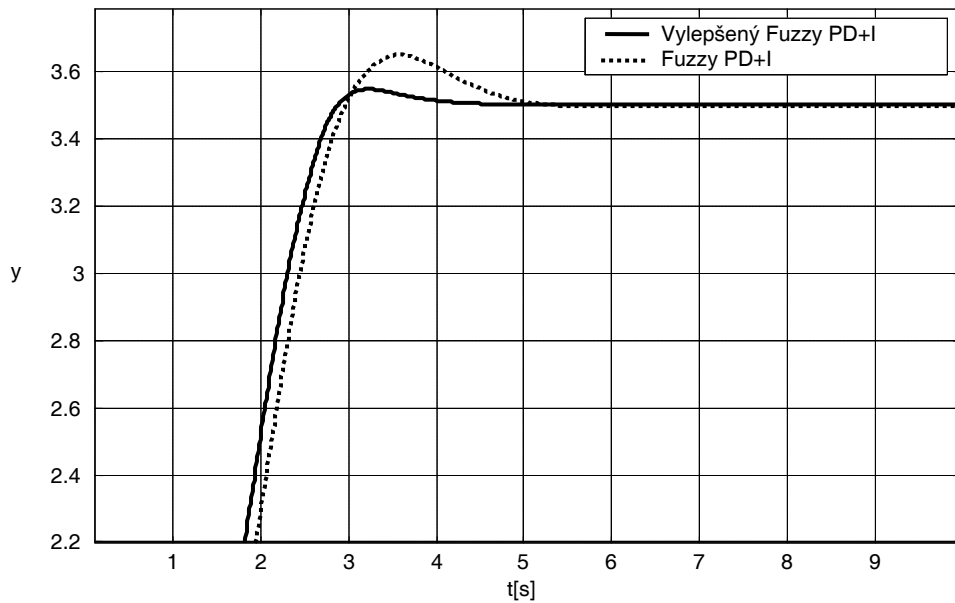
Obr. 20. Časový priebeh regulovanej veličiny y a riadiaceho zásahu u pri aplikácii fuzzy PD regulátora a konvenčného PID regulátora

Na obr. 21 je zobrazený časový priebeh výstupnej veličiny a riadiaceho zásahu pri kombinácii dvoch fuzzy regulátorov typu PD a PI (obr. 21, obr. 22). Dosiahla sa tým nulová regulačná odchýlka a tzv. „mäkko aperiodický priebeh“ pri ktorom sa regulovaná veličina blíži k svojej referenčnej premennej zhora. Výber funkcií príslušnosti bol realizovaný heuristicky a preto štruktúra regulácie nie je optimálna.

V súčasnosti je možné aplikovať štandardné metódy v prostredí Matlab- Simulink (*genfis*, *anfis*), ktoré umožňujú optimalizovať štruktúru fuzzy regulátora a automatický výber tak počtu ako aj rozloženia funkcií príslušnosti (obr. 23).



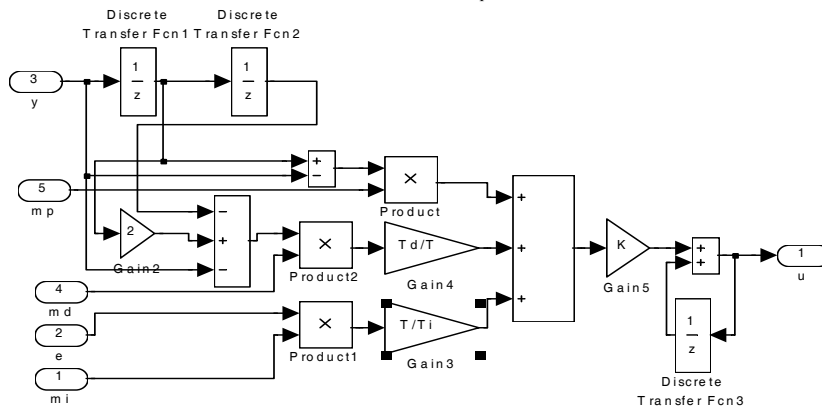
Obr. 21. Časové priebehy veličín u pre fuzzy PD+I regulátor a konvenčný PID regulátor



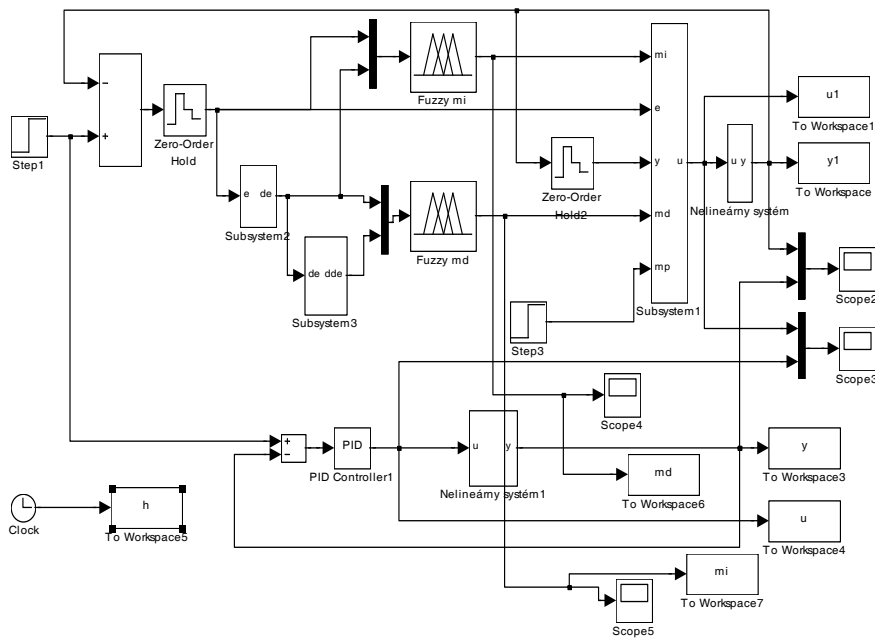
Obr. 22. Časový priebeh výstupnej regulovanej veličiny pri aplikácii fuzzy PD+I regulátora s optimalizáciou tvaru funkcií príslušnosti

5.6 Adaptívny fuzzy PID regulátor

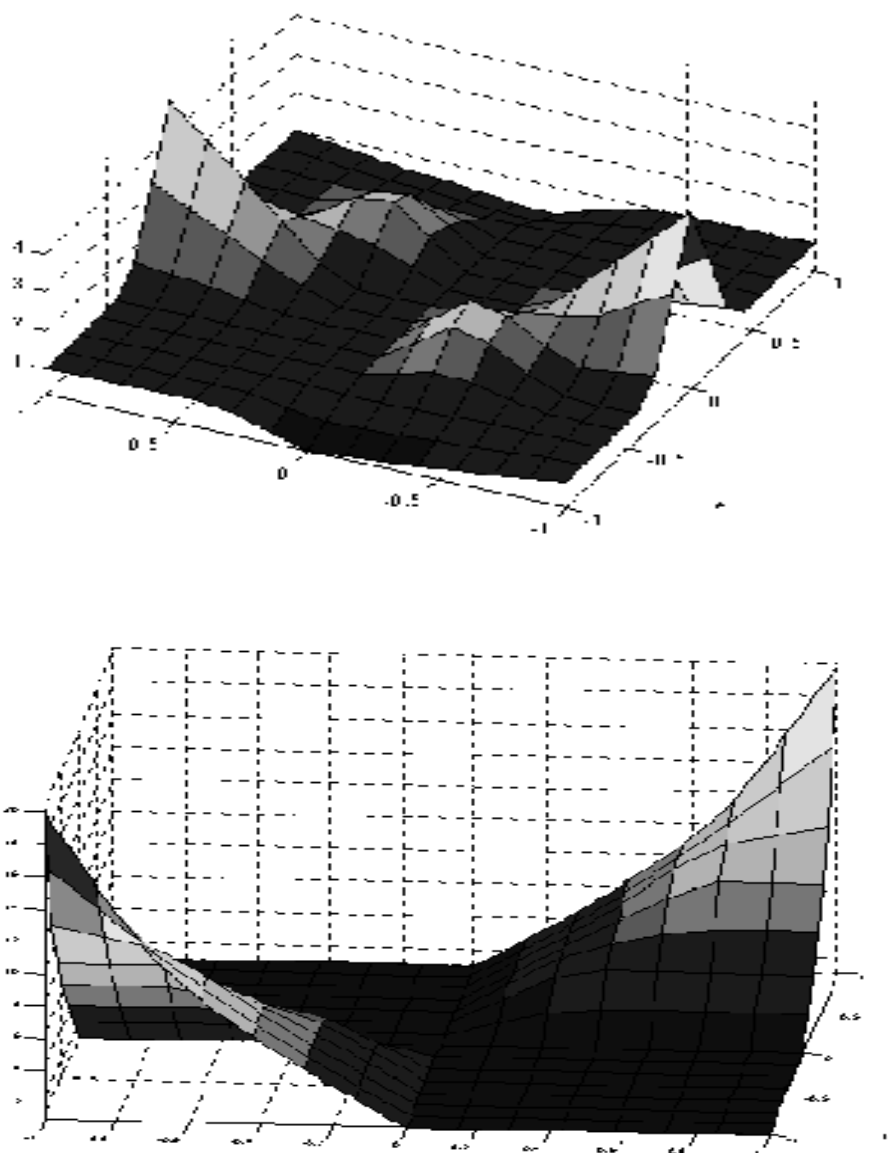
Zlepšenie kvality regulácie a priamu aplikáciu fuzzy logiky do prestavovania koeficientov PID regulátora môžeme optimalizovať ich nastavovaním pomocou fuzzy systému. Ako základná štruktúra bola zvolená prírastková forma PID regulátora upravená o možnosť nastavovania parametrov regulátora a to zložky I, D pomocou fuzzy systému, obr. 23, obr. 24), (koeficienty m_p , m_i , m_d).



Obr. 23. Schéma simulácie prírastkového PID s nastavovaním jeho parametrov



Obr. 24. Bloková schéma simulácie adaptívneho fuzzy PID regulátora pre nelineárny systém



Obr. 25. Vymedzenie priestoru pre fuzzy adaptívne nastavovanie koeficientov m_p , m_i , m_d

Pre jednotlivé zložky regulátora I, D boli zvolené pre fuzzy systémy funkcie príslušnosti rovnomerne rozložené na celom rozsahu pre jednotlivé vstupy (e , d_e , d_{de}), trojuholníkového tvaru. Typ defuzzifikácie centroid. Výstupom sú koeficienty m_i (6 výstupných konštánt), m_d (4 výstupné konštanty).

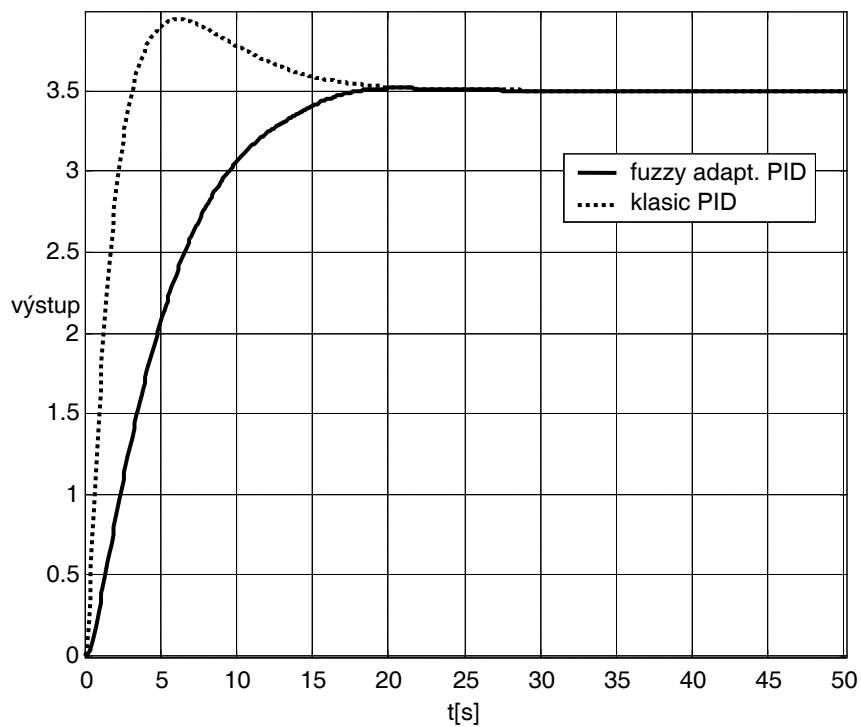
Pre proces fuzifikácie boli aplikované nasledovné metódy:

Fuzifikácia:

| | |
|------------|-----|
| And metóda | min |
| Or metóda | max |
| Implikácia | min |
| Agregácia | max |

Defuzifikácia: metóda centroidu (COA)
Funkcie príslušnosti typ : trojuholníkové

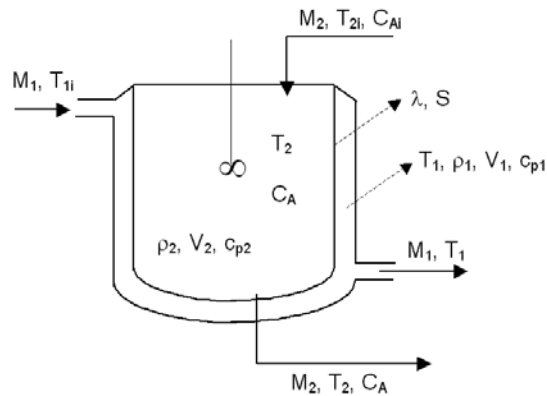
Na obr. 26 sú zobrazené časové priebehy výstupnej regulovanej veličiny získané pri aplikácii adaptívneho fuzzy PID regulátora a konvenčného regulátora. Kvalita regulácie F PID oproti konvenčnému PID regulátoru je výrazne vyššia (bez preregulovania).



Obr. 26. Časový priebeh regulácie s adaptívnym fuzzy PID regulátorom a konvenčným PID regulátorom pre nelineárny model systému.

5.7 Aplikácia fuzzy logiky pre modelovanie a riadenie chemického reaktora

Chemický reaktor predstavuje nelineárny dynamický systém. V tejto časti kapitoly bude demonštrovaná metodika modelovania a riadenia nelineárneho systému pomocou moderných metód fuzzy logiky. Princiálna schéma chemického reaktora je uvedená na obr. 27.



Obr. 27. Princiálna schéma chemického reaktora

Pre porovnanie fuzzy modelu reaktora a analytického modelu budeme vychádzať z nasledovného opisu prostredníctvom nelineárnych diferenciálnych rovníc:

Materiálová bilancia:

$$\frac{dC_A}{dt} = \frac{Q_2}{V_2} (C_{Ai} - C_A) - K(T_2)C_A \quad (3)$$

Tepelná bilancia plášťa reaktora:

$$\frac{dT_1}{dt} = \frac{Q_1}{V_1} (T_{1i} - T_1) + \frac{\lambda S}{V_1 \rho_1 c_{p1}} (T_2 - T_1) \quad (4)$$

Tepelná bilancia vnútra reaktora:

$$\frac{dT_2}{dt} = \frac{1}{\rho_2 c_{p2}} (-\Delta Hr) K(T_2) C_A + \frac{Q_2}{V_2} (T_{2i} - T_2) - \frac{\lambda S}{V_2 \rho_2 c_{p2}} (T_2 - T_1) \quad (5)$$

$$\text{kde: } K(T_2) = k_0 \exp\left(\frac{-a}{T_2 + 273,15}\right) \quad (6)$$

Pre modelovanie a simuláciu boli uvažované nasledovné hodnoty parametrov reaktora:

Vstupy do reaktora :

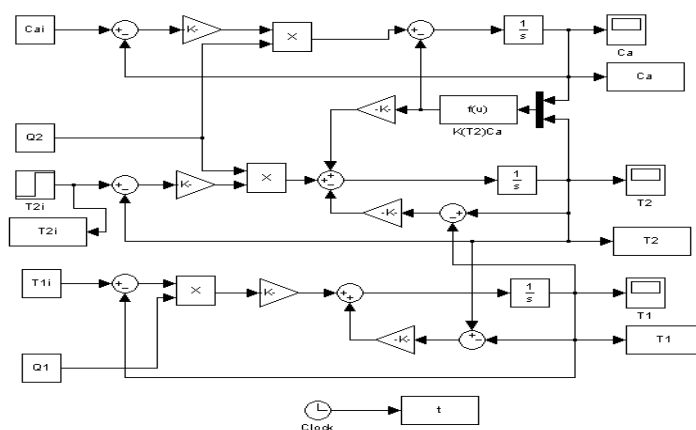
$$\begin{aligned} k_0 &= 1,01 \cdot 10^{12} \text{ s}^{-1}, a = 10\,067,36 \text{ K}, \Delta H_r = 250\,800 \text{ J} \cdot \text{mol}^{-1} \\ c_{p1} &= c_{p2} = 4\,187 \text{ J} \cdot \text{kg}^{-1} \cdot \text{K}^{-1}, \rho_1 = \rho_2 = 1\,000 \text{ kg} \cdot \text{m}^{-3} \\ V_1 &= 0,02 \text{ m}^3, V_2 = 0,05 \text{ m}^3, \lambda = 800 \text{ J} \cdot \text{m}^{-2} \cdot \text{K}^{-1} \cdot \text{s}^{-1} \\ S &= 1,25 \text{ m}^2 \end{aligned}$$

- teplota vody na vstupe do plášťa reaktora T_{1i} [$^{\circ}\text{C}$],
- teplota vody na vstupe do reaktora T_{2i} [$^{\circ}\text{C}$],
- objemový prietok vody do plášťa reaktora Q_1 [$\text{m}^3 \cdot \text{s}^{-1}$],
- objemový prietok vody do reaktora Q_2 [$\text{m}^3 \cdot \text{s}^{-1}$]
- koncentrácia látky A na vstupe do reaktora C_{A1} [$\text{mol} \cdot \text{m}^{-3}$].

Výstupy z reaktora:

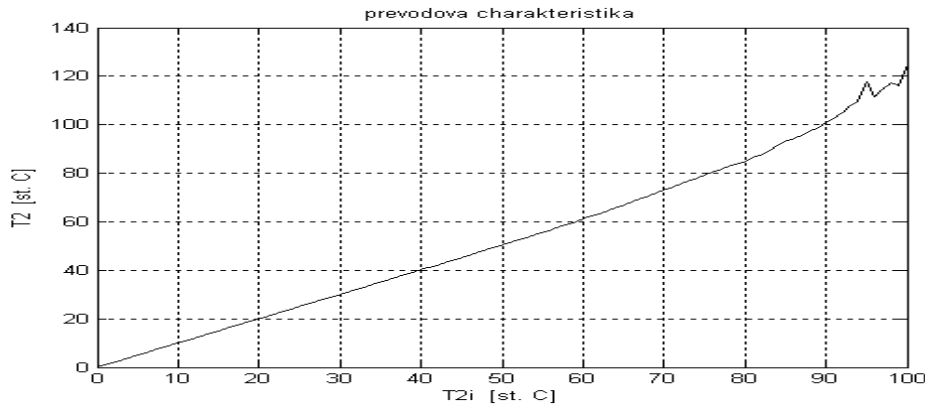
- teplota vody na výstupe z plášťa reaktora T_1 [$^{\circ}\text{C}$],
- teplota vody v reaktore T_2 [$^{\circ}\text{C}$] a
- koncentrácia látky A na výstupe z reaktora C_A [$\text{mol} \cdot \text{m}^{-3}$].

Nelineárny dynamický systém je pomerne zložitý s viacerými vstupmi a výstupmi, pre demonštráciu kvality modelovania fuzzy množinami budeme uvažovať iba jeden vstup/jeden výstup (SISO - single input / single output). Výstupom z reaktora je teplota vody v reaktore T_2 a ako vstupnú veličinu uvažujeme teplotu vody na vstupe do reaktora T_{2i} . Pretože neboli získané údaje priamo z reálneho procesu, fuzzy model bol vytvorený z údajov získaných experimentovaním a simuláciou na matematickom modeli, ktorý bol vytvorený v prostredí Matlab – Simulink, s využitím pôvodného opisu nelineárnymi diferenciálnymi rovnicami.



Obr. 28. Schéma simulácie chemického reaktora v prostredí Matlab Simulink

Závislosť výstupnej T_2 veličiny od vstupnej T_{2i} (od 0 °C až 100°C) vyjadruje prevodová charakteristika vyjadrujúca vplyv nelinearity modelu. Výrazný vplyv nelinearity je až pri vyšších teplotách vstupnej teploty $T_{2i} = 90^\circ\text{C}$.



Obr. 29. Prevodová charakteristika chemického reaktora

Výsledný fuzzy model je vytvorený pre celý rozsah I/O veličín, (vstupná veličina $T_{2i} = 0$ až 100 °C).

Pre sériu vstupných skokov teploty T_{2i} boli merané odpovedajúce hodnoty výstupnej veličiny T_2 . Pre vytvorenie optimálnej štruktúry (optimálny počet funkcií príslušnosti, optimálne pokrytie vstupno-výstupného priestoru) boli použité funkcie z programového systému Matlab [2].

Na vytvorenie vlastného modelu boli použité funkcie *genfis2* a *anfis* umožňujúce vytvoriť optimálnu štruktúru fuzzy modelu. Funkcia *genfis2* využíva zhľukovú analýzu (Subtractive clustering), ktorá patrí k hierarchickým zhľukovacím metódam. Na základe vstupno – výstupných údajov z experimentov (vstupy a výstup fuzzy systému) vytvorí počiatočnú štruktúru fuzzy modelu (FIS), ktorá sa ďalej využije ako vstup pre funkciu *anfis*. *Genfis2* vyberie optimálnu sadu pravidiel, ktoré modelujú správanie systému. Zhľuková analýza sa využije na určenie počtu pravidiel a podmienkových funkcií príslušnosti a potom sa aplikuje metóda najmenších štvorcov na určenie optimálnych parametrov rovníc každého pravidla. Funkcia *anfis* dokončí tvorbu fuzzy modelu. Ako vstupy využíva tréningové dáta (vstupy a výstup z fuzzy systému) a počiatočnú štruktúru FIS (z *genfis2*). Podobné algoritmy pre určenie optimálnej štruktúry fuzzy modelov na základe I/O údajov sa využívajú aj pre praktické priemyselné aplikácie fuzzy algoritmov riadenia, kde optimálna štruktúra fuzzy modelov predstavuje dominantný problém.

Ukážka tvorby fuzzy modelu v Matlabe pre merané vstupno-výstupné údaje:

```
in_fis = genfis2(x, y, 0.5);
fis_T2 = anfis([x y],in_fis,100);
```

kde

in_fis – výstupná Takagi Sugeno fuzzy štruktúra z funkcie *genfis2*

fis_T2 – výstupná Takagi Sugeno štruktúra z funkcie *anfis*

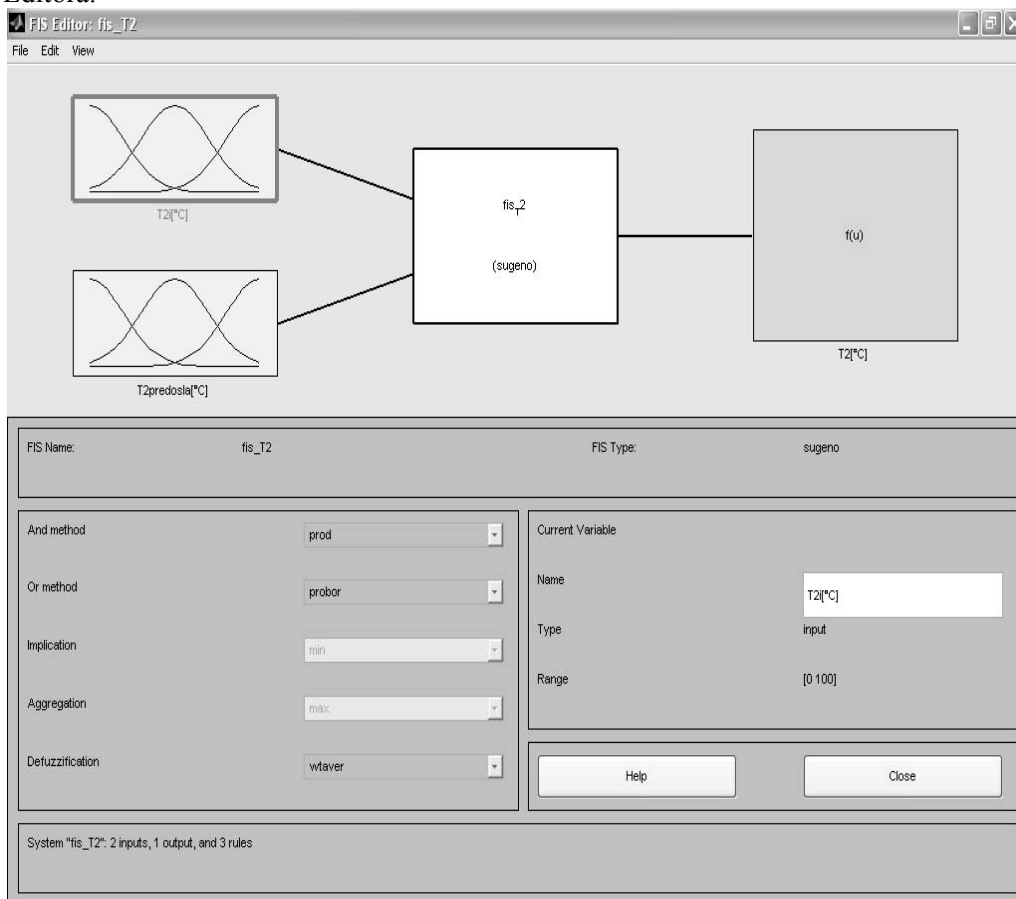
x – matica hodnôt vstupných premenných

y – vektor hodnôt výstupnej premennej

0.5 – radii – určuje rozsah zhlukov; radii $\in <0, 1>$

100 – počet epoch tréningu

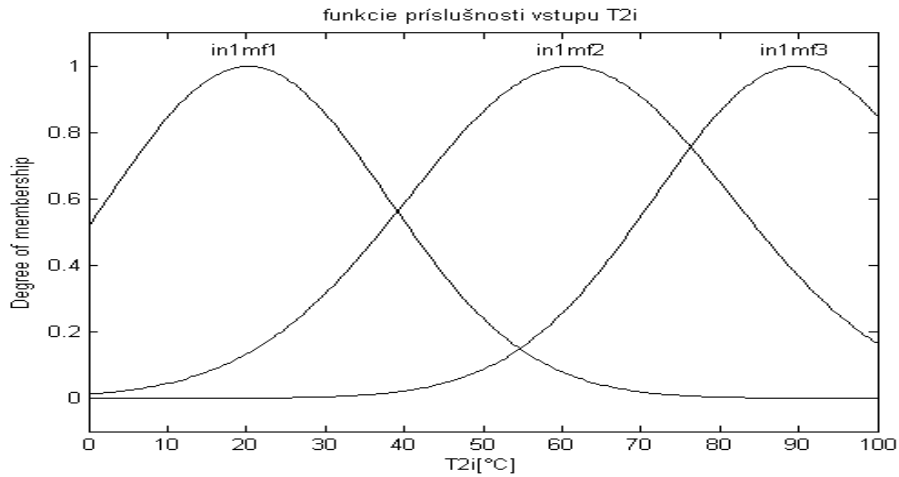
Výsledná štruktúra fuzzy modelu je realizovaná prostredníctvom FIS Editoru (obr. 30) v prostredí Matlab po zadaní príkazu „fuzzy“ a vložení vypočítanej optimálnej štruktúry fuzzy modelu „Load“ z položky „File“ v hlavnom menu FIS Editoru.



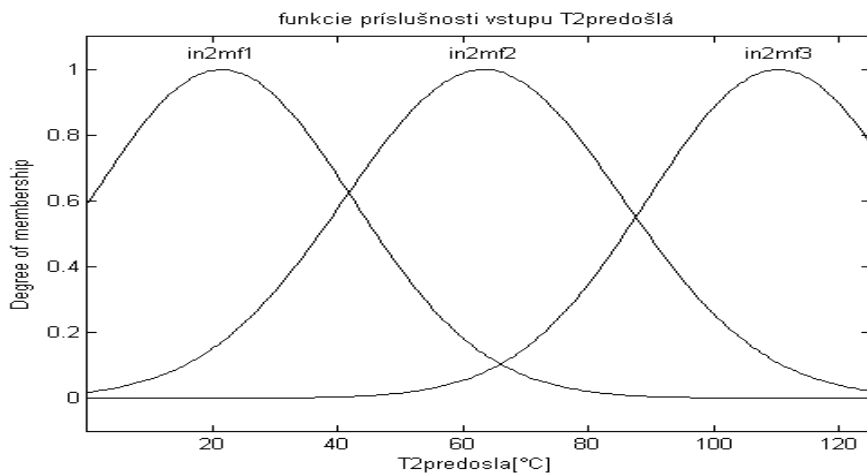
Obr. 30. FIS Editor pre testovanie fuzzy modelu

Na obr. 31 a obr. 32 sú zobrazené výsledné funkcie príslušnosti pre vstupy T_{2i} a $T_{2predosla}$. Na modelovanie systému boli použité tri pravidlá (3 funkcie príslušnosti). Všetky funkcie príslušnosti sú typu „Gauss“. Sú rozmiestnené viac menej rovnomerne

a rozsah premenných zodpovedá rozsahu trénovacích dát (T_{2i} : 0 až 100 °C, $T_{2\text{predošlé}}(T_2)$: 0 až 125 °C).



Obr. 31. Funkcie príslušnosti vstupu T_{2i}



Obr. 32. Funkcie príslušnosti vstupu $T_{2\text{predošlé}}$

Výstupom sú tri lineárne funkcie s nasledujúcimi parametrami:

| | | | |
|----------|----------|--------|----------|
| out1mf1: | [0,01983 | 0,98 | 0,00183] |
| out1mf2: | [0,01867 | 0,9812 | 0,02842] |
| out1mf3: | [0,02225 | 0,9854 | -0,3838] |

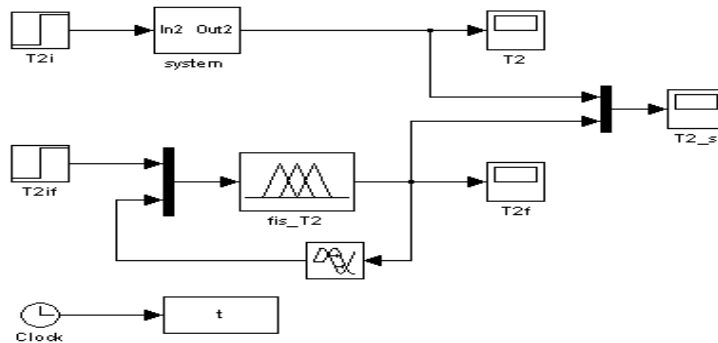
Fuzzy pravidlá:

```

if in1mf1 and in2mf1 then out1mf1
if in1mf2 and in2mf2 then out1mf2
if in1mf3 and in2mf3 then out1mf3

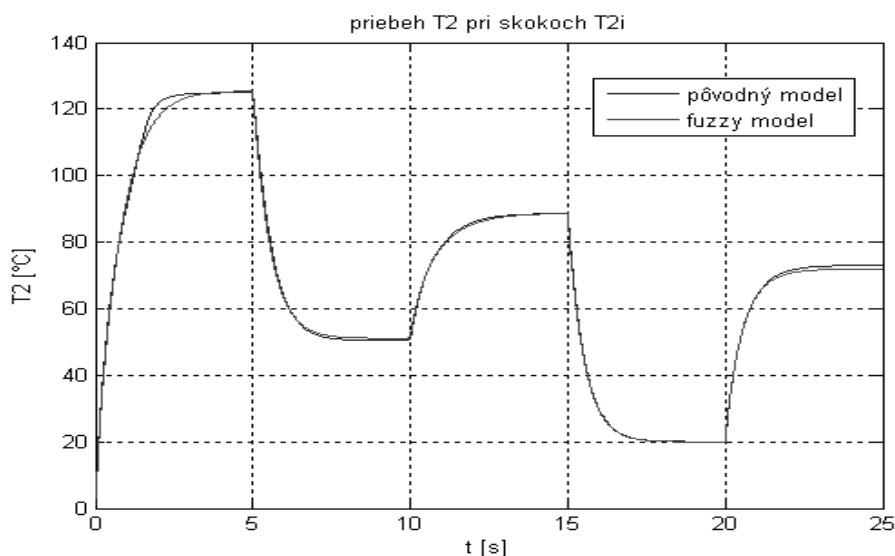
```

Overenie fuzzy modelu bolo realizované v prostredí Matlab-Simulink porovnaním s pôvodným nelineárnym modelom. Pre tento účel bola vytvorená schéma na obr. 33. FIS štruktúra (fis_T2) reprezentuje fuzzy model. Vstupmi sú hodnoty T_{2i} (T_{2if}) a predchádzajúce hodnoty výstupu T_2 (hodnoty z výstupu prechádzajú na vstup cez blok oneskorenia s oneskorením 1 perióda vzorkovania). Priebeh výstupnej veličiny T_2 je zobrazený v bloku T_{2f} . Pôvodný model reprezentuje blok systém. Vstupom sú hodnoty T_{2i} (T_{2i}) a priebeh výstupu T_2 je v bloku T2. Výsledné priebehy T_2 z oboch modelov sa dajú porovnať v bloku T_{2_s} .



Obr. 33. Bloková schéma simulácie pôvodného a fuzzy modelu

Pre simuláciu boli porovnávané priebehy výstupnej veličiny T_2 pre rôzne hodnoty skokových zmien vstupnej veličiny T_{2i}



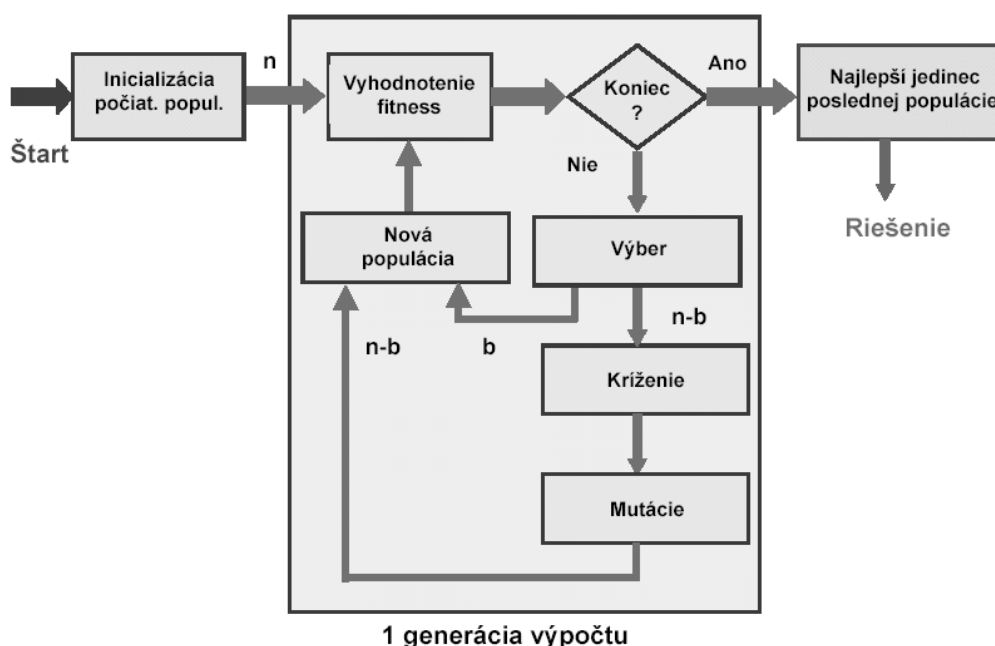
Obr. 34. Porovnanie pôvodného a fuzzy modelu pre rôzne skoky T2i

6 Aplikácia fuzzy logiky a genetických algoritmov

Genetické algoritmy sú súčasťou pokročilých metód výpočtovej inteligencie. Genetické algoritmy sú inšpirované známou teóriou evolúcie. Jednoducho povedané, problémy nájdenia najlepšieho riešenia sú evolučným procesom, s cieľom nájdenia najlepšieho riešenia. Proces prirodzenej evolúcie je jednoduchý, ale veľmi výkonný a robustný. Univerzálne platný pre jednoduché aj zložité organizmy. Neexistuje v ňom spätná väzba. GA sú schopné vyviaznuť z okolia lokálneho extrému a približovať sa ku globálnemu optimu. Uskutočňujú prehľadávanie vo viacerých smeroch súčasne. Nevyžadujú pomocné informácie o vývoji riešenia ako je napr. gradient účelovej funkcie. Požadujú len vyhodnotiť účelovú funkciu v každom bode priestoru. Sú schopné riešiť optimalizačné problémy so stovkami premenných, ale patria k časovo najnáročnejším metódam optimalizácie.

6.1 Genetický algoritmus

Algoritmus začína skupinou riešení (reprezentovanou chromozómami) nazývanou populácia. Riešenia z jednej populácie sú vybrané a používané na vytvorenie novej populácie. Toto je motivované tým, že nová populácia bude lepšia ako predošlá. Riešenia, ktoré sú neskôr vybrané na formovanie novej populácie sú vybrané podľa ich vhodnosti. Čím sú vyhovujúcejšie, tým majú väčšiu pravdepodobnosť reprodukcie. Toto sa opakuje až do splnenia nejakej ukončujúcej podmienky. Vývojový diagram GA je znázornený na obr. 35.



Obr. 35. Vývojový diagram genetického algoritmu

6.2 Návrh hybridného fuzzy regulátora + GA

V tejto časti kapitoly je demonštrované hybridné riadenie pomocou fuzzy logiky a genetických algoritmov. Uvažujeme opäť aplikáciu pre nelineárny model chemického reaktora opísaného nelineárnymi diferenciálnymi rovnicami:

$$\dot{x}_1(k) = -50x_1(k) - 10x_1^2(k) + u(10 - x_1(k))$$

$$\dot{x}_2(k) = 50x_1(k) - 100x_2(k) + u(-x_2(k))$$

$$y(k) = x_2(k)$$

Cieľom kombinovaného prístupu je aplikácia fuzzy logiky a GA pre optimalizáciu počtu a rozloženia funkcií príslušnosti tak, aby bola dosiahnutá čo najlepšia kvalita regulácie reprezentovaná zvolenou *fitness* funkciou.

Pre riadenie reaktora bol aplikovaný fuzzy PD+I regulátor s Mamdaniho metódou implikácie. Počet vstupov je 2 (regulačná odchýlka a derivácia regulačnej odchýlky). Výstupom je optimálny riadiaci zásah. Ako najlepší spôsob defuzzifikácie pre danú aplikáciu sa ukázala ťažisková metóda. Na základe modelovaných a simulovaných riešení boli zvolené rozsahy univerza pre vstupy a výstup. Pre riešenie bol vytvorený podporný programový systém, podľa ktorého si používateľ v interaktívnom režime môže modelovať a riadiť priebeh optimalizácie počtu funkcií príslušnosti, pokrytia univerza a nastavenia pravidiel.

GA upravuje prvotnú používateľskú voľbu parametrov funkcií príslušnosti (pozícia a šírka). Báza pravidiel sa na začiatku volí lineárna. Počet pravidiel je závislý

od počtu funkcií príslušností. GA nastaví každé pravidlo, ktoré môže nadobúdať hodnoty ľavého alebo pravého suseda.

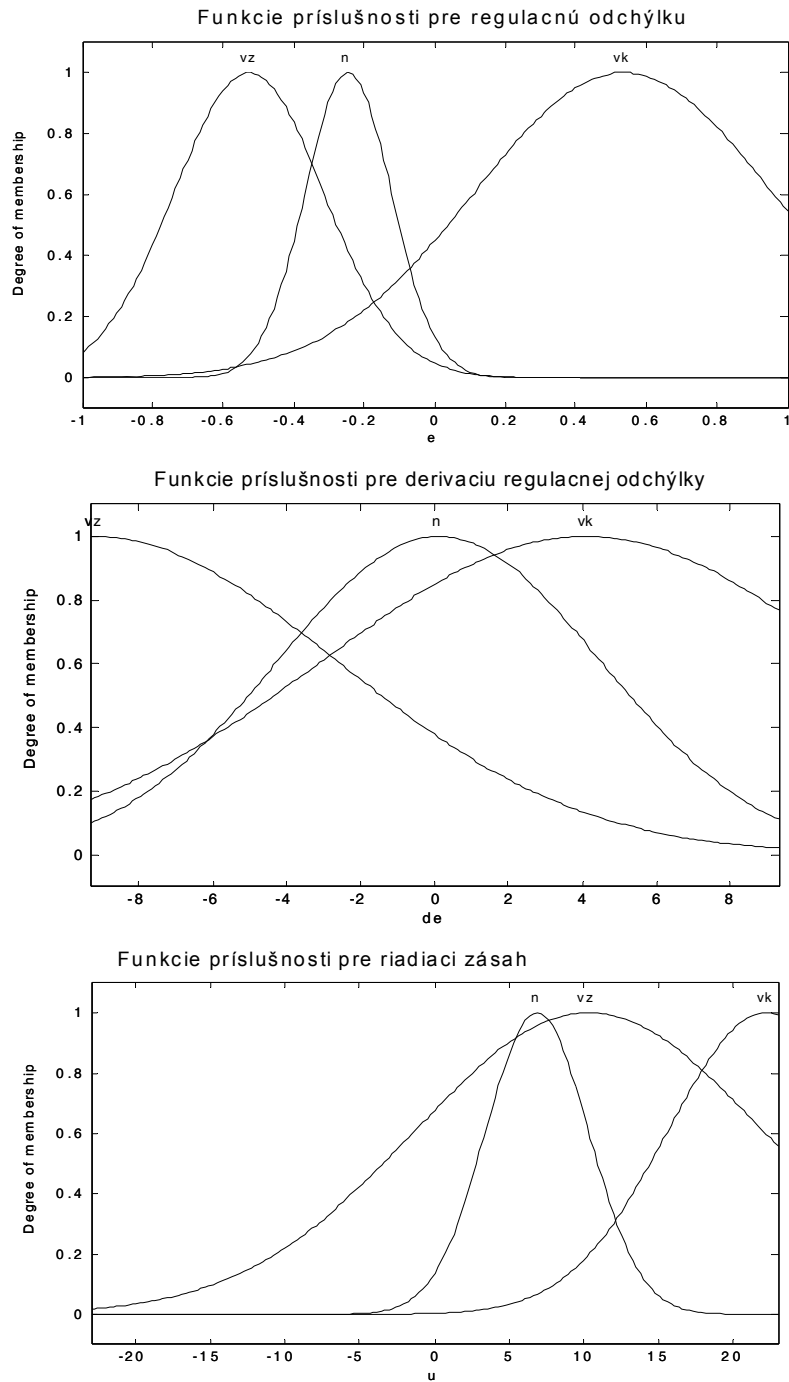
Zmena parametrov funkcií príslušnosti ako aj zmena parametrov bázy pravidiel sa realizuje súčasne.

- Pre riadenie nelineárneho systému boli zvolené tri funkcie príslušnosti (gaussove funkcie príslušnosti), 3 pre každý vstup a výstup. Rozsahy jednotlivých vstupov a výstupov sú nasledovné:

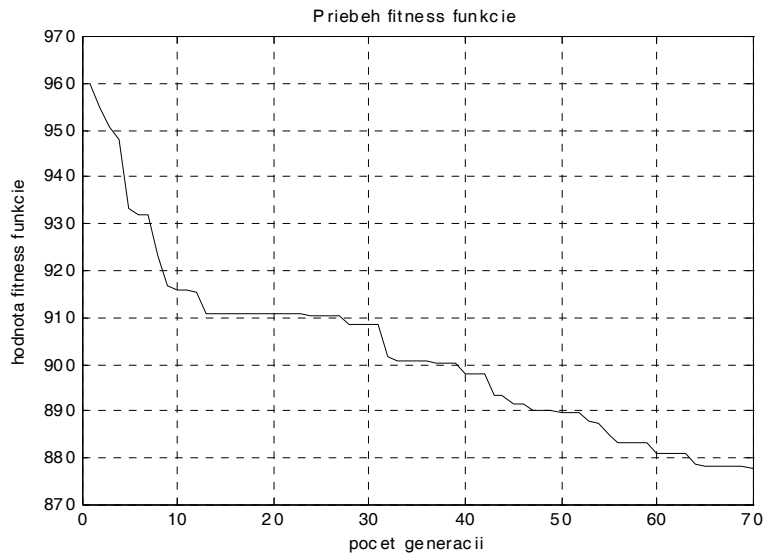
$e [-1; 1]$, $d_e [-10; 10]$, $u [-23; 23]$.

- Počet generácií bol zvolený 70 a veľkosť populácie bola 30, (čo sa neskôr ukázalo ako dostatočné). Po spustení genetického algoritmu sme dostali ďalej uvedené nastavenie fuzzy regulátora.
- Optimálne rozloženie funkcií príslušnosti pre vstup regulačnej odchýlky, deriváciu regulačnej odchýlky a riadiaceho zásahu je znázornené na obr. 36.
- Priebeh hodnôt fitness funkcie počas vykonávania GA je znázornený na obr. 37.
- Konečná báza pravidiel je zobrazená v Tab. 5.

Riešením je optimálna štruktúra fuzzy regulátora. Vzhľadom na náročné časové riešenie je vhodné realizovať testovanie a ladenie algoritmu dopredu off-line a po nájdení optimálneho riešenia sa algoritmus realizuje priamo v C-kóde alebo assembléri pre daný mikropočítač.



Obr. 36. Optimálne rozloženie funkcií príslušnosti (regulačnej odchýlky, derivácie regulačnej odchýlky a riadiaceho zásahu) po optimalizácii a nastavení GA



Obr. 37. Pribeh hodnôt fitness funkcie pri optimalizácii syntézy fuzzy regulátora

Optimálna báza pravidiel po optimalizácii je uvedená v tabuľke 5

Tab. 5. Optimálna báza pravidiel

| | e | | | | |
|----|----|----|----|----|----|
| de | VZ | SZ | SZ | SZ | N |
| | VZ | SZ | SZ | N | SK |
| | SZ | N | N | SK | SK |
| | SZ | N | SK | SK | SK |
| | N | SK | SK | VK | VK |

6.3 Aplikácia GA pre optimálne pokrytie univerza

Pre optimálne pokrytie univerza bol aplikovaný Toolbox GA v prostredí Matlab-Simulink [11]. Z toolboxu boli aplikované nasledovné funkcie: *vygenerovanie náhodnej populácie*, *výber najlepších reťazcov*, *turnajový výber*, *ruletový výber*, *náhodný výber reťazcov*, *viacbodové križenie rodičov*, *obyčajná mutácia*, *aditívna mutácia*, ktoré sú pre danú aplikáciu potrebné. GA cyklicky menil tvar, rozloženie funkcií príslušnosti a bázu pravidiel.

Pre optimalizáciu nastavenia bola aplikovaná fitness (kriteriálna funkcia) v tvare:

$$\text{Fit}(h)=a*\text{sum}(\text{abs}(e))+b*\text{sum}(\text{abs}(dy))+c*\text{sum}(\text{abs}(du))$$

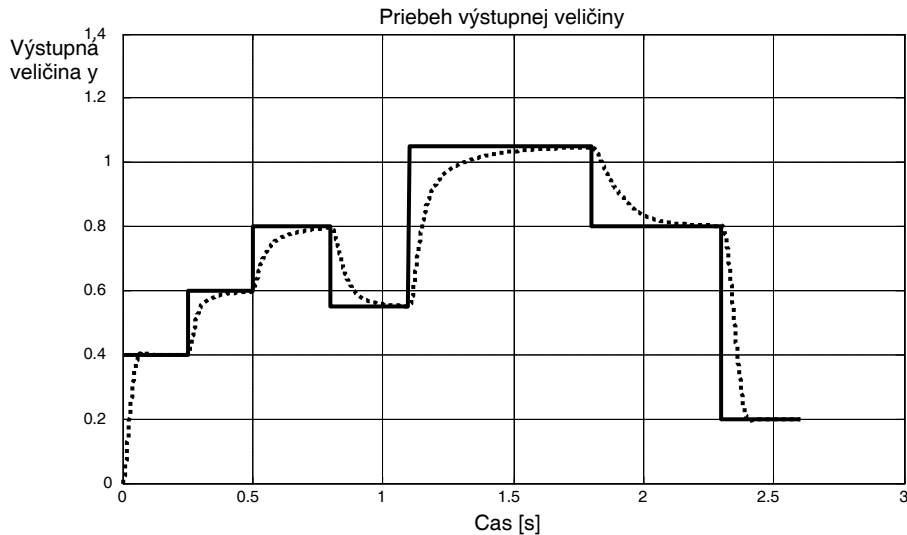
pričom koeficienty a , b , c závisia aj od typu riadeného procesu. (Symbol e reprezentuje regulačnú odchýlku, pomocou ktorej sme minimalizovali dobu regulácie. dy je derivácia výstupu z nelineárneho systému, pomocou ktorého bola realizovaná minimalizácia kmitov regulovanej veličiny, du je derivácia riadiaceho zásahu, pomocou ktorého boli minimalizované oscilácie riadiaceho zásahu.

Pre zovšeobecnenie prístupu bolo vytvorené grafické menu umožňujúce aplikovať navrhovanú metodiku pre široké spektrum riadenia nelineárnych procesov hybridnými technikami využívajúcimi GA a fuzzy logiku. Na obr. 37 je znázornené interaktívne pracovné menu, ktoré ponúka používateľovi na výber *zmenu parametrov - počet generácií, rozsahy vstupov a výstupov, čas simulácie, periódu vzorkovania a váhy jednotlivých koeficientov fitness funkcie*

Obr. 38. Interaktívny návrh hybridného fuzzy+GA regulátora

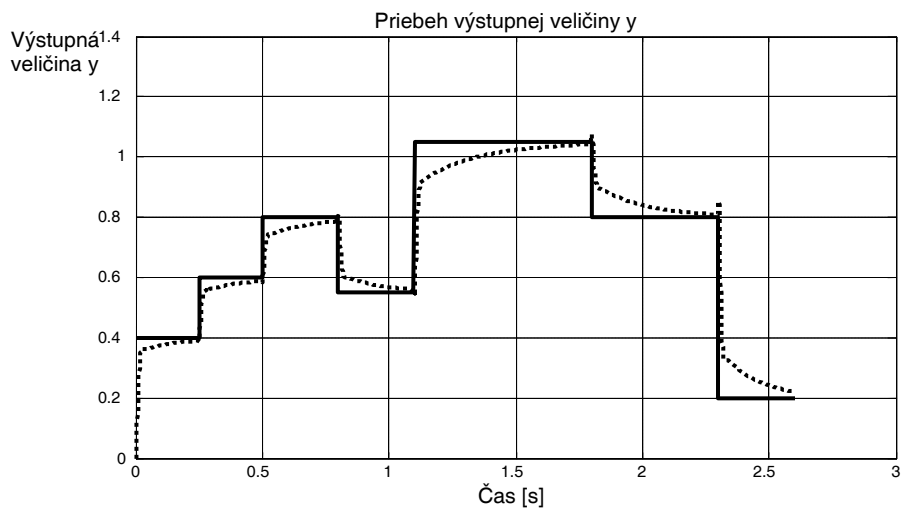
Položka návrh regulátora spúšťa GA a tým aj nastavovanie parametrov fuzzy regulátora. Po skončení GA sa zobrazí časová odozva výstupnej veličiny, referenčnej

veľičiny a akčného zásahu pre navrhnutý fuzzy regulátor + GA a priebeh hodnôt fitness funkcie. Položka zobraz regulátor zobrazí navrhnutý regulátor ako fis štruktúru. Užívateľ si môže z grafického menu vybrať z viacerých typov nelineárnych procesov.



Obr. 39. Časový priebeh výstupnej veličiny pri aplikácii hybridného fuzzy regulátora a GA

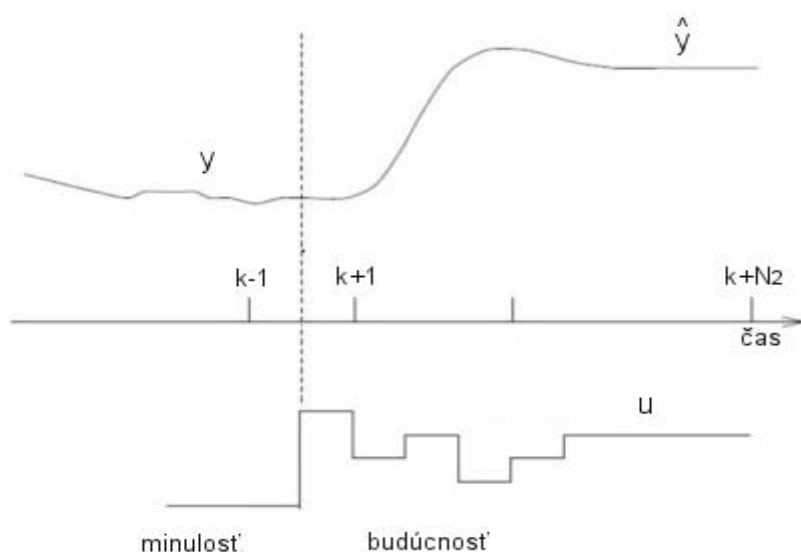
Z priebehu výstupnej veličiny je dobre vidieť dobrú kvalitu regulácie pri zmene referenčnej hodnoty. Fuzzy PD+I zabezpečuje minimálne prerogulovanie, a stabilizuje proces aj vo vzdialenejších oblastiach od pracovného bodu. Na rozdiel od toho PID regulátor navrhnutý pomocou GA už nevykazoval také uspokojivé výsledky ako Fuzzy PD+I ako môžeme vidieť na obr. 40.



Obr. 40. Časová odozva výstupnej veličiny PID regulátora ktorého parametre sú nastavené GA

7 Pokročilé formy riadenia na báze fuzzy logiky

Prediktívne algoritmy riadenia patria do skupiny moderných metód riadenia. Pri riadení sa využíva model procesu na stanovenie optimálneho riadiaceho zásahu. Jedným zo základných parametrov prediktívneho regulátora je *horizont predikcie* N_2 , ktorý určuje, o koľko krokov dopredu algoritmus riadenia umožňuje optimalizovať svoju činnosť. Graficky je priebeh regulácie pomocou prediktívneho regulátora znázornený na obr. 41.



Obr. 41. Regulácia prediktívnym regulátorom.

Obr. 41 zobrazuje priebeh regulácie SISO systému, pričom k je aktuálny diskretný časový okamih a nasledujúce vzorky ($k+1, \dots$) sú oproti k posunuté vždy o ďalšiu periódu vzorkovania. Pomocou horizontu predikcie je možné definovať nasledujúce vektory:

$$\begin{aligned} u &= [u(k), u(k+1), \dots, u(k+N_2-1)]^T \\ \hat{y} &= [\hat{y}(k+1), \hat{y}(k+2), \dots, \hat{y}(k+N_2)]^T \\ w &= [w(k+1), w(k+2), \dots, w(k+N_2)]^T \end{aligned} \quad (7)$$

Vektor u zohľadňuje budúce riadiace zásahy, vektor y budúce predpokladané výstupy procesu, získané z modelu a w je vektor nasledujúcich referenčných hodnôt. Prvá hodnota optimalizovaného vektora u sa použije ako skutočný realizovaný riadiaci zásah.

Na výpočet optimálneho prediktívneho riadiaceho zásahu, ktorý bude aktuálne vstupovať do systému, sa využíva vhodne zvolená *kritériálna funkcia*, ktorá ohodnocuje jednotlivé riešenia.

$$J = \sum_{i=N_1}^{N_2} (\hat{y}(k+i) - w(k+i))^2 + \lambda_u \sum_{i=1}^{N_u} u(k+i-1)^2 \quad (8)$$

Minimalizácia akčného zásahu sa zamení za minimalizáciu zmeny akčného zásahu. Takýto tvar kriteriálnej funkcie vyhladí priebeh akčného zásahu a v ustálenom stave, pri nulovej regulačnej odchýlke, ho bude tlačiť na konštantnú hodnotu.

Koeficient lambda slúži na vyváženie vplyvov regulačnej odchýlky a zmeny akčného zásahu na hodnotu kriteriálnej funkcie. Parameter N_u sa nazýva ovládaci horizont (control horizon) – určuje, koľko nasledujúcich zmien akčných zásahov sa berie do úvahy pri prediktívnej regulácii. Hodnota N_u sa volí tak, aby platilo: $1 \leq N_u \leq N_2$

Na tvorbu kriteriálnych funkcií je možné využiť aj ďalšie hodnoty, získané analýzou kvality regulácie. Sú to napríklad:

- veľkosť prekmitu
- počet prekmítov
- tlmenie kmitania
- doba nábehu
- doba ustálenia

V aplikáciách sa však používajú zriedka, kvôli zložitejšiemu výpočtu, oproti vyššie spomínaným funkciám. Taktiež vzťah medzi parametrami regulátora a týmito hodnotami je veľmi nelineárny, čo komplikuje optimalizáciu.

Základnou požiadavkou predikčného modelu, je čo najpresnejšie predpovedať budúce stavy regulovaného systému. Medzi najbežnejšie modely podľa patria:

- Auto-Regressive eXogenous (ARX)
- Auto-Regressive Moving Average eXogenous (ARMAX)
- Auto-Regressive Integrated Moving-Average eXogenous (ARIMAX)
- Finite Impulse Response (FIR)
- Finite Step Response (FSR)

Vo všeobecnosti je predikčný model vyjadrený diferenčnou rovnicou v tvare :

$$y(k) = \frac{q^{-d} B(q^{-1})}{A(q^{-1})} u(k-1) + \underbrace{\frac{C(q^{-1})}{D(q^{-1})A(q^{-1})}}_{\text{model šumu}} e(k) \quad (9)$$

kde d je oneskorenie a výstup v čase k závisí od vybudenia akčným zásahom a šumu.

Polynómy A , B , C a D sú diskkrétne polynómy v z -oblasti v tvare:

$$\begin{aligned} A(q^{-1}) &= 1 + a_1 q^{-1} + \dots + a_{n_A} q^{-n_A} \\ B(q^{-1}) &= b_0 + b_1 q^{-1} + \dots + b_{n_B} q^{-n_B} \\ C(q^{-1}) &= 1 + c_1 q^{-1} + \dots + c_{n_C} q^{-n_C} \\ D(q^{-1}) &= 1 + d_1 q^{-1} + \dots + d_{n_D} q^{-n_D} \end{aligned} \quad (10)$$

pričom $n_A n_B n_C n_D$ sú rády príslušných polynómov. Špecifikom pre modely FIR a FSR je, že neuvažujú model šumu. Modely ARX, ARMAX, ARIMAX a iné im blízke modely sa líšia len rozdielmi v modelovaní šumu. Medzi jednu z hlavných výhod prediktívneho riadenia možno považovať zakomponovanie ohraničujúcich podmienok priamo do funkcionálu. Môže sa jednať o ohraničenie akčnej veličiny, zmeny akčnej veličiny alebo aj veľkosti riadenej veličiny. Pri prediktívnom riadení sa zvyknú používať niektoré z nasledujúcich ohraničení:

$$\begin{aligned} U_{\min} < u < U_{\max} \\ \Delta U_{\min} < \Delta u < \Delta U_{\max} \\ Y_{\min} < y < Y_{\max} \end{aligned} \quad (11)$$

Na minimalizáciu niektorého z uvedených kritérií sa využíva kvadratické programovanie, do ktorého je možné zakomponovať ľubovoľné ohraničenie, či už na zmenu alebo absolútnu veľkosť ľubovoľnej premennej.

Okrem vyššie spomínaných najčastejšie používaných modelov je možné aplikovať aj iné, menej známe modely. Medzi ne patria napríklad modely tvorené natrénovanými neurónovými sieťami, fuzzy modely, a ďalšie. Môže sa jednať prakticky o akýkoľvek model, ktorý je schopný predpovedať budúce stavy modelovaného systému. Takéto alternatívne modely sa využívajú v prípadoch, keď je proces vysoko nelineárny a štandardné modely neposkytujú uspokojivé výsledky, respektíve vôbec nie sú schopné modelovať takéto systémy.

7.1 Generalized predictive control (GPC) prediktívne metódy s využitím metód fuzzy logiky

Pre návrh prediktívneho riadenia využívame ako matematický model Takagi-Sugeno typ modelu

$$\begin{aligned} R^i : \quad & \text{Ak } x_1 \text{ je } A_1^i \text{ a } \dots \text{ a } x_m \text{ je } A_m^i \\ & \text{potom } y^i = p_0^i + p_1^i + \dots + p_m^i \quad i = 1, 2, \dots, n \end{aligned} \quad (12)$$

kde R^i je i -te fuzzy pravidlo, A_j^i predstavuje funkciu príslušnosti (Gaussovu), ktorá je definovaná na danom univerze, x_i je i -ta vstupná premenná, m je počet vstupných premenných a n je počet pravidiel:

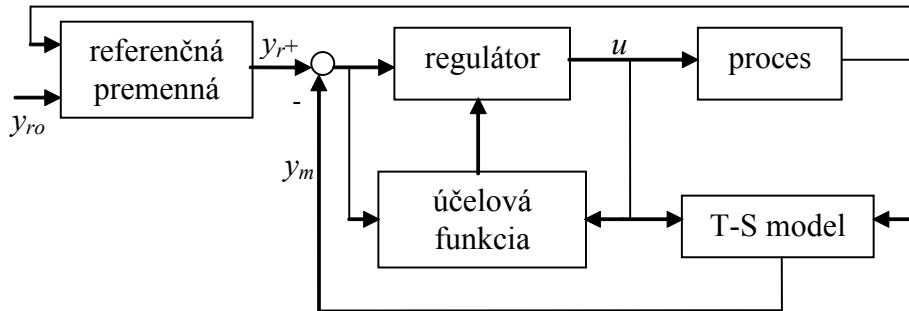
$$A_{ij}(x_j) = \exp \left[\frac{-(x_j - v_{ij})^2}{2\delta_{ij}^2} \right] \quad (13)$$

kde v_{ij} sú centrá (zhlukov), δ_{ij}^2 šírky funkcie príslušnosti. Algoritmus môže nájsť maticu $U = [\mu_{ij}]_{c \times n}$ a centrá klastrov $V = [v_1, v_2, \dots, v_c]$ minimalizáciou účelovej funkcie

$$J(Z, \Lambda, V) = \sum_{i=1}^c \sum_{j=1}^N (\mu_{ij})^m d^2(z_j, v_i) \quad (14)$$

z_j je množina vzoriek $z = [z_1, z_2, \dots, z_N]$, $d(z_j, v_i)$ je vzdialenosť vektora z_j od prototypu klastra v_i .

Štruktúra fuzzy prediktívneho regulátora s využitím T-S modelu je na obr. 42.



Obr. 42. Bloková schéma fuzzy prediktívneho regulátora

T-S model môže byť reprezentovaný lineárnou formou CARIMA modelu

$$A(z^{-1})y(k) = B(z^{-1})u(k-1) + \xi(k) / \Delta \quad (15)$$

kde $\Delta=1-z^{-1}$, $y(k)$ a $u(k)$ sú vstupy a výstupy systému, $\xi(k)$ predstavuje poruchu (biely šum). $A(z^{-1})$, $B(z^{-1})$ sú polynómy opísané ako

$$\begin{aligned} A(z^{-1}) &= 1 + a_1 z^{-1} + \dots + a_{n_a} z^{-n_a} \\ B(z^{-1}) &= b_0 + b_1 z^{-1} + \dots + b_{n_b} z^{-n_b} \end{aligned} \quad (16)$$

Kriteriálna funkcia pre prediktívne riadenie s TS modelom je v tvare

$$J = \sum_{j=N_1}^{N_2} [\hat{y}(k+j) - w(k+j)]^2 + \sum_{j=1}^{N_u} \lambda [\Delta u(k+j-1)]^2 \quad (17)$$

kde $\hat{y}(k+j)$ je predikovaný výstup j -krokov dopredu, $w(k+j)$ je referenčná premenná, N_1 a N_2 je minimálny a maximálny horizont predikcie, N_u je horizont riadenia a λ je váhový koeficient. Predikcia j -krokov dopredu môže byť získaná ako

$$\hat{y}(k+j) = G_j(z^{-1})\Delta u(k+j-1) + F_j(z^{-1})y(k) \quad (18)$$

kde $G_j(z^{-1}) = E_j(z^{-1})B(z^{-1})$, $E_j(z^{-1})$ a $F_j(z^{-1})$ sú polynómy získané z diofantickej rovnice

$$1 = E_j(z^{-1})\Delta A(z^{-1}) + z^{-j}F_j(z^{-1}) \quad (19)$$

kde $\deg(E_j)=j-1$, $\deg(F_j)=n_a$. Výraz (18) môžeme napísať ako

$$\hat{y} = G\tilde{u} + f \quad (20)$$

$$\text{kde } \hat{y} = [\hat{y}(k+1), \dots, \hat{y}(k+N)]'$$

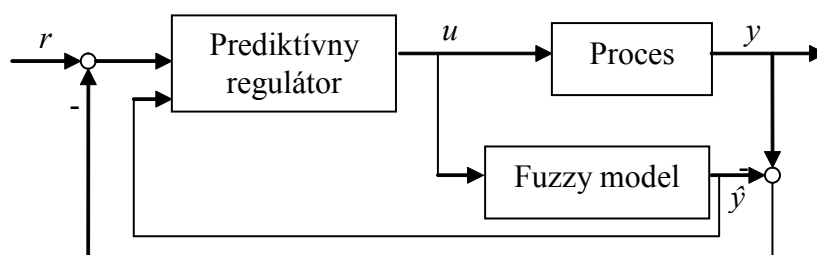
$$\tilde{u} = [\Delta u(k), \dots, \Delta u(k+N-1)]'$$

Označením $w = [w(k+1), \dots, w(k+N)]'$ a minimalizáciou dostaneme optimálne riešenie

$$\tilde{u} = (G^T G + \lambda I)^{-1} G^T (w - f) \quad (21)$$

7.2 Fuzzy prediktívne riadenie s využitím metódy vnútorného modelu (IMC)

Na obr. 43 je uvedená štruktúra fuzzy prediktívneho riadenia s využitím IMC štruktúry (Internal Model Control- Metóda vnútorného modelu). Nelineárny proces je opísaný fuzzy model, ktorý predikuje budúce sa správanie procesu. Pretože sú fuzzy modely väčšinou nelineárne, v každej perióde vzorkovania musí byť riešený nekonvexný optimalizačný problém. Existujúce iteratívne metódy pre nekonvexnú optimalizáciu často konvergujú k lokálnym riešeniam, čo obmedzuje praktické využitie tejto metódy [3]. Aby sa obišli tieto optimalizačné problémy, môžeme znovu aplikovať genetické algoritmy alebo diskretnú optimalizáciu založenú na metóde vetiev a hraníc (branch-and-bound method). Výhodou metódy je, že vždy nájde globálne optimum pre riadiaci zásah a vie zohľadniť aj ohraničenia na vstupné a výstupné premenné.



Obr. 43. Implementácia fuzzy prediktívneho riadenia s využitím IMC štruktúry

Sú tu riešené dve úlohy: optimalizácia účelovej funkcie za účelom určenia optimálneho akčného zásahu a explicitná predikcia výstupu z procesu na horizonte predikcie.

Najčastejšou používanou formuláciou je minimalizácia kvadratickej účelovej funkcie podľa už uvedeného CARIMA modelu:

$$J = \xi \left\{ \sum_{i=1}^p (r(k+i) - \hat{y}(k+1))^2 + \sum_{i=1}^m \alpha(i) [\Delta u(k+i-1)] / F(k) \right\} \quad (22)$$

kde p je horizont predikcie, r je referenčný signál, \hat{y} je predikovaný výstup z modelu, Δu zmena vstupnej veličiny, α je konštantný váhový faktor penalizujúci budúcu zmenu akčného zásahu na horizonte riadenia m , ξ označuje strednú hodnotu.

Budúce výstupy z fuzzy modelu na horizonte predikcie sú:

R_i : if $y(k)$ is A_{i1} and...and $y(k-n_y+1)$ is A_{iny} and
if $u(k)$ is B_{i2} and...and $u(k-n_u+1)$ is B_{inu} then

$$y(k+1) = \sum_{j=1}^{n_y} a_{ij} y(k-j+1) + \sum_{j=1}^{n_u} b_{ij} u(k-j+1) + c_i, \quad i=1, \dots, K \quad (23)$$

kde A_{il} , B_{il} sú fuzzy množiny a a_{ij} , b_{ij} a c_i sú parametre konsekventu. Váňovaný predikovaný výstup $y(k+1)$ modelu je:

$$y(k+1) = \frac{\sum_{i=1}^K \mu_i(y(k), \dots, u(k-n_u+1)) y_i(k+1)}{\sum_{i=1}^K \mu_i(y(k), \dots, u(k-n_u+1))} \quad (24)$$

kde μ_i je funkcia príslušností daná ako:

$$\mu_i(y(k), \dots, u(k-n_u+1)) = \mu_{A_{i1}}(y(k)) \dots \mu_{A_{iny}}(y(k)) \mu_{B_{i2}}(u(k-1)) \dots \mu_{B_{inu-1}}(u(k-n_u+1)) \quad (25)$$

Keď antecedent v neobsahuje vstup $u(k)$, model výstupu $y(k+1)$ je afinný k vstupu $u(k)$. V tejto súvislosti označujeme normalizovanú funkciu príslušností ako:

$$\mu_i(y(k), \dots, u(k-n_u+1)) = \frac{\mu_i(y(k), \dots, u(k-n_u+1))}{\sum_{i=1}^K \mu_i(y(k), \dots, u(k-n_u+1))} \quad (26)$$

Substitúciou (26) a (25) do (25):

$$y(k+1) = \sum_{i=1}^K \lambda_i(y(k), \dots, u(k-n_u+1)) \left[\sum_{j=1}^{n_y} a_{ij} y(k-j+1) + \sum_{j=2}^{n_u} b_{ij} u(k-j+1) + c_i \right] + \sum_{i=1}^K \lambda_i(y(k), \dots, u(k-n_u+1)) b_{i1} u(k) \quad (27)$$

Nelineárna afinná forma výstupu z fuzzy modelu:

$$y(k+1) = g(y(k), \dots, y(k-n_y+1), u(k-1), \dots, u(k-n_u+1)) + h(y(k), \dots, y(k-n_y+1), u(k-1), \dots, u(k-n_u+1)) u(k) \quad (28)$$

Pri predpoklade, že v kroku $(k+1)$ je model výstupu rovný referenčnej premennej $y(k+1)=w(k+1)$, riadiaci zásah je možno vyjadriť zo vzťahu:

$$u(k) = \frac{w(k+1) - g(y(k), \dots, y(k-n_y+1), u(k-1), \dots, u(k-n_u+1))}{h(y(k), \dots, y(k-n_y+1), u(k-1), \dots, u(k-n_u+1))} \quad (29)$$

S uvažovaním (28) získame výslednú formu riadiaceho zásahu:

$$u(k) = \frac{w(k+1) - \sum_{i=1}^K \mu_i(y(k), \dots, u(k-n_u+1)) \sum_{j=1}^{n_y} a_{ij} y(k-j+1) + \sum_{j=2}^{n_u} b_{ij} u(k-j+1) + c_i}{\sum_{i=1}^K \mu_i(y(k), \dots, u(k-n_u+1)) b_{i1}}$$

7.3 Prediktívne fuzzy riadenie nelineárneho systému – izotermického regulátora

Uvažujme už riešený príklad izotermického reaktora, (Van Vusseov reaktor), reakcia (podľa kinetickej schémy):



kde A je výťažok, B je požadovaný výrobok, C a D sú nežiaduce vedľajšie výrobky.

Pri navrhovanom modeli sú odhadované koeficienty k_2 a k_3 malé v porovnaní s k_1 pri príslušnom výbere katalyzátorových podmienok. Koncentrácia B vo výrobku môže byť riadená prostredníctvom vstupnej rýchlosti prietoku a /alebo pôsobením teploty. Výťažok prietoku obsahuje len cyclopentadién malej koncentrácie C_{Af} . Predpokladáme konštantnú hustotu a ideálny čas delenia vnútri reaktora, porovnávaní rovnováhy masy pre príslušné koncentrácie cyclopentadiénu a požadovaných výrobkov C_A a C_B , prostredníctvom analytického modelu:

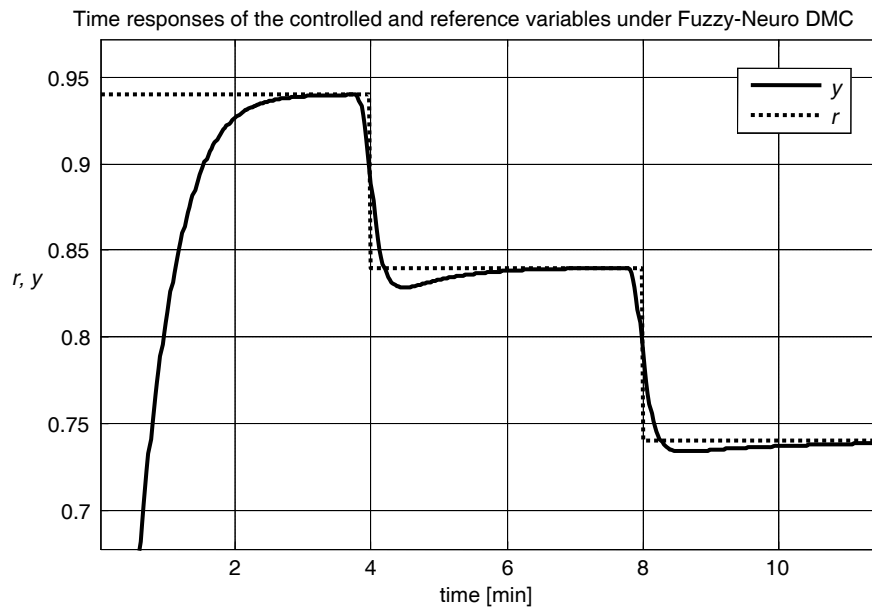
$$\begin{aligned} \dot{C}_A &= -k_1 C_A - k_3 C_A^2 + \frac{F}{V} (C_{Af} - C_A) \\ \dot{C}_B &= k_1 C_A - k_2 C_B - \frac{F}{V} C_B \\ y &= C_B \end{aligned} \quad (31)$$

Normalizáciou procesných premenných v okolí pracovného bodu a substitúciou hodnôt fyzikálnych konštánt modelu procesu dostaneme výsledný model:

$$\begin{aligned} \dot{x}_1(t) &= -50x_1(t) - 10x_1^2(t) + u(10 - x_1(t)) \\ \dot{x}_2(t) &= 50x_1(t) - 100x_2(t) + u(-x_2(t)) \\ y(t) &= x_2(t) \end{aligned} \quad (32)$$

kde koncentrácia komponentu A je označená x_1 , koncentrácia komponentu B je x_2 a vstupná rýchlosť prietoku je u .

Pre uvedený analytický model reaktora boli testované a overené tri typy fuzzy prediktívnych algoritmov. Pre zovšeobecnenie prediktívnych algoritmov bol navrhnutý programový systém umožňujúci modelovať, riadiť a simulovať vybrané typy nelineárnych modelov s návrhom troch typov prediktívnych algoritmov (DMC—Dynamic Matrix Control, GPC Generalized Predictive Control a IMC predictive Control [9,10]. Na obr. 44 je zobrazený výsledný priebeh výstupnej regulovanej veličiny pri zmene referenčnej premennej.



Obr. 44. Časový priebeh výstupnej regulovanej veličiny pri aplikácii fuzzy prediktívneho algoritmu riadenia.

8 Záver

Kapitola *Aplikovaná fuzzy logika* predstavuje opis vybraných aplikácií fuzzy logiky v rôznych oblastiach. Simulačné, modelovacie a riadiace algoritmy prezentované v tejto časti boli zväčša úspešne realizované a testované v prostredí Matlab- Simulink a prostredníctvom mikropočítačových systémov Freescale. Do praxe je zavádzaných stále viac týchto jednoduchých ale vysoko efektívnych metód a v najbližších rokoch sa dá očakávať prienik aj do ďalších oblastí ako je zdravotníctvo, ekonómia, biotechnológie a mechatronické systémy. V súlade s rozvojom nových algoritmov numerickej matematiky a optimalizačných algoritmov možno očakávať čoraz väčšie množstvo aplikácií fuzzy logiky s využitím princípov prediktívneho a robustného riadenia.

Pod'akovanie: Táto časť publikácie vznikla s podporou grantu VEGA 1/0822/08 a projektu APVV č. VMSP-P-0059-09.

Literatúra

- [1] Altus, M., Bělohávek R.: *Relax 2005*, Grant IGA MZ ČR ND7665-3/2003 Ministerstva zdravotnictví ČR.
- [2] Dokumentácia k programu MATLAB 7.2
- [3] Hudzovič P.: *Optimalizácia*. Vydavateľstvo STU, 2000, Bratislava
- [4] Lazarová, E., Ušalová, E. : *Využitie fuzzy logic Toolboxu v geotechnike*, Ústav geotechniky, SAV Košice, 2001
- [5] Leondes C. T.: *Fuzzy Logic and Expert Systems Applications*. Academic Press, San Diego, 1998
- [6] MacVicar-Whelan, P. J.: Fuzzy sets for man-machine interactions. *Int. J. Man-Machine Studies* 8 (1977) 687–697
- [7] Misak, P. Vymazal, T.: *Výpočet odhadu pevnostních charakteristik betónu pomocou fuzzy inference*, Fakulta stavební, Ústav stavebního zkušebnictví, v Brně 2008
- [8] Novák V.: *Základy fuzzy modelování*. BEN – technická literatura, 2000, Praha
- [9] Paulusová, J., Kozák, Š.: Predictive Fuzzy Control Application. *2nd International Conference on ADVANCED CONTROL CIRCUITS & SYSTEMS (ACCS'08)*, March 30 - April 2, 2008, Cairo – Egypt, 21_NFG122.pdf.
- [10] Paulusová, J., Kozák, Š.: Robust Predictive Fuzzy Control. *7th Portuguese Conference on Automatic Control, CONTROLO'2006*, Lisbon, Portugal: 11-13 September 2006, MA-7-1.
- [11] Sekaj, I., Foltin, M.: Matlab toolbox - genetické algoritmy. 11. ročník konference MATLAB 2003. Praha, Česká republika: 25.11.2003, s. 514-519.
- [12] Soeterboek R.: *Predictive Control - A Unified Approach*. Prentice Hall, 1992.
- [13] Spalek, J., Janota, A. Efektívne a bezpečné riadenie dopravy v tuneli. In: *Tunel*. - ISSN 1211-0728. - Roč. 13, č.1 (2004), s. 40-42 .
- [14] Stehlíková, B., Gažo, J.: *Použitie fuzzy množín pri konštrukcii hraníc deskriptorov*. Genetické zdroje rastlín, Nitra: SPU, 1999, s. 55-59 ISBN 80-7137-668.
- [15] Vítečková, M.: *Seřízení regulátorů metodou inverze dynamiky*. 1. vyd., dotisk vr. 2000. Ostrava: skripta FS VŠB - TU Ostrava, 1998.
- [16] Zadeh L. A.: *Fuzzy Logic, Neural Networks, and Soft Computing*. 1972
- [17] Záhorová, V. Určení nástupištní koleje pro zpožděný přijíždějící vlak pomocí fuzzy regulátoru. In sborník příspěvků konference *Intelligentní systémy pro praxi: XI ročník. konference*, Bohdaneč, 30.-31. ledna 2008

Technika sémantických tabiel v logike

Vladimír KVASNIČKA a Jiří POSPÍCHAL¹

Abstrakt. Technika sémantických tabiel patrí tak v klasickej logike, ako aj v neklasických logikách k základným technikám modernej matematickej logiky; tvorí efektívny prostriedok k štúdiu sémantických vlastností rôznych logík (menovite k špecifikácii relácie sémantického vyplývania \models). Taktiež možno ukázať, že sémantické tablá interpretované v inverznom poradí možno chápať ako prirodzenú dedukciu.

1 Úvodné poznámky

V našej každodennej skúsenosti pojem „tablo“ má silne technický akcent, v tomto texte budeme pod ním rozumieť obrázok alebo schému. V matematickej logike metóda sémantického tabla [5,12,15,16] je formálnou procedúrou dôkazu, ktorá, aj keď má v rôznych logikách mnoho spoločných črt, má v týchto logikách vždy aj špecifické neopakovateľné črty.

Prvá dôležitá vlastnosť sémantických tabiel je, že nám slúži ako efektívna metóda nepriameho dôkazu toho, či formula φ je tautológia. Ak sa nám podarí dokázať pomocou sémantických tabiel, že negácia $\neg\varphi$ je kontradikcia, potom pôvodná formula φ musí byť tautológia. Táto technika je založená na diagramatickej interpretácii transformácie logickej formuly do tvaru disjunktívnej normálnej formy (DNF). Táto transformácia je znázornená pomocou sémantických tabiel. Môžeme teda konštatovať, že sémantické tablá znázorňujú transformáciu formuly do tvaru DNF, ktorý je mimoriadne vhodný na verifikovanie skutočnosti, či daná formula je kontradikcia (platí jednoduchá veta, že DNF formula je kontradikcia práve vtedy, ak jej každá konjunktívna klauzula obsahuje dvojicu komplementárnych literálov, napr. p a $\neg p$).

Druhá významná vlastnosť sémantických tabiel je, že pomocou nich môžeme pomerne ľahko zostrojiť model danej formuly, t. j. nájdeme tie pravdivostné interpretácie premenných, pre ktoré je formula pravdivá. Každá vetva tabla je priradená konkrétnej pravdivostnej interpretácii. Táto vlastnosť nám umožňuje formulovať mnoho viet o konštrukcii tautologického vyplývania [12] danej formuly z predpokladov, ktoré tvoria teoretický základ pre alternatívnu výpočtovú realizáciu

¹ Fakulta informatiky a informačných technológií, STU Bratislava, Ilkovičova 3, 842 16 Bratislava, E-mail: kvasnicka@fiit.stuba.sk, pospichal@fiit.stuba.sk

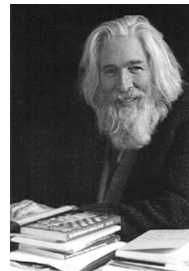
dôkazu tautologičnosti formúl alebo dáva návod ako modifikovať predpoklady, aby z nich vyplývala daná formula a poskytuje mnoho ďalších zaujímavých vlastností (napr. abdukcii). Za tvorcov metódy sémantických tabiel možno pokladať trojicu matematikov a logikov E. W. Betha, J. Hintikka a R. Smullyana, ktorý ju vytvorili na prelome 50.-60. rokov minulého storočia (pozri obr. 1).



(A) E. W. Beth



(B) J. Hintikka



(C) R. Smullyan

Obr. 1. Tvorcovia techniky sémantických tabiel. (A) Belgický matematik a logik Evert Willem Beth (1908 –1964), ktorého práce podstatne ovplyvnili základy matematiky. (B) Fínsky filozof a logik Jaakko Hintikka (1929) patrí medzi významné osobnosti, ktoré sa zaslúžili o rozvoj modálnej logiky. (C) Americký matematik, filozof, logik a koncertný umelec Raymond Smullyan vytvoril modernú verziu sémantických tabiel.

1.1 Bethove sémantické tablá

Belgický logik a matematik E. W. Beth v r. 1955 [2,3] zaviedol *sémantické tablá* pre určenie toho, či daná výroková formula φ je tautológia alebo nie. Tento dôkaz bol vykonaný nepriamo tak, že dokazoval kontradikčnosť negácie $\neg\varphi$. Jeho postup nasledovne ilustrujeme pomocou dôkazu tautologičnosti známeho zákona výrokovej logiky „*reductio ad absurdum*“ $\varphi = (p \Rightarrow q) \wedge (p \Rightarrow \neg q) \Rightarrow \neg p$, ktorý sa často využíva v matematike pri dôkazu sporom, pozri tab 1.

Tabuľka je rozdelená na pravdivú a nepravdivú časť. V prvom riadku nepravdivej časti je umiestnená formula (1), vzhľadom k centrálnej spojke (implikácia v obdĺžniku) táto formula môže byť rozdelená na pravdivú (2) a nepravdivú (3) časť. Nepravdivú podformulu (3) preniesieme do pravdivej časti (4). Pravdivá podformula (2) s centrálnou spojkou konjunkciou je rozdelená na dve pravdivé podformuly (5) a (6). Pravdivá podformula (5) je rozdelená vzhľadom k centrálnej implikácii na pravdivú (7) a nepravdivú (8) podformulu. Podformula (8) je v kontradikcii s podformulou (4), takže stĺpec pod (8) je „uzavretý“ (ďalej už nie je predlžovaný). Podformula (6) je rozdelená na dve podformuly (9) a (10), ktoré sú umiestnené do stĺpca pod (7). Ľavý stĺpec je uzavretý, pretože (7) a (9) sú v kontradikcii. Podobne, aj pravý stĺpec je uzavretý, pretože (4) a (10) sú v kontradikcii. Môžeme teda konštatovať, že sme dokázali kontradikčnosť formuly $\neg\varphi = \neg((p \Rightarrow q) \wedge (p \Rightarrow \neg q) \Rightarrow \neg p) \equiv (p \Rightarrow q) \wedge (p \Rightarrow \neg q) \wedge p$, čiže formula φ je tautológia.

Tabuľka 1. Bethovo sémantické tablo pre formulu $(p \Rightarrow q) \wedge (p \Rightarrow \neg q) \Rightarrow \neg p$

| | | | |
|---|------------|--|--|
| nepravdivá | | (1) $(p \Rightarrow q) \wedge (p \Rightarrow \neg q) \Rightarrow \neg p$ | |
| pravdivá | | nepravdivá | |
| (2) $(p \Rightarrow q) \wedge (p \Rightarrow \neg q)$ | | (3) $\neg p$ | |
| (4) p | | | |
| (5) $p \Rightarrow q$ | | | |
| (6) $p \Rightarrow \neg q$ | | | |
| Pravdivá | nepravdivá | | |
| (7) q | (8) p | | |
| pravdivá | nepravdivá | | |
| (9) $\neg q$ | (10) p | | |

Ako vyplýva z tohto jednoduchého ilustračného príkladu, použitie Bethovej techniky sémantických tabiel je pomerne ťažkopádne a neprehľadné. Avšak už Beth ukázal úzky vzťah medzi jeho sémantickými tablami a Gentzenovou prirodzenou dedukciou. Použil tzv. *princíp podformuly*, podľa ktorého, ak daná formula má dôkaz, potom aspoň jedna z jej hlavných podformúl (vzhľadom k centrálnej spojke formuly) musí byť dokázateľná, čo nie je nič iné ako Gentzenov rozšírený Hauptsatz [22]. Takže aplikujúc inverzný postup z Bethovho tabla, realizujeme prirodzenú dedukciu založenú na elementárnych transformáciách eliminácie a introdukcie konjunkcie alebo disjunkcie.

1.2 Hintikkov prístup modelových množín

Fínsky filozof a logik J. Hintikka [9,10] skoro súčasne s Bethom vypracoval alternatívnu sémantickú metódu, ktorá je založená na tzv. modelových množinách. Podobne ako Beth, aj jeho primárnym záujmom bolo dokázať tautologičnosť formuly φ tak, že negácia $\neg\varphi$ je kontradikcia. Hlavnou výhodou jeho prístupu bola skutočnosť, že požitá metóda je ľahko aplikovateľná nielen pre výrokovú logiku ale aj pre modálne logiky.

Hintikkov prístup je založený na *modelových množinách*, ktoré sú špecifikované takto: Nech τ je daný model výrokovej formuly (pravdivostná interpretácia výrokových premenných), k tomuto modelu priradíme množinu formúl, ktoré sú pravdivé pre interpretáciu-model

$$\mu = \{ \varphi; \text{val}_\tau(\varphi) = 1 \}$$

t. j. modelová množina μ obsahuje formuly, ktoré sú pravdivé v rámci modelu M . Pomocou tohto formalizmu môžeme formulovať tieto tri tvrdenia

- (a) $(\neg\varphi \in \mu) \equiv (\varphi \notin \mu)$ (φ je atomická premenná)
 (b) $((\varphi \wedge \psi) \in \mu) \equiv ((\varphi \in \mu) \wedge (\psi \in \mu))$
 (c) $((\varphi \vee \psi) \in \mu) \equiv ((\varphi \in \mu) \vee (\psi \in \mu))$
 (d) $((\varphi \Rightarrow \psi) \in \mu) \equiv ((\neg\varphi \in \mu) \vee (\psi \in \mu))$

Pomocou jednoduchého ilustračného príkladu ukážeme ako Hintikkova metóda „pracuje“. Opäť budeme študovať formulu $\varphi = (p \Rightarrow q) \wedge (p \Rightarrow \neg q) \Rightarrow \neg p$, jej negácia má tvar konjunkcie, $\neg\varphi = (p \Rightarrow q) \wedge (p \Rightarrow \neg q) \wedge p$. Predpokladajme, že $\neg\varphi \in \mu$

$$((p \Rightarrow q) \sqcap ((p \Rightarrow \neg q) \wedge p)) \in \mu \quad (12)$$

Použitím (b) na centrálnu spojku konjunkcie dostaneme

$$(p \sqsupseteq q) \in \mu \quad (13)$$

$$((p \Rightarrow \neg q) \sqcap p) \in \mu \quad (14)$$

Aplikáciou (d) na (13) dostaneme

$$(\neg p \in \mu) \vee (q \in \mu) \quad (15)$$

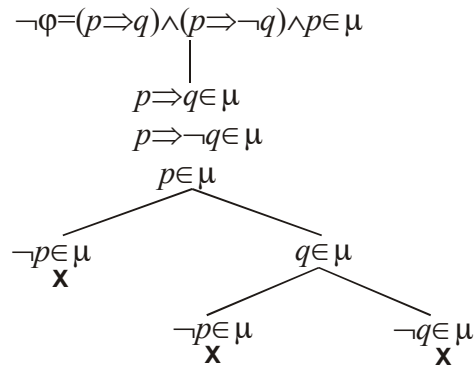
Podobne, aplikáciou (b) na (14) dostaneme

$$(p \Rightarrow \neg q) \in \mu \quad (16)$$

$$p \in \mu \quad (17)$$

Konečne, použitím (d) na (16) dostaneme

$$(\neg p \in \mu) \vee (\neg q \in \mu) \quad (18)$$



Obr. 2. Diagramatická reprezentácia postupu rozkladu formuly $\neg\varphi = (p \Rightarrow q) \wedge (p \Rightarrow \neg q) \wedge p$ pomocou vzťahov (a-d).

Pre názornosť vyjadríme túto postupnosť rozkladu formuly $\neg\varphi$ pomocou diagramu znázorneného na obr. 2 (pre úplnosť poznamenajme, že Hintikka tento diagram nepoužíval). Pomocou tohto obrázku ľahko zistíme, že takáto modelová množina neexistuje, existujúce kontradikcie $p \in \mu$, $\neg p \in \mu$ a $q \in \mu$, $\neg q \in \mu$ jednoznačne implikujú, že táto množina nemôže existovať. To potom znamená, že formula $\neg\varphi$ je kontradikcia (t. j. nikdy nie je pravdivá), čiže jej negácia (t. j. formula φ) je tautológia, čo bolo potrebné dokázať.

Môžeme konštatovať, že obe metódy, tak Bethova metóda, ako aj Hintikkova metóda, nie sú veľmi prívetivé k užívateľovi. Bethova metóda je založená na dvojstĺpcových tabuľkách (sémantických tabľách), kde sa jednotlivé stĺpce rozdeľujú na ďalšie dva podstĺpce, atď. Hintikkova metóda založená na modelovej množine je tiež pomerne ťažkopádna. Obe metódy vyžadujú formálne a notačné zjednodušenie, ktoré by ich urobilo viac prívetivými k užívateľovi.

1.3 Smullyanov diagramatický prístup

Americký filozof a logik R. J. Smullyan [19-21] podstatne zjednodušil metódu sémantických tabiel (ktoré nazýval analytické tablá, čím chcel zdôrazniť skutočnosť, že metóda je založená na rozklade formúl na dve podformuly špecifikované centrálnou logickou spojkou) do tvaru, ktorý sa používa aj v súčasnosti. Začneme so špecifikáciou Smullyanovho *označeného systému*, kde každá formula je označená buď T (pravdivá) alebo F (nepravdivá). Označené formuly pôsobia buď konjunktívne alebo disjunktívne, pričom konjunktívne formuly sú zoskupené do formúl *typu A* a disjunktívne formuly sú zoskupené do formúl *typu B*. Skupina typu *A* obsahuje formuly α_1 a α_2 , zatiaľ čo skupina typu *B* obsahuje formuly β_1 a β_2 . Podobným spôsobom, v ktorom sa nepoužívajú písmena T a F pre označenie formúl, sú špecifikované aj neoznačené formuly. Smullyan definoval tak pre označené ako aj neoznačené formuly dve tabuľky (pozri tab. 2, 3 a 4)

Tabuľka 2. Formuly typu A

| α | α_1 | α_2 |
|----------------------------------|---------------|------------|
| $T(\varphi \wedge \psi)$ | $T\varphi$ | $T\psi$ |
| $F(\varphi \vee \psi)$ | $F\varphi$ | $F\psi$ |
| $F(\varphi \Rightarrow \psi)$ | $T\varphi$ | $F\psi$ |
| $T(\neg\varphi)$ | $F\varphi$ | |
| $F(\neg\varphi)$ | $T\varphi$ | |
| $(\varphi \wedge \psi)$ | φ | ψ |
| $\neg(\varphi \vee \psi)$ | $\neg\varphi$ | $\neg\psi$ |
| $\neg(\varphi \Rightarrow \psi)$ | φ | $\neg\psi$ |
| $\neg\neg\varphi$ | φ | |

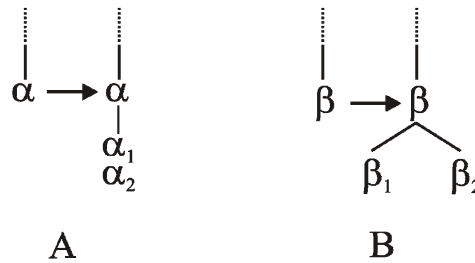
Tabuľka 3. Formuly typu B

| β | β_1 | β_2 |
|-------------------------------|---------------|------------|
| $F(\varphi \wedge \psi)$ | $F\varphi$ | $F\psi$ |
| $T(\varphi \vee \psi)$ | $T\varphi$ | $T\psi$ |
| $T(\varphi \Rightarrow \psi)$ | $F\varphi$ | $T\psi$ |
| $\neg(\varphi \wedge \psi)$ | $\neg\varphi$ | $\neg\psi$ |
| $(\varphi \vee \psi)$ | φ | ψ |
| $(\varphi \Rightarrow \psi)$ | $\neg\varphi$ | ψ |

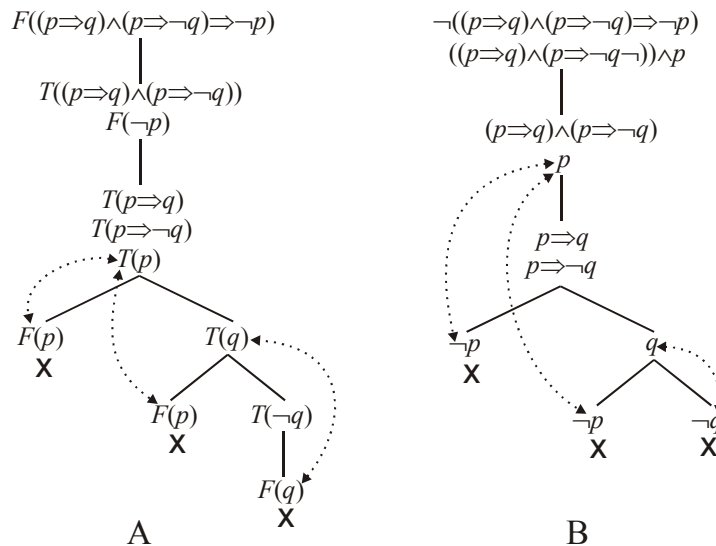
Tabuľka 4. Formuly typu B pre ekvivalenciu

| β | β_1 | β_2 |
|-----------------------------|---------------------------|-------------------------------|
| $T(\varphi \equiv \psi)$ | $T\varphi \wedge T\psi$ | $F\varphi \wedge F\psi$ |
| $F(\varphi \equiv \psi)$ | $T\varphi \wedge F\psi$ | $F\varphi \wedge T\psi$ |
| $(\varphi \equiv \psi)$ | $\varphi \wedge \psi$ | $\neg\varphi \wedge \neg\psi$ |
| $\neg(\varphi \equiv \psi)$ | $\varphi \wedge \neg\psi$ | $\neg\varphi \wedge \psi$ |

V Smullyanovej metóde sa postupne zostrojuje sémantické tablo vo forme koreňového stromu, kde úlohu koreňa hrá formula $\neg\varphi$, strom sa tvorí (predlžuje) tak, že pre A formuly sa strom nevetví (formuly sa píšú pod seba), zatiaľ čo pre B formuly dochádza k vetveniu stromu, každá vetva obsahuje jednu novú výslednú formulu (pozri obr. 3).



Obr. 3. Predlžovanie sémantického tabla podľa Smullyana. Diagram A znázorňuje predĺženie konjunktívnej α formy, diagram B znázorňuje predĺženie disjunktívnej β formy.



Obr. 4. Znázornenie sémantického tabla podľa Smullyana pre dôkaz tautologičnosti formuly $\varphi = (p \Rightarrow q) \wedge (p \Rightarrow \neg q) \Rightarrow \neg p$. Diagram A obsahuje označené formuly, diagram B obsahuje neoznačené formuly. Každá vetva sémantického tabla je označená symbolom **X**, t. j. formula φ je tautológia.

Ilustračný príklad použitia Smullyanovej metódy k dôkazu tautologičnosti formuly $\varphi = (p \Rightarrow q) \wedge (p \Rightarrow \neg q) \Rightarrow \neg p$ (ktorá sa nazýva v klasickej výrokovej logike „*reductio ad absurdum*“) je znázornený na obr. 4. Sémantické tablo v tomto prístupe je reprezentované koreňovým stromom, kde vrchným symbolom (koreňom) je ohodnotená formula $F(\varphi)$, použitím tab. 2-4 jednotlivé vrcholy predlžujeme, tento proces ukončíme vtedy, keď všetky koncové vrcholy stromu – tabla sú ohodnotené elementárne formuly neobsahujúce logické spojky. V prípade, že každá vetva tabla obsahuje dvojicu kontradiktórnych ohodnotených elementárnych formúl, danú vetvu označíme symbolom **x** (hovoríme, že daná vetva je uzavretá). V opačnom prípade, ak daná vetva neobsahuje dvojicu kontradiktórnych ohodnotených elementárnych formúl, potom túto vetvu označíme symbolom **o**, hovoríme, že daná vetva je otvorená. V prípade, že Smullyanovo sémantické tablo s vrcholom $F(\varphi)$ má všetky vetvy uzavreté, potom formula φ je tautológia.

2 Boolove funkcie

Formula výrokovej logiky $\varphi(x_1, x_2, \dots, x_n)$, ktorá obsahuje n výrokových premenných (atomických formúl) x_1, x_2, \dots, x_n , môže byť na sémantickej úrovni interpretovaná ako zobrazenie vektora binárnych argumentov na binárnu funkčnú hodnotu. Takéto zobrazenie sa nazýva **Boolova funkcia** [11]

$$\varphi: \{0, 1\}^n \rightarrow \{0, 1\} \quad \text{alebo} \quad y = \varphi(x_1, x_2, \dots, x_n) \quad (19)$$

ktorá priradí n binárnym premenným (argumentom x_1, x_2, \dots, x_n) binárnu funkčnú hodnotu y .

Pre lepšie pochopenie významu tohto prístupu pre výrovkovú logiku uvažujme tento výrok, ktorý obsahuje štyri výrokové premenné

$$\varphi(x_1, x_2, x_3, x_4) = (((x_1 \wedge x_2) \vee (x_3 \Rightarrow x_4)) \Rightarrow x_1) \quad (20)$$

Táto „výrovková funkcia“ má $16=2^4$ rôznych špecifikácií premenných. Použitím tabuľkovej metódy môžeme vypočítať pravdivostné (funkčné) hodnoty tejto funkcie (pozri tab. 5).

Tabuľka 5. Boolova funkcia 4 premenných

| # | x_1 | x_2 | x_3 | x_4 | $\varphi(x_1, x_2, x_3, x_4)$ |
|-------|-------|-------|-------|-------|-------------------------------|
| 1 | 0 | 0 | 0 | 0 | 0 |
| | | | | | |
| 16 | 1 | 1 | 1 | 1 | 1 |

K tomu, aby sme dostali teóriu Boolových funkcií do priamej súvislosti s výrovkovou logikou, definujme elementárne Boolove funkcie 1- a 2-premenných, pomocou ktorých sa zostrojujú všeobecné Boolove funkcie (pozri tab. 6 a 7). Tretia unárna funkcia $u_3(x)$ reprezentuje výrovkovú spojku negácie, ostatné tri unárne funkcie nemajú vo výrokovej logike analógiu.

Tabuľka 6. Unárne Boolove funkcie

| # | x | u_1 | u_2 | u_3 | u_4 |
|-------------|-----|-------|-------|----------|-------|
| 1 | 0 | 0 | 0 | 1 | 1 |
| 2 | 1 | 0 | 1 | 0 | 1 |
| log. spojky | | | x | $\neg x$ | |

Tabuľka 7. Binárne Boolove funkcie

| # | x_1 | x_2 | f_1 | f_2 | f_3 | f_4 | f_5 | f_6 | f_7 | f_8 | f_9 | f_{10} | f_{11} | f_{12} | f_{13} | f_{14} | f_{15} | f_{16} |
|-------------|-------|-------|-------|----------|-------|-------|-------|-------|----------|--------|--------------|----------|------------|----------|------------|---------------|------------|----------|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 3 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 4 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| log. spojky | | | | \wedge | | x_1 | | x_2 | \oplus | \vee | \downarrow | \equiv | $\neg x_2$ | | $\neg x_1$ | \Rightarrow | \uparrow | |

Celkový počet binárných Boolových funkcií je $16=2^4$, avšak iba štyri z nich majú vo výrokovej logike svojich reprezentantov, v tab. 7 sú vyznačené tmavými stĺpcami.

Nech φ a ψ sú dve formuly výrokovej logiky s rovnakými výrokovými premennými, $\varphi = \varphi(p_1, p_2, \dots, p_n)$ a $\psi = \psi(p_1, p_2, \dots, p_n)$. Nech $\tau = (\tau_1, \tau_2, \dots, \tau_n) \in \{0, 1\}^n$ je interpretácia týchto výrokových premenných, t. j. pre každé $\tau \in \{0, 1\}^n$ platí: $val_{\tau}(p_i) = \tau_i$. Tak napríklad, pre $n = 3$, premenné p_1, p_2, p_3 sú interpretované $\tau = (0, 1, 1) \in \{0, 1\}^3$ takto: $val_{\tau}(p_1) = 0$, $val_{\tau}(p_2) = 1$, $val_{\tau}(p_3) = 1$, t. j. premenná p_1 je nepravdivá a premenné p_2, p_3 sú pravdivé.

Definícia 1.

Hovoríme, že formuly $\varphi = \varphi(p_1, p_2, \dots, p_n)$ a $\psi = \psi(p_1, p_2, \dots, p_n)$ sú (logicky) **ekvivalentné**, čo zapisujeme $\varphi \sim \psi$, vtedy a len vtedy, ak ich pravdivostné hodnoty sú rovnaké pre každú interpretáciu $\tau \in \{0, 1\}^n$,

$$(\varphi \sim \psi) =_{def} (\forall \tau)(val_{\tau}(\varphi) = val_{\tau}(\psi)) \quad (21)$$

Poznamenajme, že symbol ' \sim ' je binárna relácia a nie binárna logická spojka, aj keď svojou definíciou je veľmi blízka spojke ' \equiv '.

Pre naše ďalšie úvahy o význame teórie Boolových funkcií vo výrokovej logike zavedieme dva dôležité pojmy a ukážeme, že pomocou nich môže byť „analyticky“ vyjadrená každá Boolova funkcia, ktorá je špecifikovaná len tabuľkou svojich funkčných hodnôt (pozri tab. 5). Každá formula výrokovej logiky (a teda aj každá Boolova funkcia, ktorá používa len klasické binárne funkcie) môže byť prepísaná do ekvivalentného konjunktívneho alebo disjunktívneho tvaru. Práve tieto špeciálne tvary Boolových funkcií (alebo aj výrokových formúl) majú význam pre konštrukciu „analytických“ funkcií určených len tabuľkou. Zavedieme nasledujúcu terminológiu:

Definícia 2.

1. **Literál** je výroková premenná alebo jej negácia, t. j. $l = p$ alebo $l = \neg p$, dva literály l a l' sa nazývajú komplementárne, ak sú tvorené výrokovou premennou a jej negáciou, t. j. $l = p$ a $l' = \neg p$.
2. **Konjunktívna klauzula** je vytvorená pomocou konjunkcie literálov ($l_1 \wedge l_2 \wedge \dots$). Podobne, **disjunktívna klauzula** je vytvorená pomocou disjunkcie literálov ($l'_1 \vee l'_2 \vee \dots$).
3. **Konjunktívna normálna forma (KNF)** je tvorená pomocou konjunkcie disjunktívnych klauzúl ($(l_1 \vee l_2 \vee \dots) \wedge (l'_1 \vee l'_2 \vee \dots) \wedge \dots$). Podobne, **disjunktívna normálna forma (DNF)** je tvorená pomocou disjunkcie konjunktívnych klauzúl ($(l_1 \wedge l_2 \wedge \dots) \vee (l'_1 \wedge l'_2 \wedge \dots) \vee \dots$).

Príklad 1. Príkladom disjunktívnej resp. konjunktívnej normálnej formy sú tieto dve formuly

$$(p_1 \wedge \neg p_2 \wedge \neg p_3 \wedge p_4) \vee (\neg p_3 \wedge p_5 \wedge \neg p_6) \\ (\neg p_1 \vee \neg p_2 \vee \neg p_3) \wedge (\neg p_3 \vee p_5)$$

Ukážeme, že každá výroková formula φ (Boolova funkcia s klasickými spojkami) môže byť prepísaná do ekvivalentnej disjunktívnej resp. konjunktívnej normálnej formy, $\varphi \sim \varphi_{DNF}$ resp. $\varphi \sim \varphi_{KNF}$ tvaru. Tento postup je založený na použití disjunktívneho tvaru implikácie, De Morganových zákonov a distributívnych zákonov pre konjunkciu a disjunkciu. Uvažujme formulu $\varphi = (p \vee q) \Rightarrow (r \wedge \neg q)$, použitím disjunktívneho tvaru implikácie túto formulu prepíšeme do tvaru

$$\neg(p \vee q) \vee (r \wedge \neg q)$$

Aplikáciou De Morganovho zákona pre negáciu disjunkcie dostaneme požadovaný disjunktívny tvar

$$\varphi_{DNF} = (\neg p \wedge \neg q) \vee (r \wedge \neg q)$$

kde $\varphi \sim \varphi_{DNF}$. Podobne, ako v predošlom ilustračnom príklade, ukážeme, že každá výroková formula môže byť prepísaná taktiež aj do konjunktívneho tvaru. Budeme študovať rovnakú formulu ako v predošlom príklade, jej disjunktívny tvar $(\neg p \wedge \neg q) \vee (r \wedge \neg q)$ je ďalej upravovaný pomocou distributívnych zákonov pre konjunkciu a disjunkciu

$$((\neg p \wedge \neg q) \vee (r \wedge \neg q)) \equiv (\neg p \vee (r \wedge \neg q)) \wedge (\neg q \vee (r \wedge \neg q)) \equiv \\ (\neg p \vee r) \wedge (\neg p \vee \neg q) \wedge (\neg q \vee r) \wedge (\neg q \vee \neg q) \equiv \\ (\neg p \vee r) \wedge (\neg p \vee \neg q) \wedge (\neg q \vee r) \wedge \neg q$$

Potom

$$\varphi_{KNF} = (\neg p \vee r) \wedge (\neg p \vee \neg q) \wedge (\neg q \vee r) \wedge \neg q$$

Veta 1.

(1) Pre každú formulu φ existuje ekvivalentná formula, ktorá má tvar **disjunktívnej normálnej formy** $\varphi_{DNF} \sim \varphi$

$$\forall \varphi \exists \varphi_{DNF} (\varphi \sim \varphi_{DNF}) \quad (22)$$

(2) Pre každú formulu φ existuje ekvivalentná formula, ktorá má tvar **konjunktívnej normálnej formy** $\varphi_{KNF} \sim \varphi$

$$\forall \varphi \exists \varphi_{KNF} (\varphi \sim \varphi_{KNF}) \quad (23)$$

Disjunktívna normálna forma pre kontradikciu (konštantná formula vždy nepravdivá) je tvorená disjunkciou konjunktívnych klauzúl, z ktorých každá obsahuje

dvojicu komplementárnych literálov, $\varphi_{DNF} = \left(\underbrace{l \wedge \tilde{l}}_0 \wedge \dots \right) \vee \left(\underbrace{l' \wedge \tilde{l}'}_0 \wedge \dots \right) \vee \dots \equiv 0$.

Podobne, konjunktívna normálna forma pre tautológiu (konštantná formula vždy pravdivá) je tvorená konjunkciou disjunktívnych klauzúl, u ktorých každá obsahuje

dvojicu komplementárnych literálov, $\varphi_{KNF} = \left(\underbrace{l \vee \tilde{l}}_1 \vee \dots \right) \wedge \left(\underbrace{l' \vee \tilde{l}'}_1 \vee \dots \right) \wedge \dots \equiv 1$. Tieto

zaujímavé vlastnosti môžeme zhrnúť do nasledujúcej vety.

Veta 2.

(1) Formula φ je kontradikcia práve vtedy, ak jej ekvivalentná disjunktívna normálna forma φ_{DNF} obsahuje v každej konjunktívnej klauzule dvojicu komplementárnych literálov.

(2) Formula φ je tautológia práve vtedy, ak jej ekvivalentná konjunktívna normálna forma φ_{KNF} obsahuje v každej disjunktívnej klauzule dvojicu komplementárnych literálov.

Prvý alternatívny dôkaz vety 1: upriamime našu pozornosť najprv na konštrukciu ekvivalentnej formuly v disjunktívnom tvare. Budeme uvažovať len tie interpretácie premenných τ , ktoré vytvárajú jednotkovú pravdivostnú hodnotu formuly φ , t.j. pre dané τ platí $val_{\tau}(\varphi) = 1$. Podobne pre dané τ nové funkčné premenné (literály)

$$x_i^{(\tau)} = \begin{cases} x_i & (\text{ak } val_{\tau}(x_i) = 1) \\ \neg x_i & (\text{ak } val_{\tau}(x_i) = 0) \end{cases} \quad (24)$$

Konjunkcia týchto premenných môže byť chápaná ako pomocná Boolova funkcia

$$\Psi_{\tau}(x_1, x_2, \dots, x_n) = x_1^{(\tau)} \wedge x_2^{(\tau)} \wedge \dots \wedge x_n^{(\tau)} \quad (25)$$

Táto konjunkcia má jednotkovú funkčnú hodnotu (je pravdivá) len pre také hodnoty premenných, ktoré sú totožné s binárnymi hodnotami interpretácie τ , pre všetky ostatné hodnoty premenných je funkčná pravdivostná hodnota nulová. Potom disjunktívna forma funkcie φ_{DNF} , ktorá je ekvivalentná formule φ , $\varphi_{DNF} \sim \varphi$, má tvar

$$\Phi_{DNF} = \bigvee_{\substack{\tau \\ (val_{\tau}(\varphi)=1)}} x_1^{(\tau)} \wedge x_2^{(\tau)} \wedge \dots \wedge x_n^{(\tau)} \quad (26)$$

kde disjunkcia beží cez všetky τ , pre ktoré $val_{\tau}(\varphi) = 1$.

Analogickým spôsobom zostrojíme aj konjunktívnu formu Φ_{KNF} formuly φ , kde $\Phi_{KNF} \sim \varphi$. V tomto prípade budeme uvažovať len také interpretácie τ , ktoré majú nulovú pravdivostnú funkčnú hodnotu, t.j. pre dané τ platí $val_{\tau}(\varphi) = 0$. Definujme pre dané τ nové funkčné premenné (literály)

$$\tilde{x}_i^{(\tau)} = \begin{cases} x_i & (\text{ak } val_{\tau}(x_i) = 0) \\ \neg x_i & (\text{ak } val_{\tau}(x_i) = 1) \end{cases} \quad (27)$$

Disjunkcia týchto premenných tvorí pomocnú Boolovu funkciu

$$\tilde{\Psi}_{\tau}(x_1, x_2, \dots, x_n) = \tilde{x}_1^{(\tau)} \vee \tilde{x}_2^{(\tau)} \vee \dots \vee \tilde{x}_n^{(\tau)} \quad (28)$$

Konjunktívna forma Φ_{KNF} je

$$\Phi_{KNF} = \bigwedge_{\substack{\tau \\ (val_{\tau}(\varphi)=0)}} \tilde{x}_1^{(\tau)} \vee \tilde{x}_2^{(\tau)} \vee \dots \vee \tilde{x}_n^{(\tau)} \quad (29)$$

kde konjunkcia beží cez všetky špecifikácie τ , pre ktoré $val_{\tau}(\varphi) = 0$.

To, akým spôsobom zostrojíme Boolovu funkciu, či v disjunktívnej alebo konjunktívnej forme, je určené počtom jednotkových resp. nulových funkčných hodnôt. Tak napr. ak v tabuľke sú dominantné nulové (jednotkové) funkčné hodnoty, potom je výhodné použiť konjunktívnu (disjunktívnu) normálnu formu, týmto výberom sa minimalizuje rozsah zostrojovanej formy. V prípade, keď tabuľka obsahuje približne rovnaký počet nulových a jednotkových funkčných hodnôt, konštrukcia Boolovej funkcie je približne rovnako obtiažná v oboch formách.

Príklad 2. Vykonajte konštrukciu Boolových funkcií, ktoré sú určené funkčnými hodnotami uvedenými v tab. 8, v disjunktívnej a/alebo konjunktívnej forme pomocou formúl (26) a (29).

Tabuľka 8. Určenie dvoch formúl α a β pomocou funkčných hodnôt

| # | x_1 | x_2 | x_3 | α | β |
|---|-------|-------|-------|----------|---------|
| 1 | 0 | 0 | 0 | 0 | 1 |
| 2 | 0 | 0 | 1 | 0 | 1 |
| 3 | 0 | 1 | 0 | 1 | 0 |
| 4 | 0 | 1 | 1 | 0 | 1 |
| 5 | 1 | 0 | 0 | 0 | 1 |
| 6 | 1 | 0 | 1 | 1 | 1 |
| 7 | 1 | 1 | 0 | 0 | 0 |
| 8 | 1 | 1 | 1 | 0 | 1 |

V prvom kroku vykonáme konštrukciu formuly α , ktorej funkčné hodnoty sú určené v tab. 8. V tomto prípade počet výskytov jednotkových funkčných hodnôt je podstatne menší ako nulových hodnôt, preto konštrukciu vykonáme v disjunktívnej forme. Ku konštrukcii použijeme teda len riadky 3 a 6, interpretácie premenných sú $\tau_3 = (x_1/0, x_2/1, x_3/0)$ a $\tau_6 = (x_1/1, x_2/0, x_3/1)$. Príslušné konjunkcie $\psi_\tau(x_1, x_2, \dots, x_n)$ (25) majú tvar

$$\psi_{\tau_3}(x_1, x_2, x_3) = \neg x_1 \wedge x_2 \wedge \neg x_3$$

$$\psi_{\tau_6}(x_1, x_2, x_3) = x_1 \wedge \neg x_2 \wedge x_3$$

Použitím (26) dostaneme konečný tvar zostrojovanej formy

$$\alpha = (\neg x_1 \wedge x_2 \wedge \neg x_3) \vee (x_1 \wedge \neg x_2 \wedge x_3)$$

V druhom kroku vykonáme konštrukciu konjunktívnej formy formuly β , ktorej funkčné hodnoty sú špecifikované tabuľkou 8. Príslušné interpretácie τ , ktoré sú priradené nulovým funkčným hodnotám majú tvar: $\tau_3 = (x_1/0, x_2/1, x_3/0)$ a $\tau_7 = (x_1/1, x_2/1, x_3/0)$. Týmto interpretáciám sú podľa (28) priradené disjunktie

$$\tilde{\psi}_{\tau_3}(x_1, x_2, x_3) = x_1 \vee \neg x_2 \vee x_3$$

$$\tilde{\psi}_{\tau_7}(x_1, x_2, x_3) = \neg x_1 \vee \neg x_2 \vee x_3$$

Použitím (29) dostaneme konečný tvar zostrojovanej konjunktívnej formy

$$\beta = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3)$$

Druhý alternatívny dôkaz vety 1: použijeme metódu z príkladu 1. Táto metóda spočíva v tom, že formulu φ postupne prepisujeme do tvaru φ_{DNF} pomocou známych ekvivalencií výrokovkej logiky, akými sú De Morganove zákony a distributívne zákony medzi disjunkciou a konjunkciou. Dôkaz vykonáme indukciou vzhľadom k syntaktickému stromu formuly φ . V tabuľke 9 sú uvedené základné formuly pre prepis do tvaru DNF.

Tabuľka 9. Elementárne transformácie prepisu formuly φ na φ_{DNF} .

| # | Pôvodná formula | Transformovaná formula |
|---|----------------------------------|--|
| 1 | $(\alpha \wedge \beta)$ | $\rightarrow (\alpha) \wedge (\beta)$ |
| 2 | $(\alpha \vee \beta)$ | $\rightarrow (\alpha) \vee (\beta)$ |
| 3 | $(\alpha \Rightarrow \beta)$ | $\rightarrow \neg(\alpha) \vee (\beta)$ |
| 4 | $(\alpha \equiv \beta)$ | $\rightarrow (\neg(\alpha) \wedge \neg(\beta)) \vee ((\alpha) \wedge (\beta))$ |
| 5 | $\neg(\alpha \wedge \beta)$ | $\rightarrow \neg(\alpha) \vee \neg(\beta)$ |
| 6 | $\neg(\alpha \vee \beta)$ | $\rightarrow \neg(\alpha) \wedge \neg(\beta)$ |
| 7 | $\neg(\alpha \Rightarrow \beta)$ | $\rightarrow (\alpha) \wedge \neg(\beta)$ |

| | | | |
|----|-------------------------------------|---------------|--|
| 8 | $\neg(\alpha \equiv \beta)$ | \rightarrow | $(\neg(\alpha) \wedge (\beta)) \vee ((\alpha) \wedge \neg(\beta))$ |
| 9 | $\alpha \wedge (\beta \vee \gamma)$ | \rightarrow | $(\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$ |
| 10 | $\alpha \vee (\beta \wedge \gamma)$ | \rightarrow | $(\alpha \vee \beta) \wedge (\alpha \vee \gamma)$ |

Prvé štyri riadky tejto tabuľky obsahujú transformácie elementárnych logických spojok do tvaru konjunkcie alebo disjunkcie literálov. Ďalšie štyri transformácie 5-8 obsahujú transformácie negácie elementárnych logických spojok do tvaru konjunkcie alebo disjunkcie literálov. Posledné dve transformácie 9 a 10 reprezentujú použitie distributívnych zákonov medzi konjunkciou a disjunkciou, ktoré musia byť použité k zjednodušeniu upravovanej formuly.

Príklad 3. Vykonajte transformáciu formuly

$$\varphi = (p \equiv q) \wedge (r \vee \neg(r \vee \neg p))$$

do tvaru DNF pomocou elementárnych transformácií z tabuľky 9. Tento prepis bude vykonaný pre názornosť ako postupnosť elementárnych krokov:

Krok 1: Centrálna konjunkcia (tvoriaca koreň príslušného syntaktického stromu) v φ je prepísaná pomocou elementárnej transformácie 1

$$\begin{aligned} \varphi &= \left(\underbrace{(p \equiv q)}_{\alpha} \wedge \underbrace{(r \vee \neg(r \vee \neg p))}_{\beta} \right) \\ &\downarrow \\ &(p \equiv q) \wedge (r \vee \neg(r \vee \neg p)) \end{aligned}$$

Krok 2: Centrálna spojka ekvivalencie na ľavej strane je prepísaná pomocou elementárnej transformácie 4 a centrálna spojka disjunkcie pravej strane je prepísaná pomocou 2 a 6

$$\begin{aligned} &\left(\underbrace{(p)}_{\alpha} \equiv \underbrace{(q)}_{\beta} \right) \wedge \left(\underbrace{(r)}_{\gamma} \vee \underbrace{(\neg(r \vee \neg p))}_{\delta} \right) \\ &\downarrow \\ &\left((\neg(p) \wedge \neg(q)) \vee ((p) \wedge (q)) \right) \wedge \left((r) \vee (\neg(r) \wedge \neg(\neg p)) \right) \\ &\downarrow \\ &\left((\neg p \wedge \neg q) \vee (p \wedge q) \right) \wedge \left((r) \vee (\neg r \wedge p) \right) \end{aligned}$$

Krok 3: Centrálna spojka konjunkcie je prepísaná (roznásobená) pomocou 9

$$\begin{aligned}
& \left(\underbrace{(\neg p \wedge \neg q) \vee (p \wedge q)}_{\alpha} \right) \wedge \left(\underbrace{(r)}_{\beta} \vee \underbrace{(\neg r \wedge p)}_{\gamma} \right) \\
& \quad \downarrow \\
& \left(\left(\underbrace{(\neg p \wedge \neg q) \vee (p \wedge q)}_{\beta} \right) \wedge \underbrace{r}_{\alpha} \right) \vee \left(\left(\underbrace{(\neg p \wedge \neg q) \vee (p \wedge q)}_{\beta'} \right) \wedge \underbrace{(\neg r \wedge p)}_{\alpha'} \right) \\
& \quad \downarrow \\
& \left((\neg p \wedge \neg q \wedge r) \vee (p \wedge q \wedge r) \right) \vee \left(\underbrace{(\neg p \wedge \neg q \wedge \neg r \wedge p)}_0 \vee \underbrace{(p \wedge q \wedge \neg r \wedge p)}_{p \wedge q \wedge \neg r} \right) \\
& \quad \downarrow \\
& (\neg p \wedge \neg q \wedge r) \vee (p \wedge q \wedge r) \vee (p \wedge q \wedge \neg r)
\end{aligned}$$

To znamená, že výsledná formula φ_{DNF} je určená poslednou formulou z predchádzajúcej schémy formúl

$$\varphi_{DNF} = (\neg p \wedge \neg q \wedge r) \vee (p \wedge q \wedge r) \vee (p \wedge q \wedge \neg r)$$

Pomocou tohto jednoduchého ilustratívneho príkladu sme ukázali, že každá formula výrokovej logiky môže byť prepísaná do ekvivalentného DNF tvaru pomocou postupnosti elementárnych transformácií z tabuľky 9. Spôsob dôkazu vety 1 môže byť charakterizovaný ako úplná indukcia vzhľadom k syntaktickému stromu danej formuly. Idúc zhora nadol, vždy pomocou vhodnej elementárnej transformácie z tabuľky 9 vykonáme vhodný prepis formuly tak, aby bol bližšie k tvaru DNF formuly. Poznamenajme, že analogický dôkaz môže byť vykonaný aj pre konštrukciu KNF formuly.

Podľa vety 1 ku každej výrokovej formule φ existuje jej DNF a KNF formula, ktorá je s ňou ekvivalentná, $\varphi \sim \varphi_{DNF}$ resp. $\varphi \sim \varphi_{KNF}$. V mnohých prípadoch, tento tvar formuly je zbytočne zložitý, použitím jednoduchých vlastností konjunkcie a disjunkcie (napr. $p \wedge p \equiv p$ a $p \wedge 1 \equiv p$) môžeme formulu podstatne zjednodušiť do ekvivalentného tvaru DNF alebo KNF. Podľa vety 2 formula φ je kontradikcia (tautológia) práve vtedy, ak jej DNF (KNF) tvar obsahuje v každej klauzule dvojicu komplementárnych literálov. Potom taktiež formula φ je *splniteľná* práve vtedy, ak jej DNF tvar obsahuje aspoň jednu klauzulu, ktorá neobsahuje dvojicu komplementárnych literálov.

Príklad 4. Pretransformujte formuly $\varphi = (p \Rightarrow q) \Rightarrow (\neg q \Rightarrow \neg p)$ a $\psi = \neg\varphi$ do DNF resp. do KNF.

$$\varphi = (p \Rightarrow q) \Rightarrow (\neg q \Rightarrow \neg p) \equiv \neg(\neg p \vee q) \vee (q \vee \neg p) \equiv (p \wedge \neg q) \vee (q \vee \neg p)$$

Použitím distribučných zákonov pre disjunkciu a konjunkciu získame tieto ekvivalentné formuly

$$\varphi_{DNF} = (\neg p) \vee (q) \vee (\neg q)$$

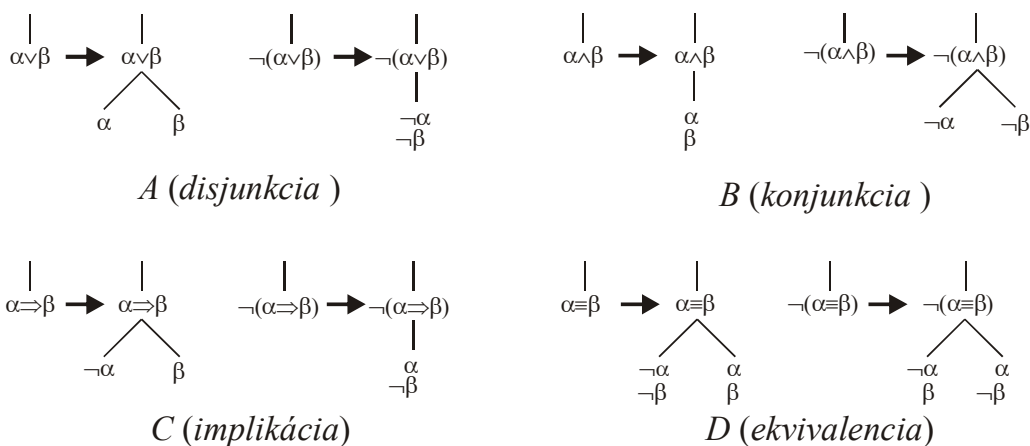
$$\Phi_{KNF} = \left(\underbrace{p \vee \neg p \vee q}_1 \right) \wedge \left(\underbrace{\neg p \vee q \vee \neg q}_1 \right) \equiv 1$$

podobným spôsobom zostrojíme aj ekvivalentné funkcie pre Ψ

$$\Psi_{DNF} = \left(\underbrace{p \wedge \neg p \wedge \neg q}_0 \right) \vee \left(\underbrace{p \wedge q \wedge \neg q}_0 \right) \equiv 0$$

$$\Psi_{KNF} = (p) \wedge (q) \wedge (\neg q)$$

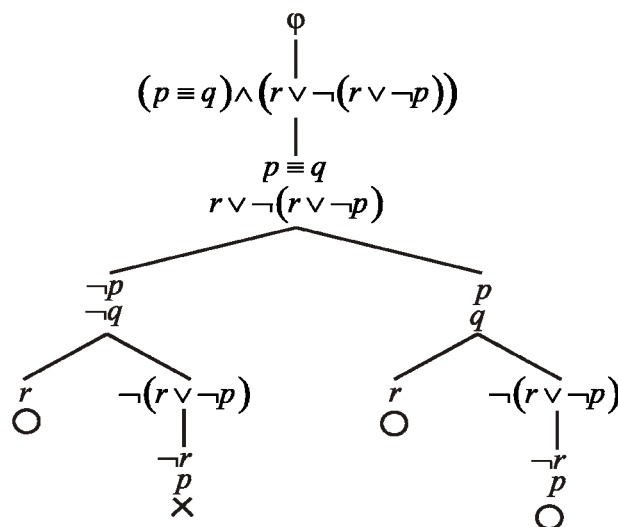
Pre funkciu ϕ jej ekvivalentná funkcia ϕ_{KNF} obsahuje v každej disjunktívnej klauzule dvojicu komplementárnych literálov, čiže funkcia ϕ je tautológia. Podobne, pre funkciu ψ jej ekvivalentná funkcia ψ_{DNF} obsahuje v každej konjunktívnej klauzule dvojicu komplementárnych literálov, čiže funkcia ψ je kontradikcia.



Obr. 5. Základné módy tvorby binárneho stromu v metóde sémantických tabiel. (A) Rozklad disjunkcie a jej negácie, (B) rozklad konjunktie a jej negácie, (C) rozklad implikácie a jej negácie, a (D) rozklad ekvivalencie a jej negácie. Tieto schémy „predlžovania“ sémantického stromu majú svoj základ v Tab. 9, ktorá obsahuje pravidlá pre transformáciu formuly ϕ do DNF tvaru.

3 Metóda sémantických tabiel vo výrokovej logike

Proces transformácie formuly ϕ do DNF tvaru môže byť reprezentovaný koreňovým stromom (nazývaným *sémantické tablo*), ktorý je vytváraný použitím vhodných predĺžení z obr. 5. Aplikáciou týchto pravidiel zostrojíme sémantické tablo (koreňový strom) pre transformáciu formuly ϕ do DNF, pozri obr. 6. Tie vetvy stromu, ktoré obsahujú komplementárne literály sú označené symbolom 'x' a nazývajú sa *uzavreté vetvy*. Podobne, tie vetvy, ktoré neobsahujú komplementárne literály sú označené symbolom 'o' a nazývajú sa *otvorené vetvy*. Ak sémantické tablo obsahuje len uzavreté vetvy, potom sa nazýva *uzavreté sémantické tablo*, v opačnom prípade, ak obsahuje aspoň jednu otvorenú vetvu, potom sa nazýva *otvorené sémantické tablo*. Sémantické tablo priradené formule ϕ je označené $\mathcal{T}(\phi)$.



Obr. 6. Sémantické tablo pre formulu $\varphi = (p \equiv q) \wedge (r \vee \neg(r \vee \neg p))$ z príkladu 5. Metóda presne kopíruje transformáciu formuly do DNF tvaru (jednotlivé kroky tejto transformácie sú uvedené v príklade 3). Koncový vrchol označený symbolom 'X' znamená, že príslušná vetva stromu je uzavretá a nepravdivá (obsahuje komplementárne literály). Koncový vrchol označený symbolom 'O' znamená, že príslušné vetvy sú otvorené a môžu byť jednoznačne priradené konjunktívnym klauzulám z $\varphi_{DNF} = (\neg p \wedge \neg q \wedge r) \vee (p \wedge q \wedge r) \vee (p \wedge q \wedge \neg r)$. V tomto prípade existujú špecifikácie premenných $\tau_1 = (p/0, q/0, r/1)$, $\tau_2 = (p/1, q/1, r/1)$ a $\tau_3 = (p/1, q/1, r/0)$, pre ktoré je formula pravdivá.

Príklad 5. Zostrojte sémantické tablo pre formulu $\varphi = (p \equiv q) \wedge (r \vee \neg(r \vee \neg p))$. Zostrojené sémantické tablo (pozri obr. 6) je otvorené, z čoho priamo vyplýva, že formula φ je splniteľná.

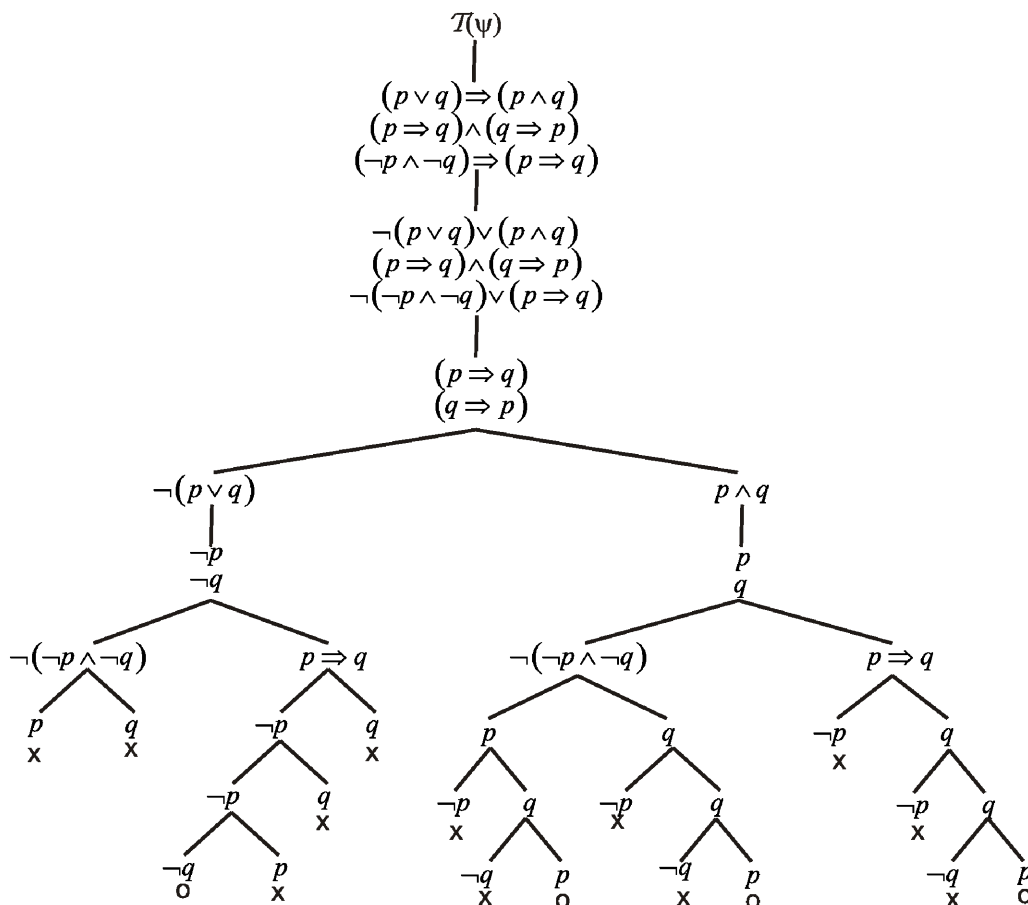
Na základe platnosti vety 2 môžeme formulovať vetu platnú pre sémantické tablá

Veta 3.

- (1) Ak $\mathcal{T}(\varphi)$ je uzavreté (otvorené) sémantické tablo, potom formula φ je kontradikcia (splniteľná).
- (2) Ak $\mathcal{T}(\neg\varphi)$ je uzavreté sémantické tablo, potom formula φ je tautológia; v prípade, že sémantické tablo obsahuje aspoň jednu otvorenú vetvu, potom formula φ nie je tautológia.

Druhá vlastnosť tejto vety úzko súvisí s prvou vlastnosťou. Keď chceme zistiť, či formula φ je tautológia, potom zostrojíme sémantické tablo $\mathcal{T}(\neg\varphi)$, ak je toto tablo uzavreté, potom formula $\neg\varphi$ podľa prvej vlastnosti je kontradikcia, alebo formula φ je tautológia. Môžeme teda konštatovať, že v rámci techniky sémantických tabiel

tautologičnosť formuly φ sa zistí nepriamo tak, že ak pre formulu $\neg\varphi$ zistíme je kontradikčnosť, potom pôvodná formula φ je tautológiou.



Obr. 7. Sémantické tablo $T(\psi)$ z príkladu 6 pre konjunkciu $\psi = \varphi_1 \wedge \varphi_2 \wedge \varphi_3$, kde formuly z konjunkcie sú z teórie $T = \{(p \vee q) \Rightarrow (p \wedge q), (p \Rightarrow q) \wedge (q \Rightarrow p), (\neg p \wedge \neg q) \Rightarrow (p \Rightarrow q)\}$. Pomocou otvorených vetví môžeme zostrojiť model $M(T) = \{\tau_1 = (p/1, q/1), \tau_2 = (p/0, q/0)\}$.

Príklad 6. Pre teóriu $T = \{(p \vee q) \Rightarrow (p \wedge q), (p \Rightarrow q) \wedge (q \Rightarrow p), (\neg p \wedge \neg q) \Rightarrow (p \Rightarrow q)\}$ zostrojte pomocou sémantického tabla model $M(T)$. Ak jednotlivé formuly z tejto teórie označíme symbolmi φ_1, φ_2 , resp. φ_3 , potom ich konjunkcia $\psi = \varphi_1 \wedge \varphi_2 \wedge \varphi_3$ tvorí koreň sémantického tabla zobrazeného na obr. 7. Tablo obsahuje štyri otvorené cesty, pričom tri obsahujú rovnaké literály, týmto otvoreným

cestám môžeme priradiť interpretácie $\tau_1 = (p/1, q/1)$ a $\tau_2 = (p/0, q/0)$, potom model tejto teórie obsahuje tieto dve interpretácie

$$M(\mathbf{T}) = \{\tau_1 = (p/1, q/1), \tau_2 = (p/0, q/0)\}$$

Veta 4.

- (1) Teória $\mathbf{T} = \{\varphi_1, \varphi_2, \dots, \varphi_n\}$ je konzistentná práve vtedy, ak sémantické tablo $\mathcal{T}(\varphi_1 \wedge \dots \wedge \varphi_n)$ je otvorené. Model $M(\mathbf{T})$ teórie $\mathbf{T} = \{\varphi_1, \varphi_2, \dots, \varphi_n\}$ je tvorený interpretáciami literálov z otvorených vetiev.
- (2) Teória $\mathbf{T} = \{\varphi_1, \varphi_2, \dots, \varphi_n\}$ je nekonzistentná práve vtedy, ak sémantické tablo $\mathcal{T}(\varphi_1 \wedge \dots \wedge \varphi_n)$ je uzavreté.

V čom spočíva výhoda sémantického tabla pred formálnymi manipuláciami s formulou φ , ktoré ju transformujú do DNF tvaru? Aplikácia distribučných zákonov pri úprave formuly do DNF tvaru je pomerne náročnou operáciou a preto je výhodné prenechať ju diagramatickej metóde konštrukcie sémantického tabla. Druhý, nemenej dôležitý aspekt konštrukcie je uzavretie tej vetvy, ktorá obsahuje komplementárne literály. Predlžovanie takejto vetvy už neprináša žiadnu novú skutočnosť z pohľadu toho, či daná formula je kontradikciou alebo je splniteľná. Prípadné ďalšie výskyty dvojíc komplementárnych literálov už nemení nič na skutočnosti, že daný konjunkt v DNF je nepravdivý. Preto táto možnosť „okamžitého“ uzavretia vetvy pri konštrukcii sémantického tabla obvykle patrí medzi významné zjednodušenia jeho konštrukcie, celé veľké podstromy v sémantickom table môžu byť ignorované ako nevýznamné.

Vo výrokovej logike je dokázané, že formula φ je logickým dôsledkom množiny predpokladov (teórie) $\mathbf{T} = \{\varphi_1, \varphi_2, \dots, \varphi_n\}$, $\mathbf{T} \vdash \varphi$, vtedy a len vtedy, ak $\vdash \varphi_1 \wedge \dots \wedge \varphi_n \Rightarrow \varphi$, t. j. keď táto formula logicky vyplýva z axiomatického systému výrokovej logiky. Pomocou sémantických tabiel je problém $\vdash \varphi_1 \wedge \dots \wedge \varphi_n \Rightarrow \varphi$ formulovateľný veľmi efektívne pomocou vetvy:

Veta 5.

Formula φ je logickým dôsledkom množiny formúl $\mathbf{T} = \{\varphi_1, \varphi_2, \dots, \varphi_n\}$, $\mathbf{T} \vdash \varphi$, vtedy a len vtedy, ak sémantické tablo $\mathcal{T}(\neg\psi)$, kde $\neg\psi = \neg(\varphi_1 \wedge \dots \wedge \varphi_n \Rightarrow \varphi) \equiv (\varphi_1 \wedge \dots \wedge \varphi_n \wedge \neg\varphi)$, je uzavreté.

Táto veta je priamym dôsledkom skutočnosti, že metóda sémantického tabla je vlastne špecifický spôsob prepisu formuly do DNF pomocou operácií, ktoré sú ekvivalencie (napr. De Morganove relácie, distributívne vzťahy medzi disjunkciou a konjunkciou a pod.). Čiže môžeme konštatovať, že pre danú formulu φ je prepis na φ_{DNF} čisto syntaktický prístup, ktorý nepoužíva žiadne sémantické interpretácie.

Príklad 7. Koná sa oslava, Ján, Júlia, Klára a Štefan sú potenciálni účastníci tejto oslavy. K tomu, aby sme mohli zapísať študovaný problém pomocou formúl výrokovej logiky, zavedieme tejto štyri výroky:

| | |
|-----|------------------------|
| p | Ján pôjde na oslavu |
| q | Júlia pôjde na oslavu |
| r | Klára pôjde na oslavu |
| s | Štefan pôjde na oslavu |

Avšak, ako to často býva, ich účasť je ohraničenú tromi podmienkami

| | |
|------------------------------|--|
| $p \vee q$ | Ján alebo Júlia pôjdu na oslavu. |
| $q \Rightarrow (r \wedge s)$ | Ak Júlia pôjde na oslavu, potom na oslavu pôjde tak Klára ako aj Štefan. |
| $\neg p \Rightarrow s$ | Ak nepôjde na oslavu Ján, potom pôjde na oslavu Štefan. |

Naším cieľom je zistiť, za akých podmienok sa Štefan zúčastní oslavy, t.j. budeme riešiť problém, kedy z teórie

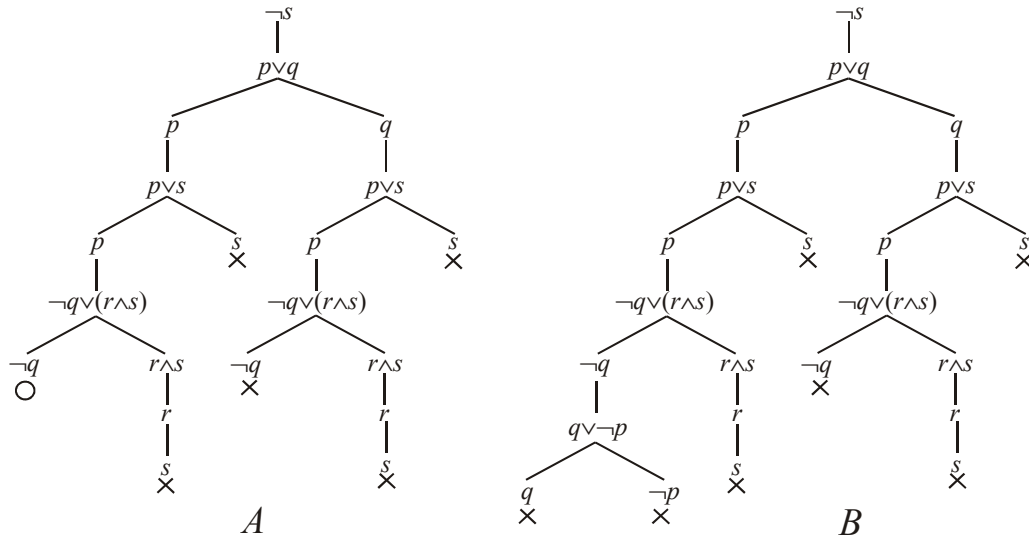
$$T = \{p \vee q, q \Rightarrow (r \wedge s), \neg p \Rightarrow s\}$$

vyplýva, že Štefan sa zúčastní oslavy, $T \vdash s$, alebo ekvivalentné vyjadrené takto pomocou formuly (pozri vetu 5)

$$((p \vee q) \wedge (q \Rightarrow (r \wedge s)) \wedge (\neg p \Rightarrow s)) \Rightarrow s$$

Túto formulu môžeme jednoducho prepísať pomocou jej negácie do tvaru

$$(p \vee q) \wedge (q \Rightarrow (r \wedge s)) \wedge (\neg p \Rightarrow s) \wedge \neg s \tag{30}$$



Obr. 8. (A) Otvorené sémantické tablo zostrojené pre teóriu T špecifikovanú (30) z príkladu 7, (B) uzavreté sémantické tablo zostrojené pre rozšírenú teóriu (31) z rovnakého príkladu.

Aplikujeme metódu sémantických tabiel k analýze kontradikčnosti formuly (30), aby sme zjednodušili generované sémantické tablo, v prvom kroku odstránime z formuly implikácie

$$(p \vee q) \wedge (\neg q \vee (r \wedge \neg s)) \wedge (p \vee s) \wedge \neg s$$

Príslušný strom sémantického tabla \mathcal{T} je znázornený na obr. 8, diagram A. Toto sémantické tablo nie je uzavreté, obsahuje jednu vetvu, ktorá je otvorená, čiže formula (30) nie je kontradikciou, je len splniteľná. To znamená, že neplatí $\mathbf{T} \vdash s$, t.j. výrok s nie je logickým dôsledkom teórie \mathbf{T} .

Môžeme si položiť zaujímavú otázku, ako rozšíriť teóriu \mathbf{T} na novú teóriu \mathbf{T}' (kde $\mathbf{T} \subset \mathbf{T}'$), aby výrok s už bol dôsledkom tejto novej teórie. K tomuto účelu nám dobre posluží sémantické tablo \mathcal{T} z obr. 8, diagram A. Naším cieľom bude také rozšírenie teórie \mathbf{T} , aby otvorené vetvy tabla sa stali uzavretými. Teóriu rozšírime o tento výrok

| | |
|-----------------------------|---|
| $\neg q \Rightarrow \neg p$ | Ak Júlia nepôjde na oslavu, potom aj Ján nepôjde na oslavu. |
|-----------------------------|---|

Rozšírená teória má tvar

$$\mathbf{T}' = \{p \vee q, q \Rightarrow (r \wedge \neg s), \neg p \Rightarrow s, \neg q \Rightarrow \neg p\} \quad (31)$$

potom formula (30) je rozšírená o príslušnú konjunkciu, ktorá je priradená novej podmienke z rozšírenej teórie \mathbf{T}'

$$(p \vee q) \wedge (q \Rightarrow (r \wedge \neg s)) \wedge (\neg p \Rightarrow s) \wedge (\neg q \Rightarrow \neg p) \wedge \neg s$$

Sémantické tablo pre túto formulu je znázornené na obr. 8, diagram B. Z obrázku vidíme, že sémantické tablo pre takto rozšírenú teóriu (31) sa už sa stalo uzavretým, $\mathbf{T}' \vdash s$.

Z tohto jednoduchého príkladu vyplýva, že metóda sémantického tabla je vhodným prístupom vtedy, keď sa snažíme rozšíriť danú teóriu tak, aby nové rezultujúce tablo bolo už uzavreté, zo stromovej štruktúry tabla sa dá jednoducho odvodiť, akým spôsobom sa má vykonať také rozšírenie teórie, aby sémantické tablo bolo uzavreté. Detaily tohto prístupu budú špecifikované v odseku 3.2 v rámci štúdia abdukcie vo výrokovej logike.

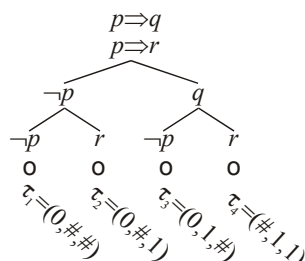
3.1 Konštrukcia tautologického vyplývania pomocou sémantických tabiel

Prvú úlohu, ktorú budeme riešiť v tomto aplikačnom odseku kapitoly, bude úloha zostrojiť pomocou sémantických tabiel model $M(\mathbf{T})$, ktorý je množinou pravdivých interpretácií formúl z teórie $\mathbf{T} = \{\varphi_1, \varphi_2, \dots, \varphi_n\}$

$$M(\mathbf{T}) = \{\tau_1, \tau_2, \dots, \tau_a\} \quad (32)$$

Danú úlohu rieši veta 4, podľa ktorej interpretácie z množiny $M(\mathbf{T}) = \{\tau_1, \tau_2, \dots, \tau_a\}$ sú určené pomocou otvorených vetví sémantického tabla

$\mathcal{T}(\varphi_1 \wedge \varphi_2 \wedge \dots \wedge \varphi_n)$, pričom každej otvorenej vetve môžeme priradiť interpretáciu $\tau \in M(\mathbf{T})$, pre ktorú sú všetky literály na danej vetve pravdivé.



Obr. 9. Sémantické tablo $\mathcal{T}((p \Rightarrow q) \wedge (q \Rightarrow r))$ pre teóriu $\mathbf{T} = \{p \Rightarrow q, p \Rightarrow r\}$, v tomto prípade každá vetva je otvorená, čiže môžeme k nej priradiť interpretáciu τ_i , pre $i = 1, 2, 3, 4$.

Príklad 8. Uvažujme teóriu $\mathbf{T} = \{p \Rightarrow q, p \Rightarrow r\}$, našim cieľom bude zostrojiť množinu $M(\mathbf{T})$. Podľa vety 4 zostrojíme sémantické tablo pre konjunkciu formúl z teórie \mathbf{T} , ktoré je znázornené na obr. 9. Teória \mathbf{T} má štyri interpretácie, pre ktoré sú všetky formuly pravdivé

$$\tau_1 = (0, \#, \#)$$

$$\tau_2 = (0, \#, 1)$$

$$\tau_3 = (0, 1, \#)$$

$$\tau_4 = (\#, 1, 1)$$

kde symbol '#' znamená ľubovoľný znak 0/1. Ľahko sa presvedčíme, že pre takto špecifikované interpretácie, formuly z teórie \mathbf{T} sú pravdivé.

Ak poznáme množinu $M(\mathbf{T})$, potom môžeme upriamiť našu pozornosť na konštrukciu formuly φ , ktorá je pravdivá pre každú interpretáciu $\tau \in M(\mathbf{T})$, t. j. je tautologickým dôsledkom teórie \mathbf{T} . Definujme premenné pre danú interpretáciu $\tau = (\tau_1, \tau_2, \dots, \tau_n) \in M(\mathbf{T})$

$$p_i^{(\tau)} = \begin{cases} p_i & (\text{ak } \tau_i = 1) \\ \neg p_i & (\text{ak } \tau_i = 0) \\ 1 & (\text{ak } \tau_i = \#) \end{cases} \quad (33)$$

Potom môžeme definovať konjunktívnu klauzulu (pozri (24) a (25))

$$\Psi_\tau(p_1, p_2, \dots, p_n) = p_1^{(\tau)} \wedge p_2^{(\tau)} \wedge \dots \wedge p_n^{(\tau)} \quad (34)$$

Pomocou tejto klauzuly definujme výslednú funkciu

$$\varphi(p_1, p_2, \dots, p_n) = \bigvee_{\tau \in M(\mathbf{T})} p_1^{(\tau)} \wedge p_2^{(\tau)} \wedge \dots \wedge p_n^{(\tau)} \quad (35)$$

ktorá je pravdivá pre každú interpretáciu $\tau \in M(\mathbf{T})$

$$(\forall \tau \in M(\mathbf{T})) (val_{\tau}(\varphi) = 1) \quad (36)$$

Týmto sme dokázali, že funkcia (35) je logický dôsledok teórie \mathbf{T} , t. j. $\mathbf{T} \vdash \varphi$.

Veta 6.

Ak teória $\mathbf{T} = \{\varphi_1, \varphi_2, \dots, \varphi_n\}$ je konzistentná, t. j. $M(\mathbf{T}) \neq \emptyset$, potom môžeme zostrojiť pomocou (35) takú formulu φ , ktorá je logickým dôsledkom teórie \mathbf{T} , $\mathbf{T} \vdash \varphi$.

Pomocou tejto vety môžeme teda zostrojiť „minimálny tvar“ formuly φ , ktorá tautologicky vyplýva z teórie \mathbf{T} . Môžeme rôznym spôsobom rozšíriť túto formulu φ do tvaru φ_{ext} , ktorý taktiež tautologicky vyplýva z teórie \mathbf{T}

$$\varphi_{ext} = \varphi \vee \chi \quad (37)$$

kde χ je ľubovoľná formula. Ľahko sa presvedčíme o tom, že aj rozšírená formula φ_{ext} je platná pre ľubovoľné χ . Pre každé $\tau \in M(\mathbf{T})$ a pre každú formulu $\varphi_i \in \mathbf{T}$ platí $val_{\tau}(\varphi_i) = val_{\tau}(\varphi) = val_{\tau}(\varphi_{ext})$.

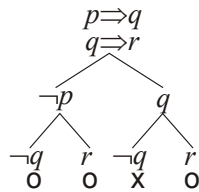
Príklad 9. Budeme pokračovať v ďalšom riešení príkladu 8. Pomocou formuly (35) a interpretácií τ_i , pre $i = 1, 2, 3, 4$, ktoré boli zostrojené v príklade 8 zostrojíme formulu

$$\begin{aligned} \varphi(p, q, r) &= (\neg p) \vee (\neg p \wedge r) \vee (\neg p \wedge q) \vee (q \wedge r) \\ &= \left((\neg p) \wedge \underbrace{(1 \vee r \vee q)}_1 \right) \vee (q \wedge r) \\ &= (\neg p) \vee (q \wedge r) = (p \Rightarrow q \wedge r) \end{aligned}$$

Táto formula tautologicky vyplýva z predpokladov obsiahnutých v teórii $\mathbf{T} = \{p \Rightarrow q, p \Rightarrow r\}$

$$\{p \Rightarrow q, p \Rightarrow r\} \models (p \Rightarrow q \wedge r)$$

Príklad 10. Majme teóriu $\mathbf{T} = \{p \Rightarrow q, q \Rightarrow r\}$, našou úlohou bude nájsť takú formulu φ , ktorá tautologicky vyplýva z tejto teórie, $\mathbf{T} \models \varphi$. Sémantické tablo pre teóriu $\mathbf{T} = \{\varphi_1 = p \Rightarrow q, \varphi_2 = q \Rightarrow r\}$ je znázornené na obr. 10.



Obr. 10. Sémantické tablo pre teóriu $\mathbf{T} = \{p \Rightarrow q, q \Rightarrow r\}$, ktoré obsahuje tri otvorené vetvy.

Jednotlivým otvoreným vetvám tabla z obr. 10 podľa (35) môžeme priradiť tieto konjunktívne klauzuly

$$\begin{aligned}\varphi_{\tau_1} &= \neg p \wedge \neg q \\ \varphi_{\tau_2} &= \neg p \wedge r \\ \varphi_{\tau_3} &= q \wedge r.\end{aligned}$$

Použitím (35) dostaneme

$$\varphi = (\neg p \wedge \neg q) \vee (\neg p \wedge r) \vee (q \wedge r) \equiv \left(\neg p \wedge \underbrace{(q \Rightarrow r)}_{\varphi_2} \right) \vee \left(\underbrace{(p \Rightarrow q)}_{\varphi_1} \wedge r \right) = (\neg p \wedge \varphi_2) \vee (\varphi_1 \wedge r)$$

Pretože požadujeme pri definícii tautologického vyplývania, aby formuly φ_1, φ_2 boli pravdivé pre každé $\tau \in M(\mathbf{T})$, potom formulu φ môžeme zjednodušiť

$$\varphi = (\neg p \wedge 1) \vee (1 \wedge r) \equiv p \Rightarrow r$$

Týmto sme dokázali, že z teórie \mathbf{T} tautologicky vyplýva $p \Rightarrow r$, čiže

$$\{p \Rightarrow q, q \Rightarrow r\} \models (p \Rightarrow r)$$

3.2 Sémantické tablá a problém abdukcie vo výrokovvej logike

Paradigma abdukcie ako špeciálneho typu nededuktívneho usudzovania bola zavedená americkým filozofom a logikom Charlesom Peirceom (pozri ref. [18]). V jeho pôvodnej definícii usudzovania rozlišoval klasický typ usudzovania: dedukciu a dva neklasické typy usudzovania: indukciu a abdukciu. Abdukciu vyjadril nasledujúcou schémou usudzovania

$$\frac{\begin{array}{l} q \\ p \Rightarrow q \end{array}}{p}$$

ktorá v rámci klasického deduktívneho usudzovania je evidentne nekorektná (napr. formula $(q \wedge (p \Rightarrow q)) \Rightarrow p$ nie je tautológia a tvorí základ „zlozvyku“ usudzovania nazývaného zdôraznenie dôsledku). Niekedy sa takto formulovaná abdukcia nazýva „inverzný modus ponens“. Peircov ilustračný príklad abdukcie má tvar

$$\frac{\begin{array}{l} \text{tieto fazule sú biele} \\ \text{každá fazuľa z tejto nádoby je biela} \end{array}}{\text{tieto fazule sú z tejto nádoby}}$$

Alternatívny pohľad na abdukciu je ten, že sa jedná o proces tvorby hypotézy, ktorý vysvetľuje pozorovanie q na základe známeho poznatku (teórie) reprezentovaného implikáciou $p \Rightarrow q$.

Moderný prístup k špecifikácii abdukcie [1,17] spočíva v jej formulovaní veľmi blízkeho deduktívnemu usudzovaniu. Nech $\mathbf{T} = \{\varphi_1, \varphi_2, \dots, \varphi_n\}$ je konzistentná teória,

$M(\mathbf{T}) \neq \emptyset$, a nech φ je pozorovanie, ktoré chceme vysvetliť pomocou danej teórie \mathbf{T} , avšak $\mathbf{T} \not\models \varphi$, t. j. $M(\mathbf{T}) \not\subseteq M(\varphi)$. Z týchto dôvodov v rámci množiny hypotéz – formúl $\mathbf{H} = \{\alpha_1, \alpha_2, \dots, \alpha_p\}$ hľadáme takú hypotézu $\alpha \in \mathbf{H}$, ktorá spolu s teóriou deduktívne vysvetľuje pozorovanie φ

$$\mathbf{T} \cup \{\alpha\} \models \varphi \quad (38a)$$

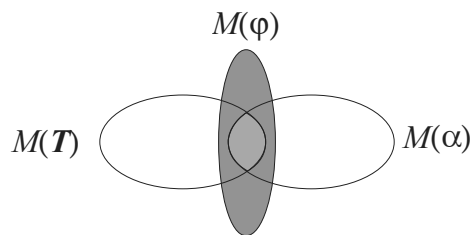
čo je ekvivalentné množinovo-teoretickej relácii (pozri obr. 11)

$$M(\mathbf{T}) \cap M(\alpha) \subseteq M(\varphi) \quad (38b)$$

kde predpokladáme, že aj rozšírená teória \mathbf{T} o hypotézu α je konzistentná

$$M(\mathbf{T} \cup \{\alpha\}) = M(\mathbf{T}) \cap M(\alpha) \neq \emptyset \quad (38c)$$

Poznamenajme, že množinovo-teoretická relácia (38b) tvorí teoretický základ nášho prístupu k riešeniu takto formulovanej abdukcie, t. j. hľadaniu hypotézy α . Umožňuje nám „algebraizovať“ proces abdukcie, kde centrálnu úlohu hrá relácia (38b), pričom dominantnú úlohu v tomto procese bude hrať technika sémantických tabiel.



Obr. 11. Diagramatické znázornenie množinovej relácie $M(\mathbf{T}) \cap M(\alpha) \subseteq M(\varphi)$, ktorá tvorí podmienku pre existenciu tautologického vyplývania $\mathbf{T} \cup \{\alpha\} \models \varphi$.

Na záver tejto úvodnej časti o abdukcii uvedieme niekoľko poznámok o tom, prečo abdukcia patrí medzi *nededuktívne módy usudzovania*, aj keď relácia (38a) má striktno deduktívny charakter. Hlavný dôvod k tomuto odlíšeniu spočíva v skutočnosti, že v počiatočnej etape usudzovania stojíme pred problémom výberu hypotézy α z množiny možných hypotéz, ktoré zachovávajú konzistentnosť teórie $\alpha \in \mathbf{H}$ a tiež umožňujú tautologické vyplývanie (38a) pozorovania φ z rozšírenej teórie $\mathbf{T} \cup \{\alpha\}$. Tento výber sa deje mimologickými prostriedkami, kde sa používajú rôzne heuristiky o jednoduchosti (a ekonomičnosti). V tejto súvislosti pripomeňme pravidlo *Occamovej britvy*, podľa ktorého „*dôvody nesmú byť zložitejšie, než ako je potrebné*“, moderná alternatívna formulácia tohto pravidla je „*najjednoduchšie vysvetlenie pozorovaného javu, ktoré je konzistentné so súčasným stavom vedy, je s najväčšou pravdepodobnosťou korektné vysvetlenie*“. Ako ilustratívny príklad použitia tejto hypotézy uvedieme nasledujúci negatívny príklad:

Vrátil som sa domov neskoro v noci, aj keď som sľúbil manželke, že prídem ešte pred TV správami. Nepríjemne zapáchajúci cigaretovým dymom a alkoholom, stojím pred problémom ako vysvetlím manželke túto skutočnosť, že som prišiel domov tak neskoro. Porušenie zásady Occamovej britvy je

vysvetlenie, že podvečerom na streche fakulty pristálo UFO a ja som bol vyzvaný dekanom fakulty, aby som s nimi komunikoval, čo sa nepríjemne dlho pretiahlo a bol som s nimi prinútený piť víno a fajčiť cigarety, aj keď už som dávno prestal fajčiť a piť.

Môžeme teda konštatovať, že problém výberu hypotézy je minimalizačný problém,

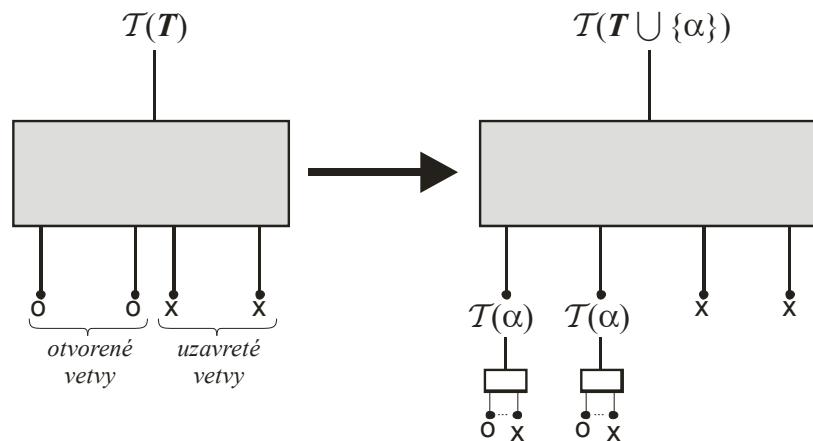
$$\alpha_{opt} = \arg \min_{\alpha \in H} f(\alpha) \quad (39)$$

kde $f(\alpha)$ je „účelová funkcia“, ktorá vyhodnocuje „ekonomičnosť“ danej hypotézy α , ktorá je založená na mimologických prostriedkoch. Práve riešenie tohto problému (39) je hlavným dôvodom toho, prečo je abdukcia pokladaná za nededuktívny mód usudzovania, aj keď riešenie relácie (38a) je už striktno deduktívne.

V našich ďalších úvahách o abdukcii budeme potrebovať vlastnosť sémantických tabiel, ktorá súvisí s ich predĺžením o nový poznatok α . Táto vlastnosť je formulovaná do nasledujúcej definície.

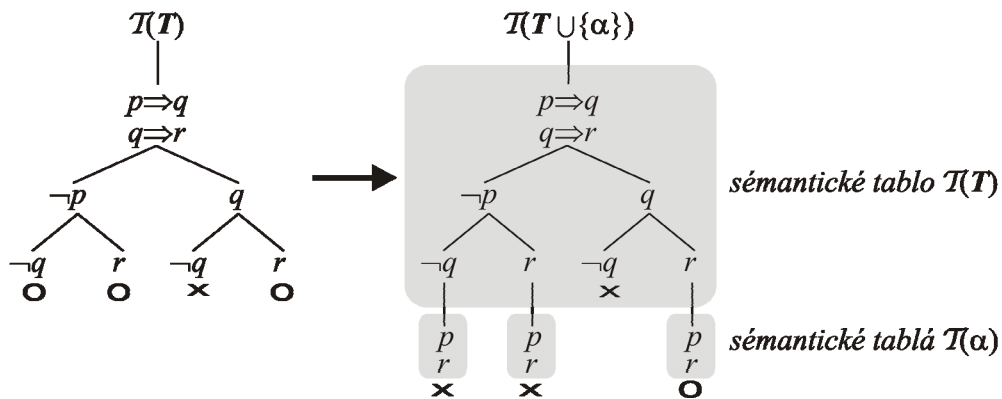
Definícia 3.

Nech $\mathcal{T}(T \cup \{\alpha\})$ je sémantické tablo s koreňovým vrcholom, ktorý je tvorený konjunkciou elementov z teórie T a formuly α , t. j. $\Theta = \varphi_1 \wedge \varphi_2 \wedge \dots \wedge \varphi_n \wedge \alpha$. Hovoríme, že toto sémantické tablo je vytvorené predĺžením otvorených vetiev tabla $\mathcal{T}(T)$ o formulu α (pozri obr. 12).

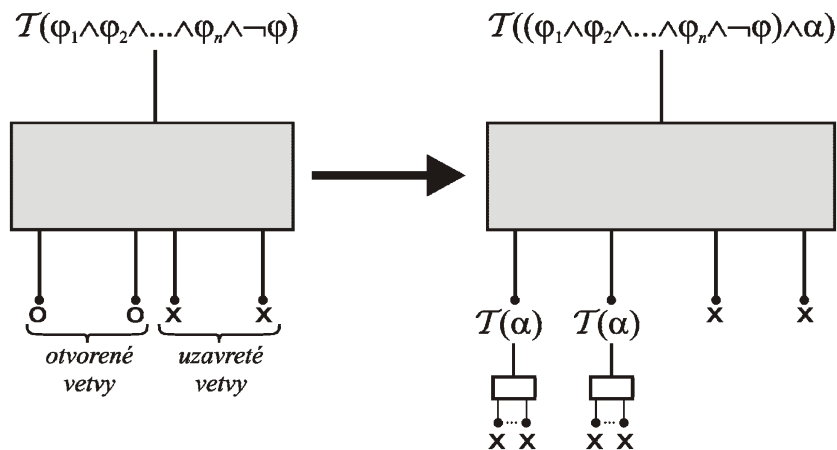


Obr. 12. Sémantické tablo $\mathcal{T}(T \cup \{\alpha\})$ vzniklo z tabla $\mathcal{T}(T)$ predĺžením otvorených vetiev o sémantické tablá $\mathcal{T}(\alpha)$, pričom vytvorené vetvy môžu byť tak otvorené ako aj uzavreté.

Príklad 11. Nech teória $T = \{p \Rightarrow q, q \Rightarrow r\}$ a $\alpha = p \wedge r$. Podľa definície 3 sémantické tablo $\mathcal{T}(T \cup \{\alpha\})$ je vytvorené predĺžením tabla $\mathcal{T}(T)$ o tablo $\mathcal{T}(\alpha)$, pričom predlžujeme len otvorené vetvy tabla $\mathcal{T}(T)$, pozri obr. 12 a 13.



Obr. 13. Znázornenie predĺženia sémantického tabla z príkladu 11. Tento “modulárny” prístup ku konštrukcii sémantického tabla $\mathcal{T}(T')$ pre teóriu špecifikovanú ako zjednotenie dvoch podteórií, $T' = T \cup \{\alpha\}$, umožňuje efektívne analyzovať procesy usudzovania založené na rozširovaní pôvodnej teórie o nové poznatky.

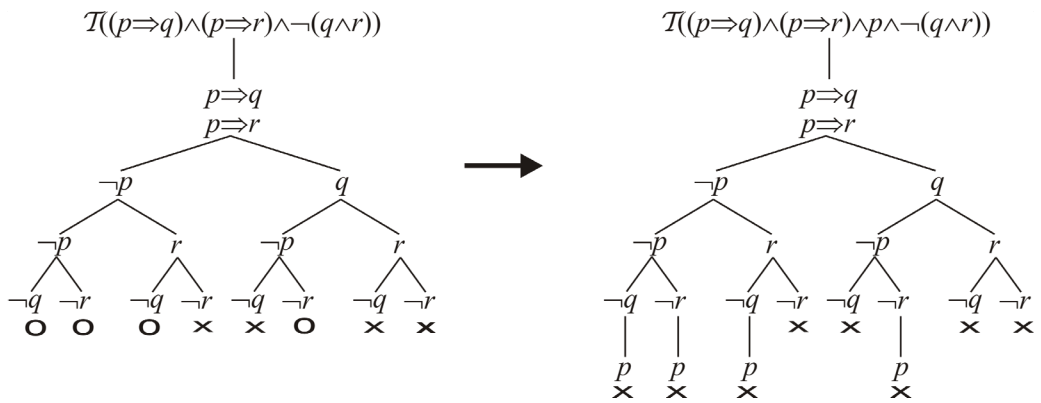


Obr. 14. Použitie techniky predĺženia sémantického tabla k riešeniu problému abdukcie, t. j. hľadaniu takej formuly α , aby platilo $T' = T \cup \{\alpha\}$. Hľadaná formula α musí mať taký tvar, aby rezultujúce rozšírené sémantické tablo $\mathcal{T}(-\Theta)$ bolo uzavreté.

Pomocou techniky predĺženia sémantického tabla budeme študovať abdukciu, aplikovaním predĺženia upravíme reláciu $T \not\models \varphi$ tak, aby poznatok φ tautologicky vyplýval z rozšírenej teórie $T \cup \{\alpha\}$, t. j. $T \cup \{\alpha\} \models \varphi$. Nech $T = \{\varphi_1, \varphi_2, \dots, \varphi_n\}$ je východisková teória, ktorá obsahuje n poznatkov reprezentovaných formulami $\varphi_1, \varphi_2, \dots, \varphi_n$, pričom platí $T \not\models \varphi$, alebo formula $\varphi_1 \wedge \varphi_2 \wedge \dots \wedge \varphi_n \wedge \neg\varphi$ nie je

kontradikcia². Našou snahou bude „modelovať“ dodatočný poznatok α tak, aby „rozšírená“ formula $\neg\Theta = \varphi_1 \wedge \varphi_2 \wedge \dots \wedge \varphi_n \wedge \alpha \wedge \neg\varphi$ už bola kontradikcia, t. j. aby príslušné sémantické tablo $\mathcal{T}(\neg\Theta)$ malo všetky vetvy uzavreté. K tejto konštrukcii použijeme techniku rozšírenia sémantického tabla špecifikovanú definíciou 3, ktorej konkretizácia pre abdukciu je znázornená na obr. 14.

Príklad 12. Nech $T = \{p \Rightarrow q, p \Rightarrow r\}$ je konzistentná teória a $\varphi = q \wedge r$ je požadovaný dôsledok z tejto teórie. Vytvoríme formulu $\neg\Theta = \varphi_1 \wedge \dots \wedge \varphi_n \wedge \neg\varphi \equiv (p \Rightarrow q) \wedge (p \Rightarrow r) \wedge \neg(q \wedge r)$, sémantické tablo $\mathcal{T}(\neg\Theta)$ má otvorené vetvy, z čoho plynie, že $\varphi = q \wedge r$ nie je tautologickým dôsledkom teórie $T = \{p \Rightarrow q, p \Rightarrow r\}$, pozri obr. 15.



Obr. 15. Ilustračný príklad abdukcie, keď sémantické tablo $\mathcal{T}(\neg\Theta)$ je rozšírené o abdukčný predpoklad – hypotézu $\alpha = p$, ktorý uzatvára sémantické tablo.

Ak rozšírime pôvodnú teóriu o nový predpoklad p , potom z $T \cup \{p\}$ už vyplýva požadovaný dôsledok $\varphi = q \wedge r$. Z tohto elementárneho príkladu vyplýva, že pomocou techniky sémantických tabiel môžeme ľahko rozšíriť pôvodné predpoklady z teórie T o nový poznatok α tak, aby z rozšírenej teórie $T \cup \{\alpha\}$ vyplýval požadovaný predpoklad $T \cup \{\alpha\} \models \varphi$, pričom $T \not\models \varphi$.

3.3 Vzťah medzi prirodzenou dedukciou a sémantickými tablami

Cieľom tejto podkapitoly je ukázať, že medzi prirodzenou dedukciou [5,12,21] a sémantickými tablami existuje úzka súvislosť. Podľa vety 3 platí, že ak sémantické

² Poznamenajme, že ak platí $T \vdash \varphi$, potom formula $\varphi_1 \wedge \varphi_2 \wedge \dots \wedge \varphi_n \Rightarrow \varphi$ je tautológia a teda jej negácia $\varphi_1 \wedge \varphi_2 \wedge \dots \wedge \varphi_n \wedge \neg\varphi$ je kontradikcia.

tablo $\mathcal{T}(\neg\varphi)$ je uzavreté, potom formula φ je tautológia. Sémantické tablo $\mathcal{T}(\neg\varphi)$ môže slúžiť v inverznom poradí ako návod pre dôkaz formuly φ pomocou prirodzenej dedukcie, pričom potrebné vstupné výrokové premenné a ich negácie sú tvorené pomocou triviálnych kontradikcií typu $\neg p \wedge p \equiv \neg(p \Rightarrow p)$. Prirodzenú dedukciu budeme realizovať pomocou diagramov (pozri obr. 16), takže pri konštrukcii tejto diagramatickej reprezentácie prirodzenej dedukcie budeme používať spôsob, ktorý je v podstate inverziou sémantického tabla. Použité schémy usudzovania pre tvorbu diagramatickej reprezentácie prirodzenej dedukcie sú uvedené v tab. 10. V tomto prípade sa používajú len tie najjednoduchšie schémy usudzovania, ktoré pomocou konjunkcie, disjunktie a implikácie vytvárajú z dvoch aktuálnych podformúl φ a ψ dôkazu novú formulu $\varphi \clubsuit \psi$.

Tabuľka 10. Elementárne schémy usudzovania pre sémantické tablá a dôkaz pomocou prirodzenej dedukcie založenej na sémantickom table

| Logická spojka | sémantické tablá | prirodzená dedukcia |
|----------------|--|--|
| Negácia | $\begin{array}{c} \neg\neg p \\ \downarrow \\ p \end{array}$ | $\begin{array}{c} p \\ \downarrow \\ \neg\neg p \end{array}$ |
| Implikácia | $\begin{array}{c} \neg(p \Rightarrow q) \\ \downarrow \\ p \\ \neg q \end{array}$ $\begin{array}{c} p \Rightarrow q \\ \swarrow \downarrow \searrow \\ \neg p \quad q \end{array}$ | $\begin{array}{c} \neg p \quad q \\ \swarrow \downarrow \searrow \\ p \Rightarrow q \end{array}$ $\begin{array}{c} \neg p \quad q \\ \swarrow \downarrow \searrow \\ p \Rightarrow q \end{array}$ $\begin{array}{c} p \quad \neg q \\ \swarrow \downarrow \searrow \\ \neg(p \Rightarrow q) \end{array}$ |
| Disjunktia | $\begin{array}{c} \neg(p \vee q) \\ \downarrow \\ \neg p \\ \neg q \end{array}$ $\begin{array}{c} p \vee q \\ \swarrow \downarrow \searrow \\ p \quad q \end{array}$ | $\begin{array}{c} p \quad q \\ \swarrow \downarrow \searrow \\ p \vee q \end{array}$ $\begin{array}{c} p \quad q \\ \swarrow \downarrow \searrow \\ p \vee q \end{array}$ $\begin{array}{c} \neg p \quad \neg q \\ \swarrow \downarrow \searrow \\ \neg(p \vee q) \end{array}$ |
| Konjunktia | $\begin{array}{c} p \wedge q \\ \downarrow \\ p \\ q \end{array}$ $\begin{array}{c} \neg(p \wedge q) \\ \swarrow \downarrow \searrow \\ \neg p \quad \neg q \end{array}$ | $\begin{array}{c} p \quad q \\ \swarrow \downarrow \searrow \\ p \wedge q \end{array}$ $\begin{array}{c} \neg p \quad \neg q \\ \swarrow \downarrow \searrow \\ \neg(p \wedge q) \end{array}$ $\begin{array}{c} \neg p \quad \neg q \\ \swarrow \downarrow \searrow \\ \neg(p \wedge q) \end{array}$ |

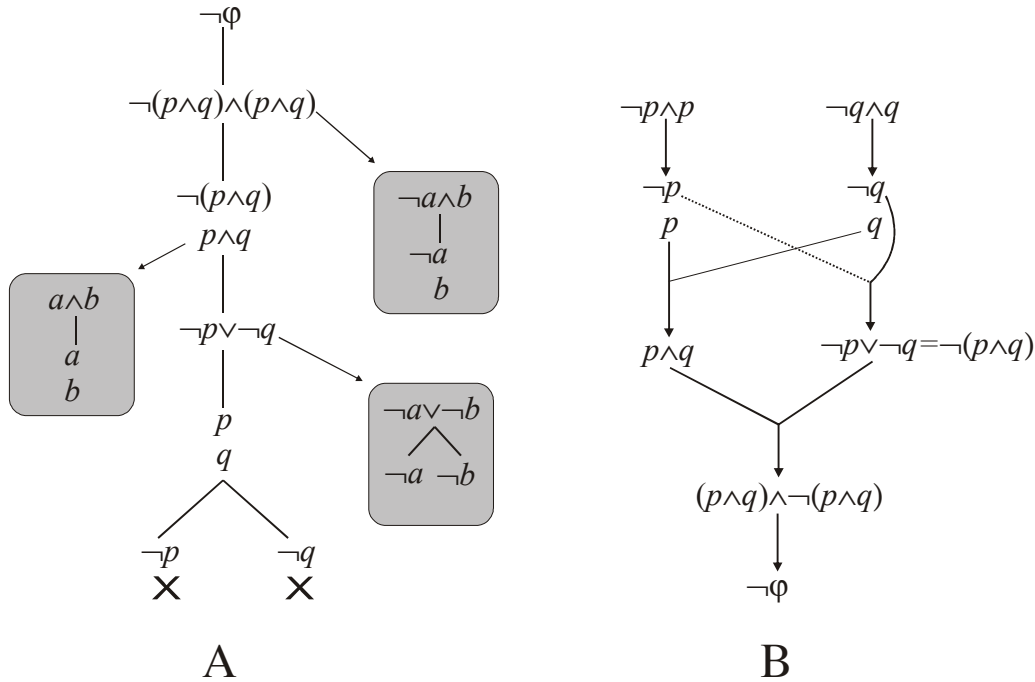
Poznámka: Prerušované čiary znamenajú, že týmto čiaram priradené formuly sú ľubovoľne – brané *ad-hoc*, nemusia byť odvodené v predchádzajúcej časti dôkazu.

Ako ilustračný príklad uvažujme pravidlo z tab. 6 pre prirodzenú dedukciu a implikáciu, predpoklady sú φ a $\neg\psi$, dôsledok je $\neg(\varphi \Rightarrow \psi)$, čo môžeme vyjadriť takto

$$\frac{\begin{array}{l} \varphi \\ \neg \psi \end{array}}{\neg(\varphi \Rightarrow \psi)}$$

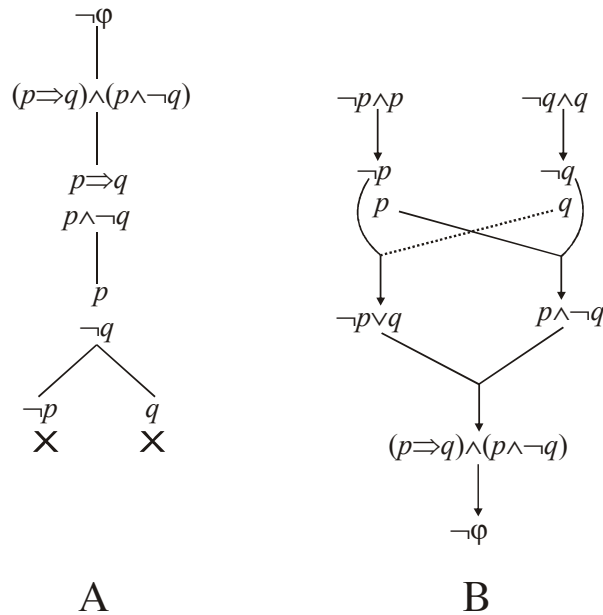
Lahko sa presvedčíme, že formula $\varphi \wedge \neg \psi \Rightarrow \neg(\varphi \Rightarrow \psi)$ je tautológia, čiže schéma usudzovania je korektná. Takýmto jednoduchým spôsobom môžeme preveriť všetky pravidlá z tab. 10 pre prirodzenú dedukciu.

Príklad 13. Dokážte tautologičnosť formuly $\varphi = \neg(p \wedge q) \Rightarrow (\neg p \vee \neg q)$ pomocou sémantického tabla a potom vykonajte dôkaz tejto formuly pomocou prirodzenej dedukcie založenej na sémantickom table. Sémantické tablo $\mathcal{T}(\neg\varphi)$ je znázornené na obr. 16. Pretože tablo je uzavreté, formula φ je tautológia. Inverziou (čítaním zdola nahor) môžeme zostrojiť dôkaz formuly $\neg\varphi$ len pomocou elementárnych schém usudzovania z tab. 10 (pozri diagram B obr. 16).



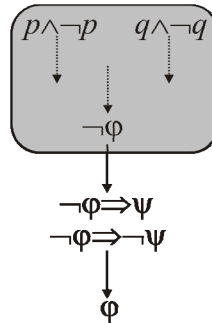
Obr. 16. (A) Sémantické tablo $\mathcal{T}(\neg\varphi)$ pre formulu $\varphi = \neg(p \wedge q) \Rightarrow (\neg p \vee \neg q)$ kde $\neg\varphi = (p \wedge q) \wedge \neg(p \wedge q)$ má všetky vetvy uzavreté, t. j. formula φ je tautológia. Použité schémy rozširovania sémantického tabla sú uvedené v tmavých blokoch stojacích vedľa stromu sémantického tabla (pozri obr. 5). (B) Prirodzená dedukcia pre danú formulu na základe duálneho sémantického tabla pre dôkaz, že daná formula je tautológia (pozri diagram A). Prirodzená dedukcia je zahájená tvorbou literálov z elementárnych kontradikcií $p \wedge \neg p$ a $q \wedge \neg q$. Prirodzená dedukcia je založená na inverznom postupe tvorby sémantického tabla $\mathcal{T}(\neg\varphi)$, pomocou elementárnych schém uvedených v Tab. 10.

Příklad 14. Dokážte tautologičnosť formuly $\varphi = (p \Rightarrow q) \Rightarrow (\neg p \vee q)$ pomocou sémantického tabla a potom pomocou neho zostrojte dôkaz formuly pomocou prirodzenej dedukcie. Sémantické tablo je znázornené na obr. 17, diagram A. Pomocou tohto sémantického tabla zostrojíme dôkaz formuly $\varphi = (p \Rightarrow q) \Rightarrow (\neg p \vee q)$ metódou prirodzenej dedukcie (pozri obr. 17, diagram B).



Obr. 17. (A) Duálne sémantické tablo pre formulu $\varphi = (p \Rightarrow q) \Rightarrow (\neg p \vee q)$, tablo je uzavreté, preto formula je tautológia. (B) Dôkaz formuly $\varphi = (p \Rightarrow q) \Rightarrow (\neg p \vee q)$ pomocou prirodzenej dedukcie, ktorá bola navrhnutá pomocou duálneho sémantického tabla z diagramu A.

Na základe týchto dvoch ilustračných príkladov môžeme konštatovať, že metóda sémantických tabiel je silne previazaná s prirodzenou dedukciou. Obrazne môžeme povedať, že tieto dve metódy sú navzájom v „inverznom“ vzťahu. V prvom kroku pre formulu φ pomocou sémantického tabla $\mathcal{T}(\neg\varphi)$ zistíme, či táto formula je tautológia. Ako áno (sémantické tablo je uzavreté), potom inverzným „čítaním“ tabla zdola nahor môžeme postupne zostrojiť graf prirodzenej dedukcie pomocou elementárnych schém usudzovania z tab. 10, ktorého výsledkom je formula $\neg\varphi$. Skutočnosť, že na základe sémantického tabla $\mathcal{T}(\neg\varphi)$ sme schopní zostrojiť dôkaz formuly $\neg\varphi$ (pritom o $\neg\varphi$ vieme, že je kontradikcia) pomocou prirodzenej dedukcie, nie je chybou nášho postupu. Formálne môžeme pokračovať v našom dôkaze tak, že dokázanú formulu $\neg\varphi$ rozšírime pomocou implikácie o formulu ψ a taktiež aj $\neg\psi$, pozri obr. 18.



Obr. 18. Formálne dokončenie dôkazu pomocou prirodzenej dedukcie, ktorá je založená na inverznom slede „príkazov“ zo sémantického tabla $\mathcal{T}(\neg\varphi)$. Postup je založený na zákone výrokovej logiky „*reductio ad absurdum*“ $(\neg\varphi \Rightarrow \psi) \wedge (\neg\varphi \Rightarrow \neg\psi) \Rightarrow \varphi$. Horný blok znázorňuje počiatočný stav dôkazu v ktorom rezultuje formula $\neg\varphi$, ktorá je v nasledujúcej etape dôkazu pomocou „*reductio ad absurdum*“ pretransformovaná na očakávanú formulu φ .

Tento prístup k prirodzenej dedukcii môžeme taktiež chápať ako konštruktívny dôkaz úplnosti, t. j. ak formula φ je tautológia, potom aj logicky vyplýva (existuje jej dôkaz) z daného systému axióm výrokovej logiky.

Veta 7.

Ak formula φ má uzavreté sémantické tablo $\mathcal{T}(\neg\varphi)$, t. j. je tautológia, potom pomocou „inverzie“ tohto tabla môžeme zostrojiť dôkaz formuly φ pomocou prirodzenej dedukcie.

4 Sémantické tablá v predikátovej logike

Technika sémantických tabiel pre predikátovú logiku je jednoduchou modifikáciou sémantických tabiel z výrokovej logiky (pozri obr. 2) o rozšírenia, ktoré zahŕňajú univerzálne a existenčné kvantifikátory. Tieto rozšírenia sú založené na nasledujúcich zákonoch predikátovej logiky

$$(\forall x)\varphi(x) \Rightarrow \varphi(t) \quad (40a)$$

$$(\exists x)\varphi(x) \Rightarrow \varphi(a) \quad (40b)$$

$$(\forall x)(\forall y)\varphi(x, y) \Rightarrow \varphi(t, t') \quad (40c)$$

$$(\exists x)(\exists y)\varphi(x, y) \Rightarrow \varphi(a, a') \quad (40d)$$

$$(\exists y)(\forall x)\varphi(x, y) \Rightarrow \varphi(t, a) \quad (40e)$$

$$(\exists x)(\forall y)\varphi(x, y) \Rightarrow \varphi(a, t) \quad (40f)$$

$$(\forall x)(\exists y)\varphi(x, y) \Rightarrow \varphi(t, f(t)) \quad (40g)$$

$$(\forall y)(\exists x)\varphi(x, y) \Rightarrow \varphi(f(t), t) \quad (40h)$$

kde t je ľubovoľný objekt (individuum) z univerza U , a je daný konštantný objekt (individuum) z univerza U . Poznamenajme, že dvojice formúl (40e-f) a (40g-h) sú ekvivalentné, použitím zámény objektov $x \leftrightarrow y$ môžeme jednu formulu prepísať na druhú formulu. Určité interpretačné problémy spôsobuje posledná formula (40f), ktorá môže byť chápaná ako „jemná“ modifikácia formuly (40e) s inverzným poradím kvantifikátorov. Ako ukážeme neskoršie, táto skutočnosť má dramatický dopad na korektnú interpretáciu ľavej strany formuly (40f). Posledné dve formuly sú upravené procesom *skolemizácie* takto:

$$\begin{aligned} (\forall x)(\exists y)\varphi(x,y) &\equiv (\forall x)\varphi(x, f(x)) \Rightarrow \varphi(t, f(t)), \\ (\forall y)(\exists x)\varphi(x,y) &\equiv (\forall y)\varphi(f(y), y) \Rightarrow \varphi(f(t), t), \end{aligned}$$

kde f sa nazýva *Skolemova funkcia*. Predĺženia predikátových formúl v teórii sémantických tabiel sú znázornené na obrázok 19.

$$\begin{array}{ccc} \begin{array}{c} | \\ (\forall x)\varphi(x) \end{array} \longrightarrow & \begin{array}{c} | \\ (\forall x)\varphi(x) \\ | \\ \varphi(t) \end{array} & \begin{array}{c} | \\ (\exists x)\varphi(x) \end{array} \longrightarrow & \begin{array}{c} | \\ (\exists x)\varphi(x) \\ | \\ \varphi(a) \end{array} \end{array}$$

eliminácia jedného kvantifikátora

$$\begin{array}{ccc} \begin{array}{c} | \\ (\forall x)(\forall y)p(x,y) \end{array} \longrightarrow & \begin{array}{c} | \\ (\forall x)(\forall y)p(x,y) \\ | \\ p(t,t') \end{array} & \begin{array}{c} | \\ (\exists x)(\exists y)p(x,y) \end{array} \longrightarrow & \begin{array}{c} | \\ (\exists x)(\exists y)p(x,y) \\ | \\ p(a,a') \end{array} \\ \\ \begin{array}{c} | \\ (\exists x)(\forall y)p(x,y) \end{array} \longrightarrow & \begin{array}{c} | \\ (\exists x)(\forall y)p(x,y) \\ | \\ p(a,t) \end{array} & \begin{array}{c} | \\ (\exists y)(\forall x)p(x,y) \end{array} \longrightarrow & \begin{array}{c} | \\ (\exists y)(\forall x)p(x,y) \\ | \\ p(t,a) \end{array} \\ \\ \begin{array}{c} | \\ (\forall x)(\exists y)p(x,y) \end{array} \longrightarrow & \begin{array}{c} | \\ (\forall x)(\exists y)p(x,y) \\ | \\ p(t,f(t)) \end{array} & \begin{array}{c} | \\ (\forall y)(\exists x)p(x,y) \end{array} \longrightarrow & \begin{array}{c} | \\ (\forall y)(\exists x)p(x,y) \\ | \\ p(f(t),t) \end{array} \end{array}$$

eliminácia dvoch kvantifikátorov

Obr. 19. Doplnenie základných rozšírení (pozri obr. 5) metódy sémantického tabla pre predikátovú logiku.

Upriamime našu pozornosť na formuly (40a-h). Prvé dve formuly (40a-b) sú priamym dôsledkom zákonov konkretizácie z predikátovej logiky. Žiaľ, tieto jednoduché rozšírenia sémantických tabiel pre formuly s jedným predikátom sú platné len pre tie formuly s dvoma predikátmi, ktoré neobsahujú „zmiešaný“ súčin existenčného a univerzálneho kvantifikátora.

Nech $\varphi = (\exists x)(\forall y)p(x, y)$ má alternatívny tvar

$$\varphi = (\exists x)(\forall y)p(x, y) \equiv \bigvee_{x \in U} \bigwedge_{y \in U} p(x, y) \equiv \left(\bigwedge_{y \in U} p(a_1, y) \right) \vee \dots \vee \left(\bigwedge_{y \in U} p(a_n, y) \right) \quad (41a)$$

kde $U = \{a_1, a_2, \dots, a_n\}$. Pretože sa jedná o disjunkciu konjunktívnych klauzúl, stačí vybrať len jednu pravdivú klauzulu a tá vyhovuje formule

$$(\exists x)(\forall y)p(x, y) \Rightarrow \left(\bigwedge_{y \in U} p(a, y) \right) \quad (41b)$$

kde $a \in U$ je vybraný objekt z univerza. Konjunktívna klauzula na pravej strane musí byť pravdivá pre každé $y \in U$, potom

$$(\exists x)(\forall y)p(x, y) \Rightarrow p(a, t) \quad (42)$$

kde $t \in U$ je ľubovoľný objekt z univerza U . Môžeme povedať, že tento výsledok by sme získali aj postupným použitím pravidiel (40a-b).

Pre formulu (40g) je situácia o niečo zložitejšia, táto formula má tento alternatívny tvar

$$\psi = (\forall x)(\exists y)p(x, y) \equiv \bigwedge_{x \in U} \bigvee_{y \in U} p(x, y) \equiv \left(\bigvee_{y \in U} p(x_1, y) \right) \wedge \dots \wedge \left(\bigvee_{y \in U} p(x_n, y) \right) \quad (43)$$

kde $U = \{x_1, x_2, \dots, x_n\}$. Pre dané x_i stanovenie takého $y \in U$, aby disjunkcia $\bigvee_{y \in U} p(x_i, y)$ bola pravdivá, musí prebiehať vzhľadom k x_i , t. j. $a = y = f(x_i)$, kde $f(x)$

je Skolemova funkcia, ktorá určuje jednoznačne y vzhľadom k danému $x = t$. Potom platí implikácia

$$(\forall x)(\exists y)p(x, y) \Rightarrow p(t, f(t)) \quad (44)$$

kde $t \in U$ je ľubovoľný objekt z univerza U . Poznamenajme, že Skolemova funkcia $f(t)$ priradí ľubovoľnému $t \in U$ nejaké vybrané $a = f(t)$ tak, aby formula (44) bola pravdivá. Podobným spôsobom môžeme študovať aj formuly (40c-d), ktoré obsahujú dva univerzálne alebo existenčné kvantifikátory. Podobnými úvahami, ako v predchádzajúcich dvoch prípadoch dostaneme formuly

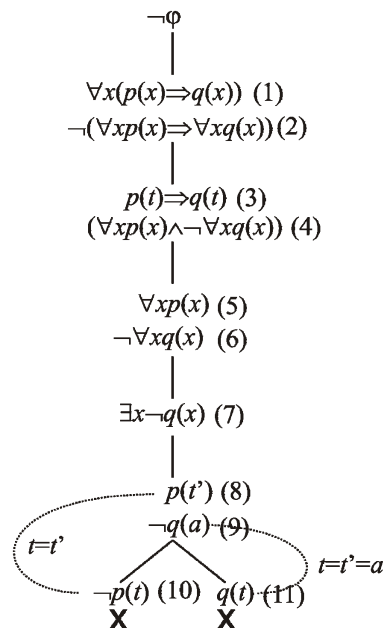
$$(\forall x)(\forall y)p(x, y) \Rightarrow p(t, t') \quad (45a)$$

$$(\exists x)(\exists y)p(x, y) \Rightarrow p(a, a') \quad (45b)$$

kde $t, t' \in U$ sú ľubovoľné objekty a $a, a' \in U$ sú vybrané objekty.

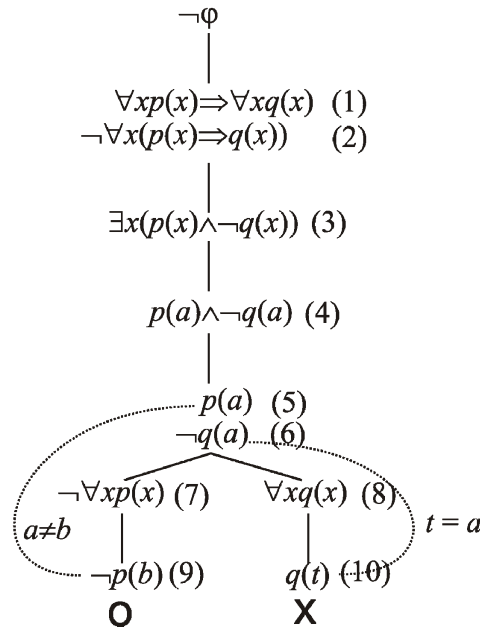
Príklad 15. Použitím techniky sémantického tabla dokážeme, že formula $\varphi = \forall x(p(x) \Rightarrow q(x)) \Rightarrow (\forall x p(x) \Rightarrow \forall x q(x))$ je zákonom predikátovej logiky, pozri obr. 20. Jednotlivé formuly v sémantickom table sú indexované (čísla v zátvorkách) a majú tento význam:

- Pôvodná formula (koreň sémantického tabla) $\neg\varphi$ bola predĺžená aplikáciou schémy predĺženia $\neg(\alpha \Rightarrow \beta)$, pozri obr. 5, diagram C, vznikli nové formuly (1) a (2).
- Na formulu (1) bola použitá schéma predĺženia z obr. 19, odstránili sme univerzálny kvantifikátor a všeobecná individuová premenná x bola nahradená ľubovoľnou individuovou konštantou t , vznikla formula (3).
- Formula (4) vznikla aplikáciou ekvivalencie $\neg(\alpha \Rightarrow \beta) \equiv \alpha \wedge \neg\beta$ na formulu (2).
- Formuly (5) a (6) vznikli predĺžením (4) podľa schémy z obr. 5, diagram B.
- Formula (7) vznikla z (6) ekvivalentným prepisom $\neg\forall xR(x) \equiv \exists x\neg R(x)$.
- Formula (8) vznikla z (5) použitím predĺženia z obr. 19, odstránili sme univerzálny kvantifikátor a individuovú premennú x sme nahradili ľubovoľnou individuovou konštantou t' .
- Formula (9) vznikla z (7) aplikáciou predĺženia z obr. 19, odstránili sme existenčný kvantifikátor a individuovú premennú x sme nahradili vybranou individuovou konštantou a .
- Vzniknuté sémantické tablo má dve vetvy. Ľavá vetva je uzavretá, pretože môžeme vytvoriť kontradikciu tak, že položíme $t = t'$ (poznajme, že tieto konštanty sú ľubovoľné). Pravá vetva je taktiež uzavretá, kontradikciu vytvoríme tak, že položíme $t = a$. Tieto dve podmienky sú realizovateľné súčasne, čo môžeme napísať ako $t = t' = a$.



Obr. 20. Sémantické tablo formuly $\varphi = (\forall x)(p(x) \Rightarrow q(x)) \Rightarrow ((\forall x)p(x) \Rightarrow (\forall x)q(x))$. Pretože tablo $\mathcal{T}(\neg\varphi)$ je uzavreté, formula φ je tautológia, čo bolo potrebné dokázať.

Príklad 16. Použitím techniky sémantického tabla dokážeme, že formula $\varphi = (\forall x p(x) \Rightarrow \forall x q(x)) \Rightarrow \forall x (p(x) \Rightarrow q(x))$ nie je zákonom predikátovej logiky, pozri obr. 21.



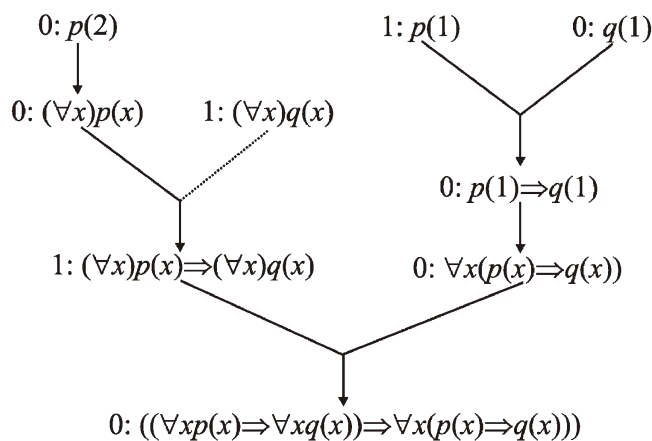
Obr. 21. Sémantické tablo pre negáciu formuly $\varphi = ((\forall x)p(x) \Rightarrow (\forall x)q(x)) \Rightarrow (\forall x)(p(x) \Rightarrow q(x))$. Tablo $T(\neg\varphi)$ nie je uzavreté, formula φ nie je tautológia, je len splniteľná s interpretáciou zostrojiteľnou pomocou otvorenej vetvy.

Podobne, ako v predchádzajúcom príklade, aj teraz pristúpime k podrobnej špecifikácii každého riadku sémantického tabla z obr. 21:

- Pôvodná formula (koreň sémantického tabla) $\neg\varphi$ bola predĺžená aplikáciou schémy predĺženia $\neg(\alpha \Rightarrow \beta)$, pozri obr. 5, diagram C, vznikli formuly (1) a (2).
- Formula (3) vznikla z (2) aplikáciou ekvivalencií $\neg(\forall x)R(x) \equiv (\exists x)\neg R(x)$ a $\neg(\alpha \Rightarrow \beta) \equiv \alpha \wedge \neg\beta$.
- Formula (4) vznikla z (3) použitím predĺženia z obr. 19, odstránili sme existenčný kvantifikátor a individuová premenná x bola nahradená danou individuovou konštantou a .
- Formuly (5) a (6) vznikli z (4) použitím predĺženia z obr. 5, diagram B.
- Formuly (7) a (8) vznikli z (1) použitím predĺženia z obr. 5, diagram C.

- Formula (9) vznikla z (7) použitím predĺženia z obr. 19, odstránili sme univerzálny kvantifikátor a individuová premenná x bola nahradená danou individuovou konštantou b .
- Ľavá vetva sémantického tabla neprodukuje vo všeobecnosti kontradikciu (nie je uzavretá), potenciálna možnosť jej tvorby pomocou (5) a (9) nemôže existovať, pretože vo všeobecnosti platí $a \neq b$.
- Pravá vetva je uzavretá, tu vytvoríme kontradikciu podmienkou $t = a$.
- Sémantické tablo nie je uzavreté, preto formula φ nie je tautológia.

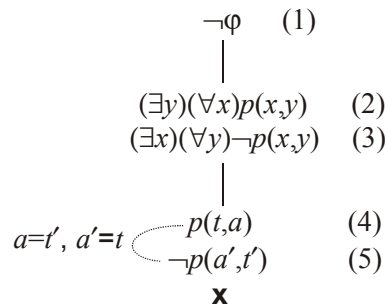
Poznamenajme, že otvorená ľavá vetva sémantického tabla môže byť použitá na konštrukciu takej interpretácie, pre ktorú je formula φ je nepravdivá, t. j. požiadavka tautologičnosti formuly φ je falzifikovaná. Z ľavej vetvy pre hodnoty predikátov p a q dostaneme tieto hodnoty: $p(a)=1, \neg q(a)=1, \neg p(b)=1$. Z prvých dvoch podmienok odvodíme ich konjunkciu $p(a) \wedge \neg q(a)=1$, čo môžeme zovšeobecniť pomocou existenčného kvantifikátora $\exists x(p(x) \wedge \neg q(x)) = \exists x \neg(p(x) \Rightarrow q(x)) = \neg \forall x(p(x) \Rightarrow q(x))=1$, Z tretej podmienky $\neg p(b)=1$ môžeme vyvodit': $\exists x \neg p(x) = \neg \forall x p(x)=1$. Tento výsledok môžeme pomocou disjunkcie rozšíriť, $\neg \forall x p(x) \vee \forall x q(x) = \forall x p(x) \Rightarrow \forall x q(x)=1$. Teraz môžeme pristúpiť k výpočtu pravdivostnej hodnoty formuly φ , $((\forall x)p(x) \Rightarrow (\forall x)q(x)) \Rightarrow (\forall x)(p(x) \Rightarrow q(x)) \equiv (1 \Rightarrow 0) \equiv 0$. Tento dôležitý moment využitia sémantického tabla pre konštrukciu falzifikácie tautologičnosti danej formuly môžeme aj znázorniť diagramaticky, pozri obr. 22.



Obr. 22. Diagramatická falzifikácia tautologičnosti formuly $((\forall x)p(x) \Rightarrow (\forall x)q(x)) \Rightarrow (\forall x)(p(x) \Rightarrow q(x))$ na základe počiatočných predpokladov $p(a)=1, \neg q(a)=1, \neg p(b)=1$, ktoré boli získané z ľavej vetvy sémantického tabla z obr. 21. Posledný riadok diagramatickej interpretácie znamená, že pre dané podmienky pravdivostná hodnota formuly je nepravda, t. j. formula nemôže byť tautológiou.

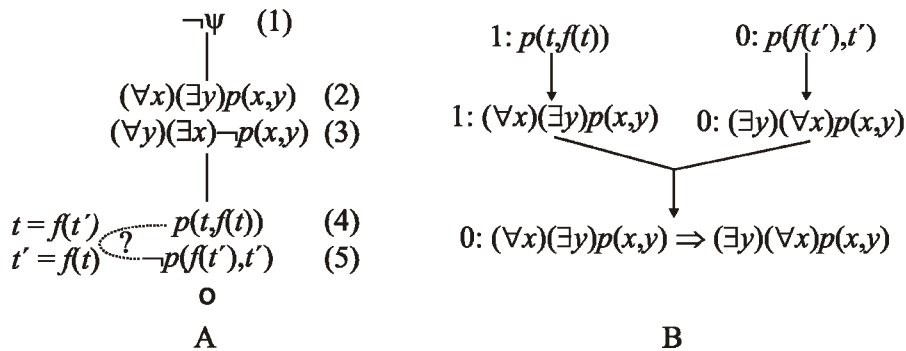
Tieto dva ilustračné príklady ukazujú na potenciálnu vhodnosť sémantických tabiel pre konštrukciu pravdivostnej interpretácie formúl predikátovej logiky, kde majú nezastupiteľnú úlohu pre konštrukciu pravdivostných hodnôt, pretože v predikátovej logike je tabuľková metóda nepoužiteľná.

Príklad 17. Použitím techniky sémantického tabla dokážeme, že formula $\varphi = (\exists y)(\forall x)p(x, y) \Rightarrow (\forall x)(\exists y)p(x, y)$ je zákonom predikátovej logiky, pozri obr. 23.



Obr. 23. Sémantické tablo pre negáciu formuly $\varphi = (\exists y)(\forall x)p(x, y) \Rightarrow (\forall x)(\exists y)p(x, y)$, jediná vetva tohto tabla je uzavretá, čiže formula φ je tautológia. Pri prechode od formúl-vrcholov (2-3) k formulám-vrcholom (4-5) sme použili pravú schému predĺženia z druhého riadku obrázku 19.

Príklad 18. Použitím techniky sémantického tabla falzifikujte tautologičnosť formuly $\psi = (\forall x)(\exists y)p(x, y) \Rightarrow (\exists y)(\forall x)p(x, y)$, pozri obr. 24.



Obr. 24. (A) Sémantické tablo negácie formuly $\psi = (\forall x)(\exists y)p(x, y) \Rightarrow (\exists y)(\forall x)p(x, y)$, jediná vetva tabla je otvorená, čiže formula ψ nie je tautológia. (B) Pomocou tohto sémantického tabla môžeme zostrojiť taký model formuly ψ , v ktorom je formula nepravdivá, čiže sme falzifikovali jej tautologičnosť.

5 Sémantické tablá v modálnej logike [4,6,7,12,15]

Použijeme jednoduchý prístup ako zaviesť modálnu logiku pomocou výrokovej logiky. Množina logických spojok výrokovej logiky, $\{\wedge, \vee, \Rightarrow, \equiv, \neg\}$, je rozšírená o ďalšie dve unárne spojky \Box a \Diamond . Formula modálnej logiky $\Box\varphi$ znamená 'nutne platí φ ', analogická formula $\Diamond\varphi$ znamená 'možno platí φ '. Tieto dve spojky nie sú nezávislé, navzájom sú zviazané reláciou $\Box\varphi \equiv \neg\Diamond\neg\varphi$, alebo $\Diamond\varphi \equiv \neg\Box\neg\varphi$. Pôvodne boli tieto dve logické spojky \Box a \Diamond chápané ako validita resp. splniteľnosť. Analogické spojky boli neskoršie použité pri formulácii kauzálnej, deontickej, epistemickej a veľmi populárnej temporálnej logiky.

Ako jednoduché rozšírenie klasickej výrokovej logiky pomocou modálnych spojok boli študované rôzne typy modálnych logík americkým logikom C. I. Lewisom, ktorý je pokladaný za zakladateľa modernej modálnej logiky [13]. Žiaľ, jeho spôsob prezentácie modálnych logík bol veľmi ťažkopádny, aj keď zo súčasného pohľadu väčšina jeho výsledkov sa ukázala korektnými. Zásluhou Kurta Gödela [8] tento systém bol prevedený do súčasného prehľadného hierarchického axiomatického systému, kde zapájaním rôznych axióm dostávame rôzne typy modálnych logík. V tomto prístupe veľmi dôležitú úlohu hrá nové pravidlo odvodzovania (okrem štandardného pravidla modus ponens) nazývané pravidlo nutnosti³ $\varphi/\Box\varphi$, ktoré umožňuje inferenciu $\Box\varphi$ z východiskového predpokladu φ . Najdôležitejšie axiómy tohto systému sú

| | |
|--------------------|---|
| (Ax ₀) | $\Box(\varphi \Rightarrow \psi) \Rightarrow (\Box\varphi \Rightarrow \Box\psi)$ |
| (Ax ₁) | $\Box\varphi \Rightarrow \varphi$ |
| (Ax ₂) | $\Box\varphi \Rightarrow \Box\Box\varphi$ |
| (Ax ₃) | $\Diamond\varphi \Rightarrow \Box\neg\Box\neg\varphi$ |

Najdôležitejšie systémy vždy obsahujú formulu (Ax₀); systém, ktorý obsahuje len túto axiómu a pravidlo nutnosti sa nazýva **K** na počesť Saula Kripkeho, pretože mal mimoriadne postavenie pri vzniku jeho teórie modelu modálnej logiky. Ostatné najdôležitejšie systémy modálnej logiky vznikli zahrnutím týchto axióm

| | |
|------------|---|
| K: | (Ax ₀) |
| T: | (Ax ₀) + (Ax ₁) |
| S4: | (Ax ₀) + (Ax ₁) + (Ax ₂) |
| B: | (Ax ₀) + (Ax ₁) + (Ax ₃) |
| S5: | (Ax ₀) + (Ax ₁) + (Ax ₂) + (Ax ₃) |

Všeobecnou snahou boli pokusy navrhnuť spoločný model pre modálne logiky rôzneho typu, ktorý by prechádzal na iný typ modelu len jednoduchou zmenou základných „variabilných parametrov“ modelu. Tieto pokusy boli neúspešné až do zavedenia Kripkeho modelu.

Kripkeho model modálnej logiky je podstatne zložitejší ako ten, ktorý bol použitý v závere predchádzajúcej kapitoly. Pravdivostné ohodnotenie danej formuly modálnej logiky bude založené na rekurzii podľa zložitosti formuly, pričom môžeme prechádzať

³ Toto pravidlo v rámci sémantickej interpretácie môžeme jednoducho interpretovať tak, že ak formula φ je tautológia, potom aj $\Box\varphi$ musí byť tautológia.

z daného „sveta“, kde vyhodnotenie realizujeme, do iného „alternatívneho sveta“, kde využívame pravdivostné ohodnotenie jej podformúl. Snáď na tejto úvodnej úrovni môžeme použiť vysvetľujúcu analógiu so situáciou, keď chceme riešiť pravdivosť výroku $\Box\varphi$, kde φ = 'vlak do Žiliny odchádza o 14.37 hod'. Ak nám všetci kolegovia v práci (reprezentujúci Kripkeho alternatívne svety) vyhodnotia výrok φ ako pravdivý, potom môžeme povedať, že formula $\Box\varphi$ je pravdivá, alebo výrok 'vlak do Žiliny odchádza o 14.37 hod' je nutne pravdivý výrok. Alternatívne, ak nám výrok φ vyhodnotí aspoň jeden kolega ako pravdivý, potom hovoríme, že výrok 'vlak do Žiliny odchádza o 14.37 hod' je možné pravdivý, čo označíme ako $\Diamond\varphi$.

Kripkeho model $M = (W, R, w_0, val)$ je usporiadaná štvorica [4,7,12,15]. Symbol $W = \{w, w', \dots\}$ je neprázdna množina svetov, symbol $R \subseteq W \times W$ je binárna relácia, $w_0 \in W$ je odlišný svet pre daný model, a val je zobrazenie, ktoré pravdivostne ohodnocuje výrokové premenné z $P = \{p, q, \dots, p', q', \dots\}$ pre daný svet $w \in W$, $val: W \times P \rightarrow \{0, 1\}$. Symbol $val(w, p) = 1$ (0) znamená, že premenná $p \in P$ je vo svete $w \in W$ pravdivá (nepravdivá). Binárna relácia R špecifikuje tzv. dostupné svety, ak $(w, w') \in R$, potom hovoríme, že svet w' je dostupný zo sveta w . Binárna relácia R môže byť formálne nahradená podmnožinami $\Gamma(w) \subseteq W$, kde $\Gamma(w) = \{w' \in W; (w, w') \in R\}$ obsahuje svety, ktoré sú dostupné zo sveta w . Odlišný svet w_0 špecifikuje použitý model M pre daný svet w_0 . Ako alternatívu k tomuto označeniu zavedieme reláciu 'sémantického vyplývania' \models

$$(w \models p) =_{def} (val(w, p) = 1)$$

$$(w \not\models p) =_{def} (val(w, p) = 0)$$

kde $w \models p$ ($w \not\models p$) čítame ako 'vo svete w je výrok p pravdivý' ('vo svete w je výrok p nepravdivý'). Zovšeobecnie zobrazenia val (alebo relácia sémantického vyplývania) pre neatomické výrokové formuly φ a ψ má tvar:

$$\text{negácia} \quad (w \models \neg\varphi) =_{def} (w \not\models \varphi) \quad (46)$$

$$\text{konjunkcia} \quad (w \models \varphi \wedge \psi) =_{def} (w \models \varphi) \wedge (w \models \psi) \quad (47)$$

$$\text{disjunkcia} \quad (w \models \varphi \vee \psi) =_{def} (w \models \varphi) \vee (w \models \psi) \quad (48)$$

$$\text{implikácia} \quad (w \models \varphi \Rightarrow \psi) =_{def} (w \not\models \varphi) \vee (w \models \psi) \quad (49)$$

$$\text{spojka 'nutne'} \quad (w \models \Box\varphi) =_{def} (\forall w' \in \Gamma(w))(w' \models \varphi) \quad (50)$$

$$\text{spojka 'možne'} \quad (w \models \Diamond\varphi) =_{def} (\exists w' \in \Gamma(w))(w' \models \varphi) \quad (51)$$

V tejto tabuľke formula (51) je redundantná, danú podmienku môžeme odvodiť z ekvivalencie $\Diamond\varphi \equiv \neg\Box\neg\varphi$. Takto naformulovaný Kripkeho model často nazývame aj ako **Kripkeho sémantika** pre výrokovú modálnu logiku. Kripke pri konštrukcii tohto modelu bol stimulovaný Leibnizom, ktorý spomína možné svety, ktoré sú obsiahnuté

v myslí Boha, ktorý však stvoril z týchto alternatívne možných svetov ten najlepší svet. Musíme však poznamenať, že táto „pitoreskná“ terminológia Kripkeho sémantického modelu spôsobila mnohé nedorozumenia a hádky pri diskutovaní filozofických základov modálnej logiky.

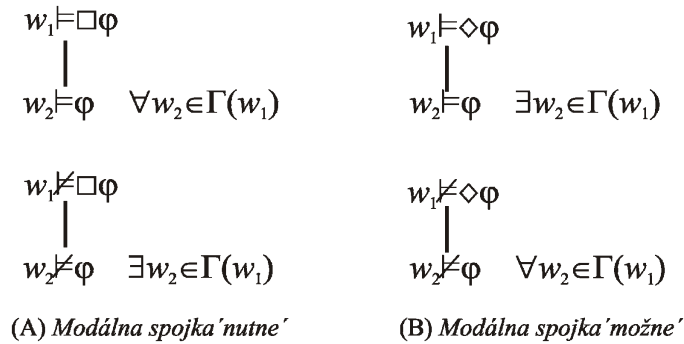
Na záver tejto kapitoly zavedieme štandardnú notáciu modálnej logiky, ktorá bude vyjadrovať skutočnosť, že formuly $\Box\varphi$ a $\Diamond\varphi$ sú pravdivé vo svete w v modeli M , $w \models \Box\varphi$ resp. $w \models \Diamond\varphi$; nepravdivosť týchto formúl sa vyjadruje $w \not\models \Box\varphi$ resp. $w \not\models \Diamond\varphi$. Potom podmienky (50) a (51) z tabuľky uvedenej vyššie môžeme prezentovať takto

$$(w \models \Box\varphi) =_{def} \begin{cases} \bigwedge_{w' \in \Gamma(w)} (w' \models \varphi) & (pre \Gamma(w) \neq \emptyset) \\ \mathbf{1} & (pre \Gamma(w) = \emptyset) \end{cases} \quad (52)$$

$$(w \models \Diamond\varphi) =_{def} \begin{cases} \bigvee_{w' \in \Gamma(w)} (w' \models \varphi) & (pre \Gamma(w) \neq \emptyset) \\ \mathbf{0} & (pre \Gamma(w) = \emptyset) \end{cases} \quad (53)$$

kde bolo vykonané aj zovšeobecnenie formúl pre prípad, že podmnožina $\Gamma(w)$ obsahujúca dostupné svety zo sveta w je prázdna.

Metóda sémantických tabiel pre modálnu logiku je jednoduchým rozšírením tohto prístupu platného pre výrokovú logiku tak, že zavedieme nové rozšírenia sémantického tabla pre modálne spojky \Box a \Diamond , pozri obr. 25.

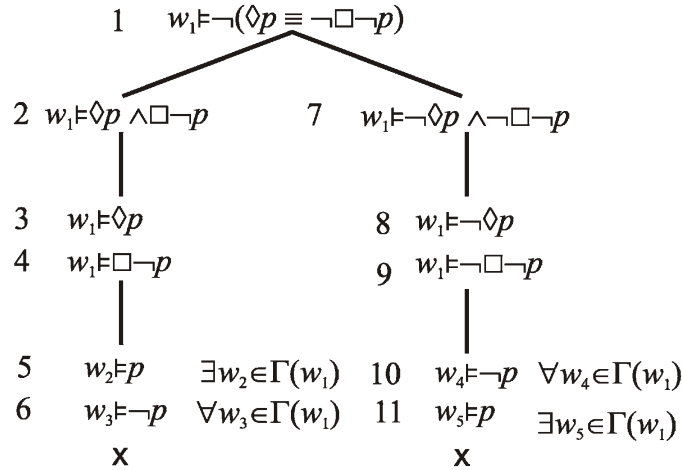


Obr. 25. Rozšírenie metódy sémantických tabiel pre modálnu logiku. V dolnej časti diagramu vpravo sú uvedené objekty z množiny W (možné svety), pre ktoré formula platí.

Príklad 19. Metódou sémantického tabla ukážte, že formula $\varphi = (\Diamond p \equiv \neg\Box(\neg p))$ je tautológia. Vzniknuté sémantické tablo je znázornené na obr. 26, kde pre jednoduchosť sú jednotlivé uzly tabla označené číslami 1, ..., 6 s nasledujúcim komentárom:

1. Koreň tabla obsahuje negovanú formulu $\neg\varphi$, ktorá je ekvivalentnými manipuláciami prepísaná do konjunkcie $\neg(\Diamond p \Rightarrow \neg\Box(\neg p)) \wedge (\neg\Box(\neg p) \Rightarrow \Diamond p)$,

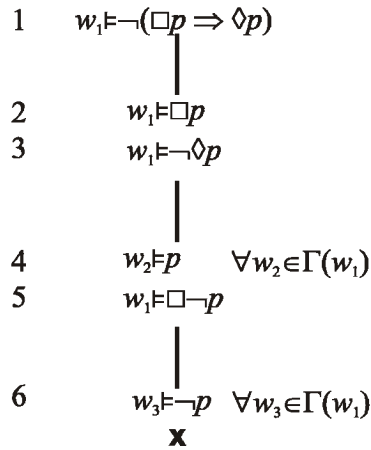
- pričom predpokladáme, že ekvivalentný prepis formuly $\neg\varphi = (\diamond p \wedge \Box \neg p) \vee (\neg \diamond p \wedge \neg \Box \neg p)$ je uvažovaný vo svete $w_1 \in W$.
2. Tento uzol sémantického tabla obsahuje podformulu $\diamond p \wedge \Box \neg p$ vo svete w_1 , ktorá pochádza z pôvodnej formuly $\neg\varphi = (\diamond p \wedge \Box \neg p) \vee (\neg \diamond p \wedge \neg \Box \neg p)$.
 - 3.-4. Tieto dva uzly sú priradené podformulám $\diamond p$ a $\Box \neg p$, obe uvažované vo svete $w_1 \in W$, ktoré vznikli z predchádzajúceho uzla 2 jeho rozkladom podľa konjunkcie.
 5. Uzol vznikol z uzla 3 odstránením modálneho operátora \diamond , vzniknutá podformula p je uvažovaná vo svete $\exists w_2 \in \Gamma(w_1)$.
 6. Uzol vznikol z uzla 4 odstránením modálneho operátora \Box , vzniknutá podformula $\neg p$ je uvažovaná vo svetoch $\forall w_3 \in \Gamma(w_1)$. Táto vetva sémantického tabla je **uzavretá**, pretože dvojica komplementárnych literálov p a $\neg p$ môže existovať v rovnakom svete $w_2 = w_3$. Podobným spôsobom môže byť komentovaná aj druhá vetva sémantického tabla obsahujúca uzly 7.-11., ktorá je taktiež uzavretá. Týmto sme dokázali, že formula φ je **tautológia**, jej pravdivostná hodnota nezávisí na zobrazení v , ktoré špecifikuje pravdivostnú hodnotu atomických premenných v jednotlivých svetoch z W a taktiež nie je závislá od tvaru podmnožín $\Gamma(w)$ zavedených pri definícii Kripkeho modelu.



Obr. 26. Sémantické tablo formuly $\varphi = (\diamond p \equiv \neg \Box (\neg p))$

Príklad 20. Metódou sémantického tabla ukážte, že formula $\varphi = (\Box p \Rightarrow \diamond p)$ je tautológia, riešenie je uvedené na obr. 27. Popis jednotlivých uzlov sémantického tabla z obr. 27:

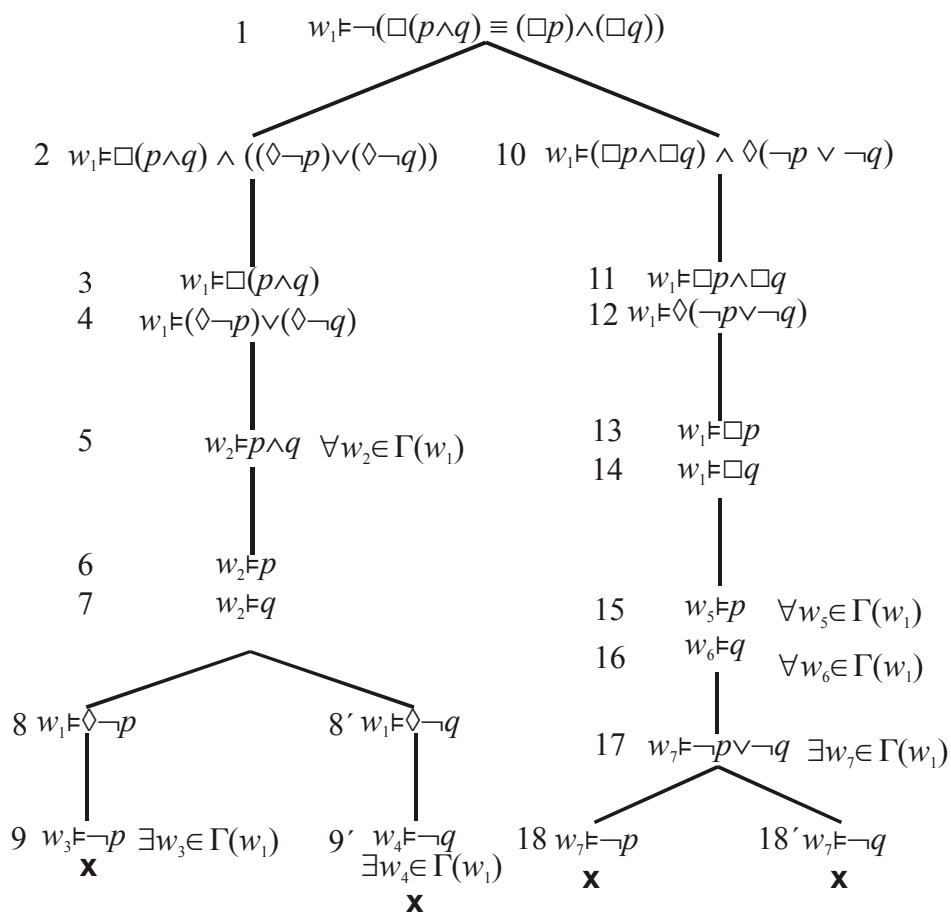
1. Uzol sémantického tabla je inicializovaný formulou $\neg\varphi = \neg(\Box p \Rightarrow \Diamond p)$, ktorá je uvedená aj v upravenom ekvivalentnom tvare $\neg\varphi = \Box p \wedge \neg\Diamond p$, pričom formula $\neg\varphi$ je uvažovaná vo svete $w_1 \in W$.
2. Uzol obsahuje podformulu $\Box p$ vo svete $w_1 \in W$.
3. Uzol obsahuje podformulu $\neg\Diamond p \equiv \Box\neg p$ vo svete $w_1 \in W$.
4. Uzol obsahuje literál p , ktorý vznikol z uzlu 2 odstránením spojky \Box , podformula je uvažovaná vo svetoch $\forall w_2 \in \Gamma(w_1)$.
5. Uzol vznikol z uzlu 3 jeho prepisom pomocou formuly $\neg\Diamond p \equiv \Box\neg p$, pričom podformula $\Box\neg p$ je uvažovaná vo svete $w_1 \in W$.
6. Uzol vznikol z 5 odstránením modálnej spojky \Box , podformula $\neg p$ je uvažovaná vo svetoch $\forall w_3 \in \Gamma(w_1)$, komplementárne literály z uzla 4 a 6 musia existovať v rovnakom svete $w_2 = w_3$, t. j. vetva je uzavretá a formula φ je tautológia.



Obr. 27. Sémantické tablo formuly $\varphi = (\Box p \Rightarrow \Diamond p)$.

Príklad 21. Metódou sémantického tabla zistíme, že formula $\varphi = ((\Box(p \wedge q)) \equiv ((\Box p) \wedge (\Box q)))$ je tautológia, riešenie je uvedené na obr. 28. Nebudeme už popisovať každý uzol sémantického tabla, ktoré má 4 uzavreté vetvy, ktoré idúc zľava doprava sú tieto:

1. vetva, obsahuje uzly 1-9, dvojica komplementárnych literálov je p (vo svetoch $\forall w_2 \in \Gamma(w_1)$) a $\neg p$ (vo svete $\exists w_3 \in \Gamma(w_1)$), potom pre $a = w_3 = w_2$ táto komplementárna dvojica súčasne existuje vo svete $a \in \Gamma(w_1)$.



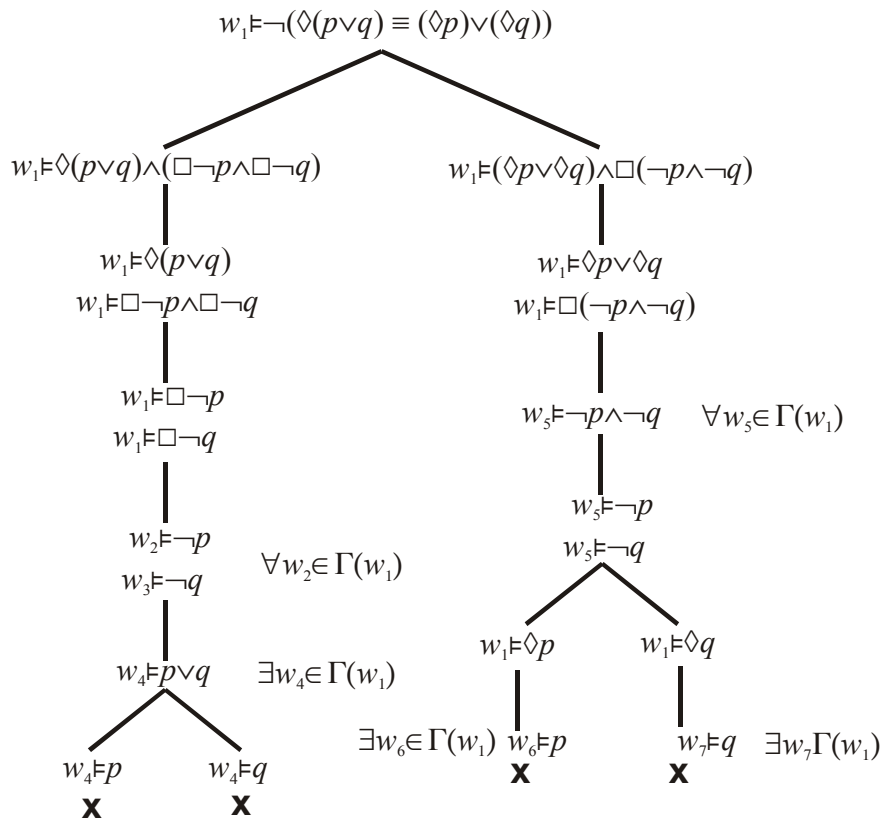
Obr. 28. Sémantické tablo formuly $\varphi = ((\Box(p \wedge q)) \equiv ((\Box p) \wedge (\Box q)))$

2. vetva, obsahuje uzly 1-7, 8', 9', dvojica komplementárnych literálov pre túto vetvu obsahuje q (vo svetoch $\forall w_2 \in \Gamma(w_1)$) a $\neg q$ (vo svete $\exists w_4 \in \Gamma(w_1)$), potom pre $a = w_4 = w_2$ daná komplementárna dvojica súčasne existuje v rovnakom svete $a \in \Gamma(w_1)$.
3. vetva, obsahuje uzly 1, 10-18, dvojica komplementárnych literálov je p (vo svetoch $\forall w_5 \in \Gamma(w_1)$) a $\neg p$ (vo svetoch $\forall w_7 \in \Gamma(w_1)$), potom pre svet $a = w_5 = w_7 \in \Gamma(w_1)$ daná komplementárna dvojica súčasne existuje v rovnakom svete $a \in \Gamma(w_1)$.
4. vetva, obsahuje uzly 1, 10-17, 18', dvojica komplementárnych literálov je q (vo svete $\exists w_6 \in \Gamma(w_1)$) a $\neg q$ (vo svetoch $\forall w_7 \in \Gamma(w_1)$), potom pre svet

$a = w_6 = w_7 \in \Gamma(w_1)$ daná komplementárna dvojica súčasne existuje v rovnakom svete $a \in \Gamma(w_1)$.

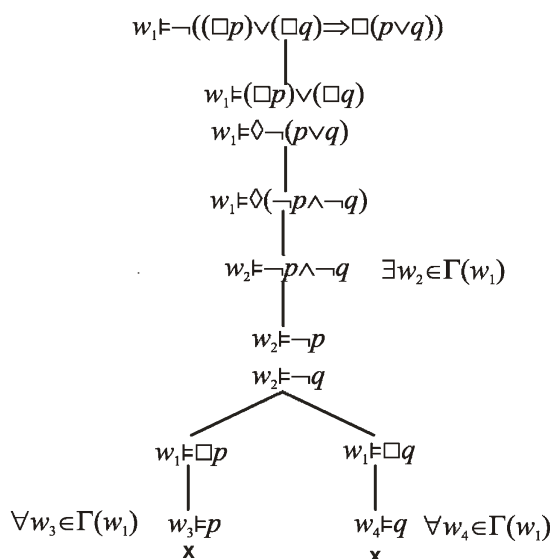
Týmto sme ukázali, že všetky vetvy sú uzavreté, t. j. formula $\varphi = ((\Box(p \wedge q)) \equiv ((\Box p) \wedge (\Box q)))$ je tautológia.

Příklad 22. Dôkaz tautologičnosti formuly $\varphi = (\Diamond(p \vee q) \equiv (\Diamond p) \vee (\Diamond q))$ pomocou sémantického tabla je znázornený na obr. 29. Nebudeme špecifikovať jednotlivé elementárne kroky konštrukcie sémantického tabla, zdôrazníme len, že všetky vetvy tabla sú uzavreté, t. j. formula φ je tautológia (ktorá je pravdivá pre každé zobrazenie v a v každom svete w).



Obr. 29. Sémantické tablo formuly $\varphi = (\Diamond(p \vee q) \equiv (\Diamond p) \vee (\Diamond q))$

Příklad 23. Dôkaz tautologičnosti formuly $\varphi = (\Box p) \vee (\Box q) \Rightarrow \Box(p \vee q)$ pomocou sémantického tabla je znázornený na obr. 30, ktorý obsahuje dve uzavreté vetvy, t. j. formula je tautológia.

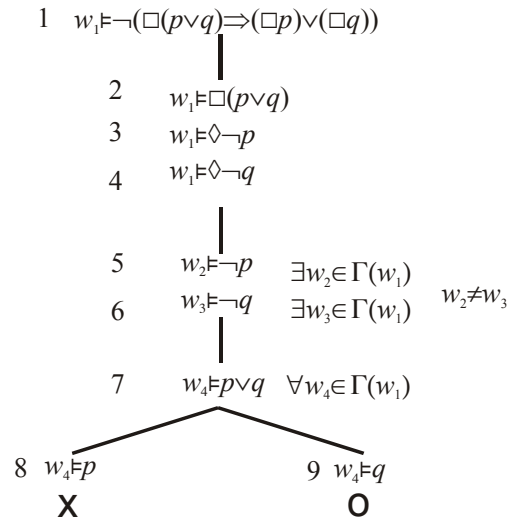


Obr. 30. Sémantické tablo formuly $\varphi = (\Box p) \vee (\Box q) \Rightarrow \Box(p \vee q)$

Priklad 24. Falzifikácia tautologičnosti formuly $\varphi = \Box(p \vee q) \Rightarrow (\Box p) \vee (\Box q)$ pomocou sémantického tabla je znázornená na obr. 31. Tablo obsahuje dve vetvy, jedna vetva (ľavá) je uzavretá, zatiaľ čo pravá vetva je otvorená, pretože svet w_4 je fixovaný už z ľavej vetvy. Tabuľka 11 obsahuje druhý alternatívny dôkaz, že formula $\Box(p \vee q) \Rightarrow (\Box p) \vee (\Box q)$ nie je tautológia (alebo presnejšie, falzifikáciu predpokladu tautologičnosti formuly). Pomocou sémantického tabla z obr. 31 sme si zvolili pravdivostné hodnoty premenných p a q , potom postupne zostrojujeme podformuly danej formuly, až na záver dostaneme, že pre dané pravdivostné hodnoty premenných p a q v rôznych svetoch z množiny $W = \{w_1, w_2, w_3\}$, kde podmnožiny dosiahnuteľných svetov sú špecifikované takto: $\Gamma(w_1) = \{w_1, w_2, w_3\}$, $\Gamma(w_2) = \{w_2\}$, $\Gamma(w_3) = \{w_3\}$, je formula vždy nepravdivá vo svete w_1 , čiže bola falzifikovaná jej tautologičnosť.

Tabuľka 11. Sémantická interpretácia podformúl formuly $\Box(p \vee q) \Rightarrow (\Box p) \vee (\Box q)$

| Formula | w_1 | w_2 | w_3 |
|---|-------|-------|-------|
| p | 1 | 0 | 1 |
| q | 1 | 1 | 0 |
| $\Box p$ | 0 | 0 | 1 |
| $\Box q$ | 0 | 1 | 0 |
| $\Box p \vee \Box q$ | 0 | 1 | 1 |
| $p \vee q$ | 1 | 1 | 1 |
| $\Box(p \vee q)$ | 1 | 1 | 1 |
| $\Box(p \vee q) \Rightarrow (\Box p \vee \Box q)$ | 0 | 1 | 1 |



Obr. 31. Sémantické tablo pre falzifikáciu tautologičnosti formuly $\varphi = \Box(p \vee q) \Rightarrow (\Box p) \vee (\Box q)$.

5.1 Formulácia modálnej predikátovej logiky

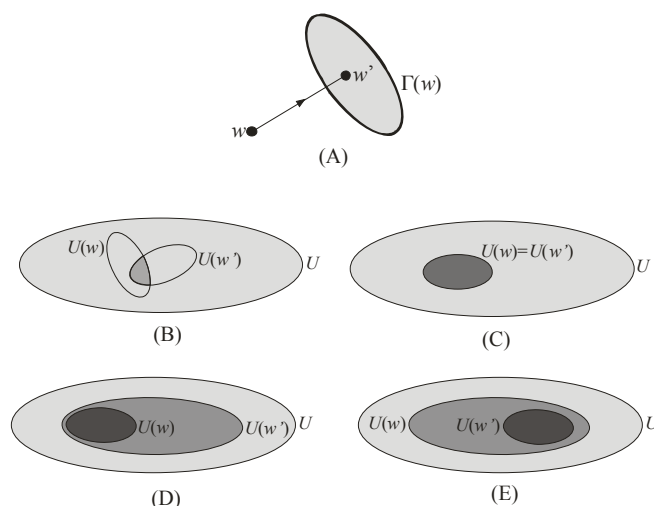
V tejto kapitole vykonáme kombináciu predikátovej (pozri kapitoly 4) a modálnej logiky, jej výsledkom bude modálna predikátová logika, ktorá súčasne obsahuje tak unárne modálne spojky \Box a \Diamond , ako aj kvantifikátory \forall a \exists . Syntax modálnej predikátovej logiky jednoducho získame, ak napríklad formuláciu syntaxe predikátovej logiky rozšírime o obe modálne spojky. Sémantika takto vzniknutej modálnej predikátovej logiky žiaľ už nie je taká jednoduchá, aká je sémantika tak modálnej logiky (založenej na Kripkeho modeli dostupných svetov), ako aj predikátovej logiky. Nové zovšeobecnenie spočíva v tom, že univerzum U závisí od sveta $w \in W$ v ktorom je uvažovaná formula φ modálnej predikátovej logiky študovaná. Pre každý svet $w \in W$ vyberieme podmnožinu $U(w) \subseteq U$, ktorá špecifikuje kvantifikátory vzhľadom k danému svetu. Budeme uvažovať tieto tri alternatívy (pozri obrázok 32)

$$(w, w') \in R \Rightarrow \begin{cases} U(w) = U(w') & (\text{konštantné univerzum, diagram } C) \\ U(w) \subseteq U(w') & (\text{rastúce univerzum, diagram } D) \\ U(w) \supseteq U(w') & (\text{zmenšujúce sa univerzum, diagram } E) \end{cases} \quad (54)$$

Pre modálnu predikátovú logiku budeme špecifikovať Kripkeho sémantický model v zjednodušenom tvare

$$\mathcal{M} = (W, R, \mathcal{V}, w_0, \{U(w)\}_{w \in W}) \quad (55)$$

kde $W = \{w, w', w'', \dots\}$ je **množina svetov** a $R \subseteq W \times W$ je binárna relácia **dostupnosti** nad množinou W . Diagram A na obr. 32 ilustruje túto reláciu alternatívnym spôsobom pomocou podmnožín $\Gamma(w) \subseteq W$, kde $\Gamma(w) = \{w'; (w, w') \in R\}$ obsahuje susedov – nasledovníkov sveta w . Výraz \mathcal{V} špecifikuje **pravdivostnú interpretáciu** formúl modálnej predikátovej logiky, obsahuje tak pravdivostnú interpretáciu predikátovej logiky, ako aj modálnej logiky. Pretože prezentujeme zjednodušený sémantický model modálnej predikátovej logiky, nebudeme bližšie špecifikovať sémantický interpretačný postup obsiahnutý v operátore \mathcal{V} . Výraz w_0 špecifikuje **vybraný svet** z W , vzhľadom ku ktorému je zostrojený model \mathcal{M} . Konečne, $\{U(w)\}_{w \in W}$ je **množina univerzít** špecifikovaných pre všetky možné svety z W . Predpokladáme, že tieto množiny vyhovujú jednej z relácií z (54), t. j. sú konštantné, rastúce, alebo zmenšujúce sa.



Obr. 32. (A) Schematické znázornenie dostupných svetov $w' \in \Gamma(w)$ z aktuálneho sveta w . (B) Vo všeobecnosti podmnožiny $U(w) \subseteq U$ sú rôzne a nemusia vyhovovať žiadnej relácii medzi podmnožinami. (C) Znázornenie **konštantného univerzá**, kde w a w' sú ľubovoľné svety z W . (D) **Rastúce univerzá**, pre dvojicu $w' \in \Gamma(w)$ (alebo $(w, w') \in R$) platí $U(w) \subseteq U(w')$. (E) **Zmenšujúce sa univerzá**, pre dvojicu $w' \in \Gamma(w)$ platí $U(w) \supseteq U(w')$.

Tabuľka 12. Elementárne formuly modálnej predikátovej logiky

| | | | |
|---|---------------------------------------|----|--|
| 1 | $w_0 \models \diamond(\exists x)P(x)$ | 1' | $w_0 \models (\exists x)\diamond P(x)$ |
| 2 | $w_0 \models \diamond(\forall x)P(x)$ | 2' | $w_0 \models (\forall x)\diamond P(x)$ |
| 3 | $w_0 \models \square(\exists x)P(x)$ | 3' | $w_0 \models (\exists x)\square P(x)$ |
| 4 | $w_0 \models \square(\forall x)P(x)$ | 4' | $w_0 \models (\forall x)\square P(x)$ |

Nech φ je formula modálnej logiky, skutočnosť, že táto formula je pravdivá vo zvolenom svete $w_0 \in W$ budeme znázorňovať výrazom $w_0 \models \varphi$. V tab. 12 je uvedených 8 elementárnych formúl modálnej predikátovej logiky s predikátmi $P(x)$, budeme študovať ich pravdivosť vo vybranom svete w_0 , pričom budeme predpokladať, že univerzá sú konštantné, rastúce, alebo znižujúce sa.

Pre ilustráciu zvoleného postupu budeme podrobne študovať metódou prirodzenej dedukcie formulu 1 z tab. 12, $w_0 \models \diamond(\exists x)P(x)$, budeme vyšetrovať, či táto formula implikuje formulu $w \models P(a)$, kde $\exists w \in \Gamma(w_0)$ a $\exists a \in U(w)$

| | | |
|----|--|-----------------------------|
| 1. | $w_0 \models \diamond(\exists x)P(x)$ | |
| 2. | $w \models (\exists x)P(x)$ | $\exists w \in \Gamma(w_0)$ |
| 3. | $w \models P(a)$ | $\exists a \in U(w)$ |
| 4. | $(w_0 \models \diamond(\exists x)P(x)) \Rightarrow (w \models P(a))$ | |

Riadok 2 vznikol z riadku 1 tak, že sme odstránili vo svete w_0 modálny operátor \diamond , v nasledujúcom kroku (riadok 3) sme konkretizovali existenčný kvantifikátor z formuly na riadku 2. Na záver (riadok 4) sme pomocou vety o dedukcii previedli predpoklad na ľavú stranu dôsledku pomocou implikácie. Týmto sme ukázali, že pôvodná formula $w_0 \models \diamond(\exists x)P(x)$ implikuje záver $w \models P(a)$. Tento postup môžeme zopakovať aj v inverznom poriadku

| | | |
|----|--|-----------------------------|
| 1. | $w \models P(a)$ | $\exists a \in U(w)$ |
| 2. | $w \models (\exists x)P(x)$ | $\exists w \in \Gamma(w_0)$ |
| 3. | $w_0 \models \diamond(\exists x)P(x)$ | |
| 4. | $(w \models P(a)) \Rightarrow (w_0 \models \diamond(\exists x)P(x))$ | |

Týmto sme dokázali, že platí ekvivalencia $(w_0 \models \diamond(\exists x)P(x)) \equiv (w \models P(a))$, t. j. relácia $w_0 \models \diamond(\exists x)P(x)$ je pravdivá práve vtedy, keď je pravdivá $w \models P(a)$ v niektorom svete susednom so svetom w_0 , $\exists w \in \Gamma(w_0)$, a pre niektorý objekt $\exists a \in U(w)$. Týmto spôsobom dostaneme tieto formuly

$$w_0 \models \diamond(\exists x)P(x) \Rightarrow w \models P(a) \quad (\text{pre } \exists w \in \Gamma(w_0), \exists a \in U(w)) \quad (56a)$$

$$w_0 \models (\exists x)\diamond P(x) \Rightarrow w \models P(a) \quad (\text{pre } \exists w \in \Gamma(w_0), \exists a \in U(w_0)) \quad (56a')$$

$$w_0 \models \diamond(\forall x)P(x) \Rightarrow w \models P(t) \quad (\text{pre } \exists w \in \Gamma(w_0), \forall t \in U(w)) \quad (56b)$$

$$w_0 \models (\forall x)\diamond P(x) \Rightarrow w \models P(t) \quad (\text{pre } \exists w \in \Gamma(w_0), \forall t \in U(w_0)) \quad (56b')$$

$$w_0 \models \square(\exists x)P(x) \Rightarrow w \models P(a) \quad (\text{pre } \forall w \in \Gamma(w_0), \exists a \in U(w)) \quad (56c)$$

$$w_0 \models (\exists x)\Box P(x) \Rightarrow w \models P(a) \quad (\text{pre } \forall w \in \Gamma(w_0), \exists a \in U(w_0)) \quad (56c')$$

$$w_0 \models \Box(\forall x)P(x) \Rightarrow w \models P(t) \quad (\text{pre } \forall w \in \Gamma(w_0), \forall t \in U(w)) \quad (56d)$$

$$w_0 \models (\forall x)\Box P(x) \Rightarrow w \models P(t) \quad (\text{pre } \forall w \in \Gamma(w_0), \forall t \in U(w_0)) \quad (56d')$$

Tieto elementárne formuly modálnej predikátovej logiky môžeme zhrnúť ako elementárne predĺženia v metóde sémantických tabiel pre súčasné odstránenie modálnej spojky a kvantifikátora (pozri tab. 13).

Tabuľka 13. Výsledky metódy sémantických tabiel pre elementárne formuly modálnej predikátovej logiky z tab. 12

| | | | | | |
|---|---|---|----|--|---|
| 1 | $w_0 \models \Diamond(\exists x)P(x)$ \downarrow $w \models P(a)$ | $\exists w \in \Gamma(w_0)$ $\exists a \in U(w)$ | 1' | $w_0 \models (\exists x)\Diamond P(x)$ \downarrow $w \models P(a)$ | $\exists a \in U(w_0)$ $\exists w \in \Gamma(w_0)$ |
| 2 | $w_0 \models \Diamond(\forall x)P(x)$ \downarrow $w \models P(t)$ | $\exists w \in \Gamma(w_0)$ $\forall t \in U(w)$ | 2' | $w_0 \models (\forall x)\Diamond P(x)$ \downarrow $w \models P(t)$ | $\forall t \in U(w_0)$ $\exists w \in \Gamma(w_0)$ |
| 3 | $w_0 \models \Box(\exists x)P(x)$ \downarrow $w \models P(a)$ | $\forall w \in \Gamma(w_0)$ $\exists a \in U(w)$ | 3' | $w_0 \models (\exists x)\Box P(x)$ \downarrow $w \models P(a)$ | $\exists a \in U(w_0)$ $\forall w \in \Gamma(w_0)$ |
| 4 | $w_0 \models \Box(\forall x)P(x)$ \downarrow $w \models P(t)$ | $\forall w \in \Gamma(w_0)$ $\forall t \in U(w)$ | 4' | $w_0 \models (\forall x)\Box P(x)$ \downarrow $w \models P(t)$ | $\forall t \in U(w_0)$ $\forall w \in \Gamma(w_0)$ |

$$\begin{array}{c}
 w_0 \models \Diamond(\exists x)P(x) \Rightarrow (\exists x)\Diamond P(x) \\
 \downarrow \\
 w_0 \models \Diamond(\exists x)P(x) \wedge (\forall x)\Box \neg P(x) \\
 \downarrow \\
 \begin{array}{l}
 w_0 \models \Diamond(\exists x)P(x) \\
 w_0 \models (\forall x)\Box \neg P(x)
 \end{array} \\
 \downarrow \\
 \begin{array}{l}
 w \models (\exists x)P(x) \quad \exists w \in \Gamma(w_0) \\
 w_0 \models \Box \neg P(t) \quad \forall t \in U(w_0)
 \end{array} \\
 \downarrow \\
 \begin{array}{l}
 w \models P(a) \quad \exists a \in U(w) \\
 w' \models \neg P(t) \quad \forall w' \in \Gamma(w)
 \end{array}
 \end{array}$$

Obr. 33. Sémantické tablo formuly $\Diamond(\exists x)P(x) \Rightarrow (\exists x)\Diamond P(x)$. Ak položíme $w = w'$ a $a = t$, posledné dva riadky produkujú kontradikciu, potom formula (57a) je tautológia pre konštantné univerzá a taktiež aj pre zmenšujúce sa univerzá (pozri obr. 34).

Teraz obrátíme našu pozornosť na zložitejšie formuly modálnej predikátovej logiky, menovite budeme študovať **Barcanovej formuly** [14]

$$\diamond(\exists x)P(x) \Rightarrow (\exists x)\diamond P(x) \quad (57a)$$

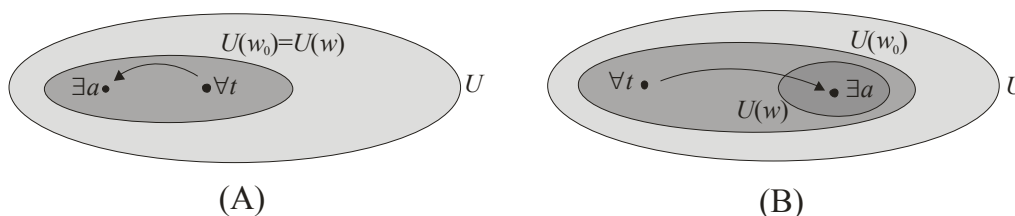
$$(\forall x)\Box P(x) \Rightarrow \Box(\forall x)P(x) \quad (57b)$$

a **konverzné Barcanovej formuly**

$$\Box(\forall x)P(x) \Rightarrow (\forall x)\Box P(x) \quad (58a)$$

$$(\exists x)\diamond P(x) \Rightarrow \diamond(\exists x)P(x) \quad (58b)$$

Tieto formuly zohrali dôležitú úlohu v rozvoji teórie modálnej predikátovej logiky, kde boli použité ako základný testovací príklad pre rozvoj rôznych sémantických teórií tohto typu logiky. V prvom kroku použijeme metódu sémantických tabiel (pozri obr. 33) pre zistenie pravdivosti Barcanovej formuly (57a)



Obr. 34. (A) diagram znázorňuje prípad, keď univerzá sú konštantné, v tomto prípade môžeme stotožniť ľubovoľný objekt t s vybraným objektom a . (B) diagram znázorňuje situáciu, keď univerzá sú znižujúce sa, t. j. $(w_0, w) \in R \Rightarrow U(w_0) \supseteq U(w)$, v tomto prípade ľubovoľný objekt t môže byť stotožnený s daným objektom a .

Podobným spôsobom môžeme študovať aj ostatné Barcanovej formuly (57b) a (58a-b), výsledky sú zhrnuté v tab. 14.

Tabuľka 14. Tautologičnosť Barcanovej formúl pre rôzne typy univerzálnych množín

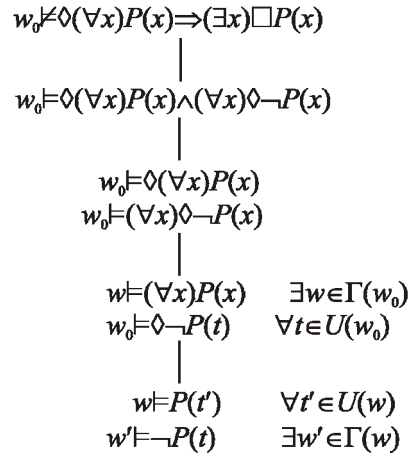
| # | Formula | konštantné univerzum | rastúce univerzum | zmenšujúce sa univerzum |
|----------|--|----------------------|-------------------|-------------------------|
| 1 | $\diamond(\exists x)P(x) \Rightarrow (\exists x)\diamond P(x)$ | áno | nie | áno |
| 2 | $(\forall x)\Box P(x) \Rightarrow \Box(\forall x)P(x)$ | áno | nie | áno |
| 1 \Box | $\Box(\forall x)P(x) \Rightarrow (\forall x)\Box P(x)$ | áno | áno | nie |
| 2 \Box | $(\exists x)\diamond P(x) \Rightarrow \diamond(\exists x)P(x)$ | áno | áno | nie |

Z tab. 14 vyplýva zaujímavý poznatok, že pre konštantné univerzá modálne operátory „komutujú“. To znamená, že ak sa použije klasický Kripkeho sémantický model pre modálnu predikátovú logiku s konštantnými univerzami (napr. pre každé $w \in \mathcal{W}$ nech platí $U(w) = U_0 \subseteq U$), potom všetky Barcanovej formuly sú tautológie.

Na záver tejto kapitoly budeme študovať metódou sémantického tabla formulu modálnej predikátovej logiky, ktorá nie je tautológiou

$$\diamond(\forall x)P(x) \Rightarrow (\exists x)\Box P(x) \tag{59}$$

Na obr. 35 je znázornené sémantické tablo pre túto formulu, pretože vo všeobecnosti $w \neq w'$, študovaná formula (59) nie je tautológia.



Obr. 35. Sémantické tablo dôkazu tautologičnosti formuly $\diamond(\forall x)P(x) \Rightarrow (\exists x)\Box P(x)$.

Vlastnosť netautologičnosti formuly (59) môžeme verifikovať tak, že pomocou sémantického tabla z obr. 35 navrhne taký model v ktorom táto formula je nepravdivá, t. j. nie je tautológia (môžeme povedať, že sme falzifikovali predpoklad tautologičnosti). Nech model \mathcal{M} je špecifikovaný dvojzložkovým univerzom $U = \{a, b\}$ a $\Gamma(w_0) = \{w, w'\}$, $\Gamma(w) = \Gamma(w') = \emptyset$. Postulujeme, že vo svete w je predikát P pravdivý pre všetky objekty z univerza U a vo svete w' je tento predikát P nepravdivý pre všetky objekty z univerza U , pozri tab. 15.

Tabuľka 15. Konštrukcia jednoduchého modelu pre falzifikáciu tautologičnosti formuly (59).

| | w_0 | w | w' |
|--|-------|-----|------|
| $P(a)$ | * | 1 | 0 |
| $P(b)$ | * | 1 | 0 |
| $w_0 \models \diamond\neg P(a)$ | 1 | * | * |
| $w_0 \models \diamond\neg P(b)$ | 1 | * | * |
| $w \models (\forall x)P(x)$ | * | 1 | * |
| $w_0 \models (\forall x)\diamond\neg P(x)$ | 1 | * | * |
| $w_0 \models \diamond(\forall x)P(x)$ | 1 | * | * |
| $w_0 \not\models (\exists x)\Box P(x)$ | 0 | * | * |
| $w_0 \models \diamond(\forall x)P(x) \Rightarrow (\exists x)\Box P(x)$ | 0 | * | * |

To znamená, že pre formulu $\diamond(\forall x)P(x) \Rightarrow (\exists x)\Box P(x)$ existuje model v ktorej je nepravdivá, čiže formula nie je tautológia.

6 Záver

Cieľom tejto kapitoly bolo ukázať použitie techniky sémantických tabiel v klasických aj neklasických logikách. Idea sémantických tabiel má svoje prvopočiatky už v Gentzenovej prirodzenej dedukcii, ktorá bola vytvorená 30. rokoch minulého storočia, jej základy boli vytvorené v polovici 50. rokov minulého storočia logikmi Bethom a Hintikkom a do konečnej formy rozpracované v polovici 60. rokov minulého storočia logikom Smullyanom. Mimoriadne postavenie majú sémantické tablá v modálnej logike, kde tvoria vlastne jedinú univerzálnu metódu na stanovenie pravdivosti formúl modálnej logike. Taktiež, ako bolo ukázané mladým logikom Saulom Kripkem počiatkom 60. rokov minulého storočia, pomocou nich môžu byť jednotlivé modálne logiky klasifikované vlastnosťami binárnej relácie dostupnosti medzi možnými svetmi, čo podstatne zjednodušuje pôvodný ťažkopádny syntaktický prístup k špecifikácii rôznych modálnych logík. Môžeme teda konštatovať, že technika sémantických tabiel má v súčasnosti prominentné postavenie v matematickej logike, ako jednoduchý a názorný prostriedok na stanovenie tautologičnosti formúl, určenie modelov teórií, v inverznom poradí sú sémantické tablá identické Gentzenovej prirodzenej dedukcii, a čo je najdôležitejšie, pre logiky používajúce Kripkeho sémantiku dostupných svetov (modálne logiky a ich zovšeobecnenie do predikátovej formy) technika sémantických tabiel je vlastne jedinou technikou na vyšetovanie formúl na sémantickej úrovni.

Pod'akovanie: Táto kapitola vznikla za podpory grantovej agentúry VEGA SR v rámci grantových úloh 1/0804/08 a 1/0141/10.

Literatúra

- [1] Aliseda, A.: *Abductive Reasoning. Logical Investigations into Discovery and Explanation*. Springer, Berlin 2006.
- [2] Beth, E. W.: *The Foundations of Mathematics*. North-Holland, Amsterdam 1959.
- [3] Beth, E. W.: Semantic Entailment and Formal Derivability. *Mededelingen der Koninklijke Nederlandse Akademie van Wetenschappen*. **18** (1955) 309-342.
- [4] Blackburn, P., van Benthem, J., Wolter, F. (eds.): *Handbook of Modal Logic*. Elsevier, Amsterdam 2007.
- [5] D'Agostino, M., Gabbay, D. M., Hahnle, R., Posegga, J. (eds.): *Handbook of Tableau Methods*. Springer, Berlin 1999.
- [6] Fitting, M., Mendelsohn, R. L.: *First-Order Modal Logic*. Kluwer Academic Publishers, Dordrecht 1998.

-
- [7] Garson, J. W.: *Modal Logic for Philosophers*. Cambridge University Press, Cambridge 2006.
- [8] Gödel, K.: Eine Interpretation des intuitionistischen Aussagenkalküls. *Ergebnisse eines mathematischen Kolloquiums* **4** (1932) 39-40.
- [9] Hintikka, J.: A new approach to sentential logics. *Soc. Scient. Fennica, Comm. Phys.-Math.* **17** (1953) 2.
- [10] Hintikka, J.: Form and content in quantification theory. *Acta Philosophica Fennica* **8** (1955) 8–55.
- [11] Kvasnička, V., Pospíchal, J.: *Algebra a diskrétna matematika*. Vydavateľstvo STU, Bratislava 2008.
- [12] Kvasnička, V., Pospíchal, J.: *Matematická logika*. Vydavateľstvo STU, Bratislava 2006.
- [13] Lewis, C. I.: *A Survey of Symbolic Logic*. University of California Press, Berkeley 1918.
- [14] Marcus, R. B.: A Functional Calculus of First Order Based on Strict Implication. *Journal of Symbolic Logic* **11** (1946) 1-16.
- [15] Peregrin, J.: *Logika a logiky*. Academia, Praha, 2004.
- [16] Peregrin, J., Svoboda, V.: *Od jazyka k logice*. Academia, Praha 2009.
- [17] Priest, G.: *An Introduction to Non-Classical Logic*. Cambridge University Press, Cambridge 2004.
- [18] Šefránek, J.: *Inteligencia ako výpočet*. IRIS, Bratislava, 2000.
- [19] Smullyan, R. M.: A unifying principle in quantification theory. *Proceedings of the National Academy of Sciences* **49** (1963) 828–832.
- [20] Smullyan, R. M.: Analytic natural deduction. *Journal of Symbolic Logic* **30** (1965) 123–139.
- [21] Smullyan, R. M.: *First-Order Logic*. Springer-Verlag, Berlin 1968 (slovenský preklad *Logika prvého rádu*. ALFA, Bratislava 1979).
- [22] Szabo, M. E. (ed.): *The Collected Papers of Gerhard Gentzen*. North-Holland, Amsterdam 1969.

Náčrt teórie potencialít jazyka matematiky

Ladislav KVASZ¹

Abstrakt. Cieľom predkladanej state je analyzovať rôzne spôsoby ako jazyk ovplyvňuje kogníciu. Snažím sa odlíšiť šesť rôznych avšak navzájom previazaných vlastností jazyka, ktoré možno označiť ako abstrakčný potenciál, diferenciacny potenciál, explikačný potenciál, klasifikačný potenciál, regularizačný potenciál a unifikacny potenciál. Pre každý z nich je uvedená stručná charakterizácia a následne na niekoľkých konkrétnych príkladoch je ilustrovaný ich kognitívny prínos.

1 Úvod

Vzťah jazyka a kognície je zložitý a k jeho skúmaniu možno pristupovať z rôznych hľadísk. Jedno z hľadísk na skúmanie vzťahu jazyka a kognície ponúka história matematiky, ktorá skúma vývoj našej schopnosti argumentovať, dokazovať, počítať či konštruovať matematické objekty. Faktom, ktorý asi nemá zmysel popierať je, že dnes vieme toho zdôvodniť, dokázať, vypočítať či skonštruovať nepomerne viac ako naši predchodcovia v 15. či 17. storočí. Jedným z faktorov, ktorý zásadným spôsobom ovplyvňuje rozvoj matematického myslenia je jazyk matematiky. V knihe *Patterns of Change, Linguistic Innovations in the Development of Classical Mathematics* [19] som sa pokúsil opísať jazykové inovácie, ktoré ovplyvnili rozvoj matematiky. Ukázalo sa, že v rozvoji matematiky možno rozlíšiť tri typy lingvistických inovácií.

Prvý typ lingvistických inovácií možno ilustrovať na línii vedúcej od *syntetickej geometrie cez analytickú geometriu po fraktálnu geometriu*, a nazval som ho **re-prezentácie**. Pre tento typ je charakteristická zásadná zmena spôsobu konštrukcie objektov (pomocou kružidla a pravítka, ako to robí syntetická geometria; pomocou súradnej sústavy a formule, ako to robí analytická geometria; alebo pomocou iteratívneho procesu, ako sa to deje vo fraktálnej geometrii). Druhý typ inovácií možno ilustrovať na línii *euklidovská geometria, projektívna geometria, Lobačevského geometria, Kleinov erlangenský program*. Nazval som ho **objektácie**. Objektácia, na rozdiel od predošlého typu zmien, nemení objekty, ale skôr priestor, do ktorého sú tieto objekty zasadené. Tretí typ inovácií nazývam **re-formulácie**. Spája rôzne axiomatizácie euklidovskej geometrie. Výklad jednotlivých typov inovácií možno nájsť v spomínanej knihe. Zvláštnosťou tejto knihy je, že pri opise rôznych typov zmien jazyka matematiky sú jednotlivé typy vyložené každý iným spôsobom.

¹ Katedra matematiky a didaktiky matematiky, Ped. Fakulta Univerzity Karlovy, M.D. Rettigové 4, 116 39 Praha a Katedra filozofie, Filozofická Fakulta Katolíckej Univerzity, Hrabovská cesta 1, 03 401 Ružomberok, E-mail: kvasz@fmph.uniba.sk

Pri výklade re-prezentácií sú analyzované *potenciality jazyka* ako *logická sila*, *expresívna sila*, *explanatorická sila* a *integratívna sila*², pričom vývin matematiky je v rovine re-prezentácií vyložený ako nárast logickej, expresívnej, explanatorickej a integratívnej sily jej jazyka. Matematika je schopná postupne dokázať stále *silnejšie vety*, opísať stále *zložitejšie objekty*, vysvetliť stále *jemnejšie súvislosti* a vo svojom univerze odкрýva stále *vyšší stupeň jednoty*. Vývin matematiky v rovine re-prezentácií vykazuje teda znaky pokroku. Naproti tomu objektácie sú v knihe charakterizované pomocou *formálnych aspektov jazyka* ako *epistemický subjekt*, *horizont*, *pozadie* a *ideálne prvky*³, pričom vývin určitej matematickej teórie je interpretovaný ako postupnosť zmien subjektu, horizontu, pozadia a ideálnych prvkov. V rovine objektácií je tak vývin opísaný nie ako pokrok, ale ako striedanie štruktúr. Je prirodzené položiť si otázku, či je tento stav *nevyhnutný* a jednotlivé spôsoby výkladu sú jednoznačne spojené s príslušnými typmi zmien, pri opise ktorých boli použité, alebo je toto spojenie *náhodné* a určitý spôsob výkladu možno použiť aj na opis iných typov zmien.

Zdá sa, že správna je druhá alternatíva a aj u re-prezentácií je možné nájsť ich formálne aspekty a u objektácií je možné opísať ich potenciality. O splnenie druhej z týchto úloh sa usiluje predkladaná stať. Jej cieľom je *pre objektácie nájsť potenciality jazyka*, ktoré by mohli pomôcť porozumieť dynamike objektácií podobne, ako pojmy logická či expresívna sila jazyka pomáhajú porozumieť dynamike re-prezentácií⁴. Keď sa to podarí, je možné vytvoriť jednotiaci pohľad na vývin jazyka matematiky a vytvoriť zoznam vlastností (nazývam ich potencialitami jazyka, lebo sa viažu k možnostiam, ktoré jazyk otvára pre myslenie), umožňujúcich porozumieť zmenám, ktoré jazyk vnáša do myslenia. Teória potencialít jazyka tak môže byť zaujímavá pre každého, kto sa zaoberá skúmaním vzťahu jazyka a kognície. V stati som zvolil jazyk matematiky preto, lebo tento jazyk má stabilnú štruktúru a tak jeho zmeny možno ľahko sledovať. Verím však, že podobné zmeny možno nájsť aj v iných jazykoch.

Re-prezentácie, objektácie a re-formulácie opísané v *Patterns of Change* predstavujú zmeny jazyka v rôznych hĺbkach. V prípade každej z týchto zmien má potencialita jazyka iný charakter. V prípade re-prezentácií som potenciality jazyka označil termínom *sila*, pričom som zaviedol *logickú*, *expresívnu*, *explanatorickú* a *integratívnu* silu. Termín sila označuje schopnosť jazyka vytvoriť čosi zásadne nového, schopnosť radikálne zmeniť celkový spôsob myslenia. Ako príklad možno uviesť Descartovu analytickú geometriu, ktorá vytvorila úplne nové univerzum

² Podrobný výklad týchto pojmov možno nájsť v knihe *Patterns of Change* ([19], s. 14 – 84) alebo v stati [13]. Pre slovenského čitateľa je dostupnejšia kniha *Gramatika zmeny* [12], v ktorej sa uvedený výklad nachádza na stranách 63 až 95. V stati sa budem odvolávať na *Patterns of Change*, lebo oproti slovenskej knihe obsahuje niektoré upresnenia. Pre sledovanie výkladu sú však tieto upresnenia nepodstatné, a preto budem uvádzať aj odvolávky na slovenský text.

³ Výklad týchto pojmov možno nájsť v ([19], s. 107 – 200), resp. v ([12], s. 109 – 142) a v statiach [11] a [18].

⁴ Táto stať sa usiluje spojiť analytickú metódu s historickým pohľadom. Analytická filozofia v ranom období svojho rozvoja odmietala historickú perspektívu. Avšak od 60-tych rokov dvadsiateho storočia dochádza v analytickej filozofii k otvoreniu sa voči historickej perspektíve. Stačí spomenúť práce J. Hintikka či M. Friedmanna. Pozoruhodný príspevok, ktorého pojmá vývinu formálnych metód je veľmi blízke pojatiu v tejto stati, predstavuje kniha Vojtěcha Kolmana *Filosofie čísla* ([9]).

geometrických objektov a radikálne pretvorila celú matematiku aj fyziku. Na vzniku analytickej geometrie možno ilustrovať nárast logickej, expresívnej, explanatorickej i integratívnej sily jazyka matematiky (pozri [19], s. 37 – 47; resp. [12], s. 80 – 88). Ukazuje sa však, že zoznam štyroch potencialít, pomocou ktorých sú charakterizované re-prezentácie, možno doplniť o ďalšie dve – o *metodickú* a *metaforickú* silu. Ich výkladu je venovaná prvá časť tejto state.

V prípade objektácií navrhujem potenciality jazyka, ktorých nárast tieto zmeny sprevádza, označiť termínom *potenciál*, pričom budem hovoriť o *abstrakčnom*, *diferenciačnom*, *explikačnom*, *klasifikačnom*, *regularizačnom* a *unifikačnom* potenciáli jazyka. Použitím spojenia *potenciál jazyka* chcem naznačiť, že potenciality jazyka zviazané s objektáciami už nie sú tak prevratné ako *sily*. Potenciál neprináša niečo úplne nového. Vedie k obohateniu možností, ktoré boli v jazyku implicitne prítomné. Ako príklad možno vziať objav neeuklidovskej geometrie. Nepopieram význam tohto objavu, ale nemožno tvrdiť, že by neeuklidovská geometria zmenila univerzum tvarov, porovnateľne s tým, ako ho zmenila analytická geometria. Väčšina objektov neeuklidovskej geometrie je nám z euklidovského sveta dobre známa. V neeuklidovskej geometrii získavajú tieto objekty rad prekvapujúcich vlastností, ale ich súvis s euklidovskými objektmi nie je prerušený. Napríklad kosínusová veta má síce v neeuklidovskej geometrii odlišný tvar ako v euklidovskej geometrii, ale stále je to kosínusová veta, teda veta umožňujúca určiť veľkosť uhla v trojuholníku pomocou dĺžok jeho troch strán. Navyše kosínusová veta euklidovskej geometrie je limitným prípadom kosínusovej vety geometrie neeuklidovskej ([19], s. 127; resp. [12], s. 117). Teda aj keď jazyk neeuklidovskej geometrie má väčší *abstrakčný*, *diferenciačný*, *explikačný* a *unifikačný* potenciál než jazyk euklidovskej geometrie, čo sa *logickej*, *expresívnej*, *explanatorickej* a *integratívnej* sily týka, majú ju obe geometrie – euklidovská aj neeuklidovská – rovnakú. Opisu potencialít jazyka spojených s objektáciami je venovaná druhá časť state.

Na záver sa pokúsim opísať potenciality jazyka ktoré sa menia pri re-formuláciách. V tomto prípade navrhujem potenciality nazývať *cnosťami*. Cnosť nespočíva ani vo vytvorení niečoho zásadne nového (ako *sila*), ani v explicitnom vyjadrení niečoho implicitne už prítomného (ako *potenciál*). Ide skôr o schopnosť vybrať vhodný variant z množstva alternatív, ktoré sprístupnil potenciál. Medzi cnosťami, pomocou ktorých možno opísať re-formulácie patria *všeobecnosť*, *presnosť*, *prirodzenosť*, *jednoznačnosť*, *jednoduchosť* a *univerzálnosť*. Re-formulácie majú za cieľ zovšeobecniť tvrdenia, zvýšiť presnosť výpočtov, dosiahnuť prirodzenosť konštrukcií, jednoznačnosť formulácií, zjednodušenie dôkazov a univerzálnosť metód. Výkladu cností je venovaná záverečná časť state.

2 Doplnenie potencialít re-prezentácií

V *Patterns of Change* je každá re-prezentácia charakterizovaná pomocou štyroch potencialít:

1. *logickej sily*, ktorá ukazuje nakoľko zložité formuly je možné v jazyku dokázať
2. *expresívnej sily*, ktorá ukazuje, čo nového, čo sa v predošlých štádiách vymykalo vyjadreniu, teraz jazyk umožňuje vyjadriť

3. *explanatorickej sily*, ktorá ukazuje, ako nový jazyk umožňuje vysvetliť zlyhania jazyka ktoré boli na predošlom štádiu nepochopiteľné
4. *integratívnej sily*, ktorá ukazuje, ako nový jazyk umožňuje vidieť jednotu a poriadok tam, kde na báze predošlého jazyka sa ukazovali len navzájom nesúvisiace prípady

Zdá sa, že s každou re-prezentáciou sú spojené ešte ďalšie dve potenciality, a to **metodická sila a metaforická sila**. Tieto potenciality som si pri písaní *Patterns of Change* neuvedomil, lebo som sledoval dynamiku vnútorného rozvoja matematiky, kým metodická a metaforická sila jazyka sa prejavujú skôr pri aplikácii matematiky. Metodická a metaforická sila jazyka matematiky mali rozhodujúcu úlohu pri vedeckej revolúcii, keď umožnili vznik *zmiešaných disciplín* (metodická sila), a *metaforickej oblasti paradigmy* (metaforická sila – pozri [16]).

Metodickú silu jazyka matematiky možno rozpoznať, keď si uvedomíme, že s každou re-prezentáciou je spojená určitá typická metóda. Napríklad s algebrou sa spája analytická metóda a s analytickou geometriou metóda redukcie. Sú to štandardné postupy, ktoré príslušný jazyk ponúka a ktoré sú použiteľné pri riešení širokého spektra úloh. Podobne **metaforickú silu** jazyka matematiky si môžeme uvedomiť pri čítaní literatúry z biológie, ekonómie, psychológie či sociológie, kde sa matematický jazyk používa skôr ako zdroj analógií a metafor než v striktnom technickom zmysle.

2.1 Náčrt vývinu metodickej sily jazyka matematiky

Slovo náčrt som do názvu state ako aj do názvu tejto jej kapitoly zaradil preto, lebo uviesť pre každú reprezentáciu od aritmetiky až po teóriu množín výklad metódy, ktorá ilustruje metodickú silu jej jazyka, tak ako som to pre ostatné sily uviedol v *Patterns of Change* by viedlo ku značnému nárastu rozsahu state. Preto iba načrtnem príslušné metódy.

a. ELEMENTÁRNA ARITMETIKA bola z hľadiska explanatorickej a integratívnej sily charakterizovaná ako neexplanatorická a neintegratívna. Podobne ju možno charakterizovať aj ako **nemetodickú**, čo znamená, že postupy riešenia problémov, ktoré sa dochovali zo starého Egypta či Babylonu sú *ad hoc* a chýba im akákoľvek metodická jednota.

b. SYNTETICKÁ GEOMETRIA prináša pozoruhodnú metódu, ktorú použil Euklides v *Základoch*. Jej výklad možno nájsť v komentári Thomasa Heatha ([3], s. 129–131). Dôkaz určitej propozície či riešenie určitého problému pozostáva z piatich krokov. Prvým je **prótasis** (πρωτασις), t.j. vyslovenie tvrdenia, pri ktorom je propozícia sformulovaná vo všeobecnom tvare. Napríklad Propozícia 8 Knihy IV znie: „Do daného štvorca vpísať kruh.“

Druhým krokom je **ekthesis** (εκθεσις), t.j. preformulovanie tvrdenia do konkrétneho tvaru, v ktorom sa zavedie označenie: „Nech *ABCD* je daný štvorec, teda požaduje sa vpísať kruh do daného štvorca *ABCD*.“ Ekthesis je sprevádzaná náčrtom, ku ktorému sa označenie vzťahuje a v ktorom jasne vyznačí, čo je dané. Po **ekthesis** nasleduje často **diorismós** (διορισμός) t.j. upresnenie, v rámci ktorého sa vyjasní, za akých podmienok má úloha vôbec zmysel.

Tretím krokom je *kataskeyé* (κατασκευή), t.j. konštrukcia, pri ktorej sa prvky dané v *ekthesis* postupne dopĺňajú aby sa vytvoril objekt vyžadovaný v úlohe. Práve tento krok zvedol Kanta k presvedčeniu, že matematika je založená na konštrukciách a jej dôkazy majú syntetický charakter. Podľa Hintikku je tento krok nutný iba preto, lebo logika Euklidovej doby neumožňovala existenciu objektov postulovať, a preto namiesto postulovania bolo nutné objekty konštruovať [17].

Štvrtým krokom je *apódeixis* (αποδείξις), t.j. samotný dôkaz v užšom zmysle. Ide o to, že ani pri čisto dôkazových úlohách dôkaz spravidla nemôže nastúpiť hneď po *ekthesis*, lebo útvar, ktorý vystupuje v tvrdení je nutné doplniť o pomocné prvky (rôzne body, úsečky a oblúky kružníc), ktoré v zadaní nevystupujú, ale o ktoré sa opiera argumentácia dôkazu. Jednou z najkrajších ilustrácií takéhoto doplnenia je Euklidov dôkaz Pytagorovej vety. Až po tom, ako Euklides spustil výšku na preponu trojuholníka ktorou rozdelil štvorec nad preponou na dva obdĺžniky, stalo sa možným dokázať, že obsah jedného obdĺžnika je zhodný s obsahom štvorca nad jednou odvesnou, a obsah druhého obdĺžnika je zhodný s obsahom štvorca nad druhou odvesnou. Až keď sa útvar, ku ktorému sa vzťahuje tvrdenie vety, doplní, možno začať s dôkazom. U Euklida *apódeixis* nasleduje až po skončení *kataskeyé*, čo znamená, že je jasne oddelená syntetická, konštrukčná časť od samotného dôkazu, ktorý je čisto logickou argumentáciou.

Záverečným piatym krokom je *sympérasma* (συμπερασμα), t.j. záver, spočívajúci v návrate ku všeobecnému diskurzu, ktorý sme opustili v *ekthesis*, kedy sme všeobecné tvrdenie nahradili konkrétnym prípadom. *Sympérasma* spočíva spravidla v zopakovaní tvrdenia („Preto v danom štvorci bol vpísaný kruh.“). Vidíme, že euklidovský dôkaz kombinuje analytické kroky so syntetickými. Ako upozornili Hintikka a Friedman, Euklidov dôkaz nemohol byť plne analytický, lebo logika, ktorú mal k dispozícii bola príliš slabá. Tam, kde moderný dôkaz používa existenčnú axiómu, pomocou ktorej zabezpečí existenciu určitého objektu, musí Euklides tento objekt skonštruovať.

c. **ALGEBRA** prináša u Viëta *analytickú metódu*. Keď ju porovnáme s Euklidovou metódou, zistíme, že Viëte mení Euklidovu *ekthesis*. Pri *ekthesis* Euklides postupoval od toho, čo je dané k tomu, čo sa hľadá (čo je jedným z významov slova syntetický – k tomu čo je dané postupne pridávať nové prvky). Viëtovou inováciou jazyka algebry bolo zavedenie dvoch druhov písmen: „označme neznámu veličinu pomocou písmena *A* alebo nejakej inej samohlásky *E, I, O, U* alebo *Y* a dané veličiny pomocou písmen *B, G* a *D* alebo inými spoluhláskami“ ([25], s. 340). Táto inovácia umožňuje zapísať vzťahy medzi známymi a neznámymi prvkami v podobe algebraických rovníc a nahradiť *kataskeyé* ich riešením. Na riešenie rovníc zavádza Viëte postupy ako *antitéza* (prenesenie člena na druhú stranu rovnice pri súčasnej zmene znamienka), *hypobibazmus* (predelenie celej rovnice neznámou) a *parabolizmus* (predelenie rovnice koeficientom pri najvyššej mocnine neznámej).

Analytické umenie sa podľa Viëta skladá z troch častí. Prvou je *zetetické umenie*, teda umenie nájsť vzťah medzi veličinou, ktorú hľadáme, a veličinami, ktoré sú dané. *Zetetické umenie* je rozvinutím Euklidovej *ekthesis*, pričom algebraická *symbolika* umožňuje označiť prvky dané v zadaní, ako aj prvky neznáme. Geometrická „*ikonika*“ čosi také neumožňuje. Druhou je *poristické umenie*, pri ktorom sa skúma správnosť takto získanej rovnice (či spĺňa princíp homogenity, teda či sa v ňom nesčítavajú dĺžky

s obsahmi a objemami). Táto časť pripomína Euklidovu *diorismós*. Treťou zložkou je **exegetické umenie** pri ktorom sa z rovnice vyjadri veličina, ktorú hľadáme ([25], s. 320 – 321). Exegetické umenie nahrádza *kataskéyē*. Celé trojdielne analytické umenie Viète definuje ako vedu správneho hľadania v matematike. Svoje pojednanie končí slovami: „*analytické umenie, po tom ako mu bola daná trojitá forma zetetiky, poristiky a exegetiky si právom pripisuje problém všetkých problémov, ktorým je: nenechať žiaden problém nevyriešený.*“

Analytická metóda vo Vièetovej formulácii má rad idiosynkretických čŕt. V Descartovej *Geometrii* sa s ňou stretávame v zrelšej podobe. Descartes napríklad vynechal poristické umenie, keď zrušil princíp homogenity. Odhliadnuc od podobných technických detailov je užitočné porovnať analytickú metódu s Euklidovou, aby sme si uvedomili nárast metodologickej sily jazyka pri prechode od syntetickej geometrie k algebre. V *Patterns of Change* je rozdiel *logickej sily* jazyka syntetickej geometrie a jazyka algebry vyložený porovnaním úsečky neurčitej dĺžky s neznámou. Zásadný rozdiel medzi úsečkou neurčitej dĺžky, pomocou ktorej dosahuje všeobecnosť svojich dôkazov geometria, a neznámou, ktorá je základom všeobecnosti v algebre, je v tom, že geometrický obrázok **nesie informáciu o postupnosti krokov**, ktoré viedli k jeho vytvoreniu. Táto okolnosť je príčinou rozdielu logickej sily syntetickej geometrie a algebry. Analýza *metodickej sily* jazyka odhaľuje ďalší rozdiel. Geometria sice môže pomocou úsečky neurčitej dĺžky vyjadriť neznámu, **nedisponuje však prostriedkom na odlišenie neznámej od parametra**. Geometria dokáže vyjadriť *všeobecný objekt* určitého druhu (napríklad všeobecný trojuholník) tak, že nešpecifikuje jeho prvky; nedokáže však vyjadriť *všeobecné vzťahy*. Preto Euklides musí pri konštrukcii postupovať synteticky, využívajúc konkrétne vzťahy medzi objektmi uvedené v zadaní, ich pomocou vytvárať ďalšie konkrétne objekty v konkrétnych vzťahoch, až kým nedosiahne výsledný objekt. Viète naproti tomu vďaka vyjadreniu rozdielu medzi neznámou a parametrom dokáže aj *vzťahy* vyjadriť všeobecne (pomocou rovníc s parametrami) a potom ich upravovať.

d. ANALYTICKÁ GEOMETRIA, okrem toho, že od algebry preberá analytickú metódu, prináša aj vlastnú metódu, ktorou je **metóda redukcie**, keď pomocou súradnicovej sústavy prevedie geometrickú úlohu na úlohu algebraickú a tú rieši. U Descarta bola metóda redukcie geometrie kriviek na algebru polynómov súčasťou všeobecnej metódy redukcie, na ktorej bola založená karteziánska prírodoveda. Podľa Descartovho mechanistického pojatia prírody každý prírodný proces možno redukovať na mechanický pohyb rozpriestranenej substancie. Mechanický pohyb možno previesť na geometriu trajektórií jednotlivých častíc. Geometriu trajektórií, ako sme už uviedli, možno previesť na algebraický problém riešenia rovníc, ktoré príslušné trajektórie zadávajú.⁵ Descartes bol presvedčený, že algebraické rovnice sa podarí vyriešiť, čím po spätnom preklade výsledkov do jazyka prírodovedy dostaneme odpoveď na pôvodnú prírodovednú otázku.

Tu, podobne ako pri Vièetovej analytickej metóde, sa nesmieme zastaviť na idiosynkretickom charaktere Descartovej redukcie. Newton ukázal, že na opis

⁵ Až tento posledný krok predstavuje metódu redukcie, ilustrujúcu nárast metodologickej sily jazyka. Ostatné prvky veľkolepého karteziánskeho projektu redukcie sú skôr ilustráciou metaforickej sily jazyka algebry. Zo žiadneho z nich sa nepodarilo vytvoriť fungujúci metodický postup.

prírodných zákonov sú vhodnejšie diferenciálne rovnice než rovnice algebraické. Ani vieru v mechanický charakter všetkých pohybov sa nepodarilo udržať – mechanistický pohľad na prírodu vyvrátila špeciálna teória relativity, keď ukázala neudržateľnosť éteru, tvoriaceho mechanický substrát elektromagnetických vln. Napriek všetkým nedostatkom Descartovej metódy redukcie nemožno poprieť zásadný význam redukcie pre vedu. Stačí spomenúť redukciiu chémie na kvantovú mechaniku či redukciiu dedičnosti na molekulárnu biológiu. Redukcia je jednou z najúspešnejších metodologických stratégií vedy. Bez nadsádzky možno povedať, že ideu redukcie veda preberá od kartezianstva. Analytická geometria bola asi prvým jazykom, ktorý umožnil redukciiu jednej disciplíny na inú. Logická analýza jazyka analytickej geometrie môže preto pomôcť pochopiť, čo tento zásadný nárast metodologickej sily umožnilo.

e. DIFERENCIÁLNY A INTEGRÁLNY POČET prináša *metódu modelovania*. Táto mohla vzniknúť, pretože diferenciálny a integrálny počet má symbol pre ľubovoľnú funkciu. To, čo algebra dosiahla v rovine veličín – všeobecné pravidlá pre manipuláciu – prenáša matematická analýza do oblasti funkcií. Jazyk matematickej analýzy disponuje symbolom pre *neznámu funkciu*. Vďaka tomu môže zapísať vzťah neznámej funkcie k iným (znáмым) funkciám daným v zadaní úlohy (prípadne k deriváciám neznámej funkcie) v tvare diferenciálnej rovnice, integrálnej rovnice alebo rovnice variačného počtu analogicky s tým, ako Viëta zapisoval vzťah neznámej veličiny k iným (znáмым) veličinám daným v zadaní úlohy (prípadne k mocninám a odmocninám neznámej veličiny) v tvare algebraickej rovnice. Všade tam, kde hľadáme zákonitosť priebehu určitého procesu (t.j. vlastne neznámu funkciu) nám metóda modelovania poskytuje podobne účinné nástroje, ako ponúka analytická metóda pri hľadaní neznámych veličín. Od kozmologických modelov vo fyzike cez evolučné modely v biológii a epidemiologické modely v medicíne až po modely trhu v ekonómii ponúka tak metodická sila jazyka matematickej analýzy mohutný nástroj na porozumenie skutočnosti.

f. ITERATÍVNA GEOMETRIA je pomerne mladá disciplína. Výklad metodologickej sily jej jazyka nechávam ako otvorený problém pre ďalšie skúmania.

g. PREDIKÁTOVÝ POČET prináša *modernú axiomatickú metódu*. Antická logika nepoznala existenčný kvantifikátor v tom zmysle, ako ho chápeme dnes. V klasických axiomatických systémoch, napríklad v Euklidových *Základoch*, sa existencia nepostuluje, ale zabezpečuje sa konštrukčne. Predikátový počet zaviedol existenčný kvantifikátor, čím umožnil existenciu postulovať. Tento aspekt jazyka predikátového počtu umožnil zrod modernej axiomatickej metódy. Od Peanových axióm aritmetiky cez Hilbertove axiómy geometrie a Zermelove axiómy teórie množín po axiomatické zavedenie grúp, modulov a algebier je moderná axiomatická metóda založená na existenčnej kvantifikácii a je tak prejavom metodologickej sily jazyka predikátového počtu.

Je otázne, či je to jediný metodický prínos predikátového počtu, a či je to jeho najvýznamnejší prínos. Keď jazyky s omnoho slabšou logickou a expresívnou silou než akú má predikátový počet vytvorili *analytickú metódu*, *metódu redukcie* či *metódu modelovania*, je pravdepodobné, že skutočná metodická sila predikátového počtu sa ešte neprejavila a tak jeho skutočný metodický prínos sa odhalí až v budúcnosti a bude významnejší než axiomatická metóda, ktorá je dosť obmedzená. Rovnako je však možné, že ostatné vedné disciplíny nedokážu zatiaľ metodický potenciál jazyka predikátového počtu zužitkovať, lebo ich vlastné jazyky sú príliš slabé na to, aby

dokázali plnú metodickú silu tohto jazyka prijať. To by znamenalo, že na metodologickú revolúciu spojenú s aplikáciou predikátového počtu v prírodných vedách iba čakáme. Je možné, že táto revolúcia bude ešte významnejšia, ako bolo Newtonovo zavedenie diferenciálnych rovníc do fyziky.

2.2 Dôsledky zavedenia metodickej sily jazyka

Uvedený náčrt prináša dva problémy. Jedným je nutnosť korigovať výklad síl jazyka uvedený v *Patterns of Change*. V niektorých prípadoch som príklad ilustrujúci v skutočnosti metodickú silu jazyka, keďže som si existenciu tejto sily neuvedomoval, použil ako ilustráciu niektorej inej sily. Napríklad Viètovu *analytickú metódu* som použil ako ilustráciu *integratívnej sily* jazyka algebry. Do istej miery je to oprávnené, lebo táto metóda skutočne *zjednocuje* izolované postupy do jednotnej metódy. Keď však pridáme k silám aj metodickú silu, je vhodnejšie Viètovou analytickou metódou ilustrovať túto novú silu. Integratívnu silu jazyka algebry budeme preto musieť ilustrovať iným príkladom. Ponúka sa tu pojem *algebraickej formy*, ktorú zaviedol Michael Stifel roku 1544 vo svojej knihe *Arithmetica integra*, keď pripustil, aby koeficienty rovnice nadobúdali aj záporné hodnoty, čím zlúčil do jediného polynómu (či polynomickej formy) tri typy kubických rovníc, ktoré Cardano riešil osobitne (pozri [19], s. 165 – 177). Takže ako prvá sa ponúka úloha prejsť ilustrácie potencialít reprezentácií uvedené v *Patterns of Change* a všetky miesta kde je príklad manifestujúci *metodickú silu* jazyka použitý ako ilustrácia niektorej inej potenciality, je nutné opraviť.

Ako druhá sa ponúka úloha porozumieť dynamike nárastu metodickej sily jazyka. Podobne ako pri ostatných potencialitách, aj v prípade metodickej sily jazyka jednotlivé re-rezentácie na seba nadväzujú a ide o to odhaliť, ako určitá metóda tvorí predpoklad nasledujúcej. V prípade logickej sily je úsečka neurčitej dĺžky, ktorá tvorí základ logickej sily syntetickej geometrie, predstupňom pojmu neznámej, tvoriaceho základ logickej sily algebry. Neznáma predstavuje symbolické vyjadrenie úsečky neurčitej dĺžky, pričom táto symbolizácia je základom prevahy logickej sily jazyka algebry nad logickou silou jazyka syntetickej geometrie. Pri opise analytickej metódy som ukázal ako Viète nadväzuje na postupnosť krokov Euklidovej metódy, keď *ekthesis* nahrádza *zetetickým umením* a *kataskeyé exegetickým umením*. Keď porovnáme zodpovedajúce kroky Vièta a Euklida vynikne metodická prevaha jazyka algebry, ktorý umožňuje pracovať s parametrami rovnako všeobecne ako s neznámou, čo geometria nedokáže. Bolo by zaujímavé porovnať aj ostatné ilustrácie metodickej sily jazyka, teda metódu redukcie a metódu modelovania a odhaliť dynamiku rozvoja metodickej sily.

2.3 Náčrt vývinu metaforickej sily jazyka

Ďalšou potencialitou jazyka matematiky je metaforická sila. Na rozdiel od zvyšných piatich potencialít, v tomto prípade výklad nebude tak systematický ako doteraz, lebo (aspoň pri predbežnom pohľade sa zdá, že) metaforická sila niektorých jazykov je omnoho menej významná ako iných. Preto uvediem iba tie prípady, kedy je možné prenikanie metafor z matematiky do ostatných oblastí doložiť. Metaforická sila zakladá otvorenosť jazyka matematiky, jeho schopnosť prenikať do nových oblastí a odhaľovať nečakané súvislosti. Kým metodická sila jazyka dáva istotu a možnosť spoľahnúť sa na výsledky analýz (ak tieto boli získané pri dodržaní pravidiel metódy), metaforická sila nás otvára novému a robí nás schopnými vstúpiť do kontaktu s neznámym.

a. PYTAGOREIZMUS s jeho úspešnou snahou uchopiť pomocou čísel harmóniu v hudbe či astronómii, spolu s menej úspešnou snahou pomocou čísel opísať také vlastnosti ako spravodlivosť či dokonalosť možno vyložiť ako prejav metaforickej sily jazyka aritmetiky

b. ARISTOTELOVSKÝ KOZMOS s hierarchickým usporiadaním prirodzených miest spolu so sférami jednotlivých prvkov od zeme po oheň, okolo ktorých obiehajú sféry planét a rotuje sféra hviezd, možno vyložiť ako prejav metaforickej sily jazyka syntetickej geometrie. Jazyk geometrie tu nevystupuje ako metodický nástroj presného opisu a striktného dokazovania, ako ho používajú Archimedes či Ptolemaios, ale je skôr súborom metafor. Ako píše o Aristotelovej teórii Philip Frank: „*Teória pohybu bola veľmi dôležitá pre život človeka, lebo podporovala jeho vieru v hierarchickú štruktúru spoločnosti. Dodávala odvahu mravnému správaniu ľudských bytostí.*“ ([4], s. 18). Práve metaforický charakter jazyka umožnil aby sa v obraze kozmu zrkadlili sociálne projekcie.

c. KARTEZIÁNSKE UNIVERZUM, alebo aspoň jeho určité črty, možno vyložiť ako prejav metaforickej sily jazyka algebry. Všeobecnosť, ktorú do algebry vnáša pojem neznámej, Descartes používa metaforicky, ako základ ontologického opisu univerza. Univerzum chápe nie ako súbor konkrétnych súcién, ale ako prejav jednej univerzálnej substancie. Abstraktná všeobecnosť, ktorou sa karteziánske univerzum tak zásadne odlišuje od všetkých univerz skonštruovaných pred ním aj po ňom, má skôr povahu metafory než jasne artikulovanej idey (asi preto Descartove *Princípy filozofie* Voltaire označil za román).

d. LAPLACEOV DÉMON ktorý sa objavil v úvode *Filozofickej eseje o pravdepodobnosti*: „*Predstavme si Inteligenciu, ktorá by v určitom okamihu času poznala všetky sily pôsobiace v prírode a polohy všetkých častíc z ktorých pozostáva svet; predpokladajme ďalej, že táto Inteligencia by bola schopná podrobiť všetky tieto dáta matematickej analýze. ... Potom by nič neostalo neurčité pre túto Inteligenciu. Minulosť a budúcnosť by bola odkrytá jej očiam.*“ ([21], s. 4) je vlastne metaforicky vyložená veta o existencii a jednoznačnosti riešení diferenciálnych rovníc. Preto ho možno chápať ako snahu použiť jednoduché mechanické systémy, ako sú matematické kyvadlo či Keplerova úloha o sústave dvoch telies, ako metaforu pre fungovanie vesmíru ako celku. Laplaceov démon je preto prejavom metaforickej sily jazyka diferenciálneho a integrálneho počtu, keď vlastnosti jednoduchých systémov sú použité ako metafora pre vlastnosti celku.

Aj z tohto malého počtu príkladov vidíme, že metafory majú pre rozvoj poznania veľký význam. Ako ukazuje príklad pytagorejskej aritmetiky, metafory môžu otvoriť bránu pre pochopenie novej triedy javov. Použitie číselných pomerov pre opis harmónie sa ukázalo ako šťastná voľba, ktorá viedla ku vzniku teórie hudobnej harmónie. Fyzikálna teória kmitania strún rozpracovaná Eulerom a d'Alembertom v 18. storočí, teda 24 storočí po pytagorejskom objave, do veľkej miery potvrdila pôvodné intuície, na ktorých pytagorejci analógiu aritmetiky a hudby zakladali. Naproti tomu *Laplaceov démon* ilustruje opačný pól, ktorý možno nazvať *uhranutie metaforou*. Určitá metafora môže na dlhý čas zabrániť vidieť fakty, ktoré sa vymykajú obrazu vytvorenému na základe metafory. Predstava, že vesmír je hodinový stroj (ktorého minulosť a budúcnosť možno vypočítať) vznikla na základe niekoľko málo mechanických úloh, kedy sme skutočne schopní správanie systému predvídať s dostatočnou presnosťou. Turbulentný vodný prúd mali vedci pred očami od nepamäti, a na jeho základe mohlo byť zrejme,

že to s predpovedaním budúcnosti vesmíru nie je tak jednoduché. Laplace vedený analógiou s jednoduchými systémami však tento príklad ignoroval a metafora sveta ako hodinového stroja dlho strašila v knihách vedcov a filozofov, až kým ju objav deterministického chaosu na konci 19. storočia navždy nezapudil.⁶

3 Náčrt potencialít objektácií

V prípade objektácií možno rozlíšiť šesť potencialít jazyka, ktoré sú analogické s logickou, expresívnou, explanatorickou, metodickou, metaforickou a integratívnou silou jazyka. Sú to **abstrakčný, diferenciačný, explikačný, klasifikačný, regularizačný a unifikačný** potenciál. Každý z nich budem ilustrovať na niekoľkých príkladoch, ukážem ako sa líši od sily jazyka, ktorá stojí na zodpovedajúcom mieste v zozname potencialít re-prezentácií a načrtnem súvis s príslušným formálnym aspektom jazyka, pomocou ktorého sú objektácie popísané v *Patterns of Change*.

S objektáciami je spojený abstrakčný zdvih, prechod na vyššiu úroveň abstrakcie. Od Aristotela pochádza výklad abstrakcie ako odhliadania od určitých aspektov vecí a podržania či zohľadňovania iba vybratých aspektov. Tak, keď odhliadneme od farby, tvrdosti a materiálu určitého predmetu, napríklad kovovej gule, a podržíme iba jeho tvar, máme pocit, že sme abstrahovaním dospeli od skutočného predmetu ku geometrickému útvaru, v našom prípade ku geometrickej guli. Preto je rozšírený názor, že veda dospieva ku svojim predmetom v procese abstrakcie, keď odhliada od aspektov, ktoré sú z hľadiska jej diskurzu nepodstatné, a podrží iba tie, ktoré chce skúmať.

Uvedený výklad abstrakcie je podľa mňa nesprávny. Problematické je jeho *psychologické zakotvenie*. Ak spolu s Fregem a Husserlom odmietame psychologizmus vo filozofii matematiky, musíme ho odmietnuť aj vo výklade abstrakcie. Mojm cieľom je *psychologický* výklad abstrakcie ako procesu odhliadania od niektorých aspektov skutočnosti, nahradiť *lingvistickým* výkladom abstrakcie ako jazykovej redukcie, t.j. priemetu opisu skutočnosti do určitého jazykového rámca. Domnievam sa, že tento výklad je plauzibilnejší. Dokáže vysvetliť rozdiely v abstrakcii v rôznych disciplínach. Keď kovovú guľu opisuje geometria, klasická mechanika a teória pružnosti, odhliadajú od rôznych aspektov. Od farby odhliadnu všetky tri, ale kým prvé dve disciplíny abstrahujú aj od tvrdosti kovu (a predkladajú guľu ako nedeformovateľný objekt) teória pružnosti vysvetľuje tvrdosť gule pomocou síl. To, ktoré aspekty skutočnosti je potrebné pri abstrahovaní zachovať, je určené jazykovým rámcom danej disciplíny. Preto budem namiesto o procese abstrakcie hovoriť o *abstrakčnom potenciáli jazyka*. Tým psychologický proces odhliadania nahradím lingvistickým parametrom – potencialitou jazyka.

⁶ Zdá sa, že podobne uhrančivou metaforou bola aj metafora jazyka ako obrazu, na ktorej sa zakladá traktátovská teória jazyka a následne aj veľká časť filozofie logického pozitivizmu. Preto porozumieť tomu, aké metafory nadobúdajú nad myslením uhrančivú moc, je zaujímavý problém. Určite by nebolo ťažké zostaviť katalóg „uhrančivých metafor“ v dejinách vedy a pokúsiť sa nájsť ich spoločné črty.

3.1 Abstrakčný potenciál jazyka

*Abstrakčným potenciálom jazyka*⁷ nazývam schopnosť jazyka reprezentovať určitú situáciu ako špeciálny prípad niečoho všeobecného. Abstrakčný potenciál úzko súvisí s objektáciami, a môže slúžiť pri charakterizácii objektácií podobne ako logická sila jazyka slúži pri charakterizácii re-representácií. Ako príklad *nárastu abstrakčného potenciálu* jazyka pri objektácii uvediem Kleinov *Erlangenský program*. Klein predložil klasifikáciu geometrických systémov, keď jednotlivé geometrie vnoril do projektívnej roviny, ktorá slúžila ako neutrálna báza. Každá z geometrií sa stala určitou štruktúrou, ktorú Klein identifikoval s jej grupou transformácií a potom klasifikoval príslušné grupy. Vnorené, neutrálna báza a štruktúra sú aspekty jazyka, pomocou ktorých je kleinovská objektácia opísaná v *Patterns of Change* ([19], s. 133 – 143; resp. [12], s. 127 – 132). Teraz ide o to uvedomiť si, že Kleinom sa geometria dostáva na vyššiu úroveň abstrakcie. Prejavom tejto vyššej úrovne je Kleinova definícia geometrie ako štúdia invariantov grúp transformácií. Geometria je tak špeciálnym prípadom niečoho všeobecnejšieho – štúdia invariantov grúp. Z nášho hľadiska je dôležité, že tento pokrok bol umožnený prechodom k jazyku s vyšším abstrakčným potenciálom. Kleinov prínos nebol umožneným nejakým odhliadaním (nie je jasné, ako možno u geometrie odhliadnuť od jej euklidovskej štruktúry), ale zmenou jazykového rámca.

K analogickému zdvihu došlo aj v diferenciálnej geometrii, kde Riemann zovšeobecnil pojem geometrie keď geometriu definoval ako štruktúru vznikajúcu na diferencovateľnej variete (neutrálnej báze jazyka) pomocou určitej kvadratickej formy (ktorá hrá úlohu analogickú Kleinovým grupám transformácií). Pri tomto abstraktnom pohľade, keď metriku (v rovine) chápeme ako formu

$$ds^2 = g_{11}dx^2 + 2g_{12}dxdy + g_{22}dy^2,$$

vidíme, že v euklidovskej metrike, kde $ds^2 = dx^2 + dy^2$ máme vlastne $g_{11} = g_{22} = 1$ a $g_{12} = 0$. Teda euklidovská metrika je špeciálnym prípadom niečoho všeobecnejšieho: riemannovskej metriky. Tento abstrakčný zdvih možno opísať tak, že „*nemé koeficienty prehovorili*“. Koeficienty 1 a 0 sú „*nemé koeficienty*“ lebo ich prítomnosť si (na rozdiel od koeficientov typu $\frac{1}{2}$ či 7) neuvedomujeme. V dôsledku objektácie tieto koeficienty „*prehovorili*“, čo znamená, že ich hodnota sa zmenila na takú, ktorú vnímame. Keď koeficienty nadobudnú hodnotu inú než 0 alebo 1, uvedomíme si možnosť alternatív tam, kde sme pôvodne žiadne alternatívy nevideli. Otvorenie sa alternatívam, ku ktorému dochádza pri objektácii, vyjadrujem spojením „*nemé koeficienty prehovorili*“.

Niečo podobné nastalo aj v Kleinovom prípade, kedy sme okrem známych geometrií, ako euklidovská či Bolyai-Lobačevského geometria, začali ako plnohodnotné geometrické systémy vnímať mnohé nové geometrie, ako je napríklad afinná geometria. Aj v tomto prípade je slovné spojenie o *prehovorení nemých koeficientov* prilihavé, lebo euklidovská podgrupa je v projektívnej grupe vymedzená tým, že v maticiach, ktoré zadávajú transformácie euklidovskej podgrupy, sú určité

⁷ Presnejšie by bolo hovoriť o abstrakčnom potenciáli formy jazyka, lebo každá forma prináša nárast abstrakčného potenciálu, a teda určitá úroveň tohto potenciálu sa viaže vždy na určitú formu jazyka. Nebudem sa však držať tejto striktnnej terminológie, lebo by to robilo vyjadrovanie zbytočne ťažkopádny.

koeficienty nulové, t.j. majú charakter nemých koeficientov. Nové, Kleinom vyčlenené geometrie sa od euklidovskej líšia práve tým, že na miestach, kde sú v maticiach euklidovských transformácií nuly alebo jednotky, majú tieto geometrie rôzne čísla. Preto možno povedať, že *prehovorenie nemých koeficientov je prejavom nárastu abstrakčného potenciálu jazyka pri objektácii*.

3.1.a Porovnanie abstrakčného potenciálu s logickou silou

Je dôležité poukázať na súvis a na rozdiel medzi nárastom *abstrakčného potenciálu* pri objektácii a nárastom *logickej sily* pri re-rezentácii. Nárast logickej sily jazyka pri re-rezentácii je spojený s radikálnou zmenou syntaxe, keď pribudne nový typ premenných (objektové premenné v algebre, funkcionálne premenné v diferenciálnom a integrálnom počte). Naproti tomu nárast abstrakčného potenciálu jazyka pri objektácii nemení syntax jazyka, nepribúda žiaden nový typ výrazových prostriedkov. Ide iba o to, že určité prvky jazyka, ktoré boli pôvodne považované za konštantné, pevné a nemenné (t.j. boli „nemé“, či „dané“), zrazu „prehovoria“.

3.1.b Abstrakčný potenciál a hľadisko

Je prirodzené očakávať súvis medzi abstrakčným potenciálom jazyka a *hľadiskom*. Hľadisko je v *Patterns of Change* zavedené ako jeden z aspektov formy jazyka. V jazyku algebry bolo pritom hľadisko priamo identifikované s číslom 0, teda ako to, voči čomu ostatné veličiny získavajú svoju hodnotu. Pritom 0 je typická hodnota „nemého parametra“. Zabudovanie hľadiska do jazyka, opísané v *Patterns of Change* ako okamih, kedy hľadisko získava explicitné vyjadrenie v jazyku, zodpovedá „prehovoreniu nemého parametra“ ako je opísané pri výklade abstrakčného potenciálu jazyka.

Náš pohľad na abstrakciu je opačný než pohľad tradičný. Abstrakcia nespočíva v odmyslení si niečoho, ale naopak, v „primyslení si“ niečoho, v rozšírení jazyka o nový prvok, menovite o hľadisko. Zabudovanie hľadiska do jazyka umožní *variovateľnosť* určitého aspektu, ktorý bol pôvodne nemenný. Práve tým, že hľadisko získa explicitné vyjadrenie v jazyku, rodí sa možnosť tento nový jazykový prvok meniť. Teda abstrakcia nemá pôvod vo veciach, nevzniká odmyšľaním si vlastností vecí, ale práve naopak, zdrojom abstrakcie je subjekt. Tým, že sa hľadisko explicitne zabuduje do jazyka, tým, že sa spredmetní, je možné od neho „získať odstup“, a tento novozískaný odstup konštituuje abstrakčný zdvih. Odstup je tak vždy odstupom od seba, alebo presnejšie, odstupom od spredmetnenej časti seba. Až pri spätnom pohľade, keď sa z pozície bohatšieho jazykového rámca pozeráme na pôvodný rámeč, máme pocit, že on vznikol „odmyslením“ si variovateľnej zložky.

Tradičný výklad abstrakcie je nesprávny, lebo prv ako si môžeme variovateľnosť odmyslieť, musela sa „otvoriť“, predtým ako možno určitý parameter „umlčať“ musel napred „prehovoriť“. Preto je z epistemologického hľadiska chybné interpretovať euklidovskú geometriu ako geometriu, v ktorej abstrahujeme od zmiešaných členov metrického tenzora a od rozdielu diagonálnych členov (aj keď z matematického hľadiska je takáto charakterizácia presná). A niečo podobné robí klasická definícia abstrakcie. Robí to však v kontextoch, ktoré sú natoľko zložité, že si absurdnosť takéhoto kroku neuvedomíme. V príklade s kovovou guľou máme pocit, že o čo tu ide je odmyslenie si farby, tvrdosti a materiálu gule; avšak tomuto kroku muselo predchádzať uvedomenie si, že farba, tvrdosť a materiál sú variovateľné parametre,

ktoré možno od objektu „odlúčiť“. Tradičný výklad abstrakcie neuchopuje abstrakciu v smere abstrakčného zdvihu (ako *prechod od jazyka s nižším abstrakčným potenciálom k jazyku s vyšším abstrakčným potenciálom*) ale naopak: jazyk s vyšším abstrakčným potenciálom (v prípade gule to je jazyk s „variovateľnými“ pojmi ako farba, tvrdosť či materiál) považuje za samozrejmosť. V rámci tohto silnejšieho jazyka (s odlučiteľnými atribútmi farby, hmotnosti, etc.) tradičné pojmá abstrakcie chápe potom abstrakciu ako využitie tejto variovateľnosti na odmyslenie si určitých aspektov (teda ako umlčanie určitých parametrov jazyka či pripísanie im nulovej hodnoty), napríklad farebnosti, hmotnosti, či tvrdosti. Variovateľnosť sa od počiatku považuje za niečo samozrejme a abstrakcia potom vyzerá ako určité ochudobnenie. To je však podľa mňa zásadný omyl. **Abstraktné je vždy bohatšie, lebo to konkrétne je jeho špeciálnym prípadom**, ku ktorému vždy možno prejsť špecifickou voľbou určitých parametrov.⁸

3.2 Diferenciačný potenciál jazyka

Ako druhú potencialitu jazyka spojenú s objektáciami by som rád uviedol **diferenciačný potenciál**. Jeho nárast si možno uvedomiť na objave neeuklidovskej geometrie ([19], s. 124 – 133; resp. [12], s. 115 – 123). V euklidovskej geometrii splyva pojem *priamky* (čo je primitívny pojem geometrie, t.j. pojem vystupujúci v axiómoch) a *ekvidištanty* (čo je množina bodov ležiacich v jednej polorovine určenej danou priamkou v rovnakej vzdialenosti od nej). Rozsah týchto pojmov je v euklidovskej geometrii identický, a tak si matematici často neuvedomovali ich rozdiel. Viaceré chybné dôkazy Euklidovho piateho postulátu sa zakladali na tom, že ich autori zostrojili ekvidištantu a pracovali s ňou ako s priamkou. Ale predpokladať, že ekvidištanta je priamka je ekvivalentné s piatym postulátom, a teda príslušný dôkaz predpokladal čo chcel dokázať.

Jazyk, ktorý zaviedli tvorcovia neeuklidovských geometrií (Gauss, Bolyai a Lobačevskij), prináša možnosť jasného rozlíšenia medzi priamkou a ekvidištantou, lebo tieto objekty majú v neeuklidovskej geometrii rôzny tvar. Pritom rozdiely medzi nimi sú tým nápadnejšie, čím väčšia je krivosť priestoru. Iba v euklidovskej rovine, ktorej krivosť je nulová, tieto objekty splyvajú a vzniká dojem, že ide o objekt jediný. Odlíšenie priamky a ekvidištanty je posun typický pre objektácie. Objektáciu môžeme vyložiť ako „*prehovorenie nemého parametra*“ (v tomto prípade krivosti priestoru). V euklidovskej geometrii má krivosť nulovú hodnotu, čo je typická hodnota nemého parametra. Pri objektácii uvedený parameter „prehovorí“, nadobudne nenulovú hodnotu, a v dôsledku toho sa ekvidištanta odlíši od priamky. Pri objektácii teda narastá diferenciačný potenciál jazyka.

3.2.a Porovnanie diferenciačného potenciálu s expresívnou silou

Podobne ako v predošlom prípade, aj teraz je poučné porovnať **diferenciačný potenciál** s príbuznou potencialitou jazyka na úrovni re-prezentácií, ktorou je **expresívna sila**. Nárastom *expresívnej sily* pri re-prezentácii jazyk získa schopnosť generovať termy

⁸ Pomýlenému chápaniu abstraktnosti sa niektorí filozofi (napríklad Hegel) usilovali dodať zdanie hĺbky a tak namiesto vyjasnenia celú situáciu iba viac zamotali. Zmiešali abstraktnosť so schematickosťou.

nového druhu, termy, ktoré boli na predošlých štádiách nepredstaviteľné. Tak algebra priniesla mocniny ľubovoľného stupňa, čo syntetická geometria nedokázala. Nárast expresívnej sily jazyka súvisí s novými syntaktickými pravidlami, ktoré riadia prácu s týmto druhom termov. Naproti tomu pri náraste *diferenciačného potenciálu jazyka* objekty, ku ktorých odlišeniu dôjde, už existovali. Ekvidištanty sú bežné objekty euklidovskej geometrie – sú vytvorené podľa pravidiel jej syntaxe. Na ich získanie nemusíme meniť jazyk. Problém je len, že v euklidovskej geometrii splyývajú s priamkami. Keď vytvoríme množinu bodov ležiacich v konštantnej vzdialenosti od priamky, v euklidovskej rovine vznikne priamka. Nárast *diferenciačného potenciálu jazyka* neprináša nové objekty, prináša však *rozlíšenia tam, kde prv rôzne objekty splyývali*.

3.2.b Diferenciačný potenciál a horizont

Diferenciačný potenciál je prirodzené dať do súvisu s *horizontom*, čo je ďalší aspekt formy jazyka. Horizont totiž (podľa Wittgensteina) predstavuje hranice sveta zobrazeného jazykom. Čím predmety ležia bližšie k horizontu, tým ich vidíme menšie a tým ťažšie ich dokážeme navzájom odlíšiť. Pri zmene horizontu jazyka (ako napríklad pri prechode od euklidovskej geometrie k Bolyai-Lobačevského geometrii) môže dôjsť k odlišeniu prvkov ktoré splyývali (priamka a ekvidištanta). Zmena horizontu prináša tak nárast *diferenciačného potenciálu*.

Podobne keď Desargues zaviedol reprezentáciu reprezentácie, mohol nekonečne vzdialené body jednej roviny premietnuť na obyčajnú priamku druhej roviny. Desargues tak vytvoril jazyk, v rámci ktorého boli nekonečne vzdialené body jednej roviny explicitne vyjadrený na druhej rovine. S touto zmenou je opäť spojený nárast *diferenciačného potenciálu jazyka*, keď nový jazyk dokáže odlíšiť správanie sa paraboly a hyperboly v nekonečne: parabola sa nekonečne vzdialenej priamky iba dotýka, kým hyperbola túto priamku pretne.⁹

3.3 Explikačný potenciál

Explikačným potenciálom rozumiem schopnosť jazyka racionálne vysvetliť určité postupy, ktoré prv figurovali ako triky bez akéhokoľvek vysvetlenia. Explikačný potenciál umožňuje urobiť explicitným niečo, čo predtým nebolo možné vyložiť, niečo, čo bolo iba *ad hoc*. Napríklad keď Cardano vyriešil rovnicu tretieho stupňa, použil pri tom *ad hoc trik*. Vďaka predpokladu o tvare výsledku sa mu podarilo previesť úlohu o riešení rovnice tretieho stupňa na úlohu o riešení pomocnej rovnice, ktorá bola druhého stupňa ([19], s. 167 – 172; [18]; [14] a [15]). Podobne postupoval Ferrari, ktorému sa podarilo previesť úlohu o riešení rovnice štvrtého stupňa na riešenie pomocnej rovnice tretieho stupňa. V oboch prípadoch však bola redukcia

⁹ Pojem horizontu je jedným z ústredných pojmov alternatívnej teórie množín. Petr Vopěnka diskutuje zmeny horizontu, ktoré vykladá ako „posúvanie“ či „približovanie“ horizontu, teda z hľadiska našej klasifikácie ako re-formulácie. Re-formulácia však neprináša zmenu formy jazyka, neprináša preto ani nárast *diferenciačného potenciálu*. Preto na posúvaní horizontu nie je možné založiť výklad žiadneho netriviálneho javu, a už vonkoncom nie pohybu, ako sa o to spomínaný autor opakovane pokúša.

stupňa rovnice dielom *šťastnej náhody*. Zakladala sa na *ad hoc trikoch*, pre ktoré chýbalo konceptuálne porozumenie.

Lagrangeovi sa podarilo vysvetliť, prečo zdanlivo náhodné triky viedli k riešeniu. Ukázal, že to nebola náhoda. Pochopil, že Cardanove a Ferrariho triky majú spoločné jadro, ktoré zachytil pojmom *rezolventa*. Ukázal, že pre každú rovnicu tretieho a štvrtého stupňa existujú výrazy vytvorené ako kombinácie koreňov, ktoré redukujú stupeň rovnice. Výrazy, ktoré sa prv javili ako nahodilé zoskupenie písmen, Lagrange pomenoval, **vyčlenil ich** ako samostatné formálne objekty a rozpoznal ich úlohu v procese riešenia rovníc. Takto prechod k jazyku s vyšším explikačným potenciálom umožnil racionálne vyložiť postupy, ktoré sa na štádiách s nižším explikačným potenciálom javili ako *ad hoc triky*. To ukazuje, že prítomnosť *ad hoc* postupov v určitej teórii je príznakom nedostatočného explikačného potenciálu jazyka, a môže byť predzvesťou ďalšej objektácie. Podobne ako nárast diferenciacného potenciálu umožňuje odlišiť objekty, ktoré prv splývali, nárast explikačného potenciálu umožňuje pochopiť racionálny základ postupov, ktoré sa prv javili ako iracionálne triky.

3.3.a Porovnanie explikačného potenciálu s explanatorickou silou

Je užitočné porovnať **explikačný potenciál** s príbuznou potencialitou re-reprezentácií, ktorou je **explanatorická sila**. Nárast explanatorickej sily sme charakterizovali tým, že umožňuje vysvetliť zlyhania jazyka predošlej reprezentácie. Napríklad jazyk analytickej geometrie umožnil vysvetliť zlyhanie jazyka algebry, ktoré bolo z pohľadu samotnej algebry nepochopiteľné. Keď sa Cardano pokúšal riešiť rovnicu $x^3 = 7x + 6$, ktorá má koreň $x = 3$, a postupoval podľa štandardného postupu riešenia rovníc tohto typu, dospel k výrazu $x = \sqrt{\quad}$, ktorý mal pod znakom druhej odmocniny záporné číslo. To predsa nedáva zmysel! Žiadne číslo umocnené na druhú nemôže byť záporné. Odmocnina zo záporného čísla neexistuje. Postup zlyhal. Cardanovi jazyk v ktorom pracoval neumožňoval pochopiť prečo. Prečo vedú štandardné postupy k absurdným výsledkom?

Newton vďaka prechodu od jazyka algebry k jazyku analytickej geometrie, teda k jazyku s vyššou explanatorickou silou, dokázal túto záhadu vysvetliť. Algebraické rovnice zadávajú v rovine geometrické krivky. Riešiť rovnice znamená určiť priesečníky kriviek (so súradnou osou alebo kriviek navzájom). Keby postupy riešenia rovníc vždy fungovali, znamenalo by to, že sa príslušné krivky musia vždy preŕať (v bode, ktorého súradnice získame pomocou algebraického postupu). Skúsenosť ale ukazuje, že mnohé krivky sa nepretínajú. Preto postupy riešenia algebraických rovníc musia zlyhať. Zlyhanie algebraických postupov, ktoré z pohľadu algebry pôsobí nepochopiteľne, dokáže analytická geometria vysvetliť vďaka nárastu explanatorickej sily jazyka. Algebraické postupy musia zlyhať, aby dali geometrickým objektom dostatočnú slobodu sa vzájomne nepretínať.

Tento príklad dobre ilustruje rozdiel medzi explanatorickou silou a explikačným potenciálom. **Explanatorická sila** umožňuje **vysvetliť zlyhanie jazyka** na predošlom štádiu, teda vysvetliť, prečo určité postupy nemôžu existovať. Naproti tomu **explikačný potenciál** sa týka úspešných postupov, postupov, ktoré viedli k správne riešeniu. Tieto postupy mali iba ten nedostatok, že boli *ad hoc* – nebolo jasné, prečo fungujú. Nárast explikačnej sily jazyka umožňuje racionálne **vyložiť fungovanie ad hoc trikov**.

Podobne ako v prípade objektov „prehovoria nemé parametre“, tu sa explikujú implicitné vzťahy a súvislosti, ktoré umožnia v trikoch nájsť racionálne jadro.

3.3.b Explikačný potenciál a indivíduá

Individuá sú (popri hľadisku a horizonte) ďalším aspektom formy jazyka. Husserlova téza – bez tela niet telesa – ukazuje, že konštitúcia indivíduí jazyka je formálna. Subjekt vyčleňuje vo svojom okolí ako indivíduá tie entity, ktoré rozpoznáva ako sebe podobné. Teleso, t.j. individuum jazyka mechaniky, je niečo, k čomu sa možno telesne vzťahovať. Potom, ako sa v jazyku spredmetní hľadisko, sú postupne identifikované objekty analogické tomuto spredmetnenému hľadisku. Tak ako spredmetnenie hľadiska zakladá abstrakčný potenciál, a ako horizont zakladá diferenciačný potenciál, zavedenie indivíduí zakladá explikačný potenciál jazyka. Lagrange rozpoznal racionálne jadro v Cardanových a Ferrariho trikoch tak, že skúmal veličiny invariantné voči permutáciám koreňov. Podobne, ako je hľadisko v algebre vyčlenené ako neutrálny prvok pre sčítanie, Lagrange vyčlenil rezolventu ako veličinu vykazujúcu invariantnosť voči určitým permutáciám koreňov. Tak ako v mechanike sa teleso vyčleňuje na základe telesnosti, čo je vlastnosť epistemického subjektu mechaniky, Lagrange vyčleňuje indivíduá jazyka algebry pomocou invariantnosti, čo je vlastnosť charakteristická pre epistemický subjekt algebry. Individuá (rezolventy a invarianty), ktoré predošlá forma jazyka nedokázala individualizovať, hrali rozhodujúcu úlohu v Lagrangeovej explikácii Cardanových a Ferrariho trikov. Pre predošlé formy jazyka veličiny ako $x_1x_2 + x_3x_4$ predstavovali náhodné termy bez akejkoľvek individuality.

3.4 Klasifikačný potenciál jazyka

Konštituovanie nových objektov, ktoré je základom nárastu explikačného potenciálu jazyka, prináša nové možnosti *klasifikácie* objektov a vytvorenia prehľadu o prvkoch *univerza teórie*. Ako príklad možno uviesť Kleinovu klasifikáciu geometrií. **Klasifikačným potenciálom** jazyka rozumiem jeho schopnosť vniesť určitý systém do univerza objektov a javov teórie. Klein bol schopný sformulovať otázku, aké geometrie sú možné, t.j. otázku klasifikácie geometrií vďaka tomu, že geometrie stotožnil s podgrupami projektívnej grupy, a potom metódami algebry klasifikoval tieto podgrupy. Klasifikácia často mení poriadok v univerze. Objekty, ktorých význam bol pôvodne určovaný na základe kontextu ich objavu dostávajú nový význam na základe svojho miesta v systéme. Často sa to, čo bolo považované za jedinečné stáva typickým.

Základom klasifikácie je často určitá **identifikácia**, stotožnenie klasifikovaných objektov s určitými prvkami štruktúry. Príkladom je Kleinova *identifikácia* geometrie s grupou transformácií, alebo *identifikácia* logických spojok s booleovskými funkciami, ktorá je základom klasifikácie týchto spojok. Veci ktoré boli prv jedinečné, či špecifické sa identifikujú s určitým prvkom štruktúry, a na základe tejto identifikácie dostávajú svoje miesto v systéme. Pritom je to schopnosť jazyka niečo vyložiť ako niečo iné, čo umožňuje klasifikáciu. Keď povieme, že *logické spojky sú booleovské funkcie*, alebo keď Klein povie, že *absolúta je invariant grupy transformácií*, je to táto jazyková identifikácia, ktorá vytvára predpoklady pre klasifikáciu.

3.4.a Porovnanie klasifikačného potenciálu s metodickou silou

Podobne ako v predošlých prípadoch aj tu je užitočné porovnať *klasifikačný potenciál* s príbuznou potencialitou v rovine re-prezentácií, ktorou je *metodická sila*. Metodická sila jazyka umožňuje vytvoriť štandardné postupy na riešenie širokého spektra úloh. Naproti tomu klasifikačný potenciál nie je zameraný na riešenie úloh, t.j. na získavanie nových poznatkov, ale na utvorenie celkového obrazu o tom, čo už poznáme. Keď „*prehovoria nemé parametre*“, tak okrem už známych objektov sa vynorí rad objektov zodpovedajúcich novým hodnotám parametrov; objektov, ktoré na seba prv nepútať pozornosť. Ďalšie objekty poskytne explanatorická sila a s ňou spojená individualizácia náhodných termov jazyka. Prvky takto obohateného univerza možno zoradiť podľa „*narastania parametra*“, čím sa odhalia vzťahy, ktoré boli prv skryté. Klasifikácia univerza objektov slúži často na stabilizáciu metódy, lebo prvky sa spravidla rozdeľujú podľa toho, ako sa správajú pri použití metódy. Svedčia o tom často už aj samotné názvy tried objektov, ako sú napríklad „*integrovateľné funkcie*“ či „*diferencovateľné funkcie*“.

3.4.b Klasifikačný potenciál a kategórie

Ďalší aspekt, ktorým je v *Patterns of Change* charakterizovaná forma jazyka, sú kategórie, teda fundamentálne pojmy, ktoré v dôsledku ich fundamentálnosti jazyk neumožňuje explicitne vymedziť ([19], s. 205 – 209). Ilustráciou posunu v kategoriálnom systéme geometrie je zmena, ktorú použil Alfred Tarski vo svojej práci ([24]), kedy namiesto troch typov základných objektov (bod, priamka, rovina) zaviedol jediný typ – body. Priamky a roviny definoval pomocou relácie kolíneárnosti a konplanárnosti. Výhodou tohoto prístupu je sprehľadnenie logickej stavby teórie. Vďaka zmene kategoriálneho systému geometrie mohol Tarski geometrické pojmy klasifikovať podľa toho, akú formulu potrebujeme na ich definíciu. Vidíme, že zmena kategoriálneho systému teórie úzko súvisí s klasifikačným potenciálom jej jazyka.

3.5 Regularizačný potenciál

Regularizačným potenciálom jazyka rozumiem schopnosť jazyka pomocou rozšírenia svojho oboru individuí premeniť určité výnimočné (či singulárne) prípady v štandardné (či regulárne). Napríklad v euklidovskej geometrii sa dve priamky v rovine spravidla pretínajú. Výnimkou sú rovnobežky. Preto bol Euklides nútený každý dôkaz vetviť a oddelene uvažovať prípad kedy sa dve priamky preťali a prípad, kedy boli rovnobežné. V projektívnej geometrii sa rovina doplní o nekonečne vzdialené body, a aj rovnobežky sa začnú pretínať (v týchto nekonečne vzdialených bodoch). Preto nie je nutné vetviť dôkazy a rozlišovať rôzne prípady. Tu sa ukazuje dvojitý význam pojmu regularizácia. Na jednej strane sa regularizácia týka objektov, keď sa výnimočné objekty (rovnobežky) pridaním nekonečne vzdialených bodov stanú štandardnými. Súčasne sa regularizácia týka postupov, keď sa alternatívy, ktoré Euklides musel uvažovať oddelene, zahrnú pod jediné pravidlo. Po regularizácii všetky objekty podliehajú pravidlám a pravidlá možno aplikovať bez výnimky. Regularizácia je nástrojom, ktorý matematici často používajú. Napríklad v algebre rozšírenie číselného oboru o komplexné čísla umožňuje na jednej strane odmocniny zo záporných čísel interpretovať ako regulárne výrazy, a na druhej strane postupy riešenia rovníc

získavajú všeobecnú použiteľnosť. Tézou tohto článku je, že táto regularizácia je umožnená jazykom, jeho regularizačným potenciálom.

Vďaka regularizácii nadobúdame schopnosť určité postupy plne *formalizovať*. (Formalizáciu považujem takto za prejav regularizačného potenciálu jazyka). Formalizácia je veľmi dôležitá. Ide vlastne o zavŕšenie prechodu od implicitného k explicitnému. Ústredná idea, pomocou ktorej sú v *Patterns of Change* vyložené objektácie, je, že niečo čo sa iba ukazovalo, a teda patrilo k forme jazyka, sa stáva explicitným. Tento proces sa završuje regularizáciou. Spravidla pri tom dochádza ku spredmetneniu formy jazyka.

3.5.a Porovnanie regularizačného potenciálu s metaforickou silou

Je zaujímavé porovnať *regularizačný potenciál* s analogickou potencialitou v rovine re-prezentácií, ktorou je *metaforická sila*. Metaforická sila jazyka je základom pre prenesenie pojmov určitej matematickej disciplíny do oblasti ležiacej mimo matematiky, čím sa matematika otvára novým typom problémov, situácií, súvislostí a objektov. Naproti tomu regularizačný potenciál neprináša žiaden nový druh problémov, situácií, súvislostí či objektov, ale iba pre to, čo sa už dávno pozná, nachádza súbor jednotných regulárnych pravidiel. Pritom nemožno poprieť, že v rozprávaní o nekonečne vzdialených bodoch v geometrii alebo o komplexných číslach v algebre je určitá miera „metaforického“ rozšírenia pojmu bod respektíve číslo. Preto podobne ako v predošlých prípadoch, aj teraz je súvislosť medzi regularizačným potenciálom a metaforickou silou jazyka zrejmy. Ale na druhej strane je zrejmy aj rozdiel medzi nimi, lebo na prácu s nekonečne vzdialenými bodmi či s komplexnými číslami vieme udat' plne explicitný súbor pravidiel, bez akéhokoľvek metaforického významu.¹⁰

3.5.b Regularizačný potenciál a ideálne prvky

Vzťah regularizačného potenciálu a *ideálnych prvkov*, čo je ďalší aspekt formy jazyka, je zrejmy z uvedených príkladov. V projektívnej geometrii bola regularizácia dosiahnutá pridaním nekonečne vzdialených bodov, v ktorých sa pretínajú rovnobežky; v algebre bola regularizácia dosiahnutá pridaním komplexných čísel, pomocou ktorých možno vyjadriť odmocniny zo záporných čísel. Nekonečne vzdialené body rovnako ako imaginárne čísla sú ideálne prvky príslušného jazyka. Teda aj tu existuje jednoznačný vzťah medzi potencialitou a zodpovedajúcim aspektom formy jazyka.

3.6 Unifikačný potenciál jazyka

Ako poslednú potencialitu jazyka spojenú s objektáciami chcem uviesť *unifikačný potenciál* jazyka. Je to schopnosť jazyka vytvoriť jednotiaci pohľad tam, kde predtým existovali rôzne nesúvisiace či dokonca navzájom si protirečiacie opisy. Ako príklad

¹⁰ Je zaujímavé, že predtým ako sa podarilo nájsť striktné pravidlá pre prácu s nekonečne vzdialenými bodmi či komplexnými číslami, tieto pojmy boli používané viac-menej metaforicky (nekonečne vzdialené body u Kuzánskeho, komplexné čísla u Cardana a Bombelliho). Preto vzťah metaforickej sily a regularizačného potenciálu jazyka je, zdá sa, geneticky ešte užší ako vo zvyšných prípadoch.

možno uviesť Kleinovo zjednotenie geometrií. Pred Kleinom boli euklidovská a neeuklidovská geometrie považované za navzájom sa vylučujúce. Ak je priestor euklidovský, nemôže byť neeuklidovský, a naopak. Klein dokázal jednotlivé geometrie vyložiť z jednotného hľadiska ako geometrie patriace k rôznym podgrupám *projektívnej grupy*. Pojem projektívnej grupy vniesol jednotu do systému geometrií, keď každej geometrii vyčlenil náležité miesto ako podgrupy tejto grupy. Kleinov *Erlangenský program* sme spomínali pri výklade klasifikačného potenciálu. Tam sme však kládli dôraz na jednotlivé podgrupy projektívnej grupy, ktoré zodpovedajú jednotlivým geometriám. Pri výklade unifikáčného potenciálu jazyka chceme upozorniť na ďalší aspekt, ktorý je s týmto novým chápaním geometrie spojený, totiž na samotnú projektívnu grupu a jej schopnosť vyjadriť *vzájomné vzťahy* medzi geometriami a tým ich *začleniť do jednotného systému*. Teraz nejde o to, že grupa transformácií tej-ktorej geometrie vzniká zúžením projektívnej grupy, t.j. konkrétnou voľbou určitých parametrov. Dôraz je teraz na tom, že projektívna grupa umožňuje jednotlivé geometrie zaradiť do jednotného rámca a skúmať ich vzájomné vzťahy.

3.6.a Porovnanie unifikáčného potenciálu s integratívnou silou

Uvedená ilustrácia *unifikáčného potenciálu* pripomína *integratívnu silu jazyka*, čo je potencialita charakterizujúca re-prezentácie. Nárast integratívnej sily možno ilustrovať na zjednotení kriviek antickej geometrie, ktoré v 17. storočí priniesla analytická geometria. Antickí matematici poznali krivky ako *Dioklesova cissoida*, *Nikomedova konchoida* či *Dinostratova kvadratrix*. Tieto krivky boli navrhnuté pri snahe riešiť úlohu trisekcie uhla či duplicity kocky. V antickej matematike však predstavovali izolované, navzájom nesúvisiace objekty. Až keď sa zrodila analytická geometria, bolo možné tieto krivky zaradiť do jednotnej schémy a ukázať, že *cissoida* je kubická krivka, *konchoida* bikvadratická a *kvadratrix* transcendentná [23]. Výnimočné objekty antickej geometrie tým zapadli do jednotného systému. Osud týchto kriviek v rukách Descarta a jeho nasledovníkov by tak mohol pripomínať osud neeuklidovských geometrií v rukách Kleina. V oboch prípadoch boli nesúvisiace krivky (respektíve odporujúce si teórie) zaradené do systému, v rámci ktorého im bolo vymedzené miesto a získali jasne definovaný vzťah k ostatným (*cissoida* k ostatným kubickým krivkám, neeuklidovská geometria k ostatným geometriám). Táto podobnosť je však zavádzajúca.

Prostriedky, pomocou ktorých bola dosiahnutá *integrácie* izolovaných fragmentov teórie kriviek respektíve *unifikácia* rôznych geometrií sú odlišné. Spomínané krivky antickej geometrie vybočovali z Euklidovho kánonu. Pre každú z nich bol navrhnutý osobitný mechanický nástroj, ktorým bolo možné krivku zostrojiť. Popri kružidle a pravítku, čo sú štandardné nástroje euklidovskej geometrie, vznikol rad nových mechanických pomôcok. Keď Descartes zjednotil tieto krivky a pevne ich zabudoval do rámca analytickej geometrie, neurobil to tak, že by vymyslel nejaký univerzálny mechanický nástroj, nejaké krivítko, ktoré by umožnilo nakresliť každú z uvažovaných kriviek. (Takýto postup by zodpovedal unifikácii, ktorá prináša všeobecný pojem, pod ktorý zahrnie zdanlivo nesúvisiace prípady.) Descartes naopak zavrhol mechanický spôsob kreslenia kriviek a začal ich generovať úplne inak. Krivky rozbil na body a tie vynášal pomocou algebraických formúl do súradnicovej siete. *Cissoidu*, *konchoidu* i *kvadratrix* tým úplne pretvoril, rozbil ich na najmenšie časti a tieto nanovo poskladal

pomocou formúl. Mechanické kreslenie nahradil algebraickým generovaním. Jednota bola do univerza kriviek vnesená zvonka, pomocou *algebraického nástroja*. To algebraická formula ukazuje, či je krivka kubická či bikvadratická. Klein na rozdiel od Descarta nemení spôsob generovania objektov. Kým Descartes priniesol do geometrie nový reprezentačný nástroj, ktorým rozšíril jej univerzum o tisícky nových kriviek, Klein žiadne nové objekty nepriniesol. Urobil iba explicitnou štruktúru, ktorá v základoch geometrie už vždy bola.

Integratívna sila, ktorá je spojená s Descartovým dielom, spočíva v zjednotení izolovaných objektov tým, že sa rozšíri univerzum, a na pozadí tohto bohatšieho univerza každý z izolovaných objektov získa svoje pevné miesto. Medzi syntetickou geometriou v rámci ktorej sa zrodila *cissoida*, *konchoida* a *kvadratrix* a analytickou geometriou, ktorá odhalila ich jednotu, leží algebra s pojmom formule. Integrácia uvedených kriviek do jednotného rámca bola umožnená až týmto symbolickým medzistupňom. Klasifikácia kriviek na kubické, bikvadratické, ... transcendentné, je algebraická klasifikácia. Naproti tomu **unifikačný potenciál**, ktorý sa viaže ku Kleinovmu dielu, sa prejaví keď „*prehovoria nemé parametre*“ a jednotlivé izolované objekty sa priradia rôznym hodnotám parametrov. Žiadne nové objekty nevznikajú. Kleinovo vyčlenenie podgrúp projektívnej grupy neprináša do geometrie ani externý aparát analogický Descartovým polynómom. Transformácie odjakživa patrili ku geometrii a to, že uzavretý súbor transformácií bol prvý raz nazvaný grupou práve v algebre je historická kontingencia. Pojem grupy mohol vzniknúť aj v geometrii, ved' v geometrii existuje aspoň toľko grúp ako v algebre. Klein neprináša unifikáciu euklidovskej a neeuklidovskej geometrie prostriedkami nejakého vonkajšieho aparátu ako Descartes, ale tým, že prenikol hlbšie do štruktúry samotnej geometrie. Nástroj, pomocou ktorého geometrie zjednocuje neprichádza zvonka, nie je to kultúrne podmienený artefakt, akým algebraická symbolika bezpochyby je, ale je to štruktúra, ktorá je v geometrii implicitne obsiahnutá od počiatku.

Dalším príkladom nárastu unifikačného potenciálu jazyka je, keď projektívna geometria vyložila elipsu, hyperbolu a parabolu ako rôzne polohy toho istého objektu. Všetko, čo sa zjednocuje, elipsa, parabola aj hyperbola, bolo v jazyku vyjadriteľné. Chýbalo vedomie ich jednoty. Naproti tomu v prípade kriviek *cissoida* či *konchoida* boli mimojazykové v tom zmysle, že v rámci jazyka euklidovskej geometrie boli nedefinovateľné. Nárast *integratívnej sily* tak spája opisy, ktoré prísne vzaté do jazyka nepatrili, boli to opisy vzniklé prekročením hraníc jazyka pomocou nedovolených nástrojov.¹¹ Boli to akési fragmenty, ktoré predznamenávali smer ďalšieho rozvoja. *Unifikačný potenciál* sa naopak týka výrazov, ktoré sú vytvorené podľa pravidiel syntaxe jazyka. V dôsledku zmeny formy jazyka sa tieto fragmenty navzájom prepoja. Identita zjednocovaných prvkov ostáva zachovaná, spôsob ich generovania sa nemení.

¹¹ To je zaujímavé. Nelegitímne postupy tvoriace oblasť zmiešaných disciplín pri ideácii (pozri [3]) sa tak vyskytujú aj pri re-prezentáciach. Na rozdiel od ideácie, kde sú zmiešané disciplíny motorom vedeckej revolúcie a v jej priebehu vytvoria úplne nový typ vednej disciplíny, v rámci ktorej sa legitimizujú, pri re-prezentácii sú nelegitímne postupy napokon eliminované. Mechanické pomôcky Dioklesa či Nikomeda Descartes z matematiky odstránil, keď príslušné krivky skonštruoval metódami analytickej geometrie. Naproti tomu Archimedove či Ptolemaiove postupy, ktoré sa rovnako vymykali ideálu antickej vedy ako postupy Dioklesa či Nikomeda, vo vede pretrvali a pohltila ich experimentálna metóda.

Avšak prepojenia, ktoré boli implicitné, v jazyku nevyjadriteľné a iba sa ukazovali, sa stali explicitne vyjadriteľnými.

3.6.b Unifikačný potenciál a pozadie

Aspekt formy jazyka, ktorý je úzko spojený s unifikačným potenciálom je pozadie. V geometrii je pozadím priestor, ktorý možno chápať ako spredmetnenie aktov jazyka, ako súbor všetkých miest, kam môžeme umiestniť hľadisko. Zaujímavú analýzu pôvodu pojmu priestor podáva Poincaré vo *Vede a Hypotéze*. Odvodzuje ho priamo z transformácií hľadiska (pozri tiež [19], s. 140 – 143). A práve grupa transformácií predstavuje prostriedok, pomocou ktorého Klein uskutočňuje unifikáciu rôznych geometrických systémov. Preto unifikačný potenciál jazyka je konštituovaný pozadím.

3.7 Ilustrácia potencialít objektácií

U čitateľa mohol náš výklad vyvolať pochybnosti, lebo príklady ilustrujúce jednotlivé potenciality jazyka sú značne nesúrodé. Celková stavba teórie určite lepšie vynikne, keď sa podarí všetky potenciality ilustrovať na jedinom príklade. Tak jasnejšie vyniknú súvislosti medzi jednotlivými potencialitami. Na ilustráciu všetkých potencialít možno použiť dôkaz neskonštruovateľnosti pravidelného sedemuholníka pomocou kružidla a pravítka. Podrobnosti dôkazu, na ktoré tu niet miesta, možno nájsť v ([1], s. 117 – 146). Problém neskonštruovateľnosti sedemuholníka je zaujímavý aj sám o sebe, lebo sa okolo neho točí veľká časť Wittgensteinových *Prednášok o základoch matematiky* [2]. Analýza ukáže, že v prirodzenom jazyku, v ktorom tento problém analyzuje Wittgenstein, ho nemožno pochopiť. Dôkaz neskonštruovateľnosti pravidelného sedemuholníka vyžaduje niekoľko objektácií, v dôsledku ktorých vznikne jazyk s dostatočným abstrakčným, diferenciačným, ..., unifikačným potenciálom, aby bolo možné neskonštruovateľnosť dokázať. Keď sa na dôkaz pozrieme z hľadiska potencialít, je možné rozlíšiť hlavné kroky dôkazu.¹²

Prvým krokom je využiť **abstrakčný potenciál jazyka** a pozrieť sa na geometrické konštrukcie z abstraktného hľadiska. Základom dôkazu neskonštruovateľnosti sedemuholníka je abstraktný prístup ku geometrickým konštrukciám, pri ktorom si všimame iba to, aké veličiny sa pri určitej konštrukcii vytvárajú. Nevšimame si jednotlivé útvary, ktoré geometer postupne vytvára, ale iba súradnice bodov a koeficienty v rovniciach priamok a kružníc, ktoré pri konštrukcii používa. Túto úroveň abstrakcie dosiahneme tak, že body 0 a 1 prehlásime za dané (t.j. spredmetníme hľadisko; 0 a 1 sú typické nemé koeficienty) a pokúsime sa charakterizovať, čo sa deje počas jedného kroku geometrickej konštrukcie.

Aby sme charakterizovali jednotlivé kroky geometrickej konštrukcie, potrebujeme siahnúť po **diferenciačnom potenciáli jazyka**, ktorý umožňuje rozlíšiť veličiny podľa stupňa polynómu, ktorého sú koreňom. Tak zistíme, že existujú veličiny, ktoré sú koreňom rovníc druhého stupňa (napríklad $\sqrt{2}$ alebo $\sqrt{7}$), veličiny, ktoré sú koreňom rovníc tretieho stupňa (napríklad $\sqrt[3]{5}$ alebo $\sqrt[3]{2}$), etc. Keď sa z tohto pohľadu pozrieme na jednotlivé kroky euklidovskej konštrukcie, zistíme, že v jednom

¹² Nasledujúci text predpokladá porozumenie technickej stránke daného dôkazu. Ak sa čitateľ nechce zdržiavať technickými detailami, môže celú kapitolu 2.7 vynechať.

kroku môže pribudnúť (ako priesečník kružnice a priamky alebo ako priesečník dvoch kružníc) iba veličina, ktorá je koreňom rovnice nanajvyššieho druhého stupňa, pričom koeficienty tejto rovnice sú veličiny už skonštruované. Takto sa podarilo presne rozlíšiť, aké veličiny vznikajú v jednotlivých krokoch geometrickej konštrukcie.

V treťom kroku dôkazu neskonštruovateľnosti sedemuholníka použijeme **explikačný potenciál jazyka**. Ak chceme ukázať, že sa niečo nedá skonštruovať, musíme charakterizovať súbor všetkých objektov, ktoré možno skonštruovať. Musíme položiť otázku, ako vyzerá súbor skonštruovateľných veličín. Naším cieľom nie je posúdiť jednu konkrétnu konštrukciu (ako problém nastoľuje vo svojich prednáškach Wittgenstein), ale musíme charakterizovať všetky konštruovateľné objekty. Až na tejto úrovni je úloha riešiteľná. Musíme sa teda pokúsiť *explikovať* pojem konštruovateľnosti. Ukazuje sa, že pojem konštruovateľnosti možno explikovať pomocou pojmu „kvadratické rozšírenie poľa“. Presnejšie povedané, skonštruovateľné sú práve tie geometrické útvary, ktorých význačné body majú súradnice, ktoré možno získať konečným počtom kvadratických rozšírení poľa racionálnych čísel.

Po úspešnej explikácii konštruovateľnosti pomocou pojmu poľa môžeme pristúpiť k otázke, ako možno dokázať, že sa sedemuholník nedá skonštruovať. Na to stačí ukázať, že nejaký jeho prvok nepatrí do poľa skonštruovateľných čísel. Tu používame **klasifikačný potenciál jazyka**. Používame ho v triviálnej forme, keď sa súbor všetkých veličín rozpadá na dve triedy, na triedu skonštruovateľných a triedu neskonštruovateľných veličín, pričom ukážeme, že vrcholy sedemuholníka (s výnimkou jedného) patria do tej druhej triedy. Nesmieme sa nechať oklamať triviálnosťou tejto klasifikácie. Aj klasifikácia na dve triedy je klasifikácia. Jadro dôkazu spočíva v prevedení úlohy ukázať, že sa niečo nedá skonštruovať na úlohu, že nejaká veličina nepatrí do určitej triedy. A práve to je úloha klasifikačného potenciálu jazyka, že skutočnosť „rozkrája“ na triedy tak, že na vyriešenie daného problému stačí ukázať, že nejaký prvok niekam nepatrí. Prvé tri kroky dôkazu spočívali vo vytvorení určitej triedy (1. – uvedomenie si, že to bude trieda veličín; 2. – odlíšenie kvadratických veličín od ostatných; 3. – pochopenie, že konštruovateľné veličiny tvoria pole). Vďaka tomu môžeme dôkaz nekonštruovateľnosti previesť na otázku prináležania určitého prvku do takto vytvorenej triedy.

Aby sme túto otázku mohli zodpovedať, potrebujeme využiť **regularizačný potenciál jazyka**, a siahnuť ku komplexným číslam. Vrcholy pravidelného 7-uholníka uchopíme ako korene polynómu $z^7 - 1 = 0$ v komplexnej rovine. Po elementárnych úpravách tohto polynómu, ktoré možno nájsť v uvedenej knihe, možno dospieť k rovnici tretieho stupňa ($u^3 + u^2 - 2u - 1 = 0$), o ktorej musíme ukázať, že žiaden jej koreň nepatrí do poľa skonštruovateľných veličín.

Aby sme to ukázali využijeme v poslednom, šiestom kroku dôkazu **unifikačný potenciál jazyka**. Keď chceme ukázať, že náš polynóm tretieho stupňa nemá koreň v poli skonštruovateľných čísel, použijeme nasledovný fakt: ak je číslo $\alpha = p + q\sqrt{w}$ koreňom príslušného polynómu v poli, ktoré vzniklo pomocou konečného počtu kvadratických rozšírení racionálnych čísel, tak je koreňom tohto polynómu aj číslo $\beta = p - q\sqrt{w}$. Dá sa však ukázať, že pre náš polynóm tretieho stupňa nie je možné, aby mal takéto dva korene. Preto nemôže mať žiaden skonštruovateľný koreň. Použili sme symetriu $\varphi: (a + b\sqrt{w}) \rightarrow (a - b\sqrt{w})$, a problém je v tom, že rozšírenia, ktoré

vytvárame pomocou kružidla a pravítka majú túto symetriu, kým rovnica opisujúca sedemuholník takúto symetriu mať nemôže. Uvedená symetria predpokladá unifikáciu univerza, a preto argument založený na takejto symetrii sa zakladá na integratívnej sile jazyka.

Príklad ukazuje ako potenciality jazyka umožňujú viesť dôkaz. Som presvedčený, že uvedené potenciality reálne existujú a matematika ich využíva rovnako, ako využíva logické princípy. Naša ilustrácia je výnimočná, lebo v nej vystupujú všetky potenciality. To vo všeobecnosti nebýva splnené, rovnako ako určitý dôkaz nemusí využiť všetky axiomy. Verím však, že možnosť predviesť všetky prvky v jednom príklade dodáva teórii potencialít na plauzibilitu. Uvedený príklad možno chápať ako určitú polemiku s Wittgensteinom, lebo ukazuje, že dôkaz neskonštruovateľnosti pravidelného sedemuholníka je možný až po rade objektácií, ktoré vytvoria jazyk s dostatočným abstrakčným, diferenciačným, explikačným, klasifikačným, regularizačným a unifikačným potenciálom. Otázky, ktoré Wittgenstein vo svojich prednáškach kladie v prirodzenom jazyku, sú v rámci prirodzeného jazyka nezrozumiteľné. Wittgenstein pozná technický jazyk, ktorý umožňuje dané otázky zodpovedať, avšak pri svojich úvahách sa tvári, ako keby tieto otázky mali zmysel aj v prirodzenom jazyku. Prirodzený jazyk nedokáže odlíšiť skonštruovateľné útvary od neskonštruovateľných a preto otázka či je sedemuholník skonštruovateľný nedáva v prirodzenom jazyku zmysel. Každý, kto si myslí, že zmysel dáva, nevedomky podsúva pod pojem konštrukcie technický jazyk geometrie.

3.8 Zhrnutie

Predošlé analýzy umožňujú vytvoriť si predstavu o *aspektach formy jazyka*, ktoré konštitujú jednotlivé *potenciality* objektácií. Prvým je hľadisko, ktorého spredmetnenie zakladá nárast **abstrakčného potenciálu jazyka**. Na to nadväzuje zpohybnenie (či uvoľnenie) tohto spredmetneného hľadiska, ktoré prináša spredmetnenie horizontu, čo zakladá nárast **diferenciačného potenciálu jazyka**. Tretím je identifikácia referentov určitých termov jazyka ako indivíduí, čo je základom nárastu **explikačného potenciálu jazyka**. Na ňu nadväzuje rozdelenie takto obohateného oboru indivíduí do základných tried, ktoré zakladá nárast **klasifikačného potenciálu jazyka**. Piatym je rozšírenie univerza o ideálne prvky, ktoré zakladá nárast **regulačného potenciálu jazyka**. Na to nadväzuje aktualizácia aktov subjektu v podobe pozadia, ktoré je základom nárastu **unifikačného potenciálu jazyka**.

Pôvodne boli tieto aspekty – hľadisko, horizont, indivíduá, kategórie, ideálne prvky a pozadie – prvkami *formy jazyka*. Boli preto implicitné, ich nositeľom bol epistemický subjekt. Tým, že sa tieto prvky spredmetnia v jazyku, jazyk sa obohatí o určité formálne výrazy (formálne jednak v tom zmysle, že pochádzajú z formy jazyka, a okrem toho aj v tom zmysle, že nerefekujú k ničomu v univerze jazyka), čo umožní **nárast zodpovedajúcich potencialít**. Tým, že sa jazyk stane nositeľom uvedených formálnych prvkov, oslobodí od tejto úlohy epistemický subjekt, ktorý môže začať budovať novú formu jazyka. Z pohľadu tejto novej formy sú aspekty predošlej formy, keďže sú spredmetnené v jazyku, „obyčajnými“ prvkami jazyka, a preto nová forma ich môže plne využívať. To má za následok, že jednotlivé potenciality (abstrakčný, diferenciačný, ... a unifikačný) zviazané s novou formou budú vyššie ako v prípade predošlej formy. Takže aj v prípade objektácií môžeme, rovnako

ako v prípade re-prezentácií, hovoriť o *raste* a nie len o striedaní (viac-menej rovnocenných) foriém.

Ďalšou zaujímavou vlastnosťou teórie re-prezentácií je, že pre každý z nástrojov symbolickej či ikonickej reprezentácie možno určiť jeho logické a expresívne medze. V prípade objektácií už nemožno hovoriť o medzi, teda o objektívnej hranici, ktorú by bolo možné prostriedkami neskoršieho jazyka presne vytýčiť. Ale aj v prípade objektácií sa možno stretnúť s javom pripomínajúcim medze. Pri spätnom pohľade na vývin viacerých disciplín často cítime, že v určitých situáciách prílišná konkrétnosť jazyka bráni vyriešiť určitý problém či porozumieť určitej súvislosti. Na zachytenie takejto situácie navrhujem zaviesť termín **abstrakčná prekážka**. Abstrakčná prekážka sa v histórii prejavuje aj v neochote prijať určitý objav alebo poznatok, pretože takéto prijatie by vyžadovalo prejsť k jazyku s vyšším abstrakčným potenciálom. Abstrakčná prekážka sa pomerne často prejavuje vo vyučovaní matematiky. Učitelia majú sklon žiakom predkladať učivo, ktorého stupeň abstraktnosti je vyšší než je abstrakčný potenciál vlastného jazyka žiakov. Potom sú učitelia konfrontovaní s abstrakčnou prekážkou v podobe neschopnosti žiakov prijať predkladaný poznatok, ktorý je inak celkom jednoduchý. Pre učiteľa je záhadou, čo žiakom bráni poznatok prijať. Ako typický príklad možno uviesť pojem grupy v algebre. Grupa je definovaná pomocou štyroch pomerne jednoduchých vlastností a ako taká by mala byť ľahko prijateľná. Ale nie je.¹³

Podobne bude asi možné zaviesť **diferenciačnú prekážku**, ktorá sa od expresívnych medzí jazyka bude líšiť opäť v tom, že nepôjde o objektívnu nemožnosť určitého rozlíšenia, ale len o to, že jazyk neumožňuje príslušné rozlíšenie prirodzene fixovať. Viaceré chybné dôkazy Euklidovho piateho postulátu sa zakladali na mylnej zámene ekvidistanty za priamku. Táto chyba súvisí s tým, že v euklidovskej geometrii sú zhodou okolností ekvidistanty priamkami a preto nie je možné rozdiel medzi nimi jednoducho fixovať. Ale táto prekážka nemá charakter objektívnych medzí. Kolegovia spomenutých matematikov boli v predložených dôkazoch schopní nájsť chybu, t.j. uvedomiť si, že medzi ekvidistantou a priamkou je predsa len rozdiel. Pre štúdium diferenciačných prekážok spojených s určitým jazykom možno skúmať typické chyby v riešení problémov či v dôkazoch viet.

4 Náčrt potencialít re-formulácií

Zdá sa, že aj na úrovni re-formulácií, kde jazyk matematiky ponúka celý rad variantov určitého tvrdenia, definície či dôkazu, tieto varianty nie sú rovnocenné. Aj pri re-formulácii, t.j. prechode od jednej formulácie k inej (ktorá s ňou nie je synonymná) môžeme v niektorých prípadoch hovoriť o objektívnom rozdieli určitej potenciality. Povaha tohto rozdielu ale nemá charakter *nárastu sily*, ako pri re-prezentáciách, kde

¹³ Tento fenomén je v didaktike matematiky známy pod názvom epistemická prekážka. Mojim cieľom nie je pre známu vec navrhovať nový názov. Ide mi o to tento fenomén zabudovať do predkladanej teórie. Ak je toto zabudovanie správne, znamená to, že epistemické prekážky súvisia s formou jazyka a ich prekonanie sa deje prostredníctvom objektácií, teda prostredníctvom zabudovania formy jazyka (hľadiska, horizontu, ... a pozadia) do jazyka. Predkladaná teória tak poskytuje teoretický rámec pre diagnostiku a prekonávanie týchto prekážok.

nový jazyk umožňuje vyjadriť (definovať, dokázať) niečo zásadne nového, čo bolo prv nevyjadriteľné. Podobne povaha tohto nárastu nemá charakter *zvýšenia potenciálu*, ako pri objektáciách, kde nový jazyk prináša objektívne vykázateľné zvýšenie abstraktnosti či jednoty. Nová kvalita, ktorú prináša so sebou re-formulácia, má skôr povahu určitej *cnosti*.¹⁴ Termínom *cnosť* chcem naznačiť, že uvedená potencialita sa neviaže na jazyk ako celok, ale jej nositeľom je hovorca jazyka. Okrem toho jednotlivé potenciality re-formulácií nie sú navzájom prepojené, ako tomu bolo pri objektáciách či reprezentáciách. Jednotlivé cnosti sú izolované, nesúvisiace kvality.

4.1 Všeobecnosť

Asi najrozšírenejšou aktivitou, tvoriacou náplň prevažnej časti práce profesionálneho matematika je *zovšeobecňovanie* známych výsledkov. Jeho cieľom je nájsť k danej vete alternatívnu formuláciu, ktorá by zahŕňala širšie spektrum prípadov a túto dokázať. Často je zovšeobecnenie daného tvrdenia spojené s konceptuálnou zmenou – s abstrakčným zdvihom. Vtedy je nárast všeobecnosti tvrdenia dôsledkom vyššieho abstrakčného a unifikačného potenciálu nového jazyka. Niekedy však pri zovšeobecnení nedôjde ku zmene konceptuálneho rámca – pôvodné tvrdenie aj jeho zovšeobecnenie sú sformulované v tom istom rámci. Napriek tomu, že matematik zotrúva v tom istom pojmovom rámci, môže sa mu podariť zovšeobecníť dané tvrdenie. Vtedy máme do činenia s „čistým“ zovšeobecnením.¹⁵ Príklady takéhoto zovšeobecnenia sú opísané v knihe *Proofs and Refutations* [20]. Všeobecnejšie výsledky sa v matematickej komunite cenia aj vtedy, keď zovšeobecnenie nie je významné z hľadiska aplikácií.

4.2 Presnosť

Matematika je tradične charakterizovaná ako presná disciplína a presnosť je asi najcharakteristickejšou vlastnosťou jej tvrdení. Preto zvyšovanie presnosti rôznych

¹⁴ Cnosť chápem v duchu MacIntyreovej koncepcie ako viazanú na určitú spoločenskú prax (tu je ňou matematika), v rámci ktorej sa dosahujú určité, tejto praxi interné dobré (matematické poznatky). Vychádzam z MacIntyreovej práce *Zráta cnosti* (MacIntyre 1981; pozri tiež Kuna 2009). Ak je toto použitie MacIntyreovej koncepcie legitímne, naznačuje, že MacIntyreova teória opisuje etické hodnoty toho najmenšieho rádu. Ak odlišenie úrovne re-reprezentácií, objektácií a re-formulácií je zmysluplné aj v etickom diskurze, tak sa zdá, že MacIntyreova teória uchopuje z týchto troch rovín iba rovinu re-formulácií. Len za predpokladu stabilného konceptuálneho rámca, v ktorom opisujeme prax, je možné dohodnúť sa o dobrách, na ktoré sú viazané cnosti. Akonáhle v diskurze o hodnotách dôjde k objektácii či k re-reprezentácii, systém cností prestáva byť relevantný.

¹⁵ Táto úvaha platí aj pre ostatné cnosti. V ďalšom texte budem o cnosti hovoriť iba vtedy, keď obe formulácie, pre ktoré danú cnosť vymedzujeme, sú v tom istom pojmovom rámci. Poradie jednotlivých cností nie je náhodné, ale sleduje poradie potencialít objektácií. Určitá potencialita objektácií vytvára priestor pre uplatňovanie zodpovedajúcej cnosti. Abstrakčný potenciál vytvára priestor v rámci ktorého je možné usilovať o všeobecnosť podobne, ako diferenciačný potenciál vytvára priestor v rámci ktorého možno usilovať o presnosť, klasifikačný potenciál vytvára priestor pre sledovanie prirodzenosti, metodický potenciál vytvára priestor pre jednoznačnosť, regularizačný potenciál vytvára priestor pre jednoduchosť, a unifikačný potenciál vytvára priestor pre univerzálnosť.

tvrdení či výpočtov, aj keď sa na prvý pohľad môže zdať ako samoúčelné, patrí medzi štandardné matematické aktivity. **Presnosť** je tak druhou re-formulačnou cnosťou. Čím presnejšie sa podarí sformulovať určitú definíciu, tvrdenie či argument, tým viac sa blížime k ideálu, ktorý matematika sleduje. Ako ilustráciu zvyšovania presnosti možno uviesť určovanie hodnoty čísla π , ktoré je z praktického hľadiska úplne zbytočné.

4.3 Prirodzenosť

Prvý dôkaz určitého tvrdenia či prvé odvodenie určitej formule je často komplikované, neprehľadné a získané pomocou umelého triku. Z vecného hľadiska je to jedno: keď máme dôkaz, vieme, že tvrdenie platí a môžeme ísť ďalej. Ale predsa sa v matematickej komunite cení, keď sa nájde alternatívny dôkaz, ktorý ten istý fakt dokazuje prirodzenejším spôsobom. Prirodzenosť je iba dodatočná kvalita, základnou kvalitou, na ktorej stojí celá matematika, je pravdivosť.¹⁶ Preto proti neprirodzeným dôkazom nemožno zásadne namietat'. Ale každý matematik cíti, že neprirodzený dôkaz nie je ten pravý. Preto **prirodzenosť** je ďalšou re-formulačnou cnosťou.

4.4 Jednoznačnosť

Mnohé matematické pojmy a tvrdenia majú pôvod v skúsenosti získanej prácou s matematickými objektmi. Pokiaľ skúsenosť, na báze ktorej vznikajú, nie je dostatočne bohatá a tvorcovia nového pojmu či tvrdenia neuvážili dostatočný počet prípadov, môže sa stať, že ich definície či tvrdenia budú nejednoznačné.¹⁷ Pri svojej formulácii mali na mysli štandardný prípad, či intendovanú interpretáciu. Až dodatočne sa začnú vynárať „neštandardné prípady“ či „neintendované modely“, pri ktorých pôvodné formulácie nedávajú zmysel. Preto sú matematici nútení meniť svoje formulácie, aby dosiahli **jednoznačnosť** aj v situáciách, ktoré neboli pôvodne zamýšľané. Ako ilustrácie tohto procesu možno uviesť spresňovanie definície pojmu mnohostena, opísané vo vyššie uvedenej Lakatosovej knihe.

¹⁶ Pravdivosť nie je cnosť, teda vlastnosť, ktorá odlišuje „dobrú“ matematiku od tej „menej dobrej“. Pravdivosť je základný predpoklad matematiky ako takej. Preto pravdivosť nie je potencialitou ani v rovine re-formulácií, ani v rovine objektácií, ani v rovine re-prezentácií. Pravdivosť nie je vlastnosť, ktorá by odlišovala dve formulácie, dva konceptuálne rámce, či dve reprezentácie – jednu ako menej pravdivú a druhú ako viac pravdivú. Pravda je konštitutívny základ celej matematiky, všetkých jej re-prezentácií, objektácií a re-formulácií.

¹⁷ V knihe Representation and Productive Ambiguity in Mathematics and the Sciences Emily Grosholz opisuje nejednoznačnosť ako vlastnosť vedeckého jazyka, ktorá umožňuje objav nových poznatkov. Hovorí o produktívnej nejednoznačnosti. Na príkladoch ukazuje ako nejednoznačnosť reprezentačných nástrojov prispela k významným objavom vo vede. Preto by sa mohlo zdať užitočné zahrnúť nejednoznačnosť medzi potenciality. Podľa mňa je ale vhodnejšie príklady, ktoré Grosholz vykladá ako ilustrácie nejednoznačnosti jazyka, vyložiť inak. Grosholz považuje jazyk za jednotný celok a analýzou príkladov dospela k názoru, že tento celok je nejednoznačný. Podľa mňa je jednoduchšie v jazyku odlišiť jednotlivé reprezentačné nástroje, a namiesto nejednoznačnosti jazyka hovoriť o nejednoznačnosti prekladu medzi týmito nástrojmi. V príkladoch ktoré Grosholz čerpá z matematiky možno nejednoznačnosť takto interpretovať. To samozrejme neznižuje hodnotu jej analýz. Ukazuje to však, že tieto analýzy možno dať do súladu s teóriou potencialít jazyka.

4.5 Jednoduchosť

Niektoré dôkazy používajú aparát, ktorý je omnoho náročnejší než jazyk v ktorom je sformulované dokazované tvrdenie. Jedným z príkladov je dôkaz *veľkej Fermatovej vety*. Kým znenie tejto vety je prístupné stredoškólakovi, jej dôkaz vyžaduje zvládnutie náročných partií matematiky, ktoré nie sú v dosahu bežného matematika. Takže aj keď je tvrdenie dokázané, jeho dôkaz je väčšine matematikov neprístupný. Preto sa cení, keď sa pre určité tvrdenie podarí nájsť jednoduchší dôkaz. Napríklad zákon rozdelenia prvočísel bol dokázaný roku 1896 Hadamardom a de la Vallée-Poussinom. Použili pri tom teóriu funkcií komplexnej premennej, čo je náročnejší aparát než ten, v ktorom je sformulovaná samotná veta. Dôkaz bez použitia funkcií komplexnej premennej podali roku 1948 Selberg a Erdős. Matematici uprednostňujú jednoduchšie dôkazy, preto *jednoduchosť* je ďalšia re-formulačná cnosť.

4.6 Univerzálnosť

Pri riešení určitého problému sa spravidla nájde viacero postupov vedúcich k výsledku. Často je určitý postup aplikovateľný iba na daný problém, je mu tak povediac „šitý na mieru“. Niekedy sa však stáva, že danou metódou možno vyriešiť aj iné problémy, ktoré síce už vyriešené sú, ale nová metóda vyzdvihuje analógie medzi problémami. Dobrou ilustráciou je lineárna algebra, kde sústavy lineárnych rovníc možno riešiť rôznymi spôsobmi. Väčšina učebníc však uprednostňuje Gaussovu eliminačnú metódu, lebo ju možno široko aplikovať. Aj keď je s ostatnými metódami ekvivalentná, ponúka univerzálny prístup k rôznym úlohám. Použitie univerzálnych postupov robí text zrozumiteľnejším a ľahšie prijateľným. Preto *univerzálnosť* je re-formulačnou cnosťou.

Pod'akovanie: Chcel by som poďakovať Tomášovi Čanovi a Róbertovi Macovi za cenné pripomienky k rukopisu state. Príspevok je súčasťou grantového projektu VEGA 1/0453/09 *Vedecká racionalita, jej historické predpoklady a filozofické medze*.

Literatúra

- [1] Courant, R. and Robbins, H.: *What is mathematics?* Oxford University Press, New York 1978.
- [2] Diamond, C.: *Wittgenstein's lectures on the foundations of mathematics Cambridge 1939*. Cornell University Press, New York.
- [3] Euclid: *The Thirteen Books of the Elements*. Preklad Thomas Heath, Dover, New York 1956.
- [4] Frank, Ph.: *Philosophy of science*. Prentice-Hall, Englewood Cliffs, N. J.
- [5] Friedman, M.: Kant's theory of geometry. *The Philosophical Review* **94**, s. 456-506.
- [6] Grosholz, E.: *Representation and Productive Ambiguity in Mathematics and in the Sciences*. Oxford University Press, Oxford.

-
- [7] Hintikka, J.: Kant and the Tradition of Analysis. In: P. Weingartner, Ed., *Deskription, Analytizität, und Existenz, Salzburg*, Pustet, S. 254-272.
- [8] Klein, J.: *Greek Mathematical Thought and the Origin of Algebra*. MIT Press 1968.
- [9] Kolman, V.: *Filosofie čísla. Základy logiky a aritmetiky v zrcadle analytické filosofie*. Filosofía, Praha.
- [10] Kuna, M.: MacIntyreova politická filozofia. *Filozofia* **2009**
- [11] Kvasz, L.: History of Geometry and the Development of the Form of its Language. *Synthese*, s. 141-186.
- [12] Kvasz, L.: *Gramatika zmeny*. Chronos, Bratislava.
- [13] Kvasz, L.: Changes of Language in the Development of Mathematics. *Philosophia Mathematica* **8**, s. 47-83.
- [14] Kvasz, L.: Epistemologické aspekty dejín klasickej algebry. *Filozofia* 2000/10, s. 788-808.
- [15] Kvasz, L.: Epistemologické aspekty dejín modernej algebry. *Filozofia* 2001/5, s. 309-331.
- [16] Kvasz, L.: Niekoľko poznámok ku vzťahu prírodných a spoločenských vied. *Organon F*, s. 157-172.
- [17] Kvasz, L.: Hintika a Friedman o Kantovej filozofii geometrie. In: P. Sousedík (ed.): *Jazyk – logika – veda*. Filosofía, Praha, s. 233 – 251.
- [18] Kvasz, L.: History of Algebra and the Development of the Form of its Language. *Philosophia Mathematica* Vol. 14, s. 287-317.
- [19] Kvasz, L.: *Patterns of Change, Linguistic Innovations in the Development of Classical Mathematics*. Birkhäuser Verlag, Basel.
- [20] Lakatos, I.: *Proofs and Refutations*. Cambridge University Press, Cambridge.
- [21] Laplace, P. S.: *A Philosophical Essay on Probabilities*. John Wiley and Sons, New York 1902.
- [22] MacIntyre, A.: *Ztráta cnosti. K morální krizi současnosti*. Oikoymenh, Praha 2004.
- [23] Savelov, A. A.: *Ploskije krivyje, sistematika, svojstva, primenenija*. GIFML, Moskva.
- [24] Tarski, A.: *A Decision Method for Elementary Algebra and Geometry*. University of California Pres, Berkeley 1951.
- [25] Viète, F.: *Introduction to the Analytical Art*. In: Klein 1934, s. 313-353.

Riešenie problémov s ohraničeniami

Marián MACH¹

Abstrakt. Ohraničenia sú všadeprítomné a nie je možné sa im vyhnúť, neustále je potrebné zohľadňovať rôzne obmedzujúce požiadavky. Na druhej strane reprezentácia problémov pomocou explicitného popisu ohraničení je veľmi prirodzenou a názornou. Predkladaný text sa sústreďuje na úlohy pracujúce s konečnými doménami, ktoré majú kombinatorickú povahu. Je ich možné riešiť pomocou pomerne značného počtu prístupov, pochádzajúcich z rôznych oblastí. Pozornosť je sústredená na metódy splňania ohraničení, evolučné algoritmy a metódy, umožňujúce využiť špecifické znalosti o riešených úlohách.

1 Úvod

Či sa jedná o plánovací problém, rozvrhovanie, pridelovanie zdrojov, návrhové úlohy alebo iné problémy, neustále je potrebné zohľadňovať rôzne požiadavky ako napr. dostupnosť zdrojov, časové závislosti, charakteristiky použitých elementov, ap. Ak si uvedomíme koľko úloh technickej praxe možno formulovať ako úlohy s ohraničeniami, tak je zřejmý dôvod, prečo metódy riešenia úloh s ohraničeniami majú takú dôležitosť.

Problémy s ohraničeniami je vo všeobecnosti možné rozdeliť do dvoch základných tried – úlohy pracujúce s konečnými alebo nekonečnými doménami. Predkladaný text sa sústreďuje na prvú skupinu. Problémy z tejto skupiny majú kombinatorickú povahu, čo je zdrojom pomerne značných požiadaviek kladených na metódy používané pre riešenie týchto problémov – značná časť týchto problémov totiž patrí do kategórie NP-úplných problémov.

Úlohy s ohraničeniami je možné riešiť pomocou pomerne značného počtu prístupov pochádzajúcich z rôznych oblastí, väčšinou z oblasti operačného výskumu a umelej inteligencie. Predkladaný text sa zameriava na tri triedy prístupov – deterministické algoritmické riešenie, stochastické riešenie a riešenie založené na doménových znalostiach, pričom v rámci každého z nich venuje pozornosť iba jednému typu metód – konzistenčným algoritmom, evolučným algoritmom a znalostným modelom. Tieto prístupy prezentujú pomerne rozmanitú vzorku metód, pretože zahŕňajú:

¹ KKUI, FEI, Technická univerzita, Letná 9, 042 00 Košice, E-mail: Marian.Mach@tuke.sk

- ako metódy garantujúce nájdenie riešenia (ak toto existuje), tak aj metódy, ktoré toto negarantujú
- ako metódy pracujúce iba s jedným riešením, tak aj metódy s paralelným spôsobom práce
- ako metódy špecializované na riešenie problémov s ohraničeniami, tak aj metódy použiteľné aj pre riešenie iných typov úloh
- ako metódy abstrahujúce od špecifických detailov konkrétnych úloh, tak aj metódy stavajúce práve na týchto detailoch.

2 Problémy s ohraničeniami

Ako ilustračného reprezentanta kombinatorického problému s ohraničeniami nad konečnými doménami použijeme známy hlavolam Sudoku, ktorého jedna inštancia je zobrazená na Obrázku 1. Hlavolam pozostáva z mriežky vytvorenej deviatimi riadkami, deviatimi stĺpcami a deviatimi 3x3 štvorcami. Spolu obsahuje 81 pozícií (pozícia 11 reprezentuje ľavú hornú a 99 zase pravú dolnú pozíciu – prvá cifra značí riadok a druhá zase stĺpec), pričom niektoré pozície sú obsadené zatiaľ čo iné sú voľné. Cieľom je na každú voľnú pozíciu priradiť jednu číslicu z $\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ tak, aby v žiadnom riadku, stĺpci ani štvorci neboli dve rovnaké číslice.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | 8 | 2 | | 5 | | 6 | |
| | 5 | 3 | 9 | | | | 7 | |
| | | | | | | | 1 | |
| | | | 3 | | | 1 | | |
| 7 | | | | | 8 | 9 | 5 | |
| 9 | | | | | | | | 8 |
| | 4 | | | 2 | | | 9 | 5 |
| | | 6 | | | | | | 7 |
| | | | | 5 | | 4 | | |

Obrázok 1. Ilustračný problém s ohraničeniami nad konečnými doménami.

Z formálneho hľadiska je problém s ohraničeniami vo všeobecnosti definovaný pomocou troch množín:

- množina premenných $V = \{V_1, V_2, \dots, V_n\}$
- množina domén $D = \{D_1, D_2, \dots, D_n\}$
- množina ohraničení $C = \{C_1, \dots, C_n, C_{1,2}, \dots, C_{n-1,n}, \dots, C_{1,2,\dots,n}\}$.

Premenné reprezentujú tie udalosti, stavy atď., hodnoty ktorých sú relevantné pre riešenie konkrétneho problému. V príklade na Obrázku 1 je najprirodzenejšou reprezentáciou taká, keď každá pozícia mriežky je reprezentovaná samostatnou premennou. Výsledkom bude súbor 81 premenných, kde V_{11} reprezentuje pozíciu 11 a V_{99} zase pozíciu 99.

Každá doména $D_i \in D$ zodpovedá premennej $V_i \in V$ a vlastne definuje možné hodnoty, ktoré premenná V_i môže nadobúdať. V našom prípade sú použité konečné domény enumeračného typu. Všetky domény sú rovnaké – množina deviatich číslic 1 až 9.

Nad premennými z množiny V je definovaná množina ohraničení. Indexy jednotlivých ohraničení označujú árnosť týchto ohraničení a premenné, nad ktorými tieto ohraničenia sú definované. Tak napr. ohraničenie $C_{i,j}$ je binárnym ohraničením nad premennými V_i a V_j .

Definované ohraničenia stanovujú prípustnosť hodnôt premenných v rámci nejakého kontextu. Týmto kontextom sú hodnoty iných premenných (počet týchto premenných závisí od árnosti ohraničení). Napríklad, ak nejaké ohraničenie je definované nad podmnožinou premenných, potom toto ohraničenie pre každú premennú z tejto podmnožiny stanovuje prípustnosť priradenia hodnoty z jej domény v kontexte tvorenom hodnotami ostatných premenných danej podmnožiny. Jedinou výnimkou sú unárne ohraničenia, pri ktorých nie je potrebné brať do úvahy hodnoty iných premenných.

Pre skupinu n premenných je teda možné definovať unárne, binárne, ... až n -árne ohraničenia. Celkový počet ohraničení, ktoré je možné definovať pre n premenných je

$$\sum_{i=1}^n \binom{n}{i} \quad (1)$$

pričom i označuje árnosť ohraničení. Je teda zrejmé, že je možné definovať až n unárnych ohraničení a jedno n -árne ohraničenie. Maximálny počet binárnych ohraničení je daný vzťahom $(n^2 - n)/2$. Okrem n -árneho ohraničenia všetky ostatné sú definované iba nad nejakou podmnožinou premenných (premenne z tejto podmnožiny sa označujú ako relevantné voči príslušnému ohraničeniu). V našom prípade teda je možných až 81 unárnych ohraničení, 3240 binárnych, 85320 ternárnych ohraničení, atď.

Všetky ohraničenia sú definované nad podmnožinami premenných originálnej množiny a obmedzujú kombinácie hodnôt, ktoré premenné daných podmnožín môžu nadobudnúť. Extrémnymi prípadmi sú ohraničenia, ktoré nepripúšťajú žiadne alebo naopak pripúšťajú všetky možné kombinácie hodnôt premenných, nad ktorými sú tieto ohraničenia definované. V prvom prípade nie je možné nájsť také riešenie, pre ktoré by takéto extrémne ohraničenie bolo splnené. V druhom prípade je také ohraničenie v množine ohraničení zaradené iba formálne a možno ho z tejto množiny vypustiť.

Pre reprezentáciu nejakého ohraničenia je možné použiť vymenovanie jednotlivých kombinácií hodnôt (či už povolených alebo zakázaných) alebo ohraničenie možno zadať nejakým predpisom, ktorý pre každú kombináciu hodnôt premenných umožňuje rozhodnúť, či daná kombinácia porušuje dané ohraničenie alebo

nie. To, ktorý tvar je využitý, závisí od konkrétnej úlohy a spôsobu práce s ohraničeniami. Náš demonštračný príklad priamo zo svojej definície ponúka 27 rôznych 9-árnych ohraničení (jedno pre každý z deviatich riadkov, stĺpcov a štvorcov). Aj keď je možné takéto ohraničenie reprezentovať vymenovaním prípustných kombinácií hodnôt relevantných premenných, bolo by to nepraktické pre počet týchto kombinácií (najmenší počet by bol pre ôsmy stĺpec – 24 prípustných možností, najväčší pre tretí riadok – 40320 prípustných možností).

V praktických úlohách sa vyskytujú ohraničenia rôznych árností. Pretože však unárne ohraničenia je možné zohľadniť príslušnou redukciou domén jednotlivých premenných (resp. naopak je ich možné rozšíriť na binárne ohraničenia) a ohraničenia vyšších árností môžu byť redukované na binárne ohraničenia (aj keď niekedy za cenu zavedenia nových pomocných premenných) [1] [39], väčšina prác sa zaoberá práve binárnymi problémami s ohraničeniami – problémami, v ktorých všetky ohraničenia sú definované najviac nad dvoma premennými.

Použitý ilustračný problém používa špeciálny typ ohraničení – „alldiff“ ohraničenia (všetky premenné v rámci takéhoto ohraničenia musia mať navzájom rôzne hodnoty). Pri tomto type sa 9-árne ohraničenie dá nahradiť 84 ternárnymi alebo 36 binárnymi ohraničeniami rovnakého typu (po jednom pre každý výber troch alebo dvoch premenných z daných deviatich relevantných premenných).

Pri reprezentácii predpisom je možné ohraničenie použiť iba na testovanie, či nejaká kombinácia hodnôt premenných je z hľadiska daného ohraničenia prípustná alebo nie. Príkladom môže byť ternárne ohraničenie: „tri premenné z toho istého riadku (resp. stĺpca či štvorca) musia mať navzájom rôzne hodnoty“. Takýmto ohraničením môže byť napr. $C_{12,13,22}$. Reprezentácia vymenovaním umožňuje nielen iba testovať ale aj použiť ohraničenie generatívnym spôsobom – priamo získať kombinácie hodnôt relevantných premenných, ktoré sú validné z pohľadu daného ohraničenia. Príkladom môže byť binárne ohraničenie $C_{21,31}$ pri vymenovaní majúce tvar množiny $\{(1,2), (1,3), \dots, (9,7), (9,8)\}$ so 72 dvojicami.

Okrem ohraničení na rôznosť hodnôt kombinácií premenných problém ohraničuje priamo niektoré premenné. Toto reprezentujú unárne ohraničenia dané vymenovaním prípustných hodnôt napr. $C_{13}=\{8\}$.

Keď úloha pre niektoré premenné resp. kombinácie premenných nedefinuje ohraničenia priamo, je možné takéto ohraničenia formálne zaviesť. Aby takéto ohraničenie nemenilo definíciu úlohy, povoľuje všetky možné kombinácie hodnôt relevantných premenných. Tak je možné zaviesť napr. $C_{31,42}=\{(1,1), \dots, (9,9)\}$ s 81 dvojicami hodnôt, $C_{11}=\{1, \dots, 9\}$ s deviatimi hodnotami a $C_{31,32,42}$ so 729 trojicami hodnôt.

Keďže je potrebné splniť všetky ohraničenia (teda aj unárne), je možné znížiť komplexnosť problému redukciou domén jednotlivých premenných. Po tejto redukcii premennej V_i bude zodpovedať redukovaná doména, ktorá vznikne z pôvodnej domény vypustením všetkých prvkov nespĺňajúcich unárne ohraničenie viažúce sa na premennú V_i .

Táto redukcia sa môže rozšíriť aj na definície ohraničení. Ak binárne ohraničenie povoľuje nejakú takú kombináciu hodnôt, ktorá obsahuje hodnotu, porušujúcu unárne ohraničenie viažúce sa k príslušnej premennej, potom je možné toto binárne

ohraničenie „sprísniť“ a danú kombináciu hodnôt nepovolíť. Takýmto spôsobom dochádza k zvyšovaniu prísnosti ohraničení (danej pomerom počtu kombinácií hodnôt povolených daným ohraničením voči všetkým možným kombináciám hodnôt).

Napríklad keďže unárne ohraničenie C_{13} obsahuje iba jednu hodnotu 8 zatiaľ čo binárne ohraničenie $C_{12,13}$ obsahuje 72 dvojíc, je možné $C_{12,13}$ sprísniť iba na osem prípustných dvojíc $C_{12,13} = \{(1,8), (2,8), \dots, (7,8), (9,8)\}$. Podobne je možné sprísniť na osem dvojíc aj $C_{12,22}$. A následne ternárne ohraničenie $C_{12,13,22}$ možno na základe $C_{12,13}$ a $C_{12,22}$ sprísniť na sedem prípustných trojíc (ak by bolo reprezentované vymenovaním a nie predpisom).

Obidva uvedené prípady (vymenovanie a predpis) ilustrujú explicitný tvar ohraničenia. Ohraničenie však môže byť vyjadrené aj implicitným spôsobom – samotné ohraničenie nie je priamo uvedené ale je reprezentované iba prostredníctvom iných ohraničení, z ktorých môže byť indukované. Príkladom je $C_{41,32}$. Keďže zahŕňa dve pozície v rozličných stĺpcoch, riadkoch aj štvorcoch, nijako neohraničuje hodnoty na daných dvoch pozíciách. Avšak napr. ohraničenia C_{22} , $C_{22,32}$, C_{15} , $C_{14,15}$ indukujú neprípustnosť dvojíc $(7, \dots)$ a $(\dots, 5)$ pre $C_{41,32}$ ktoré tým pádom by malo povoľovať iba 64 dvojíc. Takúto indukciu je možné (aspoň čiastočne) urobiť pri reprezentácii problému, ale zvyčajne sa to nerobí a prenecháva sa to príslušným metódam pre riešenie úloh s ohraničeniami.

Ilustračný príklad bude teda reprezentovaný tak, že unárne ohraničenia budú dané podľa Obrázku 2, binárne ohraničenia budú reprezentované vymenovaním povolených dvojíc, pričom budú sprísnené v závislosti od obsahu unárnych ohraničení, a ternárne a viacárne ohraničenia budú dané predpisom.

Domény premenných vlastne definujú priestor. Je to vlastne priestor všetkých potenciálnych riešení (kandidátov riešení) – všetkých možných kombinácií hodnôt z domén premenných. Ak každá z domén bude reprezentovaná ako jedna súradnica tohto priestoru, potom každý bod tohto priestoru bude reprezentovať jednu kombináciu hodnôt premenných. Tento priestor takto vznikne ako kartézsky súčin domén premenných – ako množina všetkých kombinácií hodnôt jednotlivých premenných.

3 Algoritmické splňanie ohraničení

Táto trieda metód bola v minulosti označovaná rôznymi názvami napr. *konzistentné značkovanie* (consistent labeling) či *spĺňajúce priradenia* (satisficing assignment). Napokon sa pre ňu vžilo označenie *spĺňanie ohraničení* (constraint satisfaction).

Jednou z prvých aplikačných domén sa stala interaktívna grafika, kde prvým systémom bol Sutherlandov Sketchpad [35] začiatkom šesťdesiatych rokov. Jeho pokračovateľom je Borningov ThingLab [5]. Tieto systémy dovoľovali používateľovi kresliť a manipulovať s geometrickými obrázkami, podliehajúcimi ohraničeniam, po displeji počítača.

Prvé práce spojené s oblasťou algoritmov momentálne zahŕňaných pod týmto spoločným názvom siahajú do polovice sedemdesiatych rokov. Popudom k rozvoju v tejto oblasti sa stali algoritmy používané pre interpretáciu trojrozmerných scén na základe dvojrozmerných obrazov (bolo potrebné zohľadňovať ohraničenia pri interpretácii čiarových obrazov – napr. trojrozmerná interpretácia nejakej čiary ako

hrany musí byť rovnaká na oboch jej koncoch). Vtedajší študent MIT AI Lab (Laboratória pre umelú inteligenciu na Massachusettskom technologickom inštitúte) David Waltz si uvedomoval kombinatorickú explóziu znižujúcu použiteľnosť vtedy známych interpretačných metód a preto pre jej zmenšenie vytvoril filtračný algoritmus na odstránenie nekonzistentných interpretácií [38].

Ďalší ranný vývoj v tejto oblasti je spojený s takými menami ako Eugene C. Freuder, John Gaschnig, Robert M. Haralick, Alan K. Mackworth a Ugo Montanari [23] [13] [17]. Došlo k abstrahovaniu vlastností algoritmov, k ich oddeleniu od aplikácií a k formovaniu oblasti do takeého tvaru, v akom je známa dnes. Popisy základných algoritmov pre spĺňanie ohraničení je možné nájsť napr. v prehľadových publikáciách [22] a [10]. Z on-line dostupných rozsiahlejších publikácií sú k dispozícii [36] (v angličtine) a [25] (v slovenčine).

Jednou z tried algoritmov sú *konzistenčné algoritmy* (consistency enforcing algorithms). Podstatou týchto algoritmov je orezanie priestoru kandidátov riešení na nejaký jeho podpriestor. Ak úloha má jedno alebo viac riešení, tak tento výsledný podpriestor v ideálnom prípade bude obsahovať iba také body, ktorých každá súradnica je súčasťou aspoň jedného riešenia (ak úloha má iba jedno riešenie, tak tento podpriestor bude obsahovať iba jeden bod). Ak však úloha nemá riešenie ktoré by splnilo všetky definované ohraničenia, tak priestor kandidátov je možné redukovať na prázdny podpriestor.

Samotná redukcia je založená na filtrácii. Pri filtrácii sa vyhľadávajú také hodnoty z domén premenných, ktoré nie sú súčasťou žiadneho riešenia. V doménach premenných sú ponechávané iba hodnoty, ktoré sú konzistentné s definovanými ohraničeniami.

V procese filtrácie môže dochádzať nielen k redukcii domén ale aj k redukcii ohraničení, keď ohraničenie je sprísňované – nejaká kombinácia hodnôt premenných (ktoré sú relevantné pre dané ohraničenie) je označená ako neprípustná. Obidva tieto typy redukcie sú navzájom previazané. Zakázanie nejakej kombinácie hodnôt môže viesť k zbytočnosti niektorých hodnôt v doménach premenných alebo naopak zakázanie nejakej hodnoty v doméne nejakej premennej implikuje neprípustnosť všetkých kombinácií hodnôt v ktorých daná hodnota vystupuje.

Podobne môže dochádzať k redukcii medzi ohraničeniami rôznych árností. Zakázanie nejakej kombinácie i premenných (sprísnenie i -árneho ohraničenia) môže mať za následok sprísnenie $(i+j)$ -árneho ohraničenia (vypustenie všetkých kombinácií obsahujúcich ako svoju súčasť danú kombináciu i premenných). Alebo sprísnenie $(i+j)$ -árneho ohraničenia môže podnietiť vypustenie všetkých tých kombinácií z i -árneho ohraničenia, ktoré už nie sú konzistentné s ohraničením s vyššou árnosťou.

Uvedená filtrácia môže mať reťazový charakter. Vypustenie nejakej hodnoty z domény nejakej premennej môže prostredníctvom sprísnenia nejakého ohraničenia spôsobiť redukcii domény inej premennej. Podobne sprísnenie jedného ohraničenia môže prostredníctvom redukcie domény nejakej premennej mať za následok sprísnenie iného ohraničenia definovaného nad danou premennou. Toto reťazenie môže nastať podobným spôsobom aj medzi ohraničeniami rôznych árností. Takýto reťazový proces filtrácie sa označuje ako *šírenie ohraničení* (constraint propagation).

Výhodou konzistenčných algoritmov je, že ak robia „fair“ šírenie ohraničení (t.j. každé ohraničenie, ktorého premenné sa zmenili, je znovu aktivované), tak takéto šírenie vedie k jedinečnému stavu siete ohraničení – a teda nezáleží na poradí, v ktorom sú jednotlivé ohraničenia uvažované [15].

Konzistenčné algoritmy pracujú s pojmom *konzistencie*. Jedna z používaných podôb tohto pojmu je (i,j) -konzistencia, kde v úlohe hodnôt i a j môžu vystupovať prirodzené čísla z intervalu $\langle 1, n-1 \rangle$, pričom ich súčet je menší alebo rovný n . Voľne by sa dalo povedať, že sa jedná vlastne o konzistenciu podmnožín premenných z množiny všetkých premenných V a hodnota i vlastne udáva kardinalitu týchto podmnožín. Úlohou konzistenčných algoritmov je túto konzistenciu zabezpečiť. Konzistenčné algoritmy vlastne zabezpečia, že nejaká subsieť siete ohraničení je konzistentná voči svojmu okoliu. Pritom pod týmto okolím sa chápu nejaké premenné, ktoré sa nevyskytujú v danej subsieti – teda takýchto okolí pre nejakú subsieť môže existovať vo všeobecnosti viac. Formálne je možné povedať, že sieť ohraničení je (i,j) -konzistentná, ak platí:

Nech ľubovoľným i premenným sú priradené také hodnoty z ich domén, aby všetky ohraničenia definované nad touto i -ticou premenných boli splnené. Pre ľubovoľných ďalších j premenných je možné vybrať také hodnoty z ich domén, že všetky ohraničenia definované nad vzniknutou $(i+j)$ -ticou premenných sú splnené.

Ohraničenie definované nad i premennými je (i,j) -konzistentné, ak jeho explicitné vyjadrenie nepripúšťa žiadnu takú kombináciu i hodnôt, ktorá nie je prípustná indukovanou podobou daného ohraničenia (v príklade na Obrázku 1 napr. explicitná podoba $C_{11,12}$ obsahuje 72 prípustných dvojíc, avšak indukovaná podoba tohto ohraničenia obsahuje prípustných dvojíc menej – napr. vďaka ohraničeniam $C_{11,13}$ a $C_{12,13}$ nie sú prípustné dvojice, obsahujúce hodnotu 8). Inak povedané, každá kombinácia hodnôt povolená explicitnou podobou ohraničenia sa dá konzistentne rozšíriť o hodnoty ďalších j premenných.

Takto definovaná (i,j) -konzistencia teda vlastne znamená, že sieť ohraničení je práve vtedy (i,j) -konzistentná, ak neexistuje žiadne také ohraničenie definované nad skupinou i premenných, ktoré by nebolo (i,j) -konzistentné. Sieť ohraničení je prísne (i,j) -konzistentná, ak je súčasne (k,j) -konzistentná pre všetky hodnoty k , pre ktoré je splnená nerovnosť $1 \leq k \leq i$ (prísna konzistencia nejakého stupňa znamená konzistenciu daného stupňa ako aj všetkých nižších stupňov).

Použitie konzistenčných algoritmov je potom možné chápať ako transformáciu siete ohraničení na ekvivalentnú ale explicitnejšiu sieť, pričom existujúce ohraničenia sú použité pre odvodenie ďalších a ich pridanie do siete. Zabezpečenie (i,j) -konzistencie potom má podobu sprísňovania i -árnych ohraničení tak, aby boli konzistentné až s $(i+j)$ -árnymi ohraničeniami. Ak nejaké ohraničenie nie je explicitne dané, tak je ho možné dedefinovať – bude jednoducho reprezentovať ohraničenie, pripúšťajúce všetky kombinácie hodnôt z domén príslušných premenných.

Pojem $(1,1)$ -konzistencie znamená, že v doméne ľubovoľnej premennej V_i sa nachádzajú iba také hodnoty, ktoré ak spĺňajú ohraničenie C_i , tak sú zároveň

konzistentné s doménou ľubovoľnej inej premennej V_j , $j \neq i$. To znamená, že pre každé $h_i \in D_i$ existuje taká hodnota $h_j \in D_j$, že dvojica (h_i, h_j) neporušuje ohraničenia definované nad premennými V_i a V_j (napr. $C_{i,j}$ a C_j). Takýto typ konzistencie v sieti ohraničení zahŕňa vždy dva uzly reprezentujúce dve premenné a preto sa nazýva *hranovou konzistenciou* a označuje sa AC (arc consistency).

Použitý ilustračný príklad na Obrázku 1 nie je hranovo konzistentný. Dôvodom je napr. to, že C_{31} nie je konzistentné s $C_{31,38}$, $C_{31,51}$ a $C_{31,61}$ – obsahuje totiž aj hodnoty 1, 7 a 9, pričom však tieto hodnoty nemôžu byť konzistentne rozšírené o hodnoty premenných V_{38} , V_{51} alebo V_{61} bez porušenia príslušného binárneho ohraničenia.

Podobne (2,1)-konzistencia znamená, že ak z domén ľubovoľných dvoch rôznych premenných V_i a V_j je možné vybrať takú dvojicu hodnôt, ktorá neporušuje žiadne ohraničenie, tak táto dvojica je konzistentná s doménou ľubovoľnej inej premennej V_k , pričom $k \neq i$ a $k \neq j$. To znamená, že pre každú dvojicu (h_i, h_j) ($h_i \in D_i$ a $h_j \in D_j$) neporušujúcu žiadne z trojice ohraničení $C_{i,j}$, C_i a C_j existuje nejaká hodnota $h_k \in D_k$ taká, že trojica (h_i, h_j, h_k) neporušuje žiadne z ohraničení $C_{i,j,k}$, $C_{i,k}$, $C_{j,k}$ a C_k .

Táto konzistencia vlastne zaručuje, že ľubovoľná kombinácia dvoch hodnôt z domén dvoch rôznych premenných, povolená priamym ohraničením definovaným nad týmito dvoma premennými, je súčasne povolená všetkými možnými cestami dĺžky 2 medzi týmito dvoma premennými. Preto sa nazýva konzistenciou *po ceste* a označuje PC (path consistency).

Použitý ilustračný príklad nie je konzistentný po ceste. Dôvodom je napr. to, že $C_{12,22}$ nie je konzistentné s $C_{12,22,13}$ a $C_{12,13}$ – obsahuje totiž aj dvojicu (8,5), pričom však táto nemôže byť konzistentne rozšírená o hodnotu premennej V_{13} bez porušenia príslušného ternárneho ohraničenia.

Podobne (1,2)-konzistencia znamená, že v doméne ľubovoľnej premennej V_i sa nachádzajú iba také hodnoty, ktoré ak spĺňajú ohraničenie C_i , tak sú zároveň konzistentné s doménami ľubovoľných iných dvoch premenných V_j , $j \neq i$ a V_k , $k \neq i$ a $k \neq j$. To znamená, že pre každé $h_i \in D_i$ neporušujúce ohraničenie C_i existujú také hodnoty $h_j \in D_j$ a $h_k \in D_k$, že trojica (h_i, h_j, h_k) neporušuje žiadne z ohraničení $C_{i,j,k}$, $C_{i,j}$, $C_{i,k}$, $C_{j,k}$, C_j a C_k .

Táto konzistencia vlastne zaručuje, že ľubovoľná hodnota z domény premennej umožní indukovať také ohraničenie nad dvojicou ďalších dvoch premenných, ktoré je povolené priamym ohraničením definovaným nad týmito dvoma premennými. Preto sa nazýva konzistenciou *inverznou po ceste* a označuje PIC (path inverse consistency).

Použitý ilustračný príklad nie je inverzne konzistentný po ceste. Dôvodom je napr. to, že C_{11} nie je konzistentné s $C_{11,13}$ a $C_{11,12,13}$ – obsahuje totiž aj hodnotu 8, pričom však táto hodnota nemôžu byť konzistentne rozšírené o hodnoty premenných V_{12} a V_{13} bez porušenia príslušného ternárneho ohraničenia.

Analogickým spôsobom je možné definovať konzistencie vyšších stupňov. Keďže (i,j)-konzistencia znamená, že i-tice hodnôt je možné konzistentne rozšíriť o ďalších j hodnôt, dosahovanie takejto konzistencie vlastne znamená nutnosť úpravy (sprísnovania) i-árnych ohraničení. Dôležité je uvedomiť si, že definícia konzistencie vyššieho stupňa nevyžaduje konzistenciu stupňa nižšieho a naopak. Platnosť resp. neplatnosť (i,j)-konzistencie vo všeobecnosti nič nehovorí o platnosti alebo neplatnosti (k,l)-konzistencie pre $i \neq k$ alebo $j \neq l$. A teda (i,j)-konzistencia neovplyvňuje priamo

domény premenných v prípade $i > 1$. Preto po dosiahnutí vyššieho stupňa konzistencie musí nasledovať zabezpečenie aj nižších stupňov, aby sa sprísnenie ohraničení vyššej árnosti rozšírilo aj na ohraničenia nižších árností a cez unárne ohraničenia ovplyvnilo domény premenných.

Z praktických dôvodov sa konzistencie vyšších stupňov nepoužívajú kvôli vysokým výpočtovým nárokom na ich zabezpečenie (majú exponenciálnu zložitosť). My sa budeme venovať iba tým najjednoduchším, ktoré ovplyvňujú priamo unárne ohraničenia premenných a tým vlastne aj priamo ich domény. Cenou za dosiahnutie iba nízkeho stupňa konzistencie je nie vždy dostatočný stupeň redukcie ohraničení. Vo všeobecnosti totiž garanciu, že v doménach premenných ostanú iba hodnoty participujúce aspoň na jednom z riešení, môže poskytnúť iba prísna (i,j) -konzistencia v prípade, že $i + j = n$.

Pretože existujú rôzne typy konzistencie, musia existovať aj rôzne navzájom odlišné algoritmy, ktoré umožňujú tieto konzistencie dosahovať. Vo všeobecnosti, od toho aký typ konzistencie daný algoritmus zabezpečuje, závisí aj zložitosť (a tým aj praktická použiteľnosť) tohto algoritmu.

Pre zabezpečenie hranovej konzistencie je potrebné porovnávať navzájom domény dvojíc premenných. Ak jedna doména obsahuje takú hodnotu, pre ktorú nie je možné vybrať z domény druhej premennej žiadnu hodnotu bez toho, aby binárne ohraničenie definované nad týmito dvoma premennými bolo porušené, tak je nutné danú hodnotu z domény prvej premennej vypustiť.

Naviac je potrebné zohľadniť fakt, že aj keď v určitom momente je nejaká premenná V_i hranovo konzistentná voči premennej V_j , toto sa môže neskôr zmeniť po redukcii domény D_j prislúchajúcej premennej V_j – a teda vlastne redukcia domény premennej V_j zapríčiňuje nutnosť návratu ku kontrole premennej V_i .

Algoritmus pre zabezpečenie hranovej konzistencie môže mať viacero podôb s rôznou zložitosťou a priestorovými nárokmi, danými jednak spôsobom implementácie a jednak mierou redundantných testov. V literatúre je možné nájsť celý rad rôznych implementácií: AC1 opakovane revidujúcu premenné ak nastala nejaká zmena v unárnych ohraničeniach, AC2 realizujúcu iba jednu úplnú iteráciu cez všetky premenné [23] (je ekvivalentný Waltzovmu filtračnému algoritmu [38]), AC3 používajúcu dynamickú frontu uzlov čakajúcich na kontrolu [23], AC4 založenú na pojme podpory hodnôt premenných hodnotami iných premenných [27], AC5 v generickom tvare umožňujúcu špecializáciu okrem iného aj na AC3 a AC4 [37], AC6 narábajúcu s nutnou podporou hodnôt premenných [3], AC6+ [4] a AC7 [14] založené na symetrii ohraničení a AC8 dekomponujúcu problém do série menších problémov [20].

Principiálny algoritmus AC pre zabezpečenie hranovej konzistencie môže mať podobu podľa Programu 1 (fragment reálneho kódu v jazyku Common Lisp).

```

(defun vyhovuju-2-hodnoty-ohraniceniu (preml hodn1 prem2 hodn2)
  (member (cons hodn1 hodn2)
          (binarne-ohranicenie preml prem2) :test #'equal))

(defun vyhovuje-hodnota-premennej (premenna1 hodnota1 premenna2)
  (some #'(lambda (hodnota2)
            (vyhovuju-2-hodnoty-ohraniceniu
             premenna1 hodnota1 premenna2 hodnota2))
        (unarne-ohranicenie premenna2)))

(defun vyhovuje-hodnota-premennym (premenna hodnota)
  (every #'(lambda (premenna2)
             (vyhovuje-hodnota-premennej premenna hodnota premenna2))
         (zavisle-premenne premenna)))

(defun AC (&optional (zoznam *zoznam-premennych*))
  (unless (null zoznam)
    (let ((premenna (first zoznam)) zmena)
      (dolist (hodnota (unarne-ohranicenie premenna))
        (unless (vyhovuje-hodnota-premennym premenna hodnota)
          (redukcia-unarneho-ohranicenia premenna hodnota)
          (setf zmena t))))
      (if (not zmena)
          (AC (rest zoznam))
          (AC (union (rest zoznam) (zavisle-premenne premenna)))))))

```

Program 1. Fragment implementácie AC algoritmu.

Algoritmus pracuje so zoznamom premenných, ktorých unárne ohraňenia je potrebné skontrolovať (a prípadne sprísiť). Tento zoznam na začiatku obsahuje zoznam všetkých premenných. Postupne sa z neho odoberajú premenné a kontroluje sa ich konzistencia. Ak sa zmení unárne ohraňenie niektorej premennej, do tohto zoznamu sú pridané (ak už z neho boli odstránené) všetky tie premenné, ktoré na danej premennej závisia prostredníctvom binárnych ohraňení. Algoritmus končí až sa zoznam vyprázdni.

Kontrola hodnoty unárneho ohraňenia nejakej premennej V_i sa deje voči všetkým premenným, ktoré sú závislé na V_i prostredníctvom binárnych ohraňení (iba premenné viazané binárnym ohraňením s V_i môžu byť ovplyvnené zmenou C_i). Akonáhle sa zistí nekonzistencia hodnoty voči niektorej z týchto premenných, kontrola končí neúspechom, pričom kontrola danej hodnoty voči zostávajúcim závislým premenným sa vynechá.

Pri kontrole hodnoty unárneho ohraňenia nejakej premennej V_i voči inej premennej V_j sa kontroluje iba dovtedy, pokiaľ sa nezistí konzistencia. Akonáhle sa zistí, kontrola končí úspechom a kontrola danej hodnoty z C_i voči ostatným hodnotám z C_j sa vynechá.

Dve hodnoty (jedna z C_i a druhá z C_j) sú konzistentné vtedy, ak sa daná kombinácia nachádza v zozname povolených dvojíc binárneho ohraňenia $C_{i,j}$.

Aj PIC algoritmus môže mať viacero podôb s rôznou zložitou a priestorovými nárokmi. V literatúre je možné nájsť napr. PIC1 pamätajúci si iba vypúšťané hodnoty ale s veľkou časovou zložitou a PIC2 s optimálnou časovou zložitou [7]. Princiálna podoba algoritmu PIC je veľmi podobná uvedenému AC algoritmu.

Rozdiel je iba v tom, že funkcia *vyhovuje-hodnota-premennym* v tomto prípade kontroluje danú hodnotu voči nie jednej ale dvom premenným, pričom daná hodnota musí vyhovovať nielen každej z kontrolovaných premenných osobitne (teda je potrebné dvojnásobné volanie funkcie *vyhovuje-hodnota-premennej*) ale aj obom súčasne a teda pribudnú dve nové funkcie *vyhovuje-hodnota-dvom-premennym* (musí byť nájdená aspoň jedna kombinácia hodnôt tých dvoch premenných podporujúca kontrolovanú hodnotu) a *vyhovuju-3-hodnoty-ohraniceniu* (kontrola voči relevantnému 3-árnemu ohraničeniu).

Existujú algoritmy, ktoré umožňujú dosiahnuť väčšiu redukciu ohraničení ako ponúka nejaký stupeň konzistencie, avšak menšiu ako ponúka vyšší stupeň konzistencie (je to vlastne spájanie algoritmov do kooperačných schém) [2]. Príkladom je *redukovaná konzistencia po ceste* RPC (reduced path consistency) alebo alternatívy *k-RPC* a *max-RPC* [9]. RPC algoritmus pracuje v móde ako AC (porovnáva dvojice hodnôt premenných) a len v situácii, keď hodnota premennej V_i je konzistentná iba s jednou hodnotou z unárneho ohraničenia inej premennej V_j , dochádza k prepnutiu do módu PC a daná dvojica hodnôt je kontrolovaná, či môže byť konzistentne rozšírená o nejakú hodnotu tretej ľubovoľnej premennej V_k . Ak nejaká premenná toto rozšírenie neumožňuje, na rozdiel od PC algoritmu (ktorý by sprísnil zodpovedajúce binárne ohraničenie $C_{i,j}$) dochádza k redukcii unárneho ohraničenia C_i vypustením pôvodne kontrolovanej hodnoty.

Na rozdiel od RPC, variant *k-RPC* sa prepína do módu PC keď hodnota premennej V_i je konzistentná iba s k alebo menej hodnotami z unárneho ohraničenia inej premennej V_j (musí byť možnosť konzistentného rozšírenia na tri hodnoty aspoň pre jednu z hodnôt premennej V_j). Alternatíva *max-RPC* sa prepína do módu PC pre všetky hodnoty unárneho ohraničenia premennej V_j , s ktorými je konzistentná skúmaná hodnota premennej V_i .

Algoritmus by sa podobal algoritmu AC. Rozdiel by bol v tom, že v tomto prípade funkcia *vyhovuje-hodnota-premennej* by nekontrolovala iba či testovaná hodnota je podporovaná nejakou hodnotou druhej premennej, ale by aj určovala koľkými hodnotami je podporovaná. Pri počte rôznom od 1 by sa okamžite rozhodlo o úspechu alebo neúspechu. V prípade podpory iba jednou premennou by bola volaná nová funkcia *vyhovuju-dve-hodnoty-premennym* (dvojica hodnôt musí byť konzistentná s každou premennou zo zoznamu všetkých ďalších premenných), následne ďalšia nová funkcia *vyhovuju-dve-hodnoty-premennej* (dvojica hodnôt musí byť konzistentná aspoň s jednou hodnotou tretej premennej) a už spomínaná *vyhovuju-3-hodnoty-ohraniceniu*.

Okrem spomínaných algoritmov je možné použiť aj tzv. singletonový konzistenčný algoritmus, ktorý pracuje systémom pokus-omyl [8]. Validitu každej hodnoty v unárnom ohraničení premennej kontroluje tak, že dané unárne ohraničenie zúži na práve túto hodnotu a aplikuje nejaký redukčný algoritmus. Ak dôjde v procese redukcií k vyprázdneniu nejakého ohraničenia (signalizujúcemu neexistenciu riešenia), kontrolovanú hodnotu vypúšťa, inak ju ponecháva. V závislosti na tom, aký algoritmus je použitý na redukciu, je možné získať napr. SAC (singleton arc consistency), SPIC (singleton path inverse consistency) či SRPC (singleton restricted path consistency), atď.

Principiálna podoba tohto algoritmu má tvar podľa Programu 2 (opäť fragment reálneho kódu v jazyku Common Lisp).

```
(defun singleton-test-hodnoty (prem hodnota redukny-algoritmus)
  (let ((validita-hodnoty t))
    (ulozenie-stavu :unarne-ohranicenia)
    (nastavenie-unarneho-ohranicenia prem hodnota)
    (funcall redukny-algoritmus)
    (when (vyprazdnenie-ohranicenia)
      (setf validita-hodnoty nil))
    (obnovenie-stavu :unarne-ohranicenia)
    validita-hodnoty))

(defun singleton (algoritmus)
  (do ((zmena t) ((not zmena))
      (setf zmena nil)
      (dolist (premenna *zoznam-premennych*)
        (dolist (hodnota (unarne-ohranicenie premenna))
          (unless (singleton-test-hodnoty premenna hodnota algoritmus)
            (redukcia-unarneho-ohranicenia premenna hodnota)
            (setf zmena t)))))))

(defun SAC () (singleton #'AC))
(defun SPIC () (singleton #'PIC))
(defun SRPC () (singleton #'RPC))
```

Program 2. Fragment implementácie všeobecného singletonového algoritmu a jeho využitie pre implementáciu algoritmov SAC, SPIC a SRPC.

Je potrebné však povedať, že singletonový algoritmus realizuje mnohonásobné opakovanie redukčného algoritmu (pre každú potenciálnu hodnotu každej premennej), čo sa výrazne prejaví na výpočtovej náročnosti. Preto je vhodné pred jeho začiatkom znížiť počet hodnôt premenných, na ktoré bude musieť byť aplikovaný (napr. použitím AC).

Ukážme si ako je možné použiť popísané algoritmy pre riešenie ukážkového príkladu podľa Obrázku 1. Ten istý prípad je znázornený aj na Obrázku 2, pričom na jednotlivých pozíciách mriežky sú zobrazené unárne ohraničenia jednotlivých premenných (na obsadenej pozícii existuje iba jedna možnosť, na neobsadenej zatiaľ prichádzajú do úvahy všetky možnosti). Ak sa v takejto situácii použije AC algoritmus, tak napr. pre C_{17} prebehne redukcia. Postupne budú vypustené hodnoty: 1 pre chýbajúcu podporu v C_{38} , 2 kvôli C_{14} , 4 pre C_{97} , 5 pre C_{16} , 6 pre C_{18} , 7 pre C_{28} , 8 pre C_{13} a 9 pre C_{57} . Takýmto spôsobom bola C_{16} zredukovaná na jednu hodnotu a následne bolo možné hodnotu 3 vypustiť z ohraničení všetkých premenných reprezentujúcim ten istý riadok, stĺpec a štvorec ako prislúcha C_{16} . Výsledkom bolo orezanie unárnych ohraničení podľa Obrázku 3.

V situácii podľa Obrázku 3 už použitie AC neprináša ďalšiu redukciu ohraničení. Ak sa použije napr. RPC, tak redukcia unárnych ohraničení môže pokračovať. Pri skúmaní hodnoty 8 z C_{77} voči C_{27} sa zistila podpora iba hodnotou 2. Keďže podpora bola iba jednou hodnotou, prepol sa mód a začalo sa skúmať, či kombinácia 8 z C_{77} a 2 z C_{27} je konzistentná voči unárnemu ohraničeniu každej zo zvyšných premenných.

Keďže z C_{87} nebolo možné danú kombináciu hodnôt validne doplniť žiadnou hodnotou, tak hodnota 8 bola vypustená z C_{77} . Ďalšími redukciami sú napr. vypustenie 2 z C_{67} (rovnaký dôvod ako bol pre vypustenie 8 z C_{77}), 6 z C_{67} (v konflikte s aktuálnym C_{77}), atď. Postupnou redukciami sa dospeje do situácie na Obrázku 4.

| | | | | | | | | |
|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|
| 1 2 3 4 5 6 7 8 9 | 1 2 3 4 5 6 7 8 9 | 8 | 2 | 1 2 3 4 5 6 7 8 9 | 5 | 1 2 3 4 5 6 7 8 9 | 6 | 1 2 3 4 5 6 7 8 9 |
| 1 2 3 4 5 6 7 8 9 | 5 | 3 | 9 | 1 2 3 4 5 6 7 8 9 | 1 2 3 4 5 6 7 8 9 | 1 2 3 4 5 6 7 8 9 | 7 | 1 2 3 4 5 6 7 8 9 |
| 1 2 3 4 5 6 7 8 9 | 1 2 3 4 5 6 7 8 9 | 1 2 3 4 5 6 7 8 9 | 1 2 3 4 5 6 7 8 9 | 1 2 3 4 5 6 7 8 9 | 1 2 3 4 5 6 7 8 9 | 1 2 3 4 5 6 7 8 9 | 1 | 1 2 3 4 5 6 7 8 9 |
| 1 2 3 4 5 6 7 8 9 | 1 2 3 4 5 6 7 8 9 | 1 2 3 4 5 6 7 8 9 | 3 | 1 2 3 4 5 6 7 8 9 | 1 2 3 4 5 6 7 8 9 | 1 | 1 2 3 4 5 6 7 8 9 | 1 2 3 4 5 6 7 8 9 |
| 7 | 1 2 3 4 5 6 7 8 9 | 1 2 3 4 5 6 7 8 9 | 1 2 3 4 5 6 7 8 9 | 1 2 3 4 5 6 7 8 9 | 8 | 9 | 5 | 1 2 3 4 5 6 7 8 9 |
| 9 | 1 2 3 4 5 6 7 8 9 | 1 2 3 4 5 6 7 8 9 | 1 2 3 4 5 6 7 8 9 | 1 2 3 4 5 6 7 8 9 | 1 2 3 4 5 6 7 8 9 | 1 2 3 4 5 6 7 8 9 | 1 2 3 4 5 6 7 8 9 | 8 |
| 1 2 3 4 5 6 7 8 9 | 4 | 1 2 3 4 5 6 7 8 9 | 1 2 3 4 5 6 7 8 9 | 2 | 1 2 3 4 5 6 7 8 9 | 1 2 3 4 5 6 7 8 9 | 9 | 5 |
| 1 2 3 4 5 6 7 8 9 | 1 2 3 4 5 6 7 8 9 | 6 | 1 2 3 4 5 6 7 8 9 | 1 2 3 4 5 6 7 8 9 | 1 2 3 4 5 6 7 8 9 | 1 2 3 4 5 6 7 8 9 | 1 2 3 4 5 6 7 8 9 | 7 |
| 1 2 3 4 5 6 7 8 9 | 1 2 3 4 5 6 7 8 9 | 1 2 3 4 5 6 7 8 9 | 1 2 3 4 5 6 7 8 9 | 5 | 1 2 3 4 5 6 7 8 9 | 4 | 1 2 3 4 5 6 7 8 9 | 1 2 3 4 5 6 7 8 9 |

Obrázok 2. Ilustračný problém s unárnymi ohraničeniami premenných.

Ak by sa však v Obrázku 3 namiesto RPC použilo PIC, tak napr. pri skúmaní C_{37} sa každá hodnota z tohto ohraničenia bude kontrolovať voči dvojiciam premenných aby sa overilo, či je s unárnymi ohraničeniami týchto premenných konzistentná. Hodnota 2 bude vypustená, pretože pri porovnaní voči C_{27} a C_{87} nebolo možné vybrať takú kombináciu rôznych hodnôt tak, aby obe boli rôzne od hodnoty 2. Z analogického dôvodu bolo potrebné vypustiť aj hodnotu 8 z C_{37} .

Pokračovať v redukcii unárných ohraničení v situácii ilustrovanej Obrázkom 4 možno napr. prostredníctvom algoritmu max-RPC. Pri kontrole validity hodnoty 4 z C_{49} voči C_{39} je síce táto hodnota podporovaná dvomi hodnotami z C_{39} , avšak ani v jednom prípade nie je možné rozšírenie o tretiu hodnotu buď z C_{19} alebo C_{29} (v závislosti na použitej hodnote z C_{39}). Toto vedie k vypusteniu hodnoty 4 z C_{49} . Z rovnakého dôvodu dôjde k vypusteniu hodnoty 4 aj z ohraničenia C_{59} .

| | | | | | | | | |
|-----------------|----------------|--------------|--------------|--------------|--------------|------------|----------|-------------------|
| 1 4 | 1 7 9 | 8 | 2 | 1 4 7 | 5 | 3 | 6 | 4 9 |
| 1 2 4 6 | 5 | 3 | 9 | 1 4 6 8 | 1 4 6 | 2 8 | 7 | 2 4 |
| 2 4 6 | 2 7 9 | 2 4 5 7 9 | 4 7 8 | 3 4 6 7 8 | 3 4 6 7 | 2 5 8 | 1 | 2 4 9 |
| 2 4 5 6 8 | 2 6 8 | 2 4 5 | 3 | 4 7 9 | 2 4 6 7 9 | 1 4 6 | 2 4 6 | 2 4 6 |
| 7 | 1 2 3 6 4 | 1 2 6 4 | 1 4 6 | 1 4 6 | 8 | 9 | 5 | 2 3 4 6 |
| 9 | 1 2 3 6 4 5 | 1 2 6 4 5 | 1 4 5 6 7 | 1 4 6 7 | 1 2 4 6 7 | 2 6 7 | 2 3 4 | 8 |
| 1 8 | 3 4 | 1 7 | 1 7 8 | 6 2 | 1 7 | 3 6 8 | 9 | 5 |
| 1 2 3 5 8 | 1 2 3 8 9 | 6 | 1 4 8 | 1 4 8 9 | 1 3 4 9 | 1 3 4 8 | 2 8 | 2 3 7 |
| 1 2 3 8 | 1 2 3 7 8 9 | 1 2 7 9 | 1 7 8 | 6 5 | 1 7 9 | 3 6 7 | 4 | 2 3 1 2 3 6 |

Obrázok 3. Ilustračný problém s unárnymi ohraničeniami premenných po aplikovaní algoritmu AC.

Redukcia ohraničení by mala inú podobu, ak by v situácii na Obrázku 4 bol použitý algoritmus SAC. Napríklad je možné začať skúmať hodnotu 2 v C_{27} . Ak by sa C_{27} zredukovalo iba na túto hodnotu, tak pre zabezpečenie hranovej konzistencie je nutné redukovať C_{29} na hodnotu 4 (teraz hodnota 2 nemá oporu v C_{27}) a C_{39} na hodnotu 9 (hodnota 2 nemá oporu v C_{27} a hodnota 4 zase v C_{29}). Následkom úprav C_{29} a C_{39} bude C_{19} redukovaná na prázdnu množinu – a to je signál, že pôvodný predpoklad (redukcia C_{27} na hodnotu 2) bol chybný a teda hodnota 2 je odstránená z C_{27} . Opakovaným použitím SAC sa získa situácia podľa Obrázku 5.

Ak sa v situácii podľa Obrázku 5 použije SPIC, tak možno napr. redukovať C_{15} na hodnotu 4. Následné skúmanie hodnoty 4 v C_{25} spôsobí jej vypustenie (nebude možné ju konzistentne rozšíriť o ďalšie dve hodnoty ak jedna z nich bude braná z C_{15}). Podobný osud postihne aj hodnotu 4 v C_{55} a C_{65} . Ak potom sa bude uvažovať ľubovoľná hodnota z aktuálneho C_{25} , tak sa nebude dať doplniť žiadnou kombináciou hodnôt z C_{55} a C_{65} – a teda C_{25} sa vyprázdni. Znamená to, že prvotné zúženie C_{15} na hodnotu 4 nie je možné a táto hodnota je vypustená z C_{15} . Z úplne rovnakého dôvodu je nutné vypustiť z C_{15} aj hodnotu 1.

| | | | | | | | | |
|-----------------|----------------|-----------------|-----------------|----------------------|-----------------|------------|----------|-------------------|
| 1 4 | 1 7 9 | 8 | 2 | 1 4 7 | 5 | 3 | 6 | 4 9 |
| 1 2 4 6 | 5 | 3 | 9 | 1 4 6 4 6 8 | 1 4 6 | 2 8 | 7 | 2 4 9 |
| 2 4 6 | 2 7 9 | 2 6 4 7 9 | 2 4 6 7 8 | 3 4 6 7 8 | 3 4 6 7 | 5 | 1 | 2 4 9 |
| 2 4 5 6 8 | 2 8 | 2 4 5 | 3 | 2 4 6 7 9 | 2 4 6 7 9 | 1 | 2 4 | 2 4 6 |
| 7 | 1 2 3 6 4 | 1 2 6 4 | 1 4 6 | 1 4 6 | 8 | 9 | 5 | 2 3 4 6 |
| 9 | 1 2 3 6 4 5 | 1 2 6 4 5 | 1 4 5 6 | 1 4 6 | 1 2 4 6 | 7 | 2 3 4 | 8 |
| 1 3 8 | 4 | 1 7 | 1 7 8 | 2 7 | 1 3 7 | 6 | 9 | 5 |
| 1 2 3 5 8 | 1 2 3 8 9 | 6 | 1 4 8 | 1 3 4 8 9 | 1 3 4 9 | 2 8 | 2 3 8 | 7 |
| 1 2 3 8 | 1 2 3 7 8 9 | 1 2 7 9 | 1 7 8 | 6 7 8 | 5 | 1 3 7 9 | 4 | 2 3 1 2 3 8 |

Obrázok 4. Ilustračný problém s unárnymi ohraničeniami premenných po aplikovaní algoritmov AC a RPC.

| | | | | | | | | |
|-----------------|----------------|-----------------|-----------------|----------------------|---------------|--------|--------|-------------|
| 1 4 | 1 7 9 | 8 | 2 | 1 4 7 | 5 | 3 | 6 | 4 9 |
| 1 2 4 6 | 5 | 3 | 9 | 1 4 6 4 6 8 | 1 4 6 | 8 | 7 | 2 4 9 |
| 2 4 6 | 2 7 9 | 2 6 4 7 9 | 2 4 6 7 8 | 3 4 6 7 8 | 3 4 6 7 | 5 | 1 | 2 4 9 |
| 2 4 5 6 8 | 2 8 | 2 4 5 | 3 | 7 7 9 7 9 | 7 7 9 | 1 | 2 4 | 6 |
| 7 | 1 2 3 6 4 | 1 2 6 4 | 1 4 6 | 1 4 6 | 8 | 9 | 5 | 3 |
| 9 | 1 2 3 6 4 5 | 1 2 6 4 5 | 1 4 5 6 | 1 4 6 | 1 2 4 6 | 7 | 2 4 | 8 |
| 1 3 8 | 4 | 1 7 | 1 7 8 | 2 7 | 1 3 7 | 6 | 9 | 5 |
| 1 3 5 8 | 1 3 8 9 | 6 | 1 4 8 | 1 3 4 8 9 | 1 3 4 9 | 2 8 | 3 8 | 7 |
| 2 3 8 | 2 3 7 8 9 | 2 7 9 | 6 7 8 | 5 | 3 7 9 | 4 | 3 8 | 1 |

Obrázok 5. Ilustračný problém s unárnymi ohraničeniami premenných po aplikovaní algoritmov AC, RPC a SAC.

Pre porovnanie výkonnosti jednotlivých algoritmov boli použité problémy z [16]. V ponuke bolo 100 úloh v štyroch kategóriách obtiažnosti. Ako doplnok bolo použitých aj 10 úloh z [21], zaradovaných medzi najťažšie doposiaľ objavené inštancie daného hlavolamu. Na základe priemerného stupňa orezania unárnych ohraničení (a tým pádom aj domén premenných) boli jednotlivé algoritmy zoradené takto: $AC < RPC = PIC < \text{max-RPC} < SAC < SRPC = SPIC < \text{Smax-RPC}$. V tomto prípade teda AC dosiahol najmenší stupeň redukcie a Smax-RPC zase najväčší. Na daných úlohách RPC mal rovnakú výkonnosť ako PIC (a teda aj SRPC ako SPIC). Toto zodpovedá zoradeniu algoritmov napr. podľa [9]. Rozdiel bol iba v tom, že PIC dosahuje minimálne také orezanie ako RPC (teda môže aj väčšie, nielen rovnaké ako to bolo v našom prípade).

Na súbore 100 úloh z [16] dokázali jednotlivé algoritmy riešiť problém nasledovne: AC úplne vyriešil 23 úloh, RPC a PIC vyriešili po 46 úloh, max-RPC 56 úloh, SAC bol úspešný v 97 prípadoch a SRPC, SPIC a Smax-RPC mali sto percentnú úspešnosť. Pri použití [21] boli všetky uvedené algoritmy neúspešné – hoci bol dosiahnutý určitý stupeň redukcie, ich schopnosti redukovať ohraničenia nestačili ani na jednu z úloh. Prvým úspešným algoritmom bol až DSAC (double singleton arc consistency), algoritmus analogický k SAC ibaže redukujúci binárne ohraničenia. Tento už vyriešil všetkých desať úloh (a taktiež všetkých sto z predchádzajúceho súboru).

4 Stochastické prehľadávanie v akcii

Existuje pomerne veľký počet algoritmov založených na stochastických princípoch, ktoré je možné použiť pre riešenie úloh s ohraničeniami. Ako príklad použijeme *evolučné algoritmy* (evolutionary algorithms). Tieto algoritmy tvoria širokú skupinu prístupov a smerov, ktorých jednotiacim prvkom je využívanie princípov prirodzeného výberu a génovej dedičnosti (od snahy presne kopírovať tieto mechanizmy cez využívanie ich zjednodušených podôb až po iba vzdialenú inšpiráciu týmito princípmi).

Hlavnými impulzmi (aj keď to neboli prvé impulzy v tejto oblasti) pre rozvoj týchto algoritmov sa na prelome šesťdesiatych a sedemdesiatych rokov stali práce I. Rechenberga [30] a H.P. Schwefela [32] v oblasti riešenia optimalizačných úloh z oblasti hydrodynamiky, práce L.J. Fogela, A.J. Owensa a M.J. Walsha [12] v oblasti riešenia predikčných úloh a práce J.H. Hollanda [18] z oblasti skúmania fenoménu adaptácie v prírode a jeho prenosu do umelých systémov. Tieto práce sa stali hybným momentom pre vznik troch samostatných smerov, ktoré sa spočiatku vyvíjali izolovane, neskôr však dochádza medzi nimi ku kontaktu a vzájomnému prelínaniu ideí. Následkom toho je vznik algoritmov spájajúcich prvky typické pre rôzne smery. Popisy základných prvkov týchto algoritmov je možné nájsť on-line napr. v prehľadovej publikácii [24].

Nedávny rozvoj evolučných algoritmov je spájaný ako s rozvojom teórie tak aj s aplikovaním týchto algoritmov na najrozličnejšie problémy. Špeciálne s ich využívaním pre riešenie úloh s ohraničeniami je spájané meno Zbigniewa

Michalewicz [26] (oblasť numerickej optimalizácie) a Agostona E. Eibena [11] (oblasť kombinatorickej optimalizácie).

Pre riešenie kombinatorických problémov s ohraničeniami je možné použiť takú podobu evolučného algoritmu, ktorá je vlastne jednou z populačných verzií algoritmu *lokálneho prehľadávania* LS (local search). Domény premenných vlastne definujú priestor prehľadávania. Je to priestor všetkých potenciálnych riešení – všetkých možných kombinácií hodnôt z domén premenných. LS generuje a skúma rozličné kombinácie priradení hodnôt z domén premenných týmto premenným. Každá takáto kombinácia je považovaná za kandidáta riešenia a cieľom je nájsť takého kandidáta, ktorý je skutočným riešením – teda priradenia premenných neporušujú žiadne z existujúcich ohraničení.

Nepopulačný LS pracuje vždy iba s jedným kandidátom. V inicializačnej fáze je vygenerovaný počiatočný kandidát, ktorý sa stáva aktuálnym kandidátom. Ďalšie prehľadávanie prebieha iteračným spôsobom. V každej iterácii je vybraný nový kandidát, pričom výber sa deje z lokálneho okolia aktuálneho kandidáta. Vybraný nový kandidát sa následne stáva aktuálnym kandidátom a algoritmus prechádza na ďalšiu iteráciu. Pri populačnej verzii je aktuálny kandidát nahradený populáciou (množinou) kandidátov a v každej iterácii je vybraná populácia nových kandidátov.

Principiálny EA algoritmus môže mať nasledovnú podobu uvedenú v Programe 3 (fragment reálneho kódu v jazyku Common Lisp).

```
(defun EA ()
  (let ((populacia (vyhodnotenie (inicializacia-kandidatov))))
    (dotimes (i *max-gen* nil)
      (setf populacia
            (vyhodnotenie (generovanie-kandidatov populacia)))
      (when (some #'najdene-riesenie-p populacia)
        (return-from EA populacia))))))

(defun generovanie-kandidatov (populacia)
  (let ((nova-populacia nil))
    (dotimes (i *velkost-populacie*)
      (push (vyber-z-okolia (zuzenie-populacie populacia))
            nova-populacia))
    nova-populacia))
```

Program 3. Fragment implementácie EA algoritmu.

Na začiatku vo funkcii EA je populácia kandidátov inicializovaná tak, že požadovaný počet kandidátov je vygenerovaný náhodným spôsobom a títo kandidáti sú následne ohodnotení. Každému kandidátovi je priradená vhodnosť – číslo udávajúce ako dobre ten-ktorý kandidát spĺňa existujúce ohraničenia. V následných iteráciách sú vytvárané nové a nové populácie kandidátov. Ak niektorá novovytvorená populácia obsahuje skutočné riešenie, tak sa iteračné hľadanie ukončí. Ak skutočné riešenie nie je objavené, tak po vykonaní predpísaného počtu iterácií je neúspech hľadania signalizovaný symbolom nil.

Generovanie novej populácie začína vytvorením prázdnej populácie, do ktorej sa postupne pridávajú noví kandidáti. Vytvorenie každého nového kandidáta sa deje

rovnakým spôsobom: najprv sa z aktuálnej populácie vyberie podmnožina kandidátov a následne sa z okolia týchto vybratých kandidátov vyberá nový kandidát.

Reprezentácia kandidátov, spôsob určovania vhodnosti kandidátov, zúženie populácie na nejakú jej podmnožinu, definícia okolia kandidátov a operátory pre výber kandidátov z tohto okolia ako aj riadiace parametre a ich nastavenie sú špecifické pre konkrétnu inštanciu evolučného algoritmu. Popísaný algoritmus vlastne reprezentuje algoritmický rámec, ktorý možno ďalej špecializovať vhodnou voľbou uvedených prvkov algoritmu.

Ilustrujme to na dvoch príkladoch. Prvým bude algoritmus založený na usporiadaní pozícií mriežky. Ak kandidáti budú reflektovať iba tie pozície v mriežke, ktoré je potrebné ešte doplniť, tak v prípade podľa Obrázku 1 každý kandidát musí zohľadniť 56 voľných pozícií. Predpokladajme, že každý kandidát bude reprezentovať jedno z možných usporiadaní týchto voľných pozícií. Príkladom takéhoto kandidáta je poradie pozícií mriežky

$$87, 17, 27, 37, 39, 19, 29, 26, \dots, 21 \quad (2)$$

kde prvá číslica v každej dvojici označuje riadok a druhá zase stĺpec, pričom celkovo toto usporiadanie obsahuje 56 rôznych dvojíc.

Určenie vhodnosti takéhoto kandidáta nie je možné priamo, ale najprv je potrebné z jeho reprezentácie skonštruovať priradenie hodnôt premenným a až následne ohodnotiť toto priradenie. Postupne sa uvažujú jednotlivé pozície podľa zadaného poradia. Pre každú voľnú pozíciu sa skúsi priradenie hodnoty 1 na danú pozíciu. Ak to nie je možné (lebo by došlo k porušeniu niektorého z binárnych ohraničení definovaných nad premennou, zodpovedajúcou skúmanej pozícii), tak sa skúsi priradenie hodnoty 2, atď. Postupne sa skúšajú jednotlivé hodnoty 1 až 9 (stále v tom istom poradí). Akonáhle niektorá zo skúšaných hodnôt môže byť priradená, tak sa toto priradenie vykoná a nasledujúce hodnoty sa už neskúšajú. Následne sa prechádza na ďalšiu pozíciu v poradí. Ak sa pre skúmanú pozíciu nenájde ani jedna vhodná hodnota, tak pozícia ostáva neobsadená a opäť sa prechádza na nasledujúcu pozíciu v poradí. Počet neobsadených pozícií bude použitý ako vhodnosť daného jedinca. To znamená, že algoritmus vlastne rieši minimalizačný problém, keď hľadané riešenie je reprezentované kandidátom transformovateľným na priradenie všetkých premenných bez nutnosti ponechávať niektoré pozície neobsadené.

Tento postup by pri použití pre hore uvedené poradie (2) konštruoval nasledovné priradenie: hodnotu 2 pre pozíciu 87 pretože hodnota 1 je už na pozícii 47; hodnotu 3 pre pozíciu 17 pretože 1 je už na 47 aj na 38 a 2 bola v predchádzajúcom kroku priradená pozícii 87; hodnotu 8 na pozíciu 27; hodnotu 5 na 37; 2 na 39 a 4 na 19. Ďalším krokom by malo byť určenie vhodnej hodnoty pre pozíciu 29, avšak keďže každá z hodnôt by spôsobila porušenie aspoň jedného ohraničenia, tak táto pozícia musí ostať prázdna. Pokračovalo by sa hodnotou 1 na 26 atď.

Vhodnosť kandidátov, zaradených v populácii, sa použije pri zužovaní populácie na dvoch kandidátov. Toto sa deje na pravdepodobnostnom základe, pričom kandidát s menším počtom neobsadených pozícií je vybraný s väčšou pravdepodobnosťou než kandidát s viacerými neobsadenými pozíciami. Je to možné realizovať náhodným výberom dvoch kandidátov z aktuálnej populácie a porovnaním ich vhodností, pričom je vybraný kandidát s menšou hodnotou.

Z okolia vybraných dvoch kandidátov sa vyberie nový kandidát a zaradí sa do novej populácie. Pre definovanie okolia sa v evolučných algoritmoch používajú operátory – do okolia patria všetci tí kandidáti, ktorých je možné vytvoriť pomocou týchto operátorov. Používajú sa dva typy operátorov – mutácia a kríženie.

Prvý z nich vytvára nových kandidátov zo štruktúry jedného kandidáta, zatiaľ čo druhý pre tvorbu nových potrebuje dvoch kandidátov na svojom vstupe. Cieľom krížacieho operátora je kombinovať štruktúrne prvky dvoch kandidátov do nových kombinácií tak, aby výsledný nový kandidát obsahoval časti pochádzajúce z oboch vstupných kandidátov. Jednou z možností je OX kríženie, ktoré chápe usporiadania, vytvárajúce kandidátov, ako cyklické – po poslednej pozícii v poradí nasleduje prvá pozícia. Nový kandidát sa buduje tak, že z jedného usporiadania sa vyberá súvislý segment, určený dvomi náhodne generovanými deliacimi bodmi. Zvyšné miesta v usporiadaní sa doplnia z druhého usporiadania tak, aby sa čo najviac zachovalo relatívne poradie v tomto druhom usporiadaní.

Pre ilustráciu okrem vyššie uvedeného usporiadania podľa (2) uvažujeme ešte druhé usporiadanie

$$11, 12, 15, 17, 19, 21, 25, \dots, 98, 99. \quad (3)$$

Nech boli náhodne generované deliace body medzi druhým a tretím prvkom usporiadania ako aj medzi šiestym a siedmym. Zo štruktúry prvého usporiadania (2) sa tak prenáša do výsledného kandidáta segment: 27, 37, 39, 19. Pred tento segment je potrebné doplniť dve pozície a zaň zase 50 pozícií z druhého usporiadania (3). Keďže poradie sa chápe cyklicky, tak druhé usporiadanie možno súčasne zapísať ako

$$25, \dots, 98, 99, 11, 12, 15, 17, 19, 21 \quad (4)$$

pričom sa začína prvkom za druhým deliacim bodom (t.j. za koncom fragmentu, preneseného z (2)). Z takto vyjadreného poradia je potrebné vypustiť tie pozície mriežky, ktoré už sú v novom kandidátovi (boli prenesené z prvého usporiadania), čím sa získa

$$25, \dots, 98, 99, 11, 12, 15, 17, 21. \quad (5)$$

Takto upraveným poradím sa cyklicky doplní nový kandidát do výsledného tvaru postupnosti pozícií

$$17, 21, 27, 37, 39, 19, 25, \dots, 98, 99, 11, 12, 15. \quad (6)$$

Cieľom mutačného operátora je dosiahnuť, aby nový kandidát obsahoval nielen prvky nachádzajúce sa vo vstupnom usporiadaní ale aj prvky, ktoré v ňom pôvodne neboli. Zámerom je malá zmena pôvodného usporiadania. Jednoduchou možnosťou malej zmeny je vzájomná výmena dvoch pozícií mriežky v usporiadaní jedinca, pričom tieto dve pozície sa vyberú náhodne. Takúto výmenu je možné niekoľkokrát zopakovať – a keďže vhodný počet výmen v rámci jedného usporiadania nie je očividný, je možné voľbu tohto počtu ponechať opäť na náhodu.

Pri vyberaní nových jedincov je možné používať krížiaci a mutačný operátor v rozličných kombináciách. Najčastejší spôsob použitia je použitie oboch operátorov na pravdepodobnostnom základe.

Druhým ilustračným príkladom algoritmu bude algoritmus založený na hodnotách vložených do mriežky. Ak kandidáti budú reflektovať všetky pozície v mriežke, ktoré v nej existujú, tak každý kandidát musí zohľadniť 81 pozícií. Predpokladajme, že

každý kandidát bude reprezentovať jedno z možných priradení hodnôt všetkým pozíciám mriežky. Príkladom takéhoto kandidáta je poradie hodnôt

$$3, 1, 8, 2, 3, 5, 9, 6, 4, \dots, 4, 2, 4 \quad (7)$$

kde prvá hodnota reprezentuje hodnotu na pozícii 11, druhá na pozícii 12, tretia na 13, ..., posledná zodpovedá hodnote umiestnenej na pozícii 99. Tieto hodnoty rešpektujú prvotné priradenie hodnôt (teda pre úlohu podľa Obrázku 1 na treťom mieste musí byť hodnota 8, na štvrtom 2, na šiestom hodnota 5, atď.), variácie sú možné iba v prípade voľných pozícií.

Určenie vhodnosti takéhoto kandidáta je teraz možné priamo, bez nutnosti vytvárať dodatočné konštrukcie. Jednoducho sa skontrolujú všetky binárne ohraničenia, či hodnoty na relevantných pozíciách splňajú alebo porušujú dané ohraničenia. Tak pri kontrole jedinca podľa (7) je zrejmé, že hodnota 3 na pozícii 11 porušuje $C_{11,15}$ a $C_{11,23}$ a hodnota 4 na 19 porušuje $C_{19,99}$. O tom, či hodnota 1 na pozícii 12 porušuje nejaké ohraničenie alebo nie, rozhodnú hodnoty, ktoré nie sú viditeľné v (7). Počet porušených binárnych ohraničení bude použitý ako vhodnosť daného jedinca. To znamená, že algoritmus opäť rieši minimalizačný problém, keď hľadané riešenie je reprezentované kandidátom s nulovým počtom porušených binárnych ohraničení.

Rovnako ako v predchádzajúcej verzii algoritmu, aj teraz sa vhodnosť kandidátov, zaradených v populácii, použije pri zužovaní populácie – tentoraz však iba na jedného kandidáta. Opäť sa toto deje na pravdepodobnostnom základe, pričom kandidát s menším počtom porušených ohraničení je vybraný s väčšou pravdepodobnosťou než kandidát s viacerými porušenými ohraničeniami (v základnej verzii všetky ohraničenia majú rovnakú váhu, avšak sú možné aj verzie s rôznymi váhami, dokonca s dynamickým určovaním váh ohraničení). Obdobne k predchádzajúcej verzii algoritmu, je to možné realizovať náhodným výberom dvoch kandidátov z aktuálnej populácie a porovnaním ich vhodností, pričom je vybraný kandidát s menšou hodnotou.

Z okolia vybraného kandidáta sa vyberie nový kandidát a zaradí sa do novej populácie. Pre definovanie okolia sa teraz použije iba jeden operátor – mutácia. Meniť sa pritom môžu iba tie hodnoty v kandidátovi, ktoré zodpovedajú hodnotám priradeným voľným pozíciám mriežky. Hodnoty definované úlohou sa nemenia. V úlohe mutačného operátora sa používa operátor $[\#, \#, \#]$, kde znak na prvej pozícii reprezentuje počet menených hodnôt v kandidátovi, druhá pozícia zase spôsob výberu hodnôt, ktoré sa majú meniť, a tretia pozícia vyber hodnôt ktoré nahradia menené hodnoty.

Počet menených hodnôt môže byť vopred pevne daný (napr. $[3, \#, \#]$) alebo jeho výber sa nechá na náhodu. Príkladom môže byť $[r(3), \#, \#]$ reprezentujúci náhodný výber hodnoty z $\{1, 2, 3\}$ s použitím uniformnej distribúcie pravdepodobnosti.

Voľba pozícií kandidáta, ktoré majú byť zmenené, má vo všeobecnosti dva základné tvary: $[\#, r, \#]$ a $[\#, b, \#]$. Prvý znamená náhodný výber s rovnakou pravdepodobnosťou pre všetky pozície kandidáta (random), zatiaľ čo druhý signalizuje použitie nejakej heuristiky (bias). Jedna z možných heuristík je určiť váhu každej pozície v mutovanom kandidátovi, pričom váha je daná ako počet binárnych ohraničení, ktoré sú porušené hodnotou na danej pozícii kandidáta. Keďže pre

mutovanie sa vyberú pozície s najväčšou váhou, vždy sa menia tie hodnoty, ktoré majú najväčší podiel na porušovaní existujúcich ohraničení.

Podobne voľba nových hodnôt, ktoré majú nahradiť menené hodnoty, má vo všeobecnosti dva základné tvary: [#, #, r] a [#, #, b]. Prvý opäť znamená náhodný výber s rovnakou pravdepodobnosťou pre všetky hodnoty z príslušnej domény (random), zatiaľ čo druhý opäť signalizuje použitie nejakej heuristiky (bias). Možnou heuristikou je váženie hodnôt podľa toho, koľko ohraničení porušujú.

Keďže evolučný algoritmus je algoritmom stochastickým, tak neexistuje záruka nájdenia riešenia aj keď toto riešenie skutočne existuje (a najmä v podmienkach obmedzeného času). Pre ľahšie problémy sa zvyčajne nájde riešenie skoro v každom behu, zatiaľ čo pre problémy obtiažnejšie je riešenie nachádzané menej často. Pri testovaní je preto každý problém riešený niekoľkokrát pri rôznych počiatočných nastaveniach generátora náhodných čísel.

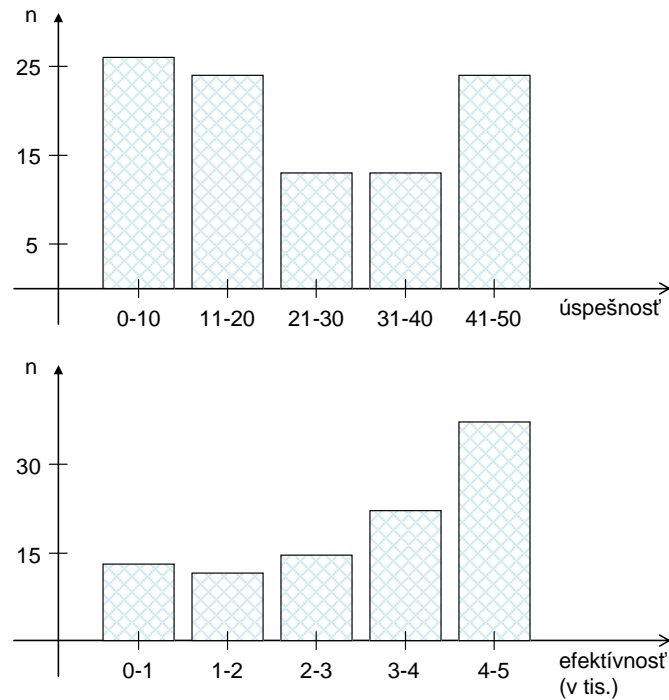
Výkonnosť algoritmu je často meraná prostredníctvom dvoch charakteristík. Miera úspešnosti algoritmu je udávaná ako počet úspešných hľadání voči počtu všetkých hľadání pri opakovanom riešení toho istého problému. Efektívnosť algoritmu sa navonok prejavuje ako rýchlosť nájdenia riešenia. Táto rýchlosť môže byť meraná napr. počtom potrebných iterácií algoritmu a určuje sa ako priemerná hodnota z viacerých realizácií.

Prezentovaný variant evolučného algoritmu, založeného na usporiadaní pozícií mriežky, bol pri testovaní použitý s populáciou skladajúcou sa zo sto kandidátov, pričom jedno hľadanie riešenia bolo obmedzené na 5000 iterácií. Riešenie každého problému bolo opakované 50 krát. Pri porovnaní výkonnosti pri riešení problémov z [16] boli dosiahnuté výsledky, ktoré sú prezentované na Obrázku 6.

Horná časť obrázku prezentuje histogram úspešnosti, ktorý je pomerne vyrovnaný. Pri riešení 24 rôznych mriežok bol algoritmus schopný nájsť riešenie vo viac než 40 pokusoch z celkového počtu 50 pokusov, pričom v siedmich prípadoch bol úspešný v každom pokuse. Na druhej strane mal problémy s 26 mriežkami, pri ktorých bol úspešný maximálne v 10 pokusoch, pričom v prípade dvoch mriežok nedokázal nájsť riešenie ani raz. Pri zohľadnení obtiažnosti jednotlivých mriežok, uvádzaných v použitom zdroji, pri riešení ľahkých úloh algoritmus dosahoval úspešnosť 100%, pri riešení stredných úloh 75%, pri ťažkých úlohách 43% a pri najťažších (zákerných) úlohách bola jeho úspešnosť iba 18%.

Dolná časť obrázku reprezentuje histogram potrebného priemerného počtu iterácií pre riešenie jednotlivých úloh (vo výpočte sa v prípade, že v niektorom behu algoritmus nedokázal vyriešiť úlohu, uvažuje maximálny počet 5000 iterácií). Je zrejmé, že prevažujú vyššie počty potrebných iterácií nad nižšími, teda častejšie sa objavovali dlhšie doby hľadania ako kratšie. V prípade 37 mriežok bol potrebný priemerný počet viac než 4000 iterácií, zatiaľ čo iba v prípade 13 mriežok bolo riešenie nájdené v priemere skôr než za tisíc iterácií.

Pri použití [21] bol algoritmus pri rovnakom nastavení menej úspešný – iba pri polovici úloh dokázal nájsť aspoň raz riešenie, avšak ani pre jednu úlohu nedokázal nájsť riešenie viac ako dva razy.



Obrázok 6. Výsledky dosiahnuté evolučným algoritmom založeným na usporiadaní pozícií mriežky.

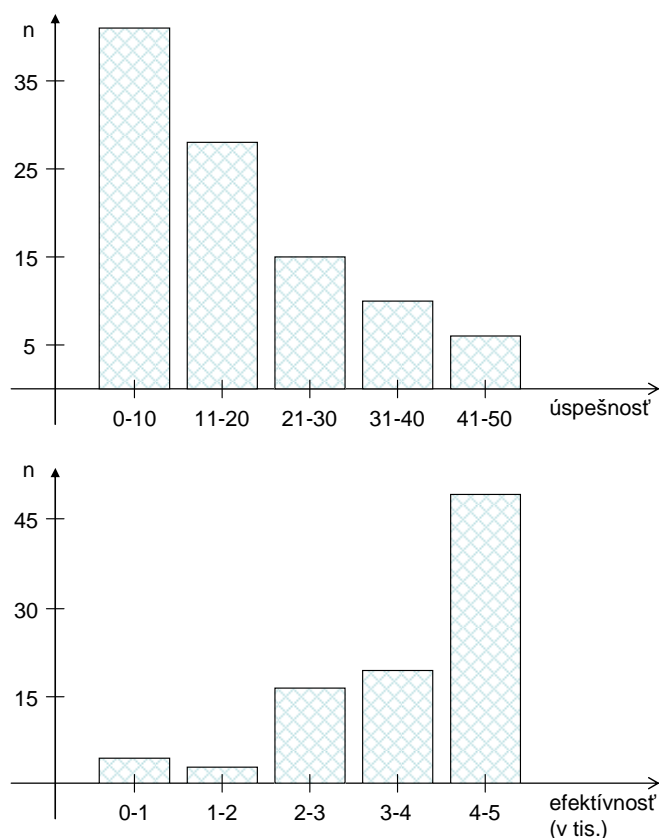
Prezentovaný variant evolučného algoritmu, založeného na hodnotách vložených do mriežky, bol pri testovaní tiež použitý s populáciou skladajúcou sa zo sto kandidátov, obmedzením riešenia na 5000 iterácií a opakovaním riešenia každého problému 50 krát. Pre testovanie bol použitý $[r(8), b, r]$ operátor, ktorý sa pri pokusoch na ľahkej inštancii hlavolamu osvedčil najviac. Pri porovnaní výkonnosti pri riešení problémov z [16] boli dosiahnuté výsledky, ktoré sú prezentované na Obrázku 7.

Histogram úspešnosti je menej vyrovnaný ako v predchádzajúcom prípade a signalizuje zhoršenie úspešnosti. Iba pri riešení 6 rôznych mriežok bol algoritmus schopný nájsť riešenie vo viac než 40 pokusoch z celkového počtu 50 pokusov, pričom v štyroch prípadoch bol úspešný v každom pokuse. Na druhej strane mal problémy až s 41 mriežkami, pri ktorých bol úspešný maximálne v 10 pokusoch, pričom v prípade jednej mriežky nedokázal nájsť riešenie ani raz. Pri riešení ľahkých úloh algoritmus dosahoval úspešnosť 100%, pri riešení stredných úloh 51%, pri ťažkých úlohách 28% a pri najťažších (zákerných) úlohách bola jeho úspešnosť iba 12%.

Histogram potrebného priemerného počtu iterácií pre riešenie jednotlivých úloh tiež odráža zhoršenie výkonnosti. Počet mriežok, kde bol potrebný priemerný počet viac než 4000 iterácií, sa zvýšil na 50, zatiaľ čo iba v prípade 8 mriežok bolo riešenie nájdené v priemere skôr než za dva tisíc iterácií.

Rovnako ako v predchádzajúcom prípade, pri použití [21] bol algoritmus pri rovnakom nastavení menej úspešný – avšak jeho výkon bol lepší ako výkon verzie

algoritmu založeného na usporiadaní pozícií mriežky. Dokázal nájsť aspoň raz riešenie pre osem mriežok, pričom pre niektoré z nich dokázal nájsť riešenie tri razy.



Obrázok 7. Výsledky dosiahnuté evolučným algoritmom založeným na hodnotách vložených do mriežky.

5 Sila znalostného prístupu

Podstatou tohto prístupu k riešeniu úloh je orientácia na konkrétne znalosti o tom, ktorom riešenom probléme. Keďže takýmto spôsobom je možné zohľadniť špecifiká riešeného problému a nielen jeho všeobecné charakteristiky, spoločné pre celú skupinu úloh nejakého typu, tak je možné očakávať vytvorenie vysoko efektívneho postupu pre nájdenie riešenia tohto problému.

Korene znalostných prístupov siahajú do druhej polovice šesťdesiatych rokov, keď sa stali základom pre tvorbu systémov označovaných ako *expertné* [29]. V počiatkovom období sa ako vedúci princíp používal *prenos znalostí* (knowledge transfer), ktorý neumožňoval diferenciáciu. Zmena nastala v osemdesiatych rokoch a bola inicializovaná prácou Allena Newella [28]. Táto zmena súvisela s prechodom k

princípu *modelovania znalostí* (knowledge modelling). Postupne začali vznikať znalostné modely rozlišujúce rozličné typy znalostí, pričom každý z týchto typov bol charakterizovaný nie spôsobom reprezentácie alebo zdrojom pôvodu, ale úlohou, ktorú hral v uvažovanej doménovej oblasti. Vznik týchto prvých modelov je spojený s takými menami ako William J. Clancey [6], Balakrishnan Chandrasekaran [19] a Luc Steels [33].

Jedným z výsledkov neskoršieho vývoja v tejto oblasti sú viacvrstvé znalostné modely. Príkladom môže byť štvorvrstvý expertízny model z KADS metodológie alebo trojvrstvý znalostný model podľa jej nasledovníka – metodológie CommonKADS [31]. Tieto modely reprezentujú prostriedok uľahčujúci ako samotné získavanie znalostí tak aj prenos týchto znalostí pre riešenie iných úloh.

Vrstvy týchto modelov je možné v podstate rozdeliť do dvoch skupín. Doménová vrstva obsahuje konkrétne znalosti týkajúce sa riešeného prípadu. Ostatné vrstvy okrem iného obsahujú napr. inferenčnú úroveň obsahujúcu znalosti o inferenciách a znalostných objektoch nad ktorými tieto inferencie sú realizované. Inferenčná vrstva tak reprezentuje akúsi šablónu uvažovania, ktorá nie je závislá na konkrétnom prípade, a teda môže byť znovu použitá aj pre iné problémy. Doménová vrstva reprezentuje použitie takejto šablóny pre riešenie konkrétneho problému. Jej úlohou je definovať obsah jednotlivých objektov, použitých na všeobecnej inferenčnej úrovni – znalostí pre realizáciu jednotlivých inferenčných krokov, znalostných štruktúr nad ktorými je realizovaná inferencia a podporných znalostí.

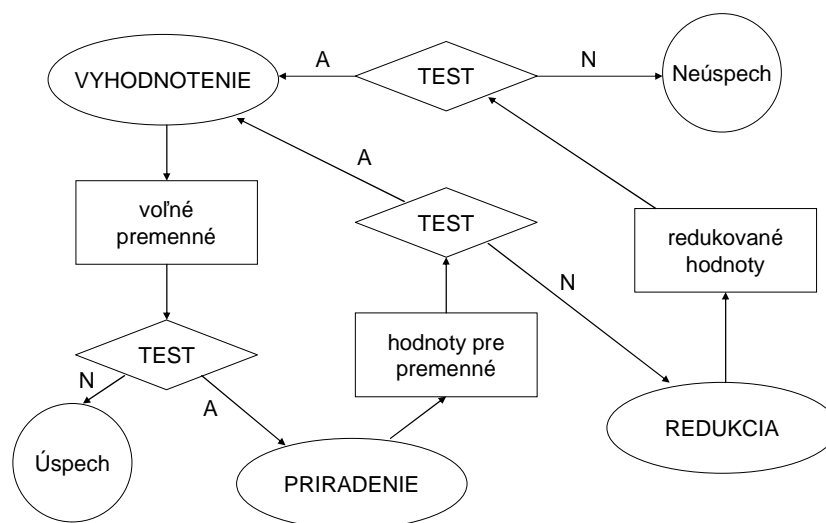
Výhodou takéhoto prístupu je, že ak sa návrhárovi podarí identifikovať typ úlohy ktorú je potrebné riešiť, potom voľba nejakého modelu pre riešenie takéhoto typu úloh mu jasne špecifikuje, ktoré znalosti z doménovej oblasti je potrebné získať.

Pri použití princípu znalostného modelovania pre riešenie úloh s ohraničeniami rolu znalostných štruktúr, nad ktorými je realizovaná inferencia, hrajú premenné a ich domény. Výsledkom nejakého inferenčného kroku môže byť napr. naviazanie premennej na nejakú hodnotu, uvoľnenie premennej, alebo zmena domény premennej. Iným výsledkom môže byť samotná premenná alebo hodnota ako výsledok selekcie. Definované ohraničenia zase môžu vystupovať v úlohe podporných znalostí. Inferenčné kroky môžu byť usporiadané v nejakej štruktúre podľa zvolenej stratégie riešenia.

Ukážkou môže byť napr. *stratégia ER* (extend and revise). Inferenčná vrstva znalostného modelu, zodpovedajúca tejto stratégii, pozostáva z niekoľkých navzájom prepojených aktivít znázornených na Obrázku 8.

Tieto aktivity operujú nad dvomi typmi znalostných štruktúr – premennými a doménami týchto premenných. Pri manipulácii s hodnotami v doménach sa určuje, ktoré hodnoty budú skutočne priradené premenným a ktoré budú vyradené z domén ako nepoužiteľné.

Stratégia sa zameriava na postupné budovanie riešenia. Na začiatku je riešenie prázdne, žiadnej premennej nie je priradená hodnota – všetky premenné sú voľné. Keďže stratégia neobsahuje žiadne opravné aktivity (nedochádza k uvoľňovaniu premenných ani k zmene ich hodnôt), premennej je možné priradiť hodnotu iba v prípade, že toto priradenie nemôže byť v následnej fáze spochybnené.



Obrázok 8. Inferenčná vrstva stratégie ER.

Inferencie navzájom prepájajú tri rôzne aktivity: *vyhodnotenie*, *priradenie* a *redukciu*. Tieto aktivity je možné charakterizovať nasledovne:

- Vyhodnotenie skúma aktuálny stav viazanosti jednotlivých premenných. Výstupom je zoznam voľných premenných, ktorým je ešte potrebné priradiť hodnotu. Činnosť je algoritmická, nevyžadujúca žiadne doménové znalosti.
- Priradenie v skutočnosti pozostáva z dvoch krokov. V prvom kroku sa vyberie voľná premenná a v druhom sa určuje hodnota, ktorú je potrebné priradiť tejto premennej. Výber voľnej premennej môže byť založený na znalostiach (umožňujúcich priamo vybrať jednu premennú alebo poskytujúcich poradie v ktorom je vhodné premenné skúmať) alebo premenné môžu byť jednoducho skúmané v preddefinovanom poradí (ktoré môže byť aj náhodné). Výber hodnoty musí byť založený na znalostiach, pretože je možné vybrať iba takú hodnotu, ktorá neporušuje žiadne ohraničenie (a to ako aktuálne tak ani v budúcnosti). Počet takýchto viazaní premenných môže byť rôzny – aktivita sa môže realizovať so snahou naviazať iba jednu z voľných premenných ale aj so snahou naviazať všetky tie voľné premenné, u ktorých to je v danom štádiu možné. Výstupom je zoznam dvojíc premenná-hodnota.
- Redukcia má za cieľ redukovať domény premenných vypúšťaním nevhodných hodnôt. Pozostáva tak isto z dvoch krokov – výberu domény, ktorá bude redukovaná, a určení hodnoty, ktorá bude z vybranej domény vypustená. Podobne ako v predchádzajúcej aktivite, výber domény môže byť založený na znalostiach alebo nepoužíva žiadne doménové znalosti. Pre určenie hodnoty, ktorá bude vyradená, sú potrebné doménové znalosti. Snahou môže byť vyradiť jednu alebo čo najviac hodnôt z jednej alebo čo najviac domén. Výstupom je zoznam dvojíc doména-hodnota.

Okrem týchto troch hlavných aktivít sú použité aj testovacie aktivity. Keďže tieto iba skúmajú, či zoznamy, vygenerované hlavnými aktivitami, sú prázdne alebo nie, nepotrebujú žiadnu znalostnú podporu.

Prepojenie aktivít vytvára dva cykly. Základným cyklom je prepojenie vyhodnotenia a priradenia. Akonáhle nie je možné vykonať žiadne priradenie, v rozšírenom cykle sa prechádza na redukciu. V prípade úspešnej redukcie sa opäť prechádza na menší cyklus. V prípade, že nie sú identifikované žiadne voľné premenné, je k dispozícii hotové riešenie. Ak sú ešte niektoré premenné voľné a súčasne nie je možné vykonať ani jedno priradenie a ani redukciu, tak nie je možné na základe dostupných znalostí zostaviť riešenie (ktoré môže alebo nemusí existovať).

Aby bolo možné nejaký znalostný model použiť, je potrebné doň vložiť konkrétne znalosti. Pre realizáciu uvedeného modelu je potrebné mať aspoň znalosti, kedy je možné premennej priradiť nejakú hodnotu a kedy je možné nejakú hodnotu vypustiť z nejakej domény.

Znalosti pre výber voľnej premennej pre priradenie hodnoty a výber domény pre redukciu je samozrejme možné nahradiť nejakou všeobecne použiteľnou metódou, napr. náhodným výberom alebo systematickým skúmaním, avšak je zrejmé, že to pravé urýchlenie riešenia sa získa práve použitím doménovo závislých heuristik.

Pre použitý ilustračný príklad na Obrázku 1 je možné využiť uvedenú inferenčnú časť modelu a doplniť ju doménovými znalosťami (uvedené znalosti sú iba ilustračné a v žiadnom prípade si nerobia nárok na úplnosť). Doplnenie znalosťami pre realizáciu priradenia:

- **P1:** ak v nejakom útvare (riadku, stĺpci alebo štvorci) na nejakej pozícii je prípustná iba jedna hodnota, tak tá hodnota sa umiestni na danú pozíciu bez ohľadu na to, či táto hodnota je prípustná aj pre inú pozíciu daného útvaru alebo nie;
- **P2:** ak v nejakom útvare (riadku, stĺpci alebo štvorci) je pre nejakú hodnotu prípustná iba jedna pozícia, tak tá hodnota sa umiestni na danú pozíciu bez ohľadu na to, či pre danú pozíciu sú prípustné aj ďalšie hodnoty alebo nie;

ako aj znalosťami pre realizáciu redukcie:

- **R1:** pre nejakú pozíciu je neprípustná taká hodnota, ktorá je už umiestnená v útvare (riadku, stĺpci alebo štvorci), do ktorého skúmaná pozícia patrí;
- **R2** (heuristika locked candidates): ak nejaká hodnota síce ešte nie je umiestnená v štvorci, ale je prípustná iba pre pozície na priesečníku tohto štvorca a jedného riadku (stĺpca), tak táto hodnota potom nie je prípustná pre tie pozície daného riadku (stĺpca), ktoré nie sú súčasťou daného štvorca. Analogicky, ak nejaká hodnota nie je ešte umiestnená v riadku (stĺpci), ale je prípustná iba pre pozície tohto riadku (stĺpca) a jedného štvorca, tak táto hodnota nie je prípustná pre tie pozície daného štvorca, ktoré nie sú súčasťou daného riadku (stĺpca);

- **R3** (heuristika naked pairs): ak v nejakom útvare (riadku, stĺpci alebo štvorci) pre dve rôzne pozície sú prípustné iba dve hodnoty, rovnaké pre obe pozície, tak tieto dve hodnoty nie sú prípustné pre ostatné pozície daného útvaru;
- **R4** (heuristika hidden pairs): ak v nejakom útvare (riadku, stĺpci alebo štvorci) dve rôzne hodnoty sú prípustné iba pre dve pozície, rovnaké pre obe hodnoty, tak pre tieto dve pozície nie sú prípustné žiadne iné hodnoty.

Keďže kombinácia P1 a R1 v uvedenom modeli je vlastne ekvivalentom zabezpečenia hranovej konzistencie, tak výsledkom ich použitia je situácia na Obrázku 3. V tejto situácii by použitie P2 zapríčinilo napr. umiestnenie hodnoty 1 na pozíciu 99, pretože aj keď táto hodnota je prípustná pre viac pozícií posledného riadku, je prípustná iba pre túto jednu pozíciu v rámci posledného stĺpca. Následne sa pozícia 77 stáva jedinou prípustnou pozíciou pre hodnotu 6 v rámci pravého dolného štvorca a preto jej môže byť táto hodnota priradená.

Rovnako je možné na Obrázku 3 ilustrovať aj použitie znalostí R2, R3 a R4. Pri použití R2 je napr. možné hodnotu 4 určiť ako neprípustnú pre pozície 49 a 59 (hodnota je v rámci pravého horného štvorca prípustná iba v priesečníku tohto štvorca a posledného stĺpca), hodnotu 3 ako neprípustnú pre pozície 82 a 92 (hodnota je v rámci prvého stĺpca lokalizovaná na priesečník tohto stĺpca a ľavého dolného štvorca) a hodnotu 7 ako neprípustnú pre pozície 64, 65 a 66 (hodnota v rámci štvrtého riadku je obmedzená na priesečník tohto riadku a stredného štvorca).

Použitie R3 by na základe toho, že v siedmom stĺpci sú hodnoty 2 a 8 jedinými prijateľnými hodnotami pre pozície 27 a 87, umožnilo identifikovať nevhodnosť hodnoty 2 pre pozície 37 a 67 ako aj nevhodnosť hodnoty 8 pre pozície 37 a 77.

Aplikácia R4 by mala za následok, že keďže v rámci štvrtého riadku sa hodnota 7 ako aj hodnota 9 môžu vyskytovať iba na pozíciách 45 a 46, tak hodnoty 4 a 6 nie sú prípustné pre pozíciu 45 a hodnoty 2, 4 a 6 zase nie sú prípustné pre pozíciu 46.

Pri porovnaní výkonnosti pri riešení problémov z [16] použitie všetkých šiestich znalostných heuristík umožnilo vyriešiť všetky úlohy zo všetkých štyroch kategórií. Použitie iba nejakej podmnožiny znalostných heuristík viedlo k menšej úspešnosti s výskytom úloh, s ktorými si model vybavený touto podmnožinou nedokázal poradiť.

Pri absencii R2 bolo vyriešených 95 úloh, pri R3 to bolo 99 úloh a pri R4 zase 97 úloh. Nepoužitie P1 viedlo na vyriešenie 93 úloh zatiaľ čo nepoužitie P2 na 83 úloh.

Na rozdiel od konzistenčných algoritmov, jednotlivé znalostné heuristiky nie je možné zoradiť podľa ich „sily“ pri riešení úloh – nedá sa povedať, ktorá znalosť je lepšia vo všeobecnosti. Často sa totiž stáva, že kombinácia heuristík umožní vyriešiť nielen tie úlohy, ktoré bolo možné vyriešiť jednou z kombinovaných heuristík, ale aj tie úlohy, s ktorými si žiadna osobitne použitá heuristika neporadila. Vzniká takto synergický efekt, ktorý poskytuje výkonnosť vyššiu ako je súčet výkonností jednotlivých zložiek.

Pri použití [21] bol model s uvedenými znalostnými heuristikami neúspešný, nedokázal vyriešiť ani jednu z úloh. Pre ich riešenie je nutné použiť niektoré z heuristík pre zložitejšie problémy [34].

6 Záver

Prirodzene vystupuje do popredia otázka, ktorý z uvedených prístupov je vhodnejší pri riešení nejakého konkrétneho prípadu. Táto otázka sa nedá zodpovedať na všeobecnej úrovni, pretože odpoveď je závislá od charakteristík riešeného problému a podmienok hľadania riešenia. Je dôležité si uvedomiť, že rozličné typy metód sú založené na rozličných princípoch a to môže spôsobovať ich rozličnú vhodnosť alebo nevhodnosť.

Kritickým faktorom pre použitie redukčných metód je množstvo propagácií, ktoré umožňujú ohraničenia. Ak podstata problému je taká, že pomerne malým množstvom úsilia je možné dosiahnuť veľké orezanie priestoru prehľadávania, tak to je vhodné použitie pre tento typ metód. Ich použiteľnosť sa však znižuje s nutnosťou zabezpečovať vyššie stupne konzistencie kvôli veľkému nárastu časových nárokov oproti prípadom, keď postačuje zabezpečenie iba nižšieho stupňa konzistencie.

Pre rozšírenie použiteľnosti sú konzistenčné metódy spájané s prehľadávacími algoritmami, keď priestor je redukovaný iba čiastočne a samotné riešenie je vyhľadávané prehľadávaním tohto redukovaného priestoru nejakým algoritmom, ktorého základ tvorí prehľadávanie s možnosťou navracania.

Doménou evolučných algoritmov sú najmä problémy, ktorých rozmer neumožňuje ich riešenie redukčnými a deterministickými prehľadávacími metódami. Dosiahnuté výsledky pritom favorizujú tie varianty evolučných algoritmov, ktoré sa v nejakej miere vzdali svojej univerzálnosti zohľadňovaním charakteru ohraničení a štruktúry vzájomnej závislosti ohraničení. Dobré výsledky boli dosiahnuté aj použitím adaptívnych verzií, ktoré sa v určitej miere dokázali prispôsobiť konkrétnemu riešenému problému.

Pri použití znalostného princípu narastajú časové nároky na formulovanie problému, pretože súčasťou tejto formulácie je aj proces získavania potrebných znalostí, nutných pre vykonávanie jednotlivých krokov znalostného modelu. Pri dostupnosti vhodných znalostí je však možné riešiť úlohy veľmi efektívne, pretože je možné k jednotlivým ohraničeniam (a ich porušeniam) pristupovať individuálnym spôsobom so zohľadnením ich podstaty. V prípade dostupnosti znalostí tento prístup umožňuje riešiť aj veľmi zložité problémy, neriešiteľné alebo iba obtiažne riešiteľné inými metódami.

Literatúra

- [1] Bacchus, F., Van Beek, P.: On the conversion between non-binary and binary constraint satisfaction problems. In *Proc. of the 15th National Conference on Artificial Intelligence AAAI-98*, Madison, Wisconsin, 1998, pp. 311-318.
- [2] Berlandier, P.: A symbiotic approach to arc and path consistency checking. In Pinto-Ferreira, C. – Mamede, N.J. (eds.): *Progress in Artificial Intelligence (LNAI 990)*. Springer Verlag, Heidelberg, 1995, pp. 107-114.
- [3] Bessiere, Ch.: Arc-consistency and arc-consistency again. *Artificial Intelligence* **65** (1994) 179-190.

-
- [4] Bessiere, Ch., Regin, J-Ch.: Using bidirectionality to speed up arc-consistency processing. In Meyer, M. (ed.): *Constraint Processing* (LNCS 923). Springer Verlag, Heidelberg, 1995, pp. 157-169.
- [5] Borning, A.: The programming language aspects of ThingLab, a constraint oriented simulation laboratory. *ACM Transactions on Programming Languages and Systems* **3** (1981) 252-387.
- [6] Clancey, W.J.: Heuristic classification. *Artificial Intelligence* **27** (1985) 289-350.
- [7] Debruyne, R.: A property of path inverse consistency leading to an optimal PIC algorithm. In *Proc. of the European Conference on Artificial Intelligence ECAI-00*, Berlin, 2000, pp. 88-92.
- [8] Debruyne, R., Bessiere, Ch.: Some practicable filtering techniques for the constraint satisfaction problem. In *Proc. of the 15th Int. Joint Conference on Artificial Intelligence IJCAI'97*, Nagoya, Japan, 1997, pp. 412-417.
- [9] Debruyne, R., Bessiere, Ch.: From restricted path consistency to max-restricted path consistency. In *Proc. of Third Int. Conference on Principles and Practice of Constraint Programming*, Linz, 1997, pp. 312-326.
- [10] Dechter, R.: *Constraint processing*. San Francisco, CA: Morgan Kaufmann 2003.
- [11] Eiben, A. E. et al.: Graph coloring with adaptive evolutionary algorithms. *Journal of Heuristics* **4** (1998) 25-46.
- [12] Fogel, L.J., Owens, A.J., Walsh, M.J.: *Artificial intelligence through simulated evolution*. New York: John Wiley 1966.
- [13] Freuder, E.C.: Synthesizing constraint expressions. *Communications of the ACM* **21** (1978) 958-966.
- [14] Freuder, E.C.: Using metalevel constraint knowledge to reduce constraint checking. In Meyer, M. (ed.): *Constraint Processing* (LNCS 923). Springer Verlag, Heidelberg, 1995, pp. 171-183.
- [15] Guesgen, H-W., Hertzberg, J.: Some fundamental properties of local constraint propagation. *Artificial Intelligence* **36** (1988) 237-247.
- [16] Gould, W.: *Pravda Sudoku kniha 1*. Bratislava: Denník Pravda 2005.
- [17] Haralick, R.M., Shapiro, L.G.: The consistent labeling problem: part I. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **1** (1979) 173-184.
- [18] Holland, J.H.: *Adaptation in natural and artificial systems*. Ann Arbor: University of Michigan Press 1975.
- [19] Chandrasekaran, B.: Design problem solving: a task analysis. *The AI Magazine* **11** (1990) 59-71.
- [20] Chen, Y.: Arc consistency revisited. *Information Processing Letters* **70** (1999) 175-184.
- [21] Inkala, A.: *AI Sudoku Top 10*, 2006, on-line: http://www.aisudoku.com/index_en.html
- [22] Kumar, V.: Algorithms for constraint satisfaction problems: a survey. *The AI Magazine* **13** (1992) 32-44.

-
- [23] Mackworth, A.K.: Consistency in networks of relations. *Artificial Intelligence* **8** (1977) 99-118.
- [24] Mach, M.: *Evolučné algoritmy: prvky a princípy*. Košice: Elfa 2009, on-line: <http://neuron.tuke.sk/~machm/book-eapp-sk.html>
- [25] Mach, M., Paralič, J.: *Úlohy s ohraničeniami: Od teórie k programovaniu*. Košice: Elfa 2000, on-line: <http://neuron.tuke.sk/~machm/book-uo-sk.html>
- [26] Michalewicz, Z.: Genetic algorithms + data structures = evolution programs, 3rd edition. New York: Springer Verlag 1996.
- [27] Mohr, R., Henderson, T.C.: Arc and path consistency revisited. *Artificial Intelligence* **28** (1986) 225-233.
- [28] Newell, A.: The knowledge level. *Artificial Intelligence* **18** (1982) 87-127.
- [29] Popper, M., Kelemen, J.: *Expertné systémy*. Bratislava: Alfa 1989.
- [30] Rechenberg, I.: *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Stuttgart: Frommann-Holzboog 1973.
- [31] Schreiber, G. et al.: Knowledge engineering and management. The CommonKADS methodology. Cambridge, MIT Press 2000.
- [32] Schwefel, H.P.: *Kybernetische Evolution als Strategie der experimentellen Forschung in der Stroemungstechnik*. Diplomová práca, Technická univerzita, Berlín, 1965.
- [33] Steels, L.: Components of expertise. *The AI Magazine* **11** (1990) 29-49.
- [34] SuDoKu-Cracker – Levels of difficulty, on-line: <http://www.the24net.com/en/sudoku/help/?id=levels>
- [35] Sutherland, I.: Sketchpad: a man-machine graphical communication system. In *Proc. of the IFIP Spring Joint Computer Conference*, 1963, pp. 329-345.
- [36] Tsang, E.: *Foundations of constraint satisfaction*. San Diego: Academic Press 1993, on-line: <http://www.brasil.net/edward/fcs.html>
- [37] Van Hentenryck, P., Deville, Y., Teng, Ch.: A generic arc-consistency algorithm and its specializations. *Artificial Intelligence* **57** (1992) 291-321.
- [38] Waltz, D.: Understanding line drawings of scenes with shadows. In Winston, P.H. (ed.): *The psychology of computer vision*. McGraw Hill, New York, 1975, pp. 19-91.
- [39] Weigel, R., Blik, Ch., Faltings, B.V.: On reformulation of constraint satisfaction problems. In *Proc. of the 13th European Conference on Artificial Intelligence*, Brighton, UK, 1998, pp. 254-258.

Dynamika sietí

Mária MARKOŠOVÁ¹

Abstrakt. Kapitola pojednáva o teórii takých sietí, v ktorých sa počet uzlov s časom mení. Nazývame ich dynamickými sieťami. Vlastnosti a štruktúra dynamických sietí sa podobajú sieťam, ktoré vytvárajú prirodzené samoorganizované procesy (sociálne siete, www, internet), preto modelovanie dynamiky sietí je dôležité nielen z teoretického, ale aj z praktického hľadiska. V tejto práci popisujem dynamiku základných modelov rastúcich sietí a na základe matematickej analýzy jednotlivých modelov poukazujem na fakt, že konečná štruktúra siete veľmi závisí od spôsobu, akým sa sieť v čase mení. S dynamickými procesmi v sieti súvisí aj hierarchické usporiadanie uzlov. V ďalších častiach prezentujem modely, ktoré objasňujú ako môže teória sietí prispieť k pochopeniu štruktúry sietí známych z praxe. Poukážem tiež na to, čo majú mnohé reálne siete spoločné. V závere konkrétnejšie popíšem rôzne aplikácie teórie sietí a perspektívy ďalšieho výskumu.

1 Úvodné poznámky

Informatik sa veľmi často stretáva so sieťami tak v teórii, ako aj v praxi. Určite vyhľadáva informácie na internetovej sieti, alebo posiela maily. Možno má aj vlastnú www stránku, na ktorej prezentuje svoju prácu. Na nej môže napríklad umiestniť odkazy na iné, preňho alebo pre jeho spolupracovníkov, zaujímavé stránky. Aj mnohé informatické teoretické problémy súvisia so sieťami. Patrí k nim teória šírenia signálov po sieťach, kde sa zaujíname, ako musia byť rozmiestnené prijímače, vysielače alebo zosilňovače signálu, aby to na jednej strane nebolo neekonomické, ale aby sme zároveň zachovali dostatočnú kvalitu signálu. Môže nás tiež zaujímať odolnosť internetovej, alebo aj iných sietí, voči náhodným poruchám, či cieľovým útokom. Možno napríklad potrebujeme vedieť, aká štruktúra siete je voči poruchám a útokom najodolnejšia.

So sieťami sa však nestretávajú len informatici. Inžinieri spravujú cestné siete, elektrické siete, či siete dopravných komunikácií a tiež ich potrebujú optimalizovať tak, aby plnili svoju úlohu, ale zároveň neboli predimenzované. Cestná sieť má byť dostatočne hustá, aby pokryla potreby dopravy, ale na druhej strane by pre krajinu nemala byť ekologickou a ekonomickou záťažou. Telefónna sieť a mobilná sieť má

¹ Katedra aplikovanej informatiky, Fakulta matematiky, fyziky a informatiky, Univerzita Komenského, Mlynská dolina, 824 48 Bratislava, E-mail: markosova@ii.fmph.uniba.sk

umožňovať komunikáciu medzi ľuďmi, má byť aj dostatočne stabilná a spoľahlivá, ale zároveň nie príliš drahá.

Vidíme, že celý civilizovaný svet je nejakým spôsobom „zosieťovaný“. Preto sa v posledných desiatich rokoch začali o siete zaujímať aj vedci z iných, od informatiky vzdialených, vedných odborov [1, 2, 3, 4]. Biológovia aj informatici už niekoľko desaťročí skúmajú neurónovú sieť mozgu [4]. Donedávna nebolo veľa možností skúmať, do akej miery procesy myslenia súvisia s tým, ako sú neuróny pospájané. Nevedelo sa ako sieťová štruktúra mozgu ovplyvňuje jeho schopnosti. Neboli na to nástroje a ani vhodná teória. S rozvojom teórie dynamických sietí sa množili pokusy študovať mozog novými metódami. Pomohli tomu aj také neinvazívne techniky ako je *funkčná magnetická rezonancia* (fMRI) [5]. Skúmajú sa aj iné biologické siete, ako napríklad interakčné siete proteínov [6] a tiež, v spolupráci s psychológmi a kognitívnymi vedcami, aj sociálne siete a jazykové siete [2, 7, 8]. Pozornosť, nielen z teoretického, ale aj praktického hľadiska, pútajú ad hoc siete. To sú také, ktoré sa vytvárajú v ohraničenom čase, a potom zanikajú. Patria k nim napríklad funkčné siete mozgu [3]. Ich výskum je priamo závislý na zdokonaľovaní zachytávania signálov mozgu pomocou funkčnej magnetickej rezonancie [5]. Signály z mozgu získané touto technikou ukazujú, že ak mozog koná nejakú kognitívnu úlohu, niektoré oblasti mozgovej kôry pracujú synchronne, aj keď možno nie sú priamo prepojené pomocou neurónov. Tieto tvoria takzvanú funkčnú sieť, ktorá má, ako sa ukazuje, vlastnosti podobné vlastnostiam iných prirodzenými procesmi vznikajúcich sietí [9, 10, 33].

Keďže matematickou reprezentáciou siete je graf, človek, ktorý sa chce vážne venovať teórii dynamických sietí, musí sa nutne zaoberať s teóriou grafov [11]. Na druhej strane sieť, do ktorej pribúdajú alebo z nej ubúdajú uzly, je vlastne dynamickým systémom. Dynamické systémy študuje fyzika [12]. Matematicky ich možno popísať pomocou diferenciálnych, diferenciálnych, alebo dokonca integro – diferenciálnych rovníc. Preto ak chceme robiť v teórii dynamických sietí seriózny výskum, musíme sa venovať aj spomenutým oblastiam matematiky a fyziky.

Mnoho praktických problémov možno vyriešiť pomocou modelov rastúcich sietí, ktoré sú vlastne integro – diferenciálnymi rovnicami. Toto zjednodušenie, totiž že uzly do siete hlavne pribúdajú a počet uzlov, ktoré sa zo siete strácajú, je voči nim zanedbateľný, je z hľadiska praxe opodstatnené. Preto sa v tejto časti knihy budeme venovať hlavne rastúcim dynamickým sieťam. Rastúcou sieťou je napríklad internet – o tom zrejme nikoho netreba osobitne presviedčať. Neustále do nej pribúdajú noví účastníci a ich počet ďaleko prevyšuje počet tých, ktorí sa od internetu odpoja. Ďalšie zjednodušenie, ktoré urobíme, je takéto: Predstavme si, že ak náhodne vyberieme dva uzly siete, hrana medzi nimi buď existuje, alebo neexistuje. Tieto siete nazývame binárnymi. Aj keď hovorím o zjednodušení, dúfam, že vás presvedčím, že aj výskum binárnych sietí prináša zaujímavé problémy, ale aj otázky a praktické poznatky.

V druhej časti práce sa budem venovať základným vlastnostiam grafov. V tretej zasa popíšem reálne siete a ich vlastnosti a vo štvrtej časti na príklade základných modelov ukážem, ako vlastnosti sietí súvisia s ich dynamikou, teda so spôsobom rastu tej ktorej siete. Budeme sa zaoberať modelom s náhodným a preferenčným pripájaním uzlov a tiež modelom sietí so zrýchleným rastom. Podkapitola päť je venovaná niektorým lokálnym procesom, ktoré ovplyvňujú vlastnosti a štruktúru sietí

a hierarchické usporiadanie uzlov. V šiestej podkapitole pojednám o aplikáciách teórie sietí. Siedma časť poukáže na to, aké sú perspektívy a možnosti ďalšieho výskumu.

2 Krátky pohľad do teórie grafov

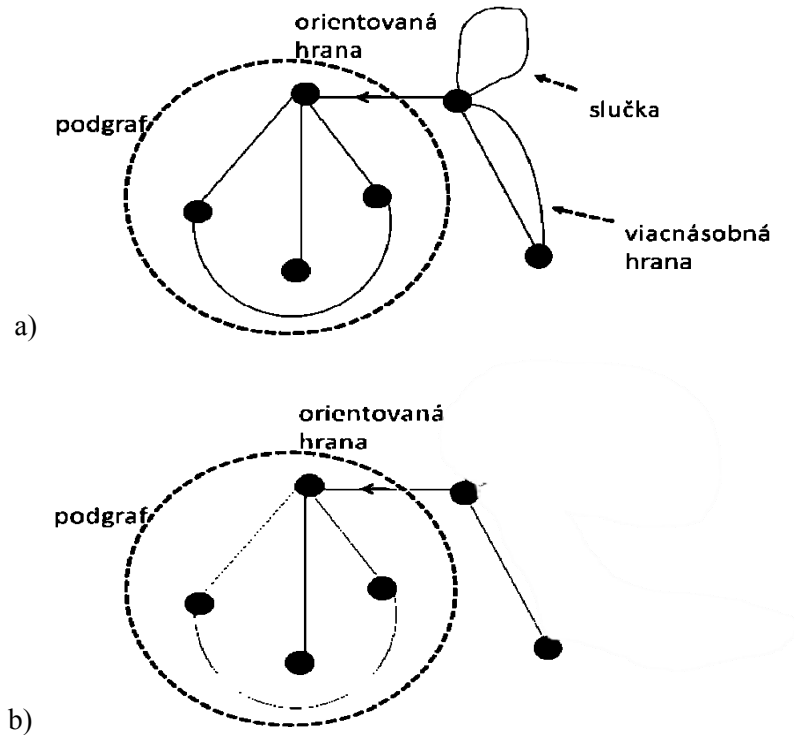
V tejto časti krátko nahliadneme do matematickej teórie grafov. Zavediem v nej niekoľko nevyhnutných základných pojmov a vzťahov. Teória grafov je v súčasnosti veľmi rozvinutou oblasťou matematiky, a preto toto pojednanie v žiadnom prípade nemôže byť vyčerpávajúce. Tých, ktorí by potrebovali obsiahlejšie informácie, odkazujem na bohatú literatúru z teórie grafov, napríklad na výbornú knihu *Introduction to Graph Theory* od D. B. Westa [11].

Graf G je definovaný pomocou dvoch množín; množiny vrcholov $V(G)$ a množiny hrán $E(G)$. Pokiaľ nebude povedané inak, budeme tieto množiny považovať za konečné. Prvkami množiny $V(G)$ sú vrcholy grafu (resp. identifikátory vrcholov grafu), prvkami množiny $E(G)$ sú usporiadané dvojice. Ak je hrana grafu orientovaná, potom prvým členom dvojice je identifikátor vrcholu, z ktorého hrana vychádza, druhým členom identifikátor vrcholu, do ktorého hrana vchádza. Takéto hrany s fixným poradím vrcholov v usporiadanej dvojici nazývame *orientované*. Ak neexistuje na hrane preferovaný smer, na poradí vrcholov v usporiadanej dvojici nezáleží. Vrcholy zvykneme nazývať aj uzlami grafu.

Na obrázku 1a vidíme graf. Má jednu *slučku*, teda hranu, ktorá začína aj končí na tom istom uzle. Ak dva uzly spája viacero hrán, hovoríme o *viacnásobnej hrane*. Prerušovanou kružnicou je označený jeden z *podgrafov* nášho grafu. Jedna z hrán grafu na obrázku 1 je *orientovaná*, jej orientácia je označená šípkou. V posledných piatich rokoch nastal zvýšený záujem o vývoj a aplikáciu vnorených výpočtových systémov do mnohých oblastí života. Najväčšiu aplikačnú oblasť vnorených systémov predstavujú procesy ovládania a riadenia tak v spotrebnej elektronike, ako aj v priemyselných procesoch. Vnorené systémy riadenia využívajú pre svoju činnosť algoritmy založené na konvenčných a inteligentných technikách modelovania a riadenia. Nové inteligentné metódy modelovania a riadenia sú dnes implementované vo forme algoritmov do všadeprítomných výpočtových systémov. Inteligentné metódy predstavujú v oblasti modernizácie HW a SW vnorených systémov evolučný smer, umožňujúci podstatným spôsobom zvýšiť kvalitu riadenia mnohých technologických a výrobných odvetví v priemysle, procesov v zdravotníctve, biotechnológiách, genetickom inžinierstve, doprave a v službách.

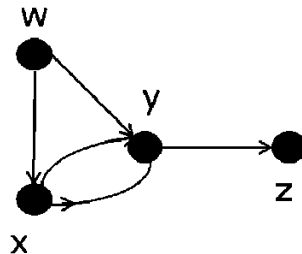
Jedným z hlavných cieľov tejto časti príspevku je poukázať na vývoj, algoritmizáciu a aplikáciu inteligentných metód modelovania a riadenia procesov s využitím moderných evolučných metód, založených na fuzzy množinách, umelých neurónových sieťach a genetických algoritmoch vo vnorených systémoch.

Pod pojmom *jednoduchý graf* rozumieme taký graf, ktorý nemá viacnásobné hrany ani žiadne slučky. Príklad jednoduchého grafu je na obrázku 1b. Vznikol z grafu na obrázku 1a vymazaním násobných hrán a slučiek.



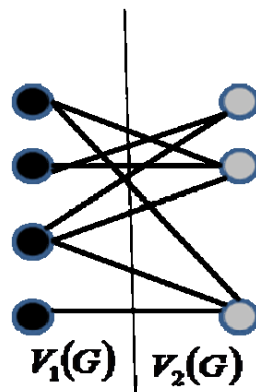
Obr. 1. Graf a jeho prvky. a) Graf so slučkou a násobnou hranou. b) Jednoduchý graf.

Na obrázku 2 vidíme ďalší príklad *orientovaného* grafu, kde všetky hrany majú preferovaný smer. Každému uzlu možno priradiť jeho stupeň. *Stupeň uzla x* , k_x , predstavuje súčet počtu hrán, ktoré do uzla vchádzajú alebo z neho vychádzajú. Takéto hrany nazývame *incidentnými* s vrcholom x . V orientovanom grafe rozoznávame počet hrán k_x^{in} , ktoré do uzla x vchádzajú a počet hrán k_x^{out} , ktoré z uzla x vychádzajú. Potom stupeň uzla x , k_x , je daný súčtom $k_x = k_x^{in} + k_x^{out}$. V ďalšom texte, pokiaľ nebude povedané inak, uvažujeme len neorientované grafy.



Obr. 2. Orientovaný graf. Uzol x má stupeň $k_x^{in} = 1$, $k_x^{out} = 2$, a teda $k_x = 3$.

Pod *bipartitným grafom* rozumieme taký graf, ktorého množinu uzlov $V(G)$ môžeme rozdeliť na dve dizjunktné podmnožiny (partície) $V_1(G)$, $V_2(G)$ kde $V(G) = V_1(G) \cup V_2(G)$ je zjednotením oboch množín. Hrany existujú len medzi uzlami z rôznych partícií (obrázok 3). Ako príklad môže slúžiť graf, v ktorom v jednej partícii sú ľudia a v druhej partícii ich zamestnania. Hrana existuje vtedy, ak človek mal v živote dané zamestnanie.



Obr. 3. Bipartitný graf. Hrany medzi uzlami existujú len vtedy, ak uzly patria do dvoch rôznych podmnožín množiny vrcholov $V(G)$.

Pod pojmom *kompletný graf*, rozumieme taký neorientovaný jednoduchý graf, kde sú uzly poprepájané hranami spôsobom každý s každým. Dva jednoduché neorientované grafy nazývame *komplementárnymi*, ak majú rovnaký počet uzlov a po stotožnení uzlov ich hrany vytvoria kompletný graf. Pod *komplementom* grafu G rozumieme teda taký graf G' , ktorý má rovnaký počet uzlov ako graf G a je komplementárny ku G .

Matematicky možno grafy reprezentovať rôznymi spôsobmi, napríklad pomocou *matice susednosti*, alebo pomocou *incidenčnej matice*. Riadky aj stĺpce matice susednosti označujú vrcholy a prvky matice susednosti reprezentujú počty hrán medzi jednotlivými dvojicami vrcholov. Riadky incidenčnej matice označujú uzly a stĺpce hrany. Prvok matice má hodnotu 1 ak je daná hrana incidentná s daným uzlom (teda v prípade orientovaných grafov z neho vychádza, alebo doňho vchádza) a majú hodnotu 0 v opačnom prípade. Matica susednosti a incidenčná matica pre graf na obrázku 2 je:

$$\begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 2 & 0 \\ 1 & 2 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} - \text{matica susednosti} \quad \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} - \text{incidenčná matica}$$

Samozrejme, sú možné aj iné reprezentácie, ktoré vyzdvihujú ďalšie vlastnosti grafov. Tak napríklad, ak odhladneme od počtu hrán medzi dvoma uzlami v matici susednosti a upravíme prvky matice tak, aby boli rovné jednej ak existuje hrana medzi dvoma uzlami a nule ak nie, dostaneme binárnu maticu susednosti. Ak od matice susednosti odčítame diagonálnu maticu, ktorej diagonálne prvky obsahujú stupeň príslušného uzla, získame Laplaceovu maticu. Jej riadky a stĺpce sa sumujú na nulu a obsahuje informácie jednak o tom, ako sú rozložené hrany v grafe, jednak o tom, ako dobre sú zapojené jednotlivé vrcholy. Ak sa chceme niečo dozvedieť o štruktúre najkratších vzdialeností medzi uzlami, môžeme graf reprezentovať pomocou matice vzdialeností. Riadky a stĺpce tejto matice sú vrcholy a prvky matice predstavujú najmenší počet hrán, ktorý treba prejsť, aby sme sa od jedného vrcholu dostali k inému. Ak graf nie je orientovaný, táto matica je symetrická.

Dva neorientované grafy G a H sú *izomorfné*, ak dokážeme nájsť také vzájomne jednoznačné zobrazenie, ktoré zobrazí všetky uzly grafu G na uzly grafu H a všetky hrany grafu G na hrany grafu H . Platí, že hrana medzi dvoma vrcholmi v grafe G existuje vtedy a len vtedy, keď existuje aj medzi ich obrazmi v H . Matice susednosti oboch grafov sú rovnaké, ak upravíme číslovanie vrcholov. Graf sa nazýva *samokomplementárnym*, ak je izomorfný so svojím komplementom. Grafy môžeme zaraďovať do izomorfných tried, ktorými sú napríklad cesty na n uzloch, cykly na n uzloch a podobne. Počet jednoduchých grafov, ktoré môžeme vytvoriť na množine n

rozlíšených uzlov je daný vzťahom $m = 2^{\binom{n}{2}}$.

Od jedného uzlu grafu k inému sa pohybujeme po hranách. Ak majú orientáciu, pohyb je možný len v jej smere, ak nie, môžeme sa pohybovať oboma smermi. *Sled* je taký pohyb po grafe, ktorý môže opakovať hrany aj vrcholy, zatiaľ čo pri *ťahu* sa hrany opakovať nesmú. *Eulerovský ťah* prejde cez všetky hrany práve raz. Ak graf má *cestu*, potom sa musíme po hranách pohybovať tak, aby sa ani vrcholy ani hrany neopakovali. Neorientovaný graf je *súvislý*, ak existuje cesta medzi každou dvojicou rozdielnych vrcholov. Známa Eulerova veta tvrdí, že súvislý graf má uzavretý eulerovský ťah (t.j. prvý a posledný vrchol ťahu je rovnaký) práve vtedy, keď všetky vrcholy sú párneho stupňa. Je tomu tak preto, lebo ak ťah neopakuje hrany, jednou hranou do vrcholu vchádzame a druhou z neho vychádzame. *Maximálny súvislý komponent* grafu G je taký podgraf, ktorý je súvislý a nie je obsiahnutý v žiadnom inom súvislom podgrafe grafu G .

Stupeň uzla x , k_x , je jednou zo základných veličín, ktoré v grafe meriame. Neorientovaný graf G je *k -regulárny*, ak stupne všetkých jeho uzlov sú rovnaké a majú hodnotu k . Keď sčítame stupne všetkých uzlov grafu, zistíme, že

$$\sum_{i=1}^N k_i = 2e \quad (1)$$

V rovnici (1) N je počet uzlov v grafe G a e označuje počet hrán. Túto hodnotu dostaneme napríklad aj z Laplaceovej matice grafu, keď sčítame jej záporne vzaté diagonálne prvky. Často sa počíta *priemerný stupeň uzla* v grafe:

$$\langle k \rangle = \frac{1}{N} \sum_{i=1}^N k_i \quad (2)$$

Pod najkratšou cestou medzi dvoma uzlami grafu myslíme tú, ktorá prechádza najmenším počtom hrán. O tom, ako majú k sebe uzly „ďaleko“ vypovedá miera nazývaná *priemernou vzdialenosťou*, alebo *separáciou uzlov* v grafe. Pre veľké grafy sa separácia uzlov počíta ako priemerná hodnota najkratších vzdialeností pre náhodne volené dvojice uzlov.

O tom, ako dobre je prepojené okolie uzla x , vypovedá jeho *klasterizačný koeficient* C_x , ktorý je pomerom počtu skutočne existujúcich hrán medzi susedmi uzla x a počtu hrán, ktoré by medzi susedmi boli, keby susedstvo uzla x tvorilo kompletný graf:

$$C_x = \frac{e_x}{\binom{k_x}{2}} \quad (3)$$

V rovnici (3) e_x je počet skutočne existujúcich hrán medzi susedmi uzla x a k_x je stupeň uzla x , a teda aj počet jeho susedov. Platí tiež, že $\binom{k_x}{2} = \frac{(k_x - 1)k_x}{2}$.

Klasterizačný koeficient C celého grafu G je priemernou hodnotou klasterizačných koeficientov všetkých uzlov

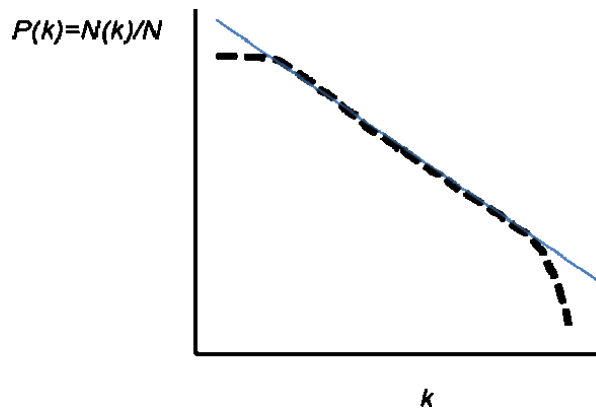
$$C = \frac{1}{N} \sum_{i=1}^N C_i \quad (4)$$

kde N je celkový počet uzlov grafu a C_i je klasterizačný koeficient uzla i daný vzťahom (3).

Ak chceme štruktúru grafu popísať presnejšie, nevystačíme len s priemernými hodnotami ako sú klasterizačný koeficient C (4) a priemerný stupeň uzla $\langle k \rangle$ (2). Viacej nám o nej povie *rozdelenie stupňa uzlov* a *rozdelenie klasterizačných koeficientov*. Rozdelenie stupňa uzlov je funkcia, ktorá opisuje, ako sa normovaný počet uzlov majúcich stupeň k , mení s hodnotou k . Môže napríklad vyzeráť tak, ako na obrázku 4, alebo napríklad aj tak, ako na obrázku 9. Podobne rozdelenie klasterizačných koeficientov je funkcia, ktorá opisuje, ako sa priemerný klasterizačný koeficient C_k uzlov majúcich stupeň k mení s veličinou k . Neskôr uvidíme, že obe tieto distribúcie veľmi úzko súvisia s dynamikou a štruktúrou siete.

Dôležitým pojmom v teórii grafov je *náhodný graf*. Prielom do teórie náhodných grafov urobili Erdos a Rényi a svoje výsledky zverejnili v známej práci On random graphs [13]. Erdos a Rényi v práci definujú dva modely náhodných grafov. Nazvime ich model A a model B.

V modeli A máme daný počet uzlov N a definovanú pravdepodobnosť p , že zvolená dvojica uzlov je spojená hranou. Každý graf o e hranách sa vyskytuje s pravdepodobnosťou $G_p = p^e (1-p)^{\binom{N}{2}-e}$. Náhodný graf typu B má určený počet uzlov N a počet hrán e , ktorými sú uzly náhodne poprepájané. Oba modely majú v limite veľkého počtu uzlov rovnaké vlastnosti, ak pravdepodobnosť $p = e / \binom{N}{2}$.



Obr. 4. Rozdelenie stupňa uzlov v grafe. Na osi x je stupeň uzla a na osi y počet uzlov, ktoré majú stupeň k normovaný na celkový počet uzlov grafu. Zobrazujeme v dvojitej logaritmickej škále, teda v lineárnej časti sa distribúcia správa podľa vzťahu $P(k) \propto k^{-\gamma}$.

Podľa Erdosa a Rényiho [13] takmer každý graf má nejakú vlastnosť Q , ak pravdepodobnosť mať túto vlastnosť sa blíži k jednej s rastúcim počtom uzlov grafu.

Nie je účelom tejto práce rozoberať vlastnosti náhodných grafov. Zaujímavosť odkazujem na ďalšiu literatúru [11, 13]. Pre naše potreby je dôležité vedieť, že pre značný rozsah hodnôt pravdepodobnosti p distribúcia stupňa uzlov náhodného grafu má tvar funkcie s peakom (Poissonovo rozdelenie) (viď. obrázok 9). Znamená to, že v náhodných grafoch máme mnoho uzlov majúcich stupeň v úzkom okolí maxima distribučnej funkcie a pravdepodobnosť mať uzol s väčším alebo menším stupňom so vzdialenosťou od tohto typického stupňa, definovaného maximom rozdelenia, rýchlo klesá. Klasterizačný koeficient v náhodnom grafe je

$$C_{rand} \approx \frac{\langle k \rangle}{N} \quad (5)$$

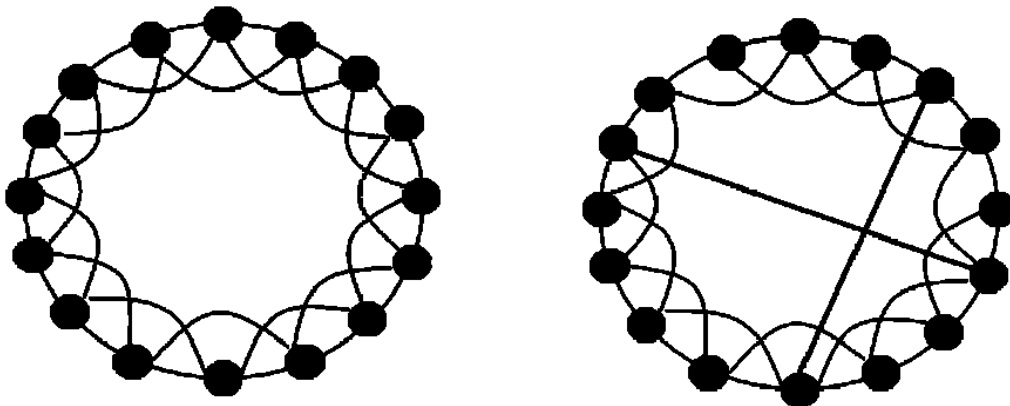
ak počet uzlov N je veľký. V takom prípade priemernú vzdialenosť môžeme vypočítať ako

$$L_{rand} \approx \frac{\ln(N)}{\ln(\langle k \rangle)} \quad (6)$$

V rovniciach (5) a (6) N je počet uzlov a $\langle k \rangle$ je priemerný stupeň uzla grafu. Z (5) vidíme, že pre veľký počet uzlov sa klasterizačný koeficient v náhodnom grafe blíži k nule. Ďalej z rovnice (6) možno usúdiť, že separácia vrcholov rastie s počtom vrcholov grafu len pomaly, logaritmicky. Náhodný graf je teda taký graf, v ktorom majú uzly k sebe pomerne blízko.

Nie všetky grafy však majú také isté vlastnosti ako náhodné grafy. Ak graf vzniká procesom, ktorý navrhli Watts a Newman [14], potom sa jeho klasterizačný koeficient nebude s rastúcim počtom uzlov blížiť k nule, aj keď priemerná vzdialenosť medzi uzlami bude rásť s počtom uzlov logaritmicky, tak ako v náhodných grafoch. Takéto siete nazývame *sietami malého sveta*. Charakterizuje ich pomerne veľký klasterizačný koeficient kombinovaný s malou separáciou uzlov. Akú majú siete malého sveta štruktúru, ukážeme na Newmanovom a Wattsovom modeli [1, 14] (obrázok 5).

Predstavme si, že máme retiazku vrcholov, pozostávajúcu z veľkého množstva vrcholov, $N \gg 1$. Každý vrchol je hranami priamo spojený so štyrmi najbližšími susedmi (obrázok 5a). Konce retiazky spojíme (vytvoríme tak periodické hraničné podmienky). Potom postupne každú z hrán s istou pravdepodobnosťou prepojíme tak, že odpojíme jeden jej koniec a zapojíme ho na náhodne vybraný uzol (obrázok 5b).



Obr. 5. Štruktúra sietí malého sveta. a) Sieť s pravidelnou štruktúrou, kde klasterizačný koeficient $C = \frac{1}{2}$ a priemerná vzdialenosť rastie s rastúcim počtom uzlov ako $L \propto N$. b)

Štruktúra siete malého sveta. Vďaka náhodným krátkym spojeniam sa radikálne skraca separácia uzlov ($L \propto \ln(N)$) ale, ak krátkych spojení je relatívne málo, lokálna štruktúra siete ostáva prakticky neporušená ($C \approx \frac{1}{2}$).

Ak je táto pravdepodobnosť presmerovania hrany veľmi malá, pre konečný počet uzlov nemusíme dostať žiadne náhodné krátke spojenie. Ak sa blíži k jednej, potom temer každá hrana v sieti je náhodne presmerovaná a dostaneme náhodný graf. Niekde medzi týmito dvoma krajnosťami sa však nachádza oblasť malého sveta, kde pár náhodných krátkych spojení zabezpečí malú separáciu uzlov, ale nenaruší príliš

pravidelnú sieťovú štruktúru. Siete malého sveta tak stoja na pomedzí medzi pravidelnými, mriežkovými grafmi a náhodnými grafmi [1, 14].

Ak má sieť takú distribúciu stupňa uzlov ako na obrázku 4, potom vlastne do veľkej miery platí, že

$$P(k) \propto k^{-\gamma} \quad (7)$$

Exponent γ sa nazýva *škálovacím exponentom* distribúcie stupňa uzlov. Sieť s distribúciou stupňa uzlov typu (7) nazývame *bezškálovou*, pretože na rozdiel od náhodných grafov (obrázok 9) v nej neexistuje žiadna význačná škála, žiaden typický stupeň uzla. Jediná škála je daná konečnosťou počtu uzlov a hrán v sieti, teda veľkosťou siete.

Reálne siete mávajú veľmi často bezškálovú štruktúru. Zároveň v nich, spolu s bezškálovou štruktúrou, často pozorujeme aj isté hierarchické usporiadanie uzlov. Ako sa toto usporiadanie prejavuje na merateľných vlastnostiach siete? O hierarchii v sieťach budeme podrobnejšie hovoriť neskôr. Študovali ju, okrem iných, aj autori Ravasz a Barabási [15], Vásquez [16] a Náther, Markošová a Rudolf [17]. Z týchto prác vyplýva, že hierarchické usporiadanie uzlov sa odzrkadľuje v mocninatej distribúcii priemerných klasterizačných koeficientov uzlov majúcich istý stupeň k

$$C(k) \propto k^{-\delta} \quad (8)$$

kde δ je nový škálovací exponent.

Často pozorovaná kombinácia bezškálovosti a hierarchickej organizácie v reálnych sieťach prirodzene vedie k otázke, či existujú jednoduché, v prírode pozorovateľné procesy, ktoré by v konečnom dôsledku viedli k takýmto vlastnostiam. V ďalších častiach ukážeme, že takéto procesy skutočne existujú.

3 Reálne siete a ich vlastnosti

So sieťami sa človek súčasnosti stretáva celý život. Máme svoje siete sociálnych kontaktov, denne používame internet, mobilnú a telefónnu sieť, premiestňujeme sa pomocou siete ciest, železníc a leteckých trás a myslíme pomocou neurónovej siete nášho mozgu. Mnohí z nás sú napojení na Facebook, poprípade majú stránku na www sieti. Preto štúdium sietí nie je samoúčelné. Ako sa vlastnosti sietí menia s časom, ako sú siete odolné voči poruchám, ako dobre sa pomocou sietí šíria signály, ako dynamika sietí ovplyvňuje ich štruktúru, aké procesy v sieťach vedú k tým ktorým žiaducim, či nežiaducim vlastnostiam, o tom všetkom je užitočné mať predstavu skôr, ako nejakú sieť začneme vytvárať. Okolo seba však pozorujeme siete, ktoré vytvorila sama príroda. Jedna z nich, neurónová sieť mozgu, pracuje v našich hlavách. Ako si vlastne príroda poradila? Ako vytvorila siete optimálnych vlastností? V tejto časti preskúmame niektoré zo sietí, ktoré vznikli viac menej samočinne, bez predošlého plánu, len procesom samoorganizácie.

3.1 Internet

V roku 1999 Faloutsos a jeho spolupracovníci merali vlastnosti internetovej siete [18]. V tom čase bola veľmi aktuálna otázka optimalizácie štruktúry internetu tak, aby komunikácia bola čo najefektívnejšia. Pokusy umelými zásahmi optimalizovať internet neboli príliš úspešné, pretože sa mlčky predpokladalo, že internet je vlastne náhodným grafom. Faloutsos a jeho kolegovia prví ukázali, že internet je bezškálovou sieťou typu (7) a že vlastnosti internetovej siete sa tak výrazne líšia od vlastností v tom čase populárnych náhodných grafov. Neskôr sa tiež zistilo, že takáto bezškálová štruktúra dobre odoláva náhodným poruchám, ale je citlivá na cieľené útoky smerované na uzly s veľkým stupňom [33]. Objav Faloutsosovho tímu jednak vysvetlil neúspechy, ale aj otvoril nové otázky. Týkali sa hlavne procesov, ktorými internet rastie a ich dôsledku na jeho celkovú štruktúru. Viacej svetla do týchto otázok vniesla práca Yooka, Jeonga a Barabásiho [19].

Títo traja autori experimentálne študovali procesy rastu internetovej siete. Internet je na najnižšej úrovni rozlíšenia sieť uzlov - routerov spojených linkami. Uzly sú rozmiestnené na dvojrozmernom povrchu zemegule tak, že vytvárajú fraktálnu štruktúru (s fraktálnou dimenziou $D_f = 1.5$), ktorá silne koreluje s hustotou svetovej populácie. Merania Yooka, Jeonga a Barabásiho ukázali, že internet rastie inkrementálne a jeho štruktúra je určená dvoma základnými kompetitívnymi procesmi:

1. Pravdepodobnosť linky medzi dvoma uzlami klesá lineárne s fyzickou vzdialenosťou medzi nimi.
2. Pravdepodobnosť, že sa nový uzol pripojí k starému je priamo úmerná stupňu tohto uzla (*preferenčné pripájanie*).

Prvý proces je daný tým, že noví užívatelia internetu majú tendenciu pripojiť sa k čo najbližšiemu uzlu, pretože je to najjednoduchšie a aj najekonomickejšie. Druhý proces, ako ukážeme v ďalšom, je zodpovedný za bezškálovú štruktúru internetovej siete. Teda pravdepodobnosť, že sa nový uzol pripojí k starej sieti je daná takýmto vzťahom

$$\pi(k_j, d_{ij}) \sim \frac{k_j^\alpha}{d_{ij}^\sigma} \quad (9)$$

kde k_j je stupeň uzla j a d_{ij} je vzdialenosť medzi uzlom i a j . α a σ sú preddefinované exponenty, ktoré riadia silu preferenčného pripájania a vplyv vzdialenosti. Merania ukazujú, že pre internetovú sieť platí $\sigma = 1$, $\alpha = 1$. Ak meníme tieto exponenty, alebo meníme fraktálnu dimenziu priestoru rozmiestnenia uzlov, potom je topologická štruktúra siete iná, ako skutočná bezškálová štruktúra internetu. Internet tak zaberá isté výnimočné postavenie v priestore parametrov (α, σ, D_f) .

Výskum internetu stále pokračuje a prináša nové zaujímavé výsledky. Okrem internetovej siete na úrovni routerov sa intenzívne študuje internetová sieť na doménovej úrovni. Niekoľko prác je venovaných aj www sieti, kde jednotlivé www

stránky tvoria uzly a hrana medzi uzlami existuje, ak na jednej stránke je hypertextový odkaz na inú stránku [1, 33, 35]. Takejto sieti sa budeme podrobnejšie venovať neskôr.

3.2 Siete v biológii

Výskumy v poslednej dobe potvrdzujú, že aj biologické siete sú často bezškálové, majú vlastnosť sietí malého sveta, poprípade hierarchicky zoskupené uzly. Ako príklad môžu slúžiť metabolické siete [20], siete proteínových interakcií [6], poprípade siete potravinových reťazcov [21]. Napríklad proteínové interakčné siete sú vytvárané vo vnútri každej bunky. Uzlami sú jednotlivé proteíny a hrana existuje vtedy, keď dva proteíny fyzicky interagujú. V metabolických sieťach sú uzly (metabolity) spojené hranou vtedy, keď sú buď substrátmi potrebnými k nejakej metabolickej reakcii, alebo produktami tej istej metabolickej reakcie.

Hierarchická organizácia uzlov sa študovala najmä v metabolických reťazcoch. Hlavne tieto siete sú príkladom biologických bezškálových a hierarchicky organizovaných sietí, ktorých štatistické miery spĺňajú zákony (7) a (8) [22, 23]. Navyše sú aj sieťami malého sveta.

Noort, Snel a Huynen [24] študovali siete génov na príklade *Saccharomyces cerevisiae*. V tejto sieti sú gény uzlami a hrana sa medzi nimi kladie vtedy, ak sú dva gény koregulované. Autori ukázali, že takéto siete majú charakter siete malého sveta a sú zároveň bezškálové. Či v danej sieti existuje nejaká hierarchia, autori neskúmali.

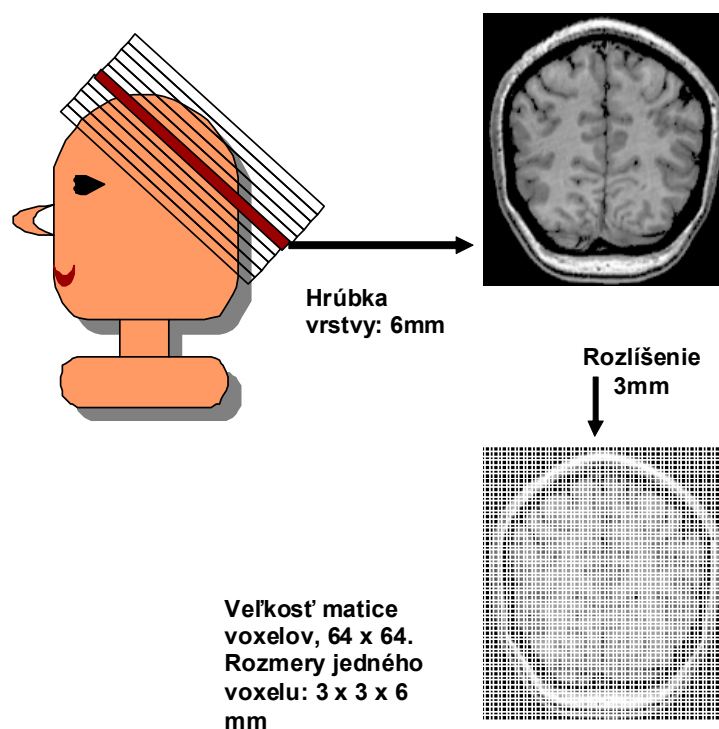
V metabolických, ale aj iných biologických sieťach, však nejde len o ich topologické vlastnosti, ale hlavne o to, aké majú tie ktoré interakcie (hrany) význam. Model biologickej siete musí teda počítať s tým, že nie každá hrana má rovnakú váhu, čo celú situáciu veľmi komplikuje. Nie je to tak dávno, čo sa študujú aj modely sietí s váhovanými hranami [25, 47]. Študujú sa však najmä numerickými metódami, pretože analytické výpočty v takýchto sieťach sú omnoho komplikovanejšie, ako v jednoduchších binárnych sieťach.

3.3 Funkčné siete mozgu

Funkčná sieť mozgu je príkladom ad hoc siete. Iným príkladom ad hoc siete je aj mobilná telefónna sieť. V mobilnej telefónnej sieti sú uzlami ľudia a hrana medzi nimi je prítomná len vtedy, keď dvaja ľudia spolu práve telefonujú pomocou mobilného telefónu. Ad hoc sieť, ako napríklad funkčná sieť mozgu, alebo mobilná telefónna sieť, sa tak v čase neustále mení a za stabilnú ju možno považovať len na krátku dobu. Funkčnú sieť mozgu možno skúmať len vtedy, keď je mozog zaujatý plnením nejakej kognitívnej úlohy. Štúdium funkčných sietí mozgu umožnil až pokrok neinvazívnych metód zobrazovania mozgu in vivo. Jednou z takých metód je v úvode spomenutá metóda funkčnej magnetickej rezonancie [5].

Funkčná magnetická rezonancia využíva zmeny pomerov okysličenej a neokysličenej krvi (BOLD metóda) a tiež fakt, že prietok krvi tými oblasťami mozgu, ktoré sú pri plnení danej kognitívnej úlohy aktivované, je merateľne zvýšený. Zvýšenie je úmerné intenzite aktivity (metóda perfúzie). Scanovanie mozgu pomocou funkčnej magnetickej rezonancie nám pomôže pochopiť obrázok 6. Snímaná osoba leží

v akomsi tuneli, v ktorom má k dispozícii mikrofón a slúchadlá. Do slúchadiel dostáva pokyny od experimentátora, alebo iné signály nevyhnutné k tomu, aby vykonávala dopredu dohodnutú kognitívnu úlohu. Tou úlohou môže byť napríklad opakovanie slov v neznámom jazyku, preklad viet, počúvanie hudby, alebo trebárs aj niečo úplne jednoduché, ako rytmický pohyb prstami jednej, či druhej ruky. Signál sa sníma on line a pomocou vhodného softvéru sa premieta ako séria rezov mozgom na monitor experimentátora sediaceho v druhej miestnosti.



Obr. 6. Schéma usporiadania pri funkčnej magnetickej rezonancii. Mozog sa scanuje po vrstvách, ktoré majú hrúbku niekoľko milimetrov. V každej vrstve sa sníma FMRI signál s rozlíšením 3 mm. Rozlíšenie predstavuje veľkosť jedného *voxelu*, jeho rozmery môžu byť napríklad $3 \times 3 \times 6$ mm, ak vzdialenosť medzi snímanými vrstvami je práve 6 milimetrov.

Experimentátor tak môže priamo sledovať, ktoré časti mozgu počas danej kognitívnej úlohy pracujú intenzívne, ktoré menej a ktoré vôbec nie. Všetky vrstvy mozgu sa oscanujú v priebehu zhruba dvoch sekúnd. Dáta sa uchovávajú, a potom rôznym spôsobom spracúvajú. Pri predspracovaní sa odstránia irelevantné signály (napr. tie, ktoré produkuje biela hmota mozgová) a časovo sa zosynchronizujú tak, aby sme signály zo všetkých vrstiev mohli považovať za zosnímané v tom istom časovom okamihu. Predspracované dáta sa potom uchovávajú vo forme trojrozmernej matice voxelových signálov a neskôr používajú na riešenie rôznych výskumných úloh a otázok.

Jedným zo spôsobov spracovania FMRI dát je aj vytváranie funkčných sietí mozgu. Prví sa o to pokúsil tím Danteho Chialva zo Španielska a USA [3, 26]. Predpokladali, že ľudský mozog je v podstate nelineárny dynamický systém, ktorý pracuje na hranici kritického stavu. Fyzikom je známe, že kritické stavy v nelineárnych dynamických systémoch sú doprevádzané sériou mocninných zákonov podobných rovniciam (7) a (8) [12]. Mocninné zákony znamenajú, že v systéme sú prítomné ďalekodosahové korelácie a to aj napriek tomu, že interakcie medzi elementami dynamického systému sú len krátkodosahové. Ako príklad môže slúžiť kopa piesku [27]. Ak prikladáme veľmi pomaly jednotlivé zrnká na kopy piesku, ktorá má malý sklon, väčšinou sa nič nedeje. Nanajvýš sa spustí malá lavínka, ktorá hneď zaniká. Ak však kopa rastie, pri istom kritickom sklone priloženie jediného zrnka piesku môže spustiť lavínu, ktorej veľkosť je porovnateľná s veľkosťou celej kopy. Veľkosť lavíny môžeme napríklad merať počtom zrníčok, ktoré sa počas trvania lavíny pohli. Keď urobíme distribúciu veľkosti lavín, teda na os x vynesieme veľkosť lavíny a na os y počet lavín danej veľkosti, dostaneme v logaritmickú škálu obrázok podobný obrázku 4. Teda normovaný počet lavín závisí od veľkosti podľa mocninného vzťahu (7), kde k teraz nepredstavuje stupeň uzla, ale veľkosť lavíny a $P(k)$ normovaný počet lavín veľkosti k .

Pohľad na mozog ako na dynamický systém je pohľadom zhora nadol. Vlastnosti mozgu, o ktorých experimentátori už dávno vedia, by sa mali objaviť ako dôsledok kritického správania sa nelineárneho dynamického systému. Napríklad by bolo zaujímavé sledovať, prečo neuróny pri plnení kognitívnych úloh pracujú synchronizovane a ako sa z jednotlivých elektrických impulzov vytvára vedomá myseľ. Vytváranie vedomej mysle sa dá porovnať s fenoménom kultúry, ktorá sa vynorí ako dôsledok individuálnych intelektuálnych počínov.

Ak je mozog skutočne nelineárny dynamický systém pracujúci v kritickom stave, mali by sme pozorovať takéto javy:

1. Na väčších škálach by sme mali pozorovať ďalekodosahové korelácie v čase i priestore.
2. Na malých škálach by sme mali pozorovať lavínovitú činnosť.
3. Adaptívne procesy v mozgu by mali mať lavínovitý charakter, mali by teda trvať veľmi krátko.

Lokálne lavíny neuronálnej aktivity ako prví pozorovali Beggs a Plenz [29]. Čo sa ďalekodosahových korelácií týka, tie sa vyjavujú práve vo funkčných sieťach mozgu. Z nameraných FMRI dát ich môžeme vytvoriť takto. Pod uzlami potenciálnej funkčnej siete budeme rozumieť jednotlivé voxely. Hranu medzi dvoma voxelmi položíme vtedy, keď sú časové signály medzi nimi skorelované. Mieru korelácie medzi voxelom i a j určuje lineárny korelačný koeficient $r(i, j)$

$$r(i, j) = \frac{\langle s_i(t) s_j(t) \rangle - \langle s_i(t) \rangle \langle s_j(t) \rangle}{\sigma(s_i(t)) \sigma(s_j(t))} \quad (10)$$

kde $\sigma^2(s(t)) = \langle s^2(t) \rangle - \langle s(t) \rangle^2$, $s(t)$ je aktivita voxelu v čase t (meraná v nejakých jednotkách) a $\langle \cdot \rangle$ predstavuje časové priemerovanie. Hrana medzi dvoma voxelmi i a j vznikne vtedy, keď korelačný koeficient $r(i, j)$ prekročí istú preddefinovanú prahovú hodnotu r_c . Väčšinou sa prahový koeficient r_c volí v rozmedzí hodnôt 0.6 až 0.8 . Čím menší prahový koeficient použijeme, tým viac uzlov sieť má, ale zároveň tým viac náhodných korelácií zachytávame. Čím je koeficient r_c vyšší, tým lepšie sú aktivity voxelov skorelované, ale naša sieť môže mať príliš málo uzlov na to, aby sme mohli merať jej vlastnosti.

Keď Chialvov tým zmeral distribúciu stupňa uzlov funkčnej siete vzniknutej počas plnenia jednoduchkej kognitívnej úlohy, zistil, že sa správa podľa rovnice (7). Sieť, ktorá sa vytvorila, má tak charakter bezškálovej siete. Posúvaním prahovej hodnoty korelačného koeficientu môžeme pozorovať ako sa charakter distribúcie mení.

Práca Chialvu a jeho kolegov tak ukázala, že v mozgu existujú ďalekosahové korelácie medzi voxelmi [3, 26]. Teória nelineárnych dynamických systémov hovorí [12], že také korelácie môžu naznačovať, že systém pracuje na hranici poriadku a chaosu. Bližšie o funkčných sieťach mozgu pojednám v podkapitole o aplikáciách teórie sietí, keď popíšem naše vlastné spracovanie dát a funkčné siete vznikajúce pri rytmickom pohybe prstov.

Na záver už len krátka poznámka. Je veľmi zaujímavé sledovať FMRI merania mozgu z miestnosti experimentátora. Ja som sa ich zúčastnila počas študijného pobytu na Univerzitnej klinike v Tubingene na jar roku 2009. Meraný študent mal za úlohu rytmicky zatínať a uvoľňovať päsť, najprv niekoľko minút pravú, a potom ľavú, podľa pokynov experimentátora. Na monitore sme sledovali ako sa aktivita mozgových oblastí presúva z ľavej pologule na pravú. Keď sme sa naučili, ktoré časti mozgu sú zamestnané v prípade, keď pracuje pravá päsť, a ktoré, keď pracuje ľavá, požiadal experimentátor študenta, aby prestal skutočne zatínať päste a len si predstavoval, ktorú päsť práve zvierá. Našou úlohou bolo čítať jeho myseľ a z meraných signálov uhádnuť, či práve myslí na pravú, alebo ľavú ruku. Skúsení experimentátori hádali so sto percentnou istotou, my ostatní sme uhádli tak 70 percent prípadov. Ak niekedy budete mať takú možnosť, veľmi odporúčam zúčastniť sa podobného experimentu.

3.4 Sociálne siete

Štruktúra sociálnych sietí sa stala predmetom výskumu už v šesťdesiatych rokoch minulého storočia. Populárnym sa stal napríklad Milgramov experiment [2], ktorý vyústil v sformulovanie známeho zákona šesťstupňovej separácie v sociálnej sieti.

Milgram študoval štruktúru sociálnej siete, a preto si vymyslel takýto experiment. Rozoslal sériu listov na náhodne vybrané adresy po celých Spojených štátoch. Adresátov poprosil, aby odpoveď poslali na adresu známeho brookera v Bostone, ale nie priamo. Mali využiť sieť svojich sociálnych kontaktov a poslať list tomu, s kým sa dobre poznajú a o kom predpokladajú, že by mohol poznať niekoho, kto pozná niekoho... kto pozná onoho brookera. Cieľom experimentu bolo určiť, koľkými rukami list v priemere prejde, kým sa dostane k adresátovi. Milgram zistil, že človek potrebuje

okolo šesť sprostredkovateľov k tomu, aby sa dostal k inému, úplne neznámemu človeku. Tento zákon šesťstupňovej separácie v podstate potvrdzuje, že sociálne siete majú vlastnosti sietí malého sveta. Dokonca aj samotný pojem *sieť malého sveta* zaviedol Milgram.

Prečo je tomu tak? Skúsme uvažovať. Vaša aj moja sociálna sieť má, dalo by sa povedať, istú štruktúru. Vytvárame si skupiny priateľov, napríklad na báze spoločných záujmov, poprípade na pracovnom základe. Ľudia v rámci jednej skupiny sa zvyčajne nepoznajú len s vami, ale aj navzájom. Priemerný klasterizačný koeficient v sociálnej sieti môže byť preto dosť vysoký. Zároveň však v našej sociálnej sieti existujú aj krátke spojenia. Vzniknú napríklad tak, že náš priateľ sa odsťahuje niekam ďaleko, a v prípade potreby nám tak umožní kontakt s ľuďmi, ku ktorým by sme sa inak vôbec nedostali.

Separácia dvoch osôb v sieti profesionálnych kontaktov, kde napríklad v hereckej sieti herec je uzol a všetci herci, ktorí s ním hrali v niektorom filme, sú s ním spojení hranami, alebo napríklad vedec je uzol a všetci tí, čo s ním napísali článok sú s ním spojení hranami, je ešte menšia [1]. Neskoršie štúdie ukázali [30, 31, 32], že sociálne siete majú aj vlastnosti bezškálovej siete, lebo sa v nich uplatňuje preferenčné pripájanie uzlov. Znamená to, že človek, ktorý je známy a má veľa kontaktov, má väčšiu pravdepodobnosť získať ďalšie kontakty, ako ten, čo ich nemá. Aj slávny herec bude skôr angažovaný do nejakého nového filmu ako začiatočník, známy vedec má väčšiu nádej napísať prácu s iným vedcom, ako niekto úplne neznámy.

V roku 2001 sa v Nature objavil zaujímavý článok, ktorý skúmal štruktúru iného typu sociálnej siete, siete sexuálnych kontaktov [32]. Autori analyzovali dáta zozbierané vo švédskom sociálnom výskume. Dotazník skúmal sexuálne správanie 2810 náhodne vybraných osôb vo veku od 17 do 74 rokov. Pretože sieť sexuálnych kontaktov nie je stabilná a s časom sa mení, autori sa najprv pýtali na počet partnerov za posledných 12 mesiacov. Predpokladali, že zmeny počas takej krátkej doby je možné zanedbať. V sieti sexuálnych kontaktov je jedinec uzlom a počet kontaktov predstavuje stupeň uzla k . Dobrou matematickou reprezentáciou siete sexuálnych kontaktov je bipartitný graf. V tomto grafe patria muži do jednej partície a ženy do druhej. Od prípadov, kde tomu tak nie je, môžeme odhliadnuť.

Distribúcia stupňa uzlov pre ženy aj mužov je veľmi podobná. V podstate splňa zákon (7), so škálovacími exponentami $\gamma = 2.54$ pre mužov a $\gamma = 2.31$ pre ženy. Ďalšia otázka sa týkala počtu partnerov, ktorých respondenti vystriedali počas celého doterajšieho života. Distribúcia stupňa uzlov v takejto sieti je podobná ako v prvom prípade, len s trochu inou hodnotou škálovacích exponentov ($\gamma = 2.1$ pre mužov a $\gamma = 1.6$ pre ženy). Sieť sexuálnych kontaktov, podobne ako iné sociálne siete, má tak bezškálový charakter. Pretože je aj sociálnou sieťou, dá sa predpokladať, že bude mať aj vlastnosti sietí malého sveta.

3.5 Zhrnutie

Na príklade niekoľkých veľmi rôznych reálnych sietí sme si mohli všimnúť, že často krát majú podobnú štruktúru. Charakterizuje ju bezškálovosť, poprípade hierarchia

v usporiadaní uzlov a tiež to, že sú sieťami malého sveta. Hierarchia v sieťach sa však skúma len pár rokov, preto staršie práce sa touto otázkou nezaobierajú. Fakt, že reálne siete majú podobné vlastnosti, vedie k oprávneným úvahám o tom, že musí existovať jednoduchý a v prírode pozorovaný spôsob rastu sietí, ktorého výsledkom sú pozorované vlastnosti reálnych sietí. V ďalších častiach tejto práce ukážeme o aké procesy ide.

4 Modely rastúcich sietí

Ak chceme porozumieť tomu, ako súvisí dynamika sietí s ich štruktúrou, môžeme študovať vhodné modely. V tejto časti predstavím základné modely rastúcich sietí. Potom sa pokúsím o niekoľko modifikácií týchto základných modelov, na ktorých ukážem, že finálna štruktúra siete je citlivo závislá na detailoch lokálnej dynamiky [9, 10, 33].

Aby sme získali istý vzhľad do dynamiky modelov, je vhodné ich najprv simulovať na počítači. Skôr ako by sme sa snažili vyriešiť rovnice, získame takto istý odhad ako sa štruktúra siete mení v dôsledku dynamiky. Môžeme napríklad zmerať to, čo sa zmerať dá, teda charakteristiky, ktoré so štruktúrou siete súvisia. Môžeme zmerať distribúciu stupňa uzlov, priemerný stupeň, distribúciu priemerných klasterizačných koeficientov uzlov majúcich istý stupeň, poprípade iné charakteristiky nasimulovanej siete.

Rastúce siete môžeme na počítači modelovať takto:

1. Na začiatku máme pár uzlov pospájaných niekoľkými hranami. Simuláciu môžeme napríklad štartovať z malého kompletného grafu. Na tom v podstate príliš nezáleží, lebo vlastnosti veľkej siete počiatkový graf vôbec neovplyvní.
2. Nech každú časovú jednotku odkiaľsi z vesmíru priletí jeden uzol a pripojí sa m hranami k starým uzlom. Uzol môžeme identifikovať pomocou času s , v ktorom sa objavil. Uzol, ktorý prišiel do siete v čase 1 má index $s = 1$, ten, čo prišiel v čase 2 má $s = 2$ a podobne. *To, akým spôsobom pripájanie uzlov prebieha, je veľmi dôležité z hľadiska celkovej štruktúry siete.*
3. Pridáme ďalšie lokálne procesy, ktoré sa dejú v tom istom časovom okamihu ako pripájanie nového uzla.
4. Opakujeme od bodu 2. Sieť, ktorej finálnu štruktúru skúmame, musí byť dostatočne veľká, čím dlhšie simulácia beží, tým presnejšie zmeriame charakteristiky siete.

Rôzne modely sietí sa líšia v implementácii bodu 2. a 3. K bodu 2. treba povedať, že sú dva základné spôsoby pripájania uzlov k sieti: náhodné pripájanie a preferenčné pripájanie [9, 10, 33]. Pri náhodnom pripájaní pravdepodobnosť, že sa nový uzol pripojí jedným koncom hrany k nejakému starému uzlu, je rovnaká pre všetky staré uzly, teda

$$\pi_r = \frac{1}{N(t)} \quad (11)$$

kde $N(t)$ je počet uzlov v sieti v čase t . Ak sa uzol pripojí do siete preferenčne, pravdepodobnosť pripojenia jedným koncom hrany k nejakému starému uzlu s je úmerná stupňu tohto uzla, teda

$$\pi_p = \frac{k_i}{\sum_{i=1}^N k_i} \quad (12)$$

V rovnici (12) k_i je stupeň uzla i . Proces rastu siete s obidvoma spôsobmi pripájania uzlov možno opísať aj matematicky. Ak použijeme prvý spôsob pripájania, dostaneme model s náhodným pripájaním, ak použijeme druhý spôsob, získame Barabási – Albert model (BA model) s preferenčným pripájaním uzlov [9, 10].

4.1 Model s náhodným pripájaním uzlov

Predstavme si rastúcu sieť s náhodným pripájaním uzlov [10, 33]. Spustíme na počítači vyššie opísaný proces pre najjednoduchší prípad $m=1$ a urobíme napríklad desať simulácií rastúcej siete. Keď sa pozeráme na jeden konkrétny uzol s , v čase t bude mať v prvej simulácii istý stupeň k_s^1 . V druhej simulácii bude mať v čase t vo všeobecnosti stupeň $k_s^2 \neq k_s^1$. Podobne je tomu aj v ďalších simuláciách. Stupeň konkrétneho uzla je tak náhodnou premennou. Preto ak chceme vedieť, ako sa stupeň uzla s mení s časom, môžeme skúmať len priemerný stupeň uzla s a jeho časovú závislosť.

Môžeme sa preto najprv opýtať, aká je pravdepodobnosť $p(k, s, t)$, že uzol, ktorý prišiel do siete v čase s , bude mať v čase t nejaký stupeň k :

$$p(k, s, t) = \frac{1}{N(t)} p(k-1, s, t-1) + \left(1 - \frac{1}{N(t)}\right) p(k, s, t-1) \quad (13)$$

Prvý člen na pravej strane rovnice (4.3) znamená, že k uzlu s so stupňom $k-1$ (ktorý mal v čase $t-1$) sa v čase t pripojila jedna hrana a stupeň tohto uzla sa zvýšil na k . Hrana sa pripojila s pravdepodobnosťou $\frac{1}{N(t)}$ (11). Druhý člen v rovnici (13) hovorí, že uzol s už v čase $t-1$ mal stupeň k a nič sa nestalo.

Ako vyriešiť rovnicu (13)? V prvom rade musíme vedieť počet uzlov siete v danom čase. Keďže simuláciu štartujeme z malého počiatočného grafu, ktorý v konečnom dôsledku veľmi nezaváži, a keďže každú časovú jednotku prichádza do siete jeden uzol, $N(t) \approx t$. Potom rovnicu (13) možno prepísať takto:

$$p(k, s, t) = \frac{1}{t} p(k-1, s, t-1) + \left(1 - \frac{1}{t}\right) p(k, s, t-1) \quad (14)$$

Keď túto rovnicu vynásobíme časom t , upravíme a prejdeme k spojitej limite (čo pre veľké siete môžeme urobiť) [10, 33], diferenčná rovnica (14) sa zmení na diferenciálnu

$$t \frac{\partial p(k, s, t)}{\partial t} = - \frac{\partial p(k, s, t)}{\partial k} \quad (15)$$

Ak (15) vynásobíme veličinou k a preintegrujeme cez k v medziach od 0 do ∞ dostaneme s využitím vzťahu $k(s, t) = \int_0^{\infty} k p(k, s, t) dk$ konečne rovnicu pre priemerný stupeň uzla s , v čase t , $k(s, t)$:

$$\frac{\partial k(s, t)}{\partial t} = \frac{1}{t} \quad (16)$$

Riešením tejto rovnice je

$$k(s, t) = 1 - \ln\left(\frac{s}{t}\right) \quad (17)$$

kde sme využili hraničnú podmienku $k(s, s) = 1$, teda, že uzol, ktorý prišiel do siete v čase s má v dobe svojho zrodu stupeň jedna (jednou hranou sa pripojil k starým uzlom). Vieme teda, ako sa priemerný stupeň uzla s mení s časom. To, čo nás však zaujíma ešte viac je štruktúra siete, teda to, koľko akých uzlov v sieti je a ako sú poprepájané. Aká je napríklad distribúcia stupňa uzlov? Dá sa to analyticky vypočítať? Dobrou správou je, že dá [10, 33]. Potrebujeme v podstate spočítať koľko uzlov daného stupňa sa v takto skonštruovanej sieti nachádza a prenormovať počtom všetkých uzlov:

$$P(k, t) = \frac{1}{t} \int_0^t ds \delta(k - k(s, t)) = - \left[t \frac{\partial k(s, t)}{\partial s} \right]_{s=k(s, t)}^{-1} \quad (18)$$

V tejto rovnici $N(t) = t$ je počet uzlov v sieti a delta funkcia sa rovná jednej ak $k = k(s, t)$. Ak integrujeme cez všetky uzly s , spočítame tak vlastne počet tých uzlov, ktoré majú práve stupeň k . Pravú stranu rovnice (18) získame využitím vlastností delta funkcie [10, 33, 36]. S použitím rovníc (17) a (18) dostaneme pre veľké časy distribúciu stupňa uzlov siete rastúcej pomocou procesu náhodného pripájania nových uzlov

$$P(k) \propto e^{-\alpha k} \quad (19)$$

kde $\alpha = \ln(2) > 0$. Vidíme, že v takejto sieti je veľa uzlov s malým stupňom. Počet uzlov s väčším stupňom exponenciálne rýchlo klesá k nule.

Model siete s náhodným prepájaním uzlov je dobrý vtedy, ak dokáže vysvetliť štruktúru sietí, ktoré rastú podobnými procesmi. Neskôr spomeniem nejaké príklady.

4.2 Barabási – Albert model

Skúsme sa teraz pozrieť podrobnejšie na iný model, v ktorom sa uzly pripájajú do siete preferenčným spôsobom [9, 10, 33]. Opäť zvolíme najjednoduchší prípad, keď každý nový uzol sa do siete pripojí jednou hranou ($m = 1$). Úprava rovníc pre prípad $m > 1$ je veľmi jednoduchá a pre čitateľa bude užitočným cvičením. Pravdepodobnosť pripojenia konca hrany k uzlu je pri preferenčnom pripájaní daná rovnicou (12). Takýto model rastúcej siete bol prvýkrát analyzovaný roku 1999 dvoma autormi a nesie aj ich meno: Barabási – Albert model (BA model) [9]. Barabási a Albert, inšpirovaní častým výskytom bezškálovej štruktúry v reálnych sieťach, vznikajúcich samoorganizovanými procesmi, hľadali jednoduchú a v prírode pozorovateľnú dynamiku, ktorá by prirodzene viedla k vzniku takejto štruktúry. V práci [9] dokázali, že preferenčné pripájanie uzlov je takou dynamikou.

Ak proces simulujeme na počítači a pozorujeme aký stupeň má v čase t istý uzol s , zistíme, že stupeň uzla s je, podobne ako v prípade náhodného pripájania nových hrán, náhodnou premennou. Preto môžeme niečo povedať len o tom, ako sa s časom mení priemerný stupeň uzla s , $k(s, t)$. Podobne ako v predošlom modeli sa pýtame, aká je pravdepodobnosť, že uzol s má v čase $t+1$ stupeň k :

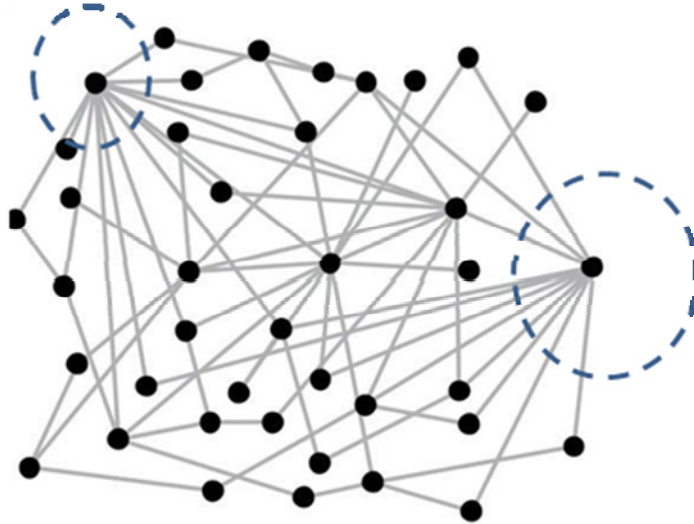
$$p(k, s, t+1) = \frac{k-1}{\sum_{i=1}^t k_i} p(k-1, s, t) + \left(1 - \frac{k}{\sum_{i=1}^t k_i}\right) p(k, s, t) \quad (20)$$

Členy na pravej strane rovnice (20) majú podobný význam ako v prípade náhodného pripájania uzlov, až na to, že pravdepodobnosť pripojenia je úmerná stupňu uzlov, a normovacia konštanta je preto súčtom všetkých stupňov uzlov (12). Podľa rovnice (1) je tento súčet rovný dvojnásobku počtu hrán v sieti. Keďže každú časovú jednotku prichádza do siete jedna nová hrana, pre veľké t je počet hrán v systéme $e(t) = 2t$. Ak, samozrejme, odhliadneme od tých pár hrán v počiatočnom malom

grafe. Teda môžeme položiť $\sum_{i=1}^t k_i = 2t$.

Ak to dosadíme do (4.10), dostaneme

$$p(k, s, t+1) = \frac{k-1}{2t} p(k-1, s, t) + \left(1 - \frac{k}{2t}\right) p(k, s, t) \quad (21)$$



Obr. 7. Vizualizácia siete vytvorenej preferenčným pripájaním uzlov, kde $m = 2$ je počet hrán prinesených novým uzlom. Z obrázku vidieť, že sieť pozostáva z uzlov rôznych stupňov. Dva uzly s veľkým počtom susedov (huby) sú vyznačené trhanou čiarou. Najviac uzlov je však takých, ktoré majú malý stupeň. Simulované pomocou Network Workbench Tool [34].

Keď rovnicu (21) pre násobíme $2t$, upravíme a prejdeme k spojitej limite, dostaneme diferenciálnu rovnicu

$$2t \frac{\partial p(k, s, t)}{\partial t} + \frac{\partial k p(k, s, t)}{\partial k} = 0 \quad (22)$$

a po prenasobení (22) faktorom k a integráciou oboch strán rovnice cez dk v medziach od 0 po ∞ dostaneme rovnicu pre priemerný stupeň uzla s , $k(s, t)$

$$\frac{\partial k(s, t)}{\partial t} = \frac{k(s, t)}{2t} \quad (23)$$

Riešením (23) je

$$k(s, t) \propto \left(\frac{t}{s}\right)^\beta, \quad \beta = \frac{1}{2} \quad (24)$$

a ak použijeme vzťah pre výpočet distribúcie stupňa uzlov (18), dostaneme

$$P(k) \propto k^{-\gamma}, \quad \gamma = 3 \quad (25)$$

Dá sa ukázať, že pre bezškálové siete platí [10, 33]

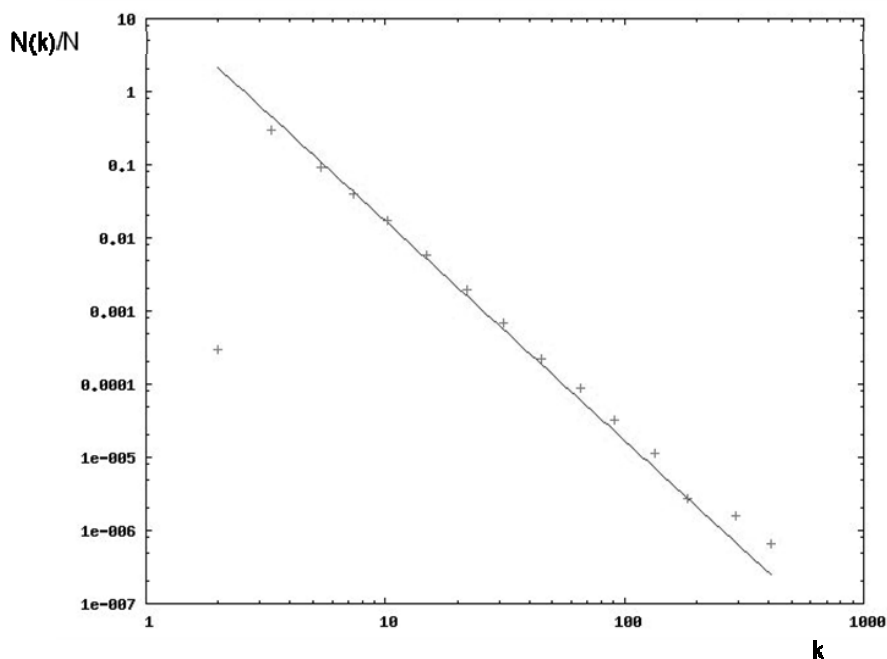
$$\gamma = 1 + \frac{1}{\beta} \quad (26)$$

a teda exponent γ možno získať aj bez počítania distribúcie (18). Ako možno nájsť vzájomnú závislosť exponentov v bezškálových sieťach? Z rovnice (24) vieme, že ak zafixujeme čas, potom platí $k \propto s^{-\beta}$. Rovnica (18) zasa hovorí, že $k^{-\gamma} \propto \frac{\partial s}{\partial k} \propto k^{-1-1/\beta}$. Porovnaním ľavej a pravej strany tejto rovnice a s použitím vzťahu (25) dostaneme (26).

To, čo sme práve matematicky dokázali (18, 24, 25), nie je nič iné, ako dokázané tvrdenie, že preferenčné pripájanie uzlov je ten mechanizmus, ktorý vytvára bezškálový typ siete. Navyše preferenčné pripájanie nových uzlov k sieti nie je neznáme ani v realite. Predstavme si napríklad sociálnu sieť. Už sme spomenuli, že človek (uzol), ktorý je známy a populárny (má veľa kontaktov, a teda vysoký stupeň), má omnoho väčšiu šancu získať nové kontakty a ďalšie priateľstvá, ako ten, ktorého nikto nepozná. Aj v citačnej sieti ten článok (uzol), ktorý prináša nejaký významný objav, má omnoho väčšiu šancu byť citovaný v iných prácach [35]. Podobne je tomu aj v iných reálnych sieťach, spomínali sme napríklad sieť profesionálnych kontaktov [30, 31]. Ale je to tak aj vo www sieti [33, 37, 38], v ktorej uzly sú predstavované www stránkami a hrany hypertextovými odkazmi. Stránka významnej osobnosti, alebo inštitúcie má určite väčšiu pravdepodobnosť, že sa na ňu budú odkazovať z iných stránok, ako úplne neznáma www stránka.

Pravdepodobne som vás presvedčila, že preferenčné pripájanie uzlov môže byť dôvodom bezškálovej štruktúry reálnych sietí. Preto je BA model základným modelom, ktorý pomáha porozumieť ich štruktúre a vystihuje základné kvalitatívne ba dokonca aj kvantitatívne vlastnosti mnohých reálnych sietí. Kvantitatívne preto, lebo merania v reálnych sieťach ukazujú, že škálovacie exponenty γ (25) sú v nich obvyčajne v rozmedzí hodnôt 2.5–3.0 [10, 33]. Uvidíme neskôr, že zmena škálovacieho exponentu je spôsobená dodatočnými dynamickými procesmi v sieti.

Mocninná distribúcia stupňa uzlov (obrázok 8) naznačuje, že v sieti sa vyskytujú uzly všetkých stupňov. Jediným ohraničením stupňa je počet hrán v sieti. Sieť má zaujímavú, sebe podobnú štruktúru, ktorá sa veľmi často vyskytuje aj v reálnych sieťach (obrázok 7). Niet v nej žiaden význačný, typický uzol s typickým stupňom. Práve preto sa takáto sieť nazýva bezškálovou.



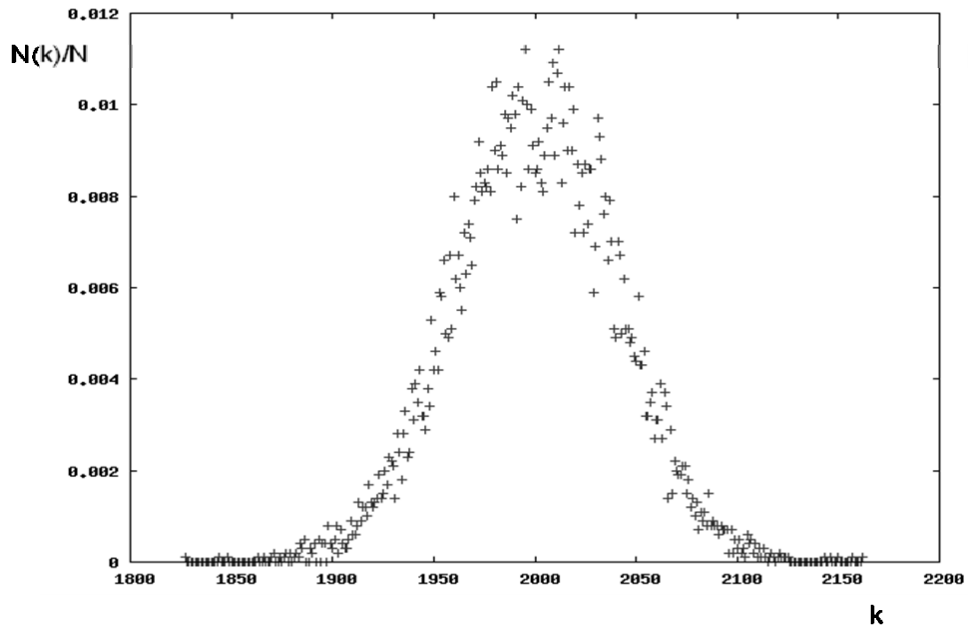
Obr. 8. Distribúcia stupňa uzlov pre BA model. Sieť má $N=30000$ uzlov, $m=2$. $N(k)$ predstavuje počet uzlov stupňa k . Obrázok je v logaritmickej škále a lineárna závislosť potvrdzuje rovnicu (25). Exponent γ možno zmerať ako tangens smerového uhla priamky.

V BA modeli, ako sme videli vyššie, je pravdepodobnosť pripojenia novej hrany k starému uzlu úmerná stupňu tohto uzlu. To vedie k mocninnej distribúcii stupňa uzlov so škálovacím exponentom $\gamma = 3$ (25). Skúsme BA model mierne pozmeniť [10]. Predstavme si, že π_p (12) je úmerné $k+A$, kde A je nejaká konštanta. Potom rovnicu (23) možno modifikovať takto

$$\frac{\partial k(s,t)}{\partial t} = m \frac{k(s,t) + A}{\int_0^t (k(s,t) + A) ds} \quad (27)$$

V rovnici (27) m je počet koncov hrán, ktorými sa nový uzol pripojí k starým uzlom. Normovaciú konštantu, predstavovanú integrálom v menovateli, vypočítame takto: Zintegrujeme obe strany rovnice (4.17) podľa ds a dostaneme

$$\int_0^t \frac{\partial k(s,t)}{\partial t} ds = m \quad (28)$$



Obr. 9. Distribúcia stupňa uzlov pre náhodný graf (Eordos, Rényi model A), pravdepodobnosť hrany medzi dvoma uzlami je 0.1, počet uzlov $N=10000$. $N(k)$ predstavuje počet uzlov majúcih stupen k . Simulované s pomocou Network Workbench Tool [34].

S použitím vzťahu $\frac{\partial}{\partial t} \int_0^t k(s,t) ds = k(t,t) + \int_0^t \frac{\partial k(s,t)}{\partial t} ds = n + m$, kde $k(t,t) = n$

je hraničná podmienka, dostaneme $\int_0^t k(s,t) ds = (n + m)t$ a teda:

$$\int_0^t (k(s,t) + A) ds = (n + m + A)t \quad (29)$$

Rovnica (27) tak bude

$$\frac{\partial k(s,t)}{\partial t} = m \frac{k(s,t) + A}{(n + m + A)t} \quad (30)$$

čo je diferenciálna rovnica, ktorá má riešenie

$$k(s,t) \propto \left(\frac{t}{s}\right)^\beta, \quad \beta = \frac{m}{m + n + A} \quad (31)$$

Odtiaľ dostaneme, že $\gamma = 1 + \frac{1}{\beta} = 2 + \frac{n+A}{m}$. Ak $A=0$ a $n=m$, dostávame γ exponent BA modelu. Konštanta $n > 0$, pretože uzol v čase svojho príchodu do siete sa k starému uzlu vždy pripojí aspoň jednou hranou. Ak aj $n+A > 0$, hodnoty exponentov γ sú v intervale $(2, \infty)$. Konštanta A tak nemení charakter distribúcie stupňov uzlov, mení však veľkosť škálovacieho exponentu. Exponent závisí od dodatočnej atraktivity (fitness) uzla.

4.3 Siete so zmiešaným preferenčným a náhodným pripájaním

Predstavme si teraz model rastúcej siete, v ktorom sa mieša preferenčné a náhodné pripájanie nových hrán k starým uzlom [10]. Počet nových hrán, ktorý pribudne každú časovú jednotku do siete, nech je $n_r + m$ a nech sa m koncov nových hrán pripojí k uzlu s preferenčným spôsobom a n_r koncov nových hrán náhodne. Druhé konce hrán sú pripojené k novému uzlu. Potom rovnicu (27) možno modifikovať takto

$$\frac{\partial k(s,t)}{\partial t} = \frac{n_r}{t} + m \frac{k(s,t) + A}{\int_0^t (k(s,t) + A) ds} \quad (32)$$

V tejto rovnici prvý člen na pravej strane predstavuje pravdepodobnosť pripojenia n_r koncov hrán náhodným spôsobom a druhý pravdepodobnosť pripojenia m koncov hrán preferenčným spôsobom. Priemerný stupeň uzla s sa môže meniť vďaka obidvom spôsobom pripojenia nových hrán. Nech opäť $k(0,0) = 0$ a $k(t,t) = n$. Podobným spôsobom aký bol použitý pre modifikáciu BA modelu, môžeme vypočítať integrál v rovnici (32) a dostaneme

$$\frac{\partial k(s,t)}{\partial t} = \frac{m}{n_r + m + n + A} \times \frac{\left(k(s,t) + A + \frac{n_r}{m} (n_r + m + n + A) \right)}{t} \quad (33)$$

Riešenie rovnice (33) poskytuje

$$k(s,t) \propto \left(\frac{t}{s} \right)^{\frac{m}{(n_r + m + n + A)}} \quad (34)$$

a vezmúc do úvahy vzťah (26) exponent γ pre distribúciu stupňa uzlov bude

$$\gamma = 2 + \frac{n_r + n + A}{m} \quad (35)$$

Podľa rovnice (35) vidíme, že bezškálová sieť môže vzniknúť aj tak, že časť nových hrán sa do siete pripojí náhodným spôsobom. Jediné, čo sa zmení, je škálovací exponent γ .

Ak sa rozhodnete dočítať tento text až po časť 6, dozviete sa aj o ďalších typoch analyticky riešiteľných sieťových modelov. Predstavím v nej niektoré naše výsledky v tejto oblasti. Zavediem ďalšie procesy, napríklad presmerovanie starých hrán, vznik nových hrán medzi starými uzlami a ukážem, ako tieto dodatočné efekty ovplyvňujú celkovú štruktúru siete. Uvidíte tiež, že modely rastúcich sietí nie sú samoúčelnou matematickou ekvilibristikou, ale hlavne nástrojom, ktorý dokáže vysvetliť to, čo sa deje v reálnych sieťach.

4.4 Siete so zrýchleným rastom

V základnom BA modeli, ktorý sme rozobrali v časti 12, počet hrán v systéme je lineárnou funkciou času, lebo do siete každú časovú jednotku pribudne nový uzol, ktorý prinesie konštantný počet nových hrán. Takýto lineárny rast nemeň priemerný stupeň uzlov v sieti. V danej verzii BA modelu suma všetkých stupňov uzlov je $2t$, prípadne $2mt$ (11), ak $m > 1$ je počet hrán, ktoré prinesie do siete každý nový uzol. Počet uzlov v sieti v čase t je $N(t) = t$. Odtiaľ možno vypočítať, že priemerný stupeň

uzla $\langle k \rangle = \frac{2t}{t} = 2$ prípadne $\langle k \rangle = \frac{2mt}{t} = 2m$. Ukazuje sa však, že takýto lineárny

rast nie je v reálnych sieťach úplne typický. Merania dokazujú, že napríklad vo www sieti, v internetovej sieti (na úrovni domén), ako aj v sieti profesionálnych kontaktov priemerný stupeň uzla s časom pomaly, ale iste rastie [39].

Pre BA model je distribúcia stupňa uzlov určená škálovacím exponentom $\gamma = 3$. V predošlých častiach sme ukázali, že toto nie je jediná možná hodnota škálovacieho exponentu pre lineárne rastúce siete. Ak chceme odhadnúť dolnú hranicu intervalu škálovacích exponentov, môžeme napríklad využiť aj fakt, že v takýchto sieťach je priemerný stupeň uzla konečný. Musí teda platiť, že $\langle k \rangle = \int_1^{\infty} kP(k)dk < \infty$.

Ak $P(k) \propto k^{-\gamma}$, potom $\langle k \rangle = \int_1^{\infty} k^{-\gamma+1} dk < \infty$ a teda, ako už bolo iným spôsobom

ukázané, $\gamma > 2$. Škálovací exponent γ_{BA} pre BA model túto nerovnosť spĺňa.

Keď meriame distribúciu $P(k)$ v reálnych bezškálových sieťach, zistíme, že mocninne sa správa len v istých medziach, ktoré sú tvorené dvoma hranicami k_0 a k_{cut} . Prvá hranica znamená, že uzol s nižším stupňom sa v danej sieti nenachádza.

Druhá hranica je spôsobená konečnou veľkosťou siete a počet uzlov s väčším stupňom je rádu jedna. Ak použijeme normalizačnú konštantu

$$\int_{k_0}^{\infty} P(k)dk = 1 \quad (36)$$

potom rovnici (36) vyhovuje takýto tvar distribúcie stupňa uzlov

$$P(k) \sim [(\gamma - 1)k_0^{\gamma-1}] k^{-\gamma}, \quad k_0 < k < k_{cut} \quad (37)$$

Ak $\gamma > 2$, druhú hranicu k_{cut} odhadneme z kumulatívnej distribúcie

$$P_{cum} = \int_k^{\infty} P(k) dk \quad (38)$$

ktorá počíta pravdepodobnosť výskytu uzlov majúcich stupeň väčší ako istý stupeň k . Ak je počet takých uzlov rádovo jedna, potom z rovnice (37), (38) a vzťahu $tP_{cum}(k_{cut}(t)) \sim 1$ vyplýva

$$k_{cut} \sim k_0 t^{\frac{1}{\gamma-1}} \quad (39)$$

Toto všetko platí pre siete s mocninnou závislosťou distribúcie stupňa uzlov, teda tie, ktoré voláme bezškálovými. Jediná škála, ktorá tu zohráva nejakú úlohu, je daná prirodzenými hranicami samotného problému. To, že tieto hranice existujú, sťažuje meranie škálovacieho exponentu pre reálne siete. Napríklad meranie veľkých škálovacích exponentov γ je vďaka závislosti (39) nemožné, pretože lineárna časť distribúcie je veľmi krátka aj vtedy, keď je sieť veľmi veľká. Keďže škálovací exponent sa meria ako tangens smerového uhla pre distribúciu zobrazenú v dvojitej logaritmickú škále, potrebujeme, aby lineárna časť bola dostatočne dlhá pre algoritmy fitujúce dáta priamkou.

Preskúmame teraz konečne siete so zrýchleným rastom [39], v ktorých priemerný stupeň uzla nie je konštantný, ale rastie lineárne s časom

$$\langle k \rangle \propto t^a, \quad a > 0 \quad (40)$$

kde a je exponent rastu. Ak uvažíme, že počet hrán v systéme je daný vzťahom (1) a počet uzlov v čase t je $N(t)=t$, potom $\sum_{i=1}^N k_i = \langle k \rangle N(t)$, potom odtiaľ dostaneme, že

$$e(t) \propto t^{a+1} \quad (41)$$

Pre zrýchlený rast môžeme predpokladať, že distribúcia stupňa uzlov bude nestacionárna a správa sa takto [39]:

$$P(k, t) \sim t^z k^{-\gamma} \quad (42)$$

kde $z > 0$ je nový škálovací exponent. $P(k)$ sa podľa zákona (42) správa len v hraniciach $k_0(t) < k < k_{cut}(t)$. Podobným postupom, ako pre siete bez zrýchleného rastu, len s tým rozdielom, že použijeme (42), určíme, že

$$k_0(t) \sim t^{\frac{z}{(\gamma-1)}} \quad (43)$$

a tiež

$$k_{cut}(t) \sim t^{(z+1)/(\gamma-1)} \quad (44)$$

Rovnica (43) platí dokonca pre $\gamma > 1$. Pozrime sa na dva prípady, $1 < \gamma < 2$ a $\gamma > 2$. Treba mať stále na pamäti, že uvažujeme len prípad, keď uzly do siete len pribúdajú a žiaden z nich zo siete neodbudne.

1. $1 < \gamma < 2$

Pre siete so zrýchleným rastom rastie priemerný stupeň uzlov s časom podľa rovnice (40). Preto ak vypočítame $\langle k \rangle = \int_{k_0}^{k_{cut}} k P(k) dk$ a použijeme (42) a (44),

dostaneme $t^a \sim \int_{k_0}^{k_{cut}} kt^z k^{-\gamma} dk \sim t^{-1+(z+1)/(\gamma-1)}$, lebo integrál je dominovaný

hornou hranicou. Odtiaľ vidíme, že $a + 1 = (z + 1)/(\gamma - 1)$ a efekt konečnosti systému daný k_{cut} sa prejaví so zohľadnením (41) a (43) takto

$$k_{cut} \sim t^{a+1} \sim e(t) \quad (45)$$

Čiže hornou hranicou stupňa uzla je maximálny stupeň, ktorý je pre danú sieť vôbec možný. Horná hranica pre tento prípad teda vlastne neexistuje, resp. je daná len časovo závislým počtom hrán v sieti. Z predošlých úvah je možné určiť aj vzájomnú závislosť škálovacích exponentov a , γ a z :

$$\gamma = 1 + \frac{z+1}{a+1} \quad (46)$$

Ak sa má γ nachádzať v intervale $(1, 2)$, exponent z musí byť menší ako a . Ak je distribúcia stacionárna, teda $z = 0$ (42), potom $\gamma = 1 + \frac{1}{a+1}$.

2. $\gamma > 2$

V tomto prípade integrál $t^a \sim \int_{t^{\frac{z}{\gamma-1}}} kt^z k^{-\gamma} dk \sim t^{z-z(\gamma-2)/(\gamma-1)}$ je dominovaný dolnou

hranicou.

Odtiaľ dostávame

$$\gamma = 1 + \frac{z}{a} \quad (47)$$

Ak má byť γ väčšia ako dva, potom musí platiť, že $z > a > 0$ a tiež aj to, že stacionárna distribúcia stupňa uzlov v tomto prípade vôbec neexistuje. Je tomu tak preto, že ak $z = 0$, exponent γ nemôže byť väčší ako 2. To je však práve prípad, ktorý rozoberáme.

Tak ako v predošlej časti, nech aj pre siete so zrýchleným rastom platí: $p(k, s, t)$ je pravdepodobnosť, že uzol, ktorý došiel do siete v čase s má v čase t stupeň k . Priemerný stupeň uzla s , v čase t sa potom dá vypočítať zo vzťahu

$$k(s, t) = \int_0^{\infty} kp(k, s, t) dk \quad (48)$$

Predpokladajme, že pre bezškálové siete so zrýchleným rastom je $k(s, t)$ dané takýmto vzťahom [39]

$$k(s, t) \propto t^{\delta} \left(\frac{s}{t} \right)^{-\beta} \quad (49)$$

kde δ a β sú škálovacie exponenty. Dá sa ukázať [39], že $p(k, s, t)$ je

$$p(k, s, t) = t^{-\delta} \left(\frac{s}{t} \right)^{\beta} g \left(kt^{-\delta} \left(\frac{s}{t} \right)^{\beta} \right) \quad (50)$$

kde g je škálovacia funkcia. Z rovnice (49) a zo vzťahu $P(k, t) = \frac{1}{t} \int_0^t p(k, s, t) ds$

a tiež (42) možno odvodiť vzťahy pre škálovacie exponenty

$$\gamma = 1 + \frac{1}{\beta} \quad (51)$$

$$z = \frac{\delta}{\beta} \quad (52)$$

Keď vezmeme do úvahy rovnice pre škálovacie exponenty, môžeme konečne odvodiť finálne vzťahy pre $p(k, s, t)$ a $P(k, t)$:

$$p(k, s, t) = \frac{s^{\frac{1}{(\gamma-1)}}}{t^{(z+1)/(\gamma-1)}} \mathcal{G} \left(k \frac{s^{\frac{1}{(\gamma-1)}}}{t^{(z+1)/(\gamma-1)}} \right) \quad (53)$$

$$P(k, t) = t^z k^{-\gamma} G(kt^{-(z+1)/(\gamma-1)}) \quad (54)$$

V (54) $G(\cdot)$ je škálovacia funkcia. V prípade, že $z = 0$, máme siete, v ktorých nepozorujeme zrýchlený rast.

V tejto časti práce som uviedla všeobecnú teóriu sietí so zrýchleným rastom. Vzťahy pre siete s lineárnym rastom, reprezentované napríklad BA modelom sú vlastne limitným prípadom sietí so zrýchleným rastom. V nasledujúcej podkapitole zistíme, ako modely sietí so zrýchleným rastom modelujú reálne situácie. Neskôr, v časti dotýkajúcej sa praktických aplikácií teórie sietí, ako ilustráciu toho, že teória sietí nie je samoučelná, ukážem aj niektoré reálne siete, ktoré vznikali procesom zrýchleného rastu.

4.5 Ilustratívne modely pre siete so zrýchleným rastom

Predpokladajme, že naša sieť rastie tak, ako to bolo opísané v úvode tejto časti. Predpokladajme ďalej, že pripájanie uzlov je preferenčné, teda $\pi_p \sim k(s, t) + A(t)$ a $A(t)$ nech je nejaká dodatočná, vo všeobecnosti časovo závislá veličina, ktorú nazveme fitness, alebo aj atraktivita uzla. Nech počet hrán, ktoré každú časovú jednotku prinesie do systému nový uzol, je daný mocninným zákonom $c_0 t^a$.

Nech najprv $A(t) = A$ je od času nezávislou konštantou. Potom pre priemerný stupeň uzla, ktorý prišiel do siete v čase s a my ho pozorujeme v čase t platí

$$\frac{\partial k(s, t)}{\partial t} = c_0 t^a \frac{k(s, t) + A}{\int_0^t [k(s, t) + A] ds} \quad (55)$$

Počiatkové a hraničné podmienky sú $k(0, 0) = k(t, t) = 0$. Ak rovnicu (55) vyriešime, dostaneme výsledok

$$k(s, t) = A \left(\frac{t}{s} \right)^{a+1} = A \left(\frac{s}{t} \right)^{-(a+1)} \quad (56)$$

a použijúc vzťah (51) pre škálovací exponent distribúcie stupňa uzlov získame

$$\gamma = 1 + \frac{1}{a+1} \quad (57)$$

Predstavme si na chvíľu, že čas t zafixujeme. Potom $k(s, t) \propto s^{-(a+1)}$ a preto, lebo stupeň uzla nemôže rásť nad všetky medze, $a + 1 > 1$. Odtiaľ vidíme, že $\gamma < 2$ (57). Takáto sieť je teda modelom prípadu $1 < \gamma < 2$, spomínaného v časti 4.4. Ak vypočítame distribúciu stupňa uzlov pomocou rovnice (18), dostaneme od času nezávislú, stacionárnu distribúciu

$$P(k) = \frac{A^{1/(a+1)}}{a+1} k^{-\left(1+\frac{1}{a+1}\right)} \quad (58)$$

Vzhľadom na (46) pre $z = 0$ tento výsledok nikoho neprekvapí.

Predpokladajme teraz, že dodatočná atraktivita (fitness) uzla v sieti nie je konštantná, ale sa mení s časom. Konkrétne, nech $A(t) = B \frac{c_0 t^a}{a+1}$, $B > 0$, B nech je konštanta. Podobne ako v predošlom prípade môžeme vypočítať distribúciu stupňa uzlov. Ukazuje sa, že je nestacionárna, a teda závislá od času

$$P(k, t) \sim t^{a(1+B)/(1-Ba)} k^{-\left(1+\frac{1+B}{1-Ba}\right)} \quad (59)$$

ak k je omnoho väčšie ako t^a . Vidíme, že ak $Ba < 1$, potom škálovací exponent

$$\gamma = 1 + \frac{1+B}{1-Ba} > 2 \quad (60)$$

Takáto sieť je teda modelom druhého prípadu, opísaného v časti 4.4, kde $\gamma > 2$. Týmito výpočtami som vlastne ukázala, že mocninný rast priemerného stupňa uzlov v sieti vedie na mocninnú distribúciu stupňa uzlov.

Teória sietí so zrýchleným rastom umožňuje pochopiť aj vlastnosti niektorých reálnych sietí. Ukážem niekoľko konkrétnych príkladov. O sieťach jazyka a o svojich vlastných výsledkoch, týkajúcich sa tohto špeciálneho typu sietí s časovo závislou fitness budem hovoriť neskôr. Na tomto mieste ukážem modely distribúcie kapitálu v spoločnosti, ktorá sa s časom nejakým spôsobom mení. Ukážem tiež ako je bohatstvo rozdelené v spoločnosti, ktorá stagnuje, potom v tej, ktorá sa rozvíja a nakoniec v takej, ktorá zaniká [39].

Aby sme takéto spoločenstvo mohli modelovať, zavedieme niekoľko zjednodušení. Budeme predpokladať, že každú časovú jednotku sa narodí len jeden človek, a že nik nezomrie. Ďalej predpokladáme, že neexistuje žiadna inflácia a peniaze sa nikde nestrácajú. Pod stabilnou spoločnosťou budeme rozumieť takú, kde sa množstvo peňazí vlastnených jedincom s časom nemení. Ak sa spoločnosť rozvíja, tento kapitál s časom rastie a v zanikajúcich spoločnostiach, naopak, klesá. Ak je distribúcia kapitálu k , $P(k)$, daná mocninným zákonom $P(k) \propto k^{-\gamma}$ a $\gamma < 2$, potom spoločnosť nie je férová. Nachádza sa v nej veľa ľudí, ktorí vlastnia veľmi málo, ale aj pomerne veľa boháčov. Naopak ak $\gamma > 2$, spoločenstvo je férové v tom, že počet

veľmi bohatých ľudí rýchlo klesá s rastúcim bohatstvom. Ak $P(k)$ klesá exponenciálne rýchlo, spoločnosť je superférová, pretože v nej veľmi majetní jedinci prakticky neexistujú. V tejto spoločnosti sú si všetci rovní, ale rovní v chudobe.

Takáto predstava o férovosti spoločnosti sa môže zdať trochu divnou. Za férovú skôr zvykneme považovať takú spoločnosť, v ktorej je veľký podiel relatívne bohatých jedincov a nie tú, kde všetci majú málo. Autori práce [39] však férovosť definovali tak, ako som to opísala vyššie.

Predstavme si, že tok kapitálu do spoločnosti je mocninou funkciou času, teda je úmerný $\langle k \rangle \propto t^\alpha$. Exponent α hovorí o tom aká spoločnosť je. Ak $\alpha = 0$, spoločnosť je stabilná, ak je $\alpha < 0$ ($\alpha > 0$) spoločnosť zaniká (rozvíja sa).

Zo skúsenosti všetci vieme, že platí jednoduché pravidlo, ktoré sa dá vyjadriť vetou „peniaze priťahujú ďalšie peniaze“. Toto pravidlo je v spoločnosti často vyvažované tým, že časť bohatstva sa rozdeľuje rovnakým dielom medzi všetkých jej členov. Spravodlivosť totiž velí, že vo väčšine spoločností má každý jedinec nárok na istý štartovací kapitál. Patria k nemu peniaze, ale aj duchovné bohatstvo, predstavované napríklad vzdelaním. Aj vzdelanie totiž môže priniesť človeku nejaké peniaze. Isté množstvo kapitálu môže človek dostať už pri narodení. Počiatočný kapitál jedinca môže s časom rásť, alebo môže ostať celý čas konštantný.

V ďalšom budeme predpokladať, že jednotliví členovia spoločnosti dostanú nejaký počiatočný kapitál pri narodení v čase s , kde $s < t$. Pretože, podľa predpokladu, každú časovú jednotku sa narodí len jeden člen spoločnosti, čas narodenia s je aj identifikačným znakom, pomenovaním jedinca. Predpokladajme ďalej, že $k(s, t)$ je priemerný kapitál osoby, ktorá sa narodila v čase s a my sa s ňou stretáme v čase t .

Stagnujúce spoločenstvo

Nech m_s je kapitál, ktorý dostane osoba s pri narodení. Nech $A > 0$ je dispozícia jedinca zarábať peniaze. (Všimnime si, že táto dispozícia (fitness) môže byť celý život konštantná, alebo môže s časom rásť, napríklad vďaka vzdelaniu, či klesať, napríklad kvôli chorobe.) Predpokladáme, že schopnosť jedinca zarábať peniaze sa s časom nemení. Každú časovú jednotku do spoločnosti prichádza kapitál m , z ktorého jedinec dostane istý podiel. Zlomok p z tohto kapitálu je distribuovaný rovnako každému, časť $(1-p)$ si jedinci privlastňujú podľa zákona „peniaze prinášajú peniaze“, teda preferenčne. Individuálne bohatstvo osoby s sa bude s časom vyvíjať takto

$$\frac{\partial k(s, t)}{\partial t} = \frac{pm}{t} + (1-p)m \frac{k(s, t) + A}{\int_0^t (k(s, t) + A) ds} \quad (61)$$

s počiatočnou podmienkou $k(0, 0) = 0$ a s hraničnou podmienkou $k(t, t) = m_s$. Keď rovnicu (61) vyriešime a pomocou metód spomínaných v predošlých častiach tejto

kapitoly získame distribúciu stupňa uzlov, zistíme, že má mocninný charakter (25) s exponentom

$$\gamma = 2 + \frac{pm + m_s + A}{(1-p)m} > 2 \quad (62)$$

Pre stabilné spoločnosti platí, že sú férové, ak teda férovosť znamená relatívnu rovnosť bohatstva jedincov, aj keď je to rovnosť v chudobe.

Spoločenstvo, ktoré sa rozvíja alebo zaniká

Predpokladajme teraz, že štartovací kapitál, ktorý jedinec dostane pri svojom narodení, nie je konštantný, ale je úmerný priemernému bohatstvu v spoločnosti $m_s(t) = dmt^\alpha$, kde d je kladná konštanta. Priemerné bohatstvo v spoločnosti sa s časom mení. Nech sa spoločenské bohatstvo opäť distribuuje čiastočne férovo (teda rovnakým dielom každému) a čiastočne preferenčne (teda tí, čo majú viac, aj viac dostanú). Nech zlomok férovo distribuovanej zložky je p , a zlomok preferenčne distribuovanej zložky nech je $1-p$. Pre jednoduchosť tiež predpokladajme, že každý má rovnakú (nulovú) dodatočnú schopnosť zarábať peniaze ($A(s,t) = 0$). Potom sa priemerné individuálne bohatstvo osoby s mení s časom takto

$$\frac{\partial k(s,t)}{\partial t} = \frac{mt^\alpha p}{t} + (1-p)mt^\alpha \frac{k(s,t)}{\int_0^t k(s,t) ds} \quad (63)$$

Počiatková podmienka aj v tomto prípade je $k(0,0) = 0$ a hraničná podmienka $k(t,t) = dmt^\alpha$. Riešenie rovnice (63) vedie k distribúcii bohatstva, ktorá závisí od parametrov p , d a α [39].

1. Ak $\alpha > (1-p)/(p+d)$, potom distribúcia bohatstva je exponenciálna a spoločnosť je superférová.
2. Ak $\alpha < (1-p)/(p+d)$, distribúcia má mocninný charakter s exponentom
$$\gamma = 2 + \frac{(1+\alpha)(p+d)}{1-p-\alpha(p+d)}.$$

Ak v druhom prípade $\alpha = -1$, potom $\gamma = 2$. To je práve hranica prechodu medzi férovou ($\gamma > 2$, $\alpha > -1$) a neférovou spoločnosťou ($\gamma < 2$, $\alpha < -1$). Tento bod prechodu nezávisí od hodnôt p a d , čo naznačuje, že ani ak sa veľká časť nového majetku rozdeľuje férovo, nezabráni to neférovej distribúcii kapitálu v zanikajúcej spoločnosti. Tento záver nijako neprekvapuje, pretože je v súlade s našou skúsenosťou. Pripomínam, že spoločnosť považujeme za zanikajúcu ak $\alpha < 0$, za rozvíjajúcu sa ak $\alpha > 0$ a stabilnú ak $\alpha = 0$. Treba mať na pamäti, že férovosť a rozvoj sú dva rôzne pojmy.

Vyššie opísané modelovanie toku kapitálu v spoločnosti je zrejme príliš zjednodušené. Vnímový čitateľ si pravdepodobne všimol, že okrem iných zjednodušení sme mlčky predpokladali aj neohraničenú dobu života jednotlivca. Ale aj napriek tomu, že modely majú od reality ďaleko, predsa len ukazujú, ako možno teóriu sietí použiť v praxi. Ak by sme chceli realistickejší model, museli by sme citlivejšie zvážiť, ktoré vplyvy sú pre distribúciu kapitálu dôležité a ktoré nie. Ak by sme však zahrnuli príliš mnoho vplyvov, možno by sme dostali model, integro - diferenciálnu rovnicu, ktorú by bolo ťažké analyticky vyriešiť. Pri tvorbe modelu musíme vždy brať do úvahy dva aspekty. Model by mal byť čo najrealistickejší, ale, ak vôbec chceme získať nejaké výsledky, aj riešiteľný. Samozrejme, môžeme urobiť počítačový model a numericky skúmať, ako sa správa. Analytické výsledky sú však vždy cenné, pretože sú presné.

Teória sietí so zrýchleným rastom modeluje situácie, ktoré sú často krát bližšie k realite ako napríklad BA model a jeho variácie. Tieto modely na základe princípu preferenčného pripájania uzlov vysvetľujú vznik bezškálových sietí. Nevysvetľujú však ako sa v sieťach vytvára ďalšia dôležitá vlastnosť, a to je hierarchická organizácia uzlov. Nepomôže nám ani teória sietí so zrýchleným rastom. V piatej časti preto poviem o hierarchii niečo viac.

5 Hierarchické siete

V kapitole venovanej reálnym sieťam sme spomenuli, že ich bezškálová štruktúra je často doprevádzaná hierarchickou štruktúrou [15, 16, 17]. Z predošlých častí tejto práce vieme, že znakom bezškálovej siete je mocninná distribúcia stupňa uzlov

$$P(k) \propto k^{-\gamma} \quad (64)$$

Znakom hierarchického usporiadania je zasa mocninná distribúcia priemerných klasterizačných koeficientov uzlov majúcich stupeň k [15], teda

$$C(k) \propto k^{-\delta} \quad (65)$$

kde δ je nový škálovací exponent.

Základná otázka, ktorú si odborníci v teórii sietí musia položiť je: Ak sú bezškálové a zároveň hierarchické siete v realite také časté, musí existovať jednoduchý a prirodzený dynamický proces, ktorým sieť rastie, a ktorý vytvára hierarchickú bezškálovú štruktúru. Riešeniu tejto otázky sa, okrem iných, venovali aj autori prác [15, 16, 17]. Ako jedni z prvých si ju položili Ravászová a Barabási [15]. V práci [15] navrhli proces rastu siete, ktorý vytvára finálnu bezškálovú hierarchickú štruktúru. Numerickou simuláciou sa im podarilo ukázať, že hierarchická a zároveň bezškálová sieť rastie tak, že sa v každom časovom okamihu k sieti nepripája len jeden uzol, ako je tomu napríklad v BA modeli, ale viacej uzlov tvoriacich istú pevnú štruktúru.

Ravászovej a Barabásiho (RB) proces si najlepšie objasníme pomocou obrázku. Na obrázku 10 vidíme, že rast siete začína z počiatočného päť uzlového modulu, ktorý tvorí kompletný graf (obr. 10 a). Tento modul sa v nasledujúcom kroku štyrikrát skopíruje a jeho kópie sa pripoja k pôvodnému modulu tak, že ich obvodové uzly sa pripoja každý jednou hranou k centrálnemu uzlu pôvodného modulu (obr. 10b). Nová,

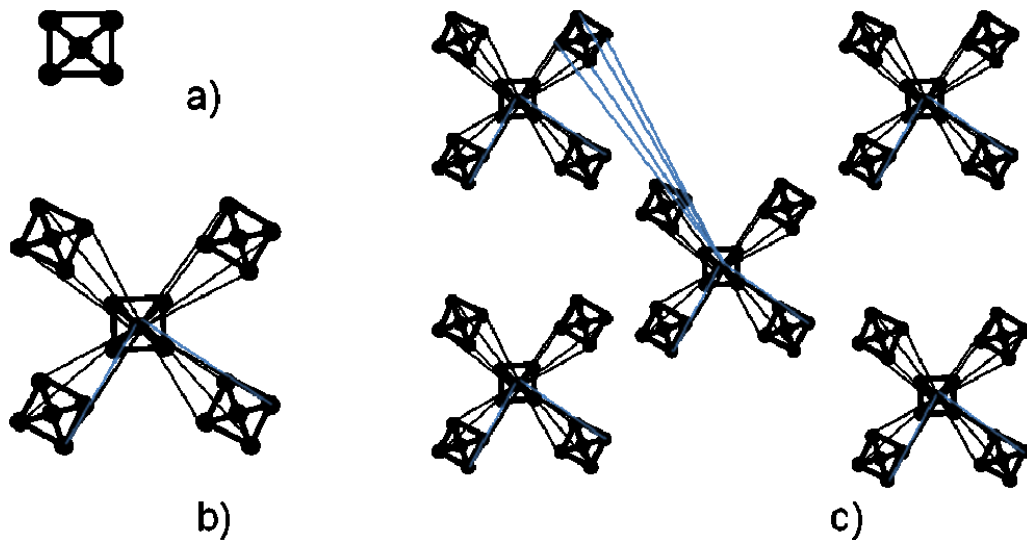
25 uzlová štruktúra sa opäť štyrikrát skopíruje a jej obvodové uzly sa pripoja k centrálnemu uzlu pôvodného malého päť uzlového modulu (obr. 10c). Centrálny päťuzlový modul kópií sa nepripája. Tento proces opakujeme tak dlho, pokiaľ sieť nie je dostatočne veľká na to, aby sa mohli zmerať jej vlastnosti, napríklad distribúciu stupňa uzlov a distribúciu priemerných klasterizačných koeficientov uzla stupňa k . Obe distribúcie spĺňajú zákony (64) a (65) s exponentami $\gamma = 2.161$ a $\delta = 1$, teda sieť, ktorá vznikne RB procesom rastu, je hierarchická a bezškálová zároveň.

Ravászová a Barabási, vedomí si toho, že pravidelné pripájanie nejakého modulu k sieti sa v realite príliš nepozoruje, utvorili ďalšiu, náhodnú verziu svojho modelu. Sieť opäť začína rásť z rovnakého päť uzlového počiatočného modulu. Podobne ako v predošlom prípade sa tento modul v ďalšom kroku štyrikrát skopíruje. Zlomok p uzlov týchto kópií sa pripojí preferenčným spôsobom k uzlu centrálnemu modulu (obrázok 11). Sieť, ktorá vznikne takýmto procesom, je opäť sieťou hierarchickou a bezškálovou, čo znamená, že jej vlastnosti sú vyjadrené zákonmi (64) a (65). Škálovacie exponenty γ a δ však závisia od p tak, že oba s rastúcim p klesajú. Závislosti $\gamma(p)$ a $\delta(p)$ sú zatiaľ známe len z numerických simulácií, ich analytické vyjadrenie nepoznáme.

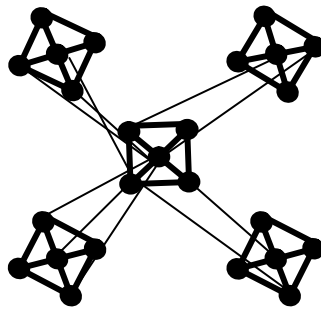
Ďalším autorom, ktorý sa zaoberá vznikom hierarchickej organizácie uzlov v sieťach, je Alexei Vásquez [16]. Ako príklad mu slúži www sieť a citačná sieť. O štruktúre sietí však Vásquez uvažuje trochu iným spôsobom ako Ravászová a Barabási [15]. Zamýšľal sa nad tým, ako surferi, blúdiaci po sieti, objavujú a zároveň tvoria a modifikujú jej štruktúru. Vásquez si položil otázku: Môžu surferi len pomocou náhodného blúdenia po sieti zistiť ako sieť vyzerá? Skôr ako sa pustíme do riešenia tejto otázky, definujme si, čo rozumieme pod www sieťou a citačnou sieťou.

Pod www sieťou myslíme takú sieť, v ktorej uzlami sú www stránky, a smerovaná hrana vznikne vtedy, keď sa na stránke nachádza hypertextový odkaz na nejakú inú stránku. V druhej, citačnej sieti, uzlom je článok. Tento uzol spájajú smerované hrany so všetkými uzlami (článkami), ktoré sú v ňom citované. Matematickou reprezentáciou oboch týchto sietí je orientovaný graf.

Surfer, ktorý surfuje v takýchto sieťach, ich štruktúru spočiatku vôbec nepozná. Blúdením po sieti ju pre seba objavuje, ale zároveň aj aktívne dotvára. Ak surfujeme napríklad po www stránkach, robíme to v podstate dvoma spôsobmi. Niekedy použijeme vyhľadávací program, pomocou ktorého skočíme náhodne na nejakú stránku. Ak sa nám stránka páči a jej obsah je zaujímavý, môžeme sa v ďalšom kroku rozhodnúť sledovať jeden z hypertextových odkazov na nej umiestnených, aby sme sa o problematike, ktorej sa stránka venuje, dozvedeli viac. Alebo je pre nás stránka, na ktorej sa práve nachádzame nezaujímavá, a v ďalšom kroku jednoducho náhodne skočíme na inú stránku.



Obr. 10. Proces rastu hierarchickej bezškálovej siete podľa autorov Ravászovej a Barabásiho [15]. Začína sa z kompletného grafu o piatich uzloch a). V nasledujúcom kroku sa urobia štyri identické kópie základného modulu a uzly týchto kópií (okrem stredových) sa pripoja každý jednou hranou k centrálnemu uzlu pôvodného modulu b). Takto vzniknutý 25 uzlový modul sa opäť štyrikrát skopíruje a celý proces sa opakuje, lebo každý uzol každej kópie (okrem stredových uzlov a stredného päťuzlového modulu kópie) sa pripojí jednou hranou k centrálnemu uzlu základného modulu. Na obrázku c) sú tieto hrany pre prehľadnosť väčšinou vynechané. Celý proces podobným spôsobom pokračuje ďalej.



Obr. 11. Ravász – Barabási model hierarchickej bezškálovej siete s náhodnosťou. Sieť rastie podobne ako jej deterministická verzia. Rozdiel spočíva v tom, že v každom kroku sa pripojí len zlomok p novo pridaných uzlov.

Podobne, keď študujeme novú odbornú problematiku, hľadáme články, ktoré by bolo dobré si prečítať. Prvú prácu k danej téme si buď vyhľadáme pomocou vyhľadávacieho programu, alebo nám ju niekto odporučí (náhodný skok na uzol). Ďalšie práce získame buď tak, že si stiahneme niektoré z tých článkov, ktoré daný článok citoval (sledovanie linky siete), alebo opäť použijeme vyhľadávací program či odporúčenie kolegu, ktorý nám ponúkne nový náhodný výber prác.

Po sieti nemusí blúdiť len jeden surfer, môže ich byť aj viacero. Surferi tým, že po sieti blúdia, ju aj modifikujú, tvoria. Z času na čas totiž každý z nich pridá k už existujúcim uzlom nejakú novú hranu. Aby som bola konkrétnejšia, pri blúdení po www sieti nás niektorá stránka môže natoľko zaujať, že na svoju stránku si na ňu pridáme hypertextový odkaz. Poprípade v citačnej sieti, ak je nejaký článok dobrý a užitočný pre náš výskum, citujeme ho v našom vlastnom článku.

Vásquez vyššie popísaný proces formalizoval takto: položil si otázku s akou pravdepodobnosťou bude istý uzol i navštívený jedným surferom. Táto pravdepodobnosť je súčtom pravdepodobnosti náhodného skoku na uzol i a súčtom pravdepodobnosti toho, že sa k uzlu i dostaneme po hrane od niektorého z jeho susedov j :

$$v_i = \frac{1 - q_e}{N} + q_e \sum_j J_{ij} \frac{v_j}{k_j^{out}} \quad (66)$$

V rovnici (66) N je počet uzlov v sieti, J_{ij} predstavuje maticu susednosti (viď časť o teórii grafov) a k_j^{out} je počet hrán vychádzajúcich z uzla j . Veličina q_e označuje pravdepodobnosť toho, že surfer, nachádzajúci sa na nejakom uzle, sa rozhodne pokračovať putovaním pozdĺž hrany. S pravdepodobnosťou $1 - q_e$ surfer skočí na iný, náhodne vybraný uzol.

Keďže matica susednosti je zvyčajne veľká a riedka, rovnica (66) sa rieši pomocou aproximácie

$$v_i = \frac{1 - q_e}{N} + q_e \Theta k_i^{int} \quad (67)$$

Tu Θ predstavuje priemernú hodnotu pravdepodobnosti, že uzol, ktorý má hranu smerujúcu k uzlu i je navštívený surferom, pričom Θ je podielom priemernej pravdepodobnosti $\langle v \rangle$ a priemerného počtu hrán, ktoré smerujú von z uzla [16]:

$$\Theta = \frac{\langle v \rangle}{\langle k^{out} \rangle} \quad (68)$$

Ako sme už spomínali, surfer pri blúdení po sieti siet' aj dotvára. Občas pridá k nejakému uzlu novú hranu. Preto sa s časom nemení len počet objavených uzlov

$$\frac{\partial N}{\partial t} = v_a \quad (69)$$

ale aj počet hrán v sieti

$$\frac{\partial e}{\partial t} = v_s q_v \langle v \rangle N \quad (70)$$

V predošlých rovniciach ν_a je počet uzlov pridaných do siete za jednotku času, ν_s je počet surferov a q_v pravdepodobnosť toho, že jeden surfer pridá do siete novú hranu. Riešením rovníc (69) a (70) sú závislosti

$$N = \nu_a t \quad (71)$$

$$e = \nu_s q_v \langle \nu \rangle N t \quad (72)$$

Priemerný počet hrán, ktoré do uzlov siete vchádzajú je taký istý ako počet tých, ktoré z nich vychádzajú

$$\langle k^{out} \rangle = \langle k^{in} \rangle = \frac{e}{N} = \frac{\nu_s q_v \langle \nu \rangle N}{\nu_a} \quad (73)$$

kde e je opäť celkový počet hrán a N počet uzlov v sieti. Z rovníc (68) a (73) môžeme Θ vyjadriť pomocou merateľných veličín ako

$$\Theta = \frac{\nu_a}{q_v \nu_s N} \quad (74)$$

Keď toto všetko vieme, vieme zistiť aká bude výsledná štruktúra siete? Bude hierarchická a bezškálová? Inými slovami, vieme analyticky vypočítať distribúciu stupňa uzlov a distribúciu priemerných klasterizačných koeficientov? Vásquez vo svojej práci [16] dokazuje, že vzniknutá sieť naozaj je bezškálová a hierarchická zároveň.

Aby sme to mohli analyticky ukázať, potrebujeme najprv vypočítať, aká je pravdepodobnosť toho, že stupeň uzla, majúceho stupeň k^{in} vzrastie o jednotku. Táto pravdepodobnosť je daná vzťahom

$$A(k^{in}) = q_v \nu(k^{in}) \quad (75)$$

teda súčinom pravdepodobnosti q_v pridania hrany surferom a pravdepodobnosti, že jeden surfer navštívi uzol so stupňom k^{in} . Ak do (75) dosadíme (67) a (74), dostaneme

$$A(k^{in}) = \frac{1}{N} \left[q_v (1 - q_e) + q_e \frac{\nu_a}{\nu_s} k^{in} \right] \quad (76)$$

Pomocou (76) je potom možné vypočítať distribúciu stupňa uzlov. Môžeme napísať tzv. rovnicu pomerov (rate equation), vzťah, ktorý hovorí, ako sa počet uzlov daného stupňa mení s časom [10, 16]

$$\frac{\partial n_{k^{in}}}{\partial t} = \nu_s A_{k^{in}-1} n_{k^{in}-1} - \nu_s A_{k^{in}} n_{k^{in}}, \quad k^{in} \neq 0 \quad (77)$$

Prvý člen rovnice (77) znamená, že uzol so stupňom k^{in} dostaneme tak, že sa k uzlu so stupňom $k^{in} - 1$ pripojí jeden koniec hrany. Počet uzlov so stupňom k^{in} sa naopak zmenší, ak sa k nejakému uzlu s takýmto stupňom pripojí koniec hrany. Toto popisuje druhý člen rovnice (77). Ak rovnicu (77) upravíme a pravú stranu prepíšeme pomocou derivácie (čo pre dostatočne veľké siete môžeme urobiť [10, 33]), dostaneme

$$\frac{\partial n_{k^{in}}}{\partial t} = -v_s \frac{\partial A_{k^{in}} n_{k^{in}}}{\partial k^{in}} \quad (78)$$

Rovnicu (78) vyriešime, ak použijeme vzťah (76) a predpoklad konečnej stacionárnej distribúcie stupňa uzlov $P_{k^{in}}$. Potom $n_{k^{in}} = NP_{k^{in}}$ a z (78) po úpravách dostaneme

$$P_{k^{in}} C_1 = [C_2 - C_3 k^{in}] \frac{\partial P_{k^{in}}}{\partial k} \quad (79)$$

kde C_1, C_2, C_3 sú konštanty. Riešením rovnice (79) je vzťah [16]

$$P_{k^{in}} \propto (k^{in})^{-\gamma}, \quad \gamma = 1 + \frac{1}{q_e} \quad (80)$$

Sieť, ktorú dostaneme Vásquezovým procesom prehládavania a dotvárania má teda bezškálovú štruktúru. Škálovací exponent γ závisí od veličiny q_e .

Čo však možno povedať o hierarchickej organizácii uzlov? Vásquez analyticky dokázal [16], že

$$C_k \propto k^{-\delta}, \quad \delta = 1 \quad (81)$$

teda, vzniknutá sieť je aj hierarchická.

Dá sa to ukázať takýmto spôsobom: Klasterizačný koeficient uzla i je daný vzťahom (3). To, čo sa s časom mení, je počet hrán medzi susedmi uzla i [16]:

$$\frac{\partial e_i}{\partial t} = q_v (q_e \Theta k_i^{in} + q_e v_i) \quad (82)$$

kde e_i je počet hrán medzi susedmi uzla i a ostatné veličiny znamenajú to isté, čo v predchádzajúcich vzťahoch. Použitím rovníc (67, 74, 76) a so zohľadnením toho, že $\frac{\partial k_i^{in}}{\partial t} = A(k_i^{in})$ možno za predpokladu, že k je dostatočne veľké, odvodiť

$$\frac{\partial e_i}{\partial t} \approx (1 + q_e) \frac{\partial k_i}{\partial t} \quad (83)$$

Integrovaním tejto rovnice za predpokladu, že $e(k^{in} = 0) = 0$ dostaneme klasterizačný koeficient C_k

$$C_k \approx \frac{2(1 + q_e)}{k} \quad (84)$$

čo nie je nič iné ako (81).

Vázquezom popísané blúdenie po sieti však nie je procesom rastu siete, aj keď si možno predstaviť, že s ním úzko súvisí. Novoobjavený uzol môžeme považovať za uzol práve pridaný do siete. Neskôr uvidíme, že model rastúcej siete, v ktorom sa kombinujú lokálne zákony pripájania hrán a pripájanie uzlov, možno pretransformovať na Vásquezovu sieť a analyticky vyriešiť [17].

6 Rôzne aplikácie teórie dynamických sietí

V tejto časti ukážem niekoľko aplikácií teórie dynamických sietí v jazykovede, biológii aj informatike. Budem uvažovať aj o tom, akým prínosom môže byť táto relatívne nová teória pre umelú inteligenciu.

6.1 Slovná sieť

Teória sietí priniesla zaujímavý pohľad na štruktúru lexikónu ľudského jazyka. Skúsme sa pozrieť na lexikón ako na slovnú sieť. Predstavme si, že slovo je uzol a ak dané slovo nejakým spôsobom interaguje s inými slovami, môžeme povedať, že je s nimi spojené hranou.

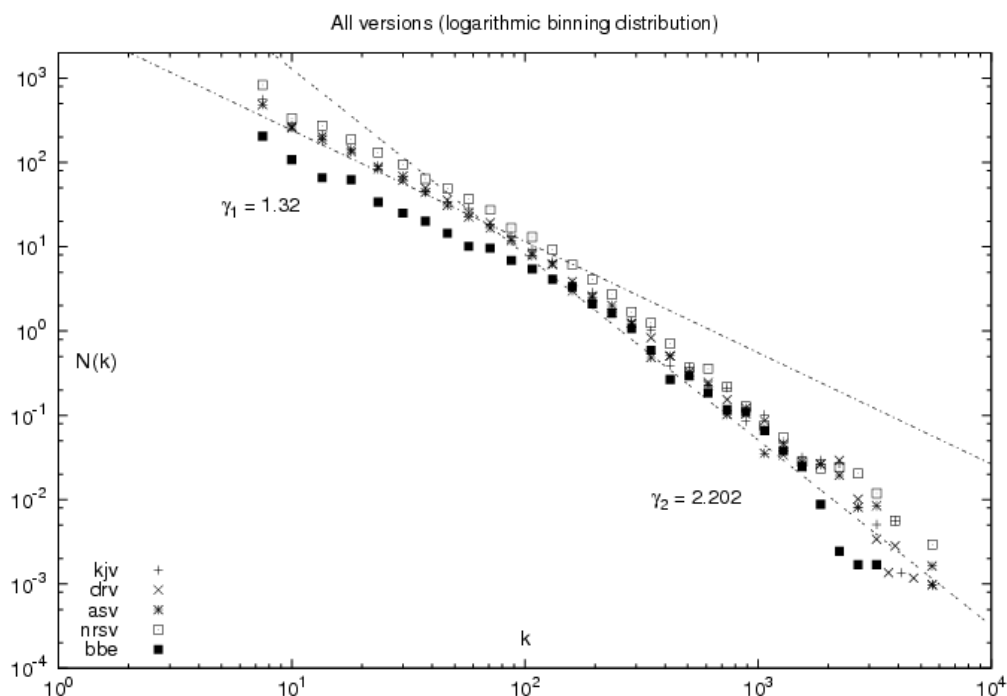
Čo to však znamená „slovo interaguje s iným slovom“? V zásade sú možné dva spôsoby interakcie slov. Prvý spôsob zvyčajne *sémantický* [7] a druhý spôsob *syntaktický* aspekt jazyka [8, 40, 41].

Predstavme si, že interakciu slov definujeme pomocou výkladového slovníku. Slovo, ktoré má v slovníku heslo, nech je uzlom. Všetky slová, ktoré naše slovo opisujú a sú zároveň aj samé slovníkovými heslami, považujeme s daným slovom za spojené hranou. Takýmto spôsobom môžeme vybudovať *sémantickú* slovnú sieť. Jej vlastnosti skúmal Motter a jeho kolegovia [7]. Merali distribúciu stupňa uzlov a ukázali, že s rastúcim stupňom klesá exponenciálne. Takáto slovná sieť (word web) teda nie je bezškálová.

Ak je však interakcia slov definovaná susedstvom vo vete, dostaneme *syntaktickú* slovnú sieť. V nej slovo opäť predstavuje uzol a všetky slová, ktoré sa vo vetách vyskytujú ako jeho najbližší susedia, sú s týmto uzlom spojené hranou. Takáto sieť sa teda buduje na základe veľkej databázy textov dostatočne dobre reprezentujúcich skúmaný jazyk. Sieť sa tvorí takto: zvolíme nejaké slovo a vyhľadávame ho v textoch databázy. Zároveň vyhľadáme aj slová, ktoré sú najbližšími susedmi daného slova vo vetách, kde sa naše slovo vyskytuje. Potom to isté urobíme so susedmi zvoleného slova.

Ferrer a Solé analyzovali vlastnosti syntaktickej slovnej siete zostrojenej na základe textov anglického národného korpusu [8]. Distribúcia stupňa uzlov tejto siete

naznačuje, že ju možno považovať za bezškálovú (5.1) avšak s tým rozdielom, že pozorujeme dva škálovacie režimy (obrázok 12) s dvoma rôznymi škálovacími exponentami. Pre uzly s menším stupňom škálovací exponent $\gamma_1 = 1.5$ a pre uzly s vyšším stupňom $\gamma_2 = 2.7$. Druhá hodnota škálovacieho exponentu je blízka (ale nie totožná) s hodnotou $\gamma_{BA} = 3.0$.



Obr. 12. Distribúcia stupňa uzlov pre syntaktickú slovnú sieť, ktorá bola skonštruovaná na báze anglických prekladov Biblie. Niektoré preklady sú staršie (Douay Rheims verzia, djv, rok vydania 1582; King James verzia, kfv, 1611), iné moderné (American Standard verzia, asv, 1901; Basic English verzia, bbe, 1941; New Revisited Standard verzia, nrsv, 1989). bbe je špeciálny prípad, lebo text bol umelo zjednodušaný. Siete majú tento počet uzlov: drv – 11423, kfv – 11624, asv – 10105, nrsv – 14985, bbe – 4961.

V snahe vysvetliť, prečo v syntactickej slovnej sieti existujú dva škálovacie režimy, vzniklo niekoľko prác [40, 41]. My sme experimentovali s rôznymi anglickými prekladmi Biblie [42] a rôznymi anglickými textami z projektu Gutenberg [43]. Na obrázku 12 vidíme distribúciu stupňa uzlov biblických syntaktických sietí. V oboch prípadoch sme pozorovali podobný efekt ako Ferrer a Solé, len sme namerali trochu iné škálovacie exponenty γ_1 a γ_2 . Domnievame sa, že to môže byť spôsobené horšou štatistikou, pretože naša sieť má desať krát menej uzlov ako sieť Ferrera a Solého. Počet slov v Biblii je totiž silne obmedzený a je mnoho takých slov, ktoré sa v Biblii vôbec nevyskytujú. V tomto smere Biblia pripomína špecializovaný odborný text.

Preto je dosť dobre možné, čo potvrdzuje aj analýza textov z projektu Gutenberg [43], že $\lambda_1 = 1.5$ a γ_2 má hodnotu len o niečo menšiu ako $\gamma_{BA} = 3$.

Ferrer a Solé sa domnievajú, že existencia dvoch škálovacích režimov v distribúcii stupňa uzlov je spôsobená rozdielnou dynamikou periférie a jadra jazyka [8]. Pod jadrom jazyka rozumieme slová, ktoré tvoria základ jazyka používaný všetkými príslušníkmi danej jazykovej skupiny, bez ohľadu na vek, pohlavie, vzdelanie, či iné aspekty. Jadro jazyka obyčajne obsahuje okolo 10^4 slov, pomocou ktorých možno vyjadriť takmer všetko. V slovnej sieti sú slová z jadra jazyka väčšinou tie, ktoré sú s inými slovami dobre prepojené. Predstavujú teda obyčajne tie uzly slovnej siete, ktoré majú vyšší stupeň. Slová z jazykovej periférie sú jednak slangové výrazy, jednak odborná terminológia, jednak nové slová, teda tie, ktorým nemusí rozumieť každý. V slovnej sieti tieto slová majú zvyčajne menší stupeň, pretože sa vo vetách používajú v obmedzenom kontexte. Podľa Ferrera a Solého dynamika jadra jazyka je v podstate ustálená a možno ju modelovať pomocou BA procesu, zatiaľ čo pre jazykovú perifériu to neplatí.

S týmto vysvetlením nesúhlasia Dorogovtsev a Mendes [41]. Autori práce [41] uvažujú takto: Kontext, v ktorom sa slová používajú, sa s časom mení. Väčšinou sa obohacuje. Vďaka tomu sa nové hrany do slovnej siete nepridávajú len s novými uzlami, ale môžu vzniknúť aj medzi slovami, ktoré sú už v slovníku dlhšie. Dorogovtsev a Mendes vytvorili model, ktorého základom je BA model (viď. časť 4.2), obohatený o nový proces tvorby hrán medzi starými slovami. Podobne, ako v BA modeli

1. Na začiatku rastu siete máme pár uzlov pospájaných niekoľkými hranami. O aký počiatkový graf sa jedná, na tom v podstate príliš nezáleží, lebo vlastnosti veľkej siete počiatkový graf neovplyvní.
2. Nech každú časovú jednotku odkiaľsi z vesmíru priletí jeden uzol a pripojí sa preferenčne m hranami k starým uzlom.
3. V tom istom okamihu vznikne medzi starými uzlami $2ct$, ($0 \leq c \ll 1$) nových hrán. Tieto nové hrany prepoja staré uzly preferenčným spôsobom.
4. Opakujeme od bodu 2. Sieť, ktorej finálnu štruktúru skúmame, musí byť dostatočne veľká, čím dlhšie simulácia beží, tým presnejšie zmeriame charakteristiky siete.

Dorogovtsevov – Mendesov model (DM model) je matematickým vyjadrením vyššie uvedených efektov:

$$\frac{\partial k(s,t)}{\partial t} = (m + 2ct) \frac{k(s,t)}{\int_0^t k(s,t) ds} \quad (85)$$

V rovnici (85) integrál predstavuje normováciu konštantu, teda súčet všetkých stupňov uzlov. $k(s, t)$, tak ako v predošlých podobných rovniciach, predstavuje priemerný stupeň uzla, ktorý došiel do siete v čase s a pozorujeme ho v čase t . m je

počet hrán, ktoré do siete prinesie každý nový uzol a $2ct$ je počet koncov hrán, ktoré prepoja staré uzly preferenčným spôsobom.

DM model (85) sa dá analyticky vyriešiť. Ak preintegrujeme obe strany rovnice cez s v hraniciach od 0 do t ($\int_0^t ds$), dostaneme

$$\int_0^t ds \frac{\partial k(s,t)}{\partial t} = (m + 2ct) \frac{\int_0^t k(s,t) ds}{\int_0^t k(s,t) ds} = m + 2ct \quad (86)$$

Odtiaľ s použitím hraničnej podmienky $k(t,t) = m$ a vzťahu známeho z matematiky [36] $\frac{\partial}{\partial t} \int_0^t k(s,t) ds = k(t,t) + \int_0^t \frac{\partial}{\partial t} k(s,t) ds = 2m + 2ct$ dostaneme

$$\int_0^t k(s,t) ds = 2mt + ct^2 \quad (87)$$

Keď (87) dosadíme do (85) a rovnicu (85) vyriešime, dostaneme výsledok [41]

$$k(s,t) = \left(\frac{t}{s}\right)^{\frac{1}{2}} \left(\frac{2m + ct}{2m + cs}\right)^{\frac{3}{2}} \quad (88)$$

čo vedie na distribúciu ako na obrázku 12 [41]. Pre uzly s indexom $s \ll t$ (teda pre slová, ktoré majú šancu mať veľký stupeň, lebo prišli do siete skoro) škálovací exponent $\gamma_2^{DM} = 3$ a pre uzly s indexom $s \sim t$ škálovací exponent $\gamma_1^{DM} = 1.5$.

DM model teda zakomponovaním procesu vzniku nových hrán medzi starými uzlami dokázal vysvetliť vznik dvojrežimovej distribúcie stupňa uzlov (obrázok 12). Pozorný čitateľ si možno uvedomil aj to, že rovnica (85) je vlastne rovnicou s časovo závislou atraktivitou (fitness) uzla, ak $A(t) = 2k(s,t)ct$. Je tu však ešte jeden problém. Napriek tomu, že DM model je kvalitatívne úspešný, merania Ferrera a Solého [8], ako aj naše vlastné [42, 43] ukazujú, že medzi experimentálnymi dátami a modelom je istá nepresnosť. V oboch prípadoch nameraný exponent $\gamma_2^{measured}$ v strmšej časti distribúcie stupňa uzlov nedosahuje hodnotu 3, ktorú predikuje DM model. V menej strmej časti distribúcie je to v poriadku a $\gamma_1^{DM} = \gamma_1^{measured} = 1.5$. Usúdila som teda [40], že to nie je náhoda a DM model nie je úplne postačujúcim modelom syntaktickej slovnej siete.

Aby som objasnila rozdiel medzi nameraným a vypočítaným škálovacím exponentom pre uzly s veľkým stupňom, navrhla som obohatiť DM proces rastu siete o ďalší efekt, ktorý by nebol v rozpore s pozorovaným vývojom lexikónu jazyka.

Skúsme spoločne trochu uvažovať. To, čo vyjadruje DM model, totiž, že do lexikónu nielenže pribúdajú nové slová, ale aj staré slová sa dostávajú do nového kontextu, je pravda. Napríklad slovo Boh sa najprv používalo na označenie Slnka, Mesiaca, hromu, alebo iných prírodných úkazov. Dnes má toto slovo zložitý psychologicko – nábožensko – filozofický obsah. Počas vývoja slovníka (rastu slovnej siete) sa tento pojem začal používať v takých kontextoch (vznikli nové hrany medzi starými slovami), ktoré s pôvodným významom majú už máločo spoločné. Avšak slová nielen získavajú, ale aj strácajú kontext. Napríklad slovo počítač ešte pred pár desiatkami rokov znamenalo pomenovanie zariadenia zaberajúceho celú veľkú halu. Dnes si pod týmto pojmom skôr predstavíme malé PC, poprípade notebook. Veta „Odnies mi počítač na stôl, prosím.“ dnes nikoho neprekvapí. Ešte nedávno by sa však váš asistent takejto prosbe veľmi začudoval. Ale príkaz „Nájdite veľkú miestnosť pre umiestnenie nového počítača“ by ho vôbec neprekvapil. Slovo „počítač“ sa v minulosti v kontexte „na stole“ nepoužívalo. Používalo sa v iných kontextoch, ktoré už dnes prestali platiť. Vzhľadom na podobné úvahy som pôvodný DM proces rastu siete modifikovala takto:

1. Na začiatku máme pár uzlov pospájaných niekoľkými hranami. O aký počiatkový graf sa jedná na tom v podstate príliš nezáleží, lebo vlastnosti veľkej siete počiatkový graf príliš neovplyvní.
2. Nech každú časovú jednotku odkiaľsi z vesmíru priletí jeden uzol a pripojí sa preferenčne m hranami k starým uzlom.
3. V tom istom okamihu vznikne medzi starými uzlami $2ct$, ($0 \leq c < 1$) nových hrán. Tieto nové hrany prepoja staré uzly preferenčným spôsobom.
4. Simultánne s javmi 2. a 3. sa náhodne zvolí m_r starých uzlov, od ktorých sa odpojí koniec jednej hrany a prelinkuje sa preferenčne k inému uzlu.
5. Opakujeme od bodu 2. Sieť, ktorej finálnu štruktúru skúmame, musí byť dostatočne veľká, takže čím dlhšie simulácia beží, tým presnejšie zmeriame charakteristiky siete.

Model matematicky popisujúci horeuvedené procesy, som nazvala *modelom s prelinkovaním hrán* (edge rewiring, teda RW model). Matematicky ho možno reprezentovať integro – diferenciálnou rovnicou

$$\frac{\partial k(s,t)}{\partial t} = (m + 2ct + m_r) \frac{k(s,t)}{\int_0^t k(s,t) ds} - \frac{m_r}{t} \quad (89)$$

V rovnici (89) prvý člen na pravej strane popisuje preferenčné pripájanie $m + 2ct + m_r$ koncov hrán. m z nich prinesie nový uzol, ct hrán ($2ct$ koncov hrán) sa vytvorí medzi starými uzlami a m_r hrán sa prepojí. Druhý člen rovnice (89) vyjadruje odpojenie jedného konca hrany od m_r náhodne zvolených uzlov.

Proces presmerovania hrán neovplyvňuje súčet všetkých stupňov uzlov, a teda integrál v menovateli rovnice (89) je rovnaký ako v DM modeli (87). Ak ho dosadíme do rovnice (89) a takto upravenú rovnicu vyriešime, dostaneme

$$k(s, t) \propto \left(\frac{t}{s}\right)^{\frac{m+m_r}{2m}} \left(\frac{2m+ct}{2m+cs}\right)^{2-\frac{m+m_r}{2m}} \quad (90)$$

Toto riešenie vedie na takú istú distribúciu stupňa uzlov ako v DM modeli, ale s rozdielnymi škálovacími exponentami [40]. Ak $s \ll t$ z rovnice (90) vypočítame $\gamma_2^{RW} = 2 + \frac{m-m_r}{2m}$. V prípade $s \sim t$, $\gamma_1^{RW} = 1.5$. Ak počet náhodne vybraných starých uzlov, od ktorých sa odpojí jeden koniec hrany, je menší ako celkový počet hrán prinesených novými uzlami, škálovací exponent pre uzly s veľkým stupňom bude menší ako $\gamma_2^{DM} = 3$ a väčší ako 2. V menej strmej časti distribúcie sa efekt prepájania hrán na hodnote škálovacieho exponentu neprejaví.

Naše merania biblickej slovnej siete ukazujú [42], že ak $\gamma_2^{RW} = 2.13$ a ak novo prichádzajúce slovo má v priemere okolo 10 spojení so starými slovami, potom $m_r \approx 7.7$. Vidíme teda, že RW model je celkom realistický. Efekt prelinkovania hrán, ktorý bol pridaný k DM modelu, vysvetlil systematickú chybu medzi predpoveďou teórie a dátami. Navyše fakt, že slovo s časom mení kontext, patrí k prirodzeným javom vo vývoji lexikónu ľudského jazyka. RW model tak presnejšie vystihuje namerané dáta a odstraňuje nedostatky predošlých modelov.

6.2 Funkčná sieť mozgu v konkrétnom experimente

Ako som už spomenula v tretej časti, funkčné siete mozgu sú príkladom ad hoc sietí. V týchto sieťach za uzol pokladáme voxel a hrana medzi dvoma voxelmi existuje vtedy, keď sú aktivity voxelov časovo skorelované [3].

Počas svojho pobytu na Department of Computer Science, Otago University, Dunedin, Nový Zéland, som spolu s kolegyňami Lubicou Beňuškovou a Liz Franz, mala možnosť študovať funkčnú sieť mozgu, extrahovanú z nameraných dát [44]. Niekoľko meraných osôb sa podieľalo na experimente, pri ktorom mali striedavo rytmicky ťukať palcom a ukazovákom pravej a ľavej ruky. Rytmus ťukania určoval zvukový signál, frekvencia ťukania približne zodpovedala jednému ťuknutiu palcom a ukazovákom za sekundu.

Merali sme štyri zdravé osoby. Dve z nich boli ženy (subjekt 1 a 3, vid' tabuľka 1, 2) vo veku 55 a 46 rokov. Dvaja muži (subjekt 2 a 4) mali 60 a 23 rokov.

Pred zberom fMRI dát je vždy potrebné, aby mozog relaxoval. Požiadali sme preto frekventantov, aby chvíľku pokojne odpočívali. Potom nasledovalo prvé, šesťnásťsekundové referenčné meranie, počas ktorého merané osoby neplnili žiadnu úlohu. Skúšobné meranie malo len pripraviť našich frekventantov na experiment, a preto sme ho pri extrakcii funkčných sietí nebrali do úvahy. Po tomto zahrievacom kole nasledovalo 6 dvadsaťsekundových meracích cyklov, počas ktorých osoby plnili

danú úlohu (task cyklus), striedaných šiestimi dvadsaťsekundovými cyklami oddychu (rest cyklus). FMRI signál mozgu sme merali v jednotlivých voxeloch. Voxely sú uložené v ôsmych vrstvách (obrázok 6). Matica voxelov v jednej vrstve má veľkosť 64×64 , teda celkový počet voxelov je $64 \times 64 \times 8 = 32768$. Jeden merací cyklus pozostáva z desiatich meraní. Keďže každý merací cyklus trvá dvadsať sekúnd, je zrejmé, že scanner získa dáta z celého mozgu v priebehu dvoch sekúnd.

Namerané dáta sme potom spracovali nasledovným postupom. V prvom rade sme pre každý subjekt spriemerovali všetky rest aj task merania. Potlačili sme tým šum a zvýraznili signál. Pred spriemernením sa hodnoty signálov v cykloch rest aj task pohybovali v rozmedzí 0 až ± 500 jednotiek. Je známe [45], že rozdiel hodnôt rest a task signálu nie je veľký, v najlepšom prípade to predstavuje päť percent. Preto sme sa zaujímali aj o pozitívne aj o negatívne nadprahové časové korelácie signálov. V ďalšom kroku sme spočítali korelačný koeficient pre 80 miliónov náhodne zvolených párov voxelov. Korelačný koeficient sme určili pomocou vzťahu

$$r(i, j) = \frac{\langle s_i(t) s_j(t) \rangle - \langle s_i(t) \rangle \langle s_j(t) \rangle}{\sigma(s_i(t)) \sigma(s_j(t))} \quad (91)$$

kde $\sigma^2(s(t)) = \langle s^2(t) \rangle - \langle s(t) \rangle^2$, $s(t)$ je aktivita voxelu v čase t (meraná v nejakých jednotkách) a $\langle \cdot \rangle$ predstavuje časové priemerovanie. Do úvahy sme brali iba tie voxely, ktoré mali nadprahovú aktivitu (100 jednotiek v prvom prípade, alebo 200 jednotiek v druhom prípade). Pre obidve nadprahové aktivity sme zvolili prahový korelačný koeficient $r_c = 0.8$. Výsledky pre rôzne prahové aktivity sa v zásade v ničom nelíšia, preto sme ďalšie hodnoty prahových aktivít nepoužili. Skúmali sme však, ako bude vyzerat' extrahovaná funkčná sieť mozgu ak by bol prah aktivity voxelov nulový. V tomto prípade sa výsledky výrazne líšili. Sieť stratila akúkoľvek zaujímavú štruktúru a stala sa náhodnou. Domnievame sa, že napriek spriemerovaniu signálov, malé aktivity voxelov sú vlastne tvorené šumom. Vlastnosti extrahovaných funkčných sietí pre obidve prahové hodnoty aktivít voxelov sú zhrnuté v tabuľkách 1 a 2.

Aktivitu voxelov sme považovali za významne skorelovanú, ak vypočítaná hodnota korelačného koeficientu bola väčšia ako $r_c = 0.8$ (korelácia), alebo menšia ako $r_c = -0.8$ (antikorelácia). Štatistické a topologické charakteristiky siete sme vypočítali s pomocou voľne dostupného nástroja na analýzu sietí Network Workbench [34].

Tabuľka 1. Vlastnosti funkčnej siete mozgu extrahovanej z nameraných dát. Prah aktivity je 100 jednotiek, korelačný prah je $r_c = 0.8$. N je počet uzlov siete, C predstavuje klasterizačný koeficient siete, L je separácia uzlov a γ je škálovací exponent mocninnej časti distribúcie stupňa uzlov.

| Subject | N | C | | L | | γ | |
|---------|------|-------|-------|------|------|----------|------|
| | | task | rest | task | rest | task | rest |
| 1 | 5530 | 0.038 | 0.038 | 5.12 | 5.10 | 1.99 | 2.04 |
| 2 | 5415 | 0.047 | 0.047 | 4.50 | 4.49 | 1.44 | 1.11 |
| 3 | 5720 | 0.047 | 0.047 | 4.31 | 4.04 | 1.39 | 1.39 |
| 4 | 5535 | 0.038 | 0.036 | 5.12 | 5.13 | 2.26 | 2.23 |

Tabuľka 2. Vlastnosti funkčnej siete mozgu extrahovanej z nameraných dát. Prah aktivity je 200 a korelačný prah je taký istý ako v predošlej tabuľke, $r_c = 0.8$. Ostatné veličiny znamenajú to isté ako v predchádzajúcej tabuľke.

| Subject | N | C | | L | | γ | |
|---------|------|-------|-------|------|------|----------|------|
| | | task | rest | task | rest | task | rest |
| 1 | 4260 | 0.036 | 0.038 | 5.41 | 5.42 | 2.16 | 2.08 |
| 2 | 4420 | 0.047 | 0.047 | 4.74 | 4.79 | 1.54 | 1.33 |
| 3 | 5030 | 0.047 | 0.047 | 4.37 | 4.12 | 1.36 | 1.34 |
| 4 | 4265 | 0.040 | 0.038 | 5.41 | 5.40 | 2.05 | 2.38 |

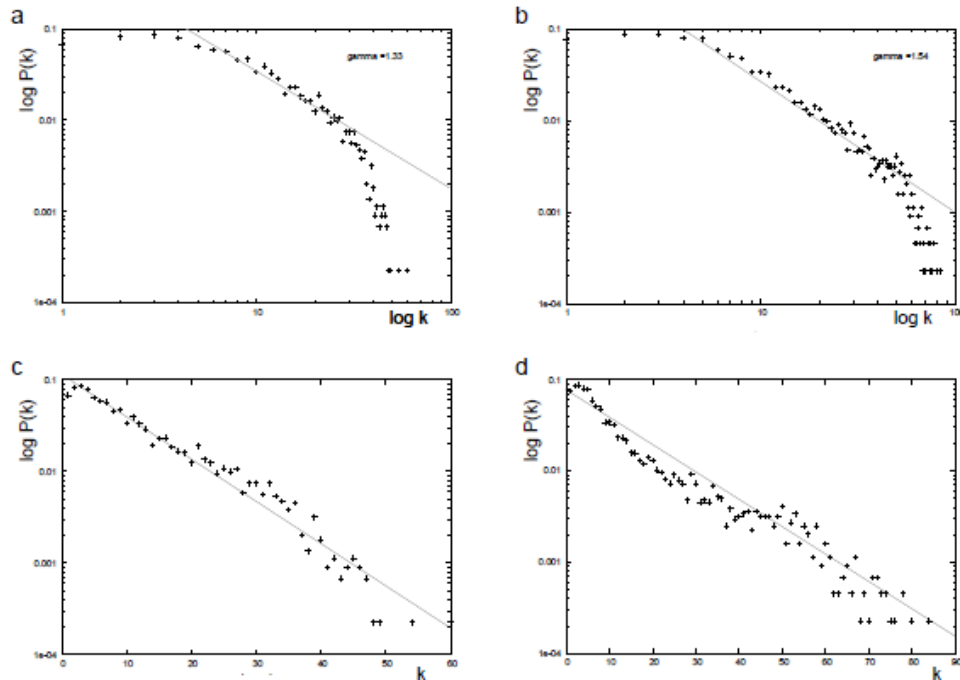
Aby sme zistili, či extrahované siete sú sieťami malého sveta, vypočítali sme klasterizačný koeficient a separáciu uzlov náhodného grafu s rovnakým počtom uzlov a s rovnakým priemerným stupňom uzla ako majú funkčné siete:

$$C_{random} = \frac{\langle k \rangle}{N} \quad (92)$$

$$L_{random} \sim \frac{\ln(N)}{\ln(\langle k \rangle)} \quad (93)$$

kde N je počet uzlov a $\langle k \rangle$ je priemerný stupeň uzla. Pre siete ekvivalentné našim extrahovaným funkčným sieťam $C_{random} = 10^{-3} - 10^{-4}$ a $L_{random} \approx 4$. Či sieť má vlastnosti siete malého sveta možno určiť pomocou tzv. indexu malého sveta [1]

$$\sigma = \left(\frac{C}{C_{random}} \right) / \left(\frac{L}{L_{random}} \right) \quad (94)$$



Obr. 13. Distribúcia stupňa uzlov vo funkčnej sieti extrahovanej pre subjekt 2. a) zobrazenie v log – log škále pre cyklus rest, b) zobrazenie v log – log škále pre cyklus task, c) zobrazenie v log – linear škále pre cyklus rest, d) zobrazenie v log - linear škále pre cyklus task.

Vo všetkých našich sieťach bol tento index vysoký ($\sigma \sim 10-100$). Z toho usudzujeme, že funkčné siete extrahované z nameraných dát, sú zároveň sieťami malého sveta.

Sú funkčné siete mozgu aj bezškálovými sieťami? Ak sme lineárnu časť log – log zobrazenia distribúcie stupňa uzlov preložili priamkou, dostali sme pre všetky siete škálovacie exponenty s hodnotami v rozmedzí $1.0 < \gamma < 2.5$. Treba však dodať, že nie vždy bola lineárna časť distribúcie dostatočne výrazná. Platí to hlavne pre siete získané v rest cykle. Z tabuliek 1 a 2 možno vidieť, že hodnoty škálovacích exponentov pre prípad rest a task sa pre jednotlivé merané osoby príliš nelíšia. Rozdiely však pozorujeme medzi rôznymi osobami. Môže to byť len náhoda, avšak je dosť dobre možné, že pre tieto individuálne rozdiely existuje nejaký, zatiaľ nezistený, dôvod. Na to, aby sme ho mohli nájsť, potrebujeme väčší štatistický súbor meraní.

Fakt, že hodnoty škálovacích exponentov v rest a task cykle sa nelíšia, potvrdil aj Eguíluz a jeho spolupracovníci [26]. Zistil, že rozdiely neexistujú ani vtedy, keď je kognitívna úloha náročnejšia (počúvanie hudby).

Ak sa lepšie pozrieme na obrázok 13, vidíme, že tvar distribúcie stupňa uzlov sa pre rest a task cyklus dosť líši. Hoci na obrázku sú zobrazené len distribúcie jedného subjektu, rozdiel je viditeľný vo všetkých štyroch prípadoch. Spočíva v tom, že lineárna časť distribúcie je výraznejšia v prípade task cyklu ako v prípade rest cyklu

(obrázok 13a, b). Znamená to, že bezškálová štruktúra funkčnej siete sa zvyrazňuje, ak mozog pracuje na nejakej úlohe. Je výraznejšia dokonca aj vtedy, ak je táto úloha taká jednoduchá ako rytmické ťukanie prstami. Spomenutý rozdiel možno ešte lepšie pozorovať pri zobrazení v log – linear škále (obrázok 13c, d), kde je závislosť lineárnejšia v cykle rest, čo by naznačilo skôr exponenciálne klesajúcu distribúciu stupňa uzlov.

Na záver konštatujeme, že všetky siete, nezávisle od toho, či boli extrahované z dát v cykle rest alebo task, majú charakter sietí malého sveta. Zdá sa tiež, že funkčné siete majú vyšší stupeň organizácie v štádiu task, čo vidieť z toho, že distribúcia stupňa uzlov v log – log zobrazení má v task perióde výraznejšiu lineárnu časť ako v rest perióde. Aby sme však mohli urobiť presnejšie závery, potrebujeme omnoho väčší štatistický súbor meraní.

Určite by bolo zaujímavé vedieť, aký je vzťah funkčnej a anatomickej neurónovej siete mozgu. Niektoré štúdie dokazujú [46], že neurónová sieť mozgu je síce sieťou malého sveta, ale nie je bezškálovou sieťou. Z hľadiska prác Equíluzza a kol. [26] a van den Heuvela a kol., a tiež Chialva [3], ktorí tvrdia, že funkčné siete mozgu sú bezškálové, je tento výsledok prekvapujúci. Naša práca však ukazuje, že bezškálovosť štruktúry funkčných sietí mozgu, minimálne v perióde rest, možno spochybniť.

Domnievame sa, že ďalší výskum funkčných sietí mozgu, ktorý by študoval funkčné siete mozgu vo vzťahu k jeho anatomickej štruktúre, by mohol do veci vniesť viacej svetla.

6.3 Hierarchia v rastúcej sieti s lokálnymi pravidlami

V kapitole 5 sme hovorili o hierarchických bezškálových sieťach. Rozobrali sme dva procesy, pomocou ktorých vzniká v sieťach hierarchia, a to bez toho, aby sa narušila bezškálová štruktúra. Prvý proces pripájania fixnej štruktúry uzlov je tak trochu umelý [15]. Aj keď ho autori znáhodnili, v skutočnosti len ťažko možno nájsť sieť, ktorá by rástla pripájaním pravidelnej štruktúry uzlov.

Na druhej strane Vásquezov model [16] hovorí skôr o tom, ako objavujeme, poprípade dotvárame sieť, ktorá už existuje. My sa však zaoberáme rastúcimi sieťami. Je preto namieste otázka, či existuje nejaký jednoduchý proces rastu siete, podobný napríklad BA procesu, ktorý by sme poznali zo skúsenosti, a ktorý by prirodzeným spôsobom vytváral hierarchické bezškálové siete. Preferenčné pripájanie uzlov síce vedie na bezškálovú, ale nie hierarchickú štruktúru siete [9, 10, 33].

Zamysleli sme sa nad touto otázkou [17] a vytvorili sme model rastúcej siete, ktorý sme nazvali modelom riadeným klasterizáciou (CD model, clustering driven model). Sieť riadená klasterizáciou rastie takýmto procesom:

1. Na začiatku máme pár uzlov pospájaných niekoľkými hranami. O aký počiatočný graf sa jedná na tom v podstate príliš nezáleží, lebo vlastnosti veľkej siete počiatočný graf vôbec neovplyvní.
2. Nech každú časovú jednotku odkiaľsi z vesmíru priletí jeden uzol a pripojí sa m hranami k starým uzlom. Uzly sú indexované časom svojho príchodu do siete s .

3. Spôsob pripojenia je nasledovný. Jedna hrana sa pripojí s najväčšou pravdepodobnosťou k tomu uzlu, ktorý má najväčší klasterizačný koeficient. Pripojí sa k nemu s pravdepodobnosťou

$$\pi(s) = \frac{C_s}{W(t)} \quad (95)$$

alebo

$$\pi_{new}(s) = \frac{1 + C_s}{t + W(t)} \quad (96)$$

kde $W(t) = \sum_{i=1}^t C_i$ a C_i nie je nič iné ako klasterizačný koeficient uzla i (3).

Môžeme implementovať obe klasterizácie riadené pravdepodobnosťami pripájania hrán, pravdepodobnosť (92) má však tú výhodu, že je nezávislá od počítačného modulu. *Ostatných $m - 1$ hrán sa pripojí náhodne k susedom uzla i .*

Dovolím si ešte pár poznámok k rovnicam (95), (96). Ak je klasterizačný koeficient uzla s nulový, potom pravdepodobnosť $\pi(s)$ (95) pripojenia hrany k tomuto uzlu je nulová, zatiaľ čo z rovnice (96) vyplýva, že aj v tomto prípade existuje malá, nenulová pravdepodobnosť $\pi_{new}(s)$ pripojenia hrany k uzlu s .

$W(t) = \langle C \rangle t$, kde $\langle C \rangle$ je priemerný klasterizačný koeficient uzlov siete.

$\langle C \rangle$ je pre dostatočne veľké t nezávislý od času, a preto sa pravdepodobnosti (95) a (96) pre $t \rightarrow \infty$ správajú rovnako. Ako vidieť z obrázku 14, siete modelované pomocou pravdepodobností (95) a (96) aj vizuálne vyzerajú veľmi podobne.

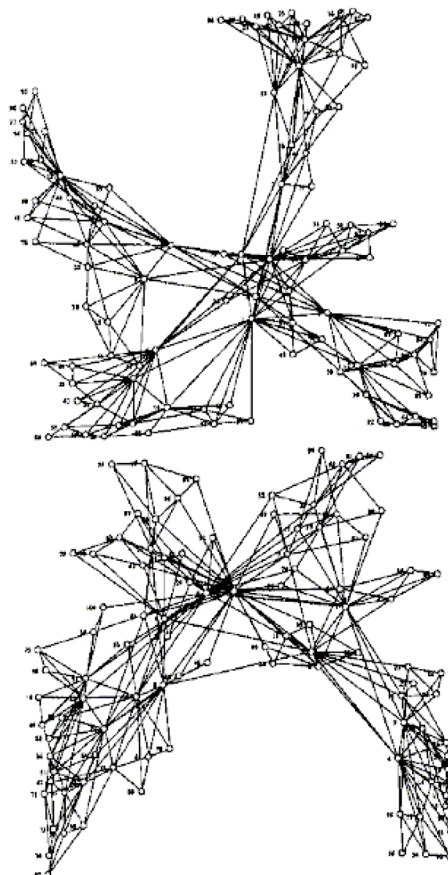
4. Opakujeme od bodu 2.

Sieť, ktorej finálnu štruktúru skúmame, musí byť dostatočne veľká, pretože čím dlhšie simulácia beží, tým presnejšie zmeriame charakteristiky siete. Integro – diferenciálnu rovnicu, ktorá popisuje hodnoverne tento proces, sa nám zatiaľ nepodarilo napísať. Podarilo sa nám však analyticky ukázať, že zjednodušená verzia CD modelu vedie na hierarchickú a bezškálovú štruktúru siete.

V zjednodušenej verzii CD modelu (nazvanej SCD modelom, simplified clustering driven model) každý uzol prichádzajúci do siete sa do nej pripojí len dvoma hranami. Jedna hrana sa pripojí k nejakému uzlu s v sieti a to s pravdepodobnosťou úmernou klasterizačnému koeficientu starého uzla s (95), (96). Druhá hrana sa pripojí náhodne na niektorý z uzlov z najbližšieho susedstva uzla s . Pod najbližším susedstvom rozumieme také uzly, ktoré sú s uzlom s spojené hranou. Dve nové hrany, starý uzol,

jeho sused a nový uzol a tiež hrana medzi starým uzlom a jeho susedom tak vytvárajú trojuholník (obrázok 15).

Akú reálnu situáciu by mohol takýto proces modelovať? Domnievam sa, že to môže byť napríklad vznik záujmových skupín. Predstavme si, že existuje nejaká skupina turistov. Turista, ktorý sa k nej chce pripojiť, najpravdepodobnejšie nájde vedúceho tejto skupiny, človeka, ktorý plánuje výlety, poprípade udržiava a aktualizuje stránku na internete. Nový potenciálny člen klubu kontaktuje túto kľúčovú osobu. Kľúčová osoba turistickej skupiny obyčajne pozná väčšinu členov, ktorí sa poväčšine tiež vzájomne poznajú. V jazyku sietí to znamená, že kľúčová osoba má veľký klasterizačný koeficient. Keď sa nový turista v klube objaví, nadviaže kontakty aj s inými členmi skupiny. Samozrejme, niektorí z nich sú členmi aj iných turistických klubov, poprípade iných záujmových skupín (klastrov) a sprostredkujú tak kontakt medzi rôznymi skupinami.



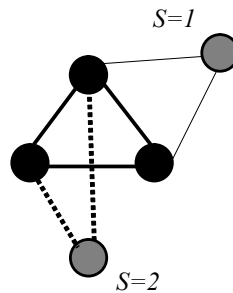
Obr. 14. Vizualizácia CD modelu. Horný obrázok predstavuje sieť, kde sa uzly pripájajú spôsobom (95), dolný obrázok spôsobom (96). Obe siete majú sto uzlov a $m = 3$.

Fakt, že v každom časovom okamihu sa v sieti vytvorí nový trojuholník, je z hľadiska analytických výpočtov veľmi dôležitý. Klasterizačný koeficient uzla s je daný vzťahom

$$C(s) = \frac{e(s)}{\binom{k(s)}{2}} \quad (97)$$

kde $e(s)$ je počet hrán medzi susedmi uzla s a $k(s)$ je stupeň uzla s . Pretože v SCD modeli v každom kroku vznikne v sieti nový trojuholník uzlov, počet hrán $e(s)$ je daný rovnicou $e(s) = k(s) - 1$. Ak toto dosadíme do (97), potom v SCD modeli platí

$$C(s) = \frac{2}{k(s)} \quad (98)$$



Obr. 15. Spôsob pripájania uzlov v SCD modeli. Začíname z malej siete o troch uzloch (čierny uzly a čierna plná hrana). Z nich každý má klasterizačný koeficient rovný jednej. V nasledujúcom okamihu ($t=1$) pripojíme uzol s indexom $s=1$, ktorý prinesie dve nové hrany ($m=2$, tenká plná čiara). Pripojenie jednej hrany k uzlu i je riadené klasterizáciou, druhá sa náhodne pripojí do susedstva uzla i . Takto proces pripájania uzlov pokračuje ďalej.

Ďalšia otázka, ktorú si možno položiť znie: Čo je v SCD procese najdôležitejšie pre vznik bezškálovej štruktúry v sieti? Je to pripájanie uzlov riadené klasterizáciou? Aby sme odpovedali na túto otázku, zohľadníme fakt (98) a vyriešime BA model, v ktorom nie je preferencia pripojenia hrany k uzlu s úmerná k , ale k^{-1} :

$$\frac{\partial k(s,t)}{\partial t} = \frac{k(s,t)^{-1}}{\int_0^t k(s,t)^{-1} ds} \quad (99)$$

Riešenie tejto rovnice je [17]

$$k(s, t) = \left[2^{\frac{1}{2}} \left[\frac{1}{2} - \log_a \right] \left(\frac{t}{s} \right) \right]^{\frac{1}{2}} \quad (100)$$

s $a = 2.964$. Z časti 4 tejto práce vieme, že riešenie (99) nevedie na takú distribúciu stupňa uzlov, ktorá charakterizuje bezškálovú sieť. Pripájanie hrán riadené klasterizáciou teda nezodpovedá za bezškálovú štruktúru vzniknutej siete. Toto možno potvrdiť numerickými simuláciami modifikovaného SCD modelu, v ktorom pripojenie prvej hrany nie je riadené klasterizáciou, ale ju pripojíme k uzlu s náhodným, alebo preferenčným spôsobom. Druhú hranu opäť pripojíme náhodne na bezprostredného suseda uzla s . Numerické simulácie ukazujú, že sieť ostáva aj v takýchto prípadoch bezškálovou.

Prečo? Odpoveď je jednoduchá. Pripájanie druhej hrany do susedstva uzla s je v podstate skrytým preferenčným pripájaním, pretože uzol s veľkým stupňom bude susedom mnohým uzlom. Preto pravdepodobnosť, že sa k nemu druhá hrana pripojí, je väčšia ako pre uzol s malým stupňom, ktorý je susedom len malému počtu uzlov. Ako vieme z BA modelu, preferenčné pripájanie je základným mechanizmom vzniku bezškálovej štruktúry sietí.

Treba si uvedomiť aj ďalší fakt. SCD model je vlastne ekvivalentný a možno ho transformovať na Vásquezov model s jedným surferom ($v_s = 1$) a s jednotkovou pravdepodobnosťou zvýšenia stupňa navštíveného uzla ($q_v = 1$). Nazvime tento variant Vásquezovho modelu V modelom. Vo V modeli blúdenie po sieti začína skokom surfera na náhodne zvolený uzol. V nasledujúcom kroku s pravdepodobnosťou q_e sleduje hranu incidentnú s týmto uzlom a s pravdepodobnosťou $1 - q_e$ skočí na nový, náhodne vybraný uzol. Distribúciu stupňa uzlov pre V model vypočítame z tejto rovnice pomerov [17]

$$\frac{\partial n_k}{\partial t} = A_{k-1} n_{k-1} - A_k n_k \quad (101)$$

pričom (101) platí pre $k \neq 0$ a A_k je pravdepodobnosť, že uzol so stupňom k zvýši svoj stupeň o jednotku. Počet uzlov objavených surferom za jednotku času je aj v tomto prípade

$$\frac{\partial N}{\partial t} = v_a \quad (102)$$

SCD model sa od V modelu líši aj spôsobom, akým sa doňho pridávajú nové hrany. V najjednoduchšej verzii V modelu v každom kroku pridávame do siete buď jednu hranu, alebo jednu hranu a jeden uzol. V SCD modeli v každom okamihu pridáme dve hrany a jeden uzol. Preto, aby sme mohli SCD model pretransformovať na

V model, musíme položiť $q_e = \frac{1}{2}$. Navyše vzťah pre A_k (76) upravíme tak, že zohľadníme (98):

$$A_k = \frac{1}{N} \left[(1 - q_e) \frac{2}{k \langle C \rangle} + q_e v_a k \right] \quad (103)$$

V (103) $\langle C \rangle$ je priemerný klasterizačný koeficient. Ak do (101) zavedieme predpoklad $n_k = NP(k)$, kde $P(k)$ je stacionárna pravdepodobnosť toho, že uzol má stupeň k , potom v spojitaj aproximácii dostaneme

$$\frac{\partial NP(k)}{\partial t} = - \frac{\partial A_k NP(k)}{\partial k} \quad (104)$$

Keďže $P(k)$ nezávisí od času z rovnice (102) a (104) dostaneme

$$\frac{\partial P(k)}{\partial k} = P(k) \frac{v_a(1 + q_e) + \frac{2(1 - q_e)}{k^2 \langle C \rangle}}{(q_e - 1) \frac{2}{k \langle C \rangle} - q_e v_a k} \quad (105)$$

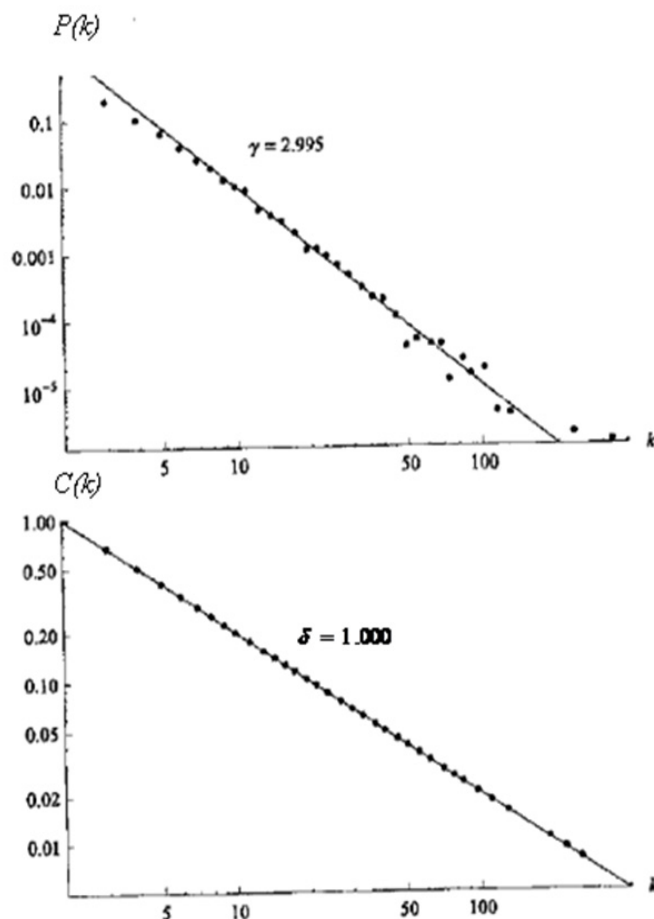
Riešenie tejto rovnice vedie na

$$P(k) \propto k^{-\gamma}, \quad \gamma = 1 + \frac{1}{q_e} \quad (106)$$

čo je v súlade aj s Vásquezovým riešením a potvrdzuje tak fakt, že pripájanie hrán riadené klasterizáciou nie je z hľadiska bezškálovej štruktúry rozhodujúce.

Klasterizačný koeficient v SCD modeli je daný vzťahom (98). Preto pre SCD model v rovnici (65) škálovací exponent $\delta = 1$. Ak vieme, že pre SCD model $q_e = \frac{1}{2}$, potom škálovací exponent distribúcie stupňa uzlov $\gamma = 3$ (106). Tieto výsledky potvrdila aj numerická simulácia SCD procesu (obrázok 16).

Vráťme sa teraz ku všeobecnému CD modelu, kde v každom kroku pribúda do systému viacej hrán ($m > 2$). Numerické simulácie CD modelu ukazujú, že základný rozdiel medzi CD modelom a Vásquezovým modelom je ten, že γ vo Vásquezovom modeli závisí od parametra q_e . V CD modeli $\gamma = 3$ pre všetky hodnoty m . Toto platí aj pre modely, kde pripojenie prvej novej hrany je náhodné, poprípade preferenčné.



Obr. 16. Namerané distribúcie stupňa uzlov a klasterizačných koeficientov v SCD modeli, v ktorom $m = 2$ a počet uzlov $N = 20000$.

Pre všeobecný CD model sme zmerali aj iné charakteristiky [17]. Zmerali sme napríklad priemerný klasterizačný koeficient a separáciu uzlov. Priemerný klasterizačný koeficient je konštantný a sieť má aj vlastnosti siete malého sveta, pretože priemerná najkratšia vzdialenosť medzi uzlami rastie s počtom uzlov siete logaritmicky. Keď meníme počet pripájaných hrán m , numerická simulácia CD modelu potvrdzuje nezávislosť exponentu γ od m , zatiaľ čo exponent δ riadiaci distribúciu priemerných klasterizačných koeficientov uzla stupňa k sa s m mení ako $\delta \propto m^{-\beta}$, kde $\beta = 0.1$.

Skonštatujme na záver niekoľko faktov. V práci [17] súhlasíme s Vásquézom [16], že spôsob pripojenia prvej hrany nie je rozhodujúci pre výslednú topológiu siete. To, čo je naozaj dôležité, sú lokálne pravidlá, teda fakt, že $m - 1$ hrán sa vždy náhodne

pripojí do susedstva vybraného uzla. Proces rastu siete, kde sa uplatňujú takéto lokálne pravidlá, vedie na bezškálovú a zároveň hierarchickú topológiu siete. Na bezškálovú preto, že lokálne efekty zabezpečujú virtuálne preferenčné pripájanie nových hrán. Ak má totiž nejaký uzol veľký stupeň, potom má veľa susedov a pravdepodobnosť, že jedna z $m - 1$ nových hrán sa pripojí práve k nemu je o to vyššia. Na hierarchickú štruktúru vedie takýto proces preto, že v každom kroku sa do siete pripája viac menej pravidelný útvar, pozostávajúci z jedného nového vrcholu a m nových hrán. Ako sme videli v SCD modeli, je týmto útvarom trojuholník uzlov a hrán. V tomto SCD model pripomína Ravász – Barabási model hierarchickej siete [15]. Môžeme teda skonštatovať, že sme našli proces rastu siete, ktorý je prirodzený, a v ktorom lokálne procesy dominujú a vedú k tomu, že sieť je zároveň bezškálová a hierarchická.

6.4 Siete a umelá inteligencia

Umelá inteligencia je informatická vedná disciplína, ktorá hojne využíva poznatky z iných vedných oblastí. Jednou z úloh, ktoré si umelá inteligencia kladie, je prinútiť stroje k tomu, aby ich správanie pripomínalo inteligentné správanie človeka. Takéto stroje – roboty potom môžu nahradiť človeka všade tam, kde je to nebezpečné, poprípade pre ľudský organizmus príliš namáhavé. Na druhej strane, umelá inteligencia si kladie aj iné ciele, ciele blízke tým, ktorými sa zaoberajú kognitívne vedy. Jej úlohou je aj lepšie porozumieť, čo to vlastne inteligencia je. Ako sa inteligentné živé tvory učia? Ako sa rozhodujú, ako konajú a čo toto konanie ovplyvňuje?

Teória sietí môže byť nápomocná umelej inteligencii pri oboch jej hlavných úlohách. Dost' preto prekvapuje, že sa v umelej inteligencii doteraz prakticky nevyužíva. Komunikačné siete robotov by mali byť navrhnuté tak, aby zabezpečovali bezproblémové šírenie signálu. Aj komunikačná sieť iných umelo-inteligentných agentov (nech je to hoci aj ad hoc sieť) by mala splňať isté parametre, aby komunikácia netrpela slabým signálom, veľkým šumom a podobnými nedostatkami. Umelej inteligencii by mohla pomôcť aj znalosť toho, že istá sieťová štruktúra odoláva lepšie náhodným poruchám, iná zasa lepšie cieľovým útokom [33].

Porozumenie ľudskej inteligencii je úzko spojené s výskumom mozgu. Pretože mozog je vo svojej podstate sieťou poprepájaných neurónov, spojitosť teórie sietí a umelej inteligencie sa tu priam ponúka. Aj výskum funkčných sietí mozgu môže pomôcť k porozumeniu kolektívneho sa správania neurónov pri jeho činnosti.

Verím, že nie je ďaleko doba, keď umelá inteligencia a teória sietí nájdu spoločnú reč a obohatia nás o nové zaujímavé výsledky.

7 Perspektívy ďalšieho výskumu

Ďalší výskum sietí, tak ako doteraz, pôjde dvoma smermi: teoretickým a aplikačným. V oblasti teórie sietí sa napríklad aj na našej fakulte pokračuje v štúdiu hierarchickej štruktúry sietí. Nie sú zaujímavé len mechanizmy, ktorými hierarchická štruktúra v sieťach vzniká, ale aj stabilita tejto štruktúry. Do akej miery možno porušiť lokálne zákony pripájania nových hrán, aby sa hierarchická štruktúra zachovala? Je tento

prechod pozvoľný, alebo má charakter náhleho skoku? Čo sa stane s bezškálovosťou, keď sa naruší hierarchia? Toto všetko sú zaujímavé a aktuálne otázky.

Ďalšia vetva teoretického výskumu povedie k tvorbe modelov, ktoré budú ešte lepšie vystihovať vlastnosti reálnych sietí. Binárne siete pomaly treba nahradiť komplikovanejšími modelmi, napríklad aj takými, v ktorých každá hrana existuje len s istou pravdepodobnosťou. Pre šírenie signálov je dôležitá aj priepustnosť hrany – tu sa teória sietí dotýka perkolačnej teórie. Prvé lastovičky v tomto smere výskumu sa už objavili [47].

Z hľadiska praxe je dôležitý aj výskum ad hoc sietí. Vo výskume mozgu bude pokračovať skúmanie funkčných sietí a závislosti ich štruktúry od obtiažnosti kognitívnej úlohy. Významná je aj otázka ako funkčné siete mozgu súvisia s anatomicou štruktúrou mozgu. Dá sa odvodiť ako bude funkčná sieť vyzerat', ak poznáme anatomicú štruktúru oblastí mozgu, zodpovedných za vnímanie daného typu signálu?

Teória sietí začína pomaly prenikať aj do biológie. Skúmajú sa rôzne typy interakčných sietí [6] a tento výskum bude určite pokračovať ďalej aj v budúcnosti. Biologický výskum, podobne ako výskum sociálnych sietí, bude úzko spojený s pokrokom v teórii a s analýzou realistickejších sieťových modelov.

Verím, že tento prehľad teórie dynamických sietí ukázal čitateľovi krásu, ale aj úskalia ich výskumu. Ak sa čitateľ rozhodne prispieť k nemu svojím dielom, účel tejto kapitoly bol viac ako naplnený.

Pod'akovanie: Ďakujem Dr. Edite Máčajovej z Katedry informatiky FMFI za starostlivú recenziu časti venovanej teórii grafov. Táto práca bola podporená grantami VEGA 2/7087/27, 1/0361/08.

Literatúra

- [1] Watts D. J.: *Small Worlds: The Dynamics of Networks between Order and Randomness*, Princeton, Princeton University Press, Princeton, NJ, 1999.
- [2] Milgram S.: *The Small World Problem*, *Psychology Today* **2**, 60, 1969.
- [3] Chialvo D. R.: *Critical Brain Networks*, *Physica A* **340**, 756, 2004.
- [4] Beňušková L., Kasabov N.: *Computational Neurogenetic Modelling*, Springer, New York, 2007.
- [5] Buxton, R. B.: *Introduction to Functional Magnetic Imaging*, Cambridge University Press, 2002.
- [6] Goh K. I., Khang B., Kim D.: *Graph Theoretic Analysis of Protein Interaction Networks of Eukaryotes*, *Physica A* **357**, 512, 2005.
- [7] Motter A. E., de Moura A. P. S., Ying – Cheng L., Dasgupta P.: *Topology of the Conceptual Network of Language*, *Phys. Rev. E* **65**, 065102 (R), 2002.
- [8] Ferrer i Cancho, R., Solé R. V.: *The Small World of Human Language*, *Proc. Royal Soc. London B*, 2261, 2001.

-
- [9] Barabási A. L., Albert R.: *Emergence of Scaling in Random Networks*, Science **286**, 509, 1999.
- [10] Dorogovtsev S. N., Mendes J. F. F.: *Evolution of Networks*, Adv. Phys. **51**, 1079, 2002.
- [11] West D. B.: *Introduction to Graph Theory*, Prentice Hall, NJ USA, 2001.
- [12] Schuster H. G.: *Deterministic Chaos. An Introduction* Willey - VCH Weinheim, Germany, 2005.
- [13] Eordos, Rényi: *On Random Graphs*, Publicationes Mathematicae **6**, 290, 1959.
- [14] Newman M. E. J.: *Models of Small World*, J. Stat. Phys. **101**, 819, 2000.
- [15] Ravasz E., Barabási A. L.: *Hierarchical Organization in Complex Networks*, Phys. Rev. E **57**, 026112, 2003.
- [16] Vásquez A.: *Growing Networks with Local Rules*, Phys. Rev. E **67**, 056104, 2003.
- [17] Náther P., Markošová M., Rudolf B.: *Hierarchy in the Growing Scale Free Networks with Local Rules*, Physica A **388**, 5036, 2009.
- [18] Faloutsos M., Faloutsos P., Faloutsos Ch.: *On Power – law Relationship of the Internet Topology*, ACM SIGCOMM'99, 1999.
- [19] Yook S., H., Jeong H., Barabási A. L.: *Modeling the Internet's Large – scale Topology*, Proc. Nat. Acad. Sci **99**, 13382, 2002.
- [20] Almaas E., Kovacs B., Vicsek T., Oltvai Z., N., Barabási A. L.: *Global Organization of Metabolic Fluxes in the Bacterium Escherichia Coli*, Nature **427**, 839, 2004.
- [21] Montoya J. M., Solé R. V.: *Small World Patterns in Food Webs*, J. Theor. Biol, **214**, 405, 2002.
- [22] Ravasz E., Somera A. L., Mongru D. A., Oltvai Z. N., Barabási A. L.: *Hierarchical Organization of Modularity in Complex Networks*, Science **30**, 1551, 2002.
- [23] Jeong H., Tombor B., Albert R., Oltvai Z. N., Barabási A. L.: *The Large – scale Organization of Metabolic Networks*, Nature **407**, 651, 2000.
- [24] van Noort V., Snel B., Huynen M.: *The Yeast Coexpression Network Has a Small– world, scale – free architecture...*, EMBO Rep., **5**, 280, 2004.
- [25] Newman M. E. J.: *Analysis of Weighted Networks*, Phys. Rev. E **70**, 056131, 2004.
- [26] Eguíluz V. M., Chialvo D. R., Cecchi G., Baliki M., Apkarian A. V.: *Scale – free Brain Functional Networks*, Phys. Rev. Letters **94**, 018102, 2005.
- [27] Bak P.: *How Nature Works*, Copernicus, Copernicus Press, NY, USA, 1996.
- [28] Bak P., Tang Ch., Wiesenfeld K.: *Self Organized Criticality*, Phzs. Rev. A **38**, 364, 1988.
- [29] Beggs J. M., Plenz D.: *Neuronal Avalanches in Neocortical Circuits*, J. Neuroscience **23**, 11167, 2003.

-
- [30] Newman M. E. J.: *The Structure of Scientific Collaboration Networks*, Proc. Natl. Acad. Sci. USA **98**, 404, 2001.
- [31] Newman M. E. J.: *Who is the Best Scientist? A Study of Scientific Collaboration Networks*, Phys.Rev. E64, 016132, 2001.
- [32] Liljeros F., Edling CH. R., Amaral L. A. N., Stanley E. H., Aberg Y.: *The web of Human Sexual Contacts*, Nature **411**, 907, 2001.
- [33] Albert R., Barabási, A. L.: *Statistical mechanics of complex networks*, Rev. Mod. Phys. **74**, 47, 2002.
- [34] NWB Team: *Network Workbench Tool*, Indiana University, Northwest University, and University of Michigan, <http://nwb.slis.indiana.edu>, 2006.
- [35] Vásquez A.: *Statistics of Citation Networks*, cond-mat/0105031v1, 2001.
- [36] Chow T. L.: *Mathematical Methods for Physicists*, Cambridge University Press, Cambridge CB2 2RU, UK, 2001.
- [37] Serrano M. A., Maguitman A., Boguna M., Fortunato S., Vespigniani A.: *Decoding the Structure of the WWW: Facts versus Sampling Biases*, cs/0511035v2, 2006.
- [38] Dorogovtsev S. N., Mendes J. F. F., Samukhin A. N.: *WWW and Internet Models...*, Cond-mat/0009090v1, 2000.
- [39] Dorogovtsev S. N., Mendes J. F. F.: *Accelerated Growth of Networks*, Cond – mat/0204102, 2002.
- [40] Markošová M.: *Network Model of Human Language*, Physica A **387**, 661, 2008.
- [41] Dorogovtsev S. N., Mendes J. F. F.: *Language as an Evolving Web*, Proc. Royal Soc. London B, **268**, 2603, 2001.
- [42] Markošová M.: *Modely jazyka ako dynamickej siete*, Modely mysle, editori Kvasnička, Kelemen, Pospíchal, Europa, 146, 2008.
- [43] Markošová M., Náther P.: nepublikované.
- [44] Markošová M., Franz L., Beňušková L.: *Topology of Brain Functional Networks: Towards the Role of Genes*, Editori M. Koeppen, N. Kasabov, G. Coghill, Advances in Neuro-Information Processing, ICONIP 2008, LNCS 5506, Springer, Berlin/Heidelberg, 111, 2009.
- [45] Franz L.: súkromný rozhovor.
- [46] Humphries M. D., Gurney K., Prescott T. J.: *The brainstem Reticular Formation is a Small World, not Scale Free*, Network, Proc. Biol. Sci **723**, 503, 2006.
- [47] Fagiolo G., Reyes J., Schiavo S.: *On the Topological Properties of the World Trade Web: A Weighted Network Analysis*, arXiv: 0708 43591v1, 2008.

Včelí úl' ako metafora riešenia problémov v umelej inteligencii

Pavol NÁVRAT a Anna BOU EZZEDDINE¹

Abstrakt. Prírodou alebo biologicky inšpirované metódy riešenia problémov získavajú na dôležitosti. Potvrďuje sa ich potenciál pre navrhovanie nových, ale aj zefektívnenie známych metód riešenia rôznych druhov problémov. Správanie sa sociálneho hmyzu poskytuje mnohé inšpirácie. V poslednej dobe sa pozornosť popri mravcoch začína obracať aj na iné druhy. My študujeme modely správania sa včiel medonosných. Inšpirovaní ich správaním sme navrhli model a experimentovali s ním pri riešení rôznych tried problémov, najmä na vyhľadávanie na internete.

1 Úvod

Pri hľadaní metód riešenia zložitých problémov sa vedci často inšpirujú správaním živočíchov, ktoré pozorujú v prírode. Na prvý pohľad sa môže zdať správanie jednotlivých živočíchov veľmi primitívne. Na druhý pohľad, vnímajúc ich ako celok, niektoré živočíšne druhy sa prejavujú správaním, ktoré sa môže vyhodnotiť tak, že jednoducho, a pritom veľmi efektívne riešia zložité problémy. Častou inšpiráciou býva problém zberu potravy mravcov, včiel alebo rýb. Ak sa pozorované jednoduché pravidlá neskôr prevedú do algoritmov, vznikajú jednoduché, ale veľmi účinné nástroje na riešenie rôznych zložitých problémov.

I keď možný okruh živočíchov, ktorých správanie vykazuje zaujímavé črty, naznačujúce schopnosť akoby riešiť problémy je širší, najväčšia pozornosť výskumníkov sa sústreďuje na živočíchy s rozvinutým sociálnym správaním. Je to pochopiteľné. Ako sme už vyššie naznačili, u takýchto živočíchov nejde ani tak o ich individuálny výkon ako konateľa – riešiča problému, ako o výkon roja (úľa, mraveniska, svorky atď.), t.j. viackonateľského, vulgo multiagentového systému. Ich emergentné správanie z nich robí riešič nejakej triedy problémov.

Hmyz je z tohto hľadiska veľmi zaujímavou triedou živočíšnej ríše. Vývojovo nestojí síce príliš vysoko, niektoré druhy však vykazujú sociálne správanie, ktoré je impozantné aj čo do počtu zúčastňujúcich sa jedincov, aj čo do výsledných prejavov.

¹ Fakulta informatiky a informačných technológií, STU Bratislava, Ilkovičova 3, 812 19 Bratislava, E-mail: navrat@fiit.stuba.sk, anna.bou.ezzeddine@stuba.sk

Druhy, vykazujúce výrazné sociálne správanie, sa niekedy označujú ako tzv. sociálny hmyz. Sociálny hmyz žije vo viac alebo menej organizovaných spoločenstvách s týmito črtami:

1. spolupráca v starostlivosti o mladé pokolenie,
2. reprodukčné rozdelenie práce so sterilnými jedincami pracujúcimi za jedincov zamestnaných reprodukciou,
3. prekrytie aspoň dvoch generácií vývojových štádií schopných pracovať v kolónii.

Sociálny hmyz zaujíma v oblasti umelého života kľúčové postavenie najmä kvôli relatívnej jednoduchosti správania sa jedinca v spojení so zložitým kolektívnym správaním. Spoločenstvá sociálneho hmyzu sú schopné vyvinúť prostriedky na kolektívne riešenie úloh. Zložitosť týchto úloh o mnoho rádov prevyšuje schopnosti jednotlivca. Dokážu riešiť úlohy bez centrálného vedenia a bez existencie dopredu daných fixných štruktúr aj napriek existencii výrazných vnútorných šumov.

Rôzne druhy sociálneho hmyzu, ako napríklad mravce, včely, osy, alebo termity majú na prvý pohľad odlišné správanie. Pri podrobnejšom štúdiu ich správania sa a činností nájdeme veľa rovnakých princípov.

Napríklad mravce majú schopnosť vytvárať dynamicky sa meniace štruktúry a sú schopné nájsť najkratšiu cestu od zdroja potravy do hniezda (mraveniska) bez použitia zraku. Sú schopné prispôbiť sa zmenám prostredia a nájsť novú najkratšiu cestu v prípade, ak stará je nepoužiteľná preto, že ju zatarasila prekážka.

Hlavným prostriedkom, ktorý mravce používajú na vytváranie a udržiavanie spojenia, je feromónová stopa. Zanechávajú určité množstvo feromónov a uprednostňujú smer, na ktorom je ich viac. V tomto prípade mravce, ktoré si zvolili kratšiu cestu, rýchlejšie zrekonštruujú novú cestu ako tie, ktoré si vybrali dlhšiu. Kratšia cesta získa vyššie množstvo feromónov za jednotku času.

Pochopenie emergentných schopností kolónii mravcov a kolektívneho správania sa spoločenstva bolo inšpiráciou pri návrhu nových metód distribuovaných výpočtov.

Včely v prípade hľadania potravy vďaka používaniu vzdušných ciest nemajú možnosť označovať si cestu ako mravce. V tomto prípade si včely odovzdávajú informácie o vzdialenosti zdroja potravy tancom.

V roku 1973 rakúsky zoológ Karl Ritter von Frisch dostal Nobelovu cenu za objavenie jazyka tancujúcich včiel [9]. Ak sa zdroj nachádza blízko úľa, včela tancuje tzv. kruhový tanec, ktorý neobsahuje informáciu o smere potravy. Pri vzdialenejšom zdroji včely zakomponujú do tanca aj informáciu o smere a vzdialenosti potravy. Tanec sa začne rozťahovať a nadobúda tvar osmičky.

V tejto kapitole predstavíme niekoľko prác, v ktorých autori opisujú správanie sa sociálneho hmyzu, najmä však správanie sa včiel a možnosti využitia metafory včiel na získanie informácií.

V týchto prácach včely predstavujú multiagentový systém. Hlavným problémom v týchto systémoch je spôsob dorozumievania sa agentov (v prípade včiel je to tanec). Agenti nemajú globálny pohľad na riešený problém, vidia iba svojim lokálnym

pohľadom. Systém by mal byť schopný vysporiadať sa s obmedzeniami, ktoré si jednotlivé agenty kvôli svojmu lokálnemu pohľadu neuvedomujú. Od agentov sa vyžaduje, aby rozumne spolupracovali (kooperovali). Jednou z ciest spolupráce medzi agentmi bez explicitnej komunikácie je využitie inteligencie roja [4], [32].

2 Samoorganizácia v biologických systémoch

V tejto podkapitole detailnejšie uvedieme, čo rozumieme pod pojmom inteligentné správanie sa sociálneho hmyzu a zadefinujeme pojem inteligencia roja.

V ďalšej časti opíšeme spôsob samoorganizácie vo včelom spoločenstve a experimentálne overenie spôsobu, ako si včelie spoločenstvo vyberá medzi zdrojmi potravy podľa [28].

2.1 Inteligencia roja

Inteligentné správanie sa sociálneho hmyzu vzniká prostredníctvom nepriamej komunikácie medzi jedincami spoločenstva známej ako princíp stigmergie.

Termín stigmergia podľa [3] použili v 1950, aby pomenovali typy vzájomných vzťahov medzi sociálnym hmyzom, ktoré vznikali napríklad pri stavbe mohýl u termitov, mravenísk u mravcov, úl'ov u včiel. Stigmergia je mechanizmus spontánnej nepriamej koordinácie medzi jednotlivcami spoločenstva. Je to forma samoorganizácie, kde spoločenstvo vytvára zložité, zjavne inteligentné štruktúry bez nutnosti akéhokoľvek plánovania a riadenia. Podporuje efektívnu spoluprácu medzi veľmi jednoduchými živočíchmi, ktorí nemusia mať akúkoľvek pamäť a inteligenciu. Neskôr tento termín preniesli aj do iných odborov vrátane informatiky.

Termín inteligencia roja (swarm intelligence) bol podľa [4], použitý v umelej inteligencii v roku 1989 pri mobilných robotických systémoch. Inteligencia roja opisuje kolektívne správanie sa v decentralizovanom samoorganizovanom systéme, ktorý môže byť prírodný alebo umelý.

Inteligencia roja v umelej inteligencii je schopnosť systémov zložených z neinteligentných agentov s obmedzenými individuálnymi schopnosťami spoločne dosiahnuť celkové inteligentné správanie sa roja [33].

Agent sa v tomto zmysle chápe ako jednotlivec schopný vnímať a pozorovať okolité prostredie a následne podniknúť niektorú z dostupných akcií. K týmto akciám patrí aj menenie samotného prostredia, v ktorom sa agent nachádza.

Jednotlivci nemajú žiadne vedomosti o riešení daného problému a inteligentné správanie sa objavuje z akcií roja takýchto agentov. Medzi hlavné vlastnosti inteligencie roja patrí:

1. flexibilita - roj sa môže rýchlo adaptovať na meniace sa prostredie,
2. robustnosť - kolónia je úspešná, aj keď niektorí jedinci zahynú,
3. samoorganizácia - kolónia nemá žiadnu riadiacu jednotku,

4. decentralizácia - presun riadiacej moci na nižšie orgány-jedincov,
5. paralelizmus - súbežné vykonávanie rovnakých akcií viacerými jedincami.

2.2 Samoorganizácia vo včelom spoločenstve

V knihe [6] v kapitolách 12, 15, 16 sa autori zaoberajú opisom samoorganizácie včelieho spoločenstva. Spoločenstvo si pomocou jednoduchých pravidiel vyberá najlepší zdroj nektáru. Včely vylietavajú do okolia a hľadajú potravu pre spoločenstvo. Keď včela nájde potravu, vracia sa s ňou do úľa, a pritom prináša pre ostatné včely aj správu o zdroji potravy.

V tomto procese hľadania potravy má včela tieto možnosti:

1. pokračovať v zháňaní potravy bez lákania ostatných včiel,
2. zdieľať informáciu o zdroji potravy tancovaním, pritom im oznamuje smer, kvalitu a vzdialenosť zdroja. Na tento zdroj sa snaží zlákať aj ostatné včely,
3. opustiť zdroj potravy a nechať sa zlákať inou včelou, ktorá propaguje kvalitnejší zdroj potravy.

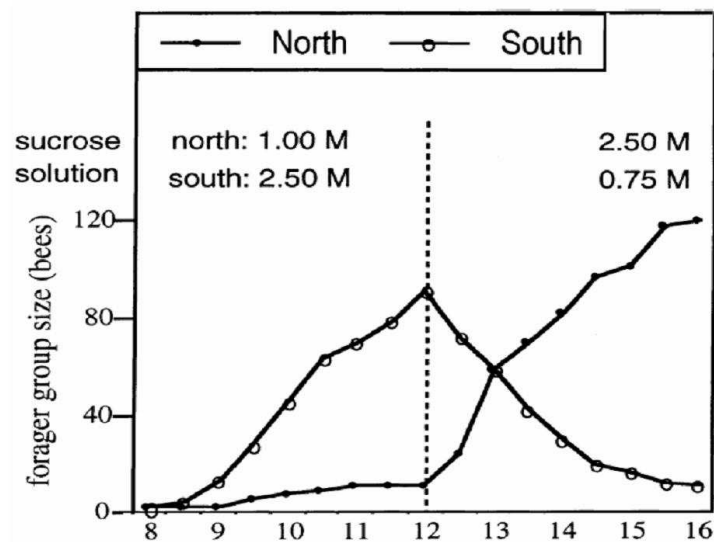
2.3 Včelie spoločenstvo a jeho výber medzi zdrojmi potravy

V práci [29] autori experimentálne dokázali, že rozhodovanie včely v procese hľadania potravy je založené na veľmi obmedzených informáciách získaných z ňou navštívených zdrojov. Napriek takémuto jednoduchému správaniu sa každej včely spoločenstvo si vyberie najlepší zdroj potravy. Tento zdroj si vyberá na základe miery tancovania za lepší zdroj a opúšťania nekvalitného zdroja.

Vo vyššie spomenutej práci autori vykonali experiment zameraný na preskúmanie toho, ako si včelie spoločenstvo vyberá medzi zdrojmi potravy. Experiment vykonali na púšti. V blízkosti včelieho spoločenstva umiestnili dva zdroje potravy. Prvý umiestnili 400 m južne od kolónie a druhý 400 m severne od kolónie. Zdroje mali rôznu koncentráciu nektáru. Severný, menej kvalitný, mal koncentráciu 1.0 jednotiek, južný bol kvalitnejší s koncentráciou 2.5 jednotiek. Experiment prebiehal v čase od 8:00 do 16:00. O 12:00 zdroje vymenili, t.j zmenila sa kvalita zdrojov, takže severný mal koncentráciu 2.5 jednotiek a južný iba 0.75 jednotiek. Na začiatku experimentu natrénovali 12 včiel, aby leteli k severnému zdroju potravy a 15 včiel, aby letelo k južnému zdroju potravy.

Pri pozorovaní sa zistilo, že počet včiel znášajúcich potravu z kvalitnejšieho zdroja sa časom zväčšoval a počet včiel znášajúcich potravu z menej kvalitného zdroja sa postupne znižoval. Po zamenení zdrojov sa zmenila aj situácia. Severný, po zámene kvalitnejší, zdroj navštevovalo postupne čoraz viac včiel a južný, menej kvalitný, zdroj začali včely opúšťať.

Obrázok 1 znázorňuje výsledok tohto experimentu. Horizontálna os znázorňuje čas (8:00 – 16:00), vertikálna os znázorňuje počet včiel, ktoré nosia potravu z vyznačeného zdroja.



Obr. 1. Experiment: Ako si včelie spoločenstvo vyberá medzi zdrojmi potravy [29].

3 Modely inšpirované správaním sa sociálneho hmyzu

Veľká časť prác, ktoré sa zaoberajú správaním sa sociálneho hmyzu a jeho využitia v informačných technológiách, si berie za vzor model správania sa mravcov. Neskôr vznikajú modely, ktoré sú inšpirované správaním sa včelieho spoločenstva. Nie každý z týchto modelov rieši rovnaký problém, preto sa modely líšia, ale podstata správania sa včiel zostáva rovnaká.

3.1 Modely správania sa mravcov

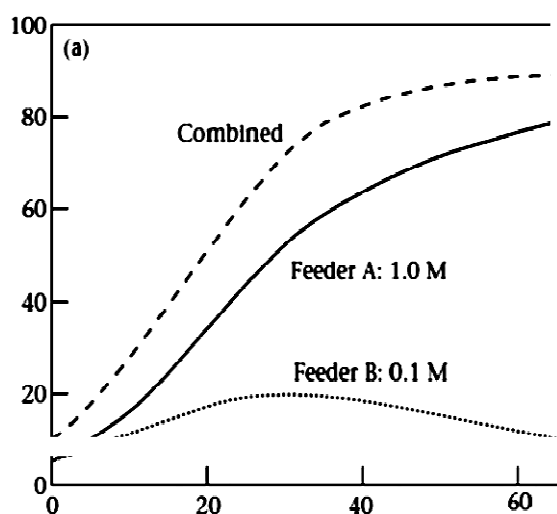
V práci [30] sa autori zaoberali výskumom faraónových mravcov a distribúcie pracovnej sily medzi dva rôzne kvalitné zdroje potravy. Ak bol dostupný iba jeden zdroj potravy, mravce ho navštevovali bez ohľadu na jeho kvalitu. Pokiaľ boli dostupné dva rovnaké zdroje, spoločné využívanie oboch zdrojov zostalo na rovnakej úrovni ako využívanie jedného zdroja. Pri použití dvoch zdrojov s rôznymi kvalitami potravy boli mravce schopné rozpoznať ten lepší a navštevovať ho častejšie a početnejšie ako menej kvalitný. Výsledok tohto experimentu je znázornený na obrázku 2.

Autori práce [10] sa pri vytváraní nového modelu pre vyhľadávanie na internete inšpirovali mravcami. Táto práca začína empirickým pozorovaním. Odkazy často reprezentujú autorove úmysly spojiť stránku, na ktorej sa tento odkaz nachádza, s inou stránkou, vzhľadom na spoločnú tému. Takže ak stránka obsahuje tému, ktorá zaujíma používateľa, a ak si používateľ prezerá práve túto stránku, s veľkou

pravdepodobnosťou navštívi aj odkazovanú stránku, pretože sú pravdepodobne zamerané na tú istú tému alebo problém.

Postupnosť odkazov reprezentuje možnú cestu jedného agenta, kde na konci každého prieskumu môže byť vypustená feromónová stopa. Keď sa agent nachádza na určitom zdroji, má k dispozícii iba dve informácie: kvalitu zdroja na základe porovnania jeho obsahu s používateľovým dopytom a množstvo feromónových hormónov na výstupných odkazoch.

Pokiaľ sa na ceste nachádzajú relevantné informácie, bude sa feromónová stopa zosilňovať a množstvo prilákaných mravcov porastie.



Obr. 2. Porovnanie počtu mravcov pri zbere potravy z dvoch zdrojov [30].

3.2 Modelovanie kolektívneho hľadania potravy včelieho spoločenstva

V práci [32] sa autori zaoberajú modelovaním kolektívneho zbierania potravy prostredníctvom včiel. Individuálne orientovaný model je skonštruovaný tak, aby simuloval kolektívne správanie sa včiel pri zbere. Každá včela sa riadi rovnakým súborom pravidiel správania.

Cieľom autorov v tejto práci bolo vytvoriť simulačný model, ktorý by dosahoval podobné výsledky ako pozorovali v práci [29]. V tomto modeli existuje niekoľko kategórií včiel:

1. Zamestnaná včela: včela, ktorá pozná pozíciu výnosného zdroja, nosí z neho potravu a nenasleduje tancujúce včely.
2. Nezamestnaná včela: včela, ktorá nenesie potravu.
3. Včela začiatovníčka.

4. Včela prieskumníčka: nepozná zdroj potravy a začne ho hľadať spontánne.
5. Zlákaná včela: začne hľadať zdroj potravy potom, ako pozoruje tancujúce včely. Pozná približnú pozíciu zdroja, ale nepozná jeho kvalitu.
6. Skúsená včela: včela, ktorá pozná pozíciu a kvalitu zdroja potravy.
7. Včela inšpektor: spontánne sa reaktivuje, vykonáva prieskumné lety a navštívi známy zdroj, aby znovu zistila jeho kvalitu.
8. Reaktivovaná včela: reaktivuje sa potom, ako odpozoruje od tancujúcich včiel potvrdzujúcu informáciu.
9. Včela prieskumníčka: spontánne začne hľadať nový zdroj potravy potom, ako sa jej predchádzajúci vyčerpal.

Prostredie, v ktorom sa včely pohybovali, predstavoval štvorec s dĺžkou hrany 1200 m a úlom umiestneným v jeho strede. 400 m na sever a juh od úľa sa nachádzali dva zdroje, každý rôznej kvality. Včelám bol tiež obmedzený pohyb do vzdialenosti 500 m od úľa.

Jediná časť modelu, ktorá bola prevzatá z reálneho úľa, je tančiareň. Jej podstatou je, že včely, ktoré pozorujú tanec, nevidia všetky tancujúce včely, ale iba malú časť (konkrétne 4%²). Rozloženie včiel v tančiarni je náhodné, a preto je šanca pozorovať určitú tancujúcu včelu nezávislá od priestorového rozloženia včiel v tančiarni.

Včela svoju kariéru začína ako nezamestnaná začiatovníčka. Nemá žiadne znalosti o zdrojoch potravy. V tomto prípade môže spontánne začať hľadať zdroj potravy a stáva sa z nej včela prieskumníčka, alebo po pozorovaní tancujúcich včiel sa stáva zlákanou včelou, ktorá má približnú informáciu o polohe zdroja.

Keď včela nájde zdroj potravy, zapamätá si jeho polohu a kvalitu a začne z neho znášať potravu. Takto sa včela stáva zamestnanou včelou. Včela sa opäť môže stať nezamestnanou včelou v prípade, že sa zdroj potravy vyčerpá.

Ak si včela v pamäti uchováva informácie o zdroji, nazývame ju skúsenou nezamestnanou včelou.

Ak včela vykoná prieskumný let ku zdroju, ktorý kedysi opustila, nazývame ju inšpektorom.

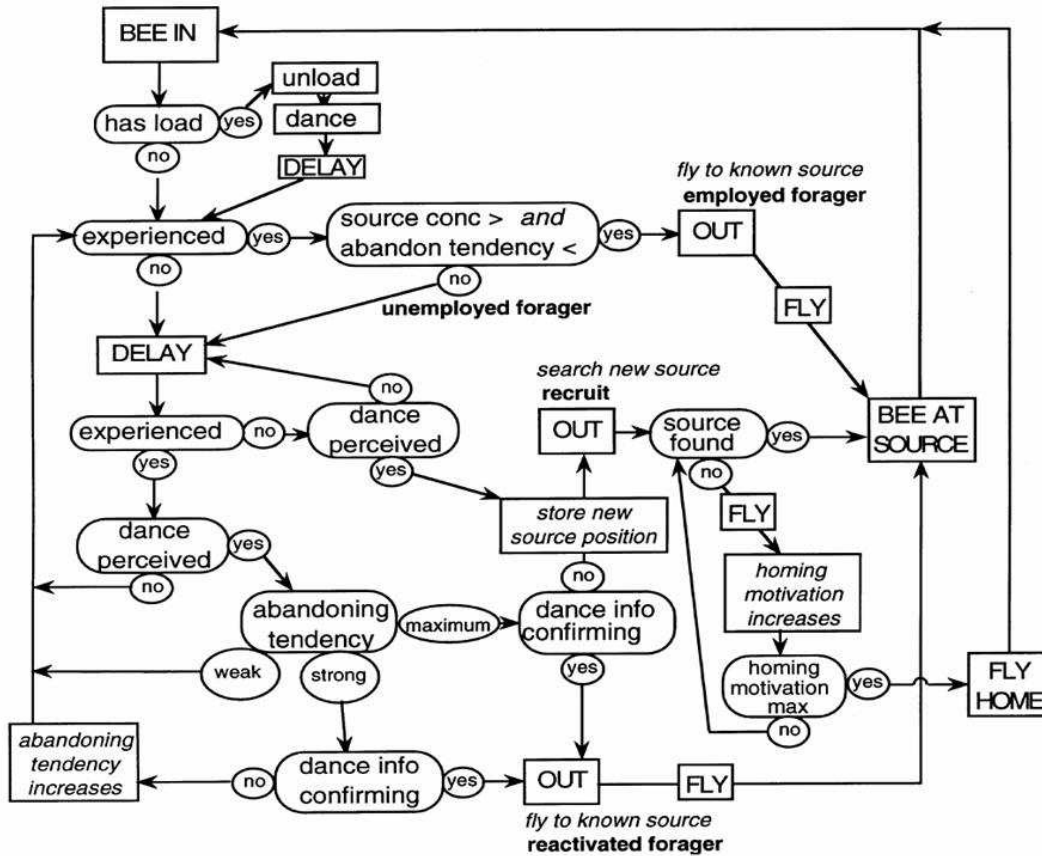
Ak nezamestnaná skúsená včela odpozoruje od tancujúcich včiel pozíciu zdroja, ktorý sa podobá na ňou zapamätaný zdroj, stáva sa reaktivovanou včelou.

Ak nezamestnaná skúsená včela odpozoruje od tancujúcich včiel pozíciu predtým neznámeho zdroja, stáva sa odvedenou včelou.

Včela môže začať spontánne hľadať nové zdroje potravy v prípade, že nebude nasledovať žiadnu tancujúcu včelu.

Tieto pravidlá správania sa včely sú schematicky znázornené na obrázku 3.

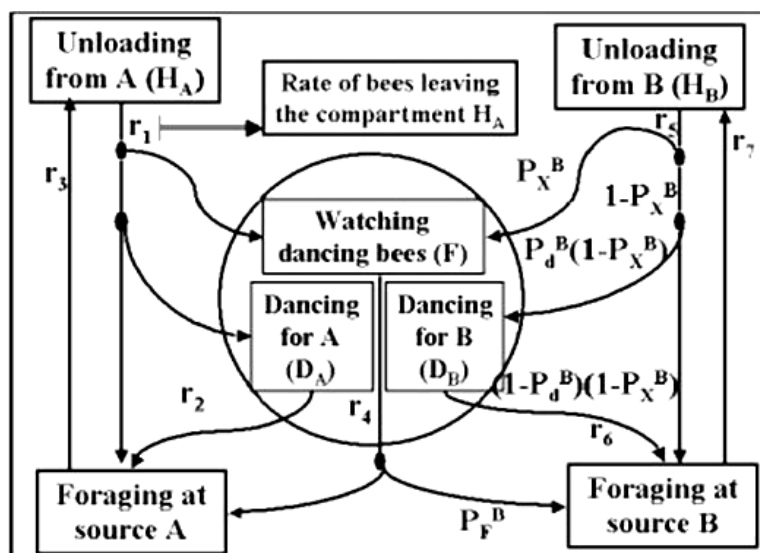
² Tančiareň je kruhová miestnosť s polomerom 25 cm. Každá včela môže vnímať tanec z maximálne 5 cm. Z toho vyplýva, že včela môže vnímať iba ($5^2 / 25^2 = 4\%$) zo všetkých včiel.



Obr. 3. Diagram správania sa včiel pri zbere potravy [32].

3.3 Matematický model získavania potravy z dvoch zdrojov

V práci [5] je uvedený matematický model popisujúci dynamickú interakciu medzi včelami v procese nosenia potravy z dvoch zdrojov (zdroj A, zdroj B).



Obr. 4. Matematický model získavania potravy z dvoch zdrojov [5].

Modelovaná situácia je znázornená na obrázku 4, kde je vyznačených sedem rôznych častí. Každá časť predstavuje jeden z možných stavov, v ktorom sa včela môže nachádzať.

V tomto modeli sú A a B dva rôzne zdroje:

1. časť H_A , časť H_B : vyloženie potravy zo zdrojov A a B ,
2. časť D_A , časť D_B : tancovanie za zdroje potravy A a B ,
3. časť F : pozorovanie tancujúcich včiel.

Na rozloženie celkového počtu včiel nosiacich potravu z dvoch zdrojov na sedem rôznych častí vplývajú dva faktory:

1. itenzita, s ktorou sa včela pohybuje medzi jednotlivými časťami (r_{1-7})
2. pravdepodobnosť, s ktorou si včela vyberie jednu z dvoch ciest v rozhodovacích bodoch (na obrázku sú znázornené ako čierne body).

V nasledujúcom príklade si opíšeme správanie sa včely, ktorá priniesla potravu zo zdroja B .

Po tom, ako včela vyloží v úli potravu, prichádza k prvému rozhodovaciemu bodu, kedy sa rozhoduje, či opustí zdroj $B(P_X^B)$, alebo bude pokračovať v zbieraní potravy zo zdroja $B(1 - P_X^B)$.

Ak sa rozhodne pokračovať v nosení potravy zo zdroja $B(1 - P_X^B)$, môže sa rozhodnúť, že pre tento zdroj zláka ďalšie včely ($P_d^B(1 - P_X^B)$).

Ak sa včela rozhodne nepokračovať v nosení potravy zo zdroja B , ide pozorovať tancujúce včely a rozhodne sa, ktorú z tancujúcich včiel bude nasledovať (P_F^B).

Pravdepodobnosť P_F^B sa vypočíta:

$$P_F^B = \frac{D_B d_B}{D_B d_B + D_A d_A} \quad (1)$$

d_A a d_B sú časy tancovania včely za zdroj A alebo zdroj B . Čas tancovania je priamo úmerný kvalite zdroja, teda, čím je zdroj kvalitnejší, tým je väčšia pravdepodobnosť, že pozorujúce včely odpozorujú vášny tanec a nechajú sa zlákať na zdroj potravy. D_A a D_B je počet včiel tancujúcich za zdroj A alebo zdroj B .

Jednotlivé pravdepodobnosti predstavujú:

1. P_X^A a P_X^B : pravdepodobnosť opustenia zdroja A alebo zdroja B ; táto pravdepodobnosť sa vypočíta pri každom výlete včely,
2. P_d^A a P_d^B : pravdepodobnosť tancovania za zdroj A alebo zdroj B ,
3. P_F^A a P_F^B : pravdepodobnosť nasledovania včely tancujúcej za zdroj A alebo zdroj B .

3.4 Multi-agentový odporúčajúci systém

Na základe matematického modelu z [5] navrhli v [14] a prezentovali v [15] a [16] prototyp multi-agentového odporúčajúceho systému s využitím metafory správania sa spoločenstva sociálneho hmyzu - tancujúcich medonosných včiel. Vychádzali z prác [28] a [27], aplikujúc rozdielny prístup k riešeniu problému. Táto problematika bola spracovaná aj v [13].

Model z [5] rozšírili tak, aby bolo možné použiť viacero zdrojov potravy. Každý agent predstavuje včelu s týmito vlastnosťami:

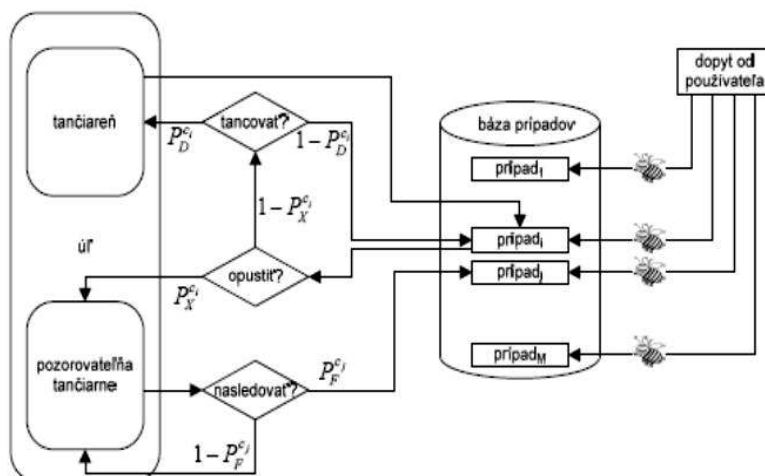
$P_x^{c_i}$ pravdepodobnosť opustenia zdroja potravy c_i ,

$P_D^{c_i}$ pravdepodobnosť tancovania za zdroj c_i ,

$P_F^{c_j}$ pravdepodobnosť nasledovania iných včiel na zdroj c_j .

Pri dopyte od používateľa sa včely-agenty snažia nájsť v báze prípadov (zdrojov $c_{1...m}$) najlepší zdroj potravy.

Na obrázku 5 je znázornený model z [16].



Obr. 5. Model multi-agentového systému [16].

Úľ sa skladá z dvoch oddelení, tančiarne a pozorovateľne tančiarne. Báza prípadov reprezentuje prístupné zdroje nektáru ($1...m$). Kosoštvorce reprezentujú možnosti rozhodovania včiel.

V odporúčajúcom systéme sa používateľova požiadavka porovná so všetkými prípadmi v báze prípadov. Prípado, ktorý najlepšie spĺňa požiadavku, sa ponúka používateľovi. V tomto prípade podobnosť je použitá na to, aby sa včela rozhodla, či bude pokračovať v nosení potravy zo zdroja, alebo prejde na iný. Včela navštívi bázu prípadov, porovná používateľov dopyt s navštíveným prípadom c_i na základe čoho vypočíta pravdepodobnosť $P_X^{c_i}$. Prípado s najmenšou vzdialenosťou od používateľovho dopytu je najpodobnejší prípad, teda prípad, ktorý reprezentuje najlepší zdroj. Prípado c_i je zdroj, ktorý včela momentálne navštevuje. Prípado c_j predstavuje zdroj, ktorý bol navštívený tancujúcou včelou, ktorú sleduje včela z pozorovateľne a rozhoduje sa, či ju bude nasledovať, alebo nie.

Kosoštvorce predstavujú miesta, kde sa včely rozhodujú. Včela sa môže rozhodnúť, že bude pokračovať v nosení potravy zo zdroja (zostáva pri zdroji c_i), potom môže pre tento zdroj lákať iné včely (navštívi oddelenie tančiareň), alebo bude pokračovať v nosení potravy zo zdroja c_i bez pomoci iných včiel. Ak sa rozhodne, že zdroj c_i opustí, navštívi oddelenie pozorovateľňa tančiarme odkiaľ pozoruje tancujúce včely a rozhoduje sa, ktorú z tancujúcich včiel bude nasledovať.

Výhoda odporúčajúceho systému založeného na opísanom princípe je, že systém vždy niečo odporučí. Niekedy je však lepšie neodporučiť nič, ako odporučiť veľmi nepresne, čo v tomto prípade predstavuje nevýhodu takéhoto systému.

Autori na rozšírenie a aktuálnosť bázy prípadov navrhujú použiť ďalšiu vrstvu agentov, ktorí by hľadali aktualizované informácie a zaradovaním do bázy prípadov by ju rozširovali.

V experimentoch, ktoré autori s týmto systémom vykonali, bolo dokázané, že čím je zdroj kvalitnejší (čím viac sa zhoduje s dopytom používateľa), tým viac včiel ho navštevuje. Autori nechceli poukázať iba na schopnosť zhukovania sa včiel okolo najlepšieho zdroja, ale aj na dynamickú reakciu celého spoločenstva pri objavení lepšieho zdroja. Skúmali aj rýchlu reakciu systému pri zmene používateľovho dopytu.

4 Návrh modelu včelieho úľa

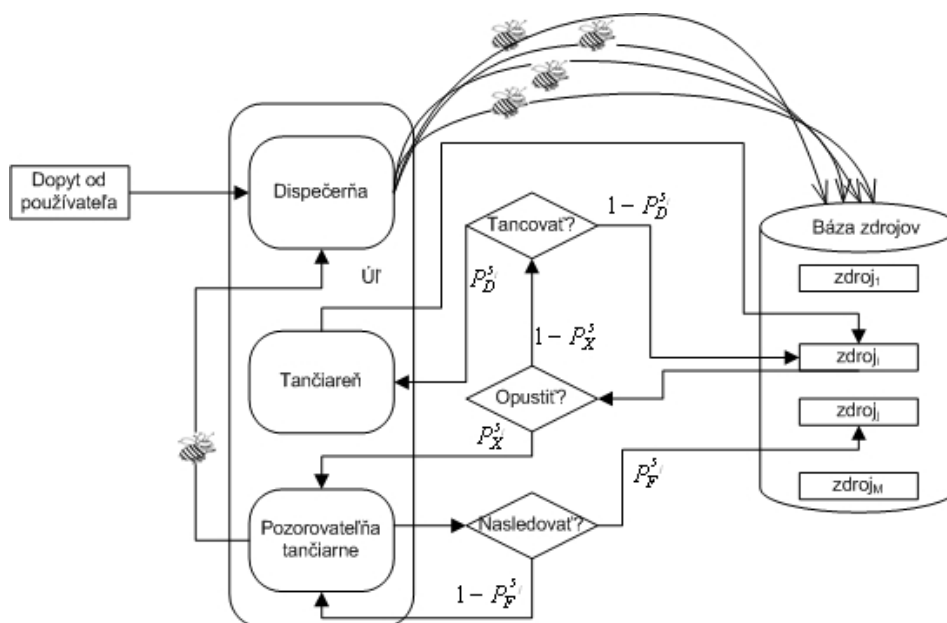
Autori [18] sa pri návrhu modelu inšpirovali modelom predstaveným v [16]. Boli zistené niektoré nedostatky modelu, z ktorého vychádzali a ich hlavným cieľom bolo ich odstránenie s prihliadnutím na to, aby sa nenarušil model včelieho spoločenstva.

K hlavným nedostatkom predtým opísaného systému bolo počiatkové priradenie jednej včely na jeden zdroj potravy. Pri takomto priradení vznikajú tieto problémy:

- na dosiahnutie prijateľných výsledkov treba použiť toľko včiel-agentov, koľko je zdrojov, takže pri veľkom počte zdrojov by boli nároky na systém neprijateľné,
- ak sa včele na začiatku priradí zdroj, ktorého kvalita je vysoká (napríklad 95%), napriek tomu existuje pravdepodobnosť (hoci iba 5%), že bude zlákaná inou včelou. V tomto prípade, ak včela tento zdroj opustí, neexistuje možnosť návratu k tomuto zdroju. Existujú iba dva spôsoby, ako sa včela dostane na zdroj. Buď na základe počiatkového priradenia, alebo v prípade zlákania inou včelou. Týmto sa môže veľmi kvalitný zdroj stať neprístupným pre celé spoločenstvo.

Hlavné požiadavky pri vytváraní modelu boli:

- zmenšenie potrebného počtu včiel;
- zmenšenie pravdepodobnosti trvalého opustenia kvalitného zdroja.



Obr. 6. Model správania sa včelieho spoločenstva podľa [18].

Oproti pôvodnému modelu sme do časti úl pridali oddelenie nazvané dispečerňa. Z tohto oddelenia včely vylievajú náhodne do bázy zdrojov, kde hľadajú zdroje potravy. Pri používateľovom dopyte včela cez dispečerňu vyletí k náhodnému zdroju.

Ďalšou zmenou v modeli bolo usmernenie nerozhodných včiel, ktoré v predchádzajúcom modeli mohli neobmedzene dlho zostať v pozorovateľni tančiarnie, čím sa znižoval počet výkonných včiel. V tomto modeli včely, ktoré sledujú v pozorovateľni tancujúce včely a do určitého času si nevyberú včelu, ktorú budú nasledovať, prechádzajú do dispečerne a odtiaľ na náhodný zdroj potravy.

V modeli sme pridali aj informačný šum pri komunikácii medzi včelami (odovzdávanie informácie o zdroji tancovaním). Miera informačného šumu určuje, ako nepresne tancujúca včela oznámi polohu zdroja pozorujúcej včele. Keďže včely neuvažujú priestorovú polohu zdrojov, informačný šum predstavuje pravdepodobnosť, s akou tancujúca včela oznámi nepresnú adresu zdroja pozorujúcej včele. V prípade zvýšeného informačného šumu sa zvyšuje pravdepodobnosť, že včela sa pošle na náhodný zdroj potravy.

Poslednou zmenou v modeli bolo pridanie chyby vyhodnocovania zdroja ERR . Chyba nadobúda hodnoty z intervalu $\langle 0,1 \rangle$. Táto chyba sa berie do úvahy pri výpočte podobnosti používateľovho dopytu a zdroja. Kvalita Q sa určuje zo vzťahu $Q = Q \pm Q * ERR$. V prípade, že v uvedenom vzťahu dostaneme hodnotu väčšiu ako 1, hodnota Q bude nastavená na 1.

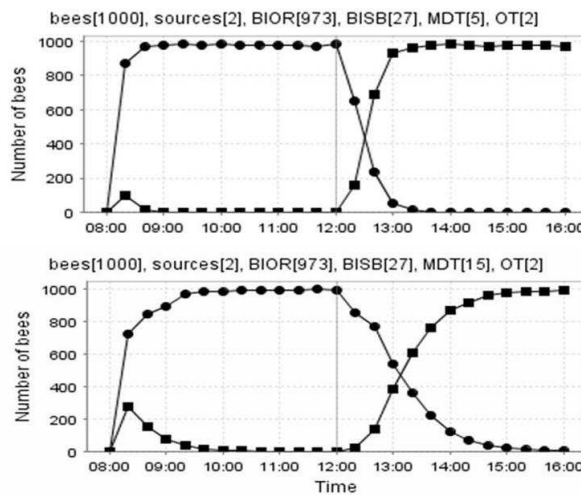
Parametre modelu:

- celkový počet včiel v modeli $N(BIOR + BISB)$,
- počiatočná distribúcia včiel: $BIOR$ predstavuje počet včiel v pozorovateľni, $BISB$ počet včiel, ktoré vyletia do bázy zdrojov,
- maximálny čas tancovania za určitý zdroj potravy MDT , konkrétny čas tancovania je závislý na kvalite zdroja ($MDT * kvalita$),
- maximálny čas, ktorý včela strávi v pozorovateľni OT , sa nemení, ale včela pred uplynutím tohto času môže byť zlákaná inou včelou,
- informačný šum $NOICE$, presnosť pri vymieňaní informácie medzi tancujúcou a pozorujúcou včelou,
- chyba vyhodnotenia kvality zdroja ERR .

4.1 Experimenty s parametrami modelu [18]

Správnosť navrhnutého modelu sme overili experimentom ako v [29] (výber z dvoch zdrojov potravy pri skutočnom včelom spoločenstve). Pri experimente boli súčasne sledované nastavenia nových parametrov systému.

Pri zvyšovaní parametra MDT (maximálny čas tancovania) sa znižovala dynamika systému - včely dlhšie tancujú, čím sa zvyšuje pravdepodobnosť zlákania iných včiel, z ktorých následne viaceré tiež budú tancovať, teda rozhodovanie celého spoločenstva je pomalšie. Na obrázku 7 je znázornený výsledok tohto experimentu.



Obr. 7. Experiment s parametrom MDT

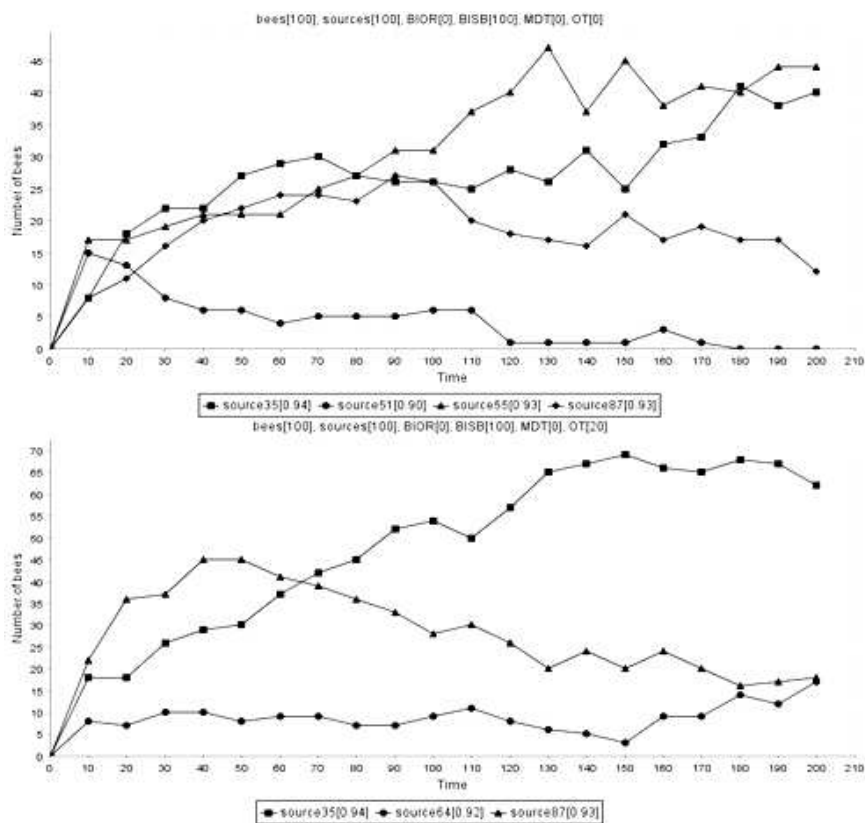
Cieľom experimentu s parametrami *NOICE* a *ERR* bolo skúmať vplyv zavedenia týchto parametrov do modelu. Boli vykonané experimenty, pri ktorých boli parametre modelu menené:

NOICE nastavený na 15% (tancujúca včela oznámi nepresne polohu zdroja pozorujúcej včele o 15%) a *ERR* na 0. *ERR* nastavený na 15% (kvalita zdroja je $\pm 15\%$ z reálnej kvality zdroja) a *NOICE* na 0.

Zavedenie šumu sa môže javiť ako zámerné vnášanie nepresností do modelu. Pri experimentoch sa ukázalo, že zavedenie šumu môže mať priaznivý vplyv na rýchlosť objavovania nových zdrojov. Pri väčšom šume včely častejšie vylietavajú a objavujú nové zdroje.

Pri znižovaní hodnoty parametra *OT* (čas pozorovania tancujúcich včiel) sa zvyšuje dynamika systému ako celku. Pri hodnotách blízkyh nule však nastávajú časté výkyvy správania sa spoločenstva, pretože včely často vylietavajú na náhodné zdroje.

Výsledky experimentu sú znázornené na obrázku 8.



Obr. 8. Experiment s parametrom *OT*, x-ová os predstavuje čas a y-ová os počet včiel.

4.2 Model včiel s pamäťou

Model včiel s pamäťou vychádza z modelu [18]. Pamäť včely je veľmi jednoduchá a okrem naposledy navštíveného zdroja si pamätá pozíciu a kvalitu doteraz najkvalitnejšieho zdroja, ktorý dovtedy navštívila. Ak je kvalita nového zdroja väčšia ako kvalita zapamätaného zdroja, najkvalitnejším sa stáva nový zdroj. Oproti predchádzajúcemu modelu nastáva zmena správania sa včely v pozorovateľni tančiarni. V prípade, že včela v tančiarni propaguje kvalitnejší zdroj ako najkvalitnejší zdroj zapamätaný včelou, môže ju nasledovať alebo sa môže rozhodnúť vrátiť sa k zapamätanému zdroju, prípadne prechádza do dispečerne.

V experimentoch sa overila predpokladaná skutočnosť, že takouto úpravou sa včely tak, ako predtým, sústreďovali okolo jedného zdroja potravy, avšak výsledky odporúčania boli mierne kvalitnejšie a na kvalitné odporúčanie vystačoval menší počet včiel. Experimenty s včelami s pamäťou boli publikované v [19].

4.3 Experiment s homogénnymi a nehomogénnymi včelami

Homogénne včely v tomto experimente predstavovali včely, ktoré vyhodnocujú kvalitu zdroja ako celok.

Nehomogénne včely vyhodnocujú kvalitu iba určitej zložky zdroja. Ak má zdroj viacero zložiek, je potrebné, aby existovalo viacero druhov včiel. Každá nehomogénna včela bude vyhodnocovať kvalitu konkrétnej zložky zdroja. Za najkvalitnejší zdroj sa bude považovať ten, z ktorého na konci simulácie bude nosiť potravu najviac včiel bez ohľadu na to, z akého sú druhu.

V experimente bol použitý trojzložkový zdroj, ktorý pozostával z dvoch slov a jedného čísla. Zdrojom bola hľadaná pracovná pozícia, ktorej prvou zložkou bolo mesto, druhou pracovná pozícia a tretou plat. Postupne bolo použitých 10, 100, 1000 a 2500 včiel a rôznych 1000 zdrojov.

Už pri použití 10 včiel úľ vyhľadal najlepší zdroj, bol však pomalší. Dobré výsledky sa dali dosiahnuť už použitím 10 krát menšieho počtu včiel ako bol počet zdrojov, vynikajúce výsledky boli dosiahnuté pri použití 4 krát menšieho počtu včiel ako zdrojov.

V prípade nehomogénnych včiel sa použili tie isté začiatkové podmienky ako v predchádzajúcom prípade. V tomto prípade sa však nedosiahli najlepšie výsledky. V prípade, že boli viaceré zdroje, ktoré obsahovali niektorú zo zložiek napríklad rovnaké mesto, včely sa rovnomerne rozdistribovali medzi zdrojmi a postupne sa navzájom blokovali.

Výsledky týchto experimentov boli detailnejšie popísané v práci [19].

4.4 Použitie modelu včiel pri počítaní PageRanku

Algoritmus PageRank je založený na činnosti imaginárneho surfera, ktorý náhodne kliká na linky. Po každom kliknutí sa surfer rozhoduje, či klikne znova. Pravdepodobnosť, že surfer bude pokračovať, sa nazýva faktor útlmu. Viaceré štúdie

sa zaoberali rôznymi faktormi útlmu, podľa [25] tento faktor nadobúda vo všeobecnosti hodnotu približne 0,85.

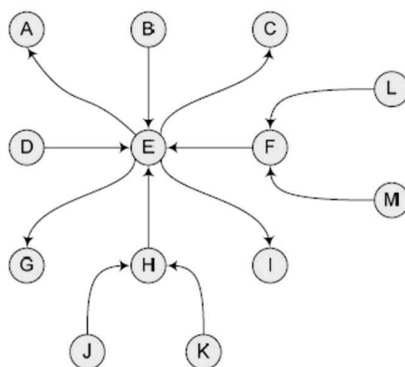
Všeobecný vzorec na výpočet PageRanku je podľa [25]:

$$PR(p_i) = 1 - d + d \sum_{p_j \in M(p_i)} \frac{PR(p_j)}{L(p_j)} \quad (2)$$

kde p_i je stránka, ktorej PageRank počítame, $M(p_i)$ je množina stránok, ktoré odkazujú na stránku p_i a $L(p_j)$ počet vychádzajúcich liniek zo stránky p_j .

Na výpočet dostatočne presnej hodnoty PageRanku každej stránky je potrebných niekoľko iterácií. Počas každej iterácie sa prepočíta PageRank každej stránky v kolekcii. Počet iterácií závisí od počtu stránok v kolekcii a od zložitosti prepojenia jednotlivých stránok.

Na obrázku 9 webové stránky predstavujú vrcholy grafu a orientované hrany grafu prepojenia medzi stránkami.



Obr. 9. Jednoduchý graf prepojenia webových stránok.

Pri iteratívnom prístupe sa vypočíta PageRank stránky A , potom postupne PageRank stránok B, C, \dots, M . Ideálne by bolo vypočítať PageRank stránok, od ktorých sú ostatné stránky závislé.

Ak uvažujeme situáciu, ktorá je zobrazená na obrázku 9 a chceme vypočítať Page Rank stránky E , v tomto prípade potrebujeme poznať PageRank stránok B, D, H, F a následne Page Rank stránok J, K a L, M .

Uvažujme použitie včiel s mierne modifikovaným správaním. Ak včela priletí na zdroj E (vrchol E), zistí kvalitu zdroja, teda PageRank E , ale súčasne by zistila, že na presnejšie určenie PageRanku E je potrebné určiť PageRank vrcholov D, B, F, H .

Ak by bol zdroj E kvalitný, včela pri návrate do úľa by s veľkou pravdepodobnosťou išla do tančiarne, kde by však netancovala za zdroj E , ale za

jeden zo zdrojov odkazujúcich na E (zdroje D, B, F, H). Výber jedného zo zdrojov je náhodný. Týmto sa včela v tančiarni snaží zlákať iné včely na zdroje, ktorých Page Rank je potrebný na presnejšie určenie Page Ranku vrcholu, ktorý navštívila. Po ukončení tancovania sa vráti včela na vrchol E , a ak zlákala iné včely na odkazujúce vrcholy, môže presnejšie určiť PageRank vrchola E . V prípade, že po aktualizácii je kvalita zdroja vysoká, existuje vysoká pravdepodobnosť, že včela bude propagovať v tančiarni jeden z odkazujúcich zdrojov. Celý cyklus sa teda opakuje a Page Rank vrcholu E sa postupne približuje k teoretickej hodnote. Podobné pravidlá platia pre všetky zdroje.

Experimenty publikované v [20], [22] ukázali, že použitie včiel na výpočet Page Ranku môže byť výhodné a v mnohých prípadoch rýchlejšie ako klasický iteračný prístup.

Na rozdiel od klasického prístupu metóda výpočtu PageRanku pomocou včiel neposkytuje vždy presné výsledky, čo je spôsobené povahou modelu - nie je zaručené, že včely navštívia všetky zdroje. Tento nedostatok možno aspoň čiastočne odstrániť zvýšením počtu včiel.

Výhodou je možnosť kontinuálneho výpočtu PageRanku v prevádzke. Napríklad Google prepočítava vždy za určitý čas PageRank celého ním indexovaného webu. Pri použití modelu včiel by včely nepretržite navštevovali webové stránky (vrcholy grafu) a aktualizovali by PageRank jednotlivých stránok.

5 Vyhľadávanie na webe

Práca [12] podáva komplexný prehľad súčasného stavu algoritmov simulujúcich inteligenciu včelieho roja. Podrobne sa opisujú rôzne spôsoby využitia včelieho spoločenstva v informačných systémoch. Ako je aj v tejto práci uvedené, prioritou nášho výskumu je vyhľadávanie na webe, on-line vyhľadávanie inšpirované správaním sa včelieho roja.

V súčasnosti pracuje väčšina vyhľadávačov nad off-line databázou, ktorá obsahuje indexované stránky z časti Internetu. Obsah týchto stránok sa neustále aktualizuje a pribúdajú nové stránky.

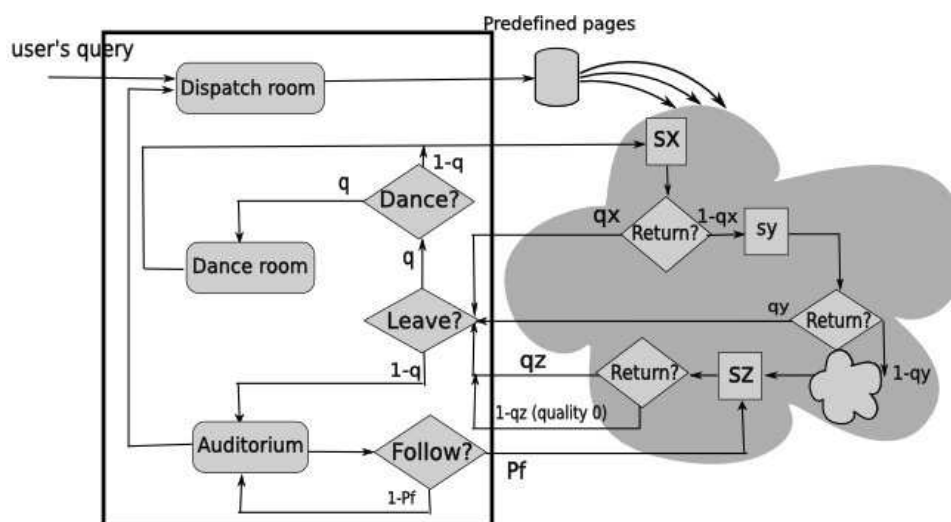
Rozsah Internetu je obrovský a nie je možné prehľadať všetky stránky, preto je nutné on-line vyhľadávanie usmerniť iba na najľubnejšie cesty. Každý používateľ má pri vyhľadávaní svoje preferencie. Tie môže vyjadriť vybratím skupiny stránok, ktoré sa venujú oblasti, ktorú preferuje. Z tejto skupiny stránok sa začína vyhľadávanie. Webová stránka teda predstavuje zdroj, jej poloha URL adresa a dosiahnuteľnosť iného zdroja predstavuje hypertextový odkaz na inú stránku (zdroj).

Pôvodný model [18] je všeobecný a dá sa použiť v rôznych doménových oblastiach. Definuje správanie sa včiel v úli. Správanie sa včiel mimo úľa pre vyhľadávanie na Internete sme dodefinovali v [21].

Včela si po vyletení z úľa vyberie náhodný zdroj x , určí jeho kvalitu a s pravdepodobnosťou q_x sa vráti do úľa, alebo s pravdepodobnosťou $1 - q_x$ si náhodne vyberie iný zdroj, dostupný zo zdroja x . Proces sa opakuje, pokiaľ včela

nenarazí na zdroj z , z ktorého už žiadny iný zdroj nie je dostupný. V tomto prípade sa včela s pravdepodobnosťou q_z vráti do úľa s kvalitou zdroja q_z , inak s kvalitou 0. Návrat včiel s nulovou kvalitou sme zaviedli kvôli tomu, aby tento zdroj nebol zvýhodnený oproti ostatným, pretože z neho sa všetky včely vrátia do úľa a mohli by tento zdroj propagovať viac oproti ostatným. Tým, že sa včely vrátia späť do úľa, propagujú kvalitné stránky alebo ak také stránky nenájdu, môžu pozorovať a nasledovať iné včely. Vyhľadávanie sa usmerňuje na kvalitnejšie zdroje. Za stránku, ktorú úl odporučil, sa považuje stránka, ktorá bola dlhodobo najviac propagovaná v tančiarni.

Na obrázku 10 je model včelieho úľa s dedefinovaným správaním sa včiel mimo úľa.



Obr. 10. Model včelieho úľa so zameraním na vyhľadávanie na webe.

Najdôležitejšou funkciou modelu je zistenie kvality Q nájdenej stránky. Kvalita zdroja je z intervalu $\langle 0,1 \rangle$, pričom 0 je najnižšia a 1 najvyššia kvalita. Výsledná kvalita sa skladá z troch častí:

1. Kvalita vzdialenosti

Vzdialenosť dvoch stránok definujeme ako počet krokov, ktoré včela potrebuje, aby sa z jednej domény dostala na druhú. Čím je počet krokov menší, tým je väčšia pravdepodobnosť, že dané stránky sú obsahovo príbuzné. Na kvantifikáciu vzdialenosti boli navrhnuté vzťahy:

$$DIST_{max} = 0 \Rightarrow q_{dist} = Q_{DIST} \tag{3}$$

$$d > DIST_{max} \Rightarrow q_{dist} = 0 \tag{4}$$

$$d \leq DIST_{\max} \Rightarrow q_{dist} = Q_{DIST} - d \frac{Q_{DIST}}{DIST_{MAX}} \quad (5)$$

kde $DIST_{\max}$ je maximálna vzdialenosť doletu včely, Q_{DIST} je maximálna hodnota čiastkovej kvality a d je vzdialenosť od pôvodnej stránky.

2. Kvalita počtu výskytov

Aby sa rozlíšil počet, koľko krát sa výraz na stránke objavil, bola navrhnutá funkcia, ktorej hodnota na začiatku rýchlejšie rastie, potom sa rast spomalí a asymptoticky sa blíži k definovanej maximálnej hodnote kvality. Funkcia má tvar:

$$q_{count} = \frac{-1}{2 \left(n + \frac{1}{2Q_{COUNT}} \right)} + Q_{COUNT} \quad (6)$$

kde n je počet výskytov daného slova a Q_{COUNT} je maximálna hodnota definovaná pre túto čiastkovú kvalitu.

3. Kvalita výskytu v nadpise

Pri vyhľadávaní sa dá očakávať, že výraz vyskytujúci sa v nadpise je zaujímavejší ako ten istý výraz vyskytujúci sa v článku. V takomto prípade je dôležité sledovať, ako hierarchicky vysoko sa slová v nadpisoch vyskytujú, nie celkový počet výskytov. HTML definuje nadpis celej stránky ako $\langle title \rangle$ a $\langle h_1 \rangle$ až $\langle h_6 \rangle$ pre nadpisy a podnadpisy stránok. Každéj úrovni bolo priradené číslo 0($\langle title \rangle$) až 6($\langle h_6 \rangle$). Bola navrhnutá funkcia:

$$q_{header} = Q_{HEADER} - h * \frac{Q_{HEADER}}{HEADER_{MAX} + 1} \quad (7)$$

kde h je minimum zo všetkých hodnôt nadpisov, v ktorých sa hľadané slovo vyskytuje; Q_{HEADER} je maximálna hodnota tejto čiastkovej kvality; $HEADER_{MAX}$ je najväčšia hĺbka v hierarchii nadpisov zobrazená pri výpočte do úvahy.

V tomto modeli je kvalita rozdelená do troch častí, a to kvalita počtu, nadpisov a vzdialenosti. Kvalita počtu reprezentuje počet výskytov hľadanej frázy v tele dokumentu (okrem zdrojového kódu), kvalita nadpisov počet výskytov frázy v nadpisoch, pričom zohľadňuje aj ich dôležitosť a kvalita vzdialenosti klesá s rastom vzdialenosti nájdennej stránky od preddefinovaných. Autori si vybrali dva druhy určenia, ktorá stránka je najlepšia a najviac spĺňa používateľove predpoklady:

- počet včiel, ktoré tancujú pre danú stránku na konci experimentu,
- celkový počet včiel, ktoré tancovali pre danú stránku počas celého experimentu.

Priklonili sa k druhej alternatíve, ale podotkli, že v reálnom vyhľadávaní by bolo rozumnejšie použiť prvú možnosť, keďže pri vyhľadávaní nás zaujíma najmä výsledok a nie jeho proces.

5.1 Sledovanie vyvíjajúceho sa príbehu

Ak sa čitateľ zaujíma o nejakú konkrétnu udalosť, ktorá je mediálne sledovaná, vyhľadávanie, triedenie a spracovanie informácií je časovo veľmi náročné. Je potrebné neustále monitorovanie postupného vývinu udalosti. Pre takéhoto používateľa je vhodné vytvoriť systém, ktorý by niektoré tieto úlohy riešil automaticky.

Naším cieľom bolo vytvoriť systém, ktorý je schopný vyhľadávať a sledovať informáciu na osobnom počítači niekoľko hodín až dní, zamerať vyhľadávanie na sľubné oblasti a neviaznuť pritom v lokálnom extréme.

Po zozbieraní je nutné množinu informácií filtrovať a zoskupovať. Spravodajské portály obsahujú predovšetkým články, komentáre, diskusiu, blogy, ankety. Z hľadiska sledovania informácie sú dôležité články. Komentáre, blogy, ankety spracúvame, aby sme získali názory na určité udalosti, prípadne ponúkaný tovar [24]. Vyhľadané a vytriedené informácie sa zoradili podľa dátumu publikovania, čím postupne vzniká príbeh s časovou následnosťou.

Inšpirovaní prácou [17] sme použili koncept energie z agenta InfoSpider. Včela prehľadáva okolie, pokiaľ má dostatok energie. Ak nájde kvalitný zdroj potravy, jej energia sa zvýši, pri návšteve každého zdroja sa energia spotrebuje. Ak energia klesne na nulu, včela sa musí vrátiť do úľa.

Kvalitu zdroja vypočítame ako súčet čiastkových kvalít (kvalita počtu výskytov, kvalita výskytu v nadpise).

Biologické pozorovania ukazujú [2], že včely sa pri zbieraní potravy delia na prieskumníčky (scouts) a nasledovníčky (recruits). Prieskumníčky hľadajú potravu samostatne, nasledovníčky sa pri zbieraní riadia informáciami od iných včiel. Podľa teórie neúspešnej nasledovníčky [1] sú prieskumníčky vlastne neúspešné nasledovníčky, teda včely, ktoré v tančiarni nenašli včelu, ktorú by sa rozhodli nasledovať. Ak je dostatok potravy, je veľká pravdepodobnosť, že včela sa stane nasledovníčkou. Ak je však potravy nedostatok, včely sa stávajú prieskumníčkami a hľadajú vlastný zdroj.

V našom modeli sa včela v auditórii dá považovať za voľnú včelu. Ak si vyberie nejakú včelu z tančiarnie a bude nasledovať jej zdroj, stane sa nasledovníčkou. Ak si nevyberie žiadnu včelu a pôjde do dispečernej, stáva sa prieskumníčkou. Prieskumníčky a nasledovníčky sa vo vnútri úľa správajú rovnako, mimo úľa sa ich správanie mení. Prieskumníčka po objavení zdroja s nenulovou kvalitou sa vráti do úľa a pôjde ho propagovať.

5.2 Prípadová štúdia

Príbeh, ktorý sme sa rozhodli pre experimentálne účely sledovať, boli prezidentské voľby na Slovensku. Okrem časovej incidencie, t.j. faktu, že sa takýto príbeh odvíjal práve v čase, keď sme už mali metódu pripravenú na experimentovanie a aj sme potrebovali uskutočniť experiment, zavážilo aj to, že išlo o príbeh, ktorý bol dostatočne sledovaný verejnosťou a médiami, vrátane informácií na webe. Priebeh volieb sme sledovali trikrát – po prvom kole volieb, počas kampane pred druhým kolom, počas volebného dňa a v noci počas druhého kola volieb.

Štartovacie stránky predstavovali spravodajské portály:

www.sme.sk a www.pravda.sk.

Nastavené parametre modelu sú uvedené v tabuľke 1 a parametre počítania kvality v tabuľke 2.

Tabuľka 1. Parametre modelu

| | |
|----------------------------|--------|
| Počet včiel | 20 |
| Maximálna dĺžka tancovania | 10 kôl |
| Maximálny čas v hľadisku | 5 kôl |
| Počiatočná energia | 1 |
| Úbytok energie | 0.05 |

Tabuľka 2. Parametre počítania kvality

| | |
|---|----------------|
| Počiatočná energia | 1 |
| Prírastok energie | Kvalita zdroja |
| Úbytok energie pri prechode na nový zdroj | 0.02 |
| Maximálna kvalita počtu výskytov | 0.8 |
| Maximálna kvalita výskytu v nadpise | 0.2 |
| Maximálne číslo nadpisu | 3 |

Po prvom kole prezidentských volieb postúpili dvaja kandidáti: Iveta Radičová a Ivan Gašparovič. Cieľom prvej časti štúdie bolo sledovať aktivity jedného z kandidátov - Ivety Radičovej.

Experiment bežal v noci z 25. na 26. marca. Hľadanými kľúčovými slovami boli rôzne tvary mena Radičová. Počas experimentu sa našlo 899 stránok, z toho 215 s nenulovou kvalitou. Pri ručnom prehľadávaní bolo iba 81 označených za stránky s relevantným obsahom a z nich 34 ako stránky s informačným obsahom. Relevantné stránky patrili do piatich kategórií: informačná stránka (42), zoznam stránok (21), diskusia (11), grafický obsah (4), komentáre (3).

Nájdene stránky publikovali v rôznych dátumoch a včely zozbierali informácie o pohybe Ivety Radičovej od 10. februára do času krátko pred ukončením experimentu. V období medzi prvým a druhým kolom volieb sa na južnom Slovensku objavili letáky sľubujúce v mene Ivety Radičovej autonómiu pre túto oblasť. Cieľom druhej časti štúdie bolo sledovať túto udalosť.

Experiment prebiehal 30. marca od 12:45 do 19:30. Hľadanými kľúčovými slovami boli rôzne tvary slov letáky a južné Slovensko.

Včely objavili 3557 stránok, z ktorých iba 50 malo nenulovú kvalitu. Relevantných bolo 37 stránok, informačný obsah malo 8 stránok. Stránky boli zaradené do kategórií: diskusia (21), informačná stránka (8), zoznam stránok (6), komentáre (2).

Cieľom poslednej prípadovej štúdie bolo vytvoriť príbeh o dvoch hlavných kandidátoch počas volebného dňa a postupné zverejnenie výsledkov. Vyhľadávanie začalo od 4. 4. 2009 o 14:30 a bolo ukončené v nasledujúci deň ráno o 7:00. Hľadané kľúčové slová boli rôzne tvary slov výsledky volieb a mená kandidátov.

Bolo vyhľadaných 4615 rôznych stránok z toho 742 s nenulovou kvalitou, relevantných bolo 217 stránok. Stránky boli podľa obsahu rozdelené do 7 kategórií: informačná stránka (85), zoznam stránok (66), diskusia (29), blog (18), grafický obsah (11), komentáre (5), anketa (3).

Zaujímavosťou z nájdených stránok je, že medzi vyhľadanými stránkami sa objavili i portály písané v cudzom jazyku – v angličtine písaný www.spectator.sk a v maďarčine písané www.ujso.sk. Tieto stránky boli ponechané ako relevantné, nakoľko sa vyjadrovali k téme.

Z výsledkov vyplýva, že algoritmus je schopný sledovať informáciu, a túto informáciu aj spätne vyskladať. Rozlišovanie rôzneho obsahu stránky a spresnenie určenia relevantnosti stránky je problémom, ktorému sa v súčasnosti venujeme.

6 Stochastická optimalizácia funkcií

Správanie sa včiel inšpirovalo viaceru autorov. Ich záujem sa sústredil najmä na:

- modelovanie a simuláciu správania sa včiel s cieľom čo najpresnejšie zachytiť správanie sa biologických včiel,
- konštruovanie algoritmov, ktoré sú inšpirované správaním sa včiel.

Inšpirovali sme sa modelom [26]. Je to včelí algoritmus na stochastickú numerickú optimalizáciu matematických funkcií. Výsledky tohto algoritmu sú veľmi dobré a pri experimentoch sa porovnávajú s výsledkami iných algoritmov.

Ďalší včelami inšpirovaný algoritmus opísali v [34]. Autori úspešne aplikovali algoritmus na problém obchodného cestujúceho. K algoritmom používajúcim včeliu metaforu z oblasti optimalizácie patria aj práce [31] a [11].

Model zobrazený na obrázku 6 predstavuje všeobecný model správania sa včiel v úli. Pri jeho použití na optimalizáciu funkcií je potrebné dodefinovať správanie sa včiel mimo úľa. Zdroje v tomto prípade sú rôzne vektory hodnôt argumentov funkcie.

6.1 Nové parametre modelu

Do modelu sme zaviedli dva nové parametre:

1. parameter veľkosti kroku: včely mimo úľa môžu navštíviť viac ako jeden zdroj pred návratom do úľa. Keď včela prechádza zo zdroja na zdroj, pripočíta každému argumentu funkcie náhodnú hodnotu z intervalu $\langle -\text{veľkostkroku}, +\text{veľkostkroku} \rangle$.
2. parameter energie: k použitiu tohto argumentu nás inšpirovala práca [17]. Je to energia na aktivitu včely mimo úľa. Ak včela stratí energiu, vracia sa do úľa bez zdroja.

6.2 Zisťovanie kvality zdroja

Pri zisťovaní a výpočte kvality zdroja sa použili dva aspekty: použitie rankingu a mapovanie všetkých hodnôt optimalizovanej funkcie z $< -\infty, \infty >$ do $< 0, 1 >$.

Pretože navrhovaný model môže pracovať iba s hodnotami z intervalu $< 0, 1 >$ a funkcie sú vo všeobecnosti definované na intervale $< -\infty, \infty >$, potrebujeme funkciu, ktorá môže mapovať každú funkčnú hodnotu do intervalu $< 0, 1 >$. Nazvime túto funkciu $\text{Map}(x)$.

Funkcia musí vyhovovať podmienkam uvedeným v (8), (9), (10).

$$\forall x, y \in R : x < y \Leftrightarrow \text{Map}(x) < \text{Map}(y) \quad (8)$$

$$\forall x, y \in R : x > y \Leftrightarrow \text{Map}(x) > \text{Map}(y) \quad (9)$$

$$\forall x, y \in R : x = y \Leftrightarrow \text{Map}(x) = \text{Map}(y) \quad (10)$$

Napríklad funkcia uvedená v (11) spĺňa tieto podmienky.

$$\text{Map}(x) = \frac{1}{\pi} \left(\text{arctg}(x) + \frac{\pi}{2} \right) \quad (11)$$

Ak hľadáme globálne maximum, použije sa táto funkcia, pre globálne minimum použijeme vzorec $1 - \text{Map}(x)$.

Funkcia je nelineárna, čo však nie je problémom, pretože model využíva ranking.

Špecifický vzorec, ktorý sme použili, je:

$$\text{rank} = \frac{1}{1 - NB} * ((1 - \varepsilon) * P + \varepsilon - NB) - k \quad (12)$$

kde NB je počet včiel, ε je malá konštanta (použili sme hodnotu 0.001), P je pozícia danej kvality v poradí zoradených všetkých kvalít indexovaná od 1, k je konštanta, ktorá zabezpečuje, aby žiadna kvalita nemala hodnotu 1 (použitá hodnota 0.2).

6.3 Výsledky experimentov

Pri použití základného modelu a modifikácií opísaných v predchádzajúcej časti sme nechali náš algoritmus, aby sa pokúsil optimalizovať funkcie zobrazené v tabuľke 3. Tieto funkcie optimalizovali v [26].

V tabuľke 4 uvádzame výsledky zobrazené v [26] doplnené o výsledky našich experimentov [23].

Tabuľka 3. Funkcie a ich parametre

| ID | Function name | Interval | Global optima |
|----|-------------------|-----------------|---|
| 1 | Rosenbrock 2D | [-1.2, 1.2] | X(1,1) F=0 |
| 2 | Rosenbrock 2D | [-10,10] | X(1,1,1,1) F=0 |
| 3 | Goldstein & Price | [-2,2] | X(0,-1) F=3 |
| 4 | Martin & Gaddy | [0,10] | X(5,5) F=0 |
| 5 | Rosenbrock 4D | [-1.2, 1.2] | X(1,1,1,1) F=0 |
| 6 | De Jong | [-2.048, 2.048] | X(1,1) F=3905.93 |
| 7 | Branin | [-5, 10] | X(-22/7, 12.275) X(22/7, 2.275) X(66/7, 2.475) F=0.3977272 |
| 8 | Hyper Sphere | [-5.12, 5.12] | X(0,0,0,0,0,0) F=0 |

Tabuľka 4. Experimentálne výsledky získané ako priemerná hodnota z výsledkov experimentov, ktoré sa 100 krát opakovali. S: Simplex Method, NS: Stochastic Simulated Annealing Optimisation Procedure, GA: Genetic Algorithm, ANT: Ant Colony System, BA: Bees' Algorithm, BHM: nami navrhnutý včelí algoritmus, *****: dáta neboli uvedené.

| ID | S | NS | GA | ANT | BA | BHM |
|----|-------|-------|-------|-------|-------|-------|
| 1 | 10780 | 4508 | 10212 | 6842 | 631 | 2409 |
| 2 | 12500 | 5007 | ***** | 7505 | 2306 | 16019 |
| 3 | ***** | ***** | 5662 | 5330 | 999 | 6773 |
| 4 | ***** | ***** | 2844 | 1688 | 526 | 645 |
| 5 | 21177 | 3053 | ***** | 8471 | 28529 | 68249 |
| 6 | ***** | ***** | 10160 | 6000 | 868 | 6699 |
| 7 | ***** | ***** | 7325 | 1936 | 1655 | 1822 |
| 8 | ***** | ***** | 15468 | 22050 | 7113 | 17152 |

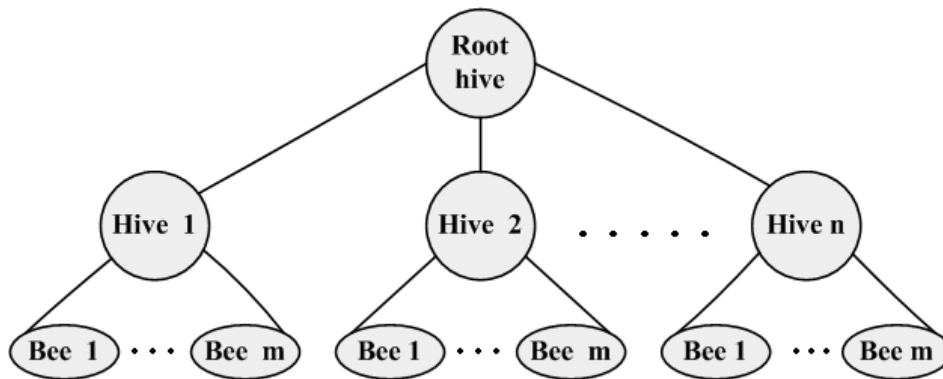
7 Hierarchia včelieho úľa

Existujú algoritmy, ktoré používajú rôzne druhy hierarchie. Napríklad hierarchický optimalizačný algoritmus časti roja [8] využíva hierarchiu častí na vyriešenie problému s predčasnou konvergenciou roja. Oddeluje tu lepšie riešenia od horších do rôznych hierarchických stupňov.

Práca [7] sa tiež zaoberala konceptom hierarchie. V [7] použili jednoduchý príklad hierarchického problému: daný je obrazec v tvare šesťuholníka, ktorý sa skladá z iných šesťuholníkov a cieľom je nájsť symetrický tvar zložený z týchto obrazcov. Na uskutočnenie tejto úlohy autor použil simulované žíhanie bez hierarchie.

Náš koncept hierarchie je z väčšej časti založený na práci [7]. Náš model včelieho úľa sa delí na dve časti – úľ a včely.

Obrázok 11 ukazuje konštrukciu hierarchie: vo vyššej, nie základnej úrovni, úľ reprezentuje ostatné úle, zatiaľ čo iba na základnej úrovni úľ reprezentujú pravé včely.



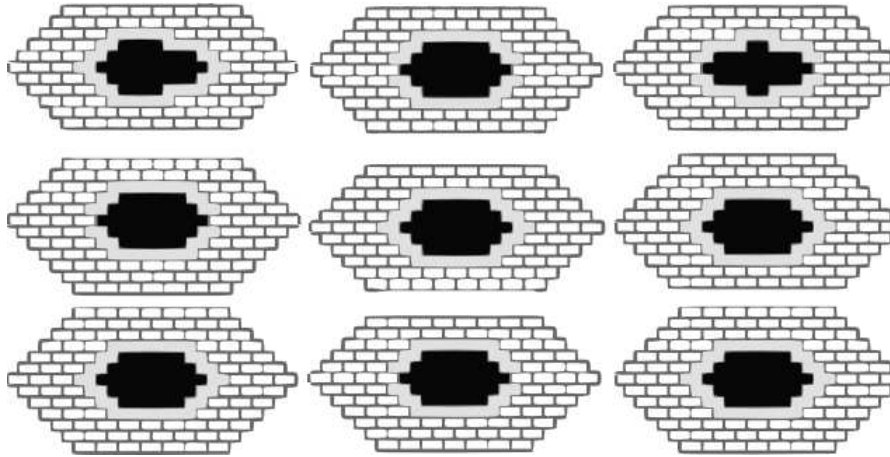
Obr. 11. Hierarchia včelieho úľa.

Vyššia úroveň úľa je vďaka nášmu návrhu stále úľom, pretože pracuje ako úľ, aj keď reprezentuje ostatné úle, ale tie sú zase navrhnuté tak, aby sa správali ako včely. Logickým odôvodnením nášho návrhu je, aby úle vyššej úrovne boli schopné ovplyvniť správanie úľov nižšej úrovne (bez potreby vedomosti, či je to včela alebo úľ).

7.1 Experimenty založené na hierarchii

V [7] sa predpokladalo, že počas experimentu sa nájde symetrický tvar. S vyššie popísaným modelom sme previedli sériu experimentov. Zhromaždené výsledky experimentov vidíme na obrázku 12, ktorý ukazuje 9 behov za sebou idúcich experimentov.

Ako môžeme vidieť na tomto obrázku, v každom behu okrem prvého sa dosiahli symetrické výsledky.



Obr. 12. Experimenty zamerané na formovanie symetrického tvaru.

8 Záver

Metafora včelieho úľa sa v súčasnosti dostáva do popredia záujmu vedcov, ktorí sa zaoberajú metódami riešenia rôznych problémov, aj keď prvé pokusy z tejto oblasti sa datujú dve desaťročia dozadu.

Včelí úl je komplexný systém zložený z včiel, ktorých správanie sa definuje pomocou jednoduchých pravidiel.

Experimentálne sa dosiahli dostatočne solídne výsledky na podporu tvrdenia, že metafora včelieho úľa je atraktívnou alternatívou vyžadujúca ďalšie štúdium.

S navrhnutým modelom včelieho úľa sme uskutočnili hľadanie na webe, sledovanie vývoja príbehu spätne na webe, tvarovanie symetrie a stochastickú optimalizáciu funkcií.

Dosiahnuté sľubné výsledky však vyžadujú ďalší výskum pred tým, ako urobíme definitívne závery.

Pod'akovanie: Prácu finančne podporila grantová úloha VEGA 1/0508/09.

Literatúra

- [1] Beekman, M., Gilchrist, A.L., Duncan, M., Sumpter, D.J.T. (2007). *What makes a honeybee scout?* Behavioral Ecology and Sociobiology: 61:pages 985- 995, 2007.
- [2] Biesmeijer, J.C., de Vries, H. (2001). *Exploration and exploitation of food sources by social insect colonies: A revision of the scout-recruit concept.* Behavioral Ecology and Sociobiology, 49(2-3), pages 89-99, 2001.

-
- [3] Bonabeau, E., Theraulaz, G., Deneubourg, J.L., Aron, S., Camazine, S. (1997). *Self-organization in social insects*. Trends in Ecology and Evolution, vol 12, pp. 188-193.
 - [4] Bonabeau, E., Thraulaz, G., Dorigo, M. (1999). *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press.
 - [5] Camazine, S., Sneyd, J. (1991). *A model of collective nectar source selection by honey bees: Self-organization through simple rules*. Journal of Theoretical Biology, 149(4):547-571.
 - [6] Camazine, S., Deneubourg, J.L., Franks, N.R. ET AL. (2003). *Self-organization in Biological Systems*. Princeton University Press.
 - [7] Cerny, V. (2000). *Searching for the Best Model of the World from Simulated Annealing to Hierarchical Systems (in Slovak)*, IRIS Bratislava , pages 27 - 44, 2000.
 - [8] Chuan, L., Quanyuan, F. (2007). *A Hierarchical Subpopulation Particle Swarm Optimisation Algorithm*, In Proc.: ISKE-2007 Proceedings, Advances in Intelligent Systems Research, October 2007.
 - [9] Gadagkar, R.: (1996): *The Honeybee Dance-Language Controversy: Robot Bee Comes to the Rescue*. Resonance, January 1996.
 - [10] Gasparetti, F., Micarelli, A. (2004). *Swarm Intelligence: Agents for Adaptive Web Search*. 16th European Conference on Artificial Intelligence. 2004, s. 1019-1020.
 - [11] Karaboga D. (2005). *An Idea Based on Honey Bee Swarm For Numerical Optimisation*, Technical Report-TR06. Erciyes University, Engineering Faculty, Computer Engineering Department 2005.
 - [12] Karaboga, D., Akay, B.(2009). *A survey: algorithms simulating bee swarm intelligence*. Artificial Intelligence Review, 2009 - Springer
 - [13] Lorenzi, F., Ricci, F. (2004). *Case-based reasoning and recommender systems*. Technical report, IRST.
 - [14] Lorenzi, F., Sherer Dos Santos, D., Bazzan, A.L.C. (2005). *Case-based Recommender Systems: a unifying view*. In LNCS/LNAI State-of-the-Art Survey book on Intelligent Techniques in Web Personalization.
 - [15] Lorenzi, F., Sherer Dos Santos, D., Bazzan, A.L.C. (2005). *Case-based recommender systems inspired by social insects*. In XXV Congresso da Sociedade Brasileira de Computacao, 22-29 jul 2005
 - [16] Lorenzi, F., Sherer Dos Santos, D., Bazzan, A.L.C. (2005). *Negotiation for task allocation among agents in case-based recommender systems: a swarm-intelligence approach*. In IJCAI-05 Workshop on Multi-Agent Information Retrieval and Recomander Systems. Edinburg, page 23-27, 2005.
 - [17] Menczer, F., Monge, A.E. (1999). *Scalable Web Search by Adaptive Online Agents: An InfoSpiders Case Study*. Intelligent Information Agents, Springer.

- [18] Návrat, P., Kováčik, M. (2006). *Web Search Engine as a Bee Hive*. In: 2006 IEEE/WIC/ACM International Conference on Web Intelligence, Hong Kong, Los Alamitos, IEEE Computer Society, page 694-701.
- [19] Návrat, P., Kováčik, M., Bou Ezzeddine, A., Rozinajová, V. (2007). *Vyhľadávanie informácií pomocou včiel*. In: Znalosti 2007, Ostrava, ISBN 978-80-248-1279-3, pages 63-74.
- [20] Návrat, P., Kováčik, M., Bou Ezzeddine, A., Rozinajová, V. (2007). *Metafora včelieho úľa - model vyhľadávania a odporúčania informácií*. Kognícia a umelý život 2007, Smolenice, pages 249-258.
- [21] Návrat, P., Jastrzemska, L., Jelínek, T., Bou Ezzeddine, A., Rozinajová, V. (2007). *Exploring Social Behaviour of Honey Bees Searching on the Web*. In: Y. Li, V.V. Raghavan, A. Broder, H. Ho: 2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology (Workshops), Silicon Valley, USA, 2-5 November 2007. - Los Alamitos USA : IEEE Computer Society, 2007, ISBN 0-7695-3028-1, pages 21-25.
- [22] Návrat, P., Kováčik, M. Bou Ezzeddine, A. Rozinajová, V. (2008). *Web search engine working as a bee hive* In: Web Intelligence and Agent Systems: An International Journal. Amsterdam : IOS Press, 2008. – ISSN 1570-1263. - Vol.6, s. 441-452
- [23] Návrat, P., Jelínek, T., Jastrzemska, L. (2009). *Bee Hive At Work: A Problem Solving, Optimizing Mechanism*. World Congress on Nature and Biologically Inspired Computing (NaBIC'09) & 8th International Conference on Computer Information Systems and Industrial Management Applications (CISIM2009) PSG College of Technology, Coimbatore, India Dec 09-11, 2009.
- [24] Návrat, P., Bou Ezzeddine, A., Slizik, L. (2010). *Mining Overall Sentiment in Large Sets of Opinions*, AWIC2009, In. Advances in Intelligent Web Mastering - 2, AISC 67, pp. 167–173. Springer-Verlag Berlin Heidelberg 2010.
- [25] Page L. , Brin S. (1998). *The Anatomy of Large-Scale Hypertextual Web Search Engine*. Computer Networks and ISDN Systems, 30 (1-7): pages 107-117, 1998.
- [26] Pham D.T., A. Ghanbarzadeh, E. Koc, Otri S., Rahim S., Zaidi, M. (2006). *The Bees Algorithm – A Novel Tool for Complex Optimisation Problems*, http://conference.iproms.org/the_bees_algorithm_a_novel_tool_for_complex_optimisation_problems
- [27] Resnick, P. ET AL. (1994). *An open architecture for collaborative filtering of netnews*. In Proceedings ACM Conference on Computer-Supported Cooperative Work, pages 175-186.
- [28] Schafer, J.B., Konstan, J.A., Riedl, J. (2001). *E-commerce recommendation applications*. Data Mining and Knowledge Discovery, 5(1/2):115-153.
- [29] Selley, D., Camazine, S., Sneyd, J. (1991). *Collective decision-making in honey bees: How colonies choose nectar sources*. Behavioral Ecology and Sociobiology, 28:277-290.

- [30] Sumpter, D. J. T., Beekman, M. (2003). *From nonlinearity to optimality: pheromone trail*. Animal Behaviour. 2003, s. 273-280.
- [31] Teodorovic, D., Dell'Orco, M. (2005). *Bee Colony Optimisation: Cooperative Learning approach to Complex Transportation Problems*, In Proc.:16th Mini - EURO Conference and 10th Meeting of EWGT. Poznan 2005.
- [32] Vries, H. - Biesmeijer, J.,C. (1998). Modelling collective foraging by means of individual behaviour rules in honey-bees. Behav Ecol Sociobiol 44: 109-124.
- [33] White, T. a Pagurek, B. (1998). *Towards Multi-Swarm Problem Solving in Networks*. Multi Agent Systems - International Conference on Volume. 1998.
- [34] Wong, Li-Pei, Chi Yung Puan, Malcolm Yoke-Hean Low, Chin Soon Chong (2008). *Bee Colony Optimisation algorithm with Big Valley landscape exploitation for Job Shop Scheduling problems*. Winter Simulation Conference 2008: pages 2050-2058.

Vnímanie a pomenovávanie farieb a farebných kategórií

Kristína REBROVÁ a Martin TAKÁČ¹

Abstrakt. Kategorizácia je jednou z mnohých populárnych tém na poli kognitívnych vied pokladaná za esenciálny mechanizmus myslenia a kognície ako takej. Skutočnosť, že vnímanie farieb je univerzálnou črtou ľudskej vizuálnej percepcie, nám umožňuje skúmať a porovnávať kategorizáciu farieb v akejkoľvek kultúre či jazyku, či umelom systéme. Cieľom tejto kapitoly je opísať základné mechanizmy vnímania farieb, teórie o tom, ako ich rozlišujeme a pomenujeme a predstaviť najvýznamnejšiu svetovú štúdiu pomenovávania farieb World Color Survey. Dáta z tejto štúdie boli použité v simulácii kategorizácie farieb na báze sémantiky rozlišovacích kritérií, ktorá uzatvára túto kapitolu ako aplikačný pohľad na problematiku.

1 Úvod

Kategorizácia, teda proces, v ktorom narábame s rozličnými entitami ako s rovnakými, je esenciálnym mechanizmom myslenia a kognície ako takej [32]. Umožňuje nám rozlišovať objekty vo svete a vzťahy medzi nimi. Podieľa sa na vnímaní, chápaní, usudzovaní, plánovaní, reči a rôznych ďalších základných kognitívnych aktivitách. Jednou z možných ciest ku pochopeniu ľudskeho myslenia a usudzovania je skúmanie kategorizácie v jazyku [15]. Skutočnosť, že vnímanie farieb je univerzálnou črtou ľudskej vizuálnej percepcie, nám umožňuje skúmať kategorizáciu a pomenovávanie farieb v akejkoľvek kultúre, jazyku, či umelom biologicky motivovanom systéme a výsledky z takýchto skúmaní navzájom porovnávať, hľadať všeobecné mechanizmy a odlišnosti.

Od začiatku druhej polovice minulého storočia sa sformulovalo viacero hypotéz o tom, ako rozlišujeme a pomenujeme farby. Vo všeobecnosti ich možno rozdeliť do dvoch skupín, na univerzalistické a relativistické [12]. Relativisti tvrdia, že pomenovania pre farby a im zodpovedajúce farebné kategórie vznikajú na základe arbitrárnych jazykových konvencií. Univerzalisti naopak zastávajú názor, že každý

¹ Katedra aplikovanej informatiky, Fakulta matematiky, fyziky a informatiky, Univerzita Komenského, Mlynská dolina, 824 48 Bratislava, E-mail: kristina.rebrova@fmph.uniba.sk, takac@ii.fmph.uniba.sk

jazyk obsahuje konečné množstvo základných farebných termínov, pričom zodpovedajúce farebné kategórie, charakteristické ich farebným rozsahom a najlepšimi reprezentantmi (prototypmi), sú univerzálne pre všetky jazyky a kultúry na svete.

Cieľom tejto kapitoly je uviesť čitateľa do problematiky vnímania, rozlišovania a pomenovania farieb, predstaviť základy, na ktorých stojí univerzalistický pohľad na kategorizáciu farieb a s ním súvisiaci model kategorizácie farieb inšpirovaný prototypovou teóriou [24]. Najprv popíšeme základné teórie o vnímaní farieb, ktoré boli sformulované ešte pred objavením svetlocitlivých buniek a vizuálnych neurálnych ciest, no napriek tomu dobre popisujú princíp fungovania vnímania farieb ako taký. Najmä druhá z nich, teória opozičných procesov, zohráva dôležitú úlohu pri skúmaní kategorizovania farieb. Základné farby, ktoré definuje, sa považujú za jadro univerzálnej množiny základných farebných kategórií a opozičné procesy, ktoré medzi nimi pôsobia, za mechanizmus na pozadí vzniku farebných termínov v jazyku.

Dalej popíšeme fyziológiu a fenomenológiu farebného vnímania, aspekty dôležité pre získanie jednotného pohľadu na vnímanie a rozlišovanie farieb, teda na ich perceptuálnu kategorizáciu. Zdanlivým odbočením bude časť o farebných modeloch a priestoroch, ktorá hovorí o tom, ako farby reprezentujeme matematickým spôsobom. Možnosť exaktnej reprezentácie farieb je však veľmi dôležitá pre to, aby sme ich mohli objektívne porovnávať, zaznamenávať a skúmať naše reakcie na ne. Túto obsahovo najbohatšiu časť kapitoly zavŕšime krátkym úvodom do fenoménu kategorizácie ako takej, prototypovej teórie a lexikálnej kategorizácie farieb s priamou nadväznosťou na tretiu časť. Tá už bude priamo pojednávať o hlavnej teórii o základných farebných kategóriách formulovanej v roku 1969 a o svetovej farebnej štúdií (WCS) vytvorenej pre získanie silnej empirickej podpory pre túto teóriu. Popíšeme jej metodológiu, výsledky a ich využitie, teda najvýznamnejšie štúdie pre potvrdenie univerzalistických hypotéz.

V poslednej časti predstavíme model kategorizácie farieb založený na prototypovej teórii, konkrétne na sémantike rozlišovacích kritérií a z neho vytvorenú simuláciu natrénovanú na dátach z WCS. Cieľom tejto časti je nielen ukázať vhodný mechanizmus pre modelovanie kategorizácie farieb, ale tiež overiť kategorizačnú schopnosť rozlišovacích kritérií pomocou reálnych dát.

2 Vnímanie farieb

Otázka definovania farby ako takej je v mnohých ohľadoch diskutabilná. O farbe možno povedať, že je to vlastnosť alebo vedľajší produkt svetla – elektromagnetického žiarenia s vlnovou dĺžkou od 380nm do 740nm, ktoré dopadá na sietnicu oka a je ďalej spracúvané perцепčným aparátom, ktorý produkuje zmyslový vnem [32]. Keď hovoríme o farbe objektu, myslíme tým vlastne vlnovú dĺžku svetla, ktoré tento objekt odráža. Je všeobecne známe, že materiál, o ktorom hovoríme, že je svetlý, odráža viac svetla ako tmavý materiál. Takto popisujeme farbu na základe jej vzniku, vnemu a fyzikálneho pôvodu.

Farba je však tiež vlastnosťou ľudského mozgu alebo mysle. Farbu si totiž vieme predstaviť aj bez použitia perцепčného systému. Preto väčšinou definujeme farbu nepriamo ako psychologický vnem s tromi komponentmi – odtieňom, jasom

a saturáciou [9]. Odtieň sa vzťahuje na kvalitu tohto vnemu, teda určitú farebnú hodnotu, ktorú vyjadrujeme rôznym spôsobom. Jas alebo svetlosť sú dva pojmy, ktoré vystihujú to, aké množstvo svetla zdanlivo vychádza z objektu, na ktorom farbu pozorujeme. Jas charakterizuje vlastnosť farby ako takej a svetlosť zas vlastnosť povrchu objektu, na ktorom farbu pozorujeme. Saturácia, chroma alebo farebnosť sú tri pomenovania pre sýtosť resp. šedosť farby. Čím je saturácia vyššia, tým je farba sýtejšia.

Rozsah elektromagnetického žiarenia, ktoré je pozorovateľné ľudským okom, nazývame viditeľné (optické) spektrum alebo jednoducho svetlo. Bežným omylom je predstava, že vlnová dĺžka pôsobiaceho svetla priamo určuje, akej farby budú predmety, ktoré toto svetlo osvetľuje. Je tomu tak väčšinou len v prípade takzvanej voľnej farby, kedy uvažujeme akoby absolútnu hodnotu svetla resp. farby bez porovnania s okolím. No v princípe náš percepčný systém nevie vnímať absolútne vlnové dĺžky alebo jas, iba ich porovnávať vzhľadom na rôzne časti scény. Text v tejto časti (2.1 až 2.5) bude vychádzať prevažne z [6] a [8].

2.1 Základné teórie vnímania farieb

Počas histórie skúmania farebného videnia boli popísané dva základné a navzájom si odporujúce princípy, a to trichromatická teória a teória opozičných procesov.

Trichromatická alebo Young-Helmholtzova teória vnímania farieb je založená na pokusoch s miešaním farieb a tvrdí, že v oku existujú práve tri typy farebných receptorov (červený, modrý, zelený), ktoré posielajú do mozgu informáciu o farbe vnímaného svetla a tiež to, že tieto 3 farebné kanály sú potrebné, ale aj postačujúce pre namiešanie akejkoľvek farby. Napriek tomu, že sa opiera o výsledky z mnohých experimentov, a že ju môžeme nájsť aj vo fyziológii ľudského oka, trichromatická teória nevie vysvetliť existenciu fenoménu opozičných farieb (viď nižšie) a farebných paobrazov². Tiež nevie vysvetliť to, že ľudia s poruchou vnímania jednej zo základných farieb, dichromati (viď nižšie) môžu vidieť bielu a žltú farbu, aj napriek tomu, že ich bez tretieho kanálu nie je možné namiešať.

Teória opozičných procesov tiež hovorí o troch základných kanáloch farebného vnímania, ktoré sú však úplne inej povahy. Ide o kanály párov takzvaných opozičných farieb, ktoré nikdy nie je možné vnímať súčasne. Sú to zelená – červená, modrá – žltá a čierna – biela. Neexistuje nič také ako červenkastá zelená alebo žltkastá modrá. Zmiešaním týchto dvoch opačných farieb dostaneme farbu neutrálnu, teda pri akomkoľvek miešaní neutralizuje jedna z páru opačných farieb druhú. Tento princíp je konzistentný s farebnými paobrazmi (červený obraz indukuje zelený paobraz) a korešponduje aj s ľudským vnímaním. Reakcie na opačné páry farieb sa pri

² Termín paobraz alebo afterimage sa používa na pomenovanie efektu, pri ktorom sa nám pred očami zjaví niečo ako negatív pozorovaného obrazu – paobraz, ktorý nie je výsledkom priameho vizuálneho vnemu (môžeme pri tom aj zavrieť oči) ale predchádzajúceho zosilneného vnemu, napríklad jasnosťou obrazu a dĺžkou vystavenia obrazu oku. Tento efekt sa používa napríklad pri optických ilúziách. Pri vnímaní farieb spôsobuje silný vnem jednej z opozičných unikátnych farieb – zelenej, červenej, modrej, žltej, čiernej a bielej paobraz opačnej farby.

spracúvaní fotoreceptormi navzájom inhibujú. Opozičné procesy fungujú aj na úrovni neurálnych ciest.

Napriek tomu, že mali tieto teórie sprvu snahu si konkurovať, ani jedna z nich nie je samostatne postačujúca pre vysvetlenie celého princípu farebného vnímania. Ako bližšie popíšeme v ďalšej časti, v ľudskom oku sú skutočne práve tri typy farebných receptorov, avšak ich vzájomné pôsobenie umožňuje aj opozičné procesy, ktoré sa ďalej prenášajú vyššie do mozgu a pôsobia aj na konečné spracovanie farebného vnemu.

2.2 Fyziológia farebného vnímania

Ľudský vizuálny perцепčný systém funguje na princípe absorpcie svetla špeciálnymi receptormi, bunkami citlivými na svetlo [18]. Fotoreceptory spracúvajú jednotlivé dávky svetelnej energie, čiže fotóny dopadajúce na sietnicu. Rozlišujeme dva druhy fotoreceptorov – čapíky a tyčinky – pomenované podľa ich charakteristického tvaru. Za vnímanie farieb zodpovedajú čapíky, ktoré sú citlivejšie, ale aj náročnejšie na svetlo.

Na rozdiel od väčšiny cicavcov vníma človek farby tromi druhmi čapíkov. Tie možno rozlíšiť podľa vlnovej dĺžky svetla, na ktoré reagujú. Delíme ich na červené, zelené a modré alebo L, M a S-čapíky. Červené alebo L-čapíky (long-wavelength sensitive cones) vnímajú svetlo s veľkou vlnovou dĺžkou od 500 nm po 700 nm s vrcholom citlivosti okolo 564-580 nm. Zelené M-čapíky reagujú na o niečo kratšiu vlnovú dĺžku, 450-630 nm s vrcholom okolo 534-545 nm a modré S-čapíky na svetlo krátkej vlnovej dĺžky 400-500 nm s vrcholom okolo 420-440. Všetky majú rôzne, no prelínajúce sa citlivosti. Dôležitým faktom je, že čapík akéhokoľvek druhu môže určiť intenzitu a vlnovú dĺžku svetla k nemu prichádzajúceho len v porovnaní s jedným alebo viacerými inými typmi čapíkov.

Červené a zelené čapíky zabezpečujú vnímanie spektra s väčšími vlnovými dĺžkami a vo všeobecnosti zodpovedajú za rozlišovanie svetlého a tmavého. Krátkovlnné, modré čapíky sa naopak používajú na vnímanie farebného kontrastu, preto ich je asi desaťkrát menej než ostatných. V princípe sú si dva druhy dlhovlnných čapíkov veľmi podobné a majú spoločný farebný základ – žltú farbu. Červené čapíky sú teda červeno-žlté a zelené zeleno-žlté. Tento princíp súhlasí s oboma uvedenými farebnými teóriami a korešponduje aj s rozdelením najsvetlejšej časti farebného spektra, žltej farby, na dve časti – zelenkastú a červenkastú. Trivariancia umožňuje miešaním paralelného zeleno-červeného a modro-žltého systému vnímanie viacerých farieb, ako napríklad magenta (svetlá fialová) alebo cyan (zeleno-modrá, svetlá tyrkysová), ktoré nie sú súčasťou farebného spektra.

Svetelný stimul sa po spracovaní čapíkmi prenáša cez optický nerv do talamu napojeného na bočné kolienkovité telieska (LGN). V ňom sa v súlade s teóriou opozičných procesov prenáša do primárnej vizuálnej kôry (V1) červeno-zelený signál pomocou parvocelulárneho kanálu a modro-žltý pomocou koniocelulárneho [3]. Cez tieto kanály sa môže šíriť signál len pre jednu z páru vzájomne sa inhibujúcich farieb, preto skutočne nie je možné vnímať zelenkastú červenú a podobne. Opozičné procesy fungujú aj v primárnej vizuálnej kôre, kde sa signál spracúva dvojitémi opozičnými

bunkami, ktoré sa združujú v špeciálnych valčekovitých štruktúrach (blobs). Tieto špeciálne opozičné bunky reagujú na množstvo zelenej a červenej farby v rôznych častiach scény s tým, že najlepšie reagujú, ak sú farby objektov kontrastné (zelená vedľa červenej). Z tejto oblasti sa signál šíri ďalej do V2 a V4, až sa dostane do spodnej časti spánkového laloku (IT kôra), ktorá integruje informáciu o farbe s informáciou o tvare a iných priestorových vnemoch. Cesta cez oblasti V1 – V2 – V4 po IT kôru sa nazýva ventrálna alebo „čo“ dráha, ktorá zodpovedá za rozlišovanie farieb, detailov, textúr, tvárí, udržanie pozornosti a celkovo slúži na identifikáciu a kategorizáciu objektov spracúvaných vizuálnym perцепčným systémom [31].

2.3 Poruchy vnímania farieb

Existuje sedem typov porúch vnímania farieb spôsobených odchýlkou v citlivosti alebo úplnou absenciou niektorých druhov farebných čapíkov. Vrodenou poruchou vnímania farieb trpí na svete približne 8% mužov a približne 0.5% žien [25].

Prvou a najbežnejšou je anomálna trichromacia. Človek trpiaci touto poruchou má síce všetky tri druhy farebných čapíkov, avšak jeden z nich je zastúpený v menšom počte, s menším množstvom pigmentu alebo nastavený na neobvyklú vlnovú dĺžku. Preto dochádza k zmenenému farebnému vnemu najmä pri jasnejších alebo viac saturovaných odtieňoch základných farieb, najčastejšie červenej a zelenej. Ak je porucha protanomalická, dochádza k posunu citlivosti na vlnovú dĺžku od červenej k zelenej, teda niektoré objekty, ktoré zdravé oko vidí ako červené (príp. oranžovej alebo podobnej farby), sa javia zelené alebo zelenkasté. Pri najbežnejšej, deuteranomalickej poruche je oslabená zelená farba. Tritanomália, porucha modrých čapíkov, sa vyskytuje len zriedka.

Ďalším typom poruchy vnímania farieb je dichromacia, teda úplná absencia jedného z troch typov čapíkov. O ľuďoch trpiacich touto poruchou možno povedať, že sú skutočne farboslepí, pretože nemôžu za žiadnych okolností vnímať určité farebné odtiene. Niektorí dichromati používajú na rozoznávanie zelenej a červenej farby slabšie receptory – tyčinky, ktoré zdravý človek používa len na rozlíšenie tmavého a svetlého. Pri tejto poruche existujú opäť tri typy protanopia, deuteranopia a tritanopia, teda absencia červených (protanopia), zelených (deuteranopia) alebo modrých (tritanopia) čapíkov.

Posledným typom je monochromacia, veľmi zriedkavá porucha, pri ktorej chýbajú až dva druhy farebných čapíkov. Človek trpiaci touto poruchou vidí akoby čiernobielo s určitým farebným odtieňom podľa toho, ktorý typ čapíkov má. Podobná, tiež málo sa vyskytujúca je achromacia – úplná absencia farebných čapíkov. Človek trpiaci touto poruchou vníma len tyčinkami, je ochudobnený nielen o vnem farby, ale aj o ostrosť a jas videnia.

2.4 Fenomenológia farebného videnia

Pri najlepších optických podmienkach existuje zhruba milión rozlíšiteľných farieb – kombinácií rôznych odtieňov, saturácie a jasu [9]. Ľudské oko je schopné rozlíšiť okolo 150 spektrálnych odtieňov a okolo 450 odtieňov šedej. Tento počet môžeme ešte

obohatiť o mimospektrálne farby, ktoré vznikajú miešaním spektrálnych farieb, zmenou kontrastu a podobne.

Rozlišovanie farieb je ovplyvnené priestorovou separáciou porovnávaných objektov, počtom dimenzií, v ktorých sa farby líšia, polohou farby vo farebnom spektre, veľkosťou pozorovaného objektu, saturáciou a jasom. Tiež záleží na časovom odstupe od vzhladnutia farby. Najlepšie si zapamätáme farby, ktoré sú dobrými reprezentantmi, na ich rozpoznanie nám stačí kratší čas. Lepšie rozlišujeme farby v prípade, že máme medzi daným objektom a jeho farbou vžitú asociáciu, napríklad červené jablko, zelená tráva a podobne. Pri rozpoznávaní farby objektu je dôležitá aj jeho poloha vzhľadom na sieťnicu.

Je dôležité uvedomiť si, že vizuálny percepčný systém rozlišuje farby nie na základe absolútnych vlnových dĺžok, ale porovnávaním vlnových dĺžok a jasú prichádzajúceho z rôznych častí scény, teda ide podobne ako pri iných aspektoch vizuálnej percepcie o rozlíšenie objekt – pozadie.

Farba vnímaného objektu sa odvíja aj od farby pozadia, čiže sa nám môže zdať na rôznych pozadiach vždy trochu iná, aj keď objekt a osvetlenie sú rovnaké. Jedným z efektov, ktoré vyvoláva pozadie objektu, je takzvaný princíp simultánneho farebného kontrastu, pri ktorom môže pozadie vyvolať dojem, že má objekt ľubovoľnej farby farebný nádech doplnkovej farby k farbe pozadia. Napríklad na jasne červenom pozadí sa môže objekt javiť zelenší a na modrom zase žltší. Tento princíp najlepšie funguje vtedy, keď tvorí farba objektu a farba pozadia dvojicu opozičných farieb. Na druhej strane, keď ide o odtiene podobné, môže pozadie vnímanú saturáciu farby objektu znižovať. Teda ak sú objekt a pozadie podobných odtieňov a pozadie je jasné, objekt sa zdá poznateľne šedší než na neutrálnom pozadí. Farba pozadia môže farbu objektu tiež zdanlivo stmaviť alebo zosvetliť. Tmavé pozadie objekt zosvetlí a svetlé stmaví. Ak je objekt a pozadie podobnej farby a svetlosti (napríklad čierny objekt na tmavomodrom pozadí), nastáva takzvaný asimilačný efekt – splynutie objektu s pozadím.

Ďalší z faktorov vplývajúcich na vnímanie farieb je veľkosť pozorovaného objektu, respektíve veľkosť jeho obrazu na sieťnici. Čím je objekt na sieťnici menší, tým je jeho farba ťažšie rozlíšiteľná. Tmavé farby (napr. modrá) konvergujú k čiernej a svetlé (napr. žltá) zas k bielej.

Jasnosť farby závisí na mnohých faktoroch. Ovplyvňuje ju svietivosť materiálu, pozadie, vlastnosti osvetlenia a prispôsobivosť pozorovateľa naň, dĺžka trvania pohľadu na daný objekt a jeho veľkosť. Bezold-Brückov posun je jav, pri ktorom dochádza k zmene vnímania farebného odtieňa spôsobenej zmenou intenzity osvetlenia. So stúpajúcim jasom osvetlenia sa spektrálne farby s vlnovou dĺžkou pod 500nm posúvajú k modrej a nad 500nm k žltej (napr. objekt červenej farby sa bude javiť červeno-oranžový).

Pri slabom osvetlení sa sieťnica stáva citlivejšou na kratšie vlnové dĺžky a menej citlivou na dlhšie. Napríklad modrá a červená kvetina, ktoré majú za denného svetla rovnaký jas, sa za šera budú výrazne líšiť. Modrá kvetina ostane rovnako jasná alebo bude vyzeráť ešte jasnejšie, no červená bude tmavá, až nerozlišiteľná. Táto adaptácia sa nazýva Purkyného efekt.

K posunu vo vnímaní farieb, konkrétne farebného odtieňa, dochádza aj vtedy, ak do monochromatického (jednofarebného) osvetlenia pridáme biele svetlo. Ide

o takzvaný Abneyho efekt. Ak nám napríklad svieti modré svetlo na biele pozadie, pridaním bieleho svetla do svetelného zdroja sa nám pozorovaná farba bude zdať fialovejšia, zelená žltšia a červená ružovejšia. Napriek tomu, že by sme očakávali len nárast svetlosti farby, dochádza aj k zdanlivému zníženiu vlnovej dĺžky (posun v spektre smerom doľava). Zaujímavé je, že k tomuto efektu dochádza len v ľudskom oku, prístrojmi sa zachytiť nedá [19].

Chromatická adaptácia alebo stálosť farby je vlastnosť našej percepcie, pri ktorej pozorovaná farba vychádza nielen z bezprostredného vnemu, ale aj z predošlej skúsenosti. Na základe tejto vlastnosti si za akýchkoľvek svetelných podmienok adaptujeme náš percepčný systém tak, že rozoznávame farbu daného objektu stále rovnako. Princíp tohto javu sa zakladá na fakte, že percepcia farby objektu nezávisí len od vlnovej dĺžky svetla, ktoré naň dopadá, ale hlavne od vlnovej dĺžky žiarenia odrážaného povrchom objektu putujúceho do oka pozorovateľa. Rôzne zdroje svetla produkujú žiarenie s rôznou spektrálnou kompozíciou. Správne proporcie všetkých vlnových dĺžok má len slnečné žiarenie. Klasické žiarovky s volfrámovým vláknom vytvárajú skôr žiarenie dlhovlnné, ladené do červena, naopak žiarivky produkujú skôr krátkovlnné, modré svetlo. Rôzne spektrálne kompozície konkrétnych zdrojov svetla potom spôsobujú, že materiál odráža svetlo inak, a teda logicky má inú farbu. Naš percepčný systém sa svojím spôsobom takýmto zmenám prispôsobuje. Adaptácia na svetlo spôsobuje, že pri prevažne dlhovlnnom osvetlení strácame citlivosť na dlhovlnné svetlo, no nie na krátkovlnné, pretože modré čapíky žiarenie s väčšou vlnovou dĺžkou nevnímajú. Toto prispôsobovanie je do určitej miery obmedzené. Pri bežných podmienkach má slnečné žiarenie na obed charakter krátkovlnného žltého svetla a večer dlhovlnného červeného. V tomto osvetlení bude červené jablko stále červené, pretože jeho pigment odráža červené svetlo. No keď ho osvetlíme modrým svetlom, bude sa javiť tmavošedé alebo čierne, pretože na jeho povrch nebude dopadať žiadne červené žiarenie.

Chromatická adaptácia môže podobne ako efekty farby pozadia spôsobiť indukovanie doplnkovej (opozičnej) farby. Napríklad adaptácia zraku pohľadom na červený štvorec môže spôsobiť, že následne pozorované objekty žltej alebo bielej farby budú vyzerat' zelenkasto. Tento jav môže tiež znížiť zdanlivú saturáciu, alebo posilniť či znížiť jasnosť. Keď sa napríklad zrak adaptuje na jasnú scénu, následne vzhliadnutá tmavšia scéna bude vyzerat' tmavšie než bežne.

2.5 Reprezentácia farieb: farebné modely a priestory

Ako sme uviedli vyššie, farbu väčšinou definujeme nepriamo ako psychologický vnem s tromi komponentmi. Či už sú tieto charakteristické údaje o farbe v zložení: odtieň, jas a saturácia, alebo iné číselné alebo inak exaktne vyjadriteľné parametre, vytvárame z nich štruktúry, a tak popisujeme rôzne farebné priestory. Farebný priestor si môžeme predstaviť ako štandardný Euklidovský vektorový priestor, ktorého osi reprezentujú vlastnosti alebo základné zložky farby. V tejto časti popíšeme najznámejšie farebné modely a priestory, ich význam a využitie.

Pojem farebného priestoru sa často stotožňuje s farebným modelom, čo je abstraktný matematický model reprezentujúci farby ako štruktúry s tromi, niekedy

štyrmi, číselnými parametrami vystihujúcimi komponenty farby. V skutočnosti ale pojem farebného priestoru zahŕňa nielen štruktúru, ale aj mapovaciu funkciu do absolútneho farebného priestoru, teda do reálneho sveta. V absolútnom farebnom priestore je každá farba jednoznačná, teda jasne definovaná bez odvolávania sa na externé faktory. Farebný priestor charakterizuje jeho *gamut*, čo je množina všetkých farieb (všetkých možných hodnôt vektorového priestoru), ktoré popisuje. Je dôležité uvedomiť si, že gamut sa odvíja od štruktúry a realizácie farebného priestoru.

Vo všeobecnosti rozlišujeme z hľadiska miešania farieb dva princípy. Aditívny farebný model, alebo RGB, funguje na princípe miešania troch základných farieb spektra – červenej, zelenej a modrej. Toto spájanie je založené na skladaní frekvencií svetelných zdrojov, práve preto sa tento systém volá aditívny. Postupným skladaním dostaneme najsvetlejšie – biele svetlo. Tento systém síce čiastočne korešponduje s našou perцепčnou sústavou, no z fyzikálneho hľadiska bežne nefunguje. Napriek tomu si tento model našiel široké uplatnenie pri elektronických zariadeniach ako monitory počítačov, televízne obrazovky alebo všeobecne pri osvetľovaní tmavých scén, napríklad v divadle.

Najznámejšie farebné priestory založené na tomto modeli sú sRGB a AdobeRGB. Gamut alebo hustota farebného priestoru opísaného RGB modelom závisí od jeho implementácie. Najbežnejšia je 24-bitová implementácia, pri ktorej na každú farebnú zložku pripadá 8 bitov, teda 256 rôznych hodnôt na 1 zložku (0...255). Každý farebný priestor založený na 24-bitovom RGB modeli má potom presne stanovený gamut - $256 \times 256 \times 256 \approx 16.7$ miliónov farieb. Niektoré implementácie používajú kvôli zväčšeniu hustoty diskretných farieb na vyjadrenie jednej zložky 16 bitov, čo sa potom využíva pri konvertovaní alebo porovnávaní s „hustejšími“, zložitejšími farebnými priestormi (viď nižšie).

Subtraktívny farebný model pracuje na opačnom princípe ako aditívny. Najlepším príkladom tohto modelu sú tlačene farby, ktoré vidíme na bielom papieri. Farba je v tomto modeli reprezentovaná ako svetlo tých frekvencií, ktoré nie sú povrchom absorbované. Čím viac farieb spojíme, tým tmavšia farba vznikne, pretože sa pridá ďalšia frekvencia do spektrálnej kompozície odrážaného svetla. Tomuto modelu hovoríme subtraktívny preto, lebo každá farba odráža jej príznačné svetlo a absorbuje žiarenie iných frekvencií. Základné komponenty tohto farebného modelu sú svetlé farby – tyrkysová (cyan), žltá (yellow) a fuchsiová (magenta). Ich zmiešaním po pároch vzniká červená, zelená a modrá. Kompletným zmiešaním týchto základných farieb však nemôžeme dostať čiernu, len niečo, čo sa na ňu podobá. Preto sa pri tlačené nastavujú na modeli CMY ale CMYK, kde sa pridáva samostatná čierna farba. Podobne ako z RGB, aj z tohto modelu existuje mnoho odvodených farebných priestorov pre rôzne sady atramentov a charakteristík tlače.

HSV/HSB, hue (odtieň), saturation (saturácia), value/brightness (jas) alebo lightness/luminance (svetlosť) pri HSL modeli, sú tri parametre vystihujúce farbu tak, ako si ju prirodzene predstavujeme, teda na základe odtieňa, saturácie a jasnosti svetlosti. Základný rozdiel medzi týmito dvoma modelmi je v poslednom parametri, ktorý sa môže na prvý pohľad javiť rovnaký. V skutočnosti pojem jasnosti poukazuje na jasnosť farby a svetlosť zas hovorí o vlastnosti materiálu. Oba tieto farebné modely sú odvodené od RGB, a teda aj ich realizácia prostredníctvom farebného priestoru je

závislá na konkrétnom farebnom priestore RGB. Môžeme ich popísať ako nelineárne deformácie farebnej RGB kocky, pričom HSV možno graficky znázorniť ako farebný kužeľ a HSL ako dvojitý kužeľ, dvojitý hexagonálny kužeľ alebo guľu. Prvá zložka u oboch modelov, farebný odtieň, sa zobrazuje do kruhu, pričom ďalší rozmer, saturácia, klesá od okraja smerom do stredu a jas stúpa smerom nahor. HSV/HSB sa používajú v počítačových programoch, práve pre ich názornosť a jednoduchosť.

Biologický model farebnej percepcie je takzvaný trojzložkový farebný model, ktorý si môžeme predstaviť ako trojrozmerný euklidovský priestor. V ňom sú osi x , y , z , priradené trom druhom farebných receptorov, dlhovlnným (L), stredno-vlnným (M) a krátkovlnným (S) čapíkom. Počiatkový bod $(S,M,L) = (0,0,0)$ prislúcha čiernej farbe. Biela nie je v rámci tohto priestoru jasne definovaná, pretože sa skôr vzťahuje na požiadavku vyváženia bielej (white balance) alebo na stav osvetlenia v priestore. Ľudský trojzložkový farebný priestor má tvar pokriveného kužeľa siahajúceho akoby do nekonečna. V skutočnosti možno extrémnymi svetelnými intenzitami náš perцепčný systém presaturovať, ale aj poškodiť. K styku s takýmto extrémnym osvetlením pri bežných podmienkach vôbec nedochádza.

Najsýtejšie, teda najviac satureované farby, sa v tomto priestore nachádzajú na vonkajšom okraji. Zaujímavý je fakt, že v skutočnosti neexistuje hnedé alebo šedé svetlo, len oranžové a žlté svetlo s intenzitou nižšou ako v okolitých častiach fyzického priestoru, v ktorom sa pozorovateľ nachádza. Trojzložkový farebný model má charakter aditívneho miešania farieb, pričom ide o skladanie vektorov na základe Grassmanovho zákona. Ten hovorí, že ak je daná farba kombináciou dvoch monochromatických farieb (s jednou vlnovou dĺžkou), potom jej vnímaná hodnota, prevedená do farebného priestoru, je sumou hodnôt dvoch farieb, z ktorých vznikla.

Jedným z prvých reálnych biologicky motivovaných farebných priestorov bol CIE 1931 XYZ, vytvorený Medzinárodnou komisiou pre ilumináciu (Commission Internationale de l'Eclairage) v roku 1931. Bol vytvorený na základe experimentálnych meraní, ktoré spočívali v nastavovaní pomerov, resp. množstva farebného žiarenia pri aditívnom miešaní troch základných odtieňov – červenej, modrej a zelenej s nastaviteľnou úrovňou jasu. Každý proband dostal najprv farebnú vzorku – ukážku, ktorú potom použitím mechanického zariadenia namiešaval – modeloval (matching) zo základných farieb. Na rozdiel od RGB systémov je rozsah číselných hodnôt parametrov X , Y , Z priestoru CIE 1931 pre každú dimenziu iný. Parameter X sa pohybuje v rozmedzí od 0 do 95.047, Y od 0 do 100.000 a Z od 0 do 108.883. Najväčším problémom farebného systému CIE 1931 je, že neposkytuje priamy spôsob na vyjadrenie perceptuálnych rozdielov medzi farbami.

Perceptuálny farebný priestor [11] je taký, v ktorom je vzdialenosť medzi dvoma diskretnými hodnotami (farbami) úmerná tomu, ako by ich rozdielnosť alebo podobnosť vnímal ľudský perцепčný aparát. Rozdielne farby sú od seba v priestore ďaleko, podobné zas blízko.

Pravdepodobne najznámejší perceptuálny farebný priestor je CIEL*a*b*, založený na nelineárnej transformácii biologicky motivovaného CIEXYZ a Teórii opozičných procesov. Podobne ako HSL a HSV používa prirodzenejší spôsob vyjadrovania farieb a to pomocou farebných odtieňov a svetlosti. Prvý komponent L^* reprezentuje jas farby (luminance), komponenty a^* a b^* vyjadrujú farebný odtieň dvojrozmerným

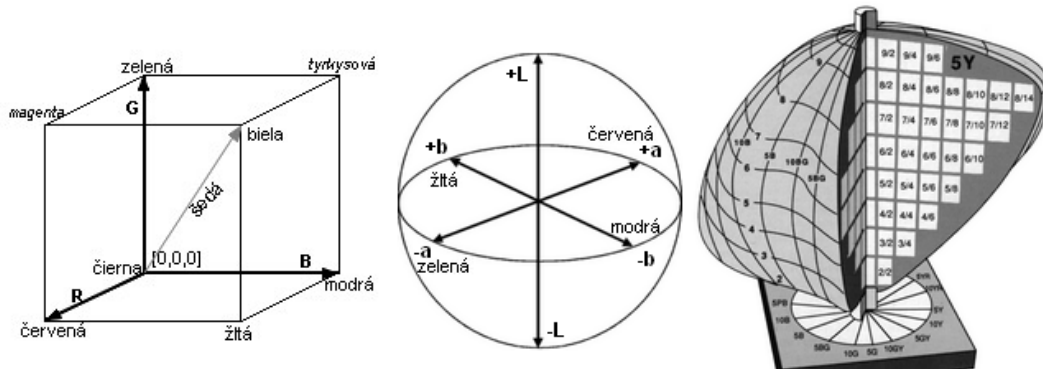
spôsobom na základe opozičných farieb. Rozsah komponentu L^* je od 0 (čierna) do 100 (biela). Komponent a^* hovorí o pozícii farby medzi zelenou a magentou (fuchsiovou), pričom negatívne hodnoty tohto parametra indikujú zelenú a pozitívne magentu. Podobne b^* sa pohybuje medzi modrou v negatívnych hodnotách a žltou v pozitívnych. Hodnota a^* a b^* sa bežne pohybuje v rozmedzí -128 až +128. Všimnime si, že systém je navrhnutý tak, že skutočne korešponduje s teóriou opozičných procesov, pretože v ňom nie je možné namiešať zelenkastú červenú alebo žlto-modrú. V bode, v ktorom sú koordináty a^* a b^* na nule, sa nachádza takzvaný šedý bod zastupujúci odtiene šedej (v rôznych stupňoch svetlosti).

Nelineárne vzťahy medzi tromi komponentmi tohto farebného systému sú vytvorené tak, aby napodobňovali logaritmické reakcie ľudského oka. Na rozdiel od CIE XYZ je tento systém závislý od stanovenia bieleho bodu (white point). Kým nie je tento údaj stanovený, nemôže tento priestor definovať absolútne farby. Bežne sa však biely bod určuje podľa štandardu vzhľadom na systém, resp. aplikáciu, ktorá tento farebný priestor používa. Napríklad pri bežnom grafickom software (napr. Adobe Photoshop) sa najviac používa norma CIE standard illuminant D50.

Farebný priestor CIE $L^*a^*b^*$ je možné znázorniť viacerými spôsobmi, obvykle sa používa kruhový tvar možných farebných odtieňov. Zobrazuje sa ako guľa s tromi na seba kolmými osami reprezentujúcimi jeho koordináty. Tiež sa zvykne zobrazovať čiastočne pre jednu alebo viac hodnôt parametra jas L^* ako kruh, niekoľko kruhov na osi L^* , výsek z gule a podobne.

Ďalším významným perceptuálnym priestorom je Munsellov farebný systém, ktorý budeme spomínať aj v ďalšej časti tejto kapitoly v súvislosti so svetovou farebnou štúdiou (WCS). Tento komerčný farebný priestor vznikol v prvej polovici 20. storočia a jeho autorom bol Albert H. Munsell. Základnými zložkami tohto modelu sú podobne ako pri HSL a HSV modeloch tri zložky prirodzene popisujúce farbu a to: odtieň, hodnota (svetlosť) a chroma, čo je parameter zhruba korešpondujúci so saturáciou (sýtosťou farby). Podobne ako farebné priestory CIE, je aj tento založený na experimentálnom meraní.

Munsellov systém definuje päť základných farebných odtieňov – červenú, žltú, zelenú, modrú a fialovú. V druhom stupni definuje ďalších päť odtieňov, teda 10 základných odtieňov, z ktorých je každý ešte rozdelený na 10 podúrovní, čo spolu dáva 100 číselných hodnôt pre rôzne farebné odtiene. Najčastejšie sa zobrazuje ako kruhovitá teleso, kde jeden kruhový výsek reprezentuje jeden odtieň (viď obrázok nižšie). Doplnkové farby teda ležia oproti sebe a aditívnym zmiešaním produkujú neutrálnu šedú s rovnakým jasom. Hodnota, alebo jas, sa mení vertikálne zdola nahor a má len 10 úrovní (0 je čierna a 10 biela). Chroma, zobrazovaná po kružnici s hodnotami meniacimi sa smerom od stredu reprezentuje čistotu farby, pričom v samom strede je saturácia nulová. Tento parameter nemá žiadne horné obmedzenie, maximálne dosahuje okolo 30 úrovní. Rôzne časti farebného priestoru majú rôzne maximálne hodnoty tohto parametra. Napríklad žlté farby majú viac rôznych hodnôt ako svetlá fialová, čo je podmienené povahou oka a fyzikálnymi vlastnosťami farebného stimulu.



Obr. 1. Schémy významných farebných priestorov. Tento obrázok zobrazuje schematický náčrt troch významných farebných priestorov, zľava RGB, CIEL*a*b* a Munsellovho farebného priestoru (obrázok prevzatý z [10]). Všimnime si, že ich tvary nie sú ani zďaleka rovnaké, čo súvisí s ich mierou výstižnosti ľudskej percepcie. Zvláštny tvar Munsellovho farebného priestoru budeme spomínať aj v časti 3.2 v súvislosti so zdôvodnením perceptuálnej dominancie niektorých farebných odtieňov.

2.6 Pomenovanie a kategorizácia farieb

Kategorizácia je proces, v ktorom narábame s rôznymi entitami, akoby boli rovnaké [32], zaradujeme ich do kategórií. Tento mechanizmus sa uplatňuje pri vnímaní (rozoznávaní), usudzovaní, plánovaní, reči a mnohých ďalších kognitívnych aktivitách. Môžeme povedať, že kategorizácia je všadeprítomným a fundamentálnym javom na pozadí ľudského rozmýšľania.

Už od čias antických filozofov sa o kategóriách zmysľalo ako o nejakých uzavretých nádobách, v ktorých sa nachádzajú prvky, ktoré majú spoločné určité vlastnosti s tým, že každý z prvkov musí spĺňať nutné a postačujúce podmienky pre začlenenie do kategórie, ktoré spätne definujú kategóriu ako takú. Podľa takéhoto prístupu prvky buď patria alebo nepatria do kategórie (Boolovská funkcia členstva v kategórii), ako je to u klasických množín. Tento pohľad na kategorizáciu bol prekonaný zhruba v polovici 20. storočia.

Na problém definatorických vlastností poukázal Wittgenstein [33] na základe konceptu hra. Všimnime si, že neexistuje jediná vlastnosť, ktorá by bola spoločná pre všetky hry, okrem tej „byť hrou“, či už ide o detské hry, šport a iné, neexistuje jednoznačná definícia. Wittgenstein v tomto prípade hovorí o takzvaných rodinných podobnostiach. Aj členovia jednej rodiny patria do jednej kategórie aj v prípade, že sa medzi nimi nenájde jedna spoločná vlastnosť. Ďalším krokom k modernej teórii kategorizácie bolo vytvorenie teórie fuzzy množín. Do množiny tohto špeciálneho druhu prvky nielen patria alebo nepatria (0 alebo 1), ale patria „do určitej miery“ vyjadrenej číslom z intervalu $\langle 0,1 \rangle$.

Najvplyvnejšou z moderných teórií kategorizácie bola prototypová teória Eleanor Roschovej [24]. Tá tvrdí, že prvky jednej kategórie nemusia mať jednu určujúcu vlastnosť, čiže môžu byť odlišné, ale majú byť vlastnosťami navzájom prepojené,

napríklad ak a, b, c, d, e sú vlastnosti, tak môžeme mať kategóriu $\{ab, bc, cd, de\}$. Pomocou špeciálnych psychologických experimentov Roschová potvrdila, že sa v kategóriách vyskytujú prototypové javy, a teda že existujú „najlepšie reprezentanty kategórií“, prvky, ktoré zdieľajú mnoho vlastností s ostatnými prvkami kategórie a málo s prvkami iných kategórií. Tiež objavila, že v hierarchických štruktúrach, napríklad taxonómiách, existujú takzvané kategórie základnej úrovne. Pomenúvajú ich krátke slová, ktoré sú používané častejšie než pomenovania z vyšších či nižších úrovní taxonómie. Tieto slová sú prvé, ktoré sa deti učia pri pomenovávaní okolitého sveta a prvé termíny, ktoré vznikajú pre daný koncept v jazyku. Napríklad vo vete „Pes sedel na verande“ nám ani nenapadne vymeniť slovo „pes“ za „cicavec“ (hierarchicky vyššie) alebo „labrador“ (hierarchicky nižšie, špecifickejšie slovo). Slovo „pes“ v tomto prípade tvorí kategóriu základnej úrovne. Táto úroveň nemusí byť pre všetky taxonómie rovnaká, záleží hlavne na funkčnosti a zľahčení rozlišovania, pomenovávania, komunikácie a kognície ako takej. Tento princíp funguje aj pri pomenovávaní farieb.

Kategorizáciu farieb možno rozčleniť na lexikálnu a perceptuálnu, ktoré spolu úzko súvisia. Lexikálna farebná kategorizácia spočíva v delení farebných vnemov na triedy korešpondujúce so symbolmi farebného sveta daného jazyka [32]. Perceptuálna zase triedi farby bez ohľadu na znalosť jazyka. Tento druh sme popísali v sekcii vyššie. V jazyku pomenovávame vnímané farebné kategórie pomocou farebných termínov. Priekopníci výskumu lexikálnej farebnej kategorizácie boli Berlin a Kay [2], ktorí skúmali najmä to, či sú termíny pre farebné kategórie univerzálne pre všetky jazyky na svete. Dospeli k rovnakým záverom ako Roschová, že aj v pomenovávaní farieb existujú kategórie základnej úrovne, označované základnými termínmi. Tie sú známe všetkým hovorcóm daného jazyka, ktorí pomocou nich vedú konzistentne pomenovať akúkoľvek farbu. Množstvo základných farebných termínov je v jazyku obmedzené a platí pre ne, že:

1. sú monolexémické, teda nie sú zložené z iných, menších význam nesúcich slov (napríklad modrá, nie zelenomodrá)
2. sú dostatočne všeobecné, teda
 - a) nepomenúvajú farbu, ktorá už je súčasťou inej väčšej farebnej kategórie (napríklad šarlátová nie je základná, lebo je súčasťou červenej)
 - b) ich používanie nie je obmedzené na úzky okruh objektov (napríklad termín blond sa používa len pre farbu vlasov)
3. sú psychologicky významné a všeobecne známe (napríklad žltá, nie šafránová, hnedá, nie béžová a podobne)

3 Základné farebné kategórie a World Color Survey

Teória o základných farebných kategóriách bola formulovaná a publikovaná v roku 1969 v knihe „*Basic Color Terms: Their Universality and Evolution*” [2] Brenta Berlina a Paula Kaya. Tí o povahe základných farebných kategórií, resp. termínov, ktoré ich pomenúvajú, tvrdili nasledovné:

1. Existuje množina základných farebných kategórií s obmedzeným počtom prvkov univerzálna pre všetky jazyky.
2. Farebné termíny vystihujúce kategórie z tejto množiny pribúdajú do jazykov v presnom poradí, ktoré je možné interpretovať ako evolúciu týchto termínov.

Táto uzavretá množina pozostáva z kategórií zodpovedajúcich anglickým, ale aj našim termínom: čierna, biela, červená, zelená, žltá, modrá, oranžová, hnedá, ružová, fialová a šedá. Evolúcia farebných termínov podľa Berlina a Kaya začína pri rozdelení pomenovaní farebných vnemov na čiernu a bielu, resp. tmavú a svetlú. Ako ukázal neskorší výskum Eleanor Roschovej, ide skôr o rozdelenie na čiernu a studené farby (napr. modrá, zelená) a bielu a teplé farby (červená, oranžová, žltá) [23]. Roschová ukázala, že v tomto prípade ide o takzvané kompozitné kategórie, ktoré sú charakteristické viacerými kvalitatívne odlišnými prototypmi (najlepšími reprezentantmi). Napríklad pre kategóriu teplé farby by sa najlepší reprezentanti zhodovali s našou bielou, žltou a červenou a prípadne aj oranžovou. Pokračujúc pri vysvetľovaní evolučného pridávania farebných termínov sa opäť stretávame s teóriou opozičných procesov. V ďalšom štádiu by mala podľa tejto hypotézy nasledovať červená, neskôr k nej opačná zelená, ďalej modrá alebo žltá a neskôr jej protipól. V predposlednom kroku, keď už teda jazyk obsahuje 6 základných opozičných farieb, prichádza na rad hnedá a v poslednom kroku už v ľubovoľnom poradí zvyšné štyri.

Empiricky boli tieto tvrdenia podložené výsledkami zo série farebných experimentov veľmi podobných tým zo svetovej farebnej štúdie opísaných nižšie. Napriek tomu, že na potvrdenie teórie experimenty v podstate stačili, boli v nich podstatné metodologické nedostatky [14]. Experimenty boli vykonané v 20 svetových jazykoch, počnúc angličtinou a inými európskymi jazykmi až po swahilčinu či mandarínčinu, avšak počet probandov pre jeden jazyk nebol väčší ako 3, čo sa vzhľadom na univerzálnosť prehlásení len ťažko berie za dostatočne širokú experimentálnu bázu. Ďalšou problematickou skutočnosťou bolo to, že experimenty sa konali len v rámci Berkeley, nie v prirodzených komunitách a to, že probandi boli väčšinou anglicky hovoriaci emigranti a niektorí z nich svoj materinský jazyk ani len dostatočne neovládali. Reakciou na tieto skutočnosti a nedostatky pôvodného experimentu bolo vytvorenie štúdie svetového rozsahu.

3.1 World Color Survey

Projekt World Color Survey alebo Svetová farebná štúdia bol odštartovaný v roku 1975 [4]. Jeho hlavným cieľom bolo vytvoriť veľkú bázu empirických dát a na jej základe potvrdiť, vyvrátiť či modifikovať pôvodné hypotézy Berlina a Kaya. Na

experimente sa zúčastnili probandi zo 110 jazykov bez písma a industrializovanej kultúry. Po ukončení experimentálnej fázy okolo roku 1980 nasledovalo spracovanie dát do elektronickej podoby, ich prekontrolovanie a prvotné analyzovanie. Finálna podoba databázy vo forme niekoľkých vzájomne logicky prepojených textových súborov bola sprístupnená na internete v roku 2003 [5].

Na experimente sa zúčastnilo v každom jazyku v priemere 24 probandov, mužov aj žien, prednostne monolingválnych. Pozostával z dvoch úloh. V prvej časti bolo probandom individuálne prezentovaných 330 farebných vzoriek vybraných z Munsellovho farebného priestoru, v 40 odtieňoch od R2.5 v prvom po RP10 v poslednom stĺpci so svetlosťou odstupňovanou v 8 riadkoch zhora nadol od najsvetlejšej po najtmavšiu farbu. Desaturované, bezfarebné odtiene boli reprezentované v nultom pravom stĺpci od bielej na vrchu, cez 8 stupňov šedej zodpovedajúcim svetlosti príslušného riadku až po čiernu úplne dolu. Nasledujúci obrázok zobrazuje schematické rozloženie Munsellových koordinátov všetkých farebných vzoriek. Pre lepšiu predstavu prikladáme aj výsledky z farebného experimentu v slovenčine z [20], kde sa okrem iného nachádza farebný obrázok palety farebných vzoriek, či vizualizácie výsledkov experimentov z vybraných jazykov z WCS.

Inštrukcie k prvej časti experimentu viedli probandov k tomu, aby používali základné farebné termíny, teda slová z „úzkej množiny jednoduchých slov, ktorými je možné pomenovať akúkoľvek farbu“, teda pomenovávať farebné vzorky tak, aby slovám, ktoré použili, rozumeli všetci hovorcovia ich jazyka (vlastnosti základných farebných termínov sme uviedli na konci predošlej časti). Farebné vzorky im boli prezentované v preddefinovanom náhodnom poradí rovnakom pre všetkých probandov. V druhej fáze im bola predstretá mriežka so všetkými 330 vzorkami (usporiadanými podľa odtieňa a svetlosti), na ktorej mali pre všetky použité termíny vybrať jedno alebo viac políčok, teda farebných vzoriek, ktoré boli podľa nich najlepšími reprezentantmi danej kategórie.

Ako sa ukázalo vo výstupných dátach, v niektorých jazykoch pravdepodobne došlo k nepochopeniu inštrukcií u probandov. Je to jasne ukázateľné napríklad v prípade, že sa medzi výsledkami pomenovávacej úlohy okrem slova „hnedá“ vyskytli zároveň aj termíny ako „kávová“, „kapučínová“ a „latté“, ktoré ale boli použité len menšinou probandov. V našom farebnom experimente pre slovenský jazyk [20] sme pri použití takmer rovnakých postupov odhalili ďalšie metodologické a praktické nedostatky. Jedným z nich bolo napríklad to, že probandi mali tendenciu vytvoriť si vlastnú teóriu o základných farebných kategóriách a tej sa počas experimentu držať, čo potom znižovalo mieru spontánnosti a autenticity odpovedí. Taktiež sa ukázalo, že sada o 330 vzorkách je náročná na koncentráciu, a že konzistentnosť odpovedí s pribúdajúcim počtom vzoriek klesá. Napriek všetkým metodologickým nedostatkom sa v experimente ukázalo, že aj v slovenčine je 11 základných farebných kategórií, ktoré korešpondovali s tými z pôvodnej štúdie Berlina a Kaya.

zistiť, či existuje medzi jazykmi s rôznym počtom termínov určitá hierarchia, ktorá by naznačovala, že sa základné farebné termíny v jazykoch časom vyvíjajú (napríklad evolučne starší jazyk má spoločnú kategóriu pre modrú a zelenú a evolučne novší má zelenú a modrú zvlášť).

K predbežnému spracovaniu WCS dát došlo ešte pred ich zverejnením v roku 2003. Prvým, intuitívnym krokom v analýze týchto dát bolo nájdenie víťazných pomenovaní, teda skutočných základných farebných termínov. Ako sme uviedli vyššie, dáta obsahovali nielen pomenovania univerzálne, ale aj veľmi špecifické, tie sa však vyskytovali zriedka. *Víťazné pomenovanie* pre jednu farebnú vzorku v jednom jazyku je to slovo, ktoré bolo na označenie tejto vzorky probandami použité najčastejšie, podobne ako pri algoritme „víťaz berie všetko“. *Farebný lexikón jazyka* je potom množina termínov, z ktorých každý figuroval ako víťazný aspoň pre jednu z 330 farebných vzoriek. Ako sme uviedli v [20], rozdiel medzi počtom víťazných a vymenovaných termínov sa pre niektoré jazyky veľmi výrazne líšil, čo mohlo okrem iného poukázať na zlé pochopenie úlohy u probandov a tiež v neposlednom rade zhoršiť výsledky akejkoľvek analýzy.

Prvým, predbežným spôsobom analýzy WCS dát bola ich vizualizácia do takzvaných máp termínov [4], ktoré boli vytvárané pomocou jednoduchých ASCII znakov do tvaru stimulačného materiálu, teda Munsellovej mriežky. Porovnaním s výsledkami z pôvodnej štúdie Berlina a Kaya bolo možné priamo sledovať paralely medzi preindustriálnymi WCS jazykmi a vyspelými jazykmi, napríklad angličtinou, pričom tieto sledovania boli priaznivo naklonené k pôvodným hypotézam. V [14] pristupovali autori k analýze výsledkov experimentov skôr teoretickým spôsobom. Výsledkom bol konceptuálny systém pre zakategorizovanie všetkých typov farebných lexikónov z WCS podľa toho, koľko základných farebných termínov obsahujú. Tento systém nadväzuje na pôvodnú teóriu o evolučných štádiách, pričom nové štádia reflektovali existujúce dáta z WCS. Autori tento systém popisujú na základe prípadových štúdií vybraných jazykov z WCS, bez celkového dôkazu o použiteľnosti systému pre všetky jazyky.

Na základe námietok voči takémuto intuitívnemu spracovaniu závislému na ľudskom faktore boli vykonané matematické analýzy všetkých WCS dát pomocou štatistických metód. Prvá a najznámejšia analýza bola [13], založená na skúmaní klasterizácie farebných kategórií naprieč jazykmi WCS a porovnaní s jazykmi z industrializovaných spoločností z pôvodnej štúdie. Na to, aby bolo možné merať klasterizáciu farebných termínov, bola zavedená jednotná reprezentácia pomocou geometrických centier farieb, ktoré boli pomenované daným termínom. Pri vytváraní centier sa najprv pre každý termín T z jazyka L pre každého hovorca, ktorý ho použil, vytvoril samostatný geometrický útvar v priestore CIEL*a*b* a geometrické stredy týchto útvarov boli spriemernené do jedného bodu, reprezentácie termínu T . Pre každý termín bolo následne nájdené jemu najbližšie políčko z Munsellovej mriežky. Nakoniec bol teda každý farebný termín z každého jazyka reprezentovaný pomocou jednej z 330 farebných vzoriek, ktoré boli použité v experimente. Porovnanie miery klasterizácie farebných termínov z WCS a fiktívnych náhodných dát ukázalo, že farebné termíny z WCS sa podobajú (zhlukujú) do väčšej než náhodnej miery. Autori podobným spôsobom ukázali aj to, že centrá termínov z WCS sa vyskytujú v blízkosti

centier pre angličtinu, čiže dokázali prvú časť univerzalistickej hypotézy, a teda že množina „použiteľných“ základných farebných kategórií je spoločná pre všetky jazyky.

Dôležitým aspektom debaty o univerzálnosti je to, či sa jedná o mechanizmus prototypový alebo iný, teda či sa farebné kategórie formujú okolo najlepších reprezentantov, ktoré vlastne predstavujú univerzálne základné farebné kategórie alebo ide o proces náhodný, pričom farebné kategórie sú určené len hranicami medzi nimi a prototypy sú len epifenoménom. Toto anti-univerzalistické tvrdenie pochádza z [22], kde autori vychádzajú z neúspešného pokusu o zopakovanie výsledkov Roschovej experimentov [23], v ktorých sa farby najlepších reprezentantov kategórií javili ako kognitívne privilegované (ľahšie zapamätateľné, identifikovateľné a pod.). Reakciou na tieto tvrdenia bola štúdia [21], v ktorej autori ukázali, že 6 základných opozičných farieb skutočne funguje ako univerzálna množina najlepších reprezentantov, že klasterizácia najlepších reprezentantov z WCS (získaných z druhej časti experimentu, vid' vyššie) je ešte vyššia než u centier termínov, a že na ich základe je možné predpovedať hranice farebných kategórií v jazyku. Toto zistenie priaznivo súvisí aj s modelom farebnej kategorizácie založenom na prototypovej teórii popísanom v poslednej časti tejto kapitoly.

Ďalšia podrobná analýza WCS dát [16] reaguje na zovšeobecňujúci charakter dvoch uvedených štúdií, konkrétne na manipuláciu s vopred vypočítanými centrami alebo najlepšími reprezentantmi, na základe ktorých mohlo dôjsť k skresleniu výsledkov. V tomto prípade sa autori zamerali na celú distribúciu kategoriálnych odpovedí na Munsellovej mriežke a to pre každého probanda zvlášť. Základnou jednotkou analýzy teda nebolo jedno políčko mriežky, ale binárny vektor reprezentujúci všetky políčka a reakcie probanda na ne vzhľadom na jeden termín, pričom z praktických dôvodov boli vynechané achromatické odtiene. Vektor jedného termínu u jedného probanda teda obsahoval pre každú z 320 zložiek 1, ak proband tento termín pre toto políčko použil, a 0 v opačnom prípade. Oblasť jednotiek vo vektore termínu teda zodpovedala mape farebných políčok pomenovaných u daného probanda týmto termínom. Tieto vektory boli spriemernené cez všetkých probandov a výsledná množina vektorov reprezentovala farebný lexikón daného jazyka. Podobne ako vo vyššie uvedených štúdiách bola na takto predspracovaných dátach vykonaná analýza klasterizácie termínov, ktorá ukázala, že všetky vymenované farebné termíny pre chromatické vzorky z celého WCS možno rozdeliť do 8 skupín identifikovateľných s kategóriami z angličtiny a to: červená, ružová, zelená, hnedá, žltá-alebo-oranžová a modrozelená. Čiže, bez ohľadu na to, koľko bolo v jazyku vymenovaných pomenovaní pre chromatické vzorky, bolo možné rozčleniť ich do 3 až 8 klastrov zhruba zodpovedajúcim univerzálnym základným kategóriám. Ak ich bolo 8, žltá, oranžová, modrá a zelená boli zvlášť, ak ich bolo menej, napríklad 3, spadali termíny skôr do kompozitných kategórií. Keď pridáme achromatické základné kategórie – bielu, čiernu a šedú, dostaneme množinu 11 základných farebných kategórií. Analýza vzájomnej zhody probandov na farebných termínoch v rámci jazyka ukázala, že k najväčšej zhode dochádza práve v miestach, v ktorých sa (zhruba) nachádza päť zo šiestich základných opozičných farieb (okrem modrej, ktorá sa väčšinou vyskytuje ako časť kompozičnej modrozelenej).

V najnovšom z uvedených článkov [17] ten istý tím autorov poukazuje aj na hypotézu o evolučnom vývoji tým, že nachádza takzvané motívy, systémy farebných kategórií, ktorých je obmedzené množstvo (3-6) a platia univerzálne pre všetky jazyky z WCS. Konkrétne sa v každom z jazykov objavuje jeden alebo aj viac týchto motívov, čo znamená, že univerzálne pravidlá môžu platiť silnejšie pre všetky jazyky ako v rámci jedného jazyka. Dôležitým aspektom ostáva pôvod týchto univerzálnych vzorov a mechanizmus, na základe ktorého k tomuto javu dochádza.

3.3 Univerzálnosť farebných kategórií

Jednou z odpovedí na otázku vzniku univerzálnych pravidiel vo vytváraní farebných kategórií je zdôvodnenie evolučné. Podľa [34] sú farby najlepších reprezentantov kategórií univerzálne kvôli tomu, že sa často vyskytujú v životnom prostredí, a preto je na ne náš perceptuálny systém určitým spôsobom naladený. Podobne môže ísť o evolučné naladenie na vlastnosti najbežnejšieho denného osvetlenia [26].

V [11] autori zas tvrdia, že vznik a univerzálnosť základných farieb (červenej, zelenej, modrej, žltej a fialovej) sa úzko viaže na tvar ideálneho perceptuálneho farebného priestoru (viď vyššie), konkrétne Munsellovho farebného priestoru. Ako vidno na obrázku 1 v časti 2.5, jeho tvar nie je úplne pravidelný. Práve v oblastiach, ktoré sú buď vypuklé alebo znížené, sa nachádzajú najlepší reprezentanti pre základné farby. Ďalej táto štúdia tvrdí, že pomenovávanie častí farebného priestoru sa musí uskutočňovať tak, aby maximalizovalo informačnú hodnotu farebných termínov. A teda, že prirodzené, prvotné rozdelenie priestoru a jeho pomenovanie je vždy tmavá/studená a svetlá/teplá, ktoré sa nachádzajú zhruba na opačných stranách farebného priestoru. Najvzdialenejšou od týchto „lokálí“ farebného priestoru je oblasť predstavujúca odtiene červenej, ktorá by mala podľa tejto teórie a súhlasne aj podľa Berlina a Kaya byť pridaná ako tretia základná farba. Predikcia toho, ktorá bude ďalšia základná farba, sa v tomto prípade sťažuje, pretože vzdialenosť medzi oblasťami klesá. Predpokladá sa, že by to mohla byť žltá alebo modrá³, ďalej zelená, fialová, ružová, oranžová, hnedá a šedá.

Okrem fyziologického aspektu ľudského percepčného aparátu treba prihliadať aj na sociálny rozmer, konkrétne na verbálnu komunikáciu medzi hovorcami jazyka. Ten bol skúmaný pomocou multiagentovej simulácie [1, 27]. Agenty sú v tomto výpočtovom modeli vybavené percepciou v CIEL*a*b* priestore, pričom vnemy kategorizujú pomocou interných reprezentácií farieb. V [1] je každá kategória reprezentovaná ako jeden bod vo farebnom priestore a príslušnosť do kategórie je funkciou vzdialenosti perceptu od prototypu kategórie.⁴ V [27] sú kategórie reprezentované zložitejším spôsobom, pomocou RBF (radial basis function) neurónov

³ Čo už nie je v súlade s pôvodnou hypotézou založenou na opozičných procesoch, pretože ako opozičná by mala prísť zelená.

⁴ Veľmi podobný prototypový prístup popíšeme v ďalšej sekcii, preto ho tu načrtneme len stručne.

pospájaných do takzvaných adaptívnych sietí.⁵ Agenty poznajú dva druhy akcií, hru na rozlišovanie a hru na hádanie, pričom prvá sa týka odlišovania rôznych farebných objektov (vlastne vzoriek) od seba a prebieha u každého agenta zvlášť a druhá naopak vyžaduje interakciu medzi dvoma agentmi. Hru na hádanie hrajú vždy dva náhodne vybrané agenty z populácie vo vopred stanovených roliach hovorca a poslucháča a jej cieľom je doplnenie chýbajúcich interných farebných kategórií u oboch agentov a zosúladenie farebných lexikónov, ktoré tieto farebné kategórie pomenúvajú a reprezentujú verejne.⁶ Tieto simulácie boli spustené s dvojakými sadami farebných vzoriek, prírodnými (z farebných fotografií) a náhodnými (počítačovými) farbami, pričom výsledky boli spracované rovnakým spôsobom ako údaje z WCS v [13] a porovnané s týmito výsledkami. Zaujímavý výsledok bol, že systém farebných kategórií, ktorý si agenty vytvorili, bol kvalitatívne podobný základným farebným kategóriám z WCS, a teda aj v umelom prostredí s použitím vhodnej reprezentácie farieb (perceptuálneho farebného priestoru) je možné nasimulovať vznik farebných kategórií pochádzajúcich z nášho univerzálneho repertoára.

4 Simulácia kategorizácie farieb

Náš model kategorizácie farieb je založený na Roschovej zisteniach o prototypových javoch. Napriek tomu, že Roschová svoju teóriu považuje za popisnú skôr než vysvetľujúcu, rozhodli sme sa použiť tieto princípy ako inšpiráciu pri vytváraní mechanizmu tvorby interných reprezentácií kategórií, v našom prípade farieb. V tomto modeli je každá kategória reprezentovaná práve jedným prototypom a členstvo prvkov v nej sa odvíja od ich vzdialenosti od prototypu, pričom prototyp aj skúmané príklady musia byť reprezentované v rovnakom priestore s použiteľnou metrikou.

V našej simulácii sme sa rozhodli vytvoriť idealizovaného umelého hovorca do každého z jazykov z WCS. Ten sa má učiť farby „ostenzívny“ spôsobom na základe všetkých odpovedí všetkých probandov z daného jazyka, akoby mu každý povedal, ako pomenúva ktorú farbu. Pre vnútornú implementáciu farebných kategórií tohto simulovaného hovorca sme sa rozhodli použiť sémantiku rozlišovacích kritérií.

4.1 Rozlišovacie kritéria

Sémantiku rozlišovacích kritérií pôvodne navrhol Šefránek [28] a implementoval Takáč [29,30]. Inšpirované Roschovej zisteniami o prototypových javoch, rozlišovacie kritéria fungujú ako *lokálne reagujúce detektory*. Každé rozlišovacie kritérium reprezentuje jednu kategóriu, teda je schopné rozhodnúť, do akej miery je jemu prezentovaný vstup členom kategórie, ktorú predstavuje. Lokálne reagujúci detektor si

⁵ Adaptívna sieť je dopredná neurónová sieť z RBF neurónov. Tie samotné sa však neučia, menia sa len ich váhy smerom do výstupného neurónu. Jednotlivé RBF uzly môžu pribúdať a odbúdať na základe zaznamenávanej úspešnosti v hrách.

⁶ Všimnite si, že rovnako ako u ľudí je možné, že je nejaká farebná kategória reprezentovaná u rôznych jedincov rôzne, ale dostatočne blízko (v tomto prípade môžeme skutočne hovoriť o blízkosti, keďže ide o body trojrozmerného priestoru) na to, aby si jedinci porozumeli, teda pomenovali rovnaké alebo podobné percepty rovnakým termínom.

môžeme predstaviť ako dvojicu, prototyp kategórie reprezentovaný bodom vo vstupnom priestore kritéria a aktivačnú funkciu kritéria, ktorá na výstupe vracia číslo z intervalu $\langle 0,1 \rangle$, reakciu na vzdialenosť vstupu od prototypu kritéria. Dôležitou vlastnosťou rozlišovacích kritérií je adaptívnosť – prototyp, rovnako ako aj parametre metriky, na ktorej je založená aktivačná funkcia, sa priebežne počítajú z príkladov tak, aby odrážali štatistické charakteristiky učiacej vzorky. Vstupný priestor nemusí byť nutne euklidovský, ako je tomu v prípade farieb reprezentovaných vo väčšine farebných priestorov.

Lokálne reagujúce detektory majú intuitívnu geometrickú interpretáciu založenú na konceptuálnych priestoroch [7]. Konceptuálny priestor je geometrický priestor s dimenziami zodpovedajúcimi atribútom reprezentovaných entít. V našom prípade, kde vstupmi pre kategorizáciu budú farby reprezentované ako body vo farebnom priestore, sa jedná o 3 rozmery jedného priestoru s jasne definovanou metriku, preto sa ďalej nebudeme venovať problému počítania vzdialeností v ľubovoľnom konceptuálnom priestore, viac v [30].

Pre nás dôležitou vlastnosťou prirodzených kategórií [7] a tiež rozlišovacích kritérií je predpoklad konvexnosti: ak dva body reprezentujú objekty, ktoré sú dobrým príkladom nejakej kategórie, aj všetky body ležiace medzi nimi by mali byť dobrým príkladom tejto kategórie. Meraním vzdialenosti vstupu od prototypu, ktorý sa nachádza v centre kategórie získame jeho „mieru príslušnosti“ do tejto kategórie: čím je vzdialenosť nižšia, tým je hodnota aktivačnej funkcie bližšie k 1, a vstup je lepším reprezentantom kategórie. Táto reprezentácia umožňuje, aby mali kategórie neostre (fuzzy) hranice, tak ako tomu je aj u kategórií v reálnom svete.

V implementácii rozlišovacích kritérií, ktorú sme použili, získava každé kritérium svoj prototyp na základe jemu prezentovaných vstupov (dostáva len pozitívne príklady), konkrétne ako ich geometrické centrum, v našom prípade priemer ich súradníc. Lokálne reagujúci detektor $r_{\vec{p}}$ (kde \vec{p} je prototyp kategórie) vráti aktiváciu exponenciálne klesajúcu s vzdialenosťou od prototypu

$$r_{\vec{p}}(x) = \exp(-k \cdot d(\vec{p}, \vec{x})) \quad (1)$$

kde k je kladná konštanta, $d(\vec{p}, \vec{x})$ je použitá metrika a \vec{x} je trojrozmerný vektor vo farebnom priestore CIEL*a*b*. Rozlišovacie kritéria, ktoré sme použili, zaznamenávajú distribúciu vstupov v kovariančnej matici Σ a pre výpočet aktivácie používajú Mahalanobisovu metriku

$$d_{\Sigma^{-1}}^2(\vec{p}, \vec{x}) = (\vec{x} - \vec{p})^T \Sigma^{-1} (\vec{x} - \vec{p}) \quad (2)$$

kde \vec{p} a \vec{x} sú stĺpcové vektory a Σ^{-1} je inverzná matica ku kovariančnej matici Σ množiny vstupov, ktoré kritérium dostalo počas tréningu. Pre bližší popis tohto mechanizmu viď. [30].

4.2 Implementácia a metódy vyhodnotenia simulácie

Simuláciu sme implementovali v jazyku Java s použitím knižníc pre kovariančné rozlišovacie kritériá a súvisiacich matematických knižníc z [29]. Prvým krokom nutným pre prácu z WCS dátami bolo spracovanie databázy vo forme textových

súborov stiahnutých zo stránky WCS [5], ktoré boli vzájomne prepojené pomocou skratiek a poradových čísel. Z nej sme vytvorili tréningové sady pre rozlišovacie kritériá v našej simulácii.

Pre každý jazyk sme vytvorili sadu rozlišovacích kritérií zodpovedajúcich všetkým farebným termínom, ktoré použili probandi z daného jazyka. Rozlišovacie kritériá predstavovali sadu interných reprezentácií umelého hovorca. Počas tréningovania dostávali kritériá ako vstup trojrozmerné vektory vo farebnom priestore CIEL*a*b*, ktoré zodpovedali farebným vzorkám, políčkam farebnej mriežky z pôvodného experimentu. Použili sme všetky odpovede všetkých probandov jedného jazyka⁷, ktoré sme postupne predkladali príslušným rozlišovacím kritériám. Ak teda proband p pomenoval políčko c termínom t , rozlišovacie kritérium zodpovedajúce tomuto termínu aktualizovalo svoj prototyp na základe farebných koordinát políčka c .

Po natrénovaní všetkých rozlišovacích kritérií pre všetky jazyky sme porovnali výsledky simulácie so štatisticky vyhodnotenými pôvodnými dátami, a to na základe víťazných kategórií⁸, vektorov aktivít a celkovo na základe klasterizácie kategórií naprieč všetkými jazykmi.

Ako sme spomenuli v predošlej časti, pre každé jedno z 330 políčok mriežky môžeme z dát pre celý jazyk vyjadriť jeho *víťaznú kategóriu* vo forme poradového čísla termínu, ktorý na jeho pomenovanie použilo najväčšie množstvo probandov, akoby sme použili algoritmus „víťaz berie všetko“. Takýmto spôsobom sme vypočítali pre každý jazyk víťazné kategórie pre všetky farebné vzorky. Okrem porovnania s víťaznými kategóriami z rozlišovacích kritérií sme pri tom prišli na závažnú skutočnosť, že počet vymenovaných kategórií a reálnych víťazných kategórií sa u väčšiny jazykov líši. V priemere je víťazných termínov cca o 13 menej ako vymenovaných, pričom maximum z tohto rozdielu bolo až 71, avšak modus bol 3, čiže môžeme povedať, že sa v rámci dát vyskytujú viac a menej „zašumené“ jazyky. Takéto rozdiely mohli vzniknúť napríklad v prípade, že probandi jednoducho nepochopili úlohu a používali vlastné subjektívne termíny (napríklad problém s hnedou a kávovou, ktorý sme spomínali v 3.2).

Jazyky sa tiež líšili v tom, do akej miery sa vedeli probandi zhodnúť na pomenovaní konkrétnych farebných vzoriek. Pre každý jazyk sme vyjadrili percento vzájomnej zhody probandov ako pomer počtu probandov, ktorí pre danú vzorku použili víťazný termín, k tým, čo použili nejaký iný a tieto pomery sme spriemernili cez všetkých 330 vzoriek. Najväčšia zhoda probandov bola zhruba 89%, najmenšia okolo 42%, priemer zo všetkých jazykov bol 67% a modus 60%.

Víťazné kategórie pre simuláciu sme počítali na základe hodnôt aktivácie rozlišovacích kritérií daného jazyka, pričom ako vstup slúžili farebné hodnoty skúmanej farebnej vzorky (políčka). Keďže sme pri vyhodnocovaní výsledkov

⁷ Pri priemernom počte 25 probandov na jeden jazyk a 330 políčkach to vychádza okolo 8250 vstupných hodnôt, ktoré boli podľa reakcií probandov rozdistribuované medzi všetky možné farebné kategórie v jazyku.

⁸ V nasledujúcom texte budeme pre jednoduchosť voľne zamieňať koncepty „farebný termín“ a „farebná kategória“. Vychádzame z predpokladu, že základné farebné termíny pomenávajú základné farebné kategórie, a keďže nás viac zaujíma univerzálnosť farebných kategórií než konkrétne slová, ktoré ich v konkrétnych jazykoch pomenávajú, nebudeme sa nimi zaoberať.

simulácie v [20] zistili, že existujú niektoré zavádzajúce prípady bezvýznamných úzko vyprofilovaných kategórií⁹, rozhodli sme sa, že budeme aktivitu rozlišovacích kritérií pri vyhodnocovaní víťazných kategórií násobiť počtom probandov, ktorí pre danú vzorku použili reprezentované pomenovanie. Pre vzorku c je v jazyku l víťaznou kategóriou tá, ktorej rozlišovacie kritérium $r_{\bar{p}}$ dáva po vynásobení aktivácie pre vstup c počtom probandov, ktorí toto pomenovanie použili N , najväčšiu hodnotu $r_{\bar{p}}(c) \cdot N$ spomedzi všetkých ostatných kritérií.

Reflektujúc námietku, že víťazné kategórie sú generalizáciou odpovedí všetkých probandov, sme sa podobne ako [16] rozhodli zaviesť porovnanie na základe distribúcie odpovedí probandov a aktivácií rozlišovacích kritérií, ktoré sme nazvali *vektory aktivít kategórií*. Podobne ako pri víťazných kategóriách sme v každom jazyku pre každú vzorku vytvorili vektor aktivít zo všetkých dostupných farebných kategórií – vymenovaných farebných termínov tak, že sme zráтали odpovede probandov podľa toho, aké termíny použili, pričom jeden komponent vektora bol rovný jednej farebnej kategórii (kategórie boli zoradené vždy v rovnakom poradí). Majme napríklad jazyk X s 10 probandmi a troma farebnými termínmi $t1$, $t2$ a $t3$. Ak pre dané políčko c odpovedali dvaja termínom $t1$, traja termínom $t2$ a piati termínom $t3$, potom aktivačný vektor \vec{a}_c bude vektor [2, 3, 5]. V prípade výsledkov simulácie sme opäť zobrali aktivity všetkých kritérií vynásobené zodpovedajúcim počtom probandov a vytvorili vektor.

V ďalšom kroku sme dva vektory aktivít, jeden z dát a jeden zo simulácie, znormovali použitím Euklidovskej normy¹⁰ a vypočítali sme ich skalárny súčin, teda kosínus uhla, ktorý tieto vektory zvierajú. Ak je skalárny súčin rovný 1, vektory sa rovnajú, ak -1 sú si navzájom opačné. Naším predpokladom bolo, že sa tieto hodnoty budú blížiť k jednej.

Posledný spôsob porovnania sme vybrali z [13]. Použili sme ten istý postup, ako sme popísali v časti 3.2. Pre každú kategóriu každého jazyka sme vypočítali jej centrum ako bod v CIEL*a*b* priestore. Mieru klasterizácie sme vyjadrili pomocou miery disperzie týchto kategórií nasledujúcim spôsobom. Pre každú kategóriu x v jazyku l sme našli kategóriu x^* v inom jazyku l^* , ktorá k nej bola spomedzi všetkých ostatných kategórií všetkých ostatných jazykov euklidovsky najbližšie. Mieru disperzie kategórií sme určili tak, že sme pre každý jazyk vypočítali sumu vzdialeností medzi jeho kategóriami a im najbližšími kategóriami z ostatných jazykov a tieto sumy navzájom sčítali.

⁹ K takejto situácii mohlo dôjsť v prípade, že sa kategória vyskytla iba u jedného alebo zopár probandov. Vzhľadom na to, že si kritériá formovali svoje prototypy priemerovaním prezentovaných vzoriek, toto konkrétne kritérium, trénované na malom množstve farebných vzoriek, dávalo podstatne väčšiu odozvu pre konkrétne vstupy, a teda mohlo „prebiť“ kritérium reprezentujúce kategóriu, ktorá bola použitá podstatným množstvom probandov, ale jej prototyp bol viac všeobecný.

¹⁰ Každý komponent vektora sme vydělili veľkosťou vektora, teda odmocninou zo sumy štvorcov jeho komponentov. Takto sme dostali dva jednotkové vektory v n -rozmernej hyperguli, kde n je počet kategórií vo vyhodnocovanom jazyku.

$$D = \sum_{l, l^* \in WCS} \sum_{x \in l, x^* \in l^*} \min vzd(x, x^*). \quad (3)$$

Vychádzajúc z overenej skutočnosti [13], že farebné kategórie v jazykoch z WCS majú disperziu menšiu než náhodné dáta, rozhodli sme sa porovnať disperziu vypočítanú z WCS dát s disperziou kategórií z našej simulácie. Tú sme vypočítali vyššie uvedeným spôsobom s tým, že centrá kategórií sme nepočítali z odpovedí probandov, ale použili sme prototypy z rozlišovacích kritérií, ktoré ich predstavovali.

4.3 Výsledky

Výsledok porovnania víťazných kategórií reprezentuje percento zhody medzi originálnymi dátami a rozlišovacími kritériami. Táto zhoda bola v priemere 90%, pričom maximálna zhoda bola 98% a minimálna 54,5%, modus bol 91,5% a medián 91,5%.

Čo sa týka porovnania vektorov aktivít kategórií z dát a rozlišovacích kritérií, ich skalárny súčin bol v priemere 0,97, najväčšia hodnota bola 0,99 a najmenšia 0,79, modus 0,97 a medián 0,97. Ideálny prípad úplnej zhody by nastal, ak by skalárny súčin týchto vektorov bol 1, čo znamená, že uhol medzi nimi by bol nulový. Z vysokých hodnôt priemeru, najčastejšej a strednej hodnoty môžeme usúdiť, že rozlišovacie kritériá boli väčšinou veľmi úspešné a len v niektorých prípadoch menej úspešné (u zhruba tretiny jazykov bola zhoda menšia než priemer).

Posledný aspekt, ktorý sme skúmali, bola disperzia farebných kategórií. Tá bola v prípade rozlišovacích kritérií výrazne (cca 3-krát) menšia než vypočítaná z originálnych dát, čo znamená, že centrá kategórií z rozlišovacích kritérií mali značne väčšiu mieru klasterizácie než reálne dáta. Tento výsledok poukazuje na to, že rozlišovacie kritéria sú schopné úspešne generalizovať trénovacie dáta. Ak by to tak nebolo, miera disperzie by sa blížila k tej vypočítanej z pôvodných dát. Podstatný rozdiel medzi hodnotami disperzie u simulácie a reálnych dátach vyplýva aj zo „zašumenosti“ výsledkov experimentov. Ako sme uviedli vyššie, počet vymenovaných a víťazných termínov sa v niektorých prípadoch výrazne líšil, podobne aj zhoda probandov na víťazných termínoch kolísala medzi polovicou a zhruba 90%. Vzhľadom na to, že disperzia sa počítala pre všetky vymenované termíny všetkých jazykov, mohli tieto bezvýznamné kategórie výsledok značne skresliť.

5 Záver

Popísali sme fungovanie perceptuálnej a lexikálnej kategorizácie farieb. Dôležitým fenoménom v tejto oblasti je existencia párov základných opozičných farieb. Sú to biela – čierna, červená – zelená a žltá – modrá. Tie nikdy nie je možné vnímať súčasne. Môžu za to opozičné princípy vyskytujúce sa na rôznych úrovniach spracovania vizuálneho vnemu, od receptorov až po neurálne dráhy v primárnej vizuálnej kôre. Týchto šesť základných farieb figuruje aj v jadre teórie o základných farebných termínoch (kategóriách).

Prostredníctvom štatistického spracovania dát zozbieraných v rámci svetovej farebnej štúdie (WCS), obsahujúcej 110 jazykov neindustrializovaných kultúr bez písma, sa dokázalo, že v lexikálnej farebnej kategorizácii existujú univerzálne princípy. Konkrétne, že farebný lexikón ľubovoľného jazyka obsahuje najviac 11 základných farebných termínov, ktorým zodpovedajú farebné kategórie. Tie pochádzajú z univerzálnej množiny farebných kategórií zodpovedajúcich našej bielej, čiernej, červenej, zelenej, žltej, modrej, hnedej, šedej, oranžovej, ružovej a fialovej. Dôraz sa kladie na šesť opozičných farieb, ktoré sú považované za evolučne staršie.

Univerzálne farebné kategórie sú oblasti, do ktorých sa zhlukujú (klasterizujú) najlepšie reprezentanty farebných termínov všetkých jazykov. Predpokladá sa, že univerzálnosť farebných kategórií pramení z farebnej kompozície životného prostredia a bežného denného osvetlenia, ideálneho funkčného rozdelenia perceptuálneho farebného priestoru a sociálnej interakcie.

V našej simulácii farebnej kategorizácie sme použili prototypový prístup k modelovaniu kategórií. Pre každý jazyk z WCS sme simulovali idealizovaného umelého hovorca, ktorý si osvojoval farebné termíny v danom jazyku. Pre internú reprezentáciu týchto termínov a im zodpovedajúcich farebných kategórií sme použili sémantiku rozlišovacích kritérií, ktoré reagujú na prezentované vstupy vyhodnocovaním vzdialenosti medzi vstupom a prototypom kategórie. Takto určujú do akej miery vstup patrí do tejto kategórie. Rozlišovacie kritérium, ktoré dáva pre daný vstup najlepšiu odozvu spomedzi ostatných kritérií, reprezentuje termín, ktorým by náš agent pomenoval prezentovaný vstup (farebnú vzorku). Výsledky simulácie ukázali, že prototypový model kategorizácie založený na sémantike rozlišovacích kritérií je vhodný pre problematiku pomenovania farieb a farebných kategórií.

Podakovanie: Táto práca bola podporená grantom VEGA 1/0361/08 a grantom KEGA 3/7300/09.

Literatúra

- [1] Belpaeme, T., Bleys, J.: *Colourful language and colour categories*, Second International Symposium on the Emergence and Evolution of Linguistic Communication, 2005.
- [2] Berlin, B., Kay, P.: *Basic Color Terms: Their Universality and Evolution*. University of California Press, Berkeley, CA, 1969.
- [3] Conway, B.R. Spatial structure of cone inputs to color cells in alert macaque primary visual cortex (V-1). *Journal of Neuroscience*, **21** (2001), pp. 2768-2783.
- [4] Cook, R. S., Kay, P., Regier T.: The World Color Survey database: History and use. In Cohen, H. Lefebvre, C.: *Handbook of Categorization in Cognitive Science*, Elsevier, St. Louis, 2005, pp. 223-242.
- [5] Cook, R. S., Kay, P., Regier T.: *WCS data archives*. 2003 (posledná zmena 14.11.2006). <http://www.icsi.berkeley.edu/wcs/data.html>
- [6] Fairchild, M.D.: *Color Appearance Models, 2nd Ed.* Wiley-IS&T, Chichester, UK. 2005.

-
- [7] Gärdenfors, P.: *Conceptual spaces*. MIT Press, Cambridge, MA, 2000.
- [8] Gouras, P.: Color Perception. In *WebVision, The organization of the Retina and Visual System*, John Moran Eye Center, University of Utah. 2008. <<http://webvision.med.utah.edu/KallColor.html>>
- [9] Green, M.: *SBFAQ Part 1: Basic Terms and Definitions*. Visual Experts. 2004. <<http://www.visualexpert.com/FAQ/Part1/cfaqPart1.html>>
- [10] ID-Studiolab. 2004. <http://studiolab.io.tudelft.nl/vormtheorie1/boulevard/onnoscolourshow/pics.html>
- [11] Jameson K.A., D'Andrade R.G.: It's not really Red, Green, Yellow, Blue: An inquiry into perceptual color space. In Hardin, C.L. and Maffi, L., *Color Categories in Thought and Language*, MIT Press, Cambridge, MA, 1997, p. 295–319.
- [12] Kay, P., Berlin, B., Maffi L., Merrifield, W.: Color naming across languages. In Hardin, C.L., Maffi, L., *Color Categories in Thought and Language*, Cambridge University Press, Cambridge, UK, 1997, pp 21–58.
- [13] Kay, P., Regier, T.: Language, thought and color: recent developments. *TRENDS in Cognitive Sciences*, **10** (2006), pp. 51-54.
- [14] Kay, P., Regier, T.: Resolving the question of color naming universals, *Proceedings of the National Academy of Sciences* **100** (2003), pp. 9085-9089
- [15] Lakoff, G.: *Women, Fire and Dangerous Things*, The University of Chicago Press, Chicago, 1987.
- [16] Lindsey D. T., Brown, A. M.: Universality of color names. *Proceedings of the National Academy of Sciences* **103** (2006), pp. 16608-16613.
- [17] Lindsey D. T., Brown, A. M.: World Color Survey color naming reveals universal motifs and their within-language diversity. *Proceedings of the National Academy of Sciences* **106** (2009), pp. 19785-19790.
- [18] Ludské Telo, GEMINI, spol. s r. o., Bratislava 1991. (The Human Body, Marshall Editions Ltd., London 1989)
- [19] Pridmore, R. Color variability of wine in a glass and general comments on contrast effects. *Color Research and Application*. **30** (2005), pp. 146-149.
- [20] Rebrová, K.: *Basic Color Categories*, diplomová práca, Univerzita Komenského v Bratislave, 2009.
- [21] Regier T., Kay P., Cook R. S.: Universal foci and varying boundaries in linguistic color categories. In Bara, B. G., Barsalou L., Bucciarelli, M., *Proceedings of the 27th Annual Conference of the Cognitive Science Society*, 2005.
- [22] Roberson, D., Davies I. & Davidoff, J.: Colour categories are not universal: Replications and new evidence from a Stone-age culture. *Journal of Experimental Psychology*, **129** (2000), pp. 369-398.
- [23] Rosch, E. Universals in color naming and memory. *Journal of Experimental Psychology*. **93** (1972), pp. 10-20.

-
- [24] Rosch, E.: Principles of Categorization, in Rosch E., Lloyd B., *Cognition and Categorization*, Lawrence Erlbaum Associates, Hillsdale NJ, 1978, pp. 27-48.
- [25] Sharpe, L.T.; Stockman A, Jägle H, Nathans J. Opsin genes, cone photopigments, color vision and color blindness. In Gegenfurtner K.R., Sharpe L.T. *Color Vision: From Genes to Perception*. Cambridge University Press, Cambridge, 1999. pp. 3-51.
- [26] Shepard, R.N.: The perceptual organization of colors. In Barkow, J. et al., *The Adapted Mind*, Oxford University Press, Oxford, 1992, pp. 495–532.
- [27] Steels, L., Belpaeme, T.: Coordinating Perceptually Grounded Categories through Language. A Case Study for Colour, *Behavioral and Brain Sciences* **28** (2005), pp. 469-489.
- [28] Šefránek, J.: Kognícia bez mentálnych procesov. In Beňušková, L. et al.: *Kognitívne vedy*, Kaligram, Bratislava, 2002, pp. 200-256.
- [29] Takáč, M.: Autonomous Construction of Ecologically and Socially Relevant Semantics. *Cognitive Systems Research* **9** (2008), pp. 293-311.
- [30] Takáč, M.: Kognitívna sémantika komplexných kategórií založená na rozlišovacích kritériách. In: Kvasnička et. al, *Mysel, inteligencia a život*. Vydavateľstvo STU, Bratislava, 2007, pp. 339 - 355.
- [31] Tanaka, K.: Inferotemporal Cortex and Object Vision. *Annual Review of Neuroscience*, **19** (1994), pp. 109-139.
- [32] Wilson R. A., Keil F. C. (Eds.): *MIT Encyclopedia of Cognitive Sciences*. MIT Press, Cambridge, MA, 1999.
- [33] Wittgenstein, L. *Philosophical Investigations*. Macmillan, New York, 1953.
- [34] Yendrikhovskij, S.N.: Computing color categories from statistics of natural images. *Journal of Imaging Science and Technology* **45** (2001), pp. 409–417.

Evoluční návrh hardware

Lukáš SEKANINA¹

Abstrakt. Kapitola shrnuje problematiku evolučního návrhu hardware, kdy je cílem pomocí evolučního algoritmu navrhnout nové výhodné implementace číslicových obvodů, analogových obvodů, antén a dalších zařízení. Na případových studiích zejména z oblasti návrhu číslicových obvodů jsou demonstrovány výhody a limity evolučního návrhu. Pozornost je rovněž věnována teoretické analýze evolučně navržených výpočetních systémů a důsledkům pro tzv. problém implementace. Dále je nastíněno, jakým způsobem by bylo vhodné interpretovat patentové právo, aby bylo užitečné i z pohledu automatického generování patentovatelných vynálezů.

1 Úvod

Použití různých optimalizačních algoritmů se stalo nedílnou součástí metod automatizovaného návrhu a implementace elektronických obvodů. Jednou z nejpobulárnějších metod je *evoluční optimalizace* [3, 13, 35]. V poslední dekádě můžeme pozorovat velký zájem o *genetické programování*, což je metoda spadající do třídy evolučních algoritmů (EA), která dovoluje elektronické obvody nejen optimalizovat, ale i automaticky sestavovat z předem zvolených komponent. Oproti optimalizačnímu přístupu, kdy jsou hledány nejvýhodnější hodnoty předem zvolených parametrů již hotového návrhu, umožňuje genetické programování vytvářet zcela nové struktury. V mnoha případech můžeme přímo hovořit o automatizovaném generování patentovatelných invencí. Počítač využívající evoluční algoritmy začíná významně konkurovat kvalifikovanému a kreativnímu inženýru-návrháři, který by dokonce mohl časem i ztratit zaměstnání [24]. Mezi nejzajímavější výsledky evolučního návrhu z oblasti elektroniky a výpočetní techniky řadíme zejména různé analogové filtry, regulátory, číslicové obrazové operátory, antény, optické systémy a komunikační protokoly (viz shrnující publikace [7, 12, 24]). U konkrétních evolučně navržených realizací bylo prokázáno, že fungují lépe (podle stanoveného kritéria) než nejlepší doposud známá řešení vytvořená konvenčními návrhovými technikami.

Kandidátní řešení produkovaná evolučním algoritmem jsou nejčastěji ohodnocována pomocí simulátoru, který je běžně používán v dané oblasti. Zajímavější

¹ Fakulta informačních technologií, Vysoké učení technické v Brně, Božetěchova 2, 612 66, E-mail: sekanina@fit.vutbr.cz

situace nastává, pokud jsou kandidátní řešení ohodnocována bez použití simulátoru, přímo v reálném prostředí. V oblasti *vyvíjejících se obvodů* (*evolvable hardware*) [7, 14, 22] je možné použít tzv. *rekonfigurovatelné obvody*. Kandidátní jedinec, se kterým pracuje EA, potom představuje buď přímo konfiguraci rekonfigurovatelného obvodu, nebo předpis, jak vytvořit kandidátní konfiguraci. Ve fitness funkci je následně testováno, do jaké míry plní kandidátní obvod požadavky zadané uživatelem. Např. se spočítá, pro kolik vstupních kombinací správně pracuje kandidátní kombinační číslicový obvod.

Použití rekonfigurovatelného obvodu namísto simulátoru má několik výhod: (1) Evoluční proces je obvykle výrazně rychlejší než při simulaci kandidátních obvodů v počítači. (2) Pokud je EA implementován přímo v cílové aplikaci, může zajišťovat dynamickou adaptaci na měnící se okolní podmínky, popř. znovuoobnovit činnost zařízení, pokud dojde k poškození části čipu. (3) EA může objevit nové implementace zadaného problému, které mohou být principiálně mimo prostor řešení, které je expert v daném oboru vůbec schopen vymyslet. Protože evoluční design probíhá v reálném fyzickém prostředí (v konkrétním čipu, za určité teploty, při existenci určitého šumu, elektromagnetického záření, při jistém kolísání napájecího napětí apod.), může dojít k nalezení řešení, které je perfektně adaptováno pro daný čip, prostor a čas. Adrian Thompson jako první ukázal, že je EA schopen najít velmi neobvyklá (a nepochopitelná) řešení, která jsou mimo dosah konvenčních návrhových technik a která *nefungují*, pokud jsou přemístěna z prostředí, kde vznikla (např. do jiného rekonfigurovatelného obvodu) [28].

Protože můžeme evolučně konstruovat elektronické obvody, můžeme rovněž konstruovat výpočetní systémy, přesněji řečeno, fyzická výpočetní zařízení. V této kapitole rovněž ukážeme, že výpočetní systémy navržené a fyzicky realizované evolučními technikami vykazují vlastnosti, které nenajdeme v existujících výpočetních systémech, které jsou běžně navrhovány inženýry. Evoluční navrhované systémy provádějí požadované výpočty, ale často není možné rozpoznat, jak a na jakém principu pracuje fyzická implementace. Důvodem je fakt, že evoluce využívá pro konstrukci těchto zařízení fyzikální a chemické vlastnosti materiálů, ze kterých jsou zařízení realizována, a různé okamžité charakteristiky prostředí, jako jsou např. teplota, elektromagnetické pole atd. Podobně je tomu např. u mozku, který je adaptován na určité prostředí a pouze v tomto prostředí funguje. Chování mozku lze také interpretovat jako výpočetní proces, ale doposud přesně nevíme, jak je tento „výpočet“ prováděn. Problémem tedy je, že neexistuje abstraktní model výpočtu a zobrazení mezi abstraktním výpočetním procesem a jeho fyzickou implementací. Můžeme se ptát, zda lze vůbec považovat evolucí vytvořená výpočetní zařízení za *výpočetní mechanismy* (computing mechanisms [19]) ve smyslu Turingova stroje.

Cílem této kapitoly je podat základní přehled o evolučním návrhu obvodů, jak s využitím simulátoru obvodů, tak i pomocí rekonfigurovatelných obvodů. Bude nás zajímat, do jaké míry může evoluční návrh konkurovat konvenčnímu návrhu elektronických obvodů (zejména v oblasti číslicových obvodů). Další část kapitoly je věnována teoretické analýze problému implementace výpočetního zařízení z pohledu evolučního návrhu. Budeme charakterizovat třídu evolučně navržených výpočetních zařízení z pohledu existujících výpočetních zařízení navržených konvenční cestou

a z pohledu živých systémů, které existují v přírodě a jimž často rovněž přisuzujeme výpočetní schopnosti. V poslední části kapitoly se zaměříme na problematiku patentování evolučně vytvořených řešení.

2 Evoluční návrh

Evoluční návrh vychází z metod evoluční optimalizace. V této kapitole vysvětlíme princip metody a popíšeme použití kartézského genetického programování pro návrh číslicových obvodů.

2.1 Problém návrhu jako problém prohledávání

Evoluční algoritmy jsou stochastické prohledávací algoritmy inspirované Darwinovou teorií evoluce a neodarwinismem [2, 13, 35]. Kandidátní řešení je reprezentováno řetězcem symbolů, který v EA nazýváme chromozom nebo genotyp. V případě evoluční optimalizace se jedná o sadu optimalizovaných parametrů, které jsou vhodně zakódovány. V případě evolučního návrhu je řešení kódováno buď *přímo*, např. jako seznam použitých komponent, s informací o jejich propojení a parametrech, nebo *nepřímo*, např. jako program, jehož vykonáním dojde k sestavení kandidátního obvodu. Množina všech chromozomů tvoří prohledávaný prostor, ve kterém evoluční algoritmus pracuje. Kvalitu kandidátního jedince určuje fitness funkce, která ohodnotí kandidátní fenotyp vytvořený z příslušného genotypu. Ve fitness funkci je reflektována specifikace problému, který je pomocí EA řešen. Nová kandidátní řešení jsou vytvářena pomocí geneticky inspirovaných operací (mutace, křížení apod.), které pracují nad chromozomy. Selektční tlak vede prohledávací algoritmus k výhodnějším částem prostoru možných řešení.

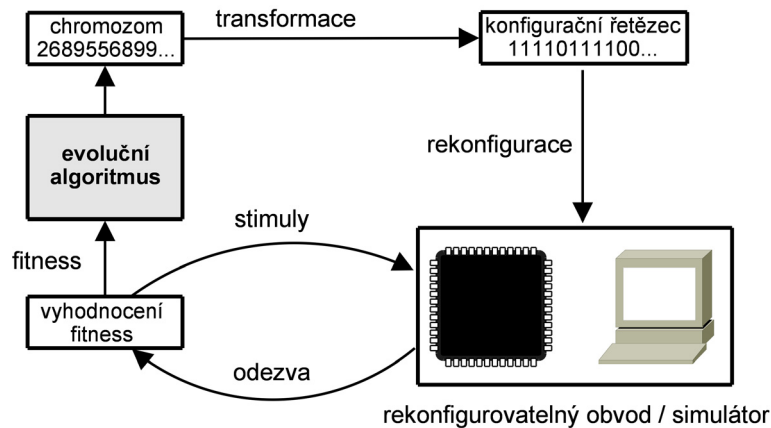
EA pracuje následovně: nejdříve je vygenerována počáteční populace (buď náhodně nebo pomocí heuristiky), jejíž prvky jsou ohodnoceny fitness funkcí. Dále, dokud není splněna ukončující podmínka, jsou prováděny následující kroky:

- Ze staré populace je vytvořena pomocí genetických operátorů nová populace.
- Každý jedinec nové populace je ohodnocen fitness funkcí.

Je tedy zřejmé, že oproti konvenčnímu návrhu je evoluční návrh založen na procesu „vygeneruj řešení a otestuj ho“. Důležité je, že změna probíhající v kandidátním řešení je náhodná. Naopak, konvenční přístup modifikuje existující řešení s určitou představou o výsledku, obvykle podle dobře definovaného a zavedeného postupu a tedy s určitým cílem.

Obrázek 1 ukazuje princip evolučního návrhu elektronického obvodu s využitím rekonfigurovatelného zařízení. Pokud EA pracuje přímo na úrovni configuračního řetězce rekonfigurovatelného obvodu, potom odpadá fáze „transformace“, tj. vytvoření konfigurace obvodu podle chromozomu. Evaluace kandidátního řešení představuje proces, který většinou nejvíce ovlivňuje dobu evoluce. Je tedy snaha ji redukovat. Např. evaluace kandidátního analogového obvodu provedená v rekonfigurovatelném

obvodu může být o několik řádů rychlejší než simulátor [25]. Zde se obvykle používají simulátory z rodiny Spice, které simulují přesně, ale pomalu.



Ob. 1. Evoluční návrh s využitím rekonfigurovatelného obvodu. Namísto rekonfigurovatelného obvodu je možné použít simulátor.

Pokud navrhujeme číslicové obvody na úrovni hradel, můžeme v některých případech simulaci obvodu výrazně urychlit „předpočítáním“ určitých hodnot, popř. paralelní simulací. Zde však platí, že s každým přidaným vstupem se doba evaluace zdvojnásobí, pokud ohodnocujeme všechny možné kombinace na vstupech. Proto je tento přístup použitelný pouze pro případy, kdy je počet vstupů malý. Pro složitější obvody musíme definovat *trénovací množinu* vstupních vektorů, která se použije pro evaluaci, a na konci evoluce ověřit výsledné řešení pomocí *testovací množiny*. Typickou aplikací je zpracování signálů a obrazů [22, 24]. Vzhledem k tomu, že není garantováno, že navržený obvod bude pracovat správně pro všechny možné kombinace na vstupech, je třída aplikací této techniky omezena. Ve speciálních případech můžeme použít i jiné principy evaluace kandidátních řešení (kapitola 3.4).

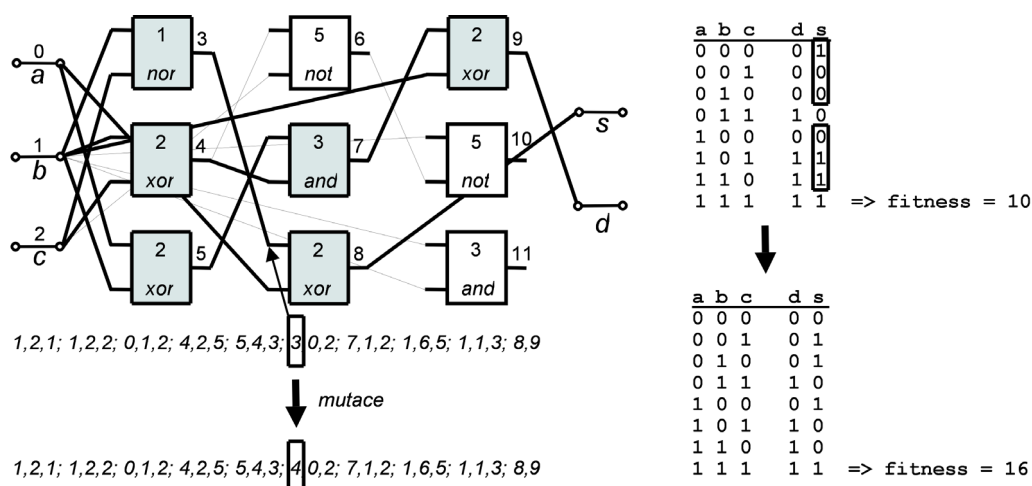
Dalšími faktory, které ovlivňují dobu evaluace kandidátního obvodu, jsou doba rekonfigurace obvodu a doba nutná pro výpočet fitness hodnoty z naměřených dat. Doba nutná pro vytvoření nové populace pomocí genetických operátorů je při dobrém návrhu aplikace většinou zanedbatelná.

2.2 Kartézské genetické programování

Předpokládejme, že je cílem navrhnout schéma zapojení obvodu pro zadanou specifikaci. Následnou realizaci na čipu zatím ponechme stranou. Kartézské genetické programování (Cartesian Genetic Programming – CGP) je vhodné pro evoluční návrh grafových struktur, a tedy i schémat číslicových obvodů [16]. Jedná se o variantu genetického programování, u které jsou kandidátní řešení reprezentována pomocí orientovaných grafů. Kandidátní obvod je modelován jako pole programovatelných elementů (uzlů grafu) o velikosti $n_c \times n_r$ (počet sloupců \times počet řádků). Označme počet

primárních vstupů obvodu n_i a počet primárních výstupů obvodu n_o . Každý z uzlů, jenž může mít až n_i vstupů, realizuje právě jednu funkci, která je vybrána z množiny dostupných funkcí F . Vstupy uzlu, který se nachází v j -tém sloupci, mohou být připojeny buď na primární vstupy obvodu nebo na výstupy uzlů umístěných v 1 až L předchozích sloupcích. Uzly stejného sloupce se nesmějí propojovat. Primární výstupy mohou být připojeny k libovolnému uzlu obvodu.

Obrázek 2 ukazuje příklad instance CGP. Každý kandidátní obvod je zakódován pomocí řetězce celočíselných hodnot (genotypu). Způsob kódování je následující: Každému primárnímu vstupu obvodu je přiřazen index z intervalu $0, \dots, n_i - 1$. K výstupům uzlů jsou postupně rovněž přiřazeny indexy, a to po sloupcích, s počáteční hodnotou n_i pro nejlevější horní uzel. Každý uzel obvodu je kódován pomocí $n_i + 1$ celočíselných hodnot. Prvních n_i hodnot určuje indexy uzlů, ke kterým budou připojeny vstupy uvažovaného uzlu. Poslední hodnota určuje kód logické funkce uzlu. Na konci chromozomu najdeme n_o hodnot definujících indexy uzlů, ke kterým jsou připojeny primární výstupy obvodu. Základní vlastností uvedeného kódování je, že zatímco délka chromozomu je vždy konstantní, velikost zakódovaného obvodu (fenotypu) je variabilní. Množina všech chromozomů určuje prohledávací prostor, ve kterém pracuje evoluční algoritmus.



Obr. 2. Příklad instance CGP s parametry $n_c = 3$, $n_r = 3$, $n_i = 3$, $n_o = 2$, $n_t = 2$, odpovídajícího genotypu a pravdivostní tabulky. Pokud je cílem navrhnout úplnou sčítačku (s operandy a a b , c je vstupní přenos, na výstupu s označuje součet a d je výstupní přenos), je získána fitness hodnota 10. Po provedení jediné mutace (3→4 na pozici 16) je možné získat plně funkční sčítačku s fitness hodnotou 16.

Základní varianta prohledávacího algoritmu, která je využita v CGP, používá populaci o velikosti $1 + \lambda$ jedinců (obvykle $\lambda = 4$) a jediný genetický operátor – mutaci, který pracuje tak, že náhodně vybere h genů (tj. h hodnot z chromozomu) a náhodně vygeneruje jejich nové hodnoty (obrázek 2 ukazuje příklad pro $h = 1$). Pomocí mutace vznikají nová kandidátní řešení problému. Počáteční populace je vygenerována

náhodně nebo ji tvoří předem připravená řešení problému. Kvalita každého kandidátního řešení je ohodnocena pomocí fitness funkce. V případě návrhu malých kombinačních obvodů je například vypočteno, kolik je schopen kandidátní obvod vyprodukovat správných bitů pro všechny možné kombinace vstupních signálů. Pro obvod na obrázku 2, který má tři vstupy a dva výstupy, je fitness hodnota 10. Po provedení jediné mutace je však možné získat plně funkční úplnou sčítačku s fitness hodnotou 16. Nová populace je vytvářena tak, že je ze staré populace vybrán jedinec s nejlepší hodnotou fitness, který je vložen do nové populace spolu se svými λ kopiemi pozměněnými pomocí mutace. V CGP se používá jedno důležité pravidlo: Pokud existuje více "nejlepších" jedinců se stejnou fitness hodnotou, použije se jako rodič ten, který nebyl rodičem v předchozí generaci. Tímto je podporována genetická diverzita populace. Evoluce končí nalezením dostatečně kvalitního jedince nebo vyčerpáním povoleného počtu generací.

Kartézské genetické programování bylo použito nejen pro návrh kombinačních obvodů, ale i v úlohách symbolické regrese, filtrace a klasifikace, pro realizaci kontrolérů, developmentálních zobrazení, sebemodifikujícího se kódu atd. [24].

3 Příklady evolučního návrhu

Abychom demonstrovali různé přístupy k reprezentaci problému a výpočtu fitness hodnoty, ukážeme, jak je možné pomocí evolučního algoritmu navrhovat malé kombinační obvody (budou obsahovat desítky hradel), obrazové filtry (tisíce hradel) a testovací obvody (miliony hradel).

3.1 Soutěž Humies

Evolučně navržené obrazové filtry i testovací obvody byly prezentovány v rámci soutěže Humies (Human competitive awards in genetic and evolutionary computation [8]), která je pořádána každoročně na konferenci GECCO – nejvýznamnější konferenci o evolučních a genetických algoritmech. V rámci soutěže je třeba představit výsledek, který byl získán pomocí technik evolučního návrhu – tedy automaticky vygenerován počítačem simulujícím evoluční proces. Komise složená z předních odborníků posuzuje, zda je takto vytvořený výsledek schopen soutěžit s lidskou invencí. Evolučně navržené obrazové filtry získaly v roce 2004 čestné uznání a testovací obvody byly v roce 2008 oceněny stříbrnou medailí. Na stránkách Humies [8] najdeme desítky výsledků, které byly vyprodukovány evolučními algoritmy a které jsou konkurenceschopné výstupům inženýra-návrháře. Kdy je tedy výsledek shledán jako „human-competitive“? Musí splňovat alespoň jedno z uvedených kritérií, které zavedl John Koza [12]:

- Výsledek byl patentován jako vynález v minulosti, je vylepšením patentovaného vynálezu nebo by dnes mohl být kvalifikován jako patentovatelný vynález.

- Výsledek je stejně dobrý nebo lepší než jiné řešení, které bylo akceptováno jako nový vědecký výsledek v době, kdy bylo publikováno v recenzovaném vědeckém časopisu.
- Výsledek je stejně dobrý nebo lepší než jiný výsledek, který byl vložen do databáze nebo archivu výsledků udržovaných mezinárodně uznávanou skupinou vědců.
- Výsledek je publikovatelný ve své podstatě jako nový vědecký výsledek nezávisle na tom, že byl mechanicky vytvořen.
- Výsledek je stejně dobrý nebo lepší než nejaktuálnější člověkem vytvořené řešení dlouho známého problému, pro který vznikala řada neustále lepších výsledků navržených člověkem.
- Výsledek je stejně dobrý nebo lepší než to řešení, které bylo považováno za pokrokové ve svém oboru, když bylo objeveno.
- Výsledek řeší problém, který má pro daný obor nepopiratelnou obtížnost.
- Výsledek zvítězil v soutěži s řešeními vytvořenými člověkem (buď ve hře proti člověku, nebo proti programu, který byl napsán člověkem).

3.2 Kombinační obvody

Předpokládejme, že máme obvodově realizovat logickou funkci o n vstupech, která je definována pomocí pravdivostní tabulky. Z pravdivostní tabulky je možné odvodit kanonický tvar v uvažovaném výpočetním modelu (např. výraz v normální disjunktí formě, úplný binární rozhodovací diagram apod.). Cílem logické syntézy je obvykle najít takové vyjádření logické funkce v uvažovaném modelu, které minimalizuje cenu realizace, tj. počet hradel, zpoždění apod. V rámci zvoleného modelu existuje množina transformací, které je možné na logický výraz aplikovat a tím měnit jeho tvar. Nikdy však nemůže dojít ke změně evaluace logické funkce, kterou výrazy reprezentují. Zvolený model a s ním spojené transformace určují prostor možných řešení. Konvenční algoritmy (např. Espresso, ABC apod.), které se v rámci syntézy používají, jsou navrženy tak, že dokážou nalézt co nejvýhodnější řešení, ale pouze v rámci zvoleného modelu, tj. v předem vymezeném podprostoru možných řešení.

Evoluční návrh vychází z toho, jaké zdroje jsou dostupné pro implementaci na konkrétní platformě. Jedná se zejména o typ výpočetních elementů, jejich možné funkce, počty vstupů a výstupů a možnosti propojování. Podle konkrétní cílové platformy je potom vytvořena taková reprezentace obvodu, která by měla umožnit postihnout libovolný obvod z množiny všech vytvořitelných obvodů. Tím je definován prohledávací prostor pro evoluční algoritmus. V rámci tohoto prostoru mohou existovat velmi výhodné implementace požadované logické funkce, které nejsou dosažitelné v konvenčních modelech.

Jednou z prvních aplikací CGP byl evoluční návrh kombinačních sčítaček a násobiček, které obsahují co nejmenší počet hradel. Pro tuto úlohu je fitness f definována

$$f = B + (n_r n_c - g), \quad (1)$$

kde B je počet správně určených výstupů (měřeno pro všechny vstupní kombinace, tj. maximální hodnota $B_m = n_o 2^{n_i}$ pro $n = n_i$) a g je počet použitých hradel. Zde je nutné poznamenat, že se výraz počet hradel ($n_r n_c - g$) začíná započítávat až v okamžiku, kdy B dosáhne maximální hodnoty B_m .

Tabulka 1 shrnuje nastavení experimentů a získané výsledky podle [16, 30]. Je zřejmé, že CGP dokáže najít násobičky (pro dvou až čtyřbitové operandy), které obsahují v průměru méně dvou vstupových hradel než násobičky získané konvenčními algoritmy.

Tabulka 1. Počet hradel v násobičkách navržených konvenčně a pomocí CGP (nejlepší ze 100 nezávislých běhů). CGP pracuje s velikostí populace 5 jedinců, $h = 3$, $L = n_c$, množina hradel obsahuje funkce x AND y , x XOR y a (not x) AND y .

| Násobička | Nejlepší konvenční | Nejlepší CGP | $n_r \times n_c$ | Max. generací |
|-----------|--------------------|--------------|------------------|---------------|
| 2b x 2b | 8 | 7 | 1 x 7 | 10 tis. |
| 3b x 2b | 17 | 13 | 1 x 17 | 200 tis. |
| 3b x 3b | 30 | 23 | 1 x 35 | 20 mil. |
| 4b x 3b | 47 | 37 | 1 x 56 | 200 mil. |
| 4b x 4b | 64 | 57 | 1 x 67 | 700 mil. |

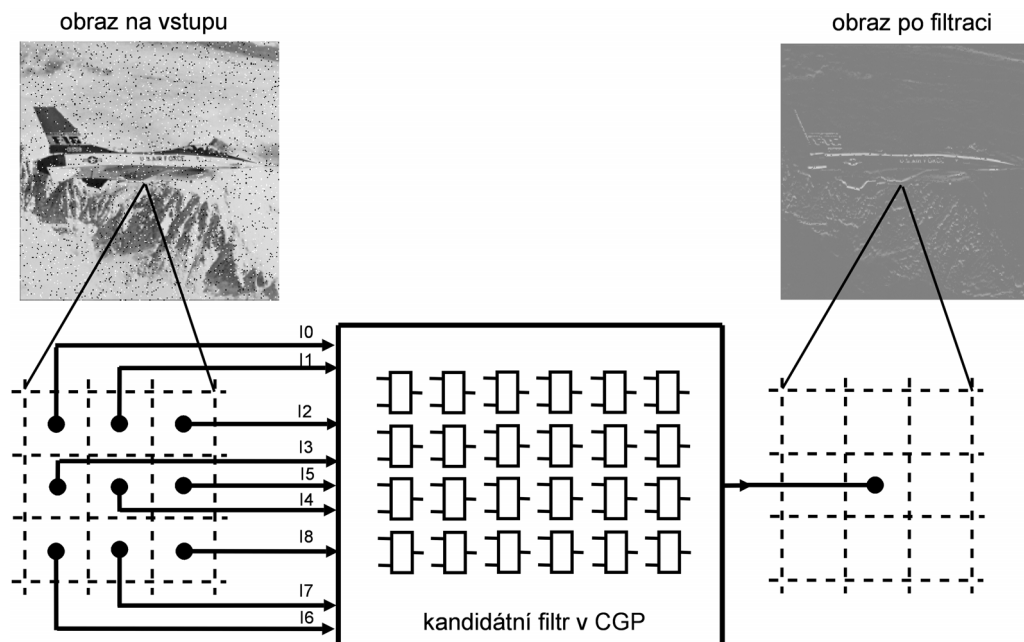
Za hlavní nevýhodu evolučního návrhu je považován fakt, že výše popsaná metoda funguje jen pro relativně jednoduché obvody. Pokud totiž budeme zvyšovat počet vstupů obvodu a současně budeme požadovat ověření činnosti obvodu pro každý možný vstupní vektor, poroste doba evaluace kandidátního řešení exponenciálně. Dalším problémem je prodlužování chromozomu, které je způsobeno požadavkem na vytváření velkých obvodů. Dlouhé chromozomy implikují velké prohledávací prostory, ve kterých evoluční algoritmy nedokážou efektivně nacházet zajímavá řešení. Pro obvody s cca 10-15 vstupy již evoluce nenajde rozumné řešení vůbec. Proto byly zavedeny různé techniky, které umožňují se výše uvedeným problémům vyhnout, např. inkrementální evoluce, modulární přístup apod. [24].

Z praktického pohledu je užitečné nezačít evoluční proces z náhodně vygenerované populace, ale z již existujícího řešení získaného konvenční metodou (např. ABC, SIS apod.). Pomocí CGP je potom možné v konvenčním řešení redukovat počet hradel i o desítky procent.

3.3 Evoluční návrh obrazových operátorů

K potlačení šumu v obraze nebo k detekci hran se nejčastěji používají nelineární číslicové filtry, které pracují tak, že novou hodnotu filtrovaného pixelu počítají pomocí vhodné nelineární funkce na základě předešlé (potenciálně poškozené) hodnoty pixelu a jeho nejbližších sousedů. Obrázek 3 ukazuje příklad filtru, jenž používá 9 pixelů na vstupu, které spolu tvoří tzv. filtrační masku o velikosti 3x3 pixely. Pokud budou pixely reprezentovány na 8 bitech, můžeme takový filtr chápat jako číslicový obvod

s 9x8 bitovými vstupy a 8 bitovým výstupem. V praxi se pro potlačení lineárního šumu běžně používají průměrovací (váhové) filtry, pro potlačení výstřelového šumu různé varianty mediánových filtrů a pro detekci hran potom specializované operátory, např. Sobelův nebo Cannyho detektor [27]. Hlavním cílem pro evoluční návrh v této oblasti je najít co nejlépe filtrující operátor pro konkrétní typ šumu a současně takové řešení, jehož implementace na čipu zabere méně plochy než zmíněné konvenční implementace.



Obr. 3. Použití CGP pro návrh obrazových filtrů. „Obraz po filtraci“ je příkladem výstupu jednoho z evolučně navržených filtrů, který dokáže současně potlačit výstřelový šum a detekovat hrany.

Evoluční návrh nelineárního filtru, který má potlačit např. výstřelový šum a současně detekovat hrany v obraze (viz výstup filtru na obrázku 3), může být proveden pomocí CGP. Kandidátní filtr však nebudeme reprezentovat na úrovni hradel, ale na úrovni funkčních bloků, které mají dva 8-bitové vstupy a jeden 8-bitový výstup a které realizují vhodné funkce, např. průměr, minimum, maximum apod. (konkrétní množina funkcí je uvedena např. v [22, 31]). Reprezentace na úrovni funkčních bloků umožňuje pracovat s relativně krátkým chromozomem (tj. malým prohledávacím prostorem), který by při práci na úrovni hradel několikanásobně narostl. Typicky je použito pole velikosti 8 x 4 funkčních bloků, populace obsahuje 8 jedinců, $h = 1$, $L = 1$ a je vytvářeno 10-100 tisíc generací [22, 24].

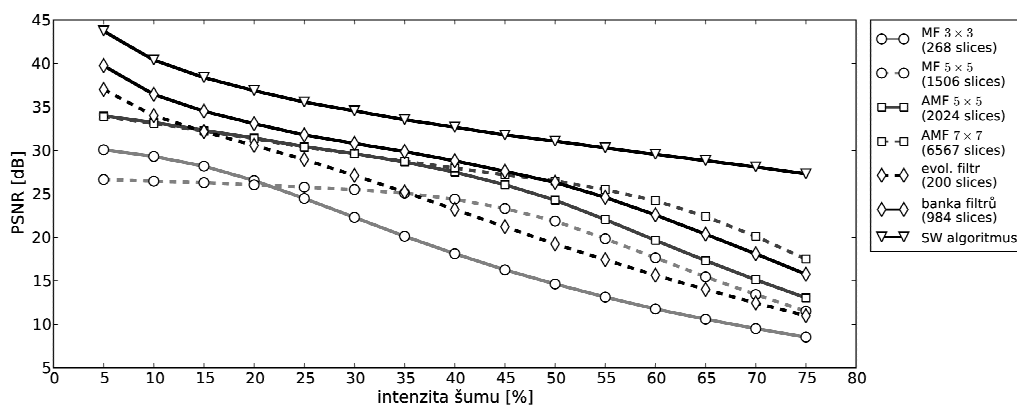
Při evaluaci kandidátního filtru není možné ověřit odezvu filtru pro všechny možné vstupní kombinace pixelů. Proto je v průběhu evoluce použit vhodně zvolený trénovací obraz. Cílem evoluce je najít filtr, který dosahuje co nejmenší odchylku ideální (nepoškozené) verze obrázku v od obrázku, který je filtrem generován. Pokud

označíme obrázek generovaný tímto kandidátním filtrem jako w , potom je fitness hodnota kandidátního filtru definována jako

$$fitness = \sum_{i=1}^M \sum_{j=1}^N |v(i, j) - w(i, j)|, \quad (2)$$

kde N a M jsou rozměry obrázku, typicky $M = N = 128$. Funkčnost a obecnost evolučně navrženého filtru je na konci evoluce ověřena pomocí sady testovacích obrázků.

V publikacích [22, 24, 31] bylo prokázáno, že evolučně navržené filtry pro různé typy šumu (např. sůl a pepř, dávkový výstřelový šum) a detektory hran dosahují minimálně stejné vizuální kvality výsledku ve srovnání s konvenčním řešením (mediánové filtry, Sobelův detektor apod.). Navíc tyto filtry, pokud by byly implementovány na čipu, zabírají podstatně méně plochy než konvenční řešení. Na obrázku 4 je znázorněna průměrná „vizuální kvalita“ vyfiltrovaného obrazu pomocí několika různých přístupů v závislosti na intenzitě impulsního šumu typu sůl a pepř. Průměrná vizuální kvalita byla získána na sadě 25 testovacích obrazů [24]. Evolučně navržený filtr (*evol. filtr*) je sice mnohem kvalitnější než konvenční mediánový filtr (*MF*) s filtrovacím jádrem 3×3 , ale kvalita filtrace značně klesá se vzrůstající intenzitou šumu. Použijeme-li mediánový filtr s větším filtrovacím oknem, je evolučně navržený filtr rovněž kvalitnější pouze do určité intenzity šumu. Navíc existuje propracovanější konvenční metoda – adaptivní mediánový filtr (*AMF*) – který poskytuje mnohem kvalitnější výsledky než klasický mediánový filtr [9]. Adaptivní medián pracuje s větším filtrovacím jádrem než 3×3 pixely a tudíž je i jeho obvodová realizace nákladnější (viz cena na obrázku 4). Pro vyšší intenzitu šumu se pomocí evolučního postupu kvalitnější filtr navrhnout nedaří, což je dáno tím, že je použito filtrovací jádro obsahující pouze 3×3 obrazové body. Nejlepší známá metoda pro odstranění šumu typu sůl a pepř [10], jejíž výsledek byl rovněž začleněn do obrázku 4 (*SW algoritmus*), není vhodná pro porovnání, protože neprovádí filtrace na základě lokální informace, ale na základě celého obrázku. Tento přístup je nevhodný pro obvodovou akceleraci.



Obr. 4. Průměrná kvalita filtrace vypočtená na sadě 25 obrázků, která byla dosažena různými filtry pro různou intenzitu šumu typu sůl a pepř. Cena filtru je vyjádřena počtem slices – konfigurovatelných elementů FPGA (viz kapitola 4.1).

Evoluční návrh však má jednu unikátní vlastnost: schopnost generovat odlišná a přitom obdobně kvalitní řešení. Nalezneme-li více (např. 3 nebo 5) kvalitních filtrů s jádrem 3x3 a necháme-li je filtrovat obraz paralelně, je velmi pravděpodobné, že většina z nich bude pro všechny devítice pixelů dávat na výstupu správnou hodnotu. Vytvoříme-li tedy obvod, který z výsledků těchto filtrů zvolí nejlepší řešení (např. pomocí mediánu), můžeme realizovat filtr (na obrázku 4 označený jako *banka filtrů*), který je schopen odstranit i šum s vyšší intenzitou. Filtr tímto způsobem vytvořený zabírá mnohem menší plochu než adaptivní mediánový filtr, přičemž dosahuje srovnatelné vizuální kvality výsledku. Evolučně navržená banka filtrů je chráněna v České republice užitným vzorem UV20017/2009 (patentová přihláška byla rovněž podána).

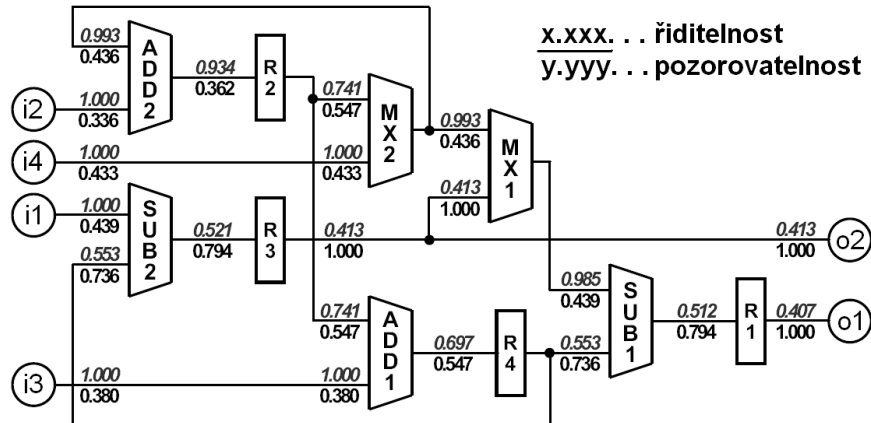
3.4 Evoluční návrh testovacích obvodů

Kromě bezvadné funkčnosti a nízkého příkonu je jedním z klíčových požadavků na současně složité číslicové obvody i snadná *testovatelnost* [17]. Obvody většinou podporují diagnostický režim, po jehož aktivaci je možné detekovat poruchy jak interních prvků tak i spojů. V diagnostickém režimu jsou na primární vstupy obvodu postupně připojovány vhodně zvolené hodnoty (tzv. testovací posloupnost, zkráceně *test*) a je sledována odezva na primárních výstupech, která je následně porovnána s předem vypočtenými referenčními hodnotami. Ideální test by měl obsahovat co nejmenší množinu testovacích vektorů a detekovat všechny požadované poruchy.

Pro návrháře číslicových obvodů je velmi důležité zjistit, jak obtížné bude vytvořit *test* pro navržený obvod, popř. přímo identifikovat obtížně testovatelné části obvodu. Pro tyto účely se používají *metody analýzy testovatelnosti* obvodů. Jejich výhodou je nižší časová složitost oproti metodám přímo generujícím test. Obecně přijímaná definice testovatelnosti bohužel neexistuje. Existuje však řada dílčích, konkrétně aplikačně orientovaných definic a z nich vycházejících metod, z nichž každá je obvykle konstruována pro jistou konkrétní úroveň popisu obvodu a pro zohlednění vybrané podmnožiny nákladů spojených s jeho testováním. Aby bylo možné existující a nově navrhované metody analýzy testovatelnosti posuzovat, je nutné disponovat sadou různě složitých testovacích obvodů, u kterých je známa míra testovatelnosti. Sada testovacích obvodů s požadovanými vlastnostmi a složitostí v této oblasti doposud neexistovala.

Práce [18] ukazuje, že je evoluční přístup schopen vytvářet obvody (sestavající po syntéze z milionu i více hradel), které mají požadovanou míru testovatelnosti a jsou tak reálně použitelné pro posuzování kvality metod analýzy testovatelnosti. Přístup použitý pro ohodnocení testovatelnosti kandidátního obvodu je založen na ohodnocení parametrů *řiditelnosti* a *pozorovatelnosti*. Řiditelnost resp. pozorovatelnost je chápána jako schopnost ovlivnit resp. změřit hodnotu signálu v daném místě obvodu. Řiditelnost resp. pozorovatelnost je vyjádřena hodnotou v rozsahu $\langle 0, 1 \rangle$. Hodnota řiditelnosti resp. pozorovatelnosti pak obvykle vyjadřuje míru snadnosti nastavení resp. zjištění hodnoty signálu v daném místě obvodu. Pro účely ohodnocení kandidátního obvodu je v našem případě vypočtena průměrná řiditelnost resp. pozorovatelnost všech

vstupů resp. výstupů u všech prvků obvodu. Obrázek 5 ukazuje příklad obvodu s vypočtenými hodnotami říditelnosti a pozorovatelnosti jednotlivých signálů obvodu podle metody [18].



Obr. 5. Číslicový obvod s ohodnocenou říditelností a pozorovatelností

Vstupem pro evoluční návrh je uživatelem zadaný počet primárních vstupů a výstupů obvodu, počet a typ prvků obvodu, požadavek na průměrnou říditelnost a pozorovatelnost a parametry algoritmu. Registry jsou vloženy automaticky na vhodná místa. Např. pro obvod na obrázku 5 by bylo zadáno:

- Počet vstupů: 4 x 8 bitů
- Počet výstupů: 2 x 8 bitů
- Komponenty: 2 x 8bit. odčítačka (SUB), 2 x 8bit. sčítačka, 2x 8bit. multiplexor
- Požadovaná průměrná testovatelnost: 0,75
- Požadovaná průměrná říditelnost: 0,75
- Velikost populace: 30
- Mutace: 0,02%
- Nahrazení populace: 95%
- Generací: 200

Vzhledem k tomu, že je uživatelem definován počet a typ komponent, obsahuje chromozom pouze informaci o propojení těchto komponent a připojení vstupů a výstupů. Algoritmus evolučního návrhu potom generuje obvody sestavené ze zadaných komponent a snaží se minimalizovat rozdíl mezi požadovanou říditelností resp. pozorovatelností a říditelností resp. pozorovatelností změřenou u konkrétního kandidátního jedince.

Zde je důležité upozornit na fakt, že fitness funkcí není vůbec posuzována funkčnost obvodu. Výstupem jsou syntetizovatelné obvody s požadovanou úrovní testovatelnosti a s požadovaným počtem hradel. Metoda byla použita k návrhu sady testovacích obvodů s různým počtem hradel a s různými parametry pozorovatelnosti

a řiditelnosti, která je k dispozici zájemcům na Internetu (viz [18]). Nejsložitější z navržených obvodů obsahuje po syntéze 1,2 milióny hradel. Tento obvod je rovněž největším obvodem, který byl (dle vědomí autorů) vytvořen pomocí evolučního návrhu.

3.5 Kdy je vhodné použít evoluční návrh?

U uvedených příkladů je zajímavé, že evoluční algoritmus pracuje s přibližně stejně velkým chromozomem (tj. v přibližně stejně velkém prohledávacím prostoru), který má velikost maximálně několik tisíc bitů. Další zvětšování velikosti chromozomu je neúnosné a evoluční prohledávání přestává poskytovat rozumné výsledky. Fakt, že evoluce může vytvořit užitečný obvod obsahující miliony hradel, je způsoben *znalostí*, která je dodána do metody řešení. Zatímco při evoluci na úrovni hradel návrhář prakticky nic nepředpokládá o výsledném řešení, u evoluce na úrovni funkčních bloků musí dodat relativně mnoho znalosti. Pokud chceme navrhovat složité obvody, je nutné, aby nemusely být v rámci výpočtu fitness hodnoty vypočteny odezvy pro všechny vstupní kombinace. Pokud stačí pracovat pouze s trénovací množinou, je možné zkrátit dobu výpočtu. Úspěch evolučního algoritmu můžeme očekávat zejména tam, kde je z různých důvodů neúplně nebo nepřesně zadaná specifikace. Obecně je možné konstatovat, že evoluční návrh je nejúspěšnější v těch oblastech, pro které platí [24]:

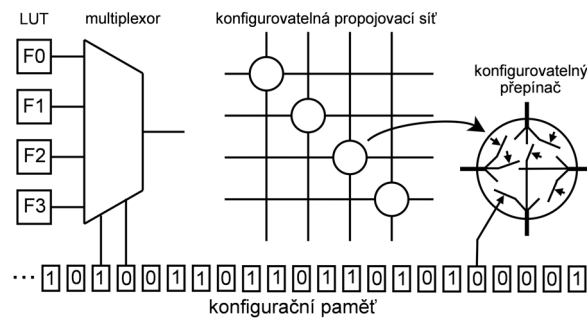
- Konvenční návrhová metoda není plně automatizovatelná a je z části založena na zkušenosti, experimentování a ad hoc postupech.
- Existuje kvalitní simulátor pro ohodnocení kandidátních řešení.
- Evaluace kandidátního řešení není příliš časově náročná.

4 Evoluce přímo v hardware

Tato podkapitola prezentuje některé příklady evolučního návrhu, kdy jsou kandidátní řešení ohodnocována ve fyzickém zařízení, které je umístěno v reálném prostředí.

4.1 Rekonfigurovatelné obvody

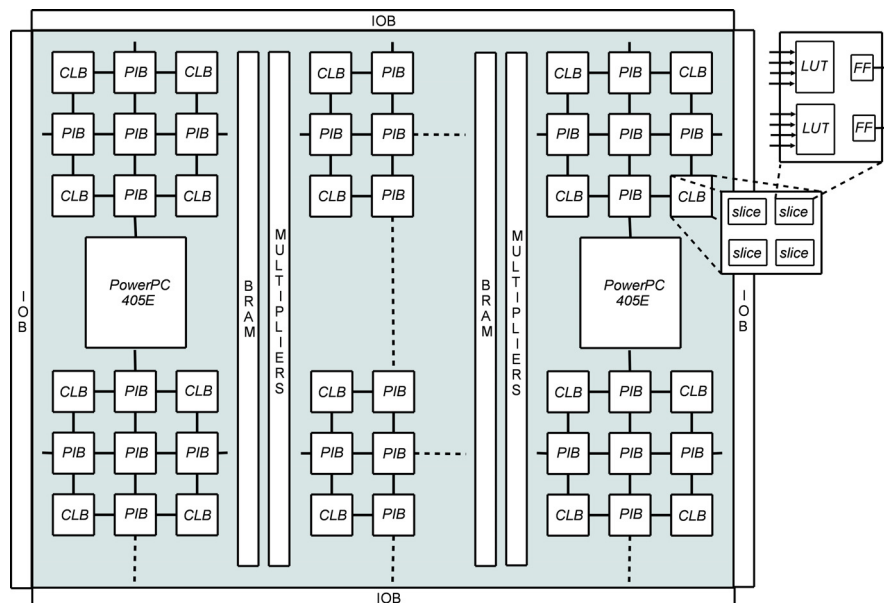
Rekonfigurovatelný obvod obvykle obsahuje pole programovatelných elementů, jejichž funkce, propojení a připojení k primárním vstupům a výstupům je definováno obsahem konfigurační paměti. Tato paměť je nejčastěji typu RAM a její obsah může být relativně rychle modifikován. Obrázek 6 ukazuje, že obsah této paměti ovládá programovatelné propojky, které určují funkci zařízení.



Obr. 6. Princip rekonfigurace propojení číslicového obvodu. Multiplexor připojí dle nastavení konfiguračních bitů jednu z funkčních jednotek F0-F3 na svůj výstup (vlevo). Propojovací síť je realizována pomocí konfigurovatelných přepínačů.

Hradlová pole FPGA

V oblasti číslicových obvodů patří mezi nejpobulárnější rekonfigurovatelné obvody programovatelná hradlová pole FPGA. Typická struktura FPGA firmy Xilinx je uvedena na obrázku 7. Jedná se o dvourozměrné pole programovatelných prvků, které obsahuje konfigurovatelné logické bloky (CLB), programovatelné propojovací bloky (PIB) a konfigurovatelné vstupně/výstupní bloky (IOB). CLB sestává z tzv. *slice*, přičemž každý *slice* obsahuje generátory logických funkcí realizované pomocí náhledových tabulek (lookup table, LUT) s 3 až 6 vstupy (v závislosti na variantě produktu), klopné obvody a pomocnou logiku. Konfigurační soubor je uložen v paměti SRAM, která je součástí čipu.



Obr. 7. Konceptní schéma obvodu FPGA řady Virtex Pro [33]

Obvody FPGA se liší množstvím a typem konfigurovatelných prvků, které najdeme na čipu. Nejmodernější FPGA obsahují více než 10 tisíc CLB a navíc další specializované komponenty, jako paměti SRAM, rychlé násobičky, procesory (viz procesor PowerPC na obrázku 7), gigabitová rozhraní atd. Integrace těchto specializovaných komponent přímo na čip je důsledkem faktu, že jsou často používány, a bylo by neekonomické, aby je každý uživatel implementoval pomocí CLB a dalších standardních zdrojů.

Obvody FPGA mohou být konfigurovány externě nebo interně. V případě externí rekonfigurace, což je nejběžnější způsob, jak konfigurovat FPGA, jsou konfigurační data nahrána z externí paměti. Interní rekonfigurace je v FPGA řady Virtex podporována pomocí rozhraní ICAP (Internal Access Configuration Port), které umožňuje číst/zapisovat konfigurační data přímo kontrolérem umístěným na čipu. Některé obvody FPGA podporují dynamickou částečnou rekonfiguraci, při které je možné rekonfigurovat část čipu, zatímco jiná část čipu provádí výpočet. Podpora interní rekonfigurace je nutnou podmínkou pro realizaci adaptivního hardware na jednom čipu. Z pohledu evoluční elektroniky je nevýhodné, že výrobci FPGA neposkytují popis konfiguračního souboru FPGA.

Vzhledem k tomu, že je proces návrhu obvodů pro FPGA velmi podobný programování, je možné vytvořit výslednou aplikaci relativně rychle. Návrhář musí nejdříve popsat strukturu obvodu nebo jeho chování pomocí specializovaného jazyka (např. VHDL, Verilog, popř. stačí použít i vhodně rozšířený jazyk C – např. SystemC). Následně je zdrojový kód téměř automaticky přetransformován na konfigurační data pro FPGA. Tento proces, jenž zahrnuje syntézu, rozmístění komponent a jejich propojení v FPGA, je prováděn specializovanými CAD nástroji. Pro syntézu je možné zavést další omezení, např. definovat maximální povolenou velikost obvodu, optimalizovat pracovní frekvenci apod. Rovněž je možné simulovat mezivýsledky transformace a v případě nesplnění požadavků provést změny v návrhu a spustit transformaci znovu.

Rekonfigurovatelné analogové obvody

V oblasti analogových obvodů rovněž existují rekonfigurovatelné čipy. Nejprve uvedeme seznam nejpoužívanějších zkratk:

- FPAA (Field-Programmable Analog Array) – integrovaný obvod, který může být naprogramován tak, aby implementoval analogový obvod. Využívá flexibilní analogové bloky a konfigurovatelné propojení.
- FPMA (Field-Programmable Mixed-Analog-Digital Array) – integrovaný obvod, který obsahuje FPAA, FPGA a konfigurovatelné konvertory za účelem implementace obvodů pracujících se smíšenými signály.
- FPTA (Field-Programmable Transistor Array) – varianta FPAA, kdy základním konfigurovatelným blokem je tranzistor.
- CAB (Configurable Analog Block) – základní programovatelná buňka FPAA.

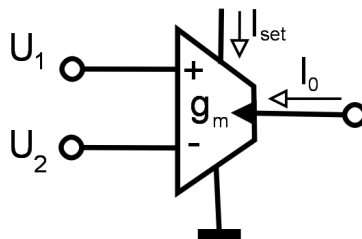
FPAA obsahují pole bloků CAB, programovatelnou propojovací síť, konfigurovatelné I/O a další podpůrné obvody. Na čipu je rovněž umístěna konfigurační paměť. Pro

konfiguraci zapojení jsou použity buď tranzistorové spínače (přenosová hradla), spínané kapacitory nebo transkonduktanční operační zesilovače. Stručně vysvětlíme princip změny konfigurace pomocí transkonduktančního operačního zesilovače, protože v současnosti vede na nejzajímavější aplikace.

Transkonduktanční zesilovače (v literatuře někdy označované také zkratkou OTA) jsou součástky podobné operačním zesilovačům. Jejich výstup však není napěťový, ale je proudový. U transkonduktančního zesilovače proto nemůžeme hovořit o napěťovém či proudovém přenosu, ale hovoříme o přenosové vodivosti (transkonduktanci). Vztah mezi výstupním proudem I_o a vstupními napětími na invertujícím (U_-) a neinvertujícím (U_+) vstupu je tedy:

$$I_o = -g_m(U_+ - U_-),$$

kde g_m je přenosová vodivost. Ta může být buď pevná (daná vnitřním uspořádáním zesilovače) nebo nastavitelná zvláštním vstupem. Vstupní i výstupní odpor transkonduktančního zesilovače je ideálně nekonečný. Na obrázku 8 je zesilovač OTA s nastavitelnou transkonduktancí, kterou je možné zvolit pomocí proudu I_{set} . Tato varianta se využívá zejména pro realizaci rekonfigurovatelných analogových filtrů, protože velikost I_{set} je možné jednoduše algoritmicky nastavovat pomocí DA převodníku. Hlavní výhodou rekonfigurovatelných obvodů na bázi OTA oproti jiným možnostem rekonfigurace (např. spínáním kapacitorů) je relativně vysoký dosažitelný pracovní kmitočet (stovky MHz).



Obr. 8. Transkonduktanční operační zesilovače s možností konfigurace g_m pomocí I_{set}

4.2 FPGA jako akcelerátor pro evoluční návrh

Obvody FPGA jsou využitelné pro akceleraci evolučního návrhu a pro realizaci vestavěných adaptivních systémů. Například obvod FPGA Virtex II Pro [33], který obsahuje 23616 slices, 49788 klopných obvodů, 852 vstupně-výstupních bloků, 232 vestavěných pamětí Block RAM (BRAM) a dva vestavěné procesory IBM PowerPC 405, byl využit pro akceleraci evolučního návrhu obrazových filtrů (kapitola 3.3). V FPGA je implementováno pole 8 x 4 programovatelných bloků, se kterým pracuje CGP. Nejedná se o evoluci na úrovni konfiguračních dat FPGA, ale na úrovni chromozomu CGP, který je uložen v uživatelem vytvořeném 384-bitovém registru. Dále je obvodově realizována fitness jednotka. Trénovací obrázky o velikosti 128 x 128 pixelů jsou uloženy v externích pamětech SRAM. Genetické operace jsou

prováděny procesorem PowerPC, který je vestavěn v FPGA. Výpočet je řízen obvodově realizovaným řadičem.

První verze akcelérátoru běžící na kmitočtu 100 MHz umožňuje urychlit evoluční návrh 44krát oproti softwarové implementaci v procesoru Celeron 2,4 GHz [31]. Za 1 sekundu dokáže akcelérátor ohodnotit až 6000 kandidátních filtrů. Jeden běh evoluce je dokončen v průměru do 10 s. Nejnovější verze akcelérátoru obsahuje čtyři paralelně pracující pole programovatelných bloků a čtyři jednotky pro výpočet fitness. Dosahuje tak urychlení až 170 oproti referenční implementaci běžící na procesoru Celeron 2,4 MHz [32].

4.3 Evoluce na úrovni konfigurace FPGA

Cílem experimentu A. Thompsona z roku 1996 bylo nalezení obvodu, který pracuje jako tónový diskriminátor v programovatelném hradlovém poli XC6216 [28]. Tento čip je jedním z předchůdců v současnosti nejpopulárnější rodiny Virtex. Čip XC6216 byl výrazně jednodušší než FPGA na obrázku 7, ale hlavně podporoval parciální dynamickou rekonfiguraci a návrháři byl znám význam každého konfiguračního bitu. Nicméně stejně jako u moderních FPGA výrobce předpokládal, že obvod bude používán výhradně pro konstrukci číslicových systémů konvenčním způsobem a neočekával, že by elementy obvodu pracovaly např. jako analogové komponenty.

Cílový obvod – tónový diskriminátor – má generovat log. 1, pokud je vstupní signál nastaven na frekvenci 10 kHz, a log. 0, pokud je vstupní signál nastaven na frekvenci 1 kHz. Evoluce byla prováděna přímo na úrovni konfiguračního řetězce a pro kandidátní obvody bylo vyhrazeno pole 10 x 10 logických bloků, což je velmi málo, pokud by měla být úloha řešena konvenčním způsobem. K zakódování řešení bylo potřeba 1800 bitů. Protože bylo na úrovni konfiguračního řetězce „vše povoleno“, mohly vzniknout netradiční asynchronní obvody a dokonce obvody, které je vhodnější označit za analogové než digitální. Zajímavé je, že programovatelné elementy XC6216 mají zpoždění v jednotkách nanosekund, kdežto požadované chování je na úrovni milisekund (nebylo povoleno připojit žádné další externí členy). Fitness funkce vyčísluje rozdíl mezi průměrným napětím na výstupu pro sérii vstupů s frekvencí 1 kHz a sérii vstupů s frekvencí 10 kHz.

Nalezené řešení je asynchronní, používá jen několik málo programovatelných elementů (konvenční řešení by bylo výrazně složitější), vykazuje netypické zpětné vazby a využívá vlastnosti konkrétního materiálu, což je naprosto nevídané na této platformě určené pro číslicové obvody. Do dnešní doby se nepodařilo přesně zjistit, jak a proč tento obvod funguje! Bylo zjištěno, že když se nalezená konfigurace nahraje do jiného FPGA (ovšem stejného typu), pak obvod nefunguje. Pokud se podle architektury FPGA a této konfigurace vytvoří model nalezeného obvodu pro simulátor, výsledky simulace nesouhlasí s naměřenými hodnotami.

Výsledek byl interpretován tak, že je EA schopen využít pro řešení problému i ty vlastnosti platformy (např. chemické a fyzikální vlastnosti konkrétního polovodiče) a prostředí (teplotu, elektromagnetické pole, ...), které běžný návrhář neuvažuje. Vzniká problém portability (přenositelnosti): řešení pracuje jen na platformě, kde bylo vyevolováno a jen při okolních podmínkách, které existovaly během evoluce.

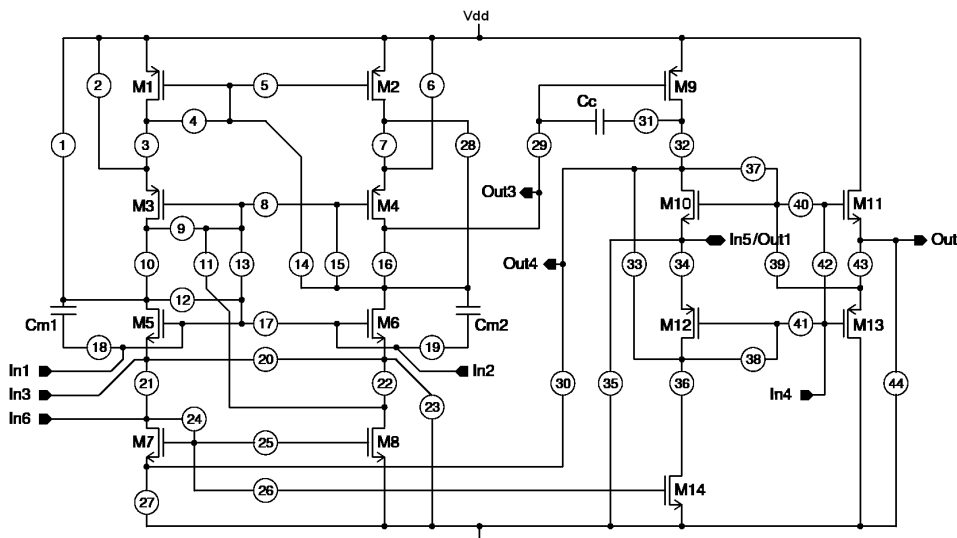
Thompson dále ukázal, že je možné evolučně v několika málo generacích „doladit“ řešení, které bylo nalezeno pro čip A, i pro čip B (stejného typu). Robustní řešení je také možné získat tak, že kandidátní obvody budeme během evoluce ověřovat v různých prostředích a na různých platformách.

Popsaný experiment demonstruje, že evoluční design umí něco, co člověk jako designér dobře nedokáže – precizně konfigurovat určité fyzikální systémy. Tento experiment byl motivací pro další výzkumníky.

4.4 Evoluce v extrémním prostředí

FPTA-2 je programovatelné pole tranzistorů, které obsahuje 64 programovatelných buněk. Buňky je možné programovat a propojovat pomocí vhodné konfigurace. Každá z buněk obsahuje 14 tranzistorů a 77 propojek (obrázek 9). Tento obvod byl použit výzkumným týmem z NASA JPL pro evoluční návrh analogových i digitálních obvodů na úrovni tranzistorů [26]. Zejména byl zkoumán vliv extrémního prostředí (teplota, radiace) na obvody vytvořené v FPTA-2 a schopnost evoluce opravit obvody poškozené tímto prostředím.

Chromozom opět přímo reprezentuje konfigurační řetězec FPTA-2. Jedna buňka je konfigurována pomocí 77 bitů, které jsou zaslány do FPTA v pěti 16b slovech. Ohodnocení kandidátního obvodu probíhá tak, že počítač generuje stimuly pro FPTA (např. při evoluci logického členu NOR na úrovni tranzistorů generuje každou ze čtyř vstupních kombinací po dobu 1 μ s – viz obrázek 10), potom čte výstup FPTA, převede ho do digitální podoby a vypočítá fitness hodnotu, která je definována jako rozdíl mezi hodnotou změřenou na výstupu FPTA a požadovanou hodnotou (ideální odezva NOR vyznačená šedou barvou na obrázku 10) pro K vzorků získaných během evaluace (v našem příkladu je to např. $K = 200$ vzorků získaných během 4 μ s).

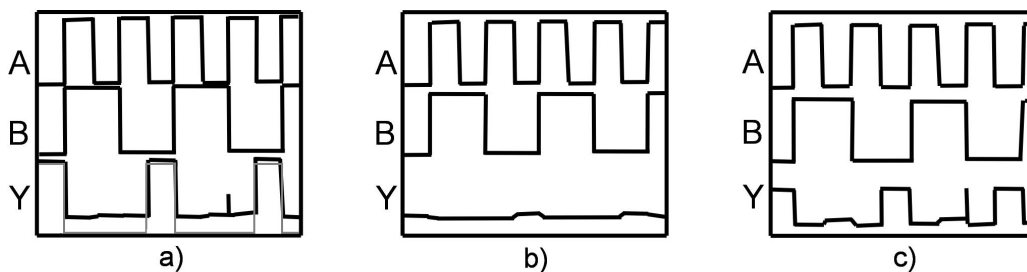


Obr. 9. Jedna programovatelná buňka čipu FPTA-2. Kroužek označuje programovatelný spínač.

Při evoluci na úrovni tranzistorů je mnohem jednodušší pozorovat vliv různých externích faktorů na výsledek evoluce než na úrovni logických členů. Častým jevem je např. situace, kdy evolučně navržený logický člen NAND pracuje korektně pouze při té frekvenci vstupních dat, která byla použita při výpočtu fitness hodnoty.

Za extrémní pro elektroniku považujeme teploty mimo rozsah -65°C až 125°C [4]. Při nízkých teplotách je nedostatek nosičů v polovodičích (protože dopanty nejsou dostatečně ionizovány). Při vysokých teplotách přestává pracovat PN přechod, nekontrolovaně přes něho protéká proud. Radioaktivní záření způsobuje u CMOS tranzistoru změnu závislosti proudu I_d na napětí hradla, což může mít neodstranitelný vliv, pokud je obvod vystaven určité dávce. SEU (Single Event Upset) je naopak jev dočasný, kdy dochází k náhodnému překlopení bitu paměti. Nahráním původní informace do paměti dojde k opravě poruchy. Tradičně se funkce elektroniky v extrémním prostředí, které je typické pro vesmírné aplikace, zajišťuje zaváděním různých kompenzačních obvodů nebo použitím speciálních materiálů.

Experimentálně bylo prokázáno, že EA může obnovit funkci obvodu (např. logického hradla, DA převodníku apod.) vytvořeného a funkčního při pokojové teplotě, který je následně vystaven vlivu extrémního prostředí a přestává v něm pracovat. Toto extrémní prostředí způsobuje, že komponenty použité v obvodu se stávají komponentami s jinými elektrickými vlastnostmi. EA je využit, aby z těchto pozměněných komponent sestavil nové řešení. Jednalo se o znovuoobnovení funkce pro nízké teploty (do -189°C), vysoké teploty (do 300°C) a radioaktivní záření (do kumulativního ozáření 250 krad) [25, 26, 34]. Příklad odezvy hradla NOR v extrémním prostředí a odezvy získané pro evolučně obnovené zapojení je znázorněn na obrázku 10.



Obr. 10. (a) Chování hradla NOR při pokojové teplotě. Šedá čára vyznačuje ideální požadovanou odezvu hradla. (b) Ztráta funkce při teplotě 326°C . (c) Obnověná funkce při teplotě 326°C . Vstupní signály jsou označeny A a B, výstup je označen Y.

4.5 Evoluce „in materio“

Při studiu experimentu A. Thompsona si J. Miller uvědomil, že EA může být velmi dobrým nástrojem pro automatickou „konfiguraci“ vhodného materiálu tak, aby prováděl požadovanou funkci – vznikl tak koncept evoluce „in materio“ [5].

Jeden z navržených scénářů vypadá takto: K vhodnému materiálu jsou na určitá místa připojeny elektrody, jejichž prostřednictvím je aplikováno elektrické konfigurační napětí. Při vhodně nastavených hodnotách konfiguračního napětí dochází ke změnám materiálu na molekulární úrovni. Byly testovány různé materiály. Pro experiment byl vybrán LCD displej. Je známo, že orientaci molekul kapalných krystalů je možné řídit elektrickým polem. Změna orientace těchto molekul potom mění optické a elektrické vlastnosti kapalných krystalů. Cílem evoluce je tedy najít taková konfigurační napětí, která způsobí takovou konfiguraci kapalných krystalů, že budou provádět požadovanou funkci. Podařilo se vyevolvovat různá chování, např. již zmíněný tónový diskriminátor, logické obvody nebo kontrolér pro robota [5].

Zde je nutné poznamenat, že není znám konvenční způsob, kterým by bylo možné takto materiál „nakonfigurovat“. Tento výsledek je významný z pohledu stále intenzivnějšího výzkumu v oblasti nanotechnologií, kdy se výzkumníci snaží různými technikami řídit vlastnosti a chování materiálů na úrovni atomů a molekul.

V oblasti molekulární elektroniky vznikla platforma nazvaná NanoCell, což je pole sebesestavujících se kovových částic propojených molekulárními přepínači [29]. Na určitá místa jsou připojeny elektrody, prostřednictvím kterých je možné aplikovat proudové pulsy a tím měnit konfiguraci NanoCell. Dochází ke změnám voltampérové charakteristiky molekuly nitroanilinu a tím i ke změně chování NanoCell. Pomocí EA byly na této platformě vytvořeny různé obvody, např. logická hradla.

4.6 Evoluce kvantových chování

Bartels a kol. se zabývali tvarováním laserových pulsů s cílem optimalizovat kvantové interference v dutém vlnovodu o průměru 175 μm , který byl naplněn argonem [1]. Tvarování bylo umožněno použitím deformovatelného zrcadla, jehož parametry byly hledány EA. Při tomto procesu vzniká řada harmonických. Některé měly být zvýrazněny, jiné potlačeny. Při experimentu bylo vygenerováno mnoho dat, která byla následně analyzována a porovnána s teoretickými predikcemi. Byla zjištěna chování, o kterých se doposud vědci domnívali, že nejsou fyzikálně vůbec možná (např. antikorelované harmonické v doméně attosekund). Podařilo se tak automatickým způsobem řídit kvantový systém, což před tím nebylo možné. Očekává se využití těchto principů v oblasti nanotechnologií.

5 Problém implementace a evoluční návrh

V předchozí kapitole jsme ukázali, že EA je schopen vytvářet elementární výpočetní systémy na konvenčních platformách (jako jsou rekonfigurovatelné číslicové a analogové obvody) i na speciálních, avšak jistým způsobem rekonfigurovatelných, platformách. Dále jsme ukázali, že EA může využít pro realizaci požadovaného chování specifické vlastnosti platformy i prostředí. Konvenční návrhové techniky prakticky nemají možnost využít tyto netradiční zdroje, které EA použít umí. Hlavním důvodem je, že tyto konvenční techniky vyžadují existenci abstraktního modelu

platformy. Evoluční návrh používající metodu „vygeneruj řešení a otestuj ho“ model platformy nepotřebuje.

Evolučně navržené výpočetní systémy představují velmi zvláštní a doposud prakticky nezkoumanou třídu výpočetních systémů. Pro další úvahy vyjdeme z předpokladu, že EA je schopen v zadané fyzické platformě v rozumné době vyevolvovat takové chování, které vyžaduje specifikace zadaná uživatelem. V dnešní době nebude složitost takových systémů velká, ale můžeme očekávat pokrok v tomto směru. Je tak vytvořena implementace, která plní požadovanou funkci. Obecně ale nerozumíme, jak a proč řešení funguje. Intuitivně předpokládáme, že takový systém je výpočetním systémem (mechanismem). Z pohledu teorie implementace to však vůbec jasné není.

5.1 Problém implementace

V literatuře orientující se na přesah informatiky do filozofie je zkoumána otázka: Kdy je fyzikální systém výpočetním mechanismem? Jako stěžejní pro odpověď na tuto otázku považujeme řešení *problému implementace*: jaký je vztah mezi abstraktním výpočtem a jeho fyzickou realizací? Existuje zde celá řada přístupů. Shrňme si základní postoje podle [23].

Putnam říká, že musí existovat zobrazení mezi abstraktními výpočetními stavy a fyzikálními stavy, kterými systém během výpočtu prochází. Je kritizován (např. Scheutzem), protože podle této definice potom jakýkoliv fyzikální systém může být chápán jako implementace libovolného výpočtu. Scheutz preferuje začít ve fyzickém světě. Popisuje způsob, jakým je možné pro vstupy, výstupy a chování fyzikálního systému (který je popsán ve zvolené fyzikální teorii) vytvořit zobrazení na abstraktní výpočetní model. Tento přístup nevyžaduje koncept fyzického stavu [21]. Piccinini definuje výpočetní mechanismus jako mechanismus, jehož účelem je získat výstupní řetězec symbolů ze vstupního řetězce symbolů na základě obecně platného pravidla, které platí pro všechny vstupy a výstupy [19]. Definuje šest požadavků, které musí fyzikální systém splňovat, aby mohl být výpočetním mechanismem – zejména nezávislý pozorovatel musí být schopen „počítání“ identifikovat na základě studia systému. Potom je možné otázku, zda je fyzikální systém výpočetním mechanismem, formulovat jako hypotézu a ověřit její platnost na základě prozkoumání fyzikálního systému. Copeland, Johnson aj. vyžadují, aby existovala konzistentní interpretace symbolů uvnitř systému během výpočtu. Tato interpretace musí být deklarována před započítáním výpočtu [11].

5.2 Třída evolučně navržených výpočetních zařízení

Jaké řešení má „problém implementace“ pro evolučně navržená výpočetní zařízení? Uvažme následující situaci: Cílem je najít konfiguraci rekonfigurovatelného zařízení (RZ), které bude realizovat funkci F požadovanou uživatelem. Abychom mohli definovat fitness funkci, musíme před započítáním experimentu zvolit interpretaci fyzického chování RZ z pohledu vstupů a výstupů RZ. Můžeme např. přiřadit vhodným napěťovým úrovním na vstupech a výstupech logické hodnoty.

Evoluční algoritmus může objevit takové řešení problému (konfiguraci) C , které bude fungovat (tj. fyzicky realizovat požadované chování – funkci F) jen v daném rekonfigurovatelném zařízení (RZ). Konfigurace C nefunguje v simulátoru ani v jiném RZ stejného typu a funguje jen za podmínek (teplota, tlak, ...), které existovaly na konci evoluce. V některých případech nejsme schopni vysvětlit, jak RZ realizuje funkci F . Inženýr prakticky není schopen konvenčním způsobem najít konfiguraci C . Jaké jsou tedy odpovědi na otázku: Je RZ konfigurované dle C a počítající F výpočetním mechanismem?

ANO: Pokud stačí, že pro zadané vstupy získáváme požadované výstupy dle interpretace, kterou jsme zavedli před tím, než byla spuštěna evoluce, potom je RZ výpočetním mechanismem. Pokud je RZ výpočetním mechanismem, potom funkce, které jsou vyčíslitelné pomocí RZ, jsou vyčíslitelné i na Turingově stroji.

NENÍ: Přísnější požadavek předpokládá, že symboly uvnitř RZ musí mít konzistentní interpretaci, která je zavedena před spuštěním evoluce. Tuto interpretaci však nelze zavést před započítím evoluce. EA může využít „cokoliv“ pro realizaci požadované funkce a my to dopředu principiálně nemůžeme odhalit. Dokonce nelze zavést tuto interpretaci ani na konci evoluce po prozkoumání RZ! Tímto zkoumáním může být realizace poškozena a vytvořené chování ztraceno. Závažnějším argumentem však je, že EA může teoreticky pro zajištění a konstrukci řešení využít fyzikální jevy, které ještě nikdy nebyly popsány (viz kapitola 4.6). Tudíž zkoumáním RZ je nemožné pochopit činnost RZ, aniž bychom tyto jevy nejdříve poznali.

Rozhodnutí o tom, zda můžeme považovat evolučně navržené zařízení za výpočetní mechanismus, závisí na volbě definice výpočetního mechanismu. Pokud souhlasíme s tím, že nezávisí na tom, jak zařízení realizuje funkci, jsme schopni interpretovat jeho vstupně-výstupní chování jako výpočet a pokud jsme schopni tuto interpretaci zavést před započítím evoluce, potom můžeme zařízení považovat za výpočetní mechanismus. Pokud však je podstatné pro výpočetní mechanismy, že existuje korespondence mezi abstraktním a fyzickým, která je identifikovatelná pro nezávislého pozorovatele, potom naše zařízení není výpočetním mechanismem. Tabulka 1 porovnává tři typy fyzicky existujících systémů, jejichž chování bývá často interpretováno jako výpočet.

Tabulka 2. Vlastnosti výpočetních zařízení

| Vlastnost | Běžné počítače | Mozek | Zařízení navržená EA |
|--|----------------|-------|----------------------|
| Reakce výstupů na zadané vstupy lze interpretovat jako výpočet | ANO | ANO | ANO |
| Abstraktní model výpočtu existuje před implementací | ANO | NE | NE |
| Uživatel může definovat požadované chování | ANO | NE | ANO |
| Inženýři mohou vyrobit | ANO | NE | ANO |

Evoluční návrh obvodů a výpočetních systémů patří do oblasti elektroniky a počítačového inženýrství. V této analýze jsme se pokusili interpretovat evoluční návrh obvodů a výpočetních zařízení z pohledu teoretické informatiky. Ukázali jsme, že evolučně vytvořená výpočetní zařízení představují specifickou podtřídu výpočetních zařízení, která vykazuje zvláštní kombinaci vlastností, kterou nemají standardní počítače ani živé systémy, jejichž chování obvykle interpretujeme jako výpočet. Obecně není možné zjistit, jak a proč vyevolvované řešení funguje. Stávající teorie implementace, které řeší vztah mezi abstraktním a fyzickým počítáním, jsou potom nepoužitelné.

Klasická teoretická informatika definuje výpočet abstraktně, nejčastěji jako posloupnost přechodů mezi definovanými abstraktními stavy. Takto definovaný výpočet může být implementován pomocí různých fyzikálních principů. Vyevolvované výpočetní systémy jsou však závislé na své fyzikální podstatě a nemohou bez ní existovat. Pokud bychom odhalili abstraktní model, podle kterého provádí výpočet, potom bychom mohli tento výpočet přenést i na jiné platformy.

6 Patentová ochrana výsledků získaných evolučním návrhem

V kapitole 3.1 jsme se zmínili, že v soutěži Humies může získat ocenění evolučně navržené řešení, které je znovuoobjevením patentovaného vynálezu, vylepšením patentovaného vynálezu nebo by dnes mohlo být kvalifikováno jako patentovatelný vynález. V této podkapitole se budeme zabývat otázkou, jak by v budoucnu mohl evoluční návrh ovlivnit chápání patentového práva. Nejdříve však popíšeme princip patentu a zejména problémy spojené s patentováním software.

6.1 O patentech obecně

Patent je výhradní právo, jenž přiznává stát vynálezu, který je nový, má tvůrčí úroveň a lze jej průmyslově využít [36]. Majitel patentu může legálně zabránit dalším osobám využívat vynález. Majiteli patentu je tak umožněno zajistit si příjem, který má kompenzovat náklady vynaložené na výzkumnou, vývojovou, realizační a další činnost spojenou se získáním patentu. Platnost patentu je obvykle 20 let a je omezena na určité teritorium. Majitel je ale povinen zveřejnit obsah vynálezu, který se tak stává veřejně známou součástí techniky. Tento princip je výhodný i pro společnost, protože podporuje vznik inovací. Ačkoliv nemůže patentovaný vynález nikdo kromě majitele legálně využívat, může se stát ihned po zveřejnění inspirací pro další vynálezce a vést k novým inovacím.

Aby byl vynález patentovatelný, musí splňovat několik požadavků. Především se musí jednat o *patentovatelný předmět*. Většinou je výslovně uvedeno, co patentovat nelze. V některých zemích například nelze patentovat software, který může být chráněn jiným nástrojem – autorskými právy. Dále přihlašovatel prokazuje *novost vynálezu* na základě podrobné rešerše stavu techniky. Vynález musí vykazovat tvůrčí úroveň, tj. musí být *nezřejmý* odborníkovi v příslušném technickém oboru (s přihlédnutím ke stavu techniky). Tento požadavek má zaručit, že patent nebude

udělen za pouhé vylepšení již existujícího řešení, ke kterému může dospět osoba s obvyklými schopnostmi a znalostmi v daném oboru. Dalším požadavkem je *průmyslová využitelnost* neboli užitečnost. Nelze totiž získat patent například za objevení nějakého teoretického fenoménu, u kterého není jasná využitelnost. A konečně, vynález musí být v přihlášce objasněn tak, aby jej mohl realizovat odborník v konkrétním technickém oboru.

V některých zemích (včetně České republiky) je možné získat tzv. *užitný vzor*, což je jednodušší než získat patent. Pro užitné vzory je typické, že požadavek na tvůrčí úroveň je nižší a řízení o udělení je kratší. Užitný vzor s příslušnou právní ochranou tak mohou být uděleny relativně rychle. Doba platnosti užitného vzoru je rovněž kratší (např. 4 roky). Současně s přihláškou užitného vzoru je obvykle podávána i řádná patentová přihláška.

6.2 Softwarové patenty

Tradičně jsou patentovány produkty a výrobní procesy. Patentová ochrana software existuje pouze krátkou dobu a jen v určitých zemích. V Evropě platí Evropská patentová úmluva, která explicitně vyjímá matematické postupy a počítačové programy ze seznamu patentovatelných předmětů. Nicméně je fakticky možné patentovat postupy, jejichž význačnou součástí je počítačový program. Naopak v USA se stalo velmi populární patentovat software. Přibližně 15% ze všech patentů je uděleno na software a počet žádostí o udělení softwarového patentu vzrostl za posledních 20 let 18krát [20].

Tento stav vede k situaci, kdy většina klíčových (částí) programů je patentována a není téměř možné vytvořit nový program, aniž by nebyl některý z existujících patentů porušen. Protože patenty na software donedávna neexistovaly, bylo možné po jejich zavedení získat celkem jednoduše patentovou ochranu na téměř jakýkoliv nový software, a přitom neporušit jiný patent. Toho využily velké společnosti, které mají prostředky na podávání a udržování širokého spektra patentů. Navíc je často velmi obtížné zjistit, zda konkrétní programová konstrukce je či není chráněna patentem, protože patentová přihláška popisující určitý vynález může obsahovat celou řadu patentových nároků, které s vynálezem souvisejí jen velmi málo. Dochází tak k blokování nových inovací, protože majitelé patentů namísto tvorby nových aplikací sledují, zda někdo neporušuje jejich patenty, a pokud se tak děje, požadují licenční poplatky [15]. Mezi velkými firmami se používá (prakticky bezplatné) křížové licencování, které dovoluje firmě A využívat patentované technologie firmy B a naopak. Malé firmy vlastníci jeden nebo několik málo patentů jsou v nevýhodě, protože za křížovou licenci sice získají možnost využívat část technologie patentovanou velkou společností, ale velká společnost získá bezplatně přístup k jejich výsledkům tvůrčího procesu.

6.3 Bude nutná nová interpretace patentového práva?

Pokud nahlédneme do patentových databází, tak zjistíme, že v souvislosti s evolučním hardware najdeme patentované:

- obvody navržené evolučně, např.:
 - US Patent 6847851: Apparatus for improved general-purpose PID and non-PID controllers (Koza et al., 2005)
 - užitný vzor UV 020017/2009: Nelineární obrazový filtr (Sekanina, Vašíček, 2009)
- evoluční algoritmy, např.:
 - US Patent 7117186: Method and apparatus for automatic synthesis of controllers (Koza et al., 2006)
 - US Patent 6526556: Evolutionary technique for automated synthesis of electronic circuits (Stoica et al., 2003)
- platformy pro evoluční návrh obvodů, např.:
 - US Patent 6728666: Evolvable circuit with transistor-level reconfigurability (Stoica et al., 2004)
 - US Patent 6222381: Self-configurable parallel processing system made from self-dual code/data processing cells utilizing a non-shifting memory (Durbeck a Macias, 2001)

V českém prostředí je možné získat patent na evolučně vytvořené řešení, popř. na rekonfigurovatelnou platformu použitou pro evoluční návrh. Není možné získat patent na nový evoluční algoritmus. V zásadě platí, že pokud evoluční algoritmus nalezne řešení problému, které splňuje požadavek na udělení patentu, je možné sepsat patentovou přihlášku popisující toto řešení a patent by měl být udělen. Není nutné uvést, že řešení bylo získáno pomocí evolučního návrhu.

V monografii [20], kterou sepsal přední americký právník zabývající se softwarovými patenty – Robert Plotkin, se uvádí, že tento stav je dlouhodobě neudržitelný a nevýhodný jak pro vynálezce, tak i pro společnost. Plotkin analyzuje situaci, která by mohla nastat v blízké budoucnosti, kdy budou volně k dispozici systémy pro evoluční návrh použitelné pro rutinní generování vynálezů. Pro ty, kteří budou disponovat finančními prostředky na poplatky za patentové přihlášky a udržování patentů, nebude příliš obtížné vygenerovat velké množství vynálezů a získat příslušné patenty. Mohlo by dojít ještě k horší situaci než je v současné době v oblasti patentování software. Tato situace se nazývá *patentová záplava* (patent flood) a vede k zablokování nových inovací, protože klíčové vynálezy jsou v rukou několika mazaných vynálezců (savvy inventors). Současně je zpomalena činnost patentového úřadu, protože posuzování patentových přihlášek je prováděno lidskou silou a při jejich velkém počtu nebudou patenty uděleny v rozumném čase. Navíc je pro člověka často velmi obtížné pochopit, jak funguje evolučně vytvořený vynález. Dalším problémem je, že evoluční algoritmus v jednotlivých bĕzích často vygeneruje různá a přitom stejně kvalitní řešení. Evolučně vytvořený patentovaný vynález by někdo mohl obejít tak, že použije evoluční algoritmus a vytvoří alternativní řešení, které nebude předmětem patentové ochrany.

Plotkin nabízí řešení popsané situace, které je založeno na nové interpretaci stávajícího přístupu k udělování patentů. Zamĕrme se na některé zajímavé postřehy související s důsledným aplikováním stávajících požadavků na přihlašované patenty.

Jedním ze základních požadavků je tvůrčí úroveň, tj. vynález musí být nezřejmý odborníkovi, který má „běžné dovednosti v příslušném technickém oboru“. Pokud by totiž posuzovatel patentu, který má rovněž „běžné dovednosti v příslušném technickém oboru“, měl k dispozici software pro evoluční návrh (ten by byl např. volně dostupný na internetu), dokázal by s jeho pomocí získat stejný nebo velmi podobný vynález. Tímto způsobem se ale řešení původně nezřejmé odborníkovi, který má „běžné dovednosti v příslušném technickém oboru“, stává zřejmým a dosažitelným pro odborníka, který má „běžné dovednosti v příslušném technickém oboru“ a tudíž není splněna jedna ze základních podmínek pro udělení patentu a žádost je třeba zamítnout.

Plotkin navrhuje, aby se v budoucnu patenty namísto evolučně vytvořeným řešením udělovaly *specifikacím*, tj. postupům, jak sestavit fitness funkci. Pokud je návrhový software s evolučním algoritmem volně dostupný, tak je to právě šikovně formulované zadání (specifikace), které vede ke vzniku nových a patentovatelných řešení. Z patentové přihlášky by ale muselo být zřejmé, že uvedená specifikace vede k patentovatelným vynálezům. Pokud by byla patentována specifikace, znamenalo by to současně ochranu i pro všechny výsledky evolučního návrhu vytvořené podle této specifikace. Patent by se potom vztahoval na celou třídu vynálezů, ne pouze na jeden, jako je tomu doposud. Zdá se, že zavést patentovou ochranu jen pro specifikace je užitečnější než pro celé evoluční návrhové systémy. Volná dostupnost návrhových systémů (nejen v oblasti evolučních algoritmů) se totiž ukázala jako velmi důležitá pro rozvoj inovací.

Doposud platilo, že patentové právo určuje, *kdo vlastní vynález*. Pokud by bylo možné patentovat specifikace, potom by patentové právo vlastně určovalo, *kdo má právo vytvářet vynálezy*. A v tom je velký rozdíl.

7 Závěr

V této kapitole byly objasněny základní principy evolučního návrhu hardware. Ukázali jsme, že existuje podstatný rozdíl mezi evolučním návrhem prováděným pomocí simulátoru a evolučním návrhem probíhajícím přímo v rekonfigurovatelném zařízení. V současnosti existují stovky evolučně navržených řešení, jenž můžeme považovat za konkurenceschopné těm řešením, které byly vytvořeny inženýry-návrháři konvenčním způsobem. Při analýze problému implementace jsme ukázali, že evolučně navržené systémy představují specifickou podtřídu výpočetních zařízení, která vykazují zvláštní kombinaci vlastností, kterou nemají standardní počítače. Z praktického pohledu vzniká problém s patentováním evolučně vytvořených vynálezů, protože stávající interpretace patentového zákona není vyhovující. Vzhledem k tomu, že je evoluční návrh stále více populární, je možné předpokládat, že zasáhne do profesního života řady dalších lidí.

Poděkování: Tato kapitola vznikla za podpory projektů GAČR P103/10/1517 *Natural computing na nekonvenčních platformách* a MŠMT 21630528 *Výzkum informačních technologií z hlediska bezpečnosti*.

Literatura

- [1] Bartels, R. et al.: Learning from Learning Algorithms: Applications to attosecond dynamics of high-harmonic generation. *Physical Review A*. **70**(1):1-5 (2004)
- [2] Bentley, P. (ed): *Evolutionary Design by Computers*. Morgan Kaufmann Publishers, San Francisco CA 1999
- [3] Drechsler, R.: *Evolutionary Algorithms for VLSI CAD*. Boston: Kluwer Academic Publishers, 1998
- [4] Extreme Temperature Electronics - Tutorial (2005)
<http://www.extremetemperatureelectronics.com>
- [5] Harding, S., Miller, J., Rietman, E.: Evolution in materio: Exploiting the physics of materials for computation. *International Journal of Unconventional Computing*. **4**(2):155–194 (2008)
- [6] Higuchi, T. et al.: Evolving Hardware with Genetic Learning: A First Step Towards Building a Darwin Machine. In: *Proc. of the 2nd International Conference on Simulated Adaptive Behaviour*. MIT Press, Cambridge MA 1993, s. 417-424
- [7] Higuchi, T., Liu, Y., Yao, X.: *Evolvable Hardware*. Springer Verlag, 2006
- [8] Human Competitive Awards in Genetic and Evolutionary Computation – Humies 2004–2010. <http://www.genetic-programming.org/hc2005/main.html>
- [9] Hwang, H., Haddad, R.: Adaptive median filters: new algorithms and results. *IEEE Transactions on Image Processing*. **4**(4):499–502 (1995)
- [10] Chan, R.H., Ho, C.W., Nikolova, M.: Salt-and-pepper noise removal by median-type noise detectors and edge-preserving regularization. *IEEE Transactions on Image Processing*. **14** 1479–1485 (2005)
- [11] Johnson, C. G.: What Kinds of Natural Processes Can be Regarded as Computations? In: *Computation in Cells and Tissues: Perspectives and Tools of Thought*. Springer, Berlin 2004, s. 327–336
- [12] Koza et al.: *Genetic Programming IV: Routine Human-Competitive Machine Intelligence*, Springer, 2005
- [13] Kvasnička, V., Pospíchal J., Tiňo P.: *Evolučné algoritmy*. Vydavatelství STU Bratislava, 2000
- [14] Lohn, J. D., Hornby, G. S.: Evolvable hardware: Using evolutionary computation to design and optimize hardware systems. *IEEE Computational Intelligence Magazine* **1**(1): 19–27 (2006)
- [15] Matyska, L.: Softwarové patenty. *Zpravodaj ÚVT MU*. **15**(4): 16-20 (2005)
- [16] Miller, J., Job, D., Vassilev, V.: Principles in the Evolutionary Design of Digital Circuits - Part I. *Genetic Programming and Evolvable Machines*. **1**(1): 8-35 (2000)
- [17] Novák, O. et al.: *Handbook of Electronic Testing*. Vydavatelství ČVUT Praha, 2005

-
- [18] Pečenka, T., Sekanina, L., Kotásek, Z.: Evolution of Synthetic RTL Benchmark Circuits with Predefined Testability. *ACM Transactions on Design Automation of Electronic Systems*. **13**(3):1–21 (2008)
- [19] Piccinini, G.: Computations and Computers in the Sciences of Mind and Brain. PhD thesis, University of Pittsburgh, 2003, 323 s.
- [20] Plotkin, R.: *The Genie in the Machine*. Stanford Law Books, 2009
- [21] Scheutz, M.: When Physical Systems Realize Functions ... *Minds and Machines*. **9**(2): 161-196 (1999)
- [22] Sekanina, L.: *Evolvable Components: From Theory to Hardware Implementations*. Springer Verlag, Berlin 2004
- [23] Sekanina, L.: Evolved Computing Devices and the Implementation Problem. *Minds and Machines* **17**(3): 311-329 (2007)
- [24] Sekanina, L. et al. *Evoluční hardware. Od automatického generování patentovatelných invencí k sebumodifikujícím se strojům*. Academia Praha, 2009
- [25] Stoica, A., Zebulum, R., Keymeulen, D., Ferguson, I., Duong, V., Guo, X.: Evolvable hardware techniques for on-chip automated reconfiguration of programmable devices. *Soft Computing - Spec. Issue on Evolvable Hardware*. **8**(5): 354-65 (2004)
- [26] Stoica, A., Keymeulen, D., Arslan, T., Duong, V., Zebulum, R., Ferguson I., Guo, X.: Circuit Self-Recovery Experiments in Extreme Environments. In: *Proc. of the 2004 NASA/DoD Workshop on Evolvable Hardware*, IEEE Computer Society, Los Alamitos 2004, s. 142-145
- [27] Šonka, M., Hlaváč, V.: *Image Processing, Analysis, and Machine Vision*. CL-Engineering, 3 vyd., 2007
- [28] Thompson, A.: *Hardware Evolution: Automatic Design of Electronic Circuits in Reconfigurable Hardware by Artificial Evolution*. Distinguished Dissertation Series, Springer, London 1998
- [29] Tour, J. M.: *Molecular Electronics*. World Scientific 2003
- [30] Vassilev, V., Job, D., Miller, J.: Towards the Automatic Design of More Efficient Digital Circuits. In *Proc. of the 2nd NASA/DoD Workshop on Evolvable Hardware*, IEEE Computer Society, 2000, s. 151-160
- [31] Vašíček, Z., Sekanina, L.: An Evolvable Hardware System in Xilinx Virtex II ProFPGA. *International Journal of Innovative Computing and Applications*. **1**(1): 63-73 (2007)
- [32] Vašíček, Z., Sekanina, L.: Efficient Hardware Accelerator for Symbolic Regression Problems, In: *5th Doctoral Workshop on Mathematical and Engineering Methods in Computer Science*, Znojmo, CZ, MUNI, 2009, s. 192-199
- [33] Xilinx: Virtex-II Pro Platform FPGAs: Complete Data Sheet (2005) URL: <http://www.xilinx.com/partinfo/ds031.pdf>.

-
- [34] Zebulum, R., Stoica, A., Keymeulen, D., Sekanina, L.: Evolvable Hardware System at Extreme Low Temperatures. *LNCS 3637*, Springer, 2005, s. 37-45
 - [35] Zelinka et al.: *Evoluční výpočetní techniky. Principy a aplikace*. Technická literatura - BEN, 2009
 - [36] Vynálezy pro budoucnost – Úvod do problematiky patentů pro malé a střední podniky. WIPO 2006 <http://www.wipo.int/ebookshop>

Jak a proč vznikají zrakové klamy

Michal ŠIMEČEK a Radovan ŠIKL¹

Abstrakt. Naše představa o vnějším světě, kterou jsme si utvořili na základě vnímání, neodpovídá přímo skutečnosti. Vznikla interpretací senzorických počitků, které vzdálené fyzikální podněty pouze zastupují. To dokládá existence zrakových (resp. percepčních klamů). Těmto jevům se dá rozumět nejen jako nějakým hříčkám, ale dají se na nich dokumentovat i mnohé poznatky o procesu vnímání. Tato kapitola se zabývá zrakovými klamy a jejich teoretickým pozadím, právě ve vztahu k vnímání jako interpretačnímu procesu. Společným základem zrakových klamů je skutečnost, že objekty, které jsou na sítnici oka zobrazeny jako stejné, jsou vnímány jako různé. Naopak je tomu u fenoménu konstantnosti vnímání, kde objekty, zobrazené na sítnici různě, jsou vnímány jako stejné. Kapitola nechce být obrázkovou sbírkou zrakových klamů, ale úvodem do vědy o klamech.

1 Úvod

Než vstoupíme do problematiky zrakových klamů, musíme si nejprve vyjasnit pár pojmů. V anglicky psané literatuře se pro pojem zrakový klam používá označení *optical illusion*. Překlad optická iluze je ovšem zavádějící. Zrakový klam primárně nesouvisí s optikou, není optickým jevem, ale jevem, ke kterému dochází při vnímání. Hranice mezi vnímáním a optikou ovšem není tak ostrá, jak by se mohlo zdát. Například fata morgana je jistě optický jev, který vzniká odrazem světla na rozhraní mezi teplou a studenou vrstvou vzduchu při zemi. Lidé, kteří tento jev nikdy neviděli, snadno podlehnou chybné interpretaci, že jde o vodní hladinu kousek pod horizontem. K jejich omylu ovšem dochází při interpretaci viděného, a ta je součástí procesu vnímání. Stejně tak se nám nebeská tělesa jeví jako stejně vzdálená, jakoby umístěná na „nebeské klenbě“, ačkoli to není pravda. Tato představa se tradovala dlouhá staletí². Řekněme, že podnět zrakového klamu by se na fyzikální rovině neměl lišit od obvyklého zrakového podnětu a k rozdílu by mělo dojít až při procesu vnímání. Pak bude možné říci, že termín zrakového, respektive percepčního klamu se týká odchylky

¹ PsÚ AV ČR, Veverčí 97, 602 00, Brno; e-mail: michal.simecek@volny.cz, sikl@psu-cas.cz.

² O představu sféry hvězd se opíral ještě Tycho de Brahe, když se snažil verifikovat Koperníkovu heliocentrickou teorii a pokoušel se na postavení hvězd měřit paralaxu vyvolanou zemským pohybem kolem Slunce [40].

mezi vnímanou realitou a objektivní nebo fyzikální realitou. Vždycky tomu tak ale být nemusí. Například obrazy nebo fotografie, které jsou v západní kultuře považovány za dvojdimenzionální vyobrazení trojrozměrné reality, a jsou tak i vnímány, nejsou obvykle za klam považovány. Pojem klam by se měl používat tehdy, kdy je vnímaný obraz v neobvyklém nesouladu s vnímanou skutečností.

Pojem klam se vyskytuje v dobré společnosti termínů jako iluze nebo halucinace. Na rozdíl od nich však nemá psychopatologické (psychiatrické) konotace. Halucinace je falešné vnímání neexistujících objektů, které vzniká bez vnější sensorické stimulace. Proti tomu iluzi nazýváme chybnou interpretaci existujícího podnětu. Jak s halucinacemi, tak i s iluzemi se setkáváme u lidí v situaci, kdy jejich percepčně-kognitivní systém nepracuje správně. Mohou být duševně nemocní nebo intoxikováni nějakou drogou, stačí ale, aby byli jen unavení nebo aby se nacházeli těsně před usnutím či chvíli po probuzení (tzv. hypnagogické halucinace). Kdy a jaká halucinace nebo iluze se objeví, závisí na stavu percipienta (resp. halucinanta) a na jeho zvláštětech. Charakter halucinací a do určité míry i iluzí je tedy do značné míry subjektivní. Mimo oblast psychopatologie ovšem existuje řada diskrepancí mezi vnímanou a objektivní skutečností, které vnímá většina lidí podobně a které jsou založeny nikoli na individuálních stavech pozorovatele, ale na charakteristických způsobech zpracování sensorické informace, které jako lidé sdílíme. Těmto případům budeme říkat smyslové klamy.

Smyslové klamy a lidská kognice vůbec, jsou do značné míry ovlivňovány prostředím a kulturou, ve které lidé žijí, a proto ne všechny smyslové klamy jsou všemi lidmi na světě vnímány stejně. Rovněž aktuální podmínky mohou systematicky ovlivnit vnímání některých klamů. Například studie prováděné ve stavu beztlíže ukazují, že se ve stavu beztlíže snižuje citlivost percipientů ke geometrickým klamům [9].

Existence klamů není výsadou zraku. Jako příklad lze uvést klam takzvaných Shepardových tónů. Jde o cyklickou řadu tónů s měnící se výškou i hlasitostí, která vytváří iluzi, že z tónů složená „melodie“ soustavně klesá nebo stoupá, přestože tomu tak ve skutečnosti není.³ Tento klam má svoji obdobu i ve vizuální oblasti. Jiným příkladem mohou být různé klamy vyvolané adaptací smyslových orgánů. Například když necháme nějakou dobu ruku ponořenou ve studené vodě, bude se nám potom naše tvář, když si na ni touto rukou sáhneme, zdát horká. Touto formou adaptace pomocí snižování nebo zvyšování prahu citlivosti smyslového orgánu jsou vybaveny všechny smyslové kanály bez ohledu na modalitu.

1.1 Zrakové klamy

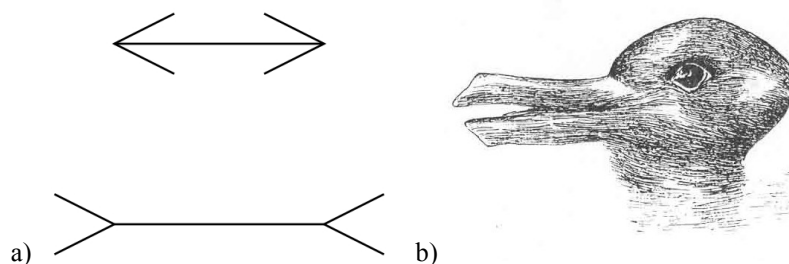
Pochopitelně nejpobulárnější ze všech klamů jsou klamy zrakové. To má jistě řadu příčin. Naše kultura je zaměřena na získávání informací pomocí zraku, a proto se velké množství lidí zabývá vizuální tvorbou – tedy i vytvářením zrakových klamů. Vytváření zrakových klamů se věnuje řada výtvarných umělců. Druhým důvodem může být, že je

³ Na internetu existuje řada vysvětlujících odkazů i ukázek. Povedená je například <http://www.netalive.org/tinkering/shepard-effect/>.

lidský zrak nějak více náchylný k možnosti nechat se oklamat. Tuto možnost prozkoumáme spolu s tématem klamů založených na kognitivních předpokladech ve druhé kapitole.

Všichni lidé tak přinejmenším tuší, co to zrakový klam je, ale celková definice může jen stěží uspokojit různé teoretické pozice, ze kterých se dá ke zrakovým klamům přistupovat. V zásadě se pracuje se dvěma rozdílnými přístupy.

První směr vychází z názoru, podle kterého je klam chyba, kterou je možné připsat jedné dílčí operaci v procesu vnímání. Určitý klam přináší konkrétní chybu percepčního procesu, která vede k nesrovnalosti mezi fyzikálním a vnímaným světem [12]. Tato definice odpovídá přístupu, podle kterého jde při vnímání o co nejvěrnější uchopení světa tak, aby vnitřní představa člověka o vnímaném okolí co nejlépe odpovídala jeho skutečným vlastnostem. Jakákoli odchylka od souladu mezi světem a jeho reprezentací je pokládána za chybu při vnímání. To je případ Müller-Lyerova klamu (obr. 1a): dvě vodorovné úsečky jsou ve skutečnosti stejně dlouhé, ačkoli se pozorovateli jeví horní úsečka jako delší.



Obr. 1. Müller-Lyerův klam a reverzibilní figura

Jiná, fenomenologická, definice je založená na zkušenosti s rozporným vnímáním. Podle této definice je klam percepční zkušenost, která je konfliktní s jinými percepčními zkušenostmi se stejným objektem [70]. Tentokrát jde méně o to, zda a jak se pozorovatel nechá oklamat, ale spíše o to, s jakou intenzitou a na jak dlouho si bude pozorovatel schopen udržet vzniklou kontradikci ve vnímání, jak dlouho bude schopen pozorovat onen rozpor. Na obrázku 1b je takzvaná reverzibilní figura: obrázek, který lze interpretovat jako hlavu zajíce nebo hlavu kačeny. Obojí je možné a k žádné odchylce proti fyzikálnímu světu zde při vnímání nedochází.

1.2 Role zrakových klamů ve výzkumu zrakového vnímání

Zrakové klamy jsou jistě fascinující a inspirativní pro uměleckou tvorbu. Mohou ale mít nějaký význam z vědeckého hlediska nebo se jedná jen o jakési kuriozity lidského vnímání? Mohlo by se na první pohled zdát, že klamy jsou jednoduše projevy nedokonalosti lidského vnímání. To bychom si mohli myslet, kdybychom přijali představu o vnímání jako o pasivním procesu. Například podle Immanuela Kanta je vnímání přímou znalostí externí reality [25]. Zastánci tohoto přístupu nebudou klamy

považovat za předmět hodný výzkumu. Ve zrakových klamech těžko spatří potenciální zdroj relevantních znalostí o samotném procesu vnímání.

Podle převládající představy člověk aktivně vstupuje do procesu vnímání a vnáší do něj své předpoklady o podobě a vlastnostech reálného světa. V takovém případě reprezentují iluzivní vjemy charakteristické chyby vnímání a jako takové mohou vypovídat o fungování percepčního systému. Gillamová [23] upozorňuje, že všechny teorie vnímání se snaží přijít na to, jak zrakový systém získává věrohodné informace o světě. Situace, kdy lidé získávají o okolním světě informace odpovídající skutečnosti, nedokážou mezi jednotlivými teoriemi rozlišovat. Zajímavé jsou právě situace, kdy tomu tak není a lidské vnímání nějakým charakteristickým způsobem selhává. Tehdy je možné zkoumat, zda se stejným způsobem chovají i různé teoretické modely percepce. Právě při výzkumu klamů se mohou naplno projevit specifické rozdíly mezi jednotlivými teoriemi vnímání.

Studium zrakových klamů probíhá v několika krocích. Nejprve je potřeba každý zrakový klam identifikovat a popsat. Je nutné pokusit se získat co nejčistší podobu studovaného klamu, oproštěnou od dalších vlivů. To často nebývá jednoduché. Objekty, které vykazují iluzivní efekt, mívají více vlastností, které se na celkovém dojmu podílí. Tyto různé vlastnosti je potřeba od sebe oddělit. V první fázi bývá formulována předběžná hypotéza o vzniku klamu. Jejím smyslem je dostatečně zobecnění iluzivního účinku a jeho fenomenologický popis. Na základě těchto prvotních formulací je možné pokusit se generovat různé varianty daného klamu.

V dalším stádiu zkoumání percepčního klamu se cesty rozdělují. Psychologie nebo evoluční biologie se bude zabývat spíše otázkou „Proč?“. Bude se zajímat o to, jakou funkci plní mechanismy zodpovědné za vznik klamu v normální percepci. Naopak fyziologicky orientovaný výzkum si bude spíše klást otázku „Jak?“. Bude se věnovat tomu, jak fungují percepční mechanismy, které jsou za vznik klamu zodpovědné [71].

1.3 Klasifikace zrakových klamů

V literatuře nalezneme mnoho dělení zrakových klamů. Některá vychází z charakteristických vlastností podnětového materiálu, jiná jsou založena spíše na klasifikaci předpokládaných procesů, které jsou za vznik iluze zodpovědné. Uplatňují se dělení založená na předpokládané úrovni zrakového systému, ve které ke vzniku iluzivního účinku dochází. Většina dělení ale představuje kombinaci obojího. Snad nejstarší skupinou klamů, která je vnímána jako konsistentní a za kterou se předpokládá specifická teorie, jsou takzvané geometrické klamy. Geometrické klamy jsou nejčastěji chápány jako důsledek aplikace prostorových předpokladů a od ostatních zrakových klamů se tím liší.

Například Changizi a kol. [38] klasifikovali 28 geometrických zrakových klamů podle kontextových parametrů, které jsou za klam zodpovědné (velikost, rychlost, světelný kontrast, vzdálenost, rozdíl, úběžník a optický tok) a podle vlastnosti, v rámci které dochází k nesouladu mezi vjemem a skutečností (velikost, rychlost, světelný kontrast, vzdálenost).

Různá dělení se ovšem mohou lišit již podle toho, co kteří autoři za zrakový klam vůbec považují. Například definice, které chápou klam jako chybu vnímání, by stěží mezi klamy zahrnuly fenomény jako je reverzibilní figura (obr. 1b) nebo anamorf. Perspektivní kresby typu *trompe l'œil* by při monokulárním vidění byly spíše případy percepční disonance než percepčního klamu [71].

V posledním půlstoletí, pravděpodobně též pod vlivem rozvoje neurologických výzkumů percepce, se ujímá dělení na klamy, založené na konkrétních neurologických mechanismech, které je způsobují, a na klamy založené na kognitivních předpokladech. V tomto smyslu například Gregory hovoří o klamech založených na percepčních mechanismech a na kognitivních strategiích. Klamy způsobované kognitivními strategiemi jsou způsobovány předpoklady o vnímaném světě, které kognitivní systém vnáší do procesu vnímání. V tomto případě se jedná o sestupnou (*top-down*) modifikaci procesu vnímání. Naopak fyziologicky podmíněné klamy jsou založeny na procesech, ke kterým dochází při zpracování sensorického signálu od jeho zachycení na sítnici směrem ke vzniku vjemu. Jde o vzestupný (*bottom-up*) proces. Gregory upozorňuje, že ačkoli rozdíl mezi těmito druhy klamů je esenciální, v praxi nemusí být jednoduché oba typy od sebe odlišit [31].

Kitaoka [43] rozděluje zrakové klamy na kognitivní a percepční. Kognitivní klamy dělí dále na vyšší (nepochopení vnímaného obrazu) a nižší úroveň (například kouzelnické iluze). Percepční klamy, které částečně odpovídají fyziologickým klamům podle jiných dělení, dále dělí na vyšší, střední a nižší. Mezi vyšší percepční klamy Kitaoka zařazuje například reverzibilní a nemožné figury. Do střední úrovně percepčních klamů řadí ty, které jsou založené na binokulárním vidění nebo klamy vzniklé vizuálním uzavřením figur (např. Kanizův trojúhelník – obr. 6a). Nejnižší úroveň percepčních klamů tvoří pohybové klamy a paobrazy.

Jinde v literatuře nalezneme kromě rozdělení na fyziologické a kognitivní klamy též kategorii umělých (uměle vytvořených) klamů. Patří sem jak Amesův pokoj, tak Hermanova mřížka. Je zde patrný poukaz na určitou nepřirozenost a umělou rafinovanost některých klamů.

Gregory [31] předkládá komprehenzivní dvoudimenzionální klasifikaci zrakových klamů, která zahrnuje i čistě optické jevy nebo naopak patologické stavy a další fenomény. První dimenze jeho klasifikace obsahuje čtyři úrovně zpracování podnětu, v nichž může klam vznikat. První, receptivní úroveň se týká potíží při přijetí podnětu receptorem, jeho převodu do nervových vzruchů a podstoupení vyšším úrovním zpracování. Na druhé úrovni dochází k problémům při zpracování signálu v nervových drahách a mozkových centrech. Klamy třetí úrovně jsou způsobeny chybným užitím znalostí, které moderují proces vnímání. Poslední úroveň se týká samotného porozumění viděnému. Druhou dimenzi Gregoryho klasifikace představují fenomenologické důsledky iluzivní percepce. Gregory rozlišuje absenci vjemu, víceznačnost, nestabilitu vjemu, distorzi vjemu a fiktivní a paradoxní vjem.

V tomto textu se dále soustředíme na teorie, které jednotlivé zrakové klamy vysvětlují, a pokusíme se propojit fyziologický a kognitivistický přístup ke zrakovým klamům. V poslední kapitole se budeme věnovat klamům jako výrazovým prostředkům lidské komunikace a jejich schopnosti nést význam. Téma zrakových klamů je neobyčejně široké a literatura je nesmírně bohatá. Všechna dílčí témata není

možné ani jen zmínit, natož zevrubně představit. V tomto textu se proto nebudeme zabývat například barevným, respektive barvou podmíněným klamům. Tyto se v černobílé publikaci nakonec špatně představují. Rovněž se nebudeme věnovat klamům, které jsou založené na pohybu nebo zdánlivém pohybu. Stranou necháme i jevy, které pracují s binokulárním viděním, jako jsou stereoskopické obrazy nebo autostereogramy (známé jako *magic pictures*) a dalšími fenomény, které představují svébytná a dosti rozsáhlá témata, která se oblasti zrakových klamů také dotýkají.

2 Kognitivní teorie zrakových klamů

Kognitivní teorie zrakových klamů jsou založeny na předpokladech, které náš kognitivní aparát používá při zpracování zrakových informací, aby dokázal vizuální podnět zpracovat do pro něj smysluplného tvaru. Přitom platí, že jako ostatní živočichové se i my lidé potřebujeme orientovat v prostředí, které má podstatnou hloubkovou dimenzi. I když máme k dispozici binokulární vidění, jsou naše „přímé“ zrakové informace o trojrozměrném uspořádání našeho okolí značně omezeny. Abychom vůbec mohli získat představu o prostorovém uspořádání toho, co vidíme, musíme přijmout řadu předpokladů a zjednodušení, která za určitých okolností mohou vést k chybné percepci. Některé z těchto chyb můžeme nazvat zrakovými klamy.

Předpokladům, které vstupují do procesu vnímání a interpretace vnímaného, se věnují zejména ti výzkumníci, kteří preferují konstruktivistický pohled na celou problematiku, zatímco stoupenci Gibsonovy teorie přímého vnímání se těmito detaily nezabývají [18]. Konstruktivistický přístup nahlíží na vnímání jako na proces zpracování informace [1], [2], [18], při kterém se ze sensorického signálu vytváří znalost. Z významově nediferencovaných sensorických dat vznikají znalosti právě aplikací předpokladů, které vnímajícímu dávají smysl, a které tak významem opatří i původní sensorická data. Vnímající se, podle konstruktivistického modelu, aplikací předpokladů vystavuje riziku omylu, ke kterému může za určitých okolností v rámci procesu interpretace dojít. Tento omyl je však něco jiného než důsledek špatného fungování percepčního systému. Zrakové klamy jsou v rámci konstruktivistického přístupu chápány jako doklady aplikace předpokladů a zároveň ukazují na určitou zranitelnost percepčně-kognitivního systému, která z toho vyplývá. Ačkoli různých předpokladů, které by mohl člověk při zrakovém vnímání potřebovat a využívat, by se dalo najít celá řada, v pozadí jsou vždy dva základní principy, od kterých se premisy lidského zrakového vnímání teoreticky odvíjí. Jsou to předpoklad jednoduchosti a obvyklosti.

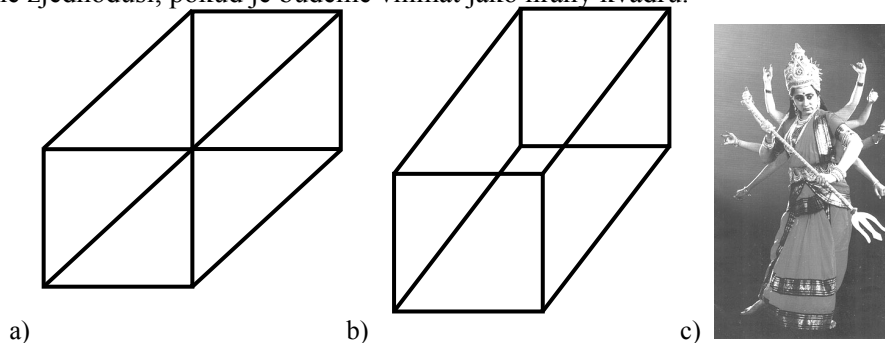
2.1 Předpoklad jednoduchosti a obvyklosti

Podle principu jednoduchosti (*simplicity principle*) je nejednoznačná scéna interpretována tím nejjednodušším způsobem, který přichází v úvahu, vzhledem k její vizuální presentaci. Princip jednoduchosti vychází z gestaltistického zákona dobré figury [46]. V zásadě jde o to, dokázat popsat viděnou scénu co nejjednodušeji, ve

smyslu co nejmenšího počtu operací [59] nebo co nejkratšího popisu. O tom hovoří teorie primitivního kódování [8].

Stejně jako prostředí kolem nás vykazuje určité vlastnosti, které obvykle (dle naší zkušenosti nebo statisticky) platí, též objekty v tomto našem prostředí mají obvykle určité charakteristické vlastnosti, na kterých lze založit vcelku účinné předpoklady. Znalost prostředí a věcí kolem nás je důležitá pro vnímání, protože ze surových sensorických stimulů pomáhá vytvářet smysluplnou zkušenost. Náš zrak je adaptován na vnímání takových objektů, které obvykle ve svém prostředí potkáváme a jejichž správné vnímání je pro naše přežití důležité. Tomuto principu se někdy říká princip obvyklosti (*likelihood principle*) [33]. Typickým případem takového předpokladu, který je generalizací určité zkušenosti s předměty, je předpoklad vztahu mezi velikostí a váhou. Větší objekty máme tendenci vnímat zároveň jako těžší [27].

Již na úrovni rozlišení figury a pozadí mohou vstupovat na scénu první prostorové předpoklady. Například dvojrozměrné obrazy některých jednoduchých třírozměrných objektů, které se promítají na plochu sítnice, mohou být složitější pro popis, než původní prostorové objekty. Příkladem principu jednoduchosti může být obrázek 2b, který vyvolá dojem drátěného modelu kvádrů, zato obrázek 2a se někomu jeví spíše jako plošný čárový útvar. Objekt na obrázku 2a je jednoduchý i ve své plošné podobě. Je tvořen vlastně jen devíti čarami, zatímco objekt 2b tvoří při nejmenším 12 čar s podstatně komplikovanějším rozmístěním [33]. Organizace těchto čar se ovšem značně zjednoduší, pokud je budeme vnímat jako hrany kvádrů.



Obr. 2. Situace, kdy zvítězí jednoduchost interpretace: dvě zobrazení kvádrů a tanečnice v zákrytu jako bohyně Durga.

Princip jednoduchosti se prolíná s principem obvyklosti, a to do té míry, do jaké je pro naše prostředí obvyklý výskyt spíše jednoduchých objektů. Často se však tyto principy doplňují. Například změnu jasu nějakého povrchu, která indikuje prohlubeň nebo výčnělek, budeme interpretovat podle toho, odkud obvykle přichází světlo v našem prostředí – tedy shora. Neexistuje přitom žádný důvod, proč by to zrovna tak mělo být v rámci teorie jednoduchosti [33]. Někdy naopak při vzájemném konfliktu „zvítězí“ princip jednoduchosti nad principem obvyklosti. Leeuwenberg a Boselie uvádějí kresbu dvojhlavého koně. Tato kresba by mohla být interpretována jako dva koně v zákrytu, nicméně člověk si většinou volí raději jednodušší, ač bizarnější

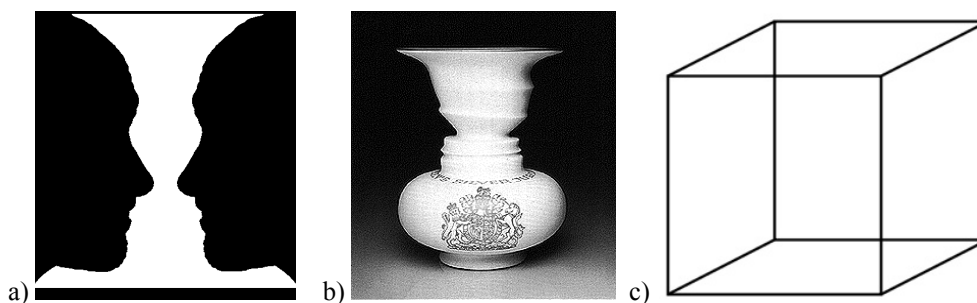
vysvětlení [33]. Na obrázku 2c vidíme situaci několika tanečnic v zákrytu. Scénu ale můžeme interpretovat i jako indickou mnohorukou bohyni Durgu. Vtip je v tom, že pokud při pokusu o co nejsnazší interpretaci scény narazíme na reprezentaci tohoto indického božstva, použijeme tuto reprezentaci k interpretaci výjevu.

Když Galileo Galilei namířil na oblohu první astronomický dalekohled, mohl se setkat s objekty, jejichž tvary na obloze neočekával. Měl zkušenost pouze s kotouči Slunce, Měsíce nebo planet, a proto interpretoval Saturnovy prstence jako dva měsíce v blízkosti planety [31]. Když se výzkumníci všeho druhu (nejen vědci, ale třeba i cestovatelé) setkávají a setkávají se skutečnostmi, které se zásadně vymykají jejich zkušenosti, mají tendenci si nové zkušenosti přizpůsobovat starým poznatkům, a to někdy i za cenu pokřivení reality. Naše kognitivní zkostnatělost tak brání příchodu revolucí v poznání [47].

Víceznačnost

Vizuální podněty jsou pro interpretujícího pozorovatele z principu víceznačné. Je to dáno již redukcí trojrozměrné informace na pouhé dva rozměry sítnicového obrazu. Tato redukce je jen z části kompenzována binokularitou a jinými prostorovými nápověďmi. Lidé ovšem mají zkušenost spíše s jednoznačným porozuměním toho, co vidí. Aby dosáhli jednoznačnosti, musí zapojit nejrůznější předpoklady, které omezují možné interpretace viděného a vedou k jedinému výsledku.

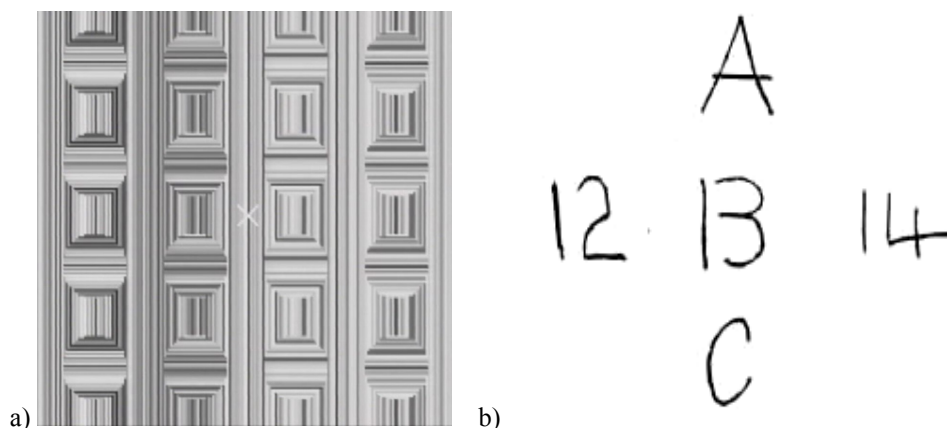
Při interpretování vizuálního podnětu velmi často záleží na kontextu. Pokud totiž bude kontext příliš chudý nebo nejasný, nemusí být člověk schopen rozhodnout, jakou interpretaci zvolit. Zůstane k dispozici víc stejně „dobrých“ možností vysvětlení viděného a všechny dosáhnou vědomí. Dánský psycholog Edgar Rubin (1886-1951) se jako první systematicky zabýval tímto fenoménem. Od něj pochází takzvaná reverzibilní figura (obr. 3a). Rubin zjistil, že v každém okamžiku může být ve vědomí pozorovatele vyjádřena pouze jediná z možností, které přicházejí do úvahy. Při vnímání se tyto možnosti ve vědomí vnímajícího dynamicky překlápají.



Obr. 3. Rubinova reverzibilní figura a fotografie vázy, Neckerova krychle.

Všimněte si, že každá z možných interpretací této figury je závislá na jiném ze dvou zmíněných principů. Vjem vázy odpovídá více principu jednoduchosti podle gestaltistických zákonů (je uzavřená, symetrická a jde jen o jeden objekt), zatímco interpretace dvou tváří je založena na principu obvyklosti, přičemž využívá naši

tendenci vidět tváře ve všem, co by je jen trochu mohlo připomínat. Pro Neckerovu krychli⁴ (obr. 3c) platí, že ji snáze budeme interpretovat jakoby shora, čelní stranou otočenou doleva. Je to tak proto, že kvádrové objekty vidáme spíše ležet na zemi, než vznášet se nám nad hlavami.



Obr. 4. Pokladnicový klam⁵ a Brunerův a Minturnův experiment

Pro některé interpretace vnímaného obrazu máme tedy větší „pochopení“ než pro jiné. Někdy naše interpretace vychází ze schopnosti vidět figury ohraničené čarami spíše než určené změnou povrchu. Tak je tomu například u pokladnicového klamu, který se dostal v roce 2006 do finále soutěže *Best Visual Illusion of the Year*. Úkolem je říci, kolik je na obrázku 4a kruhů⁶.

Gregory předpokládá, že víceznačnost vzniká, pokud se ztratí podstatná informace a vzroste nejistota [27]. Aby to tak mohlo být, museli bychom předpokládat, že zraková percepce za normálních okolností dostává dostatek informací pro generování jediného (odpovídajícího) řešení. Předpoklad, že jediné správné řešení existuje, nemusí být splněn přinejmenším u uměle vytvořených objektů, jako jsou právě reverzibilní figury.

Kromě obecných očekávání hraje při volbě interpretace roli také momentální nastavení pozorovatele. Například Bruner a Minturn [6] provedli experiment, ve kterém byla víceznačná figura (písmeno „B“ nebo číslice „13“) administrována jako součást řady písmen nebo číslic. Odpověď zkoumaných osob se řídila podle kontextu (srovnej obrázek 4b). Podobně jako v tomto experimentu platí, že kdyby se obrázek 3b nacházel v katalogu keramických výrobků, tváře by v něm nikdo nehledal.

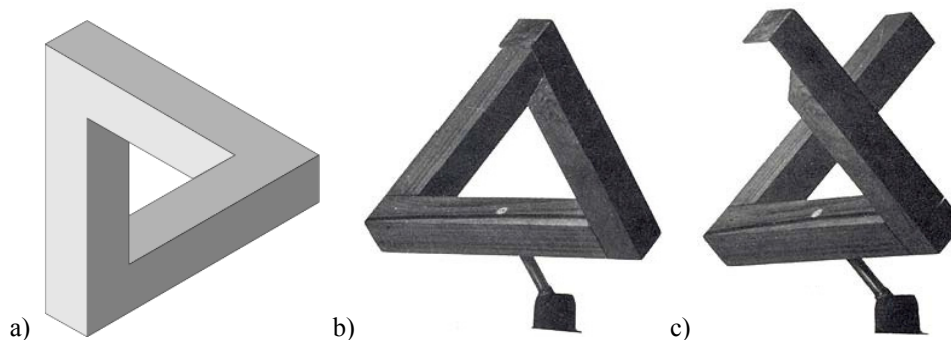
⁴ L. A. Necker byl švédský mineralog, který reverzibilitu této figury objevil poté, co se díval na vyobrazení krychlové krystalové struktury [1].

⁵ Best Visual Illusion of the Year <http://illusioncontest.neuralcorrelate.com/>

⁶ Kruhy na obrázku mají povrch tvořený vodorovnými čarami různé světlosti a jsou v matici 4x4.

Vnímání nemožných objektů a nekonzistentních scén

Těžko říci, jestli objeviteli, ale rozhodně průkopníky, ve vymýšlení nemožných objektů byli grafici M. C. Escher (1898 – 1972) a Oscar Reutersvärd (1915 - 2002). Pokud jde o nekonzistentní scény, proslavil tento výtvarný motiv hlavně malíř Rene Magritte (1898 - 1967).



Obr. 5. Penroseův trojúhelník a jedno z jeho možných trojrozměrných provedení [30].

Jednotlivé části vyobrazení nemožných objektů a nekonzistentních scén dávají jasný smysl, ale v celku jsou nekonzistentní. Při letmém pohledu nespátří pozorovatel nic divného. Teprve podrobnější analýza toho, co vidí, mu ukáže, že předkládaný objekt nebo scéna jsou nesmyslné. Existence tohoto klamu ukazuje na to, že při interpretaci celých objektů nebo scén vycházíme z jednotlivých detailů, které skládáme dohromady. Jako příklad nám poslouží útvar známý jako Penroseův trojúhelník (obr. 5a), který by se ovšem měl spíše nazývat Reutersvärdův, protože švédský grafik Oscar Reutersvärd s jeho vyobrazením přišel asi o 20 let dříve než Roger Penrose. Jde o dvojdimenzionální znázornění zdánlivě trojdimenzionálního trojúhelníku. Každý z jeho vrcholů je věrohodným způsobem znázorněn a každé toto znázornění je konzistentní se zobrazením sousedního vrcholu. Dohromady ale rozvržení hran nedává smysl. Při analýze scény se orientujeme podle zobrazení vrcholů zvlášť a nejprve identifikujeme trojdimenzionální objekt ve tvaru trojúhelníku. Obrázky umožňují zdánlivě dvě nebo více interpretací, ale ve skutečnosti ani jedna z nich není plně konzistentní.

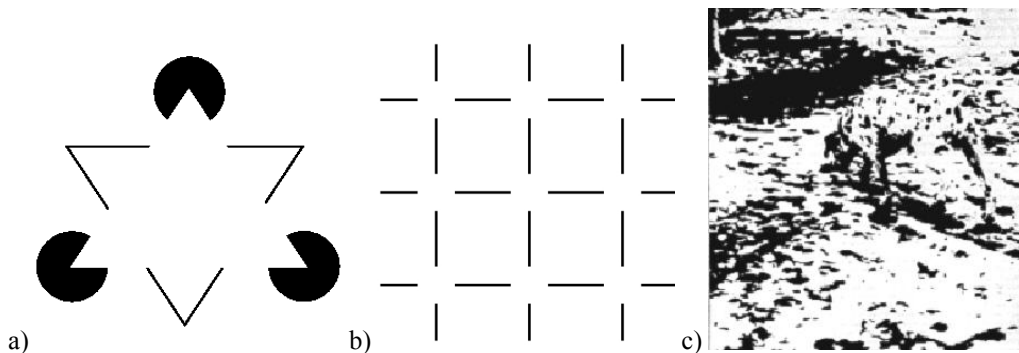
Jev nekonzistence není ovšem vázán pouze na nakreslené objekty. Hochberg [33], který tomuto jevu říká penroseovský fenomén, jej očekává všude tam, kde je objekt ve skutečnosti složitější, než odpovídá očekávání pozorovatele. Existují například pokusy vytvořit penroseovský trojúhelník v realitě. Na obrázku 5b je skutečný objekt, ovšem vyfotografovaný ze „správného“ úhlu. Z jakéhokoli jiného úhlu (obr. 5c) se ukáže jeho skutečný tvar. V reálném prostoru je tedy možné vytvořit objekty, které budou vypadat stejně jako zmíněné nekonzistentní objekty. Pro ně ovšem platí, že fungují pouze tehdy, dívá-li se na ně pozorovatel z určitého přesně daného místa a pokud je mu zabráněno v korekci pomocí prostorových nápovědí (např. binokulární disparity). Všechny tyto objekty jsou ve skutečnosti prostorově složitější, než jejich nemožná

předloha⁷. Na obrázcích si můžeme rovněž povšimnout jedné věci: vrcholy trojúhelníku na obr. 5a jsou zobrazeny tak, že vyvolávají dojem pravých úhlů. To jistě usnadňuje zvolit trojrozměrnou interpretaci. Trojúhelník se všemi úhly pravými nemůže (alespoň v nezakřiveném eukleidovském prostoru) existovat. To je právě vlastnost, která na nemožném objektu zaujala teoretického fyzika Penrose. Na obrázku 5b a 5c vidíme 3D objekt, který skutečně má pravé úhly ve vrcholech. O trojúhelník se ovšem nejedná.

Kulturní antropolog Jan Deregowski [13] zjistil, že lidé, kteří nemají sklon vnímat plošné útvary jako vyobrazení trojrozměrných objektů, dokážou snáze nakreslit předloženou kresbu neexistujícího objektu. Zatímco ti, kdo mají tendenci kresbu vnímat jako vyobrazení trojrozměrných objektů, jsou zmatení a obkreslení jim dělá potíže. Ve snaze popsat scénu co možná nejjednodušším způsobem se pouštíme do její trojdimenzionální interpretace, k čemuž nám obrázky absurdních objektů a scén lokálně nabízí řadu vodítek k trojrozměrné interpretaci. Tento pokus ovšem nakonec musí selhat, protože obraz jako celek pro nás není v trojrozměrné interpretaci konzistentní.

Vnímání neexistujících objektů

Kromě toho, že lidský zrak může podlehnout nekonzistentnímu vjemu a při troše nepozornosti označit Penroseův trojúhelník za „nějaký divný prostorový trojúhelník“, může rovněž spatřit objekt tam, kde žádný není. Nemáme však na mysli duhu, fata morgana a podobné optické jevy. Princip jednoduchosti například vede zrakový systém k předpokladu, že některé složitější vjemy mohly vzniknout vzájemným překrytím několika jednodušších objektů. Jednoduchá plastická interpretace se dosadí pro vysvětlení složitého plošného útvaru. To je případ Kanizova trojúhelníku (obr. 6a).



Obr. 6. Kanizův trojúhelník, Ehrensteinova figura a skrytá figura.

Existenci bílého trojúhelníku uprostřed obrazce naznačují výseče v černých kruzích, které trojúhelník „překrývá“ a „překrytím“ střední části druhého trojúhelníku.

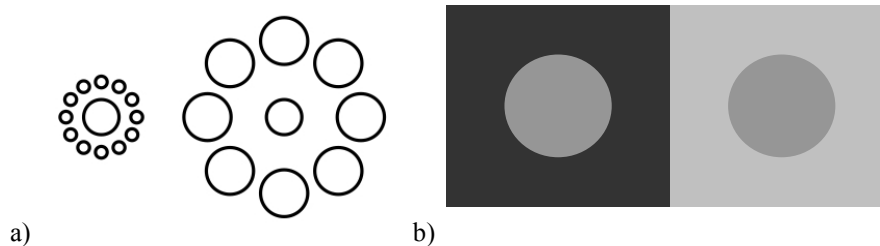
⁷ Jednu z trojrozměrných variant Penroseova trojúhelníku naleznete na adrese <http://www.coolopticalillusions.com/build-an-impossible-triangle.htm>.

Stejný jev se objeví i v ještě jednodušším případě Ehrensteinovy figury z roku 1941 (obr. 6b), kde můžeme spatřit bílé kruhy „zakrývající“ křížení černých linií. Proč vidíme právě kruhy a ne třeba čtverce, když je možné obojí? Náš zrak patrně považuje kruhy za jednodušší útvary, což dává smysl i z geometrického hlediska. Kitaoka v souladu s Gestalt psychologií hovoří v případě těchto klamů o vizuálním uzavření figury a nabízí další (barvové) fenomény [42].

Kdo nezná obrázek 6c, spatří černobíle skvrnitou plochu, která mu nebude dávat žádný význam. Když se ale dozví, že na obrázku je dalmatin, po chvíli bude schopen jej spatřit. Rozezná jasně jeho obrysy a uvidí plasticky povrch psího těla. Všechny tyto jevy jsou důsledkem dodatečné verbální informace. To ukazuje na ohromnou schopnost lidského zraku organizovat zdánlivě chaotické obrazové signály do smysluplných celků.

Závislost na kontextu

Objekty, které identifikujeme na pozadí vnímané scény, nevnímáme samostatně, ale jejich vlastnosti odhadujeme v závislosti na kontextu, ve kterém se nachází. Vnímané vlastnosti objektu, které nám jej umožňují vnímat jako figuru na pozadí, jsou hodnoceny relativně vzhledem k vlastnostem pozadí. Abychom dokázali figuru od pozadí ostře odlišit, jsou právě tyto vlastnosti vnímány vzhledem k pozadí co nejvíc kontrastně. V případě Ebbinghausova klamu (obr. 7a) je kruh uprostřed menších kruhů vnímán jako větší, než stejně velký kruh uprostřed větších kruhů. Na stejném principu funguje klam světelného kontrastu (obr. 7b), kde oba kruhy mají ve skutečnosti stejný odstín šedi.



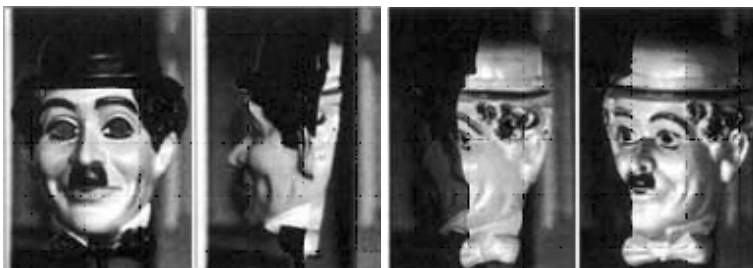
Obr. 7. Ebbinghausův klam a klam světelného kontrastu.

Jeden z nejstarších klamů, které byly kdy zkoumány, je takzvaný měsíční klam. Měsíc, který se zdá být vždy stejně veliký, pokud je vysoko na obloze, se naopak zdá lidem větší, když se nachází blízko horizontu. Všiml si toho již ve druhém století před naším letopočtem astronom Ptolemaios, který věděl, že ať už se Měsíc nachází vysoko na obloze nebo je nad obzorem, má stejnou úhlovou velikost [26]. Jedno z vysvětlení pracuje se vztahem objektu k jeho kontextu a spočívá v tom, že na obloze, kde nemáme srovnání s jinými objekty, se nám Měsíc jeví relativně malý. Ovšem nad obzorem, kde je obklopen objekty, o kterých víme, že jsou celkem velké (domy, stromy...), se nám Měsíc jeví rovněž velký. Vidíme, že vnímaná velikost objektu je v relaci s tím, jaké máme představy o velikosti objektů v jeho okolí.

Podíváme-li se na svatební fotografii lidí u vchodu do kostela, zdají se nám všichni svatebčané překvapivě malí. Je to proto, že kostelní vchod bývá obvykle velikostně předimenzován, což zpětně „zmenšuje“ ostatní objekty známé velikosti – svatebčany v tomto případě.

Tvářové klamy

Lidé (jako sociální tvorové) jsou přednostně nastaveni vnímat lidské tváře. Lidé jsou skutečnými odborníky na lidské tváře a řada dovedností, které s tím souvisí (rozpoznávání, zapamatování, schopnost odečítat emoce atd.), je vykoupena určitou percepční rigiditou. Jejím dokladem je například to, že člověk má tendenci vnímat lidskou tvář jako konvexní objekt (jako skutečnou lidskou tvář s tvářemi a nosem vystrčenými ven), a to i když jde ve skutečnosti o negativní odlitek tváře [18], [28].



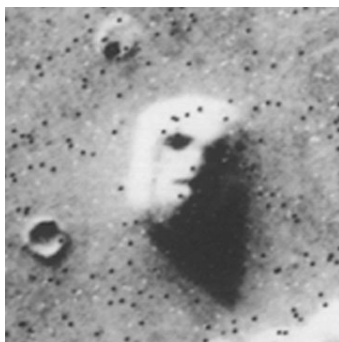
Obr. 8. Chaplinova maska v různých fázích jejího otočení [28].

Zároveň je také pro člověka obtížné registrovat různé neobvyklé deformace, provedené na obrazu tváře. Například populární klam Thatcherové (obr. 9a) ukazuje, že jsme schopni bez obtíží identifikovat podobu známé osoby, i když podobiznu vidíme vzhůru nohama. A zároveň přehlédneme významné tvarové úpravy. Podobně neznámá dívčí tvář (obr. 9b) se na první pohled jeví „divná“ a zrakové vnímání se doslova zasekne v nekonečné smyčce, ve snaze přijít té podivnosti na kloub. Tento tzv. *jazzing-effect* je nejmarkantnější při zdvojení očí [31].



Obr. 9. Klam Thatcherové a tvářový *jazzing-effect*.

Různých klamů a jiných efektů je na tvářích ukazováno hodně. Jako další příklad lze uvést zrcadlové tváře, které mění naše hodnocení charakteru i pohlaví takto zobrazované osoby nebo různé efekty pracující s rozostřením tváří a s přechody jedné tváře do druhé. Na druhou stranu ale schopnost identifikovat tváře rapidně klesá, pokud jsou podmínky uměle ztíženy (fotografie obličejů jsou rozřezané nebo jsou prezentované jako negativ).



Obr. 10. Skrytá figura a domnělá tvář na povrchu Marsu

Připravenost identifikovat objekt jako lidskou tvář může někdy směřovat až k blamáži. To se stalo v případě domnělé tváře na Marsu (obr. 10). Fotografie pořídila z oběžné dráhy Marsu orbitální část sondy Viking 1 v roce 1976 a zachycuje útvar, který má ve skutečnosti zcela jiný tvar.

2.2 Teorie založené na předpokladu existence scény

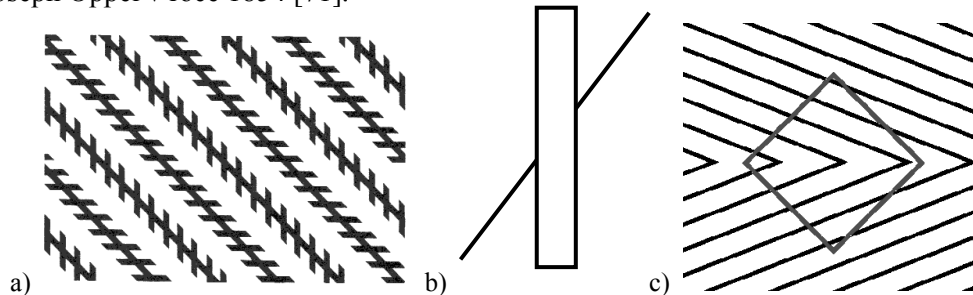
Ačkoli historie zkoumání zrakových klamů sahá hluboko do minulosti, jeden z důležitých kroků udělal německý středoškolský učitel fyziky Johann Joseph Oppel (1815–1894), který upozornil na zajímavost zrakových klamů pro vědecký výzkum. Oppel položil základy scénové (*scene-based*) teorie geometrických klamů [71]. Tuto teorii dnes rozvíjí zejména Richard L. Gregory a Barbara Gillamová.

Jak již vyplývá z jejího názvu, vychází tato teorie z předpokladu, že náš zrakový systém se při vnímání obrázku pokouší číst dvojdimenzionální figuru jako plošné zobrazení trojdimenzionální scény.

Předpoklad pravých úhlů

S předpokladem pravých úhlů jsme se již setkali, když jsme popisovali vnímání figur na obrázku 2b a 3c jako prostorových objektů, a také, když jsme vysvětlovali iluzivní účinek Penroseova trojúhelníku (obr. 5), kde předpoklad pravých úhlů vede k nekonzistentní interpretaci obrázku. Lidský zrak, a to přinejmenším zrak člověka západního kulturního okruhu, je připraven vnímat hrany objektů jako pravoúhlé.

Velká skupina takzvaných geometrických klamů bývá vysvětlována pomocí teorie *expanze ostrých úhlů* (*acute angle expansion*). Ačkoli tuto teorii dnes adoptovaly fyziologické přístupy, je její historie mnohem starší. Tento fenomén popsal již zmíněný Joseph Oppel v roce 1854 [71].



Obr. 11. Zöllnerův, Poggendorffův a Orbisonův klam

Klam na obrázku 11a publikoval roku 1860 astronom Karl Zöllner. Zöllnerův klam vzniká u linií, které jsou přetínány šikmými pruhy. Konfigurace zdánlivě mění orientaci dlouhých linií. Tyto linie, které jsou ve skutečnosti rovnoběžné, se jeví, jako by každá z nich mířila poněkud jiným směrem, a to v závislosti na směru čar, které je kříží. Konfigurace u pozorovatele navozuje chybný dojem, jakoby linie byly opatřeny pruhy kolmými na svoji celkovou orientaci. Vzniká tak prostorový dojem.

Teorie expanze ostrých úhlů předpokládá, že tyto ostré úhly jsou iluzorně zvětšovány. Zvětšení ostrých úhlů vede k iluzivnímu vykřivení dlouhých linií Zöllnerova klamu. Podle teorie vnímání scény je důvod, proč jsou všechny ostré úhly paušálně zvětšovány (směrem k pravému úhlu) tento: lidé předpokládají, že většina rohů, které se jim v oku promítají v ostrém úhlu, je ve skutečnosti pravoúhlých. Tyto pravoúhlé rohy jsou při projekci na oční sítnici perspektivně zmenšeny. Lidé předpokládají ve svém prostředí hojný výskyt pravých úhlů [33], proto ostré úhly nevědomky zvětšují směrem k úhlům pravým. Proti tomu však tupé úhly, podle teorie, pozorovatelé nemění. Taková interpretace má ovšem za důsledek, že výsledný vjem se již nemůže „vejít“ do dvou rozměrů – součet úhlů přesáhne 360° . To by mohlo vysvětlovat, proč se figura Zöllnerova klamu jeví jako plastická a jakoby vystupuje z obrazu.

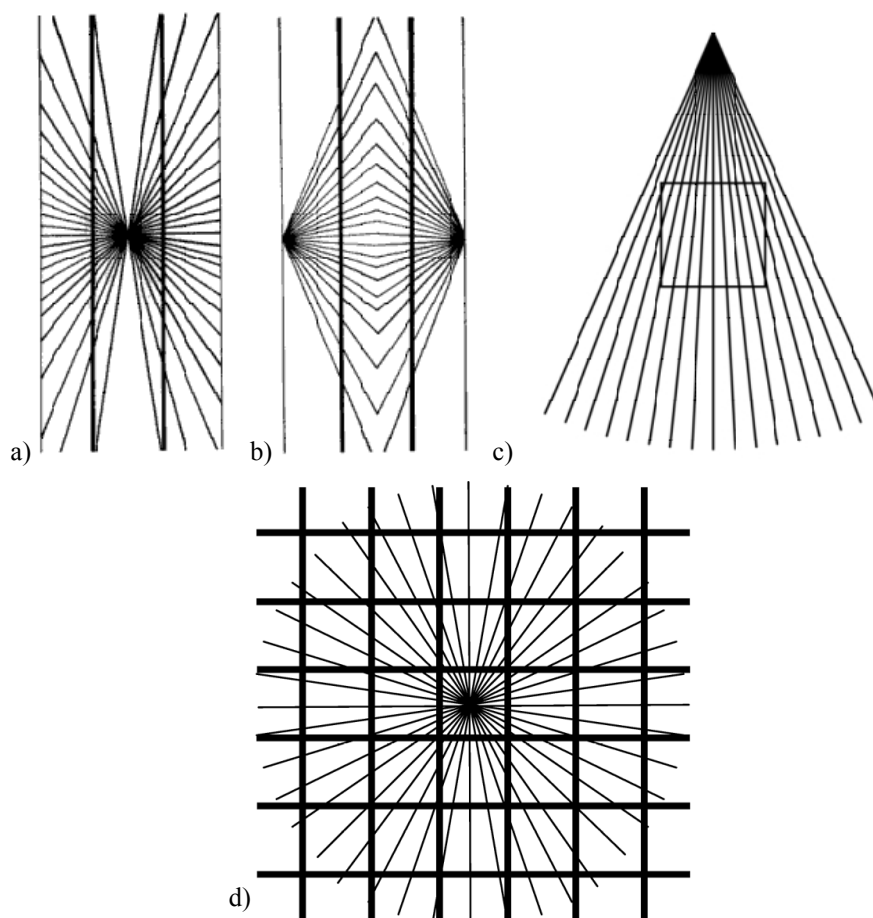
Stejně vysvětlení lze uplatnit rovněž v případě Poggendorffova klamu. Tento klam (obr. 11b), zkonstruoval J. C. Poggendorff v roce 1860 jako odpověď na klam Zöllnerův. Svým způsobem je Poggendorffova figura minimalistickou variantou té Zöllnerovy, ovšem účinek je zde jiný [64]. Efekt Poggendorffova klamu spočívá v tom, že skloněná linie, která je překryta bílým obdélníkem, se jeví přerušená a posunutá. Oba konce se jeví vůči sobě posunuté a jakoby netvořily jednu čáru [60].

Stejným způsobem jako Zöllnerův a Poggendorffův klam jsou vysvětlovány i další klamy, jako je klam Orbisonův (obr. 11c), Heringův (obr. 12a), Wundtův (obr. 12b) nebo Ehrensteinův (obr. 12c). Heringův klam (obr. 12a) je zrakový klam zkonstruovaný německým fyziologem Ewaldem Heringem (1834-1918) v roce 1861. Dvě vertikální linky se jeví jako veprostřed ohnuté směrem ven. S analogickým klamem (obr. 12b) přišel Wilhelm Wundt (1832-1920). V případě jeho klamu se vertikální linky jeví jako ohnuté dovnitř. Wundtův a Heringův klam působí obdobně jako Zöllnerův klam, ovšem zároveň využívají skutečnosti, že míra iluzivní distorze se mění s měnícím se úhlem křížení čar. Čím více se křížení vzdalují od pravého úhlu, tím větší je distorze směru dlouhé linie. To platí ale jen do určité míry. U velmi ostrých úhlů přestává expanze ostrých úhlů fungovat. Orbisonův klam (obr. 11c) využívá deformace čtverce vlivem rovnoběžných a lomících se čar v pozadí. U obrázku Orbisonova klamu si můžete všimnout, že i jeho obdélníkový okraj se jeví jako deformovaný (na levé straně vyšší).

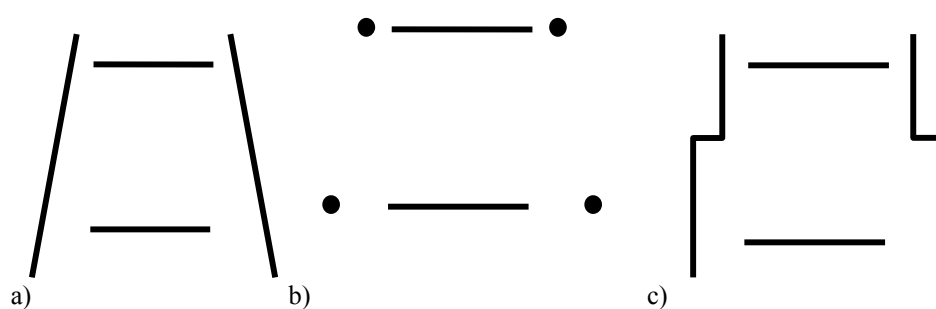
V případě Zöllnerova, Wundtova, Heringova i Orbisonova klamu vede distorze při zpracování úhlů k distorzi vnímaných orientací a tvarů. Není tomu tak ale u Poggendorffova klamu (obr. 11b). Zde distorze ve zpracování úhlů vede k nesprávnému vnímání vzájemné polohy – k iluzivnímu posunutí části čáry, která šikmo kříží středový útvar.

Expanze ostrých úhlů nemá lokální charakter [23]. Nezáleží tedy přímo na úhlu, jaký čáry svírají v místě křížení, ale na celkově převažujících směrech.

Klamy, jako je Heringův, Wundtův a Ehrensteinův, pracují se šikmými liniemi, které vychází z jednoho bodu, který by mohl představovat úběžník. Podobně je tomu u Ponzova klamu (obr. 13a) a všechny tyto klamy lze zobecnit do figury, kterou představili Changizi a kol., která je na obrázku 12d [38]. Účinek Ponzova klamu spočívá v tom, že dvě stejné vodorovné úsečky jsou uvozeny šikmými liniemi. Horní vodorovná úsečka se přitom zdá delší než ta spodní. Ponzův klam lze mimo jiné vysvětlit pomocí teorie expanze ostrých úhlů: mezi horizontálními úsečkami (pražci) a uvozujícími šikmými liniemi (kolejnicemi) lze předpokládat pravý úhel, který se ovšem vlivem perspektivního zkreslení na sítnici zmenšil.



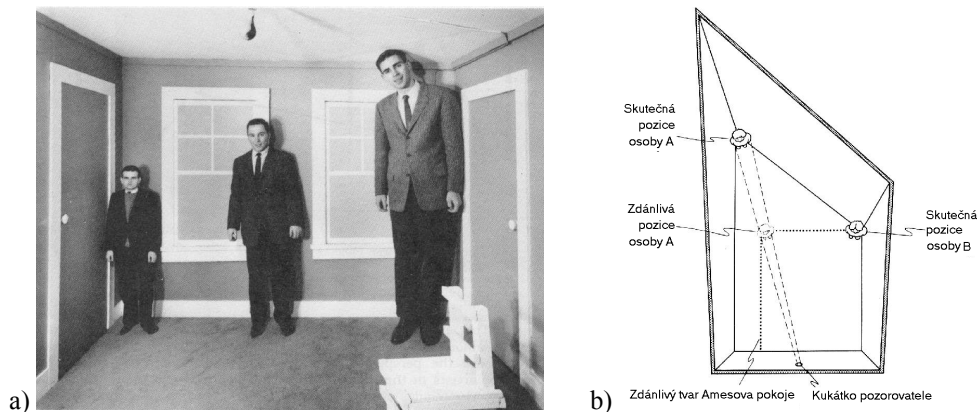
Obr. 12. Heringův, Wundtův, Ehrensteinův klam a jejich zobecňující figura [38].



Obr. 13. Ponzův klam, Ponzův klam vytvořený efektem rámování a rektilineární varianta Ponzova klamu.

Na druhou stranu, uspokojivé síly Ponzova klamu lze dosáhnout mnohem jednodušeji než zešikmenými liniemi, a to pomocí tzv. efektu rámování (obr. 13b).

I když při efektu rámování jsou přítomny jen čtyři body, jedná se o náhradu právě zešikmených (úběžníkových) čar. Ponzův klam však zcela vymizí, pokud jsou zešikmené linie nebo uvozující body nahrazeny pravoúhlou strukturou, tzv. rektilineární variantou (obr. 13c) [60].



Obr. 14. Pohled do Amesova pokoje a nákres Amesova pokoje.

Předpoklad převahy pravých úhlů ve scéně a s ním i předpoklad, že hrany ve scéně budou nejspíš vzájemně kolmé nebo naopak rovnoběžné, s sebou přináší možnost mnohem rafinovanějších iluzivních kreací, než jsou prosté geometrické klamy. Asi nejznámějším příkladem, kdy „dogmatický“ předpoklad pravých úhlů ve scéně vede pozorovatele k absurdním závěrům, je Amesův pokoj.

Amesův pokoj je místnost konstruovaná tak, že její pravý zadní roh je umístěn blíže k pozorovateli než roh levý. Výška pravé strany místnosti je zároveň menší než nalevo a všechno – sklon stropu i podlahy, dekorace na stěnách i v prostoru – je náležitě zkresleno tak, aby pozorovatel ze svého jednoho stanoviště získal dojem, že se dívá do perfektně pravoúhlé místnosti. Pozorovatel je omezen tak, že smí do místnosti nahlížet jen monokulárním kukátkem. To mu znemožňuje použít binokulární nápovědi a pohybové paralaxy. Tato omezení umožňují, že při pohledu do Amesova pokoje spatří pozorovatel hotové divy: například člověk, který přechází z jedné strany na druhou, se průběžně zvětšuje nebo zmenšuje⁸.

Klam a vnímané vzdálenosti objektů

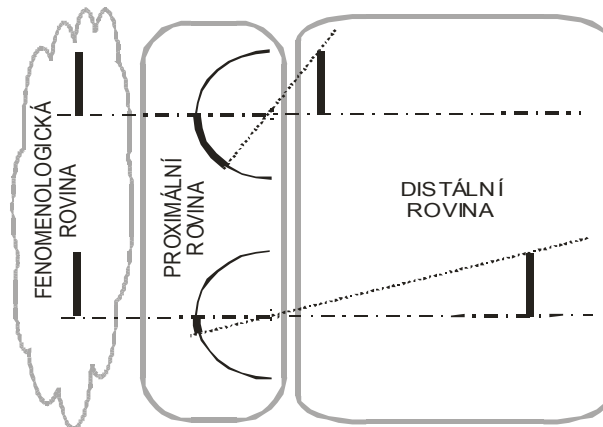
Někdy bývá proces vnímání rozdělen v zásadě do tří částí. Vnější (fyzikální nebo objektivní) rovina se nazývá distální rovinou. Na smyslových orgánech (v našem případě na sítnici oka) se zobrazuje distální objekt ve své proximální podobě. Je to rovina proximální. Transformace mezi distální a proximální rovinou je řízena zákony geometrie, optiky, popřípadě Dopplerovým principem a podobně. Při této transformaci

⁸ Na internetu lze najít řadu více či méně povedených videí z různých videoukázek Amesových pokojů, například na adrese <http://www.maniacworld.com/ames-room-illusion.html>. Ještě zajímavější ukázky se dají najít na YouTube.

ke vnímání v pravém slova smyslu ještě nedochází. Naopak dochází pouze ke ztrátě informace. Z trojrozměrného objektu se stane na sítnici oka jeho dvojdimenzionální projekce. Třetí rovinou je fenomenologická rovina, kde se z počitků stává vjem – naše psychologická reprezentace vnějšího (distálního) světa. Je přitom zřejmé, že po transformaci mezi distální a proximální doménou si reprezentace nějakého objektu nemusí se svou reálnou předlohou odpovídat [70].

Představme si dvě stejné tyče, které pozorovatel vidí ležící na zemi. Jedna tyč je umístěna dál a druhá blíž k pozorovateli. Je zřejmé, že ačkoli je skutečná (distální) délka obou tyčí stejná, jejich sítnicový (proximální) obraz bude různý. V tomto smyslu si distální a proximální rovina nebudou odpovídat – nebude mezi nimi soulad. Na fenomenologické rovině, kde vzniká naše vnitřní (psychická) představa o vnějším světě, však budeme schopni rozpoznat odlišnou vzdálenost obou tyčí a korigovat jejich velikosti proti proximálnímu obrazu na sítnici. Obě tyče budeme vnímat jako stejně dlouhé. Všimněte si, že zatímco distální a proximální rovina vnímání nejsou vzájemně v souladu a ani proximální rovina si neodpovídá s fenomenologickou, vzniká soulad mezi distální a fenomenologickou rovinou. Tomuto jevu se říká percepční stálost (*perceptual constancy*). Situace je zobrazena na obrázku 15 [70].

S pojmem percepční stálosti (respektive rozměrové nebo tvarové stálosti) přišel francouzský filosof René Descartes a své úvahy na toto téma publikoval v roce 1638 ve spise *Dioptrica* [26].



Obr. 15. Situace percepční stálosti

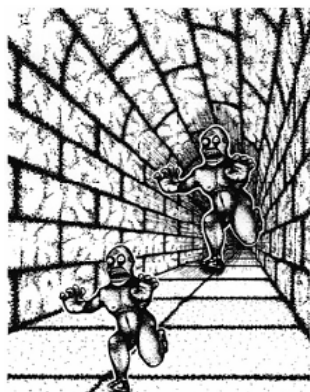
Nyní si představme poněkud odlišnou situaci, například figuru Ponzova klamu (obr. 13a). Mezi distální a proximální rovinou vnímání bude soulad, protože obě vodorovné úsečky se na sítnici oka budou promítat jako stejně dlouhé. Vnímající ovšem, navzdory tomuto souladu, použije kontextovou proměnnou k tomu, že na fenomenologické úrovni změní dosud si vzájemně odpovídající velikosti úseček a rozhodne se vnímat obě úsečky jako různě dlouhé. Zešikmené linie vytváří dojem hloubky. V této situaci sice existuje, jak jsme již napsali, soulad mezi distální a proximální sférou, nicméně percipient se pod dojmem různé vzdálenosti obou

srovnávaných úseček vzdává souladu mezi proximální a fenomenologickou rovinou a tím vzniká rovněž nesoulad mezi rovinou distální a úrovní vjemu (fenomenologickou). Nyní je zřejmé, že percipient vnímá něco jiného, než na co se ve skutečnosti dívá. Toto je obecný popis percepčního klamu.

Todorovič [70] dokonce požaduje, aby byl každý percepční klam z definice založený právě na dvou stejných podnětech a klamný dojem jejich různosti byl vytvořen manipulující (většinou kontextovou) proměnnou. Todorovič klade fenomén klamu do protikladu k fenoménu percepční stálosti, kde naopak dva podněty, které se na proximální rovině jeví jako rozdílné, vidíme jako stejné.

Něco mají ale jev percepční stálosti a zrakové klamy společné. V obou případech vzniká rozdíl mezi proximální a fenomenologickou rovinou vnímání. Pozorovatele k tomu vedou většinou kontextové okolnosti a přesvědčení, že díky vytvoření tohoto nesouladu bude schopen dosáhnout konstantního vnímání.

Podobně jako Ponzův klam je vysvětlován i Shepardův klam „Terror Subterra“ (obr. 16). Tento klam bývá také nazýván koridorový [3]. Jde o umístění dvou objektů do přirozené scény, která vyvolává co nejvýraznější dojem hloubky. Oba objekty jsou ve skutečnosti stejné a jejich velikost (vzhledem k obrázku) je stejná. Jeden objekt je v rámci zobrazené scény umístěn blíž a druhý dál. Objekt, který je ve scéně umístěn dál, se jeví jako větší.



Obr. 16. Shepardův koridorový klam „Terror Subterra“

Scéna může, ale nemusí obsahovat linie, které perspektivně míří do hloubky (jako je tomu u Ponzova klamu). Efekt se dostaví, když scéna obsahuje jakékoli nápovědi, které vytváří dojem hloubky (stačí například gradient textury) [68].

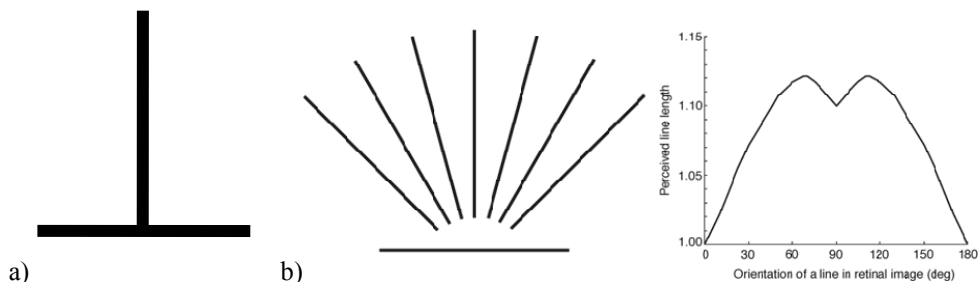
Proti představě, že koridorový klam je způsoben předpokladem existence scény, by mohla hovořit Newmanova studie [14]. Newman zjistil, že tato iluze působí i na malé děti, které přitom v obrázku nepoznají 3D scénu. Neplatí tedy, že pouze (vědomé) rozpoznání hloubkové scény umožňuje percepci hloubky v kresbě.

Koridorový klam by se mohl zdát „realističtější“ variantou Ponzova klamu. Jde vlastně o kresebnou náhradu fotografie, která vyvolává dojem hloubky. Stejný efekt, jakého dosahuje Ponzův klam, možná dokonce ještě výraznější a robustnější, by bylo

možné dosáhnout prezentováním ponzovských úseček na pozadí fotografií, které zachycují hloubkovou dimenzi. Leibowitz a kol. [49] administrovali studentům z USA, Ugandy a ostrova Guam klasický Ponzův klam, dále úsečky zobrazené na pozadí fotografie s texturní prostorovou nápovědí a na pozadí fotografie železničních kolejí (liniové i texturní prostorové nápovědi). Američtí i ugandští studenti podléhali klamu více s rostoucím množstvím přítomných nápovědí. Guamští studenti byli signifikantně odolnější proti klamům vyvolaným fotografickými pozadími, ale čistému Ponzovu klamu podléhali stejně jako ostatní. O rok později administrovali tutéž sérii obrázků ugandským vesničanům a ti vůbec nepodléhali ani Ponzovu klamu a ani klamům vyvolaným fotografiemi. Autoři z výsledků vyvozují, že odolnost vůči klamu je u guamských studentů dána jejich malou zkušeností s přirozeně se vyskytujícími úběžníkovými liniemi. Na ostrově Guam totiž nebyly (kromě krátkých úseků v přístavu) žádné silnice. Naprostá odolnost ugandských vesničanů však takto vysvětlitelná není, protože jejich prostředí rovné linie obsahuje. Autoři se domnívají, že tito lidé mají jen malou zkušenost s plošným vyobrazením prostoru a nejsou zvyklí automaticky odfiltrovat plošný rozměr takových obrázků nebo fotografií. Této domněnce se říká hypotéza *“experience with pictures”* [14].

Percepční stálost nakloněných délek

Na principu percepční stálosti lze vysvětlit nesoulad mezi viděnými délkami v horizontální a vertikální orientaci. Horizontálně vertikální anizotropie byla diskutována již Lotzem roku 1852 a Oppelem v roce 1854 [71]. Jejím jednoduchým příkladem je klam obráceného „T“ Wilhelma Wundta z roku 1858 (obr. 18a). Klam spočívá v tom, že ze dvou identických vzájemně kolmých úseček se ta horizontální zdá být výrazně kratší a je potřeba její délku zvětšit až o 30%, aby obě úsečky vypadaly jako stejně dlouhé.



Obr. 18. Klam obráceného „T“, síla horizontálního-vertikálního klamu v závislosti na sklonu svislé úsečky [34].

Oppel přišel s vysvětlením horizontálně vertikální anizotropie, které souvisí s prostorovým vnímáním. Tvrdil, že vertikální linie v námi sledovaných obrázcích často reprezentují délky ležící na zemi, tedy v hloubkovém směru [71]. Tyto délky vidíme, na rozdíl od horizontálních délek, ve výrazném perspektivním zkrácení. Proto je lidský zrak automaticky prodlužuje.

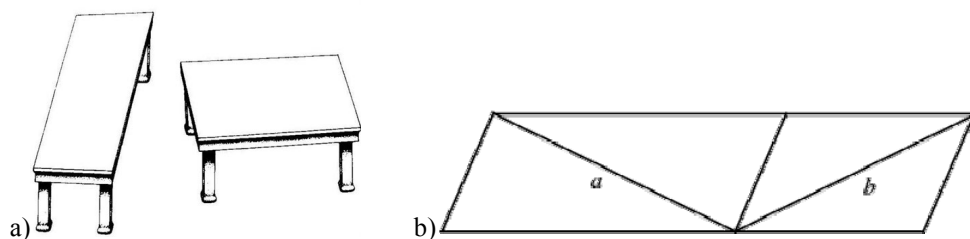
Gregory [31] uvádí, že když se klam vyvede světlou barvou na tmavém pozadí, jeví se v zešeřelé místnosti vertikální linie tak, jakoby skutečně ubíhala dozadu.

Na podporu scénové teorie přišly etnografické studie, které srovnávaly citlivost na klam obráceného „T“ u lidí, kteří žijí na otevřených pláních, lidí euroatlantického okruhu a lidí, kteří žijí v pralesním prostředí. Výsledky odpovídaly teorii. Lidé, kteří žijí na širokých prostranstvích, jsou ke klamu citliví nejvíce. U Evropanů se prokázala střední citlivost, zato lidé z pralesa byli vůči klamu odolní. Vertikální linie prostředí pralesních lidí jsou často tvořeny stojícími stromy a perspektivní zkrácení u nich není obvyklé. Ležících linií, které by se zobrazovaly vertikálně, si v zarostlém prostředí pralesní lidé neužijí [14].

Pokud se vertikála nalézá nad středem horizontální linie, jde o zobrazení, které odpovídá konvergující lineární perspektivě. Tehdy je ostatně klam nejsilnější [31]. Dále se zjistilo, že na sílu obráceného „T“ klamu má vliv vzájemná orientace úseček. Zajímavé je, že maximálního efektu klamu lze dosáhnout nikoliv při kolmosti obou úseček, ale při odchýlení svislé úsečky o 30 stupňů od vertikálního směru [34] (obr. 18b).

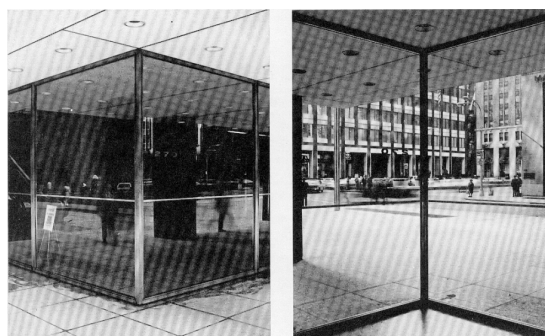
Howe a Purves [34] přišli s názorem, že účinek klamu souvisí se statistickým vztahem mezi sítnicovými délkami pozorovaných linií a jejich reálnými délkami v přirozeném prostředí. Vybaveni moderní technologií se rozhodli tento předpoklad testovat. Klam je podle nich důsledkem ekonomiky vnímání, kdy je ke svislým liniím předem přistupováno jako k obrazu, který má hloubkovou dimenzi, a je tedy potřeba ho přeškálovat. Hledali ve scéně linie a jejich orientace a přiřazovali jim fyzikální délky zjištěné laserovým dálkoměrem. Výsledky autorů korespondují s křivkou na obrázku 18b, a to i včetně poněkud překvapivého propadu v čistě vertikální orientaci.

Princip percepční stálosti se uplatní i tehdy, když jsou zobrazený objekt nebo jeho část, například jedna hrana, vnímány jako perspektivně zkrácené. Efekt se umocní, pokud okolí dělá dojem hloubkové scény. Na tomto principu funguje Shepardův klam nazvaný *Table Illusion* (obr. 19a). Levý a pravý stůl mají stejné strany, jsou pouze pootočený o 90°. Vtip je v tom, že má být srovnávaná délka těchto hran na obrázku. Protože stoly jsou nakresleny bez perspektivního zkreslení, které člověk předpokládá, jeví se levý stůl jako nápadně delší.



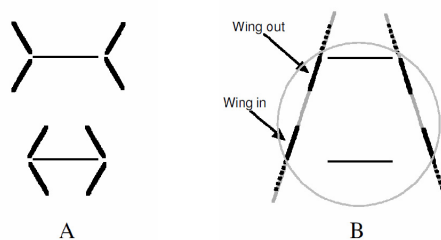
Obr. 19. Shepardova „Table Illusion“ a Sanderův paralelogram.

Stejný efekt jako v případě Shepardova klamu se ovšem dostaví i u zcela abstraktní figury na obrázku 19b, kde se úhlopříčka a jeví jako úsečka odlišné délky než úhlopříčka b. Klam popsál v roce 1926 F. Sander, po němž také dostal jméno.



Obr. 20. Ukázka Müller-Lyerova klamu v přirozeném lidském prostředí [31].

Vysvětlení tohoto jevu podle teorie založené na předpokladu existence scény vychází z představy, že konkávně uvozené hrany se obvykle nacházejí dál od pozorovatele než hrany uvozené konvexně. Zároveň může být Müller-Lyerův klam považován za útvar podobný Ponzově klamu (obr. 21) [60]. Když se v takovém případě aplikuje princip percepční stálosti, jeví se iluzivně vzdálenější (konkávně uvozená) úsečka delší než úsečka s konvexním uvozením, a ta je pak vnímána jako bližší. Pokud je scéna, kterou pozorovatel vidí, skutečně trojdimenzionální, jsou délky hran škálovány korektně a efekt Müller-Lyerova klamu se nedostaví. To podle Gregoryho představuje důkaz, že Müller-Lyerův klam není způsoben chybným zpracováním zrakového signálu, ale vzniká zapojením znalostí o prostorové deformaci délek při jejich plošném zobrazení. Gregory také publikoval fotografie scén, které tuto skutečnost přesvědčivě dokládají a vysvětlení Müller-Lyerova klamu jako důsledku předpokladu scény se jeví jako nezvratná [31].

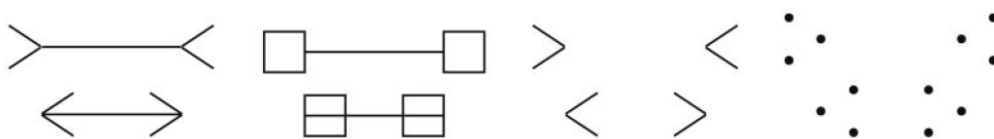


Obr. 21. Srovnání Müller-Lyerova a Ponzova klamu [60].

V našem přirozeném (dávno urbanizovaném) prostředí jsou časté existence pravých úhlů – tedy na sebe kolmých linií. Hovoří se o hypotéze tesařského světa (*carpentered world hypothesis*). Ne všechny národy však žijí v prostředí, které by bylo zaplněno pravými úhly. Riversovy studie napříč kulturami již na začátku 20. století ukazovaly, že lidé mimo euroatlantickou civilizaci jsou méně náchylní k vnímání klamů jako je Müller-Lyerův nebo horizontálně vertikální klam. Segall a kolegové zkoumali též účinek Sanderova paralelogramu. Došli k závěru, že lidé mimo západní

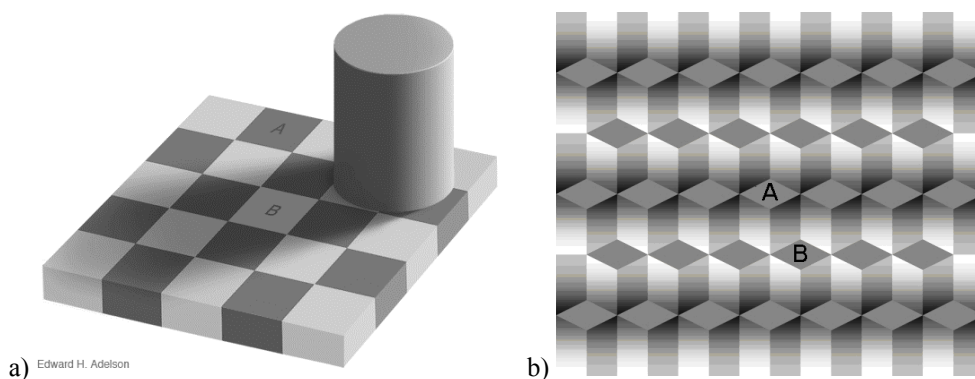
civilizaci nejspíš nejsou připraveni vnímat své prostředí jako „pravoúhle otesané“ a též nemají tolik zkušeností s plošným (realistickým) vyobrazením. Proto méně podléhají geometrickým iluzím tohoto typu [14], [49], [66].

Existují ovšem důkazy, které by mohly platnost scénové interpretace u geometrických klamů zpochybnit. Například nahrazení konvexních a konkávních šipek u Müller-Lyerova klamu jinými způsoby zakončení nezruší účinek klamu docela a další fungující modifikace klamu znemožňují prostorovou interpretaci (obr. 22). Howe a Purves hledali statistický vztah mezi konvexními a konkávními rohy u databáze fotografií a srovnávali je s jejich vzdáleností od objektivu kamery. Navzdory Gregoryho intuici nenašli vztah mezi typem rohu a jeho vzdáleností v rámci fotografovaných scén [35].



Obr. 22. Müller-Lyerův klam a některé jeho neprostorové modifikace [35].

Princip percepční stálosti se svým způsobem uplatňuje u vysvětlení všech klamů, které jsou založeny na vnímání obrazu jako prostorové scény. Na druhou stranu je tento princip mnohem širěji uplatnitelný, než jen na prostor. Jako se princip percepční stálosti uplatňuje u geometrických zrakových klamů, stejně se může uplatnit u některých zrakových klamů, které pracují s jasem.



Obr. 23. Adelsonův a Logviněnkův klam [50].

Na obrázku 23a vidíme jeden velmi působivý klam, který pracuje s percepční stálostí jasu. Je to Adelsonův klam stínu na šachovnici. Vtip tohoto klamu spočívá v tom, že tmavý čtvereček šachovnice, který je označený jako „A“ je ve skutečnosti stejně jasný, jako světlý čtvereček označený „B“. Z porozumění celé scény vidíme, že část šachovnice se čtverečkem „B“ je ve stínu, zatímco čtvereček „A“ je na světle. Automaticky přeškálováváme jas šachovnice tak, aby se nám bílé čtverečky na světle

jevily stejně jako bílé čtverečky ve stínu, a stejně tak, abychom všechny tmavé čtverečky vnímali jako stejně světlé. Bude pro nás tedy nepřijatelné, aby měl tmavý čtvereček na světle stejný jas jako světlý čtvereček ve stínu. Vysvětlení pracuje se vznikem klamu na takové úrovni kognitivního zpracování viděné scény, kdy je schopen percipient identifikovat celou šachovnici jako jeden objekt s povrchem opakujícího se šachovnicového vzoru. Zároveň, jak předpokládá Bach [3], musí být přítomen předpoklad stínu vrženého na šachovnici. U Logviněnkova klamu (obr. 23b) to ale neplatí. Zde jsou opět plochy „A“ a „B“ stejně světlé, ačkoli je obrázek zcela abstraktní. Logviněnkův a též paralelogramový klam ukazují, že efekt percepční stálosti není vázán striktně na pochopení obrazu jako scény.

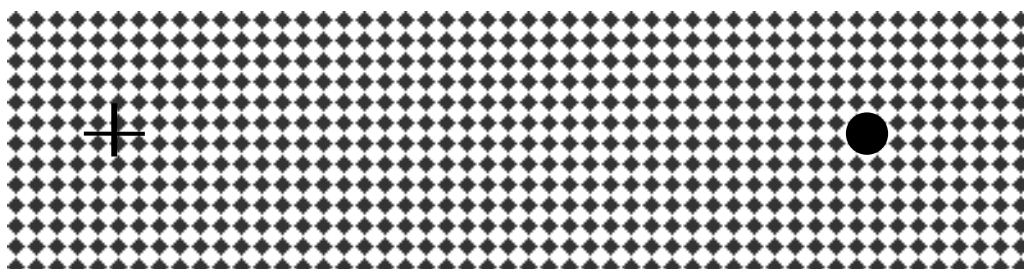
3 Fyziologické teorie zrakových klamů

Určitá skupina zrakových klamů bývá vysvětlována pomocí senzorio-neurologických procesů. Pro výzkumníky, kteří se jimi zabývají, jde o chybné fungování neurologických mechanismů, které ovšem za normálních okolností dobře plní svoji funkci.

3.1 Teorie operující na senziorické úrovni zpracování vizuální informace

Pokud pomineme čistě optické jevy a patologie, vychází nejjednodušší fenomény zrakových klamů z konstrukce lidského oka, respektive z uspořádání jeho světločivé vrstvy – sítnice neboli *retiny*. Oční sítnice je struktura oka, na kterou se promítá obraz zrakového podnětu. Sítnice je osázena receptory citlivými na světlo. Tyto receptory jsou dvojího druhu. (1) čípky, které slouží za dne a umožňují barevné vidění a (2) tyčinky, které jsou mnohem citlivější na světlo, než čípky, jejich signál je ovšem černobílý a obraz, který dávají, neostrý. Největší hustota čípků je kolem optického centra sítnice, *fovea*. Ačkoli fovea zabírá jen 2° zorného pole a přilehlé (parafoveální) oblasti 10°, zpracování signálu z této oblasti se věnuje až 80% neuronů primárního vizuálního kortexu.

Okraje sítnice jsou osázeny více tyčinkami, které nerozlišují barvy, ale zato jsou mnohem citlivější na světlo, než čípky. Profesionální i amatérští astronomové vědí, že málo jasný objekt na obloze mohou lépe spatřit na okraji svého zorného pole (říká se „koutkem oka“) než v jeho středu. Další zajímavou oblastí sítnice je tzv. slepá skvrna, kde do oka vstupují cévy vyživující sítnici a kde rovněž z oka odchází zrakový nerv. V tomto místě, které zabírá cca 5° zorného pole, jakékoli receptory citlivé na světlo chybí. Slepou skvrnu nicméně nevidíme jako „prázdnou“. Zrakový systém totiž doplňuje tuto chybějící část zorného pole, jak demonstruje pokus na obrázku 24.



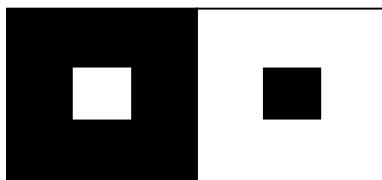
Obr. 24. Demonstrace existence slepé skvrny. Zavřete levé oko a zaměřte se na kříž vlevo. Pohybuje předlohou od sebe a k sobě (stále se dívejte na kříž), dokud nenastavíte vzdálenost, ve které černé kolečko vpravo zmizí. Prázdné místo bude jakoby vyplněno okolním vzorem.

Iradiační teorie

Když se podíváme na obrázek 25, bude se nám jevit bílý čtvereček na černém pozadí jako větší, než je černý čtvereček na bílém pozadí. Tento fenomén zkoumal Helmholtz, ale popsal jej již Leonardo da Vinci [41].

Podle *iradiační teorie (irradiation theory)* dochází k jevu expanze světla na sítnici. Expanze světla může při dalším zpracování vést k posunutí hranic objektů a ke zkreslení směrů a tvarů a napomoci tak ke vzniku komplexnějších klamů, jak uvidíme později.

Fenomén iradiace závisí na míře jasu, proto se nám na noční obloze jasnější hvězdy jeví jako větší než hvězdy s menším jasem [71]. Konečně míra jasu astronomických objektů se nazývá hvězdná velikost. Přitom hvězdy jsou ve skutečnosti prakticky svítícími body a z hlediska rozlišovací schopnosti oka skutečně žádný rozměr nemají.

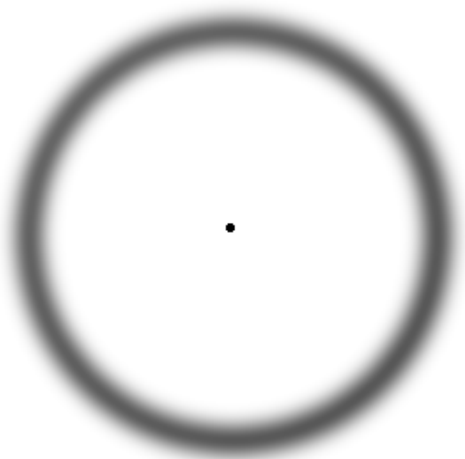


Obr. 25. Expanze světla: bílý čtverec na černém poli se jeví jako větší než černý čtverec na bílém poli.

Troxlerův efekt a paobrazy

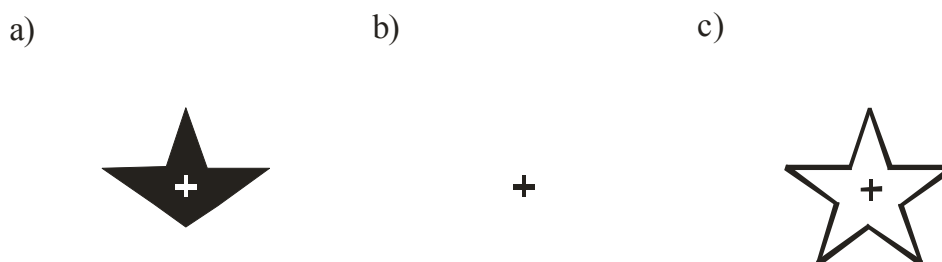
Dalším příkladem klamů, ke kterým dochází v počátečních etapách zpracování zrakové informace, jsou ty, které vznikají vlivem dlouhodobé stimulace nějakého místa na sítnici oka. V tomto místě se receptory adaptují na neměnný podnět a přestanou se aktivovat. V důsledku toho pozorovatel přestává takový podnět vidět. Tento jev je

možné demonstrovat na příkladu Troxlerova efektu (obr. 26). Budete-li sledovat černou tečku uprostřed šedé kružnice po dobu asi 15 sekund, začne šedá kružnice blednout a mizet. Aby se Troxlerův efekt dostavil, je důležité, aby podnět neměl ostré kontury, protože oční tremor (drobné mimovolní chvění očí) by vedl ke stálému obnovování hran objektu. Proto musí být kružnice na obrázku rozmazaná.



Obr. 26. Troxlerův efekt.

Po skončení dlouhodobě konstantní stimulace (zvláště pokud je objekt dostatečně kontrastní vůči svému pozadí), adaptovaným receptorům sítnice bude nějakou dobu trvat, než se přizpůsobí změněným podmínkám a mohou načas produkovat signál, který je negativem původního podnětu. Tomuto reálně neexistujícímu negativnímu podnětu se říká *paobraz* (obr. 27). Paobrazy mohou být achromatické, jako v naší ukázce, ale stejně tak mohou být i barevné. Barva vzniklého paobrazu přitom odpovídá komplementární barvě předcházejícího podnětu. V místech, kam se při expozici obrázku 27a na sítnici zobrazovala tmavá plocha černého útvaru, klesal práh citlivosti na světlo relativně proti místům, kam se na sítnici zobrazovalo bílé okolí. Když pak pozorovatel přesune svůj zrak na homogenní bílou plochu (obrázek 27b), na několik sekund spatří inverzní obrázek, dokud se prahy citlivosti nevyrovnejší. Pokud se pozorovatel po expozici figury 27a přesune na figuru 27c, výsledný paobraz vyplní kontury nového podnětu. Paobraz bude mít nyní tvar kompletní pěticípé hvězdy.



Obr. 27. Paobrazy. Dívejte se několik desítek sekund na bílý křížek na 27a, pak fixujte zrak na křížku 27b. Měli byste spatřit zářivý obraz ve tvaru obrázku 27a. Nyní zopakujte fixování na obrázek 27a a pak po několika desítkách sekund fixujte zrak na křížek uprostřed 27c. Zahlédnete, že světlý paobraz má tendenci vyplnit celou hvězdu, včetně spodních cípů, které původní podnět neobsahuje.

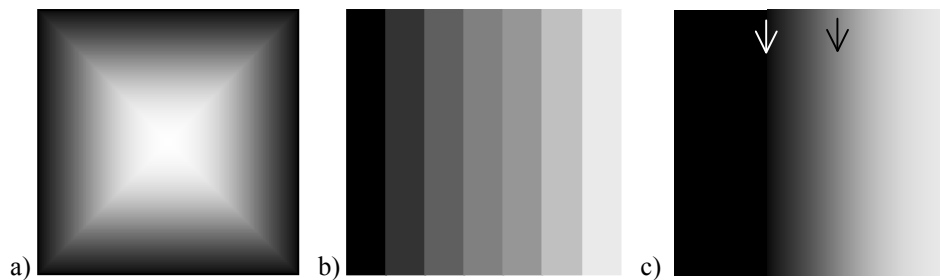
Existují daleko zajímavější konstrukce klamů založené na vyplnění paobrazu. První cenu v soutěži o nejlepší zrakový klam⁹ za rok 2008 dostal například klam z dílny Roba van Liera a Marka Vergeera, který pracuje s barevnou formou paobrazů.

Laterální inhibice

K laterální inhibici dochází podle fyziologické teorie v gangliových buňkách sítnice a je to rovněž adaptační mechanismus, který mění podle potřeb práh citlivosti světelných receptorů. Narozdíl od příkladu paobrazů zde nedochází ke vzniku efektu díky adaptaci v čase, ale díky tomu, že jsou jednotlivé neurony inhibovány podle aktivace neuronů ve svém bezprostředním okolí. Je mezi nimi negativní zpětná vazba.

Na obrázku 28a můžete vidět světlejší diagonální kříž, ve skutečnosti ale na diagonálách žádné zesvětlení není. Každý ze soustavy proužků s odstupňovanou úrovní šedi (obr. 28b) se jeví světlejší v sousedství tmavějšího a tmavší v sousedství světlejšího pásu. Tento starý klam je nazvaný po Ernstu Machovi [48]. Lineární přechod z černé po světle šedou na obrázku 28c se jeví jako by byl orámován tmavější barvou na konci přechodu do černé a světlejší barvou na konci přechodu do světle šedé. Takový přechod tam ovšem není. Nicméně jde o způsob, jak vytvořit pomocí kontrastu barvu „černější než černou“ a „bělejší než bílou“.

⁹ *Best Visual Illusion of the Year*: <http://illusioncontest.neuralcorrelate.com/>.



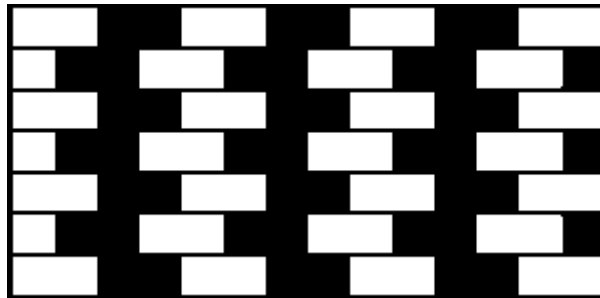
Obr. 28. Klamy založené na gradientu světlosti a principu laterální inhibice

Corney a Lotto [10] ukazují, že efektu několika zrakových klamů, které bývají interpretovány pomocí laterální inhibice, lze dosáhnout u umělé neuronové sítě, která je trénována na udržení stálé světlosti. Jejich síť například produkuje obdobný profil u klamů, jaké můžeme vidět vlastním zrakem na obrázku 28. Proces laterální inhibice je užíván i při konkrétních technických řešeních, které se týkají strojového vidění [54]. Procesy srovnatelné s laterální inhibicí se objevují i na vyšších úrovních zrakového systému a také u jiných percepčních modalit než jen u zraku.

Teorie kontrakce rohových úhlů

Münsterbergův klam (obr. 29) bývá vysvětlován zdánlivou kontrakcí rohových úhlů tam, kde je bílý roh lemován černými plochami a kde je naopak černý roh lemován bílými plochami. Vlivem tohoto jevu se úhly rohů jeví ostřejší. Tento dojem pak vede k lokálním změnám vnímaného sklonu linie mezi políčky šachovnice. Když je zrakovým systémem analyzována orientace dělicích vodorovných linií, interferuje jejich skutečný směr se zdánlivým lokálním nakloněním. Tak se vytváří dojem, že dělicí linie jsou celé nakloněné.

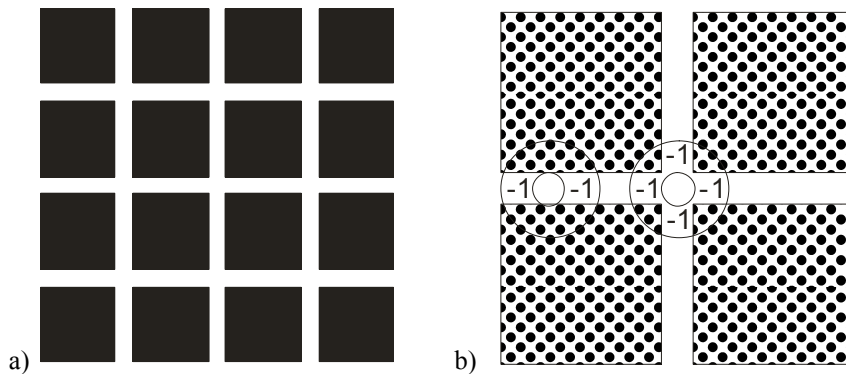
Fenomén kontrakce rohových úhlů byl lokalizován na samotný začátek procesu zrakového vnímání jako jev, ke kterému dochází ještě na sítnici. Původní představa spočívala v tom, že jde o nějaký typ *iradiace*. Kdyby byl efekt kontrakce rohových úhlů zapříčiněn *iradiací* bílé, byly by to pouze bílé rohy, které by za klam byly zodpovědné. Když se ale odstraní působení bílých rohů, nebo když se použije negativ, působení klamu zcela nezmizí [71]. To by napovídalo, že klam Münsterbergovy figury vzniká jiným mechanismem, například procesem laterální inhibice nebo ještě později, nikoli na sítnici. V každém případě ale iradiační teorie nedokáže uspokojivě vysvětlit vznik tohoto klamu.



Obr. 29. Münsterbergův klam [24].

Hermanova mřížka a laterální inhibice

Klam mřížky Ludimara Hermana (obr. 30a) bývá v literatuře popisován jako ukázkový příklad laterální inhibice [5]. V místě křížení bílých linií vzniká dojem, jako by zde člověk pozoroval prchavé ztmavení. V případě Hermanovy mřížky, je neuron snímající signál ze středu bílého kříže inhibován dvojnásobným množstvím neuronů z okolí, oproti neuronu, který přijímá signál z kteréhokoli jiného místa na spojnici (viz obrázek 30b).



Obr. 30. Hermanova mřížka a vysvětlení klamu pomocí laterální inhibice.

Můžete si všimnout, že pokud svůj zrak zaměříte na vybraný bílý kříž mřížky (obr. 30a), efekt se v tomto místě nedostaví. Podle sítnicového vysvětlení je to proto, že centrum vizuálního obrazu je promítáno na foveu, kde se nachází největší koncentrace čípků a kde tudíž k laterální inhibici dochází na mnohem menších sítnicových rozměrech obrazu.

3.2 Teorie vzniku klamů ve zrakové kůře

Ve zrakové kůře dochází ke komplexnějšímu zpracování vizuální informace. Světelné znaky již byly kódovány a obraz byl dekomponován na hrany. Nyní se bude zpracovávat orientace těchto hran, jejich tvar a relativní délka.

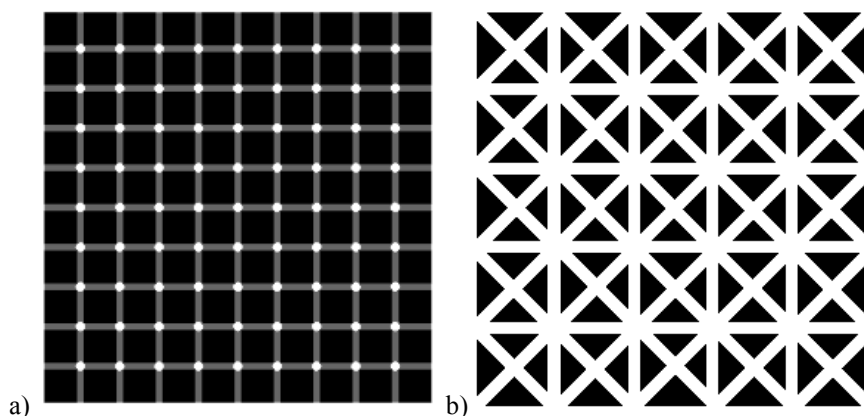
Pro klamy, které jsou vysvětlovány procesy ve zrakové kůře mozku, je typické, že u nich nezáleží na velikosti a světlosti nebo barvě jednotlivých částí. Naopak zde může záležet na orientaci obrazu vytvářejícího klam. Když se zjišťuje, zda je určitý klam závislý na zpracování světla na sítnici nebo na zpracování zrakového podnětu ve zrakové kůře, používá se pro diskriminaci těchto typů takzvané heterochromatické izoluminační zobrazení [71]. Při tomto zobrazení je klam prezentován na místo obvyklé černé a bílé pomocí dvou komplementárních barev (např. červené a azurové) o stejném jasu. Při takovémto zobrazení se sítnicové klamy neprojeví, zatímco klamy založené na komplexnějším zpracování obrazu by neměly měnit svoji působivost.

Skutečnost, že zraková kůra již v podstatě nepracuje s absolutními hodnotami světlosti, ale převážně s hranami, má své důsledky. Zakryjeme-li hranici mezi dvěma sousedními pruhy Machových pruhů (obr. 28b), budou se nám jevit tyto dva pruhy, jejichž společnou hranici jsme zakryli, jako stejně světlé. Tomuto klamu se říká Chevreulův klam a ukazuje nám, že po převedení sítnicového obrazu na hrany, nejsou absolutní hodnoty světlosti jednotlivých částí zorného pole dále zprostředkovávány. Reprezentovány jsou světlosti (respektive barvy) částí zorného pole, které jsou percepčním systémem považovány za celky.

Podobný jev, jako je laterální inhibice na sítnici, se očekává též ve zrakové kůře. Zde se ovšem jedná o inhibiční mechanismus vrstvy neuronů, která detekuje orientaci čar. Navzájem se inhibují sousední neurony, které detekují stejnou orientaci, což má zabránit neblahému působení šumu a učinit tak signál robustnější [71].

Revize Hermanovy mřížky

Již delší dobu panují pochybnosti, zda samotný proces laterální inhibice dokáže vysvětlit klam Hermanovy mřížky (obr. 30). Existují totiž různé modifikace Hermanovy mřížky, jejichž působení neodpovídá účinku laterální inhibice a vzniku klamu při zpracování podnětu již na sítnici. Ukázalo se, že efekt Hermanovy mřížky je nezávislý na rozměrech a rovněž na barevné inverzi. Naopak je závislý na orientaci mřížky a na tvaru křížících se linek. Nejznámější modifikací Hermanovy mřížky je scintilující mřížka (obr. 31a), u které lze vidět dokonce výraznější efekt než u původní Hermanovy verze. Podle principu laterální inhibice by zde měl být efekt naopak slabší. Je tomu obráceně u modifikace na obrázku 31b, kde by efekt měl být silnější, ale je spíše méně výrazný.

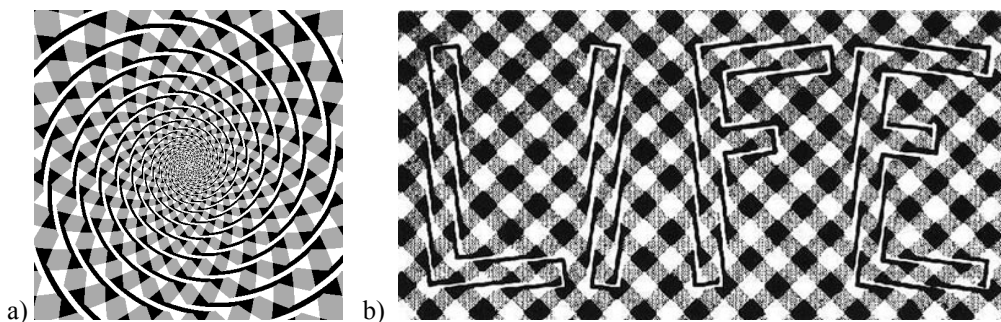


Obr. 31. Některé modifikace Hermanovy mřížky (převzato z ⁹⁾).

Ze Schillerovy laboratoře MIT pochází teorie *S1 simple-cell theory*, která předpokládá vznik klamu Hermanovy mřížky až v oblasti V1 mozkové kůry. Nová teorie odpovídá lépe zmíněným variantám Hermanovy mřížky¹⁰. Pokud je tato teorie správná, náleží klam Hermanovy mřížky do skupiny klamů, o kterých budeme pojednávat nyní. Šlo by tedy o klam, jehož vznik se předpokládá v oblasti zrakové kůry, kde se provádí složitější a komplexnější zpracování obrazu.

Klamy svinutého provazu

Ve zrakové kůře jsou u signálů detekovaných hran zjišťovány jejich orientace. Tento proces může být snadno zmaten a příkladem takto vysvětlovaného klamu je Fraserova spirála, kterou britský psycholog James Fraser publikoval v roce 1908 (obr. 32a).



Obr. 32. Fraserova spirála a klátící se nápis LIFE.

Když prstem poputujete po lince Fraserovy spirály, zjistíte, že ve skutečnosti jde o soustavu kruhů. Tento mocný efekt je způsoben tím, že linky tvořící kruhy jsou

¹⁰ Podrobnosti od autorů *S1 simple-cell theory* naleznete na webové adrese <http://web.mit.edu/bcs/schillerlab/research/A-Vision/A15-3.htm>

složeny ze vzájemně propletených tmavých a světlých čar. Linky připomínají provaz složený z tmavého a světlého pramene a odtud plyne název celé rodiny klamů, mezi které Fraserova spirála patří – *twisted cord*. Základ *twisted cord* klamů tvoří podle Fräsera linky s trojúhelníky na koncích, které Fraser nazval směrové jednotky [65].

Písmena slova LIFE (obr. 32b) vypadají, jako by se klátily na různé strany. Když si ale přiložíme pravítko k jejich svislým nebo vodorovným liniím, zjistíme, že tomu tak není. Efekt je vyvolán tím, že linie písmen jsou složeny z nakloněných černých a bílých čar. Při detekci linií a jejich orientace v primární zrakové kůře, se posuzují jejich jednotlivé části zvlášť. Vlivem šikmých linek se celé linie jeví jako zešikmené. Efekt je dále umocněn károvaným vzorem v pozadí, který maskuje skutečné složení linií.

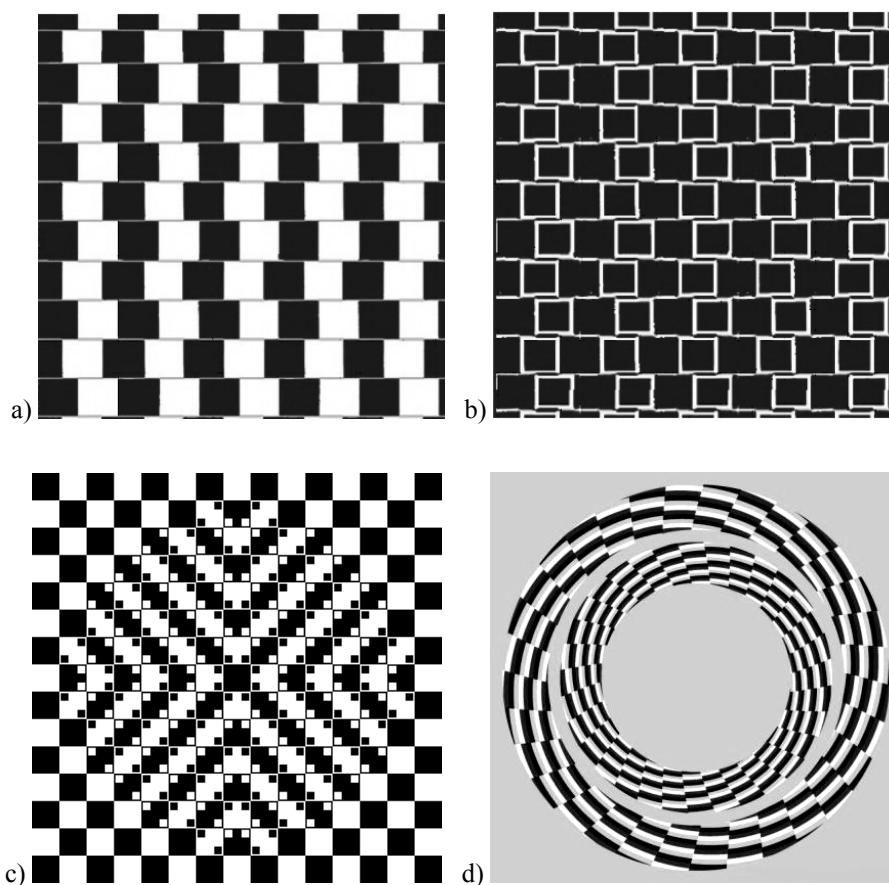
Poté, co jsou v prvních vrstvách zrakové dráhy z obrazu extrahovány linie, začíná jejich další zpracování. Zrakový systém se pokouší určit jejich orientaci. Orientace nějaké čáry však není dána jejím celkovým směrem, ale je skládána ze zjištěných orientací jednotlivých částí čáry. Obsahuje-li čára šikmé pruhy, ne příliš odkloněné od jejího skutečného směru, bude celkový zjištěný směr čáry silně ovlivněný orientacemi pruhů na čáře.

Caffe Wall

Fraser (1908) zjistil, že Münsterbergerův klam lze posílit, když bude střední čára mezi posunutými pásy šachovnice neutrální (šedé) barvy nebo, na rozdíl od černobílé šachovnice, bude barevná s prostřední intenzitou světlosti (obr. 32a). Gregory [24] klam pojmenoval *Caffe Wall Illusion*.

Jak už jsme psali, bylo původní vysvětlení začátku vzniku klamu založeno na jednoduché expanzi světlé při zpracování podnětu na úrovni gangliových buněk sítnice. To ale v případě Münsterbergovy figury a *Caffe Wall* klamu neodpovídá některým modifikacím, a proto se později uplatnilo vysvětlení vzniku pomocí mechanismu laterální inhibice. Regiony šedé dělicí čáry mezi dvěma bílými plochami jsou vnímány jako černé, zatímco ty části, které jsou obklopeny černou, jsou vnímány jako bílé. Černá pole a části dělicí čáry, které jsou vnímány jako černé, vytváří nakloněné elementy, podobné Fraserovým směrovým jednotkám. Stejně tak vytváří nakloněné elementy bílé plochy a bíle viděné části šedé dělicí linie (obr. 32b) [41].

Na tomto principu vytvořil Kitaoka novou skupinu klamů, které nazval šachovnicovými klamy. Tato forma, která je odvozena z *Caffe Wall* klamu nebo z Münsterbergovy figury, umožňuje snadné převedení do plochy a nabízí tak možnost vytvářet působivé plošné kreace (např. obr. 33c). Na stejném principu jako Kitaokův šachovnicový klam nebo klamy typu *twisted cord* fungují klamy posunuté gradace Morgana a Mouldena (obrázek 33d) a klam pruhovaného provazu (*striped cord illusion*) [41].



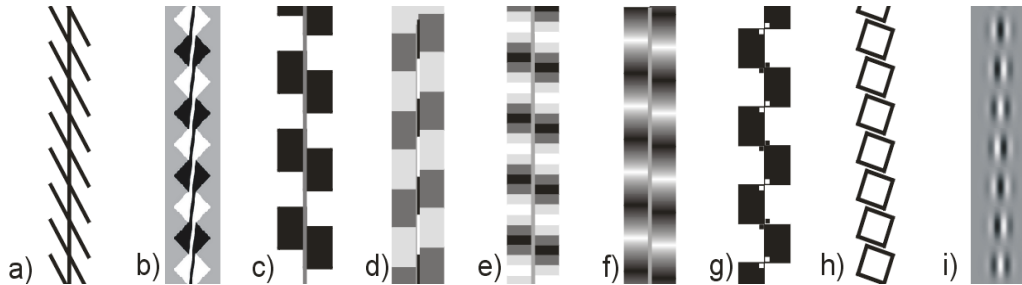
Obr. 33. Caffé wall klam (a) a simulovaná laterální inhibice u tohoto klamu (b) [41], Kitaokův šachovnicový klam (c) a varianta Fraserovy spirály na bázi posunuté gradace (d).

Magie křivých-rovných čar

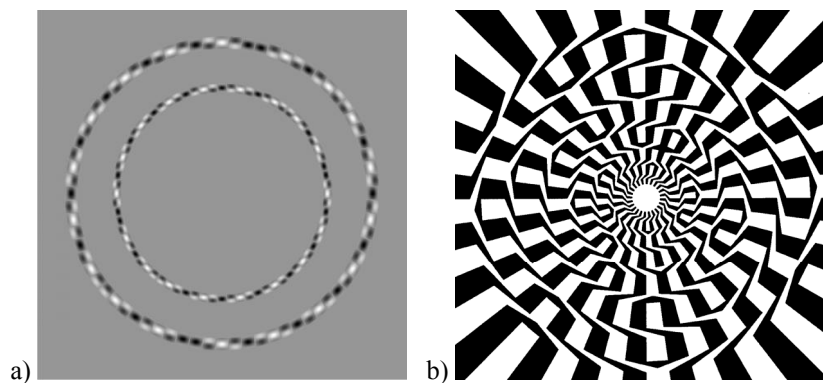
Uvedli jsme několik způsobů, kterých se dá využít pro vytvoření pokřiveného vnímání směru linií. Zatímco Zöllnerův klam a podobné geometrické klamy operují spíše na vyšších (kognitivních) úrovních vnímání, klamy typu *twisted cord*, *striped cord*, *Caffé Wall* a Kitaokovy šachovnice jsou považovány za vysvětlitelné známými neurofyziologickými mechanismy. Ačkoli těchto takzvaně fyziologických klamů existuje celá řada, jejich princip je pravděpodobně velmi podobný. Kitaoka [43], [44] uvádí řadu technik, jak docílit iluzivního dojmu křivé čáry. Využívá přitom stále stejného principu. Některé z technik, jak iluzivně pokřivit směr, potažmo tvar čáry, můžete vidět na obrázku 34. Variant lze ovšem najít mnohem více.

Fenomén tolik nevynikne, pokud jde o iluzivní změnu jen jediné linie. Mnohem pozoruhodnější je působení na ploše, ať už jde o Kitaokovy šachovnice (např. 33c)

nebo o iluzivní spirály ve Fraserově stylu (obr. 31a, 35b), nebo odvozené ze *striped cord* klamu (obr. 35a).



Obr. 34. Metody, jak vytvořit zdánlivý sklon vertikální čáry (doleva): Zöllnerův klam, Fraserovy směrové jednotky, *Caffè Wall*, modifikovaný *Caffè Wall*, diskrétní a kontinuální posunutá fáze, Kitaokův šachovnicový klam, pootočené čtverce jako směrové jednotky a *popple illusion* odvozená ze *striped cord* klamu.



Obr. 35. Iluzivní spirála v popple stylu a Wilcoxův *Twisted Circles*.

4 Od fyziologie zpět ke kognici

4.1 Zrakový klam jako nedokonalost vnímání

Zrakové klamy jsou ve vědeckém světě často chápány jako důsledek chyby percepčního systému. Zrakový aparát se mýlí tu v délce (Müller-Lyerův klam), jinde ve směru (*Caffè Wall*), tvaru (Heringův klam), nebo jasů (Adelsonův klam), špatně interpretuje celý objekt (Penroseův trojúhelník), nebo dokonce vidí, co neexistuje (paobrazy, Hermanova mřížka, Kanizův trojúhelník). Tyto chyby mají ukazovat na nedokonalost lidského zraku. Ovšem zrakový systém by byl nedokonalý v tomto smyslu pouze tehdy, pokud by jeho úkolem bylo věrohodně reprezentovat fyzikální

realitu, od které se chyba při vnímání klamu odvozuje. Tak tomu ale být nemusí. Naše zrakové dovednosti se vyvinuly tak, aby dostatečně dobře reprezentovaly svět, ve kterém žijeme, a to právě tak, jak to ve svém životě potřebujeme. Jde o to, aby náš obraz vnímaného světa byl spíše užitečný než pravdivý ve fyzikálním slova smyslu. Většina klamů je umělého původu nebo, jako je tomu u měsíčního klamu, je zcela irelevantní k obvyklým lidským potřebám. To, zda vnímáme zrakový klam odlišně od skutečnosti, by nemělo podstatně ovlivňovat naše jednání.

Goodale a Milner rozpracovávají práci Mishkina a Ungerleideové o dvou systémech zpracování zrakové informace, z nichž jeden zajišťuje vnímání pro identifikaci objektů a druhý vnímání sloužící jednání. Tento názor vychází ze zobecnění kasuistik neurologických pacientů. U některých došlo vlivem poškození mozku k narušení schopnosti vnímat (rozpoznávat, určovat a popisovat) objekty, zatímco u jiných vedlo neurologické postižení k narušení schopnosti s těmito předměty zacházet (například je chytat). Zároveň však vždy ta druhá dovednost u těchto pacientů zůstala zachována. V souvislosti s teoriemi o vzniku zrakových klamů je zajímavá hypotéza Milnera a Goodala, že identifikující zrakový systém zpracovává relativní velikosti, zatímco vizuomotorický systém vytváří reprezentaci absolutních velikostí a vzdáleností [18].

Právě tak můžeme říci, že zrakové klamy, nebo alespoň část z nich, se týkají vnímání za účelem identifikace objektů a distorze reprezentace objektu vzniká při úlohách, kdy má být klamová figura pouze pozorována. V tomto pojetí by se skutečně jednalo o chybu ve vnímání. Schopnost zacházet s objektem ovšem nemusí být zrakovými klamy zasazena. Zöllnerovy pruhy sice vnímáme nakloněné, ale patrně nebudeme mít problém dotknout se jejich konců. Například u Müller-Lyerova klamu Gillamová a Chambers [23] dokázali, že se nejedná o mispercepci pozice konců úseček, ale o chybnou percepci relativních délek těchto úseček. Řada dalších studií se zabývá vlivem zrakových klamů na jednání. Experimentuje se s různými geometrickými klamy, jako je Müller-Lyerův [18], Ponzův [6], nebo Ebbinghausův klam [32], [53]. Výsledky částečně potvrzují Milnerovy a Goodalovy předpoklady.

Představa, že kvalita lidského jednání se odvozuje přímo z kvality jeho reprezentace, jistě není stoprocentně pravdivá. V případě, kdy je percepce užita k jednání, se totiž stává vnímání nedílnou součástí akce. Lidé používají pro různé úlohy různé interaktivní strategie, kdy v průběhu akce testují, jak je jejich jednání v souladu s požadovaným cílem. Tyto strategie nepočítají s komplexní znalostí prostorového uspořádání, ale užívají zjednodušení, která vedou k cíli dostatečně dobře [18], [67].

4.2 Zpracování podnětu jako trojrozměrné reality

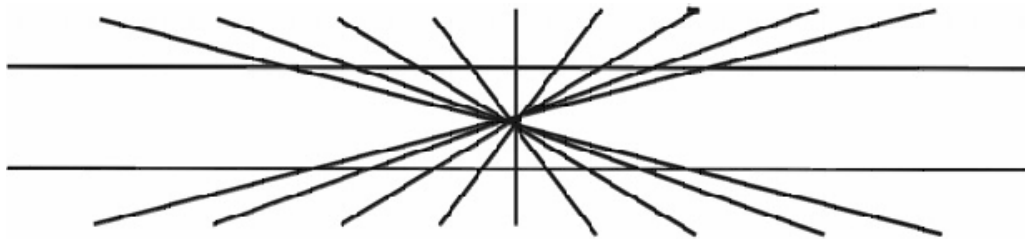
Všimněte si, že fyziologické klamy jsou vysvětlovány způsobem, který nevyžaduje znalost obsahu sledované scény. Procesy, které jsou zodpovědné za vznik fyziologických klamů, předchází pochopení toho, co sledovaná scéna obsahuje a jaký je její význam. Teorie, vysvětlující tuto třídu zrakových klamů, si vystačí s popisem mechanicky pracujících procesů. Tento způsob výkladu je dnes při výzkumu vnímání preferován, protože se může opírat o neurologické nálezy. Jak ale konstatuje

Gillamová, žádná z fyziologických teorií nedokáže vysvětlit určité druhy klamů. K jejich vzniku patrně dochází ještě na vyšší úrovni zpracování obrazové informace, kam dosud neurologický výzkum neprošel a kde vyvstává potřeba zavádět jiné fenomény, než s jakými dokáže operovat výzkum percepce na fyziologické bázi. Příkladem mohou být klamy, které vznikají při percepční organizaci nebo při vnímání prostorového uspořádání [18]. Tyto zrakové klamy nedovedeme vysvětlit bez předpokladu, že si lidský kognitivní systém již udělal určitou představu o tom, co právě pozoruje.

Gregory [27] se domnívá, že fyziologické a kognitivní zrakové klamy mají zásadně odlišné příčiny, i když uznává, že tyto příčiny mohou vést k velmi podobným efektům. Zatímco fyziologické klamy vznikají při zpracování informace jednosměrně směrem vzhůru (*bottom-up*), kognitivní klamy jsou způsobené vysoce organizovanými očekáváním, které zpětně ovlivňují základní procesy (*top-down*).

Je to ale skutečně tak, že trojrozměrný prostor může být reprezentován až na konceptuální úrovni vyšších, kognitivních procesů?

Dlouho panovala představa, že naše reprezentace trojdimenzionálního prostoru je srozumitelná až na kognitivní úrovni a že ji nelze studovat fyziologicky. Nicméně vzhledem k tomu, že se všechna zvířata, lidé a vůbec veškerý život od počátku rozvíjí v trojrozměrném prostředí, mechanismy vedoucí k trojrozměrné interpretaci sítnicového obrazu musí být zastoupeny již na neuronální úrovni. Rozlišení na jevy, které vznikají zpracováním signálu, a na jevy, které jsou zapříčiněny kognitivními strategiemi, se objevuje v moderních neurologických výzkumech jako rozdíl mezi časnými a vyššími kortikálními procesy. Zapomíná se přitom na pochopitelné propojení mezi těmito úrovněmi [71].

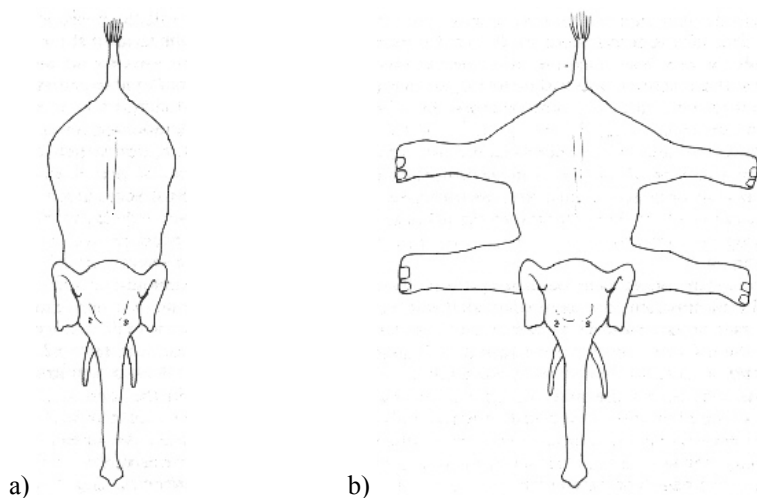


Obr. 36. Ukázka kompenzace iluzivního zakřivení u Heringova klamu [71].

Moderní neurologické výzkumy vnímání, které jsou založeny na funkčních zobrazovacích metodách, dokážou sledovat, jak probíhá zpracování vnímané scény. Nabízí možnost sledovat zpracování jednotlivých aspektů vizuálních podnětů odděleně – pozice linií, jejich délky a orientace nebo velikosti svíraných úhlů. Možná lze dokonce najít kortikální neurony, které jsou zodpovědné za zpracování těchto od sebe oddělených aspektů vizuálního podnětu [71]. Westheimer ovšem připomíná, že konečný vjem vzniká koordinovaným zpracováním všech těchto jednotlivých vlastností vizuálního podnětu. Příklad můžeme vidět na obrázku 36 (Heringova klamu), kde šikmé linie tvoří pozadí po celé délce rovnoběžných čar. Na koncích,

kde šikmé čáry schází, se iluzivní zakřivení vodorovných čar vrací zpět ke své rovnoběžnosti.

Pokud je možné vysvětlit jevy popsané jako expanze ostrých úhlů nebo nadhodnocování vertikálních linií striktně fyziologickými teoriemi, neznamená to, že by nemohlo jít o adaptaci zrakového systému na zpracování trojrozměrné reality. Westheimer [71] předpokládá, že tyto mechanismy na neuronální úrovni lze najít a má smysl jim rozumět jako předběžné reprezentaci prostoru nebo předběžnému zpracování obrazu s ohledem na trojdimenzionální realitu.



Obr. 37. Normální a rozdvojená perspektiva slona z ptačí perspektivy [14].

Zkusme se chvíli zaobírat myšlenkou, že by zvláště geometrické zrakové klamy byly způsobeny jevy předběžného zpracování trojrozměrné reality. Představme si, že pozorujeme slona, ale díváme se na něj svrchu (obr. 37a). Uvidíme hlavu, tělo, ocas. Můžeme si představit také jeho nohy – každý přece ví, že slon má nohy (předpoklad obvyklosti), i když je nevidíme (Kanizův trojúhelník). Délky těchto nohou jsou pochopitelně větší, než jak bychom je viděli z tohoto úhlu perspektivně zkrácené (Sanderův klam) a víme také, že se hřbetem mají svírat v podstatě pravý úhel (expanze ostrých úhlů). Po takovéto úvaze můžeme získat poněkud jiný obraz slona (obr. 37b).

Tyto obrázky ukázal Deregowski africkým dětem i dospělým a Evropanům s instrukcí, aby rozhodli, která z kreseb vypadá spíš jako slon svrchu. Evropanští dospělí pochopitelně jako věrohodnější označili obrázek 37a, zatímco celkově všichni Afričané stejně jako evropské děti za celkově lépe nakresleného slona považovali obrázek 37b. Deregowski tvrdí, že kresba typu rozdvojené perspektivy (*split-view perspective*) je univerzální a spíše vrozena než kulturně podmíněná [14]. Striktní lineární perspektiva kresby nebo fotografie totiž odporuje naší fenomenologické zkušenosti s vnímaným objektem. Při sledování prostorové scény automaticky užíváme stabilizující mechanismus percepční stálosti v nejširším slova smyslu [37].

Naše západní kultura si po staletí zvyká vnímat perspektivní zobrazení trojdimenzionální reality na ploše jako její nejméně ověřitelnější obraz. Je to jistě obraz nejméně ovlivněný kognitivní aktivitou kreslíře (například jeho znalostí objektu, schopností mentální rotace atd.), až se nabídla možnost takový „realistický“ obraz nahradit automaticky vzniklou fotografickou reprodukcí reality – jejím fotonovým otiskem. Věříme, že fotografie je čistě kauzálním důsledkem reality, kterou zobrazuje. Realistická kresba je napodobeninou rovněž respektující perspektivní zobrazení, která se maximálně přibližuje svému fotografickému ideálu [65]. Takovému obrazu dáváme přednost i přesto, že u něj vlivem projekce trojdimenzionálního obrazu na plochu sítnice dochází k deformaci a ztrátě informací o zobrazovaném objektu.

5 Realita, iluze nebo podvod

Když jsme se věnovali zrakovým klamům jako fenoménům, které jsou předmětem vědeckého zájmu, dokládali jsme, že tyto jevy nejsou jen žertovné anekdoty, ale že mohou mít význam pro hlubší vědecké zkoumání procesů vnímání. Hned v úvodní části jsme nastínili dvě svým způsobem protikladné definice percepčního klamu. Jedna vychází z představy vnímání jako procesu, jehož cílem je dosahovat co největší koherence vjemu s vnější (objektivní) realitou. V rámci této představy lze definovat percepční klam jako systematické zkreslení skutečnosti, kterému podléhají všichni pozorovatelé nebo alespoň jejich většina. Druhá definice naopak chápe percepční klam jako rozpor ve vnímání, který si pozorovatel může uvědomit. Lidský zrak, jak samotná existence klamů dokládá, není lepší, než je pro přežití člověka zapotřebí [23].

Zatímco první definice vytváří prostor pro podvádění, druhá definice ukazuje místo pro umění.

Podívejme se nyní na zrakové klamy poněkud z jiného úhlu. Kromě verbální komunikace lze i jiné kulturní jevy chápat jako jazykové projevy. Na zrakové klamy se můžeme dívat také jako na součást vizuálního jazykového kódu, kterému se věnuje sémiotika. Pro toto zkoumání pomineme situace, kdy se člověk setkává se zrakovým klamem náhodně a nikým nemotivovaně. Nebude nás tedy zajímat například měsíční klam. Pokud má zrakový klam svého autora, mohou prakticky nastat z hlediska tvůrce klamu dvě rozdílné situace:

- **neviditelný klam** - pozorovatel o zrakovém klamu nemá nebo nemusí vědět a má považovat iluzivní obraz za normální obraz objektu,
- **viditelný klam** - pozorovatel si má uvědomit rozdíl mezi objektem a jeho zkresleným vjemem.

V prvním případě se klamný vjem nabízí pozorovateli místo obvyklého vjemu objektu. Pozorovatel si této skutečnosti není vědom nebo ji nepovažuje za důležitou. Výsledné sdělení je takové, jaké autor sdělení očekával. Ve druhém případě je to samotný fakt klamu, který stojí v jádru sdělení, a pro tvůrce je důležité, aby si pozorovatel rozpor mezi objektem a iluzivním vjemem uvědomil, i když nemusí rozumět tomu, co se děje.

5.1 Neviditelný klam

Jost považuje zrakové klamy za nástroj lži v tom smyslu, že se percipovaný obraz tváří jako by byl obrazem skutečného objektu. Je to lež o ontologickém statusu viděného. Cílem prezentace zrakového klamu je podle Josta oklamat diváka, vnuknout mu přesvědčení, že stojí tváří v tvář skutečnosti [40].

Takové pojetí vyžaduje podle Eca [15] naivního neboli sémantického diváka. Divák má rozumět situaci tak, jak se mu nabízí. Do této chvíle bylo jedno, zda pozorovatel stojí o to, aby poznal skutečný stav věci a odhalil přítomnost zrakového klamu. Nyní musíme udělat toto rozlišení:

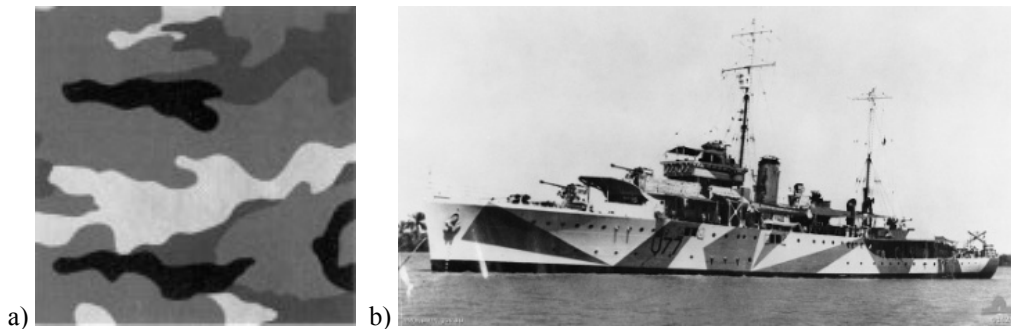
- *klam jako lež* - pozorovatel stojí o to vědět, jaká je pravá podstata objektu, na který se dívá a nepřeje si být klamán iluzivním obrazem
- *klam jako prostředník sdělení* - pozorovateli nezáleží na tom, zda vnímá pravou podstavu objektu, na který se dívá, a nezáleží mu na tom, zda je „obětí“ iluze nebo nikoli. Objekt se díky zrakovému klamu jeví pozorovateli tak, jak to autor i příjemce potřebují.

Zde se v prvním případě stává člověk obětí lži a cílem klamu je, aby byl autorův záměr klamat naplněn. Na druhé straně zde může nastat situace, kdy zrakový klam koriguje určité nedokonalosti lidského vnímání a umožňuje tak pozorovateli vnímat z jeho lidské perspektivy objekt takový, jaký by měl být, nebo jaký by si jej vidět přál. Zde můžeme jen obtížně mluvit o lži.

Zrakový klam jako lež

Zrakový klam může lhát o povaze vnímaného objektu. Cílem v takovém případě je, aby si toho pozorovatel nebyl vědom nebo aby s tím nemohl nic dělat. Zároveň je potřeba, aby divák nepoznal skutečný stav věci a nemohl učinit účinná opatření alespoň do chvíle, než udělá na základě klamných přesvědčení určité, pro autora důležité, rozhodnutí. Takové klamy jistě existují. Například Fejtek uvádí, že experimenty s reakcí zákazníků na různě osvětlené potraviny vedly ke vzniku pojmu akcentního osvětlení pro tyto produkty [19]. Například spektrální charakteristiky osvětlení mohou měnit dojem, jaký v nakupujícím vyvolávají potraviny – ty se pak zdají čerstvé a svěží. To může třeba v řeznictví hraničit s klamáním zákazníků [57]. Je zajímavé, že v těchto případech jde spíše o optické klamy, než o klamy, které vznikají v průběhu vnímání.

Jiným případem zrakového klamu, který je použit jako lež, a to někdy v pravdě proti zájmům pozorovatele, je vojenská kamufláž. Cílem kamufláže je skrýt viděný objekt tak, že je figura, která jej ve zrakovém poli percipienta reprezentuje, rozbita a rozmělněna do struktury pozadí. Kamufláž má tedy nejen stejný barevný odstín jako pozadí, se kterým má objekt splynout, ale rovněž obsahuje dostatečně výrazný vzor (který je zároveň v jistém souladu s charakterem pozadí). Tento vzor má rozbít kontury objektu. Proto někoho překvapí, jak výrazně se kontrastní vzory uplatňují u moderních vojenských uniforem a nátěrů bojové techniky (viz obrázek 38a).



Obr. 38. Moderní vzor vojenské kamufláže a námořní Dazzle kamufláž z roku 1941[4].

V námořnictví se ovšem dělaly také pokusy s tím, jak může kamufláž zmást protivníkovu schopnost odhadovat směr plavby lodi a její rychlost. Za první a druhé světové války se k tomuto účelu používala tzv. Dazzle (Razzle-Dazzle) kamufláž vytvořená v roce 1917 Britským námořním důstojníkem a výtvarníkem Normanem Wilkinsonem (obr. 38b). Šlo o komplexní vzor geometrických tvarů v kontrastních barvách, který měl zabránit nepřátelskému pozorovateli, aby percepčně uchopil celou siluetu lodi a dokázal z ní odvodit potřebné údaje [4].

Klamáním o povaze percipovaného objektu lze nazvat též vzory ochranného zbarvení živočichů. Nejde jen o pouhé splynutí s pozadím. Například motýli vytváří vzorem svých křídel dojem, že jde o obličej mnohem většího tvora. Některé druhy ryb mají na zadní části „namalované“ oko, které pro číhajícího predátora vytváří iluzi, že je ryba otočená na druhou stranu a plave opačným směrem. Podobných příkladů existuje ohromné množství. V případě přírodních mimikry ovšem chybí autor, který vědomě lež vytvořil.

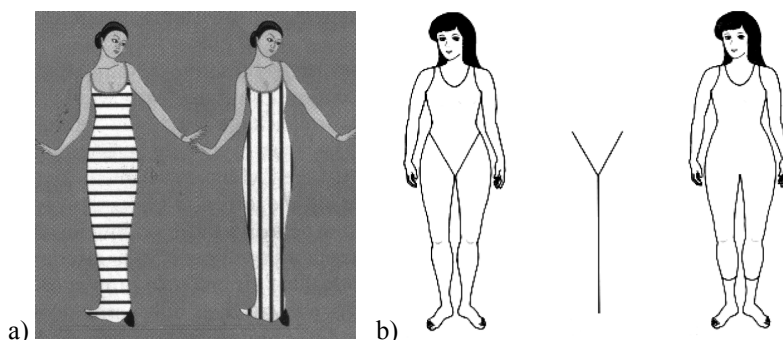
Klam zprostředkovává sdělení

Někdy je pozorovatel v nevýhodné pozici a není schopen ze svého stanoviště správně přečíst význam sdělení. Tak tomu může být i ve zcela praktických případech. Například vodorovné dopravní značení může řidič sledovat jen z poměrně kosého úhlu. Piktogramy nebo nápisy na vozovce jsou schválně deformovány tak, aby byl jejich význam z řidičova stanoviště dobře rozpoznatelný a čitelný. Ze stejného důvodu mívají například vozy záchranné služby v některých zemích vepředu zrcadlově obrácený nápis „AMBULANCE“, aby bylo toto sdělení dobře čitelné pro řidiče před sanitkou, který může nápis číst pouze ve zpětném zrcátku.

V praxi rovněž může nastat situace, kdy je potřeba zajistit čitelnost informace, která nemá optimální podobu. První verze dopravního informačního systému na české dálnici D1 pracovala s bílou plochou, na které se informační text skládal z černých pixelů. Taková tabule byla dobře viditelná i za horších světelných podmínek, ovšem text byl čitelný špatně. Teprve inverzní uspořádání umožnilo, aby se pixely vlivem iradiace „slily“ do znaků písma. Nyní jsou proto informační tabule černé a nápis je skládán z bílých pixelů.

Těmito „triky“ se dosahuje v dopravě toho, aby řidič získal co nejrychleji a bez zbytečného úsilí informace, které potřebuje k řízení. Všimněte si, že právě v silniční dopravě se hodně dbá na to, aby se kompenzovala určitá nepřírozenost prostředí, se kterou se řidiči musí vyrovnávat: nepřírozeně velké rychlosti, poloha blízko při zemi, užívání zrcadlového odrazu apod.

I jindy je potřeba v realitě kompenzovat nedokonalost lidského vnímání, která se projevuje tehdy, pokud je prostředí, ve kterém se člověk nachází, pro něj v nějakém směru nepřírozené. V architektuře to mohou být rozměry budov, které překračují lidské měřítko. Již antičtí Řekové schválně deformovali překlady svých chrámů. Rovněž sloupy antických chrámů nebyly postaveny čistě vertikálně. Teprve odchylkami od horizontál a vertikál stavby dosáhli Řekové dojmu geometrické dokonalosti. Korekce byla nutná, protože z pohledu by se stavba jevila lidem deformovaná. V sochařství se u soch v nadživotní velikosti užívá korekce, která proporčně zvětšuje horní části těla na úkor spodních partií. Vychází se z předpokladu, že socha bude sledována z výrazného pohledu, který odporuje lidské zkušenosti s vnímáním lidských postav. Tak je tomu například u sochy Davida, kterou Michelangelo vytvořil podle antických kánonů [68]. Tyto postupy slouží ke korekci nedokonalosti lidského vnímání, či spíše nějaké nevýhodnosti postavení, ze kterého je pozorovatel nucen objekt vnímat. Jejich cílem není člověka oklamat, ale naopak mu pomoci lépe zprostředkovat sdělení, o které v daném případě jde (geometrická čistota, elegance tvaru lidského těla nebo věcný obsah dopravního značení).



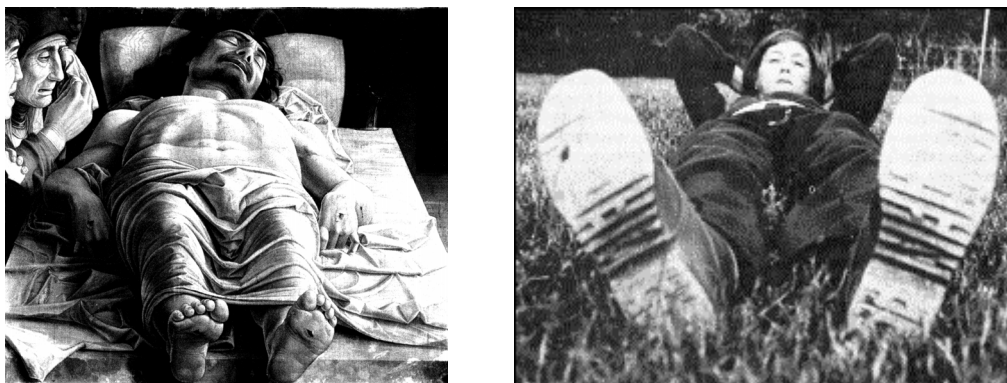
Obr. 41. Využití horizontálně vertikálního a Müller-Lyerova klamu v odívání [56].

Často se mluví o používání mechanismů zrakových klamů v odívání. Známý je například efekt horizontálních a vertikálních linií na dojem štíhlosti postavy (obr. 41a). Stejně jako make-up, slouží i takové aplikace spíše ke zdůraznění existujících vlastností než k zakrytí jiných.

Efekt „dlouhých nohou“ u vykrojených dámských plavek bývá přikládán účinku Müller-Lyerova klamu (obr. 41b). Otázkou zůstává, zda jde v tomto případě o klam, nebo se nohy dámy ve vykrojených plavkách zdají delší prostě proto, že jsou jejich kontury vytaženy k pasu [56].

Manipulace s percepčními procesy pozorovatele může sloužit k vyjádření určité skutečnosti, která má být součástí uměleckého záměru. Tak například renesanční malíř

Andrea Mantegna (1431-1506) kompenzoval perspektivní zkreslení u obrazu mrtvého Krista metodou tzv. perspektivní zkratky (obr. 42). Proč použil takové vyobrazení, jakoby neznal pravidla perspektivy? Udělal to proto, aby zachoval důstojnost výjevu.



Obr. 42. Andrea Mantegna - Mrtvý Kristus (vlevo), fotografie přibližně stejné konfigurace lidské postavy a pozorovatele (vpravo) [37].

V sociálních vztazích je totiž důležitým vodítkem sociální distance. Sociální distance má vztah k reálné vzdálenosti dvou lidí v interakci. Ve vizuálním umění hraje roli formálního parametru tzv. zjevná blízkost (*apparent proximity*). Například v případě filmové formy se rozlišuje značná blízkost kamery, která indikuje intimní nebo osobní mód, střední vzdálenost kamery reprezentuje sociální mód a vzdálené umístění kamery indikuje neosobní mód filmového sdělení [37]. Ve fotografických nebo filmových obrazech, ale také na malbách a grafikách se výrazná blízkost objektu projevuje specifickým perspektivním zkreslením obrazu. Mantegnou použitá metoda směřuje k ortografickému vyobrazení. Ortografické zobrazení vykresluje objekty jako by byly pozorovány z nekonečné vzdálenosti. Naopak použití perspektivy do obrazu vnáší stanoviště pozorovatele, které je dáno již ne jen orientací objektů, ale také vlastní vzdáleností pozorovatele od nich. Použití perspektivy vlastně vnáší do obrazu subjektivní hledisko pozorovatele, zatímco ortografické zobrazení si zachovává odosobněnou formu.

V teorii výtvarného umění existuje problém, proč středověká Evropa přestala používat poznatky perspektivy, ačkoli k nim měla přístup. Jedno z vysvětlení by mohlo znít tak, že zavedení perspektivy by středověkým umělcům překáželo v jejich pokusu zachycovat neosobní skutečnosti. Renesanční příklon k lineární perspektivě, která postuluje pozici pozorovatele, vnáší do výtvarných sdělení ideologii subjektivního pohledu [37].

McLuhan [37] vnímá fotografii jako mechanizaci lineárně perspektivní kresby. Je to mocný nástroj, který vytváří iluzi, že fotografie je „oknem do světa“. Ve skutečnosti ale fotografie (stejně jako malířský obraz nebo grafika) nezachycuje to, co může člověk skutečně vnímat. Lidské oko totiž není uvězněné do jediného pohledu – na svět

se nedíváme statickým kukátkem. Oči provádí sakadické pohyby, kterými prohledávají scénu, též naše tělo je v pohybu pro optimální dosažení cíle vnímání [67].

5.2 Viditelný klam

Množství zrakových klamů je divákovi předkládáno bez snahy něco skrývat. Existuje řada publikací, které o sobě po pravdě prohlašují, že zrakové klamy obsahují. V tomto případě jsou zrakové klamy prezentovány jako hra. Jejich iluzivní účinek je ludickým předstíráním. I v tomto případě má podle Eca smysl hovořit o pravdě a nepravdě, ale takto prezentované klamy si přímo žádají, aby byla jejich zkreslená percepce srovnávána se skutečností. V tomto pojetí je zrakový klam estetickým produktem a pozorovatel si vychutnává sílu jeho účinku srovnáváním iluzivního vjemu se skutečností [15].

V rámci objektivní skutečnosti jsme si zvykli chápat objekty tak, že jsou právě jen tím, čím jsou. V jazyce a ve všech kulturních sférách ovšem mohou objekty nabývat různých významů podle nejrůznějších pravidel. V kulturní sféře se objekty stávají znaky. No a znak, jak píše Monaco, když cituje francouzského filozofa a eseistu Brice Praina, nás nutí vidět objekt skrze jeho význam [55]. Procesu vnímání tedy můžeme rozumět i tak, že se podnětům přiřazují významy. Zrakový podnět pak může znamenat víc věcí najednou nebo to či ono podle potřeby. Může i sám sebe popírat, popřípadě může význam vyrůstat z mezery mezi podněty. Gregory [27] se pokusil některé typy zrakových klamů připodobnit k jazykovým chybám, my se dále pokusíme sledovat zrakové klamy jako nositele významu.

Zrakový klam jako šifra

Jednoduchým způsobem využití zrakových klamů je jejich užití jako klíče k žertovné šifře. Sám o sobě je objekt nečitelný, ale začne dávat smysl, když se na něj aplikuje „chybná“ percepce.

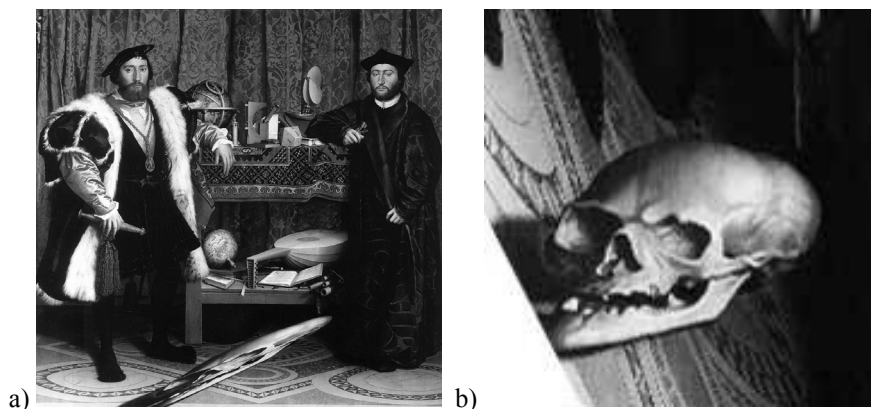


Obr. 39. Předloha pro paobraz tváře Krista.

Tímto způsobem se dá využít stínohry nebo efektu paobrazu. Vychází se z toho, že inverzní zobrazení nějakého objektu není k poznání (obr. 39). Skvrny nedávají smysl, ale pokud se na ně pozorovatel dívá nehnutě několik sekund a pak se podívá na

bílou plochu, spatří tvář muže. Oblíbené byly a jsou právě takováto vyobrazení lidské tváře, protože v inverzní podobě často nelze rozpoznat, že jde o tvář, natož určit, komu podobizna patří. Rybár si navíc všiml, že předloha na obrázku 39 obsahuje drobné skvrnky, které u originálu mohou ztěžovat identifikaci tváře. Experimentoval s předlohou stoličky, kterou schválně opatřil drobným „šumem“, který se v paobrazu nevyskytoval [61]. Tento šum umožňuje využít efektu paobrazu jako šifry.

Další rodinou technik, které bývají mezi zrakové klamy zahrnovány, jsou anamorfy. Jde o obrázky, které začínají dávat smysl při určitém zkreslení. Jsou-li viděny tak, jak jsou fyzicky vyvedeny, není jejich obsah identifikovatelný, nebo obrázky znázorňují něco jiného. Nejslavnějším a snad jedním z prvních anamorfů je obraz renesančního malíře Holbeina z roku 1533. O tomto obraze, který je nazýván *Ambasadoři* (obr. 40a), se stále vedou nekonečné debaty. Na obraze je mezi dvěma muži kobereček s poněkud divným vzorem. Teprve, pokud se na obraz díváme z určitého úhlu, promění se vzor ve vyobrazení lebky (obr. 40b).



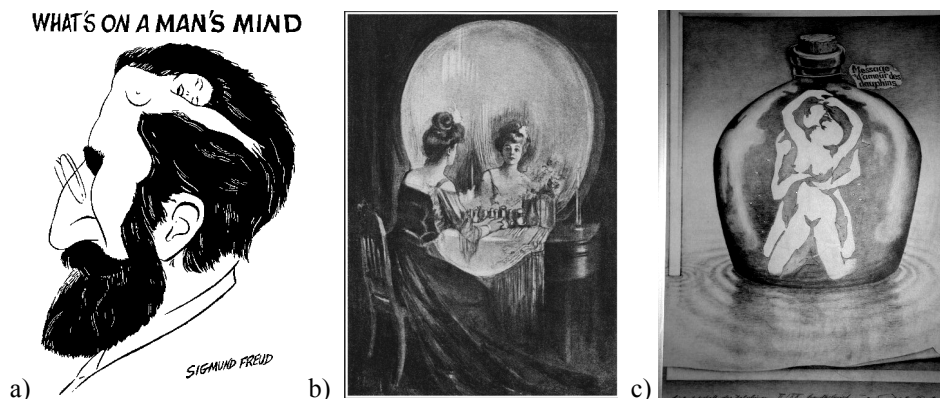
Obr. 40. Holbeinův obraz *Ambasadoři* a detail anamorfní lebky

Použití zrakového klamu jako šifry v praktickém životě se pochopitelně zdá sporné. Každý člověk má vlastně k dispozici klíč, pouze ne každý jej umí využít. Úprava pomocí anamorfní „šifry“ však může nést svůj specifický význam. V Nekesově sbírce [52] jsou ukázky pornografie z 18. století skryté pomocí anamorfy. Míra „zašifrování“ je v tomto případě prakticky nulová. Takový, vpravdě literární, případ šifry se podobá například rodičovskému zámku, kterým jsou vybaveny televizní přijímače: je to standard, ale málo kdo to používá. Říká se tím, že určité sdělení má zůstat skryto spíše, než že by toto sdělení bylo skutečně ukrýváno. Použití zrakových klamů jako šifry je spíše alegorií, než pokusem o ukrytí skutečného významu před náhodnými nebo nepovolanými percipienty.

Zrakový klam zprostředkovává další významy

Většina zrakových klamů je předkládána v takovém kontextu, že si pozorovatel může být vědom rozdílu mezi tím, co vidí, a tím, na co se ve skutečnosti dívá. Může si tak zakusit vlastní aktivitu v procesu vnímání. Příkladem mohou být reverzibilní figury.

Podobný účinek jako Rubinova reverzibilní figura (obr. 3) mají i složené obrazy nazývané *puzzle pictures*. Na obrázku 43a je karikatura Sigmunda Freuda, která zároveň s portrétem zakladatele psychoanalýzy odkazuje na jeho pansexuální motivační teorii. Nápis „co je v mužské mysli“ navíc ironicky spojuje oba významy, které kresba poskytuje, tedy teorii a jejího autora.



Obr. 43. Puzzle pictures.

Americký časopisecký ilustrátor Charles Allan Gilbert vytvořil roku 1892 klasický obrázek pod názvem „Všechno je marnivost“ (43b). Grafik na toto téma pak vzniklo značné množství. Nekesova kolekce [52] obsahuje několik francouzských variant těchto takzvaných *puzzle pictures*. V každém případě se tento morbidní motiv stal velmi populárním a byl mnohokrát použit¹¹. Tím, že jeden zrakový podnět dokáže vyvolat dva v určitém směru protikladné významy (krása – zmar) dostává tyto protiklady do vzrušujícího vztahu – přes jejich vzájemnou protikladnost zařizuje jejich syntagmatickou blízkost.

Interpretační mohutnost našeho vnímání může oscilovat podle našich kompetencí, motivací, záměrů a potřeb. Obrázek 43c je doprovázen výzvou, aby pozorovatel spočítal, kolik delfínů je v láhvi. Tento příklad nicméně není zcela poctivý, protože existují i neerotické důvody, proč člověk snáze spatří objetí nahých těl, než hejno kytovců.

Rob Gonsalves je jeden ze současných mistrů iluzivní malby a je řazený do škatulky magického realismu. Tento autor vytvořil mimo jiné obraz s názvem „Bedtime Aviation“ (obr. 44), který užívá techniku nekonzistentní scény ke zpřítomnění fantazie dětských her [63].

¹¹ Například zpěvačka Cher na desce Heart of Stone 1989.



Obr. 44. Rob Gonsalves: Bedtime Aviation, 2001 [63].

Gregory [29] rozlišuje pasivní a aktivní víceznačnost. Zatímco pasivní víceznačnost je důsledkem selhání možné interpretace vjemu, aktivní víceznačnost spočívá v nalezení více možných významů. Aktivní víceznačnost je pro Gregoryho základ poezie.

Zrakový klam podtrhuje význam objektu

Renesanční, ale hlavně barokní, iluzivní perspektivy jsou založené na omezeních, která sdílí konstrukce lineární perspektivy s fotografií. Pozorovateli je vyhrazeno jedno ideální místo, odkud má iluzivní scénu sledovat. To platí například pro fresky označované jako *trompe l'œil*, jako jsou slavná díla architekta a dekorátora Andrea Pozza. U nás se rovněž hojně uplatňoval tento „trik“ při barokních výmalbách kleneb kostelů (např. Křtiny) a jiných prostor (Strahovský klášter). Tyto fresky vytváří dojem celkového protažení prostoru do výšky většinou pomocí zobrazování dalších fiktivních pater místnosti. Cílem *trompe l'œil* výzdoby není zmást návštěvníka, ale spíš podnítit jeho fantazii a ukázat na sémantický (většinou duchovní) význam místa [68]. Pozzova výzdoba kostela Nanebevzetí svatého Ignáce v Římě například znázorňuje právě tuto mysteriózní událost (obr. 45a). Většina těch nejrafinovanějších *trompe l'œil* vyžaduje, aby stál pozorovatel na určitém ideálním místě. Mimo toto místo dochází ke zkreslení *trompe l'œil* obrazu a k rozpadu iluze.

Při dostatečném zjednodušení konstrukce takové klamavé malby a při určitém uvolnění pravidel lineární perspektivy lze dosáhnout uspokojivého efektu *trompe l'œil* i pro širší okruh míst, v rámci kterých se může pozorovatel pohybovat. Tímto způsobem dnes bývají vymalovány prázdné štíty domů v městských prolukách. Moderní použití *trompe l'œil*, které si neklade nároky na přesné stanoviště pozorovatele, sice není tolik efektní jako některé jiné kreace tohoto druhu, zato je ale určitým způsobem „demokratičtější“, protože z iluze se může těšit každý kolemjdoucí.

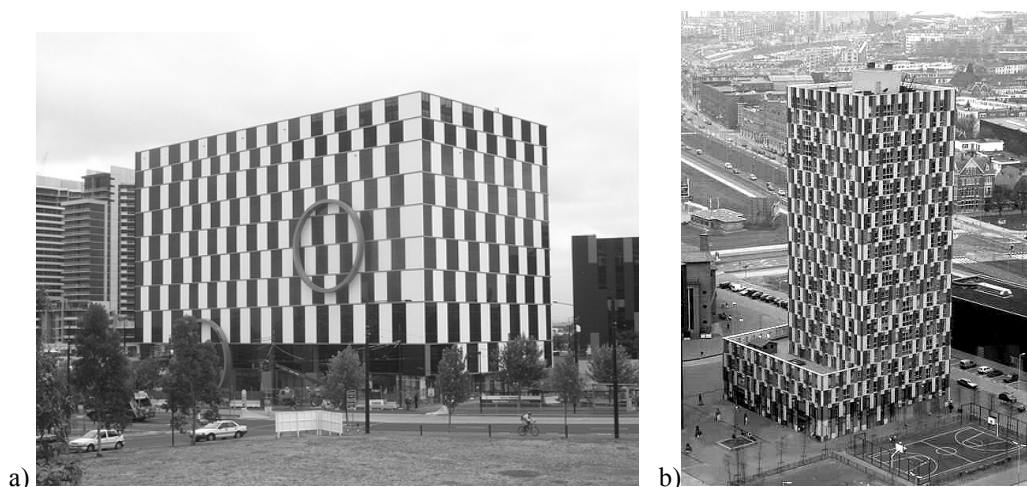


Obr. 45. a) Andrea Pozzo: Nanebevzetí sv. Ignáce v Římě, bílé čáry přibližně znázorňují skutečné hrany klenby, b) nenáročný trompe l'œil domovního štítu - Žerotínovo náměstí v Brně.

Tak, jako představují domovní štítů bez oken prázdnou plochu, která si žádá o nějaké ozvláštnění, jsou tu celé hmoty skladů a jiných technických budov, jejichž fasády není důvod jakkoli členit, ale přesto existuje snaha nenechat je ve funkčně čisté podobě. Na obrázku 46a je příklad použití *Caffe Wall* klamu pro ozvláštnění fasády skladu přístaviště v Melbourne. Budova v Rotterdamu (obr. 46b) využívá pro změnu *striped cord* klamu, díky kterému se jeví jako rozšiřující se směrem vzhůru. Kdo bude mít možnost dívat se na ni z úrovně terénu, mohl by zažít sekundární efekt tohoto klamu. Tato budova by se totiž z pohledu mohla jevit jako vyšší.

Těmito technikami lze v architektuře relativně levně dosáhnout neobvyklého dojmu nebo dokonce změny vnímané hmoty budovy. Architektura ovšem používá častěji jemnější prostředky, jako je práce s rytmem, kontrastem, užíváním horizontálních a vertikálních linií, popřípadě manipuluje s vnímaným měřítkem budov změnou velikostí detailů obvyklých rozměrů.

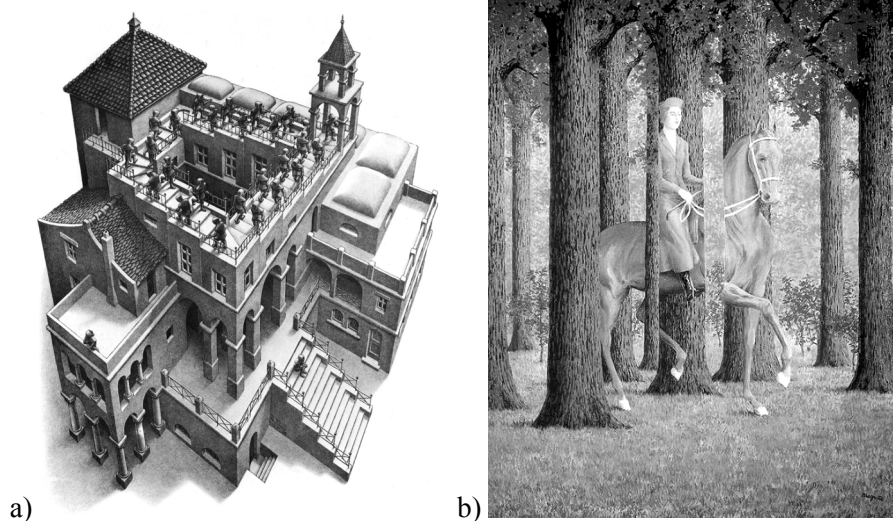
Využití specifčnosti zrakového aparátu tak může sloužit ke změně nebo naopak posílení určitých, často symbolických, vlastností architektonických objektů. Například manipulace s měřítkem staveb vede k posílení jejich vnímané a tedy i symbolické velikosti. Tak je tomu například u kolonády zámeckého komplexu Schönbrunn ve Vídni nebo u hřbitovní kaple Zelená hora ve Žďáru nad Sázavou od architekta Santiniho. Posílení symbolické velikosti budovy proto nemusí nutně znamenat více cihel [68].



Obr. 46. Budova portu 1010 na přístavišti v Melbourne (Austrálie) a budova v Rotterdamu.

Zrakový vjem popírá sám sebe

Nemožný svět ukazuje sám sebe pomocí důvodů, proč nemůže existovat. Nemožný svět není nemyslitelný, ale je sám podmínkou své nemožnosti. Je tomu tak u nemožných objektů typu Penroseova trojúhelníku stejně jako u některých literárních děl [15].



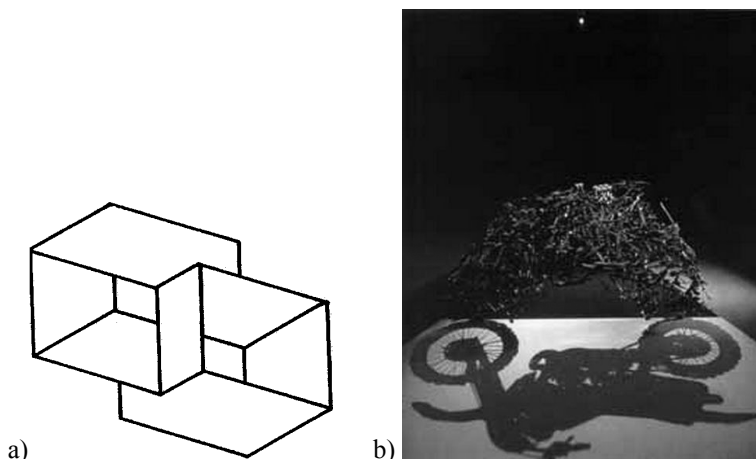
Obr. 47. M. C. Escher: Ascending and Descending, R. Magritte: Le blanc-seing.

Klasickým příkladem vyjádření absurdní skutečnosti je použití absurdních objektů a scén. Grafiky M. C. Eschera (obr. 47a) mají zneklidňující a provokativní nádech. Nutí člověka, aby je znova a znova obsedantně analyzoval a přišel věci na

kloub. Svoji absurditu skrývají pod závojem hyperrealistického vyvedení. Naopak malby nekonzistentních scén belgického surrealisty René Magritta (např. obr. 47b) svoji nemožnost nikterak neskrývají a divák si může jejich nesmyslnost užít přímo.

Všimněte si, že Gonsalvesův obraz (obr. 44) se pomocí nekonzistentní scény snaží evokovat více významů najednou. Vnímání nemožných objektů a scén totiž, jak tvrdí Gregory [29], úzce souvisí s víceznačností ve vnímání (obr. 48). Je to právě proto, že nekonzistence vjemu svádí lidskou kognici najednou do několika interpretačních směrů.

Deregowski zjistil, že lidé, kteří jsou připraveni liniovou kresbu identifikovat jako trojrozměrný objekt a kteří proto podlehnou zmatení z kresby nemožného objektu, jsou většinou příslušníci západního kulturního okruhu. Monaco to zdůvodňuje skutečností, že trojrozměrné čtení takové kresby vyžaduje znalost určitého kódu, který používáme pro znázorňování prostorových skutečností. Tento kód je kulturně podmíněný a není tedy přirozený. To se netýká jen liniových kreseb, ale také schopnosti číst realistické obrázky, fotografie nebo film [55].



Obr. 48. a) Doklad víceznačnosti nemožných objektů [29] a objekt Shigea Fukudy „Lunch with a Helmut On“ 1987 [63].

Shigeo Fukuda vytváří stínové obrazy objektů i lidí tak, že nechá světlo procházet přes hromadu různých věcí. Hromada na obrázku 69 zdánlivě vypadá jako seskupení posvařovaných vidliček, nožů a lžiček bez ladu a skladu, ovšem ve skutečnosti je rafinovaně organizovaná tak, aby stínohra, kterou vrhá díky speciálně umístěnému zdroji světla, dávala jednoznačnou interpretaci.

Stín odkazuje ke svému objektu tak, že je stín objektem vržen. Vztah mezi objektem a jeho stínem je tedy kauzální. Z hlediska sémiotiky jde o indexový znak. Podle naší praktické zkušenosti nepochybujeme, jaký objekt tento stín vrhá (v našem příkladě motocykl). Pro úplný (a vlastně jakýkoli) zážitek z Fukudova díla je nezbytně nutné, aby pozorovatel zároveň věděl, z jakého reálného objektu stínohra vzniká. Tato

znalost mu umožňuje zakusit provokující nepatřičnost interpretace, kterou si na základě indexového vztahu mezi stínem a objektem vytvořil.

Když zrakový vjem chybí

Když něco chybí, může to být i tak svým způsobem přítomno. Uvědomujeme si, že to chybí, a tím právě o tom víme. V případě Kanizova klamu (obr. 6) percipient vnímá objekt, který není na obrázku přímo zachycen. Obrazně řečeno, čte mezi řádky.



Obr. 49. Lothar Wolleh: portrét René Magritta.

Fotografie René Magritta od německého fotografa Lothara Wolleha (obr. 49) ukazuje slavného surrealistického malíře tím, co není vidět z drapérie záclon. Magritte nám zde o sobě říká právě to, co fotografie neukazuje.

V podobné situaci je člověk, který interpretuje bezsmyslné podněty, například hledá siluety zvířat a tváří ve tvaru oblaků nebo interpretuje skvrny Rorschachova testu. V tomto případě percipient vnímá podnět, kterému chybí smysl, a je na vnímajícím, aby si smysl pro sebe našel.

5.3 Věříme obrázkům?

Představme si, že nám chce někdo sdělit, že viděl slona z ptačí perspektivy, a tuto informaci nám podává obrázkovou formou. Může použít fotografii, nebo realistickou perspektivní kresbu anebo kresbu s rozdvojenou perspektivou (obr. 37). Fotografie jsme dříve považovali za index skutečnosti, něco jako nepřímý důkaz. Realistická perspektivní kresba je ikonickou napodobeninou fotografie a byla dříve vnímána jako méně věrohodná. Naivní kresbu s rozdvojenou perspektivou bychom měli tendenci vnímat jako zcela nevěrohodnou. Situace se ale v dnešní době mění. Vlivem možnosti formálně dokonalých digitálních úprav fotografií klesá věrohodnost fotografií a naopak opět stoupá věrohodnost očitých svědectví [40]. Obrázek slona 37b dokumentuje právě nejlépe takové očitě svědectví: kreslíř dokládá, že objekt identifikoval, předpokládal u něj i vlastnosti, které mohly zůstat v konkrétním okamžiku neviditelné (např. slonovy

nohy, které nejsou vidět, když je slon fotografován nebo kreslen shora) a stojí si za tím, že viděl slona a nikoli šedou plachtu přehozenou přes autobus.

6 Závěr

V první části příspěvku jsme představili zrakové klamy jako jevy, které nám mohou říci něco o povaze lidského vnímání a jsou proto předmětem vědeckého zájmu. Seznámili jsme se s problémy definice zrakových klamů a jejich klasifikací, která se snaží postihnout jak jevovou stránku, tak i teoretické mechanismy, které jsou zodpovědné za iluzivní účinek klamů.

Ve druhé části jsme představili kognitivní teorie klamů. Zabývali jsme se percepčními strategiemi, které hledají co možná nejjednodušší a zároveň dosavadní zkušenosti blízké interpretace vnímaného zrakového podnětu. Seznámili jsme se s klasickou teorií, která nahlíží na geometrické zrakové klamy jako na misinterpretaci plošných útvarů jako součástí prostorové scény, a dotkli jsme se problémů, do kterých se tato teorie v jednotlivých případech dostává.

Dále jsme se věnovali přínosu neurologického výzkumu ke studiu zrakových klamů. Neurologický, respektive fyziologický, přístup zkoumá určitou skupinu klamů jako projev chybného zpracování zrakových podnětů.

Zamysleli jsme se nad tezí o zrakových klamech jako chybné percepce a pokusili jsme se ukázat, jak relativní může být takový pohled vzhledem ke každodenním požadavkům, kladeným na vnímání zejména v součinnosti s jednáním.

Pokusili jsme se podívat na zrakové klamy jako na nositele významu a ukázali jsme příklady takového použití klamů v lidské kultuře a umění.

Kromě vědeckého zájmu o zrakové klamy jsme však ve čtenáři chtěli vzbudit též zájem o jejich estetickou hodnotu. Představili jsme zrakové klamy nejen jako kuriozity, ale jako výrazové prostředky v umělecké tvorbě.

Pro další zážitky a studium...

V dnešní době vychází řada publikací, které prezentují zrakové klamy v jejich populární podobě i jako výtvarná díla. V popularizaci vyniká značka Illusion works¹², jejímž představitelem a autorem mnoha sbírek zrakových klamů je Al Seckel. Důležitým zdrojem poznatků o zrakových klamech na internetu jsou rovněž velmi obsáhlé stránky Michaela Bacha¹³, kde je i mnoho dalších odkazů pro studium jednotlivých klamů. Na stránkách Richarda Gregoryho¹⁴, který je doyen výzkumu zrakových klamů, naleznete mnoho textů od tohoto autora. Chcete-li se hlavně dívat, lze doporučit spolu s Alem Seckelem *Eyetricks.com*¹⁵ nebo *Mighty Optical Illusion*¹⁶ a další internetové galerie. Své stránky má plodný tvůrce originálních klamů a vědec

¹² <http://www.psychologie.tu-dresden.de/i1/kaw/diverses%20Material/www.illusionworks.com/>

¹³ <http://www.michaelbach.de/ot/>

¹⁴ <http://www.richardgregory.org/>

¹⁵ <http://www.eyetricks.com/illusions.htm>

¹⁶ <http://www.moillusions.com/>

v této oblasti Akiyoshi Kitaoka¹⁷ a množství původních klamů lze nalézt na stránkách soutěže o nejlepší zrakový klam Best *Visual Illusion of the Year*¹⁸. Existuje ovšem nespočet dalších zdrojů. Příjemnou zábavu.

Poděkování: Tato kapitola vznikla za podpory grantové agentury GA ČR v rámci grantového projektu GA406/09/2003 s názvem Změny ve vnímání a paměti při dlouhodobé izolaci.

Literatura

- [1] Attneave F.: How Do You Know? *American Psychologist* **7** (1974) 493-499.
- [2] Attneave F.: Some Informational Aspect of Visual Perception, *Psychological Review* **3** (1954) 183-193.
- [3] Bach M.: *Optical Illusions & Visual Phenomena* (stránky jsou dostupné na <http://www.michaelbach.de/ot/>).
- [4] Behrens R. R.: *Camoupeedia* (web je dostupný na <http://www.bobolinkbooks.com/Camoupeedia/DazzleCamouflage.html>)
- [5] Blackemore C.: The baffled brain, in: Gregory R. L., Gombrich E. H. (eds.): *Illusion in Nature and Art*, GB, Scribners (1973) 9 – 48.
- [6] Bruner J. S. & Minturn A. L.: Perceptual identification and perceptual organization. *Journal of General Psychology* **53** (1955) 21-28.
- [7] Bruno N.: When does action resist visual illusions? *TRENDS in Cognitive Sciences* **5** (2001) 349-382.
- [8] Buffart, H. F. J. M., Leeuwenberg, E. L. J., Restle, F.: Coding theory of visual pattern completion, *Journal of Experimental Psychology: Human Perception and Performance* **7** (1981) 241-274.
- [9] Clément G., Hamilton D.: *Fundamentals of Space Medicine*. Springer a Microcosm Press 2005.
- [10] Corney D., Lotto R. B.: What Are Lightness Illusions and Why Do We See Them? *PLoS Comput Biol* **3(9)** 2007.
- [11] Cuijpers R. H., Kappers A. M. L., Koenderink J. J.: Large systematic deviations in visual parallelism, *Perception* **29** (2000) 1467-1482.
- [12] Da Pos O.: Visual illusions and effects. In Sivik, L. (Ed.), *AIC 1996 - Colour and Psychology*, Scandinavian Colour Institute AB, Stockholm, 1996, 114-118.
- [13] Deregowsky J. B.: Real space and represented space: Cross-cultural perspectives, *Behavioral and Brain Sciences* **12** (1989) 51-119.
- [14] Deregowski J. B.: Illusion and Culture, in: Gregory R. L., Gombrich E. H. (eds.): *Illusion in Nature and Art*, GB, Scribners (1973) 161 – 192.

¹⁷ <http://www.ritsumei.ac.jp/~akitaoka/index-e.html>

¹⁸ <http://illusioncontest.neuralcorrelate.com/>

- [15] Eco U.: *Meze interpretace*, Karolinum, Praha 2005.
- [16] Eco U.: *A Theory of Semiotics*. Indiana University Press, Bloomington 1979.
- [17] Eppstein W.: Percept-percept Couplings, *Perception* **11** (1982) 75 – 83.
- [18] Eysenck M. W., Keane M. T.: *Kognitivní psychologie*. Praha: Academia 2008.
- [19] Fejtek M.: Osvětlení pro čerstvé potraviny, *Světlo* **04** (2002) (dostupné na http://www.odbornecasopisy.cz/index.php?id_document=23000).
- [20] Franz V. H.: Action does not resist visual illusions, *Trends in Cognitive Sciences* **5/11** (2001) 457-459.
- [21] Franz V. H., Bühlhoff H. H., Fahle M.: Grasp effects of the Ebbinghaus illusion: Obstacle-avoidance is not the explanation, *Technical Report No. 091. Max Planck Institute for Biological Cybernetics* 2002. (dokument je dostupný na <http://www.kyb.mpg.de/publications/pdfs/pdf1624.pdf>).
- [22] Gilinsky A. S.: Perceived size and distance in visual space, *Psychological Review* **58** (1951) 460-482.
- [23] Gillam B.: Illusions at Century's End, in: Hochberg J. (ed.): *Perception and Cognition at Century's End SE*, Academic Press, USA (1998) 95 – 136.
- [24] Gregory R. L., Heard, P.: Border locking and the Café Wall illusion, *Perception* **8** (1979) 365-380.
- [25] Gregory R. L.: The confounded eye, in: Gregory R. L., Gombrich E. H. (eds.): *Illusion in Nature and Art*, GB, Scribners (1973) 49 – 96.
- [26] Gregory R. L.: Emmert's Law and the moon illusion, *Spatial Vision* **21** (2008) 407-420.
- [27] Gregory R. L.: Knowledge in perception and illusion. *Phil. Trans. R. Soc. Lond. B* **352** (1997) 1121-1128.
- [28] Gregory R.: Brainy mind, *Brit. Med. Journal* **317** (1998) 1693-1695.
- [29] Gregory R. L.: Ambiguity of 'ambiguity', *Perception* **29** (2000) 1139 – 1142.
- [30] Gregory R. L.: Perceptual illusions and brain models. *Proc. Royal Society B* **171**, 179-296.
- [31] Gregory R. L.: *Seeing Through Illusions*, NY: Oxford UP 2009.
- [32] Haffenden A. M., Schiff K. C., Goodale M. A.: The dissociation between perception and action in the Ebbinghaus illusion: Nonillusory effects of pictorial cues on grasp, *Current Biology* **11** (2001) 177-181.
- [33] Hochberg J.: Acts of perceptual inquiry: problems for any stimulus-based simplicity theory, *Acta Psychologica* **114** (2003) 215-228.
- [34] Howe C. Q., Purves D.: Range image statistics can explain the anomalous perception of length, *PNAS* **20** (2002) 13184-13188.
- [35] Howe C. Q., Purves D.: The Müller-Lyer illusion explained by the statistics of image-source relationships, *PNAS* **4** (2005) 1234-1239.
- [36] Chandler D.: *Semiotics: The Basics*, London: Routledge 2001.

-
- [37] Chandler D.: *Semiotics for Beginners* (stránky dostupné na <http://www.aber.ac.uk/media/Documents/S4B/semiotic.html>).
- [38] Changizi M. A., Hsieh A., Nijhawan R., Kanai R., Shimojo S.: Perceiving the present and a systematization of illusions, *Cognitive Science* **32** (2008) 459 – 503.
- [39] Chater N.: Reconciling Simplicity and Likelihood Principles in Perceptual Organization, *Psychological Review* **3** (1996) 566-581.
- [40] Jost F.: *Realita/Fikce - říše klamu*, Akademie múzických umění, Praha 2006.
- [41] Kitaoka A.: Apparent contraction of edge angles, *Perception* **27** (1998) 1209-1219.
- [42] Kitaoka A.: *Catalogue of illusions*, 2008 (stránky jsou dostupné na <http://www.psy.ritsumei.ac.jp/~akitaoka/cataloge.html>).
- [43] Kitaoka A.: *Comprehensive Classification of Illusions*, 2005 (stránky jsou dostupné na <http://www.psy.ritsumei.ac.jp/~akitaoka/classificatione.html>).
- [44] Kitaoka A., Pinna B., Brelstaff G.: Contrast polarities determine the direction of Café Wall tilts, *Perception* **33** (2004) 11-20 (článek je dostupný na WWW adrese <http://www.psy.ritsumei.ac.jp/~akitaoka/CafeWall.pdf>).
- [45] Koenderink J. J., van Doorn A. J. & Lappin J. S.: Direct measurement of the curvature of visual space, *Perception* **29** (2000) 69-79.
- [46] Koffka K.: *Principles of Gestalt Psychology*. New York: Harcourt, Brace & World 1935.
- [47] Kuhn T. S.: *Struktura vědeckých revolucí*, Oykomen, Praha 1997.
- [48] Lachman J. L., Butterfield E. C.: *Cognitive Psychology and Information Processing: An Introduction*, Erlbaum 1979.
- [49] Leibowitz H.: Cross-cultural and Personality Factors Influencing The Ponzo Perspective Illusion. Riegel, K., Meacham, J. A. [eds.] *The Developing Individual in a Changing World: Historical and Cultural Issues*, SE, 2009, 255-263.
- [50] Logvinenko A. D.: Lightness induction revisited, *Perception* **28** (1999) 803-816.
- [51] Loomis J. M., Da Silva J. A., Fujita A., Fukusima S. S.: Visual space perception and visually directed action, *Journal of Experimental Psychology: Human Perception and Performance* **18** (1992) 906-921.
- [52] Mannoni L., Nekes W., Warner M.: *Eyes, Lies and Illusions*, Hayward Gallery, London 2004.
- [53] Marotta J. J., DeSouza J. F. X., Haffenden A. M., Goodale M. A.: Does a monocularly presented size-contrast illusion influence grip aperture? *Neuropsychologia* **36** (1998) 491-497.
- [54] Moini A.: *Vision Chips or Seeing Silicon*, 1997 (text je dostupný na http://sl-div-bi-pb.web.cern.ch/sl-div-bi-pb/PrivatePBpage/Electronics/Electronics-Misc/Vision/vis_new.pdf).
- [55] Monaco J.: *How to Read a Film*. New York: Oxford UP 1981.

- [56] Morikawa K.: Last but not least: An application of the Müller-Lyer Illusion, *Perception* **32** (2003) 121-123.
- [57] Monzer L.: Umělé osvětlení v obytných prostorech - Světlo, barva a barevné vidění, *Světlo* **02** (2003) (dostupné na http://www.odbornecasopisy.cz/index.php?id_document=23077).
- [58] Parks T. E.: Demystifying the Poggendorff: A start, *Perception* **38** (2009) 142-144.
- [59] Pizlo Z.: Perception viewed as an inverse problem, *Vision Research* **41** (2001) 3145-3161.
- [60] Prinzmetal W., Shimamura A. P., Mikolinski M.: The Ponzo illusion and the perception of orientation, *Perception & Psychophysics* **63** (2001) 99-114.
- [61] Rybár J.: Pokus o klasifikáciu optických ilúzií. Kvasnička V., Trebatický P., Pospíchal J., Kelemen J. (eds.) *Mysel, inteligencia a život*. STU Bratislava, (2007) 223 - 232.
- [62] Seckel A.: *Nová kniha optických iluzí*, Albatros, Praha 2005.
- [63] Seckel A.: *Masters of Deception*, Sterling 2008.
- [64] Seckel A.: *Poggendorff. Illusion*, Works L. L. C. 1997 (stránka dostupná na <http://www.psychologie.tu-dresden.de/i1/kaw/diverses%20Material/www.illusionworks.com/html/poggendorff.html>).
- [65] Seckel A.: *Twisted Cord*, 1997 (text je dostupný na WWW http://www.psychologie.tu-dresden.de/i1/kaw/diverses%20Material/www.illusionworks.com/html/twisted_cord.html).
- [66] Segall M. H., Campbell D. T., Herskovit M. J.: The Influence of Culture on Visual Perception. In: Toch H., Smith C. (eds.) *Social Perception*, The Bobbs-Merrill Company, Inc. 1966.
- [67] Šimeček M., Šikl R.: Vidíme to, co vidět potřebujeme ... což je někdy pouze to, co vidět chceme, případně co jiní chtějí, abychom viděli, *Sborník konference Kognice a umělý život VII* (2005) 333 – 336.
- [68] Šimeček M., Šikl R.: Klamání oka v architektuře. Kdo koho klame? A klame vůbec? In Miovský M., Čermák I., Chrz V. (Eds.) *Umění ve vědě, věda v umění*. Praha: Grada.
- [69] Šimeček M., Šikl R.: Může být citlivost ke zrakovým klamům ovlivněná dlouhodobým pobytem v uzavřeném prostředí? *Kognice a umělý život IX* (2009) 313 – 318.
- [70] Todorovič D.: Constancies and Illusions in Visual Perception, *Psichologija* **35** (2002) 125-207.
- [71] Westheimer G.: Illusions in the spatial sense of the eye: Geometrical-optical illusions and the neural representation of space, *Vision Research* **48** (2008) 2128–2142.

Použití vrstevnatých neuronových sítí v bioinformatice

Daniel Svozil¹

Abstrakt. Bioinformatika je multidisciplinární vědecký obor stojící na hranici molekulární biologie, statistiky a počítačových věd. Zahrnuje použití výpočetní techniky pro ukládání, vyhledávání, analýzu, interpretaci a distribuci informací týkajících se biologicky relevantních makromolekul. Postupy vytěžování znalostí z dat hrají zásadní roli při analýze a interpretaci biologických dat, jejichž množství v současnosti roste exponenciálním tempem. Algoritmy a techniky používané při vytěžování znalostí z dat pocházejí z oblasti umělé inteligence zvané strojové učení. Jedním z nejúspěšnějších přístupů jsou vrstevnaté neuronové sítě. Cílem tohoto textu je obeznámit čtenáře nejen se základy a názvoslovím molekulární biologie, strojového učení a neuronových sítí, ale také popsat některé specifické problémy aplikace neuronových sítí na biologická sekvenční data.

1 Úvod do bioinformatické problematiky

Je nezpochybnitelnou skutečností, že biologie hrála a nadále hraje klíčovou roli při odhalování tajemství fungování přírodních zákonů. Moderní biologie má blízko k mnoha dalším vědeckým odvětvím, jako je především medicína, chemie (studium chemických procesů v živých organismech se zabývá biochemie) nebo fyzika (potřeba vysvětlit biologické jevy na atomární úrovni dala vzniknout biofyzice).

Kvantitativní přístup je v moderní biologii nepostradatelný, což je jednoznačně demonstrováno faktem, že většina biologického výzkumu zahrnuje aplikování některého z přehršle dostupných matematických, statistických či výpočetních nástrojů. Klasickým příkladem použití kvantitativních nástrojů z historie genetiky jsou objevy Gregora Mendela a Thomase Morgana, kteří odhalili základní principy dědičnosti jednoduchým počítáním genetických variací v rostlinách (G. Mendel) a v muškách octomilkách (T. Morgan). Nicméně žádný z takovýchto postupů kvantitativní biologie se nedá považovat za část vědní disciplíny zvané *bioinformatika*, ačkoliv tato je ze své podstaty též kvantitativní.

¹ Laboratoř informatiky a chemie, Fakulta chemické technologie, Vysoká škola chemicko-technologická, Technická 3, 166 28 Praha, E-mail: daniel.svozil@gmail.com

Co to tedy vůbec je ta tajemná bioinformatika? Je to mezioborová vědecká oblast stojící na rozhraní mezi počítačovou vědou (angl. computer science) a biologií a zahrnuje použití výpočetní techniky pro ukládání, vyhledávání, analýzu, interpretaci a distribuci informací týkajících se biologicky relevantních makromolekul, jako jsou např. proteiny či nukleové kyseliny. Bioinformatika, která je limitována na sekvenční, strukturní a funkční analýzu genů, genomů a proteomů a je tudíž možno ji chápat jako počítačovou molekulární biologii, se liší od příbuzného oboru *počítačové biologie* (angl. computational biology), jež zahrnuje všechny oblasti biologie vyžadující použití výpočetních prostředků. Do počítačové biologie spadají např. problémy jako matematické modelování ekosystémů, modelování dynamiky populace či aplikace teorie her v behaviorálních studiích apod.

Jednou z největších výzev pro počítačového vědce, který by rád pracoval v oblasti molekulární biologie, je osvojit si, alespoň na základní úrovni, mnohdy téměř odrazující spletitosti biologických znalostí a taktéž rozsáhlou biologickou terminologii. Na následujících stránkách se pokusím prezentovat minimální penzum znalostí z biologických věd, které jsou důležité pro pochopení dalšího výkladu. Většina z toho, co bude řečeno v následující kapitole, čerpá z oborů tak diametrálně odlišných, jako je molekulární biologie (zabývá se zkoumáním biologických procesů na molekulární úrovni), genetika (zabývá se dědičností a odlišnostmi mezi živými organismy vyvolanými při přenosu dědičné informace), biochemie, buněčná biologie (zabývá se studiem buněk, jejich vlastností, strukturou a interakcemi s okolním prostředím), vývojová biologie (studuje procesy růstu a vývoje jednotlivých organismů) či evoluční biologie (zabývá se výzkumem vzniku a evoluce organismů či druhů). Z toho vyplývá, že mnohé principy budou při výkladu zjednodušovány a zevšeobecňovány. Nicméně je třeba stále mít na paměti, že pro každé zobecnění, které bude učiněno, existuje ve skutečnosti mnoho výjimek. Na světě žije mnoho organismů a velmi, opravdu velmi málo znalostí lze zobecnit na všechny.

1.1 Co je život?

Ačkoliv naše intuice nám říká poměrně jasně a správně, co je to živý organismus, neexistuje žádná vědecky přesná, všeobjímající a pochopitelná definice života. Ústředním rysem života je jeho schopnost reprodukce. Leč reprodukce sama o sobě není dostatečným důvodem definovat rozmnožující se jednotku jako živou – např. počítačové programy jsou schopny vytvářet bez problémů kopie sama sebe, to z nich však nedělá živé organismy. Většina živočichů přijímá hmotné suroviny z okolí a přeměňuje je na energii, která je hlavním hnacím motorem biologických dějů udržujících organismus při životě. Nicméně např. viry (viz dále) tak nečiní, jsou totiž tvořeny téměř čistým genetickým materiálem pouze obaleným ochrannou vrstvou. Veškerá syntetická práce spojená s vytvářením (tedy reprodukcí) nových virů je pak vykonávána buňkou virem napadenou. Přesto by většina lidí o virech prohlásila, že jsou živými organismy.

Jiným přístupem k definování života je pochopit jeho elementární spojitosti a vztahy. Všechny živé organismy jsou v jistém slova smyslu příbuzné jeden druhému. Každá dvojice organismů, bez ohledu na to, jak moc jsou si podobné, vzešla kdysi

v hluboké minulosti ze společného předka. Organismy se odlišují jeden od druhého a dosahují různé úrovně komplexity díky jevu známému jako *evoluce*. Evoluce má tři hlavní složky: dědičnost (předávání charakteristických znaků z rodiče na potomka), variace (proces, díky němuž jsou potomci odlišní od svých rodičů a ne jejich pouhou kopií) a výběr (selekce, proces, který upřednostňuje reprodukci určitých jedinců na úkor jedinců jiných a tudíž posiluje v populaci určité znaky). Zřejmě nejlepší definice života říká, že život je výsledkem evolučních procesů probíhajících na Zemi. Klíčovým faktorem pro pochopení téměř všech struktur a funkcí živého organismu je *dědičnost*. Množství odlišností (variací) mezi dvěma generacemi je obecně velmi malé a evoluce probíhá na mnohem delší časové škále (život na zemi vznikl před zhruba 3,7 miliardami let). Nicméně i pro některé aspekty živých organismů platí, že se tyto změnilly od dob společného předka velmi nepatrně – říkáme, že takové faktory jsou *zakonzervované*.

Leč dědičnost sama o sobě není dostatečným mechanismem pro vyvolání evoluce, neboť perfektní dědičnost by vedla k populaci kompletně identických organismů. Aby byly organismy schopny se vyvíjet, musí docházet ke vzniku odchylek (*variací*). Mezi zdroje variací patří např. mutace (náhodné změny v dědičném materiálu), přeskupování genů nebo křížení (rekombinace). Většina variací jsou buď neutrální (tj. bez dopadu na organismus), nebo zhoubné. *Výběr* (selekce) je pak procesem, během něhož se rozhoduje, které z variací se udrží (tzv. *zafixují*) a které nikoliv. Přírozený výběr je založen na *reprodukční způsobilosti* (angl. *fitness*) každého jedince, kdy reprodukční způsobilosti je mírou toho, kolik přeživších potomků je organismus schopen vyprodukovat. Čím je organismus lépe přizpůsoben (adaptován) na své prostředí, tím větší šanci mají jeho potomci na přežití.

Evoluci je možno přirovnat prohledávání obrovského prostoru všech možných charakteristických rysů organismu. Tento prostor lze definovat dosti přesně – všechny organismem děděné vlastnosti jsou kódovány v nositelské molekule, kterou je kyselina deoxyribonukleová (viz dále). Vlastnosti jsou reprezentovány jednoduchým lineárním čtyřpísmenným kódem. Nicméně převod tohoto kódu do děděných rysů organismů (např. jako je uspořádání a propojení orgánů v těle či provázanost nervového systému) je velmi komplexním a složitým procesem. Jedno konkrétní zakódování pro jednoho konkrétního jednotlivce se nazývá *genotypem* a výsledné fyzické charakteristiky tohoto jednotlivce jsou označovány jako jeho *fenotyp* (např. genotypem je přítomnost genu pro modré oči a fenotypem je pak skutečný výskyt modrých očí). Rozdíl mezi genotypem a fenotypem je důležitý, neboť malé změny v genotypu mohou vést k obrovským změnám ve fenotypu.

Souhrn všeho genetického materiálu v organismu se nazývá *genomem*. Velikost genomu, která ovšem nemá přímou souvislost s komplexitou organismu, se liší v řádu od cca 5 000 nukleotidů² v jednoduchých virech po více než 10¹¹ nukleotidů v některých vyšších rostlinách (lidé mají genom o velikosti cca 3x10⁹ nukleotidů). Navzdory této neuvěřitelné diversitě jsou hnací silou prakticky ve všech organismech stejné základní děje. Všechny živé organismy (s výjimkou virů) jsou vystavěny z membránou obklopených buněk, v nichž probíhají do nejmenších detailů vyladěné

² Přesněji řečeno, nejde o nukleotidy, ale o jejich páry.

kaskády chemických reakcí. Zhruba tisícovka chemických sloučenin, které se účastní základních reakcí v živých buňkách, je si v rámci většiny živých organismů pozoruhodně podobná. Genetický materiál odpovědný za kódování všech těchto sloučenin je zapsán ve víceméně stejném molekulárním jazyce bez ohledu na druh organismu.

Nebiolog si často ani neuvědomuje obrovské množství různých organismů žijících na planetě Zemi. Odhadem se na Zemi vyskytuje 5 – 50 miliónů různých biologických druhů. Organismy na Zemi se dělí do tří základních skupin: Archaea, Bacteria a Eucarya (za čtvrtou skupinu je pak možno pokládat viry). Eucarya (též označována jako *eukaryota*) jsou organismy, se kterými se my, lidé, kamarádíme nejvíce. Sestávají z buněk obsahujících *jádro* (v něm je obsažen kompletní genetický materiál) a další specializované *organely* (organelou jsou např. z rostlin dobře známé chloroplasty). Doména eukaryot se pak dělí zhruba na čtyři rozdílné říše: živočichové, zelené rostliny, houby a jednobuněčné mikroorganismy zvané protisti. Baktérie (označované též jako *prokaryota*) jsou všudypřítomné jednobuněčné organismy neobsahující jádro, které jsou obklopeny membránou složením odlišnou od membrány eukaryot. Pro experimentálního biologa je hrdinkou prokaryotického světa bakterie *Escherichia coli* (v běžné mluvě mazlivě označována jako e-koli), která žije ve střevech a je potřebná pro správnou funkci zažívacího traktu. Ta se v zajetí snadno množí a je tudíž velmi dobře prostudována. V roce 1977 byla pány Carlem Woese a George E. Foxem [1] definována nová třída organismů tak velmi nepodobná eukaryotům či bakteriím jak po stránce genetické, tak po stránce morfologické, až se mnohým biologům zamžil na několik let zcela zrak. Archaea žijí v extrémně horkých sírových vývěrech na dně oceánů, v horkých kyselých pramenech či na jiných obdobně přivětivých místech. Mezi živé organismy se počítají také viry, což jsou paraziti, kteří napadají buňku a spoléhají se na její biomolekulární aparát, jenž využívají jako zdroj své vlastní obživy a reprodukce. Viry obsahují pouze velmi malé množství vlastního genetického materiálu a jejich jednoduchost a významná role v mnoha lidských nemocech z nich dělají vděčnou a aktivní výzkumnou oblast.

Na tomto místě je vhodné opravit jedno obvyklé nesprávné tvrzení, konkrétně že některé organismy jsou „primitivnější“ než jiné. Každý existující organismus byl totiž schopen přežít až do současné doby, tudíž, řečeno jinými slovy, ani jednoduché existující organismy není možno považovat za primitivní! Životní prostředí současnosti je zcela nepodobné podmínkám na Zemi v dobách, kdy se život začal vyvíjet a dokonce i ty nejjednodušší žijící organismy se vyvinuly tak, že byly schopny přežít až do dnešních časů. Je možné studovat skupiny dvou velmi vzdáleně příbuzných tvorů (např. bakterie a člověka) za účelem usuzování o vlastnostech dříve žijících organismů. Cokoliv, co mají lidé a bakterie společné, jsou rysy, které jsou s největší pravděpodobností sdíleny s jejich společným předkem žijícím před mnoha a mnoha věky. Největší vypovídající hodnotu o původu organismů na Zemi mají právě ty vlastnosti, které nám říkají, co mají tvorové mezi sebou společného; rozdíly mezi organismy jsou již daleko méně informativní.

Život na Zemi se vyvíjel postupně, od jednodušších jednobuněčných organismů až po organismy složitější, mnohobuněčné. A právě vznik mnohobuněčných organismů vedl k obrovskému nárůstu komplexity všech možných forem života na Zemi. K této

události došlo před necelou miliardou let, tedy až v poslední čtvrtině historie života. Důležitým rozdílem mezi vícebuněčnými organismy a koloniemi jednobuněčných organismů (např. korály) je skutečnost, že vícebuněčné organismy mají odlišené *pohlavní* buňky (spermie a vajíčka) od buněk *somatických* (všechny ostatní buňky kromě pohlavních). Oba dva druhy buněk jsou schopny se dělit, ale pouze pohlavní buňky mohou dát vzniknout novému jedinci. Somatické buňky bývají obvykle specializovány na jistou konkrétní činnost, patří mezi ně např. svalové, krevní či nervové buňky. Dělení somatických buněk a jednobuněčných organismů, tzv. *buněčný cyklus*, je čtyřfázový proces, v jehož závěrečné fázi, *mitóze*, dochází ke vzniku dvou dceřiných buněk. Pohlavní buňky se rozmnožují procesem známým jako *meióza*. Součástí tohoto procesu je tvorba spermií a vajíček, které každé obsahují pouze polovinu genetického materiálu.

Všechny vícebuněčné organismy začínají svůj život ve formě jediné buňky – oplodněného vajíčka. Specializované buňky vznikají z této jedné buňky procesem zvaným *buněčná diferenciací*. Vývoj dospělého jedince je pak extrémně komplexní záležitostí zahrnující nejen buněčnou diferenciaci, ale též migraci a přeskupování buněk. Zkoumáním všech těchto procesů se zabývají biologické oblasti embryologie a vývojová biologie (angl. developmental biology, je důležité si tuto disciplínu neplést s evoluční biologii, neboť evoluce se v češtině často také překládá jako vývoj).

Poté, co byly velmi stručně nastíněny rozmanitosti života a některé z klíčových prvků evoluce, se v další kapitole podíváme blížeji na to, jakým způsobem vícebuněčné organismy vůbec fungují.

1.2 Biomolekuly – stavební kameny živé hmoty

Všechny buňky, ať již Archaea, hovicí si na dně oceánu v nějakém horkém sirném prameni, nebo buňky svalové, mají stejné základní vlastnosti: obsahují *cytoplasmu* (vnitřní prostředí buňky obklopené buněčnou membránou) a genetický materiál a používají shodný mechanismus přenosu genetické informace. Studium buněk a tkání vede k pochopení jejich fyziologické funkce (např. znalost faktu, že biceps i srdce sestávají ze svalových buněk, pomáhá vysvětlit způsob, jakým cirkuluje krev v těle), nicméně způsob, jakým fungují jednotlivé buňky, zůstává na této úrovni popisu stále zahalen tajemstvím. Právě pochopení fungování buněčného aparátu v pojmech jeho chemických vlastností vedlo k menší revoluci v biologii v poslední 40-50 letech. Stále více a více životních funkcí, jako je např. dělení buňky, imunitní odezva či přenos nervového vzruchu, je detailně studováno a pochopeno jako interakce komplikovaných sítí chemických reakcí. Sloučeniny, které provádějí a regulují tyto aktivity, jsou souhrnně označovány jako *biomolekuly*. Velké biomolekuly zahrnují např. proteiny (bílkoviny), nukleové kyseliny, lipidy či cukry. Pro ně je typické, že se jedná o tzv. *makromolekuly* – velké, složité látky, které jsou tvořeny zřetězením několika málo jednoduchých stavebních jednotek za sebou. Genetický materiál buňky, tvořený deoxyribonukleovou kyselinou (DNA), udává, jakým způsobem, v jakém množství a kdy jsou tvořeny proteiny, které posléze řídí všechny procesy v buňce probíhající.

Živé systémy zpracovávají hmotu, energii a informaci. Jak již bylo řečeno dříve, mnohé aspekty živých organismů přetrvaly v průběhu téměř 4 miliardy let trvajících

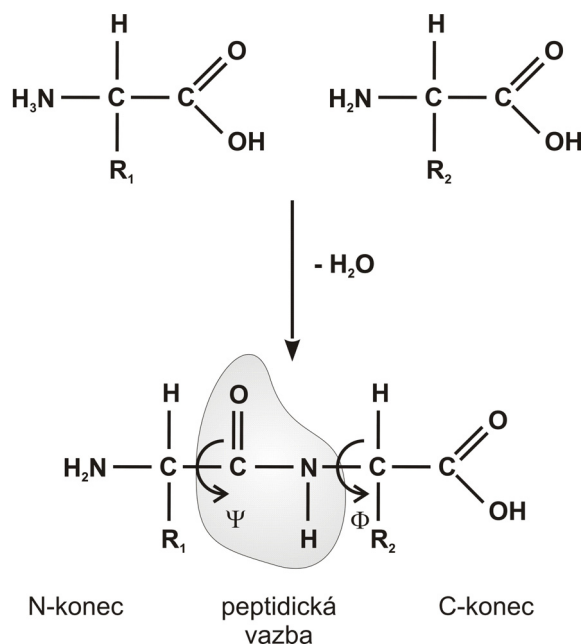
evolučních procesů. Základní funkční a strukturální chemickou jednotkou v organismu jsou proteiny, základní chemickou jednotkou uchovávání energie je fosfátová vazba v molekule *adenosin trifosfátu* (ATP) a základní jednotkou přenášející informaci jsou čtyři *nukleotidy*, z nichž sestávají kyseliny deoxyribonukleová (DNA) a ribonukleová (RNA). Chemické složení živých organismů je značně stálé i u jinak dosti odlišných životních forem: zhruba 70% buňky tvoří voda, 4% jsou malé molekuly jako cukry, ATP či ionty, proteiny tvoří cca 15%-20% obsahu buňky a DNA/RNA přispívají 2%-7% k celkové hmotnosti buňky.

Živé organismy se řídí chemickými a fyzikálními zákony včetně druhého termodynamického zákona, podle něž množství *entropie* (neuspořádanosti) ve vesmíru stále narůstá. Jediný možný způsob, jak bojovat tváří v tvář entropii je spotřebovávat energii, tím si vytvářet vnitřní pořádek (přesně to činí živé organismy) a poté přenášet energii zpět do okolí ve formě tepla. Nárůst vnitřní organizace je doprovázen nárůstem neuspořádanosti v okolním prostředí, celková entropie roste a druhý zákon tudíž zůstává zachován. Živé organismy získávají energii buď fotosyntetickou cestou (rostliny), nebo při procesu dýchání. Energie dostupná pro biochemické reakce musí být velmi obecně využitelná. Je nutné, aby byla k dispozici, pokud jí je potřeba, a pokud jí potřeba není, tak aby byla uchována pro pozdější potřeby. Uchovávání a transport energie v živém organismu je velmi delikátní záležitost, neboť při uvolňování energie nesmí docházet k porušování jiných než cílových vazeb, takže jednotkové množství energie musí být dostatečně malé, aby k něčemu takovému nedošlo, ale na druhou stranu dostatečně velké, aby bylo ekonomické energii ukládat. Nejběžnější zásobárnou a přenašečem energie je molekula adenosin trifosfátu ATP.

Hlavní komponentou živých organismů jsou bílkoviny (proteiny), které v organismu vystupují v mnoha různých funkcích. Soubor všech proteinů v organismu se nazývá *proteom*. Proteiny tvoří infrastrukturu, jež drží organismy dohromady (tzv. strukturální proteiny), katalyzují (urychlují) chemické reakce v organismu (tzv. enzymy), fungují jako přepínače zajišťující vypínání a zapínání genů nebo nám pomáhají v roli detektorů rozpoznávajících “naše” od “cizího” a zajišťujících tak patřičnou imunitní reakci. Navzdory všem těmto tak odlišným funkcím jsou proteiny tvořeny stejnými základními stavebními prvky – *aminokyselinami*. Každá aminokyselina má podobnou základní strukturu, sestává z centrálního uhlíkového atomu, na něž je na jednom konci navázána amino skupina $-NH_2$, na druhém konci karboxylová skupina $-COOH$ a na třetím konci, lze-li se takto vyjádřit, proměnný *postranní řetězec* označovaný jako $-R$ (viz Obr. 1). Naprostá většina proteinů je v organismu tvořena pouhými dvaceti tzv. esenciálními aminokyselinami, které se od sebe liší v chemickém charakteru postranního řetězce $-R$. Proteiny jsou polymery vznikající řetězením základních jednotek (tzv. monomerů) – aminokyselin – za sebou³. Toho je dosaženo vytvořením tzv. *peptidické vazby*, která vznikne spojením amino skupiny jedné aminokyseliny s karboxylovou skupinou další aminokyseliny za uvolnění jedné molekuly vody. Aminokyselina takto včleněná do řetězce se pak označuje pojmem *reziduum*. Sekvence (tj. sled) aminokyselinových reziduí je označována jako *primární struktura*.

³ Kratší úseky do cca 50 aminokyselin se označují jako *peptidy*.

Aminokyselinový řetězec má své dva konce chemicky nestejně, je třeba je odlišovat a označují se jako N-konec a C-konec.



Obr. 1. Reakcí dvou aminokyselin vzniká za odštěpení molekuly vody dipeptid. Aminokyseliny se liší postranními řetězci R₁ a R₂.

Peptidická vazba spojuje jednotlivá aminokyselinová rezidua (označovaná 20 různými jednopísmennými zkratkami) a toto propojení je možno charakterizovat dvěma stupni rotační volnosti (tj. v rámci jednoho rezidua je možno otáčet nezávisle dvěma vazbami) označovanými jako úhly Φ a Ψ (viz Obr. 1). Aminokyselinový řetězec není v buněčném prostředí v natažené formě, ale poskládá se (tzv. *foldingu*) do třírozměrné struktury, jejíž 3D uspořádání (*konformaci*) lze, mimo jiné, charakterizovat právě pomocí jednotlivých Φ a Ψ úhlů. Před více než 30 lety bylo ukázáno, že sekvence aminokyselin v proteinu jednoznačně určuje výslednou 3D strukturu proteinu, která potom udává biologickou aktivitu a funkci proteinu. A právě pochopení mechanismu vzniku třírozměrné struktury proteinu ze znalosti pouze sekvence jeho aminokyselin, tzv. *protein foldingu*, patří mezi důležité nevyřešené problémy molekulární biologie. 3D struktura proteinů je tvořena hierarchií tří hlavních uspořádání: primární struktura je pouhá sekvence aminokyselin, *sekundární* struktura je lokální uspořádání několika aminokyselinových reziduí, které zaujímají konkrétní typy konformací a *terciární* struktura je dána pozicí všech atomů v třírozměrném prostoru poté, co je protein poskládán (a je tedy představována vzájemným uspořádáním prvků sekundární struktury).

3D struktura proteinu je stabilizována především tzv. *vodíkovými vazbami*. Vodíková vazba je přitažlivá síla mezi dvěma atomy, které nesou částečné elektrostatické náboje opačného znaménka. A právě díky vodíkovým vazbám vytváří

lineární aminokyselinový řetězec dva základní typy sekundární struktury: šroubovicové struktury, tzv. *α-šroubovice* (*α-helixy*) nebo ploché struktury připomínající listy (tzv. *β-listy*). Tyto dva základní typy sekundární struktury jsou pospojovány tzv. *β-otáčkami*, které se, obdobně jako úseky nezaujímající žádnou specifickou konformaci (tzv. *náhodné klubko*), také počítají mezi sekundární strukturu.

Představují-li proteiny tažné koně biochemického světa, pak nukleové kyseliny jsou jejich vozatajové. Veškerá genetická informace v žijících organismech je uložena v deoxyribonukleové kyselině DNA a v ribonukleové kyselině RNA. Tyto biopolymery vznikají zřetěžením pouhých čtyř základních jednotek, tzv. *nukleotidů*. Každý nukleotid se sestává ze tří částí: *báze*, *cukru* (ribózy v RNA a deoxyribózy v DNA) a z fosfátové skupiny. Báze jsou čtyři a vždy dvě a dvě mají stejný chemický základ, jedna skupina se nazývá *puriny* (označované jako R), druhá *pyrimidiny* (označované jako Y). Purinové nukleotidy v DNA jsou *adenin* A a *guanin* G a pyrimidinové jsou *cytosin* C a *tymin* T, který je v RNA nahrazen *uracilem* U. Nukleotidy jsou pospojovány tzv. fosfodiesterovou vazbou (funkční obdoba peptidické vazby v proteinech), která, spolu s cukry, vytváří tzv. páteř. Také řetězce nukleotidových reziduí mají dva chemicky rozdílné konce označované jako 3' a 5' konec. Sekvence nukleových kyselin se zapisují ve směru od 5' konce ke 3' konci. Dva řetězce např. v DNA se k sobě nalepí a zůstanou u sebe díky vzniku vodíkových vazeb mezi dvěma konkrétními páry bází (tzv. *komplementarita*) – adenin se páruje s tyminem a cytosin s guaninem. Výsledné třírozměrné uspořádání dvou komplementárních řetězců DNA v prostoru je pak ona známá dvoušroubovice. Struktura RNA je složitější, neboť ribóza má oproti deoxyribóze navíc jedno místo, které je schopno tvorby vodíkové vazby.

Nyní, když jsme pronikli, tedy alespoň v mezích možností, do tajů struktury a funkce proteinů a nukleových kyselin, je možno začít skládat dohromady mozaiku molekulárních procesů probíhajících v buňce.

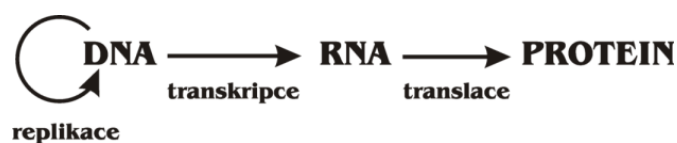
1.3 Exprese genu

Genetická informace v organismu bývá uchována v molekulách DNA označovaných jako *chromosomy*. U některých pohlavně se rozmnožujících organismů (např. u člověka) sestává každý chromosom ze dvou podobných molekul DNA, kdy každá pochází od jednoho z rodičů. Takové organismy se nazývají *diploidní* organismy, pohlavně se rozmnožující organismy s jednou DNA molekulou v chromosomech se pak nazývají *haploidní* organismy. Veškerá genetická výbava organismu vzata jako celek se nazývá *genom*. Hlavní rolí nukleových kyselin je určovat (kódovat) primární strukturu bílkovin, tzn., že konkrétní sekvence nukleotidů jednoznačně určuje, které konkrétní aminokyseliny budou v organismu syntetizovány. Na zakódování 20 aminokyselin je potřeba sekvence tří nukleotidů⁴, tzv. *kodon*. Čtyři nukleotidy mohou vytvořit celkem $4^3 = 64$ tripletů, což je více než je potřeba na kódování 20 aminokyselin (pouhá dvojice nukleotidů však kóduje jenom 16 aminokyselin). Tři

⁴ Sekvence dvou nukleotidů se označuje jako dublet, tří nukleotidů jako triplet, čtyř nukleotidů jako tetraplet atd.

z těchto kodonů použité jako signály pro ukončení aminokyselinového řetězce se nazývají *stop kodony*. Všechny ostatní kódují nějakou konkrétní aminokyselinu, což znamená, že většina aminokyselin je kódována více než jedním kodonem. Tomuto jevu se říká *degenerace*. Seznamu kodonů a jimi kódovaných aminokyselin se říká *genetický kód*, kterýžto pojem by si měli zapamatovat především novináři, již často učeně popisují, že se vědcům podařilo přečíst genetický kód. To je samozřejmě nesmysl, genetický kód je znám již po dlouhá desetiletí a je konzervován mezi různými třídami organismů, což znamená, že je, až na drobné výjimky, pro všechny žijící organismy stejný. Pokud se vědcům podařilo něco přečíst, tak to zajisté nebyl genetický kód, ale genom (tj. sekvence všech nukleotidů v kompletní genetické výbavě) nějakého organismu.

Při procesu syntézy proteinů, tzv. *proteosyntéze*, je mapována sekvence kodonů na sekvenci aminokyselin. Nicméně při tomto procesu mohou nastat rozličné komplikace. Protože kodony jsou tripletety, existují tři možné způsoby kde začít číst vlákno DNA. Například řetězec ...AATGCGATAAG... je možno přečíst jako ...AAT-GCG-ATA..., jako ...ATG-CGA-TAA... či jako ...TGC-GAT-AAG.... Každé z těchto možností se říká *čtecí rámec*. Dostatečně dlouhá sekvence kodonů bez „stop kodonů” se nazývá *otevřený čtecí rámec* (angl. open reading frame, ORF) a všechna genetická informace potřebná ke vzniku jednoho proteinu se označuje jako *gen*. Nejenom, že v DNA existují tři různé čtecí rámce, navíc je možno číst každý ze dvou řetězců ve dvoušroubovici a možný počet čtecích rámců je tedy šest. Kódující sekvence DNA (*exony*) ve většině eukaryotických organismů mají v sobě vloženy sekvence nekódující - *introny*. Kromě kódujících sekvencí obsahuje DNA velké množství dalších informací. Každá buňka v těle obsahuje stejnou DNA, nicméně každý typ buňky generuje zcela jinou sadu proteinů a dokonce i v rámci stejného typu buněk se vznikající proteiny liší v závislosti na stáří buňky. Proces vzniku proteinů z genů (tzv. *exprese genů*) musí tedy být nějakým způsobem regulován, což se děje pomocí speciálních DNA sekvencí, tzv. *signálů*. Existuje mnoho různých signálů udávajících, kde začíná a končí oblast kódující jeden protein (ta se někdy označuje jako *strukturní gen*), kde má dojít k vystřížení intronů, a především kdy a v jakých množstvích mají být proteiny syntetizovány.



Obr. 2. Schéma centrálního dogmatu molekulární biologie.

Proces mapování DNA sekvence na sbalený protein (tzv. centrální dogma molekulární biologie) v eukaryotech zahrnuje několik kroků (viz Obr. 2).

Prvním krokem je přepis (*transkripce*) části DNA do molekuly RNA označované jako messenger RNA (*mRNA*). Tento proces začíná navázáním molekuly enzymu RNA polymerázy na specifické místo v DNA, které je rozpoznáno na základě signálu označovaného jako *promotor*. RNA polymeráza katalyzuje reakci, při níž vzniká jednovláknová šroubovice mRNA (tzv. *primární transkript, pre-mRNA*), jejíž sekvence

je komplementární sekvenci přepisovaného vlákna DNA. Primární transkript není však okamžitě přeložen do aminokyselinového řetězce, ale nejprve musí dojít k tzv. sestřihu (angl. splicing). Při něm jsou z primárního transkriptu odstraněny introny a exony jsou poté spojeny za vzniku tzv. vyzrálé (angl. mature) RNA. Proces sestřihu probíhá na velkém komplexním biomolekulárním stroji zvaném *spliceosom*. mRNA po sestřihu přechází z jádra do cytoplasmy, kde se naváže na tzv. *ribozom*. Ribozom sestává z kombinace jiného typu RNA, tzv. ribozomální RNA (*rRNA*), a proteinových jednotek. Na ribozomu dochází k překlada (*translaci*), při níž na základě sekvence mRNA, která je ribozomem čtena po kodonech, vzniká proteinový řetězec o konkrétní aminokyselinové sekvenci. Proces translace závisí na další RNA molekule, zvané přenosová (angl. transfer) RNA, *tRNA*, která má na jednom konci trojici nukleotidů označovanou jako *antikodon* a na druhém konci navázanou příslušnou aminokyselinu, jež je kódována sekvencí komplementární sekvenci antikodonu. Antikodon tRNA se na ribozomu váže na mRNA na základě komplementarity bází kodonu (na mRNA) a antikodonu (na tRNA), čímž jsou aminokyseliny přinášeny do vzájemného fyzického kontaktu a vzniká z nich, za tvorby peptidických vazeb, proteinový řetězec. Proteinový řetězec pak opouští ribozom a okamžitě se skládá do třírozměrné, tzv. *nativní* (tedy funkční) konformace.

Každá buňka obsahuje v jádře uchovanou stejnou DNA, ale přesto DNA v jedné buňkách kóduje jiné proteiny než DNA v buňkách jiných. Např. DNA ve svalových buňkách kóduje proteiny nutné pro správnou funkci svalů, zatímco DNA v buňkách v oku kóduje proteiny potřebné pro tvorbu oční sítnice. Rozdíl je způsoben *regulací genové exprese*. V reakci na změny v okolním prostředí a na jiné podněty musí být v buňce v každém okamžiku exprimovány⁵ správné proteiny ve správném množství. Vysoce vyladěný a přesný mechanismus, kterým probíhá regulace genové exprese, není zatím zcela do detailů pochopen.

Celý regulační aparát sestává ze složité soustavy proteinů nalézající se v jádře. Pravděpodobně nejdůležitější z těchto proteinů jsou velmi konzervované bílkoviny zvané *histony*, jež jsou velmi pevně navázané na DNA v chromosomech v jádře. Mnohé regulační proteiny rozpoznávají specifické DNA sekvence (*kontrolní oblasti*), k nimž se váží a svojí přítomností ovlivňují expresi blízkého genu. Sekvence, které jsou zodpovědné za spuštění exprese daného genu, se nazývají *promotory*, sekvence, které naopak potlačují produkci daného proteinu, se nazývají *represory*. Buňky musí být schopny zapínat a vypínat celé kaskády genů v odezvě na různé události. Kontrolní mechanismy jednak pracují v závislosti na množství produktu v buňce již přítomného (tzv. *homeostatická kontrola*), ale musí být samozřejmě také v souladu s nepřeborným množstvím mimobuněčných (extracelulárních) signálů.

Na začátku této kapitoly byla diskutována ústřední role, kterou hraje evoluce v procesu odhalování tajemství života. Jak vyplývá z předchozího, při snaze pochopit mnoho biologických jevů je užitečné zaměřit se až na úroveň jednotlivých molekul a biochemických dějů. A právě mechanismy evoluce na molekulární úrovni jsou s postupující úrovní lidského poznání čím dál tím podrobněji prostudovány.

⁵ Že byl protein exprimován je synonymem pro fakt, že proběhla exprese tohoto proteinu, tj. že na základě informace obsažené v daném genu vznikl konkrétní protein.

Podobnosti či rozdíly mezi molekulami poskytují důležité informace o jejich struktuře a funkci. O biomolekulách majících podobnou sekvenci se předpokládá, že vznikly ze stejného společného předka. Takové molekuly se označují jako *homologní*.

Zdroje variací na molekulární úrovni jsou velmi důležité k pochopení toho, jak se jednotlivé molekuly od sebe odlišují (*divergují*). Jedním z důležitých mechanismů molekulární evoluce je *bodová mutace*, při níž dochází ke změně jednoho nukleotidu v genetické sekvenci. Mezi tyto změny patří *inserce* (vlození) nového nukleotidu, *delece* (odstranění) nukleotidu nebo *záměna* jednoho nukleotidu za jiný. Další mechanismy vnesení variability zahrnují přeuspořádání genetického materiálu na úrovni chromosomů, jako je např. *duplikace genů*, při níž jsou do genomu vloženy dodatečné kopie genu. Pohlavní rozmnožování pak přidává ke změnám v genetickém materiálu další dimenzi. DNA od dvou rodičů prodělává proces zvaný křížení (angl. *crossover*), díky němuž se geny od rodičů namíchají a jsou dále předány na potomka.

Genomové sekvenační projekty přispěly k exponenciálnímu nárůstu množství známých sekvencí uložených v biologických databázích. Projekt strukturní genomiky používá vysoce propustné (angl. *high-throughput*) experimentální techniky ve snaze zkatalogizovat strukturně-funkční informaci o všech proteinech v lidském organismu. Pokroky v technologii, jako jsou např. DNA čipy (angl. *DNA microarray*), které opět produkují enormní množství dat, odstartovaly výzkum v oblastech studia genů, proteinů a genové exprese. Co tedy charakterizuje současnou molekulární biologii, je záplava existujících dat, která je nutno analyzovat a interpretovat s cílem lepšího pochopení fungování živého organismu. Postupy vytěžování znalostí z dat (angl. *data mining*) hrají zásadní roli v porozumění rapidně expandujících zdrojů biologických dat. Aplikace těchto technik je zapotřebí pro analýzu a odkrývání zákonitostí v sekvencích či biomolekulárních strukturách, stejně tak jako pro predikci struktury biomakromolekul, vyhledávání genů, analýzu genové exprese, analýzu biochemických cest, vytěžování znalostí z biomedicínských textů, návrh nových léčiv a pro řešení mnoha dalších problémů objevujících se v genomice a proteomice.

Pod pojmem vytěžování znalostí z dat se skrývá proces analýzy velkých souborů dat za účelem nalezení vztahů v těchto datech skrytých. Algoritmy a techniky používané při vytěžování znalostí z dat pocházejí z oblasti *umělé inteligence* (angl. *artificial intelligence*), konkrétně jejího podoboru zvaného *strojové učení* (angl. *machine learning*), který se zabývá návrhem a vývojem algoritmů pro automatické rozpoznávání komplexních vzorů v datech (tato oblast bývá někdy vyčleňována zvlášť pod názvem *rozpoznávání vzorů*, angl. *pattern recognition*). Všechny výše uvedené vědní oblasti (vytěžování znalostí z dat, umělá inteligence, strojové učení, rozpoznávání vzorů) se ovšem do značné míry překrývají, jejich hranice nejsou jasně definované a v odborné literatuře se často používají jako synonyma.

V následující části budou vysvětleny základní pojmy z oblasti strojového učení, umělé inteligence a rozpoznávání vzorů.

2 Rozpoznávání vzorů

Problém rozpoznávání vzorů (angl. *pattern recognition*) v datech má za sebou již dlouhou a úspěšnou historii. Např. objevení pravidelností v atomových spektrech hrálo

klíčovou roli ve vývoji kvantové mechaniky ve 20. letech 20. století. Rozpoznávání vzorů se zabývá automatickým odhalováním pravidelností v datech za použití počítačových algoritmů a následném využití těchto pravidelností např. pro klasifikaci dat do různých kategorií.

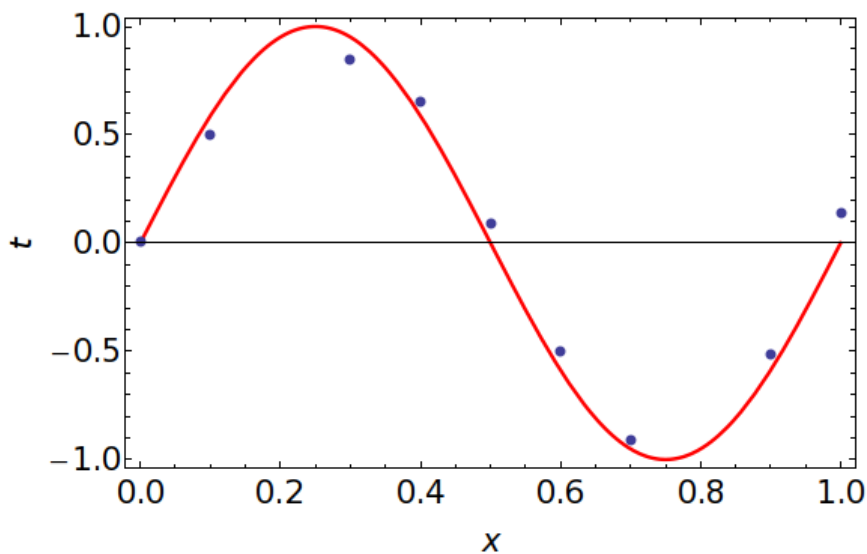
Představme si např. problém rozpoznávání ručně psaných čísel. Každé číslo sestává z 16 x 16 obrazových bodů a je možno ho tedy reprezentovat vektorem \mathbf{x} sestávajícím z 256 reálných čísel. Cílem je vybudovat takový automatický systém, který použije vektor \mathbf{x} jako vstup a na výstupu nám sdělí, o kterou z číslic 0...9 se jedná. Jde o netriviální problém, protože variabilita v zápise číslic je velká, různí lidé napíší různá čísla různě. K tomuto problému je možno se postavit čistě na heuristické bázi a rozlišovat číslice mezi sebou na základě tvarů tahů, délek a směrů rovných úseků apod., leč v praxi tento přístup vede ke značnému nárůstu množství pravidel a především výjimek z těchto pravidel, důsledkem čehož je nízká výkonnost výsledného klasifikačního systému. Mnohem lepších výsledků je možno dosáhnout nasazením vhodného algoritmu z oblasti strojového učení, při kterém se množina N číslic ($\mathbf{x}_1, \dots, \mathbf{x}_N$), zvaná *trénovací množina*, použije na nastavení parametrů adaptivního modelu. Identita číslic v trénovací množině je známa dopředu, neboť tato jsou prohlédnuta a ručně zařazena. Ze znalosti požadovaného výstupu je pak možno zkonstruovat *cílový vektor* \mathbf{t} sestávající z identit jednotlivých číslic. Pro náš příklad by cílovým vektorem byl např. binární vektor (tj. vektor, jehož prvky jsou buď nuly, nebo jedničky) o deseti prvcích, představujících deset možných číslic 0...9. Výsledek použití algoritmu strojového učení je možno vyjádřit symbolicky jako funkci $y(\mathbf{x})$, která transformuje vstupní vektor \mathbf{x} na výstupní vektor \mathbf{y} , jež je kódován stejným způsobem jako cílový vektor \mathbf{t} . Přesný tvar funkce $y(\mathbf{x})$ je určen v průběhu tzv. *trénovací* (angl. training) fáze, někdy též zvané fáze *učící*. Při ní jsou, na základě vstupních trénovacích dat, nastavovány vnitřní parametry modelu. Jakmile je jednou model natrénován, je možno ho použít k určení identity nových číslic, které tvoří tzv. *testovací množinu*. Schopnost správně kategorizovat nové případy, které se liší od těch použitých pro natrénování modelu, se nazývá *generalizace*.

Ve většině praktických aplikací bývají původní vstupní proměnné (označovány též jako příznaky, angl. features) *předzpracovány* tak, že jsou transformovány na novou sadu – prostor – proměnných ve kterém, jak všichni doufají, bude problém rozpoznávání snazší vyřešit. Např. v případě rozpoznávání ručně psaných číslic bývají obrázky číslic posunuty a škálovány tak, že každá číslice se nachází v políčku o fixní velikosti. Tento krok významně sníží variabilitu v rámci každé třídy a ulehčí tak algoritmu rozlišit mezi různými třídami. Takovému postupu, kdy jsou vstupní proměnné transformovány z jednoho prostoru do prostoru jiného, se říká *extrakce příznaků* (angl. feature extraction). Jinou možností předzpracování je *výběr příznaků* (angl. feature selection), při němž nedochází k transformaci souřadných systémů vstupních proměnných, ale pouze k výběru vhodné kombinace příznaků z jejich většího počtu. Je třeba zdůraznit, že stejným způsobem jako trénovací množina se musí předzpracovat i testovací množina.

Aplikace, ve kterých trénovací data sestávají ze vstupních vzorů spolu s příslušnými cílovými vektory, jsou známy pod označením *učení s učitelem* (angl. supervised learning). Problémy, jako je kategorizace ručně psaných číslic, se označují

jako *klasifikační* úlohy. Jestliže naopak požadovaný výstup sestává z jedné či více spojitéch hodnot, bývá taková úloha označována jako *regresní*. Pokud trénovací množina sestává pouze ze vstupních vektorů \mathbf{x} bez znalosti cílových vektorů \mathbf{t} , jedná se o *učení bez učitele* (angl. unsupervised learning), při němž je cílem identifikovat skupiny podobných vzorů (jedná se o tzv. *shlukování*, angl. clustering). Ačkoliv každá z těchto úloh vyžaduje své vlastní nástroje a techniky, mají mnohé klíčové principy shodné. Důležité koncepty rozpoznávání vzorů budou představeny na příkladu polynomické regrese.

Předpokládejme, že pozorujeme vstupní spojitou číselnou proměnnou x a chceme předpovědět hodnotu cílové proměnné t . V našem příkladu budeme jako cílovou proměnnou t používat počítačem vygenerovaná data. Trénovací množina sestává z N pozorování proměnné x zapsané jako $\mathbf{x} = (x_1, \dots, x_N)^T$ ⁶ spolu s odpovídajícími hodnotami cílové proměnné t : $\mathbf{t} = (t_1, \dots, t_N)^T$. Trénovací množina \mathbf{x} (viz Obr. 3) byla vytvořena tak, že hodnoty x_n ($n = 1, \dots, N$) pokrývají stejnoměrně interval $[0, 1]$ a prvky t_n cílové datové množiny \mathbf{t} byly získány vyčíslením funkce $\sin(2\pi x_n)$ a následným přičtením náhodného malého šumu. Tímto způsobem vygenerovaná data vykazují zjevnou regularitu, ačkoliv individuální pozorování jsou narušena náhodným šumem.



Obr. 3. Trénovací množina s $N = 10$ body (kroužky), každý z nich sestává ze vstupní proměnné x_n a cílové proměnné t_n . Křivka odpovídá funkci $\sin(2\pi x_n)$ použité při generování dat.

⁶ Horní index T značí, že po transpozici, při níž se prohodí řádky za sloupce, vznikne sloupcový vektor. V lineární algebře se totiž pod zápisem vektoru jako \mathbf{x} rozumí právě vektor sloupcový a chceme-li získat pro nás přirozenější zápis vektoru v řádce (tj. jako x_1, \dots, x_N), je třeba použít notaci \mathbf{x}^T .

Naším cílem je využít tréninkovou množinu pro předpovězení (predikci) cílové hodnoty \hat{t} pro novou, „neznámou“, hodnotu vstupní proměnné \hat{x} . Jedná se tedy vlastně o to, že se budeme snažit, na základě analýzy trénovacích dat, odhalit funkci, která data vygenerovala (tedy funkci $\sin(2\pi x)$). To je v principu složitý problém, neboť trénovací množina je konečná, z čehož plyne nutnost generalizace, a navíc jsou pozorovaná data zatížena šumem, který vnáší nejistotu do predikce \hat{t} .

Pro daný příklad začneme s použitím jednoduchého postupu aproximace křivky, kdy výslednou funkci $y(x)$ budeme aproximovat polynomem:

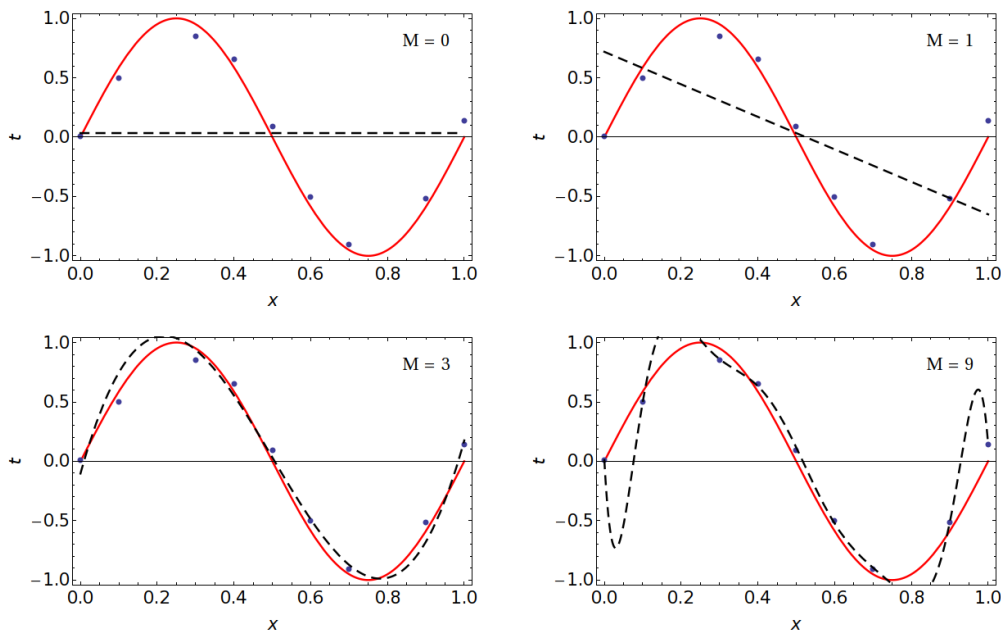
$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \dots + w_M x^M = \sum_{j=0}^M w_j x^j$$

kde M je řád polynomu a x^j je j -tá mocnina x , koeficienty polynomu w_0, \dots, w_M tvoří vektor \mathbf{w} . Ačkoliv je polynomičká funkce $y(x, \mathbf{w})$ nelineární funkcí x , je lineární funkcí koeficientů \mathbf{w} (tj. koeficienty \mathbf{w} se vyskytují pouze v součtech). Takové modely se proto označují jako *lineární modely* (a známá lineární regrese je pak pouze speciálním případem polynomičké regrese se stupněm polynomu $M = 1$). Hodnoty koeficientů \mathbf{w} , které představují vnitřní parametry modelu, jsou určeny proložení polynomu tréninkovými daty tak, že výsledná *chybová funkce* je co nejmenší. Chybová funkce kvantifikuje chybu mezi funkcí $y(x, \mathbf{w})$ a příslušnými body cílové množiny. Běžně používanou chybovou funkcí je suma čtverců chyb mezi predikovanou hodnotou $y(x, \mathbf{w})$ a odpovídající cílovou hodnotou t_n pro každý datový bod x_n :

$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N \{y(x_n, \mathbf{w}) - t_n\}^2$$

Tato míra je vždy nenegativní a nule bude rovna tehdy a pouze tehdy, pokud funkce $y(x, \mathbf{w})$ prochází trénovacími daty přesně. Problém proložení křivky můžeme vyřešit nalezením takového vektoru \mathbf{w} , pro který je chyba $E(\mathbf{w})$ nejmenší možná; toto řešení si označíme jako \mathbf{w}^* a výsledný polynom je tedy symbolicky dán jako funkce $y(x, \mathbf{w}^*)$.

Zatím jsme nastínili způsob, jak získat optimální hodnoty vektoru \mathbf{w}^* pro daný polynom o řádu M , ale volbě stupně polynomu M nebyla doposud věnována pozornost. Výběr řádu polynomu je příkladem důležitého konceptu ve strojovém učení označovaném jako *výběr modelu* (angl. model selection). Obr. 4 znázorňuje čtyři případy, kdy byla data proložena polynomy o řádech $M = 0, 1, 3$ a 9 . Konstantní polynom ($M = 0$) a polynom prvního řádu ($M = 1$) představují zjevně velmi neuspokojivou reprezentaci funkce $\sin(2\pi x)$. Jako velmi dobré řešení se jeví polynom třetího řádu, kdy chyba $E(\mathbf{w}^*)$ je již mnohem nižší. Při použití polynomu řádu $M = 9$ je chyba $E(\mathbf{w}^*)$ nulová. Nicméně je patrné (viz Obr. 4), že ačkoliv chyba je u polynomu třetího řádu nenulová, tak polynom devátého řádu osciluje a je špatnou reprezentací funkce $\sin(2\pi x)$. V případě polynomu $M = 9$ je tento schopen aproximovat trénovací množinu velmi dobře, ale jeho schopnosti generalizace jsou, ve srovnání s $M = 3$, více než mizerné. Toto nežádoucí chování je známo jako *přeučení se* (přetrénování, přefitování, angl. overfitting).



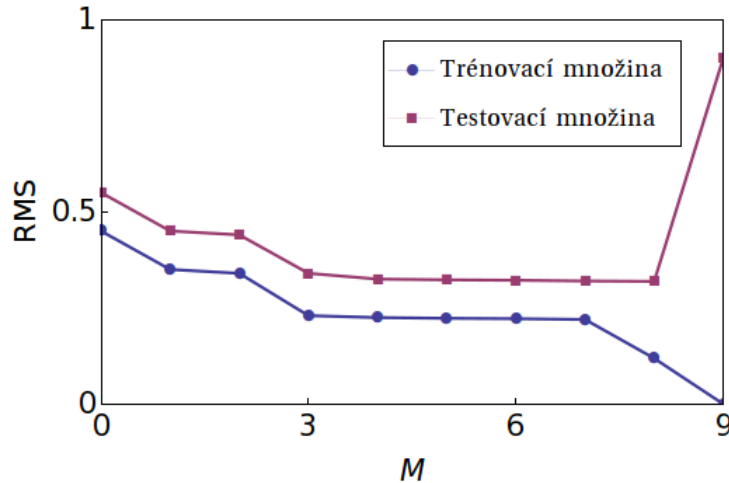
Obr. 4. Grafy polynomů o různých řádech M (šrafované křivky) proložené nagenetovanými daty (viz Obr. 3).

Jak bylo vysvětleno dříve, hlavním cílem je dosáhnout dobré generalizace umožňující predikci nových, systému dříve neprezentovaných dat. Závislosti schopnosti generalizace na stupni polynomu M je možno lépe porozumět, použijeme-li samostatnou datovou množinu, tzv. *testovací množinu*, obsahující 100 bodů, které byly vygenerovány stejně jako body množiny trénovací. Pro každou volbu stupně polynomu M pak můžeme spočítat hodnotu funkce $E(\mathbf{w}^*)$ také pro testovací data, kdy vektor \mathbf{w}^* byl získán naučením systému na datech trénovacích. Místo $E(\mathbf{w}^*)$ se často používá míra *RMS* (angl. root mean square error), definovaná jako:

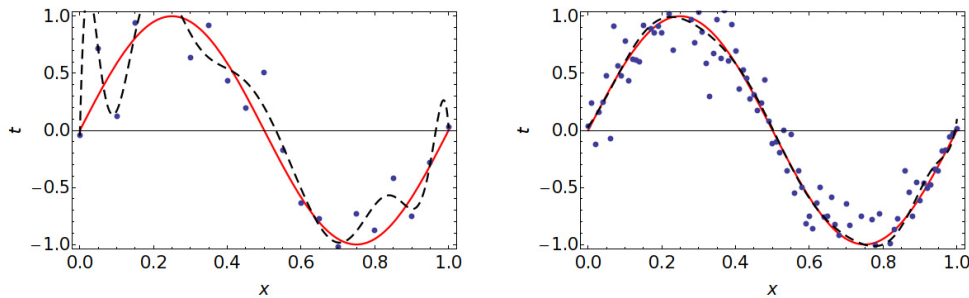
$$RMS = \sqrt{\frac{2E(\mathbf{w}^*)}{N}}$$

kdy dělení počtem prvků množiny N zajišťuje, že je možné mezi sebou porovnávat různě velké množiny a díky druhé odmocnině je *RMS* ve stejném měřítku a jednotkách jako cílová proměnná t . Grafy trénovací a testovací *RMS* chyby pro různé hodnoty stupně polynomu M znázorňuje Obr. 5. Z něj je patrné, že malé hodnoty M dávají velikou chybu jak trénovací, tak testovací, což je důsledkem malé flexibility polynomu, který tak není schopen postihnout oscilace ve funkci $\sin(2\pi x)$. Je-li stupeň polynomu v rozmezí $3 \leq M \leq 8$, chyba pro testovací množinu je rozumně malá a funkce $\sin(2\pi x)$ je reprezentována rozumným způsobem, jak je patrné pro $M = 3$ (viz Obr. 4). Pro řád polynomu $M = 9$ je chyba trénovací množiny rovna nule, neboť polynom obsahuje deset nastavitelných parametrů (tj. 10 koeficientů w_0, \dots, w_9), tzv. *stupňů volnosti*, a tyto jsou vyladěny přesně tak, že perfektně aproximují všech 10 bodů v trénovací množině.

Nicméně výsledná testovací chyba je veliká, neboť aproximovaná funkce vykazuje oscilace (viz Obr. 4).



Obr. 5. Graf trénovací a testovací RMS chyby pro různé hodnoty stupně polynomu M .



Obr. 6. Proložení polynomem o řádu $M = 12$ (čárkovaná černá čára) pro datovou množinu s $N = 20$ (vlevo) a $N = 100$ (vpravo) datovými body. Je patrné, že rostoucí velikost datové množiny potlačuje problém přefitování.

Skutečnost, že systém aproximuje funkci beze zbytku perfektně, pokud má stejný počet stupňů volnosti jako je počet prvků trénovací množiny, je faktem obecným a velmi důležitým. Neplatí pouze pro polynomy, ale pro kterýkoliv regresní algoritmus a je tudíž obrovskou chybou aproximovat funkci s počtem stupňů volnosti rovným počtu prvků v trénovací množině, neřku-li dokonce větším, jak se bohužel často v odborné literatuře stále objevuje. Je proto zajímavé prozkoumat chování našeho polynomického modelu v závislosti na velikosti trénovací množiny (viz Obr. 6). Je patrné, že pro danou komplexitu modelu (danou stupněm polynomu M) se problém přefitování stává méně vážným s růstem velikosti trénovací množiny. Jinými slovy, čím větší máme trénovací množinu, tím komplexnější model si můžeme dovolit použít. Velmi hrubá běžně doporučovaná heuristika říká, že počet datových bodů by neměl být

nižší nežli nějaký násobek (běžně se používá 5 či 10) počtu nastavitelných parametrů modelu. Nicméně počet parametrů modelu není nutně nejvhodnější měrou komplexity modelu a navíc existují algoritmy, které jsou daleko méně náchylné k přetrénování než jiné.

Jak již bylo řečeno dříve, při procesu výběru nejvhodnějšího modelu není správné usuzovat na kvalitu modelu na základě výkonnosti modelu na trénovacích datech. Máme-li k dispozici dostatek dat, je možno rozdělit datovou množinu na množinu trénovací, pomocí níž náš model naučíme, a na množinu *validační*, na níž vyhodnocujeme výkonnost modelu. Je-li ovšem design modelu iterován několikrát na základě validační množiny, pak validační množina vlastně vstupuje do procesu nastavování parametrů modelu a přetrénování se může objevit i vzhledem k množině validační. V takových případech je nezbytné zkonstruovat ještě množinu třetí – testovací – která se nijak neúčastní procesu nastavování parametrů a slouží toliko k evaluaci výsledného modelu. Dlužno podotknout, že v literatuře jsou pojmy validační a testovací množina používány velmi volně, bývají spolu často zaměňovány a je vždy potřeba se pečlivě obeznámit s pozadím experimentu a ujasnit si, zda se data z testovací/validační množiny účastní procesu výběru vhodného modelu.

V mnoha případech je ovšem množství dat limitované, a pokud chceme vystavět co nejkvalitnější model, je žádoucí použít co nejvíce dat pro trénování. Nicméně pokud je validační množina malá, odhad kvality predikce bude více ovlivněn šumem přítomným v datech. Jednou z možností jak určovat kvalitu modelu v těchto případech, je použít tzv. *křížovou validaci* (angl. cross validation). Při ní je datová množina náhodně rozdělena na K podčástí (taková křížová validace se označuje jako *K-fold validace*). Jedna z K podčástí se použije jako validační množina, zatímco ze zbývajících $K - 1$ je zkonstruována trénovací množina. Validační proces je opakován K -krát (běžně používaná je hodnota $K = 10$) s tím, že každý z K vzorků se použije jako validační množina právě jednou. K výsledků validace je poté možno zprůměrovat či jinak zkombinovat, za vzniku jednoho jediného čísla kvantifikujícího výkonnost modelu. Pro křížovou validaci $K = N$ (N je počet datových bodů v množině), při níž je $N - 1$ vzorků použito k trénování modelu a N -tý vzorek k jeho validaci, se vžilo označení *leave-one-out* (LOOCV) validace. Nevýhodou křížové validace je skutečnost, že s rostoucím K roste i počet modelů, jež je třeba natrénovat (s nejvíce extrémním případem LOOCV, která je již opravdu velmi drahá).

Jako alternativa se proto používají různá kritéria založená na teorii informace, jako je např. Akaikovo informační kritérium [2], která umožňují porovnávat různá nastavení modelu na základě jednoho jediného natrénování. Nevýhodou těchto kritérií je ovšem skutečnost, že neberou v potaz nejistotu v parametrech modelu a v praxi upřednostňují až příliš jednoduché modely. Nejlepším způsobem výběru vhodného modelu jsou pak postupy založené na Bayesovské interpretaci pravděpodobnosti, které jdou však již nad rámec tohoto úvodního textu.

V následující kapitole budou vysvětleny základní principy populárního klasifikačního a regresního modelu, kterým jsou umělé neuronové sítě.

3 Neuronové sítě

Umělé *neuronové sítě* (angl. artificial neural networks, ANN) představují stále aktivně se vyvíjející vědeckou oblast spadající do oboru umělé inteligence a mající své základy v neurobiologii. Neuronové sítě jsou modely, jež se pokoušejí napodobovat některé ze základních informací zpracovávajících pochodů odehrávajících se v mozku. Neuronové sítě našly své uplatnění především v modelování komplexních problémů. Oblast neuronových sítí se vyvinula od modelování nejjednodušších prvků (*neuronů*) zpracování a přenosu informace v mozku až po konstruování masivně paralelních sítí. V této části budou představeny základní koncepty neuronových sítí, podrobněji vysvětlen princip fungování vícevrstevných neuronových sítí a nakonec bude čtenář obeznámen s jejich využitím v bioinformatice.

3.1 Neuronové sítě a jejich schopnosti

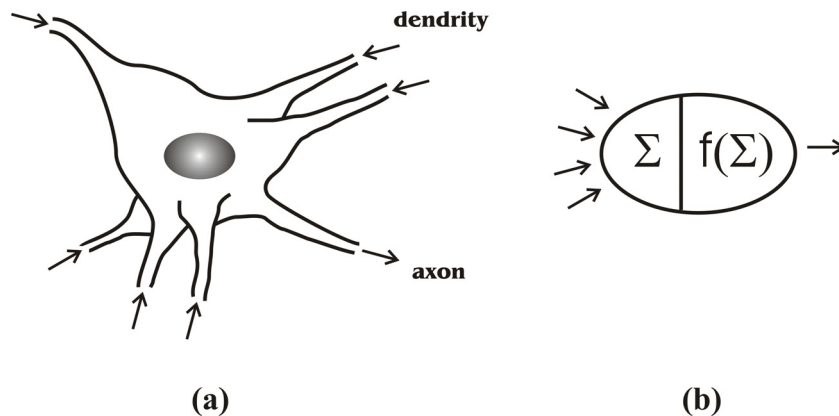
Dle širší definice neuronových sítí se jedná o soubor navzájem pospojovaných jednotek – neuronů – které jsou schopny extrahovat lineární i nelineární trendy z dat, tyto zakódovat do svých vnitřních parametrů a na jejich základě potom poskytovat spolehlivé predikce. Neurony jsou uspořádány do paralelní sítě, jejíž funkce je dána strukturou (tj. jak jsou neurony zorganizovány a navzájem propojeny), silou spojující mezi neurony (udávanou číselným parametrem, tzv. váhou w) a způsobem, jakým je vstupní signál zpracován neuronem na signál výstupní. Vzájemné uspořádání neuronů v rámci jedné sítě je označováno jako *architektura sítě*.

Existuje mnoho různých typů a architektur neuronových sítí, některé z nich fungují v módu učení s učitelem, jiné v módu učení bez učitele. Neuronové sítě jsou vhodné pro obě dvě základní úlohy strojového učení, tj. pro klasifikaci i pro regresi. Co se regrese týče, neuronové sítě jsou schopny proložit libovolně komplexní závislosti s libovolnou přesností a z tohoto důvodu bývají označovány za univerzální aproximátory [3]. Cennou vlastností neuronových sítí bez učitele je schopnost *samoorganizace* (angl. self-organization), kdy neuronové sítě nejen že naleznou shluky v datech, ale podchytí i jejich vzájemné prostorové vztahy. Zajímavou vlastností neuronových sítí je i jistá robustnost vůči změnám v jejich vnitřním uspořádání. Naučená znalost je v síti zakódována jako množina vah, kdy velikosti jednotlivých vah určují chování celé sítě. Odstraníme-li kterýkoliv z neuronů či kteroukoliv z vah, neuronová síť je, podobně jako lidský mozek, stále schopna fungovat, sice se zmenšeným výkonem. Z hlediska funkcionality představují neuronové sítě rozšíření schopností technik vícerozměrné statistické analýzy, jako je např. lineární a nelineární regrese, klastrové analýzy, metody hlavní komponenty apod. Mezi nejvíce používané a nejznámější neuronové sítě patří:

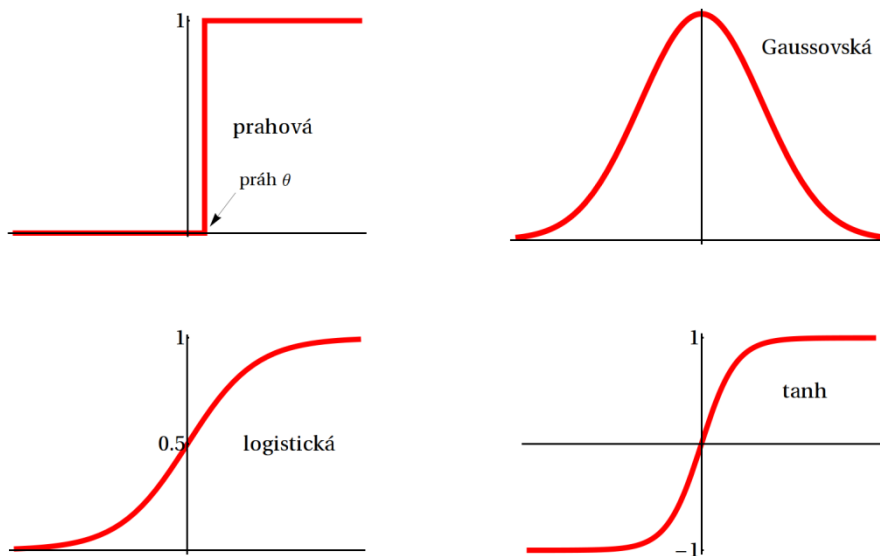
1. Jedno- či vícevrstevná síť s dopředným šířením signálu, která používá učení s učitelem a lze ji použít jak pro úlohy regrese, tak pro úlohy klasifikační
2. Kohonenova samoorganizující se mapa (angl. Self-Organizing Map, SOM), což je síť vhodná k odhalování shluků v datech a jejich prostorových souvislostí a představuje typ sítě trénované pomocí učení bez učitele.

3.2 Neuron a perceptron

Jak již bylo řečeno výše, základní stavební jednotkou umělých neuronových sítí je neuron. Ačkoliv umělý neuron a neuron v mozku nejsou zdaleka stejné a ani nevykonávají stejnou funkci, oba dva vykazují jisté podobnosti, které rozhodně stojí za povšimnutí. Biologický neuron sestává ze tří hlavních komponent (viz Obr. 7a): (1) z dendritů, které posílají vstupní signály, jež jsou váženy dle síly propojení, do těla neuronu, (2) z těla, které akumuluje vážené vstupní signály a dále je zpracovává a (3) z axonu, který přenáší výstupní signál do dalších neuronů k němu připojených.



Obr. 7. Neuron a jeho reprezentace: (a) biologický neuron, (b) matematický model neuronu.



Obr. 8. Různé typy aktivačních funkcí. Gaussovská aktivační funkce se používá v sítích RBF (angl. Radial Basis Function).

Tento proces je idealizován v umělém neuronu ukázaném na Obr. 7b, ve kterém se akumulují vážené vstupní signály (tj. jsou posčítány Σ) a následně jsou zpracovány dále (tj. jsou modifikovány pomocí funkce $f(\Sigma)$, jež je označována jako aktivační či *přenosová funkce*) za vzniku výstupního signálu. Výstupní signál je přenesen na další napojené neurony (kde tedy figuruje jako signál vstupní) a celý proces se opakuje, až je dosaženo konečného výstupu ze sítě. Váhy hrají v celém procesu významnou úlohu, neboť změnami jejich hodnot je síť schopna adaptace, tj. je schopna odhalit a naučit se vzory v datech se nalézající. Čím větší je váha, tím silnější je spojení mezi příslušnými dvěma neurony a tím více je jeden neuron aktivován neuronem druhým.

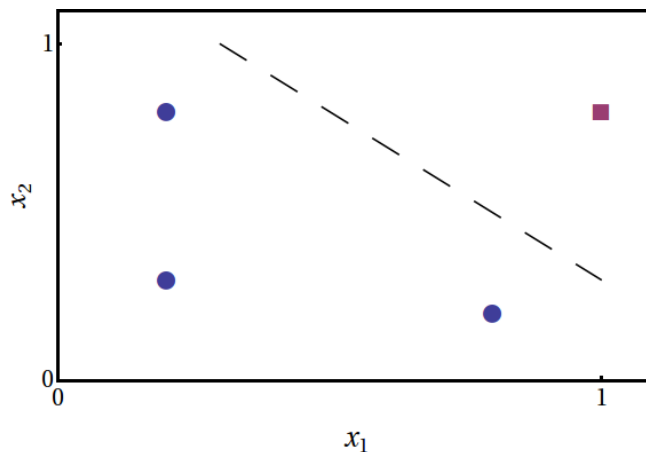
Nejjednodušší aproximací biologického neuronu je prahový neuron (angl. *threshold neuron*), který byl vyvinut pány McCullochem a Pittsem v roce 1940 [4]. Používá jednoduchou aktivační funkci ve formě nastaveného prahu, jež transformuje vstupní hodnotu buď na 0 či na 1 (viz Obr. 8).

V době návrhu tohoto neuronu nebylo známo, jak nastavit jeho váhy (tj. jak takovýto neuron učit) a tyto byly tedy fixovány na předem zvolených hodnotách. Přesto McCulloch a Pitts ukázali, že dokonce i síť složená z takovýchto jednoduchých neuronů je schopna vykonávat i značně komplexní klasifikační úkoly [3].

Fungování neuronu bude demonstrováno na jednoduchém klasifikačním problému (viz Tabulka 1 a Obr. 9).

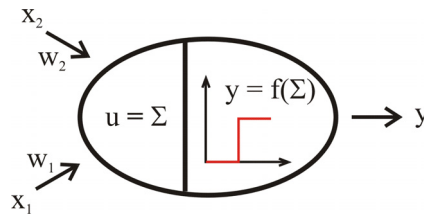
Tabulka 1. Data pro ukázkový klasifikační příklad.

| x_1 | x_2 | třída t |
|-------|-------|-----------|
| 0,2 | 0,3 | 0 |
| 0,2 | 0,8 | 0 |
| 0,8 | 0,2 | 0 |
| 1,0 | 0,8 | 1 |



Obr. 9. Grafická reprezentace ukázkového klasifikačního problému. Čárkovanou čarou je znázorněno klasifikační rozhraní prahového neuronu.

Tento problém sestává ze dvou vstupů (x_1 a x_2) a jednoho výstupu (t), který nabývá jedné ze dvou možných hodnot (0 či 1). Jedna řádka tabulky je označována jako vstupní vektor či vstupní vzor (angl. pattern). Cílem našeho jednoduchého modelového systému je správně klasifikovat vstupní vzory do dvou možných tříd 0 a 1 za pomoci prahového neuronu. Neuron má tedy dva vstupy (viz Obr. 10), jimž jsou přiřazeny váhy w_1 a w_2 , které jsou ovšem fixovány na hodnotě 1.0, systém se tedy nebude nijak učit.



Obr. 10. Konfigurace prahového neuronu pro klasifikaci ukázkového problému. Váhy w_1 a w_2 jsou fixovány na hodnotě 1.0.

Výpočet pomocí takového neuronu zahrnuje dva kroky:

1. Nejprve se vypočítá vstupní hodnota u jako suma vážených vstupů:

$$\Sigma = u = w_1 x_1 + w_2 x_2 = 1.0 \times x_1 + 1.0 \times x_2$$

Hodnota u nám pomůže při umístění prahu aktivace. Neboť suma dvou vstupů je v jedné kategorii rovna 1.8 a v druhé kategorii 0.5, 1.0 a 1.0, je zřejmé, že hodnota prahu musí být větší než 1.0 a menší než 1.8. Takže si můžeme zvolit hodnotu prahu θ třeba rovnu 1.3.

2. Poté je pomocí aktivační funkce $f(\Sigma)$ vypočtena výstupní hodnota y :

$$f(\Sigma) = y = \begin{cases} 0 & u < 1.3 \\ 1 & u \geq 1.3 \end{cases}$$

Neuron klasifikuje data správně díky našemu rozhodnutí nastavit jeho práh na hodnotu 1.3. Protože umístění prahu definuje obě dvě kategorie, $u = 1.3$ představuje tzv. klasifikační hranici, kterou je možno formulovat jako:

$$u = x_1 + x_2 = 1.3$$

$$x_2 = 1.3 - x_1$$

Jedná se tedy o přímku (viz Obr. 9) a problému, který je možno separovat přímkou, se říká problém *lineárně separovatelný*.

Pro jednoduchost je výhodné použít maticovou notaci, kdy vstupní vektor \mathbf{x} má komponenty x_1 a x_2 : $\mathbf{x} = (x_1, x_2)$ a podobně váhy mohou být uspořádány do váhového vektoru $\mathbf{w} = (w_1, w_2)$. Vážený součet vstupů $u = x_1 w_1 + x_2 w_2$ je pak reprezentován jako skalární součin $u = \mathbf{w} \cdot \mathbf{x}$.

Hlavní nevýhodou prahového neuronu je samozřejmě fakt, že tento není schopen adaptivně nastavovat své váhy na základě trénovacích dat. V roce 1949 psycholog Donald Hebb navrhl mechanismus adaptace vah neuronu, který je dnes znám jako *Hebbovské učení* [5]. Hlavním principem tohoto učení je, že změna váhy spojení mezi dvěma neurony je proporcionální součinu výstupů z těchto neuronů. To tedy znamená, že silnější excitace mezi dvěma neurony (tj. zvýšení jejich výstupů x a y) vede ke vzrůstu vah mezi nimi. Matematicky se změna vah Δw dá napsat jako $\Delta w = \beta x \cdot y$ a nová hodnota vah pak jako $w_{new} = w_{old} + \Delta w = w_{old} + \beta x \cdot y$, kde konstanta β se označuje jako *rychlost učení* (angl. learning rate). Platí, že čím vyšší β , tím rychlejší bude změna vah a naopak.

Výzkum učícího procesu v neuronových sítích se datuje zpátky k Frankovi Rosenblattovi, který v 50. letech 20. století položil základy oboru adaptivních neuronových výpočtů. Uvědomil si, že prahový neuron není vhodný pro modelování funkce činnosti mozku, neboť není dostatečně flexibilní a není schopen se učit a adaptovat na své prostředí. V důležité práci [4] Rosenblatt navrhl první model neuronu – *perceptron* – který byl schopen adaptivně modifikovat své váhy na základě dat v trénovací množině. V zásadě aplikoval Hebbovské učení na prahové neurony.

Rosenblattův perceptron má pouze dvě vrstvy – vstupní a výstupní a pracuje v režimu klasifikace. Vstupní vrstva, která vlastně slouží pouze k přivedení vstupů do sítě, se do celkového počtu vrstev nezapočítává, a proto se takovýto perceptron označuje jako *jednovrstvá neuronová síť*⁷. Síť perceptronů se učí z trénovacích dat a neurony v procesu učení mění své váhy. Neurony v perceptronu jsou prahové neurony. Neuron obdrží více vstupů, zpracuje je a vyprodukuje jeden výstup.

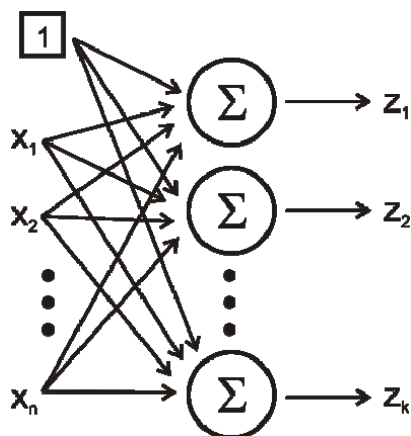
Jeden ze vstupů do neuronu, tzv. *bias*, se pokládá rovný jedné, tj. do hry vstupuje pouze jeho váha jako náhrada hodnoty prahu, a je tedy podobný konstantnímu členu v regresní analýze. Pro perceptron je využíváno učení s učitelem a zpracování signálu se v něm děje následujícím způsobem: váhy jsou inicializovány náhodným způsobem, poté je síti prezentován první vstupní vzor a proběhne příslušné zpracování, při němž jsou vstupy vynásobeny vahami, poté posčítány a upraveny prahovou funkcí na výstup. Pokud je vážená suma větší nebo rovna prahu (v případě Rosenblattova perceptronu je tento roven nule), je výstupem 1, jinak je výstupem 0. Výstup z perceptronu je poté porovnán s cílovým výstupem t pro daný vstupní vektor, a pokud je klasifikace chybná, jsou upraveny váhy aplikací Hebbovského učícího algoritmu.

3.3 Vícevrstvé sítě a algoritmus zpětného šíření chyby

Až doposud neurony, o nichž jsme mluvili, měly pouze dvě možné výstupní hodnoty: 0 a 1. V tzv. *lineárním neuronu* je výstup spojitý a pod pojmem lineární se rozumí, že výstup je lineární sumou vážených vstupů. Takový neuron je možno použít nejen pro klasifikaci, ale i pro aproximaci funkce (tj. pro regresi). První adaptivní lineární neuron ADALINE, navržený Widrowem a Hoffem [6], se učil pomocí pravidla známého jako *delta pravidlo*. Podle tohoto pravidla se váhy nastavují metodou “gradient descent” tak,

⁷ Zde opět může dojít ke zmatení pojmů, neboť někteří autoři naopak vstupní vrstvu započítávají a takovou síť pak označí za dvouvrstvou.

že se minimalizuje kvadrát chyby mezi skutečným a požadovaným výstupem. Lineární neurony je možno kombinovat do sítě s více výstupy (viz Obr. 11). Při klasifikaci každý neuron odpovídá jedné třídě. V regresních úlohách taková síť simultánně modeluje lineární vztahy mezi jednou či více výstupními proměnnými a několika proměnnými vstupními.



Obr. 11. Lineární neuronová síť s několika výstupními neurony. Čtvercový vstupní neuron s číslem jedna představuje bias.

Zatím jsme se zabývali neurony s prahovou či lineární aktivační funkcí. Nelinearitu lze zavést použitím neuronu s nelineární aktivační funkcí. Výstup z neuronu je pak dán jako vážená suma vstupů, která je transformována nelineární funkcí σ :

$$\sigma\left(\sum_{j=1}^n w_j x_j + b\right)$$

kde b představuje váhu přiřazenou biasu a w_j reprezentuje váhu j -tého vstupu. Nelineární aktivační funkce (viz Obr. 8), používané v neuronových sítích, vykazují některé důležité vlastnosti. Jsou spojité (nejsou na nich tedy žádné “ostré” hrany a jsou všude diferencovatelné, kterážto vlastnost je extrémně důležitá pro možnost jejich trénování) a jsou shora a zdola omezené (výstup je vždy omezen v určitém úzkém intervalu a nemůže dosáhnout extrémně velkých hodnot).

Nejpopulárnější nelineární aktivační funkcí je tzv. *sigmoida*, takto nazvaná neboť její tvar připomíná písmeno S. Nejvíce používanou sigmoidou je pak *logistická* funkce (viz Obr. 8):

$$y = L(u) = \frac{1}{1 + e^{-u}}$$

kde vstupem u je vážená suma vstupních hodnot $u = \sum_{j=1}^n w_j x_j + b$. Tato funkce je zdola omezená hodnotou nula a shora omezená hodnotou jedna; je-li vstupní hodnota

$u = 0$, pak výstupem $L(u)$ je hodnota 0.5. Jinou běžně používanou sigmoidní funkcí je hyperbolický tangens:

$$\tanh(u) = \frac{e^{2u} - 1}{e^{2u} + 1}$$

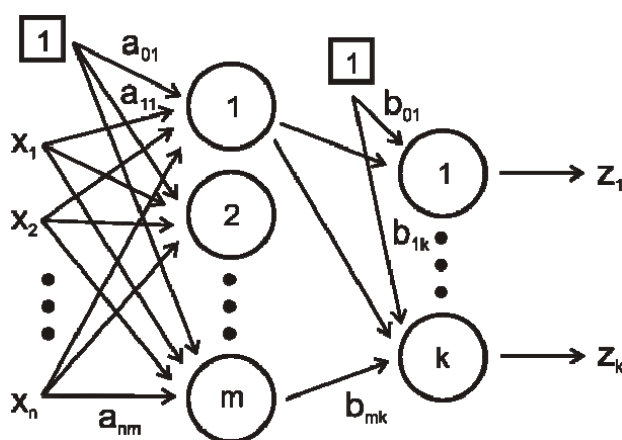
která je omezena hodnotami -1 a +1 a má v bodě $u = 0$ nulovou hodnotu (viz Obr. 8).

Při používání těchto funkcí ve výstupních neuronech je třeba si uvědomit, že výstup z neuronové sítě je porovnáván s požadovaným cílovým výstupem t a že tento tedy musí být uzpůsoben tak, aby odpovídal rozsahu danému přenosovou funkcí. Např. při použití neuronu s logistickou funkcí musí být cílový výstup t změněn tak, aby ležel v intervalu $\langle 0, 1 \rangle$. Vstupy, na druhou stranu, mohou v principu nabývat libovolných hodnot bez ohledu na rozsah aktivační funkce, ačkoliv mnohdy bývá výhodné upravit i vstupní hodnoty.

Uspořádáním nelineárních neuronů do jednovrstvé sítě získáme de facto jiný typ perceptronu. Obecně však platí, že jednovrstvé perceptrony jsou schopny řešit pouze lineárně separovatelné problémy (a to dokonce i tehdy, použijeme-li v nich nelineární neurony), jak bylo ukázáno v roce 1969 ve známé knize *Perceptrons* od Marvinina Minskyho a Seymoura Paperta [7]. Většina jejich čtenářů bohužel učinila nesprávný závěr, že podobné výsledky jsou pravdivé i pro vícevrstvé perceptrony a kniha se tak nezaslouženě zapříčinila o deset let trvající pokles zájmu o neuronové sítě. Až teprve práce Werbose [8] a Rumelharta, Hintona a Williamse [9] položily základy k úspěšné implementaci adaptace ve vícevrstevných neuronových sítích.

Přidáním další vrstvy, tzv. vrstvy *skryté*, mezi vstupní a výstupní vrstvu získáme dvouvrstvý perceptron. Její přítomnost zajišťuje schopnost sítě aproximovat nelineární vztahy, neboť právě na ní dochází ke spojitě nelineární transformaci vážených vstupů. Skrytých vrstev může být i více a obecně se takové sítě označují jako *vícevrstvé perceptrony* (MLP, angl. multi-layer perceptron). Ty jsou velmi flexibilní a jsou schopny aproximovat jakýkoliv nelineární vztah mezi vstupy a výstupy s libovolnou přesností, a to dokonce již za použití pouze jedné skryté vrstvy [3].

Pro běžné aplikace se používá dvouvrstvý perceptron (tj. perceptron s jednou skrytou vrstvou, viz Obr. 12).



Obr. 12. Architektura dvouvrstvé neuronové sítě.

V obecném případě má síť n vstupů, m skrytých neuronů a k výstupů. V případě regresních problémů se většinou používá pouze jeden výstupní neuron a v případě klasifikačních úloh se použije tolik výstupních neuronů, do kolika tříd je potřeba vstupní vzory klasifikovat. Vstupy do sítě jsou dány jako x , váhy mezi vstupní a skrytou vrstvou si označíme jako a a váhy mezi skrytou a výstupní vrstvou jako b a výstupy ze sítě si označíme jako z . Bias skrytých neuronů se modeluje jako další neuron (s výstupem rovným jedničce) ve vstupní vrstvě a váhy nesou označení $a_{01\dots 0m}$ a bias výstupního neuronu se modeluje jako neuron (bez vstupů a s jedničkovým výstupem) ve vrstvě skryté a jeho váhy nesou označení $b_{01\dots 0k}$. Vstup do j -tého skrytého neuronu u_j a výstup y_j z něj se pak vypočítají jako:

$$u_j = a_{0j} + \sum_{i=1}^n a_{ij}x_i$$

$$y_j = \frac{1}{1 + e^{-u_j}}$$

Výstupy z neuronů ve skryté vrstvě slouží jako vstupy do výstupních neuronů a vážená suma vstupů v_k a výstup z_k k -tého výstupního neuronu se vypočte jako:

$$v_k = b_{0k} + \sum_{j=1}^m b_{jk}y_j$$

$$z_k = \frac{1}{1 + e^{-v_k}}$$

Za přenosovou funkci neuronů v skryté vrstvě a ve vrstvě výstupní byla v tomto příkladě zvolena funkce logistická, nicméně je nutno podotknout, že tvar této funkce

může být libovolný a v principu je možno i použít jinou aktivační funkci pro neurony skryté a jinou pro neurony výstupní.

Nezodpovězenou otázkou u dvouvrstvého perceptronu stále zůstává, jak je tato síť trénována, tj. jakým algoritmem jsou nastavovány váhy tak, aby výstup ze sítě z odpovídal co nejvíce požadovanému cílovému výstupu t . Klíčovou roli při odvození trénovacího algoritmu hraje *kvadrát průměrné chyby* (angl. Mean Square Error, MSE), jenž je možno pro N trénovacích vzorů a jeden výstupní neuron vypočítat jako:

$$MSE = \frac{1}{2N} \sum_i^N (t_i - z_i)^2$$

Hodnota z je dána aktivační funkcí výstupního neuronu a záleží tedy na vahách mezi neurony skrytými a výstupním a na výstupech skrytých neuronů. Ty jsou pak dány vahami mezi vstupními a skrytými neurony a hodnotami vstupů do sítě. Závislost MSE na vnitřních parametrech sítě (vahách) je mnohorozměrný povrch, tzv. *chybový povrch* (angl. error surface). Optimální hodnoty vah jsou takové, pro které je hodnota MSE co nejnižší. To znamená, že na chybovém povrchu hledáme minimum, k čemuž se nejčastěji používá numerická minimalizační metoda nejprudšího spádu (angl. *steepest descent*). Metoda nejprudšího spádu používá záporně vzatou směrnicí chybového povrchu danou jeho gradientem (vektor prvních derivací chybového povrchu vzhledem ke všem nastavitelným parametrům sítě, udává směr nejprudšího nárůstu funkce) pro nalezení směru, ve kterém mají být upraveny váhy tak, aby bylo co nejrychleji dosaženo nejbližšího minima. Tento přístup používá nejpoužívanější trénovací algoritmus pro vícevrstvé síť – algoritmus *zpětného šíření chyby* (angl. backpropagation) [8, 9]. Při něm se postupuje tak, že se nejprve pro daný vstupní vzor vypočte výstup ze sítě (pro úplně první vzor prezentovaný síti se váhy nastaví na začátku učení náhodně), získá se hodnota MSE a poté se upravují zpětně váhy tak, že se nejprve upraví váhy mezi výstupní a skrytou vrstvou a posléze váhy mezi skrytou a vstupní vrstvou (postupuje se tedy „odzadu“, proto algoritmus zpětného šíření chyby). Jak daleko se tyto váhy posunou je dáno parametrem rychlosti učení ϵ , jehož hodnota záleží na studovaném problému a volí se obvykle mezi 0 a 1. Malá hodnota ϵ vede k příliš pomalému učení, na druhou stranu příliš velká hodnota může vést k „přeskočení“ hledaného minima a v konečném důsledku také k prodloužení trénování sítě. Je-li nová změna vah po epoše m (pod pojmem epocha je rozuměno vložení nového tréninkového vzoru na vstup sítě) označena jako Δw_m , pak je možno váhy w_{m+1} v epoše $m + 1$ reprezentovat jako:

$$w_{m+1} = w_m + \Delta w_m$$

$$\Delta w_m = -\epsilon d_m$$

kde znaménko (–) představuje pohyb ve směru opačném směru gradientu a ϵd_m je vzdálenost, o kterou se v tomto směru pohneme.

Dalším používaným parametrem, jehož úkolem je stabilizovat proces hledání optimálních hodnot vah, je tzv. *moment* μ . Změnu vah s použitím momentu je možno psát jako:

$$\Delta w_m = \mu \Delta w_{m-1} - (1 - \mu) \varepsilon d_m$$

kde moment nabývá hodnot mezi 0 a 1 a Δw_{m-1} je změna vah v předchozí epoše. Druhý člen na pravé straně obsahuje množství změny vah v epoše m , nicméně tentokrát vážené členem $(1-\mu)$. Tudíž o změně vah nerozhoduje pouze hodnota aktuálního gradientu, ale také velikost předchozí změny vah. Volbou hodnoty μ je tedy možno měnit chování systému: je-li $\mu = 0$, moment nehraje žádnou roli a předchozí změna se vůbec nebere v potaz, naopak je-li $\mu = 1$, je aktuální změna založena pouze na změně předchozí.

V metodě zpětného šíření chyby se všechny váhy mění stejnou rychlostí ε . Nicméně ve skutečnosti některé váhy mohou být blíže optimu a mohou mít tedy větší vliv na chybu než ostatní. Vyšší flexibility a rychlejší konvergence je možno dosáhnout adaptivní změnou rychlosti učení pro každou z vah nezávisle. Metoda adaptivní změny rychlosti učení je známa jako *delta-bar-delta* [10], někdy též TurboProp. Při této metodě má každá z vah svoji rychlost učení a ta se mění v průběhu trénovacího procesu tak, že pokud má aktuální směr změny na chybovém povrchu stejný směr jako pohyb předchozí, pak je rychlost učení zvýšena a v opačném případě je snížena.

Kromě směru nejprudšího poklesu chybového povrchu je možno brát také do úvahy i jeho zakřivení (tj. rychlost, jakou se mění vlastní rychlost změny povrchu daná jeho gradientem v daném bodě) v okolí aktuálního bodu. To je dáno maticí druhých derivací - *Hessiánem*. Mezi metody 2. řádu patří např. Quickprop [11] nebo metoda Levenberg-Marquardtova (LM) [12]. Rychlost konvergence těchto metod (především LM metody) ve srovnání s metodami 1. řádu (tedy gradientových metod) je ohromující, nicméně takto vysokého výkonu je dosaženo za cenu výrazně vyšších výpočetních nároků. Výpočet Hessiánu (a jeho následná inverze, která je principiálně těmito metodami vyžadována) pro veliké sítě učené na velikých souborech dat je náročný jak z hlediska hardwarových nároků (paměti), tak z hlediska rychlosti. Navíc inverze Hessiánu může vést k problémům s numerickou nestabilitou a v tomto případě pak metody nemusí podávat uspokojivý výkon.

Zatímco počet vstupních a výstupních neuronů bývá diktován řešeným problémem, počet skrytých neuronů nelze z povahy úlohy takto snadno odvodit. Počet skrytých neuronů určuje komplexitu rozhodovací hranice. Jsou-li vzory dobře separovány, je zapotřebí menšího množství skrytých neuronů a s rostoucí komplexitou úlohy roste i počet skrytých neuronů v síti. Počet vnitřních neuronů určuje počet vah (stupňů volnosti) v síti a, jak již bylo řečeno výše, je třeba zajistit, aby počet stupňů volnosti nepřesáhl počet vzorů v trénovací množině N . Jako empirické pravidlo, které se ukázalo fungovat správně v mnoha různých aplikacích, se často uvádí, že je potřeba zvolit počet skrytých neuronů tak, aby počet vah nepřesáhl $N/10$. Jedná se nicméně o velmi přibližné pravidlo, neboť mnoho úspěšných systémů používá více skrytých neuronů. Jiným přístupem je natrénovat několik sítí s postupně rostoucí velikostí skryté vrstvy a monitorovat chybu na testovací množině, jež u optimálního počtu skrytých neuronů dosáhne svého minima. Více "zásadovou" metodou je vyladit komplexitu sítě na základě trénovací množiny. Sem patří např. postup, kdy se začne s velkým množstvím skrytých neuronů a postupně se pak eliminují váhy či skryté neurony (angl. *pruning*).

4 Aplikace neuronových sítí v bioinformatice

Aplikace neuronových sítí pro analýzu biologických sekvencí má za sebou již dosti dlouhou historii. V roce 1982 byl úspěšně aplikován jednovrstvý perceptron na predikci vazebných míst ribozomů na základě nukleotidové sekvence jako vstupu [13]. Kvalita tohoto systému přesáhla předchozí ručně vyvinutý systém heuristických pravidel. Lineární architektura perceptronu je nicméně, pro většinu úloh rozpoznávání sekvencí, nedostatečná. Skutečné oživení v aplikacích neuronových sítí v bioinformatice znamenalo použití vícevrstvého perceptronu s trénovacím algoritmem zpětného šíření chyby v roce 1988, kdy Qian a Sejnowski úspěšně predikovali sekundární strukturu proteinů na základě jejich sekvence [14]. V následujících kapitolách budou popsány specifika aplikace vícevrstvých neuronových sítí na bioinformatické sekvenční problémy.

4.1 Kódování sekvencí jako vstupu do neuronové sítě

Důležitým problémem při aplikaci neuronových sítí na sekvenční data je kódování sekvencí biomolekul. Je-li zvolena vhodná reprezentace, která odhaluje hlavní principy dané úlohy, pak je často možno problém vyřešit jednoduchými lineárními prostředky. Ideální kódovací schéma by mělo extrahovat maximální množství informace a současně zachovávat základní předpoklad, a to sice, že podobné sekvence jsou kódovány sobě si blízkými vektory. Kódovací schémata se dělí obecně na kódování *lokální* (ta zahrnují pouze jednotlivá rezidua či krátké sekvence představující okolí – sekvenční kontext – centrálního rezidua) a *globální* (ta zahrnují dlouhé sekvenční úseky či sekvence plné délky).

V mnoha problémech sekvenční analýzy řešených neuronovými sítěmi je vstupem buď kompletní sekvence, nebo je vstup generován nepřekrývajícím se okénkem n sousedních residuí, tzv. *n-gramy*. Okénko tedy rozseká sekvenci na kratší sekvence délky n , z nichž každá představuje jeden vstupní vektor. Okénko bývá typicky umístěné tak, že upstream a downstream sekvence⁸ mají stejnou velikost, nicméně v některých případech (např. při predikci míst sestřihu v pre-mRNA) podává asymetrické okénko lepší výkon než okénko symetrické. Okénka s mezerami, která představují nesouvislou sekvenci, byla použita pro predikci promoterů nebo pro predikci počátečních míst transkripce. Na každé pozici v okénku může být A monomerů, tj. pro nukleové kyseliny je $A = 4$ a pro proteiny je $A = 20$.

V tzv. *přímém kódování* je každé jednotlivé reziduum převedeno na jeho vektorovou reprezentaci. Nejpoužívanější takovou reprezentací je *ortogonální kódování*, kdy jsou písmena abecedy (tj. např. 4 písmena pro nukleové kyseliny a 20 písmen pro proteiny) kódována ortogonálními binárními vektory. Tato kódování se označují jako BIN4, BIN20 apod. V této reprezentaci, která má výhodu v tom, že nevnáší žádné korelace mezi monomery, mohou být čtyři nukleotidy kódovány např. jako 1000 (A), 0100 (T), 0010 (G) a 0001 (C). Nevýhoda toho kódovacího schématu je

⁸ Upstream sekvence je sekvence směrem k 5' konci od nějakého konkrétního místa (např. od místa, na němž začíná transkripce). Downstream sekvence je pak sekvence směrem ke 3' konci.

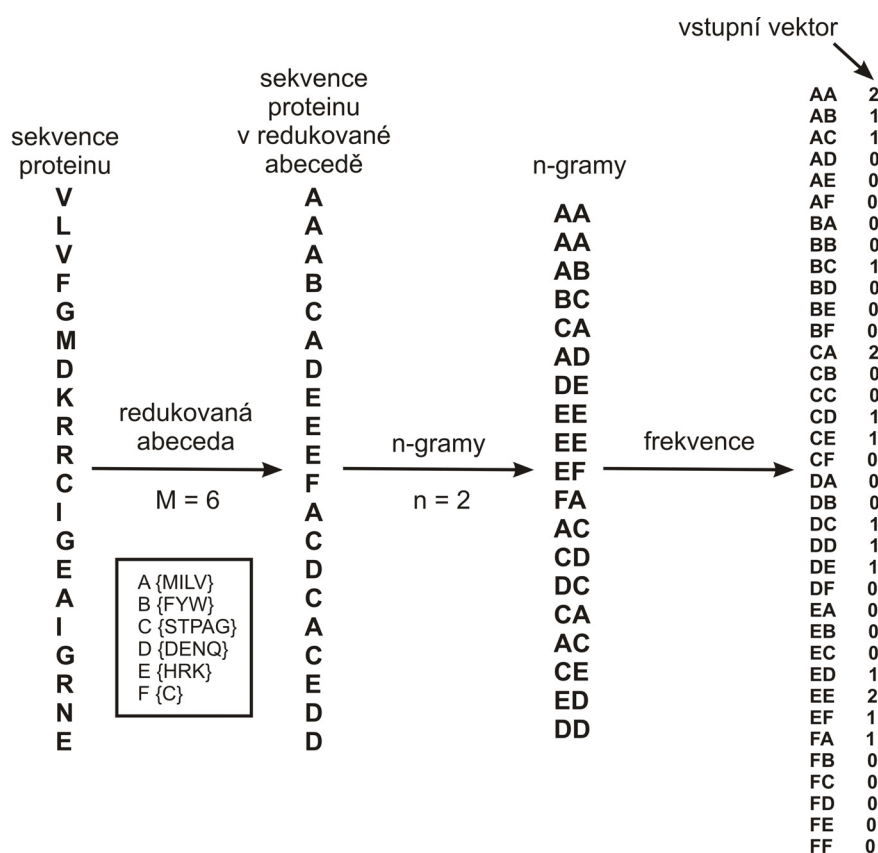
přítomnost velkého množství nul v zakódované sekvenci a z toho pramenící nehospodárnost, neboť vstupní vrstva neuronové sítě pak musí mít $A \times n$ neuronů. A písmen může být v principu kódováno pouhými $\log_2 A$ binárními jednotkami (např. 00 pro A, 01 pro T, 10 pro G a 11 pro C) nebo dokonce, v případě, kdy jsou použity spojité a nikoliv binární vstupy, jednou jedinou jednotkou (čtyři různá desetinná čísla). Takováto kompaktní kódovací schémata vedou nicméně k dramatickému nárůstu nelinearity problému, neboť jejich použitím jsou do systému zavlečeny korelace nemající žádnou biologickou relevanci. V přímém kódování je možno také reprezentovat jednotlivá rezidua reálnými čísly odpovídajícími jejich fyzikálně-chemickým vlastnostem (např. objem, jež reziduum zaujímá, jeho náboj apod.). V tomto přístupu je ovšem výkon často degradován opět díky přítomnosti nežádoucích korelací. Z výše řečeného je tedy patrné, že mezi kvalitou kódování a výkonem neuronové sítě je třeba činit kompromisy. Jako jednoznačný vítěz vychází z boje ortogonální reprezentace. U všech ostatních kódovacích schémat, zahrnujících dodatečnou informaci, je nutno tuto odfiltrout na úrovni skryté vrstvy, tzn., je třeba zvýšit počet skrytých neuronů. Pokud je navíc zahrnuto přebytečné množství fyzikálně-chemické informace, která nemusí nutně souviset s požadovaným výstupem, dojde k výraznému zhoršení učících schopností sítě.

Důležitou strategií vedoucí ke snížení nelinearity problému je použití reprezentaci založenou nikoliv na jednotlivých reziduiích, ale na dimerech (2 rezidua vedle sebe) či na trimerech (3 rezidua vedle sebe). Použití dimerů či trimerů má i svoji biochemickou relevanci. Např. v případě nukleových kyselin, u nichž je hlavním energetickým příspěvkem ke stabilitě helixu tzv. patrová interakce mezi dvěma sousedícími bázemi, je právě tato interakce zahrnuta v dimerovém modelu. Trimery pak zase odpovídají symetrickému sekvenčnímu kontextu centrálního rezidua a potažmo též odráží princip kódování jednotlivých aminokyselin pomocí tripletových kodonů. V případě nukleotidů se tak získá abeceda o 16 ($= 4^2$) či 64 ($= 4^3$) písmenech. Pokud jsou sekvence kódovány pomocí dimerů či trimerů (obecně multimerů), je možno uvažovat i jejich překryv. Výhodou kódování sekvence nukleotidů jako překrývajících se tripletů je skutečnost, že skryté neurony obdrží přímo informaci o sekvenčním kontextu daného nukleotidu, informaci, která by jinak musela být vnitřní vrstvou extrahována během učení.

Jinou strategií, která může (ale nemusí) vést ke snížení nelinearity predikčního problému, je vytvořit novou, kondenzovanou abecedu seskupením monomerů např. podle jejich fyzikálně-chemických charakteristik. Redukovanou abecedu je potom možno kódovat za použití ortogonální vektorové reprezentace. Příkladem seskupování za vzniku redukované abecedy je např. kódování nukleotidů jako purinů a pyrimidinů (R = A, G, Y = C, T). Pro proteiny lze např. použít 3-písmennou abecedu kódující náboj aminokyseliny (aminokyseliny kyselé, bazické a neutrální) nebo 3-písmennou strukturní abecedu kódující nejčastější umístění rezidua v rámci bílkoviny (rezidua externí, interní a ambivalentní). V jednom konkrétním případě bylo dokonce ukázáno [15] že protein zachovává svoji třírozměrnou strukturu i po drastickém snížení počtu reziduí z 20 na 5. To znamená, že proteiny v raných fázích evoluce byly zřejmě schopny zaujmout stabilní zfoldovanou strukturu s nižším repertoárem aminokyselin, než je k dispozici dneska.

Metody přímého kódování zachovávají pořadí reziduí v sekvenci a kódují primárně lokální informaci. Nejsou nicméně vhodné v případech, kdy je důležité uvažovat globální sekvenční rysy nebo kdy je třeba analyzovat sekvence o proměnných délkách. Takové úlohy např. zahrnují rozlišení sekvence pre-mRNA na introny a exony nebo klasifikaci proteinů. Běžně používaným *nepřímým kódováním* je *n-gram hashing*, kdy je nejprve z původní abecedy vytvořena redukovaná abeceda s počtem písmen M (např. pro $M = 6$ se nová abeceda sestává ze znaků A, B, C, D, E a F), z ní jsou pak vytvořeny n -gramy (tj. pro $n = 2$ se jedná o dvojice sousedních znaků v sekvenci tak, že okénko se posouvá bez překryvu, tj. výstupem jsou např. n -gramy AA, AF, DE, ...) a poté je spočtena frekvence (tj. počet) jednotlivých n -gramů. Vstupní vektor je pak dlouhý jako je počet n -gramů (M^n , tj. $6^2 = 36$ pro náš konkrétní příklad) a hodnoty jeho prvků odpovídají frekvenci příslušných n -gramů (celý proces viz Obr. 13). Tato metoda je invariantní vůči insercím a delecím a je možno ji bez problémů použít pro sekvence proměnlivých délek.

Mnoho strukturních nebo funkčních aspektů biomakromolekul nebývá konzervováno v sekvenci. Je dobře známo, že proteinové struktury mohou být vysoce konzervovány dokonce i tehdy, je-li jejich sekvenční podobnost velmi nízká. Co vytváří strukturu proteinů, ať již na úrovni lokální či globální, nejsou nezávislé příspěvky od individuálních pozic v sekvenci, ale spíše *kooperativita* sekvence. Tento fakt není platný pouze pro proteiny jako celé molekuly, ale i na lokální úrovni, např. pro fosforylační strukturní motiv, který musí být rozpoznán konkrétním enzymem z třídy kináz. Lokální struktury těchto sekvenčních segmentů jsou skutečně značně konzervovány navzdory jejich velké sekvenční diversitě. Neuronové sítě mají značný potenciál postihnout tuto kooperativitu díky jejich schopnosti korelovat navzájem různé vstupní hodnoty. Kooperativita vah mezi jednotlivými neurony zrcadlí relevantní korelace mezi monomery na vstupu, které jsou naopak korelovány s predikční úlohou sítí vykonávanou. Je to právě tato schopnost, která činí neuronové sítě tak úspěšné v oblasti sekvenčních analýz.



Obr. 13. n-gram hashing metoda kódování sekvence. Původní abeceda 20 písmen (viz rámeček, písmena ve složených závorkách) byla redukována na písmen 6. Z redukované sekvence byly vytvořeny 2-gramy a poté zkonstruován vstupní vektor na základě četností všech možných 2-gramů.

4.2 Výběr a extrakce příznaků

Nadměrné množství vstupních neuronů nevede pouze ke snížené schopnosti generalizace, ale též ke zpomalení trénování. Redukce množství vstupních proměnných často významně přispěje ke zlepšení výkonu neuronové sítě (zrychlení učení, snížení nebezpečí přetrénování), a to dokonce i tehdy, zůstane-li určitá část vstupní informace nepoužita. Kromě toho vede výběr vhodné podmnožiny příznaků k lepšímu pochopení procesu, který generoval vstupní data. Jak již bylo uvedeno dříve, vybíráme-li vhodnou kombinaci vstupů, provádíme tzv. výběr příznaků (angl. feature selection), pokud však transformujeme vstupní proměnné z jednoho prostoru do prostoru jiného (o nižší dimenzi, než měl původní prostor), provádíme tzv. extrakci příznaků (angl. feature extraction) [16].

Výhody technik redukce vstupních příznaků jsou však na druhé straně vyváženy faktem, že vnáší do problému další úroveň complexity. Místo pouhé optimalizace parametrů modelu pro kompletní množinu příznaků je třeba nalézt optimální parametry

sítě pro optimální podmnožinu příznaků. V principu totiž není garantováno, že optimální parametry modelu pro úplnou množinu příznaků jsou také optimální pro jejich podmnožinu. Metody výběru příznaků se odlišují právě tím, jakým způsobem prohledávají úplný prostor příznaků. Tyto metody se dělí do tří kategorií⁹: *filter*, *wrapper* a *embedded* metody.

Filter techniky určují relevantnost příznaků pouze na základě dat a pracují nezávisle na procesu výběru vhodného modelu. Ve většině filter technik se vypočítá skóre pro jednotlivé příznaky, ty s nejnižším skóre jsou vyřazeny a zbývající jsou použity jako vstup do neuronové sítě. Filter metody se dělí na jednorozměrné (angl. univariate) a vícerozměrné (angl. multivariate). Jednorozměrné metody při výběru ignorují závislosti mezi příznaky a výběr je řízen sekvenčně, tzn. je testována důležitost jednoho příznaku po druhém bez ohledu na jejich vzájemné interakce. Sem patří např. metody založené na Euklidovské vzdálenosti či na míře zvané *informační zisk* (angl. information gain). Problém zanedbání interakcí mezi příznaky je možno řešit použitím vícerozměrných technik. Ty se od sebe liší stupněm zahrnutí závislostí mezi příznaky a obecně bývají pomalejší než metody jednorozměrné. Výhodou filter přístupů je, že dobře škálují se zvyšující se dimenzionalitou vstupního prostoru, jsou výpočetně jednoduché a rychlé a nejsou závislé na použitém klasifikačním modelu. Výběr příznaků je možno provést pouze jednou a poté je možno na takto vybrané podmnožině vyhodnocovat různé typy klasifikátorů. Nevýhodou naopak je, že ignorují interakci s klasifikačním modelem.

Wrapper metody zohledňují při hledání vhodné podmnožiny příznaků též proces optimalizace vnitřních parametrů modelu, jenž je možno chápat jako prohledávání prostoru hypotéz modelem generovaných. Ve wrapper přístupu je definován způsob, kterým se prohledává prostor příznaků za vzniku různých podmnožin příznaků, jejichž kvalita je poté vyhodnocena. Pro každou množinu příznaků je naučen a testován jiný model (např. neuronová síť), takže tento přístup je přizpůsoben konkrétnímu klasifikačnímu modelu. Prohledávací metody je možno rozdělit na dvě skupiny: deterministické a heuristické. Deterministické postupy, v nichž se postupuje systematicky, jsou méně výpočetně náročné než náhodné přístupy, na druhou stranu u nich však spíše hrozí, že vygenerovaná množina příznaků je množinou toliko suboptimální (tj. prohledávací proces skončil v lokálním minimu). Příkladem takové metody je např. sekvenční dopředný výběr (angl. *sequential forward selection*), kdy se začne s prázdnou množinou příznaků a v každém kroku je do ní přidán takový příznak, který vede k největšímu zlepšení kvality klasifikace. Jiným přístupem v kontextu neuronových sítí je pak omezování sítě (angl. *network pruning*¹⁰), kdy je síť naučena za použití celé množiny příznaků a poté jsou vstupní neurony odstraňovány na základě analýzy jejich vah. Mezi heuristické postupy pak patří např. použití simulovaného žíhání či genetických algoritmů, které jsou ovšem výpočetně mnohem náročnější a snáze mohou vést k přetrénování sítě. Obecně výhodou wrapper technik je zahrnutí

⁹ Ponechávám jejich anglická označení, v jejich české terminologii neexistuje všeobecně přijímaný úzus a české překlady by zněly příliš krkolomně.

¹⁰ Anglický termín *pruning* je možno přeložit též jako prořezávání či vyvětlování, obzvlášť „vvyvětlování sítě“ mi přijde velmi výstižné.

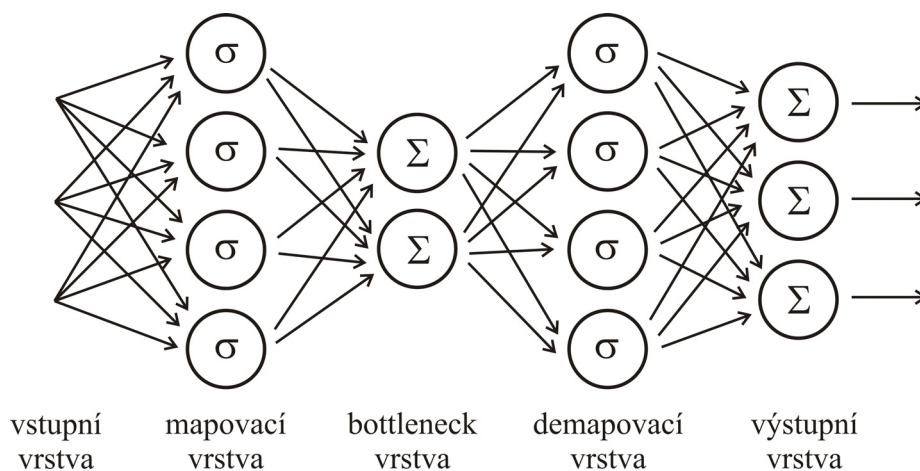
interakce s modelem do procesu výběru a též podchycení závislostí mezi jednotlivými příznaky. Ve srovnání s filter technikami však mohou snáze vést k přeučení sítě a jsou výpočetně velmi náročné.

Embedded metody, na rozdíl od filter a wrapper technik, neseparují proces učení od výběru příznaků. Hledání optimální množiny příznaků se v nich stává integrální součástí trénovacího procesu. Míra důležitosti příznaku se určuje přímo v kontextu konkrétního typu klasifikátoru a jednotlivé metody nelze tedy používat pro klasifikátory jiné, než pro které byly vyvinuty. Obdobně jako wrapper přístupy i embedded metody zahrnují interakci s učícím algoritmem, ale na rozdíl od wrapper metod bývají mnohem méně výpočetně náročné.

Jinou možností redukce dimenzionality dat je extrakce příznaků, kdy lze použít kompletní množinu příznaků a vytvořit jejich kombinace za vzniku nové množiny příznaků. Vtip spočívá v tom, že takto vytvořená nová sada příznaků má často menší dimenzi (bez signifikantní ztráty informace) než sada původní. Takto se postupuje např. ve statistické metodě zvané *metoda hlavní komponenty* (angl. principal component analysis, PCA). PCA je velmi efektivní metodou redukce dimenzionality vstupních dat a navíc zajišťuje, že nově vzniklé příznaky jsou nekorelované, neboť jsou vytvářeny jako ortogonální lineární kombinace příznaků původních. PCA je založena na pevných teoretických základech a je též možno ji využít jako efektivní techniku vizualizace dat. Nicméně pro její správné použití je třeba, aby data splňovala jisté požadavky (např. musí být normálně rozdělena) a nedává uspokojivé výsledky pro malá množství dat či pro data s vyšší mírou šumu. Některé z těchto problémů řeší nelineární varianty PCA (označované souhrnně jako NLPCA), jako je např. *kernel PCA*. Nicméně použití metod NLPCA není již zdaleka tak přímočaré a např. v případě kernel PCA je potřeba, kromě volby vhodné nelineární funkce (tzv. kernelu), také odhadnout jeho optimální parametry.

Jinou možností provedení nelineární redukce dimenzionality je použít tzv. *autoasociativní neuronové síť*. Jedná se o vícevrstvé síť s dopředným šířením, které mají stejný počet vstupních i výstupních neuronů a ve vnitřních vrstvách mají méně neuronů, než je vstupů. Cílem je natrénovat síť tak, že pro daný vstupní vektor je výstupem ten samý vektor. V případě dvouvrstvé sítě pracujeme ve vnitřní vrstvě s redukovaným množstvím neuronů a perfektní rekonstrukce vstupů tudíž není obecně možná. Dá se ukázat, že pokud se ve skryté vrstvě použijí lineární přenosové funkce, pak je takováto síť ekvivalentní lineární analýze hlavních komponent. Ovšem ani nelineární aktivační funkce ve skryté vrstvě nevnese do systému nelinearitu a taková síť je stále ekvivalentní PCA [17]. Použití dvojevrstvé neuronové sítě pro redukci dimenzionality tudíž nepřináší žádné výhody oproti standardním technikám PCA, které jsou naopak výhodnější, neboť garantují nalezení výsledku v konečném čase a poskytují navíc i hlavní komponenty setříděné.

Situace je nicméně odlišná, pokud se použijí další skryté vrstvy. Typická autoasociativní neuronová síť používaná na nelineární redukci dimenzionality má tři skryté vrstvy (viz Obr. 14), kde výstupní neurony jsou lineární a M neuronů v bottleneck vrstvě může být také lineárních. Neurony v mapovací a demapovací vrstvě jsou však nelineární.



Obr. 14. Architektura autoasociativní neuronové sítě.

Tento typ sítě efektivně provádí NLPCA a pokrývá standardní lineární PCA jako speciální případ. Nicméně minimalizace chybové funkce je nyní nelineární optimalizační problém a je třeba použít výpočetně náročných nelineárních optimalizačních technik. Navíc v praxi je třeba, pro daný problém, naučit několik takových sítí, neboť je třeba nalézt optimální počet neuronů v bottleneck vrstvě.

4.3 Zhodnocení kvality predikce

Pro zhodnocení kvality predikce bylo vyvinuto mnoho různých měr. Některé predikční postupy jsou optimalizované tak, aby poskytovaly co nejmenší počet falešně pozitivních výsledků, jiné naopak co nejvyšší počet výsledků falešně negativních. Často je též relevantní, vzhledem k povaze studovaného problému, vyhodnocovat přesnost predikce na různých úrovních. Např. při klasifikaci signálových peptidů¹¹ ze sekvence aminokyselinových reziduí je možno počítat, kolik sekvencí je správně klasifikováno jako signálové peptidy, což je lepší měrou než zjištění, kolik reziduí je správně přiřazeno do třídy signálových peptidů. Na vyšších úrovních však začínají být příslušné míry povětšinou dosti komplikované a specifické pro daný problém. Např. při detekci genu se konce predikovaného exonu mohou překrývat se známou cílovou sekvencí buď přesně, nebo pouze do určité míry. Na vyhodnocení kvality predikce umístění exonu se používají čtyři jednoduché veličiny [18]: senzitivita, specificita (obojí viz dále), chybějící exony a špatné exony. Takový přístup by však byl pro účely zhodnocení predikce sekundární struktury proteinů příliš hrubý, neboť zde nejsou, na rozdíl od exonů, hranice jednotlivých strukturních elementů (α -šroubovice a β -listy) jasně definovány. Místo toho se v tomto případě používá míra zvaná *překryv segmentu* (angl. segment overlap, SOV) [19]. SOV udává procenta korektně přiřazených nikoliv jednotlivých reziduí, ale celých segmentů sekundární struktury. Navíc nepohlíží tak

¹¹ Signálové peptidy řídí vstup prakticky všech proteinů do sekrečních (vylučovacích) drah organismu.

přísně na chyby na koncích sekundárních strukturních elementů, které jinak představují běžný problém mnoha predikčních algoritmů.

Kvůli co nejvyšší obecnosti budou popsány hodnotící míry pro úroveň jednotlivých reziduí a tyto budou diskutovány v kontextu modelového příkladu predikce sekundární struktury proteinů. Pro naše účely si označíme délku aminokyselinové sekvence N a binární strukturní data (tj. správné přiřazení N jednotlivých reziduí k patřičným sekundárním prvkům v sekvenci), dostupná pro účely porovnání s predikovaným výstupem, jako $D = d_1, \dots, d_N$. Pro jednoduchost budeme uvažovat dichotomní problém dvou alternativních tříd, např. α -šroubovice a ne- α -šroubovice. Předpokládejme, že výsledek našeho predikčního modelu je ve tvaru $M = m_1, \dots, m_N$, kde m_i jsou pravděpodobnosti reflektující naši důvěru v konkrétní přiřazení. Pravděpodobnostní hodnoty m_i je pak možno, za použití patřičné hranice, převést na binární reprezentaci 0 či 1. Základní otázkou pak je, jak vyhodnotit přesnost M a jak porovnat M s D . Je třeba si uvědomit, že otázka přesnosti predikce má silnou souvislost s počtem prvků v každé třídě. V případě predikce sekundární struktury proteinů pokrývá nešroubovicová třída cca 70% sekvencí existujících bílkovin. Pokud bychom tedy predikovali, že dané reziduum nepatří k α -šroubovici, získali bychom sice správnou, leč zcela neužitečnou, odpověď v 70% případech.

Základním zjednodušujícím předpokladem pro zhodnocení přesnosti predikce je, že pozice aminokyselin jsou stejně pravděpodobné a nezávislé (předpoklad nezávislosti a ekvivalence). To znamená, že se předpokládá, že neexistuje žádný mechanismus redukující vliv pozic reziduí blízko konců řetězce, nebo není brána do úvahy skutečnost, že predikce by se měla měnit „hladce“ (např. patří-li reziduum do třídy α -šroubovic, měl by jeho soused ve skutečnosti mít o něco vyšší pravděpodobnost, že též bude patřit do α -šroubovice).

Za předpokladu nezávislosti a ekvivalence, jsou-li D i M binární, mohou být kompletně porovnány za použití čtyř čísel:

- skutečná pozitiva (angl. *true positives*, TP) – kolikrát je d_i v helixu, patří-li příslušné m_i do helixu
- skutečná negativa (angl. *true negatives*, TN) – kolikrát není d_i v helixu, nepatří-li příslušné m_i do helixu
- falešná pozitiva (angl. *false positives*, FP) – kolikrát není d_i v helixu, patří-li příslušné m_i do helixu
- falešná negativa (angl. *false negatives*, FN) – kolikrát je d_i v helixu, nepatří-li příslušné m_i do helixu

Platí, že $TP + TN + FP + FN = N$. Hodnoty TP, TN, FP a FN se běžně uspořádávají do 2×2 tabulky zvané tabulka záměn (angl. *table of confusion*):

| | | |
|----------|-----|----------|
| | M | not- M |
| D | TP | FN |
| not- D | FP | TN |

Nicméně za použití těchto čtyř čísel není posouzení kvality dané metody jednoduché. Z tohoto důvodu se používají různé míry, popsané v následující sekci, sumarizující informace obsažené v D a M . Je však třeba si uvědomit, že při použití takových veličin dojde vždy ke ztrátě jistého množství informace.

4.3.1 Procenta

Nejjednodušším přístupem je použít procentuelní hodnoty odvozené z TP , TN , FP a FN . Například je možno použít procento správně predikovaných šroubovic:

$$PCP(D, M) = 100 \frac{TP}{TP + FN}$$

což je vlastně procentuelně vyjádřená senzitivita (viz dále). Nicméně toto číslo neposkytuje žádnou informaci o falešně pozitivních predikcích a bývá proto doplněno procentem správně predikovaných ne-šroubovic:

$$PCN(D, M) = 100 \frac{TN}{TN + FP}$$

Průměr těchto hodnot $Q_\alpha = \frac{PCP + PCN}{2}$ je běžně používaným indikátorem, ačkoliv může být klamavý v případě, kdy D i M jsou binární vektory.

4.3.2 Hammingova vzdálenost

V binárním případě je Hammingova vzdálenost mezi D a M definována jako:

$$HD(D, M) = \sum_i |d_i - m_i|$$

Tato suma se rovná celkovému počtu chyb $FP + FN$ a je tudíž ekvivalentní jednoduché procentuální míře. Tato vzdálenost neuvažuje podíl případů patřících do jednotlivých tříd. V nebinárním případě se Hammingova vzdálenost označuje jako L^1 vzdálenost.

4.3.3 Kvadratická „vzdálenost“

Kvadratická (Euklidovská) „vzdálenost“ je definována jako:

$$Q(D, M) = (D - M)^2 = \sum_i (d_i - m_i)^2$$

Přísně vzato, korektní vzdálenost je definována jako druhá odmocnina výše uvedené veličiny (viz L^2 vzdálenost v následující sekci). V binárním případě se kvadratická vzdálenost redukuje na Hammingovu vzdálenost. V případě, kdy je trénovací proces akcelerován přičtením chyb ke vstupním datům, se jako chybová funkce používá spíše logaritmická variace Q :

$$LQ(D, M) = -\sum_i \log[1 - (d_i - m_i)^2]$$

4.3.4 L^p vzdálenosti

L^p vzdálenosti jsou obecně definovány jako:

$$LP(D, M) = \left[\sum_i (d_i - m_i)^p \right]^{\frac{1}{p}}$$

V případě $p = 1$ se jedná o Hammingovu vzdálenost, v případě $p = 2$ o pravou Euklidovskou vzdálenost. Pro $p \rightarrow \infty$ je L^∞ vzdálenost dána jako $\max_i |d_i - m_i|$. Jiné hodnoty p jsou v praxi zřídka využívány. V binárním případě se L^p vzdálenost redukuje na $(FP + FN)^{1/p}$.

4.3.5 Korelace

Jedna z běžných statistických měr je korelační koeficient, někdy též označovaný jako Pearsonův korelační koeficient. V kontextu predikce sekundární struktury proteinů je tato míra v literatuře známa též jako Matthewsův korelační koeficient. Korelační koeficient leží vždy mezi -1 a $+1$ a je možno ho použít jak s binárními, tak

s nebinárními proměnnými. Korelační koeficient je měrou kvantifikující stupeň lineární závislosti mezi proměnnými (tj. jak se mění znaménko a velikost jedné proměnné v závislosti na změně druhé proměnné). V případě ideálního lineárního vztahu je roven +1 v případě ideálního “antilineárního” vztahu, kdy s růstem jedné proměnné druhá klesá, je roven -1 a v ostatních, neideálních, případech nabývá hodnot někde mezi. Pro zcela náhodný vztah je pak roven nule. Pro vektory je korelační koeficient možno psát jako skalární součin mezi normalizovanými hodnotami:

$$C(D, M) = \frac{(D - \bar{d}\vec{1})(M - \bar{m}\vec{1})}{\sqrt{(D - \bar{d}\vec{1})^2} \sqrt{(M - \bar{m}\vec{1})^2}}$$

kde $\vec{1}$ je N -rozměrný vektor sestávající výhradně z jedniček a $\bar{d} = \sum d_i / N$ a $\bar{m} = \sum m_i / N$. V případě, kdy D a M sestávají z binárních hodnot platí, že $D^2 = TP + FN$, $M^2 = TP + FP$. Po menších úpravách je možno korelační koeficient psát jako:

$$C(D, M) = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FN)(TP + FP)(TN + FP)(TN + FN)}}$$

Korelační koeficient zahrnuje všechna 4 čísla TP , TN , FP a FN a představuje mnohem lépe vyváženou míru kvality predikce než např. procenta. Další jeho výhodou je, že existuje jednoduchý statistický test umožňující určit, zda je korelační koeficient signifikantně odlišný od nuly či nikoliv. Přesto existují situace, kdy korelační koeficient není schopen poskytnout správné ohodnocení. Bude např. relativně vysoký v případech, kdy predikční algoritmus vrátí velmi málo falešných pozitiv a současně i velmi málo skutečných pozitiv.

4.3.6 Relativní entropie

Relativní entropie (angl. relative entropy, cross entropy, Kullback-Leibler (KL) contrast) mezi dvěma pravděpodobnostními vektory $\vec{x} = (x_1, \dots, x_M)$ a $\vec{y} = (y_1, \dots, y_M)$, kde $x_i, y_i \geq 0$ a $\sum x_i = \sum y_i = 1$ je definována jako:

$$H(\vec{x}, \vec{y}) = \sum_{i=1}^M \log \frac{x_i}{y_i} = -H(\vec{x}) - \sum_i x_i \log y_i$$

kde $H(\vec{x}) = -\sum x_i \log x_i$ je entropie – míra mající své základy v teorii informace [20].

$H(\vec{x}, \vec{y})$ je vždy kladná a rovna nule pouze tehdy, je-li $\vec{x} = \vec{y}$. Pro případ predikce sekundární struktury je možno ji vypočítat jako:

$$H(D, M) = \sum_{i=1}^N \left[d_i \log \frac{d_i}{m_i} + (1 - d_i) \log \frac{1 - d_i}{1 - m_i} \right]$$

což je vlastně pouze suma relativních entropií na každé z i pozic. Tuto formuli je ale možno použít pouze pro nebinární data nebo v případě, kdy pouze D je binární. Je-li M také binární, má relativní entropie $FP + FN$ nekonečných komponent a není možno ji vyčíslit.

4.3.7 Vzájemná informace (angl. mutual information)

Mutual information je další veličina, která udává, kolik nám toho jedna náhodná proměnná říká o proměnné druhé. Je to bezrozměrná kladná veličina, která se v případě použití logaritmu o základu 2 udává v bitech. Reprezentuje snížení množství nejistoty v jedné proměnné poté, co byla pozorována druhá proměnná. Velká hodnota mutual information indikuje velké snížení, malá snížení malé a nulová hodnota znamená, že proměnné jsou nezávislé.

Pro modelový příklad predikce sekundární struktury proteinů, kdy D a M jsou binární, je mutual information dána jako:

$$I(D, M) = -H\left(\frac{TP}{N}, \frac{TN}{N}, \frac{FP}{N}, \frac{FN}{N}\right) - \frac{TP}{N} \log \left[\frac{(TP + FP)(TP + FN)}{N} \right] - \\ - \frac{FN}{N} \log \left[\frac{(TP + FN)(TN + FN)}{N} \right] - \frac{FP}{N} \log \left[\frac{(TP + FP)(TN + FP)}{N} \right] - \\ - \frac{TN}{N} \log \left[\frac{(TN + FN)(TN + FP)}{N} \right]$$

kde:

$$H\left(\frac{TP}{N}, \frac{TN}{N}, \frac{FP}{N}, \frac{FN}{N}\right) = -\frac{TP}{N} \log \frac{TP}{N} - \frac{TN}{N} \log \frac{TN}{N} - \frac{FP}{N} \log \frac{FP}{N} - \frac{FN}{N} \log \frac{FN}{N}$$

je nám již známá entropie. Pro mutual information vždy platí $0 \leq I(D, M) \leq H(D)$, kde:

$$H(D) = -\frac{TP + FN}{N} \log \frac{TP + FN}{N} - \frac{TN + FP}{N} \log \frac{TN + FP}{N}$$

Často se také používá normalizovaný koeficient:

$$IC(D, M) = \frac{I(D, M)}{H(D)}$$

splňující $0 \leq IC(D, M) \leq 1$.

4.3.8 Senzitivita a specifická

Senzitivita a specifická jsou dvě statistické míry kvality binární klasifikace. Senzitivita (angl. sensitivity, v některých aplikačních oblastech též recall) je pravděpodobnost, že systém správně identifikoval pozitivní případ:

$$\text{sensitivity} = \frac{TP}{TP + FN}$$

Specifická (angl. specificity, v některých aplikačních oblastech též precision) je pravděpodobnost, že je pozitivní predikce správná:

$$\text{specificity} = \frac{TN}{TN + FP}$$

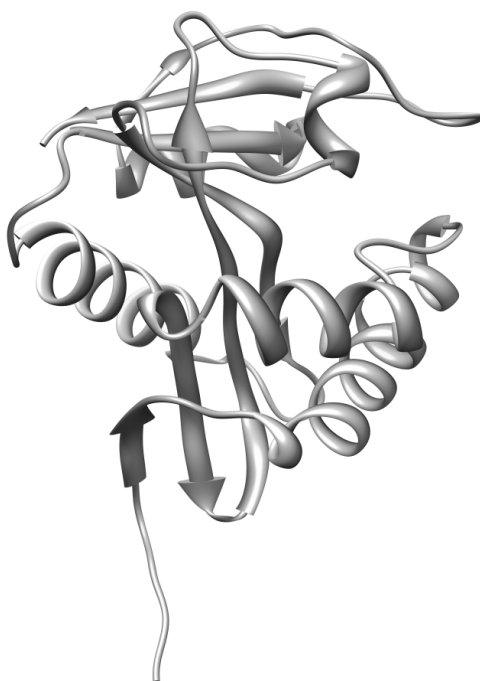
Pro případ klasifikace do dvou tříd se spojitým výstupem závisí čísla TP , TN , FP a FN na nastavení prahu pro výstup. Na základě něho se rozhoduje, zda vstupní vektor patří do jedné či do druhé třídy. Obecně platí, že mezi počtem falešně pozitivních a falešně negativních existuje rovnováha, snižováním počtu FP se zvyšuje počet FN a naopak.

Správnost predikce binárního klasifikátoru v závislosti na nastavení prahu na výstupu se sumarizuje za použití tzv. křivky „receiver operating characteristics“ (ROC), kdy se pro jistý rozsah prahu zobrazí závislost mezi senzitivitou a specifickou. Z ní je potom možno vygenerovat souhrnnou statistiku udávající kvalitu predikce, např. obsah pod ROC křivkou (angl. area under the ROC curve, AUC), a tu použít pro optimalizaci klasifikačního procesu nebo pro porovnání různých klasifikačních modelů mezi sebou.

4.4 Neuronové sítě pro predikci sekundární struktury proteinů

Z grafické vizualizace proteinu (viz Obr. 15) jsou okamžitě patrné lokální pravidelnosti v podobě opakujících se sekundárních struktur, mezi něž patří, jak již bylo uvedeno dříve, α -helixy, β -listy a otáčky (smyčky, angl. coils).

Zjištění umístění sekundárních struktur v rámci experimentálně získané 3D struktury proteinu patří mezi netriviální problémy. Většinou se provádí za použití programu DSSP [21] pracujícího na základě analýzy opakujících se vzorů potenciálních vodíkových vazeb v 3D struktuře proteinu. Nicméně schopnost přiřadit správně to, co se oku jeví jako šroubovice nebo list, je, vzhledem k limitované přesnosti xyz souřadnic atomů, netriviálním algoritmickým problémem. Dalším faktorem přispívajícím k obtížnosti problému je skutečnost, že správná identifikace vodíkových vazeb je možná pouze za použití postupů kvantové chemie, které však není možno, kvůli jejich výpočetní náročnosti, aplikovat na tak velké systémy, jako jsou proteiny či nukleové kyseliny. Při predikcích by tedy bylo užitečné soustředit se nejen na vizuální nebo topologické aspekty problému, ale navrhnout více obecné přiřazovací schéma. A přesně to je cílem metod predikce sekundární struktury ze znalosti pouhé proteinové sekvence, mezi nimiž zaujímají prominentní místo postupy založené na neuronových sítích.



Obr. 15. Vizualizace proteinu se znázorněnými prvky sekundární struktury. Na obrázku jsou patrné helikální oblasti a β -listy (jednotlivé β -řetězce jsou znázorněny pomocí plochých šipek), které jsou mezi sebou pospojovány otáčkami.

V klíčové práci, poprvé demonstrující aplikaci neuronových sítí na problém predikce sekundární struktury [14], byla použita vícevrstvá síť s jednou skrytou vrstvou. Vstupní okénko mělo lichou délku W s optimální hodnotou $W = 13$. Jako vstupu bylo použito ortogonální kódování s abecedou o 21 prvcích odpovídajících 20 aminokyselinám a jednomu terminálnímu symbolu použitému pro kódování částečných sekvencí na N- a C- koncích. Vstupní vrstva tedy obsahovala $13 \times 21 = 273$ neuronů. Typická velikost skryté vrstvy byla 40 neuronů a na výstupní vrstvě byly použity 3 sigmoidní neurony reprezentující klasifikaci do jedné ze tří tříd: α -helix (třída označovaná jako H), β -list (třída označovaná jako E, z anglického extended) či otáčku (třída označovaná jako C)¹². Celkové množství nastavitelných parametrů takové architektury je tedy $273 \times 40 + 40 \times 3 + 40 + 3 = 11\,083$. Klasifikovaná třída je určena výstupním neuronem s nejvyšší aktivitou (strategie známá jako „vítěz bere vše“, angl. winner-takes-all) a představuje klasifikaci rezidua umístěného ve středu vstupního okénka. Princip „vítěz bere vše“ zajišťuje, že interní reprezentace problému skrytou vrstvou nemusí být ideálně lineárně separovatelná. Síť byla inicializována náhodnými uniformními vahami z intervalu $\langle -0.3, +0.3 \rangle$ a naučena algoritmem zpětného šíření chyby. Trénovací množina obsahovala cca 20 000 reziduí, jejichž 3D souřadnice byly získány z volně dostupné strukturní databáze PDB [22] a HEC třídy jim byly přiřazeny

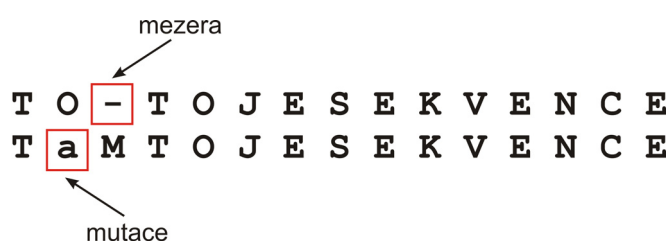
¹² Taková klasifikace se běžně označuje jako HEC klasifikace.

programem DSSP. Testovací množina měla velikost cca 1400 reziduí. Při trénování byly sekvence prezentovány na vstup v náhodném pořadí. S touto architekturou bylo dosaženo procentního podílu správně predikovaných reziduí (pro testovací množinu) Q_3 rovného 62.7%:

$$Q_3 = \frac{P_H + P_E + P_C}{N}$$

kde N je celkový počet predikovaných reziduí a P_a je počet správně predikovaných sekundárních struktur typu a . Q_3 však není nejlepší měrou kvality predikce, neboť H, E a C třídy nejsou v proteinech stejnoměrně rozděleny, ale jsou zastoupeny zhruba v poměru 30%:20%:50% (H:E:C). Mnohem vhodnější měrou je v takovém případě dříve popsáný Matthewsův korelační koeficient, který se počítá pro každou ze tří tříd a který se rovnal $C_H = 0.35$, $C_E = 0.29$ a $C_C = 0.38$. Autoři provedli řadu experimentů zahrnujících změny v architektuře neuronové sítě, zahrnutí fyzikálně-chemických charakteristik aminokyselinových reziduí do vstupních vzorů nebo zvýšení počtu klasifikovaných tříd. Hlavního vylepšení bylo nakonec dosaženo napojením předchozí architektury na druhou neuronovou síť, která konstruuje své vstupy ze spojených výstupů z první sítě. Velikost okénka druhé sítě $W = 13$ odpovídá 13 po sobě jdoucím výstupům z první sítě a vstupní vrstva tedy obsahuje 13×3 neuronů. Počet vnitřních neuronů druhé sítě byl též roven 40 a výstupní vrstva sestávala ze tří sigmoidních neuronů. S touto architekturou bylo dosaženo následující kvality predikce: $Q_3 = 64.3\%$, $C_H = 0.41$, $C_E = 0.31$ a $C_C = 0.41$. Tato a jiné obdobné studie dospěly k empirickému závěru, že pro přesnost predikce za použití lokální metody (tj. metody konstruující vstupní data pouze jako sekvenční okénko o délce W) existuje teoretický horní limit cca 70%.

Ačkoliv je obtížné porovnat výkon predikčních systémů, které byly testovány na různých datových množinách, nejvýznamnějšího vylepšení výkonu ve srovnání s předchozími metodami bylo dosaženo B. Rostem a C. Sanderem [23-25], kteří vytvořili predikční systém PHD [26]. Hlavním rysem PHD přístupu je zahrnutí evoluční informace do vstupní datové množiny. V této metodě byly použity jako vstupní vektory tzv. profily, které byly získány jako četnosti výskytu jednotlivých aminokyselin v sekvenční množině získané jako násobné přiřazení (angl. multiple alignment). Násobné přiřazení je způsob porovnávání několika sekvencí a zjišťování míry jejich vzájemné podobnosti. Proteiny s vysokou podobností jsou považovány za homologní a dá se o nich předpokládat, že se vyvinuly ze stejného počátečního předka. Při přiřazení k sobě přikládáme sekvence tak, že se snažíme maximalizovat jejich vzájemnou shodu. Toho se dosáhne tím, že do sekvencí je dovoleno vkládat mezery (angl. gaps) nebo je umožněna záměna jednoho rezidua za reziduum jiné (tzv. mutace). Celý proces znázorňuje Obr. 16.



Obr. 16. Modelový postup přiřazení dvou sekvencí. Sekvence se srovnávají tak, že co nejvíce stejných písmen leží pod sebou. K tomu je možno si dopomoci vkládáním mezer nebo povolením záměn jednotlivých písmen (mutací, bývají zvýrazňovány použitím malých písmen). Přiřazení dvou sekvencí se označuje jako párové přiřazení (angl. pairwise alignment), přiřazení více sekvencí pak jako násobné přiřazení (angl. multiple alignment).

Způsob konstrukce profilu z násobného přiřazení pak, ve zjednodušené podobě, demonstruje Obr. 17.

Z popisu PHD metody je patrné, že kvalita profilu záleží na algoritmu, který je použit pro násobné přiřazení. Metoda PSI-BLAST [28] je iterativní postup při kterém se v sekvenční databázi nejprve vyhledají sekvence podobné počáteční sekvenci a poté jsou tyto sekvence použity jako nové dotazy. Tímto způsobem je možno získat bohatší skupinu sekvencí, ovšem za cenu zahrnutí nesouvisejících sekvencí. Tohoto iterativního způsobu generování sekvencí bylo použito při získání profilu v metodě PSIPRED [29]. Nahrazení HSSP profilů používaných v PHD metodě takto vygenerovanými daty vedlo ke zvýšení predikční přesnosti sítě na $Q_3 = 76.5\%$.

Ačkoliv helixy a listy preferují určité délky, distribuce délek obou dvou typů struktur je dosti široká. Použije-li se na predikci pouze jedna neuronová síť, je nutno zvolit kompromisní velikost okénka. Větší okénko totiž může lépe postihnout dodatečné signály v dlouhých sekundárních strukturách, zatímco menší okénko naopak lépe popíše krátké struktury. V rámci jediné neuronové sítě velká či malá okénka poskytují horší celkovou predikci, ale v individuálních případech mohou být naopak více spolehlivá. Běžným přístupem je v takových případech natrénovat několik neuronových sítí s různou velikostí okénka (včetně suboptimálních příliš malých či příliš velkých okének) a jejich výsledky poté zkombinovat do jediné predikce, například jednoduchým zprůměrováním výstupů ze všech sítí. S rostoucím počtem sítí má však průměrování, díky zahrnutí šumu ze suboptimálních sítí, destruktivní dopad na kvalitu predikce. Tento problém byl řešen v práci T. E. Petersena [30], který ukázal, jak správně zkombinovat výstupy z mnoha neuronových sítí lišících se výrazně v architektuře (různé velikosti vstupního okénka, různé počty skrytých neuronů apod.). Klíčovým prvkem v této práci bylo nalezení a použití pouze takových neuronových sítí z celkového počtu 800, jejichž predikce jsou důležité pro správné přiřazení daných aminokyselinových reziduí v testovací množině. Průměrování přes tuto podmnožinu pak výrazně redukuje šum pocházející od suboptimálních sítí. Kromě správné volby množiny sítí byl výkon také vylepšen použitím tzv. výstupní expanze (angl. output expansion), kdy síť poskytuje predikce sekundární struktury odpovídající nejenom centrálnímu reziduu v okénku, ale simultánně též reziduíům sousedním. Použitím tohoto přístupu byla vylepšena predikce až na $Q_3 = 80.2\%$.

| Protein | Alignments | Profil | | | | | | | | | | | | | | | | | | | |
|---------|------------|--------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | A | R | N | D | C | Q | E | G | H | I | L | K | M | F | P | S | T | W | Y | V |
| I | I g g g | . | . | . | . | . | . | . | 3 | . | 2 | . | . | . | . | . | . | . | . | . | . |
| Y | Y Y Y Y | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | 5 |
| G | G G p p | . | . | . | . | . | . | . | 3 | . | . | . | . | . | . | 2 | . | . | . | . | . |
| P | P P P f | . | . | . | . | . | . | . | . | . | . | . | . | 1 | 4 | . | . | . | . | . | . |
| A | h A A A | 4 | . | . | . | . | . | . | . | 1 | . | . | . | . | . | . | . | . | . | . | . |
| C | C C C C | . | . | . | 5 | . | . | . | 3 | 2 | . | . | . | . | . | . | . | . | . | . | . |
| H | H H i i | . | . | . | . | . | . | . | 3 | 2 | . | . | . | . | . | . | . | . | . | . | . |
| I | I h a I | 1 | . | . | . | . | . | . | 1 | 3 | . | . | . | . | . | . | . | . | . | . | . |
| Y | s Y Y s | . | . | . | . | . | . | . | . | . | . | . | . | . | . | 2 | . | . | . | 3 | . |
| S | w S S S | . | . | . | . | . | . | . | . | . | . | . | . | . | . | 4 | . | 1 | . | . | . |
| D | D D D c | . | . | . | 4 | 1 | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| W | W W W i | . | . | . | . | . | . | . | . | 1 | . | . | . | . | . | . | . | . | 4 | . | . |
| G | G G G G | . | . | . | . | . | . | . | 5 | . | . | . | . | . | . | . | . | . | . | . | . |
| F | s s F F | . | . | . | . | . | . | . | . | . | . | . | . | 3 | . | 2 | . | . | . | . | . |
| Y | Y g g p | . | . | . | . | . | . | . | 2 | . | . | . | . | . | 1 | . | . | . | . | 2 | . |
| V | V V V V | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | 5 |
| N | s N s N | . | . | 3 | . | . | . | . | . | . | . | . | . | . | . | 2 | . | . | . | . | . |

Obr. 17. Způsob vytvoření sekvenčního profilu použitého jako vstup do neuronové sítě v PHD metodě. Nejprve bylo provedeno násobné přiřazení sekvence proteinu se sekvencemi nalézajícími se v databázi HSSP [27] (mutace jsou znázorněny malými písmeny). Poté byl zkonstruován profil, který obsahuje četnosti jednotlivých aminokyselinových reziduí (zde jsou použity jejich jednopísmenné zkratky) v násobném přiřazení (místo teček patří nuly, tečky byly použity pouze pro vizuální zvýraznění nenulových hodnot). Vstup do sítě byl pak zkonstruován tak, že byly použity četnosti všech aminokyselin v násobném přiřazení 13 následných reziduí (čárkovaný obdélník). Počet vstupních neuronů do takové sítě je tedy roven $13 \times 20 = 260$.

Zahrnuté evoluční informace do trénovací a testovací množiny použitím sekvenčních profilů vedlo k vylepšení přesnosti predikce Q_3 na hodnotu 72% s korelačními koeficienty $C_H = 0.64$ a $C_S = 0.53$.

V současné době se kvalita predikce sekundární struktury proteinů pohybuje kolem 80% a práce, které se tímto problémem zabývají, přidávají k této hodnotě pouhé desetiny procentuelních bodů. Možným způsobem vylepšení stávajících algoritmů může být např. zahrnutí informace o klasifikaci sousedního rezidua (reziduum, jež má za souseda helixové reziduum, má také větší pravděpodobnost, že bude patřit do třídy helixů) nebo zahrnutí informace o velikosti volně dostupného povrchu daného rezidua.

5 Závěr

Úspěšné aplikace neuronových sítí na četné problémy z oblasti bioinformatiky a výpočetní molekulární biologie, z nichž některé byly popsány v předchozích odstavcích, povedou v budoucnu bez pochyby k dalšímu rozvoji této vědecké oblasti. To ovšem neznamená, že by neuronové sítě měly být využívány k řešení bioinformatických problémů na úkor použití tradičních statistických postupů. Existuje mnoho způsobů, jak řešit úkoly z oblasti modelování, regrese či klasifikace a hlavním cílem by mělo být vždy použití správných technik vhodných pro daný problém. Ačkoliv mezi neuronovými sítěmi a statistickými postupy existují mnohé rozdíly, je možné mezi nimi nalézt i řadu podobností a analogií. Například lineární regresi je možno provést pomocí jednoduchého perceptronu, lineární diskriminační analýza je ekvivalentem klasifikace pomocí lineárního perceptronu, neuronová síť bez skryté vrstvy a s jediným logistickým výstupním neuronem je ekvivalentní logistické regresi běžně používané ve statistice pro analýzu binárních dat apod. Zřejmě nejpříťažlivější vlastností neuronových sítí je intuitivní vizuální způsob řešení problémů (zde hovořím především o architekturách, jako je např. Kohonenova SOM, kterým nebylo v této kapitole věnováno příliš prostoru) a zmírnění přísných formálních podmínek nutných pro správné použití statistických postupů. Ač jsou vícevrstvé neuronové sítě vhodné pro řešení hojně se v bioinformatice vyskytujících nelineárních problémů, je třeba otevřeně uznat, že nejsou všelékem a že ne vždy by jim měla být dáována přednost před klasickými statistickými postupy. Nejlepším způsobem, jak přistupovat k řešení konkrétního problému, je použít nejprve jednoduché (např. lineární) modely a nasadit sofistikovanější postupy, zahrnující i neuronové sítě, pouze tehdy, pokud je kvalita jednoduchých modelů nedostačující.

Při používání neuronových sítí, které mohou aproximovat prakticky libovolná data, je důležité zaujmout kritický pohled jak na architekturu neuronové sítě, tak na interpretaci jejích výsledků. Tento přístup se týká především volby správného počtu skrytých neuronů, kdy neuronová síť bude data přefitovávat, je-li skrytá vrstva příliš velká, ale na druhou stranu, při nízkém počtu skrytých neuronů, dojde ke snížení jejích generalizačních schopností. Rovněž způsob validace neuronové sítě je velmi důležitý, minimálním požadavkem je použití LOOCV (leave-one-out validace) a, pokud to množství dat dovolí, dostatečná pozornost by měla být věnována správnému rozdělení na trénovací, validační a testovací množinu. Nevýhodou neuronových sítí je také to, že obsahují příliš mnoho nastavitelných parametrů, což vede ke zvýšení požadavků na počet vzorků v trénovací datové množině. Neuronové sítě nemusí poskytovat správné výsledky pro malé množiny dat, kdy je naopak více na místě použití statistických metod. Největším nedostatkem neuronových sítí je ovšem velmi obtížná interpretovatelnost jejich rozhodovacích procesů. Ačkoliv bylo v této oblasti v posledních několika letech dosaženo mnoha pokroků [31], mají statistické postupy (např. regresní metody) v tomto směru jednoznačně navrch. Statistika totiž disponuje velmi bohatým a teoreticky dobře podloženým arzenálem metod statistického usuzování (angl. statistical inference).

Poděkování: Tento příspěvek vznikl za podpory Ministerstva školství, mládeže a tělovýchovy ČR v rámci řešení grantové úlohy číslo MSM6046137302.

Literatura

- [1] Woese, C. R., Kandler O., and Wheelis M. L., *Towards a natural system of organisms: proposal for the domains Archaea, Bacteria, and Eucarya*. Proc Natl Acad Sci U S A, 1990. **87**(12): p. 4576-9.
- [2] Akaike, H., *A New Look at Statistical Model Identification*. IEEE Transactions On Automatic Control, 1974. **19**: p. 716-723.
- [3] Smith, M., *Neural Networks for Statistical Modeling*. 1996, London, UK, Boston, MA: International Thompson Computer Press.
- [4] McCulloch, W. and Pitts W., *A Logical Calculus of Ideas Immanent in Nervous Activity*. Bulletin of Mathematical Biophysics, 1943. **5**: p. 115-133.
- [5] Hebb, D., *The Organization of Behavior*. 1949, Wiley, New York.
- [6] Widrow, B. and Hoff M., *Adaptive Switching Circuits*, in *IRE WESCON Convention Record*. 1960. p. 96-104.
- [7] Minsky, M. L. and Papert S. A., *Perceptrons*. 1969, Cambridge, MA: MIT Press.
- [8] Werbos, P. J., *Beyond regression: new tools for prediction and analysis in the behavioural sciences*. 1974, Harvard University: Boston.
- [9] Rumelhart, D. E., Hinton G. E., and Williams R. J., *Learning internal representation by error propagation.*, in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Rumelhart D. E. and McClelland J. L., Editors. 1988, MIT Press: Cambridge, MA. p. 318-362.
- [10] Jacobs, R. A., *Increased Rates of Convergence through Learning Rate Adaptation*. Neural Networks, 1988. **1**(4): p. 295-307.
- [11] Fahlman, S. *Faster learning variations of backpropagation: an empirical study*. in *Connectionist Models Summer School*. 1988: Morgan Kaufmann Publishers.
- [12] Bishop, C. M., *Neural Networks for Pattern Recognition*. 1996: Oxford University Press.
- [13] Stormo, G. D., et al., *Use of the 'Perceptron' algorithm to distinguish translational initiation sites in E. coli*. Nucleic Acids Res, 1982. **10**(9): p. 2997-3011.
- [14] Qian, N. and Sejnowski T. J., *Predicting the secondary structure of globular proteins using neural network models*. J Mol Biol, 1988. **202**(4): p. 865-84.
- [15] Riddle, D. S., et al., *Functional rapidly folding proteins from simplified amino acid sequences*. Nat Struct Biol, 1997. **4**(10): p. 805-9.
- [16] Saeyns, Y., Inza I., and Larranaga P., *A review of feature selection techniques in bioinformatics*. Bioinformatics, 2007. **23**(19): p. 2507-17.
- [17] Bourlard, H. and Kamp Y., *Auto-association by multilayer perceptrons and singular value decomposition*. Biol Cybern, 1988. **59**(4-5): p. 291-4.

-
- [18] Burset, M. and Guigo R., *Evaluation of gene structure prediction programs*. Genomics, 1996. **34**(3): p. 353-67.
- [19] Zvelebil, M. and Baum J., *Understanding Bioinformatics*. 2007: Garland Science.
- [20] Yeung, R. W., *A First Course in Information Theory*. 2006: Springer.
- [21] Kabsch, W. and Sander C., *Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features*. Biopolymers, 1983. **22**(12): p. 2577-637.
- [22] Berman, H. M., et al., *The Protein Data Bank*. Nucleic Acids Res, 2000. **28**(1): p. 235-42.
- [23] Rost, B. and Sander C., *Improved prediction of protein secondary structure by use of sequence profiles and neural networks*. Proc Natl Acad Sci U S A, 1993. **90**(16): p. 7558-62.
- [24] Rost, B. and Sander C., *Prediction of protein secondary structure at better than 70% accuracy*. J Mol Biol, 1993. **232**(2): p. 584-99.
- [25] Rost, B. and Sander C., *Combining evolutionary information and neural networks to predict protein secondary structure*. Proteins, 1994. **19**(1): p. 55-72.
- [26] Rost, B., Sander C., and Schneider R., *PHD--an automatic mail server for protein secondary structure prediction*. Comput Appl Biosci, 1994. **10**(1): p. 53-60.
- [27] Dodge, C., Schneider R., and Sander C., *The HSSP database of protein structure-sequence alignments and family profiles*. Nucleic Acids Res, 1998. **26**(1): p. 313-5.
- [28] Altschul, S. F., et al., *Gapped BLAST and PSI-BLAST: a new generation of protein database search programs*. Nucleic Acids Res, 1997. **25**(17): p. 3389-402.
- [29] McGuffin, L. J., Bryson K., and Jones D. T., *The PSIPRED protein structure prediction server*. Bioinformatics, 2000. **16**(4): p. 404-5.
- [30] Petersen, T. N., et al., *Prediction of protein secondary structure at 80% accuracy*. Proteins, 2000. **41**(1): p. 17-20.
- [31] Saito, K. and Nakano R., *Extracting regression rules from neural networks*. Neural Netw, 2002. **15**(10): p. 1279-88.

Index

A

abdukcia 231
ABS systém 152
abstrakčný potenciál jazyka 273
AC (arc consistency), hranová konzistencia 297
aditívny farebný model 418
agent 21, 32, 383, 390, 428
agenty sémantického webu 54
Amesův pokoj 484
aminokyselina 528
analytická metóda 267
Anaximandros 97
anestéziológia 160
anfis 182, 188
animace 26
aplikácie fuzzy logiky 148
apódeixis 267
aristotelovský kozmos 271
atomická formula jazyka 138
atómy, výrokové 127
autoasociatívni neuronové sítě 555
automobilový priemysel 152

B

banka filtrů 447
Barabási – Albert model 340
Barcanovej formuly 258
báza pravidiel 158, 167
báza znalostí 159, 167
Bethove sémantické tablá 210
bezškálová sieť 330
bioinformatika 523
biomolekuly 527
bipartitný graf 325
blendování animací 29
Boolova funkcia 215
Buffon 99

buněčná diferenciace 527

C

Caffe Wall klam 499
čapíky 414
cnosť jazyka 265
čtecí rámec 531
Cuvier 103
cytoplasma 528

D

ďalekodosahové korelácie 334
Darwin 91, 97, 99, 102, 439
dědičnost 102, 306, **525**
deformační klece 27
defuzifikácia 167, 185
diferenciačný potenciál jazyka 275
diorismós 267
disjunkcia 128, 211, 236, 247
disjunktívna klauzula 217
disjunktívna normálna forma (DNF) 209, 217
distribúvané ontológie 69
dlhodobá pamäť 12, 44
DOM, Document Object Model 58
doprava v tuneloch 156
dostupnosť svetov 255
drátový model 23
DTD, Document Type Definition 57
dynamický rezervoár 3
dynamika siete 2, 321

E

efekt rámování 483, 484
echo stavu 1, 6
ekthesis 266

ekvivalencia 130, 214, 216, 223
élan vital 108
embedded metody výběru příznaků 555
Empedokles 98
eukaryota 526
Eulerovský ťah 326
evolúcia 89, 523
- „in materio“ 455
- kvantových chování 456
- v hardware 449
evolučná epistemológia 111
evolučná psychológia 114
evolučné algoritmy 117, 306, 439, 458
evoluční návrh
- obrazových operátorů 444
- testovacích obvodů 447
evoluční obvody v extrémním prostředí 454
existenční kvantifikátor 125, 142, 239, 269
expanze ostrých úhlů 481
expertné systémy 159, 313
explanatorická síla 266, 277
explikačný potenciál 276
exprese genu 530
expresívna síla 266
extrakce příznaků 553

F

farebné kategórie 411
farebnosť, saturácia, alebo chroma 413
farebný lexikón jazyka 426
fenomenológia farebného videnia 415
fenotyp 439, 525
filament 99
filter technika výběru příznaků 554
filtrácia 296
filtry nelineární číslicové 444
fitness funkce 193, 439
FOAF (z angl. Friend Of A Friend) 64
folding 529
formálne aspekty jazyka 264
FPGA hradlová pole 450, 452
F-PID algoritmus 169, 176, 183

fragmentový identifikátor 56
funkčná magnetická rezonancia 322, 334
fuzifikácia 167, 178, 185
fuzzy logika 146
fuzzy model 160, 421
fuzzy zhukovacie metódy 156
fyzika herného sveta 28
fyziológia farebného vnímania 414

G

gamut 418
Generalized predictive control (GPC) 201
genetický algoritmus 192, 306, 442
genetický kód 531
genfis 182, 188
genom 525, 530
genotyp 525
genotypová variabilita 159
geometrické klamy 470, 481
geometrie sveta 30
geotechnika 158
graf 323
- bipartitný 325
- izomorfný 326
- kompletný 325
- náhodný 327
- orientovaný 323
- pomenovaný 81

H

Hammingova vzdálenost 559
Hebbovské učení 544
Hermanova mřížka 496
heuristika 232, 316, 439
hierarchia včelieho úľa 406
hierarchické siete 354, 369
hierarchické zhukovacie metódy 163
hierarchický prístup 42
Hintikkov prístup modelových množín 211
hmyz 382

- horizont 276
hradlová pole FPGA 450, 452
hranová konzistencia, AC (arc consistency) 297
HSV/HSB model 418
Humies, soutěž 442
hypotéza 231
- tesařského světa 489
- zkušeností s obrázky 487
- CH**
- chemický reaktor, fuzzy riadenie 186
chromatická adaptácia 417
chromosom 192, 439, 530
chyba 11, 548
chybová funkce 536
- I**
- ideálne prvky 280
iluze 33
implikácia 129, 178, 223, 236
importovanie ontológií 71
incidenčná matica 325
individuá 278
individuály 62
inferenčný mechanizmus 168, 314
informačný šum 393
integratívna sila 266, 282
inteligencia roja 383
Internet 322, 331, 398
inventár 30
inverzná konzistencia po ceste, PIC (path inverse consistency) 297
inverzní kinematika 28
iradiace 492
izomorfné grafy 326
izotermický regulátor 205
- J**
- jas 413
jazyk
- matematiky 263
- reprezentačný 54
jazyk, abstrakčný potenciál 273
-, diferenciačný potenciál 275
-, sémantický aspekt 360
-, syntaktický aspekt 360
jednoduchosť 289
jednoznačnosť 288
- K**
- k – regulárny graf 326
Kant 101
kapacita pamäte 13
karteziánske univerzum 271
kartézské genetické programování 440
kataskeyé 267
kategorizácia farieb 411, 421
Kauffman 108
klam
- nedokonalosť vnímání 501
- sdělení 507
- vzdálenosti objektů 484
- jako lež 506
- jako šifra 510
- křivých-rovných čar 500
- svinutého provazu 498
- zprostředkovává významy 512
klasifikace zrakových klamů 470
klasifikačný potenciál 278
klasterizačný koeficient 328
klauzula
- disjunktívna 217
- konjunktívna 217
kniha Předpisů 41
kodon 530
kódování sekvencí 550
kognitivní teorie zrakových klamů 472
kombinační obvody 443
komplement grafu 325
kompletný graf 325
komponent maximálny súvislý 326
konečné automaty 44
konformace 529
kongruencia 130
konjunkcia 127

konjunktívna klauzula 217
konjunktívna normálna forma (KNF)
217
konštrukcia tautologického vyplývania
228
kontext, klam 478
kontraktívnosť 7
konzistencia po ceste, PC (path
consistency) 297
konzistenčné algoritmy 296
konzistentná teória 226
konzistentné značkovanie (consistent
labeling) 295
korelace 13, 334, 365, **559**
kost 22, 24
krátkodobá pamäť 12, 43
Kripkeho model 247
kriteriálna funkcia 199
křížová validace 539
kvalita zdroja 404
kvality predikce 556
kvantifikátory 139, 239, 269
kvantová chováni, evoluce 456
kvaterniony 27

L

Lamarck 102
Laplaceov démon 271
laterální inhibice 494
lavíny 334
leave-one-out (LOOCV) validace 539
lineární algebra 27
linkovanie ontológií 70
literál 217
logická sila 265, 274
logistická funkce 545
logistika 158
lokálne prehľadávanie 307
lokálne reagujúce detektory 429
look-up-tabuľka 154
loutkář („drama manager“) 46
Lyell 102

M

makromolekuly 528
Malthus 104
Mamdani 162
mapovanie ontológií 70
matica
- incidenčná 325
- susednosti 325
- váh 4
meióza 527
mém 115
mesh 23
měsíční klam 478
messenger RNA 531
metaforická sila 266, 270, 280
metajazyk, 81
metoda hlavní komponenty 555
metóda modelovania 269
metóda redukcie 268
metodická sila 266, 270, 279
metody analýzy testovatelnosti 447
middleware 41
mitóza 527
modálna logika 246
modálna predikátová logika 254
model teórie 228
modelovanie znalostí (knowledge
modelling) 314
modus ponens 129
moment 548
morph targety 27
motion capture 24
mravce 382, 385
Müller- Lyerův klam 489
multiagentová simulácia farieb 428
multiagentový systém 383
multidisciplinárny opis 96
Munsellov farebný systém 420

N

náhodný graf 327
naturfilozofie 101
navigační body („way points“) 38

- navigační mřížky („navigation mesh“)
 40
 negácia 131
 nehierarchické zhlukovacie metódy
 164
 nekonzistentná teória 125, 226
 nekonzistentní scéna 476
 nelineární číslicové filtry 444
 nelineární dynamický model 162
 nemožný objekt 515
 neobmedzené (apeiron) 97
 neodarvinizmus 105
 NEPOMUK 80
 nepřímé kódování 552
 neuronová sieť 1, 540
 neuróny 4
 neviditeľný klam 506
 n-gram hashing 552
 n-gramy 550
 nová ontológia 108
 nukleotidy 528
- O**
- objektácie 263, 272
 oční sítnice – retina 473, 481, 485, **491**
 odporúčajúci systém 390
 odtieň 413
 ohraničenia 291
 Oken 101
 ontológie 54
 optimalizace evoluční 437
 orezanie priestoru kandidátov 296
 ortogonální kódování 550
 otvorené sémantické tablo 223
 otvorené vetvy 223
 OWL, Web Ontology Language 65
 označený systém 213
- P**
- PageRank 396
 pamät'
 - dlhodobá 12, 44
 - krátkodobá 12, 43
 pamät',
 - procedurální 42, 44
 - včely 396
 paobraz 492-493
 paprsek, vrhání („ray-casting“) 36
 patentová ochrana evolučního návrhu
 459
 patentová záplava 461
 PC (path consistency), konzistencia po
 ceste 297
 peptidické vazby 528
 percepční stálost 485
 perceptron 541
 perceptuální farebný priestor 419
 PIC (path inverse consistency),
 inverzná konzistencia po ceste 297
 PID regulátor 175
 PIMO, Personal Information Modeling
 82
 plánování 45
 Platón 98
 počítačová biologie 524
 podgraf 323
 podkoncept 63
 pohlavní buňky 527
 polygony 22
 pomenovaný graf 81
 pomenovávania farieb 411, **421**
 poristické umenie 268
 poruchy vnímania farieb 415
 potenciál
 - explikačný 276
 - jazyka 265
 - klasifikačný 278
 - regularizačný 279
 - unifikačný 281
 potenciality jazyka 264
 pozorovatelnost 447
 prahový neuron 542
 pravdivostná interpretácia 255
 predikátová logika 137, 239
 predikátový počet 269
 predikce sekundární struktury 562
 preferenčné pripájanie 331, 345
 prenos znalostí 313

- presnosť 288
prežívanie zdatných 102, 117
príbeh, vývoj 401
primární struktura 528
princíp
- jednoduchosti 472
- obvyklosti 473
- podformuly 211
prírodný výber 104
prírodná dedukcia 125, 235
prírodnosť 288
problém
- implementace 457
- redukcie 109
procedurální generování animací 27
procedurální paměť 42, 44
prokaryota 526
proof checker 145
propojení perceptů 503
prótesis 266
protein folding 529
proteom 528
protokultúra 115
prototyp farieb 412, 421
pruning 549
překážka 35
přenosová funkce 542
psychologické zakotvenie 272
puzzle pictures 512
pytagoreizmus 271
- R**
- rastúce siete 337
RDF, Resource Description
Framework 59
- Schema 62
reaktivní plán 42
reálne siete 330
reductio ad absurdum 210, 239
redukcia 315
-, problém 109
redukovaná konzistencia po ceste RPC
(reduced path consistency) 301
re-formulácie 263
regularizačný potenciál 279
rekonfigurovatelné obvody 438, 449
- analogové 451
rekurentné neurónové siete 1
relativistické teórie farieb 411
relatívni entropie 560
replikácia 115
replikátor 116
reprezentácia farieb 417
re-prezentácie 263
reprezentačný jazyk 54
reprodukční způsobilost 525
reverzibilní figura 469, 471, 474, 512
RGB model 418
riadenie spaľovania 154
řiditelnost 447
rig 22, 25
rozbalenie 90
rozlišovanie farieb 416
rozpoznávání vzorů 533
RPC (reduced path consistency),
redukovaná konzistencia po ceste 301
rychlost učení 544
- S**
- Saint-Hilaire 100
samoorganizácia 94, 383, 540
saturácia, chroma alebo farebnosť 413
sebareprodukcia 94
sekundární struktura 529
selekcija 115
sémantická paměť 44
sémantické
- mapovanie 70
- tablo 209, 223
- vyplývanie 247a
sémantický
- aspekt jazyka 360
- desktop 78
- web 53
sémantika Kripkeho 247
senzitivita 561
shlukování 535
sieť bezškálová 330

- sieť
- hierarchická 354
 - malého sveta 329, 336
 - mozgu 332, 365
 - , náhodné pripájanie uzlov 338, 345
 - rastúca 337
 - reálna 330
 - slovná 360
 - so zrýchleným rastom 346, 350
 - sociálne 335
 - , s prelinkovaním hrán (edge rewiring) 364
 - v biológii 332
- sigmoida 545
- sila
- explanatorická 266, 277
 - expresívna 266
 - integratívna 266, 282
 - jazyka 264
 - logická 265
 - metaforická 266, 270, 280
 - metodická 266, 270, 279
- singletonový konzistenčný algoritmus 301
- šírenie ohraničení (constraint propagation) 296
- škálovací exponent 330
- skolemizácia 240
- skriptovací jazyk 44
- sled 326
- slepá skvrna 491
- slice 450
- slovná sieť 360
- slučka 323
- Smullyanov diagramatický prístup 213
- sociálna evolúcia 113
- sociálne siete 335
- sociálny hmyz 383
- sociobiológia 114
- softwarové agenty 33
- softwarové patenty 460
- somatické buňky 527
- soutěž Humies 442
- specifická 561
- spektrálny polomer matice 8
- Spencer 101
- spĺňajúce priradenia (satisficing assignment) 295
- spĺňanie ohraničení 295
- splniteľná formula 222
- spoločenstvo
- stagnujúce 352
 - , ktoré sa rozvíja alebo zaniká 353
- spolupráce agentů 46
- spotrebná elektronika 149
- stavebníctvo 155
- steerovací techniky 34
- stigmergia 383
- stochastická optimalizácia 306, 403
- stopař 41
- stratégia ER (extend and revise) 314
- stromy chování 44
- stupeň uzla 324
- substitúcia 140
- subtraktívny farebný model 418
- sudoku 292, 303
- sympérasma 267
- syntaktický aspekt jazyka 360
- syntetická teória evolúcie 106
- T**
- ťah 326
- Takagi-Sugeno 155, 163
- Teilhard de Chardin 108
- tělo agenta 22
- teória
- evolúcie 89
 - grafov 323
 - konzistentná 226
 - nekonzistentná 226
 - opozičných procesov vnímania farieb 413
 - potencialít 263
 - sebeckého génu 114
 - transmutácie 103
- teorie založené na scéně 472, 476, **480**
- terciární struktura 529
- testovací množina 440, 534
- testovatelnost 447

- textura 232
theorem prover 145
transkripce 531
trénovací množina 155, 440, 534
trénovanie Echo State neurónových sietí 9
triedy, 62
trichromatická alebo Young-Helmholtzova teória vnímania farieb 413
trojzložkový farebný priestor 419
Troxlerův efekt 492
tvárové klamy 479
tvorba hypotézy 231
- U**
- unifikačný potenciál 281
uniformitarianizmus 104
univerzalistické teórie farieb 411
univerzálna teória evolúcie 106
univerzálnosť 289
- farebných kategórií 428
univerzálny darvinizmus 116
univerzálny kvantifikátor 125, 141, 239
URI, Uniform Resource Identifier 56
úrovne evolučného procesu 106
uvěřitelnost agentů 46
uzavreté sémantické tablo 223
uzavreté vetvy 223
uzol
-, priemerná vzdialenosť uzlov 326
-, priemerný stupeň 326
-, separácia 327
- V**
- variace 525
včelí úl 381, 384, 386
vedecká teória 94
vehikuly 116
vektory aktivít kategórií 432
viazaná premenná 139
vícevrstvé perceptrony 545
víceznačnost 474
- viditelný klam 510
virtuální agenti 21
virtuální vypravěčství („virtual storytelling“) 47
vířavná kategorie 431
vnímanie farieb 411, 412
vnútorný model, fuzzy prediktívne riadenie 203
vodíkové vazby 529
voľná farba 413
voxel 333
vrhání paprsku („ray-casting“) 36
výběr (selekce) 525
výběr příznaků 534
vyhledávání na webe 398
výroková logika 126
výrokové atómy 127
vyvíjející se obvody (evolvable hardware) 438
vzájemná informace (angl. mutual information) 561
- W**
- Web 2.0 73
Wikipedia 77
Witgenstein 283, 421
World Color Survey 423
wrap 23
- X**
- XML, eXtensible Markup Language, 57
- Z**
- zahnízení 110
základné farby 412
základné farebné kategórie 423
zdánlivá kontrakce rohových úhlů 495
zetetické umenie 267
zhluková analýza 163
znalostné modely 314
znalostný prístup 313

zoologická filozofia 100
zorné pole 41
zovšeobecňovanie 287
zpětné šíření chyby (angl.
backpropagation) 548
zrakový klam 467, 468
-, klasifikace 470