

Edícia výskumných textov  
Informatiky a informačných technológií

## **Umelá inteligencia a kognitívna veda I**

Vydané s podporou firmy Gratex International, a.s.

Táto publikácia vznikla za podpory grantovej agentúry VEGA SR v rámci grantových úloh VG1/4053/07 a VG1/0804/08

Vladimír Kvasnička  
Jiří Pospíchal  
Štefan Kozák  
Pavol Návrat  
Pavel Paroulek  
(editori)

# **Umelá inteligencia a kognitívna veda I**

Slovenská technická univerzita v Bratislave, 2009

---

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](http://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 16.12.2008, uznesenie č. 2008.37.1

© RNDr. Libor Běhounek, Ing. Michal Čerňanský, PhD., Ing. Peter Kostelník, PhD.,  
prof. Ing. Štefan Kozák, CSc., Mgr. Peter Krčah, prof. Ing. Vladimír Kvasnička, DrSc.,  
doc. RNDr. MUDr. Petr Maršálek, PhD., doc. Ing. Pavel Nahodil, CSc.,  
prof. Mgr. Martin Pelikán, PhD., prof. RNDr. Jiří Pospíchal, DrSc., doc. PhDr. Karel  
Pstružina, CSc., RNDr. Martin Takáč, PhD., Ing. Miloš Železný, PhD.

ISBN XXX-XX-XXX-XXXX-X

## AUTORSKÝ KOLEKTÍV KNIHY

**RNDr. Libor Běhounek**

E-mail: behounek@cs.cas.cz

**Ing. Michal Čerňanský, PhD.**

E-mail: cernansky@fiit.stuba.sk

**Ing. Peter Kostelník, PhD.**

E-mail: peter.kostelnik@gmail.com

**prof. Ing. Štefan Kozák, CSc.**

E-mail: kozak@fiit.stuba.sk

**Mgr. Peter Krčah**

E-mail: peter.krcah@ruk.cuni.cz

**prof. Ing. Vladimír Kvasnička, DrSc.**

E-mail: kvasnicka@fiit.stuba.sk

**doc. RNDr. MUDr. Petr Maršálek, PhD.**

E-mail: Petr.Marsalek@lf1.cuni.cz

**doc. Ing. Pavel Nahodil, CSc.**

E-mail: nahodil@fel.cvut.cz

**prof. Mgr. Martin Pelikán, PhD.**

E-mail: pelikan@cs.umsl.edu

**prof. RNDr. Jiří Pospíchal, DrSc.**

E-mail: pospichal@fiit.stuba.sk

**doc. PhDr. Karel Pstružina, CSc.**

E-mail: Pstruzin@vse.cz

Ústav informatiky Akademie věd České republiky, v.v.i., Pod Vodárenskou věží 2, 182 07 Praha 8, Česká republika.

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

NESS KDC, s.r.o, Puškinova 3, 040 01 Košice

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

Ústav výpočetní techniky, Univerzita Karlova v Praze, Ovocný Trh 5, 116 36 Praha 1, Česká republika

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

Ústav patologické fyziologie 1. lékařské fakulty UK, U nemocnice 5, CZ-128 53, Praha 2

Katedra kybernetiky – CAK (Centrum aplikované kybernetiky), Elektrotechnická fakulta ČVUT v Praze, Karlovo nám. 13, 122 35 Praha 2

Department of Mathematics and Computer Science, 320 CCB, University of Missouri in St. Louis, St. Louis, MO 63121, USA,

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

Katedra filosofie, Národohospodářská fakulta, Vysoká škola ekonomická v Praze, W. Churchilla 4, 130 00 Praha 3

**RNDr. Martin Takáč, PhD.**

E-mail: [takac@ii.fmph.uniba.sk](mailto:takac@ii.fmph.uniba.sk)

**Ing. Miloš Železný, PhD.**

E-mail: [zelezny@kky.zcu.cz](mailto:zelezny@kky.zcu.cz)

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

Katedra kybernetiky, Fakulta  
aplikovaných věd, Západočeská  
univerzita v Plzni,  
Univerzitní 8, 306 14 Plzeň

## Predhovor

Umelá inteligencia a kognitívna veda (UIaKV) spolu, ktoré tvoria úzko previazaný komplex vied zaoberajúcich sa riešením zložitých problémov ľudskej mysle, zaznamenali v priebehu ostatných 10 – 20 rokov mimoriadne prudký rozvoj tak v teoretickej, ako aj v aplikačnej oblasti. Aplikácia metód a algoritmov umelej inteligencie a kognitívnej vedy pri riešení zložitých problémov rozhodovania a riadenia tak v technických, ako aj v netechnických vedách zaznamenala nebývalý rozvoj hlavne vďaka vzniku metód výpočtovej inteligencie (fuzzy množín, umelých neurónových sietí a evolučných algoritmov). Metódy umelej inteligencie využívajú rôzne metafory a paradigmy z kognitívnej vedy (a neurovedy) na návrh moderných systémov riadenia za vzniku novej oblasti riadenia nazývanej „kognitívne riadenie“ a ktoré je založené na symbolickom prístupe k formulácii základných teoretických prístupov tak v umelej inteligencii, ako aj v kognitívnej vedy.

Ďalším trendom v oblasti využívania UIaKV sú tzv. vnorené softwarové a hardwarové systémy, ktoré sú reprezentované všadeprítomnými zabudovanými počítačovými systémami, ktoré tvoria súčasť moderných výrobkov, zariadení a procesov. Návrh inteligentných vnorených systémov s využitím metód UIaKV sa dnes nezaobíde bez využitia kombinovaných metodík, ktorý poskytuje kognitívna veda, ktorá skúma a modeluje princípy organizácie a fungovania tak prirodzených ako aj umelých inteligentných systémov. Pretože predmetom štúdia kognitívnej vedy sú fenomény súvisiace s myslením, mnohé inteligentné algoritmy modelovania obsiahnuté vo vnorených systémoch obsahujú ako svoju integrálnu súčasť rôzne kombinácie formálnych techník s metódami logiky, psychológie a neurovedy. Týmto prienikom poznatkov kognitívnej vedy do problematiky rozhodovania a riadenia vnorených systémov sa získajú nové a lepšie vlastnosti výrobkov a zariadení.

Komunita pracovníkov z oblasti umelej inteligencie a kognitívnej vedy (a taktiež aj z pridružených humanitných a behaviorálnych disciplín) je združená na Slovensku a v Česku okolo konferencie *Kognícia a umelý život*. Táto konferencia sa každoročne koná striedavo na Slovensku a v Česku a bude mať v budúcom r. 2010 svoje desiate výročie. Príspevky prezentované na tejto konferencii odrážajú súčasný stav oboch vedných disciplín u nás, ktoré sa búrlivo rozvíjajú a majú silne interdisciplinárny ba až transdisciplinárny charakter, integrujú pracovníkov z rôznych oblastí filozofie, biológie, chémie a informatiky. Na minuloročnej konferencii *Kognícia a umelý život VIII* konanej na Vysoké škole ekonomickej v Prahe vznikla myšlienka pravidelne vydávať edíciu *Umelá inteligencia a kognitívna veda* s pomocou komunity odborníkov pridruženej k tejto konferencii. Zámer tejto edície je, že jednotlivé publikácie – knihy budú obsahovať vybrané partie umelej inteligencie a kognitívnej vedy na úrovni bakalárskeho štúdia. Zmienená interdisciplinarita a transdisciplinarita sa odráža aj v predkladanej knihe *Umelá inteligencia a kognitívna*

*veda I*, a je dokladom toho, že vedné odbory umelej inteligencie a kognitívnej vedy našli pevné zakotvenie tak vo filozoficky a spoločensko-vedne, ako aj prírodovedne a informaticky orientovanej komunite pracovníkov v oblastiach filozofie mysle, umelej inteligencie, kognitívnej vedy a umelého života.

Ďakujeme všetkým autorom, ktorí prispeli svojimi príspevkami do tejto knihy, Ing. Jánovi Cigánkovi za pomoc pri formátovaní textu tejto knihy, spoločnosti Gratex International, a.s., reprezentovanej Ing. Ivanom Poláškom, PhD., za finančnú podporu vydania tejto knihy. Na záver ďakujeme vedeniu Fakulty informatiky a informačných technológií STU v Bratislave, menovite dekanovi fakulty prof. RNDr. Ľudovítovi Molnárovi, DrSc., za zaradenie tejto knihy do fakultného edičného radu „Edícia výskumných textov“.

V Bratislave, marec 2009

Vladimír Kvasnička  
Jiří Pospíchal  
Štefan Kozák  
Pavol Návrat  
Pavel Paroulek  
(editori)



## Obsah

## Predhovor

<b>(1) Formální fuzzy logika .....</b>	<b>1</b>
<i>L. Běhounek</i>	
1 Graduální vlastnosti a jejich logika .....	1
2 Výroková fuzzy logika .....	6
3 Rozšíření výrokové fuzzy logiky .....	19
4 Aplikace formální fuzzy logiky .....	25
5 Závěr .....	34
Literatura .....	35
<b>(2) Rekurentné neuronové siete .....</b>	<b>37</b>
<i>M. Čerňanský</i>	
1 Úvod .....	37
2 Dopředné neuronové siete .....	39
3 Rekurentné neuronové siete .....	45
4 Gradientové trénovací algoritmy .....	49
5 Trénovací algoritmy využívající Kalmanovu filtraci .....	54
6 Dynamika rekurentních neuronových sítí .....	60
7 Siete s echo stavmi .....	68
8 Aplikácie rekurentných neuronových sítí .....	72
9 Implementácia rekurentných neuronových sítí .....	76
10 Závěr .....	84
Literatúra .....	84
<b>(3) Praktický úvod do symbolickej umelej inteligencie .....</b>	<b>87</b>
<i>P. Kostelník</i>	
1 Symbolická umelá inteligencia .....	87
2 Stavový priestor, riešenie úloh a hranie hier .....	93
3 Produkčné pravidlá a expertné systémy .....	107
4 Predikátová logika a automatické dokazovanie .....	116
5 Sémantické modelovanie znalostí .....	129
6 Závěr .....	137
Literatúra .....	138
<b>(4) Inteligentné vnorené systémy .....</b>	<b>139</b>
<i>Š. Kozák</i>	
1 Analýza metód a prostriedkov inteligentnej syntézy vnorených systémov .....	139

2	Analýza súčasného stavu využívania metód modelovania, simulácie a syntézy vnorených systémov použitím konvenčných a inteligentných techník založených na fuzzy množinách, umelých neurónových sieťach a genetických algoritmoch .....	161
	Literatúra .....	192
<b>(5)</b>	<b>Evolučný návrh robotických organizmov .....</b>	<b>195</b>
	<i>P. Krčah</i>	
1	Úvod .....	195
2	Základy evolučných algoritmov .....	196
3	Metódy ovládania robotov .....	199
4	Problém reprezentácie robota v evolučných algoritmoch .....	203
5	Meranie úspešnosti robotov .....	208
6	Od simulácie k reálnym robotom .....	215
7	Záver .....	225
	Literatúra .....	225
<b>(6)</b>	<b>Darwinovská evolúcia ako algoritmus .....</b>	<b>231</b>
	<i>V. Kvasnička, J. Pospíchal</i>	
1	Úvod .....	231
2	Darwinovský systém .....	243
3	Eigenova a Schustrova teória molekulárneho darvinizmu .....	248
4	Evolúcia v multiagentových systémoch .....	264
5	Záver .....	274
	Literatúra .....	277
<b>(7)</b>	<b>Simulace v neurovědách, příklad modelu prostorového slyšení .....</b>	<b>281</b>
	<i>P. Maršálek</i>	
1	Úvod .....	281
2	Teorie zpožd'ovací linky .....	285
3	Teorie pravděpodobnostního zpoždění .....	286
4	Časová přesnost ve sluchové dráze .....	290
5	Diskuse .....	291
6	Závěr .....	295
	Literatura .....	296

<b>(8)</b>	<b>Humanoidní robotika včera, dnes a zítra ...</b>	<b>299</b>
	<i>P. Nahodil</i>	
	1 Úvod .....	299
	2 Od začátků robotizace k humanoidům .....	300
	3 Výchozí aparát vývoje inteligentního agenta – humanoida ....	309
	4 Vytváření humanoidních robotů .....	314
	5 Úspěšné realizace robotů humanoidního typu – analýza .....	322
	6 Budoucnost koexistence člověk – humanoid .....	328
	7 Závěr .....	332
	Literatura .....	333
<b>(9)</b>	<b>Evolučné algoritmy .....</b>	<b>335</b>
	<i>M. Pelikán</i>	
	1 Úvod .....	335
	2 Evolučný algoritmus .....	336
	3 Jednoduchý genetický algoritmus .....	338
	4 Genetické programovanie .....	346
	5 Evolučné stratégie .....	350
	6 Niektoré úspešné aplikácie evolučných algoritmov .....	351
	7 Záver .....	352
	Literatúra .....	353
<b>(10)</b>	<b>Funkční model mysli a jeho metodologický význam pro zkoumání operací lidského myšlení .....</b>	<b>355</b>
	<i>K. Pstružina</i>	
	1 Funkční model mysli a jeho metodologický význam .....	355
	2 Funkční model mysli - úroveň vnímání a konstitucí světa ....	359
	3 Operace lidského myšlení a intencionální obsahy mysli .....	375
	4 Tvorba novosti .....	380
	5 Závěr .....	392
	Literatura .....	392
<b>(11)</b>	<b>Konstruktivistický přístup k štúdiu kognície .....</b>	<b>395</b>
	<i>M. Takáč</i>	
	1 Úvod .....	395
	2 Teórie významu .....	396
	3 Od predverbálnych významov k jazyku: evolučný pohľad ....	400
	4 Reprezentácia významov v umelých systémoch .....	402
	5 Konštrukcia významov .....	408
	6 Výpočtové modely konštrukcie významov .....	410

7 Záver .....	416
Literatúra .....	417
<b>(12) Technologie audiovizuální syntézy řeči a její aplikace .....</b>	<b>425</b>
<i>M. Železný</i>	
1 Úvod do audiovizuální syntézy řeči .....	425
2 Datové korpusy .....	428
3 Technologie syntézy .....	432
4 Trojrozměrné snímání .....	438
5 Aplikace .....	439
6 Závěr .....	445
Literatura .....	449
<b>Index .....</b>	<b>451</b>

# Formální fuzzy logika

Libor BĚHOUNEK<sup>1</sup>

**Abstrakt.** Formální fuzzy logika je specifickou větví vícehodnotových logik, zaměřenou na logické modelování výroků o vlastnostech připisovatelných v různých stupních intenzity. Sémantika založená na t-normách umožňuje její rigorózní vybudování axiomatickou metodou neklasických logik. Tento aparát pak může sloužit k matematickému modelování graduálních vlastností a být teoretickým podkladem řízení pomocí fuzzy metod a usuzování kognitivních agentů o graduálních jevech. Formální fuzzy logiku je však třeba rozlišovat od širšího spektra fuzzy metod, které převládají v aplikacích a mají odlišné metodologické předpoklady.

## 1 Graduální vlastnosti a jejich logika

Klasická logika popisuje vztahy vyplývání mezi výroky o vlastnostech, které jsou dvouhodnotové – objekty je buďto mají, nebo nemají. Množiny vymezené takovými vlastnostmi jsou tudíž ostré – prvky do nich buď (plně) spadají, anebo (vůbec) nespádají. Mnoho vlastností, s nimiž se setkáváme v přirozeném světě, však má naopak stupňovitou, neboli – jak budeme říkat – *graduální* povahu: například vlastnosti *vysoký*, *horký*, *zelený*, *zdravý*, *nákladný*, *obtížný*, *odpočatý*, *strmý* či *blízký*. Všechny tyto (a mnohé další) atributy můžeme některým objektům připisovat ve větší či menší míře než jiným. Soubory objektů takovými vlastnostmi vymezené tudíž nemají ostrou, nýbrž graduální hranici: některé objekty do nich spadají více a jiné méně; nejde tedy o množiny ve smyslu klasické matematiky. Klasická dvouhodnotová logika existenci graduálních vlastností spíše obchází: jejich chování buď opisuje pomocí idealizací (tj. oproti skutečnému stavu věcí předpokládá, že ostrou hranici mají), nebo je modeluje pomocí jiných konstrukcí (například funkcí), upírajíc jim jejich povahu *vlastností* objektů.

Výroky o graduálních vlastnostech objektů jsou předmětem určitých formálních systémů logiky neklasické, zvaných *fuzzy logiky*. Protože však názvem „fuzzy logika“ bývají označovány i partie s logikou samotnou související jen velmi volně (jimž by více slušel název *fuzzy metody*), přidáváme zde pro odlišení adjektivum *formální*, které má vyznačit příslušnost k oboru formální logiky. Formální fuzzy logika se tedy zabývá

---

<sup>1</sup> Ústav informatiky Akademie věd České republiky, v.v.i., Pod Vodárenskou věží 2, 182 07 Praha 8, Česká republika. E-mail: behounek@cs.cas.cz

logickými vztahy výroků o graduálních vlastnostech objektů – jejich vzájemným vyplýváním, tautologičností, axiomatizovatelností, sémantickým modelováním apod. Jak uvidíme, pro graduální vlastnosti je vhodné používat jiné odvozovací zákonitosti než pro klasické vlastnosti ostré. Zákony formální fuzzy logiky se tak budou lišit od zákonů logiky klasické.

Je jasné, že o graduálních vlastnostech umíme nějakým způsobem mluvit i v rámci logiky klasické – i proto jsme se v historii bez fuzzy logiky dlouhou dobu jakž takž obešli. V případě graduálních vlastností měřených nějakou konkrétní veličinou (například *vyšoký*) můžeme totiž hovořit přímo o jejich *hodnotách*: řekneme například, že daný člověk má výšku 182 centimetrů. Namísto vlastnosti „vyšoký“ tak vlastně hovoříme o *funkci*, která tuto graduální vlastnost popisuje tím, že každému člověku přiřazuje jeho výšku. O funkcích, které graduální vlastnosti reprezentují, už lze usuzovat klasicky – samy o sobě totiž graduální nejsou: výrok „Jan je vyšoký 182 cm“ je ostrý (buď pravdivý, nebo nepravdivý). V případě takových vlastností jako *strmý* či *zelený*, kde je měřená veličina problematická,<sup>2</sup> či dokonce tam, kde ji jako u vlastností *krásný* či *obtížný* vůbec nelze nalézt, je možno existenci takové funkce prostě předpokládat (takovéto idealizace provádíme v matematickém modelování běžně). Tímto způsobem lze tedy graduální vlastnosti reprezentovat, aniž by bylo třeba opouštět klasické prostředky dvouhodnotové logiky a matematiky, a takto se ke graduálním vlastnostem také odjakživa přistupovalo.

Krokem na půl cesty směrem k formální fuzzy logice, tedy přímému hovoru o graduálních vlastnostech jakožto *vlastnostech*, byla až teorie fuzzy množin, navržená L. A. Zadehem<sup>3</sup> v roce 1965 v článku [25] a od té doby mohutně rozvíjená. Funkce reprezentující graduální vlastnost je v této teorii interpretována tak, že vyjadřuje přináležení jednotlivých objektů do „množiny“ touto vlastností vymezené (např. „množiny“ všech vyšokých lidí). Tato „množina“ samozřejmě nemůže být množinou v klasickém smyslu (neboť ty jsou vymezeny jedinečně dvouhodnotovými, ostrými vlastnostmi), nýbrž je množinou graduální (čili vymezenou graduální vlastností). Pro odlišení od klasických dvouhodnotových množin nazval Zadeh takové soubory prvků *fuzzy množinami*.<sup>4</sup> Funkci reprezentující fuzzy množinu nazval *funkcí příslušnosti* této fuzzy množiny a její hodnotu pro daný objekt *stupněm příslušnosti* tohoto objektu do dané fuzzy množiny. O funkcích příslušnosti předpokládal, že argumentům z nějaké ostře vyme-

<sup>2</sup> Např. proto, že úhlová strmost je na různých místech kopce různá (přesto chceme z hlediska strmosti porovnávat celé kopce); navíc není jasné, zda do ní počítat každou hroudu či výmol, vegetaci, atd. Podobně námitky lze uvést i v případě měření zelenosti třeba u kabátů.

<sup>3</sup> Lotfi A. Zadeh (\*1921), americký aplikovaný matematik a informatik perského původu, profesor na Kalifornské univerzitě v Berkeley, zakladatel teorie fuzzy množin. Přes vysoký věk je v roce vydání této knihy v oboru stále aktivní a hojně přednáší na konferencích po celém světě.

<sup>4</sup> „Fuzzy“ znamená v angličtině *rozmazaný* či *neostrý*, ale i například *chundelatý*, *střapatý*, *zmatený* či *přiopilý*. Zadehova volba tohoto termínu patrně přispěla k popularitě teorie fuzzy množin. Žádný překlad českým slovem se neujal a v českých textech se ustálilo nesklonné anglické adjektivum *fuzzy*.

zené klasické množiny  $X$  přiřazují hodnoty v intervalu  $[0, 1]$ .<sup>5</sup> Stupeň příslušnosti 1 pak znamená plné nálezení do fuzzy množiny a stupeň 0 plné nenáležení; mezilehlé stupně vyjadřují různou míru částečného nálezení. Charakteristické funkce klasických množin jsou tak speciálním případem funkcí příslušnosti (s hodnotami pouze 0 a 1), a klasické množiny lze tedy považovat za speciální druh fuzzy množin.<sup>6</sup>

Tento krok měl závažné konceptuální důsledky. Chápeme-li funkce (příslušnosti) jako (fuzzy) množiny, můžeme pomocí nich kupříkladu definovat (fuzzy) množinové operace průniku, sjednocení či doplňku, například tak jako v původním Zadehově článku [25].<sup>7</sup>

**Definice:** Necht'  $\mu_A, \mu_B: X \rightarrow [0, 1]$  jsou funkce příslušnosti fuzzy množin  $A, B$ . Potom jejich *průnikem*, *sjednocením* a *doplňkem* jsou fuzzy množiny dané následujícími funkcemi příslušnosti:

$$\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)) \quad (1)$$

$$\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)) \quad (2)$$

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad (3)$$

Řekneme dále, že fuzzy množina  $A$  je *částí* fuzzy množiny  $B$  (což zapisujeme  $A \subseteq B$ ), když pro jejich funkce příslušnosti  $\mu_A, \mu_B$  platí:

$$\mu_A(x) \leq \mu_B(x), \text{ pro všechna } x \in X. \quad (4)$$

Protože je fuzzy množina svojí funkcí příslušnosti jednoznačně určena, často s ní bývá přímo ztotožněna; místo  $\mu_A(x)$  pak píšeme jen  $A(x)$  nebo krátce  $Ax$ .

Zadehova teorie fuzzy množin tedy bere graduální vlastnosti reprezentované funkcemi příslušnosti vážně alespoň v tom smyslu, že jejich extenze (tj. soubory prvků jimi vymezené) chápe jako graduální *množiny*, a definuje pro ně pojmy tomu odpovídajícím způsobem. Všimá si také, že pro ně některé množinové zákony platí stejně dobře jako pro množiny ostré, zatímco jiné selhávají či platí jen omezeně. V platnosti pro ně zůstávají kupříkladu distributivní zákony

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$$

$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C).$$

<sup>5</sup> Hodnoty v jiných intervalech je možno do tohoto intervalu normalizovat. Později byly fuzzy množiny zobecněny i na jiné struktury stupňů příslušnosti, např. libovolné svazy apod. Pro jednoduchost ponecháme tato zobecnění stranou.

<sup>6</sup> Pro klasické dvouhodnotové množiny či vlastnosti se vžilo anglické adjektivum *crisp*, znamenající *ostrý*, *břítký*, *křupavý* či *řízný* (termín je podobně hravý jako slovo *fuzzy*). V českých textech bývá obvykle překládáno jako „ostrý“, mluvíme tedy o ostrých množinách jakožto protikladu či speciálním případu fuzzy množin.

<sup>7</sup> Později se ukázalo, že je často účelnější definovat tyto operace jinak, respektive že lze definovat mnoho smysluplných „průníků“, „sjednocení“ a „doplňků“ fuzzy množin, vhodných pro různé účely a situace (viz dále v oddílu 4).

Klasicky platný zákon

$$(A \cap B) \cup (A \cap \setminus B) = A$$

však obecně platí pouze pro ostré množiny: pro fuzzy množiny je platná jen inkluze

$$(A \cap B) \cup (A \cap \setminus B) \subseteq A.$$

Tento způsob nakládání s funkcemi příslušnosti jakožto (graduálními) množinami se ukázal být velice plodným, zejména v případě fuzzy relací (čili fuzzy množin dvojic popř.  $n$ -tic prvků), kde lze např. definovat takové operace jako obraz fuzzy množiny přes fuzzy relaci či zobecnit pro fuzzy relace vlastnosti reflexivity, symetrie a tranzitivitu; pomocí nich je potom možno studovat fuzzy uspořádání, relace podobnosti, fuzzy preferenční relace apod. Zkoumají se také různé varianty fuzzy algebry, fuzzy topologie, či fuzzy míry. Většina těchto pojmů našla široké praktické uplatnění v inženýrských aplikacích od rozpoznávání obrazů přes regulaci průmyslových procesů až po medicínskou diagnostiku. O teorii fuzzy množin a jejich aplikacích se lze poučit například v knihách [18], [19] či [20].

Lze říci, že celá teorie fuzzy množin i její bohaté aplikace jsou založeny právě na rozhodnutí chápat funkce příslušnosti jako graduální množiny. Přestože však tradiční teorie fuzzy množin studuje množiny vymezené graduálně, stále o nich uvažuje do značné míry klasicky, jako o funkcích příslušnosti. Fuzzy množiny zde nejsou primitivním pojmem, jako je tomu v klasické matematice u množin ostrých, nýbrž zůstávají reprezentovány ostrými objekty klasické matematiky.<sup>8</sup>

Další krok k formální fuzzy logice – tedy přímému nakládání s graduálními vlastnostmi, nezprostředkovanému klasickými ostrými reprezentacemi – učinil J. A. Goguen<sup>9</sup> ve svém článku [13] z roku 1969. Tento krok je založen na pozorování, že podobně jako operace průniku, sjednocení a doplňku ostrých množin odpovídají logickým spojkám *a*, *nebo* a *ne* (značeným po řadě  $\wedge$ ,  $\vee$  a  $\neg$ ) prostřednictvím vztahů

$$\begin{aligned} x \in A \cap B &\equiv x \in A \wedge x \in B \\ x \in A \cup B &\equiv x \in A \vee x \in B \\ x \in \setminus A &\equiv \neg(x \in A), \end{aligned}$$

tak i průnik, sjednocení a doplněk fuzzy množin odpovídá podle (1)–(3) operacím min, max a  $1 - x$ . Tyto funkce tedy můžeme považovat za reprezentace výrokových spojek *fuzzy* logiky a v analogii s logikou klasickou pak interval  $[0, 1]$  chápat jako množinu

<sup>8</sup> V důsledku toho jsou i matematické vlastnosti definované v teorii fuzzy množin obvykle ostré, nikoli graduální – kupříkladu inkluze fuzzy množin definovaná pomocí (4) je ostrým vztahem, přestože lze jistě porovnávat míru, *nakolik* je jedna fuzzy množina částí druhé (totiž nakolik dobře jedna funkce příslušnosti majorizuje druhou). Graduální pojem inkluze fuzzy množin byl sice později skutečně zaveden a je i příležitostně používán, pro jiné pojmy tradiční teorie fuzzy množin to však běžné není.

<sup>9</sup> Joseph Amadee Goguen (1941–2006), americký informatik, student L. A. Zadeha. Vedle raných prací o teorii fuzzy množin se později zabýval aplikacemi teorie kategorií v informatice a v neklasických logikách, softwarovým inženýrstvím a dalšími tématy informatiky a aplikované logiky.



*pravdivostních hodnot* fuzzy výroků, přičemž klasické hodnoty 0 a 1 jsou jejich speciálními případy znamenajícími úplnou pravdivost resp. nepravdivost. Funkce na intervalu  $[0, 1]$ , které spojky fuzzy logiky realizují, jsou zobecněním pravdivostních tabulek klasické logiky; říkáme jim proto *pravdivostní funkce*. Podobně jako se zákony teorie fuzzy množin v některých ohledech lišily od zákonů klasické teorie množin, liší se i zákony fuzzy logiky od zákonů logiky klasické. Například klasické distributivní zákony

$$\alpha \wedge (\beta \vee \gamma) \equiv (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$$

$$\alpha \vee (\beta \wedge \gamma) \equiv (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$$

platí ve fuzzy logice stejně dobře jako v logice klasické (neboť v obou případech jde vlastně jen o jiný zápis vlastností minima a maxima pravdivostních hodnot). Naproti tomu klasické zákony sporu a vyloučení třetího v podobě

$$\neg(\alpha \wedge \neg\alpha)$$

$$\alpha \vee \neg\alpha$$

ve fuzzy logice obecně neplatí – nemusejí totiž vždy nabývat hodnoty 1, tj. být úplně pravdivé: např. pro pravdivostní hodnotu  $\alpha$  rovnou 0,5 je hodnota  $\alpha \vee \neg\alpha$  rovna jen  $\max(0,5; 1 - 0,5) = 0,5$ . Tyto zákony jsou platné jen pro ostré hodnoty 0 a 1, což je i důvod, proč jsou uznávané klasickou logikou, jež se na tyto hodnoty omezuje.<sup>10</sup>

Zavedení spojek fuzzy logiky je dalším posunem k formální logice graduálních vlastností, neboť stupeň náležení do fuzzy množiny lze díky nim vyložit jako míru *pravdivosti* připsání příslušné graduální vlastnosti. Samo používání fuzzy spojek však ještě není *formální* logikou, nýbrž jen sémantickým modelem graduálních vlastností: doposud se totiž bavíme o pravdivostních hodnotách, tedy o hodnotách funkcí, kterými graduální vlastnosti reprezentujeme. Používání spojek fuzzy logiky je pouze elegantním zápisem a způsobem vyjadřování o těchto funkcích.

O formální logice graduálních vlastností můžeme hovořit teprve tehdy, umožní-li nám provozovat to, co klasická logika v případě dvouhodnotových vlastností: totiž studovat přenášení pravdivosti (zde ovšem nikoli ostré, nýbrž té graduální),<sup>11</sup> stanovit jejich pravidla, *axiomatizovat je*, a umožnit tak odhlédnout od konkrétních hodnot funkcí příslušnosti. Právě k tomu však Zadehův a Goguenův tah otevírá cestu: v příštích oddílech ukážeme, že usuzování pomocí (vhodných) spojek fuzzy logiky je možno axiomatizovat, a studovat pak formální pravidla logiky graduálních vlastností. Budeme

<sup>10</sup> Později uvidíme, že vedle  $\min$ ,  $\max$  a  $1-x$  existují daleko lepší volby spojek, pro které zákony sporu či vyloučení třetího platí. Časté zevšeobecnující tvrzení (zejména filosofických logiků), že „ve fuzzy logice neplatí zákon sporu“, je tedy poněkud nepřesné.

<sup>11</sup> S graduální pravdivostí mají někteří filosofové problém, namítajíce (většinou ovšem bez jasných argumentů), že pravdivost může být toliko dvouhodnotová. Zde však graduální pravdivostí myslíme právě a pouze připsovaný stupeň graduální vlastnosti, s nímž ve fuzzy logice *nakládáme obdobně*, jako klasická logika nakládá s pravdivostí dvouhodnotovou. Jde tedy jen o pohodlný technický termín – namísto o přenášení částečné pravdivosti bychom mohli stejně dobře mluvit jenom o přenášení stupňů graduálních vlastností. Komu však představa odstupňované pravdivosti nečiní problém, může termín chápat i doslovně.

přítom následovat systematizaci formální fuzzy logiky podanou P. Hájkem<sup>12</sup> v jeho knize [14] z roku 1998. První formální systém fuzzy logiky explicitně motivovaný teorií fuzzy množin však vytvořil již v 70. letech J. Pavelka<sup>13</sup> a rozpracoval jej V. Novák.<sup>14</sup> Dodatečně se pak ukázalo, že ojedinělé kalkuly vhodné pro logiku graduálních vlastností byly vlastně známy již dávno před teorií fuzzy množin; byly však původně motivovány jinými úvahami a jako systémy formální fuzzy logiky byly rozpoznány až relativně nedávno – zejména logika Łukasiewiczova,<sup>15</sup> jež byla jako trojhodnotová vytvořena již před rokem 1920 a později zobecněna na nekonečněhodnotovou, a Gödelova,<sup>16</sup> původně definovaná jen jako pomocný prostředek důkazu nekonečněhodnotovosti intuicionistické logiky. Vydání Hájkovy monografie [14] v roce 1998 pak stimulovalo prudký rozvoj oboru, vedoucí k vytvoření a prozkoumání desítek formálních systémů fuzzy logiky. Některými výsledky a aplikacemi tohoto výzkumu se budeme zabývat v následujících oddílech; vzhledem k dnešní rozsáhlosti oboru však bude možno podat jen úvodní informaci.

## 2 Výroková fuzzy logika

V tomto oddíle představíme základní systémy výrokové fuzzy logiky. Předpokládáme přítom elementární znalosti klasické logiky dostupné v běžných učebnicích.

V předchozím oddíle jsme vyjasnili, že fuzzy logika rozšiřuje klasické pravdivostní hodnoty 0 a 1 na celý reálný interval  $[0, 1]$ . Pravdivostní tabulky klasických výrokových spojek je zjevně možno na interval  $[0, 1]$  rozšířit nekonečně mnoha způsoby, jelikož klasickou logikou jsou fixovány pouze jejich hodnoty v bodech 0 a 1. Ne všechny funkce se však pro tyto účely hodí – pravdivostní funkce výrokových spojek by měly splňovat určité vlastnosti, které od těchto spojek očekáváme. Možností, které vlastnosti od nich vyžadovat, je opět nepřehledně; z hlediska formální logiky však lze ukázat (viz např. [2]), že některé vlastnosti spojek jsou nepostradatelné, mají-li se spojky chovat správně vůči logickému vyplývání (tj. přenášení stupňů graduálních vlastností mezi výroky). Inženýrské fuzzy metody tímto požadavkem svázané nejsou, neboť stupně příslušnosti se v nich nepoužívají k *logickému* odvozování, nýbrž k jiným

<sup>12</sup> Petr Hájek (\*1940), český matematik a logik. Vedle zásadního přínosu k formální fuzzy logice (od poloviny 90. let) se dříve zabýval i teorií množin, metamatematikou aritmetiky a metodami těžení z dat.

<sup>13</sup> Jan Pavelka (†2007) publikoval článek o fuzzy logice založený na své disertaci v roce 1978, poté se matematice přestal věnovat; jeho práce je však dodnes hojně citovaná.

<sup>14</sup> Vilém Novák (\*1951) vede Ústav pro výzkum a aplikace fuzzy modelování Ostravské univerzity. Vedle teoretického výzkumu fuzzy logiky a teorie fuzzy množin se zabývá i jejich aplikacemi a využitím pro jazykové modelování a řízení.

<sup>15</sup> Jan Łukasiewicz (1878–1956), polský matematik a logik, příslušník Lvovsko-varšavské logické školy, tvůrce prvního formálního systému vícehodnotové logiky. Vynalezl též polskou notaci, později často užívanou v informatice.

<sup>16</sup> Kurt Gödel (1906–1978), rakouský a později americký logik narozený v Brně, kde žil až do roku 1924. Je znám zejména svými větami o úplnosti predikátové logiky a neúplnosti aritmetiky. Dvoustránkový článek implicitně definující logiku zvanou dnes Gödelova publikoval v roce 1932.

účelům (zejména k jejich agregaci a následnému vyhodnocení za účelem řízení). V důsledku toho mohou být v inženýrských fuzzy metodách spojky definovány ad hoc, podle potřeb konkrétních aplikací; budování obecné teorie fuzzy metod je pak značně obtížné. Ve formální fuzzy logice jakožto logické teorii graduálních vlastností však spojky musejí splňovat určité podmínky, které nyní uvedeme a podáme i jejich intuitivní motivaci.

Začněme u konjunkce, jejíž pravdivostní funkci na  $[0, 1]$  označíme  $*$ , zatímco spojku samotnou budeme značit  $\&$ .<sup>17</sup> Na pravdivostní funkci konjunkce je přirozené klást následující podmínky (pro všechna  $x, y, z \in [0, 1]$ ):

- $x * y = y * x$  (komutativita)

Tato podmínka vyjadřuje, že v konjunkci tvrdíme *oba* výroky, nezávisle na tom, v jakém pořadí jsme je uvedli. Bylo by samozřejmě možno uvažovat i konjunkci nekomutativní, vyjadřující např. časovou závislost výroků („vstal a upadl“ je něco jiného než „upadl a vstal“). Takovou konjunkcí se však klasická logika, kterou zde zobecňujeme na graduální výroky, nezabývá: fuzzy logika s nekomutativní konjunkcí by tak zobecňovala nějakou jinou (např. temporálně orientovanou) logiku. To však není naším cílem – připuštěním graduálních výroků vedle klasických nechceme zároveň měnit význam klasické konjunkce jakožto tvrzení obou výroků nezávisle na jejich pořadí.

- $x * (y * z) = (x * y) * z$  (asociativita)

Zdůvodnění je podobné jako u komutativity: tvrdíme-li tři výroky, tvrdíme je všechny najednou, bez ohledu na to, které dva z nich jsme konjunkcí spojili nejdříve.

- Jestliže  $x \leq x'$ , pak  $x * y \leq x' * y$  (monotonie)

Tato podmínka vyjadřuje, že *zvětšení* pravdivostní hodnoty konjunktu by v žádném případě nemělo *zmenšit* hodnotu konjunkce.

- $x * 1 = x$  (neutralita 1)

Touto podmínkou se stanovuje význam hodnot 1 a 0 jako klasické (plné) pravdivosti a nepravdivosti. Podmínka doslova říká, že plně pravdivé výroky nesnižují pravdivost konjunkce. Je snadným cvičením ukázat, že z této a předchozích podmínek již vyplývá identita  $x * 0 = 0$ , jež vyjadřuje, že konjunkce s plně nepravdivým výrokiem je vždy plně nepravdivá.

---

<sup>17</sup> Formální logika vždy striktně odděluje syntaktickou a sémantickou stránku věci. Proto i ve formální fuzzy logice budeme důsledně odlišovat syntaktickou spojku, čili znak vystupující ve formulích, od její sémantiky, tj. pravdivostní funkce.

- \* je spojitou funkcí (spojitost)

Neformálně řečeno, požadavek spojitosti \* vyjadřuje, že malá (infinitesimální) změna pravdivostní hodnoty jednoho z konjunktů<sup>18</sup> by neměla vést k velké (makroskopické) změně pravdivostní hodnoty celé konjunkce.

V kladení různých podmínek bychom mohli dále pokračovat a vyžadovat například idempotenci konjunkce (neboli podmínku  $x * x = x$ ); ukazuje se však, že by to nebylo účelné. Dosavadní podmínky již omezují pravdivostní funkce natolik, že s nimi lze vybudovat dostatečně zajímavou a bohatou logiku, a další požadavky by již vlastnosti konjunkce příliš svazovaly.<sup>19</sup> Další možné podmínky je tedy vhodné uvažovat pouze jako volitelné, vhodné pro speciální účely.

Funkce splňující první čtyři z výše uvedených podmínek byly již dříve zkoumány v souvislosti s pravděpodobnostními metrickými prostory, a to pod názvem *triangulární normy* (byly totiž použity pro zobecnění trojúhelníkové nerovnosti), či zkráceně *t-normy*. Podmínky na pravdivostní funkci konjunkce lze tedy shrnout tak, že jde o *spojitou t-normu*. Vlastnosti t-norem podrobně podává kniha [17]; zde se omezíme pouze na jejich vlastnosti důležité pro formální fuzzy logiku.

Nejdůležitějšími spojitými t-normami jsou následující tři:<sup>20</sup>

Minimum, zvané též *Gödelova t-norma*:  $x *_G y = \min(x, y)$ ,  
 součin, zvaný též *produktová t-norma*:  $x *_\Pi y = x \cdot y$ ,  
 a tzv. *Łukasiewiczova t-norma*:  $x *_L y = \max(0, x + y - 1)$ .

Ke každé spojitě t-normě \* lze definovat její tzv. *reziduum*  $\Rightarrow_*$ , tj. operaci zadanou předpisem:

$$x \Rightarrow_* y = \sup \{z \mid x * z \leq y\}.$$

Reziduum spojitě t-normy je zjevně největší funkcí takovou, že platí:

$$x * (x \Rightarrow_* y) \leq y. \tag{5}$$

Díky tomu je vhodným kandidátem na pravdivostní funkci fuzzy-logické spojky *implikace* ( $\rightarrow$ ), neboť je největší – z hlediska pravdivostních hodnot tedy nejslabší – funkcí, která ještě přenáší stupeň pravdivosti výroků. Nerovnost (5) lze pak interpretovat jako graduální verzi pravidla usuzování *modus ponens* – nakolik je pravdivé  $x \& (x \rightarrow y)$ , alespoň natolik je pravdivé  $y$ .

Z definice je zřejmé, že  $x \Rightarrow_* y$  má hodnotu 1, právě když  $x \leq y$  (tento fakt bude důležitý i dále). Pro  $x > y$  vycházejí hodnoty reziduí tří základních spojitých t-norem takto:

<sup>18</sup> Přestože ke spojitosti dvouargumentové funkce obecně nestačí spojitost v jednotlivých argumentech, pro funkce splňující dosud uvedené podmínky je to již zaručeno.

<sup>19</sup> Například podmínka idempotence by už konjunkci fixovala jako minimum, v některých kontextech je však vhodnější používat pro konjunkci jiné pravdivostní funkce.

<sup>20</sup> Z těchto tří základních spojitých t-norem lze jistým způsobem (jako tzv. ordinální sumy jejich izomorfních kopií) poskládat všechny ostatní spojitě t-normy (Mostertova–Shieldsova věta).

$$\begin{aligned} x \Rightarrow_G y &= y && \text{(Gödelova implikace)} \\ x \Rightarrow_{\Pi} y &= y / x && \text{(Goguenova čili produktová implikace)} \\ x \Rightarrow_L y &= \min(1 - x + y, 1) && \text{(Łukasiewiczzova implikace)} \end{aligned}$$

V inženýrských fuzzy metodách, kde bývají používány i jiné druhy „implikací“, se reziduová implikace obvykle nazývá *R-implikací* (k dané t-normě).

Pomocí fuzzy implikace je přirozeně definovat fuzzy *negaci* (příslušnou dané t-normě) jako spojku  $\neg$  reprezentovanou pravdivostní funkcí

$$\sim_* x = x \Rightarrow_* 0.$$

Tato identita vyjadřuje chápání negace jako *reductio ad absurdum* (tj. implikování plně nepravdy). Reziduové negace tří základních t-norem vycházejí podle definice takto:

$$\begin{aligned} \sim_L x &= 1 - x && \text{(involutivní negace)} \\ \sim_G x &= \sim_{\Pi} x = \text{sign}(1 - x) && \text{(striktní negace)} \end{aligned}$$

Všimněme si, že negace Gödelovy a produktové t-normy vycházejí stejně a jsou dvouhodnotové: dávají hodnotu 1 pouze pro  $x = 0$ , a jinak hodnotu 0.

Jako poměrně jednoduché cvičení lze dokázat, že pro libovolnou spojitou t-normu  $*$  a její reziduum  $\Rightarrow_*$  platí následující vztahy:

$$\begin{aligned} \min(x, y) &= x * (x \Rightarrow_* y) \\ \max(x, y) &= \min((x \Rightarrow_* y) \Rightarrow_* y, (y \Rightarrow_* x) \Rightarrow_* x) \end{aligned}$$

Operace minima a maxima jsou tedy definovatelné pomocí  $*$  a  $\Rightarrow_*$ . Operace maxima je díky tomu vhodným kandidátem na pravdivostní funkci *disjunkce* ( $\vee$ ). Operace minima má zjevně konjunktivní význam; s výjimkou Gödelovy t-normy jde ale o odlišnou „konjunktci“, než je ta reprezentovaná t-normou  $*$ . Protože ji pomocí dosavadních fuzzy-logických spojek dokážeme beztak vyjádřit, není důvod si ji nezařadit do logického slovníku jako další konjunktivní spojku, vedle původní konjunktce reprezentované t-normou  $*$ . T-normové fuzzy logiky tak budou mít *dvě* konjunktce s různým významem (s výjimkou fuzzy logiky založené na Gödelově minimové t-normě, kde obě konjunktce splývají).

Lze poměrně jednoduše ukázat, že pro každou t-normu platí  $x * y \leq \min(x, y)$  pro všechna  $x, y \in [0, 1]$ . Minimum je tedy největší t-normou, čili z pohledu logické síly nejslabší fuzzy konjunktci. Původní konjunktci  $\&$  reprezentovanou t-normou  $*$  budeme proto nazývat *silnou konjunktci*, kdežto konjunktci reprezentovanou minimem nazveme *slabou* nebo též *minimovou* konjunktci (zkráceně min-konjunktci). Rozdíl mezi jejich významy bude vysvětlen později, na základě jejich různé inferenční role.

Logickou spojku *ekvivalence* ( $\leftrightarrow$ ) lze přirozeně definovat jako konjunktci obou směrů implikace, čili jako spojku s pravdivostní funkcí  $\leftrightarrow_*$  definovanou předpisem:

$$x \leftrightarrow_* y = (x \Rightarrow_* y) * (y \Rightarrow_* x).$$

Otázka, kterou z obou konjunkcí máme v této definici použít, se ukazuje jako bezpředmětná, neboť platí

$$(x \Rightarrow_* y) * (y \Rightarrow_* x) = \min(x \Rightarrow_* y, y \Rightarrow_* x). \quad (6)$$

Definice ekvivalence pomocí obou konjunkcí tudíž splývají.

*Důkaz:* Protože pro  $x, y \in [0, 1]$  vždy nastává alespoň jeden ze vztahů  $x \leq y$  nebo  $y \leq x$ , je vždy hodnota alespoň jedné z implikací  $x \Rightarrow_* y, y \Rightarrow_* x$  rovna 1, a na obou stranách rovnosti (6) tedy vyjde hodnota té druhé z těchto implikací. – QED

Vidíme tedy, že volba konkrétní spojitě t-normy coby pravdivostní funkce pro konjunkci nám již přirozeným způsobem určuje pravdivostní funkce ostatních výrokových spojek. Samotný výběr, kterou spojitou t-normu použít, jsme však nechali na uživateli, neboť pro různé situace se mohou hodit různé konkrétní pravdivostní funkce. Podmínky, které jsme na pravdivostní funkce nakladli, nám však zaručují jejich dobré logické vlastnosti (zejména zachovávání pravdivostních stupňů při usuzování pomocí implikace, srv. (5)).

Ke každé konkrétní spojitě t-normě  $*$  tedy můžeme definovat jazyk a sémantiku její vlastní *výrokové fuzzy logiky*  $L_*$  s pravdivostními funkcemi založenými na této t-normě. Speciálně pro Lukasiewiczovu, Gödelovou a produktovou t-normu ( $*_L, *_G$  a  $*_\Pi$ ) nazýváme jejich logiky *Lukasiewiczovou, Gödelovou a produktovou fuzzy logikou* a značíme je po řadě  $L, G$  a  $\Pi$ . Podáme nyní formální definice syntaxe a sémantiky fuzzy logiky  $L_*$  obecně pro libovolnou spojitou t-normu  $*$ .

**Definice:** *Slovník* výrokové fuzzy logiky  $L_*$  je tvořen binárními výrokovými spojkami  $\&, \rightarrow, \vee, \wedge, \leftrightarrow$ , unární výrokovou spojkou  $\neg$ , výrokovými konstantami  $\underline{0}, \underline{1}$  a výrokovými proměnnými  $p, q, r, \dots$ , z nichž tvoříme formule výrokové fuzzy logiky  $L_*$  obvyklým způsobem (formule budeme označovat velkými písmeny  $A, B, C, \dots$ ). Vzhledem ke vzájemné definovatelnosti pravdivostních funkcí je možno omezit slovník např. na primitivní spojky  $\&, \rightarrow$  a konstantu  $\underline{0}$ , a ostatní spojky považovat za definované následujícími vztahy:

$$A \wedge B \equiv A \& (A \rightarrow B) \quad (7)$$

$$A \vee B \equiv ((A \rightarrow B) \rightarrow B) \& ((B \rightarrow A) \rightarrow A) \quad (8)$$

$$A \leftrightarrow B \equiv (A \rightarrow B) \& (B \rightarrow A) \quad (9)$$

$$\neg A \equiv A \rightarrow \underline{0} \quad (10)$$

$$\underline{1} \equiv \neg \underline{0} \quad (11)$$

Abychom ušetřili závorky, stanovíme pro unární spojky nejvyšší prioritu, a pro spojky  $\rightarrow$  a  $\leftrightarrow$  prioritu nejnižší. Formulí  $\neg p \& q \rightarrow r$  budeme tedy číst jako  $((\neg p) \& q) \rightarrow r$ .

*Standardní*<sup>21</sup> sémantika logiky  $L_*$  je dána způsobem, jakým výše popsané pravdivostní funkce založené na spojitě t-normě  $*$  počítají pravdivostní hodnoty složených výroků. Pro dané ohodnocení  $e$  výrokových proměnných pravdivostními hodnotami z intervalu  $[0, 1]$  tedy definujeme hodnoty  $\|A\|_{e,*}$  složených výroků  $A$  rekurzivně takto:<sup>22</sup>

$$\begin{aligned}\|p\|_{e,*} &= e(p) \\ \|Q\|_{e,*} &= 0 \\ \|A \& B\|_{e,*} &= \|A\|_{e,*} * \|B\|_{e,*} \\ \|A \rightarrow B\|_{e,*} &= \|A\|_{e,*} \Rightarrow_* \|B\|_{e,*}\end{aligned}$$

Pro definované spojky pak již automaticky vychází počítání podle následujících vzorců:

$$\begin{aligned}\|A \wedge B\|_{e,*} &= \min(\|A\|_{e,*}, \|B\|_{e,*}) \\ \|A \vee B\|_{e,*} &= \max(\|A\|_{e,*}, \|B\|_{e,*}) \\ \|A \leftrightarrow B\|_{e,*} &= \|A\|_{e,*} \leftrightarrow_* \|B\|_{e,*} \\ \|\neg A\|_{e,*} &= \sim_* \|A\|_{e,*} \\ \|\perp\|_{e,*} &= 1\end{aligned}\tag{12}$$

*Příklad:* Mějme ohodnocení  $e$  přiřazující výroku  $p$  pravdivostní hodnotu 0,9, výroku  $q$  hodnotu 0,4 a výroku  $r$  hodnotu 0,1. Pak formule  $p \& \neg q \rightarrow r$  bude mít ve standardní sémantice logiky  $L_*$  hodnotu

$$\begin{aligned}\|p \& \neg q \rightarrow r\|_{e,*} &= \\ &= (e(p) * \sim_* e(q)) \Rightarrow_* e(r) = \\ &= (0,9 * \sim_* 0,4) \Rightarrow_* 0,1\end{aligned}$$

Konkrétně pro Łukasiewiczovu t-normu  $*_{\text{L}}$  bude mít tato formule hodnotu

$$(0,9 *_{\text{L}} \sim_{\text{L}} 0,4) \Rightarrow_{\text{L}} 0,1 = (0,9 *_{\text{L}} 0,6) \Rightarrow_{\text{L}} 0,1 = 0,5 \Rightarrow_{\text{L}} 0,1 = 0,6,$$

zatímco pro produktovou t-normu  $*_{\text{II}}$  bude mít hodnotu

$$(0,9 *_{\text{II}} \sim_{\text{II}} 0,4) \Rightarrow_{\text{II}} 0,1 = (0,9 \cdot 0) \Rightarrow_{\text{II}} 0,1 = 0 \Rightarrow_{\text{II}} 0,1 = 1.$$

Vidíme, že oba výsledky se značně liší. Je to způsobeno striktností produktové negace  $\sim_{\text{II}}$ , která činí antecedent implikace zcela nepravdivým, a tedy celou implikaci zcela pravdivou (lze snadno ověřit, že  $0 \Rightarrow_* x$  platí vždy). Stejný výsledek bychom proto dostali i pro minimovou t-normu. Výsledné pravdivostní hodnoty formulí tedy velmi závisí na tom, kterou spojitou t-normu použijeme. Zrovna tak samozřejmě záleží na tom, jaké přesně hodnoty výrokům přiřazuje ohodnocení  $e$ .

<sup>21</sup> Kromě této standardní sémantiky na intervalu  $[0, 1]$  mají t-normové fuzzy logiky i obecnější sémantiky na tzv. reziduovaných svazech. Tyto obecné sémantiky zde nebudeme uvažovat, detaily o nich lze najít například v knihách [14], [21] a zčásti i [19].

<sup>22</sup> Jedná se o přímočaré zobecnění tzv. Tarského podmínek, neboli způsobu, jakým se v klasické logice počítají hodnoty složených výroků pomocí pravdivostních tabulek.

Diskusi, podle jakých kritérií máme vybírat konkrétní t-normu a ohodnocení výroků pro daný sémantický model, však odložíme do oddílu 4. Narozdíl od konkrétního sémantického modelu, *formální* logika (ať už klasická nebo fuzzy) se totiž nikterak nestará o to, jaké jsou konkrétní pravdivostní hodnoty výroků či jak jsme k nim došli, nýbrž studuje logické zákony, které jsou platné pro *všechna* ohodnocení pravdivostními hodnotami. Od konkrétních pravdivostních hodnot tedy odhlíží a generalizuje přes ně: logický zákon je pouze to, co platí *nezávisle* na konkrétních pravdivostních hodnotách.

Skutečně můžeme pozorovat, že některé formule logiky  $L_*$  jsou platné (tj. mají hodnotu 1) bez ohledu na to, jakými čísly jsou ohodnoceny výrokové proměnné. Pro příklad vezměme Łukasiewiczovu logiku a formuli  $\neg\neg p \rightarrow p$ . Její hodnota je pro libovolné ohodnocení  $p$  rovna 1, neboť

$$\begin{aligned} \|\neg\neg p \rightarrow p\|_{e,L} &= & (13) \\ &= \sim_L \sim_L e(p) \Rightarrow_L e(p) = \\ &= (1 - (1 - e(p))) \Rightarrow_L e(p) = \\ &= e(p) \Rightarrow_L e(p) = \\ &= 1. \end{aligned}$$

Tato formule tedy vyjadřuje jeden z *logických zákonů* Łukasiewiczovy fuzzy logiky, neboli její *tautologii*. Obecně definujeme:

**Definice:** Formule  $A$  je tautologií výrokové fuzzy logiky  $L_*$ , zkráceně *\*-tautologií*, jestliže pro každé ohodnocení  $e$  výrokových proměnných platí  $\|A\|_{e,*} = 1$ .

Formule  $\neg\neg p \rightarrow p$  je sice Ł-tautologií, v produktové nebo Gödelově logice však již tautologií není: například pro  $e(p) = 0,5$  je

$$\|\neg\neg p \rightarrow p\|_{e,\Pi} = \sim_{\Pi} \sim_{\Pi} 0,5 \Rightarrow_{\Pi} 0,5 = \sim_{\Pi} 0 \Rightarrow_{\Pi} 0,5 = 1 \Rightarrow_{\Pi} 0,5 = 0,5$$

a stejný výsledek dostaneme i pro G. Jednotlivé t-normové logiky se tedy svými tautologiemi liší. Existují ale i formule, které jsou *\*-tautologiemi* pro *každou* spojitou t-normu  $*$ . Takovou formulí je zřejmě třeba  $p \rightarrow p$ , ale i mnohé další, méně triviální (příklady uvidíme vzápětí); budeme jim říkat *t-tautologie*. Tyto formule vyjadřují logické zákony nezávislé nejen na volbě konkrétního ohodnocení výrokových proměnných (tedy stanovení pravdivostních hodnot elementárních výroků), nýbrž i na konkrétní volbě spojitě t-normy reprezentující konjunkci. Jinými slovy, vyjadřují ty nejjobecnější zákonitosti usuzování o graduálních výroch, které platí v každém konkrétním sémantickém modelu splňujícím výše uvedené přirozené podmínky kladené na logické spojky. Těchto formulí je kupodivu tolik, že má dobrý smysl o nich uvažovat jako o další fuzzy logice, zobecňující všechny předchozí logiky  $L_*$ . Jelikož jde o společný základ všech logik jednotlivých t-norem, nazval ji její objevitel P. Hájek *základní fuzzy logikou* a označil BL (z angl. *basic logic*).<sup>23</sup> Formálně ji definujeme takto:

<sup>23</sup> Názvem *basic logic* je nezávisle označována i jiná neklasická logika (zobecňující logiku intuicionistickou); mimo úzký kontext formální fuzzy logiky je tedy lépe pro jednoznačnost hovořit o *basic fuzzy logic* či *Hájek's basic logic*. Později se ukázalo, že ještě základnější,



**Definice:** Formule  $A$  je tautologií výrokové fuzzy logiky BL, neboli *t-tautologií*, je-li  $*$ -tautologií pro každou spojitou t-normu  $*$ .

Tautologie BL jsou tedy plně pravdivé (tj. mají hodnotu 1) pro každé  $[0, 1]$ -ohodnocení výroků při vyhodnocení pravdivostními funkcemi založenými na libovolné t-normě. Pro ilustraci obecnosti této logiky je vhodné si uvědomit, že na rozdíl od logik jednotlivých t-norem, v případě BL nemá smysl otázka po pravdivostní hodnotě dané formule, ani když známe pravdivostní hodnoty výrokových proměnných: nejde-li totiž o t-tautologii či t-kontradikci (tj. formuli, která má v každém ohodnocení pro každou spojitou t-normu hodnotu 0), bude mít pro různé t-normy obecně *různé* pravdivostní hodnoty. Pouze t-tautologie (a t-kontradikce) mají pravdivostní hodnotu stále stejnou a vyjadřují zcela všeobecně platné zákony fuzzy logiky. Vidíme tedy, že BL abstrahuje nejen od konkrétních pravdivostních hodnot, ale i konkrétních pravdivostních funkcí, a pouze popisuje obecné logické zákonitosti platné pro graduální výroky.

Důležitou otázkou je, zda je možno uvedené fuzzy logiky *axiomatizovat*, tj. stanovit axiomy a odvozovací pravidla, pomocí nichž by bylo možno všechny jejich tautologie mechanicky odvodit. Ukazuje se, že ano: logika BL je úplně axiomatizována například následující sadou axiomů (spolu s definicemi (7)–(11)):<sup>24</sup>

$$(A \rightarrow B) \rightarrow ((B \rightarrow C) \rightarrow (A \rightarrow C)) \quad (14)$$

$$A \& (A \rightarrow B) \rightarrow B \& (B \rightarrow A) \quad (15)$$

$$(A \& B \rightarrow C) \rightarrow (A \rightarrow (B \rightarrow C)) \quad (16)$$

$$(A \rightarrow (B \rightarrow C)) \rightarrow (A \& B \rightarrow C) \quad (17)$$

$$((A \rightarrow B) \rightarrow C) \rightarrow (((B \rightarrow A) \rightarrow C) \rightarrow C) \quad (18)$$

$$\underline{0} \rightarrow A \quad (19)$$

Jediným odvozovacím pravidlem je pravidlo *modus ponens*:

z již odvozených formulí  $A$  a  $A \rightarrow B$  je možno odvodit formuli  $B$ .

Jednotlivé logiky  $L_*$  k těmto axiomům přidávají speciální axiomy platné pro příslušnou t-normu. Kupříkladu Łukasiewiczova logika je axiomatizována výše uvedenými axiomy BL plus axiomem

$$\neg\neg A \rightarrow A.$$

---

a z hlediska formální logiky patrně i důležitější je tzv. *monoidální t-normová logika* MTL všech *zleva* spojitých t-norem (ukázalo se totiž, že pro dobré logické vlastnosti stačí, je-li t-norma spojitá zleva, a že je to požadavek v jistém smyslu minimální). Protože se však BL od MTL ve svých rysech ani síle zásadně neliší, budeme se zde držet jednoduššího výkladu pro BL.

<sup>24</sup> Původní axiomatika BL obsahovala ještě dva další axiomy, K. Chvalovský [9] však nedávno pomocí počítačových programů ukázal, že byly redundantní a že zde uvedené axiomy jsou již nezávislé. Z hlediska uživatele logiky je však mnohem zajímavější znát nejdůležitější *teorémy* (tj. dokazatelné formule) dané logiky; ty uvedeme níže v tomto oddílu.

Gödelova logika rozšiřuje BL o axiom

$$A \rightarrow A \& A$$

a produktová fuzzy logika o axiomy

$$\neg(A \wedge \neg A)$$

$$\neg\neg C \rightarrow ((A \& C \rightarrow B \& C) \rightarrow (A \rightarrow B)).$$

Dokažme, že tyto axiomatické systémy jsou *korektní*, tj. že jejich axiomy jsou tautologiemi a že pravidlo modus ponens přenáší plnou pravdivost (z čehož lze již vyvodit, že generují *pouze* tautologie).

*Důkaz:* Ukažme nejprve korektnost pravidla modus ponens pro libovolnou spojitou t-normu \*. Jsou-li formule  $A$  a  $A \rightarrow B$  plně pravdivé v kterémkoli ohodnocení  $e$ , znamená to, že  $\|A\|_{e,*} = 1$ , jakož i  $\|A \rightarrow B\|_{e,*} = \|A\|_{e,*} \Rightarrow_* \|B\|_{e,*} = 1$ . Vzpomeňme však, že hodnota  $\|A\|_{e,*} \Rightarrow_* \|B\|_{e,*}$  je rovna 1, právě když  $\|A\|_{e,*} \leq \|B\|_{e,*}$ ; tedy i  $\|B\|_{e,*} = 1$ , což bylo dokázati.

Důkazy korektnosti některých axiomů jsou triviální – například korektnost axiomu (19) vyplývá přímo z definice rezidua t-normy. Axiom (15) je zase podle (7) ekvivalentní komutativitě minimové konjunkce  $\wedge$ , jež plyne dle (12) z komutativity minima. Důkaz korektnosti axiomu  $\neg\neg A \rightarrow A$  v Łukasiewiczově logice jsme viděli výše u (13). Speciální axiom Gödelovy logiky v ní platí triviálně proto, že její silná konjunkce je minimem, axiom tedy vyjadřuje, že  $\|A\|_{e,G} \leq \min(\|A\|_{e,G}, \|A\|_{e,G})$ . Důkazy ostatních axiomů z definičních vlastností t-norem a jejich reziduí jsou jednoduchým cvičením. – QED

Důkazy *úplnosti* těchto axiomatik vůči standardní t-normové sémantice (čili tvrzení, že generují *všechny* tautologie) jsou značně obtížné, proto je nebudeme předvádět; lze ale ukázat, že věta o úplnosti vůči standardní sémantice platí pro všechny uvedené výrokové fuzzy logiky. Díky větám o korektnosti a úplnosti splývají pro tyto logiky pojmy tautologičnosti a dokazatelnosti. Vedle těchto vět je známa i řada dalších metamatematických vlastností uvedených fuzzy logik – např. věta o substituci, modifikovaná věta o dedukci, aritmetická i výpočetní složitost těchto logik, definovatelnost funkcí formulemi dané logiky aj. (pro úvodní informaci lze odkázat na knihu [14], další výsledky byly publikovány v desítkách článků po roce 1998). Obrátme se však od jejich vlastností metamatematických či meta-logických (kdy usuzujeme o těchto logikách) k jejich vlastnostem logickým, tj. vlastnostem z hlediska usuzování *pomocí* těchto logik (o graduálních vlastnostech). Ukažme si, které z nejdůležitějších logických zákonů jsou ve fuzzy logice platné, a které nikoli.

Jak již bylo naznačeno, mnoho logických zákonů fuzzy logiky je shodných s logikou klasickou. Platí to zčásti i pro vlastní deduktivní aparát daný implikací a konjunkcí. Stejně jako v klasické logice je implikace tranzitivní a konjunkce vyjadřuje hromadění předpokladů vnořených implikací (srv. axiomy (14) a (16)–(17)). Platí

také graduální obdoba klasického zákona, že z nepravdivého tvrzení lze odvodit cokoli (*ex falso quodlibet*). Toto tvrzení lze formalizovat jako

$$\neg A \rightarrow (A \rightarrow B), \quad (20)$$

čili vyjádřeno v přirozeném jazyce, „je-li  $A$  nepravdivé, pak  $A$  implikuje libovolné  $B$ “. Všimněme si, že pomocí axiomu (17) lze tento teorém dále upravit na tvar

$$\neg A \& A \rightarrow B,$$

tedy tvrzení, že ze sporu ( $\neg A \& A$ ) plyne cokoliv (*ex contradictione quodlibet*).

Důkazy jednotlivých teorémů BL a ostatních fuzzy logik z axiomů jsou zdoluhavé a nudné, proto je nebudeme předvádět. Díky větě o úplnosti je lze ostatně ověřovat i sémanticky, z definičních vlastností výrokových spojek, což je ve většině případů jednoduché cvičení. Rozeberme raději podrobně, co graduální forma (20) zákona *ex falso quodlibet* ve fuzzy logice sémanticky vyjadřuje.

Formule (20) je dokazatelná, a tedy vždy plně pravdivá:  $\|\neg A \rightarrow (A \rightarrow B)\|_{e,*} = 1$  pro libovolné ohodnocení  $e$  a libovolnou spojitou t-normu  $*$ . To podle sémantiky implikace znamená, že  $\|\neg A\|_{e,*} \leq \|A \rightarrow B\|_{e,*}$ , čili  $\sim * \|A\|_{e,*} \leq \|A\|_{e,*} \Rightarrow * \|B\|_{e,*}$ . Graduální zákon *ex falso quodlibet* tedy říká, že stupeň pravdivosti implikace (daný v konkrétním modelu reziduem nějaké spojitě t-normy) je alespoň takový, jako stupeň nepravdivosti (tj. reziduová negace stupně pravdivosti) jejího antecedentu. Stručněji řečeno, nakolik je pravdivá negace  $A$ , natolik je pravdivá i každá implikace  $A \rightarrow B$ ; tedy neformálně, *nakolik* je  $A$  nepravdivé, *natolik* z něj lze vyvodit cokoli. Takovýto graduální význam mají i ostatní logické zákony fuzzy logiky.

Pokračujme vlastnostmi silné konjunkce: stejně jako v klasické logice je i v BL (a tedy tím spíše v silnějších fuzzy logikách jednotlivých t-norem) silná konjunkce komutativní, asociativní, silnější než jednotlivé konjunkty, a výrokové konstanty  $\underline{0}$  a  $\underline{1}$  se vůči ní chovají jako anihilující a neutrální prvek:

$$\begin{aligned} A \& B &\leftrightarrow B \& A \\ (A \& B) \& C &\leftrightarrow A \& (B \& C) \\ A \& B &\rightarrow A \\ A \& \underline{1} &\leftrightarrow A \\ A \& \underline{0} &\leftrightarrow \underline{0} \end{aligned} \quad (21)$$

Tyto zákony nejsou nijak překvapivé, neboť pouze odrážejí vlastnosti t-norem, které jsme přijali jako sémantiku silné konjunkce (musí tedy vskutku platit v každém sémantickém modelu BL). Tyto vlastnosti má i min-konjunkce (neboť i minimum je jednou z t-norem):

$$\begin{aligned} A \wedge B &\leftrightarrow B \wedge A \\ (A \wedge B) \wedge C &\leftrightarrow A \wedge (B \wedge C) \\ A \wedge B &\rightarrow A \\ A \wedge \underline{1} &\leftrightarrow A \\ A \wedge \underline{0} &\leftrightarrow \underline{0} \end{aligned}$$

Max-disjunkce  $\vee$  splňuje zákony duální k uvedeným zákonům o min-konjunkci:

$$\begin{aligned}
A \vee B &\leftrightarrow B \vee A \\
(A \vee B) \vee C &\leftrightarrow A \vee (B \vee C) \\
A &\rightarrow A \vee B \\
A \vee \underline{1} &\leftrightarrow \underline{1} \\
A \vee \underline{0} &\leftrightarrow A
\end{aligned}$$

Pro min-konjunkci a max-disjunkci platí i následující distribuční a de Morganovy zákony:

$$\begin{aligned}
A \& (B \wedge C) &\leftrightarrow (A \& B) \wedge (A \& C) \\
A \& (B \vee C) &\leftrightarrow (A \& B) \vee (A \& C) \\
A \wedge (B \vee C) &\leftrightarrow (A \wedge B) \vee (A \wedge C) \\
A \vee (B \wedge C) &\leftrightarrow (A \vee B) \wedge (A \vee C) \\
\neg(A \wedge B) &\leftrightarrow \neg A \vee \neg B \\
\neg(A \vee B) &\leftrightarrow \neg A \wedge \neg B
\end{aligned}$$

Obě konjunkce i disjunkce jsou rovněž monotónní vůči implikaci:

$$\begin{aligned}
(A \rightarrow B) &\rightarrow (A \& C \rightarrow B \& C) \\
(A \rightarrow B) &\rightarrow (A \wedge C \rightarrow B \wedge C) \\
(A \rightarrow B) &\rightarrow (A \vee C \rightarrow B \vee C)
\end{aligned}$$

Pro implikaci a ekvivalenci dále platí následující vztahy:

$$\begin{aligned}
A &\rightarrow A \\
A &\rightarrow (B \rightarrow A) \\
(\underline{1} \rightarrow A) &\leftrightarrow A \\
(A \rightarrow B) \vee (B \rightarrow A) & \\
A &\leftrightarrow A \\
(A \leftrightarrow B) &\leftrightarrow (B \leftrightarrow A) \\
(A \leftrightarrow B) \& (B \leftrightarrow C) &\rightarrow (A \leftrightarrow C) \tag{22}
\end{aligned}$$

Ve všech těchto logických zákonech se fuzzy logiky shodují s klasickou logikou, graduálnost výroků zde tedy nehraje podstatnou roli. Značné odlišnosti od klasické logiky jsou však v pravidlech usuzování pro negaci. Jak jsme již viděli, mimo  $\underline{1}$  ve fuzzy logikách obecně neplatí zákon dvojité negace  $\neg\neg A \leftrightarrow A$ , nýbrž jen jedna z jeho implikací:

$$A \rightarrow \neg\neg A$$

Ve fuzzy logice proto musíme obecně rozlišovat mezi výroky  $A$  a  $\neg\neg A$ , jež nejsou ekvivalentní a liší se svým významem: lze snadno ověřit, že v Gödelově a produktové logice je pravdivostní hodnota výroku  $\neg\neg A$  rovna 1, právě když má  $A$  hodnotu nenulovou; jinak je hodnota  $\neg\neg A$  rovna 0. Výrok  $\neg\neg A$  tedy v G a  $\Pi$  ostrý a znamená, že  $A$  je pravdivé „alespoň trochu“. Naproti tomu lze snadno sémanticky ověřit, že trojitá negace  $\neg\neg\neg A$  je v G i  $\Pi$  již ekvivalentní negaci jednoduché:  $\neg\neg\neg A \leftrightarrow \neg A$ . Ve fuzzy

logikách také sice neplatí zákon vyloučení třetího  $A \vee \neg A$ , v G a  $\Pi$  však platí alespoň  $\neg A \vee \neg\neg A$ , tj. zákon vyloučení třetího pro *negativní* formule (neboť ty jsou v G a  $\Pi$  již ostré).

Přibližně lze říci, že ve fuzzy logikách je třeba v souvislosti s negací usuzovat v některých ohledech „intuicionisticky“, nikoli klasicky.<sup>25</sup> Projevuje se to například u následujících vztahů, v nichž narozdíl od klasické logiky (ale shodně s intuicionistickou) obecně neplatí ekvivalence, nýbrž jen uvedené implikace:

$$\begin{aligned} A \vee B &\rightarrow \neg(\neg A \wedge \neg B) \\ A \wedge B &\rightarrow \neg(\neg A \vee \neg B) \end{aligned}$$

Rovněž zákony kontrapozice, využívané k nepřímým důkazům, platí pouze jedním směrem:

$$\begin{aligned} (A \rightarrow B) &\rightarrow (\neg B \rightarrow \neg A) \\ (A \rightarrow \neg B) &\rightarrow (B \rightarrow \neg A) \end{aligned}$$

Výjimkou z tohoto neklasického chování negace je Łukasiewiczova logika, v níž zákon dvojité negace ( $\neg\neg A \leftrightarrow A$ ) platí. V tomto a několika dalších rysech je Łukasiewiczova logika mezi ostatními fuzzy logikami výjimečná (a proto i často používaná v aplikacích). Lze si například všimnout, že všechny spojky Łukasiewiczovy logiky mají spojitě pravdivostní funkce, zatímco pravdivostní funkce implikace či negace ostatních t-normových logik jsou alespoň v některých bodech nespojitě.

Rovněž Gödelova logika se výrazně odlišuje od ostatních fuzzy logik, díky idempotentnosti své konjunkce. Jelikož v ní obě konjunkce splývají, nevyskytuje se v ní – pro ostatní fuzzy logiky typické – „počítání identických konjunktů“, jež lze ilustrovat na následujících logických zákonech. V klasické i v Gödelově logice platí jako logický zákon

$$(A \rightarrow B) \& (A \rightarrow C) \rightarrow (A \rightarrow B \& C) \quad (23)$$

To znamená, že máme-li vyvodit konjunkci  $B \& C$  z předpokladu  $A$ , stačí z  $A$  vyvodit jednotlivé konjunktivy, neboli ověřit obě implikace  $A \rightarrow B$  a  $A \rightarrow C$ . V ostatních t-normových logikách však tento zákon *neplatí*. Předpoklad  $A$  tu je totiž použit dvakrát (jednou v  $A \rightarrow B$  a podruhé v  $A \rightarrow C$ ), proto i pro závěr  $B \& C$  je nutno jej vzít dvakrát. Platí tedy pouze

$$(A \rightarrow B) \& (A \rightarrow C) \rightarrow (A \& A \rightarrow B \& C). \quad (24)$$

Tato nutnost „správného počtu identických předpokladů“ je způsobena právě neidempotentností silné konjunkce, která neumožňuje zredukovat předpoklad  $A \& A$  na

<sup>25</sup> Příbuznost fuzzy logik s intuicionistickou logikou lze přesně popsat v rámci tzv. substrukturálních logik. Gödelovu logiku lze dokonce přímo axiomatizovat jako rozšíření intuicionistické logiky o axiom  $(A \rightarrow B) \vee (B \rightarrow A)$ . Tento fuzzy-logický zákon (zvaný prelinearita) platí ve všech t-normových logikách a zčásti „zklasičuje“ intuicionistické chování negace – například de Morganovy zákony platí ve fuzzy logikách plně, zatímco intuicionisticky neplatí implikace  $\neg(A \wedge B) \rightarrow \neg A \vee \neg B$ .

jediné  $A$ : protože  $A \& A$  má obecně menší pravdivostní hodnotu než  $A$ , je předpoklad  $A \& A$  obecně silnější než  $A$ . Klasický zákon (23) platí pouze pro *minimumovou* konjunkci nebo pro ostré výroky  $A$  (proto je také platný v Gödelově a klasické logice).

Rozdíl mezi oběma konjunkcemi, který se v Gödelově logice minimumové t-normy a v klasické logice ostrých výroků stírá, je ve fuzzy logice velice důležitý. V oddílu 4 kupříkladu uvidíme, že bez neidempotentní silné konjunkce by nefungovalo řešení paradoxu hromady, jež je jednou ze základních motivací pro vybudování logiky graduálních vlastností. Objasněme si proto podrobně, jak se obě konjunkce liší a co vlastně znamenají. Již jsme zmínili, že silná konjunkce je silnější než slabá, tj. že

$$A \& B \rightarrow A \wedge B$$

(vedle přímého důkazu ve formálním kalkulu to vyplývá i z věty o úplnosti a faktu, že minimum je největší t-normou). Kdy však máme použít silnou a kdy slabou konjunkci?

Významový rozdíl mezi konjunkcemi se obzvláště zřetelně ukazuje na logických zákonech vyjadřujících, co lze z obou konjunkcí odvodit. Pro silnou konjunkci platí:

$$(A \& B \rightarrow C) \leftrightarrow (A \rightarrow (B \rightarrow C)).$$

Ze silné konjunkce  $A \& B$  lze tedy vyvodit právě ty výroky  $C$ , k jejichž vyvození potřebujeme *obě* premisy  $A$ ,  $B$  (k získání závěru  $C$  na pravé straně totiž potřebujeme nejprve vyvodit  $B \rightarrow C$  z  $A$  a poté  $C$  z  $B$ , použijeme tedy jak předpoklad  $A$ , tak  $B$ ). Pro slabou konjunkci naopak platí

$$(A \wedge B \rightarrow C) \leftrightarrow ((A \rightarrow C) \vee (B \rightarrow C)).$$

Výrok je tedy vyvoditelný ze slabé konjunkce  $A \wedge B$ , právě když jej lze vyvodit ze samotného  $A$  nebo ze samotného  $B$ , čili z *libovolného* z konjunkťů. Rozdíl mezi  $\&$  a  $\wedge$  tedy spočívá v tom, že silná konjunkce vyjadřuje použitelnost *obou* konjunkťů najednou, zatímco slabá konjunkce vyjadřuje použitelnost *libovolného* z konjunkťů pro vyvozování důsledků.

Tento rozdíl je obzvláště výrazný v případě zákona sporu. U filosofů vágnosti se lze pravidelně (např. i ve vlivné knize [24], str. 118) setkat s tvrzením, že „ve fuzzy logice zákon sporu neplatí“. To je však pravda pouze pro slabou konjunkci: formule  $A \wedge \neg A$  skutečně nemusí mít hodnotu 0, takže zákon sporu v podobě  $\neg(A \wedge \neg A)$  není t-tautologií. To je ovšem naprosto v pořádku: k vyvození sporu jsou přece potřeba *obě* dvě formule  $A$  a  $\neg A$ , zatímco jedna každá z nich sama o sobě sporná není. Sporem je tedy pouze *silná* konjunkce  $A$  a  $\neg A$ , nikoli slabá. A vskutku zákon sporu v podobě  $\neg(A \& \neg A)$  ve fuzzy logice platí, a je jen přirozené, že v podobě  $\neg(A \wedge \neg A)$  obecně neplatí.

Podobně často uváděná námitka (např. opět ve [24], str. 124), že ve fuzzy logice neplatí  $A \wedge (A \rightarrow B) \rightarrow B$ , se zakládá na mylném chápání role min-konjunkce: protože  $B$  nelze vyvodit ani ze samotného  $A$  ani ze samotného  $A \rightarrow B$ , je jen přirozené, že  $A \wedge (A \rightarrow B) \rightarrow B$  není (mimo G) logickým zákonem. Jelikož k vyvození  $B$  potřebujeme *oba* předpoklady  $A$  a  $A \rightarrow B$ , je zde namísto silné konjunkce, a zákon  $A \& (A \rightarrow B) \rightarrow B$  ve fuzzy logice skutečně platí (srv. (5)).

Rozdíl mezi konjunkcemi se může zdát smysluprázdným, jsou-li oba konjunkty stejné. Hraje však významnou roli i v tomto případě: výrok  $A \& A$  je právě proto silnější než  $A \wedge A$  (čili  $A$ ), že předpoklad  $A$  lze při silné konjunkci  $A \& A$  použít *dvakrát*, jak jsme to viděli například ve (24). Fuzzy logika tak mj. zachycuje fakt, že *vícenásobné* použití nedokonale pravdivé premisy v argumentu *snižuje* pravdivost závěru. Konkrétní aplikaci tohoto jevu uvidíme při řešení paradoxu hromady v oddílu 4.

Ukázky rozdílů mezi klasickou a fuzzy logikou na tomto místě ukončíme. Stručně lze shrnout tak, že fuzzy logiky značnou část logických zákonů s klasickou logikou sdílejí, liší se ale od ní zejména v chování negace (s výjimkou Łukasiewiczovy logiky) a silné konjunkce (s výjimkou Gödelovy logiky).

### 3 Rozšíření výrokové fuzzy logiky

V předchozím oddílu jsme popsali základní aparát výrokových fuzzy logik založených na spojitých t-normách. V tomto oddíle naznačíme, jak lze tento aparát rozšiřovat několika směry.

Často používaným rozšířením t-normových fuzzy logik je přidání unární spojky  $\Delta$  (Delta), zvané též *Baazův operátor*.<sup>26</sup> Tato spojka vyjadřuje plnou pravdivost výroku. Její standardní sémantika je dána pravdivostní funkcí definovanou takto:

$$\begin{aligned}\delta(x) &= 1, \text{ jestliže } x = 1; \\ \delta(x) &= 0, \text{ jestliže } x < 1.\end{aligned}$$

Lze ukázat, že spojka  $\Delta$  není v logických systémech diskutovaných v předchozím oddíle definovatelná, a chceme-li jí disponovat, musíme ji do logiky přidat jako novou primitivní spojku. T-normovou fuzzy logiku  $L$  obohacenou o operátor  $\Delta$  pak značíme  $L_\Delta$ . Logiku  $BL_\Delta$  lze axiomatizovat například tak, že k axiomatice logiky BL přidáme axiomy

$$\begin{aligned}\Delta A \vee \neg \Delta A \\ \Delta(A \vee B) \rightarrow \Delta A \vee \Delta B \\ \Delta A \rightarrow A \\ \Delta A \rightarrow \Delta \Delta A \\ \Delta(A \rightarrow B) \rightarrow (\Delta A \rightarrow \Delta B)\end{aligned}$$

a tzv. pravidlo  $\Delta$ -necesitace:

$$\text{z již odvozené formule } A \text{ je možno odvodit } \Delta A. \quad (25)$$

Stejná sada axiomů a pravidel je použitelná i pro úplnou a korektní axiomatizaci operátoru  $\Delta$  v logikách jednotlivých spojitých t-norem (viz [14], kap. 2.4).

Pomocí spojky  $\Delta$  lze vyjádřit mnoho věcí, jež v logikách bez  $\Delta$  obecně vyjádřitelné nejsou. Kupříkladu *ostrost* výroku  $A$  je vyjádřena formulí  $\Delta(A \vee \neg A)$ : jednoduchým

<sup>26</sup> Matthias Baaz, vídeňský logik; zabývá se především Gödelovou logikou. Spojku  $\Delta$  zavedl a axiomatizoval právě pro Gödelovu logiku, později byla zobecněna i pro jiné fuzzy logiky.

výpočtem se lze bez ohledu na použitou t-normu přesvědčit, že  $\|\Delta(A \vee \neg A)\|_{e,*} = 1$ , právě když  $\|A\|_{e,*} \in \{0, 1\}$ , zatímco pro  $\|A\|_{e,*} \notin \{0, 1\}$  je  $\|\Delta(A \vee \neg A)\|_{e,*} = 0$ . Pomocí  $\Delta$  lze rovněž vyjádřit rovnost pravdivostních hodnot formulí  $A$  a  $B$  jako  $\Delta(A \leftrightarrow B)$  a podobně pro ostré či neostré nerovnosti atp. Výroky tvaru  $\Delta A$  jsou samy ostré, takže se chovají klasicky – platí pro ně např. zákon dvojité negace  $\Delta A \leftrightarrow \neg\neg\Delta A$ , a jejich vícenásobné konjunkce lze kontrahovat, neboť platí  $\Delta A \& \Delta A \leftrightarrow \Delta A$ . Operátor  $\Delta$  tak umožňuje v rámci fuzzy logiky hovořit nejen o výrocích graduálních, ale také ostrých. Klasickou logiku pak lze chápat jako důležitý *fragment* fuzzy logiky  $BL_{\Delta}$ , který se zabývá pouze formullemi, v nichž je na všechny výrokové proměnné aplikována  $\Delta$  (takže jsou ostré). Logika  $BL_{\Delta}$  je tedy *společným* logickým rámcem pro ostré i graduální výroky.

Další často přidávanou spojkou je *involutivní negace* (krátce *involuce*), značená obvykle  $\sim$ , se standardní pravdivostní funkcí  $f_{\sim}(x) = 1 - x$ . V Łukasiewiczově logice je involucí přímo reziduová negace  $\neg$ , v jiných t-normových logikách však involuce není definovatelná, a chceme-li ji mít v jazyce, musí být přidána jako nová primitivní spojka (systém pak obsahuje dvě různé negace). T-normová logika  $L$  obohacená o involutivní negaci se obvykle značí  $L_{\sim}$ . Axiomatizaci logik s involucí již vynecháme (podrobnosti lze najít v [12]), všimneme si jen několika jejich rysů.

Involuce zřejmě splňuje zákon dvojité negace  $A \leftrightarrow \sim\sim A$  i de Morganovy zákony  $\sim(A \wedge B) \leftrightarrow \sim A \vee \sim B$ ,  $\sim(A \vee B) \leftrightarrow \sim A \wedge \sim B$ . V logikách se striktní (dvouhodnotovou) negací, jakou mají  $G$  či  $\Pi$ , je již pomocí involuce definovatelný operátor  $\Delta$ , neboť lze snadno ověřit, že ve standardní sémantice platí

$$\Delta A \leftrightarrow \neg \sim A.$$

Involutivní negace také umožňuje definovat tzv. *silnou disjunkci*:

$$A \underline{\vee} B \equiv \sim(\sim A \& \sim B)$$

Logiky s involucí tak vedle dvou konjunkcí oplývají i dvěma disjunkcemi, které se liší obdobně jako silná a slabá konjunkce (k nimž jsou duální prostřednictvím de Morganových zákonů). Lze si například všimnout, že zákon vyloučení třetího ve fuzzy logikách neplatí se slabou disjunkcí ( $A \vee \neg A$  není tautologií), platí však se silnou ( $A \underline{\vee} \neg A$  je tautologií). Důvody tohoto rozdílu jsou obdobné jako u zákona sporu (viz konec oddílu 2).

Standardní pravdivostní funkce  $\oplus_*$  silné disjunkce závisí na spojitě t-normě  $*$  použité pro konjunkci. Například v Łukasiewiczově logice je pravdivostní funkcí silné disjunkce operace omezeného sčítání  $x \oplus_L y = \min(x + y, 1)$ , jak se lze přesvědčit jednoduchým výpočtem:

$$\begin{aligned} \|A \underline{\vee} B\|_{e,L} &= \|\neg(\neg A \& \neg B)\|_{e,L} = \sim_L(\sim_L \|A\|_{e,L} *_L \sim_L \|B\|_{e,L}) = \\ &= 1 - \max((1 - \|A\|_{e,L}) + (1 - \|B\|_{e,L}) - 1, 0) = \\ &= 1 - \max(1 - (\|A\|_{e,L} + \|B\|_{e,L}), 0) = \\ &= \min(\|A\|_{e,L} + \|B\|_{e,L}, 1) \end{aligned}$$



Podobně pro produktovou t-normu vychází pravdivostní funkce silné disjunkce jako  $x \oplus_{\Pi} y = x + y - x \cdot y$ . V případě minimové t-normy je silnou disjunkcí maximum; v Gödelově logice tedy splývají jak obě konjunkce, tak obě disjunkce. Pravdivostní funkci silné disjunkce příslušné k t-normě  $*$  se též říká *t-konorma* (nebo také *S-norma*) duální k  $*$ . T-konormy jsou stejně jako t-normy komutativní, asociativní a monotonní operace na  $[0, 1]$ , na rozdíl od t-norem však mají za neutrální prvek 0 a za absorbující prvek 1.

Jiným častým rozšířením je přidání *pravdivostních konstant* pro další pravdivostní hodnoty vedle 0 a 1. Toto rozšíření jde poněkud proti snaze formální logiky od konkrétních pravdivostních hodnot abstrahovat; je ale třeba připustit, že některé pravdivostní hodnoty jsou v některých formálních fuzzy logikách beztak definovatelné (např. v  $L_{\Delta}$  lze pravdivost  $A$  ve stupni 0,5 vyjádřit formulí  $\Delta(A \leftrightarrow \neg A)$ ), a v některých kontextech (zejména aplikačních) může být výhodné mít možnost o pravdivostních hodnotách hovořit přímo pomocí logických symbolů. Rozšíření o pravdivostní konstanty se provede tak, že pro každou vybranou pravdivostní hodnotu  $r \in [0, 1]$  přidáme do jazyka t-normové fuzzy logiky  $L$  výrokovou konstantu  $\underline{r}$  a axiomy zaručující její správnou sémantickou hodnotu. Chceme-li například rozšířit Łukasiewiczovu logiku o konstantu  $\underline{0,5}$ , přidáme spolu s ní axiom  $\underline{0,5} \leftrightarrow \neg \underline{0,5}$ , který už její pravdivostní hodnotu 0,5 ve standardní sémantice vynutí.

Lze přidat i větší množství konstant najednou, například pro všechna racionální či reálná čísla  $r$  z intervalu  $[0, 1]$ . Standardní sémantika výrokové konstanty  $\underline{r}$  zastupující pravdivostní hodnotu  $r \in [0, 1]$  je definována přímočaře jako  $\|\underline{r}\|_{e,*} = r$ . Axiomatiku je pak třeba obohatit o tzv. *účetní axiomy* zaručující správnou „aritmetiku“ těchto nových konstant:

$$\begin{aligned} \underline{r} \& \underline{s} &\leftrightarrow \underline{r * s} \\ (\underline{r} \rightarrow \underline{s}) &\leftrightarrow \underline{r \Rightarrow_* s} \end{aligned}$$

pro všechna  $r, s \in [0, 1]$ , pro něž byly přidány pravdivostní konstanty (jde tedy o nekonečné množství jednotlivých axiomů, např. v produktové logice bychom měli axiomy  $\underline{0,9} \& \underline{0,9} \leftrightarrow \underline{0,81}$ ,  $(\underline{0,4} \rightarrow \underline{0,5}) \leftrightarrow \underline{1}$ , atd.).

Přidání racionálních či reálných pravdivostních konstant umožňuje formulaci zcela odlišného typu věty o úplnosti systému, nazývané *pavelkovská úplnost* (jedná se totiž o analogii Pavelkovy věty o úplnosti v jeho systému zmíněném v oddílu 1). Tato úplnost se opírá nejen o částečnou pravdivost výroků v sémantice, ale i o částečnou *dokazatelnost* teorémů v syntaxi. Věta o pavelkovské úplnosti pak tvrdí, že infimum stupňů částečné pravdivosti výroku v modelech je rovno suprému stupňů jeho dokazatelnosti. Logickým systémům, které takovou větu splňují, říkáme *pavelkovské*. Protože se však jedná o značně odlišný druh fuzzy logik, než o jakých zde pojednáváme, nebudeme se jim dále věnovat a pouze odkážeme na [21] či [14].

Důležitou fuzzy logikou s velkou vyjadřovací silou je logika  $\mathbb{L}\Pi$  (případně  $\mathbb{L}\Pi\frac{1}{2}$  s přidanou pravdivostní konstantou  $\underline{0,5}$ ), která obsahuje Łukasiewiczovské i produktové výrokové spojky. Jejich kombinací jsou už definovatelné nejenom gödelovské spojky a operátor  $\Delta$ , ale i základní aritmetické operace, a také nekonečně mnoho t-normových konjunkcí a jejich reziduí. Tato logika tak zahrnuje velké množství ostatních systémů

fuzzy logiky a poskytuje značnou vládu nad pravdivostními hodnotami prostřednictvím pravdivostních funkcí definovatelných spojek. Má přitom stále dobré metamatematické vlastnosti a lze ji axiomatizovat poměrně jednoduše – k axiomům  $\mathbb{L}$  a  $\Pi_{\Delta}$  se přidají pouze axiomy vyjadřující koincidence plné pravdivosti lukasiewiczovské a produktové implikace, distributivitu násobení (čili produktové konjunkce) vůči rozdílu (vyjádřitelnému lukasiewiczovskými spojkami) a fakt, že involutivní negace je slabší než striktní. Tato logika tak může sloužit jako společný základní systém pro práci s většinou prakticky potřebných operací. Detaily o logikách  $\mathbb{L}\Pi$  a  $\mathbb{L}\Pi\frac{1}{2}$  lze najít v pracích [11] a [10].

Kvalitativním skokem při rozšiřování výrazových možností fuzzy logiky je přechod od výrokové k prvořádové fuzzy logice. Pomocí výrokových fuzzy logik lze sice popsat jednoduché chování graduálních výroků a použít je k řešení některých logických paradoxů (viz oddíl 4). Chceme-li však modelovat složitější situace či budovat matematiku obsahující graduální pojmy, je nutno mít k dispozici prostředky pro graduální kvantifikaci fuzzy výroků a práci s predikáty a objektovými proměnnými; ty nám poskytuje právě prvořádová (neboli predikátová) fuzzy logika. Rozšíření fuzzy logiky na prvořádovou nyní stručně popíšeme; předpokládáme přitom zběžnou znalost klasické predikátové logiky.

Stejně jako v klasické logice prvního řádu, jazyk predikátové fuzzy logiky rozlišuje objekty (prvky nějakého ostrého univerza) a jejich vlastnosti či vztahy (podmnožiny a relace na tomto univerzu). Proměnné pro objekty (zvané též *individuové proměnné*) jsou značeny malými písmeny  $x, y, \dots$  a konstanty pro vlastnosti či vztahy (zvané též *predikáty*) jsou označovány velkými písmeny  $P, Q, \dots$ . Každý predikát má pevně přiřazenu svoji *aritu* určující počet jeho argumentů. *Atomické formule* mají tvar  $Px_1 \dots x_n$ , kde  $P$  je  $n$ -ární predikát a  $x_1, \dots, x_n$  jsou individuové proměnné. *Formule* je možno vytvářet z atomických formulí obvyklým způsobem pomocí výrokových spojek a kvantifikátorů  $\exists$  a  $\forall$  (je-li  $\varphi$  formule a  $x$  individuová proměnná, pak  $(\exists x)\varphi$  a  $(\forall x)\varphi$  jsou také formule). Až na přítomnost dvou konjunkcí (či dalších přidaných výrokových spojek fuzzy logiky, např.  $\Delta$ ) je tedy jazyk prvořádové fuzzy logiky stejný jako jazyk klasické prvořádové logiky.

Stejná je rovněž sémantická interpretace individuových proměnných jako prvků nějaké pevně určené neprázdné ostré množiny  $U$  (zvané univerzum modelu). V čem se obě logiky liší, je sémantika predikátů. V klasické logice jsou unární predikáty (zastupující *ostré* vlastnosti objektů) reprezentovány *ostrými* podmnožinami univerza  $U$  a  $n$ -ární predikáty (zastupující ostré  $n$ -ární vztahy mezi objekty, pro  $n > 1$ ) ostrými  $n$ -árními relacemi na  $U$  (tedy podmnožinami  $U^n$ ). Naproti tomu ve fuzzy logice jsou unární predikáty (zastupující *graduální* vlastnosti objektů) reprezentovány *fuzzy* podmnožinami univerza  $U$  (tj. funkcemi z  $U$  do  $[0, 1]$ ) a  $n$ -ární predikáty (zastupující graduální  $n$ -ární vztahy mezi objekty)  $n$ -árními fuzzy relacemi na  $U$  (tj. fuzzy podmnožinami  $U^n$ , neboli funkcemi z  $U^n$  do  $[0, 1]$ ).

**Definice:** Standardní *sémantickou interpretací* jazyka prvořádové fuzzy logiky sestávajícího z predikátů  $P_1, \dots, P_k$  s aritami po řadě  $n_1, \dots, n_k$  (též zvanou *model* či *struktura* pro tento jazyk) je tedy  $(k + 1)$ -tice  $M = \langle U, R_1, \dots, R_k \rangle$ , kde  $U$  je neprázdná ostrá množina a  $R_i$  jsou funkce  $U^{n_i} \rightarrow [0, 1]$ , pro každé  $i \leq k$ . *Ohodnocením* individuových proměnných v modelu  $M$  rozumíme funkci v přiřazující každé individuové proměnné  $x_i$

prvek  $v(x_i) \in U$ . Pravidlostní hodnota  $\|\varphi\|_{M,v,*}$  formule  $\varphi$  v modelu  $M$  vůči spojitě t-normě  $*$  při ohodnocení  $v$  je definována rekurzivně takto:

$$\begin{aligned} \|P_k x_{n_1} \dots x_{n_k}\|_{M,v,*} &= R_k(v(x_{n_1}), \dots, v(x_{n_k})) \\ \|\mathbf{0}\|_{M,v,*} &= 0 \\ \|\varphi \& \psi\|_{M,v,*} &= \|\varphi\|_{M,v,*} * \|\psi\|_{M,v,*} \\ \|\varphi \rightarrow \psi\|_{M,v,*} &= \|\varphi\|_{M,v,*} \Rightarrow * \|\psi\|_{M,v,*} \text{ (a obdobně pro případné dodatečné spojky)} \\ \|(\exists x)\varphi\|_{M,v,*} &= \sup \{\|\varphi\|_{M,v(a/x),*} \mid a \in U\} \\ \|(\forall x)\varphi\|_{M,v,*} &= \inf \{\|\varphi\|_{M,v(a/x),*} \mid a \in U\}, \end{aligned}$$

kde  $v(a/x)$  je ohodnocení, které proměnné  $x$  přiřazuje prvek  $a$  a pro ostatní proměnné se shoduje s ohodnocením  $v$ .

Kvantifikované formule  $(\exists x)\varphi$  a  $(\forall x)\varphi$  se tedy vyhodnocují jako suprémum resp. infimum pravidlostních hodnot dosažených formulí  $\varphi$  pro všechny možné prvky univerza  $U$  modelu. Pokud je formule  $\varphi$  ostrá (tj. nabývá pro všechny prvky  $a \in U$  pouze hodnot 0 či 1), shoduje se sémantika kvantifikátorů s klasickou, neboť pak formule  $(\exists x)\varphi$  nabývá hodnoty 1, právě když existuje  $a \in U$ , pro něž je formule  $\varphi$  plně pravdivá (tj. má hodnotu 1), a jinak hodnoty 0. Podobně pro ostrou formuli  $\varphi$  má formule  $(\forall x)\varphi$  hodnotu 1, právě když je  $\varphi$  (plně) pravdivá pro všechna  $a \in U$ . Pro fuzzy  $\varphi$  je však nutno chápat formule  $(\exists x)\varphi$  a  $(\forall x)\varphi$  pouze jako suprémum resp. infimum dosažených pravidlostních hodnot – tedy nikoli v klasickém smyslu „existuje  $x$  takové, že  $\varphi$ “ resp. „pro všechna  $x$  je  $\varphi$  pravdivé“, nýbrž jako vyjádření, *nakolik* existuje  $x$  takové, že  $\varphi$ , resp. *nakolik* je  $\varphi$  pravdivé pro všechna  $x$ .

Podobně jako ve výrokové fuzzy logice, některé formule jsou pravdivé ve všech modelech a pro všechna ohodnocení. Tyto formule vyjadřují obecně pravdivé logické zákony prvořádkové fuzzy logiky a stejně jako v případě klasického či výrokového je nazýváme (standardními)<sup>27</sup> tautologiemi příslušné fuzzy logiky.

**Definice:** Formule  $\varphi$  predikátové fuzzy logiky v jazyce sestávajícím z predikátů  $P_1, \dots, P_k$  se nazývá *\*-tautologií*, je-li  $\|\varphi\|_{M,v,*} = 1$  v každém modelu  $M$  pro tento jazyk při všech ohodnoceních  $v$  individuových proměnných v tomto modelu.

Formule  $\varphi$  predikátové fuzzy logiky se nazývá *t-tautologií*, je-li *\*-tautologií* pro všechny spojitě t-normy  $*$ .

Podobně jako v případě výrokové fuzzy logiky bychom nyní mohli chápat množinu všech *\*-tautologií* jako predikátovou logiku této t-normy, a množinu všech *t-tautologií* jako predikátovou verzi logiky BL. Ukazuje se však, že na rozdíl od výrokových fuzzy logik nejsou všechny množiny predikátových *\*-tautologií* a *t-tautologií* logiky axiomatizovatelné. Množiny jejich standardních tautologií proto musíme nazývat *standardními* predikátovými fuzzy logikami (značenými  $L^{\forall^s}$ , kde  $L$  je při-

<sup>27</sup> Podobně jako výroková fuzzy logika, i predikátová fuzzy logika má vedle právě popsané *standardní* sémantiky s pravidlostními hodnotami v intervalu  $[0, 1]$  také další sémantiky s hodnotami v obecnějších algebrách. Tyto sémantiky zde rovněž ponecháme stranou.

slušná výroková fuzzy logika), zatímco vlastními predikátovými fuzzy logikami  $L\forall$  budeme rozumět jejich dále uvedené axiomatické aproximace.

Dosadíme-li za výrokové proměnné v libovolné tautologii výrokové fuzzy logiky  $L$  jakékoli formule jazyka predikátové fuzzy logiky, dostaneme pochopitelně standardní tautologii této predikátové logiky (protože výrokové spojky se vyhodnocují ve výrokové i predikátové logice stejně a v obou případech uvažujeme všechna možná ohodnocení ve všech přípustných modelech). Všechny instance výrokových tautologií  $L$  můžeme proto přijmout za axiomy predikátové fuzzy logiky  $L\forall$ , neboť jsou i jejími standardními tautologiemi. Z určitých metamatematických důvodů<sup>28</sup> volíme za další axiomy predikátové fuzzy logiky  $L\forall$  všechny tautologie následujících tvarů:

$$\begin{aligned} &(\forall x)\varphi \rightarrow \varphi(y/x) \\ &\varphi(y/x) \rightarrow (\exists x)\varphi \\ &(\forall x)(\psi \rightarrow \varphi) \rightarrow (\psi \rightarrow (\forall x)\varphi) \\ &(\forall x)(\varphi \rightarrow \psi) \rightarrow ((\exists x)\varphi \rightarrow \psi) \\ &(\forall x)(\psi \vee \varphi) \rightarrow (\psi \vee (\forall x)\varphi), \end{aligned}$$

kde  $y$  je proměnná substituovatelná za  $x$  ve  $\varphi$  (tj. taková proměnná, která není na žádném místě výskytu  $x$  ve  $\varphi$  v dosahu kvantifikátoru  $(\forall y)$  či  $(\exists y)$ ),  $\varphi(y/x)$  je formule, která vznikne nahrazením všech výskytů proměnné  $x$  ve  $\varphi$  proměnnou  $y$ , a formule  $\psi$  neobsahuje žádné nekvantifikované výskyty proměnné  $x$ . Vedle výrokového odvozovacího pravidla modus ponens je odvozovacím pravidlem  $L\forall$  také pravidlo generalizace:

$$\text{z již odvozené formule } \varphi \text{ je možno odvodit formuli } (\forall x)\varphi \quad (26)$$

(pro všechny formule  $\varphi$  a individuové proměnné  $x$ ). Lze ukázat, že taková sada axiomů je korektní a úplná pro standardní Gödelovu predikátovou logiku  $G\forall^s$  (můžeme tedy psát  $G\forall^s = G\forall$ ). Není však úplná pro standardní predikátové fuzzy logiky  $L\forall^s$ ,  $\Pi\forall^s$  ani  $BL\forall^s$ , a jejich neúplnost nelze opravit přidáním žádné rekurzivní množiny axiomů (lze totiž dokázat, že množiny standardních tautologií těchto logik nejsou rekurzivně spočetné). Ukažme tedy alespoň korektnost těchto predikátových axiomů a pravidel pro každou spojitou  $t$ -normu  $*$ :

*Důkaz:* Vzpomeňme, že podle sémantiky implikace má formule  $(\forall x)\varphi \rightarrow \varphi(y/x)$  pravdivostní hodnotu 1, právě když pravdivostní hodnota  $(\forall x)\varphi$  je nejvýše rovna pravdivostní hodnotě  $\varphi(y/x)$ . To ale triviálně nastává v každém modelu při každém ohodnocení, neboť hodnota  $(\forall x)\varphi$  je infimum všech hodnot nabývaných formulí  $\varphi$  (z nichž jedna je hodnotou  $\varphi(y/x)$ ). Podobně korektnost druhého axiomu vyplývá z toho, že hodnota instance  $\varphi$  nepřevyšuje supremum takových instancí. Korektnost ostatních axiomů ponecháme jako cvičení, ověřme však ještě, že pravidlo generalizace přenáší tautologičnost: má-li  $\varphi$  ve všech modelech  $M$  při všech ohodnoceních  $v$  pravdivostní hodnotu 1, je i infimum pravdivostních hodnot v  $M$  přes všechny hodnoty  $v(x)$  rovno 1, a tedy  $(\forall x)\varphi$  má v  $M$  pravdivostní hodnotu 1, což bylo dokázati.

<sup>28</sup> Logiky  $L\forall$  jsou pak totiž korektní a úplné alespoň vůči jedné ze svých nestandardních sémantik, tvořené modely nad všemi lineárními  $L$ -algebami.

Až na výrokové odlišnosti zmíněné v oddílu 2 je i predikátová fuzzy logika do značné míry podobná logice klasické. Opatrnost při zacházení s kvantifikátory je třeba jen v některých případech, které jsou opět způsobeny příbuzností fuzzy logik s intuicionistickou logikou. Například v případě kvantifikátorových de Morganových zákonů, neboli pravidel negování kvantifikátorů, platí ve fuzzy logice (stejně jako v logice intuicionistické) obecně pouze tři implikace ze čtyř:

$$\begin{aligned} (\forall x)\neg\varphi &\leftrightarrow \neg(\exists x)\varphi \\ (\exists x)\neg\varphi &\rightarrow \neg(\forall x)\varphi \end{aligned} \quad (27)$$

Obecnou neplatnost obrácené implikace k (27) lze nahlédnout z následujícího protipříkladu kupř. v Gödelově logice. Necht' hodnoty  $\varphi$  dosahují v modelu  $M$  pro různá individua libovolně malých kladných hodnot. Pravdivostní hodnota formule  $(\forall x)\varphi$  je infimem těchto hodnot, tedy 0. Hodnota formule  $\neg(\forall x)\varphi$  je tudíž  $\sim_G(0) = 1$ . Naproti tomu hodnoty  $\neg\varphi$  jsou pro všechna individua modelu rovna 0 (neboť  $\varphi$  má v našem modelu pouze kladné hodnoty a  $\sim_G(\alpha) = 0$  pro  $\alpha > 0$ ); i jejich supremum, neboli hodnota formule  $(\exists x)\neg\varphi$ , je tedy 0. Obrácená implikace k (27), tj.  $\neg(\forall x)\varphi \rightarrow (\exists x)\neg\varphi$ , má proto hodnotu  $1 \Rightarrow_G 0 = 0$ . Tato obrácená implikace tedy v modelu  $M$  neplatí, není tudíž tautologií Gödelovy logiky (a tím spíše ani logiky  $BL\forall$ ). Stejný protipříklad funguje i pro logiku  $\Pi\forall$ , neboť má negaci se stejnou pravdivostní funkcí jako  $G\forall$ . Ekvivalence  $(\exists x)\neg\varphi \leftrightarrow \neg(\forall x)\varphi$  však platí v Łukasiewiczově logice, kde se díky zákonu dvojité negace chovají kvantifikátory v tomto směru klasicky.

Pro další podrobnosti o predikátové fuzzy logice odkážeme čtenáře na Hájkovu monografii [14] a soustředíme se nyní na některé ukázky *použití* aparátu formální fuzzy logiky pro usuzování o graduálních vlastnostech a pojmech.

#### 4 Aplikace formální fuzzy logiky

Hlavní vtíp budování formálního aparátu fuzzy logiky spočívá v tom, že její logické zákony lze nejenom studovat (jak jsme to dělali v předchozích oddílech), nýbrž i *užívat k usuzování* o graduálních vlastnostech a pojmech. Jedná se totiž především o *logiku*, tedy obecně platné zákony korektních úsudků, nejen o matematický aparát sloužící k jejich nalezení a popisu. Podobně jako lze u klasické logiky studovat pravdivostní tabulky či sémantiku kvantifikátorů a s jejich pomocí nacházet obecně platné zákony usuzování o ostrých vlastnostech objektů, tak jsme i my v předchozích oddílech studovali pravdivostní funkce a prvořadové modely za účelem nalezení logických zákonů platných pro graduální vlastnosti objektů. Avšak stejně jako hlavní význam klasické logiky nespočívá v detailním studiu jejích vlastností, nýbrž v jejím použití jakožto *pravidel* korektního usuzování o ostrých výrocích, je i hlavním účelem formální fuzzy logiky její použití pro korektní úsudky o graduálních výrocích.

Jak již bylo naznačeno v úvodním oddílu, o graduálních vlastnostech bychom mohli usuzovat i pomocí klasické logiky, a to tak, že bychom hovořili výhradně o funkcích příslušnosti. To je však nepřirozené, chápeme-li vlastnosti jako *vysoký* coby *vlastnosti*, které můžeme objektům do nějaké míry připsat: chceme-li hovořit o mno-

žině všech vysokých lidí, a nikoli jen o funkci přiřazující lidem jejich výšku, musíme v klasické logice provést idealizaci a předpokládat, že je tato množina ostrá. Tato idealizace je hned na první pohled dosti drastická, neboť vylučuje přechodové případy a nutí nás vést mezi nimi nepřirozenou ostrou hranici. V některých případech pak taková idealizace vede přímo k protiintuitivním výsledkům, projevujícím se jako „sémantické paradoxy“. Ty jsou způsobeny tím, že – jak jsme viděli – zákony logiky graduálních vlastností se mírně liší od klasických zákonů logiky ostrých vlastností, a zanedbání tohoto rozdílu není vždy zanedbatelné. Než se však dostaneme k vysvětlení takových paradoxů pomocí formální fuzzy logiky, rozeberme si nejprve, co přesně usuzování podle jejich pravidel znamená.

Vezměme například následující tautologii fuzzy logiky  $BL\forall$ :

$$Px \ \& \ (\forall y)(Py \rightarrow Qy) \rightarrow Qx \quad (28)$$

Tento logický zákon lze aplikovat na kterýkoli soubor objektů a jakékoli graduální (či ostré, neboť i ty jsou speciálním případem graduálních) vlastnosti  $P$ ,  $Q$ . Například pro množinu jablek tento zákon potvrzuje správnost úsudku vyjádřeného v přirozeném jazyce takto:

Toto jablko je červené; červená jablka jsou zralá; toto jablko je tedy zralé. (29)

Správnost tohoto úsudku je tautologičností zákona (28) v  $BL\forall$  potvrzena *nezávisle* na konkrétních (pravdivostních) hodnotách červenosti, zralosti, výběru konkrétního jablka či spojitě t-normy určující pravdivostní funkce výrokových spojek. Úsudek je platný v každém případě, kdy význam použitých výrokových spojek chápeme v souladu s jejich vlastnostmi předpokládanými v t-normových fuzzy logikách (tedy že konjunkce je komutativní, asociativní atd., implikace splňuje reziduální zákon vůči konjunkci atd.). Tyto předpoklady o významu spojek jsou nejen přirozené, ale do značné míry i vynucené požadavkem, aby logické vyplývání přenášelo stupně pravdivosti tvrzení a logické spojky toto vyplývání zachycovaly (pro detaily viz [2]).

V konkrétním případě úsudku (29) by k jeho formalizaci dostačovala i klasická logika, s idealizací červenosti a zralosti jako ostrých vlastností. Ve skutečnosti *všechny* úsudky formalizovatelné v logice  $BL\forall$  jsou také správnými úsudky podle klasické logiky, protože  $BL\forall$  i logiky  $L_*\forall$  jednotlivých spojitých t-norem jsou *slabší* než logika klasická (všechny jejich logické zákony totiž platí i pro ostré výroky, které jsou speciálními případy graduálních). Důležité je zde právě to, že klasická logika se mohla svojí idealizací dopustit chyby a uznat za správný i úsudek, který pro graduální výroky platný není. Tautologičnost formule (28) v  $BL\forall$  ale ukazuje, že v tomto případě se tak nestalo a že tento úsudek platí v plné míře *i pro* graduální vlastnosti. Později nicméně uvidíme, že klasická logika, odhlížející od graduality výroků, nesprávnou predikci skutečně v některých případech dává.

Další výhodou použití fuzzy logiky v případě úsudků, jako je (29), je možnost jejich aplikace i v případě, že jejich premisy nejsou úplně, nýbrž jen částečně pravdivé. Úsudek tedy vyjadřuje nejen to, že je-li dané jablko *zcela* červené, je také *zcela* zralé, ale i to, že je-li částečně červené, je také částečně zralé – přesněji je zralé *alespoň* ve stupni, v jakém je červené (neboť taková je sémantika plně pravdivé implikace, viz

oddíl 2). Příslušná volba škál červenosti a zralosti je na uživateli, který chce z červenosti na zralost usuzovat: zvolí-li obě škály vhodně, bude prostřední premisa „červená jablka jsou zralá“ plně pravdivá a popsané usuzování z částečné červenosti na nejméně takovou zralost bude plně platné. Zákon (28), na němž je úsudek (29) založen, však platí, ať zvolí jeho uživatel škály červenosti a zralosti jakkoli: zvolí-li je tak, že premisa „červená jablka jsou zralá“ bude pouze *zčásti* pravdivá, úsudek bude přesto v pořádku – bude však vyjadřovat pouze to, že toto jablko je zralé přinejmenším natolik, nakolik je (ve zvolené škále) červené a *nakolik* (silná konjunkce!) při zvolených škálách platí, že červená jablka jsou zralá (v konkrétním modelu tedy nakolik je velké infimum hodnot rezidua zvolené t-normy mezi červeností a zralostí). Volba, kterou spojitou t-normou se mají kombinovat premisy v konkrétním sémantickém modelu, je opět na uživateli a na jeho zvoleném chápání spojek *a*, *pak*, *ne* atd.; a podle zvoleného chápání dostane v *tomto* modelu sémantické hodnoty odpovídající *této* volbě. Úsudek (29) však bude díky tautologičnosti zákona (28) platný pokaždé, *nezávisle* na tom, jaké parametry uživatel pro svůj sémantický model konkrétně vybral. Později uvidíme, že sémantická situace může volbu vhodných spojitých t-norem (a tedy i volbu konkrétní fuzzy logiky) v některých případech zúžit.

Výše jsme uvedli, že klasická logika nemusí vždy dávat dobrý popis logických vztahů mezi graduálními výroky. V případě úsudku (29) tomu tak shodou okolností nebylo, neboť tento úsudek je správný i podle zákonů fuzzy logiky. Podívejme se však na několik případů, kdy je klasický předpoklad ostrosti zcela zavádějící.

V oddílu 2 jsme viděli, že jedním z významných rozdílů mezi klasickou a fuzzy logikou je zákon dvojité negace. Ten je zcela adekvátní pro ostré výroky klasické logiky, a v některých sémantických situacích má plné oprávnění dokonce i pro graduální výroky – totiž pokud negaci chápeme jako míru nedostatku vlastnosti, měřenou právě chybějícím množstvím. Například v případě vlastnosti *vyšoký* (měřené centimetry normalizovanými do  $[0, 1]$ ) dává dobrý smysl chápat stupeň její negace *nevvyšoký* jako měřený chybějícími centimetry. Výroky  $Vx$  a  $\neg\neg Vx$  (v přirozeném jazyce vyjádřené např. větami „Jan je vyšoký“ a „Jan není nevvyšoký“) tak lze docela dobře chápat jako ekvivalentní. V jiných sémantických situacích však graduální výroky tento zákon nespĺňují, a hodí se pro ně spíše striktní negace, jakou má Gödelova a produktová logika: vezměme třeba vlastnost *vinný* u podezřelých. Zatímco vina má různé stupně (odrážené například ve výši hrozícího trestu), její negace („nevinný“) je dvouhodnotová – podezřelý nevinný buď je, anebo není. Fakt, že je podezřelý nevinný, znamená, že jeho vina je nulová; naproti tomu vinný může být podezřelý v jakékoli nenulové míře (srovnejme podmínku, za níž je pravdivá striktní negace ve standardní sémantice  $G$  a  $\Pi$ ). Zatímco tedy vlastnost *vinný* má stupně, její negace *nevinný* je dvouhodnotová. V důsledku toho i její dvojitá negace má jen dva stupně: vlastnost *není nevinný* podezřelému (zcela) přísluší, kdykoli je jeho vina nenulová, a (vůbec) nepřislúší, je-li jeho vina nulová (opět srovnejme význam dvojité negace v  $G$  a  $\Pi$ , kde vyjadřuje, že výrok je pravdivý „alespoň trochu“). Výroky „Jan je vinný“ a „Jan není nevinný“ tedy nejsou ekvivalentní, neboť první výrok je graduální (může být pravdivý v různých stupních podle míry Janova provinění), zatímco ten druhý je dvouhodnotový (je pravdivý, jakmile je Jan jen trochu vinný). Klasický zákon dvojité negace tedy není v této sémantické situaci

(tj. pro tuto vlastnost *a při* tomto chápání negace) vhodný; naopak tuto situaci dobře popisuje striktní negace Gödelova či produktová. Klasická logika by zde svou idealizací předpovídala ekvivalenci obou výroků, jež jsme však nahlédli jako odlišné.

Uvedený příklad také ilustruje, jakým způsobem lze vybírat pravdivostní funkce spojek v konkrétních sémantických modelech (případně rozhodnout, zda lze použít silnější logiku konkrétní spojitě t-normy, a nikoli jen obecné zákony BL platné pro všechny možnosti). Je jasné, že výběr konkrétního sémantického modelu se musí dít na základě *sémantické analýzy*, podobné té z předchozího příkladu, která může vhodnou volbu pravdivostních funkcí a konkrétních pravdivostních hodnot omezit (zde nám například ukázala, kde použít involutivní a kde striktní negaci). Toto sémantické modelování, které nutně obsahuje i prvek volby uživatele a přihlídnutí k jeho cílům a požadavkům na výsledek, však už *nepatří* do oboru formální fuzzy logiky, nýbrž spadá pod formální sémantiku či logickou analýzu přirozeného jazyka. Připomeňme, že formální logika (ať klasická, nebo fuzzy) se nestará o to, který konkrétní sémantický model (tj. které ohodnocení, interpretace predikátů atd.) je použit, neboť její úlohou je popisovat zákony, které platí *ve všech* takových modelech.

Ukažme nyní patrně nejdůležitější typ použití aparátu formální fuzzy logiky pro logické modelování graduálních vlastností, jímž je řešení sémantických paradoxů typu paradoxu hromady. Tento paradox, označovaný též řeckým slovem *sórités* (které je v řečtině přídavným jménem od *sóros* – hromada), je znám již od starověku a je přisuzován řeckému filosofu Eubúlidovi.<sup>29</sup> Lze jej formulovat několika způsoby, například jako úsudek s následujícími dvěma korektně vyhlížejícími předpoklady:

Sto tisíc zrn obilí tvoří hromadu.

Odebráním jediného zrna hromada nepřestane být hromadou.

Aplikujeme-li však druhý předpoklad postupně stotisíckrát, vyvodíme paradoxní důsledek, že také *nula* zrn tvoří hromadu, což zjevně není pravda.

Proveďme logickou analýzu tohoto paradoxu. Premisy paradoxního úsudku mají následující logickou formu:<sup>30</sup>

$$\begin{aligned} & Hz_{100\,000} \\ & (\forall n)(Hz_n \rightarrow Hz_{n-1}), \end{aligned} \tag{30}$$

kde individua  $z_n$  pro  $n = 0, \dots, 100\,000$  zastupují soubory  $n$  zrn a  $H$  je predikát „být hromadou“. V klasické logice získáváme spor s očividným faktem  $\neg Hz_0$  (nula zrn ne-

<sup>29</sup> Eubúlídés z Milétu (4. století př. n. l.) byl příslušníkem megarské školy a současně Aristotelovým. Je mu připisováno celkem sedm logických paradoxů, vedle paradoxu hromady např. též známý paradox lháře.

<sup>30</sup> Formulujeme ji v predikátovém jazyce, paradox je však možno formalizovat i výrokově (kde induktivní premisu nahradíme příslušným množstvím jednotlivých implikací), nebo dokonce i zcela bez použití logických spojek, jako pouhou řadu otázek: Tvoří 100 000 zrn hromadu? Tvoří 99 999 zrn hromadu? Tvoří 99 998 zrn hromadu? atd., až po otázku: Tvoří 0 zrn hromadu? Přestože na první otázku musíme odpovědět „ano“ a na poslední „ne“, neexistuje v této řadě žádné konkrétní místo, kde by se měla odpověď změnit. Ať už ale formulujeme paradox jakkoli, podstata problému daná graduálností příslušné vlastnosti ani jeho řešení v rámci fuzzy logiky se nijak nemění.



tvoří hromadu): opakovanou aplikací indukční premisy (30) totiž můžeme postupně odvozovat pravdivost  $H_{z_{99\ 999}}$ ,  $H_{z_{99\ 998}}$ ,  $H_{z_{99\ 997}}$  atd. až po  $H_{z_0}$ . V klasické logice tedy platí

$$H_{z_{100\ 000}} \& (\forall n)(H_{z_n} \rightarrow H_{z_{n-1}}) \rightarrow H_{z_0}, \quad (31)$$

což představuje spor s  $\neg H_{z_0}$ .

Možných přístupů k tomuto paradoxu bylo v historii navrženo více (viz např. [16], [24]), s různou mírou přesvědčivosti řešení. Podívejme se, jaké sémantické modelování této paradoxní situace nabízí formální fuzzy logika. Jak lze tušit, podle formální fuzzy logiky je paradox způsoben tím, že vlastnost „být hromadou“ není ostrá, nýbrž graduální (lze ji měřit počtem zrn – čím větší počet zrn, tím více je jejich souhrn hromadou). Zdánlivý paradox vzniká právě díky tomu, že je na graduální vlastnost použit aparát klasické logiky – čili je s ní zacházeno, jako kdyby byla ostrá. Přestože tato idealizace v běžných případech příliš nevadí, situace paradoxu hromady představuje extrémní případ, jenž ji přivádí ad absurdum. Při použití logiky vhodné pro graduální vlastnosti (tj. některé fuzzy logiky) paradox mizí, neboť lze najít sémantický model, který situaci dobře vystihuje, a ukázat, že klasicky platná formule (31), jež vedla k paradoxu, ve fuzzy logice obecně *neplatí*.

Adekvátní sémantický model situace paradoxu hromady lze vytvořit nejlépe v Łukasiewiczově logice.<sup>31</sup> Zadat sémantický model  $M$  pro jazyk sestávající z jediného unárního predikátu  $H$  v prvořádové Łukasiewiczově logice (viz oddíl 3) znamená stanovit funkci  $M_H$  přiřazující individuím  $z_n$  čísla z intervalu  $[0, 1]$  (tj. vlastně fuzzy množinu interpretující extenzi vlastnosti „být hromadou“). Jaký přesně má mít taková funkce průběh, není a priori jasné – a skutečně to ani není starostí formální fuzzy logiky, nýbrž volby konkrétního sémantického modelu: znovu platí, že jaké přiřazení uživatel (sémantik–modelář) zvolí, takový model bude mít. Formální fuzzy logika zde pouze podává základní návod a upozorňuje na obecné vlastnosti, které bude mít *každý* takový zvolený model.

Sémantickou analýzou můžeme nalézt některé omezující podmínky pro funkci  $M_H$ , která určuje pravdivostní hodnoty výroků  $H_{z_n}$ . Protože platí, že čím větší počet zrn, tím spíše je jejich soubor hromadou, měla by tato funkce být rostoucí (podle sémantiky t-normových fuzzy logik to v souladu s intuicí znamená úplnou pravdivost implikace  $H_{z_n} \rightarrow H_{z_m}$  pro  $m \geq n$ ). Stejně jako v klasické logice dále připisujeme výrokům  $H_{z_{100\ 000}}$  a  $\neg H_{z_0}$  pravdivostní hodnotu 1 – označíme je za zcela pravdivé. Máme tedy za to, že 100 000 zrn již tvoří prototypickou hromadu (do hromadovitosti již takovému množství zrn nic nechybí) a že 0 zrn není hromadou vůbec ( $H_{z_0}$  má pravdivost 0). Narozdíl od klasického modelu však indukční premise (30) nepřipíšeme pravdivostní hodnotu 1 (pak bychom paradox nevyřešili, neboť při takovém ohodnocení by paradoxní formule (31) platila i ve fuzzy logice). Všimneme si totiž, že odstranění jediného zrna naší graduální vlastnost (měřenou počtem zrn!) sice nepatrně,

<sup>31</sup> Pro řešení je totiž třeba, aby pravdivostní hodnoty iterované konjunkce  $A^n$  s rostoucím  $n$  klesaly (nelze tedy použít t-normu s idempotentními prvky) a aby dosáhly pro nějaké  $n$  nuly. Tyto vlastnosti mají mezi spojitými t-normami pouze t-normy izomorfní Łukasiewiczově t-normě, vhodnou logikou je zde proto právě Łukasiewiczova logika.

ale přece jen o něco sníží. O kolik přesně, záleží opět na konkrétní volbě funkce  $M_H$ , je ale jasné, že toto snížení musí být pro každé  $n$  pouze nepatrné, neboť hromady lišící se o jediné zrnko jsou prakticky nerozlišitelné (nemohou se tedy výrazně lišit ani ve vlastnosti „býti hromadou“). Obecné řešení paradoxu (jako např. v [15]) by mělo ponechat pro funkci  $M_H$  v mezích těchto omezení nějakou volnost a předpokládat jen, že rozdíl sousedních pravdivostních hodnot je menší než nějaké velmi malé  $\varepsilon > 0$ ; zde však pro jednoduchost model přímo zafixujeme (neboť nám jde pouze o demonstraci, že v Łukasiewiczově logice nějaký takový model *existuje*) a stanovíme pravdivostní hodnotu  $H_{z_n}$  napevno jako  $n / 100\,000$  (neztrácejme však ze zřetele, že to je naše volba a že existují i jiné vyhovující modely, jež jsou pro danou sémantickou situaci možná přiměřenější, a že formální fuzzy logika od takových konkrétních modelů *abstrahuje* a zajímá se jen o obecné zákony platné pro graduální vlastnosti). Vidíme, že potom pravdivostní hodnota induktivní premisy (30) vychází podle sémantiky  $\mathbb{L}\forall$  jako

$$\inf(\min(1 - n / 10^5 + (n - 1) / 10^5), 1) = 1 - 1 / 10^5 = 0,99999,$$

tedy hodnota velmi blízká 1. To také vysvětluje, proč induktivní premisa vypadá velmi přijatelně (takže je v klasickém modelu prohlášena za pravdivou) – ona sama i každá z implikací  $H_{z_n} \rightarrow H_{z_{n-1}}$  je totiž *téměř* pravdivá. Chybějící stotisícina do její úplné pravdivosti nicméně způsobuje, že hodnota  $H_{z_n}$  pozvolna klesá od plné pravdivosti  $H_{z_{100\,000}}$  až k plné nepravdivosti  $H_{z_0}$ .

Zkonstruovaný sémantický model dokládá, že klasicky platná formule (31) v Łukasiewiczově fuzzy logice obecně neplatí. Její neplatnost souvisí s již popsanou rolí silné konjunkce, u níž je nutno brát v úvahu i počet použití premis (srv. diskusi formule (24) v oddílu 2). K odvození  $H_{z_0}$  je totiž nejprve nutno z induktivní premisy  $(\forall n)(H_{z_n} \rightarrow H_{z_{n-1}})$  vyvodit všechny implikace  $H_{z_n} \rightarrow H_{z_{n-1}}$  pro  $n = 100\,000, 99\,999, \dots, 1$ , jejichž konjunkce teprve společně s  $H_{z_{100\,000}}$  implikuje  $H_{z_0}$ . Podle snadného zobecnění zákona (24) tedy ve fuzzy logice platí jen

$$H_{z_{100\,000}} \& ((\forall n)(H_{z_n} \rightarrow H_{z_{n-1}}))^{100\,000} \rightarrow H_{z_0},$$

a tato formule již ve sporu s  $\neg H_{z_0}$  není (protože např. v našem zvoleném modelu má prostřední stotisícinásobná konjunkce pravdivostní hodnotu  $1 - 10^5 \cdot (1/10^5) = 0$ ). Lze si všimnout, že kdybychom onu téměř pravdivou premisu (30) použili jen několikrát, měli bychom sémantikou Łukasiewiczovy fuzzy logiky stále zaručenu velice vysokou pravdivostní hodnotu závěru (0,99999 při jediné aplikaci, 0,99998 při dvojnásobné, 0,99997 při trojnásobné atd.). Při „krátkých“ úsudcích se tedy takovéto skoro pravdivé premisy chovají v zásadě klasicky – závěr stále zůstává téměř pravdivý. Dlouhá série použití nedokonale pravdivé premisy však tuto nedokonalost nasčítala a přinutila nás vzít v úvahu platnost zákonů fuzzy logiky pro graduální vlastnosti (konkrétně zákona (24) místo klasického (23)). Aproximace klasickou logikou, v jiných situacích běžně použitelná, se zde již ukázala nepřijatelnou.

S podobným paradoxem se setkáme i u jiných pojmů než jen u pojmu hromady. Eubúlidés sám vytvořil jeho variantu týkající se holohlavosti: člověk se 100 000 vlasů není holohlavý, a neholohlavý člověk se vypadnutím jediného vlasu holohlavým

nemůže stát (neboť rozdíl jediného vlasu není znatelný). Snadno lze vymyslet bezpočet variant aplikovaných na nejrůznější pojmy.

Jednou z takových aplikací je i řešení paradoxu logické vševědoucnosti agentů, s nímž se setkáváme ve standardní epistémické logice. Pojem *odvoditelné znalosti* agenta (tj. souhrn všech jeho znalostí, které dokáže odvodit ze svých explicitních znalostí, daných např. obsahem jeho momentální paměti) je totiž zjevně graduální množinou – odvozovací krok stojí agenta malé, v dlouhých odvozeních však nezanedbatelné úsilí (čas, zdroje; srv. [3]). Aproximace ostrou množinou, kterou provádí klasická epistémická logika, pak vede k paradoxu logické vševědoucnosti: mezi odvoditelné znalosti agenta musejí patřit mj. všechny logické tautologie (neboť ty lze odvodit z jakýchkoli předpokladů); to je však v případě fyzických agentů zcela nereálná schopnost (tautologičnost většiny formulí s miliónem výrokových proměnných nelze ověřit ani v čase mnohonásobně přesahujícím stáří vesmíru). Použití formální fuzzy logiky na graduální pojem odvoditelnosti znalostí tento paradox odstraňuje stejným způsobem jako paradox hromady (jehož je de facto instancí, viz [1]).

Jiným známým paradoxem odstranitelným pomocí formální fuzzy logiky je tzv. Poincarého paradox. Ten klade do kontrastu intuitivní tranzitivitu relace nerozlišitelnosti (je-li  $a$  nerozlišitelné od  $b$  a  $b$  od  $c$ , pak je i  $a$  nerozlišitelné od  $c$ ) se zjevnou rozlišitelností krajních bodů dlouhé série  $a_1, a_2, \dots, a_N$  individuí, z nichž každá dvě sousední jsou nerozlišitelná (nerozlišitelnost zde rozumíme nerozlišitelnost v některém aspektu, např. barvě, poloze apod.). To je opět případ řady typu *sórités* a správná aplikace formální fuzzy logiky paradox odstraňuje. Detaily řešení přenecháme čtenáři a pouze naznačíme, že je-li relace nerozlišitelnosti  $E$  pojata jako graduální, pak *může* být tranzitivní *ve smyslu fuzzy logiky*, tj. může pro ni platit ve stupni 1 formule

$$(\forall x)(\forall y)(\forall z)(Exy \ \& \ Eyz \rightarrow Exz). \quad (32)$$

Sémanticky to však na rozdíl od klasické logiky připouští, aby stupeň nerozlišitelnosti v řadě  $a_1, a_2, \dots, a_N$  nerozlišitelných sousedních individuí klesal (podobně jako stupeň hromadovitosti u paradoxu hromady).

Další příklady aplikací aparátu formální fuzzy logiky na konkrétní graduální vlastnosti již ponecháme představivosti čtenáře. Upozorníme však na možný zdroj chybných interpretací, které se v souvislosti s používáním fuzzy logiky často vyskytují. Přestože je fuzzy logiku možno aplikovat na libovolné graduální vlastnosti či výroky, je třeba mít na paměti, že v ní jde o *logické vyplývání* přenášející pravdivostní stupně, nikoli o usuzování založené například na pravděpodobnosti, četnosti, stupních přesvědčení apod. Pravděpodobnost (jakož i četnost, přesvědčení atd.) se chová podle *jiných* zákonů, než jsou ty, jež řídí logické přenášení částečné pravdivosti mezi graduálními výroky – zejména není extenzionální vůči pravdivostním hodnotám (angl. truth-functional): pravděpodobnost složeného výroku *není* funkcí pravděpodobností jeho složek (stačí uvážit závislé a nezávislé jevy). Pravdivostní hodnota složeného výroku byla naproti tomu *definována* právě jako funkce pravdivostních hodnot jeho složek. Je tomu tak proto, že pravděpodobnost popisuje *rozdělení šancí* mezi možnými výsledky, zatímco fuzzy logika popisuje *logické přenášení* stupňů graduálních veličin.

Založíme-li tedy nějakou graduální vlastnost na pravděpodobnosti,<sup>32</sup> musíme počítat s tím, že konjunkce graduálních výroků o pravděpodobnosti obecně *neodpovídá* pravděpodobnosti konjunkce těchto výroků. Konjunkce pravděpodobností je záležitostí logickou, kdežto pravděpodobnost konjunkce záležitostí pravděpodobnostní.<sup>33</sup> Znovu zdůrazněme, že fuzzy logika se týká *logického* vyplývání mezi graduálními výroky, tedy zachovávání stupňů graduálních vlastností v úsudcích, a není náhražkou teorie pravděpodobnosti, teorie míry, epistémické logiky či jiných disciplín popisujících odlišné fenomény.

Přehlídku uplatnění formální fuzzy logiky zakončíme nastíněním širokých možností, jaké poskytuje pro matematické modelování graduálních fenoménů. Pomocí prvořadové fuzzy logiky je totiž možno budovat *formální fuzzy matematiku*, tedy matematické teorie, které připouštějí graduálnost některých svých pojmů (například umožňují korektně pracovat s pojmy jako *malé přirozené číslo*).

Součástí formální fuzzy matematiky je i formální teorie fuzzy množin, která v ní hraje podobně fundamentální roli jako teorie množin v matematice klasické. Narozdíl od tradiční teorie fuzzy množin založené na pojmu funkce příslušnosti (viz oddíl 1), formální fuzzy matematika přistupuje k fuzzy množinám – v plné analogii s klasickou teorií množin – jako k axiomaticky zachycenému *primitivnímu pojmu*. Zatímco tedy tradiční teorie fuzzy množin operuje se stupni příslušnosti  $A(x)$ , formální teorie fuzzy množin pracuje přímo se vztahem náležení  $x \in A$ . Protože v ní ale připouštíme graduálnost množin, musíme i  $x \in A$  považovat za *graduální* výrok o náležení. Při usuzování o takových výrociích používáme tudíž pravidla logiky  $BL\forall$  či jiných fuzzy logik (díky své vyjadřovací síle je k tomu obzvláště vhodná logika  $\text{Ł}\Pi$ , popsaná v oddílu 3). Zdůrazněme, že na rozdíl od tradiční teorie fuzzy množin ve formulích formální teorie fuzzy množin *nehovoříme* o pravdivostních stupních či funkcích příslušnosti: gradua-lita fuzzy množin je „zakódována“ již do samotné logiky, kterou používáme k usuzování.<sup>34</sup>

<sup>32</sup> Např. budeme-li chápat stupeň červenosti jablka jako pravděpodobnost, že je náhodně vybraný člověk označí za červené. Takováto zdůvodnění stupňů příslušnosti se v aplikované literatuře fuzzy metod často vyskytují.

<sup>33</sup> Výrok „ $A$  je pravděpodobné a  $B$  je pravděpodobné“ (formálně  $PA \ \& \ PB$ ) se totiž významně liší od výroku „ $A$  a  $B$  je pravděpodobné“ (formálně  $P(A \ \& \ B)$ ). Výrok „toto jablko je zralé a červené“ tak ani při pravděpodobnostní interpretaci obou graduálních vlastností nebude znamenat pravděpodobnost, že náhodně vybraný člověk označí jablko za zralé i červené, nýbrž jen stupeň, který jsme se jako uživatelé aparátu fuzzy logiky *rozhodli* konjunkci obou graduálních vlastností, byť zjištěných pravděpodobnostně, připisovat v konkrétním sémantickém modelu.

<sup>34</sup> Tento přístup je i metodologicky výhodný, neboť ošetření graduality zůstane „zapouzdřeno“ ve vrstvě logiky, a při zkoumání fuzzy množin samotných se jím již nemusíme zabývat. Používání neklasické logiky k matematickému usuzování je zprvu nezvyklé, lze je však nacvičit (podobně, jako třeba používání ryze konstruktivních metod v konstruktivistické matematice) – jde při tom především o to, vyhnout se při usuzování pravidlům, které platí pouze pro ostré množiny. Některé rysy fuzzy logiky však vedou k nutnosti akceptovat dosti odlišný tvar vět a definic: např. nemožnost zkrátit vícenásobný počet téhož předpokladu na jediný vede k častému výskytu teorémů tvaru  $A^n \ \& \ B^m \ \& \ C^k \ \rightarrow D$  atp., viz [8].

Jako příklad axiomu pro fuzzy množiny uveďme axiom extenzionality:

$$(\forall x)\Delta(x \in A \leftrightarrow x \in B) \rightarrow A = B. \quad (33)$$

Přestože tento axiom explicitně nehovoří o pravdivostních hodnotách, ze sémantiky fuzzy logiky víme, že jeho předpoklad vyjadřuje v každém modelu identitu stupňů příslušnosti  $x$  do  $A$  i  $B$ , tedy rovnost funkcí příslušnosti. Axiom tedy stanovuje, že fuzzy množiny s totožnými funkcemi příslušnosti jsou totožné, neboli že fuzzy množiny jsou *určeny* svými prvky. Všimněme si zde podobnosti s klasickou teorií množin, jejíž axiom extenzionality vyjadřuje totéž pro ostré množiny a používá k tomu i velmi podobnou formuli: jediným rozdílem je operátor  $\Delta$  v předpokladu (33), který v klasické teorii množin není třeba uvádět, neboť je pro ostré výroky bezobsažný.

Termy  $\{x \mid \varphi(x)\}$ , které v klasické teorii množin označují množinu všech  $x$  splňujících podmínku  $\varphi$ , je možno zavést i ve formální teorii fuzzy množin. Protože však podmínka  $\varphi$  může být graduální, je nutno tento term chápat jako označující fuzzy množinu, do níž každý prvek  $x$  patří *natolik*, nakolik splňuje  $\varphi$ ; právě to je totiž v modelech predikátové fuzzy logiky význam jeho definičního axiomu

$$y \in \{x \mid \varphi(x)\} \leftrightarrow \varphi(y). \quad (34)$$

Předvedme si letmo způsob práce ve formální fuzzy matematice (méně triviální příklady lze najít v [4]). Můžeme například zavést průnik fuzzy množin  $A, B$  jako fuzzy množinu

$$A \cap B = \{x \mid x \in A \ \& \ x \in B\}.$$

Podle (34) tedy platí  $x \in A \cap B \leftrightarrow x \in A \ \& \ x \in B$ , kde  $\&$  ovšem znamená (silnou) konjunktci fuzzy logiky. Dokažme například asociativitu průniku fuzzy množin v  $BL\forall$ :

$$(A \cap B) \cap C = A \cap (B \cap C) \quad (35)$$

*Důkaz:* Podle (34) a (21) platí následující řetězec ekvivalencí:

$$\begin{aligned} x \in (A \cap B) \cap C &\leftrightarrow (x \in A \ \& \ x \in B) \ \& \ x \in C \leftrightarrow \\ &\leftrightarrow x \in A \ \& \ (x \in B \ \& \ x \in C) \leftrightarrow x \in A \cap (B \cap C), \end{aligned}$$

tedy podle (22) je  $x \in (A \cap B) \cap C \leftrightarrow x \in A \cap (B \cap C)$ . Pravidly  $\Delta$ -necesitace (25) a generalizace (26) tudíž dostáváme

$$(\forall x)\Delta(x \in (A \cap B) \cap C \leftrightarrow x \in A \cap (B \cap C)),$$

odkud z axiomu extenzionality plyne dokazované tvrzení  $(A \cap B) \cap C = A \cap (B \cap C)$ .

Všimněme si opět, že důkaz je velmi podobný klasickému a že neoperuje s funkcemi příslušnosti, nýbrž přímo s graduálním vztahem náležení, o němž usuzujeme podle zákonů fuzzy logiky. V *modelech* formální teorie fuzzy množin přitom věta (35) vyjadřuje právě chování funkcí příslušnosti (silného) průniku tradičních fuzzy množin.

Formální teorie fuzzy množin také přirozeným způsobem zobecňuje ostré pojmy tradiční teorie množin na graduální. Používáme-li totiž k usuzování fuzzy logiku, jsou

všechny definované pojmy automaticky graduální. Formální fuzzy matematika tedy netrpí neduhem tradiční teorie fuzzy množin zmíněným pod čarou v pozn. 8 oddílu 1. Vhodné definice graduálních matematických pojmů jsou přitom často velice podobné klasickým – například graduální inkluze fuzzy množin je příhodně definována formulí  $(\forall x)(x \in A \rightarrow x \in B)$  stejně jako v klasické matematice. Tradiční ostrá definice (4) naproti tomu odpovídá formuli  $\Delta(\forall x)(x \in A \rightarrow x \in B)$ , tj. speciálnímu případu *plné* pravdivosti graduální inkluze (takovýto vztah mezi graduálními pojmy formální teorie a ostrými pojmy tradiční teorie bývá ve fuzzy matematice pravidlem). O graduální fuzzy inkluzi lze dokazovat věty stylem podobným předchozí ukázce, a získat tak graduální zobecnění ostrých vět tradiční teorie fuzzy množin.

Podrobnější popis formální teorie fuzzy množin již přesahuje zaměření této kapitoly, proto pouze odkážeme na práce [5], [7], [4] a v nich uvedenou literaturu. Shrňme tedy, že jsme se od Goguenovy motivace fuzzy logiky Zadehovou teorií fuzzy množin přes vybudování aparátu výrokové a predikátové formální fuzzy logiky dostali obloukem zpět k teorii fuzzy množin, tentokrát však budované axiomaticky nad formální fuzzy logikou. Tato „objížďka“ nám umožnila pracovat s fuzzy množinami jako primitivním pojmem, jehož graduálnost je zakódována pouze do vrstvy formální fuzzy logiky, kterou používáme k usuzování. Původní motivace zůstala zachována v *modelech* teorie: pouze ty jsou tvořeny konkrétními funkcemi příslušnosti. Teorie sama však, jak se na formální teorii sluší, od nich odhlíží, a popisuje pouze zákonitosti platné v libovolném modelu. Díky tomu lze vybudovat fuzzy matematiku na podobné metodologické bázi (popsané v [5]), na jaké je postavena klasická matematika. Takovým způsobem pojatá axiomatická fuzzy matematika je potom teorií podobného druhu, jakou je Vopěnkova alternativní teorie množin [23]: tak jako Vopěnkova teorie bere vážně pojem horizontu, formální fuzzy matematika bere vážně pojem graduální vlastnosti. Obě teorie jsou pak díky tomu schopny zachytit některé matematické disciplíny přirozeněji, než je to možné pomocí ostrých pojmů klasické matematiky.

## 5 Závěr

V tomto textu jsme se zabývali výhradně *formální* fuzzy logikou, nikoli inženýrskými fuzzy metodami a jejich aplikacemi. Ty totiž (ač rozšířenější, známější a snad i v praxi bezprostředně užitečnější) nesdílejí s formální fuzzy logikou základní metodologické předpoklady – nepopisují logické zákony *zachovávání stupňů graduálních vlastností*. Proto ani nejsou vázány požadavky na logické vyplývání vyjádřené vlastnostmi spojitých t-norem a jejich reziduí. Je to dáno rozdílnými cíli i aplikacemi obou přístupů (viz [2]): zatímco formální fuzzy logika usiluje o podání koherentního systému pro *logické dedukce* se zachováváním stupňů graduálních vlastností, fuzzy metody slouží většinou pro vzájemně nezávislé cykly řízení ve smyslu „načti data, agreguj je, a podle výsledku proved' akci“. V důsledku toho mají fuzzy metody jen velmi málo společného s logickou analýzou či sémantikou přirozeného jazyka, a bývají vytvářeny ad hoc podle potřeb konkrétních aplikací (v nichž jsou však nadmíru efektivní). Fuzzy metody jsou tak spíše balíkem inženýrských nástrojů než logickou či matematickou teorií. Literatury o fuzzy metodách je – na rozdíl od formální fuzzy logiky – k dispozici dostatek,

a lze odkázat na kompendia jako [18]. Existují i příklady aplikací formální fuzzy logiky pro kvalifikované použití některých inženýrských fuzzy metod (např. [19], [20] či kap. 7 v [14]), ty však vyžadují právě znalost aparátu, který jsme zde stručně popsali.

Formální fuzzy logika představuje přirozené rozšíření klasické logiky ostrých vlastností na vlastnosti graduální. Pro korektní nakládání s graduálními vlastnostmi jde tedy o stejně fundamentální disciplínu, jakou je z hlediska ostrých vlastností klasická logika. Graduální vlastnosti jsou přitom v přírodovědných i humanitních disciplínách (včetně kognitivní vědy) de facto rozšířenější než ostré. Na rozdíl od matematiky, v níž se obvykle setkáváme pouze s vlastnostmi dvouhodnotovými, v přirozeném světě je spíše obtížné najít vlastnost, která by v nějakém smyslu graduální nebyla.<sup>35</sup> Korektní logické usuzování o graduálních vlastnostech je proto nepostradatelné nejen pro přirozené i umělé agenty, kteří se s nimi v reálném světě setkávají doslova na každém kroku, ale i pro samotný popis těchto agentů a jejich interakcí se světem i mezi sebou navzájem. Z tohoto důvodu je studium graduálních vlastností a jejich logické modelování důležité nejen pro logiku samu, ale i pro kognitivní vědu, umělou inteligenci, filosofii, informatiku, lingvistiku i ostatní formální, přírodní a společenské vědy. Přestože je totiž formální fuzzy logika vhodná především pro *teoretické* porozumění fenoménům spojeným s graduálními vlastnostmi, využití vhodných sémantických modelů na tomto porozumění založených může přinést i užitečné praktické nástroje, tak jako důkladné teoretické porozumění pravděpodobnosti vedlo k úspěšným statistickým metodám a aplikacím.

*Grantová a institucionální podpora:* Tato kapitola vznikla za podpory Výzkumného záměru č. AV0Z10300504 „Informatika pro informační společnost: modely, algoritmy, aplikace“ a projektu 1ET100300517 programu „Informační společnost“.

## Literatura

- [1] Běhounek, L.: Dvě souvislosti mezi epistemickou a fuzzy logikou. In Kelemen, J., Kvasnička, V., Pospíchal, J. (eds.): *Kognícia a umelý život VII*, str. 37–42. Opava: Slezská univerzita v Opavě 2007.
- [2] Běhounek, L.: On the difference between traditional and deductive fuzzy logic. *Fuzzy Sets and Systems* **159** (2008) 1153–1164.
- [3] Běhounek, L.: Fuzzy logics interpreted as logics of resources. In M. Peliš (ed.): *The Logica Yearbook 2008* (v tisku).
- [4] Běhounek, L., Bodenhofer, U., Cintula, P.: Relations in Fuzzy Class Theory: initial steps. *Fuzzy Sets and Systems* **159** (2008) 1729–1772.
- [5] Běhounek, L., Cintula, P.: From fuzzy logic to fuzzy mathematics: a methodological manifesto. *Fuzzy Sets and Systems* **157** (2006) 642–646.

---

<sup>35</sup> Ukazuje to i přirozený jazyk: přibližně lze totiž říci, že přídavné jméno vyjadřuje graduální vlastnost, lze-li je stupňovat. Fakt, že většina přídavných jmen má druhý stupeň, tak dokládá téměř univerzální rozšíření graduálních vlastností.

- 
- [6] Běhounek, L., Cintula, P.: Fuzzy class theory. *Fuzzy Sets and Systems* **154** (2005) 34–55.
- [7] Běhounek, L., Cintula, P.: Fuzzy class theory: a primer v1.0. Výzkumná zpráva č. 939, Ústav informatiky AV ČR, 2006.
- [8] Běhounek, L., Cintula, P.: Features of mathematical theories in formal fuzzy logic. *Lecture Notes in Artificial Intelligence* **4529** (2007) 523–532.
- [9] Chvalovský, K.: On the independence of the axioms of BL and MTL. In Hák, F. (ed.): *Doktorandské dny '08*, pp. 28–36, Praha: Ústav informatiky AV ČR & Matfyzpress 2008.
- [10] Cintula, P.: The  $\mathbb{L}\Pi$  and  $\mathbb{L}\Pi^{1/2}$  propositional and predicate logics. *Fuzzy Sets and Systems* **124** (2001) 289–302.
- [11] Esteve, F., Godo, L., Montagna, F.: The  $\mathbb{L}\Pi$  and  $\mathbb{L}\Pi^{1/2}$  logics: two complete fuzzy systems joining Łukasiewicz and product logics. *Archive for Mathematical Logic* **40** (2001) 39–67.
- [12] Flaminio, T., Marchioni, E.: T-norm based logics with an independent involutive negation. *Fuzzy Sets and Systems* **157** (2006) 3125–3144.
- [13] Goguen, J. A.: Logic of inexact concepts. *Synthese* **19** (1969) 325–373.
- [14] Hájek, P.: *Metamathematics of Fuzzy Logic*. Dordrecht: Kluwer 1998.
- [15] Hájek, P., Novák, V.: The sorites paradox and fuzzy logic. *International Journal of General Systems* **32** (2003) 373–383.
- [16] Keefe, R., Smith, P. (eds.): *Vagueness: A Reader*. Cambridge (Mass.), London: MIT Press 1999.
- [17] Klement, E. P., Mesiar, R., Pap, E.: *Triangular Norms*. Dordrecht: Kluwer 2000.
- [18] Klir, G. J., Yuan, B.: *Fuzzy Sets and Fuzzy Logic – Theory and Applications*. Upper Saddle River, NJ: Prentice Hall 1995.
- [19] Novák, V.: *Fuzzy množiny a jejich aplikace*. Praha: SNTL 1986.
- [20] Novák, V.: *Základy fuzzy modelování*. Praha: BEN 2000.
- [21] Novák, V., Perfilieva I., Močkoř, J.: *Mathematical Principles of Fuzzy Logic*. Dordrecht: Kluwer 2000.
- [22] Poincaré, H.: *La valeur de la science*. Paris: Flammarion 1904.
- [23] Vopěnka, P.: *Úvod do matematiky v alternativnej teórii množín*. Bratislava: Alfa 1989.
- [24] Williamson, T.: *Vagueness*. London, New York: Routledge 1994.
- [25] Zadeh, L. A.: Fuzzy sets. *Information and Control* **8** (1965) 338–353.



# Rekurentné neurónové siete

Michal ČERNANSKÝ<sup>1</sup>

**Abstrakt.** Neurónové siete patria medzi úspešne používané nástroje strojového učenia. Rekurentné neurónové siete boli navrhnuté na modelovanie časových postupností a na rozdiel od klasických dopredných neurónových sietí obsahujú rekurentné prepojenia, ktoré umožňujú, aby informácia z predchádzajúceho obdobia mohla ovplyvniť aktivity neurónov v aktuálnom čase. Rekurentné siete boli úspešne použité v mnohých reálnych aplikáciách pri riešení úloh predikcie, adaptívneho riadenia, identifikácie systémov či spracovania signálu. Významným faktorom limitujúcim ich širšie použitie je komplexnosť algoritmov používaných na ich tréning.

## 1 Úvod

Umelé neurónové siete sú výpočtové modely inšpirované biologickými neurónovými sieťami, akou je napríklad aj ľudský mozog. Paralela s prírodou je v spôsobe realizácie výpočtu – výsledný zložitý výpočet je vytvorený interakciou mnohých elementov realizujúcich jednoduchšie výpočty. Bolo navrhnuté veľké množstvo typov neurónových sietí [10][17], avšak medzi najpopulárnejšie určite patria tzv. viacvrstvové perceptrónové neurónové siete, ktoré sú tréňované algoritmom spätného šírenia chyby. V ďalšom texte pod pojmom neurónová sieť budeme označovať práve neurónové siete zložené z perceptrónov.

Z inžinierskeho hľadiska sú neurónové siete populárne a často používané výpočtové modely, a to najmä vzhľadom na jednoduchosť ich použitia a schopnosť rýchlo dosiahnuť uspokojivé výsledky. Neurónová sieť pozostáva z väčšieho počtu výpočtových elementov, nazývaných neuróny, ktoré sú vzájomne pospájané cez váhové prepojenia s rôznou intenzitou. Tréning neurónovej siete pozostáva z modifikovania intenzít váhových prepojení tak, aby neurónová sieť adekvátne reagovala na vstupy, t.j. aby produkovala požadované výstupy. Príklad jednoduchej neurónovej siete je znázornený na obr. 1a.

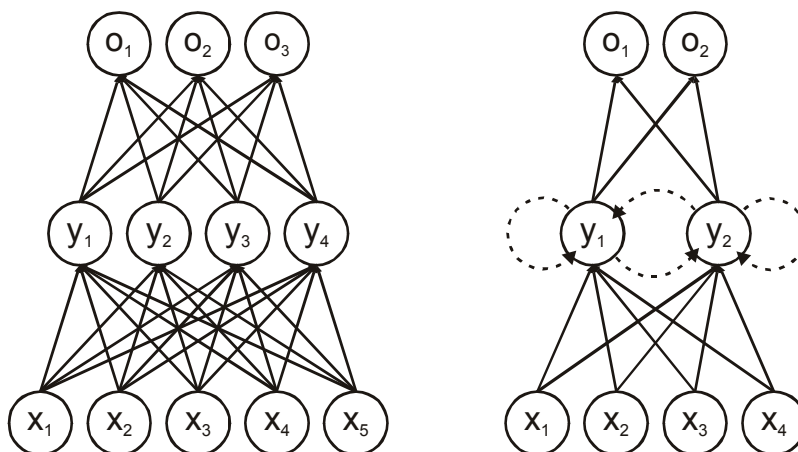
Klasické dopredné neurónové siete bývajú používané napr. pri riešení úloh klasifikácie (určenie, či objekt má danú vlastnosť) alebo regresie (nájdanie závislosti popisujúcej vzťah medzi premennými). Pre spracovanie úloh s časovým kontextom,

---

<sup>1</sup> Ústav aplikovanej informatiky, FIIT, STU Bratislava, E-mail: cernansky@fiit.stuba.sk

akou je napríklad predikcia (predpovedanie nasledujúcej hodnoty alebo hodnôt v postupnosti), boli navrhnuté rekurentné neurónové siete.

Rekurentné neurónové siete môžu byť považované za jednoduchú modifikáciu dopredných neurónových sietí, ktorá vzniká pridaním tzv. rekurentných prepojení. Rekurentné prepojenia spájajú neuróny takým spôsobom, že aktivita neurónov z predchádzajúceho časového kroku cez rekurentné prepojenie ovplyvňuje aktivitu neurónu v aktuálnom časovom kroku. Príklad rekurentnej neurónovej siete je znázornený na obr. 1b.



**Obr. 1.** Príklad (a) doprednej a (b) rekurentnej neurónovej siete. Dopredné prepojenia prenášajúce informáciu v rámci aktuálneho kroku výpočtu sú znázornené plnou čiarou a rekurentné prepojenia prenášajúce informáciu z minulého do súčasného kroku výpočtu sú znázornené prerušovanou čiarou.

Rekurentné neurónové siete boli úspešne použité vo viacerých praktických úlohách vyžadujúcich modelovanie časových postupností. Bolo navrhnutých viacero architektur, tréningových algoritmov a ich modifikácií, avšak rekurentné neurónové siete stále nie sú bežne používanými nástrojmi strojového učenia. Dôvodom je výrazne vyššia výpočtová náročnosť tréningových algoritmov v porovnaní s klasickými doprednými neurónovými sieťami. Úspešná aplikácia rekurentných sietí si tiež zvyčajne vyžaduje hlbšiu znalosť problémovej úlohy, použitého typu rekurentnej siete, ako aj zvoleného tréningového algoritmu.

Cieľom tejto práce je zoznámiť čitateľa s problematikou rekurentných neurónových sietí a uľahčiť mu ich prípadnú aplikáciu. Štruktúra tejto práce je nasledovná: v ďalšej kapitole stručne predstavíme dopredné neurónové siete a zoznámime sa so spôsobom zápisu algoritmov pre dopredné, ako aj rekurentné neurónové siete, ktorý bude použitý v nasledujúcich kapitolách. Spôsob zápisu bude demonštrovaný na známom algoritme spätného šírenia chyby. V tretej kapitole si predstavíme niektoré z architektur rekurentných neurónových sietí, s ktorými sa môže čitateľ stretnúť v odbornej literatúre. V štvrtej kapitole si vysvetlíme bežné algoritmy založené na gradientovej minimalizácii chyby a v piatej kapitole popíšeme aj

algoritmus využívajúci Kalmanovu filtráciu, ktorý v porovnaní s bežne používanými prístupmi vykazuje v mnohých parametroch výrazne lepšiu výkonnosť. V šiestej kapitole sa na úlohách modelovania postupností symbolov pokúsime preniknúť do spôsobu fungovania rekurentných neurónových sietí. Predstavíme si niektoré základné dynamické správania pozorovateľné v stavovej časti rekurentnej siete. V siedmej kapitole opíšeme dve praktické aplikácie rekurentných neurónových sietí. V poslednej kapitole využijeme navrhnutý spôsob zápisu algoritmov a navrhujeme jednoduchý a elegantný spôsob zakódovania rekurentnej siete do údajových štruktúr. Potom tréningové algoritmy formálne opísané v tretej kapitole zapíšeme v jednoduchom pseudojazyku v podobe, ktorá umožňuje ich takmer okamžité použitie v bežných procedurálnych programovacích jazykoch, akými sú napríklad jazyk C či Java.

## 2 Dopredné neurónové siete

Viacvrstvová neurónová sieť zložená z perceptrónov (angl. Multilayer Perceptron) je najčastejšie používaný typ neurónových sietí [17]. Prívlastok „dopredná“ (angl. Feedforward) používame v kontexte rekurentných neurónových sietí na zdôraznenie smeru toku informácie. V tejto podkapitole stručne zhrnieme základné informácie o dopredných neurónových sieťach a v krátkosti popíšeme algoritmus spätného šírenia chyby. Tiež sa oboznámime so spôsobom zápisu algoritmov nezávislom na konkrétnej architektúre.

### 2.1 Neurón

V oblasti umelých neurónových sietí (inšpirujúc sa neurobiológiou) pojmom neurón označujeme jednoduché elementy, z ktorých sú umelé neurónové siete zložené. Perceptrón je neurón, ktorý realizuje jednoduchý matematický výpočet. Obsahuje  $N$  vstupov  $x_1$  až  $x_N$ , ktoré sú na neho pripojené prostredníctvom váhových prepojení s intenzitami  $w_1$  až  $w_N$ . Intenzita, čiže váha prepojenia určuje, ako výrazne daný vstup ovplyvňuje výslednú aktivitu neurónu. Výpočet, ktorý neurón realizuje, je teda možné zapísať ako:

$$\tilde{o} = \sum_{i=1}^N w_i x_i + \Theta = \sum_{i=1}^N w_i x_i + w_0 = \sum_{i=0}^N w_i x_i, \quad (1)$$

$$o = f(\tilde{o}), \quad (2)$$

kde  $\tilde{o}$  označuje tzv. vnútornú aktivitu neurónu,  $o$  označuje výstupnú aktivitu neurónu a  $\Theta$  je prah neurónu. Je vhodné prah neurónu považovať za špeciálnu váhu  $w_0$  ( $\Theta = w_0$ ) smerujúcu zo špeciálneho vstupu  $x_0$  vždy nastaveného na konštantu 1. V literatúre často uvádzaný vzťah, v ktorom je od sumy  $\sum w_i x_i$  prah  $\Theta$  odčítavaný, je ekvivalentný s rovnicou (1) vzhľadom na to, že prah môže byť aj záporný. Funkcia  $f$  je tzv. aktivačná alebo prechodová funkcia. Veľmi často používaná aktivačná funkcia

je sigmoidálna funkcia. Uvádzame vzťah na výpočet funkčnej hodnoty a tiež výpočet derivácie:

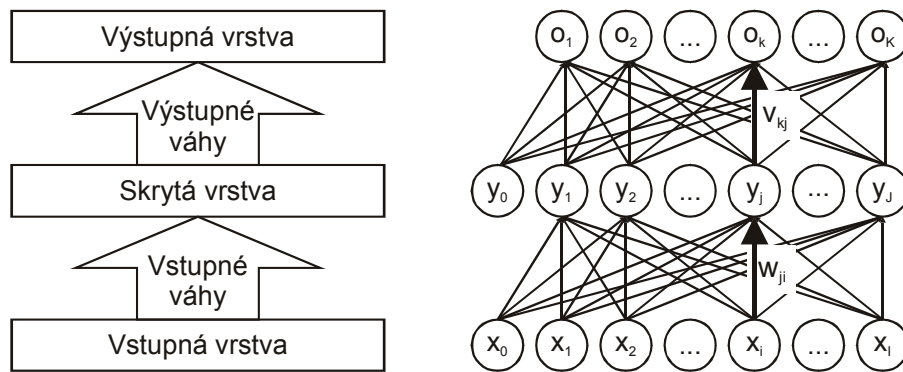
$$f(x) = \frac{1}{1 + e^{-x}}, \quad (3)$$

$$f'(x) = f(x)(1 - f(x)). \quad (4)$$

V nasledujúcom texte pojmom neurón budeme označovať perceptrón realizujúci uvedený jednoduchý výpočet.

## 2.2 Viacvrstvá neurónová sieť

Napriek tomu, že vo všeobecnosti je možné skonštruovať doprednú neurónovú sieť so skoro ľubovoľne poprepájanými neurónmi, v praxi sa zvyčajne používajú neurónové siete s neurónmi organizovanými do vrstiev. Najčastejšie používaná architektúra je trojvrstvá neurónová sieť a je znázornená na obr. 2. Vstupné neuróny nerealizujú nijaký výpočet, iba udržuujú vstupnú informáciu (a preto sa niekedy architektúra označuje ako dvojvrstvá sieť, obsahuje iba dve vrstvy „skutočných“ neurónov). Zo vstupnej vrstvy sa informácia šíri na skrytú vrstvu obsahujúcu skryté neuróny. Zvyčajne každý skrytý neurón je prostredníctvom vstupných váh prepojený s každým vstupným neurónom. Z neurónov skrytej vrstvy sa informácia šíri na výstupné neuróny na výstupnej vrstve. Opäť, každý výstupný neurón je prostredníctvom výstupných váh prepojený s každým skrytým neurónom.



**Obr. 2.** (a) Schematické znázornenie trojvrstvej neurónovej siete. (b) Trojvrstvá neurónová sieť zložená z  $I$  vstupných,  $J$  skrytých a  $K$  výstupných neurónov.

Neurónová sieť z obr. 2b má  $I$  vstupných neurónov, na ktoré sa pred výpočtom umiestni vstupná informácia –  $I$ -tica reálnych čísel  $(x_1, x_2, \dots, x_i, \dots, x_I) \in R^I$ . Aktivity skrytých neurónov –  $J$ -tica reálnych čísel  $(y_1, y_2, \dots, y_j, \dots, y_J) \in R^J$ , sú vypočítané ako:

$$\tilde{y}_j = \sum_{i=0}^I w_{ji} x_i, \quad (5)$$

$$y_j = f(\tilde{y}_j), \quad (6)$$

kde  $f$  je aktivačná funkcia (najčastejšie sigmoida) a  $\tilde{y}_j$  je vnútorný potenciál skrytého neurónu  $j$ . Aktivity výstupných neurónov –  $K$ -tica reálnych čísel  $(o_1, o_2, \dots, o_k, \dots, o_K) \in R^K$ , sú vypočítané ako:

$$\tilde{o}_k = \sum_{j=0}^J v_{kj} y_j, \quad (7)$$

$$o_k = f(\tilde{o}_k), \quad (8)$$

kde  $f$  je aktivačná funkcia (najčastejšie sigmoida alebo lineárna) a  $\tilde{o}_k$  je vnútorný potenciál výstupného neurónu  $k$ . Tento výpočet predpokladá, že špeciálne „prahové“ neuróny  $x_0$  a  $y_0$  sú nastavené na hodnotu 1, a teda je možné s prahom  $j$ -teho skrytého neurónu  $w_{j0}$  a prahom  $k$ -teho výstupného neurónu  $v_{k0}$  pracovať ako s bežnou váhou. V ďalšom texte nebudeme špeciálne odlišovať prahy a váhy neurónovej siete.

### 2.3 Trénovanie neurónových sietí - spätné šírenie chybového signálu

Významnou vlastnosťou neurónových sietí je možnosť ich natréňovať na základe predpripravených vstupno-výstupných vzoroch. Trénovacia množina je tvorená typickými vstupmi a k nim prislúchajúcimi očakávanými výstupmi. Po natréňovaní sa od neurónovej siete očakáva, že dokáže zovšeobecňovať, čiže správne spracovávať aj vstupy, ktoré trénovacia množina neobsahovala.

Najznámejším a najčastejšie používaným algoritmom na trénovanie neurónových sietí je algoritmus spätného šírenia chybového signálu (ang. Error Backpropagation alebo iba Backpropagation, BP) [29]. Umožňuje vhodným spôsobom organizovať výpočet derivácií potrebných pre zmenu váh pomocou gradientovej metódy najprudšieho spádu. Tieto derivácie sa vypočítavajú postupne od neurónov na vyšších vrstvách smerom k neurónom na nižších vrstvách.

Typický scenár trénovanie neurónovej siete je nasledovný:

#### Príprava trénovacej (a aj testovacej) množiny

Riešená úloha je transformovaná do podoby umožňujúcej jej riešenie pomocou neurónovej siete. Je potrebné rozhodnúť, čím budú tvorené vstupy do neurónovej siete a čo budeme očakávať na výstupe neurónovej siete. Výstupom z tejto fázy je trénovacia (a väčšinou aj testovacia množina) pozostávajúca z viacerých vstupno-výstupných vzorov  $(\mathbf{x}, \mathbf{d})$ , kde  $\mathbf{x} = (x_1, \dots, x_I)$  je vstupný vektor a  $\mathbf{d} = (d_1, \dots, d_K)$  je k nemu prislúchajúci vektor očakávaných hodnôt. Takýchto vzorov obsahuje trénovacia aj testovacia množina väčší počet.

### Inicializácia neurónovej siete

Inicializácia spočíva vo vytvorení neurónovej siete. Hodnoty váhových prepojení sú iniciované náhodne na malé hodnoty, napr. z intervalu  $(-0.5, 0.5)$ . Počet vstupných neurónov siete je samozrejme daný rozmerom vstupného vektora a počet výstupných neurónov je rovný rozmeru vektora požadovaných hodnôt. Počet skrytých neurónov je možné zvoliť experimentálne, typicky sú to jednotky až desiatky neurónov. Zvyčajne sa konštruujú siete s jedinou vrstvou skrytých neurónov.

### Trénovanie metódou spätného šírenia chybového signálu

Náhodne sa vyberie jeden z prvkov trénovacej množiny  $(\mathbf{x}, \mathbf{d})$  a vstupný vektor  $\mathbf{x}$  je umiestnený na vstup siete. Dopredným šírením sa vypočítajú skryté a potom aj výstupné aktivity  $\mathbf{o} = (o_1, o_2, \dots, o_k, \dots, o_K)$ . Po doprednom šírení očakávame na výstupe siete požadované hodnoty  $\mathbf{d} = (d_1, d_2, \dots, d_k, \dots, d_K)$ , ktoré sa zvyčajne od sieťou vypočítaných hodnôt  $\mathbf{o}$  líšia. Túto chybu, vzniknutú na výstupe, sa snažíme minimalizovať gradientovou metódou najprudšieho spádu. Chybu v danom kroku trénovacieho algoritmu definovanú ako

$$E = \frac{1}{2} \sum_{k=1}^K (d_k - o_k)^2, \quad (9)$$

chceme minimalizovať gradientovou metódou najprudšieho spádu, a teda váhy chceme zmeniť pomocou vzťahov:

$$\Delta w_{ji}(t) = -\alpha \frac{\partial E}{\partial w_{ji}} + \beta \Delta w_{ji}(t-1), \quad (10)$$

$$\Delta v_{ji}(t) = -\alpha \frac{\partial E}{\partial v_{kj}} + \beta \Delta v_{ji}(t-1), \quad (11)$$

kde rýchlosť učenia  $\alpha$  určuje mieru zmeny váhy. Cez momentum  $\beta$  je aktuálna zmena váhy ovplyvnená predchádzajúcou zmenou váhy. Momentum pôsobí ako istá zotrvačnosť pri menení váh, urýchľuje trénovanie a pomáha zabrániť uviaznutiu v lokálnych minimách chybovej funkcie.

Derivácie pre metódu najprudšieho spádu  $\partial E / \partial w_{ji}$  a  $\partial E / \partial v_{kj}$  sú vypočítané prostredníctvom spätne šíreného chybového signálu  $\partial E / \partial \tilde{o}_k = \delta_k^{\text{out}}$  a  $\partial E / \partial \tilde{y}_j = \delta_j^{\text{hid}}$  nasledovným spôsobom. Najskôr je vypočítaný chybový signál pre výstupné neuróny:

$$\delta_k^{\text{out}} = (d_k - o_k) f'(\tilde{o}_k), \quad (12)$$

a následne je možné vypočítať chybový signál pre skryté neuróny:

$$\delta_j^{\text{hid}} = f'(\tilde{y}_j) \sum_{k=1}^K v_{kj} \delta_k^{\text{out}}. \quad (13)$$

Po vypočítaní chybových signálov  $\delta_k^{\text{out}}$  a  $\delta_j^{\text{hid}}$  je možné vypočítať zmeny pre výstupné  $\Delta v_{kj}(t)$  a vstupné  $\Delta w_{ji}(t)$  váhové prepojenia:

$$\Delta v_{kj}(t) = \alpha \delta_k^{\text{out}} y_j + \beta \Delta v_{kj}(t-1), \quad (14)$$

$$\Delta w_{ji}(t) = \alpha \delta_j^{\text{hid}} x + \beta \Delta w_{ji}(t-1), \quad (15)$$

a váhové prepojenia následne zmeniť:

$$v_{kj}(t+1) = v_{kj}(t) + \Delta v_{kj}(t), \quad (16)$$

$$w_{ji}(t+1) = w_{ji}(t) + \Delta w_{ji}(t). \quad (17)$$

Aby sme odlíšili pôvodné a nové hodnoty pre váhové prepojenia, sú váhy aj zmeny váh v rovnici (14) až (17) uvádzané s časovými indexmi.

### Ukončenie tréovania a otestovanie siete

Algoritmus ukončíme, ak uplynul stanovený počet krokov algoritmu, popr. ak po stanovenom počte krokov algoritmu je výsledná chyba uspokojivo malá. Výslednú chybu vypočítame ako súčet čiastkových chýb vypočítaných podľa rovnice (9) po prechode cez celú tréovaciu alebo testovaciu množinu. Ak chyba ešte neklesla pod želanú úroveň, v tréovaní pokračujeme.

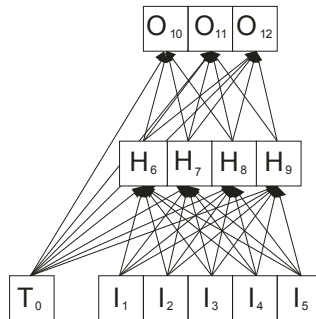
## 2.4 Alternatívny spôsob zápisu algoritmu spätného šírenia chyby

V literatúre je možné nájsť veľa spôsobov zápisu algoritmu spätného šírenia chyby. Vyššie uvedený spôsob zápisu algoritmu patrí medzi tie prehľadnejšie a ľahko pochopiteľné. Napriek tomu v tejto podkapitole uvedieme alternatívny spôsob zápisu algoritmu, ktorý s úspechom využijeme pri zápise algoritmov pre rekurentné neurónové siete. Tento alternatívny spôsob zápisu nebude závislý od konkrétnej architektúry neurónovej siete, nebude ani vyžadovať organizáciu neurónov do vrstiev, a ako uvidíme v poslednej kapitole, umožní nám jednoduchým a priamočiarym spôsobom zapísať algoritmy v pseudojazyku veľmi blízkom klasickým programovacím jazykom, akým je jazyk C či Java.

Od neurónovej siete budeme vyžadovať, aby jej neuróny boli usporiadané a očíslované tak, aby neurón s vyšším indexom bol cez svoje váhové prepojenia spojený iba s neurónmi s nižšími indexmi. Každá dopredná neurónová sieť spĺňa túto podmienku, ktorá vlastne vyjadruje fakt, že výpočet aktivít v doprednej sieti je možné realizovať dopredným spôsobom, a to postupným vypočítaním aktivít pre neuróny s indexom od 0 po  $N-1$ , kde  $N$  je počet neurónov v sieti.

Neurónová sieť s očíslovanými neurónmi je znázornená na obr. 3. Neurón s indexom 0 je špeciálny vstupný neurón vždy nastavený na konštantu 1, zvyčajne každý neurón, ktorý nie je vstupný, je na neho pripojený prostredníctvom váhového prepojenia plniaceho funkciu prahu.

Aj váhové prepojenia sú označené indexmi 0 až  $M-1$ , pričom  $M$  je počet všetkých váhových prepojení. Váhové prepojenie je vlastne trojica hodnôt  $(i, j, v)$ , kde  $i$  je index cieľového neurónu a  $j$  je index zdrojového neurónu. Váha, čiže intenzita váhového prepojenia, je označená symbolom  $v$ . Aby sme sprístupnili jednotlivé položky usporiadanej trojici reprezentujúcej váhové prepojenie, zdefiniujeme si operátory  $d, s$  a  $w$ . Operátor  $d$  (d ako angl. „destination“) nám sprístupní cieľový neurón váhového prepojenia s daným indexom, operátor  $s$  (s ako angl. „source“) nám sprístupní zdrojový neurón a  $w$  intenzitu váhového prepojenia. Nech  $(i, j, v)$  je váhové prepojenie s indexom  $l$ , tak potom  $d_l = i, s_l = j$  a  $w_l = v$ . Množinu indexov všetkých váhových prepojení označíme  $W$ .



**Obr. 3.** Dopredná neurónová sieť zložená z 5 vstupných ( $I_1$  až  $I_5$ ), 4 skrytých ( $H_1$  až  $H_4$ ) a 3 výstupných ( $O_1$  až  $O_3$ ) neurónov.  $T_0$  je špeciálny neurón s konštantnou jednotkovou aktivitou.

Dopredné šírenie je možné zapísať nasledovným spôsobom: neurón s indexom 0 je nastavený na konštantu 1, neuróny tvoriace vstupy do neurónovej siete sú nastavené na hodnoty podľa vstupného vzoru aktuálne prezentovaného sieti. Aktivity na ostatných neurónoch sú dopredným spôsobom (rastúci index  $i$ ) nastavené na základe vzťahu:

$$\tilde{x}_i = \sum_{\forall j \in W | d_j = i} w_j x_{s_j}, \quad (18)$$

$$x_i = f(\tilde{x}_i), \quad (19)$$

kde zápis  $j \in W | d_j = i$  je množina indexov takých váhových prepojení, ktorých cieľový neurón je neurón s indexom  $i$ . Sú to vlastne všetky váhové prepojenia patriace neurónu  $i$ . Cez váhové prepojenie  $w_j$  sa šíri aktivita  $x_{s_j}$  zdrojového neurónu s indexom  $s_j$ .

Spätne šírenie je možné zapísať podobne jednoduchým spôsobom. Chybový signál  $\delta_i$  neurónu s indexom  $i$  je vypočítaný ako:



$$\delta_i = \left[ (d_i - x_i) + \sum_{\forall j \in \mathcal{W} | s_j = i} w_j \delta_{d_j} \right] f'(\tilde{x}_i). \quad (20)$$

Chyba  $d_i - x_i$  vzniká iba na výstupných neurónoch, a iba keď je definovaný želaný výstup  $d_i$ . V opačnom prípade uvažujeme  $d_i = x_i$ , a teda člen  $d_i - x_i = 0$ . Pri výpočte chybového signálu  $\delta_i$  sa cez všetky váhové prepojenia vychádzajúce z neurónu  $i$  prešúria už vypočítané chybové signály. Indexy váhových prepojení vychádzajúcich z neurónu  $i$  sú určené zápisom  $\forall j \in \mathcal{W} | s_j = i$ : sú to také prepojenia indexované pomocou  $j$ , ktorých zdrojový neurón je  $i$ . Výpočet je nevyhnutné realizovať od neurónov s vyšším indexom k neurónom s nižším indexom. Po vypočítaní chybových signálov je možné zmeniť váhy podľa:

$$\Delta w_i(t) = \alpha \delta_{d_i} x_{s_i} + \beta \Delta w_i(t-1), \quad (21)$$

$$w_i(t+1) = w_i(t) + \Delta w_i(t), \quad (22)$$

kde  $\alpha$  je rýchlosť učenia a  $\beta$  je hodnota určujúca momentum.

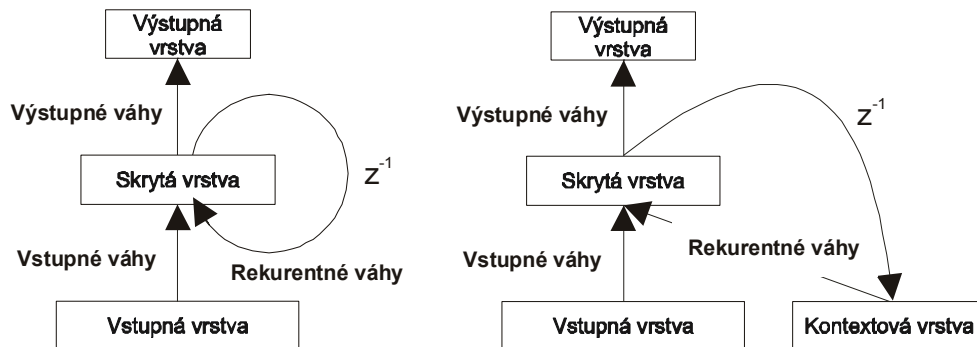
V uvedenom zápise dopredné šírenie signálu a aj spätného šírenia chybového signálu pri algoritme backpropagation nerozlišujeme vstupné, skryté a výstupné neuróny. Taktiež nepožadujeme, aby boli neuróny organizované do vrstiev, v ktorých je každý neurón z vrstvy spojený s každým neurónom predchádzajúcej vrstvy. S úspechom teda môžeme využiť uvedený zápis na tréning doprednej neurónovej siete s ľubovoľne poprepájanými neurónmi.

### 3 Rekurentné neurónové siete

Rekurentné neurónové siete (angl. Recurrent Neural Networks, RNNs) vznikli jednoduchým rozšírením dopredných sietí o váhové prepojenia prenášajúce do aktuálneho kroku výpočtu aktivity z predchádzajúceho časového kroku. Tieto prepojenia tvoriace v grafe reprezentujúceho neurónovú sieť cykly nazývame rekurentné váhové prepojenia [10][17].

#### 3.1 Jednoduchá rekurentná neurónová sieť

Jednou z prvých navrhnutých architektúr je Elmanova jednoduchá rekurentná neurónová sieť (angl. Elman's Simple Recurrent Network, SRN) [7]. Elmanova architektúra je znázornená na obr. 4.



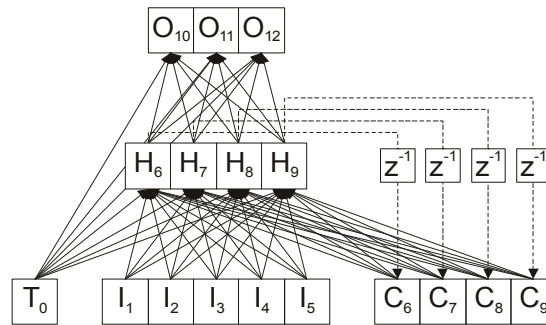
**Obr. 4.** Elmanova jednoduchá rekurentná neurónová sieť zobrazená (a) bez použitia kontextovej vrstvy a (b) s použitím kontextovej vrstvy. Časové oneskorenie je znázornené pomocou  $z^{-1}$ .

Výpočet aktivít pre skryté aj výstupné neuróny je realizovaný obdobným spôsobom, ako aj v prípade doprednej neurónovej siete. Po vypočítaní aktivít na skrytej vrstve sú tieto prekopírované na tzv. kontextovú vrstvu, z ktorej budú prostredníctvom rekurentných váh v nasledujúcom časovom kroku ovplyvňovať skryté neuróny. Takýmto spôsobom má sieť informáciu z predchádzajúceho obdobia a aktivity na kontextovej vrstve teda plnia funkciu pamäte. Je dôležité si uvedomiť, že kontextová vrstva je iba pseudovrstvou, v skutočnosti reprezentuje skrytú vrstvu z predchádzajúceho časového kroku. Rekurentné váhy na obr. 4b znázornené od kontextových ku skrytým neurónom sú v skutočnosti prepojenia zo skrytej vrstvy na seba samú (preto sa volajú rekurentné) tak, ako je uvedené na obr. 4a. Od dopredných váh sa líšia tým, že majú jednotkové časové oneskorenie, čiže sa ich prostredníctvom šíri aktivita z predchádzajúceho časového kroku.

Elmanova sieť je charakteristická tým, že pamäť je tvorená aktivitami neurónov skrytej vrstvy z predchádzajúceho časového kroku. Vo všeobecnosti ako pamäť môžu byť použité aktivity ľubovoľných neurónov a rovnako nemusia byť z bezprostredne predchádzajúceho časového kroku (môžu byť aj staršie).

### 3.2 Architektúry rekurentných neurónových sietí

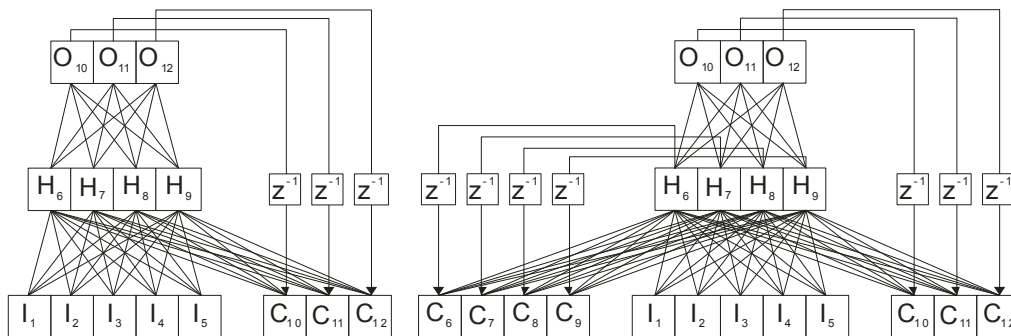
Viacere architektúry rekurentných neurónových sietí sú popísané v literatúre. Najznámejšia a najčastejšie používaná Elmanova jednoduchá rekurentná neurónová sieť už bola opísaná v predchádzajúcej časti. Pre porovnanie s ostatnými architektúrami je na obr. 5 znázornená Elmanova sieť zložená z 5 vstupných, 4 skrytých (rekurentných) a 3 výstupných neurónov.



**Obr. 5.** Elmanova jednoduchá rekurentná neurónová sieť zložená z 5 vstupných, 4 skrytých (rekurentných) a 3 výstupných neurónov. Neprerušované čiary označujú vstupné, výstupné a rekurentné váhové prepojenia a prerušované čiary s oneskorovacím členom  $z^{-1}$  označujú kopírovanie hodnôt z predchádzajúceho časového kroku.

$T_0$  je špeciálny neurón s konštantnou jednotkovou aktivitou, ktorý je pomocou váh plniacich funkciu prahov prepojený s každým skrytým a výstupným neurónom.

Na obr. 6a je znázornená tzv. Jordanova architektúra [13]. Na rozdiel od Elmanovej architektúry sú cez rekurentné prepojenia skrytej vrstvy šírené aktivity výstupnej vrstvy, a teda aktivity na výstupných neurónoch  $O_{10}$  až  $O_{12}$  prostredníctvom kontextovej vrstvy  $C_{10}$  až  $C_{12}$  tvoria pamäť neurónovej siete.

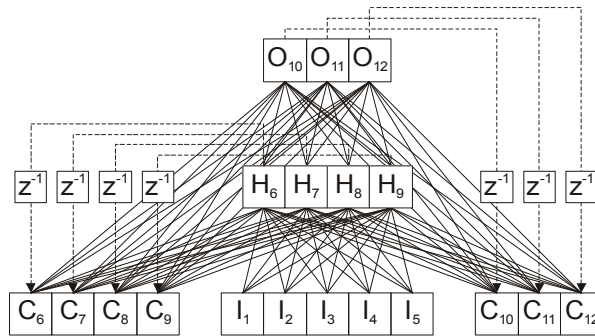


**Obr. 6.** (a) Jordanova architektúra a (b) architektúra podľa Bengia.

Táto tzv. výstupná rekurencia umožňuje pri tréovaní siete realizovať metódu „vnúteného učiaceho signálu“ (ang. Teacher Forcing). Táto metóda urýchľuje tréovanie tým, že v jeho priebehu sa na kontextovú vrstvu nekopírujú aktivity z výstupnej vrstvy, ale sa tam priamo umiestňujú požadované ideálne hodnoty. Vo všeobecnosti sa ale usudzuje, že rekurentné váhy na skrytej vrstve sú oveľa významnejšie než rekurentné váhy z výstupnej vrstvy.

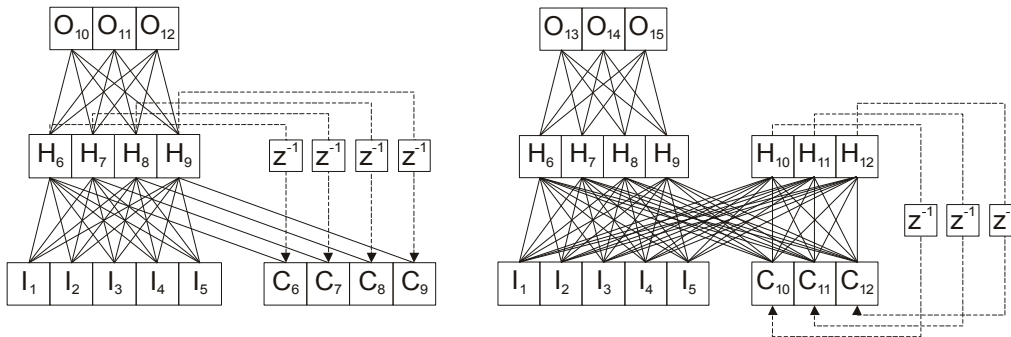
Na obr. 6b je znázornená architektúra podľa Bengia. Obsahuje ako skrytú, tak aj výstupnú rekurenciu, čiže kontextová vrstva je tvorená aj aktivitami zo skrytých neurónov ( $C_6$  až  $C_9$ ) a aj z výstupných neurónov ( $C_{10}$  až  $C_{12}$ ).

Plne prepojená rekurentná neurónová sieť Williamsa a Zipsera [37] je znázornená na obr. 7. Na rozdiel od Bengiovej architektúry prepojenia z kontextovej vrstvy smerujú aj do výstupnej vrstvy. Inak povedané, každý neurón (okrem vstupných neurónov) je prepojený cez rekurentnú váhu na každý neurón.



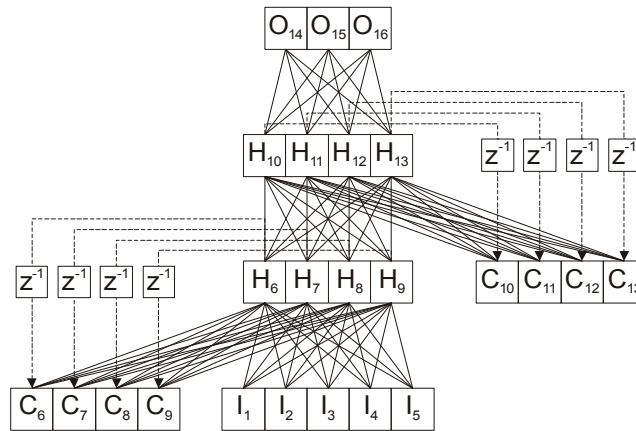
**Obr. 7.** Plne prepojená rekurentná neurónová sieť Williamsa a Zipsera.

Architektúra študovaná Frasconim, Gorim a Sodom [9] (obr. 8a) sa od Elmanovej architektúry odlišuje tým, že obsahuje iba lokálne rekurentné prepojenia. Skrytý neurón je prostredníctvom rekurentnej váhy spojený iba so sebou samým, čiže má informáciu iba o svojej predchádzajúcej aktivite. Takáto lokálna rekurencia sa považuje za nedostatočnú pre mnohé úlohy.



**Obr. 8.** (a) Architektúra podľa Frasconiho, Goriho a Sodu a (b) architektúra podľa Tiňa.

Zaujímavá je Tiňova architektúra [30] zobrazená na obr. 8b. Vyznačuje sa úplne oddelenou stavovou a asociačnou časťou siete. Asociačná časť siete je tvorená prvou skrytou vrstvou (neuróny  $H_6$  až  $H_9$ ) a výstupnou vrstvou. Stavová časť siete zodpovedajúca za vytvorenie a udržiavanie kontextovej informácie je tvorená druhou skrytou vrstvou (neuróny  $H_{10}$  až  $H_{12}$ ), ktorá je prostredníctvom rekurentných váh prepojená sama na seba (na obr. znázornené cez kontextovú vrstvu).



Obr. 9. Architektúra 5-4R-4R-3L.

Posledná prezentovaná architektúra označená podľa počtu neurónov na jednotlivých vrstvách je charakteristická dvoma skrytými vrstvami prepojenými každá sama na seba prostredníctvom rekurentných prepojení. Označenie 5-4R-4R-3L znamená, že vstupná vrstva obsahuje 5 vstupných neurónov, za ňou nasledujú dve rekurentné vrstvy každá so 4 neurónmi a nakoniec výstupná vrstva je tvorená tromi neurónmi s lineárnou aktivačnou funkciou. Tento typ architektúry bol úspešne použitý pri riešení rôznych úloh z praxe [8][25].

## 4 Gradientové tréningové algoritmy

### 4.1 Dopredné šírenie signálu

Dopredné šírenie signálu je v rekurentných neurónových sieťach realizované rovnakým spôsobom ako v prípade dopredných neurónových sietí. V rámci aktuálneho časového kroku na vstupné neuróny umiestnime aktivity zodpovedajúce aktuálnemu vstupu a postupne podľa vzťahu pre výpočet aktivity perceptrónu vypočítame aktivity neurónov. Aktivita neurónu je ovplyvňovaná cez jeho dopredné váhové prepojenie už vypočítanými aktivitami so súčasného časového kroku a cez rekurentné váhové prepojenia už vypočítanými aktivitami z predchádzajúcich časových krokov.

Formalizmus použitý na zápis algoritmov v tejto kapitole už bol predstavený pri algoritmoch dopredného šírenia a spätného šírenia chyby v doprednej sieti. Aj v rekurentnej neurónovej sieti s  $N$  neurónmi (vstupné, skryté a výstupné) je nevyhnutné usporiadať neuróny tak, aby neurón s indexom  $i$  bol cez dopredné váhové prepojenia spojený iba s neurónom  $j$  s nižším indexom  $j < i$ . Cez rekurentné prepojenia môže byť spojený aj s neurónmi s vyšším indexom. Opäť indexom 0 označíme špeciálny vstupný neurón vždy nastavený na konštantu 1 a nasledovať ho budú vstupné, skryté a nakoniec výstupné neuróny. Týmto spôsobom zabezpečíme, aby sme mohli určiť aktivity postupne od neurónu s indexom 0 až po  $N - 1$ .

Taktiež váhové prepojenia je potrebné označiť indexmi 0 až  $M - 1$ , pričom  $M$  je počet všetkých váhových prepojení. V prípade rekurentných neurónových sietí je váhové prepojenie usporiadaná štvorica  $(i, j, t, v)$ , kde  $t$  je časové oneskorenie váhového prepojenia a určuje, či je prepojenie dopredné ( $t=0$ ) alebo rekurentné ( $t>0$ ). Všetky architektúry uvedené v tretej kapitole obsahujú iba rekurentné prepojenia s  $t=1$ , čiže také, cez ktoré sa šíri aktivita iba z bezprostredne predchádzajúceho časového kroku. Architektúry s väčšími časovými oneskoreniami nie sú bežne používané. Ostatné zložky určujúce váhové prepojenie sú rovnaké ako v dopredných sieťach:  $i$  je index cieľového neurónu,  $j$  je index zdrojového neurónu a intenzita váhového prepojenia je v štvorici označená symbolom  $v$ . Na sprístupnenie jednotlivých položiek usporiadanej štvorici reprezentujúcej váhové prepojenie opäť použijeme operátory. Nech  $(i, j, t, v)$  je váhové prepojenie s indexom  $l$ , tak potom  $d_l = i, s_l = j, \tau_l = t$  a  $w_l = v$ . Množinu indexov všetkých váhových prepojení označíme  $W$ .

Použime teraz navrhnutý formalizmus na zápis dopredného šírenia signálu v ľubovoľnej rekurentnej neurónovej sieti. Štruktúra siete (architektúra) je algoritmu sprístupnená prostredníctvom operátorov  $d, s$  a  $\tau$ . Pri určovaní aktivít postupujeme od neurónov s nižším indexom k neurónom s vyšším indexom. Najskôr špeciálny neurón s indexom 0 je nastavený na konštantu 1. Na nasledujúce vstupné neuróny je umiestnený aktuálny vstup siete. Ostatné aktivity  $x_i(t)$  v časovom kroku  $t$  sú vypočítané podľa vzťahu:

$$\tilde{x}_i(t) = \sum_{\forall j \in W | d_j = i} w_j x_{s_j}(t - \tau_j), \quad (23)$$

$$x_i(t) = f(\tilde{x}_i(t)). \quad (24)$$

Zápis  $j \in W | d_j = i$  je množina indexov takých váhových prepojení, ktorých cieľový neurón je neurón s indexom  $i$ . Sú to vlastne všetky váhové prepojenia patriace neurónu  $i$ . Cez váhové prepojenie  $w_j$  sa šíri aktivita  $x_{s_j}$  zdrojového neurónu s indexom  $s_j$ . Časové oneskorenia  $\tau_j$  určuje časový krok do minulosti, ktorého aktivita je cez váhu šírená. Napr. ak je váhové prepojenie  $j$  dopredné, tak jeho  $\tau_j = 0$  a použije sa vypočítaná aktivita  $x_{s_j}(t)$  - aktivita zo súčasného kroku  $t$ . Ak je váhové prepojenie  $j$  rekurentné s časovým oneskorením 1, tak  $\tau_j = 1$  a použije sa vypočítaná aktivita  $x_{s_j}(t-1)$  - aktivita z predchádzajúceho časového kroku  $t-1$ . Aktivačná funkcia  $f$  tiež môže závisieť od neurónu, avšak uvedený zápis uvažuje rovnakú aktivačnú funkciu pre všetky neuróny v sieti.

## 4.2 Algoritmus spätného šírenia chyby

Elmanova jednoduchá rekurentná neurónová sieť bola v [7] trébovaná jednoduchým algoritmom spätného šírenia chyby, ktorý poznáme z kapitoly venujúcej sa dopredným neurónovým sieťam. Tento algoritmus nie je vo všeobecnosti vhodný na trébovanie rekurentných sietí, pretože nezohľadňuje spätné šírenie chybového signálu cez rekurentné prepojenia do predchádzajúcich časových krokov. Môže byť však úspešne použitý na jednoduchšie úlohy alebo ako referenčná metóda pre iné prístupy.

Pomocou formalizmu zavedeného pri popise dopredného šírenia uveďme vzťahy pre algoritmus spätného šírenia chyby upraveného pre rekurentné neurónové siete. Chybový signál – derivácie  $\delta_i(t) = \partial E(t) / \partial \tilde{x}_i(t)$  sú vypočítavané spätne, čiže od neurónov s vyšším indexom smerom k neurónom s nižším indexom:

$$\delta_i(t) = \left[ d_i(t) - x_i(t) + \sum_{\substack{\forall j \in W \\ |s_j = i \wedge \\ \wedge \tau_j = 0}} w_j \delta_{d_j}(t) \right] f'(\tilde{x}_i(t)). \quad (25)$$

Chyba  $d_i(t) - x_i(t)$  vzniká iba na výstupných neurónoch, a iba keď je definovaný želaný výstup  $d_i(t)$ . V opačnom prípade  $d_i(t) = x_i(t)$  a teda člen  $d_i(t) - x_i(t) = 0$ . Pri výpočte chybového signálu  $\delta_i(t)$  sa cez všetky dopredné váhové prepojenia vychádzajúce z neurónu  $i$  prešúria už vypočítané chybové signály. Indexy dopredných váhových prepojení vychádzajúcich z neurónu  $i$  sú určené zápisom  $\forall j \in W | (s_j = i) \wedge (\tau_j = 0)$ : sú to také prepojenia indexované pomocou  $j$ , ktorých zdrojový neurón je  $i$  ( $s_j = i$ ) a sú dopredné ( $\tau_j = 0$ ). Po vypočítaní chybových signálov je možné zmeniť váhy podľa:

$$\Delta w_i(t) = \alpha \delta_{d_i}(t) x_{s_i}(t - \tau_i) + \beta \Delta w_i(t - 1), \quad (26)$$

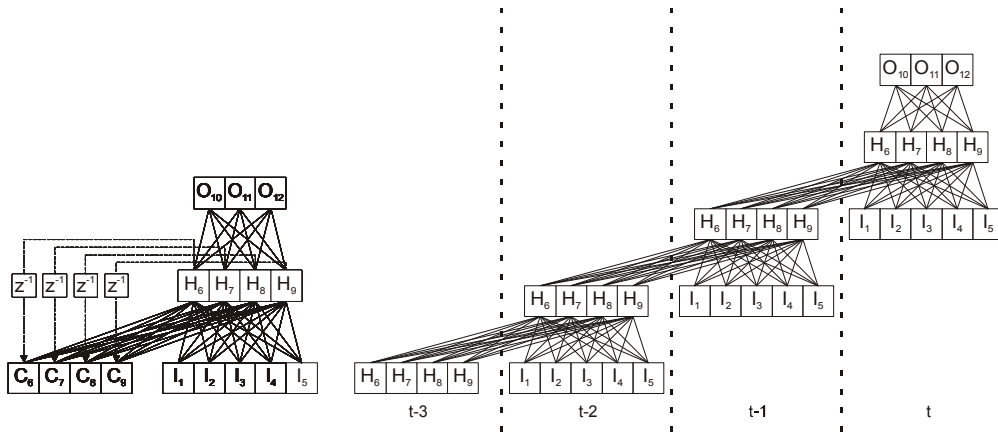
$$w_i(t + 1) = w_i(t) + \Delta w_i(t), \quad (27)$$

kde  $\alpha$  je rýchlosť učenia a  $\beta$  je hodnota určujúca momentum.

## 4.3 Algoritmus spätného šírenia chyby v čase

Rekurentné prepojenia významným spôsobom menia spôsob činnosti neurónových sietí. Aktivita neurónov sa cez rekurentné prepojenia šíri aj do nasledujúcich časových krokov a zmena váhových prepojení musí tento fakt zohľadniť. Klasické spätné šírenie mení váhy iba na základe hodnôt z aktuálneho časového kroku, aj keď chyba, ktorá vznikla v aktuálnom časovom kroku je výsledkom aktivít z predchádzajúcich časových krokov, ktoré ju prostredníctvom rekurentných váhových prepojení ovplyvnili. Riešením je šíriť chybu spätne nielen cez dopredné váhové prepojenia, ale aj cez rekurentné váhové prepojenia.

Algoritmus spätného šírenia chyby v čase (angl. Backpropagation Through Time, BPTT) umožňuje presný výpočet derivácií pre rekurentné neurónové siete. Princíp spočíva v rozvinutí rekurentnej siete v čase do potenciálne mnohovorstvej doprednej siete a v následnom použití klasického BP algoritmu. Pravdepodobne najprínosnejší príspevok týkajúci sa BPTT algoritmu je [34]. V praxi nie je rekurentná sieť rozvíjaná v čase až do času 0, ale iba istý počet krokov do minulosti označovaných ako veľkosť okna. Tomuto variantu BPTT algoritmu sa budeme venovať neskôr. Elmanova jednoduchá rekurentná neurónová sieť (obr. 10a) rozvinutá do doprednej neurónovej siete je znázornená na obr. 10b.



**Obr. 10.** (a) Elmanova jednoduchá rekurentná neurónová sieť. (b) Tá istá sieť rozvinutá v troch časových krokoch do doprednej neurónovej siete.

Uvedme teraz vzťahy pre algoritmus vychádzajúci z uvedenej myšlienky. Táto verzia algoritmu uvažuje rozvinutie rekurentnej neurónovej siete do doprednej neurónovej siete cez všetky časové kroky (pre všetky vstupy z časovej postupnosti od času  $t=0$  až do času  $t=T$ ) a následné spätné prešírenie chybového signálu a určenie zmien váh. Chybový signál  $\delta_i(t)$  je vypočítavaný spätne v čase (od časového kroku  $t=T$  po  $t=0$ ) a pre každý časový krok od neurónov s vyšším indexom smerom k neurónom s nižším indexom:

$$\delta_i(t) = \left[ d_i(t) - x_i(t) + \sum_{\substack{\forall j \in W \\ |s_j=i}} w_j \delta_{a_j}(t + \tau_j) \right] f'(\tilde{x}_i(t)). \quad (28)$$

Ak  $d_i(t)$  nie je definované, uvažujeme  $d_i(t) = x_i(t)$ , čiže na neuróne  $i$  v čase  $t$  nevzniká chyba. Chybový signál  $\delta_i(t) = 0$  pre  $t > T$ . Po vypočítaní všetkých  $\delta_i(t)$  je možné zmeniť váhy využívajúc vzťah:



$$\Delta w_i = \alpha \sum_{t=1}^T \delta_{d_i}(t) x_{s_i}(t - \tau_i), \quad (29)$$

$$w_i = w_i + \Delta w_i. \quad (30)$$

Tento variant algoritmu BPTT je obdobou tzv. dávkového tréningu (angl. Batch Learning) algoritmom BP, pri ktorom sa váhy menia až po ukončení tréningovej epochy, čiže po prezentácii všetkých vzorov z tréningovej množiny. Zmeny váh sú dané súčtom čiastkových zmien zodpovedajúcich jednotlivým vzorom. Podobne, ako dávkové tréningovanie algoritmom BP, ani tento variant algoritmu BPTT nie je vo všeobecnosti úspešný, a teda ani používaný.

Výnimku tvoria prípady, keď je tréningová množina tvorená kratšími postupnosťami, pre ktoré sú definované želané výstupné hodnoty. Napríklad úloha gramatickej inferencie spočíva v hľadaní modelu jazyka na základe pozitívnych a negatívnych príkladov slov patriacich do jazyka. Tréningová množina je tvorená reťazcami reprezentujúcimi slová, ktoré do jazyka patria (želaný výstup po prezentovaní celého slova je napr. 1) a z reťazcov, ktoré do jazyka nepatria (želaný výstup po prezentovaní slova je napr. 0). Väčšinou sú tieto slová krátke (jednotky až desiatky symbolov) a rekurentná sieť je rozvinutá do doprednej siete pre každé vstupné slovo, pričom  $T$  je dané dĺžkou tohto slova. Želaný výstup  $d_i(t)$  je teda definovaný iba v čase  $T$ . Váhové prepojenia sú následne na základe rovníc (28) až (30) zmenené.

V prípade dlhých časových postupností sa používa skrátene BPTT (angl. Truncated BPTT). Tento variant algoritmu spočíva v rozvinutí rekurentnej siete do doprednej spätne od aktuálneho času  $k=t$  do časov  $k=t-1$ ,  $k=t-2$  až  $k=t-h$ , kde  $h$  je veľkosť časového okna. Toto rozvinutie siete sa vykonáva v každom kroku  $t$  pri prechode postupnosťou od  $t=0$  až po  $T$ . Index  $t$  určuje aktuálny časový krok pri prechode postupnosťou a index  $k$  je použitý na určenie kroku v rámci siete rozvinutej v kroku  $t$ . Vo vytvorenej doprednej sieti je spätne prešírený chybový signál, ktorý ale vzniká iba v čase  $k=t$ , čiže želané výstupy  $d_i(k)$  sú definované iba v čase  $k=t$ . Chybový signál  $\delta_i(k)=0$  pre  $k>t$ . Vzťahy na výpočet chybového signálu a zmenu váh sú nasledovné:

$$\delta_i(k) = \left[ [d_i(k) - x_i(k)] + \sum_{\substack{\forall j \in W \\ |s_j=i}} w_j \delta_{d_j}(k + \tau_j) \right] f'(\tilde{x}_i(k)), \quad (31)$$

$$\Delta w_i(t) = \alpha \sum_{k=t-h}^t \delta_{d_i}(k) x_{s_i}(k - \tau_i), \quad (32)$$

$$w_i(t+1) = w_i(t) + \Delta w_i(t). \quad (33)$$

Veľkosť časového okna  $h$  sa volí 10 až 40. Z dôvodu existencie problému strácajúceho sa gradientu (angl. Vanishing Gradient Problem) [2] nemá zmysel spätne šíriť chybový signál cez viac krokov do minulosti.

#### 4.4 Rekurentné učenie v reálnom čase

Rekurentné učenie v reálnom čase (angl. Real Time Recurrent Learning, RTRL) [37] bolo navrhnuté na tréning rekurentných neurónových sietí pri úlohách vyžadujúcich priebežnú zmenu váhových prepojení v reálnom čase. Neskôr navrhnutý a v predchádzajúcej kapitole opísaný variant algoritmu BPTT – skrátene BPTT tiež umožňuje tréning váhových prepojení v reálnom čase, a to dokonca s menšími nárokmi na výpočtový čas.

RTRL je podobne ako BP a BPTT založené na gradientovej minimalizácii metódou najprudšieho spádu. Podobne ako BPTT zohľadňuje existenciu rekurentných prepojení. Rozdiel oproti algoritmu BPTT je v spôsobe výpočtu derivácií určujúcich zmeny váh. Derivácie  $\partial E/\partial w_j$  sú v RTRL vypočítané na základe derivácií  $\partial x_i/\partial w_j$ , ktoré sú vypočítavané „dopredným spôsobom“. Algoritmy BP a BPTT sú založené na spätnom šírení derivácií  $\partial E/\partial \tilde{x}_i$ , ktoré následne slúžia pre výpočet  $\partial E/\partial w_j$ . Formálne zapíšeme algoritmus RTRL pomocou nasledujúcich vzťahov:

$$\frac{\partial x_i}{\partial w_j}(t) = \left[ \delta_{id_j} x_{s_j}(t) + \sum_{\forall k \in W | d_k = i} w_k \frac{\partial x_{s_k}}{\partial w_j}(t - \tau_k) \right] f'(\tilde{x}_i(t)), \quad (34)$$

$$\Delta w_j(t) = \alpha \sum_{\forall i \in O} [d_i(t) - x_i(t)] \frac{\partial x_i}{\partial w_j}(t) + \beta \Delta w_j(t-1), \quad (35)$$

$$w_i(t+1) = w_i(t) + \Delta w_i(t). \quad (36)$$

Derivácie  $\partial x_i/\partial w_j$  sú vypočítavané postupne a dopredným spôsobom na základe už vypočítaných derivácií. Kroneckerovo delta  $\delta_{ij}$  je 1, ak  $i = j$ , inak  $\delta_{ij} = 0$ . Čiže člen  $\delta_{id_j} x_{s_j}(t)$  je vyhodnotený ako  $x_{s_j}(t)$ , čiže ako aktivita na zdrojovom neuróne váhy  $j$  iba vtedy, ak váha  $w_j$  v derivácií  $\partial x_i/\partial w_j(t)$  prislúcha neurónu  $x_i$  (vtedy  $d_j = i$  a  $\delta_{id_j} = 1$ ). Množinu indexov všetkých výstupných neurónov (neurónov, na ktorých môže vzniknúť chyba) sme označili symbolom  $O$ . Ak želané výstupné aktivity nie sú definované, uvažujeme  $d_i(t) = x_i(t)$ , a teda člen  $d_i(t) - x_i(t) = 0$ .

### 5 Tréningové algoritmy využívajúce Kalmanovu filtráciu

Medzi najúspešnejšie prístupy používané na tréning neurónových sietí patria metódy založené na Kalmanovej filtrácii [23][32]. Štandardný Kalmanov filter (KF)[15] je možné použiť v lineárnom systéme na optimálny odhad skrytého (nepozorovaného) stavu na základe pozorovaní. V prípade nelineárneho systému je možné použiť tzv. rozšírený Kalmanov filter (Extended Kalman filter, EKF), ktorý pozostáva z linearizácie nelineárnych funkcií v každom kroku a následnom použití štandardných vzťahov Kalmanovej filtrácie.

V prípade neurónovej siete je odhadovaný stavový vektor stotožnený s trénovanými váhami siete. Na výpočet derivácií pre EKF je možné použiť algoritmus podobný BPTT (EKF-BPTT algoritmus). Hlavnou nevýhodou EKF je jeho vysoká výpočtová náročnosť. Boli navrhnuté viaceré aproximačné prístupy redukujúce náročnosť výpočtu. V tomto texte predstavíme pravdepodobne najpoužívanejšiu modifikáciu znižujúcu výpočtovú náročnosť, a to tzv. oddelený Kalmanov filter. Taktiež sa budeme venovať viacprúdovému Kalmanovmu filtru, ktorý umožňuje dosiahnuť robustné a stabilné riešenie v mnohých praktických aplikáciách.

### 5.1 Kalmanov filter

Kalmanov filter je súbor vzťahov popisujúcich rekurzívne riešenie diskrétnoho lineárneho problému filtrovania. Je to efektívne riešenie problému odhadu neznámeho stavu ovplyvneného šumom metódou najmenších štvorcov [21].

Predpokladajme nasledujúci systém riadený podľa nasledujúcich vzťahov. Prvá lineárna stochastická diferenčná rovnica sa nazýva stavová rovnica, a popisuje zmenu stavu v priebehu diskrétnych časových krokoch:

$$\mathbf{x}(t) = \mathbf{F}(t)\mathbf{x}(t-1) + \mathbf{w}(t). \quad (37)$$

Stav systému  $\mathbf{x}(t)$  je lineárnou transformáciou predchádzajúceho stavu, je určený aplikovaním transformácie reprezentovanou známou prechodovou maticou  $\mathbf{F}(t)$  na predchádzajúci stav systému  $\mathbf{x}(t-1)$  a výsledok je zaťažený bielym gausovským šumom  $\mathbf{w}(t)$ . Druhá lineárna rovnica sa nazýva rovnica pozorovania a popisuje, ako sú na základe stavu systému vypočítané pozorovania:

$$z(t) = \mathbf{H}(t)\mathbf{x}(t) + \mathbf{v}(t). \quad (38)$$

Získané pozorovanie alebo výstup systému  $z(t)$  je určený aplikovaním známej výstupnej matice  $\mathbf{H}(t)$  na stav systému. Výsledok je opäť zaťažený bielym gausovským šumom  $\mathbf{v}(t)$ . Matice  $\mathbf{F}(t)$  a  $\mathbf{H}(t)$  sa v čase môžu meniť. Taktiež poznáme parametre prechodového  $\mathbf{w}(t)$  a výstupného šumu  $\mathbf{v}(t)$ , ich kovariančné matice  $\mathbf{Q}(t)$  a  $\mathbf{R}(t)$  majú všetky hodnoty okrem diagonálnych nulové:

$$\mathbf{Q}(t) = E[\mathbf{w}(t)\mathbf{w}(t)^T], \quad (39)$$

$$\mathbf{R}(t) = E[\mathbf{v}(t)\mathbf{v}(t)^T]. \quad (40)$$

Šumy  $\mathbf{w}$  a  $\mathbf{v}$  sú nekorelované s nulovou strednou hodnotou. Cieľom Kalmanovej filtrácie je vyfiltrovať čo najpresnejší odhad  $\hat{\mathbf{x}}(t)$  stavu systému  $\mathbf{x}(t)$ . Zdefinujme si chybu odhadu stavu  $\mathbf{e}(t)$  a kovarianciu odhadu chyby  $\mathbf{P}(t)$ :

$$\mathbf{e}(t) = \mathbf{x}(t) - \hat{\mathbf{x}}(t), \quad (41)$$

$$\mathbf{P}(t) = E[\mathbf{e}(t)\mathbf{e}(t)^T]. \quad (42)$$

Kalmanov filter funguje v dvojkrokovom cykle. Prvý krok je úprava v novom časovom kroku a je to jednoduchá predikcia  $\hat{\mathbf{x}}^-(t)$  z už odhadnutého stavu  $\hat{\mathbf{x}}(t-1)$  a kovariancie chyby  $\mathbf{P}^-(t)$  z existujúceho odhadu  $\mathbf{P}(t-1)$ . Získaný odhad, v ktorom ešte nie je zahrnuté výstupné pozorovanie, nazývame apriórny odhad a entity označujeme pomocou  $^-$ .

$$\hat{\mathbf{x}}^-(t) = \mathbf{F}(t)\hat{\mathbf{x}}^-(t-1), \quad (43)$$

$$\mathbf{P}^-(t) = \mathbf{F}(t)\mathbf{P}(t-1)\mathbf{F}(t)^T + \mathbf{Q}(t). \quad (44)$$

Druhý krok je úprava na základe pozorovania  $\mathbf{z}(t)$  a vykonáva opravu odhadnutého apriórneho stavu  $\hat{\mathbf{x}}^-(t)$  a kovariancie chyby  $\mathbf{P}^-(t)$ . Najskôr je vypočítaný tzv. Kalmanov zisk  $\mathbf{K}(t)$ :

$$\mathbf{K}(t) = \mathbf{P}(t)\mathbf{H}(t)^T [\mathbf{H}(t)\mathbf{P}(t)\mathbf{H}(t)^T + \mathbf{R}(t)]^{-1}, \quad (45)$$

a následne je možné vypočítať tzv. aposteriórny odhad stavu  $\mathbf{x}(t)$  a  $\mathbf{P}(t)$ :

$$\mathbf{P}(t) = \mathbf{P}^-(t) - \mathbf{K}(t)\mathbf{H}(t)\mathbf{P}^-(t), \quad (46)$$

$$\hat{\mathbf{x}}(t) = \hat{\mathbf{x}}^-(t) - \mathbf{K}(t)[\mathbf{z}(t) - \mathbf{H}(t)\hat{\mathbf{x}}^-(t)]. \quad (47)$$

Rovnice (43) až (47) popisujú jeden cyklus Kalmanovho filtra, v ktorom sa spresňuje odhad stavu na základe jediného pozorovania. Po príchode ďalšieho pozorovania v ďalšom časovom kroku  $\mathbf{z}(t+1)$  sa vykoná ďalší cyklus Kalmanovej filtrácie.

## 5.2 Rozšírený Kalmanov filter

Aplikovanie KF na nelineárne problémy môže byť vykonané niekoľkými spôsobmi. Pravdepodobne najpriamočiarejšie je aplikovanie KF na systém aproximovaný v každom kroku pomocou Taylorovho rozvoja. Taylorov rozvoj nám umožňuje aproximovať zodpovedajúco krát derivovateľnú funkciu  $f$  v okolí bodu  $a$  pomocou polynómu:

$$f(x) \approx f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \dots \quad (48)$$

Tento spôsob aplikovania KF na nelineárne systémy sa nazýva rozšírený Kalmanov filter (angl. Extended Kalman filter, EKF).

Uvažujme nasledujúcu stavovú rovnicu a rovnicu pozorovania:

$$\mathbf{x}(t) = f(t, \mathbf{x}(t-1)) + \mathbf{w}(t), \quad (49)$$

$$\mathbf{z}(t) = h(t, \mathbf{x}(t)) + \mathbf{v}(t), \quad (50)$$

kde  $f$  a  $h$  sú nelineárne funkcie. Linearizáciou týchto rovníc je možné odvodiť nasledujúce rovnice úprav na základe času:

$$\hat{\mathbf{x}}^-(t) = f(t, \hat{\mathbf{x}}^-(t-1)), \quad (51)$$

$$\mathbf{P}^-(t) = \mathbf{F}(t)\mathbf{P}(t-1)\mathbf{F}(t)^T + \mathbf{Q}(t), \quad (52)$$

a rovnice úprav na základe pozorovania:

$$\mathbf{K}(t) = \mathbf{P}(t)\mathbf{H}(t)^T [\mathbf{H}(t)\mathbf{P}(t)\mathbf{H}(t)^T + \mathbf{R}(t)]^{-1}, \quad (53)$$

$$\mathbf{P}(t) = \mathbf{P}^-(t) - \mathbf{K}(t)\mathbf{H}(t)\mathbf{P}^-(t), \quad (54)$$

$$\hat{\mathbf{x}}(t) = \hat{\mathbf{x}}^-(t) - \mathbf{K}(t)[\mathbf{z}(t) - h(t, \hat{\mathbf{x}}^-(t))], \quad (55)$$

kde  $\mathbf{F}(t)$  a  $\mathbf{H}(t)$  sú matice derivácií reprezentujúce prvý (lineárny) člen Taylorovho rozvoja:

$$\mathbf{F}(t) = \frac{\partial f(t, \hat{\mathbf{x}}(t))}{\partial \mathbf{x}}, \quad (56)$$

$$\mathbf{H}(t) = \frac{\partial h(t, \hat{\mathbf{x}}(t))}{\partial \mathbf{x}}. \quad (57)$$

Z uvedeného vyplýva, že tento prístup je možné použiť iba pre derivovateľné funkcie  $f$  a  $h$ , čo je ale v prípade neurónovej siete splnené.

### 5.3 Prispôsobenie rozšíreného Kalmanovho filtra na tréning dopredných aj rekurentných neurónových sietí

Tréning viacvrstvovej neurónovej siete zloženej z perceptrónov môže byť transformované na problém optimálneho filtrovania [36]. Neurónová sieť môže byť popísaná ako nelineárny systém:

$$\mathbf{w}(t) = \mathbf{w}(t-1), \quad (58)$$

$$\mathbf{z}(t) = h(\mathbf{w}(t), \mathbf{u}(t)) + \mathbf{v}(t), \quad (59)$$

pričom neznámy stav systému  $\mathbf{w}(t)$  je vektor hodnôt váhových prepojení neurónovej siete. Váhy v natréinovanej sieti sa nemenia, a teda prechodová matica má tvar  $\mathbf{F}(t) = \mathbf{I}$ , kde  $\mathbf{I}$  je matica identity. Pozorovanie  $\mathbf{z}(t)$  sú želané výstupné hodnoty siete a funkcia pozorovania  $h()$  je nelineárna funkcia. Jej argumentmi sú váhy siete  $\mathbf{w}(t)$  a vstupy  $\mathbf{u}(t)$ . Matica  $\mathbf{H}(t)$  je vypočítaná v každom časovom kroku  $t$  ako:

$$\mathbf{H}(t) = \frac{\partial h(\hat{\mathbf{w}}(t), \mathbf{u}(t))}{\partial \mathbf{x}}, \quad (60)$$

kde  $\hat{\mathbf{w}}(t)$  je odhad vektora hodnôt váhových prepojení neurónovej siete, čiže aktuálne váhy siete v priebehu procesu tréovania. Vektor  $\mathbf{w}(t)$  sú ideálne váhy siete, ktoré sa snažíme Kalmanovou filtráciou nájsť.

Sformulujme teda nakoniec rovnice rozšíreného Kalmanovho filtra aplikovaného na tréovanie neurónových sietí [4]. Vo vzťahoch už použijeme symboly  $\mathbf{w}$ ,  $\mathbf{d}$  a  $\mathbf{o}$ , ktoré sme používali na označenie intenzít váhových prepojení, želaných výstupov a skutočných výstupov neurónovej siete. Jeden krok tréovania neurónovej siete je opísaný nasledujúcimi vzťahmi:

$$\mathbf{K}(t) = \mathbf{P}(t-1)\mathbf{H}(t)^T [\mathbf{H}(t)\mathbf{P}(t-1)\mathbf{H}(t)^T + \mathbf{R}(t)]^{-1}, \quad (61)$$

$$\mathbf{w}(t) = \mathbf{w}(t) - \mathbf{K}(t)[\mathbf{d}(t) - \mathbf{o}(t)], \quad (62)$$

$$\mathbf{P}(t) = \mathbf{P}(t-1) - \mathbf{K}(t)\mathbf{H}(t)\mathbf{P}(t-1) + \mathbf{Q}(t). \quad (63)$$

Počet všetkých váh v sieti bude označený ako  $n_w$  a počet výstupných neurónov ako  $n_o$ . Vektor  $\mathbf{w}$  s dĺžkou  $n_w$  je vektor všetkých váh neurónovej siete (vstupných, výstupných, rekurentných, prahových, ...). Jakobián, matica  $\mathbf{H}$  s rozmermi  $n_o \times n_w$ , je vypočítaný v každom časovom kroku. V riadkoch obsahuje derivácie zodpovedajúcej výstupnej aktivity k všetkým váham. Tieto derivácie môžu byť vypočítané skoro rovnakými rutinami ako sú algoritmy RTRL alebo BPTT. Matica  $\mathbf{P}$  s rozmermi  $n_w \times n_w$  je matica kovariancie chyby obsahujúca kovariancie zodpovedajúce všetkým možným párom váh. Kalmanov zisk  $\mathbf{K}$ , matica s rozmermi  $n_w \times n_o$  slúži na úpravu váh  $\mathbf{w}$  na základe rozdielu medzi želaným výstupom  $\mathbf{d}$  a aktuálnym výstupom siete  $\mathbf{o}$ . Matica  $\mathbf{R}$  s rozmermi  $n_o \times n_o$  reprezentuje šum pozorovania a podobne ako rýchlosť učenia v gradientových algoritmov riadi rýchlosť konverencie EKF. Vyššie hodnoty šumu zodpovedajú vyššej miere neurčitosti kladenej na rozdiel  $\mathbf{d}(t) - \mathbf{o}(t)$ , čo vedie k pomalšiemu učeniu.  $\mathbf{Q}$  s rozmermi  $n_w \times n_w$  je kovariančná matica prechodového šumu. Malé hodnoty prechodového šumu sú stále uvažované, nenulové hodnoty zlepšujú konvergenciu filtra.

Hlavnou nevýhodou tohto prístupu je jeho vysoká výpočtová náročnosť. Boli navrhnuté viaceré aproximačné prístupy redukujúce náročnosť výpočtu inverznej matice uvažovaním iba diagonálnej, resp. blokovo diagonálnej matice  $\mathbf{P}$ . Tento prístup je označovaný ako oddelený EKF.

#### 5.4 Oddelený rozšírený Kalmanov filter

Princíp tréovania pomocou tzv. „oddeleného“ EKF (angl. Decoupled EKF) [26] pozostáva z rozdelenia váh do navzájom disjunktných skupín, pričom informácia o kovariancii chyby je udržiavaná iba medzi váhami v skupine. Najčastejšie sú v skupine váhové prepojenia prislúchajúce rovnakému cieľovému neurónu. Takýto prístup výrazne redukuje výpočtové nároky algoritmu. Ak má neurónová sieť  $N$  neurónov (neuvažujeme vstupné neuróny), derivácie výstupu siete voči váham

smerujúcich do neurónu  $i$  v čase  $t$  sú umiestené v matici  $\mathbf{H}_i(t)$ ,  $i \in \{1, 2, \dots, N-1\}$ . Matica kovariancie chyby pre váhy neurónu  $i$  je  $\mathbf{P}_i(t)$  a matica kovariancie šumu je  $\mathbf{R}_i(t)$ . Rovnice oddeleného EKF sú:

$$\mathbf{A}(t) = \left[ \sum_{i=1}^N \mathbf{H}_i(t) \mathbf{P}_i(t-1) \mathbf{H}_i(t)^T + \mathbf{R}_i(t) \right]^{-1}, \quad (64)$$

$$\mathbf{K}(t) = \mathbf{P}(t-1) \mathbf{H}(t)^T \mathbf{A}(t), \quad (65)$$

$$\mathbf{w}(t) = \mathbf{w}(t-1) - \mathbf{K}(t) [\mathbf{d}(t) - \mathbf{o}(t)], \quad (66)$$

$$\mathbf{P}(t) = \mathbf{P}(t-1) - \mathbf{K}(t) \mathbf{H}(t) \mathbf{P}(t-1) + \mathbf{Q}(t). \quad (67)$$

Hoci oddelený EKF umožňuje realizovať výpočet s menšou výpočtovou zložitosťou, klasický EKF s „plnou“ maticou  $\mathbf{P}$  môže viesť k lepším riešeniam a pri súčasnej dostupnosti zariadení s vyšším výpočtovým výkonom môže byť pre väčšinu úloh vhodnejší.

## 5.5 Viacprúdový rozšírený Kalmanov filter

Významnou a úspešnou modifikáciou štandardného EKF je tréovanie rekurentných sietí na viacerých „trénovacích prúdoch“ – viacprúdový EKF (Multistream EKF, MS-EKF) [8]. MS-EKF bol úspešne použitý na tréovanie rekurentných sietí na zložitých reálnych problémoch z priemyselného prostredia [24]. Princíp pozostáva z tréovania viacerých verzií tej istej siete na rôznych častiach tréovacej postupnosti, pričom každá verzia zdieľa tie isté hodnoty váhových prepojení, ale udržiava si vlastný stav siete. Tréovacia množina je teda rozdelená do viacerých „prúdoch“ a viaceré vstupno-výstupné vzory sú zohľadnené v jednom časovom kroku. Klasické tréovanie pomocou EKF s jediným „prúdom“ môže byť vhodné na časovej postupnosti s homogénnou dynamikou. Na modelovanie heterogénnych postupností s dynamikou meniacou sa v čase je vhodnejšie použiť viacprúdový EKF.

Viacprúdovým EKF je tiež možné natréovať sieť na spracovanie viacerých navzájom nezávislých úloh a natréovaný model môže byť jednoducho aplikovaný na riešenie ľubovoľnej z nich. Bežná sekvenčná prezentácia takýchto postupností na sieť tréovanú jednoprúdovým EKF môže viesť k nežiaducemu efektu, keď sa model príliš špecializuje na aktuálne prezentovaný vzor a „zabúda“ už natréované vlastnosti umožňujúce kvalitné spracovanie dávnejšie prezentovaných vzorov (angl. Recency Effect).

Pre tréovanie pomocou MS-EKF je tréovacia množina rozdelená na niekoľko častí, čo už v mnohých konkrétnych situáciách môže byť zabezpečené. Napríklad tréovacia množina môže pozostávať z podmnožín získaných za rôznych pracovných podmienok alebo v rôznych pracovných módoch sledovaného procesu. Pre tréovanie pomocou MS-EKF je teda potrebné vytvoriť  $N_s$  tréovacích množín rovnakej dĺžky, kde  $N_s$  je počet prúdov viacprúdového EKF. Každá z tréovacích množín je spracovávaná jednou inštanciou neurónovej siete a v každom časovom kroku je

spracovávaných  $N_s$  tréningových vzorov. Pre každý prúd je propagácia dopredného signálu realizovaná rovnakým spôsobom ako v  $N_s$  nezávislých sieťach. Taktiež derivácie výstupov sietí voči váham sú vypočítané nezávisle a je na to možné použiť BPTT či RTRL algoritmus. Proces tréningovania by mohol ďalej pokračovať  $N_s$  nezávislými EKF krokmi a vypočítaním celkových zmien váhových prepojení priemerovaním jednotlivých čiastkových zmien. Avšak v prípade MS-EKF sú už vypočítané derivácie združené do jedinej globálnej matice  $\mathbf{H}(t)$ . Označme  $\mathbf{H}_i(t)$  maticu derivácií výstupu voči váham pre neurónovú sieť prislúchajúcu prúdu  $i$  v čase  $t$ . Potom matica  $\mathbf{H}(t)$  môže byť vyjadrená ako  $\mathbf{H}(t) = [\mathbf{H}_1(t), \mathbf{H}_2(t), \dots, \mathbf{H}_{N_s}(t)]$ . Rovnakým spôsobom aj vektor želaných hodnôt  $\mathbf{d}(t)$  a vektor vypočítaných výstupných hodnôt  $\mathbf{o}(t)$  sú zostavené z čiastkových vektorov zodpovedajúcich jednotlivým prúdom:  $\mathbf{d}(t)^T = [\mathbf{d}_1(t)^T, \mathbf{d}_2(t)^T, \dots, \mathbf{d}_{N_s}(t)^T]$ ,  $\mathbf{o}(t)^T = [\mathbf{o}_1(t)^T, \mathbf{o}_2(t)^T, \dots, \mathbf{o}_{N_s}(t)^T]$ . Okrem týchto zmien prebieha Kalmanova filtrácia klasickým už opísaným spôsobom (rovnice (61) až (63)). Len pripomeňme, že váhové prepojenia sú zdieľané všetkými inštanciami siete.

## 5.6 Kalmanove filtre so sigma bodmi a duálna Kalmanova filtrácia

Vyššie spomenuté algoritmy patria medzi najmodernejšie prístupy a reprezentujú súčasnú úroveň vedeckého poznania v oblasti rekurentných neurónových sietí. Existuje množstvo variantov a modifikácií Kalmanovej filtrácie a mnohé sú študované v kontexte tréningovania rekurentných neurónových sietí. V krátkosti spomeňme dva zaujímavé prístupy.

Kalmanove filtre so sigma bodmi (angl. Sigma Point Kalman Filter) [14][22] spočívajú v presnejšom šírení kovariancie chýb cez jednotlivé kroky Kalmanovej filtrácie. Namiesto linearizácie sú vhodne zvolené body transformované cez nelineárny systém a matica kovariancie chýb je v ďalšom kroku odhadnutá na základe týchto transformovaných bodov.

Ďalším zaujímavým prístupom použiteľným na tréningovanie rekurentných neurónových sietí je tzv. duálna Kalmanova filtrácia (angl. Dual EKF) [6][33]. Jej princíp spočíva v súbežnom odhadovaní stavu aj parametrov systému dvojicou Kalmanových filtrov. V kontexte tréningovania neurónových sietí je jeden z filtrov používaný na adaptáciu váhových prepojení a druhý na odhad aktivít neurónov.

## 6 Dynamika rekurentných neurónových sietí

V tejto kapitole sa zameriame na dynamiku rekurentnej neurónovej siete, čiže jej správanie v čase. Špeciálne sa budeme venovať dynamike nenatrénovanej siete inicializovanej s malými váhami, pretože zaujímavé vlastnosti tejto dynamiky sú úspešne používané vo viacerých v súčasnosti populárnych prístupoch. Dynamické



správanie siete budeme pre jednoduchosť študovať na úlohách predikcie nasledujúceho symbolu.

## 6.1 Modelovanie postupností symbolov

Modelovanie postupností symbolov ma v oblasti rekurentných neurónových sietí dlhú históriu. Najznámejšia a najčastejšie používaná architektúra – Elmanova jednoduchá rekurentná neurónová sieť bola v [7] navrhnutá na študovanie jednoduchých lingvistických postupností. Práve v oblasti kognitívnej vedy sú rekurentné neurónové siete používané na modelovanie jednoduchých aj zložitejších jazykových štruktúr [18].

V kontexte modelovania postupností symbolov sú rekurentné neurónové siete často používané na predikovanie nasledujúceho symbolu v postupnosti [3]. Úlohou siete je po prezentácii časti postupnosti vyhodnotiť, aké symboly môžu nasledovať. Pre názornosť si uveďme nasledujúci príklad.

Nech je daná gramatika  $G = (T, N, P, A)$ . Množina terminálnych symbolov  $T = \{a, b, c, s\}$  je množina symbolov, z ktorých môžu byť vytvorené slová jazyka. Množina neterminálnych symbolov je  $N = \{A, B, C\}$  a je to množina pomocných symbolov, ktoré môžu byť použité iba pri generovaní slov. Množina pravidiel  $P$  je tvorená nasledujúcimi jednoduchými pravidlami:

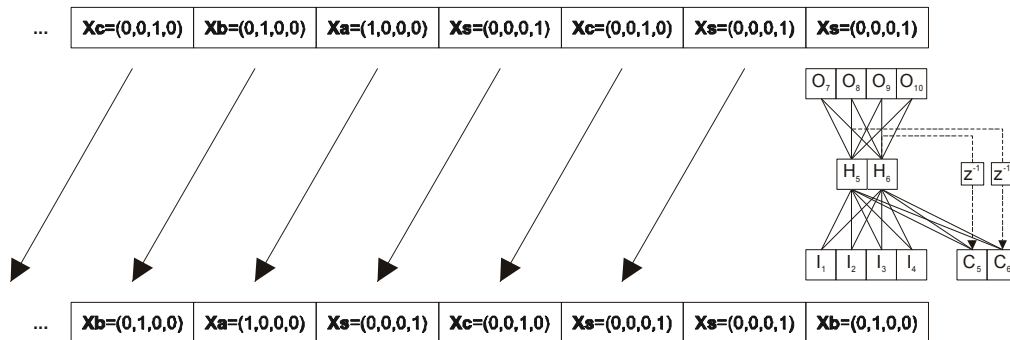
$$\begin{aligned} A &\rightarrow sA \mid bB \mid \varepsilon \\ B &\rightarrow sB \mid cC \quad . \\ C &\rightarrow sC \mid aA \end{aligned} \quad (68)$$

Štartovacím neterminálom je symbol  $A$  a  $\varepsilon$  je prázdne slovo. Táto gramatika generuje slová, v ktorých nasledujú za sebou symboly v poradí  $a$ ,  $b$  a  $c$ , medzi ktorými môže byť na ľubovoľnom mieste a ľubovoľne mnohokrát symbol  $s$ . Príklady slov patriacich do jazyka generovaných touto gramatikou sú napríklad:  $bca$ ,  $sbcssas$ ,  $sssbscasbcssa$  a pod. Postupnosť pre úlohu predikcie vytvoríme zretazením veľkého počtu slov generovaných touto gramatikou takým spôsobom, že pravidlá, ktoré je možné pri generovaní v danom okamihu použiť, sú vybrané s rovnakou pravdepodobnosťou. Takýmto spôsobom bola vytvorená postupnosť začínajúca reťazcom  $bsscsabcsssssabsssscassa...$

Takto vytvorená postupnosť je symbol po symbole prezentovaná rekurentnej neurónovej sieti, ktorej úlohou je predikovať nasledujúci symbol. Táto situácia je znázornená na obr. 11. Je zrejmé, že symboly nie je možné jednoznačne predikovať, pretože napr. po symbole  $b$  môže nasledovať  $c$  aj  $s$ , avšak nemôže nasledovať ani  $b$  ani  $a$ .

Pre riešenie úlohy sme skonštruovali jednoduchú rekurentnú neurónovú sieť, zvolili sme 2 skryté neuróny. Tento počet nám na jednej strane umožní sledovať aktivity na skrytých neurónoch v klasickom dvojrozmernom grafe, a tiež vzhľadom na jednoduchosť úlohy je plne postačujúci na jej zvládnutie. Vstupy sú tvorené 4 symbolmi  $a, b, c, s$  a pre ich zakódovanie do vstupného vektora neurónovej siete bolo

zvolené kódovanie tzv. „jeden z viacerých“ (angl. One Hot Encoding). V tomto spôsobe kódovania každému vstupu prislúcha práve jeden vstupný neurón, ktorý je aktívny (nastavený na 1) práve vtedy, keď je sieti prezentovaný daný vstup. Inak je pasívny, teda nastavený na 0. Rovnaký spôsob kódovania sme zvolili aj na výstupe siete, a teda neurónová sieť mala 4 vstupné a 4 výstupné neuróny. Skryté aj výstupné neuróny boli samozrejme vybavené prahmi a použili sme sigmoidálnu aktivačnú funkciu pre skryté a aj výstupné neuróny.



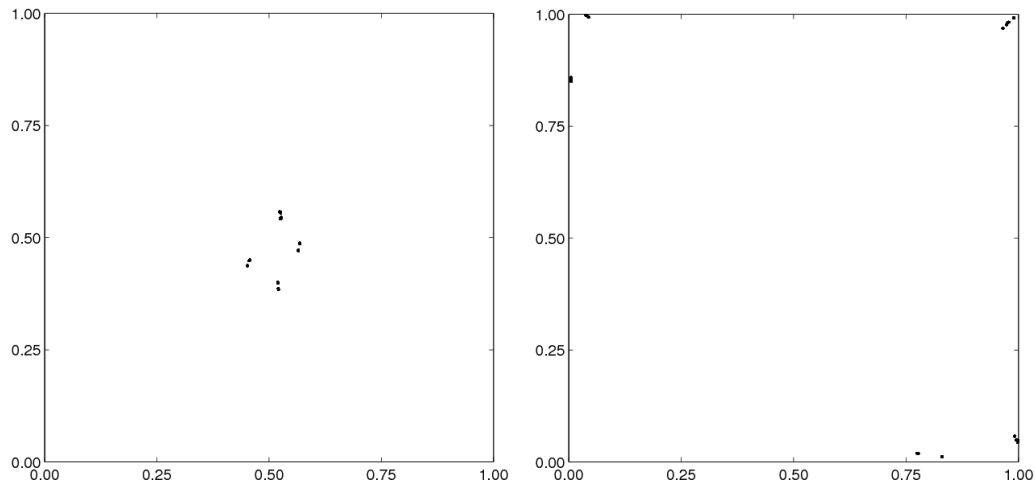
**Obr. 11.** Predikcia nasledujúceho symbolu riešená pomocou Elmanovej jednoduchšej rekurentnej neurónovej siete.

Sieť bola inicializovaná malými váhami z intervalu  $(-0.5, 0.5)$  a následne bola trébovaná algoritmom BPTT s veľkosťou okna  $h=10$ , rýchlosťou učenia  $\alpha=0.1$ , momentum nebolo použité. Trébovacia postupnosť bola vytvorená vyššie uvedeným spôsobom a obsahovala 1000 symbolov. Postupnosť bola sieti prezentovaná 10 krát, a pred každým cyklom boli vynulované aktivity na skrytých neurónoch (kontextová vrstva bola inicializovaná na nulové hodnoty). Po natrébovaní bolo pozorované, že sieť je schopná s vysokou presnosťou predpovedať nasledujúce symboly. V každom kroku postupnosť môže pokračovať jedným z dvoch symbolov. Natrébovaná sieť na výstupných neurónoch zodpovedajúcich symbolom, ktorými postupnosť mohla pokračovať, správne umiestňovala aktivitu 0.5, čo môže byť považované za pravdepodobnosť pokračovania postupnosti príslušným symbolom.

Aby sme zistili, akým spôsobom rekurentná sieť danú úlohu rieši, je potrebné analyzovať štruktúru jej stavového priestoru. Pod stavom siete budeme rozumieť aktivity na skrytých neurónoch. Na správne riešenie úlohy si totiž sieť musí vytvárať také aktivity na skrytých neurónoch, aby bola na jednej strane schopná produkovať želaný výstup v aktuálnom časovom kroku, a na druhej strane si v nich tiež musí zapamätať informáciu o kontexte, aby správne vyprodukovala nasledujúce skryté aktivity (aj v závislosti od ďalšieho vstupu samozrejme).

Aktivity na skrytých neurónoch si v priebehu prechodu postupnosťou môžeme zaznamenať a tieto aktivity – stavy siete – môžeme umiestniť ako body do grafu. Osi grafu zodpovedajú aktivitám na dvoch skrytých neurónoch. Zobrazenie stavového priestoru sme spravili nielen pre natrébovanú sieť, ale aj pre nenatrébovanú náhodne inicializovanú sieť a výsledky sú zobrazené na obr. 12. V nasledujúcich častiach

analyzujeme najskôr dynamiku natrénovanej a potom aj nenatrénovanej rekurentnej neurónovej siete.



**Obr. 12.** Reprezentácia stavového priestoru nenatrénovanej (a) a natrénovanej (b) Elmanovej jednoduchej rekurentnej neurónovej siete s 2 skrytými neurónmi vytvorená zaznamenaním aktivít skrytých neurónov pri spracovaní postupnosti symbolov.

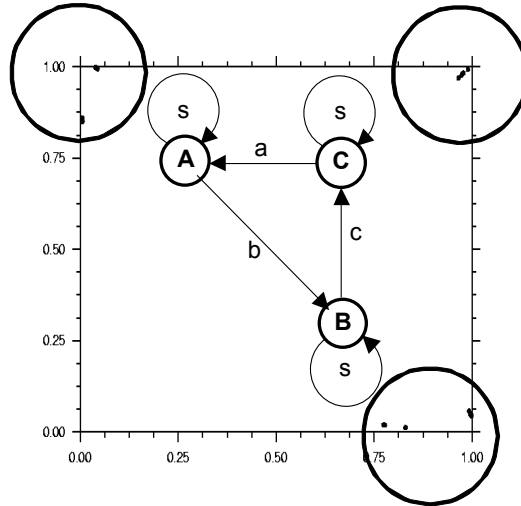
## 6.2 Dynamika natrénovanej rekurentnej neurónovej siete

Pozrime sa najskôr na dynamiku natrénovanej rekurentnej neurónovej siete. Z obr. 12b je zrejmé, že aktivity skrytej vrstvy sa koncentrujú v troch zhluchoch v rohoch priestoru  $(0,0,1,0) \times (0,0,1,0)$ . Po analýze prechodov medzi aktivitami skrytej vrstvy (reprezentovanými bodmi v grafe) v závislosti na vstupoch bolo možné stotožniť zhluky bodov so stavmi konečného automatu, ktorý si sieť trénovaním vytvorila.

Na obr. 13 je tento automat znázornený a je zrejmé, že zodpovedá gramatike, ktorou bola postupnosť symbolov vygenerovaná. Tri zhluky v rohoch grafu tvoria stavy automatu tak, ako je na obrázku naznačené. Ak je napr. stav siete v jednom z týchto zhlukov a sieť je prezentovaný symbol  $s$ , nasledujúci stav siete bude stále v tom istom zhluku. Ak je sieti prezentovaný jeden z troch symbolov  $a$ ,  $b$  alebo  $c$ , stav siete, čiže aktivity skrytej vrstvy, budú v zhluku zodpovedajúcom príslušnému stavu automatu. Takýmto spôsobom si rekurentná sieť udržiava všetku potrebnú kontextovú informáciu. Vytvorila si model časovej postupnosti, ktorý jej umožňuje predpovedať nasledujúci symbol s vysokou presnosťou.

Tento príklad demonštruje schopnosť rekurentných neurónových sietí fungovať ako konečné stavové automaty, a tým modelovať regulárne jazyky. Stavová časť siete v takomto prípade slúži na zapamätanie si konečného počtu stavov. Rekurentné siete sú ale schopné modelovať aj zložitejšie jazykové štruktúry, ktoré si vyžadujú zložitejšiu pamäť. Takýmto sú napríklad bezkontextové jazyky, ktoré môžu byť modelované

zásobníkovými automatmi. Uvedme si príklad, ako je rekurentná neurónová sieť schopná si vytvoriť v svojej stavovej časti dynamické správanie modelujúce zásobník.



**Obr. 13.** Konečný stavový automat vytvorený v stavovej časti Elmanovej jednoduchšej rekurentnej neurónovej siete

Uvažujme jednoduchú bezkontextovú gramatiku generujúcu známy jazyk  $a^n b^n$ . Formálne je gramatika definovaná takto:  $G = (T, N, P, S)$ , čiže množina terminálnych symbolov je  $T = \{a, b\}$ , množina neterminálnych symbolov je  $N = \{S\}$  a množina pravidiel  $P$  je tvorená dvojicou pravidiel:

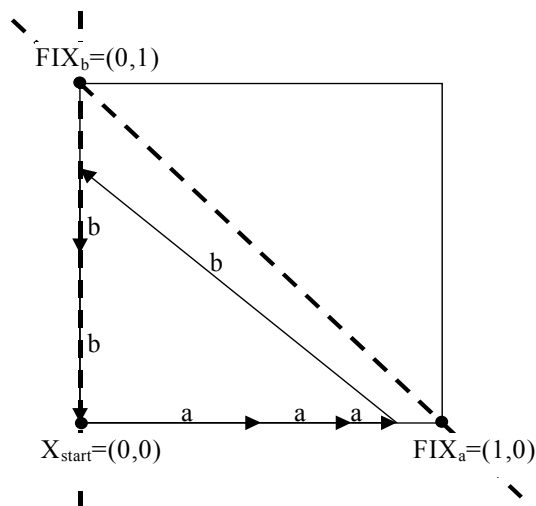
$$S \rightarrow aSb \mid \varepsilon. \quad (69)$$

Touto gramatikou je možné vygenerovať slová  $ab, aabb, aaaaabbbbb$  a pod. Postupnosť pre úlohu predikcie vytvoríme rovnakým spôsobom ako v predchádzajúcom príklade a sieť opäť použijeme na riešenie úlohy predikcie. Je vhodné si uvedomiť, že je možné jednoznačne predikovať symboly až v druhej časti slova, po prvom symbole  $b$ , a síce je možné predikovať všetky ostatné symboly  $b$ , ktorých musí byť spolu práve toľko, koľko bolo symbolov  $a$  v prvej časti slova. Pre správne fungovanie je nevyhnutné vytvoriť si mechanizmus umožňujúci počítanie symbolov.

Na obrázku obr. 14 je znázornený princíp dynamického správania natrénovanej rekurentnej neurónovej siete [35]. Rekurentné a aj vstupné váhy boli v priebehu tréningu upravené do takých hodnôt, ktoré spôsobili vytvorenie bodového atraktora pre vstupný symbol  $a$  a sedlového bodu pre vstupný symbol  $b$ . Dynamika pre vstupné symboly  $a$  a  $b$  je daná vzťahmi:

$$\begin{aligned} \omega_a(x_1, x_2) &= (0.5x_1 + 0.5, 0) \\ \omega_b(x_1, x_2) &= (0.2x_1 + 2x_2 - 1) \end{aligned} \quad (70)$$

kde  $x_1$  a  $x_2$  sú aktivity na skrytých neurónoch,  $\omega_a$  je transformácia skrytých aktivít, ak je sieti prezentovaný symbol  $a$  a  $\omega_b$  je transformácia stavu siete zodpovedajúca symbolu  $b$ . Transformácie sú v sieti vytvorené pomocou ideálne nastavených rekurentných a vstupných váhových prepojení a tiež pre jednoduchosť uvažujeme lineárnu aktivačnú funkciu.



**Obr. 14.** Zásobník vytvorený v stavovej časti rekurentnej neurónovej siete pomocou bodového atraktora a sedlového bodu.

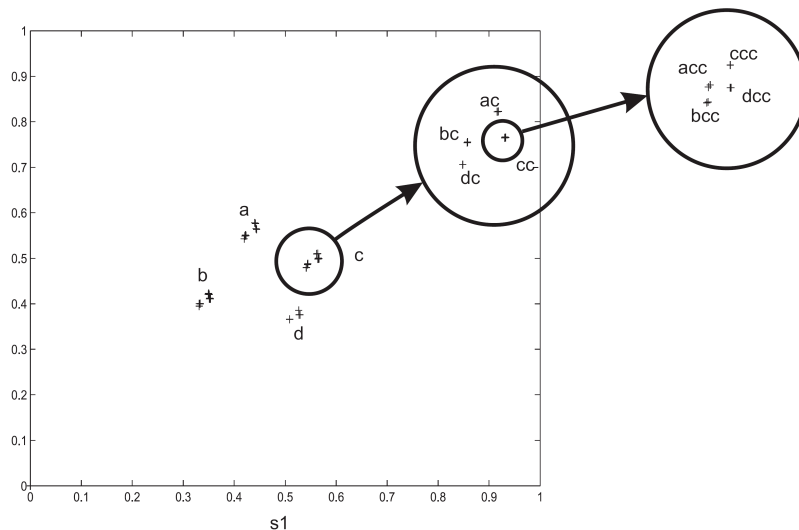
Transformácia  $\omega_a$  umiestňuje stav siete smerom k bodu  $(1.0,0.0)$ , čo je bodový atraktor pre túto transformáciu. Nekonečne veľa zreťazení tejto transformácie, čiže nekonečne veľa symbolov  $a$  postupne prezentovaných sieti, by sieť uviedlo do stavu  $(1.0,0.0)$ . Čím viac symbolov  $a$  je prezentovaných sieti, tým bližšie je stav siete k bodovému atraktoru. Takto si sieť v aktivite  $x_1$  pamätá, koľko symbolov  $a$  jej bolo prezentovaných. Transformácia  $\omega_b$  vytvára sedlový bod  $(0.0,1.0)$ , ktorý „priťahuje“ stavy zo smeru  $(1.0,0.0)$  a „odpuďzuje“ stavy v smere  $(0.0,0.0)$ . Počet symbolov  $a$  prezentovaných v sieti sa pri prvej prezentácii symbolu  $b$  premietne do aktivity  $x_2$  a následné prezentácie symbolu  $b$  ju znižujú rovnakou mierou, akou ju transformácia  $\omega_a$  zvyšovala. Takto sieť presne vie, koľko symbolov  $b$  má nasledovať za symbolmi  $a$ . Sieť si vlastne takýmto spôsobom vytvorila počítadlo, transformácia  $\omega_a$  počítadlo inkrementuje a  $\omega_b$  dekrementuje, sieť simuluje jednoduchý počítadlový automat.

Vidíme, že rekurentné neurónové siete dokážu modelovať postupnosti s regulárnymi a bezkontextovými prvkami. V [1][27] bolo ukázané, akým spôsobom je rekurentná neurónová sieť schopná vo svojom stavovom priestore vytvárať aj pamäťové štruktúry nevyhnutné na spracovanie kontextových jazykov.

Uvedené príklady vo veľkej jednoduchosti približujú, akým spôsobom rekurentná neurónová sieť môže riešiť problémy. V reálnych rekurentných neurónových sieťach s väčším počtom skrytých neurónov môže byť dynamika veľmi komplexná a jej analýza je zvyčajne náročná.

### 6.3 Dynamika nenatrérovanej rekurentnej neurónovej siete inicializovanej s malými váhami

V predchádzajúcej podkapitole sme naznačili, že v natrérovanej rekurentnej sieti je možné pozorovať dynamické správanie rôznej komplexnosti. Ale aj dynamika nenatrérovanej rekurentnej neurónovej siete inicializovanej s malými váhami má veľmi zaujímavé a potenciálne využiteľné vlastnosti [5][20][31].



**Obr. 15.** Organizácia stavového priestoru nenatrérovanej rekurentnej neurónovej siete inicializovanej malými váhami. Sieť je prezentovaná postupnosť vytvorená nad abecedou zloženou zo štyroch symbolov.

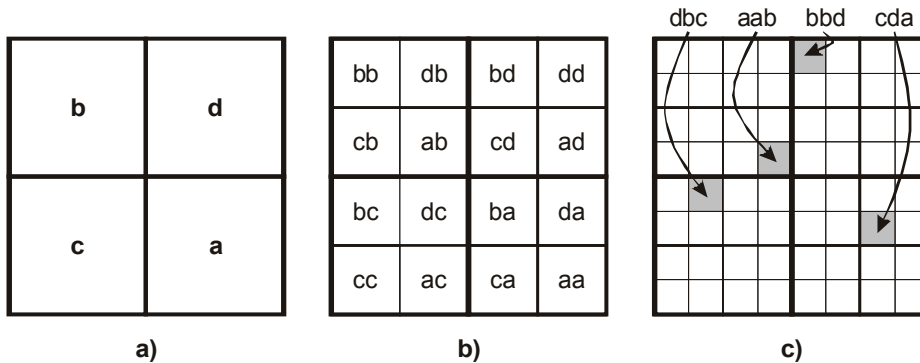
Pozrime sa bližšie na stavový priestor reprezentovaný grafom na obr. 12a. Hoci je sieť nenatrérovaná, jej stavový priestor vykazuje vysokú mieru štruktúrovanosti [16]. Stav sieti – aktivity skrytých neurónov sú organizované do zhlukov, pričom ich umiestnenie v stavovom priestore je dané postupnosťou prezentovanej sieti. Bližšie si vysvetlíme organizáciu stavového priestoru pomocou obr. 15. V stavovom priestore môžeme pozorovať 4 zhluky, pričom každý prislúcha práve jednému zo štyroch symbolov, z ktorých bola vytvorená vstupná postupnosť. Ak je rekurentnej sieti prezentovaný symbol, stav sieti sa bude nachádzať v oblasti zodpovedajúcej tomuto symbolu, a teda zo stavu sieti je možné jednoznačne určiť, ktorý symbol bol sieti prezentovaný. Ak sa ale bližšie pozrieme na zhluk zistíme, že má veľmi podobnú štruktúru ako celý stavový priestor, tiež sa skladá zo 4 podzhlukov. Ako je na obrázku

naznačené, pozícia v rámci tohto zhluku je určená predposledným symbolom prezentovaným sieti. Ak by sme sa bližšie pozreli na niektorý z podzhlukov, opäť by sme mohli pozorovať podobnú štruktúru.

Pre dôslednejšie pochopenie uvedenej vlastnosti použijeme už z predchádzajúcej časti známy formalizmus. Zmeny stavu siete budeme modelovať pomocou systému transformácií  $\Omega = \{\omega_a, \omega_b, \omega_c, \omega_d\}$ , kde jednotlivé transformácie reprezentujú zmenu stavu pri vstupnom symbole  $a$  až  $d$ :

$$\begin{aligned} \Omega &= \{\omega_x : X \rightarrow X, x \in \{a, b, c, d\}\} \\ \omega_a(x_1, x_2) &= (0.5x_1 + 0.5, 0.5x_2) \\ \omega_b(x_1, x_2) &= (0.5x_1, 0.5x_2 + 0.5) \\ \omega_c(x_1, x_2) &= (0.5x_1, 0.5x_2) \\ \omega_d(x_1, x_2) &= (0.5x_1 + 0.5, 0.5x_2 + 0.5) \end{aligned} \quad (71)$$

Kontraktívne transformácie  $\omega_x$  pracujú nad priestorom  $X = (0.0, 1.0)^2$ , ktorý reprezentuje stavový priestor rekurentnej siete. Kontraktívnosť znamená, že transformácia zobrazí stavový priestor do jeho časti, formálne  $\omega_x(X) \in X$ . Ak takéto transformácie zreťazíme, výsledný stav bude odzrkadľovať poradie transformácií tak, ako je naznačené na obr. 16. Pozícia v stavovom priestore určuje, aké transformácie boli vykonané, resp. ktoré symboly boli sieti prezentované. Požiadavka na kontraktívnosť je pre rekurentné siete zabezpečená tým, že siete sú skoro vždy inicializované s malými váhami, ako aj tým skryté neuróny sú zvyčajne vybavené aktivačnou funkciou, ktorej obor hodnôt je ohraničený.



**Obr. 16.** Mapovanie stavového priestoru podľa vykonaných transformácií. a) Rozdelenie stavového priestoru podľa poslednej vykonanej transformácie, b) podľa posledných dvoch a c) posledných troch transformácií.

Teoreticky je takýmto spôsobom možné určiť všetky transformácie/vstupy ľubovoľne ďaleko do minulosti, avšak v konkrétnej implementácii nás zvyčajne limituje presnosť počítačovej reprezentácie reálneho čísla. Avšak siete si aj tak takýmto

spôsobom pamätá aj desiatky symbolov do minulosti. Príklad vývoja stavu po prezentácií konkrétnej postupnosti je znázornený na obr. 17.

Schopnosť nenatrérovanej rekurentnej neurónovej siete mapovať vo svojom stavovom priestore históriu vstupov sme v [31] nazvali Markovovská architekturná predispozícia (angl. Markovian Architectural Bias) rekurentných neurónových sietí, pretože je možné nájsť podobnosť medzi takouto organizáciou stavového priestoru a kontextami Markovových modelov s premenlivou dĺžkou pamäti (angl. Variable Length Markov Models) [19][28].

Prezentované symboly: **b c c c c a**

Počiatočný stav: (0.5,0.5).

$$\bar{x}_0 = (0.5, 0.5)$$

$$\bar{x}_1 = \omega_b(\bar{x}_0) = (0.25, 0.75)$$

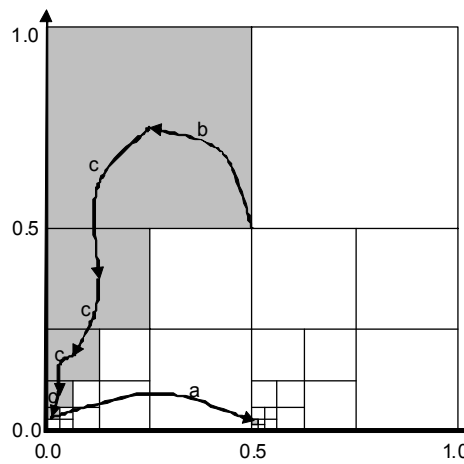
$$\bar{x}_2 = \omega_c(\bar{x}_1) = (0.125, 0.375)$$

$$\bar{x}_3 = \omega_c(\bar{x}_2) = (0.0625, 0.1875)$$

$$\bar{x}_4 = \omega_c(\bar{x}_3) = (0.03125, 0.09375)$$

$$\bar{x}_5 = \omega_c(\bar{x}_4) = (0.015625, 0.046875)$$

$$\bar{x}_6 = \omega_a(\bar{x}_5) = (0.5078125, 0.0234375)$$



**Obr. 17.** Mapovanie symbolov v stavovom priestore siete. Postupne, ako sú symboly siete prezentované, je stav siete transformovaný do zodpovedajúcich oblastí.

Takéto správanie nenatrérovovaných rekurentných neurónových sietí inicializovaných malými váhami je úspešne využívané viacerými modelmi. Najznámejším sú v súčasnosti pomerne populárne siete s echo stavmi.

## 7 Siete s echo stavmi

Siete s echo stavmi (ang. Echo State Networks, ESNs) [11][12] predstavujú nový zaujímavý prístup v oblasti rekurentných neurónových sietí a v súčasnosti patria medzi modely, ktorých vlastnosti sú aktívne študované. ESN siete sú založené na vlastnostiach náhodne inicializovaných RNN sietí, ktoré sú schopné špecifickým spôsobom reprezentovať postupnosť vstupov vo svojej skrytej vrstve. ESN sieť je v podstate klasická RNN sieť Elmanovského typu obsahujúca skrytú vrstvu zloženú z veľkého počtu náhodne poprepájaných neurónov. Táto vrstva sa v ESN sieti nazýva dynamický rezervoár. Výstupné neuróny sú použité na extrakciu zaujímavých vlastností z dynamického rezervoára. Váhové prepojenia skrytých neurónov nie sú modifikované, trérovaniu podliehajú iba výstupné neuróny. Významným prínosom ESN architektúry je možnosť využiť výpočtovo nenáročný algoritmus lineárnej regresie na adaptáciu váhových prepojení výstupných neurónov.



## 7.1 Architektúra

Architektúra ESN siete je znázornená na obr. 18. ESN sieť sa skladá z klasických vstupných, skrytých a výstupných neurónov. Charakteristickou črtou ESN sietí je dynamický rezervoár. Obsahuje stovky až tisíce neurónov, pričom tieto sú zvyčajne rekurentnými váhami iba riedko poprepájané. Neuróny v dynamickom rezervoári sú klasické perceptróny so sigmoidálnou aktivačnou funkciou  $f(x) = 1/(1 + e^{-x})$ , popr. sa používa funkcia  $f(x) = \tanh(x)$ , ktorá má rovnaký tvar ako sigmoida a interval reálnych čísel mapuje do otvoreného intervalu  $(-1.0, 1.0)$ .

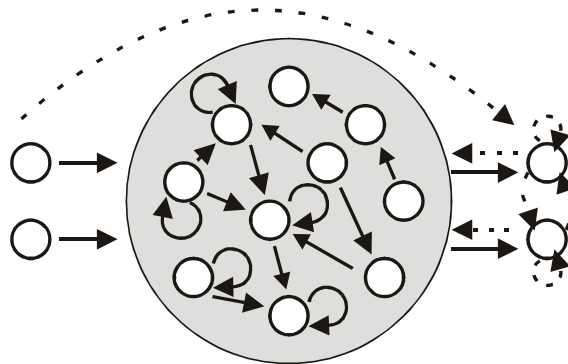
Opišme formálne fungovanie ESN siete. Nech  $\mathbf{u}(t)$  je vstupný vektor v časovom kroku  $t$ , aktivity na skrytých neurónov podliehajú nasledujúcemu vzťahu:

$$\mathbf{x}(t) = f(\mathbf{W}^{\text{IN}} \cdot \mathbf{u}(t) + \mathbf{W} \cdot \mathbf{x}(t-1) + \mathbf{W}^{\text{BACK}} \cdot \mathbf{y}(t-1)), \quad (72)$$

kde  $f$  je aktivačná funkcia skrytého neurónu,  $\mathbf{W}$  je matica rekurentných váh,  $\mathbf{W}^{\text{IN}}$  je matica vstupných váh a  $\mathbf{W}^{\text{BACK}}$  je matica tzv. spätných váhových prepojení smerujúcich z výstupných neurónov na skryté neuróny. Aktivity na výstupných neurónoch je možné vypočítať podľa:

$$\mathbf{y}(t) = f(\mathbf{W}^{\text{OUT}} [\mathbf{u}(t)^T, \mathbf{x}(t)^T, \mathbf{y}(t-1)^T]^T), \quad (73)$$

kde  $\mathbf{W}^{\text{OUT}}$  je matica váhových prepojení smerujúca zo vstupných, skrytých a výstupných neurónov na výstupné neuróny.



**Obr. 18.** Architektúra ESN siete. Prepojenia naznačené prerušovanými čiarami nemusia byť použité.

Úlohou dynamického rezervára je produkovať potenciálne zaujímavé správanie, ktoré je možné jednoduchým spôsobom pomocou výstupných váhových prepojení použiť na vytvorenie želaných výstupných aktivít. Aktivity na neurónoch rezervoára sú odozvou vstupov ovplyvňujúcich rezervoár prostredníctvom vstupných váhových prepojení. Prípadne aj výstupné aktivity môžu cez tzv. spätné váhové prepojenia ovplyvniť rezervoár podobným spôsobom ako vstupné aktivity.

Podstatnou podmienkou fungovania ESN siete je existencia tzv. „echo“ stavov v stavovom priestore siete. Od siete sa vyžaduje, aby jej stav bol „ozvenou“ vstupov prezentovaných sietí. Ak je táto podmienka splnená, na dosiahnutie vysokovýkonnej siete býva často postačujúca iba adaptácia výstupných váhových prepojení (samozrejme v závislosti na riešenej úlohe). Zvyčajne je potrebné skonštruovať sieť s mohutným a bohatým dynamickým rezervoárom, zloženým zo stoviek neurónov.

Vlastnosť existencie echo stavov znamená, že pre každý skrytý neurón  $x_i$  existuje tzv. echo funkcia  $e_i$  taká, že aktuálny stav  $x_i(t)$  neurónu môže byť vyjadrený ako  $x_i(t) = e_i(u(t), u(t-1), u(t-2), \dots)$  [11]. ESN sieť využíva kontraktívnu dynamiku náhodne inicializovaného a netrénovaného dynamického rezervoára. Aktuálny vstup  $u(t)$  prezentovaný sietí má väčší vplyv na stav siete ako vstup prezentovaný v skoršom časovom kroku. Vplyv vstupu postupne s časom a s prichádzajúcim ďalšími vstupmi slabne a rovnaká dostatočne dlhá postupnosť vstupov  $u(t), u(t-1), u(t-2), \dots$  uvedie sieť do rovnakého stavu bez ohľadu na iniciálny stav siete a predchádzajúce vstupy.

## 7.2 Trénovanie

Trénovanie ESN sietí spočíva v nastavení váhových prepojení smerujúcich do výstupných neurónov. Váhové prepojenia medzi vstupnými a skrytými neurónmi a rekurentné váhové prepojenia nepodliehajú trénovaniu. Pri trénovaní siete teda nie je potrebné šíriť chybovú informáciu cez rekurentné prepojenia, čo výrazne znižuje výpočtovú náročnosť trénovania. Napriek prípadnej nelinearite spôsobenej použitím výstupných neurónov s nelineárnou aktivačnou funkciou je zvyčajne možné na trénovanie ESN sietí použiť algoritmy lineárnej regresie.

Nevyhnutným predpokladom pre úspešné použitie takéhoto jednoduchého a rýchleho spôsobu adaptácie je vhodne nastavený dynamický rezervoár. Dynamika siete musí byť dostatočne bohatá na to, aby bolo možné z aktivít skrytých neurónov vytvoriť požadované výstupné aktivity. Táto požiadavka je zvyčajne riešená konštruovaním rezervoára s veľkým počtom neurónov. Neuróny v rezervoári sú zvyčajne riedko poprepájané, čím sa zvyšuje pravdepodobnosť, že ich výstupné aktivity budú rozdielne. Významnými parametrami rezervoára je teda jeho veľkosť (počet neurónov v skrytej vrstve, rádovo 10 – 1000 neurónov) a pravdepodobnosť existencie rekurentného váhového prepojenia (0.1 – 10 %). Ďalšou významnou a potrebnou vlastnosťou rezervoára je existencia už spomínaných „echo“ stavov. Aby mala ESN sieť potrebné vlastnosti, musí mať kontraktívnu dynamiku, alebo, inak povedané, jej rekurentné váhy musia byť patrične „malé“. Zaujímavé vlastnosti rekurentných sietí inicializovaných s malými váhami sme opísali v predchádzajúcej časti. Kontraktívnosť je v ESN sieti zvyčajne dosiahnutá tým, že matica rekurentných váh je nastavená tak, aby mala požadovaný spektrálny polomer. Spektrum matice nazývame jej vlastné čísla a spektrálny polomer matice je maximum z absolútnych hodnôt jej vlastných čísel. Parametre sa zvyčajne určujú experimentálne na základe expertnej skúsenosti a výber vhodnej siete je realizovaný bežnými technikami, ako je napríklad krosvalidácia.

Cieľom tréovania je nájdenie výstupných váhových prepojení, čiže určenie matice  $\mathbf{W}^{\text{OUT}}$ . Hodnoty ostatných váhových prepojení, čiže hodnoty v maticiach  $\mathbf{W}^{\text{IN}}$ ,  $\mathbf{W}$  a  $\mathbf{W}^{\text{BACK}}$  sú určené pri inicializácii siete a nie sú v priebehu tréovania modifikované. Tréovanie spočíva v prezentovaní vstupnej postupnosti ESN siete (vstupy  $\mathbf{u}(t)$  pre  $t=1$  až  $T$ ) a zaznamenaní vypočítaných skrytých aktivít do matice  $\mathbf{X}$ , kde stĺpce matice tvoria vektory  $\mathbf{v}(t)$  vytvorené zreťazením vektorov  $\mathbf{u}(t), \mathbf{x}(t), \mathbf{y}(t-1)$ , čiže  $\mathbf{v}(t)^T = [\mathbf{u}(t)^T, \mathbf{x}(t)^T, \mathbf{y}(t-1)^T]$ . Je dôležité, aby výstupné neuróny boli vybavené prahovým váhovým prepojením, čiže aby každý vstupný vektor  $\mathbf{u}(t)$  obsahoval aj špeciálny vstup neustále nastavený na hodnotu 1. Želané výstupné aktivity  $\hat{\mathbf{y}}(t)$  určujú maticu  $\mathbf{Y}$ . Ak výstupné neuróny majú lineárnu aktivačnú funkciu, tak stĺpce matice  $\mathbf{Y}$  sú tvorené priamo požadovanými výstupmi  $\hat{\mathbf{y}}(t)$ , v opačnom prípade, pretože je aktivačná funkcia invertovateľná, sú stĺpce  $\mathbf{Y}$  tvorené vektormi vypočítanými ako  $f^{-1}(\hat{\mathbf{y}}(t))$ . Nájdenie matice  $\mathbf{W}^{\text{OUT}}$  spočíva vo vyriešení rovnice:

$$\mathbf{Y} = \mathbf{W}^{\text{OUT}} \cdot \mathbf{X}. \quad (74)$$

Rovnica zvyčajne nemá riešenie a na nájdenie aproximatívneho riešenia je možné použiť metódu najmenších štvorcov:

$$\mathbf{W}^{\text{OUT}} = \mathbf{Y} \cdot \mathbf{X}^{\oplus}, \quad (75)$$

kde  $\mathbf{X}^{\oplus}$  je pseudoinverzná matica, ktorá môže byť vypočítaná ako:

$$\mathbf{X}^{\oplus} = \mathbf{X}^T \cdot (\mathbf{X} \cdot \mathbf{X}^T)^{-1}. \quad (76)$$

Na tréovanie ESN siete je tiež možné použiť aj iteratívnu metódu, a to rekurzívnu metódu najmenších štvorcov (angl. Recursive Least Squares, RLS), ktorá má menšie nároky na pamäť. Úprava váh sa potom riadi podľa vzťahov:

$$\mathbf{k}(t) = \frac{\mathbf{P}(t-1)\mathbf{v}(t)}{\mathbf{v}(t)^T \mathbf{P}(t-1)\mathbf{v}(t) + \gamma}, \quad (77)$$

$$\mathbf{P}(t) = \gamma^{-1} [\mathbf{P}(t-1) - \mathbf{k}(t)\mathbf{v}(t)^T \mathbf{P}(t-1)], \quad (78)$$

$$\mathbf{W}^{\text{OUT}}(t) = \mathbf{W}^{\text{OUT}}(t-1) + \mathbf{k}(t)[\hat{\mathbf{y}}(t) - \mathbf{y}(t)]^T, \quad (79)$$

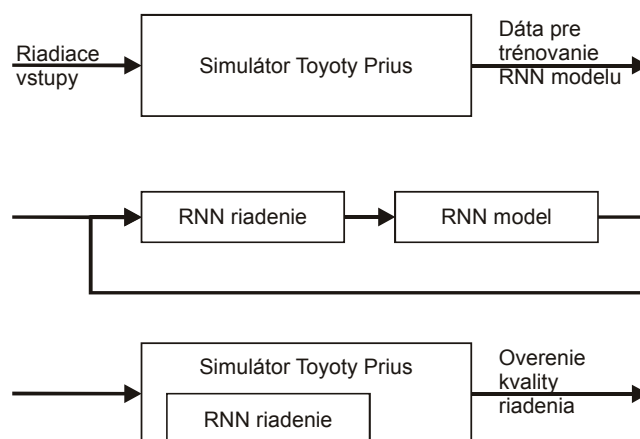
kde  $\mathbf{k}(t)$  je inovačný vektor vypočítaný v každom časovom kroku  $t$ . Vektor  $\mathbf{v}(t)$  je vytvorený zreťazením vektorov  $\mathbf{u}(t)$ ,  $\mathbf{x}(t)$  a  $\mathbf{y}(t-1)$ . Vektor želaných a vektor vypočítaných výstupných aktivít sú označené  $\hat{\mathbf{y}}(t)$  a  $\mathbf{y}(t)$ .  $\mathbf{P}(t)$  je kovariančná matica chýb odhadu váh inicializovaná vysokými hodnotami na diagonále a je upravovaná v každom kroku. Parameter  $\gamma$  je inicializovaný na hodnotu menšiu alebo rovnú 1 a určuje mieru zabúdania.

## 8 Aplikácie rekurentných neurónových sietí

Rekurentné neurónové siete boli navrhnuté na riešenie mnohých úloh z praxe. Súčasnú úroveň poznania budeme demonštrovať aplikáciou Elmanovej rekurentnej neurónovej siete adaptovanou viacprúdovým Kalmanovým filtrom na riešenie úlohy riadenia hybridného pohonu automobilu Toyota Prius [24]. Na druhej strane ESN siete patria v súčasnosti medzi modely, ktorých vlastnosti sa zatiaľ iba intenzívne študujú. Jednou z navrhnutých aplikácií ESN sietí je modelovanie nelineárneho bezdrôtového komunikačného kanála [12].

### 8.1 Riadenie hybridného pohonu Toyota Prius

Hybridné pohony sú čoraz populárnejšie, pretože predstavujú možnosť, ako výrazným spôsobom zlepšiť úspornosť prevádzky automobilov a zredukovať škodlivé emisie. Stratégia riadenia automobilu vybaveného hybridným pohonom je podstatne zložitejšia, ako v prípade klasických automobilov vybavených iba spaľovacími motormi. Hlavnou úlohou riadenia pohonu je určovanie, aká časť energie bude dodávaná z elektrického motoru a aká časť zo spaľovacieho motoru. Prítom je nevyhnutné minimalizovať spotrebu paliva a emisií, avšak je nevyhnutné zabezpečiť potrebnú pojazdnosť a spoľahlivosť, a tiež je potrebné brať do úvahy úroveň nabitia batérie. Viaceré odborné články poukázali na možnosť výrazného zlepšenia efektivity prevádzky hybridného vozidla voľbou vhodného riadenia pohonu [24].



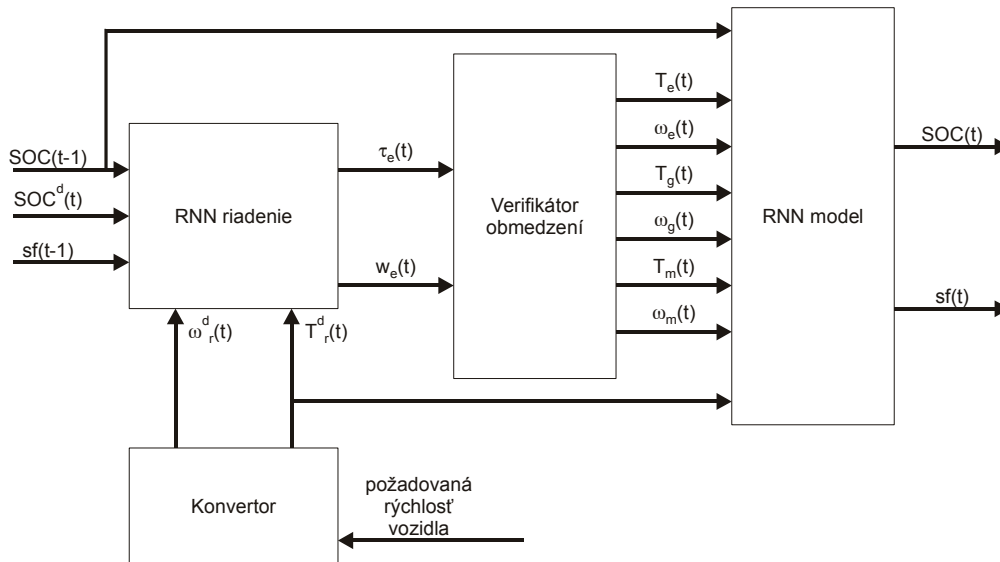
**Obr. 19.** Kroky tréningovania riadenia pohonu. Simulátor automobilu je najskôr použitý na vytvorenie dát pre natréningovanie modelu pohonovej časti automobilu. Potom je takto vytvorený model použitý na tréningovanie riadenia pohonu. Nakoniec je kvalita vytvoreného riadenia pohonu overená v simulátore automobilu. Obrázok je vytvorený podľa [24].

V tradičných schémach hybridného riadenia pohonu sú spaľovací a elektrický pohon zapojené sériovo alebo paralelne. Toyota Prius využíva inovatívny spôsob prepojenia pohonov, v ktorom je spaľovací motor prepojený s elektrickým pohonom a generátorom pomocou planetárnej prevodovky. Úlohou riadenia je udržať spaľovací

pohon v optimálnom režime s vysokým výkonom a nízkymi emisiami. Hoci je efektívnosť riadenia pohonu v automobile Toyota Prius vysoká, v [24] poukázali na možnosť jeho podstatného zlepšenia využitím rekurentnej neurónovej siete.

Pri navrhovaní riadiaceho člena realizovaného pomocou rekurentnej neurónovej siete bol využitý simulátor Toyoty Prius. Tento simulátor je distribuovaný komplexný systém, a tak jeho priame použitie je sťažené. Simulátor bol teda použitý na vytvorenie modelu simulujúceho podstatné vlastnosti automobilu z hľadiska riadenia hybridného pohonu. Tento model bol tiež vytvorený pomocou rekurentnej neurónovej siete. Kroky tréningu riadiaceho člena sú znázornené na obr. 19.

Model pohonu bol vytvorený rekurentnou neurónovou sieťou s 25 neurónmi. Tréningová množina pozostávala z údajov získaných počas 20 jazd vykonaných na simulátore. Rekurentná sieť bola tréningovaná viacprúdovým Kalmanovým filtrom a tréningovanie trvalo 3000 epoch. Časť siete bola tréningovaná na modelovanie spotreby a druhá časť na modelovanie stavu batérie elektrického pohonu.



**Obr. 20.** Blokový diagram systému na tréningovanie riadiacej časti pohonu využívajúcej rekurentnú neurónovú sieť. Symboly  $\omega$  a  $w$  označujú rýchlosti a  $\tau$  a  $T$  krútiace momenty,  $sf$  spotrebu paliva a  $SOC$  je stav nabitia batérie. Symbol  $d$  ako horný index označuje požadované hodnoty. Symboly  $e$ ,  $g$ ,  $m$  a  $r$  ako dolné indexy označujú premenné patriace elektrickému pohonu, generátoru, spaľovaciemu motoru a celkovému pohonu. Obrázok je vytvorený podľa [24].

Aj radiaca časť pohonu bola vytvorená pomocou Elmanovej rekurentnej neurónovej siete tvorenej 5 vstupmi, 10 skrytými a 2 výstupnými neurónmi. Vstupom do siete je požadovaná rýchlosť jazdy  $\omega_r^d$  a požadovaný krútiaci moment  $T_r^d$ , predchádzajúca spotreba spaľovacieho motora  $sf$ , predchádzajúci stav nabitia batérie  $SOC$  a želaný stav nabitia batérie  $SOC^d$ . Výstupom siete sú dva riadiace signály, a to

krútiaci moment spaľovacieho motora  $\tau_e$  a jeho rýchlosť  $w_e$ , z ktorých sa po kontrole obmedzení stane  $T_e$  a  $\omega_e$ . Bloková schéma tréovania je zobrazená na obr. 20.

Cieľom tréovania je vytvoriť také riadenie, ktoré bude minimalizovať spotrebu sf. Druhým, nemenej významným, cieľom je zachovať stav nabitia batérie v bezpečnej zóne definovanej ako  $SOC > 0.4$ . Stav nabitia batérie SOC je hodnota z intervalu  $\langle 0.0, 1.0 \rangle$ , kde 0.0 reprezentuje úplne vybitú batériu a 1.0 maximálne nabitú batériu. Tieto dva ciele boli skombinované do účelovej funkcie:

$$\text{cost}(t) = \lambda_1 \text{sf}(t)^2 + \lambda_2(t) (SOC^d(t) - SOC(t))^2, \quad (80)$$

kde parameter  $\lambda_1 = 1$  a parameter  $\lambda_2 = 10$  ak  $SOC(t) \geq SOC^d(t)$  a  $\lambda_2 = 50$  ak  $SOC(t) < SOC^d(t)$ .

Aj na tréovanie tejto siete autori použili viacprúdový Kalmanov filter a derivácie určovali algoritmom BPTT s oknom do minulosti nastaveným na 20. Každý prúd v jednom tréovacom cykle spracovával 50 hodnôt získaných z referenčnej jazdy, ktorej iniciálne hodnoty boli určené náhodne. Aj  $SOC^d(0)$  bolo inicializované náhodne z intervalu  $\langle 0.5, 0.8 \rangle$  a bolo udržiavané konštantné počas celého tréovacieho cyklu (jazdy). Celkovo tréovanie pozostávalo z 1200 epoch (približne 60000 zmien váhových prepojení).

Po natréovaní bola riadiaca časť umiestnená do simulátora Toyota Prius a jej výkonnosť bola porovnaná s pôvodnou riadiacou časťou. Navrhnutá metóda vykazovala v priemere o 17 % lepšiu spotrebu a zmeny stavu nabitia batérie boli v priemere o 35 % nižšie.

## 8.2 Modelovanie nelineárneho komunikačného kanála

Nelineárnym bezdrôtovým komunikačným kanálom je možné prenášať informáciu nasledovným spôsobom. Odosielateľ chce odkomunikovať postupnosť symbolov  $s(t)$ . Táto postupnosť je najskôr transformovaná do analógových hodnôt  $d(t)$ , potom je modulovaná do vysokofrekvenčného prenosového signálu a následne je odoslaná. Po prijatí je signál demodulovaný do analógového signálu  $u(t)$ , ktorý je zašumenou a poškodenou verziou originálneho signálu  $d(t)$ . Hlavným zdrojom poškodenia je šum (tepelný alebo spôsobený interferenciou signálov), potom propagácia viacerými cestami, ktorá sa prejaví ako superpozícia susedných symbolov (medzisymbolová interferencia) a nakoniec nelineárna deformácia spôsobená vysielateľom v časti zosilňovača. Aby bola táto deformácia čo najnižšia, signál býva zosilnený hlboko pod možné maximum, čo vedie k neefektívnemu využitiu možností zariadení mobilnej a satelitnej komunikácie. Poškodený signál  $u(t)$  vstupuje do ekvalizačného filtra s výstupom  $y(t)$ , ktorého úlohou je obnoviť originálny signál. Nakoniec je signál  $y(t)$  späťne transformovaný na postupnosť symbolov. Kvalita celého procesu je určená množstvom nesprávne prenesených symbolov.

Model nelineárneho kanála bol v [12] tvorený lineárnym systémom s veľkosťou pamäti 10 a následnou nelineárnou transformáciou. Vstup do kanála je postupnosť  $d(t)$  vytvorená z hodnôt  $\{-3; -1; 1; 3\}$  zodpovedajúcich symbolom  $s_1$  až  $s_4$ . Vstupno-výstupná rovnica lineárnej časti kanála je daná vzťahom:

$$q(n) = 0.08d(n+2) - 0.12d(n+1) + d(n) + 0.18d(n-1) - 0.1d(n-2) + \\ + 0.09d(n-3) - 0.05d(n-4) + 0.04d(n-5) + 0.03d(n-6) + 0.01d(n-7) \quad (81)$$

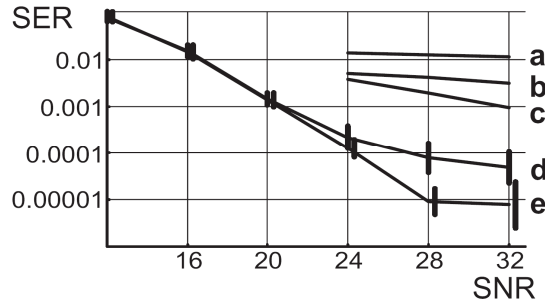
Nelineárna časť kanála nemá pamäť a je určená vzťahom:

$$u(n) = q(n) + 0.036q(n)^2 - 0.011q(n)^3 + v(n), \quad (82)$$

kde  $v$  je aditívny biely gausovský šum.

Úlohou ESN siete je zrekonštruovať pôvodný signál zo zašumených poškodených hodnôt, čiže realizovať funkciu ekvalizačného filtra. V [12] navrhnutá ESN sieť obsahovala rezervoár so 46 neurónmi. Pravdepodobnosť existencie rekurentných váhových prepojení bola 20 % a hodnoty váh boli inicializované náhodne z intervalu  $(-1.0, 1.0)$ . Následne bola matica rekurentných váhových prepojení  $\mathbf{W}$  upravená tak, aby jej spektrálny polomer bol 0.5. Úprava je realizovaná jej prenasobením konštantou  $\lambda^{\text{new}}/\lambda^{\text{old}}$ , čiže  $\mathbf{W}^{\text{new}} = \mathbf{W}^{\text{old}} \lambda^{\text{new}}/\lambda^{\text{old}}$  kde  $\lambda^{\text{old}}$  je spektrálny polomer pôvodnej matice  $\mathbf{W}^{\text{old}}$  a  $\lambda^{\text{new}} = 0.5$  je zvolený spektrálny polomer. Vstupné váhy boli vygenerované z intervalu  $(-0.025, 0.025)$ . Výstupný neurón mal lineárnu aktivačnú funkciu. Vstupom pre ESN sieť nebol priamo poškodený signál  $u(n)$ , ale hodnoty vytvorené pričítaním konštanty, a to  $u(n) + 30$ . Želaný signál použitý na tréning bol  $d(n-2)$ , čiže úlohou siete bolo minimalizovať  $(y(n) - d(n-2))^2$ . Týmto spôsobom je odozva ekvalizačného filtra posunutá oproti vstupu až o 4 časové kroky, nakoľko  $u(n)$  závisí od hodnôt  $d$  až do času  $n+2$ . Parameter zabúdania bol zvolený  $\gamma = 0.998$ .

Pre zvolené odstupy signál-šum (angl. Signal-to-Noise Ratio, SNR) od 12 do 32 db autori natrénovali 20 sietí na 5000 krokoch rekurzívnou metódou najmenších štvorcov. Počas prvých 100 krokov tréning neprebiehalo, tieto kroky slúžili na ustálenie aktivít v rezervoári. Sieť bola testovaná na  $10^7$  krokoch. Výstup siete bol konvertovaný na symboly priamočiarym spôsobom, napr. ak hodnota  $y(t) > 2$ , uvažuje sa symbol zodpovedajúci hodnote 3, čiže symbol  $s_4$ . Kvalita modelu bola určená ako podiel nesprávne prenesených symbolov ku všetkým preneseným symbolom (angl. Symbol Error Rate, SER). Výsledky experimentov sú znázornené na obr. 21. V [12] bola ESN porovnaná s lineárnym ekvalizérom so spätnou väzbou (angl. Decision Feedback Equalizer, DFE), s Volterrovým DFE a s bilineárnym DFE.



**Obr. 21.** Porovnanie chybovosti SER v závislosti na SNR (odstup signál – šum). ESN siete sú porovnané s inými prístupmi: (a) Lineárna DFE, (b) Volterra DFE, (c) bilinear DFE. (d) označuje priemer vytvorený z výsledkov viacerých ENS sietí a (e) sú výsledky najlepších sietí použitých v (d). Obrázok je prevzatý z [12].

Relatívne malá ESN sieť s rezervoárom so 46 neurónmi bola zvolená preto, aby počet adaptovaných parametrov bol porovnateľný s ostatnými modelmi. Napriek tomu bola ESN sieť schopná úlohu riešiť s výrazne lepšou chybovosťou ako iné metódy [12]. Tento príklad poukazuje na možnosť použitia ESN sietí na riešenie praktických úloh. Významnou výhodou použitia ESN sietí je možnosť ich rýchleho natrénovania, čo umožňuje experimentovať s mnohými variantmi a parametrami.

## 9 Implementácia rekurentných neurónových sietí

V predchádzajúcom texte sme algoritmy na tréning rekurentných neurónových sietí formalizovali pomocou zápisu, ktorý je možný jednoduchým spôsobom prepísať do pseudojazyka. V tejto časti teda zapíšeme algoritmy v jednoduchom pseudojazyku takým spôsobom, ktorý umožňuje ich takmer okamžité použitie v bežných procedurálnych programovacích jazykoch, akými sú napríklad jazyk C či Java.

### 9.1 Reprézentácia neurónovej siete v dátových štruktúrach

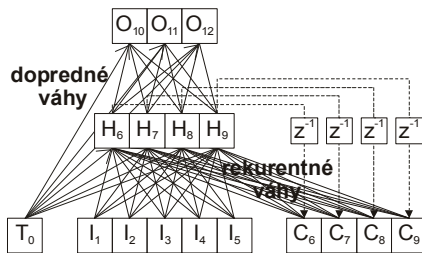
Pri zápise algoritmov pre tréning rekurentných neurónových sietí sme využívali operátory  $d$ ,  $s$ ,  $\tau$  a  $w$  pracujúce nad usporiadanou množinou váhových prepojení, ktorá určovala štruktúru siete. Každé váhové prepojenie v rekurentnej neurónovej sieti malo svoj jedinečný index  $l$  a bolo reprezentované usporiadanou štvoricou  $(i, j, t, v)$ , kde  $i$  bol index cieľového a  $j$  index zdrojového neurónu,  $t$  časové oneskorenie a  $v$  intenzita váhy. Nech  $(i, j, t, v)$  je váhové prepojenie s indexom  $l$ , tak potom  $d_l = i$ ,  $s_l = j$ ,  $\tau_l = t$  a  $w_l = v$ . Množinu indexov všetkých váhových prepojení sme označili  $W$ .

Tento spôsob zápisu štruktúry siete je možné jednoduchým spôsobom previesť na zápis v pseudojazyku. Množinu váhových prepojení si môžeme predstaviť ako tabuľku záznamov, ktoré reprezentujú váhové prepojenia. Pre jednoduchší zápis tréningových algoritmov budeme pomocou tabuľky udržiavať aj informácie o neurónoch v sieti. Na

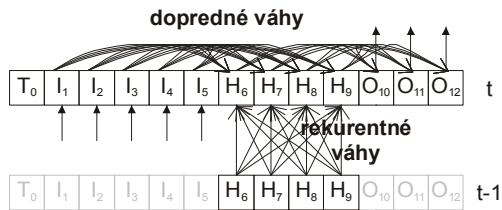


obr. 22 je znázornený spôsob reprezentácie Elmanovej rekurentnej neurónovej siete v údajových štruktúrach.

a) Elmanova architektúra - kontextová vrstva



b) Elmanova architektúra - rozvinutie v čase



c) Štruktúra váhových prepojení

Index váhy	Zdrojový neurón wSource	Cieľový neurón wDest	Časové ones. wDelay	Hodnota wValue
0	0	6	0	
1	1	6	0	
2	2	6	0	
3	3	6	0	
4	4	6	0	
5	5	6	0	
6	6	6	1	
7	7	6	1	
8	8	6	1	
9	9	6	1	
10	0	7	0	
11	1	7	0	
12	2	7	0	
13	3	7	0	
14	4	7	0	
15	5	7	0	
16	6	7	1	
17	7	7	1	
18	8	7	1	
19	9	7	1	
20	0	8	0	
21	1	8	0	
22	2	8	0	
23	3	8	0	
24	4	8	0	
25	5	8	0	
26	6	7	1	
27	7	7	1	
28	8	7	1	
29	9	7	1	

Index váhy	Zdrojový neurón wSource	Cieľový neurón wDest	Časové ones. wDelay	Hodnota wValue
30	0	9	0	
31	1	9	0	
32	2	9	0	
33	3	9	0	
34	4	9	0	
35	5	9	0	
36	6	7	1	
37	7	7	1	
38	8	7	1	
39	9	7	1	
40	0	10	0	
41	6	10	0	
42	7	10	0	
43	8	10	0	
44	9	10	0	
45	0	11	0	
46	6	11	0	
47	7	11	0	
48	8	11	0	
49	9	11	0	
50	0	12	0	
51	6	12	0	
52	7	12	0	
53	8	12	0	
54	9	12	0	

d) Štruktúra neurónov

Index neurónu	Prvá váha uFirstWeight	Posledná váha uLastWeight	Typ uType	Aktivačná funkcia uActFunc
0			T	
1			I	
2			I	
3			I	
4			I	
5			I	
6	0	5	H	SGM
7	6	11	H	SGM
8	12	17	H	SGM
9	18	23	H	SGM
10	24	28	O	LIN
11	29	33	O	LIN
12	34	38	O	LIN

**Obr. 22.** Reprezentácia Elmanovej rekurentnej neurónovej siete s 5 vstupnými, 4 skrytými a 3 výstupnými neurónmi. a) sieť znázornená s využitím kontextovej vrstvy, b) sieť znázornená v časovom kroku (rozvinutie), c) tabuľka obsahujúca informácie o váhových prepojeniach a d) tabuľka s informáciami o neurónoch.

Zápis predpokladá, že všetky neuróny v sieti sú usporiadané a oindexované. Prvý bude prahový neurón s indexom 0, potom nasledujú vstupné, skryté a výstupné neuróny. Neuróny s vyšším indexom nesmú poskytovať svoju aktivitu cez dopredné váhové prepojenie (váhové prepojenie s časovým oneskorením 0) neurónom s nižším indexom. Táto požiadavka reprezentuje fakt, že keď počítame aktivitu neurónu musíme mať už vypočítané aktivity neurónov, s ktorými je daný neurón spojený váhovými prepojeniami. Pre všetky bežne používané architektúry neurónových sietí pracujúcich v diskretnom čase je možné neuróny jednoduchým spôsobom usporiadať tak, aby táto požiadavka bola splnená. O každom neuróne je vhodné si uchovať informáciu o jeho

prvej a poslednej váhe, jeho type (prahový, vstupný, skrytý, výstupný) a napr. o jeho aktivačnej funkcii. Spôsob výpočtu aktivity bude zrejmý z ďalšieho textu. Váhy sú usporiadané najskôr podľa indexu cieľového neurónu, potom podľa indexu zdrojového neurónu. Tiež je potrebné pre každé váhové prepojenie uchovávať informáciu o jeho intenzite (vlastná váha) a pre rekurentné siete aj o jeho časovom oneskorení.

## 9.2 Dopredné šírenie signálu

Použitie uvedeného spôsobu kódovania neurónovej siete bude zrejmé z nasledovného algoritmu zapísaného v pseudojazyku (obr. 23), ktorého úlohou je výpočet aktivít na jednotlivých neurónoch zo vstupov prezentovaných sietí. Už aj táto relatívne jednoduchá súčasť každého simulátora sietí je vďaka použitiu zvoleného kódovania ešte jednoduchšia a prehľadnejšia.

NW	- počet váh
wSource[0..NW-1]	- zdrojové neuróny
wDest[0..NW-1]	- cieľové neuróny
wDelay[0..NW-1]	- časové oneskorenia
wValue[0..NW-1]	- vlastné hodnoty váh. prepojení
NU	- počet neurónov
uFirstWeight[0..NU-1]	- indexy prvých váh
uLastWeight[0..NU-1]	- indexy posledných váh
uType[0..NU-1]	- typy neurónov (THRESHOLD, INPUT...OUTPUT)
NSTEPS	- počet krokov
ACT[0..NU-1,0..NSTEPS-1]	- všetky aktivity neurónov
ACTD[0..NU-1,0..NSTEPS-1]	- derivácie aktivít neurónov
Sgm(iact)	- aktivačná funkcia
SgmDer(iact)	- derivácia aktivačnej funkcie
Input(ui,ts)	- f. vráti vstup zadaný sietí
Output(ui,act,ts)	- f. nastaví výstup siete
Target(ui,ts)	- f. vráti želaný výstup siete
ts	- aktuálny časový krok

```

1. for ui:=0 to NU-1 do
2.   begin
3.     if uType[ui] = THRESHOLD then ACT[ui,ts] := 1.0;
4.     else if uType[ui] = INPUT then
5.       ACT[ui,ts] := Input(ui,ts);
6.     else
7.       begin
8.         iact := 0.0;
9.         for wi:=uFirstWeight[ui] to uLastWeight[ui] do
10.          iact += wValue[wi]*ACT[wSource[wi],ts-wDelay[wi]];
11.          ACT[ui,ts] := Sgm(iact);
12.          ACTD[ui,ts] := SgmDer(iact);
13.        end;
14.        if uType = OUTPUT then Output(ui,ACT[ui,ts],ts);
15.      end;

```

**Obr. 23.** Zápis algoritmu pre dopredné šírenie signálu v rekurentnej neurónovej sieti.

**Obr. 23.** Zápis algoritmu pre dopredné šírenie signálu v rekurentnej neurónovej sieti.

Tento algoritmus opisuje jeden krok (krok ts) dopredného šírenia. Prvý neurón s indexom 0 zodpovedá prahovej váhe a jeho aktivita je vždy rovná 1 (riadok 3).

Aktivity na vstupných neurónoch sú nastavené podľa aktuálneho vstupu predloženého siete (riadky 4 a 5). Vnútorne aktivity na skrytých a výstupných neurónoch sú počítané násobením váh neurónov s aktivitami na zdrojových neurónoch zo zodpovedajúceho časového kroku (riadky 9 až 10). Operátor  $+=$  reprezentuje pričítanie výrazu na jeho pravej strane k premennej na ľavej strane. Pre dopredné váhy majúce časové oneskorenie 0 sú použité už vypočítané aktivity v aktuálnom časovom kroku, pre rekurentné váhy sú použité aktivity z minulých krokov. Vlastná aktivita na neuróne je vypočítaná aktivačnou funkciou na základe internej aktivity neurónu (riadok 11). Výpočet derivácie aktivačnej funkcie je potrebný iba vtedy, keď sa bude následne realizovať adaptácia váhových prepojení gradientovými algoritmi (riadok 12). Aktivity na výstupných neurónoch sú poskytnuté ako výstupy zo siete (riadok 14).

V prípade rekurentnej siete je potrebné sa zamyslieť nad iníciačnými časovými krokmi. Pre čas  $t_s$  menší ako časové oneskorenie váhy hrozí, že pole ACT bude pristupované so záporným indexom. (riadok 10). Vhodným riešením môže byť nastavenie  $t_s$  pre čas 0 na hodnotu najdlhšieho časového oneskorenia v sieti (najväčšia hodnota v poli  $wDelay$ ) a aktivity pre menšie  $t_s$  prehlásiť za iníciačný stav siete.

### 9.3 Spätne šírenie chybového signálu

Elmanova jednoduchá rekurentná neurónová sieť bola v [2] trébovaná jednoduchým algoritmom spätného šírenia chyby v čase (angl. Backpropagation, BP). BP algoritmus je najpoužívanejším prístupom na adaptáciu váh v doprednej sieti. Aj keď nie je vhodný pre trébovanie rekurentných sietí, môže byť použitý na jednoduchšie úlohy alebo ako referenčná metóda. Zápis BP algoritmu v pseudojazyku je na obr. 24.

Derivácie  $DE\_DNA[ui]$  sú postupne spätne vypočítavané, a preto musia byť na začiatku nastavené na 0 (riadok 1). Ďalšia časť zdrojového kódu slúži na výpočet derivácií (riadky 3 až 13). Začínajúc posledným neurónom s indexom  $NU-1$  sú postupne vypočítavané derivácie  $DE\_DNA[ui]$ . Hneď, ako sú spracované všetky výstupné a skryté neuróny, cyklus končí (riadok 5), lebo už boli vypočítané všetky derivácie. Ak je aktuálne spracovávaný neurón  $ui$  výstupným neurónom, rozdiel medzi želanou a skutočnou aktivitou je pripočítaný do práve vypočítavanej derivácie (riadky 6 a 7). Nakoniec, keďže všetky relevantné chybové signály boli už z neurónov s vyšším indexom spätne prešírené do aktuálne počítanej derivácie  $DE\_DNA[ui]$ , je táto hodnota prenasobená deriváciou aktivačnej funkcie (riadok 8) a je ďalej spätne šírená na všetky neuróny, ktoré sú vstupmi pre aktuálny neurón (riadky 9 až 12). Nemá zmysel spätne šíriť chybový signál na vstupné neuróny (riadok 10) a taktiež chybový signál sa nešíri cez rekurentné váhy (riadok 11). Posledná časť algoritmu je zmena váhových prepojení (riadky 15 až 20). Zmena váhy ( $DLT\_W[wi]$ ) je vypočítaná prenasobením rýchlosti učenia  $\alpha$  s chybovým signálom na danom neuróne a aktivitou na zdrojovom neuróne. K zmene váhy v súčasnom kroku je pridaná aj zmena váhy vypočítaná v predchádzajúcom časovom kroku prenasobená momentom  $\beta$  (riadky 17 a 18). Nakoniec je zmenená samotná sila váhového prepojenia  $wValue[wi]$  (riadok 19).

```

DE_DNA[0..NU-1]           - chyb. sig., derivácia chyby voči int. akt.
DLT_W[0..NW-1]           - vypočítané zmeny váh
beta                      - momentum
alpha                     - rýchlosť učenia

1. for ui=NU-1 downto 0 do DE_DNA[ui] := 0.0;
2.
3. for ui=NU-1 downto 0 do
4.   begin
5.     if uType[ui] = INPUT then break;
6.     if uType[ui] = OUTPUT then
7.       DE_DNA[ui] += Target(ui,ts)-ACT[ui,ts];
8.       DE_DNA[ui] *:= ACTD[ui,ts];
9.       for wi := uLastWeight[ui] downto uFirstWeight[ui] do
10.        if uType[wSource[wi]] = INPUT then break;
11.        else if wDelay[wi] = 0 then
12.          DE_DNA[wSource[wi]] += wValue[wi]*DE_DNA[ui];
13.        end;
14.
15. for wi:=0 to NW-1 do
16.   begin
17.     DLT_W[wi] := beta*DLT_W[wi] +
18.       alfa*DE_DNA[wDest[wi]]*ACT[wSource[wi],ts-wDelay[wi]];
19.     wValue[wi] += DLT_W[wi];
20.   end;

```

**Obr. 24.** Algoritmus spätného šírenia chybového signálu.

Jednoduchý BP algoritmus nie je vhodný na tréning rekurentných neurónových sietí, lebo nešíri chybový signál spätne aj cez rekurentné prepojenia. Jednoduchá modifikácia uvedeného algoritmu, nazývaná spätné šírenie chyby v čase, rieši tento problém.

## 9.4 Spätné šírenie chybového signálu v čase

Algoritmus spätného šírenie chyby v čase (angl. Backpropagation Through Time, BPTT) umožňuje presný výpočet derivácie chyby pre rekurentné neurónové siete. Princíp spočíva v rozvinutí rekurentnej siete v čase do potenciálne mnohovrstvovej doprednej siete a v následnom použití klasického BP algoritmu [34]. V praxi nie je rekurentná sieť rozvíjaná v čase až do času 0, ale iba istý počet krokov do minulosti označovaných ako veľkosť okna. Táto modifikácia už bola popísaná a nazýva sa skrátene BPTT. Zápis BPTT algoritmu v pseudojazyku je na obr. 25.

Podobne, ako v prípade BP algoritmu, musia byť derivácie  $DE\_DNA$  inicializované na 0. (cyklus na riadku 1). Výpočet v časovom kroku  $ts$  spočíva v spätnom šírení chybového signálu  $winSize-1$  časových krokov späť (vonkajší cyklus na riadkoch 3 až 17). Pre každý neurón  $ui$  (cyklus na riadkoch 4 až 17) a pre všetky jeho váhy (cyklus na riadkoch 10 až 16) sú derivácie  $DE\_DNA[ui,hi]$  spätne šírené cez zodpovedajúce prepojenia do derivácii v zodpovedajúcom čase (riadok 14), rekurentné prepojenia sú tiež uvažované. Pre výstupné neuróny a iba pre  $hi$  rovné 0 sa chybový signál obohatí o rozdiel medzi želanou a skutočnou hodnotou (riadok 7 a 8). Výpočet derivácie  $DE\_DNA[ui,hi]$  je ukončený prenasobením akumulovanej hodnoty deriváciou aktivačnej funkcie v zodpovedajúcom čase (riadok 9). Ako aj v prípade BP

algoritmu signál nie je šírený na vstupné neuróny (riadok 12). Rovnako je zabezpečený iba povolený prístup do poľa DE\_DNA (podmienka na riadku 13). Zmeny váh sú vypočítavané v rovnakom cykle ako spätné šírenie. Do zmeny váhy DLT\_W[wi] je pripočítaná časť pripadajúca na práve spočítaný chybový signál v čase (riadok 15). Váhy sú zmenené na riadkoch 19 až 23. Časový krok ts musí byť dostatočne vysoký, aby nedošlo k nepovolenému prístupu do poľa ACT (riadok 15).

```

DE_DNA[0..NU-1,0..winSize-1]      - spätné šírený chybový signál
winSize                             - veľkosť okna

1.  for hi=0 to winSize-1 do for ui=NU-1 downto 0 do DE_DNA[ui,hi] := 0.0;
2.
3.  for hi=0 to winSize-1 do
4.    for ui=NU-1 downto 0 do
5.      begin
6.        if uType[ui] = INPUT then break;
7.        if (uType[ui] = OUTPUT) AND (hi = 0) then
8.          DE_DNA[ui,hi] += Target(ui,ts)-ACT[ui,ts];
9.          DE_DNA[ui,hi] *= ACTD[ui,ts-hi];
10.       for wi := uLastWeight[ui] downto uFirstWeight[ui] do
11.         begin
12.           if uType[wSource[wi]] = INPUT then break;
13.           if wDelay[wi]+hi < winSize then
14.             DE_DNA[wSource[wi],wDelay[wi]+hi] += wValue[wi]*DE_DNA[ui,hi];
15.             DLT_W[wi] += alfa*DE_DNA[ui,hi] * ACT[wSource[wi],ts-hi-wDelay[wi]];
16.           end;
17.         end;
18.
19.       for wi:=0 to NW-1 do
20.         begin
21.           wValue[wi] += DLT_W[wi];
22.           DLT_W[wi] *= beta;
23.         end;

```

Obr. 25. Algoritmus spätného šírenia chybového signálu v čase.

## 9.5 Rekurentné učenie v reálnom čase

Ďalšou možnosťou ako zrealizovať výpočet derivácií chyby je použiť rekurentné učenie v reálnom čase (real-time recurrent learning - RTRL). Na rozdiel od BPTT algoritmu RTRL nepotrebuje presne definovaný časový interval určujúci rozvinutie siete. Hlavnou nevýhodou je jeho väčšia výpočtová náročnosť. Zápis RTRL algoritmu v pseudojazyku je na obr. 26.

Dopredne šírené derivácie DA\_DW zodpovedajúce iniciálnym časovým krokom sú ešte pred začiatkom tréovania vynulované. Jeden krok tréovania algoritmom RTRL je nasledovný. Pre každé váhové prepojenie  $w_i$  (cyklus na riadkoch 1 až 16) a každý neurón  $u_i$  (cyklus na riadkoch 2 až 16) okrem vstupných a prahových neurónov (riadok 4) aktuálne počítaná derivácia  $DA\_DW[ui,wi,ts]$  je postupne vytvorená doprednou propagáciou z už vypočítaných derivácií  $DA\_DW[wSource[ui],wi,ts-wDelay[ui]]$  cez váhy  $wValue[ui]$  (cyklus na riadkoch 6 až 9). Vstupné a prahové neuróny sú vynechané (riadok 8), im zodpovedajúca derivácia je vždy nulová. Riadky 11 a 12 reprezentujú explicitný efekt

ovplyvňujúci derivácie. Nakoniec je výpočet derivácie  $DA\_DW[ui,wi,ts]$  dokončený jej prenasobením deriváciou aktivačnej funkcie (riadok 14).

```

NO                                     - počet výstupných neurónov
oIndex[0..NO-1]                       - pole s indexmi výstupných neurónov
DA_DW[0..NU-1,0..NW-1,0..NSTEPS-1]   - dopredne šírené derivácie

1. for wi:=0 to NW-1 do
2.   for ui:=0 to NU-1 do
3.     begin
4.       if uType[ui] <> INPUT AND uType[ui] <> THRESHOLD then
5.         begin
6.           for uwi:=uFirstWeight[ui] to uLastWeight[ui] do
7.             if uType[wSource[uwi]] <> INPUT AND
8.               uType[wSource[uwi]] <> THRESHOLD then
9.               DA_DW[ui,wi,ts] +=wValue[uwi]*DA_DW[wSource[uwi],wi,ts-wDelay[uwi]];
10.            if wDest[wi] = ui then
11.              DA_DW[ui,wi,ts] += ACT[wSource[wi],ts-wDelay[wi]];
12.            DA_DW[ui,wi,ts] *= ACTD(ui,ts);
13.          end;
14.        end;
15.      end;
16.    end;
17.  end;
18. for wi:=0 to NW-1 do
19.  begin
20.    DLT_W[wi] := beta;
21.    for oi:=0 to NO-1 do
22.      DLT_W[wi] +=
23.        alfa*(Target(oIndex[oi],ts)-ACT[oIndex[oi],ts])*DA_DW[oIndex[oi],wi,ts];
24.    wValue[wi] += DLT_W[wi];
25.  end;

```

**Obr. 26.** Algoritmus pre rekurentné učenie v reálnom čase.

Posledný cyklus (riadky 18 až 25) implementuje zmeny váh. Najskôr je momentum  $\beta$  aplikované na staré zmeny váh (riadok 20), potom je zmena váhy  $DLT\_W[wi]$  postupne vypočítaná násobením rýchlosti učenia  $\alpha$  s chybami na výstupných neurónoch  $Target(oIndex[oi],ts)-ACT[oIndex[oi],ts]$  a derivácii  $DA\_DW[oIndex[oi],wi,ts]$  (riadky 21 až 23). Nakoniec sú na riadku 24 zmenené váhy.

## 9.6 Rozšírený Kalmanov filter

Ako sme už uviedli, medzi najúspešnejšie prístupy používané na tréning neurónových sietí patria metódy založené na Kalmanovej filtrácii. Odhadovaný stavový vektor je v prípade neurónovej siete stotožnený s tréningovými váhami siete. Na výpočet derivácií pre maticu  $\mathbf{H}$  je možné použiť algoritmus skoro identický s BPTT alebo RTRL. Matica šumu pozorovaní  $\mathbf{R}$  určuje rýchlosť konvergence podobne, ako rýchlosť učenia v klasických gradientových algoritmoch.  $\mathbf{R}$  je diagonálna matica a môže ostať v priebehu učenia konštantná alebo sa môže meniť, podobne ako rýchlosť učenia. Matica šumu procesu  $\mathbf{Q}$  je tiež diagonálna inicializovaná na malé hodnoty a pomáha udržať numerickú stabilitu výpočtu. Matica kovariancie chyby  $\mathbf{P}$  je tiež zvyčajne inicializovaná ako diagonálna. Typické hodnoty iniciálnych parametrov

pre matice môžu byť napr.  $\mathbf{R} = 100\mathbf{I}$ ,  $\mathbf{Q} = 1.10^{-5}\mathbf{I}$  a  $\mathbf{P} = 1.10^3\mathbf{I}$ , kde  $\mathbf{I}$  je jednotková matica. Zápis RTRL algoritmu v pseudojazyku je na obr. 27.

```

NO                                     - počet výstupných neurónov
oIndex[0..NO-1]                       - pole s indexmi výstupných neurónov
DO_DW[0..NO-1,0..NW-1]                - vypočítané derivácie
DO_DNA[0..NU-1,0..winSize-1]          - späťne šírený signál
winSize                                - veľkosť okna

H[0..NO-1,0..NW-1]                    - matica derivácií
P[0..NW-1,0..NW-1]                    - matica kovariancie chýb
Q[0..NW-1,0..NW-1]                    - matica šumu procesu
R[0..NO-1,0..NO-1]                    - matica šumu pozorovaní
W[0..NW-1,0..0]                       - odhadovaný váhový vektor
K[0..NW-1,0..NO-1]                    - Kalmanov zisk
D[0..NO-1,0..0]                       - vektor želaných výstupov
O[0..NO-1,0..0]                       - vektor vypočítaných výstupov

Tr(X)                                  - f. vyp. transponovanú maticu
Inv(X)                                  - f. vyp. inverznú maticu

1. for oui:=0 to NO-1 do for wi:=0 to NW-1 do DO_DW[oui,wi] := 0.0;
2.
3. for oui:=0 to NO-1 do
4. begin
5.   for hi:=0 to winSize-1 do for ui:=0 to NU-1 do DO_DNA[ui,hi] := 0.0;
6.   DO_DNA[oIndex[oui],0] := 1.0;
7.
8.   for hi:=0 to winSize-1 do
9.     for ui:=NU-1 downto 0 do
10.    begin
11.     if uType[ui] = INPUT then break;
12.     DO_DNA[ui,hi] := ACTD[ui,ts-hi];
13.     for wi := uLastWeight[ui] downto uFirstWeight[ui] do
14.     begin
15.      if (uType[wSource[wi]] = INPUT) then break;
16.      if (wDelay[wi]+hi < winSize) then
17.        DO_DNA[wSource[wi],wDelay[wi]+hi] += wValue[wi]*DO_DNA[ui,hi];
18.        DO_DW[oui,wi] += DO_DNA[ui,hi]*ACT[wSource[wi],ts-hi-wDelay[wi]];
19.      end;
20.    end;
21.  end;
22.
23. for wi:=0 to NW-1 do W[wi,0] := wValue[wi];
24.
25. for oui:=0 to NO-1 do
26. begin
27.   D[oui,0] := Target(oIndex[oui],ts);
28.   O[oui,0] := ACT[oIndex[oui],ts];
29.   for wi:=0 to NW-1 do H[oui,wi] := DO_DW[oui,wi];
30. end;
31.
32. K := P * Tr(H) * Inv(H * P * Tr(H) + R);
33. P := P - K * H * P + Q;
34. W := W + K * (D - O);
35.
36. for wi:=0 to NW-1 do wValue[wi] := W[wi,0];

```

**Obr. 27.** Rozšírený Kalmanov filter aplikovaný na tréning rekurentných neurónových sietí, derivácie sú určované spätným šírením v čase.

Hodnoty DO\_DW (derivácie výstupných aktivít vzhľadom na váhy) sú inicializované na 0 (riadok 1). Následne pre každý výstupný neurón je vykonané spätné šírenie (cyklus na riadkoch 3 až 21). Pre jedno spätné šírenie je chybový signál pre jediný výstupný neurón nastavený na 1 (riadok 6), ostatné prvky poľa DO\_DNA (derivácie výstupných aktivít vzhľadom na všetky vnútorné aktivity neurónov) sú nastavené na 0 (riadok č. 5) a sú vypočítavané v nasledujúcom cykle (riadok 8 až 20). Spätné šírenie je vykonané rovnakým spôsobom ako pre BPTT, iba namiesto výpočtu zmien váh sú vypočítavané derivácie DO\_DW. Jedno spätné šírenie sa postará o výpočet DO\_DW pre jeden výstupný neurón. Nakoniec sú nastavené matice a vektory pre KF (riadky 23 až 30) a môže byť vykonaný vlastný krok Kalmanovej filtrácie (riadky 32 až 34). Nakoniec sú vypočítané váhy umiestnené do poľa používaného pri doprednom a spätnom šírení (riadok 36).

## 10 Záver

Rekurentné siete sú na rozdiel od dopredných sietí schopné spracovávať úlohy s časovým kontextom. Napriek ich nesporne vysokému aplikačnému potenciálu rekurentné siete stále nie sú bežne používané na riešenie konkrétnych problémov. Tento text sa pokúsil priblížiť čitateľovi problematiku rekurentných neurónových sietí. Ťažisko príspevku tvorí opis trénovacích algoritmov, nakoľko ich pochopenie, ale aj implementácia, sú náročnejšie ako v prípade dopredných neurónových sietí. Algoritmy sú formálne opísané netradičným, ale do značnej miery univerzálnym spôsobom. Príspevok sa venuje aj algoritmom využívajúcim Kalmanovu filtráciu, ktoré reprezentujú súčasnú úroveň vedeckého poznania.

*Pod'akovanie:* Táto kapitola vznikla za podpory grantovej agentúry VEGA v rámci grantových úloh VG-1/0848/08 a VG-1/0822/08.

## Literatúra

- [1] Boden, M., Wiles, J.: On learning context free and context sensitive languages. *IEEE Transactions on Neural Networks* 2, 2002, 491-493.
- [2] Bengio, Y., Simard, P., Frasconi, P.: Learning Long-Term Dependencies with Gradient Descent is Difficult. *IEEE Transactions on Neural Networks* 2(5), 1994, 157-166.
- [3] Čerňanský, M., Beňušková, L.: Finite-state Reber automaton and the recurrent neural networks trained in supervised and unsupervised manner. *Artificial Neural Networks - ICANN 2001*, Vienna, Austria. 2001, 737-742.
- [4] Čerňanský, M., Beňušková, L.: Simple recurrent network trained by RTRL and extended Kalman filter algorithms. *Neural Network World* 13(3), 2003.
- [5] Čerňanský, M., Makula, M., Beňušková, L.: Organization of the state space of a simple recurrent neural network before and after training on recursive linguistic structures. *Neural Networks* 20, 2007, 236-244.



- 
- [6] Wan, E. A., Nelson, A. T.: Dual EKF methods. *Kalman Filtering and Neural Networks*, Hillsdale, N.J., 2000, 123–173.
- [7] Elman, J. L.: Finding structure in time. *Cognitive Science* 14(2), 1990, 179-211.
- [8] Feldkamp, L. A., Prokhorov, D., Eagen, C.F., Yuan, F.: Enhanced multi-stream kalman filter training for recurrent networks. *Nonlinear modeling: advanced black-box techniques*, Boston, 1998, 29-53.
- [9] Frasconi, P., Gori, M., Soda, G.: Local feedback multilayered networks. *Neural Computation* 4, 1992, 120-130.
- [10] Haykin, S.: *Neural Networks*. Prentice-Hall, New Jersey, 1994.
- [11] Jaeger, H.: The “echo state” approach to analysing and training recurrent neural networks. Technical Report GMD 148, German National Research Center for Information Technology, 2001.
- [12] Jaeger, H., Haas, H.: Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. *Science* 5667, 2004, 78-80.
- [13] Jordan, M. I.: Attractor dynamics and parallelism in a connectionist sequential machine. *Proceedings of the Eighth Conference of the Cognitive Science Society*, 1986, 531-546.
- [14] Julier, S., Uhlmann, J.: A new extension of the Kalman filter to nonlinear systems. *Proceedings of the Int. Symp. Aerospace/Defense Sensing, Simul. and Controls*, Orlando. 1997, 182–193.
- [15] Kalman, R. E.: A new approach to linear filtering and prediction problems. *Transactions of the ASME—Journal of Basic Engineering*, 82, 1960, 35–45.
- [16] Kolen, J. F.: The origin of clusters in recurrent neural network state space. *Proceedings from the Sixteenth Annual Conference of the Cognitive Science Society* 1994, 508-513.
- [17] Kvasnička, V., a kol.: *Úvod do teórie neurónových sietí*, IRIS, Bratislava 1997.
- [18] Lawrence, S., Giles, C. L., Fong, S.: Natural language grammatical inference with recurrent neural networks. *IEEE Transactions on Knowledge and Data Engineering* 12(1), 2000, 126-140.
- [19] Machler, M., Bühlmann, P.: Variable length Markov chains: methodology, computing and software. *Journal of Computational and Graphical Statistics*, 13, 2004, 435–455.
- [20] Makula, M., Čerňanský, M., Beňušková, Ľ.: Approaches based on Markovian architectural bias in recurrent neural networks. *SOFSEM 2004: Theory and Practice of Computer Science*, 2004, 257–264.
- [21] Maybeck, P., S.: *Stochastic models, estimation, and control*, 1979.
- [22] Nørgaard, M., Poulsen, N. K., Ravn, O.: Advances in derivative-free state estimation for nonlinear systems. Technical Report IMMREP-1998-15, Department of Mathematical Modelling, DTU, 1998.

- 
- [23] Pérez-Ortiz, J. A., Gers, F. A., Eck, D., Schmidhuber, J.: Kalman filters improve LSTM network performance in problems unsolvable by traditional recurrent nets. *Neural Networks*, 16(2), 2003.
- [24] Prokhorov, D. V.: Toyota Prius HEV neurocontrol and diagnostics. *Neural Networks*, 21, 2008, 458–465.
- [25] Puskorius, G.V., Feldkamp, L. A.: Neurocontrol of nonlinear dynamical systems with Kalman filter trained recurrent networks. *IEEE Transactions on Neural Networks*, 5(2), 1994, 279-297.
- [26] Puskorius, G.V., Feldkamp, L. A.: Extensions and enhancements of decoupled extended Kalman filter training. *Proceedings of international Conference on Neural Networks - ICNN*, 1997, 1879-1883.
- [27] Rodriguez, P.: Simple Recurrent Networks Learn Context-free and Context-Sensitive Languages by Counting. *Neural Computation* 13, 2001 2093-2118 .
- [28] Ron, D., Singer, Y., Tishby, N.: The power of amnesia. *Machine Learning*, 25 1996, 117-149.
- [29] Rumelhart, D. E, Hinton, G. E., Williams, R. J.: Learning internal representations by error propagation. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. I: Foundations*, 1986, 318-362.
- [30] Tiňo, P., Vojtek, V.: Extracting stochastic machines from recurrent neural networks trained on complex symbolic sequences, *Neural Network World*, 8, 1998, 517-530.
- [31] Tiňo, P., Čerňanský, M., Beňušková, Ľ.: Markovian Architectural Bias of Recurrent Neural Networks. *IEEE Transactions on Neural Networks*, 15(1), 2004, 6-15.
- [32] Trebatický, P.: Recurrent Neural Network training with the Kalman Filter-based techniques, *Neural network world* 15(5) 2005, 471-488.
- [33] Wan, E. A., Nelson, A. T.: Dual EKF methods. In *Kalman Filtering and Neural Networks*, Wiley, Hillsdale, N.J., 2000, 123–173.
- [34] Werbos, P.J.: Backpropagation through time; what it does and how to do it. *Proceedings of the IEEE*, 78, 1990, 1550-1560.
- [35] Wiles, J., Elman, J.: Learning to count without a counter: A case study of dynamics and activation landscapes in recurrent networks. *Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society*, 1995, 482 – 487.
- [36] Williams, R. J.: Training recurrent networks using the extended Kalman filter. *Proceedings of the International Joint Conference on Neural Networks*, volume 4, Baltimore, 1992, 241-246.
- [37] Williams, R. J., Zipser, D.: A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1, 1989, 270-280.

# Praktický úvod do symbolickej umelej inteligencie

Peter KOSTELNÍK<sup>1</sup>

**Abstrakt.** Cieľom tohto textu je priblížiť základné princípy fungovania systémov so symbolickou umelou inteligenciou. Spomenuté budú systémy pre riešenie úloh a hranie hier, systémy pre automatické dokazovanie, pravidlové a expertné systémy, a sémantické modelovanie. Ďalším cieľom je poskytnutie stručného sprievodcu symbolickým prístupom umelej inteligencie.

## 1 Symbolická umelá inteligencia

Plánovanie, diagnostika, rozvrhovanie, logistika, riešenie hlavolamov, hranie hier, riešenie logických úloh, klasifikácia, komunikácia v prirodzenom jazyku, automatické získavanie informácií z textov, učenie sa, vnímanie prostredia a manipulácia s objektmi, simulácia biologického metabolizmu alebo sociálneho správania sa, ... A to som vlastne ani nezačal. Čitateľ snáď bude súhlasiť, že riešenie úloh v akejkoľvek z týchto oblastí si vyžaduje množstvo znalostí a skúseností. A podľa mňa by tvrdenie, že takéto úlohy si nevyžadujú ľudskú inteligenciu, zrejme prekvapilo každého súdneho človeka. Ak by ste riešenie týchto úloh chceli ponechať len na samotný počítač, vystupuje do popredia otázka, ako by ste postupovali?

Hovorí sa, že práve systémy s umelou inteligenciou dokážu riešiť úlohy, ktoré si vyžadujú ľudskú inteligenciu. Povedané inak, systémy s umelou inteligenciou dokážu riešiť problémy takým spôsobom, ktorý ľudia pokladajú za inteligentný. Čo je to potom umelá inteligencia? Nuž, vedná disciplína patriaca pod krídla počítačovej vedy (informatiky), ktorá sa zaoberá tvorbou takých počítačových systémov, ktorých riešenie vyžaduje ľudskú inteligenciu. Jej história má svoje počiatky niekde v päťdesiatych rokoch minulého storočia, keď výpočtová technika dosiahla dostatočnú technickú úroveň napriek tomu, že informatici používali obrovské sálové počítače, ktorý svoje výsledky dolovali z hĺby diernych pások. Začali vznikať prvé netradičné programy, ktoré logicky dokazovali tvrdenia alebo hrali hru dámu, čím uvádzali do úžasu laických pozorovateľov.

Ale, dokážu vôbec počítače myslieť? Dokážu sa tieto stroje správať inteligentne? Je vôbec možné, aby riešili úlohy, ktoré si vyžadujú ľudskú inteligenciu? A ešte viac

---

<sup>1</sup> NESS KDC, s.r.o, Puškinova 3, Košice, E-mail: peter.kostelnik@gmail.com

zoširoka, môžu mať počítače myseľ, vedomie, mentálne stavy tak, ako človek? Je ľudský mozog principiálne tiež len počítač? Tieto otázky kvária vedcov z oblasti umelej inteligencie a kognitívnej vedy už celé desaťročia, od vzniku umelej inteligencie pred viac ako 50 rokmi až po súčasnosť. Nebudem sa zaoberať riešením týchto otázok, ale čitateľ musí uznať, že pre mnohých informatikov s filozofickou orientáciou je táto problematika veľmi obrovské a vďačné ihrisko na hranie.

Nedá mi nespomenúť napríklad známy Turingov test. Alan Turing v roku 1950 publikoval článok, kde okrem iného ponúkol spôsob testovania inteligencie strojov [1]. Tento test tiež zvyknú volať imitačná hra, pri ktorej, v jej jednoduchej verzii, ľudský rozhodca pomocou počítača kladie otázky v prirodzenom jazyku dvom hráčom (napr. Koľko je  $944532 \times 83352$ ? Koľko nôh má trojročný púštny gekon? Mávaš často na raňajky párky?). Jeden z hráčov je človek, druhý je počítač. Ak rozhodca nie je schopný rozhodnúť, ktorý z hráčov je počítač, počítač prešiel Turingovým testom. Typickým príkladom moderného Turingovho testu je napr. CAPTCHA [2]. To sú tie deformované písmená na obrázku, ktoré musíte vpísať do formulára pred tým, ako ho odošlete. Takto by sa malo zabrániť automatickému odosielaniu formulárov obskúrnymi internetovými robotmi. CAPTCHA je príkladom tzv. obráteného Turingovho testu, keď počítač musí rozhodnúť, či odosielateľ formulára je človek alebo stroj. V súvislosti s Turingovým testom a anonymnou komunikáciou cez počítač mi napadla ešte jedna vec – na internete nikto nevie, že osoba, ktorá práve číta tento text je veвериčka.

V roku 1980 známy filozof mysle a umelej inteligencie John Searle navrhol myšlienkový experiment, ktorý elegantne ukončil naše úvahy o tom, či počítače dokážu mať myseľ a či dokážu rozumieť tomu, čo vlastne robia. Týmto vlastne vytvoril aj pomerne silný argument proti Turingovmu testu. Experiment je známy, ako argument čínskej izby [3]. Predstavte si izbu rozdelenú na dve časti, ktoré medzi sebou môžu anonymne komunikovať (a je to jedno, či pomocou počítača alebo podsúvaním si odkazov popod dvere). V jednej časti izby je Číňan, ktorý roztopašne posieľa vety napísané po čínsky do druhej časti izby. V tej druhej časti izby je Searle, ktorý nevie po čínsky ani slovo a tiež hrubá kniha pravidiel napísaná v jazyku, ktorému Searle rozumie. Kniha pravidiel je vlastne návod, ako prepisovať čínske znaky na iné čínske znaky. Táto kniha prakticky simuluje inštrukcie, ako naložiť so vstupnými dátami a ako z nich vygenerovať dáta výstupné. Searle simuluje mechanizmus počítača, ktorý s týmito inštrukciami narába. Takže, Searle dostane vetu po čínsky, zalistuje v knihe a skúsi prepísať túto vetu podľa pravidiel na inú čínsku vetu. Túto vetu pošle späť bez toho, aby mal páru o tom, čo sa ho pýtajú a aká je odpoveď. Dialógu nerozumie. Číňan si výborne porozpráva a Searle si utrie pot zo svojho čela. Čo z toho vyplýva? Searle a kniha pravidiel vytvárajú dohromady systém, ktorý komunikuje po čínsky bez toho, aby po čínsky rozumel. Takýto systém by zrejme napriek všetkému prešiel Turingovým testom. A takto pracujú dnešné počítače. Na rozdiel od ľudí použitý systém nerozumie obsahu a významu symbolov, s ktorými pracujú. V systéme nemá myseľ a inteligenciu ako ľudia. Len sa správa, ako by ho mali<sup>2</sup>.

<sup>2</sup> Jedná sa o názor autora tejto kapitoly, oponent kapitoly sa domnieva, že z myšlienkového experimentu Searleho len vyplýva, že použitá metóda nemá efemérnu vlastnosť vedomia,

Zaujímavých filozofických otázok a odpovedí spojených nielen s inteligenciou strojov je mnoho. Ale, aby som sa konečne prepracoval k tomu, o čom chcem vlastne písať, myslím, že tu je to správne miesto, kde je čas zredukovať naše veľké ambície a pozrieť sa na veci prakticky. Umelá inteligencia, ako veda, sa nesnaží modelovať človeka alebo jeho myseľ ako celok, vo všetkej komplexnosti. Cieľom je hlavne odpozorovať, zamerať a inšpirovať sa jednoduchými základnými ľudskými myšlienkovými operáciami, činnosťami alebo biologickými štruktúrami a procesmi. Pokúsiť sa ich modelovať a použiť ich pri riešení rôzneho typu úloh, ktoré si vyžadujú inteligenciu. Podľa mňa sa zatiaľ pokojne môžeme uspokojiť s cieľom vytvárať stroje, ktoré sa inteligentne správajú. Okrem iného je celkom dobré uvedomiť si, že sa bavíme o strojoch. A tu sú reálne očakávania celkom na mieste. Zvlášť pri predstave, s akou ľahkosťou dokážu aj ľudia zmeniť svoj názor, keď sú postavení pred dostatočne zložitú úlohu (samozrejme, individuálne). Každopádne, výskum a vývoj aktívne napredujú, a to, čo je všedné a čo inteligentné sa časom mení. A mašinky sa stávajú inteligentnejšie a inteligentnejšie, až jedného dňa budú najinteligentnejšie.

## 1.1 Základné prístupy v umelej inteligencii

Umelá inteligencia, ako vedná disciplína, zahŕňa extrémne veľa poddisciplín a diametrálne odlišných prístupov, ako automaticky a efektívne riešiť celé triedy úloh rôzneho typu. Ak by ste nahnali skupinu expertov na rôzne oblasti umelej inteligencie do jednej miestnosti, zabezpečili, aby sa nepokúšali vzájomne mlátiť a nechali ich spísať všetky oblasti a prístupy, ktoré poznajú, dostali by ste zoznam, ktorého dĺžka by zmiatla každého, kto sa len trochu zaujíma o umelú inteligenciu. Ja sám si rozhodne netrúfam vymenovať všetky tieto disciplíny. Ponúkol by som dva základné prístupy, a to symbolický a sub-symbolický:

- **Symbolická umelá inteligencia** (prístup zhora – nadol): vychádza z predpokladu, že človek uvažuje a komunikuje vo forme symbolov. Tieto symboly reprezentujú objekty, udalosti, situácie, ich vlastnosti, vzájomné vzťahy. Predpokladá sa teda, že vyššie myšlienkové procesy prebiehajú na úrovni symbolov. Do tejto kategórie patria systémy pre riešenie úloh, hranie hier a logické dokazovanie, plánovače, pravidlové a expertné systémy, systémy modelujúce sémantické vzťahy (tieto budeme analyzovať), ale napríklad tiež systémy pre spracovanie prirodzeného jazyka alebo rozvrhovanie (týmto sa zaoberať nebudeme). Vo všeobecnosti tu patria systémy, ktoré nejako využívajú formálne usudzovanie.
- **Sub-symbolická umelá inteligencia** (prístup zdola – nahor): vznikla ako alternatíva, ktorá vychádza z biologickej motivácie. Namiesto symbolov sa tu používajú štruktúry, ktoré pracujú na úrovni čísel. Dôraz sa tu kladie na silu paralelných výpočtov. Sub-symbolika sa nechala inšpirovať napríklad tým, ako fyziologicky funguje ľudský mozog. Typickým reprezentantom sú tu učiace sa neurónové siete, ktoré výborne pokryli niektoré temné stránky symbolických

---

ktoré v umelej inteligencii a v kognitívnej vede je len ťažko špecifikovateľné a úpostihnutelné. Domnievame sa, že existencia mysle nevyžaduje existenciu vedomia.

systémov, ako napríklad celkom neprehliadnuteľné problémy s rozpoznávaním obrazových (či zvukových) sensorických vnemov alebo predikciou. Neurónové siete sa tiež výborne hodia na klasifikačné alebo zhlukovacie úlohy alebo, vo všeobecnosti, na úlohy reprezentovateľné matematickou funkciou, ktorej parametre sa neurónová sieť snaží nájsť učením.

Iná biologická inšpirácia vychádza z darwinovskej evolúcie. Populácia jedincov v genetických alebo evolučných výpočtoch mutuje a kríži sa medzi sebou. Každý jedinec, podobne, ako u neurónových sietí, predstavuje hľadané parametre definovanej funkcie, resp. riešenia úlohy. Lepší prechádzajú do ďalšej generácie (samozrejme, nie nevyhnutne, život takto nefunguje vždy, stačí si predstaviť intelektuála – kulturistu pod kolesami auta). Evolučné prístupy sa výborne hodia napr. na optimalizačné alebo rozvrhovacie úlohy. Len rýchlosť ich výpočtu nie je vždy práve veľmi veľká.

Ďalšou triedou základných sub-symbolických prístupov sú fuzzy systémy, ktoré majú schopnosť pracovať s neurčitými pojmami. Napr. pojem “teplota” môže byť rozdelený na viacero možných stavov, ako “chlad”, “izbová”, “teplo”, “horúco”. Jednotlivé stavy sú modelované pomocou matematických funkcií, tzv. funkcií príslušnosti. Skutočná nameraná teplota potom s istou pravdepodobnosťou môže spadnúť do viacerých stavov, napr. “izbová”: 43%, “teplo”: 57%. Pomocou takto rozdelených pojmov sú poskladané pravidlá, s ktorými systém pracuje. Používajú sa hlavne na diagnostiku alebo rýchle rozhodovanie pri riadení systémov.

Symbolické a sub-symbolické systémy síce boli vyvinuté na základe úplne iných predpokladov, ale nepretržitá práca nadšenej vedeckej komunity zabezpečila to, že oba prístupy majú čím ďalej tým väčší záber. Sú schopné riešiť omnoho viac typov úloh, symbolické systémy sa už dávno dokážu učiť z príkladov a skúseností, sub-symbolické systémy dokážu pracovať so symbolmi. Existuje množstvo hybridných prístupov, ktoré kombinujú výhody symboliky a subsymboliky. Tiež existujú úplne nové, samostatné, špecifické paradigmy, ako umelý život, behaviorálne modely alebo agentové (a multi-agentové) systémy. My v tomto texte budeme sa zaoberať základnými symbolickými prístupmi.

## 1.2 Symbolové systémy<sup>3</sup>

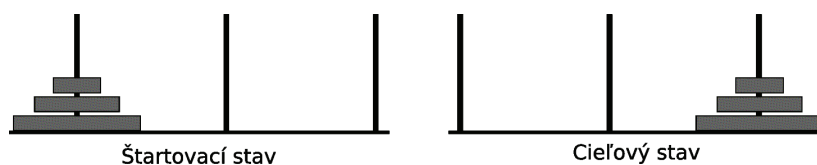
Ako bolo skôr povedané, výskum v oblasti symbolickej umelej inteligencie je založený na predpoklade, že človek uvažuje a komunikuje vo forme symbolov. Ak to chceme premeniť na výpočet, tak predpokladajme, že fenomén ľudskej inteligencie môže byť redukovaný na proces manipulácie so symbolmi. A nie je podstatné, aké médium je nosičom týchto symbolov. Nemusí to byť nevyhnutne ľudský mozog. V roku 1976 A. Newell a H. A. Simon prišli s tzv. hypotézou fyzického symbolového systému [4].

<sup>3</sup> V tejto kapitole sa prelínajú nedôsledne dva termíny „symbolický“ a „symbolový“. Prvý termín „symbolický“ patrí medzi zaužívanú terminológiu, tak v slovenskej, ako aj v českej literatúre o umelej inteligencii a kognitívnej vede.

Tento systém obsahoval množinu prvkov – symbolov, ktoré sa mohli podľa špecifických pravidiel spájať do zložitejších výrazov a štruktúr. A tieto mohli tvoriť ešte zložitejšie výrazy. Okrem symbolov a ich štruktúr systém obsahoval množinu procesov – operácií, ktoré vedeli s týmito štruktúrami manipulovať. Tieto procesy mohli vytvárať, meniť, reprodukovat' alebo likvidovať štruktúry, ktoré boli práve aktuálne. Fyzický symbolový systém je vlastne stroj, ktorý v priebehu svojej aktivity spontánne produkuje stále sa meniacu množinu symbolových štruktúr.

Pointa je v tom, že tieto symboly a z nich zložené výrazy majú predstavovať „mentálne“ štruktúry. Každý symbol, každá štruktúra by mala reprezentovať objekty, ich vlastnosti a vzťahy vo svete (najlepšie reálnom), v ktorom systém pracuje, nech sú akokoľvek zložité. Štruktúry symbolov by teda mali byť schopné modelovať s väčšou, či menšou mierou detailnosti nejakú časť reality. Na druhej strane, procesy, ktoré tie to štruktúry menia, by potom mali reprezentovať usudzovanie. Ak v jednom okamihu systém obsahuje konkrétny popis (stav) svojho prostredia a tento popis nejakým spôsobom zmení, táto zmena predstavuje nový stav prostredia, teda novú situáciu. Ak by sme takémuto systému dali požadovaný cieľový stav, tento systém by sa pokúsil nájsť takú postupnosť operácií, ktorá by jeho aktuálny stav zmenila na cieľový. Postupnosť operácií by sme potom mohli nazvať plánom, ako dosiahnuť cieľ alebo usudzovaním, ako dôjsť k požadovanému záveru. V tejto súvislosti musíme poznamenať, že nám nejde len výlučne o problematiku plánovania. Vo všeobecnosti, predmetom nášho záujmu je problém, ako symbolový systém dosahuje svoje ciele, nech ide o akýkoľvek typ úlohy a ako systém pracuje.

Viac konkrétnejšie, predstavte si, že by takýto systém vedel symbolmi reprezentovať napríklad známu úlohu Hanojských veží. Tento systém nech má k dispozícii operácie, pomocou ktorých bude vedieť prekladať jednotlivé kotúče. Povedzme, že štartovací a cieľový stav sú požadované v tvare uvedenom na obr. 1. Úlohou je prekladať kotúče dovedy, kým sa z počiatočného stavu nedosiahne cieľový stav. Samozrejme, operácie musia byť definované tak, aby sa systém nepokúšal o exotické a nezmyselné ťahy, ako napríklad nekládol väčší kotúč na menší alebo sa nejakého kotúča zbavil tak, že by ho odložil niekam vedľa ako prechodne nepotrebný. To, čo sa vo vnútri systému udeje, je typický príklad princípov fungovania symbolového systému.



**Obr. 1.** Príklad štartovacieho a cieľového stavu pre úlohu Hanojských veží.

Pre úplnosť poznamenajme, slovo „fyzický“ symbolový systém bol zavedený preto, lebo tento systém by mal skutočne existovať v reálnom fyzickom prostredí, do ktorého

by mohol aj fyzicky zasahovať. Logicky by sa preto dalo čakať, že by mal mať možnosť vnímať externé prostredie nejakými senzormi a vedieť pretransformovať tieto vnemy do symbolových štruktúr.

Simon a Newell odvážne postulovali, že fyzický symbolový systém spĺňa nevyhnutné a dostačujúce podmienky inteligentného správania sa [4]. Tento názor sa nestretol so všeobecným súhlasom, medzi hlavný protiargument patrila názor, že takýto systém síce bude vedieť hrať šach, ale nerieši napríklad problém spracovania obrazu alebo zvuku na úrovni signálov, a s tým súvisiace problémy s rozpoznávaním objektov zo senzorických vstupov, a veľa ďalších. Nás toto teraz trápiť nebude, my si tu momentálne vystačíme aj s tým, čo máme.

Väčšina konkrétnych systémov so symbolickou umelou inteligenciou celkom dobre vyhovuje predlohe, ktorú dal fyzický symbolový systém. Ak sa zameriame na to, čím sa vlastne zaoberá symbolika, vzniknú tieto dva základné body:

1. **vývoj údajových štruktúr** pre reprezentáciu znalostí,
2. **vývoj procedúr**, ktoré umožňujú cieľnú manipuláciu s týmito štruktúrami, teda uvažovanie.

Znalosti v takýchto systémoch sú zvyčajne **deklaratívne**, teda vo forme faktov, ktoré popisujú **čo** systém vie o svete, ale často aj **čo** vie o tom, **čo** vie (metavedomosti). Niekedy je potrebné mať aj **procedurálne** znalosti o tom, **ako** vykonávať konkrétnu činnosť. Aby bolo možné s reprezentáciou znalostí zmysluplne narábať, musí mať táto reprezentácia nejaké pravidlá. Systém musí vedieť popisovať svet a vyjadrovať sa o ňom v nejakom zrozumiteľnom jazyku (v ideálnom prípade zrozumiteľnom aj pre človeka). Ak by sme to chceli naštíť na fyzický symbolový systém, abecedou jazyka by boli symboly a syntaxou by boli pravidlá, ako skladať symboly do zložitejších výrazov. Každý správny jazyk má tiež sémantiku, teda pravidlá, ako interpretovať svoje výrazy. Všetky tieto javy priamo ovplyvňujú bohatosť a vyjadrovaciu schopnosť jazyka. Teda to, čo všetko je možné popísať, vyjadriť, na akej úrovni detailnosti, a podobne.

Dôležitou vlastnosťou symbolických reprezentácií je, že systém nemá znalosti o všetkom, čo sa v jeho svete môže udiť. Zďaleka nemá vymenované všetky fakty alebo všetky alternatívy situácií. Systém zvyčajne vie len základné a nevyhnutné informácie pre popis sveta. A práve **operácie**, pomocou ktorých systém uvažuje, tvoria nástroj na to, ako odvodiť z existujúcich faktov nové informácie, ktoré sa v znalostiach pôvodne nenachádzali.

### 1.3 Ciele príspevku

Cieľom je formulovať základné klasické techniky, s ktorými symbolika pracuje. Prejdeme niekoľko najznámejších a najrozšírenejších spôsobov reprezentácie znalostí a zameriame sa na základné princípy ich fungovania. Tento text je určený pre tých, ktorí s umelou inteligenciou začínajú. Preto je cieľom podať fakty úplne jednoducho, čitateľne a čo najjasnejšie. Na základnú podstatu problematiky sa budeme snažiť pozeráť prakticky, takže pre každú techniku uvedieme aj základné algoritmy, ktoré by ste si mali byť schopný bez väčších problémov implementovať. Napriek tomu, že väčšina textu sa len kľže po povrchu symbolových metód umelej inteligencie, niektoré časti textu si vyžadujú už určité aplikačné skúsenosti v matematike a algoritmizácii.



Budeme sa zaoberať so stavovým priestorom, rôznymi technikami prehľadávania, budeme automaticky riešiť hlavolamy, hrať hry a víťaziť. Vyskúšame si interaktívnu, počítačom riadenú diagnostiku pomocou expertných systémov. Ukážeme si, ako sformalizovať svet pomocou predikátovej logiky a ako prinútiť počítač automaticky dokazovať tvrdenia. A nakoniec pristúpime k štúdiu neklasických logík, ktoré umožňujú sémanticky modelovať znalosti v otvorenom svete, v ktorom to, že niečo nevieme, ešte neznamená, že to neexistuje.

Upozorňujeme, že ak by sme chceli ísť aspoň ako tak do detailov, každá z týchto oblastí by si zaslúžila prinajmenšom jednu celú knihu. V tomto texte tiež bohužiaľ nebude priestor na to, aby sme sa zaoberali existujúcimi systémami alebo rôznymi implementáciami jednotlivých prístupov. Je ich priveľa a sú špecifické. Nie je možné nejaké odporučiť bez porovnania a zdôvodnenia.

Mottom pri písaní tohto textu bolo to, čo Albert Einstein umne vyrezal Occamovou britvou do éteru: „Robme veci tak jednoducho ako sa len dá, ale nie jednoduchšie.“

## 2 Stavový priestor, riešenie úloh a hranie hier

Ak riešime nejakú úlohu netriviálnej zložitosti, máme k dispozícii celý arzenál možností, ako k riešeniu pristupovať. Ak sme postavení pred triviálnu zložitú úlohu: chceme sa napiť čaju, je dobré mať plán, ako to urobiť. To vyústi do postupnosti činností, ktoré musíme vykonať, aby sme dosiahli zvolený cieľ. Odvážne si trúfam tvrdiť, že najčastejšie: chytíme pohár, priložíme si ho k ústam a patrične nakloníme. Toto znie ako ideálne riešenie, ale všetko vždy závisí od toho, ako sme schopní naplánovať si všetky činnosti tak, aby bolo riešenie čo najlepšie. Samotné odpitie si z čaju je možné vykonať nespočítateľným množstvom spôsobov. Všetko závisí od toho, aké všetky aktivity máme k dispozícii, a čo majú tieto činnosti za následok. Pred tým, ako úspešne uchopíme šálku, to pokojne môžeme zobrať cez obývačku, v ktorej pootvárame okná. Nie je to síce ideálne riešenie, ale možné to je. Akurát to dlhšie trvá. A to mi ešte môžem poznamenať, že šálku si môžeme priložiť úplne inam a Murphyho zákony zabezpečia, aby horúci čaj skončil presne na mieste, ktoré by nikto neočakával ako konečné riešenie. Menej zdatnému riešiteľovi sa tiež môže stať, že sa k riešeniu vôbec nedostane. Napríklad opakovaním rovnakej činnosti dookola (chodť do obývačky, chodť do kuchyne, chodť do obývačky, ...). Toto my, vedci, vznešene voláme *cyklus*. Aj keď je takáto činnosť fyzicky možná, predsa je len lepšie vyhnúť sa jej.

Ak si to celé zjednodušíme a dáme tomu trochu formálnejšiu podobu, povedzme, že situácia, v ktorej sa momentálne nachádzame nám popisuje stav riešenia úlohy. Na popis situácie vieme použiť konkrétny počet príznakov. Len úplne intuitívne, príklad situácie môže byť napríklad: *som v kuchyni, čaj je v kuchyni, nedržím šálku, držím šálku*, a podobne. Povedzme, že máme k dispozícii konkrétny počet aktivít, napríklad: *chyt' šálku pravou rukou, chyt' šálku ľavou nohou, zdvihni šálku, chod' do obývačky* a ďalšie. Ak v nejakej situácii použijeme nejakú činnosť, stav sa zmení. Ak napríklad v situácii *nedržím šálku a šálka je na stole* použijem činnosť *zdvihni šálku*, nový popis stavu môže byť *držím šálku a šálka nie je na stole*. Priestor všetkých možných situácií,

ktoré môžeme dosiahnuť použitím všetkých možných činností sa volá stavový priestor. A každá situácia popisuje jeden konkrétny stav riešenia úlohy. Stavový priestor je jedna z najjednoduchších a najintuitívnejších reprezentácií znalostí vôbec. Tak sa teraz ideme zaoberať s tým, ako úlohy v stavovom priestore riešiť.

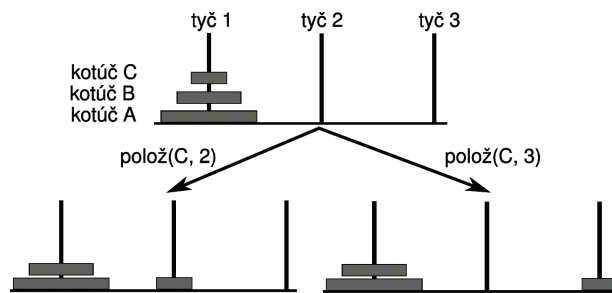
## 2.1 Riešenie úloh

Pred tým, ako sa so záujmom pustíme do samotného riešenia úloh v stavovom priestore, si ešte dáme jednoduchý a úplne konkrétny príklad, na ktorom budeme študovať details problému riešenia úloh. V predchádzajúcej časti textu som prisľúbil, že budeme prekladať kotúče na Hanojských vežiach. Tu svoj sľub plním.

Popis jedného stavu riešenia bude veľmi jednoduchý. Máme tri tyče a tri kotúče rôznej veľkosti. Kotúče môžu byť jeden na druhom nastoknuté na niektorej z tyčí. Úlohou je poprekladať kotúče zo štartovacieho do cieľového stavu (viď obr. 1.). V jednom kroku môžeme preložiť len jeden kotúč. Pravidlá sú takéto:

- prekladať môžeme len voľný kotúč, na ktorom nie je položený iný kotúč
- kotúč môžeme položiť len na väčší kotúč alebo na prázdnu tyč

V tomto prípade náš jednoduchý svet bude obsahovať tri tyče označené číslami 1, 2 a 3 (zľava doprava) a tri kotúče označené písmenami A, B, C, kde A je najväčší a C najmenší. Bohato si vystačíme s jedinou činnosťou: *prelož(Kotúč, Tyč)*. Napríklad, ak chceme preložiť najväčší kotúč na tretiu tyč, operácia bude volaná s argumentami: *prelož(A, 3)*. Teda, z každého stavu riešenia úlohy môžeme dostať toľko nových stavov, koľko kotúčov je možné preložiť všetkými možnými spôsobmi v rámci pravidiel. Keď budeme implementovať našu operáciu, je celkom na mieste zabezpečiť, aby sa nepokúšala robiť neprirozené triky a nevyťahovala kotúče zospodu alebo nekládla väčší kotúč na menší. Operácia musí vytvárať len stavy, ktoré vyhovujú pravidlám. Príklad popisu stavu pre Hanojské veže a aplikáciu použiteľných operácií je na obr. 2.



Obr. 2. Príklad reprezentácie stavu a použitia operácií pre Hanojské veže.

Vždy z nejakého stavu začíname, máme definovanú skupinu nečinne postávajúcich a prevínilo sa tváriacich cieľových stavov (cieľom nemusí byť nevyhnutne jediný požadovaný stav) a skúšame pomocou našej operácie prekladať kotúče dovtedy, kým nedosiahneme niektorý z cieľov. Ak sa tak stane, úlohu sme vyriešili a máme dôvod na

radosť z dosiahnutého cieľa. Riešenie úlohy bude vlastne postupnosť operácií, ktoré sme použili na to, aby sme poprekladali kotúče zo štartovacieho do cieľového stavu. My budeme takúto postupnosť hrdo nazývať **plán**.

**Stavový priestor je veľmi jednoduché reprezentovať grafom s orientovanými hranami.** Uzly grafu sú jednotlivé stavy a hrany reprezentujú operácie na tieto stavy nasadené. Stavový priestor je grafom preto, že zo štartovacieho stavu sa často môžete dostať do toho cieľového pomocou viacerých rôznych postupností operácií. Ďalším jednoduchým dôvodom je, že v stavovom priestore vám veľmi ľahko môžu vzniknúť cykly, stačí napríklad, ak preložíme kotúč a hneď ho preložíme späť.

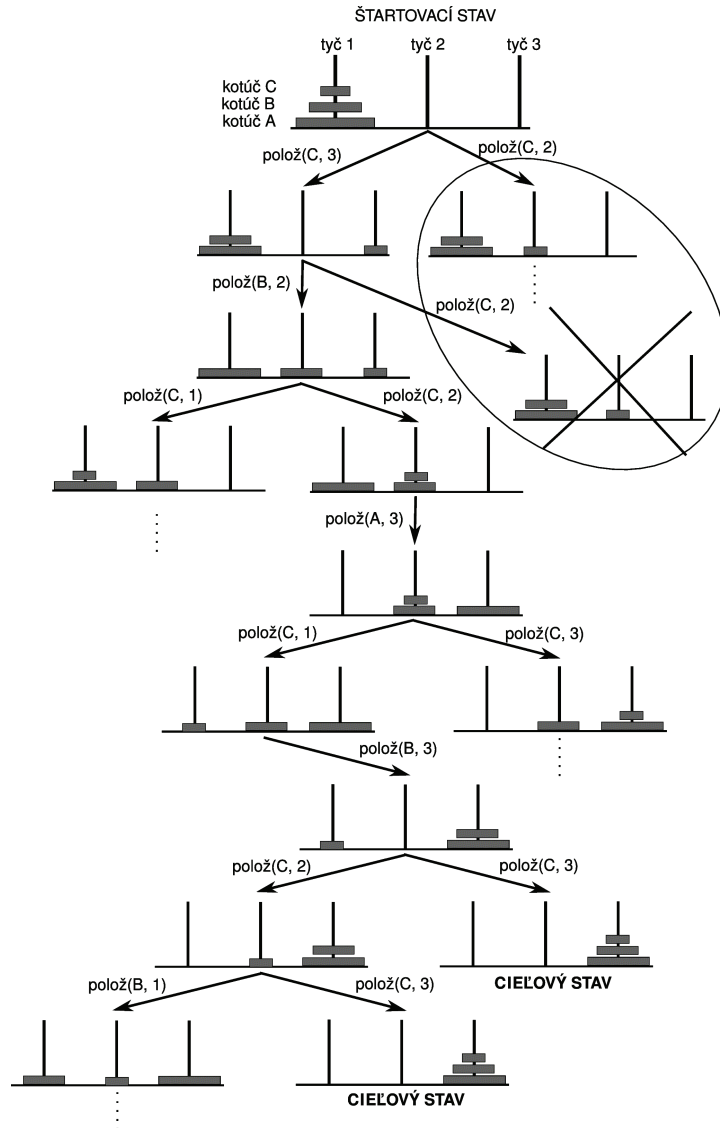
**Každá použitá operácia vytvára nové stavy,** hrany sú teda orientované v smere z pôvodného do nového stavu. Budeme hovoriť, že ak na nejaký uzol (stav) aplikujeme operáciu, tak táto operácia *expanduje* daný uzol a generuje nové uzly – *potomkov* daného uzla (ich *predka*). Riešenie úlohy nájdenia cieľového stavu je potom bez problémov možné uskutočniť, ako prehľadávanie tohto grafu. Výsledkom prehľadávania bude cesta v grafe od štartovacieho do cieľového stavu. A to je ten očakávaný plán. Je tu ale ešte zopár malých komplikácií, ktorým sa musíme s láskou povenovať, aby všetko fungovalo tak, ako má. Asi by nebolo veľmi efektívne skúšať znova a znova to, čomu sme už venovali pozornosť skôr. Nemá teda veľký zmysel prehľadávať znova stavy, ktoré už boli prehľadané skôr. Na druhej strane by bolo vynikajúce, keby sme vedeli prehľadať celý stavový priestor a na žiadny stav pri tom nezabudli (riešeniam je totiž vlastné, že môžu byť kdekoľvek, a často tam, kde by sme ich normálne nehládali). Aby sme tieto požiadavky zabezpečili, k popisu stavu a operáciám sa pridáva **riadiaca stratégia**, ktorá by mala zabezpečiť, že:

- pri prehľadávaní stavového priestoru nebude vynechaný žiaden stav (ak riešenie existuje, nájdeme ho)
- každý zo stavov bude prehľadávaný maximálne raz (takže sa nebudeme motať dookola v cykloch)

Takto definovaná stratégia prehľadávania spôsobí, že **prehľadávaná časť stavového priestoru bude strom a nie graf**. To je zabezpečené tým, že každý stav bude prehľadávaný najviac raz (a graf bez cyklov je strom). Príklad prehľadávania stavového priestoru je na obr. 3.

Riadiaca stratégia je vlastne algoritmus, ktorý určuje, kedy a v akom poradí treba aplikovať jednotlivé operácie. Jedným z neprehliadnutelných problémov prehľadávania stavového priestoru je jeho neefektivita. Ak je náš stavový priestor obrovský, môže sa stať, že úplne zbytočne prehľadáme jeho veľkú časť, ktorá riešenie neobsahuje. Veľkosť prehľadávania stavového priestoru sa dá zmenšiť tým, že do riadiacej stratégie zavedieme ďalšie znalosti o riešenom probléme. Tieto dodatočné informácie slúžia na to, aby riadiacej stratégia pomohli rozhodnúť, ktorým stavom bude výhodnejšie pokračovať, prípadne, ktoré stavy bude lepšie úplne ignorovať, ako neperspektívne. Takéto znalosti zvyčajne vychádzajú z praktických skúseností s riešením danej úlohy a nemusia nevyhnutne viesť k riešeniu. Tým, že ich použijeme, sa snažíme zvýšiť pravdepodobnosť, že jednak pretraverzujeme menšou časťou stavového priestoru a jednak sa k riešeniu dostaneme skôr. Takéto dodatočné expertné znalosti nesú honosné pomenovanie **heuristické** (heuristiky). A podľa toho, či riadiaca stratégia využíva heuristiky, sa techniky prehľadávania rozdeľujú na informované

alebo (veľmi výstižne) slepé. Pohľady teraz obrátíme k základným metódam prehľadávania.



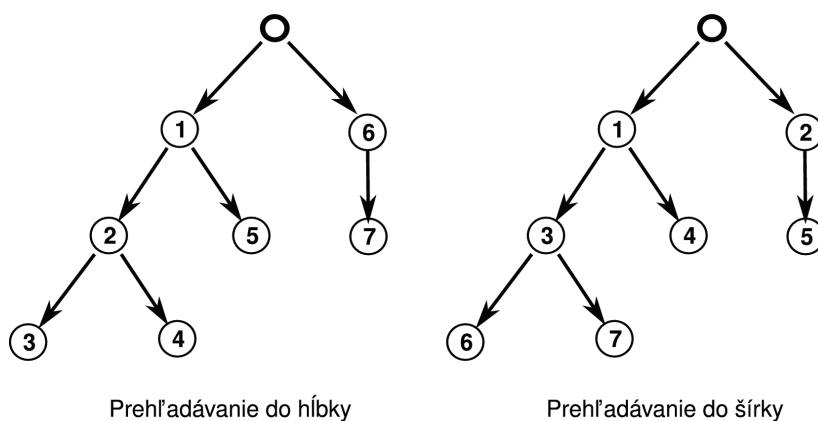
**Obr. 3.** Príklad časti stavového priestoru Hanojských veží reprezentovaného stromom.

### Základné stratégie prehľadávania

Tuto podkapitolu odporúčam veľmi pozorne prečítať, pretože techniky, ktoré budeme používať pri každej reprezentácii. Konečne pristúpime k formulácii dôležitej myšlienky, že väčšina reprezentácií znalostí sa bude dať nakoniec previesť na graf, čo sa dá vyjadriť aj pomocou centrálneho hesla symboliky: „graf si a na graf sa obrátiš“.

Tieto grafy budú hlavným predmetom nášho záujmu. Na tomto mieste ešte raz pripomínam, keďže chceme každý uzol (stav) prehľadávať maximálne jediný raz, prehľadávaná časť tohto grafu bude strom. A preto sa pri relatívne komplikovaných opisoch stratégií prehľadávania zameriame priamo na stromy.

Na tomto mieste nebudeme líčiť členité zákutia konkrétnych metód usudzovania. K tým sa dostaneme neskôr a v konkrétnejšom svetle. Teraz nám pôjde o to, ako sa dostať v strome od štartu k cieľu, teda ako nájsť cestu medzi dvoma uzlami. Máme k dispozícii dve základné možnosti, buď to zoberieme **do hĺbky** alebo **do šírky**. Prakticky povedané, rozdiel je len v tom, v akom poradí budeme expandovať uzly. Ako vlastne prehľadávanie funguje: začneme štartovacím uzlom a postupne naň aplikujeme operácie, čím vlastne generujeme strom. A to robíme dovtedy, kým nenájdeme cieľ.



Obr. 4. Príklad prehľadávania stromu do hĺbky a do šírky.

Podľa stratégie prehľadávania do hĺbky budú najskôr prehľadávané uzly, ktoré sú na konci najdlhších ciest generovaného stromu. Po expandovaní uzla sa ako ďalší uzol pre prehľadávanie vyberá najhlbší z novo vygenerovaných uzlov a takto sa pokračuje, až kým sa tento proces z nejakých dôvodov nezastaví. Ukončíme to zvyčajne vtedy, ak na stav reprezentovaný najhlbším uzlom už nie je možné aplikovať žiadnu operáciu alebo vygenerovaný stav už bol prehľadávaný (práve týmto pri prehľadaní zabraňujeme vzniku cyklov). Ak riadiaci algoritmus vysnorí uzol, pre ktorý nie je možné z nejakých dôvodov pokračovať v prehľadaní, vyberie sa najhlbší z vrcholov, ktoré ešte neboli expandované. V podstate sa vyberá ďalšia možnosť, kde bolo poslednýkrát možné rozhodnúť sa inak. Takto sa pokračuje, kým sa nenájde riešenie alebo sa neprehľadá celý stavový priestor.

Stratégia prehľadávania do šírky bude vždy prednostne expandovať uzol s najmenšou hĺbkou. Ak si predstavíte strom, tak v každom kroku prehľadávania sa expandujú všetky uzly na jednej úrovni a vygeneruje sa celá ďalšia úroveň. Prehľadávanie do šírky vždy nájde ako prvé najkratšie riešenie. Ak chcete nájsť napr.

všetky možné riešenia, hlavne z implementačného hľadiska, je lepšie prehľadávať strom do hĺbky.

Príklad použitia oboch stratégií pri prehľadávaní je znázornený na obr. 4. Uzly sú číslované v poradí, v akom boli expandované. Konkrétne algoritmy vám v ďalšom texte vždy podhodím ako prehľadávanie do hĺbky a rekurzívne, so všetkou roztopašou.

Nesmieme zabudnúť ešte na dôležitý aspekt súvisiaci so stratégiami prehľadávania. Zatiaľ sme sa bavili o poradí, v akom budú generované uzly stromu, ak začneme od štartovacieho stavu (koreňového uzla v strome). Ak sa sústredíme na smer prehľadávania, aj tu máme viac možností. Zatiaľ padol len prípad, keď začneme štartovacím a končíme cieľovým uzlom. Iste aj vás napadne, že to vôbec nie je jediná možnosť. Kvôli jednoduchosti sa budeme zaoberať hlavne s týmto typom prehľadávania. Aspoň spomeniem, že k dispozícii máme tiež opačný prístup, ktorý začína cieľovým stavom. Metóda je trochu iná. Na cieľový stav sa aplikujú operácie tak, aby sa cieľ rozdelil na podúlohy, ktorých zložitosť je menšia. A tie sa zasa rozdelia na ďalšie menšie podúlohy, tento postup opakujeme tak dlho, až dostaneme jednoduché úlohy, ktorých riešenie je triviálne. Preto sa takto generovanému stromu často nazýva reprezentácia stavového priestoru **redukciou problému**. O tom, že neexistujú metódy, ktoré postupujú oboma smermi, sa ani nepokúšajte pochybovať. Poďme sa teda pozrieť na samotné riešenie úloh.

### Slepé prehľadávanie

Naším cieľom je zabezpečiť to, aby bol náš stavový priestor kompletne prehľadaný. V tejto súvislosti sa obvykle uvádza, že použité algoritmy sa elegantne formulujú pomocou rekurzie (aj keď niekoľko nerekurzívnych algoritmov v ďalšom texte uvediem)

Poďme na pozrieť na jednoduchú a nerekurzívnu verziu algoritmu prehľadávania do hĺbky. Strom sa vždy rozrastie o stavy, ktoré vieme získať z momentálne najhlbšieho známeho uzla. Rozrastie sa však len o tie stavy, ktoré ešte neboli prehľadávané. Algoritmus používa dve globálne premenné. Zoznam OTVORENÝ, ktorý obsahuje tie stavy, ktoré ešte neboli preskúmané a UZAVRETÝ, ktorý obsahuje už prehľadávané stavy a bedlivo zabraňuje vzniku cyklov. Zo zoznamu OTVORENÝ sa na ďalšie prehľadávanie stále vyberá prvý prvok a pri každej aplikácii operácií sa nové stavy vkladajú na začiatok zoznamu OTVORENÝ. To zabezpečuje, že nové stavy budú prehľadávané skôr, a teda, že algoritmus pôjde skutočne do hĺbky. Algoritmus nám prehľadá celý stavový priestor a nájde cieľový stav, ale existuje jeden malý problém. Ak chcete, aby vám povedal plán, ako riešenie našiel (teda postupnosť použitých operácií), musíte si niekde pamätať, akým spôsobom boli jednotlivé uzly expandované (alebo to riešiť smerníkmi v C-čku). Volanie funkcie je jednoduché: `hladaj_hlboko(štart, cieľ)`, `štart` a `cieľ` sú stavy. Pred volaním algoritmu si buďte istý tým, že zoznamy OTVORENÝ a UZAVRETÝ sú prázdne. Algoritmus máte na obr. 5.

Veľmi elegantnou a v umelej inteligencii obľúbenou implementáciou prehľadávania sú rekurzívne algoritmy. Rekurzívne prehľadávanie výborne kopíruje štruktúru prehľadávaného stromu. Ak sa nájde riešenie, máte dve možnosti, ako získať

plán. Závisí to od toho, či je vaším cieľom nájsť všetky riešenia alebo sa uspokojíte s tým, ktoré sa našlo ako prvé (to, že sa riešenie našlo ako prvé, samozrejme neznamená, že je najkratšie).

```

funkcia hľadaj_hlboko(štart, cieľ){
    kým(nie_je_prázdny(OTVORENÝ)){
        stav = vyber_prvý_prvok(OTVORENÝ)
        vlož_do_zoznamu(stav, UZAVRETÝ)
        pre(i = 0; i < počet_operácií; i++){
            ak(je_aplikovateľná(operácia[i], stav)){
                nový_stav = aplikuj(stav, operácia[i])
                ak(neobsahuje(nový_stav, UZAVRETÝ)){
                    vlož_na_zočiatok(nový_stav, OTVORENÝ)
                }
            }
        }
        ak(obsahuje(cieľ, OTVORENÝ)){
            vypíš(„našiel som riešenie“)
            skonči s úspechom
        }
    }
    vypíš(„hľadaj na inom strome“)
    skonči s neúspechom
}

```

Obr. 5. Nerekurzívna verzia prehľadávania do hĺbky.

Súvisí to s tým, ako si v algoritme vyriešite budovanie cesty medzi štartovacím a cieľovým uzlom:

- **Dopredné budovanie cesty:** cestu si posielame ako lokálnu premennú, ktorú pri volaní ďalšieho vnorenia rozšírime o aktuálny stav. V každom stave prehľadávania máme vždy aktuálnu cestu vedúcu od štartu k aktuálne prehľadávanému uzlu. Ak trafíme cieľ, stačí cestu vypísať (alebo si ju zapamätať) a pokračovať v hľadaní ďalších riešení.
- **Spätné budovanie cesty:** v tomto prípade nepotrebujete žiadnu premennú. Hľadáme plán k prvému nájdenému cieľu a budeme s ním byť spokojní. V tomto prípade sa plán buduje pri vynáraní z rekurzívnych volaní. Keď nájdete riešenie, algoritmus končí a odovzdáva aktuálny stav rozšírený o operáciu, ktorou bol získaný, ako návratovú hodnotu. Každá úroveň rekurzívnej túto hodnotu dostane, pripojí k nej svoj stav a operáciu a vráti vyššej úrovni volania. Keď algoritmus ukončí rekuziu, návratovou hodnotou je plán poskladaný postupne všetkými úrovňami vnorenia.

Ukážeme si rekurzívne prehľadávanie s oboma spôsobmi budovania plánu, pretože je to veľmi užitočné. Obidva algoritmy budú používať globálnu premennú, zoznam UZAVRETÝ, v ktorom si budú pamätať už prehľadávané stavy (aby sme necyklili). Nesmieme zabudnúť tento zoznam pred každým volaním funkcie vyprázdniť. Volanie `hľadaj_plány_dopredu(štart, „“)` nám prehľadá celý stavový priestor a vypíše všetky nájdené plány. Volanie `hľadaj_plán(štart)` nájde prvý plán, ktorý je zostrojený spätným reťazením pri vynáraní z rekurzívnej. Algoritmy nájdete na obr. 6.

```

funkcia hľadaj_plány_dopredu(stav, plán){
    ak(je_cieľový(stav)){
        vypíš("mám plán: " + plán + stav)
    }
    inak{
        vlož_do_zoznamu(stav, UZAVRETÝ)
        pre(i = 0; i < počet_operácií; i++){
            ak(je_aplikovateľná(operácia[i], stav)){
                nový_stav = aplikuj(operácia[i], stav)
                ak(neobsahuje(nový_stav, UZAVRETÝ)){
                    hľadaj_plány_dopredu(nový_stav,
                                           plán +
                                           stav +
                                           operácia[i])
                }
            }
        }
    }
}

funkcia hľadaj_plán_späť(stav){
    plán = ""
    ak(je_cieľový(stav)){
        vypíš("Cieľový stav je nájdený, vynáram")
        vráť stav
    }
    inak{
        vlož_do_zoznamu(stav, UZAVRETÝ)
        pre(i = 0; i < počet_operácií; i++){
            ak(je_aplikovateľná(operácia[i], stav) AND){
                nový_stav = aplikuj(operácia[i], stav)
                ak(neobsahuje(nový_stav, UZAVRETÝ)){
                    plán = hľadaj_plán_späť(nový_stav)
                    ak(plán != ""){
                        plán = stav +
                               operácia[i] +
                               plán
                    }
                    vráť plán
                }
            }
        }
    }
    vráť plán
}

```

**Obr. 6.** Rekurzívna verzia prehľadávania do hĺbky so spätným budovaním cesty.

Plány, ktoré algoritmy skladajú, majú vždy tvar:

**stav<sub>1</sub> + operácia<sub>1</sub> + stav<sub>2</sub> + operácia<sub>2</sub> + ... + operácia<sub>n</sub> + cieľ**

Tieto algoritmy slepo (bez nejakej chytnej stratégie) prehľadajú prinajhoršom celý stavový priestor, nech by bol ako veľký. Pri komplikovanejších úlohách sa ale ľahko môže stať, že náš algoritmus narazí na počet možností obrovských rozmerov, čo môže mať za následok, že sa mu jednoducho nebude chcieť „skončiť“ (jeho časová náročnosť exponenciálne rastie s počtom stavov). A my nebudeme vedieť, či ešte stále hľadá alebo či „nedajbože“ sa nezacyklil, prípadne sa to celé môže skončiť na tom, že



počítač pripravíme o všetku operačnú pamäť. Na takéto prípady je dobré myslieť (aj keď, ak to už raz beží, zvykne byť neskoro). Jedným z použiteľných riešení je obmedziť prehľadávanie stromu len do istej hĺbky (počet úrovní vnorenia rekurzie). Buď sa riešenie nájde len v istej hĺbke stromu alebo sa nenájde, ale budeme to vedieť v rozumnom čase. Podsúva sa nám tu ale ešte jedna možnosť. Zefektívniť naše prehľadávanie pomocou dodatočných informácií o úlohe, teda použitím heuristik.

### Heuristické prehľadávanie

Slepé algoritmy systematicky prehľadajú celý stavový priestor bez akéhokoľvek uvažovania, nech sa deje, čo chce. Ak by ale sme do slepého prehľadávania implantovali dodatočné informácie o úlohe, mohli by sme mať akú takú nádej, že sa riešenie nájde skôr. Tieto informácie – **heuristiky** – často vychádzajú z praktických skúseností s daným typom úlohy a niekedy majú až intuitívny charakter. **Heuristiky sa najčastejšie používajú na hodnotenie stavov.** Takže pre konkrétny typ úlohy si definujeme funkciu, ktorá nám ku každému stavu priradí číselné hodnotenie. Toto hodnotenie znamená, nakoľko je tento stav perspektívny, teda nakoľko je vhodné sa ním vôbec zaoberať, a ak áno, kedy. Heuristické hodnotenie stavov zavedie do slepého algoritmu usporiadanie novo vygenerovaných stavov podľa ich hodnotenia. Logika káže, ak chceš ušetriť čas, zaoberaj sa najprv tým, čo vyzerá perspektívnejšie.

Je dôležité uvedomiť si, že **heuristika vôbec nemusí nájsť optimálny plán.** Dokonca, nemusí nájsť vôbec žiaden plán (to heuristiky niekedy robia). Skôr sa tu pokúšame veriť predpokladu, že ak o úlohe čosi vieme, použijeme to na nájdenie vhodného riešenie skôr, ako slepý algoritmus.

Klasickým prístupom, ktorý sa paradoxne napriek svojej jednoduchosti teší veľkej obľube (použitý všeobecne, ako prístup, nielen pre prehľadávanie stavového priestoru) je stratégia **hill climbing**. Ako už jej názov napovedá, základnou myšlienkou je pohybovať sa v stavovom priestore vždy v smere vyššieho hodnotenia stavov. Každý ďalší stav má teda lepšie (alebo aspoň také dobré) hodnotenie, ako ten predchádzajúci. A takto sa algoritmus štvorá v smere vyššieho a vyššieho hodnotenia, až kým nenarazí na cieľ, prípadne prekvapene nenarazí na fakt, že po ruke nie je žiaden nový stav s lepším hodnotením a cieľ je v nedohľadne. Keďže heuristika je len odhad, potom pre veľký komplikovanú úlohu je ťažké mať istotu, že hodnotiaci funkcia nebude stavy nadhodnocovať. Áno, nadhodnocovať znamená, že stav vyzerá perspektívnejšie, ako v skutočnosti je. Takto sa každý bežný algoritmus nechá radostne vlákať do pasce. A veľkým rizikom hill climbingu je práve jeho vďačná vlastnosť „blúdiť“ po kadejakých slepých uličkách bez toho, aby zavádil o čosi, čo by riešenie aspoň z diaľky pripomínalo. Každopádne, ak bude hodnotiaci funkcia perfektná, aj Hill Climbing bude perfektný.

Tento ľahkovážny prístup sa dá urobiť bezpečnejším celkom jednoducho. Stačí mu dať schopnosť vycúvať zo slepej uličky, keď sa mu v nej podarí skončiť a konečný stav evidentne nie je riešenie. Ďalšia stratégia, ktorá rozširuje hill climbing o schopnosť vrátiť sa do posledného stavu, kde bola možnosť výberu, sa volá **prvý najlepší** (Best First). Táto stratégia veľmi jednoducho upravuje slepé prehľadávanie. Keď generujete nové stavy, ohodnoťte ich a usporiadajte podľa hodnotenia. To je

všetko. Takto budete prehľadávať najprv tie perspektívnejšie. Ak z toho nič nebude, skúsíte menej perspektívne a prinajhoršom prehľadáte všetko. Šanca, že riešenie nájdete skôr, sa zvyšuje s kvalitou hodnotiacej funkcie. Aby bolo jasné, o čom vlastne píšem, predvediem nie práve najšťastnejšiu heuristiku v plnej paráde. Povedzme, že máme hodnotiacu funkciu pre Hanojské veže definovanú takto:

$$f(S) = x(S) + y(S)$$

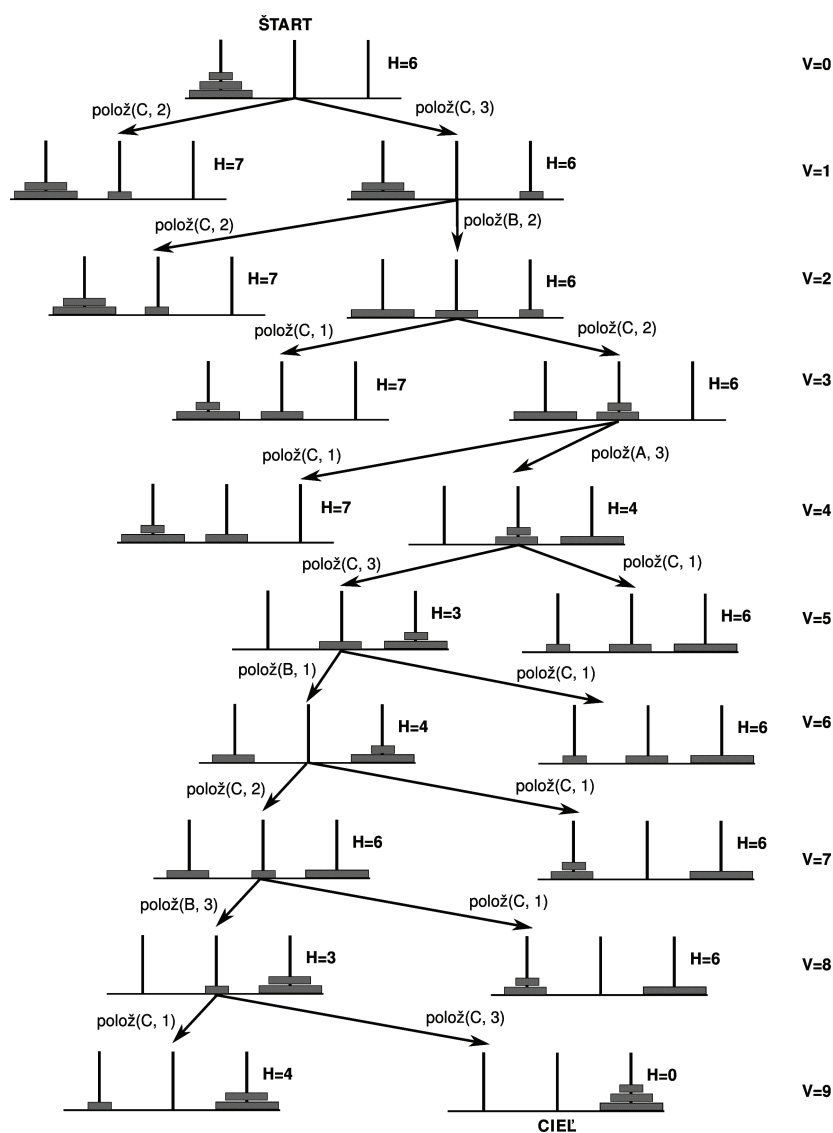
Funkcia  $f(S)$  je hodnotenie jedného konkrétneho stavu  $S$ ,  $x(S)$  je chyba horizontálnych vzdialeností pozícií kotúčov od cieľa a  $y(S)$  je chyba vertikálnych vzdialeností. Jednotlivé chyby dostaneme nasledovne:

$$x(S) = \sum d_x(\text{poz}(i) - \text{poz}(c_i)), \text{ pre } 0 < i < n$$

$$y(S) = \sum d_y(\text{poz}(i) - \text{poz}(c_i)), \text{ pre } 0 < i < n$$

kde  $n$  je počet kotúčov,  $d_x(\text{poz}(i) - \text{poz}(c_i))$ , resp.  $d_y(\text{poz}(i) - \text{poz}(c_i))$  je horizontálna, resp. vertikálna vzdialenosť aktuálnej pozície  $i$ -tého kotúča  $\text{poz}(i)$  od jeho požadovanej cieľovej pozície  $\text{poz}(c_i)$ . Pozície tyčí sú číslované zľava doprava od 1 po 3, výškové pozície kotúčov sú zdola nahor tiež číslované od 1 po 3. Povedané priamo, pre každý kotúč si spočítam, ako ďaleko je od svojej cieľovej pozície v horizontálnom a vertikálnom smere a tieto vzdialenosti sčítam. Celkové hodnotenie je súčet týchto vzdialeností pre všetky kotúče. Čím sú kotúče bližšie k svojej cieľovej pozícii, tým je hodnotenie lepšie (cieľový stav má hodnotenie 0).

Ako by vyzeral priebeh prehľadávania pre Hanojské veže pri stratégii prvý najlepší s takto postavenou heuristikou, si v celej kráse pozrite na obr. 7. Pre každý stav máte k dispozícii jeho hodnotenie a na pravej strane máte číslované úrovne rekurzívneho vnorenia algoritmu. Ak sa pozriete na priebeh prečesávania stavového priestoru, zaostríte na situáciu, ktorá sa udiala na úrovni  $V=6$ . Stav s hodnotením  $H=3$  vygeneroval dva stavy s horším hodnotením  $H=4$  a  $H=6$ . Čosi podobné sa udialo aj na úrovni  $V=7$ . Toto by sa hill climbingu nepáčilo. Je tu tiež zaujímavá situácia na úrovni  $V=5$  a rovnako na úrovni  $V=6$ . V oboch prípadoch bol zle hodnotený stav  $H=6$ , z ktorého sa dá cieľ dosiahnuť dvoma ťahmi. Vďaka hodnoteniu tento stav stratégia odložila bokom. Naša heuristika, ako je celkom jasne vidieť, vôbec nie je ideálna. Dáva však lepšie výsledky, ako slepé prehľadávanie.



Obr. 7. Príklad prehľadávania stratégiou prvý najlepší pre Hanojské veže.

Stratégiu prvý najlepší využíva aj algoritmus **A\*** (A-star) [5], ktorý nás nadchne čarom **optimálnych riešení**. Optimálne riešenie nám servíruje ako najkratší plán. A\* je len špecifickým prístupom, ako navrhovať heuristické funkcie pre stratégiu prvý najlepší. Táto metóda používa hodnotiacu funkciu zloženú z dvoch častí:

$$f(S) = g(S) + h(S)$$

Funkcia  $g(S)$  počíta cenu cesty zo štartovacieho do aktuálne prehl'adávaného stavu  $S$  a  $h(S)$  je cena cesty z aktuálneho do najbližšieho z cieľových stavov. Vzhľadom na to, že získať skutočné hodnoty zložiek funkcie  $f(S)$  je pomerne hrdinský výkon, používajú sa odhady  $g^*(S)$  a  $h^*(S)$ .

Hodnota zložky  $g^*(S)$  odhaduje cenu cesty od počiatocného stavu do aktuálne prehl'adávaného stavu  $S$ . Zvyčajne sa ako odhadovaná cena používa dĺžka skutočnej cesty. V prípade, že prehl'adávaný graf je strom je odhad  $g^*(S)$  ideálny, pretože v strome existuje len jediná cesta od počiatocného uzla do  $S$ . V prípade prehl'adávania všeobecného grafu môže byť odhad  $g^*(S)$  nadhodnotený. A aj keď stále podsúvam, že prehl'adávacie algoritmy pri prehl'adávaní generujú strom, nemusí byť tento odhad ideálny. Ak pri prehl'adávaní stromu vytvoríme konkrétny stav nejakou stratégiou, nemusí to nevyhnutne znamenať, že iná stratégia by sa k tomuto stavu nedostala inou cestou (teda skôr alebo neskôr). Toľko na margo odhadu zdanlivo známej cesty.

Hodnota zložky  $h^*(S)$  obsahuje samotnú heuristickú informáciu definovanú pre našu úlohu. Aby boli dodržané výhody algoritmu  $A^*$ , vyžaduje sa, aby hodnota  $h^*(S)$  bola vždy kladná a musí tiež platiť, že  $h^*(S) \leq h(S)$ , teda heuristický odhad ceny cesty zo stavu  $S$  do cieľového stavu nesmie byť nadhodnotený. Táto podmienka sa nazýva podmienkou prípustnosti. Čím je odhad  $h^*(S)$  bližšie k skutočnej hodnote  $h(S)$ , tým sú výsledky algoritmu lepšie. Ak by  $h^*(S) = h(S)$ , optimálne riešenie by sa našlo bez toho, aby bol prehl'adávaný jediný stav navyše. A opačne, ak by bola použitá funkcia  $h^*(S) = C$ , kde  $C$  je ľubovoľná konštanta, algoritmus  $A^*$  by degeneroval na bežné slepé prehl'adávanie (ak by boli všetky stavy hodnotené rovnako, nebola by možnosť vybrať na prednostné prehl'adanie tie perspektívnejšie).

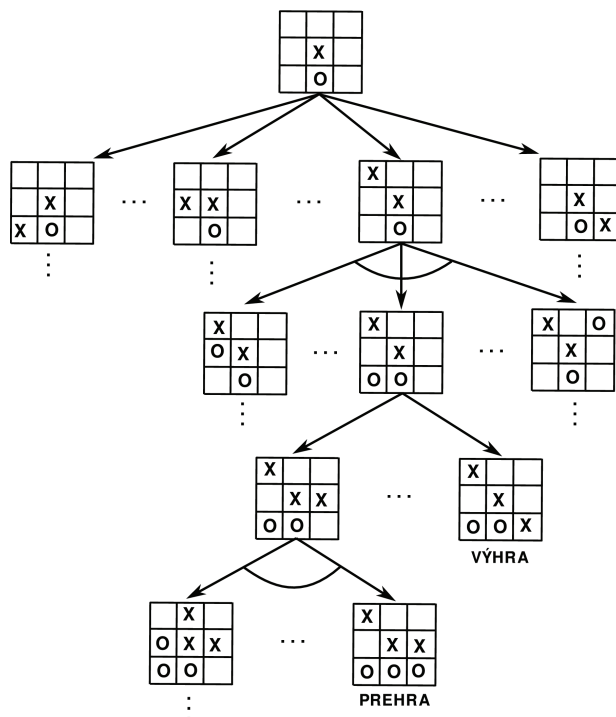
Stavovými priestormi sa prehrávajú kvantá prehl'adávacích stratégií už dobrých päťdesiat rokov. Zďaleka nie je možné venovať sa všetkým prístupom. Tak hádam aspoň tie základné, ku ktorým sme si tu privoňali, poslúžia ako inšpirácia. Ostala nám ešte jedna skupina úloh v stavovom priestore, ktorá si vyžaduje špeciálnu pozornosť a miernu zmenu prístupu. Pozrieme sa na to, ako vedia mašinky hrať hry.

## 2.2 Hranie hier

Automatické hranie hier celkom dobre zapadá do kontextu stavového priestoru. Intuitívne, ak niekoho práve porážam v šachu, v každom ťahu je hra v konkrétnom stave, každý ťah je operácia, ktorá mení stav hry. Potom dostanem dvoma ťahmi mat, je to definitívne ukončenie hry a ja sa môžem len rozpačito usmievať.

Pozrieme sa na hry, ktoré hrajú dvaja hráči. O čo by mi malo vlastne ísť pri hre? Ak sa povznesieme nad to, že budeme egoisticky myslieť len na seba, najlepší ťah je zrejme taký, ktorý nám prinesie najväčšie výhody a nášmu súperovi spôsobí vyplašený výraz. Potrebujeme teda maximalizovať našu výhodu a minimalizovať súperovu výhodu. Je v tom skrytý malý problém. Keď sa (mnohí nezmyselne) snažíme myslieť niekoľko krokov hry dopredu, máme zvyčajne viacero alternatív, ako ťahať. Pri každej možnosti ale musíme zobrať do úvahy čo najviac možných súperových ťahov. A to mierne mení situáciu, pretože priebeh hry už nebude vyzerať ako hocijaký bežný strom. Bude to **strom hry**, ktorý je kombináciou všetkých možných ťahov oboch hráčov. Ideálne by bolo mať úplný strom hry, ktorý je rozkošatený natoľko, že

obsahuje aj konečné stavy hry (výhra, prehra, remíza). Cesta od začiatočného stavu hry po niektorý z koncových uzlov by potom predstavovala jeden konkrétny priebeh partie.



Obr. 8. Časť stromu hry pre piškvorky.

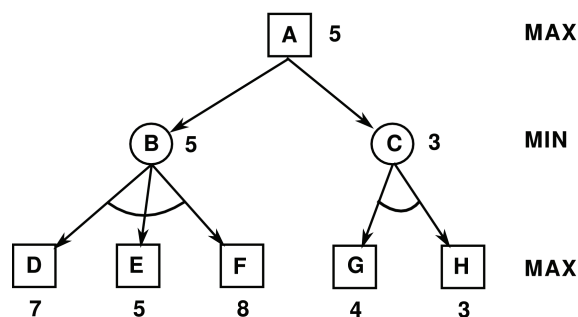
**Stromy hier sa vytvárajú z pohľadu jedného hráča** a sú reprezentované, ako tzv. **AND/OR grafy**. Striedajú sa v nich dva typy uzlov, a to:

- **OR uzol**, ktorý reprezentuje alternatívy hráča, ktorého pohľadu je strom vytvorený. Rozhodneme sa, kto by to bol čakal, pre jednu z alternatív.
- **AND uzol**, ktorý berie do úvahy všetky ťahy súpera. Kým náš hráč si vyberá jednu z alternatív ako ťahať, pri súperovej reakcii musíme brať do úvahy všetky situácie, ktoré je schopný napáchať.

V strome hry sa nám teda budú striedať OR a AND uzly. Príklad časti stromu hry je na piškvorky (tic-tac-toe) je na obr. 8. AND uzly spoznáte podľa slušivého oblúka. Úlohou je nájsť taký podstrom, ktorý nám zabezpečí výhru, pretože my radi víťazíme. Situáciu nám mierne komplikuje skutočnosť, že každá bežná hra má toľko kombinácií, že už aj hmlistá predstava toho množstva môže spôsobiť priemernému počítaču nevoľnosť. A preto sa prehľadávanie stromov hier málokedy zaobíde bez dobre odôvodnenej heuristiky.

Pri prečesávaní stromov hier by bolo ideálne mať k dispozícii úplný strom hry alebo aspoň zopár koncových uzlov, ktoré by sme si vedeli jasne a bez zbytočnej diskusie ohodnotiť. Aké by to bolo pekné? Výhra by bola 1, prehra -1, remíza 0

a forsírovali by sme známe konečné riešenie. To je ale málokedy možné (pre realistov: ak sa hra nechýli ku koncu, tak je to prakticky nemožné). Preto je dobré uvažovať len niekoľko krokov dopredu, v závislosti od zložitosti hry, tu nastupuje úloha expertov, ktorí s navrhnu hodnotiacu funkciu. Táto funkcia funguje rovnako, ako pri stavovom priestore a určuje nádejnosť situácie z pohľadu konečnej výhry. Keďže podceňovanie súperov sa neodporúča, heuristiky pri hrách zvyčajne nehodnotia stav hry na základe jej vzdialenosti od víťazstva. Zvyčajne sa hodnotí viac faktorov, ako počet figúrok, počet ohrozených figúrok, pri šachu napr. možné ohrozenie kráľa, schopnosť pohybu, apod. Čím viac hodnotených faktorov, tým lepšie. Len sa treba mať na pozore, aby nadšenie z toho, čo všetko beriete do úvahy, nezaniklo sa tom, že výpočet hodnotenia situácie bude trvať dlhšie, ako generovanie celého stromu hry.



Obr. 9. Príklad hodnotenia stromu hry stratégiou minimax.

Ako reprezentanta prehľadávania stromu hry ukážeme zrejme najznámejšiu stratégiu **minimax**. Táto stratégia sa snaží maximalizovať možnosť výhry hráča, z ktorého pohľadu sa hrá a minimalizovať možnosť výhry súpera. Princíp je veľmi jednoduchý. Vygeneruje sa zvolený počet ťahov hry a listové uzly sa ohodnotia pomocou heuristiky. Ohodnotenie sa potom šíri smerom hore, pričom sa snažíme ohodnotiť všetky uvažované ťahy takto:

- hodnota OR uzla nášho hráča bude maximum hodnôt jeho potomkov, lebo pre nášho hráča chceme len to najlepšie – stav s najlepším ohodnotením má väčšiu pravdepodobnosť, že vedie výhre
- hodnota AND uzlov súpera bude minimum hodnôt jeho potomkov, pretože mu v tejto hre nič valného neželáme – stav s najhorším ohodnotením má väčšiu pravdepodobnosť, že vedie prehre

Hypotetický príklad spätného hodnotenia stromu pomocou minimax je na obr. 9. Perspektívnosť listových uzlov je spočítaná heuristicky, ostatné uzly sú smerom hore ohodnotené minimaxom.

Rekurzívna verzia algoritmu, ktorý vám ohodnotí vygenerovaný strom hry, je na obr. 10. Sú to dve funkcie, ktoré sa vzájomne volajú. Jedna počíta hodnoty pre maximalizujúce, druhá pre minimalizujúce uzly. Keďže začínate s maximalizujúcim uzlom, volanie bude vyzeráť takto: `ohodnot_max(stav)`. Keď máte strom pre aktuálny stav ohodnotený, stačí zvoliť ťah s maximálnym hodnotením.

```

funkcia ohodnot_max(stav){
    ak(je_listový_uzol(stav)){
        vráť heuristický odhad stavu
    }
    hodnotenie = -nekonečno
    pre(i = 0; i < počet_potomkov; i++){
        hodnotenie = maximum(hodnotenie,
                               ohodnot_min(potomok[i]))
    }
    vráť hodnotenie
}

funkcia ohodnot_min(stav){
    ak(je_listový_uzol(stav)){
        vráť heuristický odhad stavu
    }
    hodnotenie = +nekonečno
    pre(i = 0; i < počet_potomkov; i++){
        hodnotenie = minimum(hodnotenie,
                               ohodnot_max(potomok[i]))
    }
    vráť hodnotenie
}

```

Obr. 10. Rekurzívna verzia stratégie minimax.

Minimax je úplne jednoduchá stratégia, ktorú je ešte možné zefektívniť napríklad zmenšením prehľadávaného priestoru. Známa je napr. stratégia alfa-beta orezania, ktorá sa zbaví tých častí stromu, ktoré už nie je potrebné prehľadávať. Základnou myšlienkou je nezaoberať sa zbytočne tými časťami stromu, ktoré už v aktuálnej situácii určite nepovedú k zlepšeniu možností. Tým sa celkom neprehliadnuteľne zredukuje čas výpočtu. V tomto texte sa ale uspokojíme s minimaxom, ako s typickou metódou automatického hrania hier.

### 3 Produkčné pravidlá a expertné systémy

Ak chceš vidieť gekona, potom chod' do ZOO. Ak to, čo vidíš, má krídla, potom je to vták. Ak je to vták a pochoduje to, tak je to pštros. Ak je pštros blízko, potom odbehni, lebo ťa d'obne. Ak budeš d'obnutý, potom budeš musieť ísť na pohotovosť. Ak budeš na pohotovosti, potom neuvidíš gekona. Ak neuvidíš gekona, potom budeš psychicky nalomený.

Zrejme, nepoviem úplný nezmysel, ak vyhlásim, že takouto formou zvykneme celkom často uvažovať o tom, ako vyhodnotiť nejakú situáciu alebo vykonať čo najvhodnejšiu činnosť. Takto formulované tvrdenia, sa volajú **produkčné pravidlá** a predstavujú špecifický typ znalostí o konkrétnej problematike. Prečo sa volajú produkčné? Pretože ak sú splnené nejaké predpoklady, málokedy to nemáva dôsledky. Ak takéto pravidlo platí, tak jeho predpoklady produkujú dôsledok. Formálne má produkčné pravidlo tvar:

**AK** podmienky **POTOM** dôsledok

Podmienkovú časť tvoria fakty čo najkomplikovanejšie poprepletané do neprehľadných logických výrazov. Fakty, ktoré spomínam, sú len jednoduché informácie, o ktorých vieme vyhlásiť, či v aktuálnej situácii sú alebo nie sú pravdivé. Dôsledková časť predstavuje nové fakty, ktoré pravidlo produkuje, ak je podmienková časť splnená. Dôsledková časť môže tiež povedať, akú akciu je potrebné za daných okolností vykonať. Ukážeme si to na konkrétnych príkladoch. Systémy, ktoré s produkčnými pravidlami pracujú, sa z nejakých zvláštnych dôvodov volajú **produkčné systémy**. Pozrieme sa, ako pracujú a následne náš zrak obrátíme k expertným systémom, ktoré na reprezentáciu znalostí využívajú produkčné pravidlá.

### 3.1 Produkčné systémy

V symbolike sú produkčné pravidlá v kategórii tých najčastejšie používaných. Sú totiž priamo predurčené na usudzovanie vďaka ich schopnosti odvodiť nové fakty, ktoré doteraz neboli známe. Produkčné systémy sú v princípe veľmi jednoduché. Bežný produkčný systém má tri časti:

- **Databáza faktov:** ak by sme chceli zaplachtiť do abstraktných výšok, povedali by sme, že plní funkciu krátkodobej pamäte. V skutočnosti je to len množina faktov, ktoré sú v aktuálnej situácii známe. Ale ináč, databáza faktov obsahuje informácie popisujúce aktuálny stav vo svete, v ktorom systém pracuje.
- **Databáza znalostí:** z predošlých abstraktných výšok ju môžeme nazvať dlhodobou pamäťou systému. Obsahuje zoznam produkčných pravidiel, teda všetky takto formulované znalosti o konkrétnej problematike.
- **Inferenčný (usudzovací) mechanizmus:** riadi použitie jednotlivých pravidiel. Ak sú splnené všetky podmienkové časti konkrétneho pravidla, toto pravidlo produkuje (niekedy hovoríme, že pravidlo je odpálené, toto nie je žiaden pokus o vtip). Odpálenie pravidla buď vygeneruje nové fakty alebo zrealizuje nejakú činnosť. Táto činnosť môže byť skutočná fyzická procedúra, ktorá sa vykoná.

Algoritmus produkčného systému pracuje v troch krokoch:

**Krok 1. Výber aplikovateľného pravidla:** inferenčný stroj prejde cez všetky pravidlá v databáze znalostí a porovná podmienkové časti každého pravidla s aktuálnym obsahom v databáze faktov. Pravidlá, ktorých predpokladové časti sú splnené, si označí ako aplikovateľné. A túto už iste registrujete mierny konflikt. Použiteľných pravidiel totiž môže byť viac.

**Krok 2. Riešenie konfliktov:** Ak v prvom kroku nájdených viac aplikovateľných pravidiel, vzniká problém, pretože v jednom kroku môžeme odpáliť len jedno pravidlo. Máme tu konflikt, ktorý treba riešiť. Metód na riešenie konfliktov je požehnané. Na riešenie konfliktov si vlastne môžete nasadiť samostatný pravidlový systém. Vo všeobecnosti sa napríklad odporúča použiť pravidlá, ktorých predpokladová časť obsahuje novšie fakty. Iná heuristika by rada skôr odpálila pravidlá so špecifickejšou podmienkovou časťou, pretože to, že sú aplikovateľné, zrejme indikuje menej



štandardnú situáciu, a podobne. Ak si neviete rady, vyberte jedno pravidlo náhodne a máte po problémoch.

**Krok 3. Aplikácia vybraného pravidla:** pravidlo zvolené v predchádzajúcom kroku sa odpáli, čo znamená, že sa vykoná jeho dôsledková časť. Dôsledky pravidla sú často vo forme nových odvodených informácií, ktoré zmenia obsah databázy faktov. Dôsledky pravidla môžu do databázy faktov pridávať nové informácie, ale tiež informácie odoberať. Takto sa aktualizuje popis situácie.

Produkčný systém môže byť použitý na to, aby sa pokúsil odvodiť zo známych faktov nejaké cieľové tvrdenie, vykonal čo najvhodnejšiu akciu alebo sa len jednoducho pokúsil aplikovať všetky možné použiteľné pravidlá a dosiahol tak aktualizáciu databázy faktov. Povedzme, že databáza faktov obsahuje množinu štartovacích informácií. Algoritmus produkčného systému sa pokúša sekvenčne aplikovať pravidlá dovtedy, kým neodvodí hľadané tvrdenie. Cieľová informácia je dosiahnutá vtedy, keď sa nachádza v databáze faktov. Táto cieľová informácia môže byť chápaná ako hypotéza, ktorú je potrebné dokázať (odvodiť) zo známych informácií. Algoritmus, samozrejme, končí aj vtedy, keď nie je možné aplikovať žiadne pravidlo. Tento typ inferencie sa volá **dopredné reťazenie pravidiel**. Existuje aj opačný prístup, **spätné reťazenie**, ktorý začína od cieľového tvrdenia. Tento príde na pretras, keď budeme hovoriť o expertných systémoch.

Predchádzajúce úvahy budú ilustrované jednoduchým príkladom. Ukážeme si triviálny príklad dopredného reťazenia. Nech štartovacia databáza faktov (DF) vyzerá takto:

$$DF = \{\text{má-krídla, chodí}\}$$

Áno, je to obyčajná množina konkrétnych symbolov. Môžu to byť reťazce, čísla alebo iné akokoľvek absurdné výrazy. Databáza faktov obsahuje to, čo v aktuálnej situácii vieme. Jediná dôležitá vec je, aby váš produkčný systém vedel tieto dáta použiť v predpokladových častiach pravidiel.

Databáza znalostí bude vyzeráť takto:

```
AK má-krídla ALEBO má-perie POTOM vták
AK vták A chodí POTOM pštros
AK vták A lieta POTOM vrana
```

Ak na naše dáta nasadíme inferenčný stroj s dopredným reťazením, udeje sa nasledovné:

1.  $DF = \{\text{má-krídla, chodí}\}$   
Aplikuj pravidlo: AK má-krídla ALEBO má-perie POTOM vták
2.  $DF = \{\text{má-krídla, chodí, vták}\}$   
Aplikuj pravidlo: AK vták A chodí POTOM pštros
3.  $DF = \{\text{má-krídla, chodí, vták, pštros}\}$   
Nie je možné aplikovať žiadne pravidlo.

Ak je použitá v predpokladovej časti pravidla logická spojka *ALEBO*, výraz bude platný, ak je pravdivá aspoň jedna z jeho častí. Výraz s logickou spojkou *A* bude platiť len vtedy, ak sú pravdivé všetky jeho časti. Náš produkčný systém teda v prvom kroku odvodil z faktu *má-krídla*, že ide o riadneho vtáka a do databázy faktov pribudla táto informácia. Podobne sa použilo aj druhé pravidlo a v konečnom dôsledku nám systém odvodil, že zo štartovacích faktov vyplýva, že ide o vtáka, a to nie o hocikakého.

Jednou z veľkých výhod reprezentácie znalostí produkčnými pravidlami je ich čitateľnosť. Všetky majú jednoduchý uniformný vzhľad, celkom jednoducho sa dá pochopiť, čo náš systém vie (samozrejme, čím je databáza znalostí väčšia, tým viac klesajú ilúzie o prehľadnosti). Pravidlá sa dajú jednoducho pridávať, meniť, likvidovať. Tu si ale treba dávať pozor, aby ste si znalosti nerozhádzali, napr. odstránením pravidiel, na ktorých výstupoch sú závislé iné pravidlá, a podobne. Pri veľkých databázach znalostí sa oplatí rozdeliť vedomosti do skupín podľa toho, čo znalosti popisujú. Takto si môžete vytvoriť moduly špecifických znalostí, ktoré medzi sebou môžete kombinovať podľa ľubovôle.

Ukázali sme základné princípy produkčných systémov a pozreli sa na jednoduché dopredné reťazenie pravidiel. Spätné reťazenie sa nám výborne hodí do kontextu expertných systémov, pri ktorých sa dotkneme interaktívnej diagnostiky realizovanej pomocou počítača.

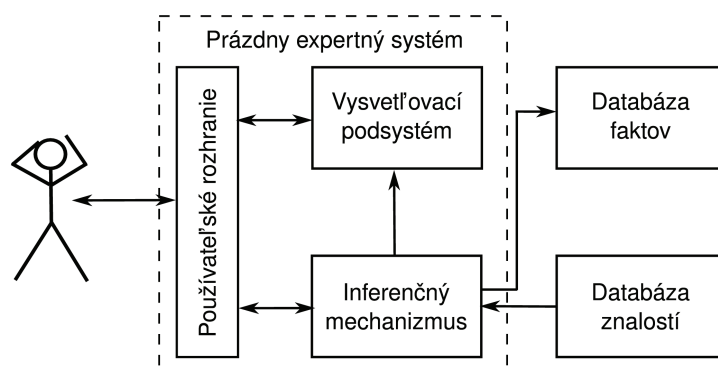
### 3.2 Expertné systémy

To, ako sa volajú, presne vystihuje to, na čo sú určené. Majú nahradiť ľudského experta z konkrétnej oblasti. Používajú sa vo všeobecnosti ako systémy pre podporu rozhodovania. Keďže je nezmysel už len pomyslieť na to, aby ste si pamätali úplne všetky detaily vo všetkých možných kontextoch, je celkom normálne, že dobre navrhnutý expertný systém môže pomôcť aj samotnému expertovi. A to teraz nehovorím o prípade, kedy ste adresátmi tvrdenia: „To je teda expert“.

V prvom rade je potrebné získať informácie od samotného špecialistu. Tieto informácie sa spracujú a navrhne sa databáza znalostí. A ťahanie rozumov z experta nie je teda nič triviálneho, ak sa o to pokúšame len neinvazívnymi metódami. Ak niekto pracuje ako špecialista v nejakej oblasti, veľa problémov dokáže riešiť intuitívne, na základe skúseností, často aj bez toho, aby bol v stave povedať vám presne, čo si riešenie problému vyžaduje. Ako to teda často vyzerá v praxi? Chytíte si najbližšieho experta, posadíte si ho na kolená a spýtate sa: „Počuj, Béla, a čo treba robiť, keď bliká tá červená kontrolka na paneli sekundárneho chladiaceho okruhu jadrového reaktora?“. A Béla sa ošáva, lebo ho v momente napadne toľko kombinácií so všetkým možným, že nevie kde začať. Tak začne menovať a vy ste za dve minúty zahltení nepreniknuteľnou hustou informačnou hmlou. Ak si vyberiete niektorú možnosť a spýtate sa prečo treba urobiť to a ono, tak sa vám môže ľahko prihodiť, že Béla povie, že on to tak cíti a to ešte máte šťastie, že nepovie: treba to urobiť preto, lebo to treba spraviť. Veru, tak je to s expertmi.

Ale, aby som dal veci na pravú mieru, je to síce zložitá záležitosť, ale existuje na to kvantum sofistikovaných systematických metód. Spracovanie experta a vytvorenie znalostnej bázy je jedna celá veda, ktorá sa volá **znalostné inžinierstvo**. A nakoniec sa

to vždy nejako dá. A ako sa hovorí medzi expertami: “To nie je problém, potrebujem čas”. My tu radšej budeme dôverčivo predpokladať, že bázu znalostí vo forme pravidiel už máme.



Obr. 11. Architektúra expertného systému.

Expertné systémy vlastne rozširujú produkčné systémy o niekoľko veľmi vd'avných vlastností. Sú špecifické tým, že sú iniciatívne. Aktívne s vami vedú dialóg a urputne sa pokúšajú poradiť vám. Ak máme povedzme systém, ktorého úlohou je zistiť vašu zdravotnú diagnózu, bude sa vás vypytovať na všetky možné symptómy dovtedy, kým sa mu nepodarí z toho, čo sa od vás dozvedel, odvodiť jednu alebo viacero možných diagnóz. Aktivitu expertného systému môžete kedykoľvek zastaviť a položiť niekoľko typov zásadných otázok, napríklad **prečo** sa systém pýta práve takúto otázku alebo **ako** dospel práve k takejto diagnóze. Toto je primárna vlastnosť expertných systémov. Všetky majú vysvetľovací mechanizmus.

Schematický obrázok architektúry expertného systému je na obr. 11. Databáza faktov, znalostí a inferenčný mechanizmus tvoria jadro produkčného systému. Navyše je tu vysvetľovací podsystém a rozhranie na komunikáciu s používateľom. Verím, že ste si na obrázku všimli označenie **prázdny expertný systém**. Na znalostných bázach vo forme produkčných pravidiel je pekné to, že **znalosti môžete vymeniť**. Ak necháte základný mechanizmus systému a vymeníte len znalosti, váš expertný systém je znenazdajky špecialistom na úplne inú oblasť. A to základné jadro bez znalostí sa volá prázdny expertný systém. Expertný systém teda môžete využiť na riešenie akejkoľvek problémovej oblasti. Závisí to len od znalostí, ktoré má k dispozícii.

### Inferencia spätným reťazením a AND/OR grafy

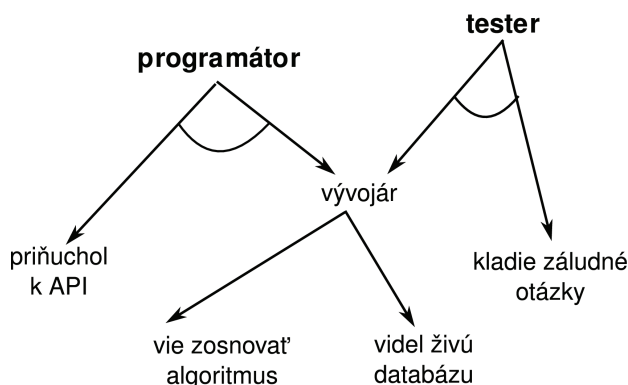
Ukážeme si prácu diagnostického expertného systému, ktorý bude využívať **spätné reťazenie pravidiel**. Ak to rozmeníme na drobné, databáza znalostí bude obsahovať pravidlá pre určenie viacerých diagnóz. Expertný systém sa postupne pokúsi dokázať platnosť každej diagnózy pomocou otázok (keďže hovorím o dokazovaní, neskôr budem namiesto diagnózy používať aj pojem hypotéza). Inferenčný mechanizmus nám tiež zabezpečí, aby sa systém nepokúšal nahlodať psychickú stabilitu používateľa tým,

že sa ho bude pýtať otázky, ktoré už nemá zmysel položiť. Napríklad, nemá zmysel pýtať sa na fakty vedúce k dôkazu hypotézy, ktorá už na základe iných odpovedí nemôže byť platná. Systém vám tiež nepoloží tú istú otázku viackrát.

Aby sa nám s databázou znalostí pohodlne pracovalo, prekloníme si ju do tzv. **AND/OR grafu**. Niečo podobného ste už videli pri stromoch hier. V tomto prípade nám bude AND/OR graf reprezentovať pravidlá v databáze znalostí. Hypotézy budeme dokazovať prehľadávaním tohto grafu. Ide o celkom jednoduchú záležitosť. Podmienka v predpokladovej časti pravidla bude zoznam faktov spojených logickou spojkou *A/ALEBO*. Budeme predpokladať, že v dôsledkovej časti bude len jeden fakt. Jedno pravidlo bude reprezentované tak, že dôsledková časť bude uzol, ktorého predkami budú fakty v podmienkovej časti. A podľa použitej spojky sa bude jednať buď o AND alebo OR uzol. Hodnota AND uzla je pravdivá, ak sú splnené všetky fakty v jeho podmienkovej časti a hodnota OR uzla platí, ak je splnený aspoň jeden fakt v jeho podmienkovej časti. Ak si takto rozložíme všetky pravidlá, dostaneme sympatický strom, kde koreňové uzly budú jednotlivé diagnózy a listové uzly budú tie najtriviálnejšie informácie.

Ako príklad si uvedieme užitočné znalosti, ktoré nám budú pomáhať na pohovore do IT firmy. Našinec môže byť odporúčaný ako programátor alebo tester. Pravidlá vyzerajú nasledovne (viď obr. 12.):

AK *vie-zosnovať-algoritmus* ALEBO *videl-živú-databázu* POTOM *vývojár*  
 AK *vývojár* A *priňuchol-k-API* POTOM *programátor*  
 AK *vývojár* A *kladie-záludné-otázky* POTOM *tester*



**Obr. 12.** Príklad AND/OR grafu. AND uzly spoznáte podľa oblúka.

Ak by sme poznali platnosť všetkých listových uzlov vedeli by sme okamžite zistiť aj platnosť všetkých hypotéz. Stačilo by ich pravdivostnú hodnotu propagovať v AND/OR grafe smerom k hypotézam. Túto informáciu ale nemáme, tak sa ju systém bude snažiť postupne zistiť. Zoberie sa prvá hypotéza a systém sa pokúsi zistiť platnosť jej predpokladovej časti. Ak fakt v predpokladovej časti nie je listový uzol,

znamená to, že tento fakt je dôsledkovou časťou iného pravidla. Systém musí opäť rovnakým spôsobom overiť platnosť tohto faktu na základe platnosti jeho predpokladovej časti. Takto sa postupuje, až kým systém nenarazí na listové uzly. A v tomto prípade je už možné zistiť platnosť uzla priamo od používateľa, a tak sa systém nesmelo spýta: “Je pravda, že *listový uzol*?”. Používateľ odpovie *áno/nie* a táto odpoveď sa vloží do databázy faktov. Pravdivosť hypotézy sa priebežne overuje z jednotlivých odpovedí používateľa. Ak sa hypotézu nepodari dokázať, systém pokračuje ďalšou hypotézou v poradí. Dialóg s expertným systémom pre náš príklad by mohol prebehnúť napríklad nasledovne (dokazuje sa prvá hypotéza *programátor*):

1. *Systém*: je pravda, že *priňuchol-k-API*?

*Odpoveď*: nie

*Komentár*: Systém začína dôkazom hypotézy *programátor*, do databázy faktov sa vložia hodnoty odpovedí {*nezloží-ho-rekurzia*: nie, *programátor*: nie}. V tomto momente je už jasné, že *programátor* sa nekoná. Systém sa pokúsi dokázať hypotézu *tester*.

2. *Systém*: je pravda, že *vie zosnovať algoritmus*?

*Odpoveď*: áno

*Komentár*: Do databázy faktov sa vloží {*vie-zosnovať-algoritmus*: áno, *vývojár*: áno}. Platnosť OR uzla *vývojár* bola kladnou odpoveďou splnená, ďalšiu otázku na splnenie platnosti uzla *vývojár* môže systém vynechať. Pokračuje sa v dokazovaní hypotézy *tester*.

3. *Systém*: je pravda, že *kladie-záludné-otázky*?

*Odpoveď*: áno

*Komentár*: Do databázy faktov sa vloží {*kladie-záludné-otázky*: áno, *tester*: áno}. Cieľová hypotéza *tester* bola dokázaná.

4. *Systém*: bola dokázaná hypotéza *tester*.

Aby sa expertný systém choval korektne, môžete rovnako dobre použiť dopredné aj spätné reťazenie pravidiel. Typickejšie je ale spätné reťazenie. Ďalšou pomocou pre diagnostiku môže byť aj to, že ešte pred spustením dialógu sa do databázy faktov vložia už známe informácie a expertný systém si pomocou ďalších dobre mierených otázok už len doplní chýbajúce informácie.

Popísaný spôsob, akým sa expertný systém postupne pokúša overiť platnosť jednotlivých hypotéz, sa dá realizovať jednoduchým rekurzívnym algoritmom. Nájdete ho na obr. 13. Funkcia `hľadaj_riešenie()` rekurzívne spustí prehľadávanie pre každú hypotézu volaním funkcie `dokáž_hypotézu(uzol)`. Algoritmus používa globálnu premennú `DB_údaiov`, ktorá obsahuje množinu zodpovedaných a odvodených faktov a ich pravdivostnú hodnotu.

Zatiaľ sme sa hrali len s čiernobielym prípadom, kedy sme mali možnosť na otázky odpovedať len: platí/neplatí. Život je ale farebný, koľkokrát až príliš. A expertné systémy nadšene pracujú s neurčitými informáciami. Neurčitosť môžete do systému zamontovať na viacerých miestach.

```

funkcia hľadaj_riešenie(){
    pre(i = 0; i < počet_hypotéz; i++){
        ak(dokáž_hypotézu(hypotéza[i]) == true){
            vypíš („Hypotéza " hypotéza[i] „ je dokázaná!“)
            koniec
        }
    }
}

funkcia dokáž_hypotézu(uzol){
    hodnota = false
    vypíš („Dokazujem: " + uzol)
    ak(je_v_databáze_údajov(uzol, DB_údajov)){
        vráť hodnota(uzol, DB_údajov)
    }
    pre(i = 0; i < počet_poduzlov; i++){
        ak(poduzol[i] je list){
            hodnota = polož_otázku(poduzol[i])
            vlož_do_DB_údajov(poduzol[i], hodnota)
        }
        inak{
            hodnota = dokáž_hypotézu(poduzol[i])
            vlož_do_DB_údajov(poduzol[i], hodnota)
        }

        ak(uzol je typu AND){
            ak(hodnota = false) vráť false
        }
        inak ak(uzol je typu OR){
            ak(hodnota = true) vráť true;
        }
    }
    ak(uzol je typu AND){
        vráť true
    }
    inak {
        vráť false
    }
}

```

**Obr. 13.** Algoritmus expertného systému realizovaný ako rekurzívne prehľadávanie AND/OR grafu spätným reťazením pravidiel.

Neurčitosť môžeme zakomponovať do predpokladových aj dôsledkových častí pravidiel. Pre predpoklady môžeme číslami definovať napríklad ich dôležitosť. Hodnota dôsledku zvyčajne určuje, s akou pravdepodobnosťou je tento dôsledok pravdivý. Táto možnosť upokojuje experta napríklad v situácii, keď stratí svoju povestnú istotu a musí dať viac na svoj odhad.

Pre používateľa je zvyčajne príjemné vedieť, že môže niečo aj nevedieť. Expertný systém mu teda môže ponúknuť celý arzenál odpovedí typu: áno, asi áno, neviem, asi nie, nie. Viachodnotová logika je v expertných systémoch celkom bežná. Takto ohodnotené odpovede sa potom šíria AND/OR grafom smerom k hypotézam a ovplyvňujú celkové ohodnotenie odvodených informácií a hypotéz. V takomto prípade je potom celkom logické pokúsiť sa dokázať viac hypotéz a ohodnotiť ich pravdepodobnosťami.

### Vysvetľovací mechanizmus

Má ho každý slušný expertný systém a slúži na vysvetlenie činnosti systému. Najčastejšie sa používajú dva typy vysvetlení, a to **prečo** systém kladie práve takúto otázku a **ako** systém dospel k odvodeniu práve tejto hypotézy.

Zvyčajnou odpoveďou na otázku *prečo* je odvodenie cesty z listového uzla, ktorý je predmetom otázky, ku koreňovému uzlu aktuálne dokazovanej hypotézy. V našom užitočnom príklade rozdeľujúcom ľudí na programátorov a testerov by dialóg mohol vyzerat' nasledovne (skontrolujte si to v AND/OR grafe na obr. 12):

*Systém:* Je pravda, že *vie-zosnovať-algoritmus*?

*Používateľ:* prečo?

*Systém:* Dokazujem hypotézu *programátor*. Predpokladám, že ak *vie-zosnovať-algoritmus*, potom *vývojár*, potom *programátor*.

*Systém:* Je pravda, že *vie-zosnovať-algoritmus*?

Takto je možné jednoducho vysvetliť, že ak systém dokazuje hypotézu "programátor", tak systém sa iniciatívne zaujíma o platnosť faktu *vie-zosnovať-algoritmus* preto, lebo predpokladá, že ak platí *vie-zosnovať-algoritmus* potom bude platiť *vývojár*, čo vedie k dôkazu hypotézy *programátor*.

Otázku *ako* môže používateľ položiť zvyčajne až keď je dokázaná hypotéza. Dožaduje sa ňou zdôvodnenia, ako systém dospel práve k takémuto záveru. Odpoveďou je zvyčajne zložitejšia štruktúra, ktorá popisuje celý inferenčný proces. Ak to budete implementovať, úplne stačí, ak po skončení inferencie ešte raz prebehnete časťou grafu, ktorá zodpovedá dokázanej hypotéze a hodnoty jednotlivých uzlov si vytiahnete z databázy faktov. Ako by vyzeralo vysvetlenie pre dokázanú hypotézu *tester* (pozrite si príklad dialógu vyššie):

*Systém:* bola dokázaná hypotéza *tester*.

*Používateľ:* ako?

*Systém:* Bola dokázaná hypotéza *tester*.

*Systém:* Ak má platiť *tester*, musí platiť *vývojár* A *kladie-záludné-otázky*

*Systém:* Dôkaz platnosti *vývojár*: áno!

*Systém:* Ak má platiť *vývojár*, musí platiť *vie-zosnovať-algoritmus*  
ALEBO *videl-živú-databázu*

*Systém:* Je pravda, že *vie-zosnovať-algoritmus* si zodpovedal: áno

*Systém:* Je pravda, že *kladie-záludné-otázky* si zodpovedal: áno

Vysvetľovací mechanizmus dáva používateľovi hrejivý pocit, že dialóg môže riadiť. Keď sa počas dialógu dozvie, čo sa systém snaží dokázať, môže počas behu prehodnotiť svoje odpovede a tým otočiť celý inferenčný proces úplne iným smerom. Najlepšie smerom k hypotézam, ktoré pokladá za pravdepodobnejšie. Ak dialóg trvá príliš dlho, je vysvetľovací mechanizmus jednak príjemným osviežením a jednak je to jedinečná možnosť, ako si pripomenúť, prípadne pochopiť, čo sa vlastne deje.

## 4 Predikátová logika a automatické dokazovanie

Sokrates je človek. Ak je niekto človek, potom je smrteľný. Z toho vyplýva, že Sokrates je smrteľný. Ak je Sokrates Parmenides a Aristoteles nie je Herakleitos, potom Diogenes je gyros. Logika, dedukcia. Z času na čas sa dá spozorovať aj u ľudí. Deduktívne usudzovanie je principiálne postavené na tom, že z konkrétnych predpokladov vyplýva konkrétny záver. Toto je na systémoch s formálnou logikou to pekné. Ak platia predpoklady, tak aj záver zaručene platí. Podobný mechanizmus sme už videli v produkčných systémoch. Teraz sa na to pozrieme o niečo formálnejšie z pohľadu logiky. Znalosti nášho systému budú reprezentované formulami predikátovej logiky, našou úlohou bude dokázať, že nejaké tvrdenie logicky vyplýva z množiny východiskových faktov. Ukážeme si, ako vytvoriť systém schopný automaticky robiť takéto dôkazy.

### 4.1 Predikátová logika prvého rádu

V symbolike sa na reprezentáciu znalostí asi najčastejšie používa predikátová logika prvého rádu (keď budem hovoriť o predikátovej logike, myslím tým predikátovú logiku prvého rádu). To je veľmi flexibilný jazyk, pomocou ktorého môžete zostrojiť zložité logické konštrukcie a s ich pomocou sformalizovať náš svet. Jazyk predikátovej logiky vám dáva široký arzenál možností, ako popísať štruktúrované objekty, udalosti, ich vlastnosti a vzájomné vzťahy. Pri (koľkokrát márných) pokusoch o vysvetlenie základných princípov automatického logického usudzovania je trochu nepríjemné to, že za ním číha pomerne zložitý matematický formalizmus. Preto sa nám zrejme nepodari obísť zopár formálnych definícií.

Predikátová logika je jazyk, a pre každý jazyk je celkom typické, že má abecedu, pravidlá, ako z abecedy vytvárať slová a zo slov výroky. Tiež je občas na mieste zaoberať sa tým, čo tieto slová a výroky znamenajú a ako ich interpretovať. Postupne cez to všetko prebehne, aby sme sa mohli priamo pustiť do logických dôkazov.

Základným stavebným kameňom jazyka predikátovej logiky sú tzv. **predikátové konštanty**. Predikátové konštanty reprezentujú konkrétne objekty zo sveta, ktorý popisujeme, napr. *sokrates*, *osa*, *anafylaktický.šok*, apod. Vlastnosti a vzťahy predikátových konštánt vieme vyjadriť tzv. **predikátovými symbolmi** (predikátmi). Predikáty sú tvrdenia o predikátových konštántach, ktoré vždy nadobúdajú hodnotu *pravdivý/nepravdivý*. Dajme si ako príklad tvrdenie *Filozof(sokrates)*. Predikát *Filozof* má jeden argument a rozdeľuje nám množinu všetkých definovaných konštánt na dve polovice. Na tú, pre ktorú platí, že *Filozof(konštanta)* a na tú, pre ktorú to nie je pravda, kedy by napríklad tvrdenie *Filozof(osa)* zjavne neplatilo. Taký bežný predikát môže mať ľubovoľne veľa argumentov. Pomocou viacargumentových predikátov vieme pohodlne definovať zložitejšie vzťahy medzi konštántami. Napríklad, *Žiak(platón,sokrates)* alebo *Daruje(platón,sokrates,hašlerka)*. Viacargumentové predikáty budú platiť pre konštanty, ktoré v takto definovanom vzťahu sú.

Zložitejšie formulácie môžeme poskladať pomocou logických operátorov:  $\neg A$  (negácia),  $A \wedge B$  (konjunkcia),  $A \vee B$  (disjunkcia),  $A \Rightarrow B$  (implikácia) a  $A \Leftrightarrow B$  (ekvivalencia). Napríklad:



„Ak je Sokrates človek, potom je Sokrates smrteľný“  
 $\text{Človek}(\text{sokrates}) \Rightarrow \text{Smrteľný}(\text{sokrates})$

„Ak je Sokrates filozof a Platón je jeho žiakom, potom aj Platón je filozof“  
 $\text{Filozof}(\text{sokrates}) \wedge \text{Žiak}(\text{platón}, \text{sokrates}) \Rightarrow \text{Filozof}(\text{platón})$

Ak chceme formulovať tvrdenia platné pre celé skupiny objektov (nie len pre jednotlivé konštanty menovite), máme k dispozícii **predikátové premenné**. Tieto premenné sú argumentami predikátov a zovšeobecňujú konštanty. Napríklad všeobecný  $n$ -árny predikát má tvar  $P(X_1, X_2, \dots, X_n)$ , kde jednotlivé  $X_i$  sú predikátové premenné. Takýto predikát nadobudne svoju pravdivostnú hodnotu až vtedy, keď sú za všetky premenné dosadené konkrétne konštanty. Aby sa dali premenné a konštanty rozlíšiť voľným okom, konštanty budeme písať s malým a premenné s veľkým prvým písmenom. A teraz tuším rozmýšľate, že keď píšeme predikáty s veľkým prvým písmenom, asi to nebude len tak. Nemýlite sa, nebude.

Tvrdenia, ktorými popisujeme svet môžu a nemusia platiť univerzálne. U niektorých chceme, aby platili len pre niektoré konštanty, u iných zas chceme, aby platili pre všetky. Aby sme to vedeli v predikátovej logike vyjadriť, máme k dispozícii tzv. **kvantifikátory**. Máme možnosť použiť **všeobecný kvantifikátor**  $\forall X$  (pre všetky  $X$ ) alebo **existenčný kvantifikátor**  $\exists X$  (existuje  $X$ ). Dajme si ako príklad nasledujúce tvrdenia:

„Každému niečo chutí.“  
 $\forall X \exists Y [ \text{Chutí}(X, Y) ]$

Keď sa chystáte zostrojiť nejaké tvrdenie, je dobré uvedomiť si význam kvantifikátorov. Tvrdenie v predchádzajúcom príklade doslovne hovorí: „Pre každého  $X$  existuje taká vec  $Y$ , že platí  $X$ -u chutí  $Y$ “. Ak by sme náhodou kvantifikátory vymenili, dostali by sme toto:

„Existuje niekto, komu chutí všetko.“  
 $\exists X \forall Y [ \text{Chutí}(X, Y) ]$

Kvantifikátory sa v logických tvrdeniach vyskytujú zásadne vo vzťahu s premennými. Premenné s nasadeným kvantifikátorom sú známe, ako **viazané**. Tie bez kvantifikátora sú tzv. **voľné**. To, že kvantifikátory nemá význam používať vo vzťahu s konštantami, hádam dáva zmysel (mne tvrdenia typu „pre každého sokrata“ dávajú celkom slušný nezmysel, zvlášť vo svetle faktu, že konštantá *sokrates* je len jedna jediná a inej takej v našom jazyku už nieto).

Doteraz nám predikátové symboly dovolili hrať sa len jednoduchú hru na *platí/neplatí*. Týmto ale predikátový počet nekončí. Ak by som to rozmenil na otázky, tak doteraz sme sa pýtali len: je pravda, že *Filozof(sokrates)*? A je pravda, že *Naháňa(achiles,korytnačka)*? Ale čo ak chceme položiť otázku: “Kto je učiteľom platóna?” S čiernobielymi predikátmi si tu zjavne nevystačíme. Z tejto nepríjemnej situácie nás dostanú tzv. **funkčné symboly** (funkcie). Na rozdiel od bežne viditeľných predikátov funkcie na výstupe produkujú konštantu, napr. *učiteľ(platón)=sokrates*. Aby sme to v tom mali poriadok, názvy funkcií budeme písať s malým začiatočným písmenom. Funkčné symboly, keďže generujú konštanty, sa objavujú ako argumenty predikátov alebo iných funkcií. Samozrejme, funkcie môžu mať ľubovoľne veľa argumentov. Argumentami funkcií môžu byť konštanty, premenné, aj iné funkcie. Napadol ma len taký zvláštny príklad:  $\forall X [\text{žiak}(X, \text{učiteľ}(X))]$ .

Pred tým, ako sa pozrieme na automatické dokazovanie, je potrebné ešte definovať (a teraz to myslím celkom vážne) **syntaktické pravidlá**, pomocou ktorých môžete formulovať tvrdenia. Pojem *pravidlá* som myslel doslovne. Tvrdenia, ktoré týmto pravidlám nevyhovujú, do jazyka predikátovej logiky prvého rádu jednoducho nepatria. Namiesto tvrdenia budem niekedy používať pojem **formula**. Aby sme predišli zmätkom, pojem *formula* znamená formálny zápis tvrdenia. Všetky definície budú rekurzívne. Poďme na to.

**Atomická formula** je  $n$ -árny predikátový symbol  $P(T_1, T_2, \dots, T_n)$ , kde jednotlivé  $T_i$  sú tzv. **termy**. Termom budeme označovať:

- ľubovoľnú konštantu
- ľubovoľnú premennú
- ľubovoľnú funkciu  $f(T_1, T_2, \dots, T_m)$ , kde  $T_j$  sú termy.

Čo nevyhovuje uvedeným pravidlám, nie je term. A čo to vlastne term je? Veľmi jednoducho, term je konštant alebo všetko, čo nakoniec ako konštant skončí (či už dosadením do premennej alebo generovaním funkciou).

**Formuly** (komplexnejšie tvrdenia) je možné vytvárať podľa nasledujúcich pravidiel:

- Atomická formula je formula.
- Ak  $A$  a  $B$  sú formuly, potom sú formuly aj výrazy:  
 $\neg A, A \wedge B, A \vee B, A \Rightarrow B, A \Leftrightarrow B$
- Ak je  $A$  formula a  $X$  je jej premenná, potom sú formulami aj výrazy:  
 $\forall X[A], \exists X[A]$

Dôležité je, aby formula, nech je akokoľvek komplikovaná, mala pravdivostnú hodnotu *platí/neplatí*. Formula je teda buď predikát alebo ľubovoľne veľa predikátov v akomkoľvek logickom vzťahu. Ak by ste mali problém od oka zistiť pravdivosť nejakej formuly, jednoducho použite pravdivostné tabuľky. Príklad pravdivostných

tabuliek pre dve formuly  $A$  a  $B$  a pre všetky logické operátory okrem negácie je v tab. 1. Interpretácii negácie sa, verím, netreba špeciálne venovať.

**Tabuľka 1.** Pravdivostná tabuľka pre jednotlivé logické operátory, kde T(F) značí pravdivý-platí (nepravdivý-neplatí).

$A$	$B$	$A \wedge B$	$A \vee B$	$A \Rightarrow B$	$A \Leftrightarrow B$
T	T	T	T	T	T
T	F	F	T	F	F
F	T	F	T	T	F
F	F	F	F	T	T

Existujú dva veľmi špeciálne prípady formúl. To sú **tautológie** – formuly, ktoré platia vždy a za každých okolností. A opačne, **kontradikcie** – formuly, ktoré neplatia nikdy. Typickým príkladom tautológie je zemité tvrdenie: „Sokrates je filozof alebo nie je filozof“ ( $Filozof(sokrates) \vee \neg Filozof(sokrates)$ ). A opačne, kontradikcia je napr.: „Platón je filozof a nie je filozof“ ( $Filozof(platon) \wedge \neg Filozof(platon)$ ).

Predikátovú logiku prvého rádu sme si ohmatali, poďme si ukázať jeden zo spôsobov, ako s ňou naložiť.

## 4.2 Automatické dokazovanie

Začneme rovno od podlahy. Rozhodol som sa, že vás bez varovania postavím pred zákerný chyták. Situácia je takáto:

Florián vlastní gekona. Každý, kto vlastní gekona, je milovník zvierat. Žiaden milovník zvierat nezašliapne žiadne zviera. Kleofáša zašliapol Florián alebo Krišpín. Kleofáš je morča a morča je zviera. Zašliapol Kleofáša Krišpín?

Neviem, ako by sa k tomuto postavil Sherlock Holmes, ale my k riešeniu pristúpime s plnou automatizáciou. Akiste ste začali tušiť, s čím tu plánujeme mať dočinenia. V plnom znení sa tomu hovorí **formálne dokazovanie**. Funguje to takto: máme množinu tvrdení – teóriu, ktorú považujeme za platnú. To znamená, že predpokladáme, že všetky tvrdenia v tejto teórii sú pravdivé. A nás bude zaujímať, či nejaké nové tvrdenie z tejto teórie vyplýva. Odložme náš chyták teraz na chvíľu bokom a skúsme si to úplne zjednodušiť. Ako triviálny objasňovací chyták nám poslúži tento príklad (v tejto chvíli sa spolieham na to, že s nasledujúcimi faktami je každý vnútorne zmierený):

Frank Sinatra je človek. Všetci ľudia sú smrteľní. Je Frank Sinatra smrteľný?

Naša teória bude: “Frank Sinatra je človek a všetci ľudia sú smrteľní”. Toto sú naše východiskové tvrdenia, o ktorých pravdivosti nepochybujeme. Úlohou je dokázať, že v našej teórii platí nové, cieľové tvrdenie: “Frank Sinatra je smrteľný” (a to napriek tomu, že mnohí by sa o tom v iných kontextoch srdcervúco snažili pochybovať). Tú otázku som si preformuloval na tvrdenie, ktoré by malo v rámci našej teórie platiť. Pred tým, ako sa budeme baviť o tom, ako na to, si tento príklad preformulujeme do predikátovej logiky:

**Teória:**

$$\{ \text{Človek}(\text{frank.sinatra}), \forall X [ \text{Človek}(X) \Rightarrow \text{Smrteľný}(X) ] \}$$

**Cieľová formula:**

$$\text{Smrteľný}(\text{frank.sinatra})$$

Opäť to úplne zjednoduším. Najprv ale zopár hrejivých slov na margo automatického dokazovania. Formálny dôkaz sa v systémoch predikátovej logiky dosahuje rovnako, ako v produkčných systémoch. Spomeňte si na produkčné pravidlá, formulované v tvare: **AK podmienka POTOM dôsledok**. Ich vďačnou vlastnosťou bolo, že dokázali odvodiť nové znalosti, ktoré formálne nemali v databáze faktov. Ak platili podmienky, potom platil dôsledok. To druhé tvrdenie v našej teórii vám nepripomína produkčné pravidlo len tak náhodou. Ono to produkčným pravidlom je. Princíp si vysvetlíme na tzv. **priamom dôkaze**. Pri priamom dokazovaní skúsime ukázať, že cieľové tvrdenie priamo vyplýva z faktov v našej teórii. Náš priamy dôkaz urobíme veľmi jednoducho. Keďže pravidlo smrteľnosti platí pre všetkých ľudí, tak platí aj pre Franka Sinatru, zvlášť, keď o ňom vieme, že je človek. Formálne by stačilo vo faktoch teórie dosadiť za premennú  $X$  konštantu *frank.sinatra*, keďže ide o rovnaký predikát *Človek*. Po substitúcii premennej  $X$  aj na druhej strane pravidla dostaneme to, čo sme formálne nevedeli, a to nový fakt: *Smrteľný(frank.sinatra)*. Okolo substitúcie premenných je celá veda a budeme sa jej podrobnejšie venovať neskôr. Každopádne, máme to dokázané!

Odvodzovacie pravidlo, ktoré som použil, sa volá **modus ponens**. A formálne vyzerá pre dve formuly  $P$  a  $Q$  takto: ak platí  $P$  a platí, že  $P \Rightarrow Q$  ( $P$  vyplýva z  $Q$ ), potom platí  $Q$ . Proces dokazovania funguje presne tak, ako pracujú produkčné systémy. Štartovacia množina faktov sú tvrdenia v teórii, každá aplikácia použiteľného modus ponens vám vygeneruje nové tvrdenie, to sa pridá do databázy faktov a tento proces sa deje dovtedy, kým sa nevyprodukuje cieľové tvrdenie. Jednoduché, nie? V rukách automatického dokazovania je to veru silná zbraň. Ako som spomínal už skôr, na systémoch formálnej logiky je pekné, že ak máte pravdivú teóriu, akúkoľvek novú informáciu odvodíte, vždy bude pravdivá. A toto sa pri automatických dôkazoch zneužíva stále. Pri priamom dokazovaní sa pomocou modus ponens snažíte odvodiť z teórie cieľové tvrdenie, pretože predpokladáte, že je pravdivé. A keď ozaj je, jedného dňa sa vám ho odvodiť podarí. My sa ale vo všetkej konkrétnosti pozrieme na opačný prístup.

### Rezolučný princíp a dôkaz sporom

Iná možnosť je urobiť **dôkaz sporom**. Formálne by sme to urobili tak, že by sme naše cieľové tvrdenie znegovali, pridali k našej teórii a ukázali, že v našej teórii existuje spor. Spor znamená, že naša teória obsahuje kontradikciu, teda napr. hovorí, že niečo platí a zároveň to neplatí.

Tu je situácia o niečo komplikovanejšia a o to viac sa jej budeme venovať. Dôkaz sporom je totiž patrí v teórii automatického dokazovania medzi tie klasické. Volá sa to **rezolučná metóda** (alebo tiež **rezolučný princíp**) a prišiel s ňou istý John Alan Robinson v roku 1965 [6]. Rezolučná metóda používa špecifické odvodzovacie tzv. **rezolučné pravidlo**, ktoré sa sekvenčne používa na formuly v teórii dovtedy, kým nie je odvodený spor. Rezolučné pravidlo však pracuje len s tzv. **klauzulárnym tvarom formúl**. A preto pred tým, ako sa dostaneme k rezolúcii, sa musíme prehrýzť cez prevod formúl do klauzulárneho tvaru.

Formulu v klauzulárnom tvare budeme volať **klauzula**. A opäť formálnejšie, **klauzula je disjunkcia literálov**:

$$L_1 \vee L_2 \vee \dots \vee L_n$$

**Literál je atomická formula alebo jej negácia**. Každú formulu v predikátovom počte prvého rádu je možné previesť na klauzulárny tvar. Chce len to niekoľko krokov.

**Krok 1. Formulu zbavíme symbolov implikácie**. Všetky implikácie sa prevedú na disjunkciu pomocou ekvivalentnej formuly:  $(A \Rightarrow B) \Leftrightarrow (\neg A \vee B)$ .

Príklad: formula

$$\forall X [A(X) \Rightarrow \forall Y [B(Y) \Rightarrow C(X, Y)]]$$

sa prevedie na tvar

$$\forall X [\neg A(X) \vee \forall Y [\neg B(Y) \vee C(X, Y)]]$$

**Krok 2. Upravíme pôsobenie symbolu negácie** tak, aby každá negácia bola viazaná najviac na jednu atomickú formulu (tu sa využívajú hlavne de Morganove zákony, komutatívny, asociatívny a distributívny zákon, atď.), len ilustračne:

$$\neg(A \wedge B) \Leftrightarrow (\neg A) \vee (\neg B)$$

$$\neg(A \vee B) \Leftrightarrow (\neg A) \wedge (\neg B)$$

$$\neg\neg(A) \Leftrightarrow A$$

**Krok 3. Štandardizácia premenných.** Premenné musíme premenovať tak, aby každý kvantifikátor mal vlastnú, unikátnu a jedinečnú premennú. Napríklad formula:

$$\forall X \left[ \neg A(X) \vee \exists Y \left[ B(Y) \wedge C(X, Y) \right] \wedge \exists Y \left[ C(Y) \wedge D(X, Y) \right] \right]$$

obsahuje dvakrát premennú  $Y$  viazanú dvoma rôznymi existenčnými kvantifikátormi. Po štandardizácii to bude vyzerat' takto:

$$\forall X \left[ \neg A(X) \vee \exists Y \left[ B(Y) \wedge C(X, Y) \right] \wedge \exists Z \left[ C(Z) \wedge D(X, Z) \right] \right]$$

**Krok 4. Zbavíme sa existenčných kvantifikátorov.** Všetky premenné viazané existenčnými kvantifikátormi je potrebné nahradiť niečím, čo by nepotrebovalo existenčný kvantifikátor, ale zároveň by to zohľadňovalo pôvodný význam formuly. Tomuto procesu sa hovorí **skolemizácia** (podľa Thoralfa Skolema). V najjednoduchšom prípade  $\exists X \left[ A(X) \right]$  sa existenčného kvantifikátora zbavíme tak, že namiesto premennej použijeme novú, unikátnu, ešte nepoužitú konštantu. Nová formula, napr.  $A(c)$ , síce nie je úplne ekvivalentná s pôvodnou, ale ak platí pôvodná formula, bude platiť aj táto nová. Aby sme odstránili existenčný kvantifikátor, povedali sme si, že ak existujú nejaké  $X$ , pre ktoré platí  $A(X)$ , potom vytvoríme jednu konkrétnu konštantu  $c$  a formálne nám bude stačiť, že to platí pre túto jednu. Existuje také  $X$ ? Existuje, u nás to bude konštantu  $c$ . Novú unikátnu konštantu musíte vytvoriť pre každú premennú viazanú existenčným kvantifikátorom.

Mierne komplikovanejšia situácia nastáva, ak sa premenná viazaná existenčným kvantifikátorom ocitne v oblasti pôsobenia všeobecného kvantifikátora. Zoberme si tvrdenie "Každému niečo chutí":  $\forall X \exists Y \left[ \text{Chutí}(X, Y) \right]$ . Pre každé  $X$  existuje také  $Y$ , ktoré  $X$ -u chutí. Tu je ale celkom zjavné, že každému  $X$ -u bude chutiť iné  $Y$ . A preto hodnota  $Y$  bude závislá od hodnoty  $X$ . V tomto prípade sa použije tzv. **skolemova funkcia**, ktorá nám túto závislosť zoberie do úvahy. Pre náš príklad to bude napr.  $Y=f(X)$ . Výsledná formula bez existenčného kvantifikátora bude vyzerat' takto:  $\forall X \left[ \text{Chutí}(X, f(X)) \right]$ . Keďže je každý takýto vzťah špecifický, aj meno funkcie musí byť pre každý takýto prípad unikátne (jednoducho vždy použijete iné meno funkcie). Pri odstraňovaní existenčného kvantifikátora premennej musí skolemova funkcia zohľadňovať všetky premenné viazané všeobecnými kvantifikátormi, na ktorých daná premenná závislá. Napríklad, hodnota premennej  $Y$  vo formule

$$\forall X_1 \forall X_2 \exists Y \forall X_3 \left[ P(X_1, X_2, Y, X_3) \right]$$

je závislá od hodnôt všetkých premenných na ľavo od nej.  $Y$  ale nie je závislé od premennej  $X_3$ . Formula preto po odstránení existenčného kvantifikátora bude mať tvar:

$$\forall X_1 \forall X_2 \forall X_3 [P(X_1, X_2, g(X_1, X_2), X_3)]$$

Skolemova funkcia sa môže volať akokoľvek. Dôležité je len to, aby jej meno bolo unikátne.

**Krok 5. Zbavíme sa všeobecných kvantifikátorov.** Formula sa prevedie do tzv. **prenexného tvaru**, čo znamená, že všetky všeobecné kvantifikátory sa presunú na začiatok formuly. Keďže sme sa v predchádzajúcom kroku zadarmo zbavili existenčných kvantifikátorov, všetky premenné sú viazané už len všeobecnými kvantifikátormi. Môžete ich bezbolestne vynechať.

**Krok 6. Úprava formuly do tzv. konjunktívnej normálnej formy.** Formula potom bude mať tvar konjunktie disjunktíí (nepopieram, že to znie príšerne). Len pre ilustráciu, konjunkcia disjunktíí vyzerá takto:

$$(A_1 \vee A_2 \vee \dots \vee A_p) \wedge (B_1 \vee B_2 \vee \dots \vee B_q) \wedge \dots \wedge (C_1 \vee C_2 \vee \dots \vee C_r)$$

Mali by ste si vystačiť s obyčajným distributívnym pravidlom. Aplikujte ho toľkokrát, koľkokrát bude treba. To, čo potrebujete, je dostať z disjunktívnej formy  $X \vee (Y \wedge Z)$  formu konjunktívnu  $(X \vee Y) \wedge (X \vee Z)$ .

**Budeme pracovať s teóriou definovanou ako konjunkcia klauzúl** (teda práve s konjunkciou disjunktíí) a to je dôvod, prečo sme potrebovali všetky pôvodné klauzuly do tohto tvaru previesť. Každá klauzula reprezentuje jedno tvrdenie v teórii. Teória má v ľudskej reči tvar: *tvrdenie<sub>1</sub> a tvrdenie<sub>2</sub> a .. tvrdenie<sub>n</sub>*. Každé z tvrdení je formula v disjunktívnom tvare (klauzula). Pre takto definovanú teóriu budeme tiež používať množinový zápis:  $\{\text{tvrdenie}_1, \text{tvrdenie}_2, \dots, \text{tvrdenie}_n\}$ .

Ak to v krátkosti zrekapitulujem, máme teóriu a úlohou je dokázať, že z tejto teórie vyplýva nejaké cieľové tvrdenie. Pri priamom dôkaze sa pokúšame cieľové tvrdenie odvodiť aplikáciou pravidiel. Pri dôkaze sporom postupujeme tak, že cieľové tvrdenie znegujeme a ukážeme, že naša teória spolu s negovaným cieľovým tvrdením obsahuje **spor**. Ak dokážeme, že teória rozšírená o negáciu cieľového tvrdenia je sporná, potom prakticky dokážeme, že cieľové tvrdenie vyplýva z pôvodnej teórie.

**Rezolučná metóda** (tiež rezolúcia) **hľadá spor v konečnej množine formúl v klauzulárnom tvare**. Bolo dokázané, že táto dokazovacia metóda je tzv. **úplná**, čo znamená, že ak v teórii existuje spor, rezolúcia ho nájde. Preto je priam preurčená na implementáciu. Rezolučná metóda využíva špeciálne **rezolučné pravidlo**, ktoré šrotuje formuly v klauzulárnom tvare. Proces dokazovania bude opäť postavený na produkčnom systéme. Všeobecný tvar toho tajomného špeciálneho pravidla vyzerá takto (hore sú predpoklady, dole odvodený dôsledok):

$$\frac{A_1 \vee \dots \vee A_i \vee X \vee A_{i+1} \vee \dots \vee A_n \quad B_1 \vee \dots \vee B_j \vee \neg X \vee B_{j+1} \vee \dots \vee B_m}{A_1 \vee \dots \vee A_i \vee A_n \vee B_1 \vee \dots \vee B_j \vee B_m}$$

Je to menej desivé, ako to vyzerá. Polopate, zoberiete si dve formuly z teórie, ktoré môžu byť predpokladmi pravidla vtedy, ak jedna formula obsahuje literál  $X$  a druhá  $\neg X$ . Tieto vzájomne negované literály sa odstránia a rezolučné pravidlo ako výstupnú formulu odvodí disjunkciu pôvodných formúl bez literálov  $X$  a  $\neg X$ . Pre istotu si dajme dva triviálne príklady:

$$\frac{A \vee B \quad B}{A} \qquad \frac{A \vee B \quad C \vee \neg B}{A \vee C}$$

Každá aplikácia rezolučného pravidla odvodí novú formulu, ktorá sa pridá do databázy faktov. Toto pravidlo sa aplikuje dovtedy, kým nie je odvodený spor vo forme prázdnej klauzuly. Odvodenie sporu vyzerá takto:

$$\frac{A \quad \neg A}{\{ \}}$$

Algoritmus rezolúcie sa dá popísať niekoľkými krokmi. Princíp je jednoduchý, takže pseudokód tu nebudem uvádzať.

**Krok 1. Štandardizácia premenných** v teórii tak, aby každá formula používala unikátne mená premenných. Žiadne dve formuly v teórii nesmú používať rovnaké premenné.

**Krok 2. Výber formúl pre rezolučné pravidlo.** Z teórie sa vyberú sa dve také formuly, kde jedna obsahuje literál  $X$  a druhá  $\neg X$ .

**Krok 3. Unifikácia.** Ak by sme pre rezolúciu vybrali napríklad formuly  $\text{Človek}(\text{frank.sinatra})$  a  $\neg \text{Človek}(X) \vee \text{Smrteľný}(X)$ , museli by sme urobiť substitúciu premennej  $X$  a konštanty  $\text{frank.sinatra}$ . Proces substitúcie sa volá **unifikácia**. Unifikovať musíme tie dva vzájomne negované literály, vďaka ktorým sa dve formuly dostali do rezolučného pravidla. Keďže sú tieto dva literály vzájomne negované, ide o ten istý predikát, akurát premenné, konštanty a funkcie v argumentoch na rovnakých pozíciách sa môžu líšiť (pozrite si pre obe formuly predikát  $\text{Človek}$  v našom príklade). A účelom unifikácie je nájsť takú substitúciu, pre ktorú by tieto dva literály vyzerali rovnako. Práve preto sa táto substitúcia sa volá **unifikátor**. Ak ju nájdeme, je aplikovaná na všetky literály v oboch formulách. Pre naše dve formuly v príklade s Frankom Sinatrom je vhodný unifikátor  $\{X / \text{frank.sinatra}\}$ . Po aplikácii unifikátora na obe formuly sa prvá formula nezmení a druhá formula s premennou  $X$



dostane novú tvar  $\neg \text{Človek}(\text{frank.sinatra}) \vee \text{Smrteľný}(\text{frank.sinatra})$ . Aby sme si urobili lepší prehľad o tom, o čom sa vlastne bavíme, ilustračné príklady unifikátorov nájdete v tab. 2. Skúste vždy použiť unifikátor na obe formuly. Výsledok musí byť rovnaký, inak o unifikátore nemá veľmi zmysel hovoriť.

Tabuľka 2. Príklady unifikátorov.

	Literály	Unifikátor	Po substitúcii
1.	$\{P(X), P(a)\}$	$\{X/a\}$	$P(a)$
2.	$\{P(X,f(Y),c), P(X,f(b),c)\}$	$\{X/a, Y/b\}$	$P(a,f(b),c)$
3.	$\{P(a,Y,c), P(X,g(b),c)\}$	$\{X/a, Y/g(b)\}$	$P(a,g(b),c)$

Pozrime sa v tab. 2. na unifikátor v druhom riadku s výsledkom  $P(a, f(b), c)$ . Množina  $\{X/a, Y/b\}$  síce je unifikátorom literálov, ale substitúcia  $X/a$  bola úplne zbytočná. Prvým argumentom v oboch formulách je premenná  $X$ . Nemá teda veľmi zmysel vymýšľať pre ňu substitúciu navyše. Rovnako dobre by sme si vystačili len s unifikátorom  $\{Y/b\}$ . Po substitúcii by mala výsledná formula tvar  $P(X, f(b), c)$ . Unifikátor  $\{X/a, Y/b\}$  teda pre tento prípad nie je najjednoduchším (najvšeobecnejším). V automatickom dokazovaní sa používajú tzv. **najvšeobecnejšie unifikátory** (Most General Unifier – MGU). Po lopate povedané, sú to unifikátory, kde nie je žiadna substitúcia navyše.

Jednoduchý algoritmus, ktorý nájde najvšeobecnejší unifikátor, ak existuje, je na obr. 14. Algoritmus má na vstupe dva literály, ktoré chceme unifikovať. Sú to tie dva vzájomne negované literály, vďaka ktorým sme si pre rezolučné pravidlo vybrali tieto dve formuly. Algoritmus postupne prechádza všetkými argumentami oboch literálov a snaží sa nájsť substitúciu ich argumentov. Len pre istotu, v oboch literáloch môžeme postupne porovnávať dvojice argumentov na rovnakých pozíciách, keďže ide o rovnaký predikát. Akurát argumenty predikátu môžu mať v oboch formulách rôzne pomenované premenné, funkcie a konštanty, pre ktoré hľadáme substitúciu. Vstupom do algoritmu sú dva literály  $S, T$ . Algoritmus používa globálnu premennú  $U$ , čo bude najvšeobecnejší unifikátor reprezentovaný ako množina. Pred štartom algoritmu musí byť premenná  $U$  inicializovaná na prázdnu množinu. Do unifikátora sa budú dopĺňať jednotlivé substitúcie počas behu algoritmu. Ak algoritmus neskončí zlyhaním, premenná  $U$  bude obsahovať substitúcie najvšeobecnejšieho kvantifikátora. Volanie algoritmu: `unifikuj(S, T)`.

Ak unifikácia zlyhá, proces rezolúcie sa vráti na krok 2 a pre rezolučné pravidlo vyberie iné dve formuly (tie budeme hľadať dotedy, kým si nenájdeme také, ktoré sa dajú unifikovať). Ak sa unifikátor nájde, aplikuje sa na formuly vybrané v kroku 2.

**Krok 4. Aplikácia rezolučného pravidla.** Pravidlo sa aplikuje na unifikované vybrané formuly a odvodí novú formulu. Ak je nová formula prázdna klauzula, dokázali sme spor a algoritmus končí s úspechom. Ak sa aplikáciou pravidla spor nenašiel, odvodená formula sa vloží do teórie a pokračuje sa krokom 1.

```

U={ }
funkcia unifikuuj(S,T) {
    pre(i=0; i<počet_argumentov; i++){
        X = argument(S,i)
        Y = argument(T,i)
        unifikuuj_term(X,Y)
    }
}

funkcia unifikuuj_term(X,Y) {
    ak(je_premenná(X)) {
        X = aplikuj_substitúciu(U,X)
        Y = aplikuj_substitúciu(U,Y)
    }

    ak(je_premenná(X) a X=Y) nerob nič
    inak ak(je_konštanta(X) a
            je_konštanta(Y) a
            X!=Y) koniec s neúspechom
    inak ak(X=f(X1, ..., Xn) a Y=g(Y1, ..., Ym)) {
        ak(f=g) {
            pre(j=1; j<n; j++){
                unifikuuj_term(XjYj)
            }
        }
        inak {
            koniec s neúspechom
        }
    }
    inak ak(nie_je_premenná(X)) unifikuuj_term(Y,X)
    inak ak(X!=Y a
            Y = f(Y1, ..., Yn) a
            vyskytuje_sa(X, {Y1, ..., Yn})) koniec s neúspechom
    inak vlož({X/Y}, U)
}

```

**Obr. 14.** Jednoduchý unifikačný algoritmus.

Použitie rezolučnej metódy si ukážeme na dvoch príkladoch. Základné princípy si ukážeme na tom jednoduchšom, ktorý som majestátne nazval “dôkaz smrteľnosti Franka Sinatru”. A potom si ukážeme, ako by prebehol dôkaz chytáku, ktorý som podhodil v úvode. Tento dôkaz bude známy ako “Kauza zašliapnutého morčat’a”.

**Príklad:** Dôkaz smrteľnosti Franka Sinatru

**Znenie úlohy:** Frank Sinatra je človek. Všetci ľudia sú smrteľní. Je Frank Sinatra smrteľný?

**Teória:**

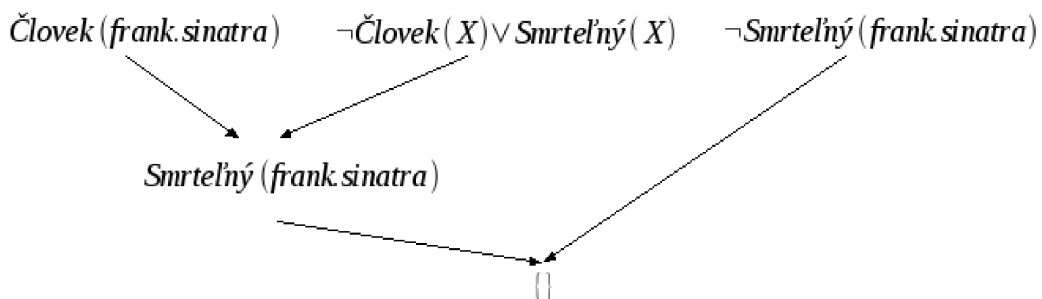
*Človek*(frank.sinatra)

$\forall X [ \text{Človek}(X) \Rightarrow \text{Smrteľný}(X) ]$

**Cieľová formula:** $Smrteľný(franksinatra)$ **Sporná teória:** $Človek(franksinatra)$  $\forall X [Človek(X) \Rightarrow Smrteľný(X)]$  $\neg Smrteľný(franksinatra)$ **Sporná teória v klauzulárnom tvare:**(1)  $Človek(franksinatra)$ (2)  $\neg Človek(X) \vee Smrteľný(X)$ (3)  $\neg Smrteľný(franksinatra)$ **Dôkaz:**

1. Aplikáciou rezolučného pravidla na formuly (1) a (2) odvodíme nové tvrdenie (4)  $\neg Smrteľný(franksinatra)$ , ktoré pridáme do spornej teórie.
2. Aplikáciou pravidla na formuly (3) a (4) odvodíme prázdnu klauzulu. Sporom sme dokázali, že Frank Sinatra je smrteľný.

Odvedenie dôkazu sa dá veľmi prehľadne zobrazit' pomocou tzv. **derivačného (odvodzovacieho) stromu**. Uzly tohto stromu reprezentujú jednotlivé formuly a hrany spájajú predpokladové klauzuly rezolvenčného pravidla s odvodenou formulou. Príklad derivačného stromu pre náš príklad je na obr. 15.



**Obr. 15.** Derivačný strom pre dôkaz smrteľnosti Franka Sinatru.

**Príklad:** Kauza zašliapnutého morčat'a

**Znenie úlohy:** Florián vlastní gekona. Každý, kto vlastní gekona, je milovník zvierat. Žiaden milovník zvierat nezašliapne žiadne zviera. Kleofáša zašliapol Florián alebo Krišpín. Kleofáš je morča a morča je zviera. Zašliapol Kleofáša Krišpín?

**Teória:**

$$\exists X [Gekon(X) \wedge Vlastní(flórián, X)]$$

$$\forall X [\exists Y [Gekon(Y) \wedge Vlastní(X, Y)] \Rightarrow MilovníkZvierat(X)]$$

$$\forall X [MilovníkZvierat(X) \Rightarrow \forall y [Zviera(Y) \Rightarrow \neg Zašliapne(X, Y)]]$$

$$Zašliapne(flórián, kleofáš) \vee Zašliapne(krišpín, kleofáš)$$

$$Morča(krišpín)$$

$$\forall X [Morča(X) \Rightarrow Zviera(X)]$$

**Cieľová formula:**

$$Zašliapne(krišpín, kleofáš)$$

**Sporná teória v klauzulárnom tvare:**

$$(1a) \quad Gekon(A)$$

$$(1b) \quad Vlastní(flórián, B)$$

$$(2) \quad \neg Gekon(D) \vee \neg Vlastní(C, D) \vee MilovníkZvierat(C)$$

$$(3) \quad \neg MilovníkZvierat(E) \vee \neg Zviera(F) \vee \neg Zašliapne(E, F)$$

$$(4) \quad Zašliapne(flórián, kleofáš) \vee Zašliapne(krišpín, kleofáš)$$

$$(5) \quad Morča(kleofáš)$$

$$(6) \quad \neg Morča(X) \vee Zviera(X)$$

$$(7) \quad \neg Zašliapne(krišpín, kleofáš)$$

**Príklad dôkazu:**

- Pomocou (7) a (4) odvodíme:

$$(8) \quad Zašliapne(flórián, kleofáš)$$

- Pomocou (8) a (3) odvodíme:

$$(9) \quad \neg MilovníkZvierat(flórián) \vee \neg Zviera(kleofáš)$$

- Pomocou (9) a (2) odvodíme:  
(10)  $\neg Gekon(D) \vee \neg Vlastní(flórián, D) \vee \neg Zviera(kleofáš)$
- Pomocou (10) a (1b) odvodíme:  
(11)  $\neg Gekon(D) \vee \neg Zviera(kleofáš)$
- Pomocou (11) a (1a) odvodíme:  
(12)  $\neg Zviera(kleofáš)$
- Pomocou (10) a (6) odvodíme:  
(13)  $\neg Morča(kleofáš)$
- Pomocou (13) a (5) odvodíme **spor**. Kauza je vyriešená, Krišpín zašliapol Kleofáša.

Ako ste si mohli všimnúť, pri dokazovaní som vždy používal najnovšie odvodené tvrdenie. Tento postup nie je povinný a už vôbec jediný. Bola to stratégia. Algoritmus, ktorý sa snaží aplikovať rezolučné pravidlo na formuly v teórii môže postupovať systematicky podľa zvolenej stratégie a môže tiež pokojne používať heuristiky, ktoré by mohli dokazovanie urýchliť.

## 5 Sémantické modelovanie znalostí

Predstavte si, že máte doma vorvaňa. Napíšte si slovo “vorvaň” na papier. A teraz skúste vymyslieť, čo o vorvaňoch viete. Možno vás napadne, že vorvaň je cicavec. Tak si napíšte na papier slovo “cicavec”, nakreslite šípku smerujúcu od slova “vorvaň” k cicavcovi a pomenujte šípku názvom “je”. To, čo ste nakreslili, je vzťah, ktorý hovorí, že “vorvaň je cicavec”. Možno vás napadne, že vorvaň žije vo vode. Urobte zas to isté, napíšte slovo “voda” a prepojte ho s vorvaňom šípkou pomenovanou “žije v”. Ak si takto napíšete všetko, čo viete o vorvaňovi, pokračujte ďalšími slovami. Zoberme si napríklad slovo “voda”. Mňa napadá, že vo vode žijú ryby. Ak by som si to nakreslil, obrázok by vyjadroval, že vorvaň je cicavec, ktorý žije vo vode a tiež, že vo vode žijú ryby. Ak by ste takto pokračovali ďalej a nechali sa vniesť do komplikovaných zákutí vlastných asociácií, dostali by ste obširnu sieť vzťahov. To, čo by ste vytvorili, sa volá **sémantická sieť**.

Jedno z prvých použití sémantických sietí popísal v roku 1968 Ross Quillian vo svojej dizertačke. Vymýšľal, ako asi u ľudí vyzerá organizácia sémantickej pamäte alebo inak, pamäte slovných konceptov. Sémantická sieť bol graf, kde uzly predstavovali slovné koncepty a orientované hrany medzi konceptami reprezentovali ich vzájomné vzťahy. Sémantické siete boli určené na reprezentáciu objektívnych a neemotívnych vlastností. Popis bol zameraný hlavne na popis objektov a ich vzťahov, nie na to, aké pocity v nás vyvolávajú. Quillianov pôvodný predpoklad bol, že význam jednotlivých pojmov sa dá reprezentovať, ako množina slovných asociácií.

Aby bolo jasné, nad čím tu teraz hlbam, predstavte si, že na vás niekto od chrbta vybalí slovo “mechúrnik”. Ak sa okamžite neurazíte, možno nazberáte silu na to, aby ste zistili, čo to ten mechúrnik je. Tak sa pozriete do encyklopédie, kde na vás číha

veta: “Mechániky alebo dvojlistovce sú skupina vývojového stupňa epitelovce”. A ste hneď múdrejší, no nie? Ale keďže váš skúmový duch nedopustí, aby ste veci neprišli na koreň, pokúsite sa zistiť, čo sú to dvojlistovce a čo sú to epitelovce. A dopátrate sa k tomu, že dvojlistovce majú telo zložené z dvoch vrstiev, kde jedna má pohybovú a obrannú funkciu a druhú majú na trávenie. O epitelovcoch zistíte, že sú vývojovým stupňom živočíchov s dobre vyvinutou diferenciáciou zárodočných listov, čo je síce skvelé, ale veľmi to s vami nepohne. Na druhej strane vás to nenechá ľahostajnými, tak sa postupne pokúšate pátrať po jednotlivých významoch slov, kým sa nedostanete k slovám, ktoré namiesto toho, aby na vašej tvári spôsobovali vydesený výraz, vám dávajú zmysel. A pomocou slov, ktorým rozumiete, sa to už dá pekne spätne poskladať do čohosi, čo význam dáva. A sémantické tu reprezentujú spôsob ako takéto pojmy, ich vlastnosti a vzťahy reprezentovať.

Sémantické siete vniesli medzi schémy reprezentácie znalostí, ako sa kedysi zvyklo hovoriť, nový, svieži vietor. Priniesli celkom iný pohľad na to, ako znalosti reprezentovať. Rôznych nevídaných variácií na tému sémantické siete je ozaj požehnané. Náhodne vybraté, bez vysvetlenia a len kvôli tomu pocitu tu podhodím definičné, výrokové, implikačné, vykonateľné, učiace sa, hybridné a kadejaké iné siete. Ak by ste sa do toho chceli seriózne oprieť, nájdete to všetko v [7]. My sa tu budeme zaoberať hlavne automatickým uvažovaním pomocou sémantických sietí a letmo sa ešte pozrieme na dva najznámejšie typy reprezentácie znalostí, ktoré sú na sémantických sieťach postavené. Konkrétne si dáme nízky prelet nad rámcami a ontológiami.

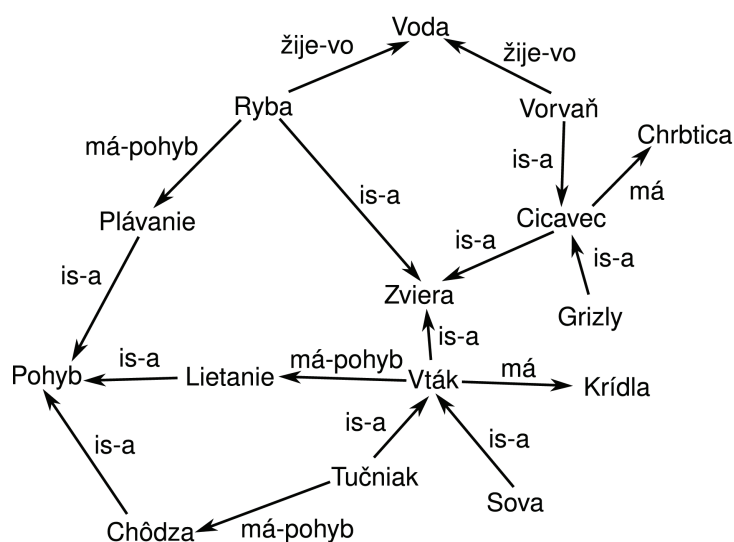
## 5.1 Sémantické siete

Istú predstavu už zrejme máte, takže začnem rovno konkrétnym príkladom. Na obr. 16. máte časť sémantickej siete, ktorej rast spustil vorvaň z úvodu. Aby sme opis sformalizovali, sémantická sieť je orientovaný graf, kde **uzly reprezentujú pojmy** (tiež tzv. **koncepty**) a **orientované hrany reprezentujú ich vzájomné vzťahy**. Pojem je potrebné chápať všeobecne, pretože to môže byť čokoľvek. Pokojne si môžeme vytvoriť sémantickú sieť, ktorá mi bude popisovať sémantickú sieť a potom aj vlastnosť sémantickej siete bude modelovaná ako pojem. Nie, mojim úmyslom nie je kompletne to domotať. Odrazíme sa od toho, že je to orientovaný graf pojmov a ich vzťahov a pri tom ostaneme.

### Vlastnosti sémantických sietí

Sémantické siete majú niekoľko špecifických vlastností. Jednou z nich je schopnosť reprezentovať hierarchické vzťahy. Ak sa pozriete na sémantickú sieť na obrázku, uvidíte napríklad to, že vorvaň je cicavec a cicavec je zviera. Hierarchický vzťah je reprezentovaný špecifickou vlastnosťou (hranou), ktorá sa volá **is-a** (z anglického “is a”, napr. “vorvaň je cicavec” – “sperm-whale is a mammal”). Táto hrana sa používa na reprezentáciu **dedičnosti**. Dedičnosť znamená, že vlastnosti nadradených pojmov v hierarchii sa automaticky prenášajú na tie pojmy, ktoré sú v hierarchii nižšie. Ak sa pozriete na našu sieť na obrázku, o vorvaňovi vieme priamo zistiť, že žije vo vode, ale

z nadradeného pojmu cicavec tiež vieme, že vorvaň má chrbticu (keďže cicavce majú chrbticu). Hrana *is-a* je tranzitívna, takže platí: ak  $A$  *is-a*  $B$  a  $B$  *is-a*  $C$ , potom  $A$  *is-a*  $C$ . Tranzitivita nám zabezpečuje, aby sa pre každý pojem odvodili všetky vlastnosti všetkých nadradených pojmov.



Obr. 16. Príklad sémantickej siete.

Ďalšou špecifickou vlastnosťou týchto sietí je schopnosť reprezentovať viacznačnosť. V sieti na obrázku si môžete všimnúť, že vtáky lietajú (“*Vták pohyb Lietanie*”). Tiež si ale môžete všimnúť, že tučniaky sú vtáky, ale zároveň sa pohybujú chôdzou. Ako je to teraz správne? Tučniaky chodia alebo lietajú? Riešenie by malo byť v takýchto prípadoch vždy skryté v inferenčnom mechanizme. Ak inferencia neberie do úvahy viacznačnosť, tak vám pre tučniaka odvodí, že chodí, ale aj to, že lieta (a ja, ako zaniatený milovník prírody, by som ozaj len nerád poznal dôvod, prečo sa tučniak dostal do vzduchu). Takémuto uvažovaniu sa hovorí **monotónne**. Pre monotónne uvažovanie platí, že ak je možné zo znalostí odvodiť nejakú informáciu, bude ju možné odvodiť aj potom, keď medzi znalosti pridáme akúkoľvek novú informáciu, hoci aj viacznačnú. Ak by bolo odvodzovanie prispôbené viacznačným informáciám, potom by boli pre skúmaný pojem odvodené z nadradených pojmov hodnoty len tých vlastností, ktoré sám nemá definované. Toto je **nemonotónne** uvažovanie, pretože pôvodná informácia by nebola po pridaní novej hodnoty vlastnosti odvodená. V tomto konkrétnom prípade ide o tzv. **defaultné** uvažovanie. Ak nemá pojem definovanú vlastnosť, tak sa ako defaultná hodnota použije hodnota vlastnosti z najbližšieho nadradeného pojmu. Monotónne uvažovanie súvisí s tzv. **predpokladom otvoreného sveta**. V otvorenom svete platí, že ak niečo nevieme, neznamená to, že to neexistuje. Preto, ak by sme sa našej siete spýtali: “Má sova zobák?”, odpoveď by mala byť “neviem”, pretože ak tam tá znalosť raz pribudne, odpoveď bude “áno”. V **uzavretom**

**svete** by bola odpoveď “nie” (takto napríklad pracujú databázy). Sémantické siete sú navrhnuté pre prácu v otvorenom svete. A to je dôvod, prečo majú s riešením viacznačnosti celkom seriózne problémy. Na druhej strane, všetko závisí od použitého inferenčného mechanizmu. Predpisy v tomto prípade v podstate neexistujú. Aby to nebolo len na strojoch, v omnoho väčšej miere to závisí aj od schopností samotného modelára a od toho, nakoľko pozná inferenčný mechanizmus (fakt je, že s týmito reprezentáciami sa dá celkom slušne hazardovať).

Doteraz sme sa bavili len o reprezentácii všeobecných konceptov a ich vzťahov. Sémantické siete však umožňujú definovať aj špeciálne uzly – **inštancie**, ktoré reprezentujú špecifické prípady konceptov. Napríklad, mám doma vorvaňa, ktorý sa volá Imrich. A ten je špecifickým prípadom všeobecného pojmu vorvaň a má svoje vlastné špecifické vlastnosti. V tomto svetle sú pojmy v sémantickej sieti akýmsi všeobecným popisom vzťahov. Pomocou inšancií je potom možné definovať skutočné vzťahy medzi reálnymi objektami.

### **Odvodzovanie znalostí v sémantických sieťach**

Sémantické siete si vyslovene žiadajú, aby boli dotazované a zvyčajne slúžia na zodpovedanie otázok. Môžeme si vyžiadať popis konkrétneho pojmu alebo inštancie. Tu by mala sieť reagovať na príkaz: “Povedz mi všetko, čo vieš o ...”. V tomto prípade odvodzovací mechanizmus začne požadovaným pojmom a pomocou hrán, ktoré smerujú od neho, popíše pojmy asociované týmito vlastnosťami. Potom pomocou vlastností týchto pojmov popíše ďalšie asociované pojmy a takto pokračuje, kým existujú hrany, pomocou ktorých je možné dostať sa ďalej. Tomuto mechanizmu sa tiež hovorí **šírenie aktivácie** (spreading activation). Keďže vo veľkých sieťach to môže trvať ozač predlho, popis sa zvyčajne robí len do istej hĺbky. Pojem je popísaný len v istom okolí. Povedzme pre okolie vo vzdialenosti 2 by sme dostali popis len tých asociovaných pojmov, ktoré nie sú prepojené cez viac ako 2 vlastnosti. Keďže sa to volá šírenie aktivácie, môžete si to predstaviť tak, že v požadovanom pojme odpálite impulz, ktorý sa šíri v smere hrán sieťou. S každou hranou intenzita impulzu klesá, a tak sa dostane len do istej vzdialenosti od štartovacieho pojmu. Ak by sme odpálili šírenie aktivácie na tučniakovi v našej sieti, výstup by bol takýto:

- *Tučniak*
  - *má-pohyb Chôdza*
    - *is-a Pohyb*
  - *is-a Vták*
    - *má Kridla*
    - *má-pohyb Lietanie*
    - *is-a Zviera*

Môžeme tiež naformulovať zložitejšiu otázku a sémantická sieť by nám mala vrátiť zmysluplnú odpoveď. Otázka je zvyčajne formulovaná ako časť sémantickej siete, ktorá sa potom porovnáva so sémantickou sieťou všetkých znalostí. Sieť otázky môže obsahovať uzly s premennými, úlohou inferenčného mechanizmu je doplniť chýbajúce



informácie. Našej ukázkovej sieti sa môžeme napríklad spýtať: “Aké zviera má krídla a chodí?”. Formálne by táto otázka vyzerala ako množina hľadaných vzťahov:

$$\{[X \text{ is-a } Zviera][X \text{ má } Krídla][X \text{ má-pohyb } Chôdza]\}$$

Inferenčný mechanizmus by odvodil, že ide o tučniaka, pretože pomocou dedičnosti by zistil, že krídla majú vtáky a chôdzou sa pohybuje tučniak, čo je tiež vták. Sieť pre túto otázku nájdete na obr. 17a). Ak by otázke vyhovovalo viac pojmov, inferenčný mechanizmus by odvodil všetky. Príkladom takejto otázky by bolo: “Aké pohyby vykonávajú zvieratá?”. Formálne by táto otázka vyzerala ako množina hľadaných vzťahov:

$$\{[X \text{ is-a } Zviera][X \text{ má-pohyb } Y]\}$$

Tu by inferenčný mechanizmus musel cez *is-a* hrany pretraverzovať ku všetkému, čo je zviera a má asociovaný nejaký pojem pomocou hrany *má-pohyb*. Odpoveď pre dvojice premenných ( $X$ ,  $Y$ ) by bola:

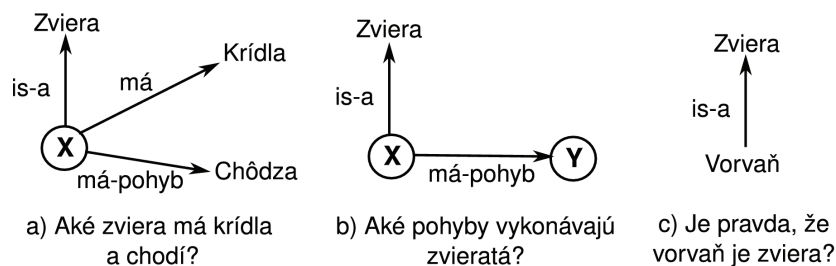
$$\{(Ryba, Plávanie), (Vták, Lietanie), (Tučniak, Chôdza), (Tučniak, Lietanie)\}$$

Dve odpovede pre tučniaka sú dôsledkom toho, ako sémantická sieť zvláda viacznačnosť. Keďže sú tieto fakty v znalostiach, je to správna odpoveď. Sieť pre túto otázku máte na obr. 17b).

Ak by sme v otázke nepoužili premenné, ale len pojmy zo siete znalostí, otázka by bola formulovaná ako: “Je pravda, že ...?”. Ak sa našej sieti spýtame napríklad: “Je pravda, že vorvaň je zviera?” Inferenčný mechanizmus cez hierarchiu dotraverzuje k faktu, že vorvaň je cicavec a cicavec je zviera. Bez váhania odpovie: “samozrejme”. Sieť pre túto otázku nájdete na obr. 17c).

Ak vytvárate sémantickú sieť zameranú na pojmy zo špecifickej oblasti (**domény**), tak vytvárate vlastne model tejto domény. Tento model reprezentuje spôsob, ako sa významy modelovaných pojmov dajú popísať, vysvetliť pomocou iných pojmov. A tomuto sa hovorí **sémantické modelovanie**.

Sémantické modelovanie a modely postavené na sémantických sieťach sú v súčasnosti snáď najpoužívanejším a technologicky najpodporovanejším typom reprezentácie znalostí. Momentálne sú najrozšírenejším prístupom sémantického modelovania tzv. ontológie, ktoré sú podporované až neuveriteľným množstvom rôznych modelovacích a dotazovacích jazykov, ale aj inferenčných mechanizmov. Zľahka sa na ne pozrieme. Cesta k ontológiám ale vedie cez tzv. rámce, na ktorých sú postavené.



Obr. 17. Príklady sémantických sietí pre otázky.

## 5.2 Rámce

Dá sa povedať, že ľudia v rôznych situáciách využívajú veľa vedomostí, ktoré vznikli na základe predchádzajúcich skúseností. Ak si omylom objednáte hotel, v ktorom ste ešte neboli, možno naivne predpokladáte, že tam bude telka, že budete mať raňajky v cene ubytovania, a že bar bude otvorený aj po polnoci. A to len preto, že v hoteloch, na ktoré ste zvyknutí to tak je. Na základe skúseností máte isté očakávania. V roku 1975 prišiel Marvin Minsky s tým, že ak ste konfrontovaní s novou situáciou, ktorej podobnú ste už zažili, v mozgovni sa vám aktivuje štruktúra – **rámec** [7]. Táto štruktúra obsahuje popis všeobecnej situácie daného typu, ktorý je získaný skúsenosťami a nová situácia do rámca zapadne s tým, že sa zmenia a doladia špecifické detaily. Toto je v kocke hlavná myšlienka rámcov.

Ak to aspoň prehľadovo zoberieme z tej technickejšej stránky, o rámcoch sa hovorí ako o rozšírení sémantických sietí. Jedným z problémov sémantických sietí je, že znalosti sú voľne porozhadzované medzi jednotlivé pojmy a ich vzťahy. Pri zložitých modeloch sa môžete v sieti veľmi ľahko stratiť. Rámce zavádzajú do sémantických sietí poriadok tým, že informácie sú reprezentované v ucelenejších jednotkách. Rámec je vlastne údajová štruktúra, ktorá reprezentuje popis objektu alebo situácie. Podobne, ako v sémantických sieťach, sa v rámcoch používajú hierarchické vzťahy a dedičnosť. V sémantickej sieti boli vlastnosti nejakého pojmu reprezentované pomocou prepojení s inými pojmi. Rámce definujú pre objekt očakávané vlastnosti pomocou **atribútov**. Hodnota atribútu môže byť reťazec, číslo, vlastne ľubovoľný dátový typ, ale aj iný rámec. Ak by sme sémantickú sieť chceli previesť na rámce, jednoducho by sme vlastnosti zložitejších pojmov reprezentované hranami grafu scvrkli do atribútov údajových štruktúr a tieto navzájom prepojili. V sémantickej sieti by sme takto urobili poriadok, pretože vlastnosti jednotlivých pojmov by sme mali pekne po hromade v údajových štruktúrach. Ak by vám to náhodou pripomínalo to, čo poznáte ako triedy v objektovo-orientovanom programovaní, je na to dôvod. Vzniklo totiž z rámcov.

Atribúty rámcov mali pomerne špeciálne vlastnosti. Atribút mohol obsahovať:

- **Hodnotu atribútu:** tá mohla byť ľubovoľný bežný dátový typ alebo sa mohla odkazovať na iný rámec.
- **Očakávanú (default) hodnotu:** ak sa vytvárala nová inštancia rámca, buď bola hodnota atribútu známa alebo sa použila default hodnota. Ak nebola definovaná ani očakávaná hodnota, zdedila sa z najbližšieho nadradeného rámca v hierarchii. Touto vlastnosťou rámce napríklad odstránili problém s viacznačnosťou.
- **Rozsah hodnôt (range):** pre atribút mohla byť definovaná konečná množina hodnôt.
- **Pripojené procedúry:** tieto procedúry reagovali na rôzne udalosti. Ak sa atribútu prihodila špecifická udalosť, spustila sa špecifická procedúra. Typickými bola napr. procedúra IF-NEEDED (ak-treba), ktorá vedela odvodiť hodnotu atribútu, ak nebola definovaná. Mohla ju získať priamo z atribútov rámca alebo sa mohla pokojne prehnat' celou hierarchiou a odvodiť ju podobne, ako pri dotazovaní sémantickej siete. Iné procedúry boli určené na kontrolu konzistencie dát, napr. IF-ADDED (ak-pridané), IF-DELETED (ak-zmazané), IF-CHANGED (ak-zmenené). Pri takýchto udalostiach procedúra napr. skontrolovala, či zmena nespôsobuje kolízie v iných dátach alebo mohla zabezpečiť ľubovoľnú inú definovanú nekalosť. Tieto pripojené programy predstavujú procedurálne znalosti, ktoré vedeli, **ako** získať informácie, resp. ako postupovať pri špecifických udalostiach. Pripojené procedúry reprezentovali **procedurálne znalosti**.

Ak to zovšeobecním a zosumarizujem, snád' nikto nebude zazerat', ak poviem, že rámce sú sémantické siete so štruktúrovanými pojmami a procedurálnymi znalosťami. S pôvodnými rámcami je vlastne už celkom ťažké stretnúť sa v praxi. Na rámcoch, ako reprezentácii znalostí, je ale veľmi dôležité to, že dali základ neuveriteľne veľkému počtu prístupov, ktoré sa aktívne vyvíjajú a využívajú v súčasnosti.

### 5.3 Ontológia

Pôvod všetkých prístupov, ktoré som doteraz spomínal, sa datuje do priestoru päťdesiatych až sedemdesiatych rokov dvadsiateho storočia. Tuto urobím menšiu výnimku, pretože ontológie sú jednak výdatne rozpracovaným formalizmom reprezentácie znalostí a jednak v súčasnosti vedú veľmi aktívny život. Ono to celé súvisí s vývojom tzv. **sémantického webu** [8]. Je to aktivita W3C konzorcia a všetkým nám ide o to, aby sme mali informácie na webe pekne sémanticky popísané, čím by sa malo skvalitniť vyhľadávanie toho relevantného, a to tak pre ľudí, ako aj pre stroje. To by mali umožniť tzv. **znalostné technológie**, ktorých cieľom je vytvoriť znalostný a významový popis informácií a umožniť tzv. **sémantickú interoperabilitu**. Sémantická interoperabilita je schopnosť dvoch (alebo viacerých strojov) vymieňať si navzájom informácie, ktoré vedia spracovať a interpretovať na základe ich významu. Mašinky by teda mali byť schopné rozumieť (tým myslím len formálnu stránku veci) informáciám,

ktoré si vymieňajú. Áno, znie to veľkolepo. Aby sa niečo takéto dalo uspokojivo implementovať, stačí dodať, že význam jednotlivých pojmov je vždy definovaný ľuďmi a stroje musia tieto modely zdieľať. Ak to veľmi nadľahčím, ak mašiny používajú rovnaký model, mali by vedieť, kedy hovoria o tom istom. A na realizáciu týchto monumentálnych cieľov sa v súčasnosti používajú práve **ontológie**.

Keďže okolo ontológií momentálne vo vedeckej a technologickej komunite prebieha riadny rozruch, máte o nich k dispozícii síce konečné, ale len s ťažkosťami spočítateľné množstvo informácií. Ak sa to pokúsim zovšeobecniť, ontológie sú reprezentáciou znalostí postavenou na sémantických sieťach a na reprezentáciu konceptov využívajú podobné štruktúry, ako rámce. Pomocou ontológií vytvárate sémantický znalostný model konkrétnej oblasti. A tieto znalostné modely je možné vzájomne prepájať do väčších celkov. Na tomto je veľmi vďačné to, že do vašej ontológie môžete dodať znalosti importované z modelu, ktorý vytvoril niekto úplne iný. A aby niečo také bolo možné, stroje musia byť schopné takýto externý model spracovať a integrovať medzi už existujúce znalosti. Je tu teda celkom slušný predpoklad, že jazyky, pomocou ktorých sú ontológie popísané, musia byť štandardizované. Jedným cieľom štandardizácie je práve to, aby boli znalosti v ontológiách popísané rovnakým jazykom. To umožňuje ich ľahké počítačové spracovanie (znalostné modely sú spracovávané rovnakým spôsobom) a tiež ďalšie špeciálne vlastnosti, ako napríklad ľahkú integráciu viacerých modelov, ľahké znovupoužitie (reuse) špecifických znalostných modelov, apod. Štandardy sa okrem jazyka pre popis znalostí týkajú aj samotných modelovaných oblastí. Povedzme napríklad, že existuje voľne dostupná ontológia popisujúca ľudí, ich činnosti a vzájomné vzťahy medzi ľuďmi. Príkladom takéhoto modelu je FOAF – Friend Of A Friend ontológia [9], často používaná pre modelovanie tzv. sociálnych sietí. Ak sa rozhodnete vytvoriť model, ktorý tiež zahŕňa informácie o ľuďoch a ich vzťahoch, volíte medzi dvoma možnosťami. Buď si vytvoríte vlastný model alebo jednoducho integrujete už existujúci model. Povedzme, že váš systém si bude potrebovať vymieňať informácie o ľuďoch s iným systémom (napríklad pomocou otázok typu: “S ktorými ľuďmi mal Béla spoločnú publikáciu?”). Ak obidva systémy používajú na popis medziľudských vzťahov rovnaký model (napr. spomínanú FOAF ontológiu), tak komunikácia bude omnoho jednoduchšia a zmysluplnejšia, pretože tieto systémy budú zdieľať rovnaký popis jednotlivých pojmov. A naopak, ak by oba systémy používali rôzne reprezentácie medziľudských vzťahov, takéto otázky by bolo najprv nevyhnutné pretransformovať z jedného popisu do iného a na takúto transformáciu je zase potrebné definovať spôsob, ako to spraviť a to by muselo byť definované pre každé dva rôzne systémy s rôznymi modelmi, atď. Aj preto vznikajú štandardy, a aj preto je dobré ich používať. Zvyšujú sémantickú interoperabilitu systémov.

Vrátim sa ešte späť k štandardným jazykom pre popis ontológií. Ak mi niekto v slabej chvíli povie slovo “ontológia”, v momente sa mi vybaví toto: RDF(S), OWL, N3, SWSL, WSMO, WSML, OWL-S, SAWSDL, SWRL, ERDF, TRIPLE, RIF, R2ML, RuleML, RDQL, SPARQL, OWL-QL, a týmto to zďaleka nekončí. Toto všetko sú štandardné jazyky pre popis ontológií, pravidiel, ktoré s ontológiami vedia pracovať a jazyky na dotazovanie ontológií. Ak sa niekto chce stať znalostným modelárom, už prvý pohľad na tento zoznam môže spôsobiť, že ho jedného dňa nájdú

čupieť po tme niekde v rohu, ako prehodnocuje svoju budúcnosť. Na druhej strane, nič nie je také zlé, ako to na prvý pohľad vyzerá. A znalostné modelovanie je technológia, ako každá iná. A našinec nemusí aktívne pracovať v oblasti technológií alebo aplikovaného výskumu, aby mu bolo jasné, že každý deň je niečo nové a musí sa nepretržite učiť. Spať na vavrínoch sa totiž nikdy neoplatí.

## 6 Záver

Cieľom tohto textu bolo priblížiť základné princípy tých najznámejších typov reprezentácie znalostí v symbolickej umelej inteligencii. Je pravda, že každý typ popisu znalostí je vhodný na riešenie iného typu úlohy. Ale ak by som sa to pokúsil zovšeobecniť, došiel by som k tomu, že systémy umelej inteligencie sú vždy poskladané z dvoch hlavných častí. V prvom rade sú tu samotné znalosti, ktoré sú u väčšiny systémov deklarované pomocou nejakého formálneho popisu. Tieto znalosti popisujú, čo systém vie. Znalosti môžu mať ale aj procedurálny charakter, môžu to byť časti kódu, ktorý vie, ako získať nejakú informáciu. Anakoniec sú tu heuristiky, pragmatické znalosti o riešenej úlohe, ktoré pomáhajú zefektívniť činnosť celého systému. Druhou časťou je odvodzovací mechanizmus, ktorý vie so znalosťami manipulovať. Zo známych faktov odvodzuje nové, ešte neznáme informácie. Zvykne sa tomu tiež vznešene hovoriť uvažovanie. A tento proces môže byť použitý na nevidané spektrum riešenia akýchkoľvek úloh, či sa už jedná o plánovanie, diagnostiku, dokazovanie alebo zodpovedanie otázok.

Ak by som chcel zdôrazniť slovo *inteligentný*, povedal by som, že ak vyvíjame systém s umelou inteligenciou, je logické predpokladať, že nebude určený na riešenie jedinej úlohy. Skôr by som očakával, že tento systém bude riešiť celú škálu úloh rovnakého typu. Niečo podobného som spomínal pri expertných systémoch. Znalosti sú formulované ako produkčné pravidlá, ak vymeníte pravidlá v systéme za iné, systém bude expertom v inej oblasti. A toto je účel reprezentácie znalostí. Odvodzovací mechanizmus pracuje stále rovnako. A vždy pracuje so znalosťami, ktoré sú formulované v nejakom jazyku (či už je to predikátová logika, či produkčné pravidlá alebo nejaký štandard na popis ontológií). Ak si teda navrhujete systém a nepoužívate na reprezentáciu znalostí nejaký štandardizovaný jazyk, navrhnite si popis znalostí tak, aby ste pomocou tejto reprezentácie vedeli popísať čo najviac možných úloh. Inferenčný mechanizmus zaujíma len forma, obsah pre neho zaujímavý nie je. Rovnako dobre svoju prácu urobí pre akékoľvek znalosti, s ktorými vie pracovať.

Aby som do toho taktu ku koncu primiesil ešte trocha hmly, poviem, že medzi jednotlivými reprezentačnými formalizmami je len veľmi tenká hranica. Zvyčajne viete prepísať jednu reprezentáciu na inú a nakoniec to v praxi zvyčajne tak či tak skončí v predikátovej logike, na ktorú sa nasadí nejaká beštiálna hyper-rezolúcia, ktorá to celé zmáčne a ani pri tom brvou nepohne. Ale to už sa celkom zbytočne rútim do pomerne členitých detailov. Týmto celým chcem opäť len zvýrazniť základný princíp. Dôležité je mať dobre navrhnutý popis znalostí. To, čo váš systém bude vedieť a spôsob, akým to bude popísané vždy závisí od úlohy. A od zvolenej reprezentácie znalostí a cieľov systému závisí zasa implementácia odvodzovacieho mechanizmu.

Verím, že tento text aspoň sčasti pomohol vniest' trocha svetla do zákutí symboliky a spôsobu, akým pracuje. A ak aspoň nejaká časť tohto textu poslúžila ako vysvetlenie alebo inšpirácia, potom tento text splnil svoj účel.

## Literatúra

- [1] A. Turing. Computing machinery and intelligence. *Mind*, 1950.
- [2] CAPTCHA: Telling Humans and Computers Apart Automatically <http://www.captcha.net/>, posledná zmena: 18.11.2008.
- [3] J. Searle. Minds, Brains, and Programs. *Behavioral and Brain Sciences* 3, 1980.
- [4] A. Newell, H.A. Simon. Computer Science as Empirical Inquiry: Symbols and Search. *Communications of the ACM*, 1976.
- [5] P.E. Hart, N.J. Nilsson, B. Raphael. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics SSC4* (2), 1968.
- [6] J. A. Robinson. A Machine-Oriented Logic Based on the Resolution Principle. *Journal of the ACM*, vol. 12, 1965.
- [7] M. Minsky. A Framework for Representing Knowledge. In *Patrick Henry Winston (ed.), The Psychology of Computer Vision*, McGraw-Hill, New York, 1975.
- [8] W3C Semantic Web Activity, <http://www.w3.org/2001/sw/>, posledná zmena: 26.1.2009.
- [9] The Friend of a Friend (FOAF) Project, <http://www.foaf-project.org/>

# Inteligentné vnorené systémy

Štefan KOZÁK<sup>1</sup>

**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 Analýza metód a prostriedkov inteligentnej syntézy vnorených systémov

Súčasná epocha sa vyznačuje agresívnym prienikom a využívaním výpočtovej techniky, numerických metód matematiky, riadiacich systémov, nanotechnológií do všetkých oblastiach ľudskej činnosti. Výpočtová technika je dnes všadeprítomná, či vo výrobkoch, domácich spotrebičoch, hračkách, priemyselných výrobkoch, priemyselných odvetviach. Moderné procesy sú dnes ovládané a riadené optimálnymi algoritmi s minimalizáciou energií, optimálnou komunikáciou a maximalizáciou kvalitatívnych parametrov. Každodenný život človeka je dnes výrazne ovplyvňovaný rozsiahlou digitálnou infraštruktúrou vplývajúcou na osobný život, priemysel, výrobné systémy, sociálne a obchodné činnosti a pod.

V súlade s uvedenými trendmi a nevyhnutnosťou integrácie informačných, komunikačných a riadiacich technológií sa dnes pred nami vytvára nedefinovaný a neviditeľný „kyberpriestor“ s rozsiahlou „kyberinfraštruktúrou“. Zvýšené požiadavky používateľov moderných výrobkov, systémov a zariadení nútia ich výrobcov zabudovať do nich vnorené výpočtové systémy a vysokointeligentný softvér umožňujúci procesy optimálne ovládať a riadiť. Vývoj nových inteligentných

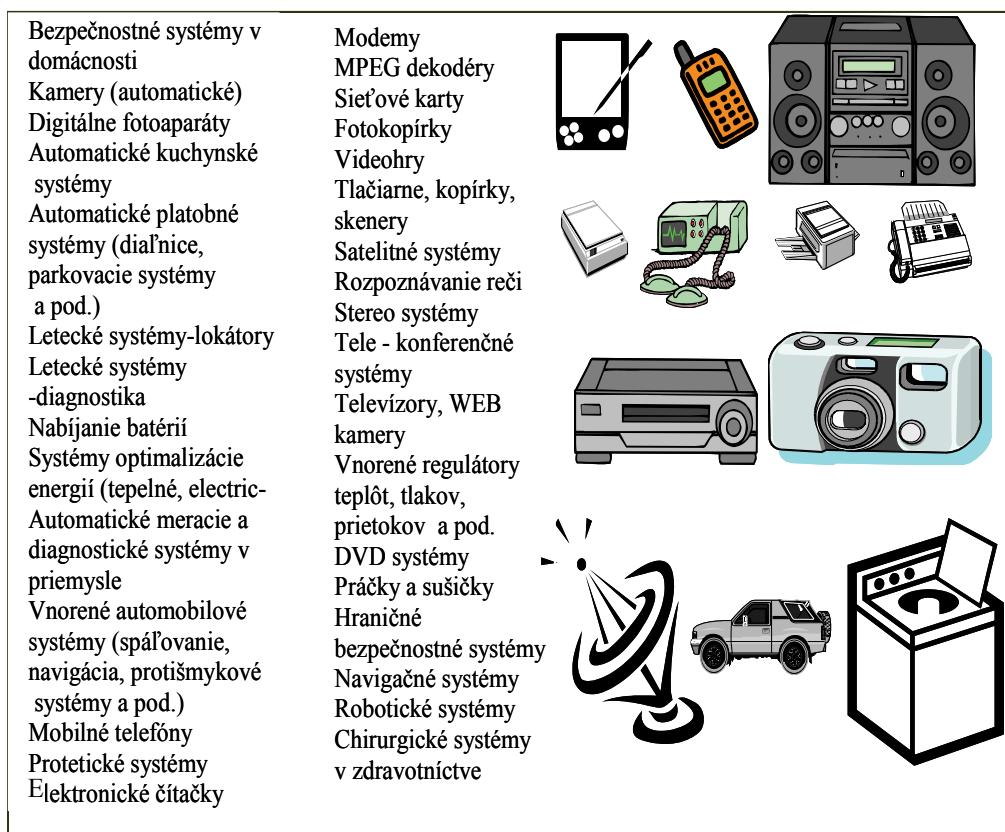
---

<sup>1</sup> Fakulta informatiky a informačných technológií, STU Bratislava, Ilkovičova 3, 812 19 Bratislava, E-mail: stefan.kozak@stuba.sk

systémov a výrobkov založených na embedded systémoch predstavuje v posledných rokoch najdynamickejšiu oblasť vývoja a aplikácie nových HW a SW prostriedkov.

Embedded systém (vnorený systém, zabudovaný systém) predstavuje systém, v ktorom je riadiaci počítač plne zabudovaný do zariadenia, ktoré ovláda a riadi pre definovanú činnosť. Vnorené počítačové systémy zabudované do výrobkov sú zväčša jednoúčelové systémy. Vzhľadom k tomu, že systém je určený pre konkrétny účel, môžu ho projektanti a vývojoví pracovníci už pri návrhu optimalizovať pre konkrétnu aplikáciu a zvýšiť tak jeho kvalitu, spoľahlivosť a cenu. Embedded systémy sú implementované zväčša do výrobkov, ktoré sa vyrábajú sériovo a vo veľkom množstve, čím je zabezpečená aj efektívnosť a ekonomická výhodnosť výroby.

Implementácii výpočtových systémov do výrobkov, systémov a zariadení odpovedá celosvetovo zavedený pojem embedded (zabudovaný, vnorený) systém. Prienik zabudovaných výpočtových systémov do výrobkov, systémov a zariadení má výrazný vplyv na ďalší vedecký pokrok a umožňuje vytvárať vysoko kvalitné výrobky a zariadenia s vysokou inteligenciou. Mohutný a rýchly prienik zabudovaných výpočtových systémov do spotrebnej elektroniky, zdravotníctva a strojárstva mení podstatným spôsobom ich kvalitu, zvyšuje ich konkurencieschopnosť a výrazne prispieva k zlepšeniu kvality života (obr. 1).



Obr. 1. Vybrané aplikácie vnorených systémov.



Vysoké kvalitatívne, bezpečnostné a ergonomické požiadavky používateľov nútia výrobcov neustále zabudovávať do svojich výrobkov vnorené výpočtové systémy s vysoko inteligentným softvérom umožňujúcim optimálne ovládať a riadiť procesy, do ktorých sú zabudované.

**Embedded (vnorený, zabudovaný) systém** je počítačový systém vložený do objektu (systému, zariadenia) za účelom riadenia, ovládania a zabezpečenia jeho požadovaných činností a funkcií, ich kvalitného a spoľahlivého vykonania, komunikácie s používateľom a manažovania zdrojov.

Na základe existujúcich analýz je možné konštatovať, že v súčasnosti až 91% výpočtových systémov predstavujú vnorené (zabudované) systémy. Vnorené systémy významne zvyšujú hodnotu mnohých výrobkov. Túto skutočnosť je možné dokumentovať, napr. dnes na automobiloch, v ktorých 20% ceny tvorí vnorený systém a dá sa očakávať, že v roku 2010 budú vnorené systémy predstavovať až 40% podielu jeho celkovej ceny. Najnovšie analýzy v oblasti priemyselnej informatiky a automatizácie potvrdzujú, že podiel vnorených systémov v priemyselných aplikáciách bude v roku 2010 tvoriť až 25 %, v spotrebnej elektronike okolo 40% a v zdravotníctve až 35%. Tempo rastu využívania vnorených systémov sa odhaduje na 10-12% ročne. Očakáva sa, že v roku 2020 budú vnorené mikroprocesorové systémy implementované vo viac ako 40 miliardách aplikácií.

## 1.1 Charakteristika a vlastnosti vnorených systémov

Zabudovaný (vnorený systém) je malý počítačový systém, ktorý je vložený do vnútra zariadenia (stroja, spotrebiča, elektronického alebo mechanického systému) za účelom dosiahnutia jeho vyššej inteligencie, bezpečnosti a spoľahlivosti, kvality riadenia, a teda aj jeho vyššej celkovej efektívnej činnosti.

Zabudovaný systém je jednoúčelový (dedicated) počítač pracujúci v reálnom čase, ako integrovaná súčasť nejakého uceleného systému. Služi na riadenie celého systému a poskytuje výpočtové služby ostatným častiam systému.

Vnorený počítačový systém zabudovaný do zariadenia nie je zväčša pre užívateľa viditeľný.

Vnorený systém je výpočtový systém obsahujúci zväčša jeden alebo niekoľko mikroprocesorov, ktorých hlavným cieľom je riadiť činnosť určitých zariadení, spotrebičov, procesov, displejov, motorov, strojov a prístrojov.

Program realizovaný mikropočítačom predstavuje dominantný prvok (mozog) systémov a zariadení, ktoré riadi a ovláda. Mikropočítač získava informácie zo snímačov, alebo od operátora a na základe programu - algoritmu ovláda elektronické, mechanické alebo vizualizačné systémy.

Jadrom vnoreného systému je algoritmus - cieľové prostredie umožňujúce program len spúšťať – realizovať (execute-only target environment), napr. mikroprocesorový systém s programom uloženým v pevnej pamäti.

V samotnom zabudovanom systéme sa program zväčša nevyvíja. Vývoj sa realizuje na hosťateľskom počítači (vývojovom systéme).

V priemyselných systémoch majú vnorené počítačové systémy najmä tieto funkcie:

zber dát a ich analýza a spracovanie, modelovanie,  
priame počítačové riadenie,  
supervízorové riadenie, optimalizácia činnosti,  
interakcia človeka so zariadením, vizualizácia procesov, pochodov a činností.

Údaje z procesu sú zbierané prostredníctvom snímačov, ktorých súčasťou sú ďalšie analógové obvody pre impedančné prispôsobenie, galvanické oddelenie a úpravu úrovni signálov (obr. 2, obr. 3). Upravený číslicový signál vstupuje do riadiaceho systému po spracovaní v A/D prevodníku a transformácii do číslicovej formy. Vstupné údaje do systému môžu byť aj v tvare logických signálov.

Základné požiadavky na vnorený systém sú:

možnosť jednoduchého pripojenia do siete (Ethernet, modem, CAN, MilCan atd.) pri použití štandardných komunikačných protokolov (TCP/IP, protokol point-to-point - PPP a iné),

možnosť grafického zobrazovania dejov v procese (ak je k dispozícii displej),

schopnosť spracovať, ukladať a posilať údaje,

schopnosť zdieľať údaje,

možnosť pripojenia na jednoduchšie štandardné periférne zariadenia (napr. tlačiareň, displej a pod.),

možnosť rozširovať vnorený systém podľa náročnosti aplikácie a vytvárať sieťové vnorené aplikácie (automobilový priemysel, vojenská technika a pod.).

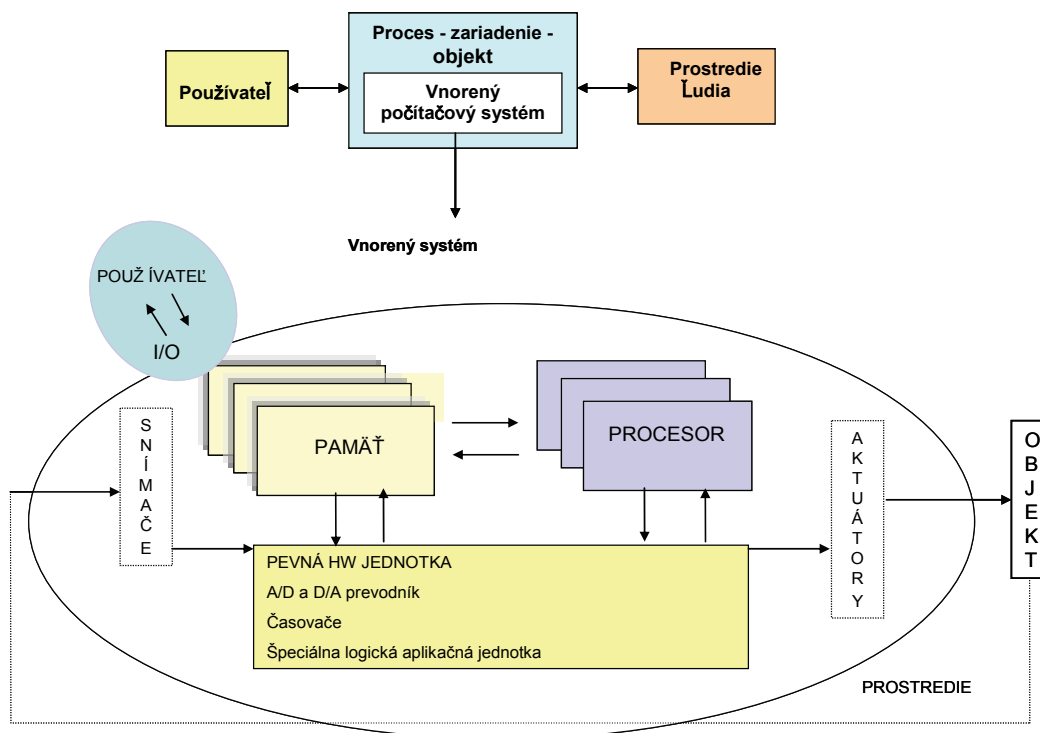
Najväčšiu aplikačnú oblasť využívania vnorených systémov predstavujú procesy riadenia a ovládania zariadení a systémov. V súčasnosti sú skoro všetky metódy a algoritmy riadenia pre vnorené systémy implementované v digitálnej forme. Digitálne algoritmy riadenia umožňujú realizovať riadenie vo vysokokvalitnej forme zabezpečujúcej splnenie požadovaných funkcií.

Existuje množstvo štruktúr riadenia (konvenčných a inteligentných), z nich najjednoduchšia je spätnoväzbová štruktúra, v ktorej zabudovaný počítač prostredníctvom algoritmu počíta riadiace zásahy, ktoré vstupujú do procesu a pôsobia na neho (obr. 3) tak, aby sa choval cieľavedome, t.j. aby jeho výstup konvergoval k požadovanej hodnote reprezentovanej referenčnou premenou vyjadrujúcou cieľ riadenia (konštanta, trajektória).

## 1.2 Využitie vnorených systémov v riadení

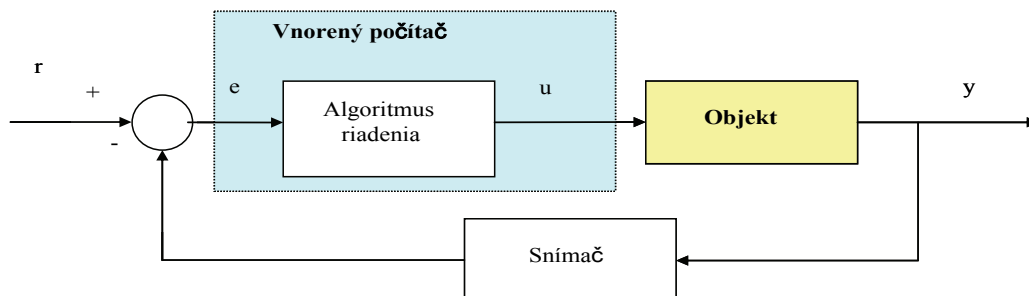
Vnorený riadiaci počítač realizuje určitú formu diskretného algoritmu riadenia (napr. PID regulátor, stavový regulátor, prediktívny regulátor alebo algoritmus riadenia založený na fuzzy logike, umelých neurónových sieťach alebo hybridných kombinovaných fuzzy-neuro a fuzzy-neuro-genetických technikách). Vnorený systém riadenia musí zabezpečovať bezpečnostné funkcie, prevádzkovú spoľahlivosť a zároveň musí sledovať, vyhodnocovať a spracovávať prevádzkové situácie a realizovať hlásenia a zobrazovanie prevádzkových situácií, sledovaných veličín a parametrov a alarmov. Pri aplikácii vnorených systémov do výrobkov spotrebnej elektroniky (televízory, mobily, práčky, kamery a pod.) interaguje používateľ so zariadením pomocou vyvedených tlačidiel, ktoré umožňujú zadávať žiadané (referenčné) parametre a iné pomocné príkazy, pre ovládanie a riadenie, ktoré sú vstupmi pre

riadiace algoritmy. Vnorený počítačový systém je teda integrovaný informačný, riadiaci a komunikačný systém, ktorý zbiera a spracováva údaje, komunikuje s používateľom a v prípade potreby indikuje a zobrazuje chybové situácie a stavy.



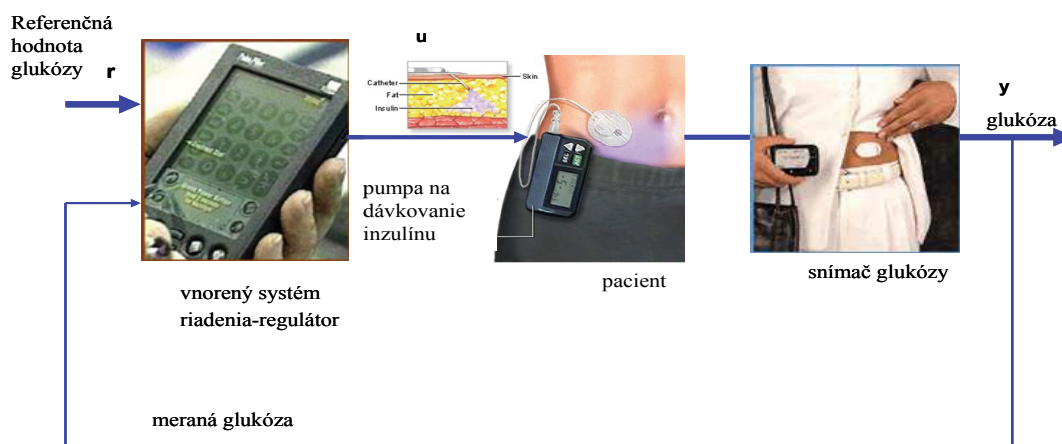
**Obr. 2.** Bloková schéma zabudovania vnoreného výpočtového systému do zariadenia.

Konvenčná štruktúra riadenia objektu s vnoreným počítačovým systémom je zobrazená na obr. 3. Cieľom riadenia je zabezpečiť, aby výstupná veličina procesu  $y(t)$  sledovala referenčnú premennú  $r(t)$  (poloha, otáčky, hladina, koncentrácia a pod.), ktorá reprezentuje požadované správanie sa procesu. V prípade odchýlky medzi výstupnou veličinou a referenčnou premennou vnorený počítačový systém na základe algoritmu riadenia vypočítava a generuje riadiaci zásah  $u(t)$  tak, aby sa výstupná veličina  $y(t)$  rovnala referenčnej premennej  $r(t)$ , čím je dosiahnutý cieľ riadenia. Konvenčné metódy a algoritmy riadenia sú zväčša realizované na znalosti matematických modelov založených na opise procesov prostredníctvom diferenciálnych, diferenčných rovníc alebo na opisoch prostredníctvom prenosových funkcií pomocou Laplaceovej a Z-transformácie. Tieto modely sa dajú získať buď matematicko-fyzikálnou analýzou procesu, alebo identifikačnými algoritmi umožňujúcimi určiť zo vstupno/výstupných meraní tak vhodnú matematickú štruktúru – matematický model, ako aj realizovať odhad jeho parametrov.



**Obr. 3.** Konvenčný spätnoväzbový obvod aplikácie vnoreného počítačového systému.

Vstupom do vnoreného počítačového systému sú informácie o odchýlke, resp. o ciele riadenia. Úplná štruktúra riadenia je zobrazená na obr. 9 a obr. 10, pričom výstup ako fyzikálna veličina je snímaný a transformovaný na elektrický signál prostredníctvom A/D (analogóvo-digitálneho) prevodníka, na výstupe z vnoreného počítačového systému sa opäť číslo transformuje na signál prostredníctvom D/A prevodníka, ktorý vstupuje do aktuátora (výkonového člena) na ovládanie objektu riadenia. Typická aplikácia vnoreného systému pre proces v zdravotníctve je na obr. 4.



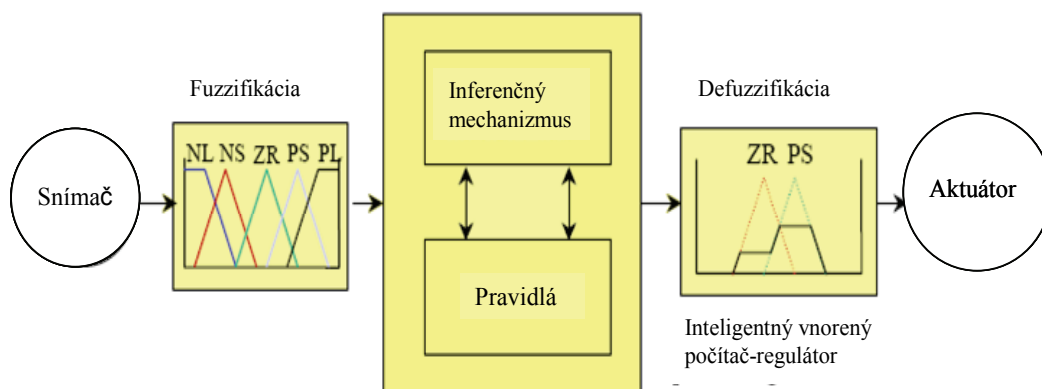
**Obr. 4.** Príklad použitia vnoreného systému pre vybraný proces v zdravotníctve.

Inteligentné algoritmy riadenia realizované na základe princípov fuzzy logiky, neurónových sietí a genetických algoritmov využívajú pre návrh algoritmu riadenia expertné analýzy, neurónové modely a na základe nich sú navrhované lingvistické a neurónové algoritmy riadenia často jednoduchšie a s lepšou kvalitou, ako v prípade opisu procesu diferenciálnymi, resp. diferenčnými rovnicami. Aplikácie vnorených počítačových systémov na báze kombinovaných hybridných metód riadenia (fuzzy logiky, neurónové siete) môže signifikantne zlepšiť kvalitu, spoľahlivosť a bezpečnosť prevádzok, systémov a zariadení. Inteligentné systémy riadenia sú rozšírením konvenčných metód riadenia. Základné štruktúry inteligentných metód riadenia sú zobrazené na obr. 5 a na obr. 6 - obr. 10.

Správna voľba vstupných a výstupných premenných fuzzy systému je základným predpokladom jeho správnej činnosti. Pre návrh inteligentných vnorených regulátorov, je dopredu dané, aké vstupné, resp. výstupné veličiny budú vo fuzzy riadiacich pravidlách vystupovať. Veľmi často sa však fuzzy vnorené regulátory navrhujú na riadenie systémov, ktoré boli doteraz riadené len ľudskými operátormi, na základe nimi sformulovaných pravidiel. Podobne sa modelujú aj systémy, ktorých správanie je možné opísať len na základe výpovedí expertov. V takýchto prípadoch je najprv potrebné vykonať analýzu týchto lingvistických pravidiel a určiť, ktoré z v nich vystupujúcich premenných nesú relevantné informácie.

Fuzzy vnorený regulátor (obr. 5) je vo svojej podstate jednoduchý vnorený výpočtový člen, ktorý realizuje v porovnaní s konvenčnými algoritmi modelovania a riadenia relatívne malé množstvo aritmetických operácií. Problémom môže byť napr. defuzzifikácia (numerický výpočet integrálu), ale táto sa často nahradzuje jednoduchými numerickými algoritmi pre výpočet integrálu.

Fuzzy inteligentné vnorené regulátory sú založené na expertných skúsenostiach. Riadiaci zákon je vyjadrený pravidlami odvodenými na základe pravidiel (kap. 2).



**Obr. 5.** Bloková schéma vnoreného fuzzy regulátora.

Cieľom operácie s názvom fuzzifikácia je priradiť ku vstupnej jazykovej premennej (teplota, tlak, poloha, hladinu a pod.) pravdivostnú hodnotu jednej alebo niekoľkých elementárnych fuzzy premenných – vstupných jazykových premenných. Každý pojem je definovaný svojou pravdivostnou funkciou, ktorá k hodnotám vstupnej jazykovej premennej priraduje hodnoty pravdivosti tohto termu (stupeň pravdivosti).

Samotná činnosť fuzzy vnoreného regulátora je založená na odvodzovacích (inferenčných) pravidlách *IF-THEN*, podobne ako je tomu v expertných systémoch. Výhodou takejto reprezentácie vedomostí je prehľadnosť a ľahká "čitateľnosť" človekom. Báza znalostí fuzzy systému je rozdelená do dvoch častí: prvou sú už spomínané definície FM, druhou sú pravidlá pre odvodzovanie, ktoré sú zvyčajne uložené v tzv. look-up tabuľke.

Inferenčný mechanizmus je jadrom fuzzy vnoreného logického riadenia. Má schopnosť simulácie ľudského spôsobu rozhodovania, založenom na fuzzy koncepcii.

Úlohou fuzzy inferenčného systému je určenie výstupu fuzzy systému pre daný vstup, keď závislosť I/O je definovaná pomocou bázy pravidiel. Existuje viacero prístupov, ale najčastejšie používanou a najjednoduchšou metódou je najprv určenie výstupu každého individuálneho fuzzy pravidla a celkový výstup potom vyjadrený kompozíciou pravidiel, napr. ako zjednotenie takýchto čiastkových výstupov

Blok defuzzifikácie transformuje výstupnú jazykovú hodnotu, ktorá vznikla kompozíciou výstupov z jednotlivých pravidiel, na „ostrú“ hodnotu, ktorá vstupuje do riadeného procesu v podobe riadiaceho zásahu cez aktuátor.

Inteligentné vnorené systémy riadenia založené na metódach umelých neurónových sietí predstavujú výrazný prínos do kvality riadenia. Masívne paralelná architektúra umelých neurónových sietí (UNS) umožňuje vykonávať výpočty pre ovládanie činnosti zariadení vysokými rýchlosťami. Pretože UNS môžu aproximovať chovanie nelineárnych systémov s vysokou presnosťou, ako aj vzhľadom na svoju univerzálnosť, sa neurónové siete javia ako veľmi silný nástroj pre návrh inteligentných algoritmov riadenia aj pre vnorené aplikácie.

Pre aplikáciu vnorených algoritmov riadenia je potrebné vytvoriť neurónový regulátor s minimálnou a optimálnou štruktúrou už vo vývojovom prostredí tak, aby bolo možné proces ovládať v reálnom čase. Vlastný vnorený algoritmus riadenia môže byť realizovaný dvoma základnými spôsobmi:

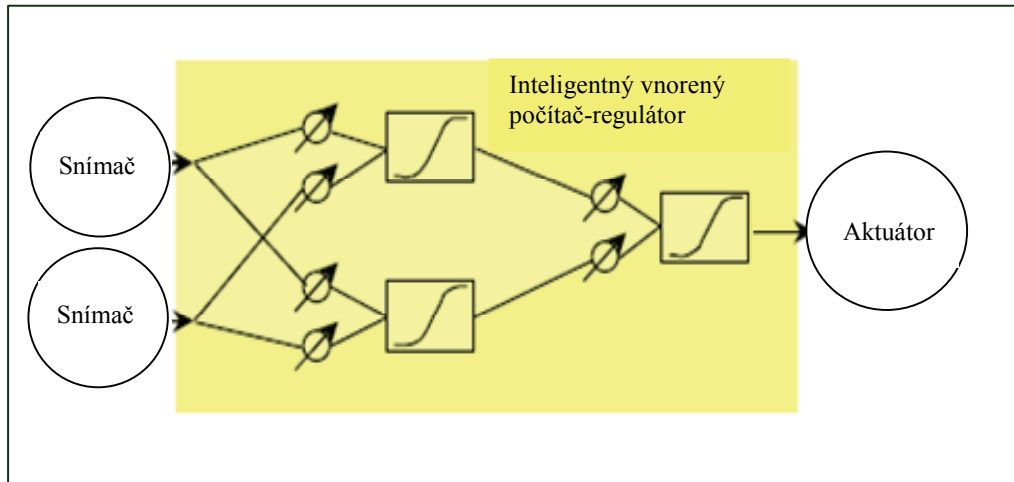
prostredníctvom matematického modelu procesu (nepriamy spôsob), ktorý je realizovaný neurónovou sieťou a je vytvorený na základe reálnych procesných údajov,

na základe natrénovania neurónového regulátora pre danú aplikáciu z meraných reálnych procesných údajov. Vnorený algoritmus je získaný na základe neurónového modelu zariadenia alebo systému meraním a tréňovaním realizovaným zväčša off-line spôsobom. Pre praktickú aplikáciu je možné využiť tak hardvérové riešenia realizácie umelých neurónových sietí na báze FPGA, ako aj softvérové riešenia prostredníctvom algoritmov riadenia, ktoré sú vygenerované vo forme kódu pre príslušný typ vnoreného procesora.

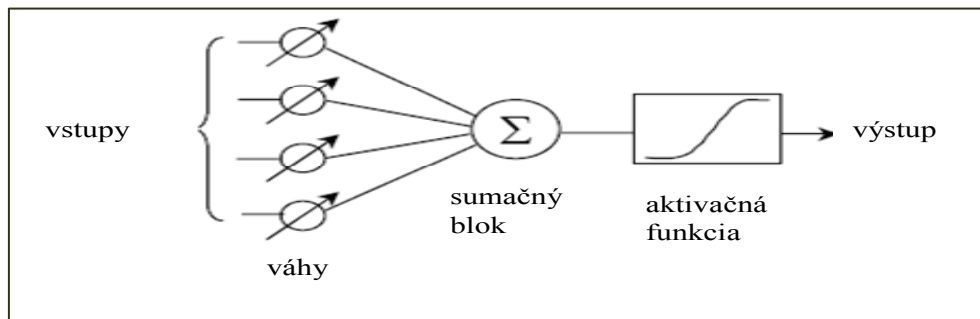
Štruktúra riadenia s použitým vnoreným regulátorom realizovaným pomocou umelých neurónových sietí je podobná ako pri konvenčnom spätnoväzbovom riadení (obr. 6). Natrénovaný vnorený algoritmus riadenia je implementovaný do príslušného vnoreného procesora a vykonáva svoju činnosť pre všetky natrénované situácie a stavy.

Inteligentné vnorené riadiace systémy na báze umelých neurónových sietí sú navrhované k zlepšeniu kvality činnosti zariadenia a integrity na širokom rozsahu pracovných podmienok. Z uvedených dôvodov umelé neurónové siete zohľadňujú chovanie sa zložitých systémov a sú schopné do svojho matematického opisu zahrnúť komplexné vlastnosti riadeného objektu, vrátane nelineárneho a časovo premenlivého správania, jeho dynamiky, premenlivých charakteristík viacerých činností a premenných, komplexnosti požadovaného cieľa, nepresností a neurčitostí v meraniach a pod.. Inteligentné vnorené systémy riadenia majú ďalšiu požadovanú vlastnosť – autonómnosť. Autonómny inteligentný vnorený riadiaci systém je taký, ktorý je schopný dosiahnuť vysoký stupeň inteligencie v nezávislosti od prostredia, ktoré ovláda a riadi.

Z existujúcich prístupov k inteligentnému riadeniu sa neurónové siete stávajú čoraz populárnejšie aj v oblasti vnorených systémov vzhľadom na fakt, že sú potenciálne schopné vysporiadať sa s niektorými ťažkosťami, ktoré sa vyskytujú v prípade riadenia s konvenčnými riadiacimi prístupmi, menovite: výpočtová komplexnosť, prítomnosť nelinearit a neurčitost'. Základná štruktúra riadenia s vloženým inteligentným regulátorom je na obr. 6.



**Obr. 6.** Principiálna schéma vnoreného regulátora na báze umelých neurónových sietí.

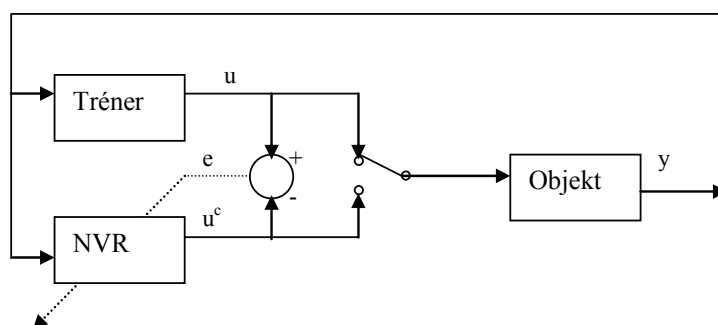


**Obr. 7.** Neurónová sieť pre modelovanie a riadenie.

Použitie neurónových sietí pre vnorené algoritmy riadenia môže byť realizované pre rôzne typy štruktúr riadenia a rôzne typy umelých neurónových sietí (viacvrstvový perceptrón, Hopfieldove siete, Kohonenové siete, ART siete a iné).

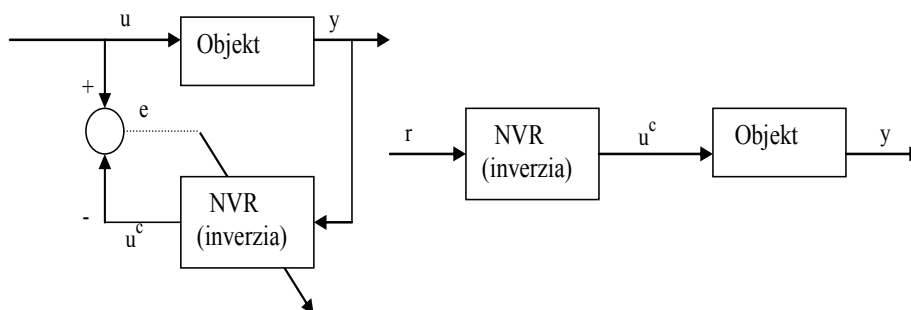
Jednu z najvhodnejších aplikácií pre vnorené systémy predstavuje supervízorové vnorené riadenie (s dohliadaním/vedením). Neurónový vnorený regulátor (NVR) (obr. 8) je vytvorený na základe tréningu paralelne s trénerom. Neurónová sieť generuje výstup použitím rovnakých vstupov, aké používa človek, a aproximuje činnosť človeka, ktorý generuje riadiaci vstup do objektu. Rozdiel medzi riadiacim

zásahom človeka a neurónového regulátora je potom použitý na adaptáciu parametrov NVR. Po dosiahnutí chyby tréningu s danou presnosťou je potom tréningový proces ukončený a NVR môže nahradiť človeka, ako vnorený neurónový regulátor v riadiacom procese pri daných prevádzkových podmienkach.



**Obr. 8.** Schéma natrénovania neurónového riadenia pre vnorené aplikácie.

Iná štruktúra inteligentného vnoreného systému môže byť realizovaná prostredníctvom tzv. inverzného modelu objektu riadenia (obr. 9). Obvykle je takéto riadenie rozdelené na priame a nepriame podľa spôsobu, ktorým sú nastavované parametre vnoreného regulátora. V priamom riadení sú parametre vnoreného regulátora priamo nastavované pri minimalizácii niektorej normy výstupnej odchýlky (rozdiel medzi realizovaným riadením a riadením z vnoreného regulátora).



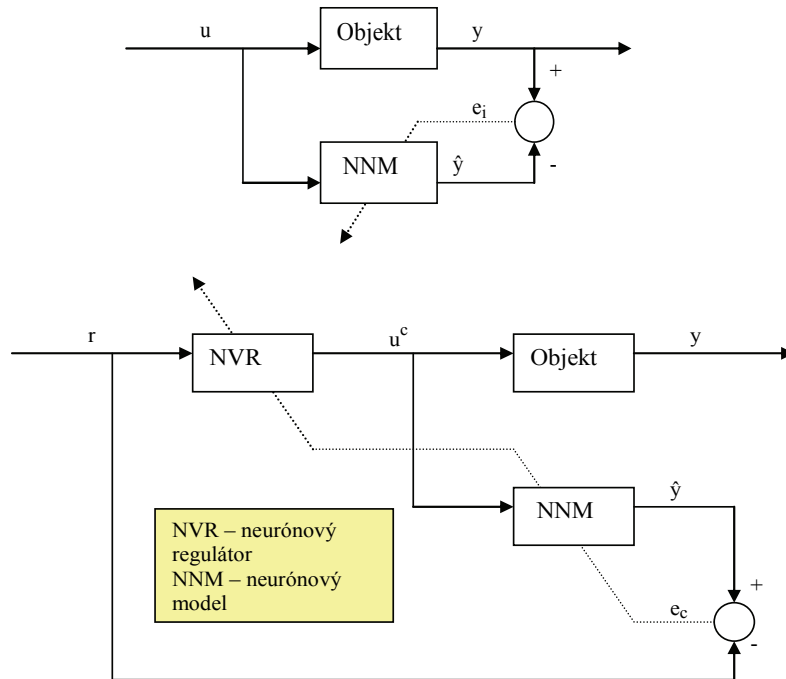
**Obr. 9.** Schéma tréningu vnoreného regulátora a jeho použitie, ako inverzného člena v kaskádovom zapojení.

Pri nepriamych algoritmoch riadenia (obr. 10) je najprv identifikovaný neurónový model objektu, a jeho parametre až v druhej etape prostredníctvom známeho neurónového modelu je tréningom vytvorený algoritmus riadenia a ovládania objektu, ktorý sa potom použije do daného zariadenia, ktoré má riadiť.

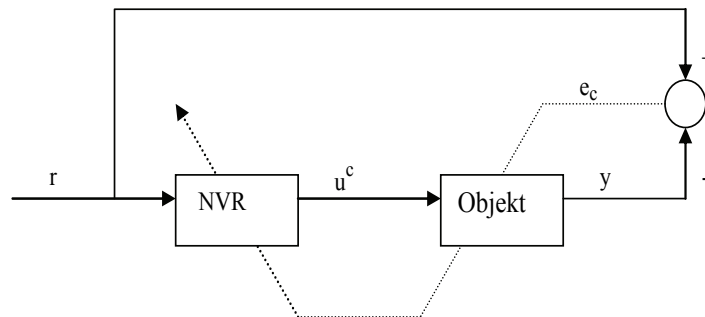
Priame ovládanie a riadenie procesu pomocou vnoreného neurónového regulátora je založené na natrénovaní neurónového regulátora na základe znalosti referenčného



signálu a výstupu z objektu riadenia. Rozdiel medzi reálnym výstupom a referenčnou (norma odchýlky) premennou je použitý na tréning vnoreného neurónového regulátora (obr. 11, obr. 12).



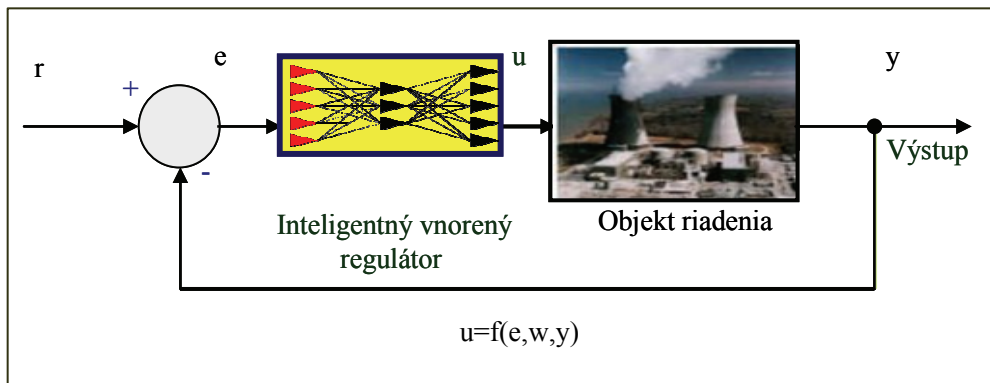
Obr. 10. Dvojetapový spôsob vývoja vnoreného algoritmu riadenia na základe NNM modelu.



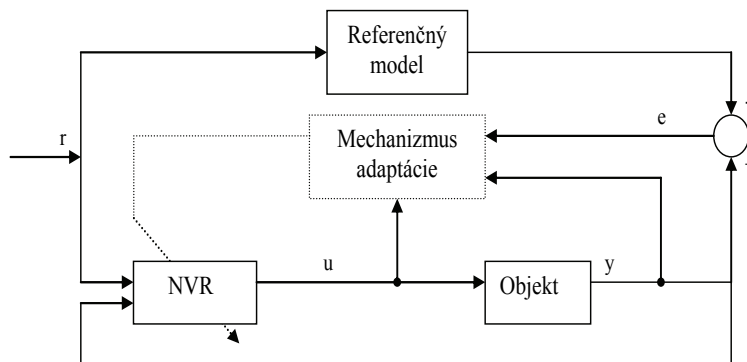
Obr. 11. Štruktúra inteligentného vnoreného regulátora na báze UNS.

Pre mnohé praktické problémy riadenia procesov pomocou vnorených počítačových systémov je možné využiť aj metodiku adaptívneho referenčného riadenia (obr.12), ktorej cieľom je sledovať referenčnú trajektóriu (referenčné

chovanie) pomocou zákona adaptácie parametrov riadiaceho algoritmu tak, aby výstupná veličina sledovala referenčnú premennú prostredníctvom referenčného modelu. Pri aplikácii tejto metodiky pre návrh inteligentného riadenia sa parametre vnoreného regulátora nastavujú na základe tréningu, kde vstupom budú hodnoty výstupu referenčného modelu, reálneho objektu a ich rozdiel. Riešením je inteligentný vnorený neurónový regulátor (algoritmus riadenia) s možnosťou implementácie pre vnorený typ procesora.



**Obr. 12.** Schéma zapojenia vnoreného inteligentného vnoreného systému riadenia.



**Obr. 13.** Bloková schéma adaptívneho neurónového regulátora pre vnorené aplikácie.

Existuje ešte mnoho ďalších verzií inteligentných systémov riadenia vhodných zväčša pre väčšie aplikácie. Tieto moderné inteligentné algoritmy sú efektívne predovšetkým pre úlohy optimálneho riadenia, prediktívneho riadenia a pre hybridné algoritmy s kombinovanou spojitou a diskretnou dynamikou. Hybridné inteligentné systémy sú ďalším vhodným kandidátom pre aplikáciu kombinovaných metód výpočtovej inteligencie (fuzzy-neuro, fuzzy-genetic, neuro-genetic a pod.). Pri modelovaní a riadení objektov základnú štruktúru fuzzy modelu tvoria vstupné a výstupné dáta, funkcie príslušnosti a báza pravidiel. Pri klasických fuzzy modeloch je výber funkcií príslušnosti ľubovoľný a báza pravidiel vytvorená na základe istých

expertných skúseností užívateľa. Funkcia príslušnosti závisí od jej parametrov a zmenou týchto parametrov meníme aj tvar funkcie príslušnosti, a tým aj vlastnosti celého fuzzy modelu. V prípade, ak máme súbor nameraných vstupných a výstupných dát procesu (systému) a chceme ho namodelovať, nie je nevyhnutné vopred poznať a nastaviť štruktúru fuzzy modelu, ale pomocou adaptívneho systému (ANFIS) môžeme navrhnúť taký fuzzy inferenčný systém, ktorého parametre funkcií príslušnosti sa budú ladiť, napr. pomocou metódy spätného šírenia chýb (backpropagation) a metódou najmenších štvorcov tak, aby daný fuzzy model mal optimálne modelovacie a riadiace vlastnosti.

### 1.3 Hybridné inteligentné vnorené algoritmy riadenia

Základné charakteristiky hybridných fuzzy-neurónových systémov vhodných pre vnorené aplikácie:

- Hybridné fuzzy-neurónové systémy (FNS) umožňujú zlepšenie vlastností a schopností existujúcich fuzzy systémov (FS) a umelých neurónových sietí (UNS),

- FS + UNS hybridné systémy (HS) (integrácia dobrých vlastností),

- FS zakomponovanie expertných znalostí o modelovanom, resp. riadenom systéme (v samotných rozhodovacích pravidlách, ale aj pri výbere jeho vstupných a výstupných premenných, tvare a spôsobe pokrytia univerzít lingvistických premenných fuzzy množinami atď.), v klasickom fuzzy systéme nemôžu byť zohľadnené existujúce reálne dáta,

UNS - vlastnosť tréningu a aproximácia ľubovoľnej funkcie, nevýhoda - návrh štruktúry intuitívne, (bez možnosti zabudovania existujúcich znalostí o modelovanom systéme do ich štruktúry),

HS – štruktúrovaná sieť - *fuzzy-neurónový systém*, ktorého parametre sa získajú, resp. doladia, pomocou niektorej metódy učenia,

Metódy učenia bez učiteľa v HS môžu byť použité na analýzu vstupných dát, určenie počtu fuzzy množín, pokrývajúcich univerzá vstupných premenných, nájdenie bázy pravidiel,

Metódy učenia s učiteľom v HS slúžia na minimalizáciu výstupnej chyby systému pri aproximácii reálnych dát. Fuzzy-neurónové systémy tak minimalizujú použitie intuície a navrhovateľových zásahov pri ich návrhu.

#### *Neuralizované fuzzy systémy (NFS)*

Hlavná myšlienka pri aplikácii NFS je založená na transformácii štruktúr fuzzy systému do zodpovedajúcej štruktúrovanej viacvrstvovej neurónovej siete, ktorej každá vrstva vykonáva niektorú z operácií fuzzy systému (fuzziifikácie, fuzzy inferenčný mechanizmus, defuzziifikáciu). Jednotlivé vrstvy takejto siete sa odlišujú typom použitých neurónov, ako aj spôsobom ich prepojenia. Niekedy sa tieto typy sietí označujú ako *adaptívne siete* a ich neuróny ako uzly siete.

K týmto typom fuzzy-neurónových systémov patria dnes už komerčne využívané ANFIS (Adaptive Neuro Fuzzy Inference System, alebo tiež Adaptive Network-based

Fuzzy Inference System), MANFIS (Multiple ANFIS), CANFIS (Coactive ANFIS) s rôznymi modifikáciami.

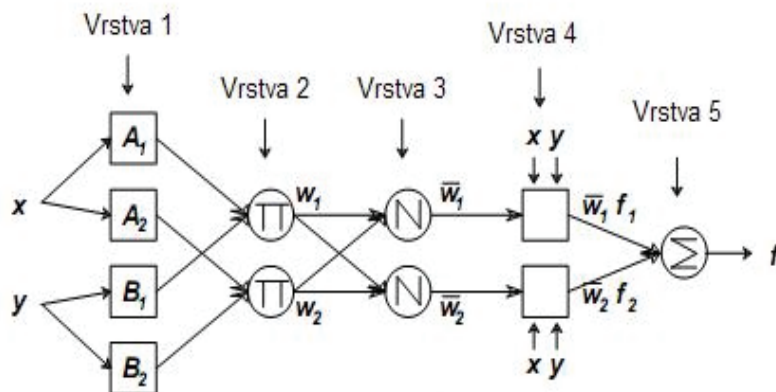
Návrh parametrov a trénovanie NFS prebieha v dvoch krokoch. V prvom kroku návrhu sa môže uplatniť maximálne množstvo expertných znalostí o riadenom systéme a výsledkom je spomínaná štruktúrovaná neurónová sieť. Existujú metódy automatického návrhu štruktúry fuzzy-neurónového systému, niektoré budú opísané v nasledujúcich častiach.

V druhom kroku sa parametre fuzzy-neurónového systému doladia pomocou učenia sa na vzorkách reálnych dát niektorou metódou učenia s učiteľom (BP).

Medzi najmodernejšie spôsoby inteligentného riadenia však patria adaptívne fuzzy-neurónové systémy, ktoré pri návrhu nevyžadujú žiadne apriórne informácie a ktorých parametre, ako počet a tvar funkcií príslušností, resp. rozhodovacie pravidlá, sa určia až pri tréovaní hrubo navrhutej neurónovej siete.

Využitie expertných znalostí v prvom kroku návrhu môže znížiť čas potrebný na dostatočne presné natrénovanie takéhoto systému, resp. zmenšiť výslednú chybu výstupu systému.

ANFIS (Adaptive Neuro-Fuzzy Inference System) bol prvý krát predstavený J.-S. Roger Jangom. V podstate je to je päťvrstvová dopredná neurónová sieť, ktorej výsledkom je fuzzy inferenčný systém typu Takagi-Sugeno (obr. 14):



**Obr. 14.** Architektúra siete ANFIS.

*Vrstvu 1* tvoria adaptívne uzly, každý uzol predstavuje lingvistickú hodnotu vstupnej premennej.

*Vrstvu 2* tvoria neadaptívne uzly, ktoré iba násobia prichádzajúce signály a na výstupe tak poskytujú váhu  $w$  pravidiel, ktorých antecedent je daný kombináciou lingvistických hodnôt jednotlivých premenných.

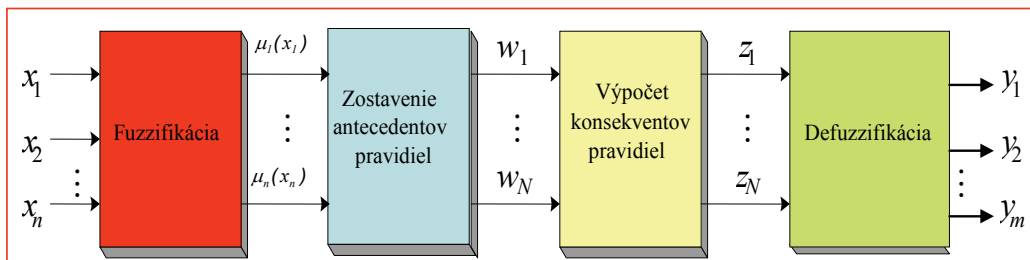
*Vrstvu 3* tvoria neadaptívne uzly, ktoré poskytujú na výstupe pomer váhy jednotlivých pravidiel a súčtu váh všetkých pravidiel  $w$ .

Vrstvu 4 tvoria adaptívne uzly, ktorých prenosové funkcie sú dané požadovaným tvarom konsekventu. Môžu byť konštantné alebo lineárne.

Vrstvu 5 tvorí jeden neadaptívny uzol, ktorý počíta celkový výstup ako súčet všetkých vstupných signálov.

#### Fuzzifikované neurónové siete (FNS)

Pod FNS (obr. 15) rozumieme zabudovanie fuzzy postupov a metodiky do neurónovej siete, jednotlivé operátory medzi premennými v uzloch siete, ako sumácia alebo násobenie, sú nahradené fuzzy logickými operátormi a samotná neurónová sieť reprezentuje bázu pravidiel fuzzy systému.



Obr. 15. Štruktúra všeobecného fuzzy-neurónového systému.

Na výpočet výstupu takéhoto hybridného systému sa použijú princípy fuzzy logiky a uvažovania. Jednotlivé parametre systému je opäť možné doladiť pomocou metód učenia. Umelé neurónové siete a fuzzy systémy sú univerzálne aproximátory funkcií, a preto ním bude aj výsledný hybridný fuzzy-neurónový systém. Vlastnosti FNS:

Vstupné veličiny sú distribuované do neurónov 1. vrstvy (fuzzifikácia), t.j. vypočítavajú sa stupne príslušnosti ostrej hodnoty vstupnej premennej k nejakej jej lingvistickej hodnote, reprezentovanej fuzzy množinou.

Operácia výpočtu hodnota konsekventu je závislá na type použitého fuzzy inferenčného mechanizmu.

Posledná vrstva siete slúži na implementáciu defuzzifikačného algoritmu, ktorého výstupom je výsledná hodnota príslušnej výstupnej veličiny.

### 1.4 Praktická realizácia a vývoj inteligentných vnorených systémov

Praktická realizácia a vývoj vnorených systémov vychádza z nasledovných základných prístupov:

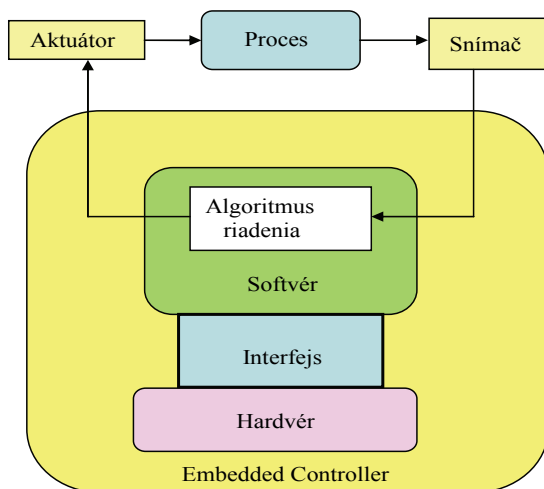
Návrh a vývoj vnoreného systému, ako jednoúčelového hardvérového systému. Hlavnou výhodou takéhoto riešenia je pomerne veľká výpočtová rýchlosť, malé rozmery, malá energetická spotreba a aj nízka cena. Takéto jednoúčelové vnorené systémy sú bez pamäte, využívajú zákaznicke obvody s vysokým stupňom integrácie, pričom realizujú zväčša jediný jednoúčelový program.

Návrh a využitie špeciálnych procesorov – ASIP (Application Specific Processor). Tieto dedikované procesory so špeciálnymi funkčnými výpočtovými funkciami majú

optimalizovaný návrh štruktúry pre riešenie definovanej triedy úloh (spracovanie signálov, filtrácia, riadenie a pod.). Hlavnou výhodou využitia a nasadenia dedikovaných vnorených systémov je pružnosť, otvorenosť, možnosť modifikácie algoritmov, vysoká výkonnosť, malé rozmery a malá spotreba energie. Špeciálnym príkladom ASIP procesora je napr. DSP (Digital Signal Processor), ktorý obsahuje špeciálne funkčné jednotky realizované hardvérovým spôsobom (násobičky, podávače a pod.) na číslicovú filtráciu a riadenie a riešenie zložitých numerických úloh (maticová algebra).

Návrh a využitie konvenčných mikroprocesorov ako vnorených počítačových systémov pre danú aplikáciu alebo ako mikroradiča je zobrazený na (obr. 16). Mikroprocesory pre všeobecné použitie majú pamäť programu, nemajú však optimalizovanú architektúru na riešenie konkrétnych úloh. Zväčša disponujú veľkým počtom registrov, aritmeticko-logickou jednotkou (Arithmetic-Logic Unit - ALU). Ich prednosťou je flexibilita, rýchly vývoj aplikácie a nízka cena.

Optimalizácia architektúry sa rieši integráciou daného mikroprocesora a periférnych obvodov na jeden čip, čím vznikne obvod nazývaný SOC (System on Chip) alebo mikroradič (jednočipový mikropočítač), ktorý obsahuje všetky funkcie ako konvenčný PC a potrebné prvky a obvody (obvody V/V). Mikroradič je taktiež realizovaný integráciou mikroprocesora a periférnych zariadení spoločne na jeden čip. Tieto periférie sú však rozdielne od periférií obvodov SOC. Zabezpečujú predovšetkým styk samotného vnoreného systému s okolím (snímače, výkonové členy a pod.) a slúžia na zabezpečenie ďalších funkcií.

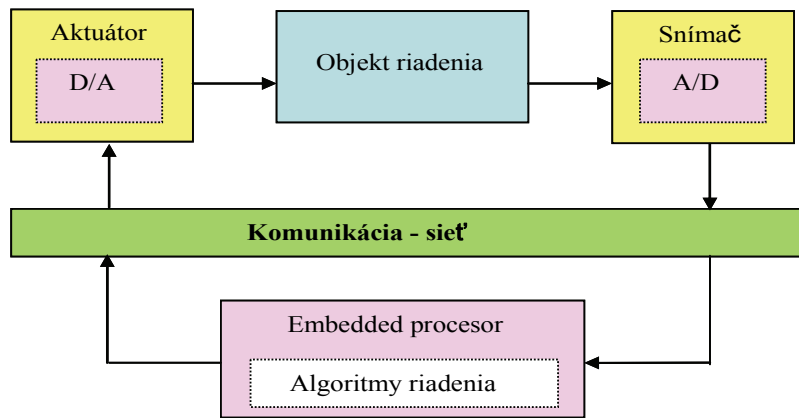


**Obr. 16.** Bloková schéma prepojenia SW a HW pre aplikáciu vnoreného systému do riadenia.

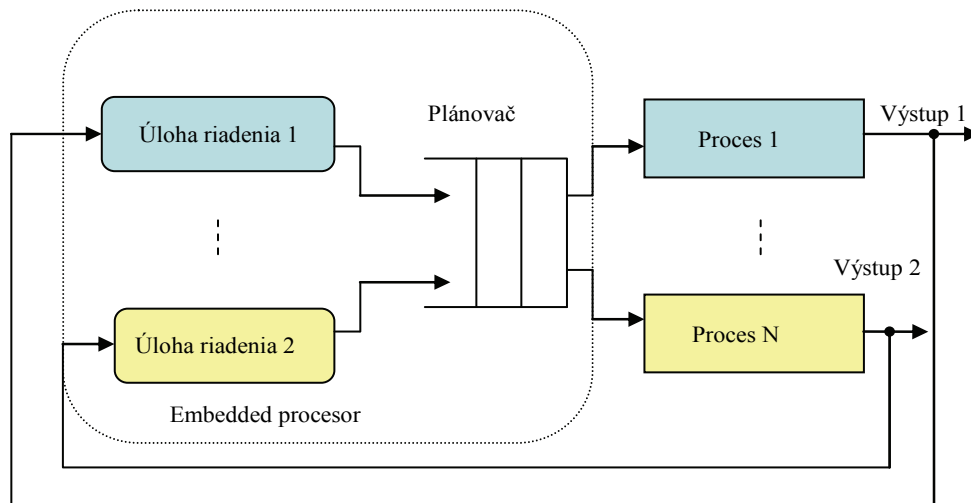
Aplikácia vnorených systémov je predovšetkým smerovaná do procesov, ktoré vnorený počítačový systém ovláda, riadi. V mnohých aplikáciách je vnorený systém súčasťou distribuovanej sieťovej štruktúry (obr. 17).

Na komplexnú riadiacu a komunikačnú činnosť sieťového vnoreného systému je potrebné poznať dynamické vlastnosti procesov, snímačov, prevodníkov, ako aj vlastnosti riadených procesov, algoritmov riadenia a znalosť aplikovania zbernice (napr. CAN, MilCan a pod.).

Jednou z ďalších aplikačných možností vnoreného systému je jeho využitie vo viacúlohovom režime riadenia (obr. 18), pri ktorom je aktivácia vnoreného systému realizovaná na základe definovaných časových režimov a sledovaná plánovačom úloh.



Obr. 17. Zapojenie vnoreného systému ako sieťového riadiaceho systému.



Obr. 18. Zapojenie vnoreného systému pri viacúlohovej (multitasking) štruktúre riadenia.

### **1.5 Analýza softvérových systémov a prostriedkov na zber a spracovanie údajov, komunikáciu, riadenie v reálnom čase a koordináciu činnosti vnorených modulov zabudovaných systémov**

V súčasnosti počet vyrobených a aplikovaných vnorených procesorov výrazne prevyšuje počet procesorov inštalovaných do osobných počítačov. Odhaduje sa, že až 98 % z celkového počtu vyrobených procesorov je inštalovaných vo vnorených systémoch a len 2% je inštalovaných v osobných počítačoch a v počítačových systémoch na riadenie.

Na praktické využitie sa dnes používa celá škála vnorených procesorov aplikovaných v spotrebnej elektronike, automobilovom priemysle, robotike a pod. (od jednoduchých štvorbitových, osembitových, šestnásťbitových, tridsaťdvabitových až po najvýkonnejšie 128 bitové a špeciálne signálové a sieťové procesory).

### **1.6 Platformy pre vnorené systémy:**

Vnorené systémy na báze procesorov určených pre osobné počítače (PC). Vývoj vnorených systémov v tejto skupine je realizovaný pomocou nástrojov používaných na platforme PC. Tato skupina vnorených systémov je označovaná ako Embedded PC (jednodoskové počítače SBC - Single Board Computer).

Druhou početnou skupinou využívajúcou pre vývoj vnorených systémov je platforma realizovaná na báze obvodov FPGA (Field Programmable Gate Array). Na základe tejto platformy sa navrhne zákaznícky obvod, ktorý nahradí štandardný procesor a ďalšie pomocné obvody. Obvody FPGA sa najčastejšie používajú pri návrhu pre vnorené systémy s veľkým výpočtovým výkonom, pretože táto architektúra umožňuje značný paralelizmus výpočtov. Ďalšou výhodou tejto platformy je robustnosť a odolnosť softvéru a možnosť použitia kryptografických algoritmov. Najrozšírenejšie v aplikáciách sú obvody FPGA od firmy Xilinx, Altera, Lattice Semiconductor, Actel, Atmel, QuickLogic, Achronix Semiconductor a MathStar. Veľká časť vnorených inteligentných algoritmov riadenia na báze fuzzy logiky a umelých neurónových sietí využíva pre praktické aplikácie obvody FPGA

Treťou najviac využívanou platformou je návrh vnoreného systému na jednom čipe – ASIC. Aplikácie sú predovšetkým v automobilovom priemysle pri meraní, riadení a komunikácii (CAN, Milan, LIN). Túto platformu ponúka dnes mnoho firiem, napr. Freescale (procesory HCO8RF2, 68HC908RF2, MPXY8020A), Atmel (ATA5756 a 5757), National Semiconductor a pod.

Jednoduché vnorené systémy zväčša pracujú bez operačného systému a programy sú vytvárané v asembleri a uložené sú v pamäti ROM, zložitejšie vnorené systémy vyžadujú operačný systém reálneho času s viacvláknovými procesmi naprogramovanými vo vyššom programovacom jazyku (C,C++, Java a pod.). Softvérové systémy pre vnorené systémy sú pre mnohé aplikácie rovnaké ako pri konvenčných počítačových systémoch.

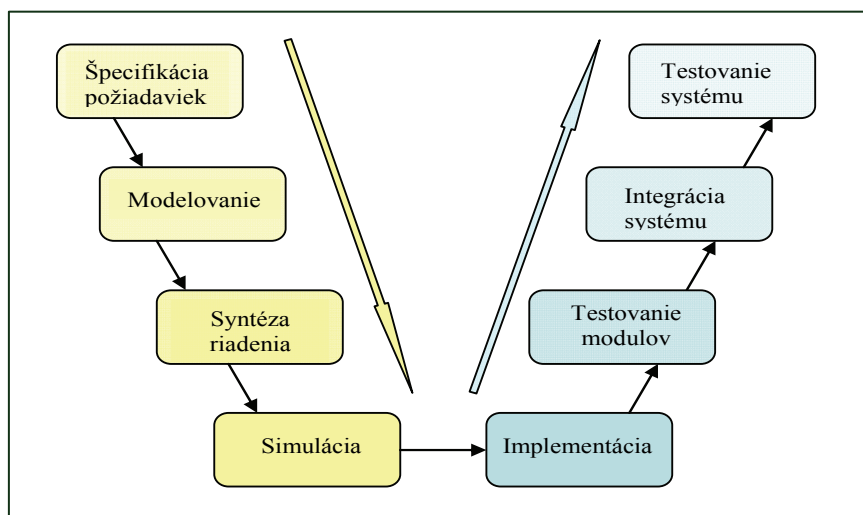


**Tabuľka 1.** Prehľad najviac používaných procesorov vo vnorených aplikáciách

Výrobcovia procesorov	Typy a označenia procesorov
ARM	ARM 7, 9, 10, 11
Atmel	MARC4, 8051, AVR 8, 32, AT 91
Analog Devices	Blackfin, SHARK, TigerSHARK, ADSP21xx
Freescale	HC(S) 08, 11, 12, 16, 5xx, 51xx, ARM, MMC2xxx, MAC7xxx, ColdFire, PowerQUICC
Microchip	PIC10, 12, 16, 18, 24F, 24H, dsPIC30, dsPIC33
NEC	78KOS, 78KOR, V850, VR 4100, 5500, 7701
Texas Instruments	TMS370, MSP430, TMS470, C2000
Toshiba	T4X, TLCS870, TLCS900, TX19,39, TX49,99

Zabudovaný softvér musí však byť veľmi spoľahlivý a odolný voči poruchám (fault tolerant), pretože samotná aplikácia musí byť zväčša v nepretržitej činnosti a nie je ju možné vzhľadom na bezpečnosť prevádzky (automobily, výťahy, navigačné systémy, chemické procesy, lekárska operácie, diagnostika a riadenie jadrových reaktorov) ľubovoľne zastaviť.

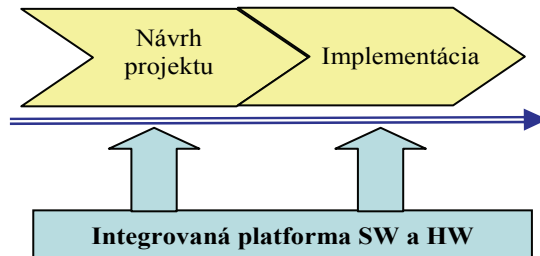
Vývoj softvéru pre vnorené systémy prebieha súčasne s vývojom hardvéru (co-design). Pre jednoduchšie aplikácie, ktoré predstavujú okolo jednej tretiny všetkých aplikácií, je vnorený systém vyvíjaný bez operačného systému reálneho času. Vývoj aplikačného softvéru je v súčasnosti realizovaný pomocou vývojových nástrojov (vývojových kitov), najmä vo vyššom programovacím jazyku C, C++ a Java. Podporné vývojové nástroje umožňujú zároveň vygenerovať strojový kód pre daný typ procesora, jeho odladenie a implementáciu (obr. 19).



**Obr. 19.** Tradičný postup návrhu softvéru pre vnorené systémy.

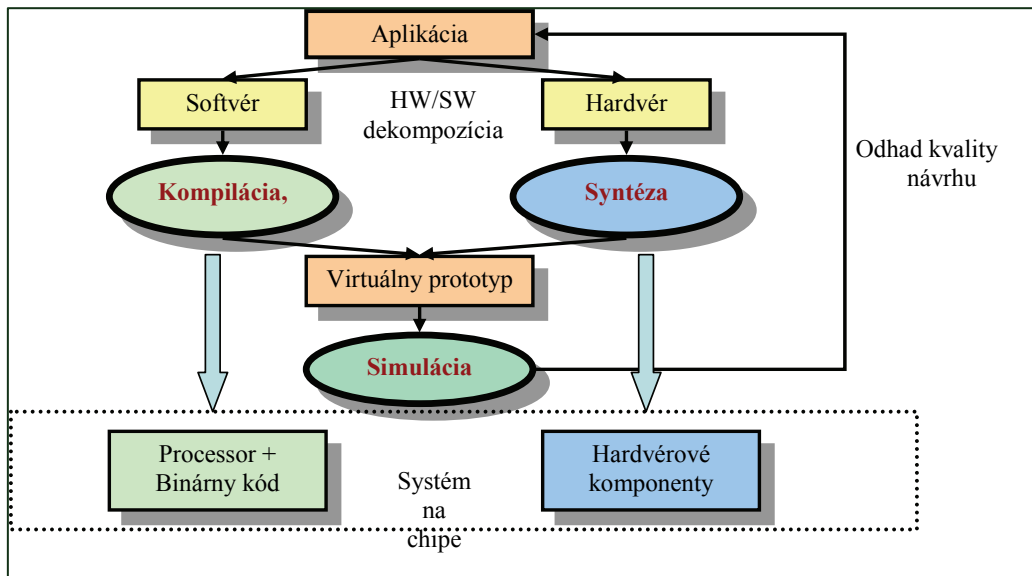
V poslednej dobe sa do stredu záujmu dostávajú nové generácie vývojových nástrojov, ktoré priamo generujú kód vo vyššom programovacom jazyku, čím sa urýchli vývoj softvéru pre konkrétnu aplikáciu. Programový systém Matlab-Simulink,

Labview, je jedným z efektívnych prostriedkov umožňujúcich pre vybrané typy platforiem a procesorov generovať kód zo schémy v Simulinku.



**Obr. 20.** Integrácia SW a HW pre vývoj aplikácie.

Vývoj softvéru pre vnorené systémy na báze FPGA a systémy na čipe využíva jazyk VHDL alebo podporné programové systémy dodávané a poskytované výrobcom. Principiálna schéma vývoja aplikácie vnoreného systému je na obr. 20 a obr. 21, kde je výrazne znázornená paralelnosť vývoja SW a HW, čím je zaručená optimálnosť návrhu vnoreného systému.



**Obr. 21.** Co - design návrhu SW a HW pre vnorené aplikácie.

Pre zložitejšie aplikácie je nutné využiť a implementovať operačné systémy reálneho času. K operačnému systému je zároveň poskytované vývojové prostredie pre vývoj používateľských programov na osobných počítačoch v prostredí Windows alebo Linux. Množstvo aplikácií, ktoré využívajú operačné systémy reálneho času (kritické

úlohy reálneho času) vo vnorených systémoch, je okolo 70%. Pre praktické aplikácie výrobcovia dnes ponúkajú viac ako 20 druhov OS (najznámejšie sú: EmbOS, Vision Embedded, Nimbe, OSEK/VDX, Tiny atď.).

**Tabuľka 2.** Prehľad najbežnejších vývojových prostredí pre vnorené aplikácie

Typ procesora	Vývojové prostredie
ATMEL	Crossware's 8051 Development Suite, LITE SUITE RKitL51
Freescall	CodeWarrior
Microchip	CCS PIC
Analog Devices	Visual DSP, EZ-Kit Lite

Vzhľadom k uvedeným požiadavkám na funkcionality, SW a HW vnorených systémov je často nutné vyvíjať vnorený softvér algoritmov spracovania údajov, riadenia, vizualizácie a komunikácie s použitím operačných systémov pracujúcich v reálnom čase (RTOS). Ako bolo uvedené v tab. 1 a tab. 2 pre vývoj vnorených aplikácií existujú dnes k dispozícii efektívne metódy a nástroje pre analýzu, špecifikáciu, návrh, verifikáciu a certifikáciu prevádzkovej spoľahlivosti. Tieto špecifické požiadavky sa uplatnia najmä u systémov s vnútorne prepojenou sieťou s vysokými požiadavkami na kritické procesy riadenia zdrojov, časových limitov a požiadavkami na minimálnu spotrebu energie. Jednou z nevyhnutných požiadaviek na činnosť vnorených systémov je aj bezchybná interakcia s okolím, čo je nevyhnutné pre dosiahnutie vysokej prevádzkovej spoľahlivosti.

## 1.7 Záver

Táto kapitola mala za cieľ vymedzenie základných pojmov, štruktúr a využitie vnorených systémov. Nasledujúca kapitola bližšie špecifikuje metodológiu a algoritmy integrovanej inteligentnej syntézy pre vnorené aplikácie. Na základe prezentovaného obsahu je možno vyčleniť niektoré špecifické vlastnosti a charakteristiky vnorených systémov.

Vnorené systémy majú nasledovné špecifické vlastnosti :

**autonómnosť:** realizované funkcie musia byť realizované automaticky bez ľudského zásahu, autonómnosť je viazaná na vysokú prevádzkovú spoľahlivosť a je vyžadovaná všade tam, kde ľudská reakcia bola príliš pomalá alebo nedostatočne predvídateľná (napr. navigácia lietadiel, vojenské systémy - bojové vozidlá, systém ABS u automobilov, mikrooperácie hlavy a pod.),

**kritickosť:** stupeň kritickosti závisí na vplyve a veľkosti odchýlok od normálneho chovania na spoľahlivé a bezpečné splnenie úloh,

**reaktivita:** vnorené systémy sú v stálom spojení s prostredím, do ktorého sú umiestnené, činnosť v reálnom čase môže byť často do určitej miery obmedzená, pretože spracovanie úloh je limitované rýchlosťou spracovania,

**inteligencia:** vzhľadom na to, že vnorené systémy pracujú autonómne a riešia jednu alebo aj viacero úloh, musia si často adekvátne vedieť poradiť so zmenou situácií na základe zabudovaných inteligentných algoritmov riadenia.

Zavedenie inteligentných prístupov do vnorených systémov prináša rad výhod, na niektoré z nich v ďalšej časti poukážeme. Aplikáciou inteligentných metód do vnorených aplikácií je možné dosiahnuť :

**Vysokú prevádzkovú spoľahlivosť:** pri vnorených aplikáciách a riadiacich činnostiach, ktoré sú často opakované a implementované predovšetkým do priemyselných oblastí a iných náročných aplikácií, je nutné zabezpečiť vysokú spoľahlivosť a chybovú odolnosť.

**Efektivitu a optimálnosť činnosti (systému, zariadenia):** ide tu o návrh takého inteligentného riešenia, ktoré zvýši efektivitu výroby, jeho spoľahlivosť a optimálnu prevádzku (minimalizácia nákladov, ergonomika a príjemné životné prostredie).

**Autonómnosť:** inteligentné vnorené riešenia plnia rovnakú funkciu ako klasické návrhy riadenia s tým, že svoju funkciu realizujú autonómne a zväčša bez interakcií s človekom.

**Modelovanie a simuláciu:** jedná sa tu o inteligentný co-design SW a HW, čím je znížená chybovosť celého návrhu a znižujú sa náklady a čas na projektovanie.

**Zníženie nákladov na údržbu:** vnorené inteligentné systémy vyžadujú minimálne nároky na údržbu a servis a sú schopné dlhodobo fungovať bez ľudskej obsluhy.

**Vysokú kvalitu a stabilitu riadenia:** implementácia inteligentných metód prispeje k zvýšeniu kvality riadenia, schopnosť adaptivity a rekonfigurácie systémov a štruktúr riadenia.

## 1.8 Literatúra ku kap. 1

- [1] QING, Li: Real-Time Concepts for Embedded Systems. CMP Books, 2003.
- [2] SROVNAL, V.: Operační systémy pro řízení v reálném čase. VŠB – Technická univerzita, Ostrava, 2003.
- [3] SRIDHAR, T.: Designing Embedded Communications Software. CMP Books, 2003.
- [4] SROVNAL, V.: Přehled operačních systémů reálného času. AT&P Journal, October 2005, Volume 12, Issue 10-12, pp. 51–53, pp. 64–65, pp. 56, ISSN 1335-2237.

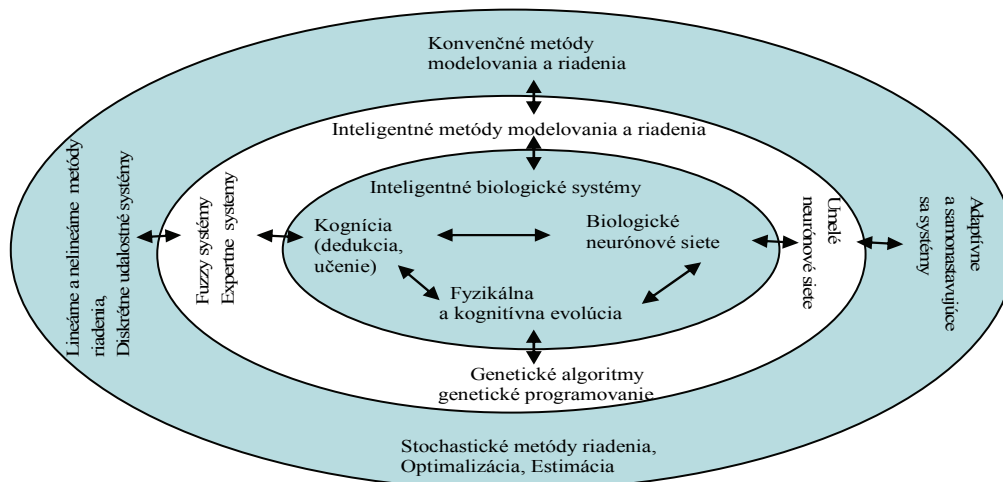
## **2 Analýza súčasného stavu využívania metód modelovania, simulácie a syntézy vnorených systémov použitím konvenčných a inteligentných techník, založených na fuzzy množinách, umelých neurónových sieťach a genetických algoritmoch**

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.

Návrh inteligentných vnorených systémov sa dnes nezaobíde bez využitia kombinovaných (hybridných) metodík výpočtovej techniky, ktoré poskytuje kognitívna veda alebo kognitivistika, teda veda o myslení, ktorá skúma a modeluje princípy organizácie a fungovania tak prirodzených, ako aj umelých intelektuálnych systémov. Pretože predmetom kognitívnej vedy sú fenomény súvisiace s myslením a využívaným expertných systémov, mnohé inteligentné algoritmy modelovania, riadenia a simulácie možno realizovať ako kombináciu formálnych techník informatiky s metódami psychológie a neurovedy. Prienikom poznatkov kognitívnej vedy do problematiky riadenia vnorených systémov sa získajú nové možnosti a schopnosti ako navrhovať, implementovať a využívať integrované inteligentné systémy a rozsiahle moderné aplikácie s vysokou mierou použitia metód umelej inteligencie a psychologickéj vedy.

**Inteligentné vnorené systémy (IVS)** môžeme definovať ako umelé integrované SW a HW systémy schopné riešiť vysoko komplexné úlohy modelovania riadenia a rozhodovania na základe napodobovania chovania sa svojich vzorov, ktorými sú živé organizmy - ľudia. Pri riešení úloh modelovania, riadenia a rozhodovania pomocou IVS využívame tak vedomostný znalostný ako aj modelový prístup potrebný k dosahovaniu a napodobovaniu schopnosti ľudí. Zavedenie inteligentných prístupov do vnorených systémov prináša rad výhod, z nich na niektoré v ďalšej časti poukážeme. Aplikáciou inteligentných metód do vnorených aplikácií je možné dosiahnuť vyššiu spoľahlivosť, optimalizáciu SW a HW na základe modelovania a simulácie, paralelného návrhu SW a HW, autonómnosť, vyššiu kvalitu a zníženie nákladov na prevádzku. Súvislosť metód výpočtovej inteligencie, ich interakcia a vzájomná previazanosť na časový a metodologický vývoj je zohľadnená a znázornená na obr. 22.



**Obr. 22.** Vývoj metód modelovania a riadenia pre aplikácie vo vnorených systémoch.

Podobne ako u inteligentných systémov konvenčného typu, tak aj u inteligentných vnorených systémov prínos ich zavedenia umožňuje efektívne riešiť nasledujúce úlohy:

interagovať prostredníctvom snímačov s okolím, vnímať a rozpoznať, čo je v ňom relevantné a dôležité pre riešenie daného problému,

predvídať zmeny v danom okolí na základe matematického modelu, simulácií a ďalších iných znalosti formulovaných najčastejšie v prirodzenom jazyku,

využívať získané vlastnosti o danom okolí a daných znalostí k logickému uvažovaniu o riešenom probléme a k nasledovnému návrhu optimálnych algoritmov pre rozhodovanie a riadenie,

plánovať spôsob realizácie vyžadovaných úloh na základe rozpoznávania situácie a predikovaných situácií s plným využitím daných znalostí pri návrhu SW algoritmov riadenia,

komunikovať a spolupracovať pri riešení zložitých úloh s inými vnorenými inteligentnými systémami, prirodzenými alebo umelými,

učiť sa na základe predchádzajúcich skúseností, pozorovaní a meraní a zovšeobecňovať poznatky pri vývoji SW,

adaptovať svoje chovanie podľa potreby a cieľov a implementovať ich do riadiacich algoritmov tak, aby zabezpečovali požadované funkcie a činnosti v požadovanom rozsahu, kvalite a presnosti.

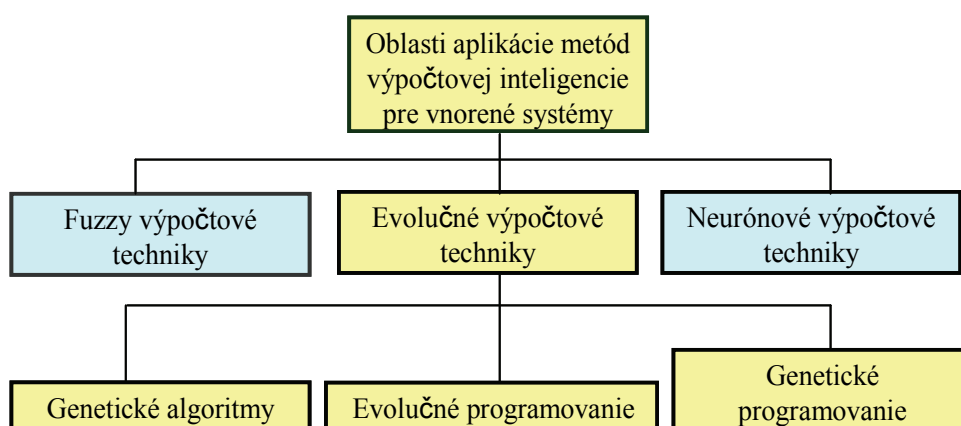
Aj keď konvenčné metódy modelovania a riadenia tvoria najväčšiu triedu aplikácií, prienik inteligentných metód do riadiacich a rozhodovacích procesov a ich potreba je nevyhnutná z dôvodu zvýšenia kvality a bezpečnosti riadiacich procesov.

Dôvodom pre mimoriadny záujem o výskum a aplikáciu inteligentných metód do vnorených systémov je ich vysoká kvalita modelovania a mimoriadna vysoko presná aproximačná schopnosť reprezentovať svojou štruktúrou a výpočtom optimálnych

parametrov zákonov riadenia chovanie sa zložitých procesov, ktoré konvenčnými technikami nie sme schopní dosiahnuť.

Inteligentné metódy modelovania a riadenia systémov (obr. 23), ktoré sú cielene vyvíjané pre aplikáciu vo vnorených systémoch sú založené na fuzzy logike, umelých neurónových sieťach a genetických algoritmov tak pre spojité (spojitá dynamika), ako aj pre hybridné systémy (kombinácia spojitaj a diskretnaj dynamiky).

Aplikácie (IVS) dnes smerujú do nových technológií a všade tam, kde je potreba výrazného zlepšenia kvality riadenia procesov za účelom minimalizácie spotreby energií, zvýšenia spoľahlivosti a bezpečnosti prevádzok, akými sú napr. chemické procesy, biotechnológie, mechatronické systémy, dopravné systémy a procesy v energetike.



**Obr. 23.** Prehľadová schéma vývoja a aplikácie metód výpočtovej inteligencie pre vnorené systémy.

Mohutný a rýchly prienik zabudovaných výpočtových systémov (tab. 3) do spotrebnej elektroniky, zdravotníctva, elektrotechniky, strojárstva mení podstatným spôsobom ich kvalitu, zvyšuje konkurencieschopnosť výrobkov, bezpečnosť a spoľahlivosť.

## 2.1 Využitie inteligentných metód modelovania a riadenia pre vnorené aplikácie na báze fuzzy množín

Fuzzy teória je moderná vedná disciplína, ktorá sa počas svojej skoro štyridsaťročnej existencie zaslúžila o riešenie mnohých technických a netechnických problémov modelovania a riadenia, ktoré sa v praxi nedali inými modelovacími prostriedkami a algoritmi riadenia zvládnuť. Osvedčila sa rádovo u niekoľko tisíc aplikácií v oblasti riadenia, diagnostiky, spracovania obrazov a prognostike. Teória fuzzy množín patrí medzi najlepšie matematicky spracované časti umelej inteligencie, pomocou ktorej môžeme modelovať, simulovať a riadiť veľkú časť procesov ľudského uvažovania. Fuzzy množiny (FM) sú zovšeobecnením klasických množín.

Matematicky ide o pomerne dobre preskúmanú a stále aktuálnu výskumnú a aplikačnú oblasť.

**Tabuľka 3.** Prehľad aplikácie inteligentných metód vo vnorených systémoch

Metódy	Charakteristika	Vnorené aplikácie
Fuzzy systémy (FS)	Vhodnosť pre vnorené aplikácie, jednoduchosť algoritmov, real-time aplikácie, FPGA implementácie na čipe.	Spotrebiče, kamery, automobilový priemysel, priemyselné procesy (tepelné procesy), dávkovacie systémy, parkovacie systémy.
Umelé neurónové systémy a siete (UNS)	Vhodnosť pre nelineárne procesy, (dopredu natrénovaná sieť, off-line použitie, vhodnosť pre procesy s pomalou dynamikou, opakované činnosti, FPGA realizácie na čipe.	Tepelné procesy so zložitou dynamikou (tepelné procesy, chemický priemysel, biotechnológie - kvasenie, sušiacie procesy a pod.)
Neuro-fuzzy systémy a siete (NFS a FNS)	Vhodná kombinácia fuzzy a neuro prístupov (ANFIS, CANFIS) pre silno nelineárne a neurčité procesy, vyššia kvalita modelovania a riadenia ako u FS a UNS, hardvérová realizácia cez FPGA .	Vybrané technologické procesy v chémii, biotechnológiách a pod. (tepelné procesy, optické systémy - kamery, spotrebná elektronika)
Genetické algoritmy (GA)	V kombinácii s FS a UNS umožňujú riešiť zložité úlohy modelovania, riadenia a rozhodovania. Vzhľadom na dlhší CPU – time použitie pre off-line úlohy .	Vybrané technologické procesy v chémii, biotechnológiách a pod.
Model checking (MCH)	Automatické inteligentné metódy pre verifikovanie a testovanie hybridných konečno-stavových konkurenčných systémov (spojenie spojitej a diskkrétnej dynamiky)	Testovanie a verifikovanie zložitých štruktúr distribuovaných systémov, návrh a výber optimálnej štruktúry prevádzky.
Multi agentové systémy	Distribuované inteligentné vnorené systémy, hybridné použitie FS +UNS + GA	Dopravné systémy, vojenská technika, navigačné systémy, pružné výrobné systémy

Podobne, ako umelé neurónové siete, aj fuzzy množiny predstavujú univerzálny aproximátor funkcií, a preto ich využitie v modelovaní a riadení rozličných typov procesov vzhľadom na jednoduchý matematický aparát je efektívne. Aj v praxi sa dostalo značného využitia a modernizácie zariadení najmä fuzzy regulátorom.

Fuzzy množiny môžeme formálne definovať nasledovne:



Nech  $X$  je univerzum všetkých uvažovaných prvkov  $x$  a nech  $M = \langle 0, 1 \rangle$  reprezentuje interval, v ktorom budeme vyhodnocovať stupne príslušnosti prvkov do fuzzy množiny. Fuzzy množina  $A$  v univerzu  $X$  je potom vyjadrená zobrazením  $A: X \rightarrow M$ , ktorá každému prvku  $x \in X$  priradzuje prvok  $\mu_A \in M$ , ktorý sa nazýva stupeň príslušnosti prvku  $x$  do fuzzy množiny  $A$ .

- ak je  $\mu_A = 0$ , potom  $x$  nepatrí do  $A$ , ak  $\mu_A = 1$ , potom  $x$  patrí do  $A$ .

- ak  $\mu_A \neq 0$  a súčasne  $\mu_A \neq 1$ , potom  $x$  čiastočne patrí do fuzzy množiny  $A$ .

Fuzzy množinu  $A$  môžeme explicitne definovať a vyjadriť nasledovne:

$$A = \{x_1/\mu_{A1}, x_2/\mu_{A2}, \dots, x_n/\mu_{An}\}, n \in \mathbf{N}, \quad A = \{x, \mu_A(x)\}, \forall x \in X$$

Praktická implementácia fuzzy množín vo vnorených aplikáciách je založená na troch základných princípoch:

Modelovanie - aproximácia ľubovoľných funkcií pomocou fuzzy množín. Klasické ostré množiny: aproximácia pomocou polynómov, splinov, ortogonálnych polynómov, autoregresných modelov typu ARX, ARMA, BJ, OE, IV, OE.

Formalizovanie znalostí vyjadrených určitými lingvistickými pojmami. Lingvistické premenné - vety - tvoria kauzálne reťazce pre predikovanie, porozumenie, rozhodovanie, učenie sa, pamätanie situácií a riadenie.

Algoritmizácia fuzzy algoritmov a generovanie kódu pre príslušný typ vnoreného procesora.

#### *Princíp návrhu inteligentných algoritmov modelovania a riadenia s využitím fuzzy množín*

Navrhnuť inteligentný systém modelovania a riadenia na báze fuzzy logiky znamená vytvoriť model správania sa systému pomocou rozhodovacích pravidiel vyjadrujúcich cieľavedomé a kauzálne správanie sa skúmaného procesu, získaných na základe skúseností experta alebo na základe meraných údajov.

#### *Princíp fuzzy množín pre úlohy modelovania a riadenia*

Modelovanie a riadenie – vyjadrenie a využitie pravidiel (približné hodnoty) prostredníctvom fuzzy množín:

if < fuzzy výrok >	then	< fuzzy výrok >
if < fuzzy výrok >	then	< fuzzy expression >
if antecedent	then	consequent

vstupy v pravidlách sú obsiahnuté v podmienkovej časti pravidiel (predovšetkým v konjunkcii),

výstupy sú obsiahnuté v konsekvente (fuzzy množina alebo matematická funkcia), počet pravidiel determinuje rozsiahlosť úlohy (konečná množina, reálny čas, jednoduchá štruktúra).

Pravidlá reprezentujú vedomosti experta - lingvisticky expertné pravidlá. Všeobecný tvar pravidla :

if  $x$  is  $A$  then  $y$  is  $B$  (jedno pravidlo)  
 if  $x_i$  is  $A_{ij}$  AND  $x_2$  is  $A_{2j}$  AND ... AND  $x_N$  is  $A_{nj}$  then  $y$  is  $B_j$

pre  $i = 1, 2, \dots, N$ , ( $N$  je počet vstupných premenných),  
 $j = 1, 2, \dots, M$ , ( $M$  je počet pravidiel).

výstupná premenná (konsekvent) je buď jedna ( $y$ ), alebo je viacrozmerná  $y \in Y$ .  
 $A, B, A_{ij}, B_{ij}$  - sú približné hodnoty modelované ako fuzzy množiny na príslušných univerzách,

*is* - je operátor pre zhodu medzi obidvoma operandami v zmysle čiastočnej príslušnosti prvku  $x_i$  do fuzzy množiny  $A_{ij}$ , resp.  $B_{ij}$ ,

podmienková časť pravidla je zostavená vo všeobecnosti z určitého počtu čiastočných podmienok, z nich každá môže byť splnená do určitého stupňa,

podmienkové časti sú spojené konjunktívnou spojkou *AND* (môže byť implementovaná rôznym spôsobom),

miera platnosti konsekventu je ovplyvnená celkovou mierou splnenia podmienok (na ľavej strane pravidla),

ak sa použije *AND* (operátor minima), je miera splnenia ľavej časti daná jedinou hodnotou a to najmenšou hodnotou zo všetkých čiastočných podmienok,

*then* - viacero spôsobov aplikácií (spojka *THEN* sa realizuje T-normou - algebraický súčin alebo silný prienik).

**Tabuľka 4.** Prehľad najpoužívanejších vzťahov pre výpočet T-normy a T-konormy

T-konorma dvoch fuzzy množín A, B.	
$\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\}$	Zadeh (maximum)
$\mu_{A \cup B}(x) = \min\{\mu_A(x) + \mu_B(x), 1\}$	Luka (obmedzený súčet)
$\mu_{A \cup B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x)$	Algebraický súčet
$\mu_{A \cup B}(x) = \begin{cases} \mu_A(x) & \text{pre } \mu_B(x) = 0 \\ \mu_B(x) & \text{pre } \mu_A(x) = 0 \\ 1 & \text{pre } \mu_A(x) \neq 0, \mu_B(x) \neq 0 \end{cases}$	Drastický (odvážny) súčet
$\mu_{A \cup B}(x) = \min\{1, \mu_A(x) + \mu_B(x) + p - \mu_A(x) \cdot \mu_B(x)\}$ pre $p \geq -1$	Sugeno

T-norma dvoch fuzzy množín A, B.	
$\mu_{A \cap B}(x) = \min \{ \mu_A(x), \mu_B(x) \}$	Zadeh (minimum)
$\mu_{A \cap B}(x) = \max \{ 0, \mu_A(x) + \mu_B(x) - 1 \}$	Luka (obmedzený súčin)
$\mu_{A \cap B}(x) = \mu_A(x) \cdot \mu_B(x)$	Algebraický súčin
$\mu_{A \cap B}(x) = \begin{cases} \mu_A(x) & \text{pre } \mu_B(x) = 1 \\ \mu_B(x) & \text{pre } \mu_A(x) = 1 \\ 0 & \text{pre } \mu_A(x) \neq 1, \mu_B(x) \neq 1 \end{cases}$	Drastický (odvážny) súčin
$\mu_{A \cap B}(x) = \frac{\mu_A(x) \cdot \mu_B(x)}{p + (1-p) \cdot (\mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x))}$ pre $p \geq 0$	Hamacher
$\mu_{A \cap B}(x) = \log_p \left( 1 + \frac{(p^{\mu_A(x)} - 1) \cdot (p^{\mu_B(x)} - 1)}{p - 1} \right)$ pre $p > 0$	Frank
$\mu_{A \cap B}(x) = 1 - \min \left\{ 1, \left[ (1 - \mu_A(x))^p + (1 - \mu_B(x))^p \right]^{\frac{1}{p}} \right\}$ pre $p \geq 0$	Yager
$\mu_{A \cap B}(x) = \left( \frac{\mu_A(x) \cdot \mu_B(x)}{\max(\mu_A(x), \mu_B(x), p)} \right)$ pre $0 \leq p \leq 1$	Dubois-Prade
$\mu_{A \cap B}(x) = 1 - \left\{ (1 - \mu_A(x))^p + (1 - \mu_B(x))^p - (1 - \mu_A(x))^p \cdot (1 - \mu_B(x))^p \right\}^{\frac{1}{p}}$ pre $p \in R$	TP

*Modelovanie nelineárnych dynamických procesov pomocou fuzzy množín (FM)*

Hlavná myšlienka pri modelovaní procesov FM : Aproximácia funkcie s neznámou štruktúrou FM – univerzálny aproximátor (fuzzy aproximácia s ľubovoľnou presnosťou dokáže aproximovať každú spojitú ohraničenú funkciu).

Veta: Ak  $U \subseteq R$  je kompaktná množina, potom ku každej ohraničenej spojitaj funkcii  $f : U \rightarrow R$  a  $\varepsilon > 0$  existujú fuzzy množiny  $A_j \subseteq U, B_j \subseteq R, j=1, \dots, m$  a defuzifikačná funkcia  $DEF: F \rightarrow R$  taká, že  $|f(x) - DEF(Bx)| < \varepsilon$ .

Typy fuzzy modelov

*Lingvistické fuzzy modely I:* antecedent a konsekvent sú fuzzy (*Fuzzy relačný model:* zovšeobecnenie lingvistických modelov relácia medzi antecedentom a konsekventom je fuzzy):

if  $x$  is  $A$  then  $y$  is  $B$  (jedno pravidlo)

Lingvistické fuzzy modely II (Takagi–Sugeno modely (TS)): antecedent je fuzzy, konsekvent je ostrá funkcia (číslo, polynóm).

Špeciálny typ modelu I, konsekvent nie je fuzzy množina (číslo-singleton)

$$\mathbf{R}_i: \text{if } x \text{ is } A_i \text{ then } y=b_i, \quad i=1,2, \dots, K$$

Výstup – defuzzifikácia:

$$y = \frac{\sum_{i=1}^K \beta_i b_i}{\sum_{i=1}^K \beta_i}$$

singleton – {univerzálnych aproximátorov} (basis function expansion)

$$y = \sum_{i=1}^K \phi_i(x) b_i$$

Zovšeobecnenie fuzzy modelu pre viacero vstupov:

$$\mathbf{R}_i := \text{if } x_1 \text{ is } A_{i1} \text{ and } \dots \text{ and } x_n \text{ is } A_{in} \text{ then } y = a + b_1 x_1 + \dots + b_n x_n$$

Nelineárne dynamické fuzzy modely :

model riadeného procesu je vyjadrený pomocou fuzzy množín ako vstupno-výstupný model,

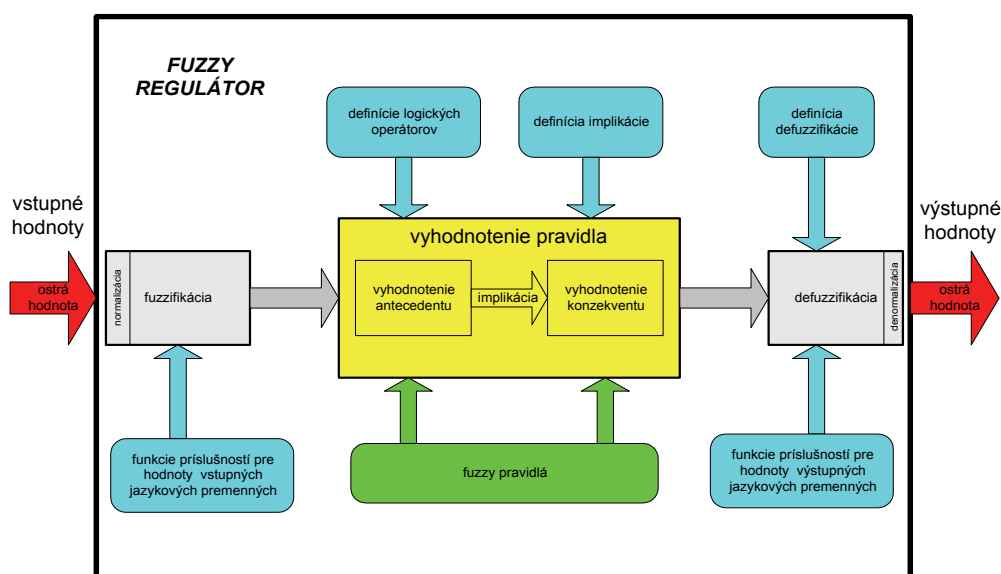
konsekventom je váhovaný výstup,  $k$ -je diskretný čas,  $y(k+1)$  je výstupná premenná v čase  $k+1$

$$\begin{aligned} \mathbf{R}_i : & \text{if } y(k) \text{ is } A_{i1} \text{ and } \dots \text{ and } y(k - n_y + 1) \text{ is } A_{iny} \text{ and } \dots \\ & \text{if } u(k) \text{ is } B_{i2} \text{ and } \dots \text{ and } u(k - n_u + 1) \text{ is } B_{inu} \text{ then} \\ y(k+1) &= \sum_{j=1}^{N_y} a_{ij} y(k-j+1) + \sum_{j=1}^{N_u} b_{ij} u(k-j+1) + c_i, \quad i = 1, \dots, K \end{aligned}$$

kde  $A_{il}$ ,  $B_{il}$  sú fuzzy množiny a  $a_{ij}$ ,  $b_{ij}$  a  $c_i$  sú ostré hodnoty parametrov konsekventu.

Návrh vnoreného systému pre ovládanie a riadenie zariadení a procesov sa opiera a využíva metodiku návrhu, ktorá používa konvenčné postupy riadenia založené na fuzzy modeloch.

Základná schéma vnoreného systému pre syntézu riadiacich algoritmov je na obr. 23.



Obr. 23. Bloková schéma návrhu vnoreného fuzzy riadenia.

## 2.2 Štruktúra fuzzy vnoreného regulátora

Tak ako aj u konvenčných algoritmov riadenia, aj vnorený počítačový systém realizujúci algoritmus riadenia má tri nadväzne funkčné bloky:

### Blok fuzzifikácie

Fuzzifikácia predstavuje prvý krok v realizácii fuzzy regulátora. Blok fuzzifikácie transformuje procesné údaje (ostré hodnoty vstupov) do intervalu  $(0,1)$  fuzzy množín. Pred operáciou fuzzifikácie je potrebné realizovať normalizáciu vstupných hodnôt. Normalizácia v tomto prípade znamená prevod ešte „ostrých“ hodnôt z objektu riadenia, ktoré bývajú často v rozsahu definovanom rôznymi typmi snímačov (napr. 0 až 10 V), na rozsah univerzít, s ktorými pracuje fuzzy vnorený regulátor. Najčastejšie sú všetky univerzá v rozsahu  $(-1,1)$ , aby bolo možné bez ďalších problémov použiť rôzne typy fuzzy množín, T-noriem, implikácií a algoritmov učenia, napr. pri realizácii hybridných algoritmov riadenia v spojení s neurónovými sieťami.

Blok bázy znalostí býva rozdelený do viacerých menších celkov:

*Báza dát* (modrá), v ktorej sú uložené tvary a počty funkcií príslušností, vstupných a výstupných premenných.

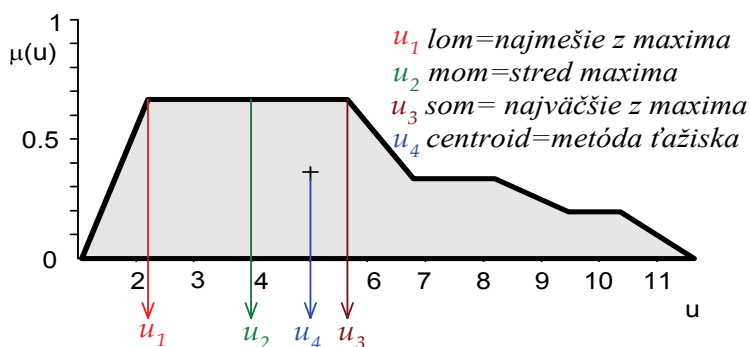
*Báza pravidiel* (zelená), v ktorej sú uložené jednotlivé pravidlá v tvare fuzzy implikácie. Všetky pravidlá sú spojené pomocou operácie agregácie do jedného zloženého fuzzy výroku vyjadrujúceho stratégiu riadenia fuzzy regulátora.

### Blok inferenčného mechanizmu

V tomto bloku je realizovaný samotný mechanizmus vyhodnotenia jednotlivých pravidiel. Blok realizuje vyhodnotenie antecedentu, implikácie a konsekventu. Výstupom z inferenčného mechanizmu býva napr. fuzzy množina (ale aj matematická funkcia) vytvorená kompozíciou z fuzzy výstupných množín jednotlivých pravidiel, ktorá slovne vyjadruje riadiaci zásah z fuzzy vnoreného regulátora.

### Blok defuzzifikácie

Transformuje výstupnú jazykovú hodnotu, ktorá vznikla kompozíciou výstupov z jednotlivých pravidiel, na „ostrú“ hodnotu, ktorá vstupuje do zariadenia cez aktuátor v podobe riadiaceho zásahu. Existuje viacero spôsobov ako získať z celkovej fuzzy plochy „ostrú“ hodnotu, ktoré sa líšia prístupom k samotnému výpočtu. Všetky vychádzajú z vyjadrenia celkovej výstupnej fuzzy plochy pomocou analytického výrazu. Tie najčastejšie sú zobrazené na obr. 24, pričom najpoužívanejšou je metóda ťažiska (centroid). Okrem spomínaných metód existuje aj mnoho ďalších líšiacich sa metodikou výpočtu ostrej hodnoty a typom riadeného procesu. Po samotnej defuzzifikácii nasleduje tzv. denormalizácia, ktorá je opakom normalizácie, a ktorej cieľom je transformácia riadiaceho zásahu definovaného v rozsahu výstupného univerza do rozsahu potrebného pre riadenie zariadenia (technológie).



**Obr. 24.** Defuzzifikačné metódy vo vnorených systémoch riadenia.

Mnohé podporné programové systémy pre výpočet riadiaceho zásahu prezentujú riešenia v numericko-grafickej forme (Matlab-Simulink). Vizualizácia riešenia a interakcia používateľa pri *Co-design* SW a HW umožňuje konfiguráciu tak metód fuzzifikácie, výber počtu pravidiel a výber takej defuzzifikačnej metódy, ktorá zaručí optimálnu kvalitu a stabilitu riešenia.

Pri návrhu fuzzy systémov riadenia sa núkajú rôzne typy algoritmov, ktoré sa líšia medzi sebou typom výberu rôznych implikačných funkcií, tvarov a počtom funkcií príslušnosti a pod. Kombináciou a výberom implikačných a defuzzifikačných metód môžeme získať rôzne typy riadiacich algoritmov. Podľa výberu týchto metód môžeme rozlišovať rôzne typy fuzzy regulátorov. Jednotlivé typy sa líšia inferenčným

mechanizmom („filozofiou“ vyhodnocovania pravidiel) a s ním súvisiacou rýchlosťou výpočtu.

**Tabuľka 5.** Vybrané metódy implikácie

Vzťah	Názov
$\mu_r(x, y) = \max(1 - \mu_A(x), \mu_B(y))$	Kleeneova a Dienesova
$\mu_r(x, y) = \min(1, 1 - \mu_A(x) + \mu_B(y))$	Łukasiewiczova
$\mu_r(x, y) = \max(\min(\mu_A(x), \mu_B(y)), 1 - \mu_A(x))$	Zadehova
$\mu_r(x, y) = \min(1, 1 - \mu_A(x) + \mu_A(x) \cdot \mu_B(y))$	Reichenbachova
$\mu_r(x, y) = \min\left(1, \frac{\mu_B(y)}{\mu_A(x)}\right)$	Goguenova
$\mu_r(x, y) = \max(1 - \mu_A(x), \mu_A(x) \cdot \mu_B(y))$	stochastická
$\mu_r(x, y) = \begin{cases} 1 & \text{pre } \mu_A(x) \leq \mu_B(x) \\ \mu_B(x) & \text{pre } \mu_A(x) > \mu_B(x) \end{cases}$	Gödelova
$\mu_r(x, y) = \begin{cases} 1 & \text{pre } \mu_A(x) \leq \mu_B(x) \\ 0 & \text{pre } \mu_A(x) > \mu_B(x) \end{cases}$	ostrá
$\mu_r(x, y) = \min(\mu_A(x), \mu_B(y))$	Mamdaniho (minimová)
$\mu_r(x, y) = \mu_A(x) \cdot \mu_B(y)$	Larsenova (súčinová)

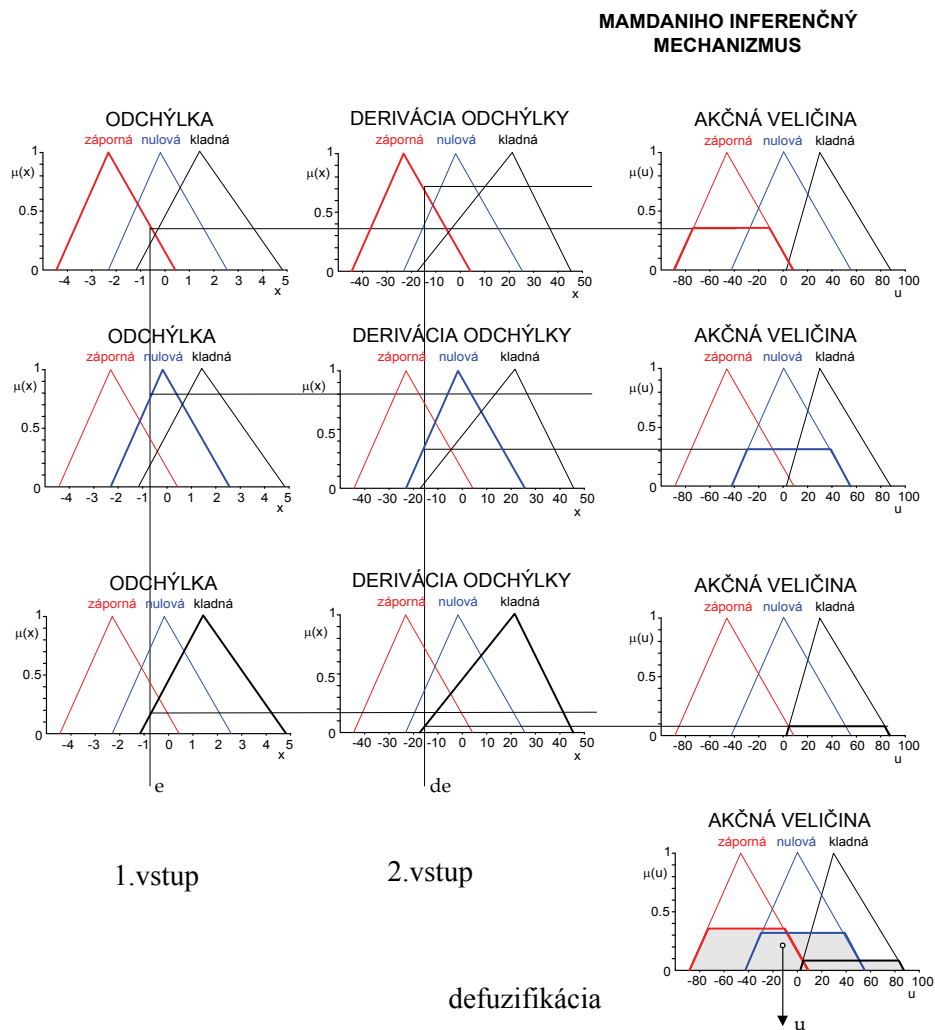
V ďalšej časti uvedieme len základné typy fuzzy algoritmov riadenia vhodných pre vnorené aplikácie:

- Mamdaniho fuzzy regulátor
- Larsenov fuzzy regulátor
- Tsukamoto fuzzy regulátor
- Sugenov fuzzy regulátor

Pre znázornenie návrhu algoritmu fuzzy riadenia pre dva vstupy (odchýlka  $e$  a diferencia odchýlky  $\Delta e$ ) a jeden výstup  $u$  (každý z nich má tri jazykové hodnoty) a tri pravidlá s rôznymi typmi inferencií, je postup výpočtu riadiaceho zásahu znázornený na obr. 25 - obr. 27. Pre Mamdaniho typ fuzzy riadenia je použitá T-norma definovaná ako minimum, T-konorma ako maximum a metóda defuzzifikácie je založená na výpočte ťažiska. Mamdaniho inferenčný algoritmus je pre vybrané tri pravidlá, dva vstupy (odchýlka a diferencia odchýlky) a jeden výstup riadiaci zásah (akčná veličina) v grafickej podobe demonštrovaný na nasledujúcom demonštračnom príklade pre možnosť použitia vo vnorených systémov riadenia s defuzzifikačnou metódou výpočtu ťažiska.

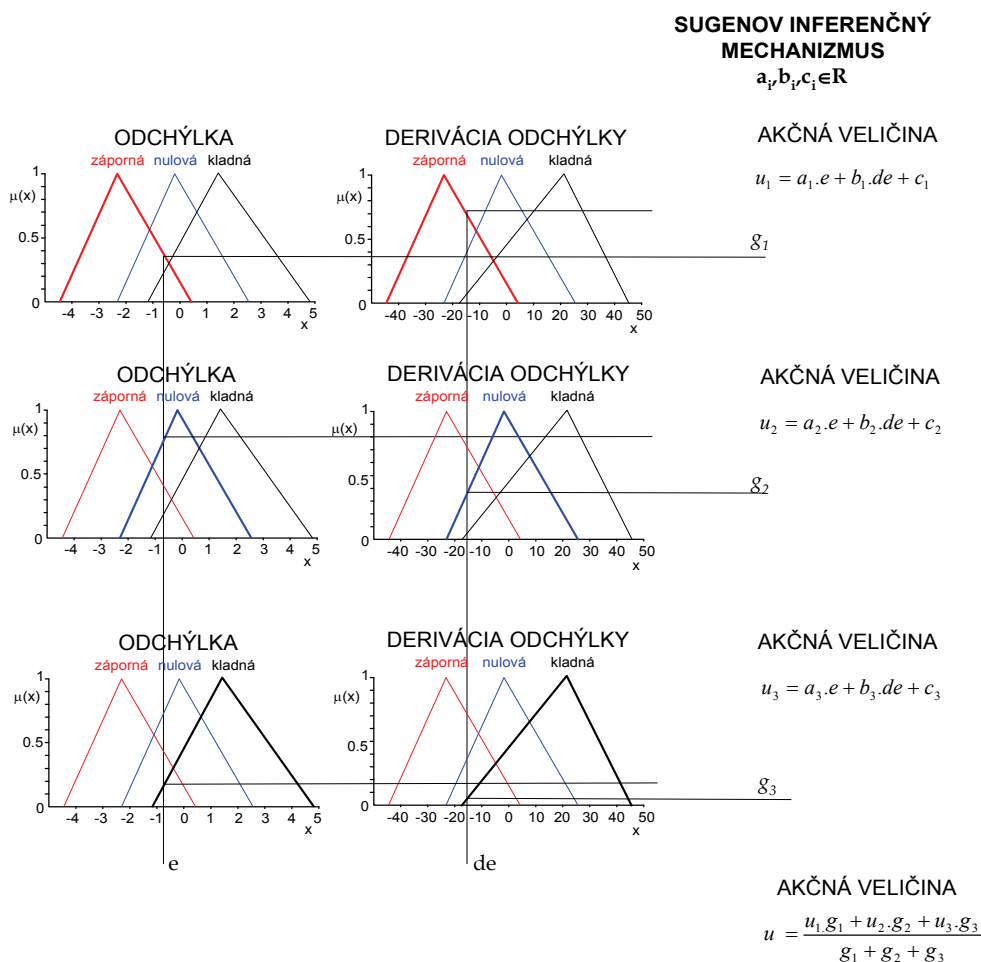
**Tabuľka 6.** Časť pravidiel fuzzy riadenia (pre demonštráciu použitia vo vnorených systémoch)

ak $e$ je záporná a $\Delta e$ je záporná potom $u$ je záporná
ak $e$ je nulová a $\Delta e$ je nulová potom $u$ je nulová
ak $e$ je kladná a $\Delta e$ je kladná $u$ je kladná



**Obr. 25.** Demonštračný príklad použitia Mamdaniho inferenčného mechanizmu pre vnorené aplikácie.





**Obr. 26.** Demonštračný príklad použitia Takagi-Sugenovho inferenčného mechanizmu pre vnorené algoritmy riadenia.

Sugenov inferenčný mechanizmus pre pravidlá uvedené v tab. 6 je znázornený na obr. 27. Na rozdiel od Mamdaniho inferenčného mechanizmu výstupom nie je fuzzy množina, ale analytická kombinácia funkcie ostrých vstupov do fuzzy regulátora. Najčastejšie to býva lineárna kombinácia vstupných premenných alebo len fuzzy singleton (číslo).

Pre jedno pravidlo a dva vstupy je konsekvent v podobe kombinácie vstupov :

if  $x_1$  is  $A1$  and  $x_2$  is  $A2$  then  $u=f(x_1,x_2)$

kde  $A1, A2$  sú fuzzy množiny (jazykové hodnoty),  $x_1, x_2$  sú vstupné premenné a  $f(x_1,x_2)$  je analytický výraz pre výpočet výstupu z pravidla. Najčastejšie je  $f(x_1,x_2)$  lineárna funkcia opísaná rovnicou  $f(x_1, x_2) = ax_1 + b \cdot x_2 + c$ .

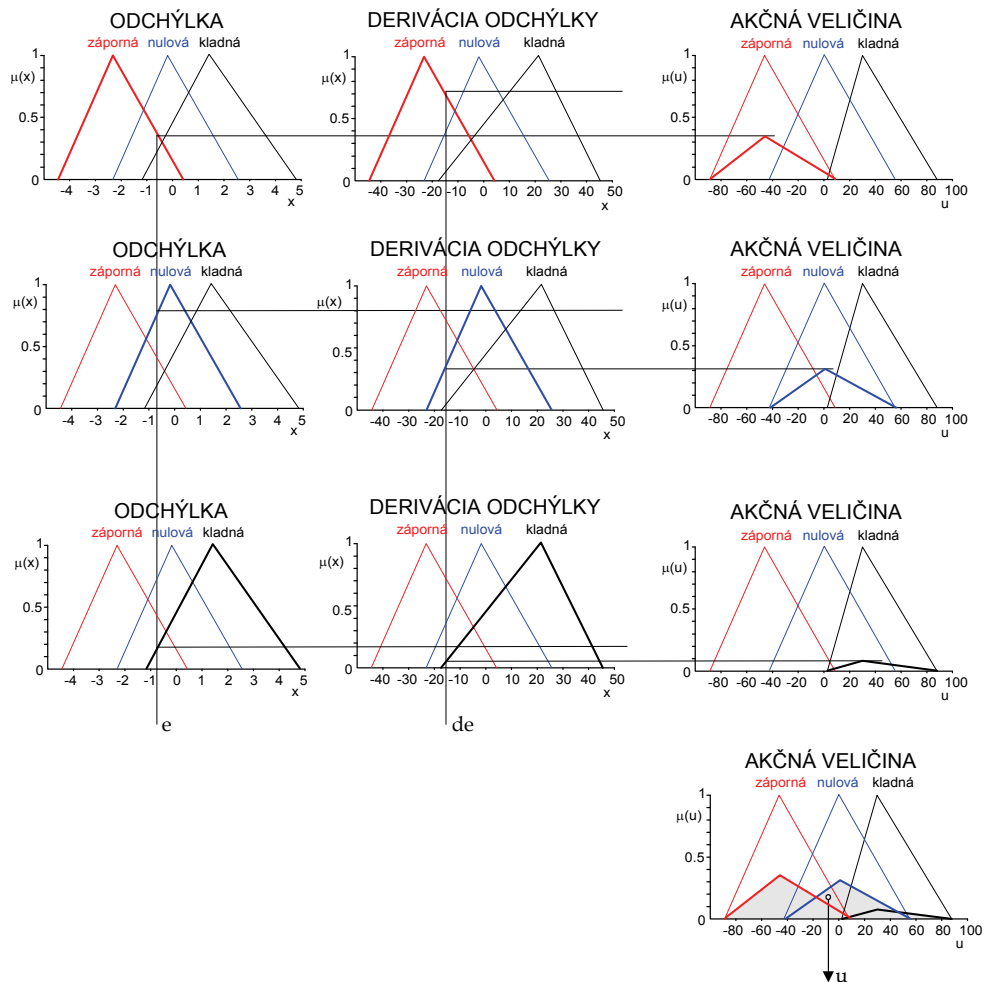
Výstup z inferenčného mechanizmu predstavuje riadiaci zásah (akčná veličina) a

je vyjadrený 
$$u = \frac{\sum_{i=1}^k w_i \cdot u_i}{\sum_{i=1}^k w_i}$$
, kde  $w_i$  je výstupná váha z antecedentu  $i$ -teho pravidla

pre  $i=1, \dots, K$ .

Ďalším vhodným kandidátom pre vnorený typ fuzzy riadenia je **Larsenov** fuzzy systém riadenia.

### LARSENOV INFERENČNÝ MECHANIZMUS



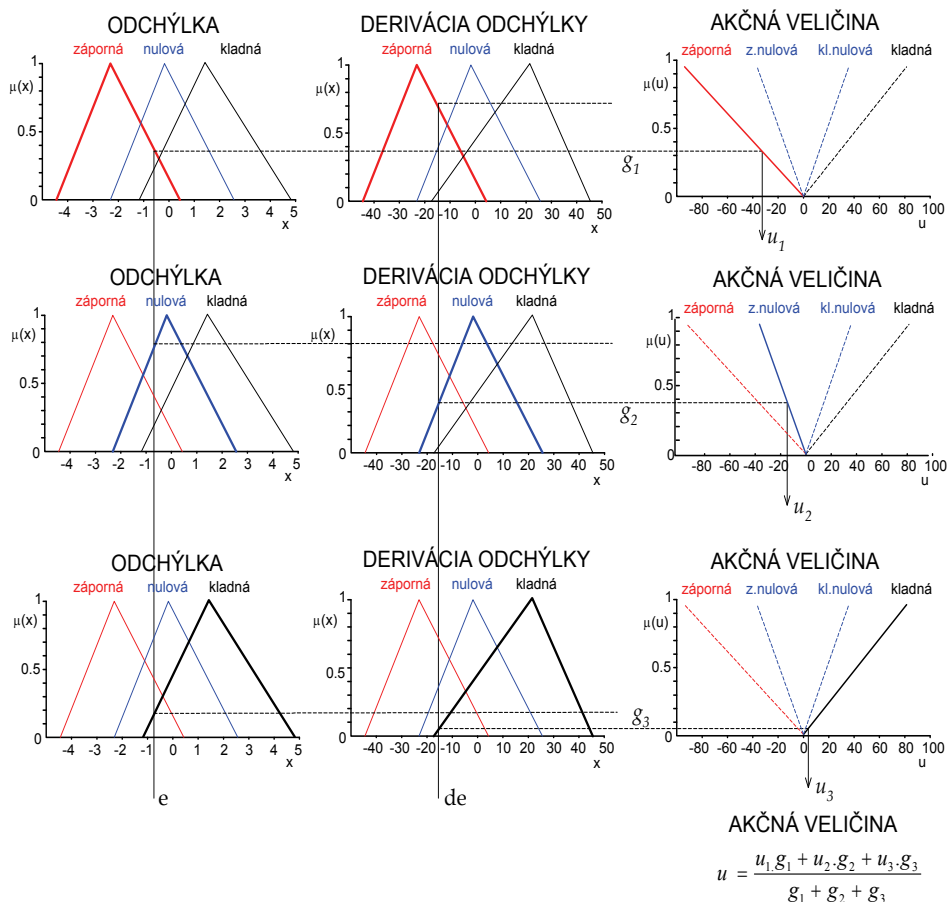
**Obr. 27.** Demonštračný príklad použitia Larsenovho inferenčného mechanizmu pre vnorené algoritmy riadenia.

Larsenov inferenčný mechanizmus má rovnaké vlastnosti ako Mamdaniho, ale implikácia je tu definovaná ako algebraický súčin. V literatúre je často uvádzaný ako modifikácia Mamdaniho regulátora.

Ďalším typom regulátora je **Tsukamoto**v fuzzy regulátor (obr. 28). Má podobné vlastnosti ako Mamdaniho regulátor, líši sa však funkciami príslušnosti (FP) antecedentov a konsekventov. FP sú definované ako lineárne monotónne funkcie. Výstupná funkcia je počítaná pomocou váhovaných výstupov jednotlivých pravidiel.

$$u = \frac{\sum_{r=1}^k \min_{i=1..n} (\mu_{i,r}(x_i)) \cdot u_r}{\sum_{r=1}^k \min_{i=1..n} (\mu_{i,r}(x_i))}$$

**TSUKAMOTOV INFERENČNÝ MECHANIZMUS**



**Obr. 28.** Demonštračný príklad použitia Tsukamotovho inferenčného mechanizmu pre vnorené algoritmy riadenia.

Vlastnosti inteligentných vnorených regulátorov založených na fuzzy logike:

**Nelineárnosť** – súvisí s kvalitatívnou povahou celého fuzzy regulátora.

**Jednoduchosť** – fuzzy regulátor je vo svojej podstate jednoduchý výpočtový člen, ktorý potrebuje relatívne malé množstvo aritmetických operácií. V súčasnosti existujú realizácie algoritmov pomocou samostatných fuzzy čipy alebo FPGA obvodov, ktoré sú špeciálne určené na riadenia systémov pomocou fuzzy vnoreného regulátora.

**Stabilita a robustnosť** – vo všeobecnosti sa regulačné obvody s fuzzy regulátorom považujú za stabilné a robustné. Vyplýva to zo statickej povahy fuzzy regulátora. Je však potrebné zväziť, najmä pri náhrade konvenčných regulátorov fuzzy vnoreným regulátorom v spätnoväzbovej slučke, a posúdiť stabilitu (kritérium Popova alebo iné), či nemôže dôjsť k nestabilite celého obvodu. Signifikantný vplyv na stabilitu a kvalitu má nevhodne zostavená báza pravidiel, nesprávny výber funkcií príslušnosti a pod.

**Statická charakteristika** – fuzzy vnorený regulátor je vo svojej podstate statický aproximátor ľubovoľnej funkcie. Jeho funkčné vlastnosti sa dajú preto vyjadriť pomocou grafov, kde je znázornená statická závislosť výstupu na všetkých možných kombináciách vstupov. Je preto nutné potrebné derivácie a integrácie vstupných a výstupných veličín vypočítať v samostatnom matematickom bloku.

**Štruktúrovaná reprezentácia znalostí** – fuzzy vnorený regulátor umožňuje vďaka svojej kvalitatívnej povahe „nahliadnuť“ do algoritmu výpočtu v tvare človeku zrozumiteľných výrokov. Túto schopnosť nemajú iné regulátory.

**Potreba zložitého „ladenia“ a konfigurácie** – fuzzy vnorený regulátor, tak ako aj ostatné algoritmy riadenia, potrebuje nastaviť svoje parametre pre konkrétny systém a pre konkrétny účel riadenia. Počet pravidiel a ich tvar sa dá realizovať heuristicky, ale aj napr. na základe metód zhlukovej analýzy.

**Všeobecnosť** – fuzzy vnorený regulátor predstavuje viackriteriálny systém. Dá sa použiť ako náhrada za konvenčné algoritmy riadenia, ale aj ako náhrada alebo kombinácia s inými typmi riadiacich algoritmov. Môže nahradiť aj človeka v dispečerskom riadení. Aké bude nasadenie fuzzy regulátora a ako sa bude správať (stratégia riadenia), možno ovplyvniť a dynamicky meniť (napr. zmenou pravidiel). Pod všeobecnosťou rozumieme aj schopnosť spojenia s inými algoritmi riadenia na optimalizáciu činnosti fuzzy regulátora. Typickým príkladom je spojenie fuzzy regulátora a neurónovej siete alebo genetických algoritmov pri priebežnom (on-line) nastavovaní fuzzy regulátora.

Pod syntézou fuzzy vnoreného algoritmu riadenia rozumieme jeho návrh pre konkrétny systém (zariadenie, objekt), pričom môžeme samotný vývoj a overenie rozdeliť na viacero fáz:

- Stanovenie počtu, tvaru a rozloženia fuzzy množín.
- Stanovenie pravidiel.
- Stanovenie inferenčného mechanizmu.
- Stanovenie metódy defuzzifikácie.

### *Stanovenie počtu, tvaru a rozloženia fuzzy množín*

Jednou z najdôležitejších otázok pri návrhu vnoreného algoritmu riadenia je určenie počtu fuzzy množín. Keďže fuzzy logika vychádza z ľudského pozorovania procesu, nemá zmysel použiť veľké množstvo funkcií príslušnosti (FP) (jazykových hodnôt) pre jednu jazykovú premennú. Človek dokáže určiť len obmedzené množstvo úrovní jeho jednotlivých zmyslov, a preto sa počet funkcií volí obvykle z intervalu 3-7(9). Tvar funkcií príslušnosti je závislý od predpokladaného využitia fuzzy regulátora. V praxi sa najčastejšie stretáme s „triangulárnou“ a „lichobežníkovou“ funkciou. Okrem uvedených FP môžeme v praxi využívať „gaussovú“ funkciu príslušnosti a jej modifikované tvary.

Tvar funkcií závisí aj od dynamiky systému s fuzzy regulátorom, ktorú potrebujeme dosiahnuť. Napr. pre rýchle fuzzy regulátory býva často použitý tvar monotónnej funkcie príslušnosti, ktorá má zanedbateľný čas výpočtu.

Samotný tvar funkcií príslušnosti, rovnako ako počet, sa stanovuje buď (intuitívne) alebo je možné použiť matematické analýzy z nameraných dát, ktoré vstupujú do fuzzy regulátora. V druhom prípade je počet funkcií príslušnosti vyšší a závisí na vstupných dátach. Príkladom je tzv. zhuková analýza „fuzzy C-means“.

Pre fuzzy množiny platí, že musia pokryť celé univerzá jednotlivých jazykových premenných. To znamená, že ani jeden bod univerza nesmie mať stupeň príslušnosti rovný nule. Najčastejšie však táto podmienka býva ešte prísnejšia. Pri nej žiadny bod univerza nesmie mať stupeň príslušnosti menší, ako zvolené číslo, často v rozsahu (0.3-0.5). Pri splnení takejto podmienky existuje vždy aspoň jedno dominantné pravidlo, ktorého existencia má vplyv na správnu funkčnosť fuzzy regulátora.

### *Stanovenie pravidiel*

Stanovením počtu funkcií príslušnosti, teda počtu hodnôt jazykových premenných, môžeme považovať všetky ich kombinácie za stavový priestor vnoreného fuzzy regulátora. Každý bunke stavového priestoru (jednej kombinácii vstupov) potom potrebujeme určiť pravidlo, podľa ktorého sa má fuzzy regulátor pre daný stav správať. Ak máme teda napr. 2 vstupy a jeden výstup, pričom každý zo vstupov a rovnako aj výstup má 3 jazykové hodnoty (funkcie príslušnosti), potom je stavový priestor regulátora rozdelený na  $3^3=27$  buniek a potrebujeme najviac 27 pravidiel (stávajú sa prípady, keď je pravidiel menej). Určenie týchto pravidiel je jedna z najdôležitejších úloh pri syntéze fuzzy regulátora.

Pravidlá získame najčastejšie pomocou informácií získaných od experta, ktorý je odborník pre riadenie daného objektu. Využívajú sa pritom rôzne metódy získavania informácií, či už pomocou dotazníkov, ktoré vyplní operátor, alebo pozorovaním jeho činnosti. Pri pozorovaní sú využívané aj metódy „identifikácie“ pravidiel pomocou neurónových sietí (napr. ANFIS). V súčasnosti existujú metódy, pomocou ktorých si „fuzzy vnorený regulátor“ hľadá správne pravidlá sám pomocou genetických algoritmov a neurónových sietí.

### *Stanovenie inferenčného mechanizmu*

Stanovenie inferenčného mechanizmu znamená určenie typu T-normy, T-konormy a implikácie, ktoré je nutné aplikovať pre vyhodnocovanie pravidiel. Ich výber je často intuitívny. Ak je možnosť vykonať simulácie systému s fuzzy regulátorom, tak je často inferenčný mechanizmus vybraný metódou pokus – omyl, až po dosiahnutie požadovaných vlastností fuzzy regulátora.

### *Stanovenie metódy defuzzifikácie*

Dôležitou súčasťou komplexného návrhu vnorených algoritmov riadenia na báze fuzzy logiky je návrh a určenie metódy defuzzifikácie. Tu sa taktiež používajú rôzne heuristické prístupy a zatiaľ neexistuje jednotný postup pre určenie toho – ktorého typu defuzzifikácie pre danú aplikačnú oblasť.

### *Kombinované fuzzy - neuro techniky pre vnorené systémy*

Moderná metodológia inteligentných vnorených systémov umožňuje efektívne prepojenie a využitie najlepších vlastností tak fuzzy množín, ako aj umelých neurónových sietí. Možnosti efektívnej spolupráce a využitia vlastností fuzzy systémov a umelých neurónových sietí je niekoľko.

Systémy integrujúce fuzzy množiny a neurónové siete sa stali asi najrozšírejšími hybridnými systémami výpočtovej inteligencie. Kombinácií týchto dvoch technológií je viacero, dokonca v rôznej literatúre možno nájsť aj viacero rozdelení týchto hybridných systémov podľa rôznych hľadísk. My sa uspokojíme s veľmi hrubým rozdelením, založeným na vonkajšom správaní sa a vnútornej realizácii toho ktorého hybridného systému. Podľa tohto kritéria rozoznávame tri skupiny takýchto systémov:

1. **Neuro-fuzzy systémy** sa navonok javia ako fuzzy systém, ktorý je ale vnútorne realizovaný neurónovou sieťou. Ak je takto zrealizovaný napríklad fuzzy regulátor, tak tento má schopnosť učiť sa (adaptovať sa) alebo automaticky skonštruovať svoju bázu znalostí. Nevýhody fuzzy systému sú kompenzované použitím neurónovej siete. Najznámejšími architektúrami neuro-fuzzy systémov sú ANFIS, NARA či FALCON.

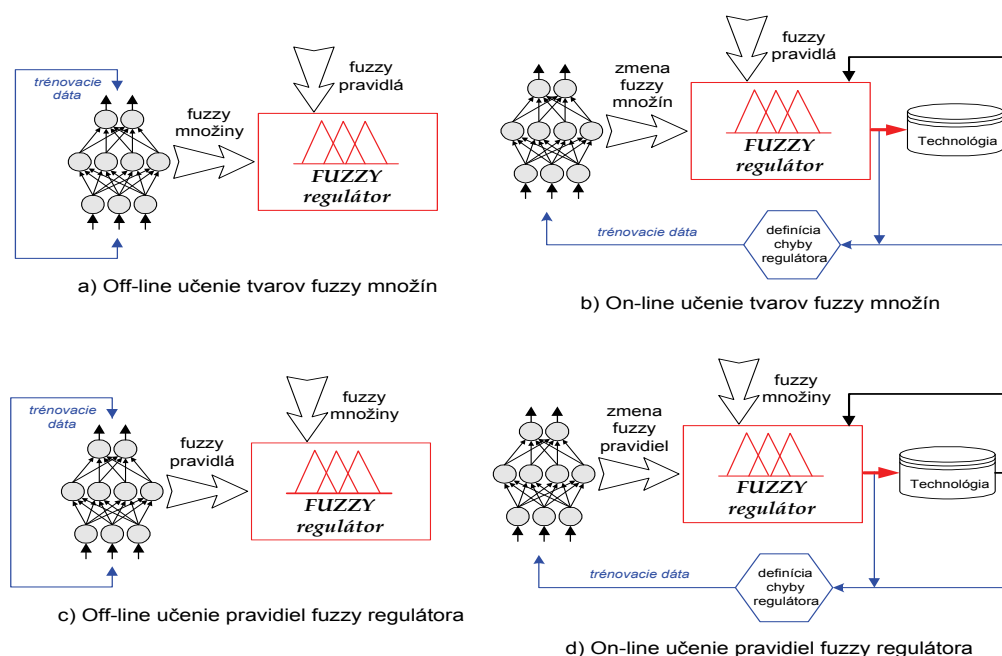
2. **Fuzzy - neurónové siete** sú neurónové siete doplnené o možnosti práce s neurčitou. Fuzzy logika je obyčajne implementovaná do neurónov, čím vznikajú tzv. fuzzy neuróny. Prioritou nasadenia fuzzy systémov je umožnenie práce s neurčitou. Toto do určitej miery dovoľuje poznatky uložené v takejto fuzzifikovanej neurónovej sieti považovať za abstraktnejšie, nakoľko takáto sieť nie je obmedzená na prácu z ostrými hodnotami.

3. Iný druh **interakcie** fuzzy systémov a neurónových sietí využíva fuzzy regulátor na adaptáciu učiacich sa parametrov siete. Týmto je zabezpečená rýchlejšia konvergenciu neurónovej siete do lokálneho minima, nakoľko možnosť zmeny učiacich parametrov neurónovej siete umožňujú modifikovať rýchlosť jej učenia sa v rôznych fázach učiaceho sa procesu.

Jednou z najdôležitejších úloh pri aplikácii vnorených fuzzy systémov je problém výberu štruktúry fuzzy systému a nastavenie jeho optimálnych parametrov. Existujú rôzne kombinácie ich prepojenia, základné štruktúry pre možné použitie vo vnorených aplikáciách sú nasledovné:

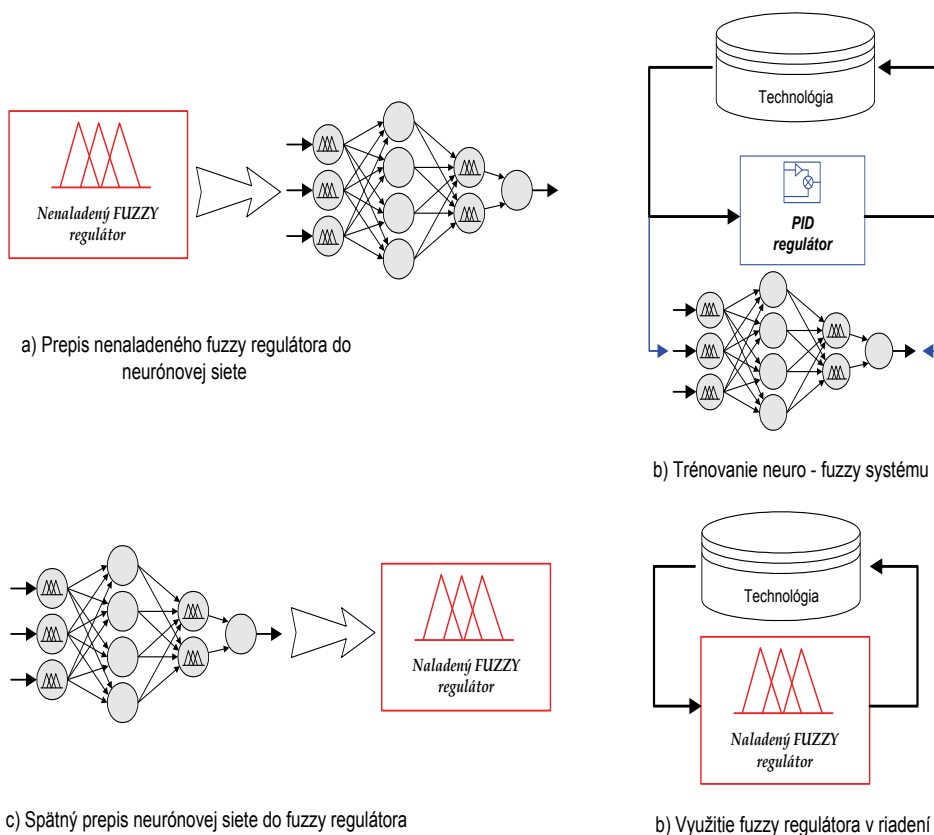
*Kooperatívna štruktúra*, neurónová sieť a fuzzy systém sú autonómne celky, pracujú nezávisle od seba a vymieňajú si informácie. Jednotlivé kooperatívne systémy možno ešte ďalej deliť, napríklad podľa cieľov a typov učenia. Príklad základnej schémy kooperatívneho systému pre spomínané rôzne ciele riadenia a typy učenia je na obr. 29. Výhodou kooperatívneho systému je ľubovoľnosť štruktúry fuzzy regulátora a neurónovej siete, nevýhodou je zložitosť prevedenia a výpočtová rýchlosť.

KOOPERATÍVNA ŠTRUKTÚRA FS a UNS



**Obr. 29.** Kooperatívna štruktúra fuzzy - neuro systémov.

*Hybridná štruktúra*, pri ktorej sú neurónová sieť a fuzzy regulátor spojené do jedného celku. Fuzzy systém je tu definovaný ako špeciálna neurónová sieť (obr. 30). Nevýhodou hybridného systému sú určité obmedzenia, pretože fuzzy - neurónová sieť je špeciálna sieť, ktorá vyžaduje matematické riešenia (problém derivovania funkcií v neuróne, nemožnosť použitia ľubovoľného učiaceho algoritmu a pod.). Je potrebné použiť konkrétny druh fuzzy regulátora a špeciálne typy neurónov v sieti. Výhodou tejto štruktúry je jednoduchosť a rýchlosť. Takýto hybridný systém sa najčastejšie používa pri on-line učení s posilňovaním (reinforcement).



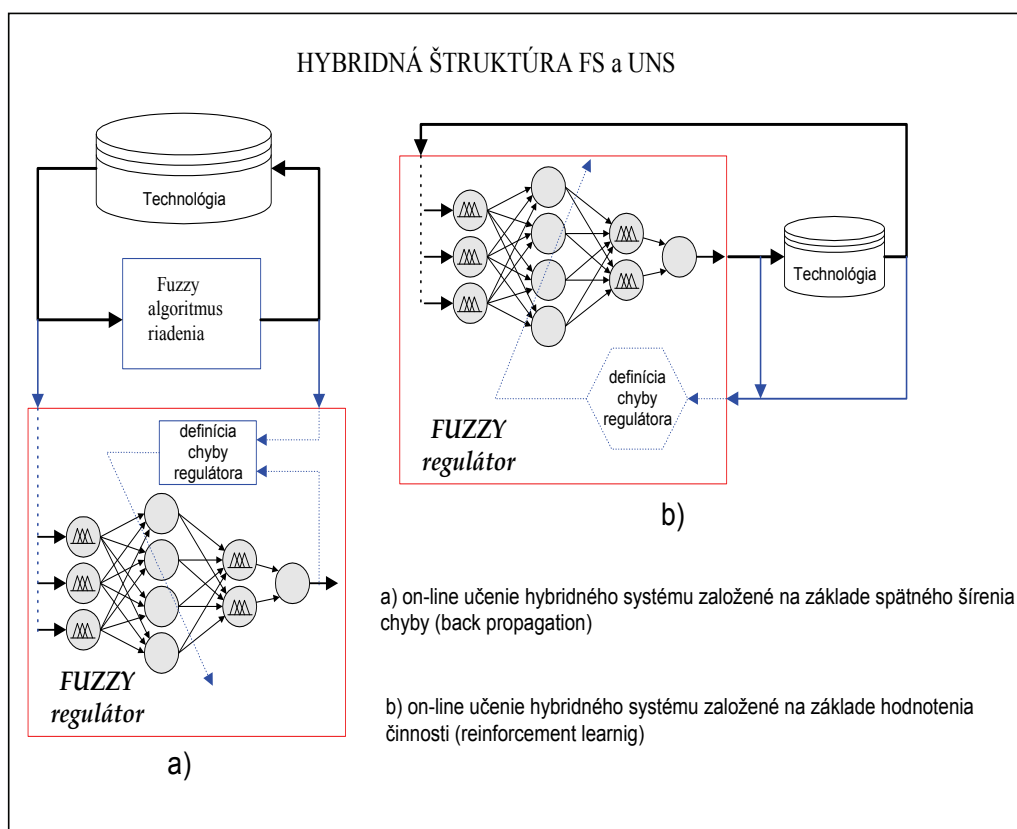
**Obr. 30.** Dvojetapový proces konfigurácie parametrov fuzzy riadenia.

Aplikácia konkrétnej kombinovanej štruktúry fuzzy – neuro, resp. neuro –fuzzy štruktúry a metodiky závisí od typu procesu, cieľov riadenia a od povahy apriórnych informácií, ktoré sú k dispozícii o danom systéme. Rovnako je dôležitá aj nutnosť poznania a existencie počiatkových dát pri jednotlivých algoritmoch nastavovania a konfigurácie parametrov.

Pri praktickej realizácii niektorých hybridných štruktúr stačí poznať len základné počty a typy funkcií príslušnosti pre jednotlivé jazykové premenné, iné metódy vyžadujú znalosť pravidiel. V súčasnosti je pre využitie uvedenej metodiky vo vnorených systémoch pomerne veľký počet typov hybridných neuro - fuzzy systémov. Medzi najznámejšie patrí systém ANFIS (Adaptive Neuro - Fuzzy Inference System), ktorý je založený na fuzzy systémoch riadenia regulátora Sugenovho typu. Ide o transformáciu fuzzy regulátora do doprednej neurónovej siete.

Pri učení takejto siete dochádza aj k zmenám parametrov samotného algoritmu fuzzy riadenia. Po naučení siete je možné tieto parametre spätne získať alebo použiť naučenú sieť priamo na vytvorenie vnoreného algoritmu riadenia. Prvú možnosť zjednodušene znázorňuje obr. 30.





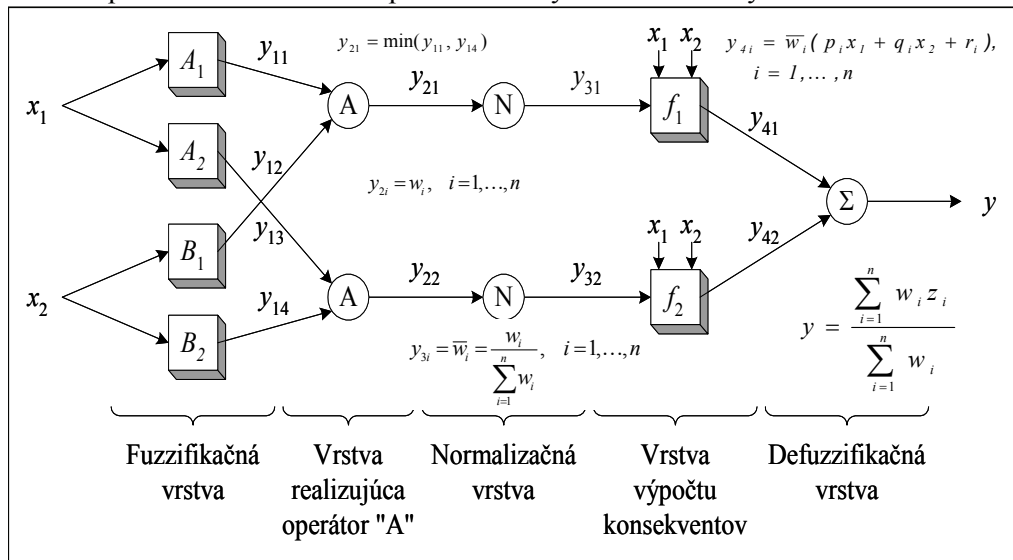
**Obr. 31.** Hybridná štruktúra konfigurácie parametrov fuzzy riadenia.

Neuro-fuzzy systém ANFIS (obr. 32) (Adaptive Network-based Fuzzy Inference System) patrí medzi najznámejšie hybridné štruktúry. Svojou architektúrou a metodikou vychádza z Takagi-Sugeno metódy návrhu fuzzy riadenia. Je reprezentovaný ako päťvrstvová dopredná neurónová sieť, ktorej učenie je založené na minimalizácii odchýlok metódou najmenších štvorcov na základe adaptácie tzv. premisných a konsekventných parametrov. Tieto parametre nie sú uložené vo váhach spomínanej neurónovej siete, ale v jej uzloch - preto je správne hovoriť o adaptívnej sieti. Návrh fuzzy riadenia - definícia funkcií príslušností a inferenčných pravidiel - je náročný a často veľmi zdĺhavý proces. FALCON (Fuzzy Adaptive Learning Control Network) je špeciálna neurónová sieť, ktorá má za cieľ tento návrh realizovať automaticky. Neuro-fuzzy systém FALCON môže po naučení fungovať ako fuzzy regulátor, alebo je možné z jeho vnútorných parametrov konfigurovať architektúru a parametre zodpovedajúceho fuzzy regulátora.

*ANFIS – hybridná inteligentná neuro - fuzzy štruktúra riadenia*

Základom ANFIS-u je Sugenov fuzzy inferenčný mechanizmus. Celá činnosť pri návrhu pozostáva z:

transformácie fuzzy systému do adaptívnej viacvrstvovej neurónovej siete,  
interpretácie rozhodovacích pravidiel fuzzy-neurónového systému.



**Obr. 32.** Štruktúra ANFIS-u pre dva vstupné premenné a jeden výstup.

Pravidlo:  $R_i : \text{ak } x_1 \text{ je } A_j \text{ a } x_2 \text{ je } B_k \text{ potom } y \text{ je } z_i = p_i x_1 + q_i x_2 + r_i,$   
 $i = 1, \dots, n \quad j \in \{1, \dots, m\} \quad k \in \{1, \dots, l\}$

kde  $m$ , resp.  $l$ , je počet fuzzy množín pokrývajúcich univerzá vstupných premenných  $x_1, x_2$ , resp.  $n$  je počet použitých pravidiel, pričom ich maximálny možný počet je rovný súčinu  $m \times l$ .

Pri použití defuzzifikačnej metódy singletonov je hodnota výstupnej premennej fuzzy systému váhovým priemerom výstupných hodnôt jednotlivých pravidiel

$$y = \frac{\sum_{i=1}^n w_i z_i}{\sum_{i=1}^n w_i} \quad w_i = \mu_{A_j}(x_1) * \mu_{B_k}(x_2)$$

(\* označuje operátor fuzzy prieniku súčin hodnôt funkcií príslušnosti vstupných premenných).

Jednotlivé vrstvy v systéme ANFIS realizujú nasledovné činnosti:

**1. vrstva** (fuzzifikácia)

**2. vrstva** (vykonáva fuzzy logickú operáciu „a“) - výstup z druhej vrstvy je priamo hodnotou váhového koeficientu  $i$ -tého pravidla,

$$y_{2i} = w_i, \quad i = 1, \dots, n$$

**3. vrstva** (normalizačná) - všetky neuróny v tejto vrstve sú nemenné a vykonávajú operáciu normalizácie váhových koeficientov jednotlivých pravidiel, vypočítaných v predchádzajúcej vrstve, podľa vzťahu

$$y_{3i} = \bar{w}_i = \frac{w_i}{\sum_{i=1}^n w_i}, \quad i = 1, \dots, n$$

**4. vrstva** (realizuje výpočet výstupov jednotlivých rozhodovacích pravidiel) - výstup z každého pravidla vo štvrtej vrstve je daný vzťahom

$$y_{4i} = \bar{w}_i (p_i x_1 + q_i x_2 + r_i), \quad i = 1, \dots, n$$

výpočet je vykonávaný v neurónoch 4. vrstvy, pričom vektor  $[p_i \ q_i \ r_i]^T$  je vektorom parametrov  $i$ -teho neurónu.

**5. vrstva (výstupná)** - počet neurónov v 5. vrstve je rovný počtu výstupných veličín fuzzy-neurónového systému, pričom každý z nich realizuje výpočet výslednej hodnoty príslušnej premennej ako súčet výstupov jednotlivých pravidiel, vynásobených normalizovanými váhovými koeficientmi, teda podľa vzťahu :

$$y = \frac{\sum_{i=1}^n w_i z_i}{\sum_{i=1}^n w_i}$$

*NEFCON (Neuro Fuzzy Controller)*

**NEFCON** je hybridný neuro-fuzzy systém založený na architektúre s názvom fuzzy perceptrón. Transformáciou fuzzy štruktúry riadenia do neurónovej siete má takto vzniknutý nový systém schopnosť rozširovať pravidlá a vyberať a meniť tvar funkcie príslušnosti podľa zadaného kritéria.

Fuzzy perceptrón je špeciálna trojvrstvová dopredná neurónová sieť. Prvá vrstva sa nazýva vstupná vrstva, druhá (skrytá) sa nazýva vrstva pravidiel a tretia sa nazýva výstupná vrstva. Synaptické spojenia medzi vrstvami pritom reprezentujú funkcie príslušnosti jednotlivých jazykových hodnôt (antecedentnú a konsekventnú časť).

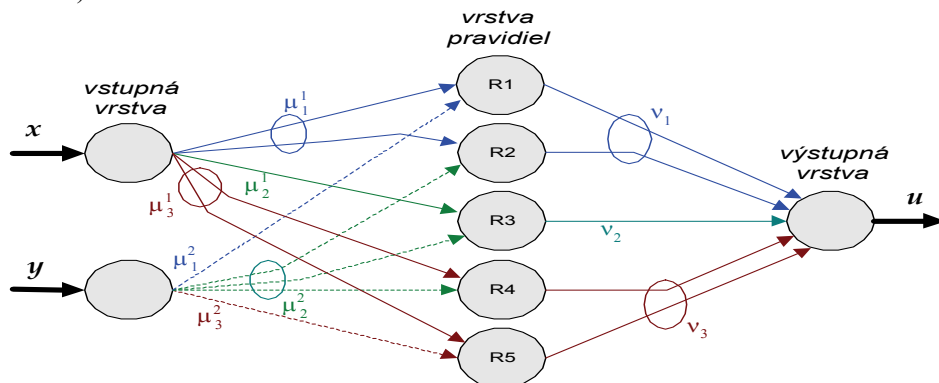
Na obr. 33 má hybridná sieť dva vstupy, jeden výstup (každý z nich má tri jazykové hodnoty) a päť pravidiel v tvare:

if  $x$  is  $A1$  and  $y$  is  $A2$  then  $u$  is  $B$

kde  $x, y$  sú vstupy do fuzzy regulátora reprezentujúce jazykové premenné a  $u$  je výstup z fuzzy regulátora.  $A1, A2, B$  sú fuzzy množiny - konkrétne jazykové hodnoty jazykových premenných reprezentované pomocou fuzzy množín s funkciami

príslušnosti  $\mu_1^1, \mu_2^1, \mu_3^1$  pre prvý vstup,  $\mu_1^2, \mu_2^2, \mu_3^2$  pre druhý vstup,  $\gamma_1, \gamma_2, \gamma_3$  pre výstup.

Štruktúra NEFCON-u na rozdiel od klasických neurónových sietí nemá synaptické spojenia definované ako reálne čísla, ale ako fuzzy množiny. Druhým veľkým rozdielom je aj to, že pri fuzzy perceptróne má viacero váh synaptických spojení rovnakú hodnotu (niektoré synaptické spojenia sú zdieľané pre viacero neurónov).



Obr. 33. Štruktúra hybridnej siete NEFCON-u.

Jednotlivé vrstvy hybridnej siete majú nasledovné vlastnosti:

Výstupná funkcia všetkých neurónov je definovaná ako identická funkcia

$$ou_i = f(in_i)$$

kde  $f(in_i)$  označuje aktivačnú funkciu.

1.vrstva

Táto vrstva sa nazýva vstupná vrstva. Každý jej neurón predstavuje jednu vstupnú jazykovú premennú. Neuróny tejto vrstvy iba priamo prenášajú vstupné signály na ďalšiu vrstvu, čiže vstupná aj aktivačná funkcia je definovaná ako identická.

$$\begin{aligned} in_i &= x_i \\ f(in_i) &= in_i \end{aligned}$$

kde  $x_i$  je hodnota  $i$ -tej vstupnej jazykovej premennej.

2.vrstva

Jej názov je vrstva pravidiel. Každý jej neurón predstavuje jedno fuzzy pravidlo. Celá vrstva predstavuje bázu všetkých pravidiel. Vstupná funkcia je definovaná ako T-norma vstupov prichádzajúcich do neurónu tejto vrstvy:

$$in_i = T(\mu_j^{(i)}(ou_i))$$

kde  $\mu_j^{(i)}(ou_i)$  je hodnota funkcie príslušnosti  $j$ -tej jazykovej hodnoty  $i$ -tej vstupnej jazykovej premennej s hodnotou  $ou_i$ ,

Aktivačná funkcia je definovaná ako identická.

$$f(in_i) = in_i$$

### 3. vrstva

Táto vrstva sa nazýva výstupná vrstva. Jej neurón predstavuje výstupnú jazykovú premennú. Tento neurón má vstupnú funkciu definovanú ako T-konormu (S-normu) svojich vstupov, čiže pravidiel s rovnakým konsekventom.

$$in = \perp (\mu_j)$$

Aktivačná funkcia je definovaná ako defuzzifikácia

$$f(in) = def\{in\}$$

Algoritmus učenia NEFCON-u používa ako základnú metódu spätného šírenia chyby (backpropagation) a môžeme ho rozdeliť na dva základné typy:

parametrické učenie – zmena tvarov a umiestnenia fuzzy množín,  
štruktúrne učenie – tvorba a modifikácia pravidiel.

Vlastnosti NEFCON-u sú určené heuristickou povahou jeho algoritmov. Na druhej strane je pomocou jednoduchej heuristiky dosiahnutý rýchlejší proces učenia, ako napr. u hybridného systému ANFIS. To predurčuje použitie NEFCON-u pri vnorených aplikáciách pracujúcich v reálnom čase.

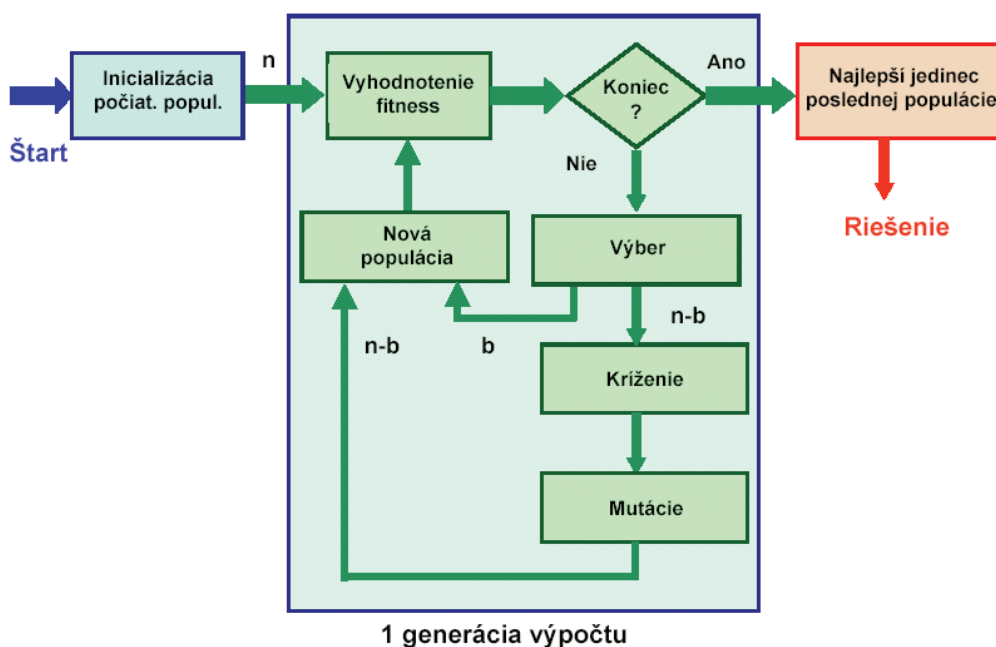
Veľkou výhodou tejto hybridnej štruktúry je jednoduchá implementácia známych pravidiel do NEFCON-u, čo umožňuje použiť apriórne informácie už pri jeho syntéze. Ďalšou výhodou NEFCON-u je jeho deterministický prístup výpočtu nasledujúceho akčného zásahu, hoci základ algoritmu je heuristický. Na rozdiel od klasických algoritmov s *reinforcement učením* regulátor nehľadá riešenie náhodnou zmenou akčného zásahu. Z toho vyplýva, že dve simulácie pri tých istých počiatkových podmienkach budú rovnaké, čo nemusí platiť pri spomínanej metóde učenia s posilňovaním alebo genetických algoritmoch.

**Genetické algoritmy** patria medzi moderné techniky riešenia zložitých úloh optimalizácie. Ich použitie je veľmi rôznorodé a aplikovateľné takmer vo všetkých oblastiach plánovania, rozhodovania, modelovania a riadenia. Výhodou je predovšetkým tá skutočnosť, že umožňujú nájsť globálny extrém v mnohých

problémoch na úkor dlhého času výpočtu, čo vylučuje ich použitie pre mnohé úlohy reálneho času.

Pri aplikácii genetického algoritmu proces hľadania riešenia generuje ďalšie body (možné riešenia), tak ako evolúcia pokračuje. Optimálne riešenia nájdené touto metódou sú často považované za dobré riešenia. Málokedy je však možné overiť skutočné optimum.

Algoritmus začína skupinou riešení (reprezentovanou chromozómami) nazývanou populácia. Riešenia z jednej populácie sú vyberané a používané na vytvorenie novej populácie. Toto je motivované tým, že nová populácia bude lepšia ako predošlá. Riešenia, ktoré sú neskôr vyberané na formovanie novej populácie, sú vyberané podľa ich vhodnosti. Čím sú vyhovujúcejšie, tým majú väčšiu pravdepodobnosť reprodukcie. Toto sa opakuje až do splnenia nejakej podmienky. Vývojový diagram GA je znázornený na obr. 34.



**Obr. 34.** Všeobecný vývojový diagram genetického algoritmu.

Hlavná myšlienka pri GA je výber lepších rodičov v nádeji, že lepší rodičia vytvoria lepších potomkov.

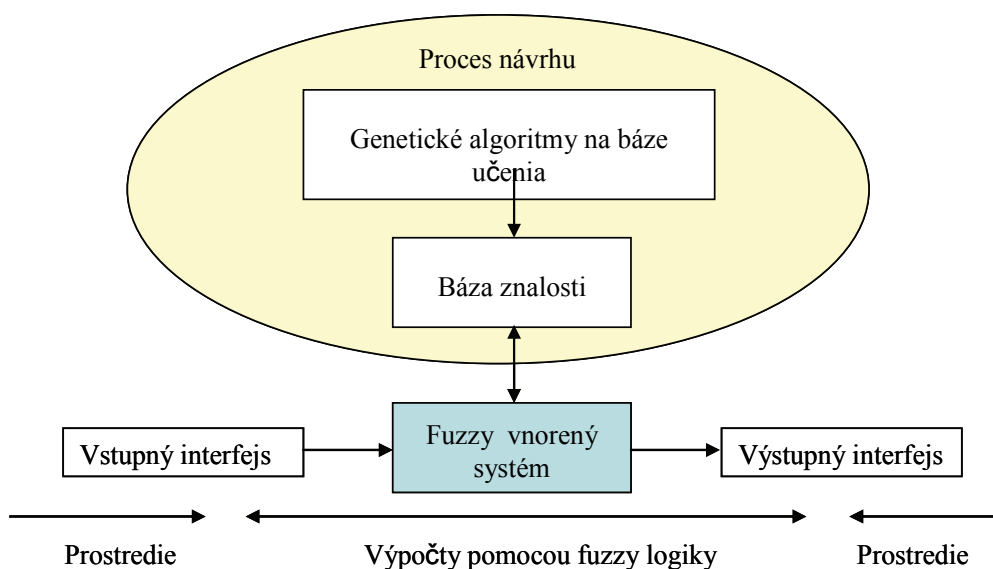
Proces prirodzenej evolúcie je jednoduchý, ale veľmi výkonný a robustný. Je univerzálne platný pre jednoduché aj zložité systémy. Základné vlastnosti GA:

Sú schopné vyviaznuť z okolia lokálneho extrémumu 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 riešeniam.



**Obr. 35.** Schéma použitia GA pri optimalizácii fuzzy modelu pre vnorené aplikácie.

GA ako univerzálny optimalizačný prostriedok (obr. 35) umožňuje v kombinovaných štruktúrach fuzzy-GA nájsť optimálnu štruktúru fuzzy vnoreného regulátora. Postup pri aplikácii GA pre optimalizáciu štruktúry a parametrov fuzzy vnoreného regulátora je nasledovný :

1. výber štruktúry riadenia,
2. výber metódy implikácie,
3. voľba počtu vstupov (napr. odchýlka, derivácia odchýlky), počtu výstupov (riadiaci zásah),
4. výber metódy defuzzifikácie,
5. určenie rozsahov univerzít pre vstupy a výstupy,
6. voľba tvarov a typov funkcií príslušnosti.

GA umožňuje optimalizáciu počtu funkcií príslušnosti, rozsahov, pozície a šírky. Hlavným výsledkom je optimálna štruktúra vnoreného algoritmu riadenia, ktorý po vložení do mikropočítača je možno zabudovať do vnoreného systému.

Optimalizácia fuzzy modelu pre realizáciu riadiacích a rozhodovacích algoritmov je realizovaná off-line vo vývojovom prostredí (napr. Matlab, Labview a pod.), z ktorých je možné zo simulačných schém vygenerovať kód aplikácie pre daný typ procesora.

### 2.3 Praktická realizácia inteligentných vnorených systémov

V posledných desiatich rokoch sa vo svete objavili rôzne praktické aplikácie inteligentného riadenia v rôznych oblastiach (priemysel, zdravotníctvo, spotrebná elektronika a pod.) na báze fuzzy logiky. Mnohé z týchto aplikácií boli realizované softvérovo, v minulosti implementované na osobných počítačoch. Takéto implementácie sú vhodné v prípade veľkého množstva aplikácií, ale sú nepoužiteľné v prípade vnorených systémov, napr. procesy v energetike, robotických systémoch a pod.

V súlade s najnovšími trendmi vývoja mikroelektroniky a požiadavkami používateľov v oblasti využívania vnorených počítačových systémov, v záujme vývoja a nasadzovania autonómnych a nezávislých RT systémov sa veľká podmnožina prístupov zameriava na použitie mikroprocesorov a špecializovaných čipov s integrovanými obvodmi (integrated circuit, IC), ako základných prvkov zabudovaných riadiacich systémov.

Ak majú mať zabudované systémy vlastnosti výpočtovej inteligencie nevyhnutnú pre interakciu s reálnym svetom, musia byť schopné zohľadňovať a vysporiadať sa s neurčitostami. Takéto systémy riadenia musia byť schopné pracovať v reálnom čase a v kontexte, napr. mobilné systémy, tak by ich regulátory boli kompaktné a s malou hmotnosťou, pretože výpočtové zdroje nesie základná doska.

Zabudovanými vnorenými výpočtovými systémami sa nazývajú elektronické systémy zabudované do daného objektu, ktorých cieľom je ovládať a riadiť tak, aby boli splnené určité funkčné požiadavky v rámci predpísaných časových obmedzení (požiadavka reálneho času). Vnorené systémy sú v mnohých prípadoch súčasťou výrobkov, ktoré obsahujú mikroprocesor s implementovanou funkciou riadenia.

Počas činnosti vnorený systém riadenia neustále interaguje s vonkajším prostredím pomocou senzorov, ktoré monitorujú stav prostredia a aktuátorov (výkonových členov), ktoré tento stav ovplyvňujú, ovládajú a riadia. Vo všeobecnosti v niektorých situáciách môžu byť inherentné šumy v senzorových dátach, ku ktorým dochádza v dôsledku nedokonalosti senzorov alebo neurčitostí prostredia, príčinou nesprávnej činnosti systému. Spracovanie nespoľahlivých senzorických údajov môže viesť k situáciám, keď rovnaký riadiaci zásah je použitý v rôznych stavoch (ktoré sa len zdajú rovnaké). V závislosti od miery odlišnosti stavov môže takéto správanie viesť ku kritickej chybe samotného zabudovaného systému alebo jeho poslania.

Ako už bolo uvedené v predchádzajúcej časti na efektívne spracovanie a zohľadnenie neurčitosti v reálnych prostrediach je potrebné použiť niektoré nové techniky aproximatívneho uvažovania. Niektoré z užitočnejších aproximatívnych techník uvažovania sú založené na pravdepodobnostných metódach, symbolickej umelej inteligencii, binárnej logike a viachodnotovej logike (fuzzy logike).

Ďalej sa zameriame na syntézu fuzzy vnorených algoritmov pri rešpektovaní neurčitostí, s ktorými sa stretávame pri vnorených systémoch a ich HW a SW realizácií. Základom fuzzy riadenia je spracovanie určitejšej informácie a úspora energie použitím pravidiel a rozhodnutí prostredníctvom prirodzeného jazyka.

V posledných rokoch mnohé priemyselné aplikácie dokázali svoju užitočnosť a praktický prínos riadenia na báze fuzzy logiky. Mnohé aplikácie fuzzy algoritmov riadenia procesov boli ako softvérové algoritmy riadenia implementované na osobných



počítačoch a PLC systémoch. Takéto implementácie sú výhodné pre mnohé väčšie aplikácie, nie však pre zabudované riadenie.

Na využitie aproximatívneho uvažovania do zabudovaných systémov sú potrebné niektoré špecifické techniky na realizáciu vnorených (zabudovaných) fuzzy regulátorov.

Zvýšený záujem o praktické riadenie na báze logiky bol vyvolaný najmä vyvinutím fuzzy čipov v polovici 80. rokov. Tvorcami prvých fuzzy čipov boli Togai and Watanabe a Yamakawa. Hlavným pojmom používaným v súvislosti s fuzzy čipmi je tzv. fuzzy aproximácia alebo interpolácia, t.j. interpolatívne uvažovanie.

Jednúčelové fuzzy regulátory sa implementujú softvérovo alebo hardvérovo podľa obmedzení návrhu daných cieľovým systémom. Podľa očakávania sú hardvérové implementácie v porovnaní so softvérovými vo všeobecnosti o niekoľko rádov rýchlejšie. Príčinou je sekvenčné spracovanie informácií vo fuzzy softvérových regulátoroch, kým v hardvérových prebieha spracovanie paralelne. Softvérové implementácie fuzzy regulátorov realizované na PC v slučke potrebujú na inferencovanie rádovo milisekundy. Hardvérové implementácie realizujú spracovanie vyššími rýchlosťami, sú kompaktnéjšie a vo všeobecnosti lacnejšie. Poukážeme na vybrané realizácie vnorených fuzzy regulátorov zabudované v zariadeniach.

#### *HW a SW riešenia vnorených inteligentných systémov*

Jedným z osvedčených prístupov k realizácii vnorených systémov je využitie konvenčných mikroprocesorov. Napr. Motorola Corporation vyvinula softvér na spracovanie inferencie fuzzy logiky a vývojový nástroj s názvom Knowledge Base Generator (KBG). KBG prekladá pravidlá a funkcie príslušnosti z prirodzeného jazyka do dátových štruktúr zrozumiteľných pre „Motorolové“ konvenčné mikrokontrolérové jednotky (MCUs). V súčasnosti podporované mikrokontroléry pozostávajú z rodiny tvorenej MC68HC11 a MC68HC05 MCUs. Zdrojový kód pre tieto viacúčelové fuzzy inferenčné nástroje je dostupný ako freeware od Motoroly. Všetky tieto nástroje uľahčujú proces integrácie fuzzy logiky do vnorených riadiacich systémov. Použitím tohto softvérového prístupu sa dajú navrhnuť vnorené fuzzy inferenčné mechanizmy, napr. so 4 vstupmi a 2 výstupmi, ktoré vyžadujú menej než 1 Kbyte pamäti.

Bohatšie prostredie pre vývoj fuzzy systémov poskytuje FIDE (Fuzzy Inference Development Environment) od firmy Apronix, Inc. Popri prepracovanej grafike vývojového a simulačného prostredia dokáže FIDE prekladať fuzzy inferenčné pravidla do RT kódu (assembler) kompatibilného s Motorolou MC68HC11, MC6805 a MC68HC05 MCUs. Tento RT kód je priamo použiteľný pre montáž a jeho implementáciu do cieľového systému pre účely zabudovaného riadenia.

V súčasnosti sú realizované na syntéze zabudovaného fuzzy riadiaceho systému, napr. pre mobilný robot, pomocou FIDE. Cieľovým systémom je malý mobilný robot s mikrokontrolérom MC68HC11 na základnej doske. Zabudovaný fuzzy regulátor pozostáva z bázy pravidiel, ktorá fuzzifikuje univerzum a generuje defuzzifikované riadiace zásahy. Riadiacimi zásahmi sú ostré hodnoty predstavujúce rýchlosť robota a riadiace smery .

Mobilný robot využíva informácie o svojom okolí, poskytované rozsahovými senzormi umiestnenými vpredu a po bokoch vozidla. V tejto aplikácii vnorený fuzzy

regulátor má vstupné údaje o čelnom, pravom a ľavom prekážkovom rozsahu poskytované senzormi; výstupmi sú smery natočenia a rýchlosti motora. Na vyjadrenie možnosti vstupných a výstupných hodnôt sa používajú lichobežníkové funkcie príslušnosti. Lichobežníkové tvary sa v mikroprocesorových aplikáciách používajú najčastejšie, pretože nevyžadujú použitie zložitých matematických operácií, ktorých realizácia by na lacných mikrokontroléroch bola obtiažná a časovo náročná.

Ďalšia aplikácia od firmy Apronix Inc. demonštruje na príklade návrhu regulátora pre dvojestupňové inverzné kyvadlo relatívnu jednoduchosť implementácie procesu návrhu použitím FIDE. Fuzzy vnorený regulátor bol systém s dvoma vstupmi a jedným výstupom s 30 pravidlami zabudovaný v 8-bitovom MC68HC11, vyžadoval 870 bytov pamäte a čas riadiaceho cyklu tesne pod 5 ms.

Pri alternatívnej realizácii sa môžu použiť jednoúčelové fuzzy mikrokontroléry ako napr. Togai jednodoskový fuzzy regulátor (single-board fuzzy controller, SBFC) alebo fuzzy mikroregulátor NLX-230 od NeuraLogix Corporation. SBFC je vybavený štandardným Togai fuzzy čipom, A/Č a Č/A čipmi a 230 štandardnou zbernicou a ďalšími obvodmi rozhrania. Dodávaný je softvér na preklad fuzzy expertného riadiaceho systému vyvinutého použitím Togai Fuzzy-C do strojového kódu pre procesor použitý na základnej doske. Hardvér umožňuje podporiť viac než 800 pravidiel a vyhodnotiť do 200 000 fuzzy pravidiel za sekundu. SBFC bol použitý v laboratórnych podmienkach na riadenie systému pásového dopravníka v aplikácii robotickej montáže. Podobne má fuzzy mikrokontrolér NLX-230 príslušný softvér napísaný v jazyku C, ktorý užívateľovi umožní nastaviť funkciu príslušnosti a pravidiel. Fuzzifikácia ako aj spracovanie pravidiel sa vykonávajú paralelne, pričom v čipovej pamäti môže byť uložených do 64 pravidiel. Fuzzy mikrokontrolér NeuraLogix bol tiež použitý v CAD laboratóriu na riadenie aplikácií v reálnom čase (napr. na riadenie už spomenutého pásového dopravníka).

Hoci zabudované fuzzy mikrokontroléry, ktoré vykonávajú inferenciu rádovo v milisekundách sú postačujúce pre mnohé praktické aplikácie riadenia, existujú ešte stále problémy riadenia vyžadujúce rýchlejšie inferencovanie. V takom prípade môžu byť potrebné hardvérové fuzzy regulátory na báze špecializovaných IC.

#### *Vysokorýchlostné fuzzy čipy pre vnorené systémy*

Vývoj vysokorýchlostných fuzzy hardvérových systémov vyžaduje paralelné spracovanie na čipe použitím vyhradených fuzzy vstupov alebo zariadení. Mnohí výskumníci doposiaľ vyvinuli zákaznicke VLSI systémy, pričom využívali hardvér pre zabudované riadenie. Zabudované hardvérové riadiace systémy sa implementujú v digitálnom alebo analógovom režime.

Fuzzy inferencovacie čipy nedávno boli použité na navigáciu mobilného robota. Ich dosky kompatibilné s VME zbernicou môžu byť konfigurované tak, aby umožnili plné inferencie s využitím do 350 pravidiel a 28 vstupov s časom odozvy 30 ms. Táto rýchlosť sa dosahuje vďaka paralelnému spracovaniu pravidiel na čipe. Uvažovanie v reálnom čase pri takýchto rýchlostiach uľahčuje reaktívne/reflexné riadenie autonómnych systémov na báze senzorov. Sofistikované hardvérové fuzzy regulátory môžu byť tiež realizované použitím čipov s pravidlami a defuzzifikačných čipov vyvinutých firmou Yamakawa. Čipové obvody pozostávajú z obvodov funkcií

príslušnosti, MIN/MAX obvodov a defuzzifikačných obvodov (prostredníctvom ťažiska).

Každý „pravidlový“ čip (čip s pravidlami) spracúva súčasne 4 pravidlá. Čas odozvy fuzzy regulátora je 670 nanosekúnd nezávisle od počtu pravidiel. Pretože tieto čipy pracujú v analógovom režime, nie sú potrebné A/Č prevody. Priame spracovanie analógových signálov umožňuje získavať vysokorýchlostné údaje. Jedna z aplikácií využívala fuzzy regulátor pozostávajúci z 11 pravidlových čipov a 1 defuzzifikačného čipu použitý na stabilizáciu inverzného kyvadla s pohárom naplneným vínom pripevneným ku kyvadlu. Rovnaký postup bol použitý s pohybujúcou sa myšou umiestnenou na plošine pripevnenej na konci kyvadla.

Vnorené fuzzy regulátory, ktoré majú v regulačnej slučke zahrnuté rozlišovanie vzorov, môžu byť realizované pomocou fuzzy neurónových čipov od Yamakawa. Tieto majú implementovaný fuzzy neurónový model, v ktorom sú synaptické váhy a sumácia konvenčných neurónových modelov modifikované fuzzy funkciami príslušnosti a operáciou MIN z teórie fuzzy množín. Výstupom fuzzy neurónu je hodnota v rozpätí 0 – 1, ktorá reprezentuje stupeň zhody so vzorom. Hoci sú čipy určené pre vysokorýchlostné rozpoznávanie vzorov, riadenie aplikácií je možné realizovať použitím fuzzy neurónových čipov v kaskáde s pravidlovými a defuzzifikačnými čipmi.

Typická architektúra na úrovni čipu predstavuje príklad aplikácie pre bezkolíznu navigáciu mobilného robota. Znamená to, že vzory môžu byť odvodzované z rozsahových senzorov umiestnených nad periferiou robota, každý fuzzy neurónový čip je možné naprogramovať na rozpoznávanie rôznych konfigurácií prekážok. Pravidlá potom môžu mapovať situáciu s prekážkami na vhodný príkaz riadenia pohybu, ktorý v každom okamihu zabráni kolízii.

#### *FPGA obvody pre realizáciu inteligentných vnorených systémov*

Jedným z prístupov realizácie IVS je Fuzzy CoCo, ktorá je evolučnou technikou založená na kooperatívnej koevolúcii na tvorbu presných a interpretovateľných fuzzy systémov.

Existujú tri prístupy v implementácii fuzzy systémov: riešenia na báze mikroprocesora (softvérový), jednoúčelové (dedicated) ASIC a na báze FPGA (Field Programmable Gate Array). Programovateľné hradlové polia (Field Programmable Gate Array – FPGA) predstavujú konfigurovateľné obvody vysokého stupňa integrácie (VLSI), ktoré sú schopné v sebe integrovať rôzne logické funkcie. V roku 1984 boli tieto obvody s konfiguračnou pamäťou založenou na technike SRAM uvedené na trh so zámerom umožniť prípadné vylepšovanie nakonfigurovaného programovateľného obvodu. V súčasnosti poskytujú obvody FPGA aj možnosť rekonfigurovania obvodov počas jeho chodu, v reálnom čase (tzv. dynamická rekonfigurácia). Na rekonfiguráciu je možno pozerať z dvoch rôznych pohľadov. Prvým je možnosť rekonfigurácie hardvéru, druhým je podpora dynamickej rekonfigurácie pomocou prostriedkov výpočtovej techniky (vývojové nástroje, CAD nástroje a pod.).

Obvody typu FPGA majú z programovateľných obvodov najvšeobecnejšiu štruktúru a obsahujú najviac logiky. Súčasný najväčší obvod FPGA obsahuje až 6-8 miliónov ekvivalentných hradiel (dvojjvstupové hradlo NAND).

Najväčšia flexibilita sa dá dosiahnuť softvérovou špecifikáciou plného systému, fuzzy systémy však predstavujú vysoko paralelné systémy a riešenia na báze mikroprocesorov fungujú horšie v porovnaní s hardvérovými verziami. Jednúčelová ASIC je najlepším riešením na dosiahnutie dobrej výkonnosti, avšak tento prístup dramaticky znižuje adaptabilitu systému. Systémy na báze FPGA poskytujú aj vyššiu výkonnosť pri paralelných výpočtoch v porovnaní so softvérovými riešeniami a lepšiu flexibilitu oproti ASIC vďaka vlastnosti dynamickej čiastočnej rekonfigurácie (DPR).

## 2.4 Záver

Táto kapitola mala za cieľ špecifikovať praktické možnosti využitia metodológie techník a poznatkov kognitívnej vedy a s ňou súvisiacich metód výpočtovej techniky pre aplikáciu vo vnorených aplikáciách. Neustály vývoj SW a HW pre vnorené aplikácie vyžaduje dnes inteligentné riešenia predovšetkým v problémoch ovládania, riadenia a optimalizácie činnosti procesov a zariadení v oblasti priemyslu, zdravotníctva, biotechnológiách a pod. Metódy výpočtovej inteligencie založené na fuzzy logike, umelých neurónových sieťach a genetických algoritmoch dokazujú schopnosť a nutnosť absorbovania poznatkov kognitívnej vedy a zabudovanie procesov myslenia, uvažovania a výberu rozhodnutí pre optimalizáciu činnosti procesov vo forme vnorených algoritmov. Táto kapitola mala preto za dominantný cieľ ozrejmiť a načrtnúť možnosti využitia inteligentných metód (softvérovými a hardvérovými prístupmi) vo vnorených aplikáciách, ktoré sa v súčasnosti vyvíjajú, overujú a implementujú pre rôzne typy procesorov a aplikácií v laboratóriu vnorených systémov, zriadených tak pre výučbu, ako aj pre výskum v tejto predmetnej oblasti na FIIT STU v Bratislave. Výsledky a získané poznatky budú obsiahnuté v pripravovanej druhej časti monografie.

*Podakovanie:* Táto časť publikácie vznikla s podporou grantu VEGA 1/0822/08.

## Literatúra

- [1] LAGO, E., C. JIMENEZ, J., LOPEZ, D. R., SANCHEZ-SOLANO, S., and BARRIGA, A., XfVHDL: A tool for the synthesis of fuzzy logic controllers, in *Design Automation and Test in Europe Conference and Exhibition*. ACM/IEEE, 1998, pp. 102.107.
- [2] LI, J., and HU, B. S.: The architecture of fuzzy PID gain conditioner and its FPGA prototype implementation. In *2nd International Conference on ASIC*. IEEE, 1996, pp. 61.65.
- [3] SANCHEZ-SOLANO, S., SENHADJI, R., CABRERA, A., BATURONE, I., JIMENEZ, C. J., and BARRIGA, A.: Prototyping of fuzzy logic-based controllers using standard FPGA development boards. In *13th International Workshop on Rapid System Prototyping*. IEEE, 2002, pp. 25.32.
- [4] TIPSUWANPORN, V., RUNGHIMMAWAN, T., INTAJAG, S., and KRONGRATANA, V.: Fuzzy logic PID controller based on FPGA for process

- control,. in *International Symposium on Industrial Electronics*. IEEE, 2004, pp. 1495.1500.
- [5] VUONG, P. T., MADNI, A. M., and VUONG, J. B: VHDL implementation for a fuzzy logic controller,. in *World Automation Congress*,2006.
- [6] KIM, D.: An implementation of fuzzy logic controller on the reconfigurable FPGA system,. *IEEE Transactions on Industrial Electronics*, vol. 47, no. 3, pp. 703.715, 2000.
- [7] OPB HWICAP Product Specification v1.3, Xilinx, 2004.
- [8] COMPTON, K., HAUCK, S.: Reconfigurable computing: A survey of systems and software,. *ACM Computing Surveys*, vol. 34, no. 2, pp. 171.210, 2002.
- [9] HOLIŠ, M., KACHAŇÁK, A.: Optimalizácia kontinuálnych procesov pomocou hybridných neuro – fuzzy systémov, Ref. na 5. medzinárodnej ved. konf. Strojné inžinierstvo 2001, Bratislava Sjf STU.
- [10] KACHAŇÁK, A., HOLIŠ, M.: Optimalizácia a automatizácia v dispečerskom riadení vykurovania, 13. medzinárodná konf. Proces Control 2001, CD-ROM.
- [11] KVASNIČKA, V., TREBATICKÝ, P., POSPÍCHAL, J., KELEMEN, J.: Mysel', inteligencia a život, STU Bratislava, 2007.
- [12] QING, Li: Real-Time Concepts for Embedded Systems. CMP Books, 2003.
- [13] SROVNAL, V.: Operační systémy pro řízení v reálném čase. VŠB – Technická univerzita, Ostrava, 2003.
- [14] SRIDHAR, T.: Designing Embedded Communications Software. CMP Books, 2003.
- [15] SROVNAL, V.: Přehled operačních systémů reálného času. *AT&P Journal*, October 2005, Volume 12, Issue 10-12, pp. 51–53, pp. 64–65, pp. 56, ISSN 1335-2237.



# Evolučný návrh robotických organizmov

Peter KRČAH<sup>1</sup>

**Abstrakt.** Roboty prenikajú stále viac a viac do nášho každodenného života, kde musia plniť čoraz zložitejšie úlohy. Klasický prístup – ručný návrh napevno naprogramovaných robotov – naráža v neustále sa meniacom prostredí na svoje hranice. Jedným z možných prístupov k riešeniu tohto problému je hľadanie inšpirácie v prírode v evolúcii živých organizmov. Táto kapitola poskytuje stručný prehľad existujúcich metód evolučného návrhu robotických organizmov.

## 1 Úvod

Evolučná robotika je technika navrhovania autonómnych robotov pomocou metód inšpirovaných evolúciou živých organizmov. Prečo používať na návrh robotov práve evolučné algoritmy? Najväčšou inšpiráciou je samotná živá príroda. Nesmierna rôznorodosť, zložitosť a účelnosť organizmov, ktorú vidíme v prírode okolo nás, nám dáva nádej, že raz budeme schopní vytvárať prístroje podobnej zložitosti automaticky pomocou evolúcie simulovanej v počítači. Samotné evolučné algoritmy sú síce mladou disciplínou, boli už však úspešne aplikované na veľké množstvo problémov v rôznych oblastiach (od návrhu elektrických obvodov [28], cez návrh algoritmov pre kvantové počítače [47] až po riešenie ťažkých algebraických problémov [49]) a v niektorých prípadoch dokonca dosahujú v danej oblasti lepšie výsledky, ako najlepšie riešenie navrhnuté človekom [2, 28]. V prípade evolučného návrhu autonómnych robotov je však podobnosť so živou prírodou asi najzreteľnejšia. Autonómne roboty majú so živými organizmami mnoho spoločného: oba musia samostatne plniť rôzne úlohy vo svete riadenom fyzikálnymi zákonmi, oba majú telo a často sadu končatín a v oboch prípadoch môže byť predmetom optimalizácie tak stavba tela, ako aj ovládanie organizmu/robotu. Tieto podobnosti robia z robotiky veľmi lákavú oblasť pre aplikáciu evolučných algoritmov. Zaujímavým dôsledkom tejto podobnosti je, že simulovaná evolúcia často dospeje k podobným riešeniam predložených problémov ako evolúcia v prírode (príkladom sú výsledky práce Karla Simsa [45]).

Termín *evolučná robotika* bol použitý prvýkrát v roku 1993 v práci Clifffa a kol. [7]. Vývoj evolučnej robotiky začal z dvoch rôznych smerov. Prvým z nich bol

---

<sup>1</sup> Ústav výpočetní techniky, Univerzita Karlova v Prahe, Ovocný Trh 5, 116 36 Praha 1, Česká republika, E-mail: peter.krcah@ruk.cuni.cz

vývoj ovládania pre jednoduché reálne roboty s nemennou konštrukciou (Khepera, e-puck). Výskum v tomto smere priniesol viaceré zaujímavé výsledky nielen v oblasti riadenia robotov, ale aj pri riešení problémov spojených s prechodom od simulácie k reálnym robotom [40]. Druhým smerom bol vývoj robotov čisto v simulácii, prístupujúci k problému skôr z perspektívy umelého života. Tým, že sa projekty v tomto prípade často nepokúšali konštruovať roboty v realite, mohli si dovoliť navrhovať roboty s omnoho bohatšími vlastnosťami a vyvíjať evolúciu aj samotnú konštrukciu robota a nielen jeho ovládanie. Vlastnosti robota a spôsob merania jeho úspešnosti (tzv. *fitness funkcia*) boli v tomto prípade obmedzované často len fantáziou autora a obmedzeniami fyzikálnej simulácie. Prvým prelomovým projektom v tejto oblasti boli virtuálne organizmy Karla Simsa uvedené v roku 1994 [45], ktoré svojou podobnosťou organizmom zo živej prírody dodnes poskytujú inšpiráciu výskumníkom v oblasti evolučnej robotiky. V poslednej dobe sa tieto dva smery začínajú stretávať v projektoch, ktoré sa pokúšajú o konštrukciu kompletného robota v realite, na základe návrhu optimalizovaného evolučným algoritmom [22, 33, 56].

Cieľom tejto kapitoly je bližšie popísať najrozšírenejšie z metód používaných pri návrhu autonómnych robotov pomocou evolučných algoritmov a popísať niektoré zaujímavé dosiahnuté výsledky. Na úvod stručne uvedieme základné pojmy používané v evolučných algoritmoch a hlavné druhy evolučných algoritmov. Potom sa budeme venovať rôznym prístupom k ovládaniu robotov (či už simulovaných alebo reálnych) a problému, ako toto ovládanie vyvíjať evolúciou. V ďalšej sekcii popíšeme problém reprezentácie robota v evolúcii a ukážeme, že voľba správnej reprezentácie môže mať pozitívny vplyv na rýchlosť evolúcie a úspešnosť výsledných organizmov. Ďalej popíšeme rôzne prístupy k vyhodnocovaniu úspešnosti robota v evolúcii, vrátane techník, ako je interaktívna evolúcia, koevolúcia a evolúcia virtuálnych ekosystémov. Na záver sa budeme venovať rôznym prístupom k problému prechodu od simulácie do reality a popisu niektorých konkrétnych projektov, ktorým sa tento prechod podarilo úspešne uskutočniť.

## 2 Základy evolučných algoritmov

Evolučné algoritmy sú optimalizačné algoritmy motivované biologickou evolúciou. Evolučná teória bola sformulovaná Charlesom Darwinom v roku 1859 a odvtedy sa stala základným nástrojom pre pochopenie biologických procesov a histórie organizmov na Zemi [10, 11]. Základné myšlienky evolúcie však nie sú viazané len na živé organizmy, ale dajú sa aplikovať na ľubovoľné „organizmy“ spĺňajúce nasledovné tri podmienky:

- organizmy musia byť schopné reprodukcie a schopné prenášať svoje vlastnosti na svojich potomkov,
- musí existovať zdroj variability v populácii (spôsobený napríklad chybným prepisom informácie z rodiča na potomka),
- a musí existovať obmedzenie (*selekčný tlak*), ktoré zabezpečí, že neprežívajú všetky organizmy, ale len tie s určitými vlastnosťami.



Prvé úspešné pokusy simulovať evolúciu v počítači a použiť ju ako stochastický optimalizačný algoritmus sa objavili v osemdesiatych rokoch minulého storočia. Priekopníkmi boli najmä John Holland, ktorý predstavil tzv. genetický algoritmus [20] a Ingo Rechenberg [42], ktorý vypracoval podobnú techniku s názvom evolučné stratégie. Od tej doby vzniklo veľké množstvo techník inšpirovaných evolúciou a aj inými biologickými procesmi, ako napríklad genetické programovanie (ktoré bližšie popíšeme v sekcii o ovládaní robotov). Najčastejšie používanou technikou pri evolúcii robotických organizmov však zostáva genetický algoritmus (GA), a preto sa v ďalšom texte budeme venovať najmä jemu.

Hlavnou myšlienkou GA je nahliadnuť na množinu všetkých prípustných riešení daného problému ako na množinu organizmov v nejakom virtuálnom prostredí. Vybrané organizmy (riešenia problému) súperia o miesto v populácii, ktorá má typicky vopred danú pevnú veľkosť. Prírodný výber je simulovaný tzv. *fitness funkciou*, ktorá meria úspešnosť organizmu v prostredí. Fitness funkcia je definovaná tak, aby zvýhodňovala organizmy, ktoré reprezentujú lepšie riešenie problému. Nová generácia je zostavená z potomkov organizmov z predchádzajúcej generácie, pričom pri výbere rodičov sú preferované organizmy s vyššou hodnotou fitness funkcie. Variabilita populácie je udržiavaná operátorom mutácie, ktorý náhodne pozmení niektorých nových potomkov. Genetické algoritmy často okrem operátora mutácie používajú operátor kríženia, ktorý simuluje sexuálnu reprodukciu a umožňuje výmenu génov medzi organizmami v priebehu evolúcie. Algoritmus končí typicky buď

- po vopred danom pevnom počte opakovaní,
- alebo v momente, keď je nájdené riešenie s dostatočne vysokou hodnotou fitness funkcie,
- alebo keď algoritmus dlhú dobu stagnuje (fitness najlepšieho organizmu v populácii nerastie alebo rastie iba málo počas viacerých generácií).

Pseudokód jednoduchého genetického algoritmu teda môže vyzeráť napríklad takto:

```
g = 0; // čítač generácií
Inicializuj populáciu P(g);
Ohodnoť populáciu P(g); // t.j. vypočítaj hodnoty fitness
while neplatíPodmienkaUkončenia() do
    g = g + 1;
    Vyber P(g) z P(g-1);
    Kríženie P(g);
    Mutácia P(g);
    Ohodnoť P(g);
End
```

Pôvodný genetický algoritmus pracuje s organizmami reprezentovanými bitovými reťazcami pevnej dĺžky. Pre GA s touto reprezentáciou organizmov bolo odvodených viacero zaujímavých teoretických dôsledkov, ktoré ukazujú, že evolúcia môže byť veľmi efektívnym spôsobom prehľadávania priestoru možných riešení s nižším rizikom uviaznutia v lokálnom optime, než je tomu u iných metód. Holland odvodil niektoré základné teoretické výsledky pomocou teórie schém, ktoré ponúkajú

vysvetlenie efektivity GA. Tieto výsledky však presahujú rámec tejto kapitoly (pre záujemcov je viac informácií možné nájsť napríklad v [17]). Presné efekty mutácie a kríženia organizmov na efektívitu evolúcie však ešte nie sú úplne teoreticky pochopené a skúmanie vlastností evolučných algoritmov je stále odkázané vo veľkej miere na experimenty.

Genetické algoritmy majú však niekoľko vlastností, ktoré ich robia cennou technikou v oblasti optimalizačných algoritmov:

- Samotný algoritmus je jednoduchý na implementáciu a pochopenie a je vo veľkej miere nezávislý na úlohe, ktorú rieši. Pri jeho použití na riešenie konkrétneho problému stačí navrhnuť reprezentáciu riešenia (spolu s genetickými operátormi) a zvoliť vhodnú fitness funkciu.
- Pri použití GA nie sú vyžadované žiadne špeciálne znalosti o chovaní fitness funkcie. GA je schopný efektívne hľadať riešenia aj pre komplikované fitness funkcie s mnohými lokálnymi extrémami. Toto umožňuje použiť genetické algoritmy na riešenie veľmi rozmanitých problémov.
- GA sú špeciálne vhodné pre problémy s veľmi veľkým prehľadávaným priestorom riešení, ktorý nie je možné prehľadávať klasickými metódami (ako je napríklad prehľadávanie do hĺbky).
- GA je typicky schopný veľmi rýchlo nájsť *nejaké* dobré riešenie problému (konkrétne chovanie závisí najmä na vlastnostiach konkrétnej fitness funkcie).

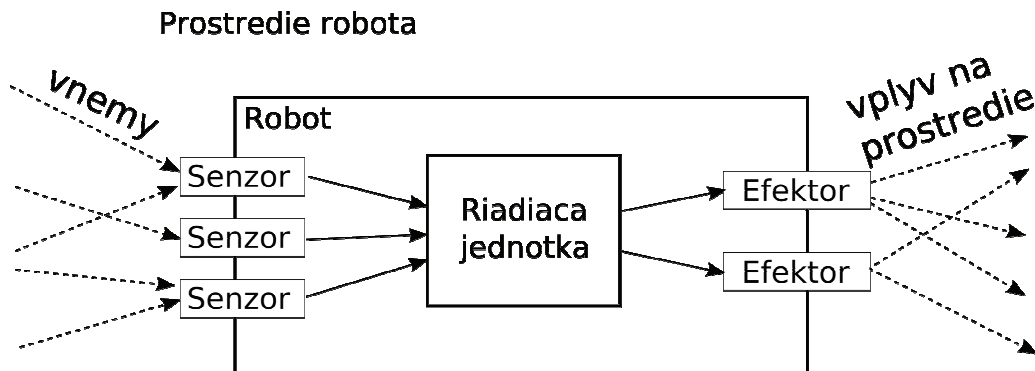
Genetické algoritmy však majú aj viacero nevýhod:

- Návrh fitness funkcie nemusí byť jednoduchý. Nesprávne navrhnutá fitness funkcia môže spôsobiť, že GA konverguje veľmi pomaly k riešeniu, alebo dokonca preferuje riešenia, ktoré síce dosahujú vysokú fitness, ale nereprezentujú dobré riešenie problému.
- Výpočet môže byť časovo veľmi náročný, pretože algoritmus typicky vyžaduje veľké množstvo vyhodnotení fitness funkcie organizmu. GA je však dobre paralelizovateľný (vyhodnocovania fitness funkcie pre jednotlivé organizmy sú navzájom nezávislé), takže problém časovej náročnosti sa dá čiastočne riešiť distribuovaním výpočtu fitness funkcie na viacero počítačov.
- Genetický algoritmus je zriedkavo schopný nájsť globálne najlepšie riešenie daného problému. Pri veľkom počte zložitých problémoch však tento nedostatok nevedí, pretože optimálne riešenie typicky nie je dosiahnuteľné ani žiadnou inou technikou.

V ďalšom texte si pri popise evolúcie robotov požičiame niektoré termíny z biológie. Pojmom *morfológia* budeme označovať „stavbu tela“ alebo konštrukciu robota, *genotyp* bude označovať genetickú informáciu, naopak pojem *fenotyp* budeme používať na označenie konkrétneho robota po jeho zostavení z jeho genotypu.

### 3 Metódy ovládania robotov

Pod pojmom robot budeme v ďalšom texte rozumieť samostatne pracujúci stroj (virtuálny alebo skutočný) vykonávajúci určené úlohy. Robot získava informácie o sebe a svojom prostredí pomocou *senzorov* (kamera, dotykový senzor, gyroskop, a pod.) a má možnosť spätne ovplyvňovať seba a svoje prostredie pomocou *efektorov* (napríklad pohyb servomotora). Samotné ovládanie robota zabezpečuje riadiaca jednotka, ktorá je schopná prijímať na svojich vstupoch informácie zo senzorov a ktorej výstupy nesú pokyny pre efekory robota (viď obr. 1).



**Obr. 1.** Schéma vzťahu robota a jeho prostredia.

Áké predpoklady kladieme pri evolúcii robotov na riadiacu jednotku? V prvom rade musí byť schopná vykazovať chovanie, ktoré je zodpovedajúce riešeniu danej úlohy. Veľké množstvo úloh skúmaných evolučnou robotikou je riešiteľné použitím riadiacej jednotky bez vnútorného stavu (t.j. jednotka si nemá možnosť uchovávať informáciu z predchádzajúcich krokov). Správanie robota je potom čisto reaktívne, t.j. robot na rovnaké podnety (informácie zo senzorov) odpovedá vždy tou istou akciou. Aj takéto na prvý pohľad jednoduché ovládanie však môže byť schopné riešiť prekvapivo zložité úlohy. Jednou z techník, ktorú môžu roboty s reaktívnym ovládaním použiť, je tzv. *senzorovo-motorická koordinácia* [40]. Táto technika spočíva v tom, že v situácii, keď robot nie je schopný rozhodnúť sa na základe aktuálnych informácií zo senzorov o ďalšom kroku, môže vykonať akciu, ktorá mu prinesie viac informácií a tým mu umožní vykonať rozhodnutie. Ak robot napríklad stojí pred objektom, ktorý však nevidí celý a nevie tak posúdiť, či sa k nemu priblížiť alebo sa od neho vzdialiť, môže situáciu často vyriešiť tým, že sa presunie na inú pozíciu, z ktorej objekt vidí lepšie, a to mu umožní vykonať rozhodnutie. Toto chovanie nevyžaduje vnútorný stav a bolo pozorované aj pri niektorých živých organizmoch [12].

Existujú však aj úlohy, kde sa roboty bez vnútorného stavu nezaobídu. Niektoré z týchto príkladov boli študované Nolfim a Floreanom [40], ktorým sa podarilo vyvinúť jednoduché roboty schopné riešiť úlohy zbierania odpadu alebo automatického hľadania zdroja energie využívajúc pri tom vnútorný stav. V druhom z menovaných príkladov dodatočná analýza neurónovej siete robota (viď sekcia 3.2) ukázala, že

výsledný robot si udržiaval vo svojej neurónovej sieti reprezentáciu prostredia, ktorú používal pri navigácii za zdrojom energie.

V evolučnej robotike sú používané na ovládanie robotov viaceré techniky, medzi inými neurónové siete, genetické programovanie, ale aj napríklad učiace sa systémy klasifikátorov (z angl. learning classifier systems) [30]. V nasledujúcom texte popíšeme niektoré z metód ovládania robotov detailnejšie.

### 3.1 Oscilujúce funkcie

Jedným z najjednoduchších spôsobov ovládania robotov je použitie sady jednoduchých oscilujúcich funkcií (ako napríklad sínus). Keďže väčšina jednoduchých motorických úkonov je periodická (ako plávanie, chodenie, plazenie a pod.) sú tieto úlohy často dobre riešiteľné len s použitím oscilátorov. Napríklad v práci Hornbyho a kol. [22] bola evolúciou vyvíjaná morfológia 3D robotov, ktorých pohyb bol popísaný výlučne oscilátormi (optimalizované boli v tomto prípade fáza aj frekvencia oscilátora). Podobný prístup je použitý v práci Simsa [45], kde je oscilátor jedným z typov prenosovej funkcie neurónu. K dosiahnutiu periodického chovania robota však nie je vždy nutné použitie explicitného oscilátora. Príkladom môže byť robot s jedným pántovým kĺbom, senzormerajúcim uhol tohto kĺbu a efektorom, ktorý aplikuje na kĺb silu požadovanej veľkosti. Priame prepojenie senzora a efektora (s opačným znamienkom) spôsobí, že sila bude pôsobiť vždy proti aktuálnemu vychýleniu, čo v prostredí bez trenia bude viesť k pravidelným osciláciám.

### 3.2 Neurónové siete

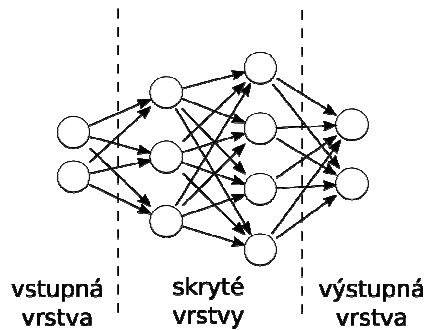
Jednou z najčastejšie používaných metód ovládania robotov sú neurónové siete. Umelé neurónové siete vznikli s ambíciou simulovať v počítači biologické neurónové siete, ktoré sú (ako vidíme v živej prírode) veľmi silným výpočtovým nástrojom schopným efektívne spracovávať veľké množstvá informácií. Neurónová sieť je tvorená navzájom prepojenou skupinou neurónov. Jeden z najjednoduchších modelov neurónu bol navrhnutý v roku 1943 McCullochom a Pittsom [37]. V tomto modeli je výstupom  $i$ -teho neurónu  $y_i$  funkcia ( $\phi$ ) sumy  $N$  vstupných signálov ( $x_1, x_2, \dots, x_N$ ) vážených silou neurónových spojov ( $w_{ij}$ ):

$$y_i = \phi\left(\sum_{j=1}^N w_{ij}x_j\right)$$

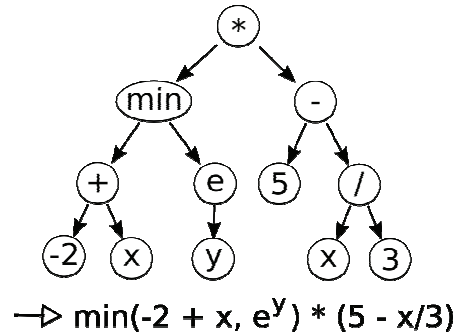
Prenosová funkcia ( $\phi$ ) môže mať rôzne formy; často používanou je napríklad sigmoidálna prenosová funkcia:

$$\phi(x) = \frac{1}{1 + e^{-kx}}$$

kde  $k$  je vhodná konštanta (typicky 1). Neuróny môžu byť usporiadané v sieti rôznymi spôsobmi. Architektúry neurónových sietí sa dajú rozdeliť na tzv. *dopredné* (neobsahujúce cykly) a *rekurentné* (môžu obsahovať cykly). Jedným z najčastejšie používaných usporiadaní je tzv. *dopredná vrstevnatá* neurónová sieť, kde



Obr. 2. Príklad doprednej vrstevnatej neurónovej siete.



Obr. 3. Reprézentácia genetického programu stromom.

sú neuróny rozložené v jednej vstupnej vrstve, ľubovoľnom počte vnútorných vrstiev a jednej výstupnej vrstve, pričom neurónové spoje vedú vždy len medzi susediacimi vrstvami smerom od vstupnej k výstupnej vrstve (príklad je na obr. 2).

Rekurentné neurónové siete si môžu práve vďaka existencii cyklov neurónových spojov dlhodobo uchovávať informáciu. Táto architektúra je preto vhodná na skúmanie robotov s vnútorným stavom, kým dopredné siete vedú k robotom s reaktívnym chovaním.

Hlavnou úlohou pri používaní neurónových sietí je nastaviť váhy jednotlivých spojov tak, aby robot ovládaný neurónovou sieťou riešil zadaný problém. Prehľad metód riešiacich túto úlohu je dostupný napríklad v [1]. Jedným z alternatívnych prístupov k optimalizácii váh siete je práve použitie evolučných algoritmov. Výskum evolúcie neurónových sietí prešiel od algoritmov optimalizujúcich váhy neurónových spojov siete s pevnou štruktúrou až ku algoritmom optimalizujúcim aj samotnú štruktúru siete (NEAT [51], GRN [43]). Neurónové siete sú veľmi často používané pri riadení robotov v evolučnej robotike, [45, 38, 39, 33] a to z viacerých dôvodov [40]:

- Priestor neurónových sietí (prehľadávaný evolučným algoritmom) je pomerne hladký. To znamená, že malé zmeny parametrov neurónovej siete (vyvolané napríklad mutáciou) často vedú k malým zmenám jej chovania.
- Adaptácia neurónovej siete môže byť vykonávaná na viacerých úrovniach: napríklad fylogenetickú úroveň (evolúciu), vývinovú (dospievaním) alebo ontogenickú (učenie počas života). Tieto úrovne môžu byť v konkrétnom algoritme rôzne kombinované.
- Pri ovládaní robota poskytujú neurónové siete priamočiary spôsob prepojenia senzorov a efektorov robota. Vstupom siete môžu byť spojité signály zo senzorov a výstupom môžu byť buď spojité, alebo diskrétné hodnoty pre efektor (podľa zvolenej fitness funkcie).
- Keďže výstup neurónu je váženou sumou vstupov, neurónové siete sú pomerne robustné voči šumu (oscilácie vstupov sú často vyhladené a nemajú dramatický vplyv na výsledok). Táto vlastnosť môže byť dôležitá pre roboty, ktoré sa často spoliehajú na zašumené hodnoty senzorov.

### 3.3 Genetické programovanie

Ďalším, často používaným prístupom, je genetické programovanie (GP). Cieľom genetického programovania je vyvíjať evolúciou počítačové programy [26]. Programy sú väčšinou v pamäti reprezentované stromovou štruktúrou, ktorá je vyhodnotená jednoduchým rekurzívnym priechodom (príklad programu je na obr. 3). Každý vnútorný uzol stromu má priradený operátor a každý list operand (premennú alebo konštantu). Reprezentácia programu v GP je tak svojou štruktúrou podobná napríklad funkcionálnemu programovaciemu jazyku Lisp. Sada operátorov (napr. +, -, \*, /) a sada operandov (napr. X, Y, Z) je vopred zadaná a často je ovplyvnená vlastnosťami riešenej úlohy. Pre takto definované programy boli navrhnuté viaceré metódy kríženia a mutácie. Mutácia môže napríklad nahradiť náhodne vybraný podstrom novým, náhodne vygenerovaným podstromom, alebo len náhodne upraviť vnútorné parametre uzlov. Kríženie typicky funguje jednoduchou zamenou náhodne vybraného podstromu v jednom jedincovi za náhodne vybraný podstrom v inom jedincovi. Pri úpravách programu je dôležité brať ohľad na vlastnosti operátorov (ako je napríklad arita) tak, aby výsledný strom aj po úprave reprezentoval korektný program.

Popísaný štandardný variant GP bol viacerými spôsobmi rozšírený. Jedným z významných rozšírení bolo napríklad zavedenie automaticky definovaných funkcií (ADF) [27]. Tieto funkcie reprezentujú celý jeden podstrom programu, ktorý tak môže byť znovu použitý na viacerých miestach v pôvodnom strome. GP boli použité viackrát úspešne na riešenie širokého okruhu problémov (za zmienku stojí napríklad návrh patentovateľných elektrických obvodov [28]), medzi inými aj v evolučnej robotike (príklady takýchto prác sú uvedené napríklad v [40]).

### 3.4 Ovládanie v hardvéri

Jednou z možností ako ovládať robota je použiť namiesto softvérových nástrojov priamo rekonfigurovateľné hardvérové komponenty. Príkladom tohto prístupu sú tzv. FPGA (Field Programmable Gate Array) obvody. FPGA je konfigurovateľný elektrický obvod zložený zo sady modulov (logických a vstupno/výstupných), ktoré sú usporiadané v pravidelnej mriežke. Pri konfigurácii FPGA obvodu je možné určiť, ako sú moduly v mriežke prepojené a pre každý logický modul je možné určiť logickú funkciu, ktorú modul vykonáva. Takáto konfigurácia môže prebiehať opakovane, čo robí z FPGA obvodov vhodnú platformu pre evolučné experimenty. FPGA obvody boli viackrát úspešne použité pri on-line evolúcii robotov, napríklad pri evolúcii robota Khepera vyhýbajúceho sa stenám [53].

Ako zaujímavosť uvedieme prevedenie myšlienky ovládania v hardvéri do extrému pri tzv. evolúcii *in materio*. Základnou myšlienkou tohto prístupu je využívať ako výpočtovú jednotku materiál s dostatočne zložitou vnútornou štruktúrou. Jedným z takýchto materiálov je tekutý kryštál. Ak do tekutého kryštálu zavedieme vstupné a výstupné elektrické kontakty, zmeny napätia na vstupných kontaktoch spôsobujú zmeny napätia na výstupných kontaktoch. Vzťah medzi vstupnými a výstupnými napätiami pritom závisí od vnútornej štruktúry materiálu. Evolúcia môže takto hľadať také zapojenie senzorov robota na vstup kryštálu, aby robot ovládaný výstupnými

hodnotami riešil zadanú úlohu. Metóda bola úspešne aplikovaná na evolúciu jednoduchej navigácie robota Khepera [19].

#### 4 Problém reprezentácie robota v evolučných algoritmoch

Hlavným cieľom evolučnej robotiky (a evolučných algoritmov ako takých) je automatizovať návrh robotov a priblížiť sa zložitou ich chovania a morfológie zložitosti živých organizmov. Evolúcia v prírode dokáže vyvíjať nesmierne zložité organizmy tvorené miliónmi buniek. Ako však dosiahnuť takú efektivitu evolučných algoritmov, aby boli schopné vyvíjať štruktúry takejto zložitosti v počítači? Ak by evolučný algoritmus mal prehľadávať takýto milión-dimenzionálny priestor všetkých možností priamo, stál by pred ťažko splniteľnou úlohou.

Pri hľadaní riešenia tohto problému si môžeme opäť vziať príklad z prírody samotnej. Pri evolúcii živých organizmov je jasne rozdelená genetická reprezentácia organizmu (*genotyp*) od fyzickej reprezentácie konkrétneho jedinca (*fenotyp*). Genotyp organizmu je používaný ako nosič informácie pri dedení a prebiehajú na ňom operácie ako mutácia a rekombinácia. Naproti tomu fenotyp označuje konkrétneho jedinca, ktorý vznikol vývinom v konkrétnom prostredí podľa predpisu uloženom v genotype. Práve v spôsobe „výroby“ fenotypu z genotypu zrejme leží odpoveď na otázku položenú v úvode. Použitie niektorého netriviálneho spôsobu prepisu genotypu na fenotyp môže napríklad umožniť jednoduchú duplikáciu častí tela organizmu a tým umožniť jeho opätovné použitie na iných miestach v tom istom organizme. Evolúcia tak môže znovu a znovu využívať elementy, ktoré sa jej raz podarilo nájsť zdĺhavým prehľadávaním bez toho, aby ich musela hľadať znovu (napríklad evolúcia nemusí vymýšľať koleso zakaždým, keď ho potrebuje niekde v robotovi, ale stačí, keď ho vymyslí raz, a pomocou duplikácie a sady transformácií ako otočenie, zrkadlové preklopenie a posun ho môže použiť viackrát na ľubovoľnom mieste na povrchu robota). Príkladom takejto kompaktnej reprezentácie v prírode je ľudský mozog s približne 100 miliardami neurónových spojov, ktorý je však reprezentovaný sadou len niekoľko desiat tisíc génov v ľudskom genóme.

Aby prepis genotypu na fenotyp zvyšoval schopnosť organizmov vyvíjať sa evolúciou, mal by mať niektoré dôležité vlastnosti [40]. Prvou z nich je *expresívna sila*, t.j. schopnosť reprezentovať veľkú škálu rôznych vlastností fenotypu. V evolučnej robotike takýmito vlastnosťami môže byť morfológia robota, jeho ovládanie, ale aj vlastnosti samotného prepisu genotypu na fenotyp. Čím má prepis vyššiu expresívnu silu, tým rôznorodejšie roboty sú evolúcii prístupné a tým má evolúcia vyššiu šancu nájsť robota lepšie adaptovaného na konkrétnu úlohu, ako pri prepise s nízkou expresívnou silou. Keď je napríklad súčasťou genotypu aj morfológia aj ovládanie robota, môže evolúcia vyvíjať obe vlastnosti naraz tak, aby spolu kooperovali pri riešení danej úlohy.

Druhou dôležitou vlastnosťou je *kompaktnosť*. Vzťah medzi zložitou genotypu a fenotypu by mal byť len veľmi slabý, takže napríklad aj jednoduché genotypy by mali mať schopnosť reprezentovať veľmi zložité fenotypy. Táto vlastnosť je dôležitá, pretože evolúcia priamo prehľadáva priestor genotypov a nie priestor fenotypov. Čím

jednoduchší je genotyp, tým menší je prehľadávaný priestor a tým efektívnejšie môže byť evolúciou prehľadávaný. Existuje viac mechanizmov ako dosiahnuť kompaktnosť. Niektorými z nich sa budeme bližšie zaoberať v ďalšom texte. Hlavnou myšlienkou je však práve modulárnosť a symetria tak, aby ten istý gén mohol byť použitý na reprezentáciu viacerých vlastností organizmu.

Ďalšou vlastnosťou, ktorú by mal spĺňať prepis genotypu na fenotyp je *schopnosť vývoja* (z angl. termínu *evolvability*), t.j. schopnosť organizmu zlepšiť svoju fitness funkciu po aplikácii genetických operátorov, ako je kríženie alebo mutácia. V tomto prípade totiž fitness funkcia výsledného organizmu záleží len na tvare samotnej fitness funkcie v okolí rodiča, na genetických operátoroch (ktoré pozmenia nejakým spôsobom genotyp potomka) a na metóde použitej pri prepise genotypu na fenotyp, na ktorej závisí, ako sa zmeny v genotype prejavajú na výslednom fenotype organizmu.

V evolučnej robotike je často používanou metódou jednoduchá priama reprezentácia fenotypu v genotype. V tomto prípade sú genotyp a fenotyp úplne (alebo skoro úplne) totožné. Toto nemusí vadiť v prípadoch, keď je riešená úloha jednoduchá (t.j. riešenie obsahuje len veľmi málo elementov), alebo ak symetria a modulárnosť vôbec nepomáhajú pri hľadaní riešenia. Výhodou je jednoduchosť implementácie, a aj preto viacero projektov zvolilo túto cestu [4, 33].

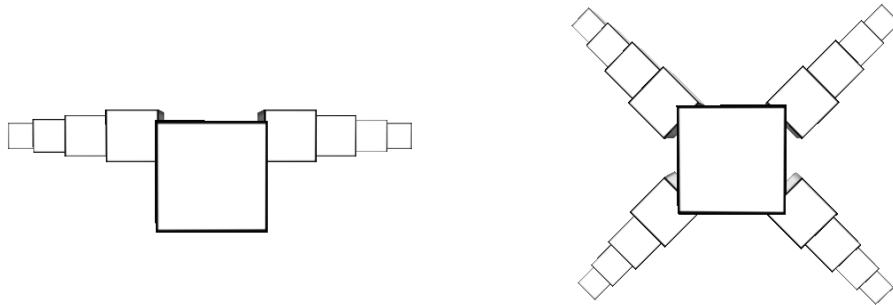
Častým nástrojom použitým k reprezentácii fenotypu v genotype sú gramatiky. Prvé použitie gramatik pri modelovaní biologických systémov siaha k Aristidovi Lindenmayrovi, ktorý predstavil systémy prepisovacích gramatických pravidiel známych ako *L-systémy* [32]. Lindenmayer bol motivovaný snahou o popis štruktúry rastlín, gramatiky je však možné použiť na popis širokej škály rôznych komplexných štruktúr, okrem iného aj na reprezentáciu robotov. Príkladom jednoduchej gramatiky môžu byť dve pravidlá “ $A \rightarrow B$ ” a “ $B \rightarrow AB$ ” a počiatkový symbol, napríklad “A”. Pri aplikácii pravidla “ $A \rightarrow B$ ” je každý výskyt symbolu “A” v reťazci nahradený symbolom “B”. Opakovaná aplikácia týchto pravidiel, počnúc zvoleným počiatkovým symbolom vytvorí sekvenciu:  $A \rightarrow B \rightarrow AB \rightarrow BAB \rightarrow ABBAB \rightarrow BABABBAB \rightarrow \dots$  a tak ďalej (v každom kroku je na každý symbol vstupného reťazca práve raz aplikované korešpondujúce pravidlo). Dve veľmi jednoduché pravidlá a počiatkový symbol môžu tak vytvárať dlhé reťazce so zložitou štruktúrou. Ak tieto reťazce interpretujeme ako príkazy pre stavbu robota, výsledkom môžu byť veľmi zložité morfológie reprezentované však veľmi kompaktným spôsobom. Rekurzívna definícia takýchto systémov pravidiel vedie k sebepodobnosti a výsledné systémy tak často vytvárajú komplexné fraktálne útvary.

Prvým systémom, ktorý aplikoval myšlienky takejto rekurzívnej konštrukcie fenotypu pri evolúcii robotov nájdeme v práci Karla Simsa z roku 1994 [45]. Predmetom evolúcie v Simsovej práci boli organizmy žijúce v 3D prostredí so simulovanými zákonmi fyziky pevných telies. Fenotyp robota je v tomto prípade reprezentovaný stromom, v ktorom vrcholy reprezentujú jednotlivé časti konštrukcie robota (kvádre) a hrany reprezentujú spoje robota. Každý vrchol grafu navyše obsahuje neurónovú sieť a ďalšie vlastnosti potrebné na konštrukciu danej časti robota (rozmery kvádra a pod.). Robot je reprezentovaný genotypom, ktorý je rovnaký ako fenotyp, až na jeden podstatný rozdiel: graf nie je strom a hrany grafu tak môžu tvoriť cykly. Graf genotypu navyše obsahuje jeden špeciálny *koreňový* uzol. Prepis genotypu na fenotyp





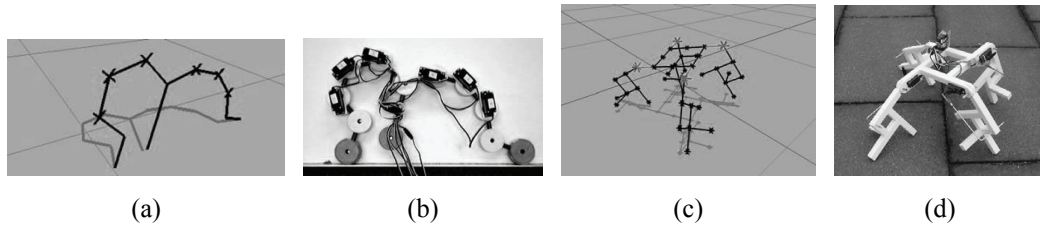
- a. Genotyp robota. Uzol **A** je koreňovým uzlom robota. Rekurzívne limity uzlov **A** a **B** sú 1 a 4.
- b. Fenotyp robota zostavený rozvinutím cyklov genotypu. Kocky **A** a **B** sú spojené hranou **c** a kocky **B** sú navzájom spojené hranou **d**.



- c. Fenotyp robota po úprave škálovania hrany **d** a zapnutí symetrie podľa osi **y** hrany **c**.
- d. Fenotyp robota po úprave uhlu otočenia hrany **c** a zapnutí symetrie podľa osi **x** hrany **c**.

**Obr. 4.** Príklad postupnej konštrukcia robota z jeho genotypu. Štruktúra genotypu (a) sa nemení počas jednotlivých krokov. Robot je zostavený postupnou aplikáciou rekurzívneho rozvinutia cyklov (b), symetrie podľa zvislej osi a škálovania (c) a symetrie podľa vodorovnej osi a otočenia (d).

prebieha tak, že je graf genotypu prechádzaný rekurzívne po hranách počnúc v koreňovom uzli, a všetky navštívené hrany aj vrcholy sú kopírované do fenotypu. V prípade, že je niektorá hrana alebo vrchol navštívená viackrát, vznikne vo fenotype viac nezávislých kópií daného elementu. Na zabránenie nekonečných cyklov má každý vrchol priradený *rekurzívny limit*, ktorý určuje, koľko najviackrát môže byť daný vrchol navštívený počas konštrukcie fenotypu. Hrana môže byť okrem toho označená ako *reflektívna* podľa niektorej z osí *x*, *y* alebo *z*. Ak je hrana reflektívna napr. podľa osi *x*, znamená to, že celý podstrom bude vo fenotype skopírovaný symetricky podľa osi *x*. Reflektívne hrany umožňujú veľmi kompaktnú reprezentáciu symetrických štruktúr. Príklad procesu tvorby fenotypu z genotypu je ilustrovaný na obr. 4.



**Obr. 5.** Príklad robotov vyvinutých evolučným algoritmom v práci Hornbyho a kol [22]. Roboty boli vyvinuté v simulácii (a, c) a následne skonštruované v realite (b, d).

Prínosom tejto metódy prepisu genotypu na fenotyp je schopnosť veľmi kompaktno reprezentovať aj zložité morfológie, najmä vďaka možnosti jednoducho opakovať rovnaké štruktúry (pomocou rekurzívneho kopírovania) a schopnosti vytvárať symetrické štruktúry mutáciou jediného parametra. Simsove virtuálne organizmy dodnes neprestávajú fascinovať svojou podobnosťou zo živou prírodou. Prepis bol následne úspešne použitý aj vo veľkom počte ďalších prác [38, 39, 30, 29, 25].

Prepis z genotypu na fenotyp použitý Simsom v mnohom pripomína L-systémy. Genotyp síce nie je vyjadrený explicitne ako zoznam prepisovacích pravidiel nejakej gramatiky, na hrany a vrcholy genotypu však môžeme nahliadnuť ako na symboly gramatiky a na koreňový uzol ako na počiatočný symbol. Každý genotyp je tak možné zapísať ako gramatiku pomocou vhodne zvolených prepisovacích pravidiel.

Existujú však aj prístupy, ktoré využívajú explicitný zápis prepisovacích pravidiel a robota konštruujú priamo z reťazca symbolov vygenerovaného gramatikou. Príkladom je práca Hornbyho a kol. [22]. V tejto práci je postupnosť znakov v reťazci použitá ako sekvencia príkazov pre LOGO-korytnačku. Príkladmi príkazov pre korytnačku (t.j. symbolov v reťazci určenom gramatikou) sú krok vpred, vzad, otočenie a pod. Stavba tela robota je určená trasou korytnačky. Kĺby sú vytvárané špeciálnym príkazom „joint“, ktorý spôsobí pohyb korytnačky vpred a vytvorenie kĺbu na novej pozícii. Kĺb má jeden stupeň voľnosti, ktorý sa mení v pravidelných osciláciách. Korytnačka môže počas svojej cesty využívať zásobník na ukladanie a čítanie svojho stavu (symboly “[” a “]”) a príkazy môžu byť vykonávané viacnásobne pevný počet krát (symboly “{” a “}”). Sila gramatiky je zvýšená pridaním parametrizovaných a podmienených prepisovacích pravidiel. Príkladom genotypu robota môže byť nasledovná postupnosť pravidiel:

$$\begin{array}{llll}
 \mathbf{A}(n) & \rightarrow \{\mathbf{A}(n-1)\}(n) & \text{pre } n > 2 & (1) \\
 & \rightarrow \text{joint}(1) \mathbf{B}(n*2) \text{ clockwise}(2) & \text{pre } n \leq 2 & (2) \\
 \mathbf{B}(n) & \rightarrow [\mathbf{B}(n/4)] & \text{pre } n > 2 & (3) \\
 & \rightarrow \text{joint}(1) \text{ forward}(1) & \text{pre } n \leq 2 & (4)
 \end{array}$$

Pri počiatočnom symbole  $\mathbf{A}(3)$  bude z tohto genotypu vytvorená postupnou aplikáciou pravidiel nasledovná sekvencia reťazcov:

```

A(3) (pomocou pravidla (1))
{A(2)}(3) (pomocou pravidla (2))
{joint(1) B(4) clockwise(2)}(3) (pomocou pravidla (3))
{joint(1) [B(1)] clockwise(2)}(3) (pomocou pravidla (4))
{joint(1) [joint(1) forward(1)] clockwise(2)}(3)

```

Robot je teda skonštruovaný pomocou výslednej sekvencie príkazov:

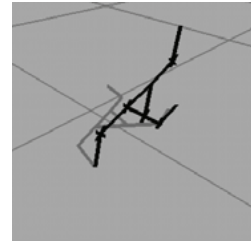
```

{joint(1) [joint(1) forward(1)]
clockwise(2)}(3) .

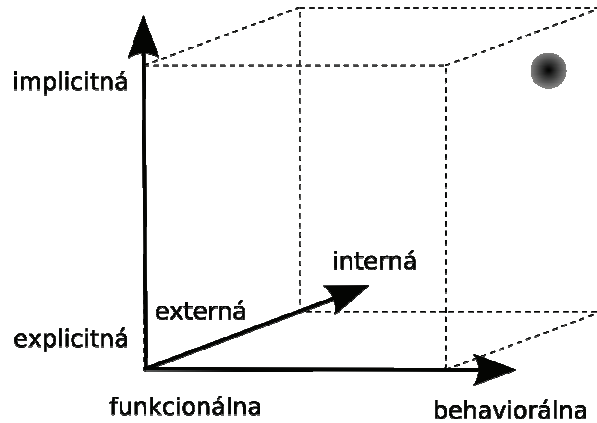
```

Výsledkom je robot zobrazený na obr. 6. Cieľom experimentov bolo vyvinúť týmto spôsobom roboty schopné pohybu po rovnom povrchu v 2D a v 3D prostredí. Výsledkom bola veľká škála robotov schopných úspešného pohybu, z ktorých niektoré boli následne skonštruované v realite (obr. 5). Jedným z prínosov práce bolo porovnanie práve popísanej metódy prepisu genotypu na fenotyp s metódou priameho prepisu, kde bola evolúciou vyvíjaná priamo sekvencia príkazov pre konštrukciu robota (t.j. genotyp a fenotyp boli totožné). Výsledky experimentov ukázali, že reprezentácia robota pomocou sady gramatických pravidiel vedie k evolúcii omnoho zložitejších robotov s vyššou hodnotou fitness funkcie oproti priamej reprezentácii. Pri prepise pomocou gramatík boli výsledné organizmy navyše zostavené z väčšieho množstva častí a obsahovali viac opakujúcich sa štruktúr, niekedy dokonca symetricky usporiadaných.

Oblasť prepisu genotypu na fenotyp je intenzívne skúmanou oblasťou evolučných algoritmov. Viaceré ďalšie metódy prepisu genotypu na fenotyp berú inšpiráciu z prírody. Príkladom je metóda Frederica Gruaua založená na simulácii bunkového rastu použitá pri evolúcii neurónových sietí [18], alebo metóda založená na procesoch génovej regulácie prebiehajúcej v živých bunkách aplikovaná na vývoj neurónových sietí [43] a morfológie simulovaného robota [3, 13]. Ďalším príkladom je algoritmus NEAT navrhnutý Stanleyem v roku 2002, ktorý efektívne vyvíja štruktúru neurónových sietí spolu s váhami neurónových spojov [51], ktorá bola nedávno rozšírená aj na evolúciu robotov [29]. Detaily týchto prístupov sú však nad rámec tejto kapitoly. Prehľad a klasifikácia metód je uvedený napríklad v práci Stanleyho a Miikkulainena [50] alebo v práci Nolfiho a Floreana [40].



**Obr. 6.** Robot skonštruovaný na základe sady prepisovacích pravidiel v práci Hornbyho a kol. [22].



Obr. 7. Schéma priestoru fitness funkcií [40].

## 5 Meranie úspešnosti robotov

Definícia úspešnosti robota (alebo *fitness funkcie*) je jednou z kľúčových častí akéhokoľvek evolučného algoritmu. Evolúcia práve na jej základe vyberá najlepšie organizmy a fitness funkcia tak nepriamo vyberá smer, ktorým sa bude prehľadávanie priestoru možných riešení uberať. Vstupom fitness funkcie je organizmus, ktorého úspešnosť chceme merať, a výstupom je reálne číslo reprezentujúce úspešnosť organizmu (typicky vyššie číslo znamená lepší organizmus). Fitness funkcia pre evolúciu robotov typicky skombinuje sadu premenných odmeraných počas testu robota do čísla reprezentujúceho úspešnosť robota pri riešení danej úlohy. Jednotlivé prístupy k návrhu fitness funkcie sa líšia podľa toho, či evolúcia prebieha na skutočných robotoch alebo v simulácii. Problémy súvisiace s prechodom od simulácie k reálnym robotom bližšie popíšeme v sekcii 6. Tu sa budeme venovať skôr všeobecným prístupom k návrhu fitness funkcií a uvedieme niektoré konkrétne príklady.

Rôzne prístupy k návrhu fitness funkcie pre evolúciu robotov môžu byť prehľadne znázornené pomocou priestoru fitness funkcií navrhnutého Nolfim a Floreanom v [40]. Tento priestor je určený tromi osami, znázornenými na obr. 7. Konkrétna fitness funkcia v tomto 3D priestore reprezentuje jeden bod. Každá z osí popisuje spojitú škálu prístupov k návrhu fitness funkcií podľa jedného z nasledovných kritérií:

- *Funkcionálna-behaviorálna* dimenzia určuje, či je fitness funkcia zameraná na meranie špecifických procesov prebiehajúcich v ovládacom systéme robota, alebo na celkové výsledné chovanie robota. Príkladom môže byť evolúcia chodenia pre robota ovládaného neurónovou sieťou. Funkcionálna fitness funkcia by sa v tomto prípade snažila ohodnotiť výstupné hodnoty jednotlivých neurónov tak, aby zodpovedali signálom

očakávaným pri chodení [31]. Naopak behaviorálna fitness funkcia meria vzdialenosť prejdenú robotom za vopred daný čas [21].

- *Explicitná-implicitná* dimenzia určuje, koľko premenných a obmedzení je v definícii fitness funkcie použitých. Príkladom môže byť implicitná fitness funkcia vyberajúca také roboty, ktoré po dostatočne dlhú dobu nevyčerpajú svoju energiu [25, 48]. Naopak explicitná fitness funkcia by obsahovala rovnicu s premennými popisujúcimi jednotlivé aspekty požadovaného chovania (či je robot schopný hľadať zdroje energie, ako rýchlo sa k nim vie priblížiť a pod.).
- *Externá-interná* dimenzia určuje, či je fitness funkcia počítaná pomocou hodnôt dostupných samotnému robotovi počas jeho ohodnocovania. Napríklad informácia o vzdialenosti robota od jeho cieľa v bludisku je externou informáciou, pretože nemôže byť vypočítaná z hodnôt senzorov robota. Táto os je špeciálne vhodná na klasifikáciu fitness funkcií pre robotov vyvíjaných v realite, kde meranie externých premenných môže byť v niektorých prípadoch značne náročnou technickou výzvou.

Ako uvádza Nolfi a Floreano [40], výber fitness funkcie a jej pozícia v popísanom priestore záleží na celi konkrétneho projektu. V prípade, že je cieľom optimalizovať sadu parametrov pre zložitý, ale dobre definovaný, problém v predvídateľnom prostredí, mala by byť fitness funkcia blízko ľavého dolného rohu priestoru. Takéto fitness funkcie (FEE – Funcionálne, Explicitné, Externé) hodnotia špecifické deje v robotovi, sú definované na základe sady premenných a obmedzení a pri meraní fitness funkcie na reálnych robotoch používajú presné externé meracie nástroje. Na druhej strane, ak má byť cieľom navrhnuť autonómne roboty operujúce v neznámom alebo nepredvídateľnom prostredí, kde sa roboty musia adaptovať na aktuálne podmienky, fitness funkcia by mala byť bližšie k pravému hornému rohu. BII (Behaviorálne, Implicitné, Interné) fitness funkcie ohodnocujú organizmy čisto na základe ich celkového správania, pomocou veľmi malého počtu premenných a fitness funkcia sa dá vypočítať z údajov priamo v robotovi.

## 5.1 Štandardný návrh fitness funkcie

Štandardným spôsobom návrhu fitness funkcie je testovať každého z robotov v populácii samostatne a na základe jeho výkonu počas testu mu priradiť ohodnotenie reprezentujúce jeho úspešnosť. Čo sa týka samotného cieľa evolúcie, veľká väčšina prác je zameraná na výkon jednoduchých motorických úkonov. Najčastejším príkladom je rýchly pohyb na pevnom povrchu [21, 22, 33, 45, 29, 30, 25]. Úspešnosť je v tomto prípade definovaná zväčša jednoducho ako vzdialenosť prejdená organizmom počas testu. Táto jednoduchá definícia však vo veľa prípadoch nemusí stačiť. Evolúcia totiž nemá tú intuitívnu predstavu „rýchleho pohybu vpred“, ktorú mal človek pri návrhu fitness funkcie, a preto často môže prísť na riešenia, ktoré zodpovedajú fitness funkcii, nezodpovedajú však tejto intuitívnej predstave. Jeden z prvých príkladov tohto problému je spomenutý už v práci Karla Simsa [45, 46]. Sims vo svojej práci uvádza problém, keď pri popísanej jednoduchej fitness funkcii evolúcia

vytvorila organizmy v tvare vysokej veže, ktorá pri svojom páde posunula ťažisko organizmu ďaleko od počiatočnej pozície a tým zvýšila jeho fitness. Sims riešil túto situáciu úpravou fitness funkcie tak, aby samotný test začal až potom, keď ťažisko robota dosiahne stabilné minimum.

Niektorými z ďalších úloh, ktoré boli použité pri evolúcii robotov, boli rôzne variácie pohybu po pevnej zemi, ako napríklad pohyb po nerovnom povrchu [21, 30], po schodoch alebo cez trhliny [30] alebo pohyb vopred daným smerom [30]. Niektoré z ďalších príkladov sú pohyb na pevnom povrchu s pomocou pohyblivej platformy [30], evolúcia katapultu [6] (cieľom robota je vystreliť kocku čo najďalej), skákanie do výšky [45, 29], rýchly pohyb vo vode [45, 29], sledovanie cieľa vo vode [45, 29] alebo dokonca lietanie [44, 36].

Problém chybnéj definície fitness funkcie nenastáva len pri evolúcii robotov schopných rýchleho pohybu, ale je veľmi častým problémom aj pri iných fitness funkciách. Ako príklad uvidíme definíciu fitness funkcie z našej práce [29], kde bolo cieľom robota nasledovať objekt v 3D vodnom prostredí. Test robota bol rozdelený na viacero po sebe idúcich časových úsekov. V každom časovom úseku bol cieľový objekt umiestnený do inej pozície vzhľadom na robota (postupnosť pozícií však bola rovnaká vo všetkých testoch). Fitness funkciou bola priemerná rýchlosť robota smerom k objektu meraná počas celého testu. Roboty vyvinuté pomocou tejto fitness funkcie boli na prvý pohľad schopné úspešne plniť zadanú úlohu a plávať k danej pozícii. Keď však bola pozícia objektu upravená (alebo bol objekt vymazaný), roboty stále nasledovali rovnakú cestu bez zmeny správania. Jednou z možností ako riešiť tento problém je meniť počas testu pozíciu cieľového objektu náhodne. Ďalším riešením je reštartovať robota kompletne pri začiatku každého časového úseku (t.j. uviesť jeho konštrukciu aj ovládanie do počiatočného stavu). Robot sa tak nemôže spoliehať na pevne danú trasu, a musí sa rozhodovať výlučne na základe aktuálnej polohy objektu. Toto riešenie viedlo k robotom, ktoré úspešne nasledovali cieľový objekt a boli schopné interaktívne reagovať na zmenu jeho polohy.

Pri návrhu fitness funkcie je možné naraziť na viacero ďalších úskalí. Jedným z príkladov je návrh príliš náročnej fitness funkcie, ktorá pri štarte evolúcie ohodnotí všetky organizmy nulou a evolúcia preto nemá pri svojom štarte k dispozícii žiadnu informáciu o tom, ktorým smerom sa pri prehľadávaní uberať. Prehľadávanie je v tomto prípade odkázané na náhodu (tá istá fitness funkcia však môže fungovať výborne pre organizmy, ktoré už majú niektoré základné schopnosti). Riešením je definovať sadu fitness funkcií rastúcej náročnosti, alebo učiť roboty úlohy postupne, tak, aby si zároveň zachovali pôvodné znalosti [5].

## 5.2 Interaktívna evolúcia

Jedným zo zaujímavých prístupov k riešeniu väčšiny problémov so zadaním fitness funkcie je nepočítať ju automaticky z vlastností robota a jeho prostredia, ale definovať ju ako ohodnotenie, ktoré je robotovi pridelené interaktívne (napríklad človekom). Takýto druh selekcie prvýkrát uviedol Richard Dawkins v roku 1986 vo svojom známom programe Biomorph [9] umožňujúcom užívateľom riadenú evolúciu 2D stromových štruktúr. V tomto programe fitness funkciu určoval človek interaktívnym

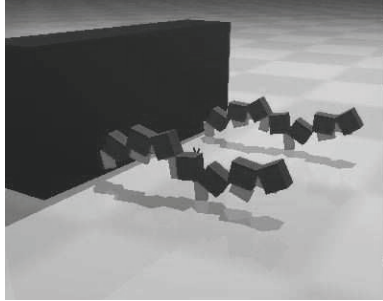
výberom organizmu z navrhovaných možností na obrazovke počítača. Vyberajúcim agentom pri interaktívnej evolúcii však nemusí byť len človek. Dawkins navrhol možnosť nahradiť človeka iným živým organizmom. Príkladom by mohol byť hmyz, ktorý smerom svojho letu určí, ktorý z virtuálnych kvetov na obrazovke počítača preferuje. Selektívny tlak by takto pracoval v prospech takých tvarov kvetín, ktoré najviac vyhovujú danému hmyzu. Hlavnou nevýhodou interaktívnej evolúcie však je, že proces výberu organizmov je časovo značne náročný a vyžaduje si často veľkú trpezlivosť zo strany človeka. V prípade, že človek vyberá riešenia trpezlivo a konzistentne, vyhýba sa interaktívna evolúcia väčšine problémov, ktoré nastávajú pri použití klasickej automatickej fitness funkcie (ako je napríklad stagnácia evolúcie alebo nepresná definícia fitness funkcie).

Interaktívna evolúcia bola viackrát použitá aj v oblasti evolučnej robotiky. Príkladom je práca Lunda a kol. [34], ktorý túto techniku použil pri evolúcii správania simulovaných robotov. Človek, pozorujúci evolúciu, mal v tomto projekte možnosť vyberať preferovaných robotov z ponuky v aktuálnej populácii na základe ich simulovaného správania. Vybrané roboty boli použité pri tvorbe novej populácie. Ďalším príkladom interaktívnej evolúcie je práca Thomasa Raya zameraná na evolúciu morfológie virtuálnych bytostí podliehajúcich simulovaným fyzikálnym zákonom [41]. Cieľom tejto práce bolo vyvinúť organizmy čisto za účelom estetickéj príťažlivosti. Projekt sa tak pohybuje na hranici evolučnej robotiky a evolučného umenia.

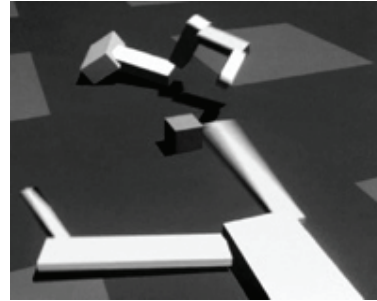
### 5.3 Koevolúcia

V popísaných prípadoch bola fitness funkcia vyjadrená vždy ako vzťah organizmu k jeho (väčšinou statickému) prostrediu. Ďalším prístupom je však popísať fitness funkciu ako vzťah organizmu k iným vyvíjajúcim sa organizmom. Takýto prístup potom vedie k tomu, že vyvíjajúce organizmy sa počas evolúcie navzájom ovplyvňujú v tzv. koevolúcii. Takýto spoločný vývoj môže prebiehať dvoma spôsobmi. Buď si organizmy navzájom konkurujú, vtedy ide o tzv. kompetitívnu koevolúciu, alebo si navzájom pomáhajú v tzv. kooperatívnej koevolúcii. V prvom prípade môžu byť výsledkom „závody v zbrojení“, kde sa každý z protivníkov musí sústavne adaptovať na meniace sa vlastnosti toho druhého (príklady tohto vzťahu vidíme veľmi často v evolúcii živých organizmov, napríklad vo vzťahu parazita k jeho hostiteľovi, dravca a jeho koristi a pod.). Naopak pri kooperatívnej koevolúcii si organizmy navzájom pomáhajú a takýto evolučný vzťah často končí vzájomnou závislosťou (príkladom z prírody je kooperácia hmyzu a rastlinných kvetov). Zlepšovanie súťažiacich strán má však svoje limity. Problém napríklad nastáva, keď sa organizmy počas evolúcie pravidelne vracajú k niektorej staršej taktike a koevolúcia sa tak pohybuje v opakujúcich sa cykloch. Spôsoby, ako riešiť tento a podobné problémy, sú však nad rámec tejto kapitoly (prehľad metód je možné nájsť v [40]).

Koevolúcia bola viackrát úspešne aplikovaná aj na evolúciu robotov. Príkladom kooperatívnej koevolúcie je evolúcia robotov za účelom posúvania predmetu, ktorý by žiadny z robotov nemohol presunúť sám [30] (viď obr. 8). Jedným z prvých príkladov kompetitívnej koevolúcie je práca Karla Simsa z roku 1994 [46], v ktorej boli simulované organizmy zložené z pevných blokov (kvádrov) simulovaných v 3D



**Obr. 8.** Kooperácia robotov pri presune bloku [30].



**Obr. 9.** Súboj simulovaných robotov o kocku [45].

prostredí riadenom fyzikálnymi zákonmi (viď obr. 9). Pri samotnom súboji boli vždy dva organizmy postavené na štartovné línie oproti sebe s kockou položenou v strede medzi nimi. Simulácia bola potom spustená po dobu 8 sekúnd. Po ukončení simulácie bola odmeraná vzdialenosť každého organizmu od kocky a fitness funkcia pre organizmy bola spočítaná nasledovným vzťahom:

$$f_1 = 1.0 + \frac{d_2 - d_1}{d_1 + d_2} \quad f_2 = 1.0 + \frac{d_1 - d_2}{d_1 + d_2}$$

kde  $d_1$  a  $d_2$  je vzdialenosť prvého a druhého organizmu od kocky po skončení testu. Takáto definícia zaručuje, že fitness je vždy v rozsahu medzi 0 a 2 pre každý organizmus, rovnako kvalitní oponenti dostanú každý ohodnotenie 1 a priemerné ohodnotenie je vždy 1. Táto formulácia nielen odmeňuje organizmus za to, že sa dostal bližšie ku kocke ako jeho oponent, ale odmeňuje aj za to, keď sa organizmu podarí udržať oponenta čo najďalej od kocky. Simsovi sa podarilo dosiahnuť medzi vyvíjanými organizmami preteky v zbrojení, keď zlepšenie jedného organizmu vyprovokovalo zmenu v druhom organizme, a to podnietilo ďalšie zmeny v prvom organizme. Príklad iného súboja nájdeme napríklad v práci Thomasa Miconiho [39], kde organizmy na rozdiel od “mierumilovných” súbojov o kocku v Simsovej práci získavali body na základe počtu a sily úderov, ktoré dokázali zasadiť svojmu protivníkovi.

Viacero prác sa venovalo výskumu kompetitívnej koevolúcie populácie lovcov a koristií. Príkladom je práca Nolfiho a Floreana [40], v ktorej evolúcia prebiehala na robotoch Khepera, alebo práca Clifffa a Millera [8], ktorej cieľom bolo vyvinúť simulované roboty ovládané neurónovou sieťou. Roboty v práci Clifffa a Millera majú kruhový pôdorys a po obvode umiestnené senzory detekujúce iné roboty. Súčasťou evolúcie bola nielen samotná neurónová sieť, ale aj pozícia senzorov na robotovi. Výsledkom experimentov boli lovci so senzormi umiestnenými vpredu (ako je tomu u očí predátorov v prírode), koristiť mala však senzory často umiestnené po stranách (ako je tomu napr. u gazely) a v niektorých prípadoch dokonca vzadu.





**Obr. 10.** Príklad robota vyvinutého v projekte Framsticks [25].



**Obr. 11.** Ekosystém v projekte DivisionBlocks [48]. Virtuálny svet pozostáva zo štvorcových platforiem ohraničených vodou. V ľavom dolnom rohu je pohľad na svet z vtáčej perspektívy.

#### 5.4 Simulácia ekosystémov

Fitness funkcia je v štandardnom genetickom algoritme uvedená v explicitnej forme. Organizmu je priradené reálne číslo hodnotiace jeho úspešnosť a na jeho základe je mu pridelený počet potomkov. Toto však nie je jediný spôsob ako pristupovať k simulácii evolúcie. Iným riešením je simulovať celú populáciu organizmov žijúcich vo virtuálnom prostredí bez explicitnej definície fitness funkcie. V tomto prostredí prežívajú tie organizmy, ktoré sú schopné presadiť sa na úkor ostatných a dosiahnuť vyšší počet potomkov. Tento prístup je oveľa bližší evolúcii v živej prírode. Nevýhodou je však nutnosť kontroly dynamiky rastu populácie (populácia môže napríklad celá vymrieť) a nižšia kontrola nad procesom evolúcie. Takáto evolúcia nemá žiadny explicitne zadaný cieľ, a preto sa jej hovorí evolúcia s otvoreným koncom (z angl. *open-ended*).

Jeden z projektov takéhoto otvoreného ekosystému predstavil Lee Spector [48] vo svojom projekte Division Blocks. Každý organizmus začína ako jednoduchý blok ovládaný neurónovou sieťou. Všetky organizmy sú postavené do spoločnej 3D arény, kde majú možnosť zbierať energiu zo simulovaného slnka, meniť svoj objem, deliť sa, spájať sa s inými kockami pomocou kĺbov a vymieňať si rôzne zdroje s okolitými organizmami (viď obr. 11). V práci nie je použitá žiadna explicitná fitness funkcia. Namiesto toho musia organizmy prísť počas evolúcie na správne spôsoby ako sa spájať s inými tak, aby prežili v ekosystéme. Výsledky ukazujú, že evolúcia je schopná vyvinúť organizmy schopné kooperácie pri výmene zdrojov.

V projekte Framsticks [25] roboty taktiež žijú v ekosystéme simulujúcom prirodzené podmienky. Robot vo Framsticks je (ako už anglický názov napovedá) zložený z flexibilných tyčiek pospájaných do stromovej štruktúry (viď obr. 10). Každá tyčka reprezentuje sval s vopred danou silou (sila svalu je súčasťou genotypu robota). Sval čerpá energiu z energetických zásob prítomných v tyčke, pričom každá tyčka má na začiatku simulácie priradené počiatkové množstvo energie (toto množstvo je pre každú tyčku uložené v genotype robota). Energia tyčky klesá pri používaní

svalu (množstvo spotrebovanej energie závisí na aplikovanej sile) a rastie po príjme stravy z prostredia. Každý robot musí byť schopný nájsť potravu dostatočne rýchlo, aby prežil v súťaži s ostatnými robotmi.

Oba spomínané prístupy vyvíjali roboty bez ambície skonštruovať ich v realite. Jedným z projektov, ktorý sa pokúsil práve o evolúciu reálnych robotov pomocou simulácie ekosystému nájdeme v práci Studera a Lipsona [52]. Cieľom evolúcie bolo skúmať replikáciu jednoduchých robotov tvaru kocky (konštrukciu týchto robotov popisujeme bližšie v sekcii 6, príklad robota je uvedený na obr. 17). Pravidelná štruktúra robota umožnila reprezentovať ekosystém jednoduchou 2D mriežkou, kde každá z kociek robota obsadzuje jedno pole mriežky. Správanie systému tak pripomína celulárne automaty. Každý z robotov je popísaný genotypom definujúcim jeho správanie. Roboty sa môžu navzájom spájať a vytvárať tak zložitejšie štruktúry. Ak robot v priebehu svojho pohybu koliduje s iným robotom, prepíše jeho ovládanie svojim, pričom mutácia je simulovaná chybou pri tomto prepise. Výsledkom samovoľnej evolúcie v tomto prostredí bolo veľké množstvo štruktúr schopných replikovať sa a rozširovať sa tak v populácii na úkor ostatných štruktúr.

## 5.5 Simulácia fyzikálneho prostredia

Dôležitou súčasťou výpočtu viacerých fitness funkcií je simulácia fyzikálneho prostredia robota. V začiatkoch evolučnej robotiky bolo viacero výskumníkov odkázaných na vlastnú implementáciu simulácie fyzikálnych zákonov [16, 45, 25]. Rozvoju fyzikálnych simulátorov však pomohol herný priemysel a dnes už existuje veľké množstvo voľne aj komerčne dostupných fyzikálnych simulátorov, čo uľahčuje vstup do poľa evolučnej robotiky ďalším výskumníkom.

Pri simulácii fyzikálnych zákonov vo virtuálnom svete je čas rozdelený na diskrétné intervaly a chovanie objektov v rámci intervalov je aproximované matematickým modelom. Takáto aproximácia umožňuje efektívny výpočet, prináša však so sebou v niektorých situáciách nestabilitu a nepresnosť simulácie. Takéto nepresnosti sú špeciálne viditeľné pri použití evolučných algoritmov, pretože roboty vyvíjané evolúciou využívajú nepresnosti simulácie vo svoj prospech. Príkladom tohto problému sú roboty v jednej z našich prác [29], ktoré počas evolúcie objavili chybu v simulátore umožňujúcu rýchly pohyb vpred pomocou ťažko postrehnuteľného kmitania jednej z častí ich konštrukcie. Frekvencia kmitania pritom bola veľmi vysoká - optimalizovaná tak, aby bola chyba v simulácii zneužitá čo najefektívnejšie. Roboty takto dosahovali veľmi vysoké rýchlosti pohybu vpred, avšak na základe pohybov, ktoré by v reálnom svete neboli možné. Riešením bola v tomto prípade detekcia oscilácií počas testu a ohodnotenie „previnilého“ robota nulovou fitness.

Nestability v dynamických fyzikálnych simuláciách môžu mať veľké množstvo príčin (napr. príliš veľké množstvo kontaktov v scéne alebo spojenie objektov s veľkým rozdielom hmotností). Riešením je týmto situáciám predchádzať buď úpravou samotného simulátora, alebo úpravou genetických operátorov tak, aby výsledné roboty nespôsobili takéto chovanie. V prípade, že toto riešenie nie je možné, môže byť problém riešený diskvalifikáciou robotov z ďalšej evolúcie na základe detekcie škodlivého chovania počas samotného testu.

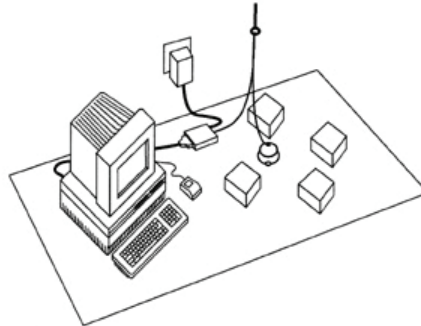
Čím viac vlastností prostredia je súčasťou simulácie, tým je simulácia vernejšia, avšak tým väčšie problémy môže pôsobiť prechod od simulovaných robotov k reálnym. Jedným z prístupov je preto zjednodušenie simulovaných fyzikálnych zákonov tak, aby bolo chovanie robotov v simulácii ľahšie prenositeľné na reálne roboty [33, 22]. Tomuto a ďalším problémom pri prechode od simulácie k reálnym robotom sa budeme bližšie venovať v sekcii 6.

Výpočet fyzikálnej simulácie síce často býva rýchlejšie než reálny čas, toto však pri nasadení v evolučných algoritmoch nemusí stačiť. Evolučný algoritmus pri prehľadávaní priestoru možných riešení potrebuje vyhodnotiť fitness často tisícky-krát a fyzikálna simulácia je vo väčšine prípadov tou výpočtovo najnáročnejšou časťou evolučného algoritmu. Existuje našťastie veľa možností ako výpočet fitness funkcie urýchliť. Jedným z najpoužívanejších prístupov je distribuovanie výpočtu na viacero výpočtových jednotiek. Tento prístup ťaží z nezávislosti výpočtu fitness funkcie pre jednotlivé roboty. Jednou z možností je využiť na tento účel akademické výpočtové siete, prípadne komunity užívateľov ochotných venovať voľný procesorový čas evolúcii robotov [33]. Výpočet však nemusí byť distribuovaný len na rôzne počítače, ale napríklad aj na procesory jedného superpočítača (toto riešenie bolo použité napríklad Karlom Simsom [45]). Jeden z relatívne nových prístupov k urýchleniu výpočtu fyzikálnej simulácie prichádza zo sveta počítačových hier. Existujúce grafické akcelerátory sú vysoko paralelné a pri správnom paralelizovaní výpočtu fyzikálnej simulácie je možné využiť ich na oveľa efektívnejší výpočet ako na štandardnom CPU (tento prístup využíva napríklad projekt Molecubes [56]). Ruka v ruke s grafickými akcelerátormi sa začínajú presadzovať aj hardvérové akcelerátory špecializované práve na výpočet fyzikálnych simulácií (zamierené najmä na herný priemysel). Predbežné výsledky nám dávajú nádej, že simulovaná evolúcia bude raz prebiehať rýchlosťou umožňujúcou jej výpočet priamo v reálnom robotovi. Takýto robot by mohol napríklad internou evolúciou dosiahnuť rýchlu adaptáciu na zmeny okolitých podmienok [4].

## 6 Od simulácie k reálnym robotom

V predchádzajúcom texte sme sa venovali najmä evolúcii robotov v simulovanom prostredí. Konečným cieľom evolučnej robotiky je však návrh skutočných robotov riešiacich úlohy v realite. K evolúcii reálnych robotov existujú dva základné prístupy. Prvým z nich je testovanie každého jedného kandidátskeho riešenia, ktoré sa objaví v priebehu evolúcie, na reálnom robotovi (tzv. on-line evolúcia). Tento prístup však vedie k viacerým praktickým problémom (napr. vysoká časová náročnosť celého procesu). Častým riešením je preto evolúcia v simulácii a prenos až výsledného robota do reality (tzv. off-line prístup). Výhodami a nevýhodami on-line a off-line evolúcie a ďalším otázkam týkajúcim sa prechodu od simulácie k realite sa podrobnejšie zaoberá sekcia 6.1.

Úloha prechodu od simulácie k reálnym robotom je obzvlášť ťažká, ak je cieľom optimalizovať okrem ovládania aj samotnú konštrukciu robota. Ak má byť možné zostaviť robota s konštrukciou, ktorá bola navrhnutá evolúciou, je nutné mať



**Obr. 12.** Schéma nastavenia experimentu pri on-line evolúcii.

k dispozícii vysoko konfigurovateľnú hardvérovú platformu schopnú reprezentovať široké spektrum rôznych konštrukcií. Niektoré z takýchto platforiem, ako aj niektoré vybrané roboty s fixnou konštrukciou, predstavíme v sekcii 6.2.

### 6.1 Praktické výzvy pri evolúcii reálnych robotov

Ako sme už zmienili v úvode, evolúcia reálnych robotov môže prebiehať jedným z dvoch spôsobov. Pri on-line prístupe je počas evolúcie fitness funkcia pre každý organizmus vyhodnocovaná priamo na reálnom robotovi. Naopak pri off-line evolúcii celá evolúcia prebieha na počítači pomocou simulovaného robota a na reálnom robotovi je spustený len víťaz evolúcie.

Typický on-line evolučný experiment prebieha teda z časti v počítači (tu beží evolučný algoritmus) a z časti na reálnom robotovi (ohodnocovanie fitness funkcie jednotlivých kandidátskych riešení). Experiment je veľmi často plne automatizovaný a nevyžaduje si tak prítomnosť experimentátora počas celého svojho behu (príklad schémy takéhoto experimentu je na obr. 12). Evolúcia priamo na reálnych robotoch má svoje nesporné výhody. Tým, že sú všetky kandidátske riešenia priebežne ohodnocované na reálnom robotovi je zabezpečené, že výsledné najlepšie riešenie bude bez problémov okamžite fungovať na reálnom robotovi (čo nie je také samozrejmé pri off-line evolúcii ako uvidíme neskôr). Ďalšou výhodou on-line prístupu je, že evolúcia má počas celého svojho behu prístup k všetkým vlastnostiam reálneho robota a jeho okolia a môže tak patrične využívať každý vhodný detail na zvýšenie fitness funkcie riešenia. Výsledkom môžu byť roboty, ktoré sú lepšie adaptované na riešenie úlohy v danom prostredí v porovnaní s robotmi vyvíjanými v simulácii.

Jedným z najväčších praktických problémov on-line prístupu je však časová náročnosť evolúcie. Samotná rýchlosť testovania konkrétneho kandidátskeho riešenia je limitovaná rýchlosťou fyzikálnych procesov, ako je pohyb motorov robota a pod. Navyše počet robotov, na ktorých prebieha testovanie, je typicky veľmi malý (vo väčšine prípadov je použitý len jeden robot). Príkladom je experiment (popísaný v [40]), ktorého cieľom bolo vyvinúť robota schopného zbierať odpad z vopred daného územia. Pri experimente bol použitý robot Khepera (bližší popis tohto robota uvádzame v nasledujúcej sekcii). Evolúcia prebiehala na reálnych robotoch a celý

experiment trval 10 dní. Problém časovej náročnosti môže byť čiastočne riešený distribuovaním ohodnocovania fitness funkcie na viacero robotov. Toto riešenie bolo použité napríklad v práci [14], kde bolo paralelne použitých 8 robotov. Jedným z problémov pri paralelnom použití viacerých robotov je však skutočnosť, že je ťažké dosiahnuť absolútne rovnaké správanie všetkých robotov, na ktorých prebiehajú testy. Výsledkom evolúcie potom môžu byť riešenia adaptované na konkrétneho robota, ktoré po presune na iného robota dosahujú nižšiu hodnotu fitness funkcie. Príčinu tohto problému nie je ľahké odstrániť, pretože rozdiely sú často spôsobené malými rozdielmi vo vlastnostiach stavebných súčiastok robota, alebo miernymi rozdielmi v okolitom prostredí. Nolfi a Floreano [40] napríklad popisujú prípad, keď minimálne zmeny v prostredí (ambientné svetlo, farba, tvar objektov) spôsobili výrazné rozdiely v citlivosti infračervených senzorov robota (napriek rovnakej technickej špecifikácii senzorov).

Reálne roboty používané pri on-line evolúcii musia splňať oproti svojim simulovaným náprotivkom množstvo ďalších podmienok. Reálny robot musí napríklad mať zabezpečený neustály zdroj energie (čo môže byť praktický problém pri plne automatizovanej evolúcii trvajúcej niekoľko dní). Niektoré z kandidátskych riešení môžu po robotovi vyžadovať netypické chovanie, ktoré môže viesť až k poškodeniu robota (obzvlášť v úvodných fázach evolúcie). Tomuto problému sa dá niekedy predísť včasnou detekciou nebezpečných konfigurácií ešte pred spustením samotného testu na robotovi. V prípade, že takáto prevencia nie je možná, musí byť zabezpečená vysoká mechanická robustnosť robota a jeho odolnosť voči zrážkam pred začatím samotného experimentu.

Ďalšie praktické obmedzenie pri on-line evolúcii plyní z faktu, že po vyhodnotení fitness funkcie pre konkrétne riešenie je nutné robota zakaždým pripraviť na nasledujúci test. V ideálnom prípade by mal byť aj robot aj jeho okolité prostredie v rovnakom stave vždy pri začatí testu, kvôli zabezpečeniu opakovateľnosti experimentov. Tento stav nemusí byť ľahké dosiahnuť v prípade zložitejších fitness funkcií. Nolfi a Floreano [40] popisujú prípad on-line evolúcie, pri ktorej bolo úlohou robota pozbierať objekty v aréne a odviezť ich mimo arénu. Pri tomto experimente bolo nutné pred každým ohodnotením fitness funkcie vrátiť robota aj objekty do arény na štartovnú pozíciu. Automatizácia tohto procesu vyžadovala od experimentátorov netriviálne úsilie. Podobným príkladom je on-line evolúcia pohybu robotického psa AIBO [15, 21], kde robot v niektorých prípadoch počas testu zlyhá a neudrží vzpriamenú polohu. Riešením v tomto prípade bolo použitie predprogramovaného pohybu robota na jeho opätovné uvedenie do vzpriamenej polohy a presun na štartovnú pozíciu.

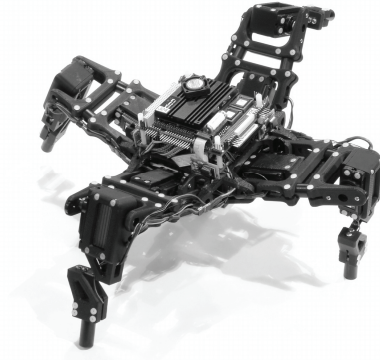
On-line evolúcia reálnych robotov teda naráža na veľké množstvo praktických prekážok (časová náročnosť celého procesu, nároky na mechanickú robustnosť robota, komplikovaná inicializácia testov atď.). Ako sme už naznačili, obľúbeným riešením týchto problémov je práve návrat k simulácii. Fitness funkcia je v tomto prípade vyhodnocovaná v simulovanom prostredí a do reality je prenesený až najlepší vyvinutý robot. Niektoré z prínosov takejto off-line evolúcie sú zjavné. Pri simulácii miznú problémy s dodávaním energie robotovi, odpadá požiadavka na mechanickú robustnosť, všetky roboty majú presne rovnaké vlastnosti a odpadajú aj problémy

s inicializáciou jednotlivých testov aj pri najzložitejších scenároch fitness funkcie. Jedným z ďalších prínosov je opakovateľnosť experimentov so zaručene rovnakým výsledkom (čo nie je samozrejmosť pri evolúcii reálnych robotov). Simulácia ďalej umožňuje detailnejšiu analýzu vyvinutého robota v porovnaní s reálnym robotom. Ako popisuje Nolfi a Floreano [40], pokus o analýzu chovania reálnych robotov v mnohom pripomína analýzu živých organizmov. U živých organizmov je totiž ťažké zistiť niečo o mechanizme ovládania analýzou mozgu *in vitro*. Častejším prístupom je preto analýza pomocou elektród zavedených do mozgu počas života daného organizmu a sledovanie aktivácie rôznych častí mozgu priamo počas riešenia daných úloh. Podobne v robotike je ťažšie analyzovať napr. neurónovú sieť reálneho robota „na papieri“, mimo samotný test, pretože jej fungovanie veľmi silne závisí na konkrétnych vlastnostiach reálneho robota. Prijateľnejšou alternatívou je preto robiť analýzu priamo počas behu experimentu na reálnom robotovi. Pri simulovanej evolúcii je situácia z pohľadu analýzy chovania ideálna, pretože neexistuje závislosť ovládania na reálnom robotovi, a preto je možné detailne analyzovať chovanie priamo v počítači.

Simulácia robota v počítači má potenciál byť omnoho rýchlejšia ako rovnaký test v realite. Dôvodov je hneď niekoľko. V prvom rade je pri simulácii robota jednoduchšie dosiahnuť distribuovanie výpočtu na viacero počítačov. Pri výpočte je samozrejme zaručené, že hodnota fitness funkcie bude rovnaká bez ohľadu na to, na ktorom počítači bola počítaná. Toto je v kontraste s distribuovaním výpočtu fitness funkcie na viacerých reálnych robotov pri on-line evolúcii, kde každý robot má trochu iné vlastnosti, a teda pre rovnaké riešenie môže vrátiť inú hodnotu fitness funkcie. Simulácia môže byť ďalej urýchlená vynechaním nepodstatných vlastností prostredia a simulovaním len tých najdôležitejších [24]. V neďalekej budúcnosti môže byť výpočet fitness funkcie ďalej urýchlený použitím špecializovaných hardvérových komponentov určených na výpočet fyzikálnych simulácií (tento prístup sme popísali v sekcii 5.5). Jedným príkladom situácie, keď môže byť časovo náročnejší výpočet simulácie, ako test na reálnom robotovi, je simulácia vizuálnych senzorov bohatých na informáciu, ako je napríklad kamera.

Hlavnou nevýhodou vývoja robotov v simulácii je však problémový prechod späť do reality po skončení evolúcie. Tento problém je tým výraznejší, čím zložitejšie fyzikálne vlastnosti robota simulujeme. Napríklad pre robota Khepera, ktorý má fixnú konštrukciu a pohybuje sa nezávislým otáčaním dvoch kolies, v mnohých prípadoch postačuje jednoduchá 2D simulácia (pohyb kruhu v rovine), bez nutnosti simulovať jednotlivé fyzikálne sily pôsobiace na rôzne časti robota. V tomto prípade je aj jednoduchšie prejsť od simulovaného robota k reálnemu. V prípade zložitejších morfológií už jednoduchá 2D simulácia často nestačí a prechod do reality môže byť omnoho problematickejší.

Jedným z riešení tohto problému je navrhnúť simulátor takej spoľahlivosti, pri ktorej akýkoľvek robot fungujúci v simulátore bude fungovať aj v realite. Priblížiť sa k tomuto cieľu je možné za cenu obmedzení vo fyzikálnej simulácii. Možnosti pohybu robota sa dajú napríklad obmedziť na také, pri ktorých je možné dostatočne spoľahlivo matematicky vypočítať výslednú polohu robota po prevedení nejakej akcie. Pokusom o toto riešenie je simulátor navrhnutý v práci [33, 22]. Robot navrhnutý pomocou evolúcie v tomto simulátore bol po ukončení evolúcie skonštruovaný v realite (viac



**Obr. 13.** Robot použitý pri experimentoch Bongarda [4]. Vid' text pre popis experimentu.

o tomto procese v nasledujúcej sekcii) a vykazoval rovnaké chovanie ako robot v simulácii. Chovanie výsledných robotov je však značne obmedzené, pretože počas evolúcie nemôžu využívať dynamické fyzikálne procesy. Simulátor podporuje jednoduché pohyby, ako chodenie alebo plazenie, ale nevie simulovať napríklad skákanie a iné procesy vyžadujúce koncept sily a zrýchlenia.

Vytvoriť podobne spoľahlivý simulátor pre plne dynamickú fyzikálnu simuláciu je problematické, pretože fyzikálne procesy majú často chaotické chovanie a sú citlivé na jemné rozdiely v počiatočných podmienkach. Alternatívnym riešením je použitie simulátora, ktorý zachytí „podstatné“ vlastnosti prostredia a vlastnosti simulátora odlišné od reality prekryje šumom tak, aby vyvinuté roboty nemohli tieto rozdiely zneužívať vo svoj prospech [23]. Ďalším prístupom je použitie takého ovládania robota, ktoré robotovi umožní adaptovať sa na rozdiely medzi simuláciou a realitou počas jeho života. Tento prístup v mnohom pripomína živé organizmy, ktoré sa rodia s predurčeným telom a mozgom, ale sú schopné počas svojho života adaptovať sa na prostredie, v ktorom vyrastajú.

Zaujímavý prístup k hľadaniu rozdielov medzi simuláciou a reálnym robotom prináša práca Bongarda a Lipsona [4], v ktorej je tento problém povýšený na samotný cieľ evolúcie. Predmetom experimentu je štvornohý robot (obr. 13), ktorého cieľom je jednoduchý pohyb zvoleným smerom. Robot je ovládaný 8. servomotormi a informácie o svojom stave získava pomocou 8. senzorov čítajúcich polohu servomotorov a dvoch senzorov naklonenia. Cieľom projektu je vyvinúť robustné roboty schopné adaptovať sa počas svojho života na poškodenie svojej konštrukcie spôsobené neznámymi externými silami. Robot však nepoužíva na zistenie poškodenia senzory priamo zisťujúce rozsah poškodení, ale miesto toho si interne udržuje model vlastnej konštrukcie, ktorý sústavne adaptuje podľa aktuálnych údajov získaných zo štandardných senzorov. Proces adaptácie prebieha nasledovným spôsobom. Na začiatku experimentu robot vykoná sadu náhodne zvolených akcií a zapamätá si dosiahnuté hodnoty senzorov. V prvej fáze adaptácie potom robot evolučným algoritmom navrhne 15 modelov, ktoré najlepšie vysvetľujú namerané výsledky prevedených akcií (modely sú testované simuláciou). Druhá fáza adaptácie potom



**Obr. 14.** Robot e-puck (vľavo), Khepera II (v strede) a simulácia robota Khepera II v simulátore Webots (vpravo).

hľadá takú akciu reálneho robota, ktorá umožní vybrať z týchto 15 modelov jeden model, ktorý najviac zodpovedá aktuálnemu stavu robota (t.j. akciu ktorá po prevedení vyvolá čo najrozdielnejšie hodnoty senzorov pre rôzne modely). Túto akciu následne vykoná skutočný robot. Tento proces je zopakovaný 16 krát, čím robot získa dostatok informácií na to, aby našiel dostatočne dobrý model vlastnej konštrukcie. Najlepší nájdený model je následne použitý na nájdenie akcie, ktorá posunie robota vpred. Výsledkom je robot, ktorý je schopný napríklad po zničení jednej zo svojich končatín dynamicky adaptovať svoje chovanie na novú situáciu a zmeniť spôsob pohybu tak, aby čo najlepšie využil aktuálnu konštrukciu. Experimenty ukázali, že táto technika zlepšuje schopnosť pohybu poškodeného robota.

## 6.2 Možnosti hardvérovej konštrukcie robota

Tradičným cieľom pri návrhu robotov pomocou evolučných algoritmov je optimalizovať ovládanie pre konkrétneho robota s vopred danou nemennou konštrukciou. V prírode je však evolúcia schopná optimalizovať aj morfológiu (t.j. stavbu tela) aj chovanie živých organizmov zároveň, ruka v ruke. Takáto dvojité adaptácia sa v prírode ukazuje nesmierne efektívna. Prečo sa teda evolučná robotika vo veľkom množstve prípadov zaoberá len vývojom ovládania a ignoruje morfológiu robota? Tento prístup je z veľkej časti spôsobený praktickými problémami pri optimalizácii morfológie reálnych robotov. Automatická konštrukcia robota s danou optimalizovanou konštrukciou je technicky veľmi náročnou úlohou. Väčšina projektov v súčasnosti optimalizuje roboty čisto v simulácii, bez snahy o ich konštrukciu v realite. Existujú však projekty, kde sa výskumníkom podarilo vyvinuté roboty po skončení simulovanej evolúcie skonštruovať, v niektorých prípadoch dokonca automatizovaným procesom. V nasledujúcom texte sa budeme venovať existujúcim projektom z hľadiska platforiem, ktoré používajú, od tých používajúcich konkrétneho robota s jednoduchou morfológiou, cez zložitejšie roboty so zložitejšou morfológiou až po projekty umožňujúce skonštruovať výsledného robota podľa plánov optimalizovaných evolúciou.

Pri výbere architektúry robota stojí každý experimentátor pred otázkou, či pre svoje pokusy postaví vlastný prototyp robota, alebo použije niektorú z existujúcich platforiem. Druhé riešenie šetrí čas nutný na návrh a konštrukciu robota, poskytuje



však menšie možnosti na úpravy robota podľa konkrétnej riešenej úlohy. Toto obmedzenie sa však dá čiastočne prekonať použitím modulárnych a rozširiteľných robotických platforiem, ktoré si môže výskumník v prípade potreby upraviť podľa vlastných potrieb.

Jednou z najrozšírenejších takýchto platforiem je robot Khepera (na obr. 14 vo variante Khepera II). Robot bol predstavený v 90-tych rokoch práve pre účely výskumu a výuky. Robot má kruhový pôdorys s priemerom 5.5cm a výšku 3cm, jeho dve kolesá sú ovládané dvoma nezávislými servomotormi (otočnými v oboch smeroch) a súčasťou konštrukcie je aj 8 infračervených senzorov, schopných v aktívnom móde detekovať predmety do vzdialenosti asi 5cm a v pasívnom móde merať intenzitu okolitého osvetlenia. Modulárny návrh robota umožňuje pridávať moduly rozširujúce základnú funkcionality (príkladmi takýchto modulov je rádio vysielateľ/prijímač, rameno na uchopovanie predmetov alebo bezdrôtová kamera). Khepera sa vďaka svojej modularite a dostupnosti stala veľmi populárnou platformou na evolučné experimenty a je používaná vo veľkom počte výskumných centier po celom svete. Pre robota existuje viacero simulátorov (napríklad komerčne dostupný simulátor Webots alebo voľne dostupné simulátory Evorobot a YAKS). Khepera bola použitá vo veľkom množstve evolučných experimentov. Prehľad niektorých off-line ale aj on-line experimentov je dostupný napríklad v knižke Nolfiho a Floreana [40].

Nasledovníkom Khepery je e-puck, uvedený v roku 2006 (obr. 14). E-puck má dlhšiu výdrž ako Khepera (na jedno nabitie vydrží 3 hodiny oproti 40 minútam u Khepery), viac senzorov (8 infračervených senzorov, 3D akcelerometer, 3 mikrofóny, farebnú VGA kameru a IR prijímač) a viaceré možnosti výstupu (sadu LED diód a reproduktor). Rozmery robota sú o niečo väčšie ako u Khepery – priemer 7cm a výška 5.5cm. Funkcionality robota môže byť, tak isto ako u Khepery, rozšírená pridávaním špecializovaných modulov.

Ďalšou niekedy využívanou možnosťou je pri evolučných experimentoch použitie niektorej z bežne dostupných stavebníc, ako je napríklad LEGO, prípadne LEGO Mindstorms. Pre špecializované úlohy môže byť prínosné použiť vhodne upravené hračky. Jedným z príkladov je projekt využívajúci malé hračkárske autíčka na výskum automatického návrhu oponentov v pretekárskych počítačových hrách [54]. Autíčka sú ovládané bezdrôtovo pomocou diaľkového ovládania, ktoré je automaticky ovládané počítačom. Polohu autíčka sníma kamera umiestnená nad arénou.

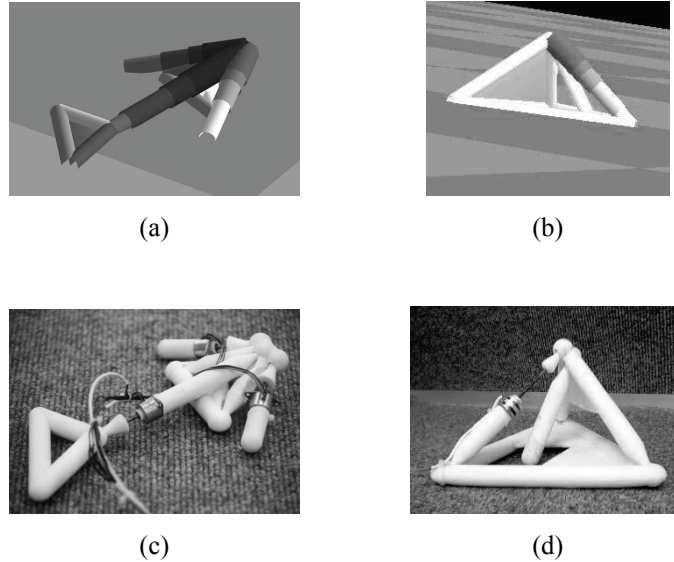
Jednou zo zložitejších platforiem pre robotické experimenty je robotický pes AIBO uvedený firmou Sony [15]. AIBO je štvornohý robot s 18 stupňami voľnosti a bohatou sadou senzorov (medzi inými napríklad video kamera, stereo mikrofón, detektor teploty, akcelerometer a sada kontaktných senzorov), dĺžkou približne 30cm a dostatočne výkonným procesorom na to, aby evolučné experimenty mohli prebiehať priamo v robotovi. Táto vlastnosť umožnila použiť robota AIBO v on-line evolučných experimentoch. V projekte Hornbyho a kol. [21], bol použitý genetický algoritmus na návrh čo najrýchlejšej chôdze robota po nerovnom povrchu. Fitness funkcia bola počítaná kompletne na robotovi pomocou údajov so senzorov robota. Celá evolúcia prebiehala plne automaticky, bez nutnosti akéhokoľvek zásahu človeka. Výsledkom experimentu bol spôsob chôdze, ktorý bol podľa autorov rýchlejší, ako najrýchlejší spôsob chôdze navrhnutý ľudským návrhárom.

Všetky projekty, ktoré sme popísali v predchádzajúcom texte, majú jednu spoločnú vlastnosť: konštrukcia robota je vždy pevná a predmetom evolúcie je iba ovládanie robota. Na konštrukcia robota môžeme nahliadnuť aj ako na obmedzenie pre evolučný algoritmus, s ktorým sa treba nejakým spôsobom vysporiadať, a nie ďalšou z vlastností, ktoré je možné optimalizovať a využiť na zvýšenie fitness funkcie. Existuje však aj rad projektov, ktoré sa snažia optimalizovať nielen ovládanie, ale aj morfológiu robota. Hlavnou inšpiráciou pre tieto projekty je samotná živá príroda. V evolučnej robotike ako prvý ukázal možnosť dosahovať zaujímavé výsledky s evolúciou morfológie Karl Sims [45] vo svojej práci zo začiatku 90-tych rokov. Sims však svoje organizmy vyvíjal výlučne v simulácii. Organizmy mali navyše veľmi bohatú morfológiu (napr. veľké množstvo typov kĺbov), čo ich robí ťažko skonštruovateľnými v realite. Projektom virtuálnych organizmov Karla Simsa však bolo inšpirovaných viacero projektov, z ktorých niektorým sa podarilo priviesť evolúciou navrhnuté organizmy k „životu“ aj v realite.

Prvé kroky v prechode zo simulácie do reality boli urobené v práci Funesa a Pollacka [16], kde bolo cieľom navrhnuť stavbu štruktúry postavenej zo stavebnice LEGO tak, aby spĺňala požadované vlastnosti. V tomto prípade síce ťažko ešte hovoriť o robotovi, pretože výsledné štruktúry boli statické a súčasťou evolúcie teda nebolo ovládanie. Práca však ukázala, že vývoj morfológie je uskutočniteľný a pripravila tak cestu ďalším projektom. Experimenty prebiehali v simulátore LEGO stavebnice vytvorenom špeciálne pre tento účel. Úlohou simulátora bolo vypočítať stabilitu štruktúry pri aplikovaní sily v niektorom jej bode v prítomnosti gravitácie. Pri meraní fitness funkcie bola navrhovaná LEGO štruktúra otestovaná v simulátore a v prípade, že štruktúra neskolabovala, niektorá jej vlastnosť bola použitá na určenie fitness funkcie (napríklad fitness funkciou pre návrh ramena žeriavu bola maximálna hmotnosť závažia, s ktorým rameno žeriavu neskolabuje). Výsledkom experimentov bolo viacero štruktúr, ktoré boli následne ručne postavené a ich funkčnosť otestovaná v realite.

Viacero prác sa pokúsilo skonštruovať v realite nielen statické štruktúry, ale pohyblivé roboty schopné plniť jednoduché úlohy. Jedným z príkladov je práca Hornbyho a kol. [22]. Hlavným cieľom projektu bolo síce študovať spôsoby genetickej reprezentácie robota (z tohto pohľadu sme sa venovali tejto práci v sekcii 4), niektoré výsledné roboty boli však úspešne skonštruované a dosahovali podobné výsledky ako roboty v simulácii.

Prvým projektom, ktorý prišiel s metódou automatickej konštrukcie robotov podľa návrhu optimalizovaného evolúciou je projekt GOLEM (Genetically Organized Lifelike ElectroMechanics) [33]. Roboty v tomto projekte boli vyvíjané off-line evolúciou, pomocou zjednodušeného simulátora, ktorý zabezpečil, že výsledné roboty budú mať rovnaké správanie v realite ako v simulácii. Robot je v projekte GOLEM reprezentovaný sústavou tyčiek, ktoré sú navzájom spojené voľnými guľovými kĺbmi na ich koncoch. Tyčky sa môžu spájať do ľubovoľnej štruktúry, takže napríklad pri spojení troch tyčiek je vytvorená pevná trojuholníková štruktúra. Robota ovláda neuronová sieť, ktorej štruktúra nie je nijako obmedzená, takže evolúcia môže vytvoriť nielen siete so spätným šírením, ale aj štandardné dopredné siete. Každý z neurónov môže byť zapojený na jednu z tyčiek. Takéto zapojenie zmení statickú tyčku

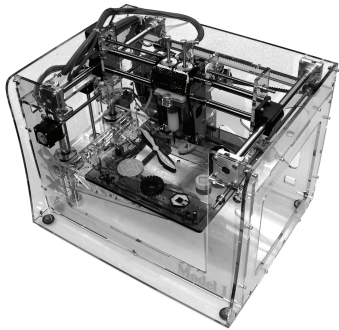


**Obr. 15.** Príklady robotov vyvinutých evolúciou v projekte GOLEM [33]. Roboty vyvinuté v simulácii (a, b) boli úspešne skonštruované v realite (c, d).

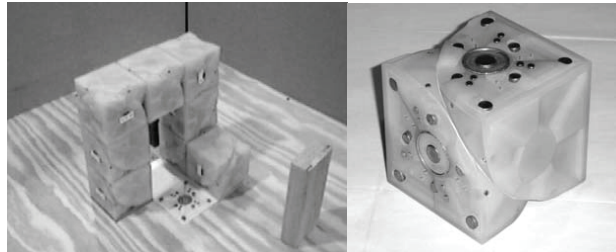
na dynamickú, ktorej dĺžka sa mení podľa aktuálnej hodnoty neurónu (v reálnom robotovi je toto chovanie realizované lineárnym motorom). Týmto spôsobom môže neurónová sieť ovládať pohyb robota. Cieľom evolúcie je vyvinúť takú neurónovú sieť a konštrukciu robota, aby výsledný robot bol schopný čo najrýchlejšieho pohybu jedným smerom. Zaujímavá na projekte GOLEM je však metóda, akou boli výsledné roboty konštruované. Na ich výrobu bola použitá automatická 3D-tlačiareň schopná vyrábať 3D objekty ľubovoľnej štruktúry podľa zadaného plánu. Tlačiareň postupuje pri konštrukcii zdola nahor po úrovniach a v každej úrovni ukladá vrstvu plastu na predchádzajúcu vrstvu. V prípade potreby je automaticky postavené dočasné „lešenie“ na podporu častí, ktoré nie sú podporované inou časťou robota. Tlačiareň je takto schopná automaticky vyrobiť plastovú konštrukciu robota a na to, aby sa robot začal pohybovať, stačí vložiť motory a elektroniku (príklady výsledných robotov je možné vidieť na obr. 15).

Projekt GOLEM bol prvým projektom, ktorému sa podarilo vyvíjať evolúciou roboty takmer úplne automaticky. Nie je ťažké predstaviť si, že v budúcnosti by podobné automatické procesy nahradili ľudských dizajnérov. Keďže výpočtová sila počítačov je čoraz lacnejšia, dá sa predpokladať, že takýto vývoj by zároveň umožnil výrazne zefektívniť (a najmä zlacniť) proces návrhu robotov určených pre riešenie konkrétnej úlohy.

Prvým krokom k tomuto cieľu je proces výroby robota plne automatizovať. Jednou z ciest takejto automatizácie je použiť 3D-tlačiareň podobnú tej použitej v projekte GOLEM, avšak schopnú tlačiť aj také časti robota, ako vodiče, elektroniku a motory. Jednou z veľmi sľubných technológií v tomto smere je projekt Fab@Home.



**Obr. 16.** Tlačiareň 3D objektov v projekte Fab@Home [35].

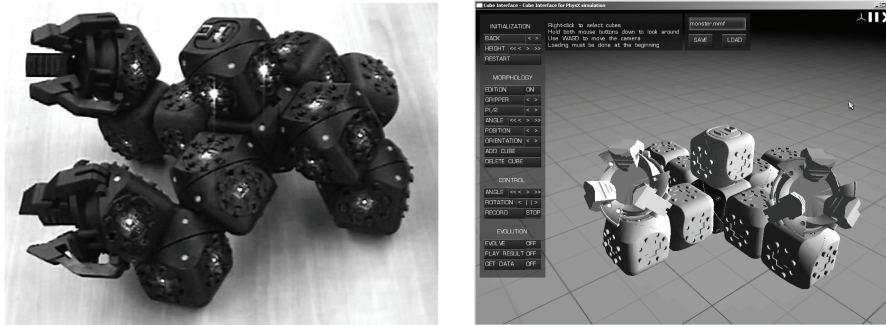


**Obr. 17.** Príklad samo-rekonfigurovateľného robota z práce Zykova a kol. [55].

Ide o akademický projekt otvorený širokej verejnosti, ktorého cieľom je vytvoriť 3D-tlačiareň schopnú vytlačiť kompletného robota. V súčasnosti je projekt v pokročilej fáze a okrem základnej schopnosti vytvárať plastové objekty boli už dosiahnuté aj niektoré z pokročilých techník, ako je napríklad výroba vodičov, jednoduchého motora, batérii alebo tranzistora [35] (vid' obr. 16).

Úplne iným prístupom k riešeniu problému, ako zostaviť robota podľa zadaného plánu, je použitie tzv. rekonfigurovateľných robotov. Rekonfigurovateľné roboty sú zložené z modulov, ktoré môžu byť navzájom zapojené, rozpojené alebo preusporiadané tak, že môžu reprezentovať širokú škálu rôznych morfológií. Samo-rekonfigurovateľné roboty sú navyše schopné takéto preusporiadanie svojej morfológie previesť samostatne. Príklad takéhoto systému nájdeme v práci Zykova a kol. [55]. Robot je zložený z 8. kociek, z ktorých každá obsahuje servomotor a je tak schopná otáčať jednu svoju polovicu voči druhej polovici. Každá kocka má okrem toho na stenách elektromagnety, ktorými je schopná pripájať sa k iným kockám (príklad je uvedený na obr. 17). Uvedený robot pozostáva síce len z 8. dielov, nie je však ťažké predstaviť si podobný systém pracujúci so stovkami kociek, ktorý by bol schopný riešiť zložitejšie úlohy. Evolúcia takého systému bola študovaná a my sme sa jej venovali v sekcii 5.4.

Jedným z problémov pri používaní rekonfigurovateľných robotov je potreba navrhnuť a skonštruovať hardvér pre jednotlivé moduly. Veľmi užitočnou pomôckou pre výskumníkov v tejto oblasti by bola robotická platforma uľahčujúca začiatky experimentovania v tomto prúde evolučnej robotiky (podobným spôsobom ako roboty Khepera uľahčujú experimentovanie s jednoduchšími robotmi s pevnou morfológiou). Jedným z pokusov o vytvorenie takejto architektúry je projekt Molecubes [56], ktorého súčasťou je otvorená hardvérová platforma a jej simulátor umožňujúci aj jednoduché evolučné experimenty (obr. 18). Stavebnica sa skladá z kociek s hranou veľkosti 5cm. Kocky plnia rôzne funkcie (zdroj energie, motor, kamera) a dajú sa skladať do ľubovoľne zložitých štruktúr použitím mechanického spoja. V predbežných experimentoch s touto stavebnicou boli úspešne vyvinuté napríklad jednoduché spôsoby pohybu vpred pre vopred danú konštrukciu robota.



**Obr. 18.** Príklad robota postaveného pomocou stavebnice Molecubes v realite (vľavo) a v simulácii (vpravo) [56].

## 7 Záver

Evolučná robotika a evolučné algoritmy samotné sú veľmi živou disciplínou, čo je doložené aj počtom každoročne publikovaných prác v tejto oblasti. Evolúcia robotov je študovaná na viacerých úrovniach – od evolúcie ovládania pre jednoduché roboty až po konštrukciu kompletných robotov, kde aj samotná konštrukcia robota je navrhnutá evolúciou. Evolúcia morfológie spolu s ovládaním má potenciál viesť k robotom omnoho lepšie adaptovaným na riešenie zadanej úlohy. Napriek technologickému pokroku a neustálej miniaturizácii je však konštrukcia týchto robotov v realite stále veľkou výzvou. Preto je dôležité, že začínajú vznikať projekty, ktoré majú ambíciu zjednodušiť konštrukciu robotov a tým výrazne uľahčiť vstup do poľa evolučnej robotiky ďalším výskumníkom [35, 56] (podobne, ako to robot Khepera umožnil pri vývoji ovládania robotov). Popri pokroku v konštrukcii robotov sa zlepšujú aj možnosti ich simulácie v počítači. Kým prví výskumníci boli často odkázaní na implementáciu vlastnej fyzikálnej simulácie, dnes už tomu tak nie je, a to aj vďaka veľkému množstvu voľne dostupných fyzikálnych simulátorov. Simulácia sa stáva bohatšou a rýchlejšou, čo je pri výpočtoch v evolučnej robotike špeciálne dôležité. Aj vďaka týmto pokrokom je evolučná robotika na sľubnej ceste k svojmu cieľu - priniesť prvýkrát v ľudskej histórii prístroj, schopný navrhovať ďalšie prístroje za nás.

## Literatúra

- [1] Beale, R. and Jackson, T., *Neural computing: an introduction*, IOP Publishing Ltd., Bristol, UK, 1990.
- [2] Bentley, P. J., editor, *Evolutionary Design by Computers*, Morgan Kaufmann, San Mateo, California, 1999.
- [3] Bongard, J., Evolving modular genetic regulatory networks, in *Proceedings of the 2002 Congress on Evolutionary Computation, CEC '02*, volume 2, pages 1872-1877, 2002.

- 
- [4] Bongard, J., Zykov, V., and Lipson, H.: Resilient machines through continuous self-modeling. *Science* **314** (2006) 1118.
  - [5] Bongard, J., Behavior chaining: incremental behavioral integration for evolutionary robotics, in *Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems*, M. A., pages 64-71, MIT Press, Cambridge, MA, 2008.
  - [6] Chaumont, N., Egli, R., and Adami, C.: Evolving virtual creatures and catapults. *Artificial Life* **13** (2007) 139.
  - [7] Cliff, D., Husbands, P., and Harvey, I.: Explorations in evolutionary robotics. *Adapt. Behav.* **2** (1993) 73.
  - [8] Cliff, D. and Miller, G. F., Co-evolution of pursuit and evasion II: Simulation methods and results, in *From Animals to Animats 4: Proceedings of the Fourth International Conference of Simulation of Adaptive Behavior*, MIT Press, 1996.
  - [9] Dawkins, R., *The Blind Watchmaker*, W. W. Norton & Company, Inc., New York, NY, USA, 1986.
  - [10] Dawkins, R., *The Selfish Gene*, Oxford University Press, 1990.
  - [11] Dawkins, R., *The Ancestor's Tale : A Pilgrimage to the Dawn of Life*, Orion Pub Co, 2004.
  - [12] Dill, M., Wolf, R., and Heisenberg, M.: Visual pattern recognition in drosophila involves retinotopic matching. *Nature* **365** (1993) 751.
  - [13] Eggenberger, P., Evolving morphologies of simulated 3D organisms based on differential gene expression, in *Proceedings of the Fourth European Conference on Artificial Life*, pages 205-213, Cambridge, MA, 1997, MIT Press.
  - [14] Ficici, S. G., Watson, R. A., and Pollack, J. B., Embodied evolution: a response to challenges in evolutionary robotics, in *Eighth European Workshop on Learning Robots*, pages 14-22, 1999.
  - [15] Fujita, M. and Kitano, H.: Development of an autonomous quadruped robot for robot entertainment. *Auton. Robots* **5** (1998) 7.
  - [16] Funes, P. and Pollack, J., Computer evolution of buildable objects, in *Fourth European Conference on Artificial Life*, pages 358-367, MIT Press, 1997.
  - [17] Goldberg, D. E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Professional, 1989.
  - [18] Gruau, F., Cellular encoding for interactive evolutionary robotics, in *Fourth European Conference on Artificial Life*, pages 368-377, Cambridge, MA, USA, 1997, MIT Press.
  - [19] Harding, S. and Miller, J. F., Evolution in materio : A real-time robot controller in liquid crystal, in *Evolvable Hardware*, pages 229-238, 2005.
  - [20] Holland, J. H., *Adaptation in natural artificial systems*, University of Michigan Press, Ann Arbor, 1975.

- 
- [21] Hornby, G. S., Fujita, M., and Takamura, S., Autonomous evolution of gaits with the sony quadruped robot, in *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1297-1304, Morgan Kaufmann, 1999.
- [22] Hornby, G. S., Lipson, H., and Pollack, J. B.: Generative representations for the automated design of modular physical robots. *IEEE Transactions on Robotics and Automation* **19** (2003) 703.
- [23] Jakobi, N.: Evolutionary robotics and the radical envelope-of-noise hypothesis. *Adapt. Behav.* **6** (1997) 325.
- [24] Jakobi, N., Half-baked, ad-hoc and noisy: Minimal simulations for evolutionary robotics, in *Fourth European Conference on Artificial Life*, pages 348-357, MIT Press, 1997.
- [25] Komosinski, M. and Ulatowski, S., Framsticks: Towards a simulation of a nature-like world, creatures and evolution, in *Proceedings of 5th European Conference on Artificial Life (ECAL99)*, pages 261-265, Springer-Verlag, 1999.
- [26] Koza, J. R., *Genetic Programming: On the Programming of Computers by Means of Natural Selection* (Complex Adaptive Systems), The MIT Press, 1992.
- [27] Koza, J. R., *Genetic Programming II: Automatic Discovery of Reusable Programs*, MIT Press, Cambridge Massachusetts, 1994.
- [28] Koza, J. R. et al., *Genetic Programming IV: Routine Human-Competitive Machine Intelligence*, Kluwer Academic Publishers, 2003.
- [29] Krcah, P., Towards efficient evolutionary design of autonomous robots, in *ICES '08: Proceedings of the 8th international conference on Evolvable Systems: From Biology to Hardware*, pages 153-164, Berlin, Heidelberg, 2008, Springer-Verlag.
- [30] Lassabe, N., Luga, H., and Duthen, Y., A new step for evolving creatures, in *IEEE-ALife'07, Honolulu, Hawaii*, pages 243-251, 2007, IEEE.
- [31] Lewis, M. A., Fagg, A. H., and Solidum, A., Genetic programming approach to the construction of a neural network for control of a walking robot, in *IEEE International Conference on Robotics and Automation*, pages 2618-2623, IEEE Press, 1992.
- [32] Lindenmayer, A.: Mathematical models for cellular interactions in development: Parts I and II. *Journal of Theoretical Biology* **18** (1968) 300.
- [33] Lipson, H. and Pollack, J. B.: Automatic design and manufacture of robotic lifeforms. *Nature* **406** (2000) 974.
- [34] Lund, H. H., Miglino, O., Pagliarini, L., Billard, A., and Ijspeert, A., Evolutionary robotics | a children's game, in *Proceedings of IEEE 5th International Conference on Evolutionary Computation*, pages 154-158, IEEE Press, 1998.
- [35] Malone, E. and Lipson, H.: Fab@home: The personal desktop fabricator kit. *Rapid Prototyping Journal* **13** (2007) 245.

- 
- [36] de Margerie, E., Mouret, J. B., Doncieux, S., and Meyer, J.-A., Artificial evolution of the morphology and kinematics in a flapping-wing mini UAV. *Bioinspir Biomim* **2** (2007) 65.
- [37] McCulloch, W. and Pitts, W.: A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biology* **5** (1943) 115.
- [38] Miconi, T. and Channon, A., An improved system for artificial creatures evolution, in *Artificial Life X: Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems*, pages 255-261, Bloomington, IN, USA, 2006, MIT Press.
- [39] Miconi, T., In silicon no one can hear you scream: Evolving fighting creatures., in *EuroGP*, pages 25-36, Springer, 2008.
- [40] Nolfi, S. and Floreano, D., *Evolutionary Robotics. The Biology, Intelligence, and Technology of Self-organizing Machines*, MIT Press, Cambridge, MA, 2001, 2001 (2nd print), 2000 (1st print).
- [41] Ray, T. S., Aesthetically evolved virtual pets., in *Artificial Life 7 Workshop Procs*, pages 158-161, 2000.
- [42] Rechenberg, I., *Evolutionsstrategie: optimierung technischer systeme nach prinzipien der biologischen evolution*, Frommann-Holzboog, 1973.
- [43] Reisinger, J. and Miikkulainen, R., Acquiring evolvability through adaptive representations, in *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 1045-1052, New York, NY, USA, 2007, ACM.
- [44] Shim, Y.-S. and Kim, C.-H.: Evolving physically simulated flying creatures for efficient cruising. *Artificial Life* **12** (2006) 561.
- [45] Sims, K., Evolving virtual creatures, in *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 15-22, New York, NY, USA, 1994, ACM Press.
- [46] Sims, K.: Evolving 3d morphology and behavior by competition. *Artificial Life* **1** (1994) 353.
- [47] Spector, L., *Automatic Quantum Computer Programming: A Genetic Programming Approach (Genetic Programming)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
- [48] Spector, L., Klein, J., and Feinstein, M., Division blocks and the open-ended evolution of development, form, and behavior, in *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 316-323, New York, NY, USA, 2007, ACM.
- [49] Spector, L., Clark, D. M., Lindsay, I., Barr, B., and Klein, J., Genetic programming for finite algebras, in *GECCO '08: Proceedings of the 10th annual conference on Genetic and evolutionary computation*, pages 1291-1298, New York, NY, USA, 2008, ACM.



- 
- [50] Stanley, K. and Miikkulainen, R., A taxonomy for artificial embryogeny. *Artificial Life* **9** (2003) 93.
  - [51] Stanley, K. O. and Miikkulainen, R.: Competitive coevolution through evolutionary complexification. *Journal of Artificial Intelligence Research* **21** (2004) 63.
  - [52] Studer, G. and Lipson, H., Spontaneous emergence of self-replicating structures in molecule automata, in *Artificial Life X: Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems*, pages 227-233, Bloomington, IN, USA, 2006, MIT Press.
  - [53] Thompson, A., Artificial evolution in the physical world, in *Evolutionary Robotics: From Intelligent Robots to Artificial Life (ER'97)*, pages 101-125, AAI Books, 1997.
  - [54] Togelius, J., Lucas, S. M., and Nardi, R. D., Computational intelligence in racing games., in *Advanced Intelligent Paradigms in Computer Games*, volume 71 of *Studies in Computational Intelligence*, pages 39-69, Springer, 2007.
  - [55] Zykov, V., Mytilinaios, E., Adams, B., and Lipson, H.: Self-reproducing machines. *Nature* **435** (2005) 163.
  - [56] Zykov, V., William, P., Lassabe, N., and H., L., Molecubes extended: Diversifying capabilities of open-source modular robotics, in *IROS Workshop on Self-Reconfigurable Robots, Systems and Applications*, 2008.



# Darwinovská evolúcia ako algoritmus

Vladimír KVASNIČKA a Jiří POSPÍCHAL<sup>1</sup>

**Abstrakt.** V práci je prezentovaná moderná paradigma darwinovskej evolúcie, ktorá ju chápe ako univerzálny optimalizačný algoritmus, aplikovateľný nielen v biologických vedách, ale aj v rôznych oblastiach tak exaktných, ako aj sociálnych a behaviorálnych vied.

*Motto:*

*Tak, ako to vidím ja, centrálnym problémom evolúcie je mechanizmus, pomocou ktorého druhy môžu spojiť najst' cestu z nižších na vyššie vrcholy.*

*(Sewall Wright [51])*

## 1 Úvod

Evolučná teória spočíva v jednoduchej myšlienke, že život je v čase premenlivý a mladšie formy života nasledujú za staršími formami. Táto myšlienka už existovala pred Darwinom, ale až on a jeho nasledovníci ju rozvinuli do podoby umožňujúcej vysvetliť nielen rôznorodosť foriem života, ale aj adaptáciu živých organizmov vzhľadom k meniacemu sa okoliu. Podľa filozofujúceho biológa Mayra [40], koncepcia evolúcie spočíva v kombinácii nasledujúcich piatich princípov:

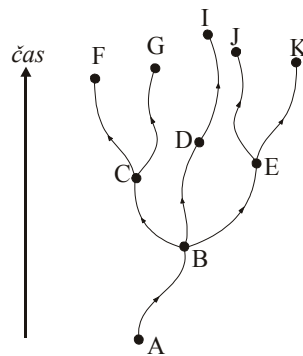
1. Živé organizmy sú *časovo premenné*, podliehajú evolučným zmenám.
2. Evolučné zmeny majú charakter *vetvenia*. Súčasný druh sú nasledovníci predchodcov existujúcich v minulosti. Podobne, súčasný druh môže tvoriť predchodcu jednému alebo viacerým budúcim druhom (pozri obr. 1).
3. Nové druhy vznikli tak, že *populácia sa rozštiepila* na izolované podpopulácie, ktoré potom navzájom evolučne divergovali za vzniku nových druhov.
4. Evolučné zmeny sú *postupné – graduálne*. Niektoré organizmy, ktoré sa odlišujú od svojich rodičov, sú schopné prežitia. Z týchto organizmov si niektoré vytvoria svoju vlastnú podpopuláciu zloženú z podobných organizmov, v ktorej sa zachovávajú ich špecifické odlišnosti.

---

<sup>1</sup> Ústav aplikovanej informatiky, Fakulta informatiky a informačných technológií STU, 84216 Bratislava, E-mail: kvasnicka@fiit.stuba.sk, pospichal@fiit.stuba.sk.

5. Mechanizmom adaptívnej zmeny je *prírodný výber*.

Darwinovým veľkým úspechom [7] bolo, že pomerne rýchlo presvedčil svojich súčasníkov o existencii evolúcie, ktorej priebeh sa dá názorne reprezentovať stromom života (pozri obr. 1). Bol evolučným gradualistom, ktorý veril v nahromadenie (akumuláciu) malých zmien v priebehu dlhých časových úsekov, ktoré sú interpretované ako evolučná zmena. Toto jeho „gradualistické“ evolučné presvedčenie bolo všeobecne prijaté až v prvej polovici 20. storočia hlavne zásluhou trojice matematikov – štatistikov Fishera, Wrighta a Haldanea, ktorí spojili populačnú genetiku s evolučnou teóriou za vzniku modernej evolučnej teórie nazývanej *neodarwinizmus*. Dôležitosť izolácie populácie vznikajúceho nového druhu zostala kontroverznou otázkou evolučnej teórie ešte o mnoho dlhšie. Tento pohľad na mechanizmus vzniku nových druhov sa stal všeobecne akceptovaným až zásluhou prác Mayra z polovice 20. storočia.



**Obr. 1.** Schematické znázornenie vzniku nových druhov (označených písmenami A, B, ...) v priebehu evolúcie. Druh A je predchodcom druhu B, druh B je spoločným predchodcom nových druhov C, D a E. Priebeh evolúcie sa dá názorne reprezentovať pomocou tohto „stromu života“.

Súčasná evolučná biológia stojí pred dvoma dôležitými problémami, ktoré sa snaží vysvetliť. Prvým problémom je *neobyčajná rôznorodosť* foriem života na našej planéte. Až darwinizmus poskytol vysvetlenie tejto skutočnosti, kde vedľa seba existuje mnoho druhov, ktoré sú morfológicky zásadne odlišné a ktoré na prvý pohľad nemajú medzi sebou skoro nič spoločné. Druhý, nemenej dôležitý problém, je *adaptácia*. Štruktúrovaná zložitosť organizmov, ich adaptácia na životné prostredie, je rovnako prekvapujúca ako rôznorodosť organizmov. Zmyslové orgány sú klasickým príkladom zložitej a jemne naladenej adaptácie. Medzi najväčšie úspechy darwinizmu patrí prírodovedecké vysvetlenie adaptácie pomocou prírodného výberu založeného na prežívaní lepšie prispôsobených organizmov (s väčším fitness) a dedení získaných vlastností. Evolučné zmeny sú určené variáciami v populácii, rozdielmi vo fitness spôsobenými týmito variáciami a genetickým prenosom. Adaptívne zmeny sa uskutočňujú aj napriek často zmieňovanej skutočnosti, že mechanizmy spôsobujúce variabilnosť populácie (hlavne mutácie) nesúvisia priamo s výhodnosťou adaptívnych

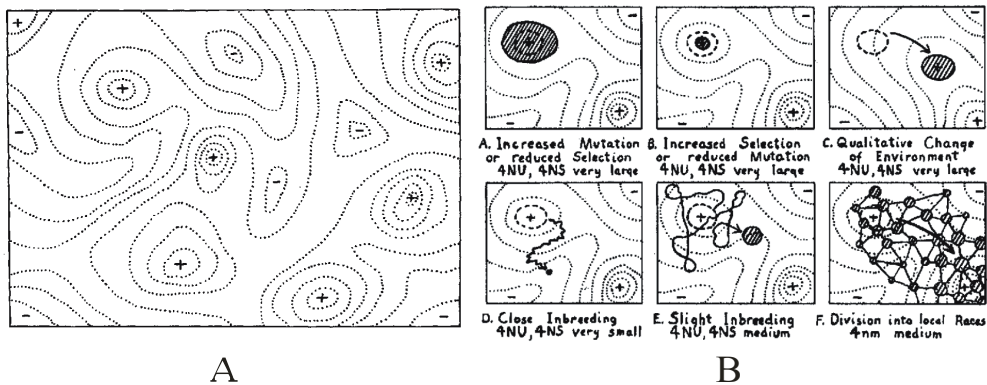
zmien. Tak napr. adaptívny posun farby peria vtáka prebieha postupne, v priebehu mnohých generácií a závisí od kumulatívneho výberu. Ak výber má vysvetliť adaptáciu, potom musí byť kumulatívny. Inovácia je výsledkom dlhej postupnosti epizód výberu, pretože pravdepodobnosť požadovanej náhlej adaptačnej zmeny prostredníctvom mutácie je zanedbateľne malá. Súčasne s adaptačnými zmenami prebieha aj proces diverzifikácie populácie, adaptácia na nové prostredie a postupná divergencia jednotlivých podpopulácií od seba.

Určité pochybnosti pri akceptovaní Darwinovej evolučnej teórie neustále pretrvávajú [4] v súvislosti s vysvetlením mechanizmov vzniku nových zložitostí. Tieto zložitosti majú komplexnú štruktúru, pričom ich architektúra je obvykle ireducibilného charakteru, t.j. nemôžeme odstrániť ani jeden element štruktúry bez straty funkčnosti celej zložitosti. V tomto smere existuje v súčasnosti najviac námietok proti evolučnej teórii, ako principiálne neschopnej vysvetliť pomocou prírodného výberu a adaptácie vznik nových kvalít, kreativitu živej prírody. Evolučný vznik nových kvalít, nových zložitostí, je obvykle sprevádzaný aj veľkou prestavbou genotypu organizmu, zmenou jeho modulárnej štruktúry a interakcie jednotlivých modulov medzi sebou. Gradualizmus klasického darwinizmu (aj neodarwinizmu) vychádza implicitne z predpokladu, že štruktúra genotypu je nemenná, môže sa meniť len zloženie jednotlivých alel genotypu, bez zmeny jeho modulárnej štruktúry a pod. Tento problém stojí pred moderným darwinizmom [5,30,31,49] ako dôležitá neriešená úloha, ktorej riešenie už bude vyžadovať úzku spoluprácu s informatikmi modelujúcimi *in-silico* evolúciu. Evolučný nárast zložitosti je formálne zložitý problém, ktorý sa nedá riešiť len na verbálnej kvalitatívnej úrovni, ale vyžaduje spoluprácu s teoretickými biológmi a informatikmi, ktorí sa venujú počítačovým simuláciám evolúcie.

V tejto sekcii uvedieme v chronologickom poradí vybrané názory, koncepcie na darwinovskú evolúciu, ktoré podľa názoru autorov zohrali významnú úlohu pri vzniku a rozvíjaní univerzálneho darwinizmu. Ako informatici pracujúci v umelej inteligencii a v kognitívnej vede, budeme preferovať také názory, ktoré viedli k pohľadu na evolúciu ako na univerzálne platný optimalizačný algoritmus, ktorého „materiálna“ realizácia - implementácia je nezávislá od pôvodnej biologickej realizácie evolúcie.

### 1.1 Wrightov pohľad na evolúciu ako na optimalizáciu [51]

V prírodnom výbere má dominantné postavenie koncepcia *fitnes* organizmov, ktorý je špecifikovaný ako relatívna schopnosť prežitia a reprodukovania sa v danom *prostredí* a v danej *populácii*. Vo všeobecnosti, *fitnes* je chápaný nielen ako atribút genotypu ale aj populácie. Tento postulát odráža naše predstavy o *fitnes* ako o veličine, ktorá kvantifikuje nielen prežívanie a schopnosť reprodukcie jednotlivých organizmov, ale aj ich vzájomnú interakciu v populácii. Vlastnosti určujúce *fitnes* sa aspoň čiastočne prenášajú z rodičov na potomkov, t.j. sú zafixované v genotype jedincov populácie. Potomkovia rodičov s väčším *fitnes* budú početnejší a taktiež aj "silnejší" ako potomkovia "slabších" rodičov.



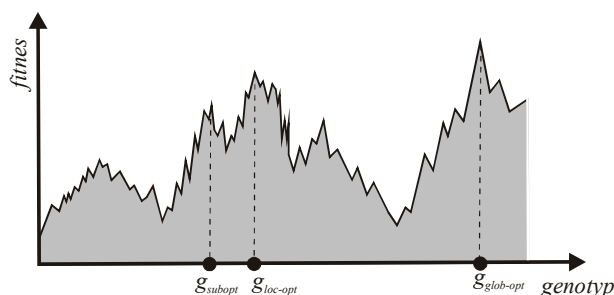
**Obr. 2.** Jeden z najslávnejších obrázkov neodarwinizmu, ktorý pochádza od amerického populačného genetika a evolučného biológa Sewalla Wrighta [51]. (A) Znárodňuje povrch fitness funkcie obsahujúci množstvo lokálnych extrémov (miním a maxim). Evolúciu charakterizoval ako adaptívny proces (v matematickej terminológii "optimalizačný proces"), kde genotyp populácie je adaptovaný tak, že dosiahne lokálne (snáď aj globálne) optimum – maximum na povrchu fitness funkcie. (B) Wright navrhoval rôzne mechanizmy, ako prekonať hlboké údolia medzi dvoma maximami. Svoju pozornosť v tomto smere sústredil hlavne na možnosť genetického driftu malej populácie, ktorá je dočasne izolovaná od hlavnej populácie a kde drift nadobúda význam s poklesom významu prirodzeného výberu. V takomto evolučnom „taviacom kotli“ dochádza k neočakávaným zmenám genotypu, čo sa môže prejaviť aj preskokom z jedného maxima na druhé maximum. Týmto spôsobom vysvetľoval Wright mechanizmus toho (pozri motto tohto príspevku), ako sa genotyp populácie môže presunúť z nižšieho vrcholu na vyšší vrchol povrchu fitness.

Koncepcia *povrch fitness* (pôvodne „adaptive fitness landscape“) (pozri obr. 2, diagramy A a B) pochádza od jedného zo zakladateľov neodarwinizmu Sewalla Wrighta, ktorý v r. 1932 navrhol [51] túto koncepciu, ktorá predstavuje jednoduchý heuristický prístup ako vizualizovať proces evolúcie populácie organizmov. Evolúcia populácie môže byť pomocou tohto obrázku interpretovaná ako pohyb populácie smerom k vrcholu s najväčším fitness. V dôsledku existencie mutácií, tento pohyb populácie na povrchu fitness má stochastický charakter. V tejto súvislosti sa zdôrazňuje, že práve vďaka mutáciám, organizmy populácie majú šancu „preskúmať“ väčšiu oblasť na povrchu fitness a tak potenciálne dosiahnuť v procese evolúcie globálne maximum na povrchu fitness. To znamená, že evolučný proces môžeme charakterizovať ako *optimalizačný proces*. Formálne túto skutočnosť môžeme vyjadriť pomocou diskrétného optimalizačného problému

$$\mathbf{g}_{opt} = \arg \max_{\mathbf{g} \in \{a,b,\dots\}^n} f(\mathbf{g}) \quad (1)$$

kde  $f(\mathbf{g})$  je fitness funkcia, ktorá priradí každému genotypu  $\mathbf{g}$  zostrojenému nad abecedou  $\{a,b,\dots\}$  kladné reálne číslo - fitness, výsledné riešenie  $\mathbf{g}_{opt}$  reprezentuje genotyp nachádzajúci sa na globálnom maxime povrchu funkcie fitness. Evolučným cieľom organizmov je optimálna adaptácia v danom prostredí. Vlastnosti povrchu fitness (extrémna multimodálnosť) určujú ako ťažko sa požadované optimálne riešenie nájde „prirodzeným“ evolučným procesom. Vychádzajúc z tejto metafory povrchu

fitness, Wright formuloval aj ciele evolúcie takto „*Ústredným problémom evolúcie je navrhnúť taký mechanizmus, ktorým organizmy môžu nájsť spojitú cestu z nižšie položeného vrcholu na vyššie položený vrchol*“ (pozri obr. 3).



**Obr. 3.** Povrch fitness bol prvýkrát použitý v Darwinovej teórii americkým biológom Sewalom Wrightom v r. 1932 [51], ktorý charakterizoval evolúciu ako optimalizačný problém (v jeho terminológii adaptačný problém) nad povrchom fitness, ktorý je charakterizovaný funkciou  $f$ . Výsledný genotyp populácie je priradený bodu – lokálne optimálnemu genotypu  $g_{loc-opt}$  – s vysokou (ale nie najväčšou) hodnotou fitness. Globálne optimálne riešenie označené  $g_{glob-opt}$  reprezentuje také riešenie, ktoré má maximálnu funkčnú hodnotu na povrchu fitness. Obrazne povedané, Darwinova evolúcia musí mať prostriedky ako prekonať hlboké údolie pri pohybe z suboptimálneho riešenia  $g_{subopt}$  k blízkemu lokálne-optimálnemu riešeniu  $g_{loc-opt}$ . Pretože Darwinova evolúcia je stochastický optimalizačný algoritmus prebiehajúci na povrchu fitness funkcie, pravdepodobnosť získania globálne optimálneho stavu nie je stopercentná, nie vždy je toto riešenie dosiahnuteľné z danej počiatočnej populácie a v reálnom čase.

## 1.2 Hollandov genetický algoritmus

Genetické algoritmy boli vynájdené informatikom Johnom Hollandom [25] počiatkom 70-tych rokov minulého storočia. Po určitej nábehovej 10-ročnej perióde rozpakov a mlčania v komunite informatikov, stali sa jednou z najbúrlivejšie rozvíjajúcich sa oblastí informatiky a umelej inteligencie [15,37]. Možno konštatovať, že spolu s neurónovými sieťami tvoria jadro novovznikajúcej oblasti nazývanej *počítačová inteligencia*, ktorá je schopná riešiť praktické inžinierske problémy z informačných technológií, ktoré vyžadujú vysoký stupeň „inteligencie“<sup>2</sup>. V tejto podkapitole ukážeme základné princípy genetických algoritmov.

Genetický algoritmus nemanipuluje priamo so živými organizmami, ale s ich genetickou reprezentáciou. Pôvodný Hollandov názov pre túto reprezentáciu bol „chromozóm“, žiaľ táto pôvodná terminológia vyvolávala medzi biológmi rôzne asociácie s čisto biologickým významom tohto termínu, ktoré nesúviseli s problematikou genetického algoritmu, preto budeme tieto informačné entity nazývať „replikátory“. Postulujeme, že replikátory majú schopnosť sa rozmnožovať – replikovať, a to buď unárne (asexuálne) alebo binárne (sexuálne). Pri unárnej replikácii

<sup>2</sup> V modernej informatike sa pod termínom „inteligentný systém“ rozumie taký „agent“, ktorého kognitívny orgán (reprezentovaný *subsymbolickou* neurónovou sieťou alebo *symbolickým* expertným systémom) je schopný adaptácie (učenia) pri zmene prostredia alebo cieľov, ktoré sa spontánne vynárajú pri pohybe v tomto prostredí.

sa zúčastňuje procesu jeden replikátor - rodič, zatiaľ čo pri binárnej replikácii sa zúčastňujú procesu dva replikátory – rodičia. Rodičia (rodič) sú kvázináhodne vybraní z populácie v závislosti na ich fitness (replikátory s väčším fitness s väčšou pravdepodobnosťou vstupujú do replikácie) a produkujú nové replikátory - potomkov. Budeme rozlišovať tieto tri zložky replikačného procesu:

- (1) *selekcia* rodičov,
- (2) *replikácia* rodičov, pričom vznikajú potomkovia, a
- (3) *návrat* potomkov do populácie.

V prvom kroku binárnej replikácie sa vyberú pomocou stochastického operátora  $O_{select}$  z populácie  $P$  dva replikátory  $\mathbf{x}_1^{old} = O_{select}(P)$ ,  $\mathbf{x}_2^{old} = O_{select}(P)$  tak, že pravdepodobnosť ich výberu je úmerná ich fitness. V druhom kroku použitím stochastického operátora replikácie  $O_{repli}$  z rodičov dostaneme potomkov  $\mathbf{x}_1^{new}$  a  $\mathbf{x}_2^{new}$ .

$$(\mathbf{x}_1^{new}, \mathbf{x}_2^{new}) = O_{repli}(\mathbf{x}_1^{old}, \mathbf{x}_2^{old})$$

V unárnej (asexuálnej) replikácii sa na tvorbe potomkov podieľa len jeden replikátor – rodič. Formálne, tento proces môžeme vyjadriť takto

$$\mathbf{x}^{old} = O_{select}(P) \text{ a } \mathbf{x}^{new} = O_{repli}(\mathbf{x}^{old})$$

V prvom kroku je kvázináhodne vybraný rodič, v druhom aplikáciou stochastického operátora reprodukcie  $O_{repro}$  sa z replikátora - rodiča vytvorí replikátor – potomok. Prečo hovoríme, že operátor reprodukcie replikátorov je stochastický? Tento operátor z rodiča (rodičov) vyrobí potomka (potomkov) určitým procesom, ktorý nie je striktno deterministický, takže potomkovia nie sú presnými kópiami rodičov, ale majú určitú malú variabilitu vzhľadom k rodičom, čím je zabezpečená variabilita populácie. V treťom kroku sa rieši problém návratu potomka do populácie. Pseudopascalovský kód takto určenej Darwinovej evolúcie je daný nasledujúcim algoritmom:

#### Algoritmus 1.

```

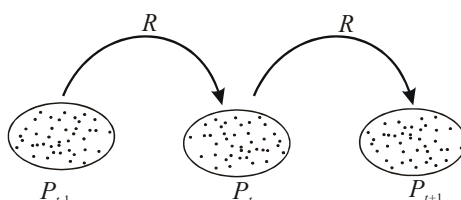
P:=náhodne vygenerovaná populácia replikátorov;
t:=0;
while t<tmax do
begin t:=t+1;
    Q:=∅;
    while |Q|<|P| do
    begin x1:=Oselect(P); x2:=Oselect(P);
        (x1', x2') := Orepli(x1, x2);
        Q:=Q∪{x1', x2'};
    end;
    P:=Q;
end;

```

Algoritmus je inicializovaný náhodnou generáciou populácie  $P$ . Evolúcia populácie je obsiahnutá vo vnútri **pokiaľ-urob**-cyklu, ktorý sa opakuje  $t_{max}$ -krát. V rámci jednej epochy sa vytvorí nová populácia  $Q$ , ktorá sa tvorí opakovanou replikáciou dvoch



rodičovských genotypov, ktoré majú vysoký fitness. Nová populácia  $Q$  nahradí pôvodnú populáciu  $P$  (pozri obr. 4).



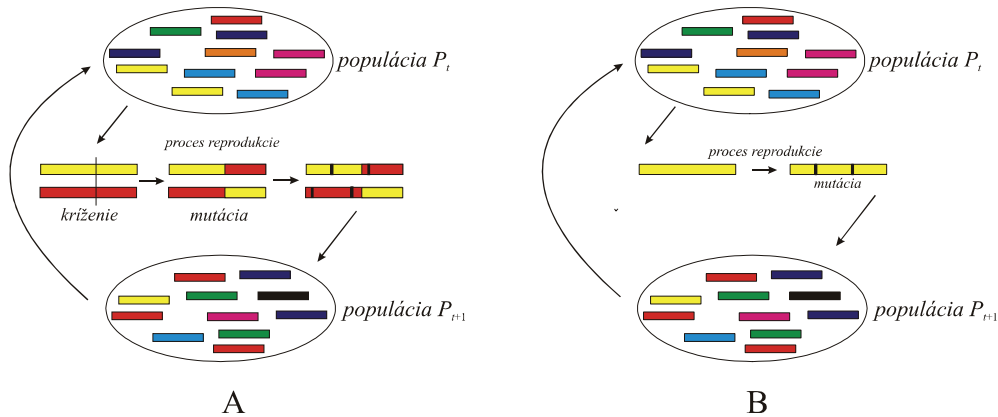
**Obr. 4.** Schématické znázornenie rekurzívneho charakteru Darwinovej evolúcie, nová populácia sa vytvára z predchádzajúcej populácie aplikáciou stochastického operátora reprodukcie  $R$ .

Darwinova evolúcia môže byť interpretovaná ako *rekurentný proces*, v ktorom nasledujúca populácia je vytvorená reprodukciou predchádzajúcej populácie (pozri obr. 4)

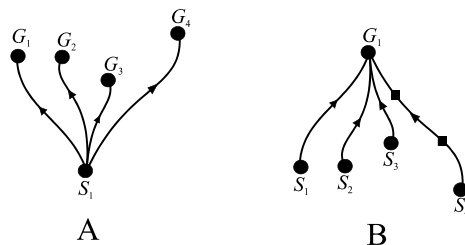
$$P_{t+1} = R(P_t)$$

Funkcia  $R$  kvázináhodne (vzhľadom k fitness replikátorov) priradí k populácii  $P_t$  nasledujúcu populáciu  $P_{t+1}$ . Cieľom tejto rekurzívnej – evolúcie darwinovského systému – je spontánna emergencia replikátorov s vysokým fitness, ktoré sa vyvinú z počiatočných, náhodne generovaných replikátorov, tvoriacich populáciu  $P_0$ . Hybnou silou evolúcie darwinovského systému je (1) prírodný výber a (2) spontánne mutácie. Z tohto pohľadu môžeme povedať, že evolúcia systému prebieha na hrane chaosu a poriadku. Mechanizmy selekcie najlepšie prispôbených replikátorov pomocou prírodného výberu a neustála zmena replikátorov v populácii dôsledkom stochastických mutácií pri replikačnom procese, sú hlavným zdrojom evolúcie systému, postupného zvyšovania priemerného fitness v celej populácii (pozri obr. 5).

Zatiaľ sme bližšie nešpecifikovali reprodukčný proces. Holland zaviedol pre binárnu (sexuálnu) reprodukciu proces kríženia, ktorý spočíva v tom, že z dvoch rodičovských replikátorov sa kopírovaním vytvorí dva nové replikátory potomkov, avšak pri tomto kopírovanom procese prvá časť sa kopíruje z 1. rodiča, zatiaľ čo nasledujúca časť sa kopíruje z 2. rodiča. Táto možnosť použitia časti replikátora 2. rodiča pri vytváraní nového replikátora sa využíva len s určitou pravdepodobnosťou (preto môže nastať situácia, že replikátor potomka je kópiou replikátora len jedného rodiča). Súčasťou reprodukčného procesu sú aj tzv. mutácie, ktoré pri tvorbe replikátorov potomkov vytvárajú s malou pravdepodobnosťou náhodné zmeny, pozri obr. 5, diagram A. Genetický algoritmus s unárnou (asexuálnou) reprodukciou je zjednodušenou verziou predošlej verzie s binárnou reprodukciou, kríženie je v tomto prípade vylúčené (jeden rodič sa nemôže sám so sebou krížiť), pozri obr. 5, diagram B.



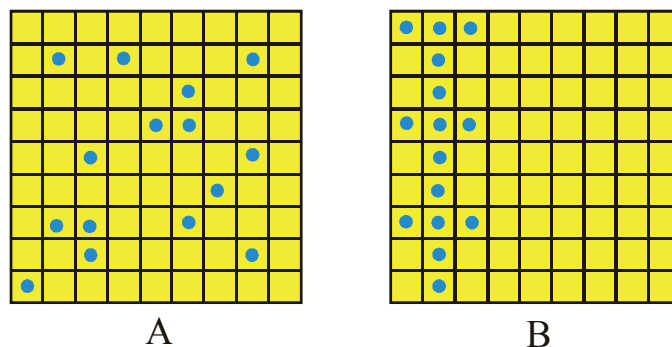
**Obr. 5. (A)** Schematické znázornenie genetického algoritmu s binárnou reprodukciou. Vychádza sa z aktuálnej populácie  $P_t$ , z ktorej opakovane vyberáme dvojice rodičovských replikátorov (pravdepodobnosť výberu je úmerná ich fitness). V reprodukčnom procese ich replikácia je založená na tzv. križení, ktoré spočíva v tom, že pri kopírovaní replikátorov najprv využívame pôvodnú časť replikátora 1. rodiča a potom (s určitou pravdepodobnosťou) aj časť replikátora 2. rodiča. Vzniknuté replikátory ešte prechádzajú procesom mutácie, pri ktorom s malou pravdepodobnosťou sa zavedú do replikátora chyby (na obrázku sú znázornené malými vertikálnymi čiarami). Takto vytvorené replikátory vytvárajú novú populáciu potomkov, ktorá, keď obsahuje rovnaký počet replikátorov ako rodičovská populácia, sa osamostatní a podobný proces reprodukcie sa opakuje. **(B)** Genetický algoritmus s unárnou reprodukciou, replikátory potomkov sa vytvárajú len jednoduchým kopírovaním rodičovských replikátorov spolu s málo pravdepodobnými mutáciami.



**Obr. 6.** Dva rôzne priebehy evolúcie v genetickom algoritme. Diagram A znázorňuje prípad, v ktorom sledujeme rôzne priebehy evolúcie pre jeden východiskový bod  $S_1$ , v dôsledku stochastického charakteru evolúcie z tohto východiskového stavu môžeme dostať rôzne finálne stavy  $G_1, G_2, G_3$  a  $G_4$ . Predpovedať, ktorý z týchto finálnych stavov dostaneme je skoro nemožné pre enormnú kombinatoriálnu zložitosť problému (obrazne môžeme povedať, že nepoznáme také výpočtové zariadenie, ktoré by bolo schopné v reálnom čase vykonať vyčerpávajúcym spôsobom rekonštrukciu všetkých možných evolučných trajektórií. Diagram B odpovedá opačnej situácii, poznáme konečný stav evolúcie a hľadáme možné evolučné trajektórie s počiatočnými stavmi  $S_1, S_2, S_3$  a  $S_4$ , ktoré by viedli k danému finálnemu stavu  $G_1$ . Podobne, ako v predchádzajúcom inverznom prípade, aj táto rekonštrukcia evolúcie so zadaným finálnym stavom je enormne zložitý kombinatoriálny problém. Určité záchytné body (reprezentované plnými štvorcami na trajektórii vedúcej z počiatočného stavu  $S_4$ ) nám môžu podstatne zjednodušiť rekonštrukciu evolučnej trajektórie; tie trajektórie, ktoré neobsahujú tieto body, sú automaticky vylúčené z našich úvah.

Genetický algoritmus má silne stochastický charakter. V dôsledku existencie náhodných mutácií, kvázináhodného výberu do procesu reprodukcie (a taktiež aj možnosti náhodného vyhynutia replikátorov populácie s vysokým fitness pred ich vstupom do reprodukčného procesu), evolučné algoritmy nemajú deterministický charakter, t.j. nemôžeme presne predvídať evolúciu populácie, pozri obr. 6.

Na druhej strane, jednoduchý organizmu, napr. baktéria je adaptívna zložitosť, pretože vykazuje tak znaky zložitosti, ako aj adaptivity na zmenu prostredia. Existencia adaptívnej zložitosti je *hlavným kritériom existencie života*. Ak na nejakej planéte nájdeme nejaké hmotné štruktúry, ktoré môžeme klasifikovať ako adaptívnu zložitosť, potom môžeme deklarovat' s vysokým stupňom istoty, že na planéte existuje (alebo existoval) život. Cieľom každej evolučnej teórie je vysvetliť vznik adaptívnej zložitosti, t.j. experimentálny fakt, ktorý niektorí pokladajú za dôkaz existencie Tvorcu.



**Obr. 7.** Dva ilustračné príklady zložitostí rôzneho typu. Diagram A znázorňuje náhodne rozmiestnené body na mriežke, diagram B znázorňuje body umiestené do tvaru trojkríža.

Budeme študovať jednoduchý ilustračný príklad zložitosti pomocou 15 objektov, ktoré sú rozmiestnené na mriežke  $9 \times 9$ , pozri obr. 7. Diagram A nám znázorňuje typické náhodné rozmiestnenie objektov na mriežke, môžeme konštatovať, že pravdepodobnosť vzniku podobného "chaotického" rozmiestnenia je skoro jednotková,  $p_A = 1 - \varepsilon$ , kde  $\varepsilon$  je malé kladné číslo. Potom pravdepodobnosť vzniku rozmiestnenia B (alebo podobného posunom alebo rotáciou) bude  $p_B = \varepsilon$ . Informačný obsah nejakého obrazca, ktorý vzniká s pravdepodobnosťou  $p$  je definovaný vzťahom  $I = -\ln_2 p$ . Potom informačný obsah našich obrazcov A a B má tieto hodnoty

$$I_A = -\ln_2(1 - \varepsilon), \quad I_B = -\ln_2 \varepsilon \quad (2a)$$

Pre jednoduchosť odhadneme len asymptotické hodnoty týchto informačných obsahov pre malé kladné  $\varepsilon$ , dostaneme

$$\lim_{\varepsilon \rightarrow 0_+} I_A = 0, \quad \lim_{\varepsilon \rightarrow 0_+} I_B \rightarrow \infty \quad (2b)$$

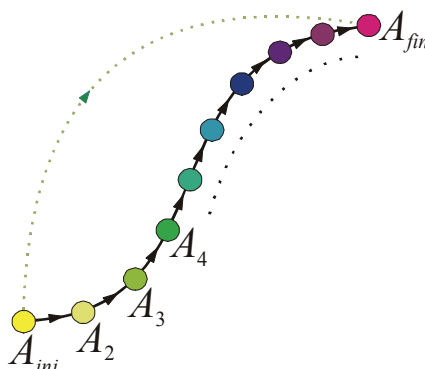
To znamená, že v súhlase s našim očakávaním, obrazec A má nulový informačný obsah, zatiaľ čo obrazec B má skoro nekonečný informačný obsah. K lepšiemu pochopeniu tohto výsledku predstavme si nasledujúcu situáciu: Prechádzame sa po morskom pobreží na neznámom ostrove a nájdeme obrazec A, na ktorom sú objekty - kamene náhodne rozmiestnené. Na základe našich predchádzajúcich úvah vieme, že tento obrazec má nulový informačný obsah, nedáva nám žiadnu správu o svojom tvorcovi. Naopak, ak by sme našli obrazec B, na ktorom kamene sú umiestnené pravidelne do tvaru troj-kríža, získali sme správu o veľkom informačnom obsahu, vieme skoro so 100% istotou, že na ostrove musí existovať tvorca obrazca.

Tento príbeh je parafrázou argumentácie anglikánskeho teológa Williama Paleyho z prelomu 18. a 19. storočia [44], ktorý vo svojej knihe „*Natural Theology*“ dôvodil, že každá adaptívna zložitost' musí mať svojho tvorca (každé hodiny majú svojho hodinára – tvorca). Tento argument sa neustále opakuje v rôznych verziách ako argument proti evolučnej teórii už skoro dvesto rokov. V súčasnosti je zmodernizovaný do podoby tzv. ireducibilnej zložitosti [4], ktorej príkladom môžu byť rôzne zložité orgány vnímania fungujúce len vtedy, ak obsahujú všetky svoje časti v správnom nastavení. Ak chýba čo len jedna súčasť tejto ireducibilnej zložitosti, stáva sa totálne nefunkčnou<sup>3</sup>. Keď rozmýšľame zjednodušene evolucionisticky, „nedokončený“ orgán vnímania nepredstavuje pre organizmus žiadnu evolučnú výhodu, čiže nemôže existovať ani selekčný tlak na jeho vznik.

Nedávno sa diskutoval v evolučnej psychológii a v kognitívnej vede podobný problém evolučného pôvodu „univerzálnej gramatiky“, pôvodne postulovanej americkým lingvistom Noamom Chomskym [6] v 60. rokoch minulého storočia, ako nutnej štruktúry našej mysle (mozgu) k tomu, aby bol schopný vysvetliť akvizíciu jazyka malým deťom počúvaním jednoduchých gramatických viet, bez nutnosti poznania gramatických pravidiel danej reči. Evolúcia nevytvorila schopnosť ľudskej reči úplne od počiatku v priebehu ostatných niekoľkých miliónov rokov, ale použila materiál, ktorý už bol k dispozícii, ktorý bol podstatne evolučne starší, ako práve vznikajúci jazyk. Evolúcia využíva už existujúce štruktúry pre nové, často prekvapujúce ciele. Anatomická štruktúra mozgu humanoidných opíc je veľmi podobná anatomickej štruktúre ľudskeho mozgu, medzi opicami a ľuďmi môžu byť pomerne presne identifikované mnohé vzájomne si odpovedajúce oblasti mozgu. Oblasť ľudskeho mozgu, ktoré sú označené za centrum reči, u opíc sú identifikované ako oblasti pre kontrolu svalov tváre a pre analýzu zvukového vstupu. Táto skutočnosť asi podstatne uľahčila evolučné „predrôtovanie“ týchto oblastí pre účely vzniku ľudskej reči. Z týchto úvah vyplýva, že ľudský jazykový inštinkt nie je dielom „zázračnej“ mutácie, alebo povedané poetickým jazykom Dawkinsa, *náhla inšpirácia slepého hodinára riadiaceho evolúciu* [9], ale dôsledkom niekoľko sto milión rokov, v ktorých evolúcia experimentovala so živočíšnou kogníciou. Adaptívny význam ľudskeho jazyka je nespochybniteľný – umožňuje hovoriť. Spolupráca pri lovení, plánovanie, koordinovanie aktivít, podieľanie sa na spoločných úlohách, sociálne

<sup>3</sup> Klasickým „úsmevným“ príkladom neredukovateľnej zložitosti je pasca na myši, ktorá obsahuje 5 súčastí: drevenú doštičku, pružinu, drôtový rám, drôtovú závlačku a plechovú poistku. Ak chýba z týchto súčastí čo len jedna časť, pasca sa stáva nefunkčnou.

väzby, manipulácia alebo podvádzanie, toto všetko môže mať úžitok zo zvyšovania schopnosti komunikovať ľudským jazykom.



**Obr. 8.** Znázornenie evolučnej tvorby komplexného orgánu (napr. oka stavovcov), proces je inicializovaný evolučným vznikom jednoduchého „protoorgánu“  $A_{ini}$ , cez prechodné riešenia  $A_i$ , až po konečné riešenie  $A_{fin}$ . Zložitosť orgánu postupne rastie, dva evolučne susedné orgány sú si podobné, evolučný prechod  $A_i \rightarrow A_{i+1}$  je dobre interpretovateľný pomocou mutácie genotypu organizmu. Každý orgán má špecifikovanú úlohu, ktorá určuje aj selekčný tlak na jeho vznik, tieto úlohy sa môžu v priebehu evolúcie podstatne meniť. Preto môžeme evolučnú úlohu vzniku komplexného orgánu formulovať ako zložitú optimalizačnú úlohu pohyblivého cieľa. Počiatočný a konečný stav je spojený čiarou, ktorá reprezentuje možnosť priameho „evolučného skoku“ z  $A_{ini}$  na  $A_{fin}$ . Táto evolučná zmena by vyžadovala zásadnú prestavbu genotypu, ktorá je jednou mutáciou nerealizovateľná, preto pravdepodobnosť takého evolučného skoku je skoro nulová.

Záver vyplývajúci z obr. 8 môžeme zovšeobecniť tak, že evolučný vznik zložitých živočíšnych orgánov neprebíhal priamočiaro za vzniku hotového orgánu požadovanej zložitosti, ale postupne využíval<sup>4</sup> staršie štruktúry organizmu, ktoré boli prispôsobené na nové potreby. Týmto spôsobom je zabezpečený neustály selekčný tlak vedúci v konečnom dôsledku na vznik komplexného orgánu prostredníctvom medzistupňov, ktorých význam môže byť veľmi vzdialený významu konečného orgánu (pozri obr. 8).

<sup>4</sup> Podľa francúzskeho molekulárneho biológa F. Jacoba [28], darwinovská evolúcia organizmov so zložitou hierarchickou štruktúrou postupuje tak, že prebieha dominantne len na najvyšších hierarchických úrovniach, pričom nižšie hierarchické úrovne sú v podstate evolučne uzavreté – invariantné. Evolúcia môže zasahovať do nižších hierarchických úrovní len veľmi vzácne a aj to len formou tzv. „bricolage“ (fušerstva), t.j. používa štruktúry, ktoré sú na danej úrovni bezprostredne k dispozícii. Napríklad, biochémia bunky bola evolučne uzavretá už pred miliardami rokov, po vzniku mnohobunkových organizmov evolúcia len veľmi vzácne zasahuje na tejto nízkej hierarchickej úrovni (preto môže existovať univerzálne platný vedný odbor biochémie). V prípade fatálnej potreby, evolúcia sa pokúša aj na tejto úrovni pomocou „bricolage“ riešiť vzniklý problém. Takouto „bricolage“ môže byť aj evolučný vznik univerzálnej gramatiky, keď evolúcia použila „predpripravené“ štruktúry v mozgu, ktoré pôvodne vykonávali iné funkcie. V súčasnej evolučnej teórii je „bricolage“ pokladaná za jeden z hlavných mechanizmov vzniku zložitých orgánov zo starších štruktúr, ktoré mohli pôvodne hrať úlohu celkom odlišnú od výsledku „bricolage“. Niektorí evoluční biológovia idú tak ďaleko, že evidentnú existenciu „bricolage“ vo fenotype druhu pokladajú za jeden z hlavných dôkazov evolučného vzniku druhov.

Niekoľko poznámok k problémom reprodukcie a dedičnosti, ktorých významnou časťou je tzv. embryogenéza, čo je spôsob vzniku a vývoja nových jedincov zo zárodkov. Budeme rozlišovať dva druhy embryogenézy:

(1) *Epigenetická embryogenéza*, kde nový organizmus vzniká delením jednej "zárodočnej" bunky, tento druh embryogenézy sa vyskytuje na našej planéte pri reprodukcii biologických systémov. Môže byť zjednodušene chápaná ako *recept* na prípravu koláča, kde je uvedené nielen to, čo koláč obsahuje, ale aj jeho príprava.

(2) *Preformačná embryogenéza*, kde nový organizmus je reprezentovaný zárodkom, ktorý obsahuje všetky detaily dospelého organizmu. Tento typ embryogenézy môžeme formálne prirovnať k plánu budovy alebo stroja, kde sú podrobne uvedené všetky detaily. Existencia tohto druhu embryogenézy je pre nás hypotetická, na našej planéte sa nevyskytuje. Snáď v budúcnosti sa preformačná embryogenéza bude vyskytovať aj na našej planéte v súvislosti s reprodukciovou umelých robotických systémov.

Pre evolúciu je primárne dôležitá tzv. *reverzibilita embryogenézy*, t.j. vzťah medzi genotypom a fenotypom, genotyp  $\Leftrightarrow$  fenotyp. Formálne povedané, či sa genotyp jednoznačne zobrazuje na fenotyp a naopak (t.j. či z fenotypu vieme dekódovať genotyp organizmu). Tento vzťah medzi genotypom a fenotypom je veľmi dôležitý pri diskusii takej evolúcie, ktorá umožňuje prenos získaných vlastností na potomkov (lamarckovská evolúcia). Preformačná embryogenéza umožňuje prenos získaných vlastností do genotypu, pretože vzťah medzi genotypom a fenotypom je jednoznačný. Táto reverzibilita neexistuje pre epigenetickú embryogenézu. Ak použijeme metaforu, podľa ktorej je genotyp niečo ako recept na prípravu koláča, potom z výslednej štruktúry koláča sa nám nepodarí jednoznačne zostrojiť recept, podľa ktorého bol vytvorený. Z tohto dôvodu lamarckovská evolúcia (dedenie získaných vlastností) nemôže existovať na našej planéte, kde živé systémy používajú výhradne len epigenetickú embryogenézu. Niekde vo vesmíre môžu existovať živé systémy, ktoré využívajú preformačnú embryogenézu, potom je možné, že medzi týmito živými systémami prebieha lamarckovská evolúcia založená na prenose získaných vlastností z rodiča na potomkov.

Dawkins poukázal vo svojej eseji [8] na fakt, ktorý sa často ignoruje, že aj keď pripustíme možnosť lamarckovskej evolúcie, ktorá predpokladá genetický prenos získaných vlastností z rodičov na potomkov, neexistencia prírodného výberu<sup>5</sup>

<sup>5</sup> Lamarckova evolučná teória [42] je založená na nasledujúcich dvoch postulátoch: (1) Orgány používaním (nepoužívaním) zvyšujú svoju perfektnosť (zanikajú), (2) získané vlastnosti sa prenášajú geneticky z rodičov na potomkov. Takto formulovaná evolučná teória je nepostačujúca na vznik adaptívnych zložitostí, pretože neobsahuje prírodný výber. Predpoklady o rastúcej perfektnosti orgánov častým používaním a o možnosti genetického prenosu získaných vlastností nie sú dostatočne špecifické k tomu, aby vznikali adaptívne zložitosti. Navyše, ako ukazujú simulačné výpočty realizované pomocou modifikovaného genetického algoritmu, aj keď doplníme Lamarckovu evolúciu o prírodný výber, prenos získaných vlastností z rodičov na potomkov zapríčiňuje vznik „unáhlenej“ konvergenencie evolúcie, konečné riešenia

spôsobuje malú špecifickosť evolúcie adaptívnych zložitostí. Ak aj doplníme Lamarckovu evolúciu o postulát prírodného výberu, táto „zmeska“ darwinovskej a lamarckovskej evolúcie je priveľmi „oportunistická, vždy vyberá len najbližšie optimálne riešenie, čo spôsobuje „zamrznutie“ evolučnej trajektórie v lokálnom minime. Lamarckovská evolúcia s prírodným výberom už môže poskytovať adaptívne zložitosti, ale tieto sú ďaleko od globálne – optimálnych riešení. Budú vznikať „evolučné kreatúry“, ktoré evolučne „zamrzli“ v nejakom suboptimálnom riešení bez šance ďalšieho evolučného vylepšovania.

### 1.3 Dennettov pohľad na evolúciu ako na algoritmus

Dennett v knihe „*Darwin's Dangerous Idea*“ [11] anticipoval Dawkinsove názory na univerzálnosť Darwinovej teórie tak, že zdôraznil jej algoritmický charakter (pozri odsek venovaný genetickému algoritmu), ktorý je nezávislý od materiálnej realizácie. Evolúcia, chápaná ako algoritmus, môže byť aplikovaná na rôznorodé systémy, v ktorých sme schopní identifikovať replikátory a ktorých podstata nemusí byť biologická. Dôvod prečo Dennett zdôrazňoval algoritmický charakter darwinovskej evolúcie, jej univerzálnosť v nezávislosti na materiálnej realizácii replikátorov bol ten, že chcel položiť solídne základy Dawkinsovej memetiky [10], v ktorej bola použitá paradigma Darwinovej evolúcie k vysvetleniu niektorých fenoménov šírenia kultúry. Pretože sa jedná o abiotickú oblasť aplikácie myšlienok Darwinovej evolúcie, považoval Dennett za potrebné najprv sa vyrovnáť s otázkou jej aplikovateľnosti mimo biologické systémy. U nás sa filozofickými aspektmi evolúcie systematicky zaoberá Dušan Gálik [21,22].

## 2 Darwinovský systém

Najvšeobecnejšia formulácia základnej idey univerzálneho darwinizmu je uskutočnená pomocou koncepcie *darwinovského systému* založeného na nasledujúcich dvoch postulátoch (určité náznaky tejto všeobecnej formulácie už boli vykonané pri popise Hollandovho genetického algoritmu):

(1) *Darwinovský systém sa skladá z populácie replikátorov* – jedincov/objektov, ktoré za určitých vhodných podmienok sú schopné *replikácie* - rozmnožovania. Replikačný proces spočíva v „kopírovaní“ jedincov do populácie, pričom toto „kopírovanie“ sa uskutočňuje s *určitými malými chybami*. Z tohto predpokladu nepresnej replikácie (s malými chybami) vyplýva, že *populácia nie je homogénna*, obsahuje jedincov – replikátory, ktoré vykazujú malú variabilitu.

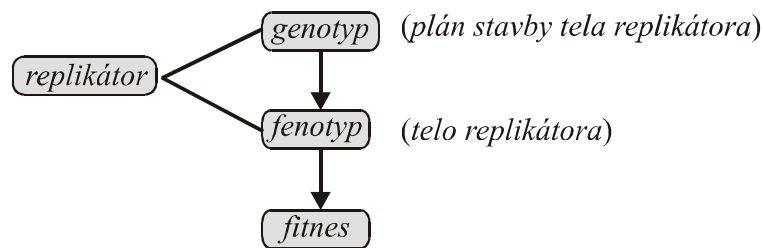
(2) Každý replikátor populácie je ohodnotený funkciou *fitness*, ktorá vyjadruje schopnosť replikátora prežiť a úspešne vstupovať do replikačného procesu. *Prírodný*

---

sú obvykle pomerne vzdialené od globálnych riešení, ktoré sú (ako postuloval už Wright) cieľom darwinovskej evolúcie. Obrazne povedané, lamarckovská evolúcia je veľmi oportunistická, využíva vždy najbližšie optimálne riešenie, čo môže podstatne negatívne ovplyvniť priebeh evolučnej trajektórie, jej „zamrznutie“ v nejakom lokálnom riešení.

výber v darwinovskom systéme spočíva v tom, že jedinci populácie nie sú vyberaní do replikačného procesu náhodne, ale *spravdepodobnosťou úmernou ich fitness* (hovoríme, že výber sa deje kvázináhodne).

V definícii darwinovského systému sa vyskytuje niekoľko pojmov, ktoré musia byť bližšie objasnené. Pojem „replikátor“ je potrebné chápať ako relatívne samostatnú entitu, ktorá existuje v nejakom prostredí a vytvára spolu s ostatnými replikátormi populáciu. Najjednoduchší prístup k implementácii replikátora je jeho stotožnenie s nejakou informáciou, ktorá kóduje jeho „telo“. Z tohto dôvodu môžeme využívať biologickú terminológiu a rozlišovať dve rôzne špecifikácie replikátora, jeho *genotyp* a *fenotyp*. Genotyp je informácia o architektúre replikátora, zatiaľ čo fenotyp je „telo“ replikátora<sup>6</sup>. Vzťah medzi genotypom a fenotypom nebudeme bližšie špecifikovať, obvykle je silne závislý na type darwinovského systému. Čo bude pre nás dôležité, je skutočnosť, že fitness replikátora je určený schopnosťou fenotypu prežívať v danom prostredí a vstupovať do výhodných interakcií s inými replikátormi z populácie, pozri obr. 9.



**Obr. 9.** Replikátor je charakterizovaný tak genotypom, ako aj fenotypom. Fitness replikátora je určený schopnosťou fenotypu prežívať v danom prostredí a vstupovať do výhodných interakcií s inými replikátormi z populácie.

Proces replikácie jedinca – replikátora je formálne chápaný ako kopírovanie jeho genotypu a vytvorenie nového fenotypu určeného kopírovaným genotypom. To znamená, že fenotyp – organizmus replikátora – môžeme chápať ako nosič (vehikel) genotypu, ktorý umožňuje jeho replikáciu. Pre zjednodušenie našich úvah chápeme proces replikácie len ako kopírovanie genotypu, pričom tento proces kopírovania je „fyzicky“ uskutočnený fenotypom replikátora. Je potrebné poznamenať, že sa jedná o veľmi silnú idealizáciu, ale umožní nám zaviesť pomerne jednoduchú algoritmizáciu univerzálneho darwinizmu (pozri algoritmus 1).

Musíme však podotknúť, že môžu existovať darwinovské systémy, kde odlišenie fenotypu od genotypu neplatí, kde sa genotyp kopíruje – replikuje iným zariadením, ako vlastným fenotypom. Dobrým príkladom tejto situácie sú biologické a počítačové vírusy, ktoré k vlastnej replikácii využívajú systémy, v ktorých parazitujú.

<sup>6</sup> Podobne, ako vo fyzike mikrosвета, kde častice majú duálny časticovo-vlnový charakter, môžeme aj fenotyp a genotyp chápať ako duálne charakteristiky replikátora. V určitých situáciách pod replikátorom rozumieme jeho genotyp, zatiaľ čo v iných situáciách pod replikátorom myslíme jeho fenotyp.



Postulujeme, že replikátor je reprezentovaný svojím *genotypom*  $x$ , ktorý, ako už bolo poznamenané, obsahuje zakódovanú informáciu o konštrukcii replikátora (pripomeňme si Dawkinsove prirovnanie genotypu k receptu na koláč). *Populácia* replikátorov je množina genotypov

$$P = \{x_1, x_2, \dots, x_p\}$$

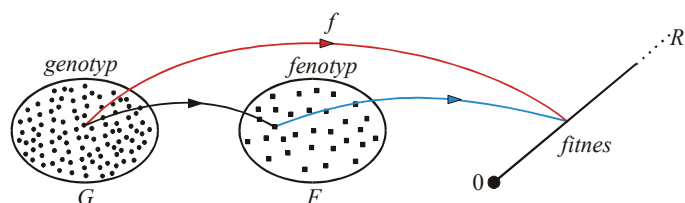
Vzájomný vzťah medzi triádou „genotyp – fenotyp – fitness“ je reprezentovaný postupnosťou dvoch zobrazení (pozri obr. 10)

$$G \xrightarrow{\text{fenotyp}} F \xrightarrow{\text{fitness}} [0, \infty)$$

kde  $G$  je množina všetkých genotypov a  $F$  je množina všetkých fenotypov. To znamená, že prvotnou veličinou je genotyp, ktorý je zobrazovaný na fenotyp, potom fenotyp je zobrazovaný na fitness (reprezentovaný kladným reálnym číslom). Prvé zobrazenie priradí každému genotypu telo – organizmus, nazývaný fenotyp. V biológii toto zobrazenie predpokladá existenciu procesu nazývaného embryogenéza, t. j. tvorba organizmu zo zárodka – genotypu. Skrátaná forma dvojstupňového zobrazovania má tvar (pozri obr. 10)

$$G \xrightarrow{f} [0, \infty)$$

kde nové zobrazenie  $f$  vznikne zložením dvoch zobrazení, zobrazovania genotypu na fenotyp a zobrazovania fenotypu na fitness.



**Obr. 10.** Znáročenie jednoduchých zobrazení genotypu na fenotyp a fenotypu na fitness, a zloženého zobrazovania genotypu priamo na fitness.

V tejto súvislosti si môžeme položiť otázku, prečo je užitočné explicitne uvažovať fenotyp ako sprostredkovateľa medzi genotypom a fitness. Táto otázka je plne legitímna z pohľadu matematiky, avšak je potrebné poznamenať, že koncepcia fenotypu reprezentuje veľmi efektívnu a plodnú heuristiku<sup>7</sup> pre interpretáciu univerzálneho darwinizmu. Cieľom evolúcie nie je obvykle požadovaný genotyp, ale fenotyp vykazujúci určité vlastnosti. Z tohto pohľadu môžeme povedať, že darwinovská evolúcia je reprezentovaná postupnosťou fenotypov, ktorých vlastnosti sú bližšie a bližšie k vlastnostiam cieľového hypotetického fenotypu.

<sup>7</sup> Heuristika v informatike znamená pravidlo, väčšinou intuitívne zavedené, ktoré poskytuje určitý návod, ako sa správať v zložitom prostredí, ktorého presný model nepoznáme. Tak napríklad, v neznámom lese je veľmi užitočnou heuristikou pravidlo, že sa používajú len označené chodníky. V mnohých prípadoch tieto označené chodníky nereprezentujú optimálnu cestu, ale zabezpečia nám, že sa dostaneme do cieľa.

Fitnes replikátora  $x \in P$  je určený dvoma rôznymi prístupmi. Prvý spôsob chápe fitnes ako atribút replikátora (vloženého do nejakého prostredia  $E$ ), interakcie daného replikátora s populáciou sa ignorujú, potom

$$fitnes(x) = F(x; E)$$

kde prostredie  $E$  je chápané ako parameter funkcie, ktorá replikátor  $x$  ohodnotí fitnes, táto funkcia je určená len pomocou jeho genotypu (a prípadného vplyvu prostredia). Druhý spôsob určenia fitnes je založený na predstave, že táto veličina je dominantne určená len interakciou daného replikátora  $x$  s ostatnými replikátormi v populácii (plus vplyv prostredia  $E$ )

$$fitnes(x) = \sum_{\substack{x' \in P \\ x' \neq x}} F(x, x'; E)$$

Tento druhý prístup postuluje, že dominantná časť fitnes je určená práve interakciou replikátorov v populácii, kde nie je dôležitá architektúra toho-ktorého replikátora (môžeme povedať, že táto prispieva nemennou časťou k fitnes). Takto je určené fitnes replikátorov v aplikáciách evolučnej teórie hier [2] (napr. pri štúdiu evolučného vzniku kooperácie v populácii). V týchto aplikáciách zahájených Maynard Smithom [39] a Axelrodom [2] už skoro pre 30 rokmi, fitnes replikátorov je určený schopnosťou spolupracovať alebo nespôlpracovať s inými replikátormi z populácie, a nie ich vnútornou architektúrou. V tomto prípade funkcia fitnes nie je fixne zadaná, ale závisí od zloženia celej populácie. Z tohto pohľadu potom môžeme tento druh evolučnej optimalizácie interpretovať ako tzv. „problém pohyblivého cieľa“, kde povrch fitnes funkcie sa dynamicky mení s evolúciou populácie. Problém emergencie sociálnych štruktúr v multiagentových systémoch bude študovaný v sekcii 4.

## 2.1 Koevolúcia

Teória darwinovskej evolúcie obvykle implicitne predpokladá existenciu jednej populácie, ktorá je predmetom evolúcie uskutočňovanej mechanizmami popísanými vyššie. Existujú však prípady, keď dva rôzne druhy replikátorov medzi sebou silne interagujú (napr. dve populácie rôznych druhov, ktoré sú v symbiotickom vzťahu, jeden druh produkuje látku, ktorá je vitálne dôležitá pre prežitie druhého druhu). V abiotických aplikáciách sú koevolučné modely dôležité pre štúdium rôznych kultúrnych fenoménov.

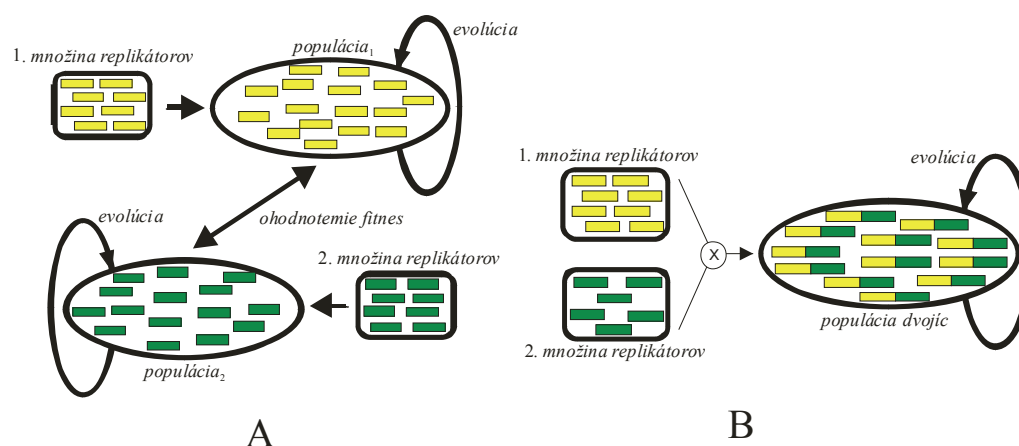
Budeme študovať dva rôzne modely koevolúcie, pozri obr. 11.

Prvý model (diagram A) vyjadruje tzv. *kompetitívny koevolučný model*, kde existujú dve kvázinezavislé populácie, ktoré sú zložené z replikátorov rôznych druhov. Interakcia medzi týmito rôznymi populáciami spočíva vo výpočte fitnes replikátorov z jednotlivých populácií, fitnes replikátora z prvej populácie je určený nielen jeho zložením (alebo jeho interakciou s ostatnými replikátormi z danej populácie), ale aj jeho interakciou s replikátormi z druhej populácie, napr.

$$fitness(\mathbf{x}) = F_1(\mathbf{x}) + \sum_{\mathbf{y} \in P_2} F_{12}(\mathbf{x}, \mathbf{y}) \quad (\forall \mathbf{x} \in P_1)$$

$$fitness(\mathbf{y}) = F_2(\mathbf{y}) + \sum_{\mathbf{x} \in P_1} F_{21}(\mathbf{y}, \mathbf{x}) \quad (\forall \mathbf{y} \in P_2)$$

Ak obe populácie chápeme ako jeden (ko)evolučný systém, potom evolučná optimalizácia môže byť taktiež interpretovaná ako „problém pohyblivého cieľa“, pretože funkcia fitness sa dynamicky mení s evolúciou populácií.



**Obr. 11.** Dva rôzne koevolučné modely. Diagram A vyjadruje kompetitívny model koevolúcie obsahujúci dve kvázinezávislé populácie, tieto populácie vzájomne interagujú prostredníctvom výpočtu fitness. Ak počítame fitness replikátorov z jednej populácie, berieme do úvahy aj interakciu s replikátormi z druhej populácie. Diagram B vyjadruje kooperatívny model koevolúcie, v tomto prípade populácia je zložená z dvojíc odlišných replikátorov. Fitness týchto dvojíc je určený jednak ich zložkami a taktiež aj ich vzájomnou interakciou.

Druhý model (obr. 11, diagram B) sa nazýva *kooperatívny koevolučný model*, v ktorom existuje len jedna populácia, ale táto populácia je zložená z komplexov (usporiadaných dvojíc) replikátorov rôzneho druhu. Tieto komplexy sú chápané ako nezávislé objekty evolúcie, ich fitness je určený napr. takto

$$fitness(\mathbf{x}, \mathbf{y}) = F_1(\mathbf{x}) + F_2(\mathbf{y}) + G(\mathbf{x}, \mathbf{y})$$

kde funkcie  $F_1$  a  $F_2$  určujú fitness jednotlivých zložiek komplexu, funkcia  $G$  špecifikuje príspevok k fitness na základe interakcie medzi zložkami komplexu. Tento model sa používa vtedy, keď jednotlivé komponenty neexistujú nezávislé, sú na sebe silne závislé. Ako pekný ilustračný príklad pre kooperatívnu koevolúciu môže slúžiť „kultúrna“ aplikácia [36] zaoberajúca sa koevolúciou génov a mémov.

### 3 Eigenova a Schustrova teória molekulárneho darwinizmu

Eigenova a Schustrova teória molekulárneho darwinizmu [13,14] postihuje elementárne aspekty evolúcie na úrovni biomakromolekúl. Tento model evolúcie poskytuje semikvantitatívnu teóriu, ktorá je schopná interpretovať mnohé pojmy a fenomény evolúcie, ako napr. úlohu neutrálnych mutácií v priebehu adaptačného procesu. Dôležitú úlohu v našom prístupe hrajú metódy *umelej chémie* [3,12,16-20,23,32-36], ktorá je v súčasnosti považovaná za oblasť úzko súvisiacu s umelým životom. Tento prístup je založený na metafore chemostatu (chemického reaktora), ktorý obsahuje "molekuly" reprezentované abstraktnými objektmi (reťazcami znakov, grafmi, formulami z nejakého jazyka, atď.). Molekuly sú v chemostate transformované pomocou "chemických reakcií" na iné prípustné objekty, pričom pravdepodobnosť týchto transformácií je striktno určená štruktúrou reagujúcich objektov. Kinetika procesov prebiehajúcich v chemostate je popísaná Eigenovými replikátorovými diferenciálnymi rovnicami, ktoré majú svoj „zdravý“ základ vo fyzikálnej chémii (zákon účinných hmotností [1]). Hlavným predmetom záujmu umelej chémie je (i) štúdium formálnych systémov založených na metafore chemostatu, ktoré sú schopné vykonávať špeciálne vysoko paralelné výpočty a (ii) návrh formálne autokatalytických systémov (molekuly sú reprezentované štruktúrovanými objektmi) pre účely *in-silico* simulácií emergencie "živých" systémov.

V súčasnosti je molekulárna evolúcia teoreticky študovaná pomocou "fyzikálneho" modelu RNA a jej foldingu [16-20,45-46]. Pre úplnosť je potrebné poznamenať, že existujú aj iné modely molekulárnej evolúcie [43], menovite pripomenieme tie, ktoré sú založené na stochastickej Kauffmanovej *KN* funkcii [27,28], ktoré sú veľmi všeobecné a poskytujú podobné výsledky ako metódy založené na RNA foldingu. Ukážeme, že použitie tohto jednoduchého modelu povrchu fitness pomocou RNA foldingu poskytuje výsledky, ktoré sú formálne veľmi blízke tým, ktoré sú predpovedané replikátorovými diferenciálnymi rovnicami, ktoré boli postulované už pred viac ako 30 rokmi Eigenom [13] ako odozva na Spiegelmanove *in-vitro* experimenty [47] s evolúciou RNA molekuly<sup>8</sup>.

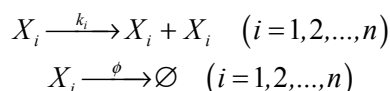
#### 3.1 Eigenove replikátory

Na prelome 60. a 70. rokov publikoval Manfred Eigen významnú prácu "*Self organization of matter and the evolution of biological macro molecules*" [13], v ktorej študoval vlastnosti chemického systému zloženého z tzv. *replikátorov* (priamo v texte tejto práce uvádza, že bol stimulovaný k jej napísaniu Spiegelmanovými [47] experimentmi *in-vitro* darwinovskej evolúcie). Replikátory sú hypotetické biomakromolekuly, ktoré majú schopnosť replikovať sa. Eigen ukázal, že takýto jednoduchý chemicko-kinetický systém formálne simuluje darwinovskú evolúciu na molekulárnej úrovni. Ďalšie zovšeobecnenie modelu replikátorov bolo vykonané Eigenom a Schusterom [14] v sérii prác so spoločným názvom "*The Hypercycles: A*

<sup>8</sup> Pozri prednášku E. Tomáška a J. Noseka [52] na seminári Kognitívne vedy III (CogSci2000), CHTF STU, Bratislava. Text prednášky je dostupný na [http://math.chtf.stuba.sk/CogSci\\_2000.htm](http://math.chtf.stuba.sk/CogSci_2000.htm).

"Principle of Natural Evolution", v ktorých sa položili základy teórie *hypercyklov*, ktoré sú v súčasnosti považované za jeden z nemnohých formálnych systémov schopných simulovať nárast zložitosti (komplexity) na rozhraní živých a neživých systémov.

Študujme molekuly - replikátory  $X_1, X_2, \dots, X_n$ , ktoré sú schopné nasledujúcich dvoch chemických reakcií



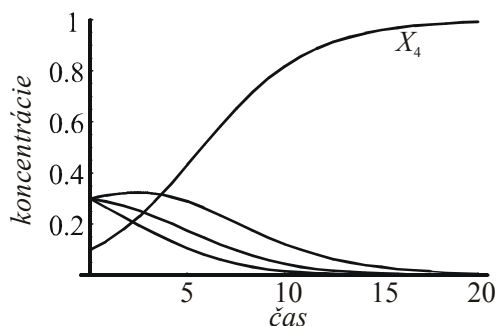
Prvá reakcia znamená, že molekula  $X_i$  sa replikuje s rýchlostnou konštantou  $k_i$  a druhá reakcia znamená, že molekula  $X_i$  zaniká s rýchlostnou konštantou  $\phi$  (ktorá sa nazýva *zried'ovací tok* a bude špecifikovaná neskoršie). Ak použijeme k týmto reakciám štandardný fyzikálno-chemický zákon účinných hmotností [1], dynamika kinetického systému obsahujúceho obe reakcie je určená diferenciálnymi rovnicami

$$\dot{x}_i = x_i (k_i - \phi) \quad (i=1,2,\dots,n)$$

kde  $0 \leq x_i \leq 1$  je koncentrácia molekuly  $X_i$ . Zried'ovací tok  $\phi$  je chápaný ako "voľný parameter", ktorý sa nastaví tak, aby suma časových derivácií koncentrácií bola nulová,  $\sum \dot{x}_i = 0$

$$\dot{x}_i = x_i \left( k_i - \sum_{j=1}^n k_j x_j \right) \quad (i=1,2,\dots,n)$$

kde podmienka  $\sum x_i = 1$  bola použitá bez straty všeobecnosti našich úvah.

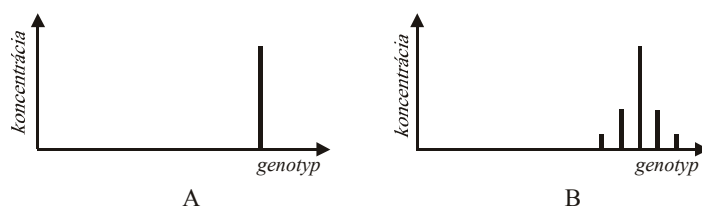


**Obr. 12.** Časový priebeh koncentrácií 4-zložkového systému s rýchlostnými konštantami  $k_1=1$ ,  $k_2=2$ ,  $k_3=3$  a  $k_4=4$ . Molekuly  $X_4$  víťazia, aj keď ich počiatočná koncentrácia bola najmenšia, v čase  $t > 15$  systém už obsahuje skoro výlučne len molekuly  $X_4$ , ostatné vymizli.

Riešenie tohto systému diferenciálnych rovníc vyhovuje asymptotickej podmienke, kde "preživa" len tá molekula, ktorá má maximálnu rýchlostnú konštantu  $k_{max}$ , zatiaľ čo ostatné molekuly vymiznú

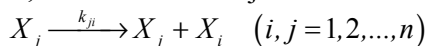
$$\lim_{t \rightarrow \infty} x_i(t) = \begin{cases} 1 & (\text{pre } k_i = k_{max} = \max\{k_1, \dots, k_n\}) \\ 0 & (\text{ostatné prípady}) \end{cases}$$

Každý typ molekuly je formálne chápaný ako zvláštny biologický druh s fitness špecifikovanou rýchlostnou konštantou  $k$ , potom prežívajú len tie molekuly - biologické druhy, ktoré sú najlepšie prispôsobené, t.j. majú najväčšiu rýchlostnú konštantu  $k_{max}$ ; všetky ostatné molekuly - druhy s rýchlostnými konštantami - fitness menšími než ako  $k_{max}$  vymiznú v priebehu časovej evolúcie chemicko-kinetického systému, pozri obr. 12. Podmienka konštantnosti sumy koncentrácie (t.j.  $\sum x_i = 1$ ) zavádza do systému dodatočný "selekčný tlak", replikujúce sa molekuly eliminujú ostatné molekuly.



**Obr. 13.** Diagram A znázorňuje asymptotickú situáciu  $t \rightarrow \infty$  pre systém replikátorov bez mutácií, ktoré sa presne replikujú. V tomto prípade asymptoticky stabilný stav systému obsahuje len replikátory s maximálnou rýchlostnou konštantou  $k_{max}$ . Diagram B znázorňuje asymptoticky stabilný stav systému, kde replikácia nie je presný proces, kde sa už uvažujú mutácie pri replikačnom procese. Stabilný stav je zložený z niekoľkých replikátorov, pričom najväčšia koncentrácia prislúcha replikátoru s maximálnou rýchlostnou konštantou  $k_{max}$ ; ostatné replikátory sa vyskytujú s podstatne menšou koncentráciou. Replikátory vyskytujúce sa súčasne s hlavným víťazom  $X_{max}$  sú tie, ktoré sú mu blízke a sú z neho tvorené mutáciami.

Pôvodná forma Eigenovho modelu replikátorov neobsahovala mutácie, replikácia molekúl bol presný proces, pri replikácii molekuly  $X_i$  vznikala len táto molekula a žiadna iná sa týmto replikačným procesom nevytvárala. Eigen jednoducho modifikoval svoj model tak, že obsahoval už aj mutácie



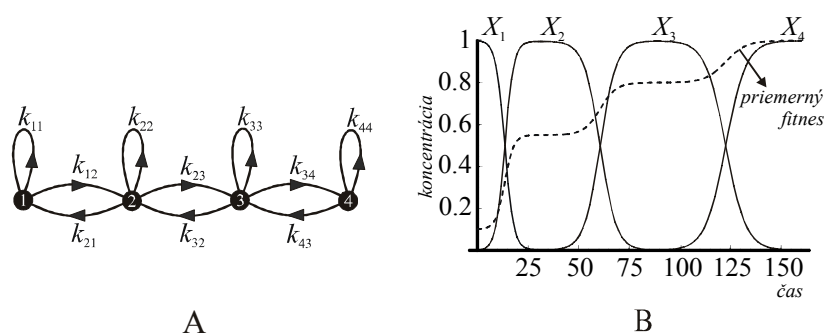
kde  $k_{ji}$  je rýchlostná konštantka tejto reakcie, pričom sa predpokladá, že matica rýchlostných konštant  $\mathbf{K} = (k_{ij})$  má dominantné diagonálne elementy, t.j. nediagonálne elementy sú podstatne (rádovo) menšie ako diagonálne elementy ( $k_{ij} \ll k_{ii}$ , pre  $i, j = 1, 2, \dots, n$  a  $i \neq j$ ). Tento dôležitý predpoklad vyplýva priamo zo skutočnosti, že "nepresné" replikácie pre  $i \neq j$  sú veľmi vzácne, t.j. produkt  $X_j$  je uvažovaný ako mutácia eduktu  $X_i$ ,  $X_j = O_{mut}(X_i)$ . Systém diferenciálnych rovníc popisujúci dynamické vlastnosti takto modifikovaného systému replikátorov má tvar

$$\dot{x}_i = x_i(k_{ii} - \phi) + \sum_{j \neq i} k_{ji} x_j \quad (i = 1, 2, \dots, n)$$

Zried'ovací tok  $\phi$  je určený tak, aby aj v tomto všeobecnejšom prípade suma časových derivácií koncentrácií bola nulová ( $\sum \dot{x}_i = 0$ , alebo  $\sum x_i = 1$ )

$$\phi = \sum_{i,j=1}^n k_{ij} x_j$$

Zavedenie slabých mutácií nemení dramaticky všeobecné vlastnosti pôvodného jednoduchého systému replikátorov bez mutácií. Menovite, v asymptotickom čase  $t \rightarrow \infty$  systém obsahuje skoro výlučne len molekuly majúce najväčšie rýchlostné konštanty  $k_{max}$ , tieto molekuly sú sprevádzané aj inými replikátormi tvorenými z víťazného replikátora (s rýchlostnou konštantou  $k_{max}$ ) pomocou nepresného replikačného procesu, pozri obr. 13.



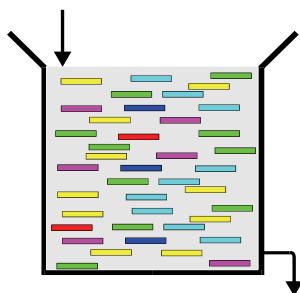
**Obr. 14.** Diagram (A) reprezentuje 4-replikátorový systém, kde replikátor  $X_i$  vytvára pomocou replikácie s chybami susedné replikátory  $X_{i-1}$  a  $X_{i+1}$ . Orientované hrany diagramu sú ohodnotené rýchlostnými konštantami, pričom diagonálne rýchlostné konštanty sú medzi sebou dobre separované a o mnoho väčšie ako nediagonálne rýchlostné konštanty. Diagram (B) znázorňuje koncentračné profily replikátorov, ktoré tvoria postupnosť koncentračných vln. Diagram taktiež obsahuje priebeh priemerného fitness, ktorý je špecifikovaný  $\bar{k} = k_{11}x_1 + \dots + k_{44}x_4$ , tvorí neklesajúcu skokovú funkciu, ktorá je typická pre Darwinovu evolúciu.

### 3.2 Replikátory a molekulová Darwinova evolúcia

Eigenov systém replikátorov s mutáciou (t.j. replikácie s chybami) je vhodným prostriedkom na simuláciu *molekulárnej Darwinovej evolúcie*. Študujeme hypoteticky reakčný systém obsahujúci postupnosť  $n$  replikátorov  $X_1, X_2, X_3, \dots, X_n$ . Tieto replikátory vyhovujú podmienkam, že  $X_i$  môže produkovať pomocou chybných replikácií susedné replikátory  $X_{i\pm 1}$ , pozri obr. 14, diagram A. Ak chemostat v čase  $t=0$  obsahuje len replikátor  $X_1$ ,  $x_1(0)=1$ , potom v priebehu evolúcie existujú koncentračné vlny, ktoré sú postupne priradené replikátorom  $X_2, X_3, \dots, X_n$ , pozri obr. 14, diagram B. Táto skutočnosť je interpretovaná ako prejav molekulovej Darwinovej evolúcie, kde fitness jednotlivých „druhov“ sú špecifikované diagonálnymi rýchlostnými konštantami  $k_{ii}$ . Evolučný proces bol naštartovaný čistou populáciou obsahujúcou len  $X_1$ . Pretože jeho replikácia nie je perfektná, príležitostne môže vzniknúť nasledujúci replikátor  $X_2$  (špecifikovaný rýchlostnou konštantou  $k_{12}$ ), ktorý má väčší fitness ako jeho predchodca  $X_1$  ( $k_{11} < k_{22}$ ). To znamená, že v tejto etape evolúcie "druh"  $X_2$  zvíťazí (t.j. jeho koncentrácia sa postupne zvýši skoro na jednotkovú hodnotu, zatiaľ čo koncentrácia

predchodcu sa asymptoticky blíži k nulovej hodnote. Tento proces sa opakuje pre replikátor  $X_2$ , ktorý je teraz chápaný ako počiatočný replikátor (hrá rovnakú úlohu ako replikátor  $X_1$  v predchádzajúcej etape). Pretože replikácia  $X_2$  nie je taktiež perfektná, s malou pravdepodobnosťou vzniká aj nový „druh“  $X_3$ , ktorý má väčší fitness ako jeho predchodca  $X_2$  ( $k_{33} > k_{22}$ ), t.j. tento nový "druh"  $X_3$  zvíťazí. Tento postupný proces je ukončený, keď sa začne objavovať posledný replikátor  $X_n$  ako dôsledok neperfektnej mutácie predchodcu  $X_{n-1}$ , v záverečnej etape evolúcie sa zvýši koncentrácia tohto finálneho produktu na jednotkovú hodnotu ako dôsledok jeho autoreplikácie.

Sumarizujúc naše výsledky, môžeme konštatovať, že Eigenov systém replikátorov simuluje Darwinovu evolúciu vtedy, keď (1) diagonálne rýchlostné konštanty sú navzájom dobre separované a (2) suma „vychádzajúcich“ rýchlostných konštánt musí byť podstatne menšia ako suma vchádzajúcich rýchlostných konštánt. Táto druhá podmienka znamená, že „pravdepodobnosť“ vzniku daného replikátora zo susedných replikátorov ako dôsledok ich nepresných replikácií je o mnoho väčšia než „pravdepodobnosť“ zániku daného replikátora v dôsledku jeho nepresnej replikácie.



**Obr. 15.** Chemostat obsahuje "zmes" molekúl – replikátorov. Replikačný akt spočíva v tom, že náhodne vybraná molekula sa zreplikuje (buď perfektne alebo s malou mutáciou), vzniknutý produkt – molekula vytesní inú náhodne vybranú molekulu (chemostat obsahuje v celej svojej histórii konštantný počet molekúl).

### 3.3 Metafora chemického reaktoru - chemostat

Uvažujme chemický reaktor - *chemostat*, ktorý obsahuje formálne objekty nazývané „molekuly“, ktoré sú reprezentované reťazcami znakov. Chemostat nie je priestorovo štruktúrovaný, v chémii sa hovorí, že je s dobrým miešaním (well stirred reactor), pozri obr. 15. Akt chemickej reakcie - replikácie spočíva v nasledujúcich troch krokoch:

- (1) náhodný výber replikátora  $M$  (ktorý je totožný s  $X_i$ ) z chemostatu,
- (2) pre takto vybraný replikátor sa uskutoční s pravdepodobnosťou úmernou rýchlostnej konštante  $k_{ii}$  proces replikácie, pričom vznik chybného replikátora  $M'$  (ktorý je totožný s  $X_j$ ) je uskutočnený s pravdepodobnosťou úmernou rýchlostnej konštante  $k_{ij}$ .
- (3) Vzniknutým novým replikátorom  $M'$  (bez mutácie alebo s mutáciou) sa nahradí iný náhodne vybraný replikátor z chemostatu (t.j. vo všetkých



elementárnych kroch máme v chemostate konštantný počet replikátorov).

Počítačová simulácia chemostatu a v ňom prebiehajúce replikačné reakcie sa dá pomerne jednoducho algoritmizovať pomocou stochastického prístupu [23]. Tento prístup je založený na skutočnosti, že ak náhodne vyberieme z chemostatu nejaký replikátor, potom pravdepodobnosť toho, že tento replikátor je totožný s nejakým daným replikátorom  $X$  sa jednoducho rovná koncentrácii tohto replikátora v chemostate,  $0 \leq c(X) \leq 1$ . Pravdepodobnosť uskutočnenia replikácie sa rieši tak, že táto pravdepodobnosť je úmerná replikačnej rýchlostnej konštante. Algoritmicky môžeme chemostat implementovať pomocou pseudo-Pascalu týmto jednoduchým algoritmom:

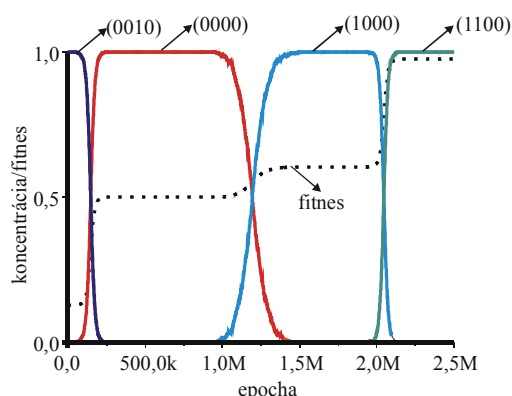
### Algoritmus 2.

```

chemostat:=náhodne vygenerovaná množina replikátorov;
t:=0;
while t<tmax then
begin
  t:=t+1;
  X:=Oselect(chemostat);
  if random<probrep(X) then
  begin
    X':=Omut(X);
    X'':=Oselect(chemostat);
    chemostat:=(chemostat+{X'})-{X''};
  end;
end;
end;
```

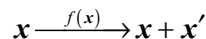
Algoritmus je inicializovaný náhodným vygenerovaním chemostatu. Pravdepodobnosť  $\text{prob}_{\text{rep}}(X)$  je určená tak, že je úmerná rýchlostnej konštante replikácie  $X$ , pravdepodobnosť mutácie je zahrnutá priamo v operátore  $O_{\text{mut}}(X')$ .

Všeobecné idey chemostatu budú ilustrované jeho použitím ako optimalizátora binárnej funkcie  $f : \{0,1\}^n \rightarrow [0,1]$ , ktorá zobrazuje binárne reťazce  $\mathbf{x} = (x_1, x_2, \dots, x_n)$  dĺžky  $n$  na reálne čísla z intervalu  $\langle 0,1 \rangle$ . Naším cieľom bude nájsť optimálne riešenie  $\mathbf{x}_{\text{opt}}$ , ktoré maximalizuje binárnu funkciu  $f$  nad množinou všetkých možných binárnych reťazcov dĺžky  $n$ .



**Obr. 16.** Priebeh frekvencií výskytu dominantných binárnych reťazcov (16) dĺžky 4. Chemostat bol inicializovaný 1000 reťazcami (0010), po  $2.5 \times 10^6$  časových krokoch dominantným konečným riešením je reťazec  $x_{fin}=(1100)$ , kde  $real(x_{opt})=12/15$  a  $f(12/15)=0.9755$ . Toto konečné riešenie je blízke optimálnemu riešeniu  $x_{opt}=(1011)$ , kde  $(x_{opt})=11/15$  a  $f(11/15)=0.9973$ , ale s „veľkou“ Hammingovou vzdialenosťou  $d=3$  (v genetickom algoritme tento efekt sa nazýva „Hammingov útes“). Táto relatívne veľká Hammingova vzdialenosť je hlavný dôvod, prečo replikátorový algoritmus nie je schopný poskytnúť globálne riešenie (1011).

Nech chemostat obsahuje molekuly, ktoré sú realizované binárnymi reťazcami  $\mathbf{x} = (x_1, x_2, \dots, x_n)$ . Uvažujme nasledujúcu monomolekulárnu reakciu – replikáciu



pričom molekula - reaktant  $\mathbf{x}'$  nahradí náhodne vybranú molekulu z chemostatu. Funkčná hodnota  $f(\mathbf{x})$  priradená chemickej reakcii je interpretovaná ako pravdepodobnosť (rýchlostná konštanta) replikácie. Podobne, ako v genetickom algoritme, selekčný tlak v chemostate je vytvorený pomocou fitness (rýchlostné konštanty), replikátory s väčším fitness majú väčšiu šancu zúčastňovať sa reprodukčného procesu (miera kvality replikátorov), a naopak, replikátory s malým fitness len vzácne vstupujú do reprodukčného procesu. Tento jednoduchý prejav princípu Darwinovho prirodzeného výberu zabezpečuje postupnú monotónne neklesajúcu gradáciu fitness v priebehu evolúcie populácie. V tomto prístupe založenom na metafore chemostatu, princíp prírodného výberu na základe fitness je rozšírený o dodatočný selekčný tlak založený na skutočnosti, že populácia v chemostate má konštantný počet jedincov. Molekula vstupujúca do reakcie je náhodne vybraná z chemostatu, kvázináhodne je rozhodnuté, či vstupuje do reakcie alebo nie (s pravdepodobnosťou úmernou jej funkčnej hodnote), vzniknutá molekula vytesní z chemostatu inú, náhodne vybranú molekulu. Produkt  $\mathbf{x}'$  je mutácia vstupujúcej molekuly  $\mathbf{x}$ ,  $\mathbf{x}' = O_{mut}(\mathbf{x})$ , kde  $O_{mut}$  je stochastický operátor mutácie, ktorý zmení jednotlivý bit s pravdepodobnosťou  $P_{mut}$ . Pseudo-Pascalovský kód tohto optimalizačného algoritmu dostaneme jednoduchou modifikáciou Algoritmu 2.

Ako ilustračný príklad študujeme chemostat pre unimodálnu funkciu určenú nad binárnymi reťazcami dĺžky 4, chemostat je množina  $P = \{\dots, (1100), \dots\} \subset \{0,1\}^4$ . Každý binárny vektor  $\mathbf{x}$  je ohodnotený racionálnym číslom z intervalu  $\langle 0,1 \rangle$

$$real(\mathbf{x}) = \frac{1}{2^4 - 1} int(\mathbf{x})$$

kde  $int(\mathbf{x})$  je nezáporné celé číslo vyjadrujúce dekadickú interpretáciu binárneho reťazca  $\mathbf{x}$ . Rýchlostná konštanta  $k$  priradená reťazcu

$$k(\mathbf{x}) = f(real(\mathbf{x})) = \frac{1}{2} (1 + \sin(2\pi \cdot real(\mathbf{x})))$$

s optimálnym riešením  $\mathbf{x}_{opt}=(1011)$ , kde  $real(\mathbf{x})=11/15$  and  $f(11/15)=0.9973$ . Chemostat bol inicializovaný 1000 binárnymi reťazcami (0010), mutačný operátor  $O_{mut}$  je špecifikovaný 1-bitovou pravdepodobnosťou  $P_{mut}=0.01$ . Z numerických výsledkov znázornených na obr. 16 vyplýva, že v chemostate spontánne emerguje binárny reťazec, ktorý je blízky k optimálnemu riešeniu.

Z tohto ilustračného príkladu vyplýva, že metafora Eigenových replikátorov poskytuje efektívny *replikátorový optimalizačný algoritmus*, ktorý je veľmi podobný štandardnému genetickému algoritmu [15,25,37], avšak je založený na úplne odlišnej metafore ako GA, metafora darwinovskej evolúcie v GA je nahradená metaforou chemostatu replikátorov. Môžeme teda na záver tejto kapitoly konštatovať, že ***Eigenova fenomenologická teória replikátorov tvorí vhodný teoretický rámec pre počítačové simulačné štúdiá molekulárnej darwinovskej evolúcie*** (t.j. biomakromolekúl, ako sú RNA alebo DNA, ktoré sú schopné replikačných procesov).

### 3.4 Folding (skladanie) binárnych reťazcov

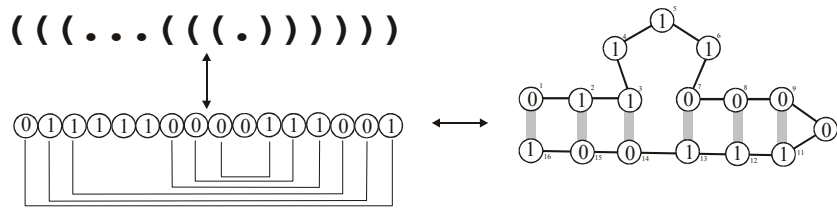
Na základe analógie s RNA molekulami zavedieme tzv. folding (sekundárnu štruktúru) aj pre binárne reťazce (pozri obr. 17), kde úlohu komplementárnych báz hrajú dvojice 0 a 1. Folding binárneho reťazca je špecifikovaný ako zoznam spárovaných binárnych komplementárnych báz  $i-j$  (pre  $i < j$ ) a nespárovaných jedincov  $k$

$$folding(\mathbf{x}) = \{i_1 - j_1, i_2 - j_2, \dots, i_p - j_p; k_1, k_2, \dots, k_q\}$$

kde jednotlivé páry sú ohraničené nasledujúcimi dvoma podmienkami:

(1) pre pár  $i-j$  platí  $j - i \geq 2$ ,

(2) pre dva páry  $i-j$  a  $k-l$ , kde  $i \leq k$ , platí  $i = k \Leftrightarrow j = l$  alebo  $k < j \Rightarrow i < k < l < j$ .



**Obr. 17.** Binárny reťazec môže byť poskladaný do 2-rozmernej štruktúry (je nazývaná sekundárna štruktúra) takým spôsobom, že komplementárne binárne dvojice sú navzájom spojené. Zátvorková reprezentácia foldingu je ukázaná v hornej časti obrázku, pričom bodky sú priradené neinteragujúcim elementom reťazca a zátvorky '(' a ')' sú priradené interagujúcim elementom.

Prvá podmienka znamená, že minimálna dĺžka tzv. „*vlasovej sponky*“ v každom foldingu musí byť aspoň 2 (pozri pravú stranu sekundárnej štruktúry na obr. 17). Ináč povedané, jednotlivé väzby – hrany nemôžu byť ohnuté, prípustná je len rotácia dvoch susedných väzieb okolo rovnakého vrcholu. Z druhej podmienky vyplýva, že každý element reťazca sa môže podieľať maximálne na jednej väzbe, alebo, že tzv. „*pseudouzly*“ sú zakázané (t.j. každý folding sa dá vyjadriť rovinným grafom, kde sa jednotlivé hrany – väzby nepretínajú). Definícia foldingu má ešte podmienku, že obsahuje maximálny možný počet dvojíc s komplementárnymi bázami; táto podmienka odráža fyzikálny význam foldingu, ako najstabilnejšej sekundárnej štruktúry. Podmienky (1-2) sú veľmi dôležité, pretože umožňujú použitie rýchleho algoritmu dynamického programovania [50] pre konštrukciu foldingu, založeného na predpoklade, že hľadanie foldingu je úloha rozložiteľná na menšie podúlohy, ktoré sú použiteľné aj pre konštrukciu riešení väčších podúloh.

Ako už bolo poznamenané, konštrukcia foldingu binárnych reťazcov bude uskutočnená pomocou metódy dynamického programovania [50]. Vo všeobecnosti, túto konštrukciu by sme mohli uskutočniť aj jednoduchým algoritmom spätného prehľadávania, avšak jeho časová náročnosť rastie exponenciálne s dĺžkou reťazca, čo sa pokladá v informatike za málo efektívny algoritmus. K tomu, aby sa zvýšila efektívnosť spätného prehľadávania, zaznamenávajú sa optimálne výsledky pre menšie podreťazce a tieto sa využívajú pre konštrukciu foldingu väčších podreťazcov, tento rekurentný proces je ukončený vtedy, keď dosiahneme celý reťazec, ktorého folding sme hľadali. Táto jednoduchá modifikácia spätného prehľadávania sa nazýva v informatike dynamické programovanie, je použiteľná však len vtedy, keď riešená úloha je rozložiteľná na menšie podúlohy, z ktorých môžeme skladať väčšie podúlohy. Časová náročnosť konštrukcie foldingu pomocou dynamického programovania je  $n^3$ , t.j. exponenciálna zložitosť je zredukovaná na kubickú zložitosť.

Ukážeme elementárnu formuláciu algoritmu na konštrukciu foldingu binárnych reťazcov. Nech  $S_{ij}$  je podreťazec zátvorkovej reprezentácie, ktorý je priradený foldingu medzi uzlami  $i$  a  $j$  (včítane), tieto podreťazce sú inicializované  $S_{ii} = \cdot$  alebo  $S_{i,i+1} = \cdot \cdot$ . Nasledujúce elementy  $S_{ij}$ , pre  $j-i \geq 2$ , sú rekurentne zostrojené pomocou formuly

$$S_{i,j} = \max_{i \leq k \leq j-1} S_{ik} \oplus S_{k+1,j}$$

kde symbol  $\oplus$  reprezentuje spojenie podreťazcov  $S_{ik}$  a  $S_{k+1,j}$ , ktoré už boli zostrojené v predchádzajúcich krokoch algoritmu (pozri Algoritmus 3). Ak  $S_{ik}$  ( $S_{k+1,j}$ ) obsahuje v pravej (ľavej) krajnej polohe symbol ‘.’, pričom oba tieto symboly sú priradené dvojici komplementárnych binárnych elementov reťazca, potom tieto symboly sú nahradené symbolmi ( a ). Symbol “*max*” znamená, že vyberieme taký index  $k$ , ktorý produkuje maximálne párovanie v  $S_{ij}$ . Výsledný folding v zátvorkovej reprezentácii je umiestnený v elemente  $S_{1n}$ .

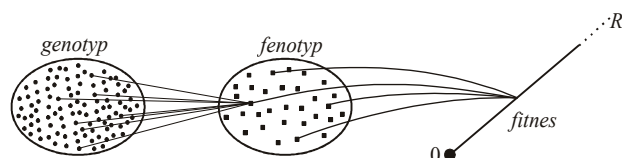
### Algoritmus 3

```

for i:=1 to n S[i,i]:=‘.’;
for i:=1 to n-1 do S[i,i+1]:=‘..’;
for d:=2 to n-1 do
for i:=1 to n-d do
begin j:=d+i;
      S[i,j]:=max{i≤k≤j-1, S[i,k]⊕S[k+1,j]};
end;
folding:=S[1,n];

```

Základný problém pri implementácii tohto algoritmu spočíva v tom, že takto špecifikovaný algoritmus zostrojí len jeden folding z mnohých možných. Z týchto dôvodov algoritmus musí byť charakterizovaný ako nejaká aktuálna „embryogenéza“ fenotypu z daného genotypu a tiež aj ako integrálna časť ohodnocovania genotypu fitness. Všetky možné foldings pre reťazce dĺžky  $n=7$  sú uvedené v Tabuľke 1.



**Obr. 18.** Schématické znázornenie zobrazenia genotypu na fenotyp a fenotypu na fitness pre prípad binárnych reťazcov. Obe zobrazenia sú typu mnoho-na-jeden, t.j. existuje silná redundancia v kódovaní fenotypu pomocou genotypu a v kódovaní fitness pomocou fenotypu. Táto vlastnosť redundancie je dôležitý predpoklad pre existenciu neutrálnych etáp darwinovskej evolúcie, v ktorých je fitness konštantný.

**Tabuľka 1.** Všetky možné genotypy reprezentované binárnymi reťazcami dĺžky  $n=7$  a im priradené fenotypy (foldingy) s hodnotami fitness

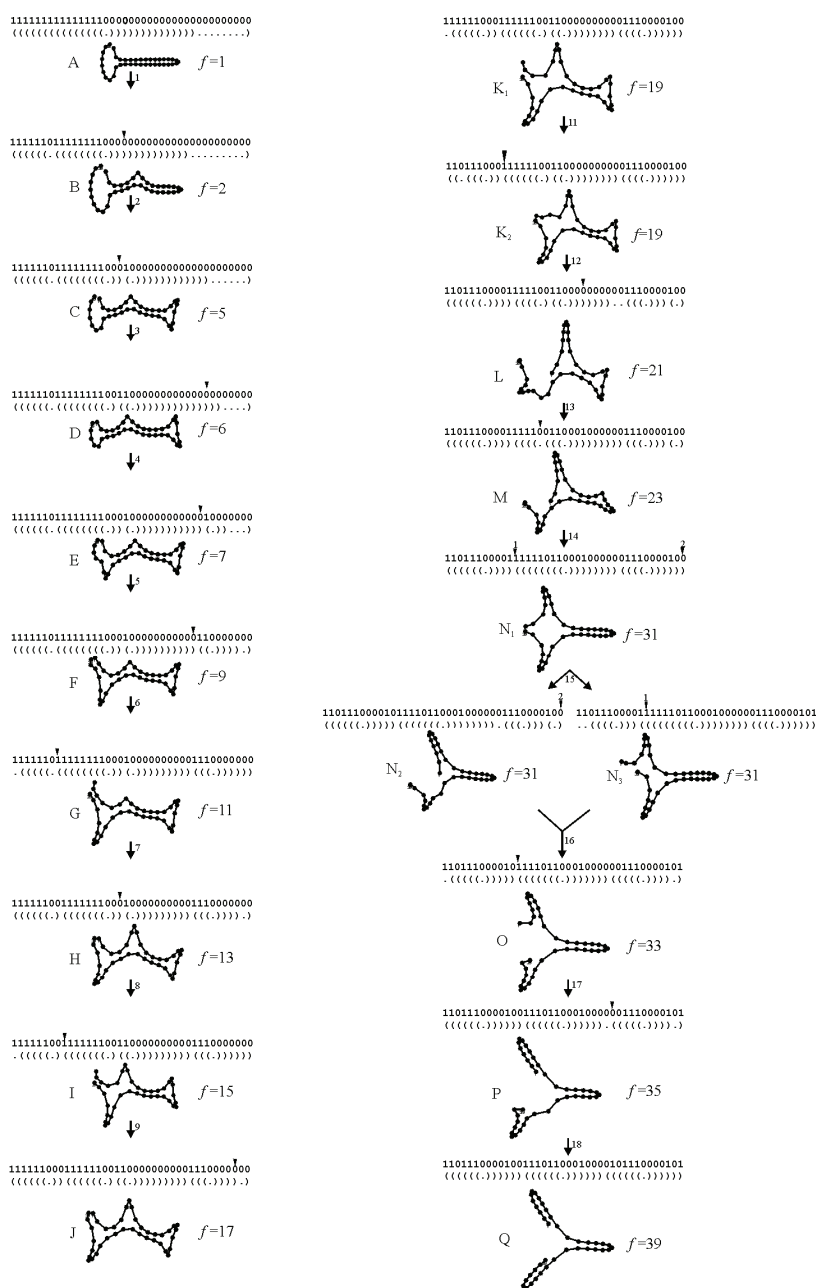
No.	Genotyp	fenotyp	fitness
1	(0000000) (1111111)	.....	1
2	(0000001) (0000100) (1111011) (1111110)	....(.)	1
3	(0000010) (1111101)	...(.).	0
4	(0000011) (0001001) (0001101) (0010010) (0010110) (0011000) (0011100) (0100011) (0100111) (0101001) (0110010) (0110110) (0111000) (0111100) (1000011) (1000111) (1001001) (1001101) (1010110) (1011000) (1011100) (1100011) (1100111) (1101001) (1101101) (1110010) (1110110) (1111100)	..((.))	0
5	(0000101) (0010001) (0101100) (0101110) (0110000) (0111010) (1000101) (1001111) (1010001) (1010011) (1101110) (1111010)	..((.))	3
6	(0000110) (0011010) (1100101) (1111001)	..((.)).	5
7	(0000111) (0001111) (0010011) (0011011) (0100101) (0101101) (0110001) (0111001) (1000110) (1001110) (1010010) (1011010) (1100100) (1101100) (1110000) (1111000)	(((.)))	7
8	(0001000) (1110111)	...(..)	0
9	(0001010) (1110101)	((.)).	0
10	(0001011) (0001100) (0001110) (0011001) (0011110) (0100001) (0100100) (0100110) (0110011) (0110100) (1001011) (1001100) (1011001) (1011011) (1011110) (1100001) (1100110) (1110001) (1110011) (1110100)	..(.).)	0
11	(0010000) (1101111)	..(....)	1
12	(0010100) (0111011) (0111110) (1000001) (1000100) (1101011)	(.)(.)(.)	1
13	(0010101) (0101010) (1010101) (1101010)	..((.))	1
14	(0010111) (0011101) (0100010) (0101000) (1010111) (1011101) (1100010) (1101000)	..(.)(.)	0
15	(0011111) (1100000)	((.)(..)	0
16	(0100000) (1011111)	..(....)	1
17	(0101011) (1010100)	((.)(.)(.)	0
18	(0101111) (1010000)	..(.)(.)	5
19	(0110101) (0110111) (1001000) (1001010)	(.)(..)	0
20	(0111101) (1000010)	..(.)(.)	3
21	(0111111) (1000000)	..(....)	3

Koncepcia foldingu umožňuje zaviesť aj triádu základných pojmov evolučnej biológie: *genotyp*, *fenotyp*, a *fitness*. Genotyp  $x$  je reprezentovaný binárnym reťazcom dĺžky  $n$ ,  $x = (x_1, x_2, \dots, x_n) \in \{0, 1\}^n$ , fenotyp  $ph(x)$  je určený foldingom reťazca  $x$ ; formálne, fenotyp môže byť vyjadrený pomocou zátvorkovej reprezentácie,  $ph(x) = fold(x) \in \{(.), (.), \cdot\}^n$ . Fitness je reprezentovaný kladným reálnym číslom, je atribútom genotypu  $x$ , jeho ohodnotenie fitness je realizované pomocou pridruženého



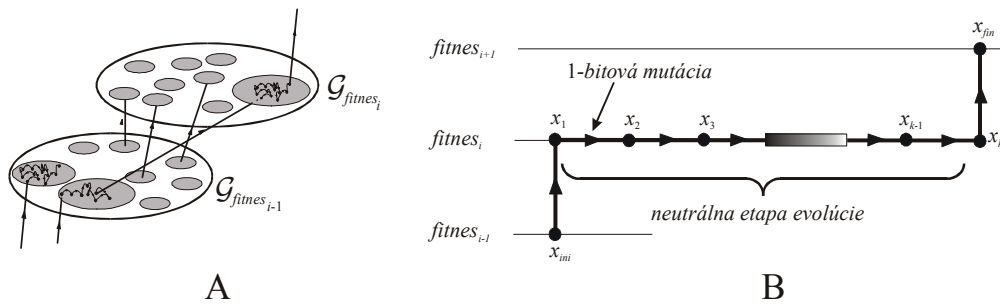






**Obr. 21.** Priebeh evolúcie v chemostate, kde počiatkový genotyp je špecifikovaný binárnym reťazcom  $x_{ini}$  a cieľový fenotyp je špecifikovaný útvarom znázorneným na obrázku 19. Identifikovalo sa 18 prechodných stavov, ktoré sú zostrojené z predchádzajúcich pomocou 1-bitovej mutácie (t.j. medzi danými binárnymi reťazcami je jednotková Hammingova vzdialenosť). Existujú dva prípady neutrálnych mutácií, prechody 11 a 15. Tieto prechody sú rôzneho typu, prvý prechod je priradený postupnosti neutrálnych mutácií, zatiaľ čo, druhý prechod reprezentuje dve paralelné neutrálne mutácie.

Táto „vrtuľová“ sekundárna štruktúra obsahuje tri „listy“, tento cieľový fenotyp bude slúžiť na ohodnocovanie genotypov (binárnych reťazcov) pomocou fitness. To znamená, že očakávame v chemostate evolučnú emergenciu takých genotypov, ktorých folding – fenotyp je špecifikovaný na obr. 19. Operátor mutácie  $O_{mut}$  je špecifikovaný pravdepodobnosťou 1-bitovej mutácie  $P_{mut}=0.0001$ . Táto pravdepodobnosť je dostatočne malá k tomu, aby v chemostate vznikali len blízke genotypy (s malou Hammingovou vzdialenosťou). Veľkosť chemostatu (počet reťazcov) bola 500. Zloženie chemostatu môže byť charakterizované pomocou dvoch rôznych typov entropie, a to genotypovej entropie a fenotypovej entropie, ktoré sú určené takto: nech  $0 \leq w(x) \leq 1$  je koncentrácia genotypov, ktorých daná veličina  $X$  (genotyp alebo fenotyp) je ohodnotená číslom  $x$ , potom  $S_X = -\sum_x w(x) \ln[w(x)]$ . V prípade, že chemostat je homogénny, t.j. obsahuje len genotypy jedného druhu (teda aj fenotypy sú rovnaké), potom tieto entropie sú nulové,  $S_{genotyp} = S_{fenotyp} = 0$ . V opačnom prípade, ak chemostat nie je homogénny, obsahuje aspoň dva rôzne genotypy a fenotypy, potom entropie sú kladné,  $S_{genotyp} > 0$ ,  $S_{fenotyp} > 0$ .

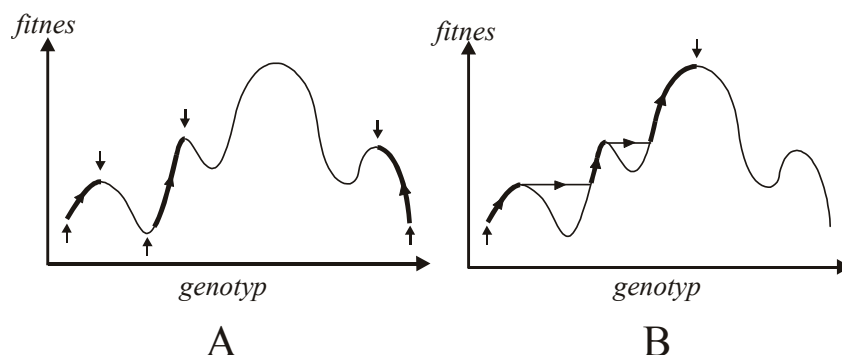


**Obr. 22.** Schematické znázornenie skoku medzi dvoma neutrálnymi etapami, kde sa fitness mení skokom  $f \rightarrow f'$ , symboly  $\mathcal{G}_f$  a  $\mathcal{G}_{f'}$  reprezentujú „neutrálne“ množiny genotypov (binárnych reťazcov) ohodnotených rovnakým fitness  $f$  resp.  $f'$ . (A) V prvej neutrálnej etape, evolúcia vykonáva lokálne prehľadávanie v rámci množiny  $\mathcal{G}_f$ , kde počiatočný genotyp reprezentuje riešenie z posledného „skoku“ a finálne riešenie je schopné „preskočiť“ na riešenie z množiny  $\mathcal{G}_{f'}$ . Potom sa neutrálna etapa znovu opakuje, až je nájdené v rámci množiny  $\mathcal{G}_{f'}$  nové riešenie, ktoré je schopné preskočiť na nasledujúcu neutrálnu množinu. (B) Podrobnejší „rozpis“ neutrálnej etapy s nemenným fitness  $fitness_i$ , postupnosť genotypov  $x_1$  až  $x_k$  tvorí neutrálnu stáziu, v ktorej sa pripravuje evolúcia na ďalší „skok“ do množiny genotypov s väčším fitness.

Numerické výsledky našich počítačových simulácií [35] molekulovej Darwinovej evolúcie sú znázornené na obr. 20 a 21. Vidíme, že vygenerovaný priebeh priemerného fitness (diagram D, obr. 20) je neklesajúca funkcia s niekoľkými relatívne dlhými neutrálnymi etapami – stázami. Diagramy A a B znázorňujú priebehy genotypovej a fenotypovej entropie, ktoré jednoznačne indukujú na konci prvej neutrálnej etapy (okolo epochy 300 000), že existuje podstatná odchýlka od genotypovej homogenity, t.j. v tejto evolučnej etape chemostat je tvorený z niekoľkých fenotypov s rovnakým fenotypom. Táto odchýlka od chemostatov homogenity môže byť považovaná ako

nutná predbežná fáza pred evolučným prechodom do nového „stavu“ špecifikovaného novým fenotypom s vyšším fitness. Rovnaké situácie môžu byť identifikované taktiež pre fenotypové prechody  $A \rightarrow B$  a  $E \rightarrow F$ . Diagram C vyjadruje koncentračné profily dominantných fenotypov, ich tvary majú podobný tvar ako koncentračné profily s koncentračnými profilmi vyplývajúcimi z Eigenovej teórie (pozri obr. 13). Tieto profily tvoria postupnosť dobre separovaných jednotkových maxim. Bodkované vertikálne pomocné čiary indikujú evolučné prechody pomocou entropií. Obr. 21 znázorňuje postupnosť fenotypov (foldingov), od počiatočného z obr. 19A, až po konečný – cieľový z obr. 19B. Podobnosť jednotlivých fenotypov z tejto postupnosti s cieľovým fenotypom určuje fitness odpovedajúcich binárnych reťazcov,  $fitness(\mathbf{x}) = s(ph(\mathbf{x}), ph_{req})$ .

Zo získaných numerických výsledkov vyplýva dôležitosť neutrálnych mutácií v molekulevej Darwinovej evolúcii, umožňujú evolúciu uniknúť z lokálnych riešení na povrchu funkcie fitness, pozri obr. 23. V mnohých prípadoch, keď daný genotyp  $\mathbf{x}$  nie je priamo spojený prostredníctvom 1-bitovej mutácie s iným genotypom  $\mathbf{x}'$  s vyšším fitness (t.j. prechod  $\mathbf{x} \rightarrow \mathbf{x}'$  reprezentuje evolučný skok), potom evolúcia stochasticky hľadá taký genotyp  $\tilde{\mathbf{x}}$ , ktorý leží v susedstve genotypu  $\mathbf{x}$  s rovnakým fitness, avšak tento nový genotyp umožňuje už evolučný skok  $\tilde{\mathbf{x}} \rightarrow \mathbf{x}'$  pomocou 1-bitovej mutácie, pričom  $fitness(\mathbf{x}') > fitness(\tilde{\mathbf{x}})$ , pozri obr. 22B.



**Obr. 23.** (A) Ilustratívny priebeh povrchu funkcie fitness pre molekulevú Darwinovu evolúciu, keď neutrálne mutácie neexistujú. Evolučná adaptácia vždy končí v najbližšom lokálnom maxime (v tomto prípade môže byť Darwinova evolúcia charakterizovaná ako lokálny optimalizátor, kde fitness populácie je „zamrznuté“ na najbližšom maxime). (B) Ak neutrálne mutácie existujú (zobrazenie genotypu na fenotyp je redundantné), potom povrch funkcie obsahuje množstvo „chodníčkov“ s konštantným fitness, ktoré umožňujú evolúciu prekonať hlbokých údolí na povrchu funkcie fitness. Darwinova evolúcia má dobrú šancu k dosiahnutiu globálneho maxima v priebehu adaptačného procesu. (Parafráza na obrázok z publikácie P. Schustra [45].)

## 4 Evolúcia v multiagentových systémoch

Cieľom tejto časti je aplikovať myšlienky univerzálneho darwinizmu k evolúcii multiagentových systémov, ktorých fitness je určený hlavne schopnosťou špecificky interagovať s ostatnými agentmi populácie. To znamená, že genotyp agentov špecifikuje spôsob interakcie s inými agentmi, pričom detaily jeho fenotypu – „telesnej schránky“ – sú ignorované. Tieto aplikácie multiagentových systémov spolu s univerzálnym darwinizmom znamenajú v súčasnosti významný prienik informatických metód k simuláciám sociálnych javov, akými sú evolučná emergencia kooperácie, altruizmu, etnickej kooperácie alebo konfliktu, normy správania sa, jazyka, mémov, atď.

Študujme populáciu agentov (organizmov),  $P = \{A_1, A_2, \dots, A_n\}$ , kde agent  $A$  je špecifikovaný genotypom  $x = x(A)$ . Fitness agentov z populácie je určený vzťahom (pozri sekciu 2)

$$fitness(A) = \sum_{\substack{A' \in P \\ A' \neq A}} F(A, A')$$

ktorý určuje fitness agenta  $A$  ako sumu „platieb“, ktoré sú priradené interakciám daného agenta s ostatnými agentmi populácie. Fitness agentov je plne určený „stratégiou“ vzájomnej interakcie, ktorá je dominantná pre ich prežívanie a reprodukciu, zatiaľ čo architektúra samotného agenta je v tomto prístupe nemenná. V tomto prístupe je fenotyp agenta reprezentovaný stratégiou interakcie s ostatnými agentmi. V zjednodušenom prístupe tieto stratégie splyvajú priamo s genotypom agenta, takže koncepcia dichotómie medzi genotypom a fenotypom agenta v tomto prístupe neplatí.

### 4.1 Sociálne dilemy

Ako vyplýva z teórie hier v ekonómii, aplikovanej tiež často na analýzu pretekov v **zbrojení medzi štátmi**, zjednodušenej zo skúmania ďalej preberaného "problému väzňov" (prisoner's dilemma – väzňova alebo väzenská dilema), kooperácia sa môže vynoriť zo sebeckého záujmu, keď je výhodná pre obidvoch. Vo všeobecnosti, keď hľadáme racionálne dôvody, či pri našej interakcii s niekým iným (s našim obchodným alebo iným partnerom) pristúpim na kooperáciu, alebo nie, naše rozhodnutie by malo byť založené na racionálnych argumentoch a nie na predsudkoch a ideologických dôvodoch [2].

Medzi sociálne dilemy napríklad patrí oblasť životného prostredia (keď ho znečistím, negatívny vplyv na mňa je malý a úspora veľká, aj keď súčet negatívnych vplyvov pre všetkých je významnejší ako moja úspora), obchodné vojny, vyjednávanie medzi krajinami a sociálne interakcie. Tu všade je výber medzi altruistickým správaním, kooperatívnym, alebo egoistickým. Na modelovanie toho sa používajú hry, ktoré dovoľujú analýzu rôznych scenárov.

Ludské spoločnosti sú veľmi komplexné kooperatívne systémy, porovnateľné iba so sociálnym hmyzom. Napriek vonkajšiemu zdaniu spoločnosť okolo nás nie je v svojom rozhodovaní riadená iba krátkodobými sebeckými cieľmi. Ako myšlienkový

dôkaz si môžete predstaviť situáciu, že by človeka vysadili do voľnej prírody. Ako dlho by asi prežil a ako by sa mu darilo? Z tejto predstavy môžeme usúdiť, že kooperácia je v rozvinutej spoločnosti nevyhnutná.

Je najskôr potrebné hľadať niečo podobné ako "neviditeľnú ruku trhu", ibaže vedúcu namiesto súťaženia k spolupráci. Tieto vplyvy najskôr nebudú určené prevažne geneticky na rozdiel od hmyzu, kde je víťazstvo kooperácie udržiavané genetickou homogénnosťou spoločností. Z pohľadu tzv. sebeckého génu [10] pomoc blízkeho príbuznému znamená pomoc sebe samému. Ľudská spoločnosť je ale z tohto pohľadu príliš heterogénna, vysvetlenie spolupráce "sebeckým génom" asi nie je to pravé.

V žargóne teórie hier poznáme súťaživé interakcie ako "hry s nulovým súčtom", čo jeden získa, druhý stratí. Tento model je ale pre prax príliš obmedzujúci, zdroje nie sú konštantné. Keď sebeckosť jedincov spôsobuje, že ich zdatnosť ako jednotlivcov aj ako celku sa znižuje, namiesto toho aby sa zvyšovala, ide o "hry so záporným súčtom".

V tejto kapitole poukážeme na jeden z možných prístupov k vysvetleniu vzniku kooperácie, ktorý je založený na simulačných výpočtoch pomocou hry nazývanej „**väzňova dilema**“.

Nech dvaja zloději boli zatknutí políciou pre vylúpenie *Národnej Pochvej Sporiteľne*, a boli separovane umiestnení do cely predbežného zadržania. Každému z páchatel'ov záleží o mnoho viac na tom, aby bol oslobodený on, ako na tom, aby bol oslobodený jeho spolupáchatel'. Pretože polícia nemala dostatok dôkazov k súčasnému usvedčeniu oboch páchatel'ov, prokurátor dal každému tento „diabolský“ návrh: „*Môžete si vybrať, buď sa priznať k lúpeži, alebo mlčať. Ak sa priznáte vy a váš komplic sa neprizná, tak potom celú vinu za lúpež dám na vášho komplicu, ktorý bude odsúdený na mnohoročný trest a vy budete oslobodený. Podobne, ak vy budete mlčať a váš komplic sa prizná k činu, potom on bude oslobodený a vy dostanete mnohoročný trest. Ak budete obaja mlčať a nepriznáte sa, potom, aj keď nerád, navrhнем súdu, aby vás oboch odsúdil iba na podmienený trest. V opačnom prípade, ak sa obaja priznáte, navrhнем vaše odsúdenie, ale sľubujem vám trest, ktorý tiež bude mnohoročný, ale kratší, ako v prípade, že by sa len jeden z vás priznal a druhý mlčal. Ak sa mienite priznať, dozorcovi nič nepovedzte, na druhý deň ráno vás znovu navštívim a môžete mi oznámiť vaše rozhodnutie*“.

“Dilema”, pred ktorou stoja obaja väzni, spočíva v tom, že nech sa druhý z nich rozhodne akokoľvek, vždy je lepšie nemlčať a priznať sa, než nepriznať svoju vinu. Táto dilema ilustruje konflikt medzi individuálnou a skupinovú racionálnosťou. Skupina, ktorej členovia preferujú svoj vlastný záujem, obvykle skončí horšie, ako tá skupina, ktorej členovia uprednostňujú skupinové záujmy. Tento problém bol navrhnutý a široko diskutovaný už počiatkom 50-tych rokov 20. storočia Merrillom Floodom a Melvinom Dresherom vo vtedy veľmi známej Rand Corporation inštitúcii, kde sa vedci snažili použiť teóriu hier k odhaleniu zákonitostí globálnej nukleárnej stratégie. Väzňova dilema sa stala často používaným prístupom k vysvetleniu vzniku kooperácie v rôznych sociálnych, ekonomických a iných systémoch, kde sa pomocou matematickej teórie hier a/alebo počítačových simulácií hľadajú „racionálne dôvody“ k tomu, prečo a za akých okolností je výhodnejšie spolupracovať, než ako nespôlpracovať.

## 4.2 Základné princípy hry "väzňovej dilemy"

"Väzňova dilema" (VD) [2] je hrou medzi dvoma hráčmi, kde každý z nich má výber z dvoch možností: spolupracovať (cooperate – C) alebo podvádzať (defect – D). Matica platieb tejto hry je daná takto

$$f = \begin{pmatrix} R & S \\ T & P \end{pmatrix} \Leftrightarrow \begin{pmatrix} 3 & 0 \\ 5 & 1 \end{pmatrix}$$

Keď obaja hráči spolupracujú, potom dostanú „odmenu“  $R$  (reward), ktorá by mala byť väčšia, ako bodový zisk „trest“  $P$  (punishment) získaný obidvoma hráčmi, keď podvádžajú. Ale keď jeden hráč podvádza (hrá  $D$ ), zatiaľ čo ten druhý spolupracuje (hrá  $C$ ), potom podvádžajúci získa výplatu bodov  $T$  (temptation - pokušenia), ktorá je väčšia ako  $R$ , zatiaľ čo spolupracujúci dostáva počet bodov  $S$  (sucker – ten, čo naletel) ktorá je menšia ako  $P$ . Platby spĺňajú nerovnosti  $T > R > P > S$  a  $2R > T + S$ . Druhá nerovnosť znamená, že celková výplata pre dvoch hráčov je väčšia keď obaja spolupracujú v porovnaní so súčtom bodov, keď jeden spolupracuje a druhý nie.

Uvažujme hráča – agenta  $A$ , jeho stratégia  $s(A)$  je súbor pravidiel, ktoré použije pri výbere ťahu  $C$  alebo  $D$  (spolupráce alebo podvádžania) proti druhému hráčovi, na základe predošlej histórie hry. Ak hra obsahuje len jednu iteráciu, potom by bolo výhodné nekooperovať - zradiť, pri polovičnej šanci stretnúť spolupracujúceho protivníka. Ak sa však iterácie s tým istým spoluväzňom opakujú a vopred nie je známy ich počet, metóda ako sa zachovať nie je jasná. Hra sa opakuje  $t_{max}$ -krát (dôležitý parameter hry), na záver hry sa spočítajú platby jednotlivých hráčov. Funkcia  $Stratégia(x, t)$  v nasledujúcom algoritme poskytuje stratégiu hráča  $x (= a, b)$  v čase  $t$ . Pre jednoduchosť v našich ďalších úvahách budeme používať stratégiu, pri ktorej hráč zvolí ťah na základe predchádzajúceho ťahu súpera. Stratégia je určená pomocou 3-bitového vektora (t.j. hráči si pamätajú predchádzajúci ťah oponenta)  $s = \{s_1, s_2, s_3\} \in \{0,1\}^3$ , kde jednotlivé zložky (0 – kooperácia, 1 – nekooperácia) majú túto interpretáciu:

(1)  $s_1$  určuje prvý ťah hráča: ak  $s_1=0$ , potom hráč kooperuje, v opačnom prípade, ak  $s_1=1$ , nekooperuje.

(2)  $s_2$  určuje ťah hráča v prípade, že druhý hráč v predchádzajúcom kroku kooperoval: ak  $s_2=0$ , potom hráč kooperuje, v opačnom prípade, ak  $s_2=1$ , nekooperuje.

(3)  $s_3$  určuje ťah hráča v prípade, že druhý hráč v predchádzajúcom kroku nekooperoval: ak  $s_3=0$ , potom hráč kooperuje, v opačnom prípade, ak  $s_3=1$ , nekooperuje.

Takáto stratégia vyžaduje len krátkodobú pamäť hráčov, požaduje len znalosť posledného kroku protivníka - druhého hráča. Celkový počet rôznych 1-stratégií je  $2^3=8$ . Keby si väzni mali pamäť viac krokov do minulosti, s každým takým krokom by exponenciálne narastala zložitosť zápisu stratégie.

Majme dve stratégie  $s(A)=(010)$  a  $s(B)=(101)$ , potom nasledujúcich 5 ťahov hráčov  $a$  a  $b$  a taktiež aj jednotlivé platby na základe matice platieb (1) sú určené pomocou tab. 2.

Tabuľka 2. Ťahy a platby hráčov

ťah	agent A		agent B	
	ťah	platba	ťah	platba
1	0	0	1	5
2	0	3	0	3
3	1	5	0	0
4	1	1	1	1
5	0	0	1	5
suma:		9		14

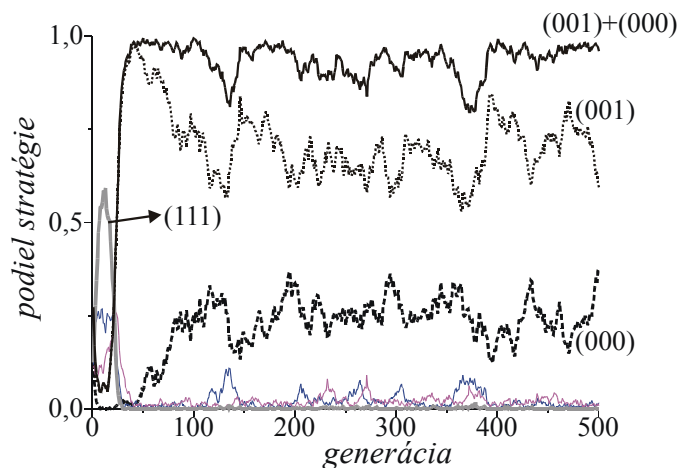
V prvom ťahu obaja hráči použijú ťahy, ktoré sú určené prvou zložkou ich stratégií. V nasledujúcich ťahoch už sú determinovaní ťahom súpera v predchádzajúcom ťahu.

### 4.3 Simulácia emergencie stabilnej stratégie pomocou genetického algoritmu

Pri plánovaní stratégie na základe predchádzajúcich skúseností s protihráčom sa často používa genetický algoritmus. Táto aplikácia genetického algoritmu prvýkrát vznikla zo spolupráce amerického politológa Roberta Axelroda [2] s informatikom a tvorcom genetických algoritmov Johnom Hollandom a rozvíjala sa ďalej pre zložitejšie modely medzinárodnej bezpečnosti odpovedajúce napr. na otázku, koľko z rozpočtu vydávať na obranu. V časovom priebehu je zaujímavé sledovať vývoj kooperatívnych a nekooperatívnych stratégií. Kooperatívne stratégie sa spočiatku ukázali menej výhodné, no v neskorších štádiách prevážili.

Základné princípy genetického algoritmu použitého k štúdiu emergencie „najvýhodnejšej“ stratégie sú rovnaké ako u klasického genetického algoritmu. Populácia je zložená z chromozómov - stratégií. Algoritmus je inicializovaný náhodne vytvorenou populáciou stratégií. Fitnes chromozómov sa určí pomocou "megaturnaja", kde všetky možné dvojice hrajú VD hru, celková platba každého chromozómu získaná v tomto megaturnaji je fitnes. Štandardná realizácia reprodukčného procesu obsahuje selekciu dvoch stratégií na základe ich fitnes (stratégie s väčším fitnes majú väčšiu pravdepodobnosť byť vybrané do reprodukčného procesu), ktorá sa obvykle realizuje pomocou Goldbergovej rulety. Vybrané stratégie vstupujú do vlastného procesu reprodukcie, ktorý obsahuje kríženie a mutáciu. Výsledné stratégie – potomkovia tvoria novú populáciu, ktorá keď obsahuje rovnaký počet potomkov, ako mala pôvodná populácia počet rodičov, nahradí rodičovskú populáciu populáciou potomkov. Numerické výsledky genetického algoritmu sú znázornené na obr. 24. Ako výsledná stratégia emergujú dve podobné stratégie (001) a (000), ktoré obe zahajujú hru kooperáciou a na kooperáciu odpovedajú kooperáciou. Líšia sa v reakcii na podvod, prvá stratégia (001) reaguje na podvod podvodom, zatiaľ čo druhá stratégia (000) „hlúpo“ reaguje na podvod kooperáciou. Stratégia (001) sa nazýva podľa Axelroda tit-

for-tat (vo voľnom preklade, ako ty mne, tak ja tebe). Menšinová stratégia „dobrák“ (000) má šancu prežiť iba popri stratégii tit-for-tat.<sup>10</sup>



**Obr. 24.** Výsledky genetického algoritmu použitého pre emergenciu výslednej stratégie, pričom populácia stratégií bola inicializovaná náhodne. V priebehu evolúcie algoritmu spontánne emergovali dve „kooperatívne“ stratégie (001) a (000), ktoré zahajujú hru kooperáciou a na kooperáciu súpera v predchádzajúcom ťahu odpovedajú kooperáciou v nasledujúcom ťahu. V dôsledku toho, že väčšinová stratégia (001) neprodukuje „podvod“, menšinová stratégia (000) nemá príležitosť reagovať spoluprácou na podvod.

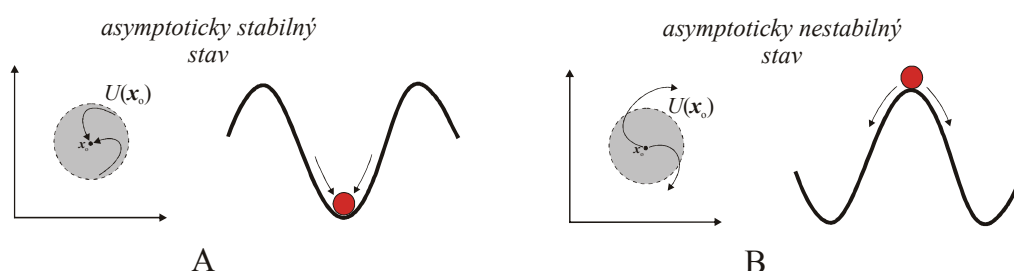
#### 4.4 Evolučne stabilné stratégie

Z evolučnej teórie hier zavádza veľmi užitočný pojem „evolučne stabilná stratégia“, ktorý je vo veľmi blízkej analógii s pojmom „asymptoticky stabilný stav“, pochádzajúcim z dynamickej teórie systémov [41], pozri obr. 25.

Asymptoticky stabilný stav je charakterizovaný podmienkou, že ak ho trochu vychýlime z jeho rovnovážnej polohy, potom v dynamickom systéme nastanú procesy, ktoré ho vrátia do pôvodnej polohy. Asymptoticky nestabilný stav sa už nevráti po vychýlení do pôvodnej polohy. Aby sme mohli použiť túto analógiu z dynamických systémov, pojem stav nahradíme novým pojmom „stratégia“, ktorý špecifikuje interakcie agenta s inými agentami populácie. Na základe analógie z teórie dynamických systémov, definujeme *evolučne stabilnú stratégiu* (ESS) [39] tak, že ak populácia  $P$  obsahuje agentov prevažne len s referenčnou stratégiou a taktiež podstatne menej iných agentov s menšinovou stratégiou, potom evolučné procesy spontánne prebiehajúce v populácii po určitom čase túto menšinovú stratégiu eliminujú z populácie.

<sup>10</sup> ďalšia úspešná stratégia založená na trocha inom princípe sa volá Pavlov. Keď jedinec dostane odmenu  $T$  alebo  $R$ , opakuje svoj krok, keď dostane  $P$  alebo  $S$ , prepne z nespôlpráce na spoluprácu alebo naopak. Ešte zložitejšie sú pravdepodobnostné stratégie.





**Obr. 25.** Znáznorenie asymptoticky stabilného (A) a asymptoticky nestabilného (B) stavu s jednoduchou fyzikálnou interpretáciou guľôčky, ktorá je buď v jamke (príklad asymptoticky stabilného stavu), alebo je na vrchole kopčeka (príklad asymptoticky nestabilného stavu). Ľavé časti obrázkov odpovedajú presnej matematickej definícii stavov, pre asymptoticky (ne)stabilný stav  $x_0$  existuje také okolie  $U(x_0)$ , že ak trajektória má počiatočný stav v tomto okolí, potom asymptoticky (pre  $t \rightarrow \infty$ ) trajektória vždy končí v bode  $x_0$  (vždy opúšťa okolie).

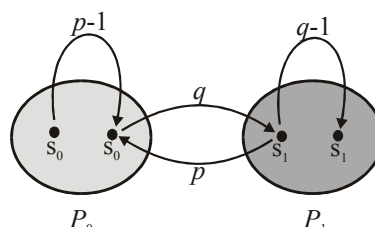
Nech populácia  $P$  obsahuje prevažne väčšinovú referenčnú stratégiu  $s_0$  a v malom rozsahu aj menšinovú stratégiu  $s_1$ .

$$P = P_0 \cup P_1, P_0 = \{s_0, s_0, \dots, s_0\}, |P_0| = p, P_1 = \{s_1, s_1, \dots, s_1\}, |P_1| = q$$

pričom predpokladáme, že počet stratégií  $s_0$  je podstatne viac ako menšinových stratégií  $s_1$ ,  $p \gg q \gg 0$  (pozri obr. 26). Fitness týchto dvoch stratégií je určený takto

$$f(s_0) = (p-1)F(s_0, s_0) + qF(s_0, s_1)$$

$$f(s_1) = (q-1)F(s_1, s_1) + pF(s_1, s_0)$$



**Obr. 26.** Výpočet fitness pre referenčnú stratégiu  $s_0$  a menšinovú stratégiu  $s_1$ . Podpopulácia  $P_0$  obsahuje  $p$  referenčných stratégií  $s_0$  a podpopulácia  $P_1$  obsahuje  $q$  menšinovej stratégie  $s_1$ .

Pretože evolúcia populácie prebieha na základe prirodzeného výberu, podmienka evolučnej stability stratégie  $s_0$  má tvar

$$f(s_0) > f(s_1) \Rightarrow (p-1)F(s_0, s_0) + qF(s_0, s_1) > (q-1)F(s_1, s_1) + pF(s_1, s_0)$$

Z predpokladu  $p \gg q \gg 0$  vyplýva, že stratégia  $s_0$  je evolučne stabilná, ak platí buď prvá podmienka (dominantná) alebo druhá podmienka (vedľajšia, keď dominantná neplatí)

$$F(s_0, s_0) > F(s_1, s_0)$$

$$F(s_0, s_0) = F(s_1, s_0) \text{ a } F(s_0, s_1) > F(s_1, s_1)$$

Podmienka ESS pre referenčnú stratégiu  $s_0$  vzhľadom k menšinovej stratégii obsahuje dve kritéria, buď je splnené prvé kritérium, alebo ak nie je splnené

(podmienka rovnosti), potom musí platiť splnené druhé kritérium, ktoré je však podstatne subtilnejšie vzhľadom k prvému kritériu (rezultujúci fitness stratégií je násobkom platieb parametra  $q$ , ktorý je však podstatne menší ako parameter  $p$ , od ktorého je odvodené prvé kritérium).

Týmto sme dostali jednoduché podmienky, ktoré špecifikujú, či daná stratégia  $s_0$  je v prítomnosti malého počtu vtreleckých stratégií  $s_1$  evolučne stabilná. Podmienky požadujú len znalosť funkcie platieb pre jednotlivé stratégie. Tieto platby urobené ako súčet platieb z piatich ťahov za sebou pre všetky možné dvojice stratégií sú ukázané v tab. 3.

**Tabuľka 3.** Súčet platieb z piatich ťahov za sebou pre všetky dvojice stratégií

	(000)	(001)	(010)	(011)	(100)	(101)	(110)	(111)
(000)	15	15	3	3	12	12	0	0
(001)	15	15	12	6	4	10	9	4
(010)	23	12	11	4	18	9	0	0
(011)	23	11	19	7	20	8	16	4
(100)	17	14	8	5	13	1	4	1
(101)	17	15	14	8	15	5	10	5
(110)	25	14	25	6	19	10	9	1
(111)	25	9	25	9	21	5	21	5

Dvojica stratégií 001 a 111 vyhovuje podmienke  $F(s_0, s_0) > F(s_1, s_0)$ , teda by teoreticky mohli byť stabilné, keby nebolo dvojíc stratégií 000 – 001 a 101 – 111. Tieto dvojice stratégií majú pri vzájomnej hre rovnaké ohodnotenie. Stratégia 000 sa teda popri stratégii 001 môže náhodnými mutáciami rozšíriť. Potom, čo sa rozšíri, môže náhodne vzniknúť ďalšia stratégia, ktorá ich využije na svoje rozšírenie. Podobne to platí aj pre dvojicu stratégií 101 – 111.

Keby stratégie 000 a 101 boli zakázané, potom by stratégie 001 a 111 boli evolučne stabilné. Každá dvojica by bola v tzv. Nashovej rovnováhe<sup>11</sup>, teda stratégia, ktorá by nahradila jednu stratégiu z dvojice 001 – 001 alebo z dvojice 111 – 111 by si zhoršila platbu. V prípade nahradenia tit-for-tat nespolupracujúcim by síce nespolupracujúci skončil horšie ako predtým, ale relatívne lepšie ako jeho spoluhráč. Vzhľadom na to, že sa ale fitness porovnáva v rámci celej populácie, nespolupracujúci by na tom bol stále zle, jeho spoluhráč tit-for-tat by nabral fitness v interakciách s ostatnými tit-for-tat (pozri obr. 27).

Podobne sa dajú študovať aj stratégie agentov, ktorí si pamätajú dva alebo viac krokov do minulosti. Spolu so zložitosťou stratégií ale narastá tendencia k nespolupráci

<sup>11</sup> Keď existuje súbor stratégií s takou vlastnosťou, že žiaden hráč nemôže profitovať na zmene svojej stratégie, zatiaľ čo ostatní hráči svoju stratégiu nezmenia, potom tento súbor stratégií je v Nashovej rovnováhe (spomeňte si na film Čistá duša – Beautiful mind).

a chaotické správanie populácie, aj keď je možné tieto tendencie prekonať napríklad ustanovením „administrátora“, ktorý by stochasticky trestal nespoluprácu.

Pomocou tohto výsledku pokúsime sa odvodiť podmienku pre ESS stratégie  $s_0$ , ktorá obsahuje len diagonálne platby. Nech nediagonálna platba pre  $s_0$  a  $s_1$  je približne určená vzťahom

$$F(s_1, s_0) \doteq \frac{1}{2} [F(s_0, s_0) + F(s_1, s_1)] \left( 1 - \frac{d(s_1, s_0)}{d_{max}} \right)$$

kde  $d(s_1, s_0)$  je „vzdialenosť“ medzi stratégiami  $s_0$  a  $s_1$  a  $d_{max}$  je jej postulovaná maximálna hodnota. Nediagonálna platba  $F(s_1, s_0)$  je teda určená ako priemer príslušných diagonálnych hodnôt krát člen špecifikujúci podobnosť medzi stratégiami. Dosadením tohto vzťahu do prvej podmienky ESS po jednoduchých algebraických úpravách dostaneme

$$F(s_0, s_0) > F(s_1, s_1) \frac{d_{max} - d(s_1, s_0)}{d_{max} + d(s_1, s_0)}$$

Nech stratégia  $s_1$  vznikne zo stratégie  $s_0$  1-bodovou (minimálnou) mutáciou, potom môžeme položiť  $d(s_1, s_0) = 1$ , podmienka pre evolučnú stabilitu  $s_0$  vzhľadom k  $s_1$  má tento konečný tvar

$$F(s_0, s_0) > F(s_1, s_1) \frac{d_{max} - 1}{d_{max} + 1}$$

Táto podmienka je podstatne jednoduchšie aplikovateľná ako jej pôvodný tvar požadujúci aj znalosť nediagonálnych platieb. Jeden z dôležitých dôsledkov tejto formule je výsledok, že ak  $s_0$  je nejaká evolučne výsledná dominantná stratégia a predpokladáme, že vedľajšie stratégie z nej odvoditeľné pomocou 1-bodovej mutácie sú veľmi blízke pôvodnej dominantnej stratégii  $s_0$ ,  $F(s_0, s_0) = F(s_1, s_1)$ , potom dominantná stratégia je evolučne stabilná, lebo  $d_{max} - 1 < d_{max} + 1$ . K tomu, aby sa vedľajšia stratégia stala dominantnou je potrebné, aby diagonálna platba  $F(s_1, s_1)$  bola podstatne väčšia ako platba  $F(s_0, s_0)$ , potom aj napriek pomerne veľkej zotrvačnosti pretrvávania pôvodnej stratégie  $s_0$  presadí sa evolučne (po peripetiách jej fluktuálneho vracania sa do dominantného postavenia) odvodená stratégia  $s_1$ .

#### 4.5 Tragédia spoločného

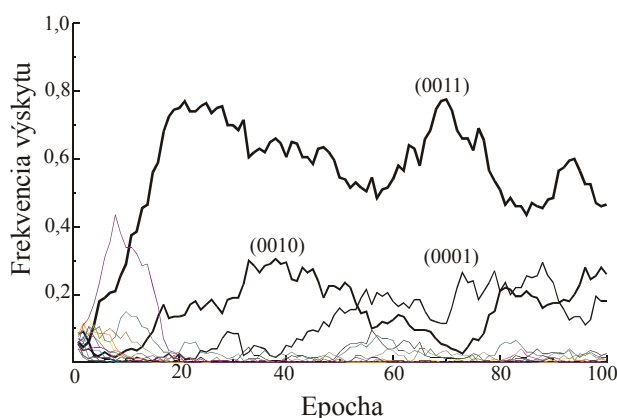
Modelový príklad egoistických tendencií erodujúcich spoluprácu je tzv. tragédia spoločného definovaná Garrettom Hardinom r. 1968 [24]: "Predstavte si pastvinu prístupnú pre všetkých. Dá sa očakávať, že každý pastier sa pokúsi držať na spoločnej pastvine toľko dobytká, ako sa len dá. Ako racionálny jedinec, každý pastier sa snaží maximalizovať svoj zisk. Explicitne alebo implicitne sa pýta: "Akú zmenu výnosu pre mňa znamená prídavok jedného alebo viac zvierat k môjmu stádu?"

1. Pozitívna časť je funkciou prínosu jedného zvierat'a. Pretože pastier dostáva všetok zisk z predaja zvierat'a, možný príspevok sa blíži + 1.

2. Negatívna časť je funkciou prídavku prílišného spásania pridaným zvierat'om. Pretože ale výsledky prílišného spásania sú zdieľané všetkými pastiermi, negatívny príspevok pre ktoréhokoľvek rozhodujúceho sa pastiera je iba zlomkom z - 1.

Keď si racionálny pastier spočíta pozitívnu a negatívnu časť prínosu, vychádza mu, že jedinú rozumnú riešenie je preňho pridať ďalšie zviera k stádu. A ešte ďalšie ...

Ale toto je záver každého racionálneho pastiera zdieľajúceho pastvinu. V tom je tá tragédia. Každý človek je uväznený v systéme, ktorý ho núti zväčšovať svoje stádo bez obmedzenia - v svete, ktorý je limitovaný. ... Sloboda v užívaní spoločnej pastviny prináša skazu pre všetkých."



**Obr. 27.** Tragédia spoločnej pastviny a jej modelovanie tromi pastiermi. Výsledok genetického algoritmu – zobrazenie frekvencie výskytu stratégií pre spoluprácu troch hráčov. Ako je vidno, vyhráva stratégia 0011, ktorá znamená, že na začiatku je výhodné spolupracovať, a pokiaľ všetci spolupracujú, aj pre mňa je výhodné spolupracovať. Pokiaľ ale iba jeden nespupracuje, je aj pre mňa výhodné nespupracovať. Ku koncu vývoja sa vyskytuje aj stratégia 0001 a 0010, ktorá pripúšťa možnosť spolupráce, aj keď jeden alebo dvaja spoluhráči z turnaja nespupracujú. To je ale výchylka spôsobená tým, že ku koncu vývoja sa vyskytujú väčšinou jedinci zo stratégie 0011, a teda situácia stretnutia s nespupracujúcimi jedincami prakticky nenastáva.

Vyššie uvedený príklad sa dá modelovať tiež pomocou vážňovej dilemy, kedy sa namiesto dvoch vážňov interagujú traja alebo viac. Zovšeobecnená vážňova dilema pre viac hráčov (konkrétna stratégia je pre troch) môže vyzeráť napríklad takto: Stratégia (0 1 1 0) v prvom kroku kooperuje, nekooperuje, keď v predošlom kroku všetci ostatní kooperovali, nekooperuje, keď v predošlom kroku jeden spoluhráč nekooperoval, kooperuje, keď v predošlom kroku všetci ostatní kooperovali. Výsledok je obr. 27.

Všeobecné vlastnosti víťaznej stratégie: hra je *zahájená kooperáciou*, ak v predchádzajúcom ťahu *všetci oponenti kooperovali, potom kooperuj*, ak v predchádzajúcom ťahu *aspoň jeden oponent nekooperoval, potom nekooperuj*.

Pretože vo väčších celkoch sa vždy nájde nekooperujúci jedinec, nastáva všeobecná nespupráca - tragédia spoločnej pastviny napr. u rybolovu v

medzinárodných vodách, pri rúbaní pralesov, starosti o spoločné zavodňovanie alebo znečisťovanie atmosféry. Politológ Axelrod tvrdí, že k tomu, aby v spoločnosti vznikla kooperácia, musia v nej existovať inštitúcie, ktoré kontrolujú - penalizujú nekooperáciu.

**Tabuľka 4.** Prvých päť ťahov zovšeobecnenej väzňovej dilemy pre troch hráčov, ktorých stratégie sú určené ako (0110), (1100), (0011)

No.	Ťah			Skóre		
	1. hráč	2. hráč	3. hráč	1. hráč	2. hráč	3. hráč
1	0	1	0	3	10	3
2	1	1	1	2	2	2
3	0	0	1	3	3	10
4	1	0	0	10	3	3
5	1	0	1	6	0	6

Celkové skóre:

24

18

24

Okrem zavedenia administrátora trestajúceho nespôluprácu sa emergenciu spolupráce dá dosiahnuť aj priestorovým rozmiestnením stratégií, ktoré potom môžu interagovať iba s blízkymi stratégiami, alebo zavedením "dobrého mena" jedincov, kedy si jedinci vymieňajú informácie o spolupráci či nespôlupráci ostatných, a k tým, čo nespôlupracovali s ostatnými, sa správajú, ako keby nespôlupracovali s nimi - tzv. moralistická agresia. Ďalšou možnosťou je aj tzv. skupinový výber, kde súťaž prebieha nielen medzi jedincami v rámci populácie, ale aj medzi populáciami. Populácia s príliš veľa sebcami má oproti ostatným populáciám nižšie priemerné skóre u jedincov, preto zanikne. U biológov je táto predstava uznávaná iba obmedzene, pretože rýchlosť výmeny jedincov je väčšia, ako rýchlosť výmeny populácií, a teda populácia sa skôr nakazí egoizmom, ako by mala zvíťaziť nad inou už egoistickou populáciou. U spoločností ale toto vysvetlenie môže mať svoj význam.

Evolučné štúdium väzňovej dilemy sa dá použiť aj na modelovanie vnútro- a medzietnických vzťahov [38]. U najjednoduchšieho modelu s dvoma etnikami medzi jedincami populácie  $P$  existujú tri typy interakcií: vnútroetnické interakcie v rámci etnickej skupiny  $A$ , vnútroetnické interakcie v rámci etnickej skupiny  $B$ , a konečne, medzietnické interakcie medzi jedincami z  $A$  a  $B$ . Postulujeme, že frekvencia medzietnickej interakcie je oveľa menšia ako frekvencia vnútroetnických interakcií v  $A$  alebo  $B$ . Jediniec je vybavený zafixovanou stratégiou väzenskej dilemy, dokáže rozpoznať etnickú skupinu druhého a jeho penalizáciu, čo je označenie nespôlupracujúcich jedincov (alebo etnika) spolupracujúcimi jedincami. Evolúcia populácie prebieha nezávisle pre obidve etnické skupiny  $A$  a  $B$ . Fitnes jedincov je vypočítaný pomocou "turnaja", kde je predpísaný počet-krát náhodne vybraná dvojica agentov, ktorí spolu hrajú jeden ťah hry väzňovej dilemy. K štúdiu medzietnických interakcií boli použité dve rozdielne evolučné stratégie.

V prvej stratégii, ktorá sa nazýva *kolektívna vina*, je potrestaná celá etnická skupina za akt nespôlupráce svojho člena voči členovi druhej etnickej skupiny. Zo

simulačných výpočtov a teoretických úvah vyplýva, že tento typ stratégie nevedie k emergencii spolupráce medzi rozdielnymi etnickými skupinami, pokiaľ nezavedieme administrátora pokutujúceho akty nespôlupráce. Keď etnické skupiny majú veľmi rozdielne veľkosti, princíp kolektívnej viny môže byť veľmi devastujúci pre menšiu etnickú skupinu, v ktorej môže zaniknúť aj vnútroetnická spolupráca.

Druhou použitou stratégiou pre podporu spolupráce je tzv. *osobná zodpovednosť*, kedy člen skupiny, ktorý nespôlpracuje v rámci medzietnických interakcií, je potrestaný osobne v rámci svojej etnickej skupiny. To znamená, že na rozdiel od princípu kolektívnej viny, v tejto stratégii existuje len jeden typ "potrestania", členovia sú označení penalizáciou osobne. Simulačné výpočty spolu s teoretickou analýzou ukazujú, že v tomto prípade môže emergovať medzietnická spolupráca.

## 5 Závěry

Metafora univerzálneho darwinizmu rozširuje princípy darwinizmu ako univerzálneho prístupu k štúdiu evolúcie nielen v biológii, ale aj v iných vedných oblastiach (chémia, sociálne vedy, informatika, umelá inteligencia, kognitívna veda,...). V informatike univerzálny darwinizmus poskytuje jedinečnú možnosť simulácií darwinovskej evolúcie prostriedkami „in silico“ (t. j. počítačom), môžeme konštatovať tieto všeobecné závery:

- Evolučné algoritmy tvoria robustný optimalizačný prostriedok pre rôzne kombinatoriálne úlohy (napr. z teórie grafov, operačného výskumu, inžinierskej praxe návrhu zložitých zariadení, a pod.).
- Evolučné multiagentové systémy umožňujú počítačové simulácie sociálnych systémov, emergenciu kooperácie a komunikácie vypracovať evolučné základy memetiky ako špeciálneho prípadu multiagentových systémov, študovať etnické konflikty v multiagentových systémoch, a pod.
- Univerzálny darwinizmus umožňuje informatike aktívne interagovať so sociálnymi a behaviorálnymi vedami.

Z ilustračných výpočtov molekulovej Darwinovej evolúcie vyplývajú tieto dôležité všeobecné závery pre univerzálny darwinizmus:

(1) Eigenova *teória replikátorov* tvorí dobrý fenomenologický základ molekulovej darwinovej evolúcie. Priebehy koncentračných profilov získaných riešením diferenciálnych rovníc popisujúcich dynamiku chemostatu replikátorov, ktoré boli získané vhodným výberom rýchlostných konštánt, poskytujú priebehy koncentrácií, ktoré sú veľmi podobné experimentálne pozorovaným [47] (pozri obr. 13).

(2) Darwinovská evolúcia prebieha ako Monodovská [42] vzájomná hra medzi *náhodou a nutnosťou*, medzi deterministickým a stochastickým charakterom, ktoré sú jej integrálnou súčasťou. Evolúcia obsahuje časti, ktoré sú plne deterministické – plne predpovedateľné (napr. zobrazenie genotypu na fenotyp), a taktiež aj stochastické časti, ktoré nemôžu byť dobre predpovedateľné, môže hovoriť len o ich štatistických charakteristikách (napr. mutácie). V tejto súvislosti sa občas používa terminológia zavedená americkým informatikom a zakladateľom umelého života Christopherom

Langtonom „na hrane chaosu“ (angl. *on the edge of chaos*). Používa sa termín „evolúcia prebieha na hrane chaosu“, ktorý sa interpretuje takto: Hlavnou hnacou silou inovácií v evolúcii sú mutácie, preto by sme sa mohli domnievať, že čím častejší je výskyt mutácií pri reprodukčnom procese, tým väčšia je pravdepodobnosť výskytu inovácie. Avšak, ak miera mutácií prekročí určitú kritickú hodnotu, efektívnosť evolúcie sa zničí a stáva sa z nej niečo, čo môžeme nazvať „slepé hľadanie“, bez vplyvu stratégie prirodzeného výberu, ktorý je schopný efektívne selektovať pozitívne mutácie. V opačnom prípade, ak miera výskytu mutácií je veľmi malá, evolúcia sa v podstate zastaví a zamrzne v aktuálnom stave. Preto môžeme povedať, že frekvencia výskytu mutácií musí byť dostatočne veľká, ale zas nie moc, aby sa evolúcia nerozpadla, hovoríme, že musí byť na hrane chaosu.

(3) Konceptia Sewalla Wrighta [51] **povrchu funkcie fitness** môže byť považovaná ako jeden z najväčších úspechov Darwinovej evolučnej teórie a taktiež aj ako jeden zo zdrojov jej univerzálneho charakteru. Táto koncepcia má veľký heuristický význam v tom, že Darwinova evolúcia môže byť interpretovaná ako špeciálny druh evolučného algoritmu [15,25,37], ktorý je schopný riešiť extrémne komplikované kombinatoriálne optimalizačné problémy. V prvej sekcii bol formulovaný Wrightov názor na evolúciu ako na optimalizačný proces na povrchu funkcie fitness (pozri obr. 2). V tejto súvislosti Wright poznamenal, že jeden z hlavných cieľov teórie evolúcie je popísať mechanizmy, ktoré umožňujú evolúcii nájsť „spojité cesty“ z jedného vrcholu do druhého vrcholu na povrchu fitness funkcie, k tomu, aby evolúcia bola schopná nájsť globálne riešenie (maximum). Ak by takáto možnosť neexistovala, potom by evolúcia vždy „zamrzla“ v lokálnom riešení, ktoré by nebola schopná opustiť. Naše modelové výpočty pomocou molekulovej Darwinovej evolúcie umožňujú popísať evolučný mechanizmus úniku z lokálnych riešení (toto riešenie je široko študované viedenským teoretickým chemikom a biológom Petrom Schusterom so spolupracovníkmi [45,46,15-20]) pomocou existencie neutrálnych mutácií, ktoré menia genotyp ale neprejavujú sa zmenou fenotypu (foldingu).

(4) Ako už bolo poznamenané v predchádzajúcom bode, použitý model molekulovej evolúcie umožňuje vysvetliť **úlohu neutrálnych mutácií** v Darwinovej evolučnej teórii. (pozri obr. 22 a 23). Existencia neutrálnych mutácií na povrchu funkcie fitness je významná pre vymanenie sa z evolučných pascí lokálnych miním. Darwinova evolúcia obsahuje dlhodobé neutrálne stázy, kde priemerný fitness populácie je konštantný (môže vykazovať malé fluktuácie), ale zloženie populácie stochasticky „vandruje“ smerom ku genotypu, z ktorého je možný „skok“ na genotyp s väčším fitness, pomocou 1-bitovej mutácie. Dĺžka týchto neutrálnych evolučných epoch je úmerná pravdepodobnosti náhodného „skoku“ na genotyp s väčším fitness. Ak je táto pravdepodobnosť veľmi malá, potom neutrálna epocha môže trvať veľmi dlho. V opačnom prípade, ak je pravdepodobnosť „skoku“ veľká, tak potom neutrálna epocha netrvá dlho. Existencia neutrálnych mutácií je podmienená typom zobrazenia genotypu na fenotyp, ak je toto zobrazenie vysoko redundantné, tak potom pre daný genotyp existuje mnoho neutrálnych mutácií, ktoré zachovávajú nemennosť fenotypu.

(5) **Časová orientácia** Darwinovej evolúcie je jednoznačná, táto skutočnosť sa prejavuje napr. existenciou neklesajúceho priebehu priemerného fitness. Genotypy s väčším fitness sa replikujú častejšie, tieto genotypy majú určitú selekčnú výhodu pred ostatnými genotypmi s malým fitness. Inými slovami, Darwinova evolúcia je taká progresívna zmena priemerného populačného genotypu, ktorá sa prejaví neklesajúcim charakterom priemerného fitness v priebehu celej evolúcie.

(6) Dve rôzne časové merítka môžu byť v evolúcii identifikované. **Adaptívna epocha** spočíva v náhlej zmene priemerného fitness (pozri obr. 20, diagram D), kde koexistujú dva rozdielne fenotypy; pôvodný fenotyp má menší fitness, zatiaľ čo nový fenotyp má väčší fitness. Pretože pravdepodobnosť replikácie genotypov je úmerná ich fitness, genotypy priradené novému fenotypu majú väčšiu pravdepodobnosť sa replikovať, než pôvodné genotypy. Pretože nové genotypy v populácii rýchlo zvíťazia, adaptívna fáza, kde staré genotypy sú nahradzované novými genotypmi, sa zdá externému pozorovateľovi ako extrémne rýchly proces. Na druhej strane, Darwinova evolúcia obsahuje **aj neutrálnu epochu**, ktorá spočíva v dlhodobej stáze, kde existujú stochastické mutácie pripravujúce nasledujúcu adaptívnu epochu.

(7) Evolučné vlastnosti chemostatového modelu sú inštruktívne špecifikované pomocou entropií dvoch typov [35]. **Fenotypová entropia** postihuje zmeny v zložení binárnych reťazcov vzhľadom k ich fenotypu (fitness), to znamená, že tento typ entropie nepostihuje neutrálne mutácie. Druhý typ entropie, **genotypová entropia**, je veľmi citlivá na každé zmeny v chemostate, vrátane aj neutrálnych mutácií. Preto súčasné sledovanie oboch entropií v priebehu evolúcie nám umožňuje odlíšenie medzi neutrálnymi a inými mutáciami na evolučnej trajektórii chemostatu.

(8) V evolúcii neexistuje zámer, ktorý by ju riadil smerom k požadovanému cieľu. Ako bolo uvedené v predchádzajúcej časti tejto práce, fitness replikátorov odráža aktuálny stav replikátorov, ich aktuálnu schopnosť vstupovať do reprodukčného procesu a prežívať v populácii. Môžeme povedať, že Darwinova evolúcia je oportunistická, kde rekurzívne obnovovanie populácie je založené len na parametroch aktuálnej populácie. Táto skutočnosť platí len v ohraničenej miere v multiagentových systémoch, kde dochádza ku koevolúcii génov a mémov, ktoré môžeme charakterizovať ako určitý manuál na prežívanie v danom prostredí. V tomto momente môže dochádzať k určitému „kultúrnemu“ zámeru, ktorý môže ovplyvňovať fitness génov.

(9) **Áké sú limity chemostatového modelu?** Použitý tvar genotypu ako binárnych reťazcov a ich zobrazenie na fenotyp pomocou foldingu reprezentuje extrémne jednoduchý model živých organizmov, ktorý postihuje také jednoduché živé organizmy, akými sú vírusy, bakteriofágy a jednoduché prokaryotické baktérie. Zložitejšie organizmy model genotypu (a jeho zobrazenie na fenotyp) musia byť študované zložitejšími modelmi genotypu, ktoré majú premenlivú dĺžku, hierarchickú štruktúru a iné možné mutácie, ako sú použité 1-bitové mutácie. Jeden z najdôležitejších problémov súčasnej teórie Darwinovej evolúcie je vysvetlenie vzniku tzv. „**neredukovateľných zložitostí**“ [4]. Tento model nie je schopný vysvetliť tento problém ani na elementárnej úrovni. Moderné snahy v „umelom živote“ sú



sústredené do oblastí, ktoré sú potenciálne dôležité pre tvorbu všeobecnejšej teórie Darwinovej evolúcie, ktorá už v nejakej miere bude postihovať tvorbu neredukovateľných zložitostí. Intenzívne sú študované modulárne aspekty genotypu [5,30,31,49] a taktiež aj modelovanie symbiózy [29,49], oba tieto problémy už vyžadujú genotypy zložitejšieho tvaru, ako ten, ktorý bol použitý v našich simuláciách (lineárny reťazec symbolov konštantnej dĺžky).

*Podakovanie:* Táto kapitola vznikla za podpory grantovej agentúry VEGA SR v rámci grantových úloh VG1/4053/07 a VG1/0804/08.

## Literatúra

- [1] Atkins, P. W.: *Physical Chemistry*. Oxford: Oxford University Press 1998.
- [2] Axelrod, R.: *The Evolution of Cooperation*. New York: Basic Books 1984.
- [3] Banzhaf, W., Dittrich, P., Eller, B.: Topological Interactions in a Binary String System. *Physica D* **125** (1999) 85-104.
- [4] Behe, M. J.: *Darwin's black box*. New York: Simon & Schuster 1996 (existuje český preklad *Darwinova černá skříňka*, Praha: Návrat domů 2001).
- [5] Callebaut, W., Raskin-Gutman, D. (eds.): *Modularity: Understanding the development and evolution of complex natural systems*. Cambridge, MA.: MIT Press 2002.
- [6] Chomsky, N.: *Aspects of the Theory of Syntax*. Cambridge, MA: MIT Press 1965.
- [7] Darwin, C.: *Pôvod druhov*. Bratislava: Kalligram, 2006 (preložil Cedzo, R.).
- [8] Dawkins, R.: Universal Darwinism. In Bendall D. S. (ed.) *Evolution from Molecules to Men*. Cambridge University Press, London, 1982, pp. 403 -425.
- [9] Dawkins, R.: *The Blind Watchmaker*. London and New York: Penguin Books 1988 (existuje český preklad: *Slepý hodinář*. Praha: Paseka 2002).
- [10] Dawkins, R.: *The Selfish Gene*. Oxford, UK: Oxford University Press 1976 (existuje český preklad: *Sobecký gen*. Praha: Mladá Fronta 2003).
- [11] Dennett, D. C.: *Darwin's Dangerous Idea - Evolution and the Meaning of Life*. London: Penguin Press 1995.
- [12] Dittrich, P.: *Artificial Chemistries* (tutorial material). A tutorial held at ECAL'99, 13-17 September 1999, Lausanne, Switzerland.
- [13] Eigen, M.: Self organization of matter and the evolution of biological macro molecules. *Naturwissenschaften* **58** (1971) 465-523.
- [14] Eigen, M., Schuster, P.: The Hypercycles: A Principle of Natural Evolution. *Naturwissenschaften* **64** (1977) 541-565; **65** (1978) 7-41; **65** (1978) 341-369.
- [15] Fogel, D. B.: *Evolutionary Computation. Toward a New Philosophy of Machine Intelligence*. New York: The IEEE Press 1995.
- [16] Fontana, W. and Schuster, P.: Continuity in Evolution: On the Nature of Transitions. *Science* **280** (1998) 1451-1455.
- [17] Fontana, W. and Schuster P.: A computer model of evolutionary optimization. *Biophysical Chemistry* **26** (1987) 123-147.

- [18] Fontana, W., Schnabl, W., Schuster, P.: Physical aspects of evolutionary optimization and adaptation. *Physical Review A* **40** (1989) 3301-3321.
- [19] Fontana, W.: Algorithmic Chemistry. In: Langton C.G. (ed.) *Artificial Life II* Reading, MA: Addison Wesley 1992, pp. 159- 210.
- [20] Fontana, W.: Pulling Strings. *Science* **314** (2006) 1552-1553.
- [21] Gálik, D.: Univezálny darvinizmus (Môže byť darvinizmus teóriou všetkého ), Zborník *Kognitívne vedy III (CogSci 2000)*, Katedra matematiky CHTF STU, Bratislava (prednáška je dostupná na internetovskej adrese [http://math.chtf.stuba.sk/CogSci\\_2000.html](http://math.chtf.stuba.sk/CogSci_2000.html)).
- [22] Gálik, D.: Univerzálna teória evolúcie a problém mémov. In Nosek, J. (ed.), *Memy ve vědě a filosofii*. Praha: Filosofia 2004, pp. 133-146.
- [23] Gillespie, D. T.: Exact Stochastic Simulation of Coupled Chemical Reactions. *J. Phys. Chem.* **81** (1977) 2340- 2361.
- [24] Hardin, G.: The Tragedy of the Commons. *Science* **162** (1968) 1243-1248.
- [25] Holland, J. H.: *Adaptation in Natural and Artificial Systems*. Ann Arbor: University of Michigan Press 1975.
- [26] Jacob, F.: *The Possible & The Actual*. New York: Pantheon Books 1982 (existuje český preklad: *Hra s možnosťami*. Praha: Karolinum 1999)
- [27] Kauffman, S. A.: *The Origins of Order: Self-Organization and Selection in Evolution*. New York: Oxford University Press 1993.
- [28] Kauffman, S.A., Hohnsen, S.: Coevolution to the edge of chaos: Coupled fitness landscape, poised states, and coevolutionary avalanches. *J. Theor. Biology* **149** (1991) 467-506.
- [29] Kvasnička, V.: An evolutionary model of symbiosis. In: P. Sinčák, J.Vasčák (eds.): *Quo Vadis Computational Intelligence?* Physica-Verlag, Heidelberg, 2000, p. 293-304.
- [30] Kvasnička, V.: An evolutionary simulation of modularity emergence of genotype-phenotype mappings. *Neural Network World* **5** (2001) 473- 491.
- [31] Kvasnička, V.: A modularity emergence of genotype-phenotype mappings. *Artificial Life* **8** (2002) 295-310.
- [32] Kvasnička, V., Pospíchal, J.: A study of autoreplicators and hypercycles by typogenetics. In: Kelemen, J., Sosik, P.: *Advances in Artificial Life, ECAL 2001*, LNAI 2159, Springer 2001, pp. 37-54.
- [33] Kvasnička, V., Pospíchal, J.: Autoreplicators and Hypercycles in Typogenetics, *Journal of Molecular Structure (Theochem)* **547** (2001) 119- 138.
- [34] Kvasnička, V.: Artificial Chemistry, Replicators, and Molecular Darwinian Evolution *In Silico*. In: P. Sinčák, J.Vasčák (eds.): *Machine Intelligence, Quo Vadis?* Heidelberg: Physica-Verlag 2002, pp. 403-425.
- [35] Kvasnička, V., Pospíchal, J.: Artificial Chemistry and Molecular Darwinian Evolution in silico. *Collection of Czechoslovak Chemical Communication.* **68** (2003) 139-177.
- [36] Kvasnička, V., Pospíchal, J.: Artificial Chemistry and Replicator Theory of Coevolution of Genes and Memes. *Collection of Czechoslovak Chemical Communication.* **72** (2007) 223-251.
- [37] Kvasnička, V., Pospíchal, J., Tiňo, J.: *Evolučné algoritmy*. Bratislava: Vydavateľstvo STU 2000.

- 
- [38] Kvasnička, V., Pospíchal, J.: A Multi-agent Study of Interethnic Cooperation. In *Multi-Agent Systems and Applications*, LNCS, Vol. 2086. Berlin: Springer Verlag 2001, pp. 415-435.
- [39] Maynard Smith, J.: *Evolution and the Theory of Games*. Oxford, UK: Oxford University Press 1982.
- [40] Mayr, E.: *Towards a New Philosophy of Biology*. Cambridge, MA: Harvard University Press 1988.
- [41] Medveď, M.: *Dynamické systémy*. Bratislava: Veda 1988.
- [42] Monod, J.: *Chance and Necessity. An Essay on the Natural Philosophy of Modern Biology*. New York: Vintage 2000 (existuje český preklad: *Náhoda a nutnosť*, Praha: vydav. Pavel Mervart 2008)
- [43] Newman, M. E. J., Engelhardt, R.: Effects of neutral selection on the evolution of molecular species. *Proc. Roy. Soc. London B* **265** (1998) 1333- 1338.
- [44] Paley, W.: *Natural Theology*, with an introduction and notes by Eddy, M. D. and Knight, D. M., London and Oxford: Oxford University Press 2006.
- [45] Schuster, P.: Molecular Insight into Evolution of Phenotypes. In Crutchfield, J.P. and Schuster, P. (eds.): *Evolutionary Dynamics - Exploring the Interplay of Accident, Selection, Neutrality, and Function*. Oxford: Oxford University Press 2002.
- [46] Schuster, P., Fontana, W.: Chance and Necessity in Evolution: Lessons from RNA. *Physica D* **133** (1999) 427-452.
- [47] Spiegelman, S.: An Approach to the Experimental Analysis of Precellular Evolution. *Quart. Rev. Biophysics* **4** (1971) 213-253.
- [48] Tomáška, Ľ., Nosek, J.: Molekulárna darwinovská evolúcia: In vivo, in vitro a in silico. Zborník *Kognitívne vedy III (CogSci 2000)*, Katedra matematiky CHTF STU, Bratislava (prednáška je dostupná na internetovskej adrese [http://math.chtf.stuba.sk/CogSci\\_2000.htm](http://math.chtf.stuba.sk/CogSci_2000.htm)).
- [49] Watson, A. R.: *Compositional Evolution. The Impact of Sex, Symbiosis, and Modularity on the Gradualist Framework of Evolution*. Cambridge, MA: MIT Press 2006.
- [50] Waterman, M. S.: *Introduction to Computational Biology. Maps, sequences and genomes*. New York: Chapman & Hall / CRC, 1995.
- [51] Wright, S.: The Roles of Mutation, Inbreeding, Crossbreeding, and Selection in Evolution. *Proceedings of The Six International Congress of Genetics*, Vol. I, 1932, pp. 356-366. (Táto klasická publikácia je dostupná na <http://www.blackwellpublishing.com/ridley/classictexts/wright.pdf>).



# Simulace v neurovědách, příklad modelu prostorového slyšení

Petr MARŠÁLEK<sup>1</sup>

**Abstrakt.** Pomocí výpočetních modelů můžeme zkoumat vybrané mozkové funkce. Funkční celky v mozku slouží vnímání, zpracování informace a motorice. V našem příkladu diskutujeme prostorové slyšení. Schopnost slyšení určit umístění zdroje zvuku pomocí dvou uší je dána fyzikálně. Popisujeme nervový obvod, který prostorové slyšení umožňuje. Máme dva alternativní modely prostorového slyšení pro nízké frekvence zvuku: jeden je založen na zpoždovací lince a druhý na pravděpodobnostním zpoždění. Oddíl diskuse má formu dialogu experimentátora a teoretika.

## 1 Úvod

Tento úvodní oddíl má dvě části, které odpovídají dvěma cílům, které sledujeme. První cíl odpovídá metodám a druhý cíl výsledkům získaným modelováním neuronů. V první části úvodu popíšeme, jak funguje prostorové slyšení z fyzikálního hlediska. V dalších oddílech pak navážeme a ukážeme, jak můžeme studovat prostorové slyšení pomocí experimentů. V druhé části úvodu popíšeme základní stavbu a funkci sluchové dráhy a její začlenění do nervového systému a na to v dalších oddílech navazuje výklad, jak pomocí modelů a simulace studujeme nervový systém. Další dva oddíly popisují dva možné algoritmy, jimiž nervové obvody mohou určovat směr, ze kterého přichází zvuk. Další oddíl ukončuje výklad rozdílů mezi oběma algoritmy tím, co musí být oběma algoritmům společné. Toto je nakonec uzavřeno diskusí, vedenou formou dialogu mezi experimentátorem a teoretikem. Pro členy komunity kolem umělé inteligence by měly být bližší části textu, které pojednávají o formalismu modelování, pro kognitivní vědy jsou určeny části popisující sluchové mechanismy.

---

<sup>1</sup> Ústav patologické fyziologie 1. lékařské fakulty UK, U nemocnice 5, CZ-128 53, Praha 2, Fakulta biomedicínského inženýrství ČVUT, nám. Sítná 3105, CZ-272 01 Kladno, E-mail: Petr.Marsalek@lfl.cuni.cz

## 1.1 Fyzikální základy prostorového slyšení

Vyjdeme z úlohy, kde na základě subjektivního vjemu máme určit polohu zdroje zvuku. Abychom mohli udávat tuto polohu, zvolme souřadný systém. Je přirozené umístit počátek souřadnic tam, kde je střed hlavy. Polohu zdroje zvuku udáme ve třech prostorových rozměrech  $r$ ,  $e$ ,  $a$ ; (1) jako vzdálenost  $r$  zdroje zvuku od subjektu, (2) jako elevace  $e$  zdroje zvuku, neboli orientovaný úhel, který svírá zdroj zvuku s vodorovnou rovinou proloženou hlavou, a konečně (3) jako azimut  $a$  zdroje zvuku, neboli orientovaný úhel, který svírá zdroj zvuku s rovinou symetrie obou uší.

Manipulací s parametry zvuku v experimentech je možné určit, které charakteristiky prostorového zvuku odpovídají těmto subjektivním rozměrům umístění zdroje zvuku.

(1) K určení vzdálenosti  $r$  zdroje zvuku přispívá to, že vysoké frekvence jsou při vedení zvuku vzduchem více tlumeny, než frekvence nízké. Čím delší dráhu zvuk ve vzduchu urazí, tím má nižší podíl vysokých frekvencí. Při subjektivním odhadu vzdálenosti zdroje zvuku hraje roli spektrální složení daného zvuku a také odhad očekávané hlasitosti zvuku.

(2) Při určení elevace  $e$  zvuku se uplatní tvar vnějšího ucha. Vnější ucho, složené ze zevního zvukovodu a ušního boltce funguje jako rezonanční prostor, jako filtr, který selektivně tlumí určité frekvence vzhledem ke směru, ze kterého zvuk přichází, a tím pomáhá určit jak elevaci, tak azimut zdroje zvuku. To je směrová charakteristika vnějšího ucha. U některých živočichů napomáhá směrovému slyšení pohyb ušního boltce vzhledem k hlavě, u člověka tato schopnost zanikla jako atavismus. Člověk i ostatní živočichové ale určují směr zvuku také pohybem celé hlavy, k tomu stačí i poslech jen jedním uchem, pokud zvuk trvá dostatečně dlouho. Při subjektivním odhadu umístění zdroje zvuku, zda je nahoře anebo dole vzhledem k vodorovné rovině vedené hlavou, ale také při odhadu orientace azimutální, zda je vepředu anebo vzadu vzhledem k rovině frontální, opět hraje roli spektrální složení daného zvuku.

(3) Pro určení azimutu  $a$  zvuku je využito opoždění zvuku a zeslabení intenzity zvuku na straně protilehlé ke straně, odkud zvuk přichází. Zvuk ze zdroje, který není umístěn v rovině symetrie obou uší, dorazí k protilehlému uchu později. Pro právě ještě rozeznatelnou odchylku směru, což je u lidí v okolí roviny symetrie obou uší kolem  $4^\circ$  (úhlové stupně), je toto zpoždění v řádu desítek  $\mu\text{s}$  (mikrosekund). Zejména u vyšších frekvencí je intenzita zvuku na protilehlé straně zeslabena. Čím je hlasitost nižší, tím jsou také později vybaveny akční potenciály v příslušném sluchovém nervu, tento efekt se tedy sčítá se zpožděním zvuku. Kromě toho obecně nižší intenzitě zvuku také odpovídá nižší frekvence nervových impulsů, neboli akčních potenciálů.

Charakteristikami zvuku, podle kterých subjektivně určujeme souřadnice  $r$ ,  $e$ ,  $a$  jsou tedy následující meziušní, neboli binaurální, nebo také interaurální rozdíly v parametrech zvuků vstupujících do levého a pravého ucha: IID (interaural intensity difference, interaurální intenzitní rozdíl, neboli intenzitní klíč), podílí se na určení azimutu  $a$ , dále ITD (interaural time delay, interaurální časové zpoždění, neboli časový klíč). ITD se podílí také na určení azimutu  $a$ . Protože časové zpoždění je detekováno jako rozdíl ve fázi zvuku (jedna perioda zvuku má různé trvání podle frekvence), alternativně se ITD může určovat jako IPD, interaural phase difference, fázový klíč.

My si vybereme ITD. Nakonec sem patří ISD (interaural spectral difference, interaurální rozdíl ve spektrech, neboli spektrální klíč), ten se podílí na určení elevace  $e$  a vzdálenosti zvuku  $r$ . ITD se uplatňuje při nižších frekvencích a IID se uplatňuje při vyšších frekvencích. Nejvíce nás zde zajímá mechanismus určování azimutu  $a$ , soustředíme se proto především na tento rozměr.

Pokud zkoumáme smyslové vnímání pomocí subjektivní výpovědi a tato výpověď je reakcí na fyzikálně definovanou stimulaci, takovým experimentem překračujeme bariéru mezi subjektivním a objektivním. Ta část smyslové fyziologie, která se zabývá experimenty se subjektivní odpovědí na objektivní stimuly, se nazývá psychofyzika. V českém a slovenském písemnictví nenajdeme bohužel mnoho knih, které by čtenáře současně uvedly do otázek psychofyziky a zároveň ho poučily o biologických základech prostorového slyšení a o jeho experimentálním studiu. Uvedme zde pramen Syku [24] s plným vědomím, že tento text byl vydán před téměř třiceti lety. O žádné podobné novější učebnici k této tématice v češtině ani ve slovenštině nevím. Možná přesnější název tohoto prvního oddílu Úvodu by mohl být: **Psychofyzikální** základy prostorového slyšení.

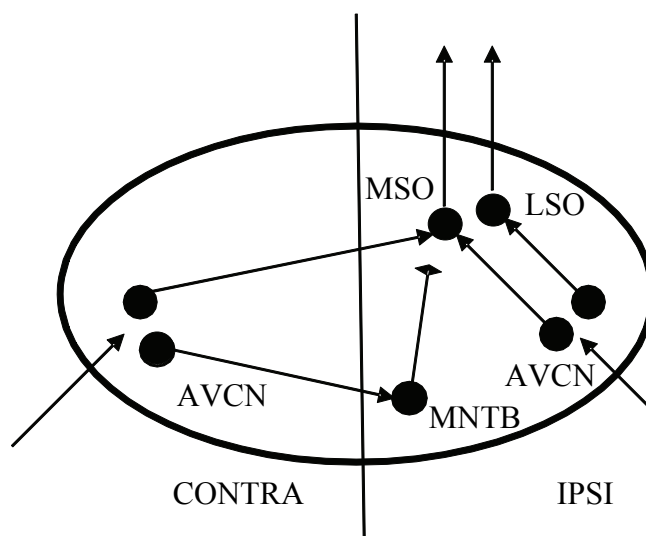
## 1.2 Základy modelování neuronů

Zopakujme stručně pojmy, používané pro popis biologických neuronů. Základní stavební a funkční jednotkou je neuron, tento neuron je též základní jednotkou, kterou modelujeme. Neurony si předávají signály pomocí impulsů, akčních potenciálů. Předání akčního potenciálu je podobné přenosu digitální jednotky v počítači. Je dobrým přiblížením si představit, že všechny akční potenciály mají stejné trvání a amplitudu. Akční potenciály se předávají přes spoje mezi neurony, nazývané synapse. V nervové tkáni savců se rozlišuje šedá hmota a bílá hmota. Šedá hmota je nahloučení neuronů, to se také nazývá (nervové) jádro. Bílá hmota jsou svazky vláken a podpůrné (gliové) tkáně, spojující jednotlivá nervová jádra, to se také nazývá (nervová) dráha. Dráhy a jádra mohou být motorické, smyslové a asociační, do toho spadá i sluchová dráha.

Neuronové mechanismy prostorového slyšení modelujeme pomocí matematických počítačových modelů. Hlavní pozornost věnujeme prvním meziušním, latinsky binaurálním neuronům ve sluchové dráze. To jsou takové neurony, v nichž se scházejí vstupy z levého a z pravého ucha, řecky by se nazývaly stereofonní, zažitý je název latinský.

Protože dále v textu občas použijeme názvy jednotlivých částí sluchové dráhy, stručně ji popíšeme. Vlásokové buňky ve vnitřním uchu převádějí zvukovou stimulaci, tedy mechanickou energii na energii elektrickou, na (postsynaptické) receptorové potenciály, které jsou dále zpracovány prvním neuronem sluchové dráhy ve formě akčních potenciálů. První neuron sluchové dráhy je v ganglion spirále, druhý neuron je v komplexu kochleárních jader a třetí, případně čtvrtý neuron sluchové dráhy, který je prvním neuronem pro obvod prostorového slyšení, je v olivě superior. Třetí, případně čtvrtý je proto, že některé větve této dráhy jdou přes více neuronů, aby se pak zase sbíhaly na další etáži sluchové dráhy, takže u nich číslování od receptorové buňky přestává být dobře definováno. Výstupy z komplexu olivárních jader, lokalizovaného

stejně jako kochleární jádra v mozgovém kmeni, potom konvergují do colliculus inferior a dále pokračují do corpus geniculatum mediale. Poslední (závěrečnou) etáží sluchové dráhy je mozková kůra, speciálně sluchová kůra. Mozková kůra je jedno velké nervové jádro, pokrývající celý zvrásněný povrch velkého mozku. Morfologicky se oliva superior skládá z mediální olivy superior, MSO, a laterální olivy superior, LSO, (viz Obrázek 1). Morfologickému rozdělení odpovídá i rozdělení funkční: MSO zpracovává nízké frekvence zvuku, u člověka v rozsahu zhruba od 200 Hz do 2 kHz a LSO zpracovává vyšší frekvence zvuku, u člověka v rozsahu asi od 1 kHz do 16 kHz. Protože tato dvě jádra mají odlišné zapojení vstupů, dva dále diskutované alternativní modely popisují pouze MSO, ačkoliv teorie pravděpodobnostního zpoždění půjde pravděpodobně v určité modifikaci aplikovat i na popis LSO.



**Obr. 1.** Nervová jádra dráhy prostorového slyšení. Na tomto obrázku je ukázáno schéma zapojení savčí nervové dráhy pro prostorové slyšení. Vlákná sluchového nervu vstupují do kochleárních jader, cochlear nuclei, CN, z nich je ukázáno jen antero-ventrální kochleární jádro, AVCN. Mediální a laterální oliva superior, MSO a LSO, dostávají vstupy právě z AVCN. V MSO se stýkají excitační axony ze stejnostranného (IPSI-laterálního) a z druhostranného (CONTRA-laterálního) AVCN. Kontralaterální vstup do LSO u savců zase prochází změnou polarity: excitační axony z AVCN se přepojují v MNTB (mediální jádro corpus trapezoideum, medial nucleus of trapezoid body) a z MNTB vedou do LSO inhibiční axony. IPSI-laterální excitační axony vedou přímo z AVCN do LSO.

Na základě elektrofyziologických akustických pokusů na zvířatech byly určeny binaurální parametry zvuku, které jsou zpracovávány v MSO a v LSO. Jsou to tyto dva binaurální klíče: první je interaurální časové zpoždění, ITD, které se u nižších frekvencí projeví také jako binaurální rozdíl ve fázi, IPD a je zpracováno v MSO a druhý klíč je interaurální rozdíl v intenzitách zvuku, IID.

Nejkratší detekovaný časový rozdíl ITD u člověka je v desítkách mikrosekund,



trvá nejméně o dva řády (stokrát) kratší dobu, než je trvání jednoho akčního potenciálu. Při prvním pohledu se zdá, že trvání akčního potenciálu je určitou dále nedělitelnou, atomickou časovou jednotkou vnímání a nervové činnosti obecně. Akční potenciál trvá 2 milisekundy (ms). K tomu na okraj dodejme, že časové úseky, které jsme schopni zaznamenat vědomě, mají trvání ještě o řád delší, jejich trvání začíná s časovým úsekem o délce 50 ms. K měření reakční doby jsou potřeba především přesné přístroje pro měření času. Na začátku 20. století experimentální psychologie především v USA začala rychle děje lidského vnímání systematicky studovat. Schopnost prostorového slyšení detekovat mikrosekundové časové rozdíly byla zajímavým problémem, který naznačoval určité paradoxy v porozumění sluchu.

## 2 Teorie zpoždovací linky

Před více než 60 lety profesor psychologické akustiky Lloyd A. Jeffress [5] formuloval teorii zpoždovací linky, která tuto schopnost prostorového slyšení detekovat mikrosekundové časové rozdíly vysvětlovala na základě detekce časové koincidence ve zpoždovací lince z nervových vláken. Jeffress formuloval svou teorii o zpoždovací lince dávno předtím, než byla známa dostatečně detailní anatomie a fyziologie nervových okruhů pro prostorové slyšení. Elektrofyziologické záznamy se v té době právě začínaly nahrávat. Tak jemná preparace, která by zaznamenala aktivitu v dráze prostorového slyšení, nebyla ještě desítky let poté k dispozici. Jeffressova teorie vysvětluje, jak by mohlo být binaurální časové zpoždění, ITD, využito prostorovým slyšením po překódování do posloupnosti akčních potenciálů, jejichž další zpracování ve sluchové dráze již nemá takové nároky na časovou přesnost. Jeffressova teorie je založena na třech předpokladech:

(1) Monoaurální spoje, neboli sekundární vlákna, která vedou z levého a pravého kochleárního jádra, se napojují na binaurální jádro, ze kterého vedou dále terciární vlákna, a časové zpoždění jejich akčních potenciálů je totožné s časovým zpožděním akustických vstupů.

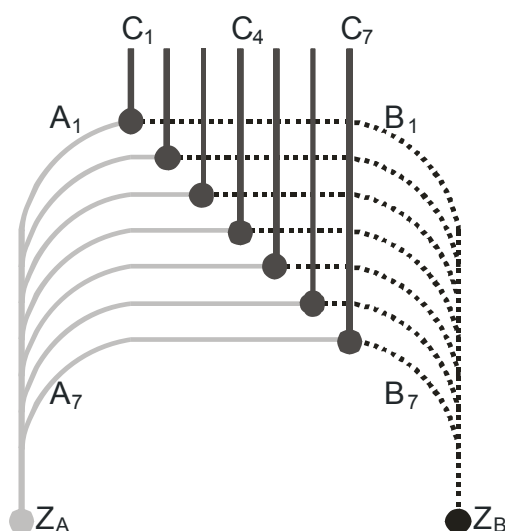
(2) Neurony binaurálního jádra jsou aktivovány pouze tehdy, pokud detekují koincidence z monoaurálních spojů z obou stran.

(3) Binaurální jádro je anatomicky zapojeno tak, že monoaurální vlákna z jedné strany se postupně prodlužují a monoaurální vlákna z druhé strany se postupně zkracují, takže v celém binaurálním jádře je aktivována pouze ta malá skupina neuronů, kde časový rozdíl ve vedení mezi jedním a druhým vstupem odpovídá ITD.

Schéma zapojení zpoždovací linky a detektoru koincidence je na Obr. 2. Časový rozdíl v axonálním vedení by tedy podle této teorie mohl být detekován zpoždovací linkou, která převádí signál zakódovaný časovým zpožděním na signál zakódovaný místem detekce v binaurálním jádře. Průběhy odpovědi binaurálních neuronů zpoždovací linky jsou ilustrovány na Obr. 4.

V experimentálním výzkumu byly potom postupně katalogizovány odpovědi neuronů v nižších etážích sluchové dráhy, a přitom byla také hledána zmíněná zpoždovací linka. U ptáků bylo morfologické i funkční zapojení zpoždovací linky skutečně popsáno [2]. U savců naopak morfologicky patrná zpoždovací linka nebyla

nalezena a některé novější experimenty [1, 12] ukazují, že příslušné neurony mohou zpracovávat prostorovou informaci na základě jiného zapojení neuronů, než do zpožďovací linky. Odpovídající teorie a modely jsou postupně vytvářeny [4]. V současné době je prostorové slyšení u savců vykládáno buďto pomocí teorie zpožďovací linky [6], anebo pomocí teorie pravděpodobnostního zpoždění [4]. Otázka jak je tomu u savců ovšem není uzavřena a proto nejsou další možné teorie a výklady vyloučeny.



**Obr. 2.** Schéma k teorii zpožďovací linky podle Jeffresse [5]. Na tomto obrázku je ukázáno, jak zpožďovací linka převádí kód daný časováním akčních potenciálů do kódu označené linky.  $Z_A$  značí zvukový signál z levého ucha a  $Z_B$  signál z ucha pravého.  $A_1-7$ , respektive  $B_1-7$ , jsou vstupy do detektorů koincidence. Výstupy z detektorů koincidence jsou označeny jako  $C_1-7$ .

### 3 Teorie pravděpodobnostního zpoždění

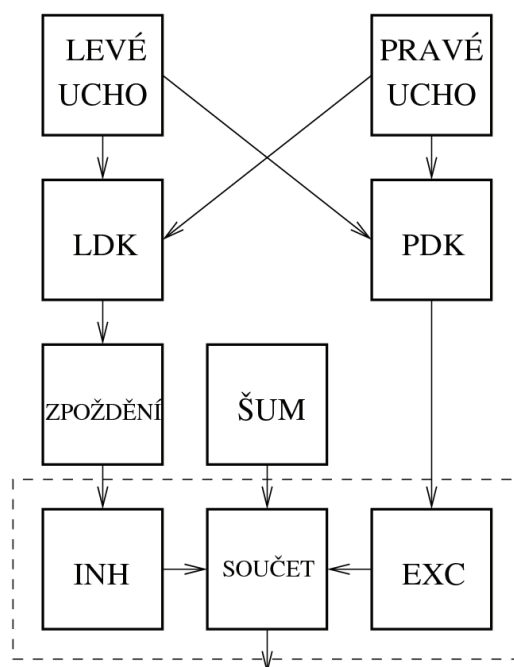
Alternativní způsob určení ITD bude také používat detekci koincidence, maximální odpověď binaurálních neuronů bude posunutá vzhledem k detekovanému fázovému rozdílu, jak to odpovídá novějším experimentům [1, 4, 12, 15]. Navrhli jsme postupně několik biologicky přípustných algoritmů zpracování binaurálních vstupů v jádrech MSO a LSO, které také zahrnují detekci koincidence. Podle toho, zda se na detektorech koincidence uplatňují jen excitační, anebo i inhibiční synapse, budeme rozlišovat excitační detekci koincidence, ECD a inhibiční detekci koincidence, ICD.

Popíšme současnou verzi našeho modelu. Model je koncipován tak, aby jednotlivá stadia zpracování signálu v modelu odpovídala stádiím zpracování signálu ve sluchové dráze, v její části předcházející a následující po MSO. Počet operací s akčními potenciály, jejich návaznost a pořadí jsou v modelu zjednodušeny, ale mechanismus zpracování informace je zachován tak, aby odpovídal biologickým možnostem sluchové dráhy od výstupu z vláskové buňky přes neuron MSO kódující

svou odpovědí azimut. Schéma modelu je zachyceno na Obr. 3.

Pro nízké frekvence (zhruba do 1,5 kHz) je vzruch na výstupu z vláskových buněk přesně sesynchronizován s fází zvuku vstupujícího do ucha. Předpokládáme nyní, že šířící se vzruch z vláskové buňky je náhodně zpožděn na další synapsích. Toto vnitřní náhodné zpoždění může být sluchovou drahou využito pro rozšíření pásma zpoždění ITD detekovaných jedním neuronem. Je to ovšem za předpokladu, že informace o přesném časování fáze, anebo náběžné hrany zvuku je zachována (a taková informace je vskutku zachována v populaci neuronů sluchové dráhy).

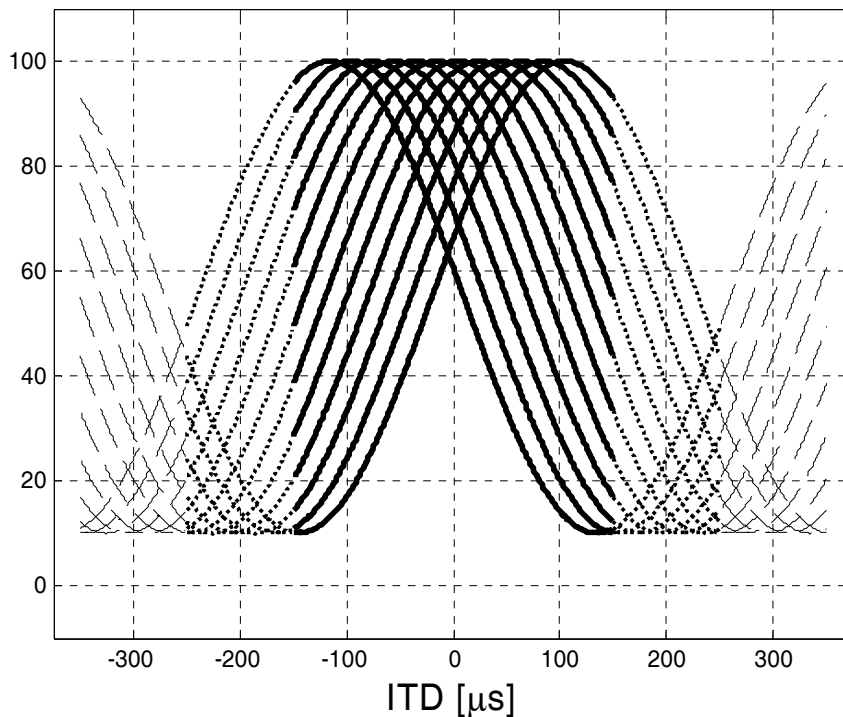
Náhodné přenosové zpoždění modelujeme v našem modelu pomocí rozdělení pravděpodobnosti času, kdy přichází akční potenciál, jedno z možných rozdělení je beta rozdělení [17]. Není to však podmínkou, obdobného chování modelu lze dosáhnout i s normálním či rovnoměrným rozdělením zpoždění. Pro jednoduchost operujeme s pojmy levá a pravá strana, model je však symetrický a tak jsou tyto pojmy záměnné.



**Obr. 3.** Schéma modelu k teorii pravděpodobnostního zpoždění. Na tomto obrázku je náhodně posunutý vstup z levé a pravé strany veden do levých a pravých detektorů koincidence (LDK a PDK). Tento vstup označený jako rámečky LEVÉ a PRAVÉ UCHO reprezentuje nejen vnější až vnitřní ucho, ale i další navazující části sluchové dráhy až po kochleární jádra. Výstup z LDK je ZPOŽDĚNÍ a veden inhibiční větví (INH). Celkový výstup z modelu je SOUČTEM spontánní aktivity (ŠUM) s výstupy z inhibiční a excitační větve (EXC).

Náhodně zpožděné signály z levého a pravého ucha se setkají na detektoru koincidence (LDK a PDK na obrázku 3). Akční potenciál se na výstupu detektorů koincidence objeví pouze tehdy, setkají-li se signály z levého a z pravého ucha

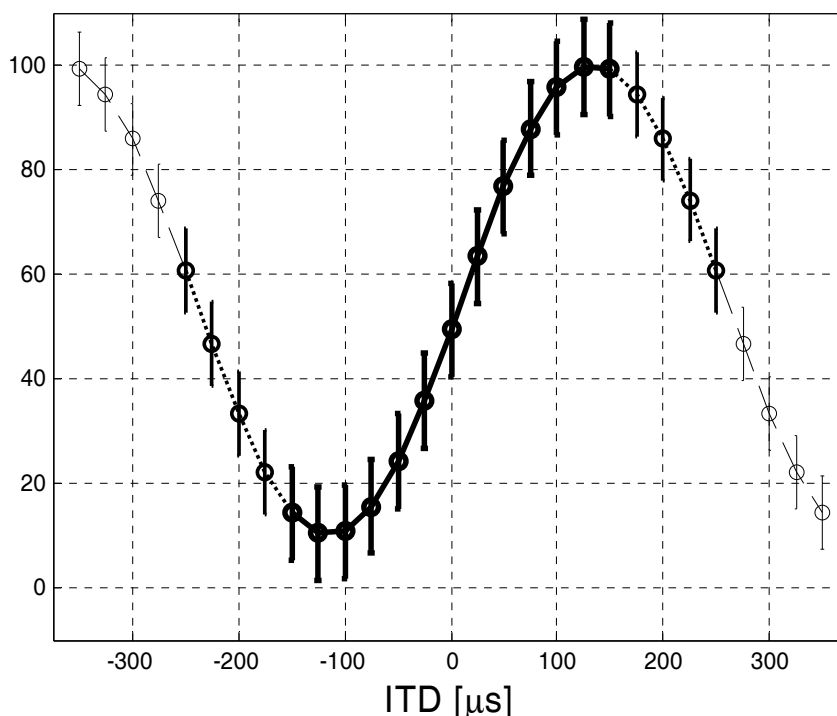
v krátkém časovém okně. Nechť například akční potenciály z levé strany jsou zpožděny. Toto zpoždění je ve sluchové dráze realizováno již před vstupem do LDK v mediálním a laterálním jádře trapézovitého tělesa (MNTB a LNTB), současně je signál invertován, dochází ke změně z vedení excitačního na inhibiční. Abychom zjednodušili implementaci modelu, předpokládejme dále, že nezáleží na tom, kde dojde ke změně z excitace na inhibici nebo kdy dojde ke zpoždění, proto je tento krok v modelu realizován až za detektorem koincidence. (Tím se bez újmy na obecnosti našeho řešení odchylujeme od anatomické posloupnosti operací na jednotlivých neuronech). Akční potenciály jsou v modelu vedeny z LDK na inhibiční synapsi v MSO (INH na obrázku 3). Akční potenciály z PDK jsou přes excitační synapsi (EXC) sečteny s výstupy (respektive odečteny od výstupů) z INH a se spontánní aktivitou (ŠUM) na neuronu SOUČET.



**Obr. 4.** Maximální odpovědi jednotlivých binaurálních neuronů ve zpožďovací lince. Na tomto obrázku jsou schématicky znázorněny aktivity jednotlivých binaurálních neuronů, vynesené v procentech na ose  $y$  v závislosti na ITD, které je na ose  $x$ . Tyto aktivity odpovídají různým zpožděním podle jejich polohy ve zpožďovací lince. Maximální odpověď každého z neuronů je v místě zpoždění, na které je naladěn. Tlustou plnou čarou je vytažena oblast relevantních ITD pro různé azimuty. Délka tohoto intervalu závisí na šířce hlavy, čím je větší vzdálenost uší, tím je delší tento interval. Tečkovanou tenkou čarou je vytažena oblast, která by ještě případně mohla být využita při kódování azimutu a čárkovaná nejtenčí čára nám připomíná, že odpovědi neuronů jsou periodickou funkcí. Dvanáct křivek znázorňuje dvanáct neuronů, které jsou naladěny na různá ITD podél zpožďovací linky.

Poloha zdroje zvuku je tímto překódována do frekvenčního kódu, to znamená, že nízká frekvence akčních potenciálů ukazuje na vlevo umístěný zdroj zvuku, vysoká frekvence naopak na zvuk vycházející zprava. Výsledná aktivita neuronu SOUČET tedy odpovídá azimutu a je schématicky ukázána na Obr. 5. Výše popsaným způsobem model využívá i inhibiční synapse, což také lépe koresponduje s experimentálními pozorováními [4, 12], než Jeffressova teorie.

Inhibiční detekce koincidence se nejspíše uplatní i ve vysokofrekvenční části obvodu prostorového slyšení, v LSO. V práci [10] algoritmus ICD srovnáváme s původní teorií o neuronovém algoritmu v LSO, kterým je odečítání frekvencí akčních potenciálů. Srovnáváme-li rychlost zpracování signálu v těchto algoritmech, ukazuje se, že námi navržený algoritmus detekuje směr zvuku rychleji.



**Obr. 5.** Maximální odpovědi všech binaurálních neuronů jedné strany podle algoritmu s pravděpodobnostním zpožděním. Na tomto obrázku je ukázána aktivita v procentech vynesena na ose y v závislosti na ITD, které je na ose x, stejně jako na předchozím obrázku. V tomto případě je ale souvislý interval ITD kódován populací binaurálních neuronů na základě jednoho pravděpodobnostního zpoždění. Odpověď neuronů nemá maximum kolem nulového zpoždění, ale kolem nulových hodnot ITD má největší derivaci (bod pro  $\text{ITD} = 0$  s aktivitou 50%), což odpovídá experimentálním pozorováním u savců [12]. Tlustá plná čára, tečkovaná tenká čára a nejtenčí čárkovaná čára vyznačují intervaly se stejným významem jako na předchozím obrázku. Sloupce s rozptylem aktivit mají ilustrovat, že odečtení zpoždění je pravděpodobnostní, nikoli deterministické.

## 4 Časová přesnost ve sluchové dráze

### 4.1 Oba algoritmy musí mít velkou, mikrosekundovou časovou přesnost

Pro další ilustraci rozměrů a časů, které diskutujeme, uveďme dva jednoduché příklady.

(1) Příklad, proč je třeba určit časové zpoždění s mikrosekundovou přesností. V psychofyzikálních experimentech bylo ukázáno, že v okolí roviny symetrie hlavy, tedy okolo azimutu  $0^\circ$  je práh rozlišení, neboli diskriminační práh  $4^\circ$  [13]. V dalším odstavci vypočítáme jako úlohu o pohybu podle učebnice elementární fyziky časové zpoždění  $t$  (ITD), které odpovídá úhlu  $4^\circ$  (tento odstavec možno přeskočit).

Šířka hlavy u člověka je 15-18 cm. (Aby výpočet hezky vyšel, uvažme vzdálenost mezi ušima menší, než stanovuje akustická norma.) Položme ji rovnu  $d = 15$  cm. Vzorec pro čas  $t$  v úloze o pohybu  $t = s/v$  ( $s$  je dráha,  $v$  rychlost) použijeme pro popis pohybu zvuku. Pro úhel  $\alpha = 4^\circ$  bude rozdíl dráhy zvuku mezi přilehlým a odvráceným uchem  $s = d \cdot \sin \alpha = 15 \cdot 0,07 = 1$  cm. Rychlost zvuku ve vzduchu je  $v = 340$  m/s = 34.000 cm/s. Představme si, že zdroj zvuku je v nekonečnu, aby cesty zvuku do pravého a levého ucha byly rovnoběžné. Pomůže také nakreslit si obrázek, lze ho nalézt v [24]. Z rozdílu vzdáleností od zdroje zvuku (pro  $\alpha = 4^\circ$ ) mezi pravým a levým uchem, to jest  $s = 1$  cm, spočteme čas, za který zvuk urazí tuto vzdálenost. Tento čas je přibližně (v sekundách)  $s/v = 1/34.000 = 0,000\ 03 = 30$   $\mu$ s (mikrosekund).

Tak se dostaneme k mikrosekundové přesnosti. Bylo demonstrováno, že různé neuronové obvody jsou schopny detekovat i kratší časové intervaly, přestože cyklus nejkratších savčích akčních potenciálů trvá déle, než 2 ms (milisekundy).

(2) Příklad, jaké rozměry musí mít zpožďovací linka. Rozměry hypotetické zpožďovací linky dle Jeffresse by potom byly následující: rozmezí azimutů  $0-90^\circ$  odpovídají zpoždění přibližně  $0-500$   $\mu$ s (= 0,5 ms). Při rychlosti vedení axony a dendrity 1 m/s urazí akční potenciál 0,5 mm za 0,5 ms, takže zpožďovací linka by měla mít asi tuto délku. Délka zpožďovací linky u sovy pálené je srovnatelná s výsledkem tohoto odhadu [2].

### 4.2 Který ze dvou neuronových algoritmů používají savci?

Věc, kterou nové experimenty nezpochybnily, je nutnost zapojení detektorů koincidence v obvodu, který zpracovává binaurální vstupy. Bez detektoru koincidence není možné signál obsahující ITD zpracovat. Mechanismus detekce koincidence může být implementován například pomocí časového okna, jehož délka trvání v čase je srovnatelná s délkou detekovaného ITD. Funkce neuronů jako detektorů koincidence je důležitá v obou teoriích.

Také jsme se zabývali otázkou, jaký algoritmus zpracování ITD může být implementován na prvních neuronech, které zpracovávají binaurální informaci, tedy na neuronech v LSO a v MSO, aby byl v souladu se současnými experimentálními výsledky. Nejvíce názorným algoritmem je sčítání časových oken určitého tvaru, a jejich porovnání s prahem [10]. K tomuto algoritmu, který úplně neodpovídá

neuronovým operacím s akčními potenciály, jsme postupně doplnili další, více biologicky realistické modifikace.

Na základě postupně budované teorie jsme schopni předpovědět výsledky některých experimentů, které dosud nebyly uskutečněny. Různé implementace výpočtu polohy zdroje zvuku z ITD by měly různě dlouhé trvání zpracování, které by při různé stimulaci umělými podněty přispělo k různým reakčním dobám v psychofyzikálním experimentu. Pomocí tohoto experimentu by se poté dalo rozhodnout, který algoritmus je skutečně implementován v neuronech olivárního komplexu. Jinou naší předpovědí je vysvětlení přítomnosti zářezu v křivce citlivosti lokalizace zvuku v závislosti na hlavní frekvenci zvuku. Tento zářez okolo 1,5 kHz pro dané parametry odpovídá zářezu popsanému v psychofyzikálních experimentech [11, 13]. Naše vlastní modely a teorie, i teorie jiných autorů [4] podporují výklad mechanismu prostorového slyšení pomocí pravděpodobnostního zpoždění.

## 5 Diskuse

Popis neuronových detektorů koincidence ve sluchových jádrech savců jsme podali v roce 2000 [8] a v tomto textu studujeme biofyzikální mechanismy detekce koincidence krátkých časových úseků ve sluchové dráze. V roce 2001, [9], jsme diskutovali o možném neuronovém kódu, použitém v nízkofrekvenční části prostorového slyšení. Dále jsme studovali excitační a inhibiční detekci koincidence v [10, 11].

Ve výkladu jsme se zde dostali k bodu, kdy přichází ke slovu diskuse. Předmět diskuse je nasnadě – je to možnost rozlišení mezi dvěma modely. Podáme ji v duchu tradice sborníků z konferencí o *Kognici a umělém životě*, totiž formou dialogu. Tento dialog proběhl zhruba před pěti lety a ne až tak idylicky, jak je to nyní podáno. V tomto bodě ještě musím varovat laskavého čtenáře, že se přesouváme z bezpečných vod do vod nejistých, řekněme jako v bazénu z míst pro neplavce do míst pro plavce. Představme si, že elektronickou korespondencí autora (A) s oponentem (B), který oponoval jednu práci autora, proběhl následující dialog:

B: Váš ilustrační příklad umožňuje vysvětlit minimální ITD o velikosti 20  $\mu$ s. Je možné psychofyzikálně pozorovat i menší ITD? Pokud ano, jak je možné vysvětlit tuto diskrepanci?

A: Speciálně detekci ITD u člověka v okolí malých úhlů s rovinou symetrie těla je možno zpřesnit tréninkem, a uvádí se dvojnásobná přesnost, to jest kolem 10 mikrosekund. Novější prameny, například Palmer, uvádějí až pětinasobnou přesnost oproti 20  $\mu$ s, to jest kolem 4 mikrosekund. Jiné hodnoty z pokusů na savcích jsou ještě daleko více impresivní. F. Rieke s kolegy ze skupiny W. Bialka [19] shrnují pokusy Simmonse [20, 21], který uvádí dva různé pokusy, jejichž cílem je určit přesnost motorické a neuronové odpovědi u netopýra. U behaviorálního pokusu má motorická odpověď netopýra přesnost 35 mikrosekund, tedy srovnatelná s netrénovanými, neboli naivními lidskými subjekty. Ale u záznamu z určitých sluchových echolokačních neuronů u netopýra byla změřena daleko vyšší časová přesnost jejich odpovědi a byl také spočítán časový rozptyl (timing jitter). Mezi výsledky je i pokus s motorickou

odpovědi (chycení hmyzu - kořisti netopýra) a oblast odpovědi se 75 % spolehlivostí, která u netopýra odpovídá ještě daleko impresivnější časové přesnosti: kolem 10-12 nanosekund. Vysvětlení těchto přesností je takové, že daná velikost časového rozptylu se může ve vzestupných jádrech, která slouží jako přepojovací stanice, zvětšovat i zmenšovat vzhledem k předchozímu jádru, přesnost fázového závěsu v dalším jádře tedy může být i vyšší než v jádře předchozím [16]. (Toto je důležitý, a ne úplně intuitivní výsledek. Předpokladem, aby toto bylo možné, je zakódování daného času populací neuronů.)

B: V jakém stavu je v současné době experimentální potvrzení, anebo vyvrácení Jeffressovy teorie?

A: Jeffressova teorie nebyla dosud potvrzena u savců. Jeffressova teorie o existenci hypotetické zpožďovací linky byla experimentálně demonstrována v nucleus laminaris sovy pálené, viz [2]. U savců dosud zpožďovací linka nebyla popsána, a jak teoretická [7], tak experimentální evidence [15] svědčí pro to, že u savců je ITD zpracováno jinak, než zpožďovací linkou.

B: Všeobecný problém modelů v této oblasti je skutečnost, že neurony sluchového nervu generují akční potenciály trvale a nejen na začátku podnětu. Kromě toho se na generování akčních potenciálů sluchového nervu zúčastňují i stochastické mechanismy. Z toho by vyplývala potřeba souběžného zpracování vstupů na mnoha neuronech ještě před korelační analýzou (časovou koincencí), aby byly vyrušeny náhodné vlivy. Jak by takové souběžné, paralelní zpracování ovlivnilo výpočet pomocí zpožďovací linky a pomocí pravděpodobnostní detekce koincidence?

A: Toto je otázka na zpracování signálu, prováděné skupinou neuronů. Zkusím otázku přeformulovat asi takto: daný časový rozdíl (ITD) kóduje populace neuronů. Jaký výsledek získáme, když odpověď buďto (1) zprůměrujeme přes skupinu neuronů, anebo když ji počítáme (2) časovým zprůměrováním na jednom neuronu?

A: (1) Uvažování předpokladu o zprůměrování přes populaci neuronů by pravděpodobnostní mechanismus zobecnilo velmi žádaným způsobem. Naopak předpoklad časové integrace (2) na jednom neuronu z binaurálního jádra by vylučoval jeho vlastnosti, které z něj dělají detektor koincidence. Zprůměrování přes populaci neuronů je rozhodně dobrým otevřeným problémem pro další výzkum, daný směr může ukazovat například práce [16], která zajímavým způsobem rozvíjí některé starší výpočty o pravděpodobnostním rozostření (anglicky jitter, zde o něm mluvíme převážně jako o rozptylu) časového signálu. Je pravděpodobné, že mechanismus pravděpodobnostní koincidence by byl ovlivněn tak, že odhad přesnosti časování na neuronech by vedl k vyšší hodnotě přesnosti a také, že by bylo možné doplnit odhad velikosti dané populace neuronů a jeho srovnání s anatomickými daty.

B: A bylo by například také možné vysvětlit takzvaný "efekt precedence" v rámci předloženého pravděpodobnostního modelu? Jev "precedence" je ovlivnění subjektivní lokalizace zdroje zvuku u dvojice následných krátkých impulsů, kde sluchový systém preferuje první a potlačuje následné impulsy, které mohou obsahovat i binaurální rozdíly, které mohou odpovídat i směru, který je konfliktní s binaurálními rozdíly u prvního impulsu.

A: Pravděpodobnostní model efekt precedence nevysvětluje. Subjektivní zdůraznění prvního zvukového impulsu zřejmě vzniká ve vyšší etáži sluchové kůry.



Teorie diskutované zde jsou založené na tom, že neurony na nižších etážích se neadaptují a nepřizpůsobují jednotlivým po sobě jdoucím zvukovým vlnám jinou váhu z hlediska možnosti vyvolat odpověď neuronů. Toto by se u sluchové kůry projevilo například jinou počáteční částí "peristimulus histogramu", získaného několikanásobným opakováním stimulace. Toto vážené (vyšší na začátku a nižší později) vyvolání odpovědi neuronů nastává ve vyšších etážích sluchové dráhy, a speciálně i v mozkové kůře.

B: Jak byste tedy shrnul, které nové výsledky přináší model pravděpodobnostního zpoždění pro pochopení prostorového slyšení? Jak by bylo možné zařadit zde předkládané these do souvislosti se současným stavem poznání neuronových kódů?

A: Teorie o pravděpodobnostním zpoždění by měla být alternativou k Jeffressově teorii v případě savců, u nichž nebyla zpožďovací linka dosud prokázána. Teorie zpožďovací linky úlohu detektorů koincidence a synchronizace předpokládá jen v prvních binaurálních jádrech, a další zpracování informace předpokládá pomocí jiných neuronových mechanismů. Pro pravděpodobnostní teorii jsou detektory koincidence a synchronizace použitelné na více místech nervového obvodu. Pravděpodobnostní teorie možná umožní sjednotit na společném základě zpracování i v LSO i v MSO.

B: Rád bych diskutoval ještě některé více detailní otázky. Jaká kombinace iontových kanálů je specifická pro sluchovou dráhu, speciálně pro zde studované neurony? Zkuste více konkretizovat Vaše obecné tvrzení, že dráha prostorového slyšení je typická menším počtem druhů iontových kanálů.

A: U neuronů v kochleárních jádrech a v mozkovém kmeni je specifický rychlý nástup hyper-polarizace, je rychlejší, než u jiných neuronů. To je způsobeno iontovými kanály pro draslík ( $K^+$ ) s rychlou odezvou a prodlouženou hyper-polarizací, která drží na určité úrovni potenciál a tím rychle ukončí trvání časového okna pro detekci koincidence. Naopak elektrické proudy draselných iontů s pomalejším nástupem (A kanál) a některé iontové kanály ovlivnitelné hormonálně, které jsou u jiných neuronů (například u neokortikálních neuronů) zodpovědné za adaptaci, nejsou přítomné [23].

B: Jaký význam u spektrálních klíčů má interaurální rozdíl, a jakou funkci má v tomto paměť?

A: Spektrální klíče jsou binaurální rozdíly (ISD) i monoaurální charakteristiky v zesílení ucha na jednotlivých frekvencích, a kromě určení azimutu jsou také využity pro lokalizaci zdroje zvuku v dalších dvou prostorových rozměrech, (1) pro určení elevace a (2) pro určení vzdálenosti zdroje zvuku. Známý zvuk, který je srovnáván s předlohou uloženou v paměti, je lokalizován přesněji. Toto opět zdůrazňuje úlohu mozkové kůry a delšího zpracování a reverberace (to jest opakování stimulu vnitřním mechanismem v nervovém obvodu) u subjektivního odhadu elevace a vzdálenosti zdroje zvuku. Odhad elevace a vzdálenosti může být i monoaurální, takže význam interaurálního rozdílu (jak ITD, tak i IID) bych jedině spatřoval v tom, že interaurální rozdíl musí být pro tyto dva prostorové rozměry zpracován ještě nějakým jiným způsobem, který kompenzuje binaurální disproporce uvnitř neuronových okruhů, které počítají spektrální informaci.

B: Jak je možné vysvětlit, že pro vysokofrekvenční komponenty stoupá minimální ITD na 200 mikrosekund? Protože tento fakt souvisí s frekvencí zvuku,

měly by z toho být vyvozeny závěry v rámci tonotopické organizace sluchové dráhy. Je možné tento efekt vysvětlit pro oba srovnávané modely?

A: Tím jste se dotkl otázek, které jsme dosud příliš nediskutovali, to jest tonotopické organizace. (Tento pojem shrnuje pozorování, že většina jader ve sluchové dráze je vrstevnatě uspořádána podle vzrůstající frekvence zvuku. Zpoždovací linka u ptáků je díky tomuto uspořádání vlastně dvourozměrná struktura, jeden rozměr má ve zpoždění a druhý ve frekvenci zvuku.) V rámci vzrůstající frekvence zvuku se mění poměr, v jakém se na určení azimutu podílejí ITD a IID. Připomeňme, že ITD je využito pro nižší a IID pro vyšší frekvence. Zajímavým psychofyzikálním experimentem je záměna poměru ITD a IID [24]. Do pokusu se záměnou poměru ITD a IID by měl vstoupit ještě třetí parametr, a to je trvání stimulu. Zde prezentovaná teorie vysvětlení nepodává, tuto otázku jsme ale také studovali v [11]. Úvahy o čase zpracování by ovšem mohly dát předpověď výsledku zmíněného pokusu a ve svých důsledcích by mohly podpořit jednu z obou srovnávaných teorií. V jednom rozpracovaném dílčím projektu, kde řešíme různé implementační otázky kolem modelu LSO, se zdá, že je možné také tento jev vysvětlit. Naše výsledky jsou ale v tuto chvíli zcela předběžné.

B: Naprosto nesouhlasím s tvrzením, že délka trvání akčního potenciálu znemožňuje vysvětlit ITD o velikosti kratší, než 20  $\mu$ s. Protože je rozhodující přesnost nástupu akčního potenciálu, ne jeho trvání! Teď jsem zvědav, co mi na to odpovíte.

A: To že jsem říkal? Pokusím se to formulovat jinak a lépe. Takže zkusme to srovnání přesnosti ITD a trvání akčního potenciálu postavit takto: Šířka okna pro detekci koincidence musí být řádově srovnatelná s délkou ITD, tedy s 20 mikrosekundami, což je řádově stokrát kratší doba, než je trvání typického akčního potenciálu ve sluchové dráze. To je určitým paradoxem (akční potenciály jsou přinejmenším podle trvání pomalé, kdežto časy, zaznamenávané pomocí nich jsou stokrát rychlejší). Pohybujeme se ve dvou různých časových měřítcích, ačkoliv by se zdálo, že nám stačí jen to malé. Takový paradox je východiskem pro budování jakékoliv teorie prostorového slyšení. Na začátku našeho rozhovoru jsem se zmiňoval o tom, že přesnost určení azimutu je možné ještě zlepšit tréninkem. Podám zde výčet několika srovnatelných hodnot časové přesnosti z literatury. Myslím, že tyto příklady ani nepotřebují dlouhý komentář. V teoretickém modelu [3] je použita přesnost 25  $\mu$ s. Autoři experimentální *in vitro* studie [18] zaznamenávají u neuronů nucleus laminaris vysokou časovou přesnost při charakteristické frekvenci zvuku 3.2 kHz (na kuřatech, tedy na ptácích). Současné experimenty (na pískomilech, tedy na savcích) i simulace ve [23] ukazují důležitost postupného aktivování depolarizačních a hyperpolarizačních proudů, které ohraničují časové okno pro detekci koincidence v řádu 100  $\mu$ s. P. Joris s T. Yinem v experimentu [6] demonstrují vektorovou sílu více než 0.5 při experimentu, kde ILD je větší než 20 dB, a to pro všechny charakteristické frekvence až do 30 kHz a v jiném experimentu Oertelová [14] ukazuje vektorovou sílu ještě přesnější (obojí u kočky). Podobnou přesnost nacházíme i u různých dalších neuronů sluchové dráhy u různých živočichů. Máte pravdu, že záleží hlavně na náběžné hraně akčního potenciálu, a jeho další průběh tedy nazvěme refrakterní fází. Vaše námitka nám doufám pomohla najít co nejlepší formulaci. Jinak vidíte, že jsem Vám snad

dostatečně přitakal.

B: Budiž. Zkusím to ale ještě jinak. Jaká je tedy přesnost fázového závěsu akčních potenciálů například v té zmíněné práci Oertelové ze skupiny P. Jorise [14]? Ta se mi líbila, připomeňte zde prosím, o co jde, a jaká časová přesnost je studována v této práci.

A: Vidím, že tato práce nás zaujala oba. Nejspíš proto, že všechna měření a úvahy o časovém rozptylu jsou tam velmi exaktní. Oertelová se spoluautory [14] měřila průběh napětí neuronů s vzorkovací frekvencí 125 kHz, takže přesnost jednoho jejich časového vzorku měla 8 mikrosekund. S tímto časovým rozlišením změřila časový rozptyl (rozostření, spike timing jitter) o velikosti 200 mikrosekund na chobotnicových buňkách kochleárních jader kočky. Zde ještě jednou připomeňme, že podobně, jako u dalších otázek, které diskutují maximální dosaženou přesnost v detekci ITD, je důležité, že daná velikost časového rozptylu se může ve vzestupných jádrech, která slouží jako přepojovací stanice, zvětšovat i zmenšovat vzhledem k předchozímu jádru. Přesnost fázového závěsu (vyjádřená vektorovou silou, číslem od nuly do jedné) v dalším jádře tedy může být i vyšší než v jádře předchozím [16]. Přesné měření časového rozptylu v jednom daném typu buněk ve vztahu ke stimulu, ke zvuku, je důležitým experimentálním protějškem k teoretické práci Reeda, Bluma a Mitchellové [16].

B: Otázkou zůstává, jak by byla ovlivněná celková reakční doba, pokud jsou latence jednotlivých mechanismů takové, jak je uvedeno v tabulkách ve Vašich pracích z roku 2005 [10, 11]?

A: Celková reakční doba je řádově delší ve srovnání s latencemi v těchto tabulkách, nejkratší bude v řádu 100 milisekund. Tabulku uvádím hlavně proto, že by se mohla vyskytnout nějaká možnost, jak nepřímo změřit časy zpracování mezi jednotlivými jádry ve vztahu k celkové behaviorální odpovědi, jako se to povedlo z mozků makaků v práci Salzmana a kolegů [22], proto ho také zde cituji. Přímé časy vedení u kočky v mozkovém kmeni změřili Joris a spolupracovníci v sérii tří článků o LSO, zakončené [6] v roce 1998. Je tedy naděje, že by v nějakém experimentu mohly být změřeny časy, odpovídající časům v našich teoretických modelech. Na závěr naší diskuse bych Vám rád poděkoval za Vaše zajímavé otázky, snad jsem Vás úplně nenudil.

## 6 Závěr

Cílem tohoto textu bylo dát příklad výpočetního modelování mozkových funkcí. Na příkladě prostorového slyšení jsme se pokusili ukázat, že i když je k dispozici množství experimentálních elektrofyziologických i psychofyzikálních dat, není snadné vyložit, jak vlastně příslušná část sluchové dráhy funguje. Fungování neuronového obvodu pro prostorové slyšení je možné vyložit přinejmenším pomocí dvou různých modelů. Je dokázáno, že model Jeffresse se zpoždovací linkou je možné použít na výklad prostorového slyšení u ptáků. Zda bude moci teorie pravděpodobnostního zpoždění vyložit analogický neuronový obvod u savců je předmětem intenzivního výzkumu.

*Poděkování:* Podpořeno z výzkumných záměrů "Molekulární biologie a patologie buňky za normy a u vybraných klinicky závažných patologických procesů", číslo 0021620806, a "Transdisciplinární výzkum v biomedicínském inženýrství II", číslo 6840770012, od MŠMT České republiky. Kromě českých daňových poplatníků díky zaslouží zejména také Ing. M. Drápal, MUDr. A. Král, PhD., Ing. J. Maršálek a Mgr. P. Šanda.

## Literatura

- [1] Brand, A., Behrend, O., Marquardt, T., McAlpine, D., and Grothe, B.: Precise inhibition is essential for microsecond interaural time difference coding. *Nature* **417** (2002) 543–547.
- [2] Carr, C. E. and Konishi, M.: Axonal delay lines for time measurement in the owl's brainstem. *Proc Natl Acad Sci USA* **85** (1988) 8311–5.
- [3] Gerstner, W., Kempter, R., van Hemmen, J. L., and Wagner, H.: A neuronal learning rule for sub-millisecond temporal coding. *Nature* **383** (1996) 76–78.
- [4] Grothe B.: New roles for synaptic inhibition in sound localization. *Nat Rev Neurosci.* **4** (2003) 540-550.
- [5] Jeffress, L. A. A place theory of sound localization. *J. Comp. Physiol. Psychol.* **41** (1948) 35–39.
- [6] Joris, P. X. and Yin, T. C.: Envelope coding in the lateral superior olive. III. Comparison with afferent pathways. *J. Neurophysiol.* **79(1)** (1998) 253–69.
- [7] Kempter, R., Leibold, C., Wagner, H., and van Hemmen, J. L.: Formation of temporal-feature maps by axonal propagation of synaptic learning. *Proc Natl Acad Sci U S A.* **98** (7) (2001) 4166-71.
- [8] Maršálek, P.: Coincidence detection in the Hodgkin-Huxley equations. *Biosystems* **58** (1-3) (2000) 83–91.
- [9] Maršálek, P.: Neural code for sound localization at low frequencies. *Neurocomputing* **38-40** (1-4) (2001) 1443–1452.
- [10] Maršálek, P. and Kofránek, J.: Spike encoding mechanisms in the sound localization pathway. *Biosystems* **79** (2005) 191-198.
- [11] Maršálek P. and Lánský, P.: Proposed mechanisms for coincidence detection in the auditory brainstem. *Biol. Cybern.* **92** (6) (2005) 445-51.
- [12] McAlpine, D., Jiang, D., and Palmer, A. R.: A neural code for low-frequency sound localization in mammals. *Nat. Neuroscience* **4** (4) (2001) 396–401.
- [13] Mills, A. W.: Auditory Localization. In Tobias, J.V., editor, *Foundations of Modern Auditory Theory*, Academic Press, New York, 1972, pp. 303–348.
- [14] Oertel, D., Bal, R., Gardner, S. M., Smith, P. H., and Joris, P. X.: Detection of synchrony in the activity of auditory nerve fibers by octopus cells of the mammalian cochlear nucleus. *Proc Natl Acad Sci USA* **97** (2000) 11773–9.

- 
- [15] Palmer, A. R., Shackleton, T. M., Mc Alpine, D.: Neural mechanisms of binaural hearing. *Acoust. Sci and Tech.* **23 (2)** (2002) 61-68
- [16] Reed, M.C., Blum, J.B., Mitchell, C.C.: Precision of neural timing: effects of convergence and time-windowing. *J. Computational Neuroscience* **13**, (2002) 35-47.
- [17] Rektorys, K., a kolektiv: *Přehled užité matematiky, svazek 1 a 2*. Prométheus, Praha, 2000.
- [18] Reyes, A. D., Rubel, E. W., and Spain, W. J. *In-vitro* analysis of optimal stimuli for phase-locking and time-delayed modulation of firing in avian nucleus laminaris neurons. *J. Neuroscience* **16** (1996) 993–1007.
- [19] Rieke, F., Warland, D., van Steveninck, R. R., Bialek, W.: *Spikes. Exploring the neural code*. MIT Press, Cambridge, Massachusetts, USA, 1997.
- [20] Simmons, J. A.: Perception of echo phase information in bat sonar. *Science* **204** (1979) 1336-1338.
- [21] Simmons, J. A.: A view of the world through the bat's ear: The formation of acoustic images in echolocation. *Cognition* **33** (1989) 155-199.
- [22] Salzman, C. D., Murasugi, C. M., Britten, K. H., and Newsome, W. T.: Microstimulation in visual area MT: Effects on direction discrimination performance. *J. Neuroscience* **12** (1992) 2331–2355.
- [23] Svirskis, G., Dodla, R., and Rinzel, J.: Subthreshold outward currents enhance temporal integration in auditory neurons. *Biol. Cybern.* **89(5)** (2003) 333–340.
- [24] Syka, J., Voldřich, L. a Vrabec, F.: *Fyziologie a patofyziologie zraku a sluchu*. Avicenum, Praha 1981.



# Humanoidní robotika včera, dnes a zítra ...

Pavel NAHODIL<sup>1</sup>

**Abstrakt.** Humanoidní robotika navrhuje roboty podobné člověku principiální stavbou těla a zejména způsobem pohybu. Jedním z cílů je úspěšná implementace komplexních chování různého typu a různé složitosti. Přesně to ale již vyřešila příroda u živočichů, od jejich reflexivních a instinktivních chování až po složité kombinování, abstrakci a anticipaci u člověka. Při návrhu se tak dnes opíráme opět i o vědy jako jsou biologie, etologie, ekonomie apod. Další důležitou vlastností humanoida by měla být schopnost adaptace a kontinuální optimalizace chování, což po učení a reprodukci zřejmě představuje tu nejdůležitější vlastnost živé hmoty a života vůbec.

## 1 Úvod

Éra robotů právě začíná a není daleko doba, kdy budou běžnou součástí našeho života. Vědci postupovali při vývoji umělého života dosud prakticky shodně s evolucí života na Zemi, od hmyzích robotů přes kybernetické čtyřnožce až po humanoidní typy robotů, které věrně napodobují lidský vzhled i chování. Současným cílem je vývoj humanoidních robotů, kteří budou schopni samostatného myšlení, učení a rozhodování. Ti se musí učit novým činnostem především pouhým »okoukáním« příslušných pohybů od lidí, vyjadřovat emoce a plnit příkazy vydávané jednoduše hlasem svých pánů a stvořitelů. Mnozí dnešní roboti už rozeznávají jak hlasy, tak obličej různých lidí, mají hloubavou povahu, a pokud nemají nic na práci, sami si doplňují »vzdělání« studiem informací. Snem výzkumníků je vytvoření molekulárního počítače, který by byl jakýmsi robotovým srdcem i mozkiem. Měl by být až milionkrát rychlejší než současné počítače a navíc už bude disponovat skutečnou umělou inteligencí, která jej předurčuje jako opravdový umělý mozek pro další generace robotů. Pracují na tom přední vědecké týmy celého světa, MIT počínaje. Kontakt s lidmi humanoidům usnadní přirozený vzhled jejich obličejů s celou škálou mimických pohybů. Lidé raději komunikují s robotem, který má určitou osobnost. Robot s lidskou tváří a chováním bude jednou pro člověka daleko přijatelnější než obyčejný mechanický pomocník.

---

<sup>1</sup> Katedra kybernetiky – CAK (Centrum aplikované kybernetiky), Elektrotechnická fakulta ČVUT v Praze, 122 35 Praha 2, Karlovo nám. 13, E-mail: nahodil@fel.cvut.cz

## 2 Od začátků robotizace k humanoidům

### 2.1 Vývoj robotů – historický přehled

Myšlenka automatů, které mohou provádět nejrůznější úkoly, pochází ze starověkého Egypta. V historii se opakují pokusy o umělé konstrukce člověku se podobajících zařízení, zejména na mechanickém principu. Tento princip byl ale Karlu Čapkovi cizí; jeho „roboti“ ve hře RUR byli koncipováni pouze na biochemické bázi.

První velmi jednoduché průmyslové roboty – spíše z dnešního pohledu manipulátory se objevily začátkem 50. let, v době studené války, kdy byl svět stravován hrozbou atomového nebezpečí. V roce 1958 *UNIMATION* vytvořil první průmyslově využitelný manipulátor s jedním robotickým ramenem. Počáteční nedůvěra byla velká. Teprve tři roky poté, v roce 1961, bylo několik prvních *UNIMATION 1900* použito úspěšně v General Motors pro lití pod tlakem. Od roku 1962 začala dodávat na trh své roboty konkurenční firma *VERSATRAN*. Firma *ASEA* navrhla prvního robota již se třemi stupni volnosti pohybu. Roboty užívané v 60. letech byly však zcela neautonomní jednotky, umístěné v pevných lokalitách do pevných pozic. Byly poměrně pomalé, nejvýše několik pohybů za minutu. Působily však už tehdy imponantně, i když mohly provádět pouze jednoduché úkoly, jako přemísťovat nebezpečné sloučeniny z jednoho místa na druhé. O trochu později přišel *Barrett Electronics* s bezdrátově řízeným elektrickým nákupním vozíkem pro obchodní dům se smíšeným zbožím. Ten byl navigován signály, vysílanými několika majáčky na příslušném podlaží. Teprve s objevem mikročipu v polovině sedmdesátých let se počítače začaly natolik zmenšovat a staly se tak všestrannými a levnými, že bylo možné začít je používat k řízení strojů v automatizované průmyslové výrobě. Objevili se první *průmyslové roboty*. V 70. letech se začalo již s osazováním výrobních linek roboty, nejprve svařovacími, montážními, tvářecími, stříkacími apod. [6]

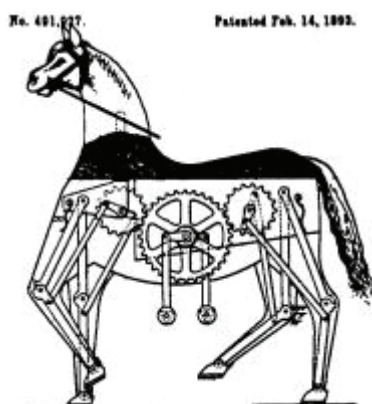
Hnacím motorem pro rozvoj robotiky a zavádění robotů do denní praxe byla a je touha po zlepšování životní úrovně, úsilí o rozvoj výroby, růst produktivity práce, sociální požadavky (hygiena, humanizace práce, monotónní a nebezpečná práce), rozšiřování sortimentu, zvyšování kvality výroby a mnoho dalších motivací [10].

### 2.2 Historie mobilních kráčivých robotů

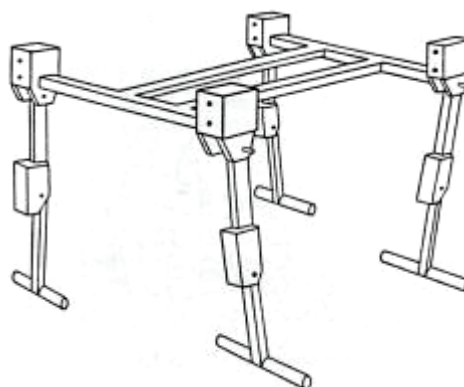
Zmiňme se krátce o některých předchůdcích robotů humanoidního charakteru. Důležitou kategorií robotů tvoří tzv. kráčivé roboty. První, historicky doložené mechanické kráčivé zařízení bylo navrženo, postaveno a velmi krátce provozováno v Anglii koncem 19. století. Byl jím *mechanický kůň*, navržený Angličanem L. A. Ryggem. Svůj návrh (Obr. 1) si autor nechal 14. února 1893 patentovat, kůň se však nedochoval, zděšení vesničané ho zničili. Zdrojem energie bylo obří hodinové péro.

V období první světové války se začaly objevovat první úspěšné realizace nejen pásových, létajících, ale i chodících strojů. Takovým byl i *kráčivý traktor* vyvinutý Thringem ve 20. letech minulého století, určený pro neschůdný, rozbahněný terén.



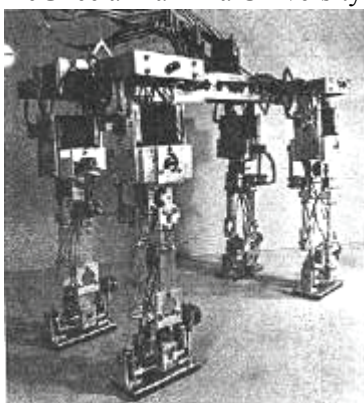


Obr. 1. Mechanický kůň, 1893.



Obr. 2. Schéma robota Phoney Poney.

S rozvojem výpočetní techniky začala v 70. letech našeho století nová éra chodících robotů. První *počítačem řízený chodící robot* (Obr. 2) vytvořili v roce 1966 McGhee a Frank na University of South Carolina. Pojmenovali ho **Phoney Poney**.



Obr. 3. Robot Collie I, 1985.

inteligentní robotiky. Výzkum a vývoj projektu COG prof. Brookse se v letech 1996-2003 úspěšně zabýval vzájemnou součinností sensorického systému robota s akčním systémem (Obr. 4). Sensorický systém byl doplněn dvěma kamerami pro 3D rekonstrukci, akční systém byl tvořen robotickou rukou [13].

O Massachusettském Institutu of Technology v americkém Bostonu, jemuž se neřekne jinak než MIT, odborníci hovoří jako o skutečné světové líhni robotů. Z jeho laboratoří už vyšly desítky nejrůznějších typů, z nichž celá řada se uplatnila v armádě, například v současnosti jsou nasazeny v Iráku, ve vesmírném výzkumu a další už jsou připravovány na své hvězdné mise na Mars a další planety. Laboratoř pro výzkum umělé inteligence v institutu vede nestor světové robotiky Rodney Brooks. Pod jeho vedením vznikají roztodivná kovová a plastická stvoření, napodobující zvířata, ptáky i člověka. Podle jeho názoru **éra robotů právě začíná a není daleko doba, kdy budou běžnou součástí našeho života**. Sám postupoval při vývoji prakticky shodně s evolucí

Vývoj kráčivých robotů má svoji dlouhou historii v Japonsku. Na University of Tokyo vyvinul v roce 1985 H. Miura 4-nohého robota Collie I s celkem 12 stupni volnosti (Obr. 3).

Jedním z prvních úspěšných projektů, který řešil úlohy spojené s vývojem humanoidních robotů, je **Project COG** realizovaný na MIT Artificial Intelligence Lab, USA. Ta se řadí ke špičkovým světovým VŠ pracovištím s výzkumem v oblasti



Obr. 4. Robot COG, 2003.

života na Zemi, od hmyzích robotů přes kybernetické čtyřnožce až po humanoidní typy robotů, které věrně napodobují lidský vzhled i chování [2].

Brooksovým cílem je nyní vývoj humanoidních robotů, kteří budou schopni samostatného myšlení, učení a rozhodování. Robot, stejně jako malé dítě se musí učit novým činnostem a pohybům od okolí, včetně vyjadřování emocí a plnění příkazů [1].

### 2.3 Inteligentní stroje – servisní roboty nové generace

Padesát let vývoje robotů rychle uběhlo. Dnes se svět mění mnohem dynamičtěji, než kdykoliv předtím a my a naše myšlení se musíme měnit s ním.

Poměrně inteligentní roboty, vybavení sice jen ještě omezenou mobilitou a autonomností, mohou dnes již naprosto spolehlivě vykonávat mnohé úkoly, určené ještě donedávna pouze člověku.

Profesor Rodney R. Brooks z MIT kdysi (1996) charakterizoval mobilního inteligentního robota třemi požadavky a to takto:

1) *Je situovaný v prostředí -Environment*, aby mohl být tímto prostředím bezprostředně ovlivňován;

2) *Musí mít tělo -Embodiment* pro vnímání dynamiky prostředí a jeho vlastností;

3) *Musí mít inteligenci -Intelligence +Mind*, aby mohl vnímat, pozorovat a učit se.

Takto výstižně definovaný robot musí navenek vystupovat jako jeden celek. Z druhého rysu plyne požadavek, že nesmí být virtuální, nesmí mít ani virtuální senzory.<sup>2</sup> Požadavkem je také *situovanost prostředí*.

Absolutní počet průmyslových a zejména servisních robotů ve světě prudce narůstá. Servisní roboty nové generace se musí umět učit, a to, co se naučí, musí také umět správně použít. Za vrchol technického umu při vývoji servisních robotů jsou považováni *roboti s lidskou podobou*, tzv. **humanoidní roboti**. Pod tímto zavedeným označením je označován robot, který je *flexibilní, umí se učit, je univerzálně použitelný* a svými *schopnostmi i vzhledem se co nejvíce blíží člověku*, přičemž jeho struktura je díky jeho mechanickému provedení mnohem robustnější a odolnější než lidské tělo. Jejich, zatím stále dost vzdáleným cílem je plně *převzít* nejen *typické úkoly člověka ve výrobě, v kanceláři nebo v domácím prostředí*, ale správně se rozhodovat a *jednat i v předem nepřipravených situacích*.

S humanoidními roboty se v této kapitole seznámíme podrobněji z různých úhlů pohledu. Od historického pozadí vývoje až po současnost, včetně jejich prvních úspěšných realizací, včetně popisů jejich technických prvků a architektury řízení.

Roboty obecně lze v současnosti považovat již za klíčovou komponentu všech moderních automatizovaných průmyslových výrob. Počet průmyslových robotů připadajících na 10 000 pracovníků ve zpracovatelském anebo výrobním průmyslu je důležitým měřítkem stupně automatizace v příslušné zemi. Podle studie *World Robotics 2008*<sup>3</sup> byl koncem roku 2007 na světě používán téměř jeden milion průmyslových a 5,5 milionu servisních robotů. Roboty pracovaly v továrnách,

<sup>2</sup> Virtuální roboty na internetu do Brooksovy definice nezapadají, humanoidi však ano.

<sup>3</sup> Studii zpracovali experti statistického odboru IFR Statistical Department ve spolupráci s Fraunhoferovým ústavem IPA (Institut für Produktionstechnik und Automatisierung).

v nebezpečných a zdraví škodlivých prostředích, v nemocnicích, domácnostech, ve veřejných budovách, pod vodou, pod zemí, na polích, ve vzduchu i v kosmu, zkrátka všude! I přes současnou hospodářskou recesi IFR (*International Federation of Robotics*) odhaduje, že do konce roku 2011 bude ve světě v provozu téměř 1,2 milionu průmyslových robotů a asi 17,5 milionů servisních robotů. Z geografického hlediska je prodej robotů velmi nerovnoměrný a na deset zemí s největším počtem nově instalovaných robotů připadá v roce 2007 více než 86 % celkového prodeje ve světě. Podle celkového počtu prodaných robotů prvenství patří Asii (zejména Japonsku a Jižní Koreji) a USA<sup>4</sup>, trh ale nejrychleji roste v Evropě (nejvíce pak v Německu). Velký boom vykazují investice do robotů v Číně, třetím největším trhu v Asii. Poptávka zde roste ve všech průmyslových odvětvích, včetně velmi rychle se rozrůstající výroby automobilů.

**Hustota použití robotů** je důležitým měřítkem stupně automatizace v příslušné zemi. Japonsko, kde na 10 000 pracovníků v průmyslu připadá 310 průmyslových robotů, je zemí s vůbec největší hustotou použití robotů ve světě a také nejvíce automatizovanou zemí světa. Na druhém místě je podle daného ukazatele Německo s hustotou 234 průmyslových robotů na 10 000 pracovníků v průmyslu.

V automobilovém průmyslu, který je z hlediska použití průmyslových robotů stále tím nejdůležitějším odvětvím je hustota použití robotů stále podstatně větší, než v ostatních odvětvích průmyslu. Na prvním místě je zde opět Japonsko s hustotou 2100 robotů na 10000 výrobních dělníků<sup>5</sup>. Na dalších místech pomyslného žebříčku jsou Itálie a Německo. Teprve s poměrně velkým odstupem následují USA, Francie, Velká Británie, Španělsko, Švédsko a Česká republika.

Vedle průmyslových robotů přichází na trh stále více *servisních robotů* – chytrých, automatizovaných pomocníků pro profesionální i soukromé použití. Podle dostupných údajů bylo na začátku roku 2008 ve světě v provozu asi 55 000 **profesionálních servisních robotů**<sup>6</sup>. Asi čtvrtinu z nich tvoří roboty pro obranné, záchranné a bezpečnostní úkoly. Tyto roboty patří stále k nejdražším. Na druhém místě<sup>7</sup> (20%) jsou *zemědělské roboty*. Poté následují *čistící roboty*, roboty *pro práci pod vodou*, *stavební a bourací roboty*, *lékařské roboty* a *mobilní inteligentní plošiny*. Menší jsou počty robotů, předurčených k *logistickým, inspekčním a prezentačním* činnostem. Servisní roboty, které provádějí náročné a nebezpečné montážní operace v továrnách, v podzemí prohlížejí kanály nebo na vzdálených planetách sbírají vzorky hornin, jsou již několik let skutečností. Mnohé procesy si lze bez nich už jen těžko představit.

Podle Mezinárodní federace pro robotiku IFR lze v nejbližších letech počítat s dalším velkým nárůstem počtu servisních robotů v soukromé sféře. Zřejmě již není daleko doba, kdy servisní roboty ovládnou i všední dny v našich domácnostech. Zejména až se stávají *autonomní servisní roboty pro osobní použití* – bez ohledu na

<sup>4</sup> Do Ameriky bylo jen v roce 2007 dodáno asi 19 600 nových průmyslových robotů, tj. o 9 % víc než v roce 2006. Přes stávající recesi se v roce 2008 odhaduje další mírný vzestup trendu.

<sup>5</sup> V japonském automobilovém průmyslu v roce 2007 připadal na každých 5 dělníků 1 robot!

<sup>6</sup> Odhadovaná pořizovací hodnota cca 10 mld. amerických dolarů.

<sup>7</sup> Zejména díky robotům pro komplexní automatizaci dojení – dojícím robotům.

to, zda již dnes vysávají prach, udržují pořádek v bytě, podávají informace nebo zalévají květiny – podaří vyvinout do stavu, kdy budou schopny vyrovnávat se i s neznámými situacemi. Tedy až budou plně vybaveny umělou inteligencí a dalšími sofistikovanými vlastnostmi. Osobní servisní robot blízké budoucnosti se bude totiž muset rychle, stále znovu a znovu orientovat v prostoru, jehož uspořádání tvůrci robota nemohou předem předvídat, a tudíž ho na ně naprogramováním nemohou připravit. Bude muset rychle reagovat i na náhlé změny v chování svého okolí a na reakce jiných.

Na jaře roku 2008 bylo ve světě v provozu asi 5,6 milionů **servisních robotů pro osobní použití**, z nichž asi 3,4 milionu robotů vykonávaly různé práce v domácnostech nejčastěji určené *k vysávání prachu, mytí podlah a oken, sekání trávy, čištění bazénů a pro mnohé další stereotypní*, na inteligenci příliš nenáročné aplikace. Nejrozšířenější robotický vysavač Roomba je klasickým příkladem, že **robot nemusí nutně být humanoidem, aby byl užitečný**. Asi dva miliony robotů jsou dnes používány *ke hraní, zábavě, výuce* apod., tj. *při aktivitách spjatých s volným časem*. Trh těchto malých mobilních osobních robotů se tak stal pozvolna nejméně 100 x rozsáhlejším, než trh profesionálních servisních robotů (o pevných a nemobilních průmyslových robotech nemluvě). Tyto roboty se vyrábějí doslova pro masovou spotřebu a jejich prodejní cena je jen zlomkem toho, co stojí servisní roboty pro profesionální použití. Trh *servisních robotů pro podporu starých a tělesně postižených osob* je zatím ještě velmi malý, ale s ohledem na demografický vývoj společnosti a stále se zdokonalující techniku se očekává, že v příštích deseti letech se tyto roboty stanou kromě humanoidů další klíčovou a expandující oblastí osobní servisní robotiky.

Podle zmíněné studie IFR se v letech 2009 až 2011 předpokládá velký růst poptávky po servisních robotech pro profesionální použití. Na trh by jich mělo v této době přijít nejméně dalších 55 000 v hodnotě 9 mld. dolarů. Největší perspektivy růstu mají podle očekávání *vojenské roboty* (zejména bezobslužná letadla a útočná i obranná vozidla), *zemědělské roboty*, *čisticí roboty* všeho druhu a *inteligentní roboty pro chirurgické operace*.

Enormní boom se očekává u *humanoidních robotů*, i když ty zatím známe převážně jen ve funkci *inteligentních hraček* nebo *sluhů*. *Příkladem* jsou robotický pejsek Sony Aibo, Tamagotchi, Sumo Robot Kit, Ludobot, Wakamaru, Enon a desítky dalších. Některé japonské podniky (Honda, Kawada, Toyota aj.) již sice vyvinuly a vyrobily **první humanoidní roboty**, které jsou opravdu *schopné přebírat vybrané typické úkoly člověka ve výrobě, v kanceláři nebo v domácím prostředí*, ale zatím byly dodávány převážně jen japonským a mezinárodním laboratořím a univerzitám pro výzkumné a vývojové účely a detailní odborné informace o nich jsou velmi kusé.

V podmínkách současné otřesené globální ekonomiky jsou však „humanoidi“ vzhledem ke svým vysokým prodejním cenám a svým jen omezeným dovednostem stále ještě ne zcela dostatečně lákavým artiklem. Jsem však přesvědčen, že tento stav je dočasný a brzy dojde k obratu.

## 2.4 Charakteristika humanoidů a nejbližší cíle jejich vývoje

Schopnosti inteligentních servisních robotů překotně vzrůstají. Průměrný lidský mozek (cca 1,500 gramů) umí zpracovat okolo sto biliónů instrukcí za sekundu - tedy stále téměř třikrát více, než stihne i ten nejvýkonnější počítač světa *BLUE GENIE* firmy IBM. Podle Mooreových zákonů by úroveň výkonu počítačového zpracování mohla vzrůstat až k bodu, kdy stroje budou moci pracovat téměř tak dobře jako lidský mozek (Minsky, 2005). Dokončovaný *molekulární počítač* tyto meze opět posune.

*“Nechme to proboha, jak to je! Nezahrávejme si s ohněm! Dostatečně užiteční roboti nepotřebují mít přeče k dispozici mozkovou kapacitu srovnatelnou plně s člověkem,”* říká profesor Hans Moravec z Carnegie Mellon University.

Pomineme-li lidský mozek a jeho těžko dosažitelnou výkonnost, vítězí ale dnes na trhu jednoznačně **trend antropomorfismu**. Antropomorfismus<sup>8</sup> označuje jev, kdy jsou na neživé předměty, přírodní síly, nadpřirozené bytosti a tedy i umělá stvoření přenášeny lidské charakteristiky, vlastnosti, či přímo lidská podoba (stavba kostry, styl chůze). Antropomorfizované stvoření se nazývá *antropomorf*.

**Android** – je antropomorfní stvoření, podobné člověku – založené obvykle na biochemické bázi a ne pouze na mechanické, jako klasičtí roboti. Roboti v R.U.R. byli tedy podle tohoto dělení vlastně androidi. Jejich úkolem je chovat se, pracovat, popřípadě uvažovat stejně jako člověk. Jelikož se rozlišuje pohlaví i mezi roboty, označuje tento pojem umělého člověka mužského pohlaví a vizáže.

**Humanoid** - robot podobný člověku principiální stavbou těla a zejména způsobem pohybu.

**Gynoid** - humanoidní typ robota, vzhledem i chováním představujícího ženu. Robota údajně ženského pohlaví, který již navenek velmi připomíná živého člověka, představil profesor Hiroshi Ishiguro z univerzity v Ósace. Model nazvaný robotka **Repliee Q1** se totiž opravdu překvapivě věrně podobá dospělé ženě (asijského původu).



Obr. 5. Repliee Q1.

**Kyborg** - složenina ze slov *kybernetický organismus*. Představuje živou bytost obohacenou o mechanické či elektronické součástky. Tyto součástky jsou obvykle nesnímatelné a jejich montáž nebo demontáž se provádí formou chirurgického zákroku. „Klasičtí“ kyborgové se zatím vyskytují pouze ve sci-fi, nicméně technicky je kyborgem i člověk s endoprotézou, naslouchátkem nebo srdečním stimulatorem. Sestrojení kybernetických protéz s kvalitou srovnatelnou, nebo přesahující kvalitu nahrazovaných orgánů, je záležitostí blízké budoucnosti.

Snad nejznámější je případ britského vědce prof. **Kevina Warwicka**, z University v Reedingu, který si nechal do ruky na tři měsíce voperovat elektronický implantát. Experiment byl podle něj úspěšný, protože dokázal pohyby své ruky přimět umělou dlaň na druhé straně Atlantského oceánu sevřít předmět [15].

<sup>8</sup> Z řečtiny: *anthrōpos* – člověk, *morphē* – tvar, forma

**Humanoidní roboti**, svými schopnostmi i vzhledem se co nejvíce blížíci člověku jsou již konstruováni s jasným cílem, poskytovat účelné a spolehlivé robotické funkce v takové estetické formě, která zvětší pravděpodobnost jejich přijetí ve společnosti. Musí být vzhlední a své okolí nevyděsit. Roboti podobající se člověku se opravdu takto konstruují mj. proto, že stroj „s lidskou tváří“ je pro člověka přijatelnější, zejména má-



**Obr. 6.** Roboti ASIMO stojící vedle lidí, 2008 .

li mu sloužit jako pomocník v domácnosti nebo např. při ošetřování dětí, či starých a nemocných lidí. Všimněte si, že vesměs všichni realizovaní humanoidi jsou opravdu vždy nižší, než dospělý člověk (měří cca max. do 145 cm).

Již brzy budou humanoidní roboti napodobovat akce člověka ve fyzickém smyslu nebo v duševním smyslu výrazně realističtější (biomimicry).

#### *Požadavky podmiňující úspěšnou realizaci humanoida*

Shrneme-li, tak **pro vytvoření humanoida** - inteligentního robota s lidskou podobou a chováním - je třeba dosáhnout **splnění těchto tří dílčích cílů**:

1) vybavit robota stejnou *schopností pohybu, jakou má člověk*, tzn. naučit ho chodit, běhat, tančit, rychle se otáčet, couvat, chodit po schodech, apod.,

2) zajistit, aby se robot uměl *spolehlivě orientovat v prostoru a dokázal se přizpůsobit lidskému světu*. Realizace je těžká, protože naše prostředí, vytvořené podle potřeb člověka, je plné překážek a nerovností a pobyt v něm je ovlivňován mnoha jen stěží předvídatelnými okolnostmi; k tomu robot potřebuje praktickou inteligenci, která mu umožní jednat na základě zkušeností a znalosti souvislostí,

3) *přizpůsobit robota lidské inteligenci*, přidat mu abstraktní znalosti, sociální inteligenci a zjednodušit jeho programování a komunikaci s ním.

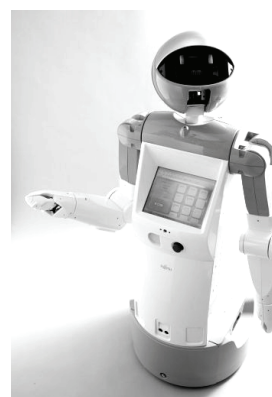
Přitom bezpochyby platí, že současný, dobře propracovaný mechatronický systém je v kombinaci s adaptivními regulačními algoritmy již docela dobrým základem humanoidních robotů budoucnosti.

Prvního humanoidního tančícího robota předvedla firma Schunk a robot Toyoda firmy Toyota překvapil na půdě ČVUT v Praze svojí čistou hrou na trubku. Firma Shadow pro roboty této kategorie vyvinula ruku pro humanoidy, analogickou k lidské, jejíž 36 pneumatických svalů umožňuje pohyb ve 24 stupních volnosti. Atraktivní veletrhů, demonstrující rychlost zpracování obrazů, je „vidící“ robot, který hraje s návštěvníky stolní tenis. Vyvinuta byla robotická mikromontáž zcela nepatrných součástí, u níž malé kompaktní kamery robota zvládají kontrolní funkce lépe, než dokáže lidské oko. V provozu je již *malý autonomní hlídací robot Secur-O-bot*. Ten rozpoznává kouř nebo nebezpečné plyny i v nepatrných koncentracích, indikuje

ohniska vznikajících požárů, pomocí infračervených kamer rozeznává osoby v temných místnostech a citlivými mikrofony registruje i sebemenší hluky.

Tito roboti už spadají do rychle se rozvíjející *kategorie sociálních robotů*, které reagují podle chování svého okolí (*Behavior Based Robotics*). Dokonce již nyní nabízejí některé převážně japonské firmy Sony, Honda své robotické výtvořky už i za spolčovníka, i když ještě stále na poměrně nízké úrovni možných jednání a akcí. Inteligentní umělá stvoření humanoidního typu jsou ale již v současnosti konstruovány tak, aby mohly poskytovat pomoc starším, nemocným a osamělým lidem (a také i jakousi první pomoc v krizových situacích). Prvními velmi jednoduchými *sociálními roboty* – spíše hračkami jsou *Tamagotchi, Aibo, Ludobot a Wakamaru, Sluha - robot Enon a další*.

Švédský *sociální robot Paro* pro duševní útěchu a relaxaci je již od roku 2007 používán v nemocnicích a domovech pro seniory. Tento překvapivě levný robot v podobě tuleního mláděte má četné senzory dotyku, obrazu a zvuku. Dokáže tedy reagovat na doteky a zvuky a je schopen interakce s člověkem. Reaguje na pohazení i na výprask, zapamatuje si, za co byl chválen či trestán. Interakcí s ním se výrazně zlepší nálada dětí upoutaných na lůžko nebo starších lidí. Paro může pomoci posílit jejich vitalitu a povzbudit je k navázání lepší komunikace i s ošetřujícím personálem.



Obr. 7. Robot Enon.

## 2.5 Inteligentní interakce mezi robotem a člověkem

### *Napodobování lidského chování*

Problém napodobování lidského chování a předjímání situací je velmi komplexní a stále otevřený. Poptávka po takových velmi nákladných robotech humanoidního typu zůstává stále nízká. „Dnes si už můžete opatřit robota, aby s Vámi hovořil, ale nemůžete po něm ještě chtít, aby s Vámi rozprávěl jako filosof Spinoza“, říká Moravec.

Nepřítomnost dalších typicky lidských orgánů – dvou plně funkčních nohou je už u většiny humanoidů zjevnější. Někteří experimentální roboti mají jakési dolní končetiny, které napodobují chůzi ještěrek, krabů nebo námořníků na vratkých nohách. Ty jsou vhodné jen pro předem připravené vesměs pouze laboratorní prostředí. Kola, či pohyblivé pásy mohou naproti tomu dobře fungovat v téměř všech prostředích a jsou také mnohem snadněji naprojektovatelná.

„Nohy – ty jsou vhodné tak pro Hollywood a umělá stvoření jako byl Terminátor“ dodává k této otázce prof. Moravec.

V poslední době vystupuje oblast *behaviorální robotiky* čím dál zřetelněji jako integrace poznatků z celé řady různorodých vědních oborů a disciplín. Všechno je vzájemně propojeno – software, hardware, mechanika, elektronická zařízení, kybernetika, etologie, aplikace progresivních myšlenek a nápadů- viz Diato [4], Franck

[5] a Lorenz [9]. To by měl nadstřešovat marketing, především co se týče podpory prodeje nápadů a jejich prvních realizací. Podle předpovědí by mohla být již za deset až dvacet let sestrojena celá nervová soustava živé bytosti. Bude tedy vyvinut vyživovaný materiál na biologickém principu. Poněkud mě to děsí, že si lidé takto zahrávají. Domnívám se v této souvislosti, že pouhé napodobování člověka nemá smysl. Život je natolik složitý, že vyrobit jeho kopii je prakticky nemožné, je to nekonečný proces.

**Inteligentní robotika** by si nyní měla dát za prioritní cíl začleňovat svá stále dokonalejší zařízení - inteligentní roboty do běžného života, aniž by se jako roboti vymezovali a aniž by se jim tak stále říkalo<sup>9</sup> [10].

#### *Umělý mozek humanoidů*

Roboty jsou co do výkonnosti a technických možností stále dokonalejší a mnohé výrobní procesy si lze bez použití robotů nyní už jen těžko představit. Servisní a zejména humanoidní roboti, jejichž vývoj výrazně pokročil, do našich všedních dnů ve větším měřítku dosud ale nevstoupili, a tak všechny jejich možnosti zůstávají stále nevyužity. Finanční náročnost takové investice je zřejmě jednou z příčin tohoto stavu. Nejsou to ale zdaleka jen otázky ekonomické, proč tomu tak je. Jedním z hlavních důvodů je to, že humanoidní roboti dnes stále ještě nejsou schopni provádět určité standardní úkony, vyžadující takové logické myšlení a rozhodování, které běžně zvládají lidé.

Na těchto problémech se proto intenzivně pracuje. Významný pokrok učinili španělští vědci. V rámci mezinárodního projektu SENSOPACK (*Sensorimotor structuring of perception and action for emerging cognition*), se jim poprvé podařilo vytvořit obvod simulující činnost mozečku (*cerebellum*) – pro senzomotorické funkce člověka velmi důležité části zadního mozku. Obvod na bázi zcela nové koncepce mikročipů nezbytných pro vytváření komplexních neuronových sítí umožňuje výrazně zdokonalit interakci mezi robotem a člověkem. Konečným cílem tohoto projektu finančně podporovaného Evropskou unií je (2006-2010), vyvinout umělý poznávací systém, který umožní především v inteligentní robotice a manipulační technice řešit komplexní „hmatové a dotykové“ úkony aktivním ohmatáváním s účelným využitím kombinace metod strojového učení UI a znalostí o biologickém mechanismu vnímání.

Na projektu aktivně spolupracují špičkoví odborníci z oborů neurofyziky, biologie, informatiky, umělé inteligence a elektroniky z dvanácti prestižních evropských univerzit a výzkumných pracovišť z devíti zemí. Vědci doufají, že jim výzkum přinese i nové poznatky pro možnost lépe léčit některá závažná neurologická onemocnění (např. Parkinsonovu chorobu).

Odborníci z univerzity v Granadě (UGR) ve Španělsku již použití nových mikročipů ověřili při vývoji prototypu umělého mozečku. Důsledně přitom napodobili komunikaci mezi mozečkem a nervovým systémem člověka. Umělý mozeček by měl umožnit humanoidnímu robotu daleko šikovněji a jemněji manipulovat s různými předměty, přičemž by jeho pohyby působily přirozeněji než dosud. Přidáním umělého

<sup>9</sup> Počítače jsou též implementovány v mnoha zařízeních, v jejichž názvu se to již nezdůrazňuje



mozečku by se také měla výrazně zlepšit interakce inteligentního robota s objekty, s okolím, z něhož je řízen, i s lidmi, s nimiž spolupracuje. To je velmi důležité zvláště pro stále žádanější kooperující robotické - MAS systémy, které využívají přímou spolupráci člověka a robota, kdy cílem je účelně navzájem zkombinovat jejich možnosti a schopnosti při využití specifických předností obou. Člověk má značnou schopnost adaptace, je kreativní a dokáže se velmi rychle přizpůsobit složitým situacím; robot naproti tomu pracuje bez únavy a s vynikající reprodukovatelností pohybu i při maximálním zatížení.

Umělý mozeček navržený španělskými vědci je nyní zabudováván do humanoidního robota, vyvinutého v Ústavu robotiky a mechatroniky Německého střediska pro leteckou a kosmickou dopravu (*Deutsches Zentrum für Luft- und Raumfahrt* – DLR) v Oberpfaffenhofenu. Přínos této etapy projektu Sensopack pro činnost a využití robotů je nesporný. Dalším cílem projektu Sensopack je vývoj umělé pokožky pro antropoidní roboty blízké budoucnosti, která by měla být podobně jako pokožka člověka citlivá na vnější podněty a velmi jemné hmatové vjemy. Navíc by měla přispět ke zlidštění vzhledu humanoidního robota tak, aby svému uživateli připomínal spíše člověka než složitý stroj. To bude nutné zejména u těch servisních a humanoidních robotů, které by měly sloužit jako pomocníci v domácnosti, nebo pečovat o malé děti, či naopak sofistikovaně ošetřovat staré, nemocné či tělesně postižené osoby [3, 11].

### 3 Výchozí aparát vývoje inteligentního agenta – humanoida

#### 3.1 Historické pozadí výzkumů Umělého života

Zastánci klasické umělé inteligence, opírající se výhradně o znalosti dlouho tvrdili, že intelekt a inteligence jsou zhruba ty samé věci a schopnost zdůvodňovat může být vyčleněna a implementována odděleně od ostatních módů myšlení. Umělou inteligenci chápali jako jednotný – unifikovatelný pochod, který lze implementovat přímo jako nějaký algoritmus – tedy sofistikovanou sekvenci logických kroků. Inteligentní systémy měly být prý proto navrhovány jako systémy, opírající se o znalosti (*Knowledge-Based*) zásadně jen od shora dolů (*top-down*). Už koncem 90. let se ale objevily zcela opačné názory. Podle nich nejlepším způsobem, jak navrhnout mechanismus, který myslí, už není snažit se nejdříve pochopit proces našeho myšlení. Daleko lepší je začít vyšetřováním struktury biologických systémů a chování bazálních mechanismů. Ty leží mnohem hlouběji než ty vědomé a jsou navíc evolucí ověřeny. Inteligence je vlastností každé populace. Navrhovat inteligentní systémy (*Behavior-Based*) se musí proto i od zdola nahoru (*bottom-up*), aby se jejich inteligence mohla vynořit (*to emerge*) – viz např. [7,8,11].

Takto nestandardně uvažující vědci vybudovali základy teorie umělého života (*ALife*) jako jakéhosi mezioborového mostu mezi evoluční biologii, etologií, sociologií a etikou na jedné straně a umělou inteligencí, počítačovou vědou a matematikou na straně druhé. Prvními „A-Lifers“, stojícími v určité opozici proti většinovým názorům

byli biologové Langton a Bagley. V architektuře počítačů to byl Hillis, v celulárních automatech Wolfram, v genetických algoritmech Holland a v programování Koza. Zmíněný robotik Rodney A. Brooks z MIT před 20 lety kritizoval stav, že se v mobilní robotice uplatňoval jen klasický přístup k Umělé inteligenci, opírající se o model světa. „*Svět sám je přece svojí nejvýstižnější reprezentací*“ píše v roce 1990 [2]. Doporučil namísto: ... „*géniů, kteří nedokážou přejít místnost začít stavět umělé pavouky a brouky, reagující instinktivně – a vůbec začít od jednoduchého života.*“ Odmítl tehdy jako první čistě lidskou předlohu chování a místo na "myšlení" vsadil na "spontánní reflexy", kterými jsou vybaveny i nejjednodušší formy života (např. hmyz se dokáže pohybovat rychle a bez problémů vyhledává cíle, přestože nemá téměř žádný mozek). Senzory (čidla) a aktuátory<sup>10</sup> Brooksových tzv. "hmyzích robotů" jsou vždy propojeny přímo. Jakmile např. hmatový senzor v noze robota ucítí překážku, dá motoru povel ke zpětnému chodu, aniž by přitom byly nutné jakékoliv výpočty. Robot přitom nevyhodnocuje, o jaký druh překážky jde, dokonce o překážkách nemá ani žádné předběžné údaje. S jednoduchým programem chování na ně však dokáže bezprostředně a většinou také naprosto správně reagovat. Takovému chování se však musel nejprve naučit. Jednak metodou pokusů a omylů, jednak radami svých tvůrců a také nastavením speciálních DNA šablon.

MIT roboti 2. poloviny 80. let nebyli převratně efektivní pouze díky svému technickému vybavení – HW, ale především díky SW, vycházejícímu z Brooksem odvážně navržené „vrstvené architektury“ (*Subsumption Architecture*) jejich řídicího systému. V programech SA je vždy řada po sobě jdoucích příkazů složena v soubor. Nejprostším souborem je pokyn "vstaň ihned po zapnutí". Následuje soubor příkazů "chůze", "zvedání nohou před překážkami" a např. "sledování zdrojů tepla". Tyto soubory (komplexy) jsou hierarchicky uspořádány do vrstev podle důležitosti. Každý komplex dovoluje nerušený průběh komplexů podřízených. Při určitých signálech, které zachytí senzory, jsou však podřízené komplexy dočasně vyřazeny. Je-li třeba se někam přesunout, je aktivován pouze komplex "chůze". Při střetu s překážkou se spouští nejbližší vyšší komplex "zvedání nohou" a komplex "chůze" je krátkodobě "mimo hru". Po zdolání překážky se opět ožíví "chůze". Pokud se robot při překonávání překážky překotí, aktivuje se instinktivně nejprve "vstávání", podobně jako u skutečného brouka, nebo malého dítěte. Toto rozvrstvení komplexů jednání, které Brooks nazýval také "*architekturou podřízenosti*", má velikou výhodu - lze ho rozšiřovat směrem nahoru o další vrstvy komplexů, aniž by přitom robot zapomněl základní "životně důležité" instinkty [2]. Takoví roboti jsou v jednání pružní a chybují minimálně. Nedokáží-li něco jedním způsobem, zkoušejí to podle jiných modelů jednání. Proti tomu klasičtí roboti, kteří bezvadně vykonávají všechny úkoly podle předem naprosto přesného programu, si v nečekané situaci najednou "nevědí rady".

Inteligentní roboti a humanoidi dneška tak nefungují již jen podle strohých a neměnných matematických postupů, ale i na základě stále se vyvíjejících emocí a zkušeností. Mají své vlastní ego, potřeby, vrtochy. Učí se mimo jiné obdobnému chování, které je vlastní živým tvorům v různých prostředích a situacích [10,14].

<sup>10</sup> Aktuátory = výkonné prováděcí prvky – např. motory nohou

### 3.2 Distribuovaná umělá inteligence a řešení úloh MAS

*Intelligence* je vlastností některých živých organismů. Vznikla a vyvíjela se v průběhu evoluce po stovky milionů let a dnes umožňuje některým živým organismům efektivně reagovat na složité projevy prostředí a aktivně je využívat ve svůj prospěch a k dosažení svých cílů.

*Umělá inteligence* (UI) jako vědní disciplína se postupně formuje v posledních třiceti letech jako průsečík několika disciplín, jakými jsou např. psychologie, neurologie, kybernetika, matematická logika, teorie rozhodování, informatika, teorie her, lingvistika atd. Její vývoj není zdaleka ukončen. Dosud neexistuje ani všeobecně přijímaná definice umělé inteligence. Marvin Minsky (1967) např. tvrdí, že „... *umělá inteligence je věda o vytváření strojů nebo systémů, které budou při řešení určitého úkolu užívat takového postupu, který – kdyby ho dělal člověk – bychom považovali za projev jeho inteligence.*“ Umělá inteligence je interdisciplinární vědou, která nemá pevně vymezený předmět zkoumání ani teoretický základ – jde spíše o soubor metod, teoretických přístupů a algoritmů, sloužících k řešení velmi složitých úloh.

#### *Distribuovaná umělá inteligence*

Sdružování jednotlivců do větších skupin je jednou z charakteristických vlastností živé hmoty. Schopnosti vzniklé skupiny výrazně převyšují jen prostý součet schopností dosahovaných každým z jejích členů.

*Týmová součinnost* má obecně tyto výhody:

- zkrácení doby řešení (vzhledem na paralelní postup),
- snížení nároků na komunikaci (předávání pouze závěrů),
- zvýšení operativnosti a spolehlivosti (rozšířitelnost týmu, vzájemná zastupitelnost).

Distribuovaná umělá inteligence (*Distributed Artificial Intelligence – DAI*) se pokouší najít metody součinnosti zúčastněných jednotlivců a vysvětlit podstatu tohoto emergujícího fenoménu. Zcela obecně se DAI zabývá součinností několika totožných či různorodých systémů při řešení společného problému. DAI se soustřeďuje především na problematiku reprezentace, zpracování a využití znalostí v těchto systémech a hledá vhodné modely způsobu rozhodování [8].

Problematika DAI se dělí obvykle na dvě oblasti:

#### *Centrálně řízené řešení distribuovaných úloh*

Řeší se pevně zvolená úloha, nejvhodnější rozdělení získávaných znalostí mezi více modulů a efektivní sdílení informací o řešení. Důraz je kladen na distribuci činností v rámci vesměs pevně definované struktury, často (centrálně) řízené.

#### *Multiagentní systémy (MAS)*

Úlohu řeší skupina volně propojených autonomních systémů spolupracujících v zájmu společného cíle. Tyto autonomní systémy se nazývají agenti, čímž se zdůrazňuje jejich

částečná samostatnost. Přitom agentem může být počítačový systém, senzor, robot nebo i člověk.

*Multiagentní systémy* kladou důraz na hledání mechanismů zajišťujících účinnou spolupráci mezi libovolnými samostatnými agenty. Autonomnost agentů je žádoucí, vazby mezi agenty jsou volnější, flexibilita systému vyšší a vnitřní život komunity bohatší.

Někteří autoři vyčleňují dnes 3 *základní směry bádání v rámci umělé inteligence*:

#### *Symbolický funkcionalismus*

Symbolický funkcionalismus je založen na dvou základních hypotézách: *funkcionalistické* hypotéze a hypotéze *fyzikálního systému symbolů*. Jinak řečeno, základními oblastmi výzkumu symbolického funkcionalismu jsou problémy reprezentace znalostí a inteligentního prohledávání stavového prostoru.

#### *Konekcionismus*

Konekcionismus předpokládá, že esence inteligence plyne ze statického propojení velkého počtu jednoduchých výpočetních jednotek. Myšlenka je inspirována mozkiem jako médiem, které zosobňuje inteligentní uvažování. Základní výpočetní jednotkou rozumíme model neuronu, který se na základě hodnoty součtu vážených vstupů excituje do aktivního stavu nebo nikoliv. Mezi hlavní oblasti, které se řadí ke konekcionismu patří *neuronové sítě*.

Neuronové sítě na rozdíl od klasických symbolických metod nejsou založeny na explicitní reprezentaci znalostí. Stejně jako v našem mozku jsou díky obrovským množstvím propojení neuronů uloženy nepřehledné dovednosti a informace, tak i v počítači robota je možné takové informace uložit nebo takové dovednosti realizovat.

#### *Robotický funkcionalismus*

Klíčová filosofie robotického funkcionalismu je založena na implementaci behaviorismu jako psychologické školy. Raději než se zabývat reprezentací inteligence, robotičtí funkcionalisté se koncentrují na funkcionalitu modelovaného systému. Jako inteligentní chování je zde chápána rozumná interakce mezi třemi entitami: systém, prostředí, úloha. V případě, že bude agent na danou úlohu a v daném prostředí reagovat inteligentně (stejně jako by reagoval člověk), je považován za agenta disponujícího schopností vykazovat inteligentní chování.

### **3.3 Agentové technologie a umělý život**

Agentové technologie jsou přirozeným rozšířením současných přístupů, opírajících se o dílčí komponenty. Staly se nejdynamičtější a nejvíc stimulující počítačovou vědou dneška. V dnešní době už čistý výzkum v pevně vymezených oblastech, jakými jsou mobilní robotika, nanotechnologie, sociální a evoluční systémy prakticky neexistuje. V praxi, v reálném světě jsou mnohem zjevnější rozsáhlé mezioborové aplikace výzkumu, zahrnující a vzájemně propojující poznatky od medicíny a řízení procesů až po virtuální podnikání a e-komerci.

Agenti jako obecně opakovaně použitelné vzory budou dále ovlivňovat aplikační oblasti v mnoha směrech:

Za prvé jsou určitou metaforou pro návrh velkých celků, distribuovaných výpočetních systémů. Za druhé jsou zásobárnou - zdrojem technologií pro zmíněné počítačové systémy. Za třetí slouží jako modely i velmi složitých reálných systémů, které můžeme nalézt například v biologii a ekonomii [9,12,14]. To platí obecně i pro oblast výzkumu umělého života.

Podle toho, zda a na jaké úrovni jsou agenti schopni zvažovat rozličné varianty řešení svého cíle, zavedli v roce 1995 Wooldridge, Jennings [16] *členění agentů* do následujících tří kategorií (intencionální, reaktivní a sociální):

- **Reaktivní agent** vždy pouze reaguje na podněty z vnějšího světa. Zná pouze omezenou množinu akcí, na něž je schopen reagovat. Upřesňuje svůj obraz světa na základě vnějších podnětů. Své výsledky může sdělovat ostatním.

- **Intencionální (deliberativní) agent** zvažuje své možnosti dosažení cíle složitěji. Při výběru jeho dalšího kroku k dosažení cíle se již odráží *motivace, intence* (chtění, záměry) agenta. Koordinace skupiny spočívá ve vzájemném informování jejích členů o tom, co momentálně akceptují (*belief*) a co plánují.

- **Sociální agent** pracuje s explicitními modely chování ostatních agentů. Samozřejmě má schopnost upravovat a aktualizovat modely, neboť je používá k přizpůsobení plánované činnosti jiných agentů.

- **Hybridní agent:** V reálném světě dnes vytvářených inteligentních robotů (včetně humanoidů) se vyskytují nejčastěji **agenti s hybridním chováním**.

Problematika inteligentních agentů i multiagentních systémů (MAS) byla v dostupných učebnicích UI mnohokrát a dostatečně do hloubky prezentována. Pro kontinuitu výkladu připomenu jen *základní vlastnosti inteligentního agenta - robota*:

### 3.4 Vlastnosti inteligentního agenta - robota podle Wooldridge

1. **Autonomnost (Autonomy)**. Inteligentní agent je schopen jednat samostatně a řídit své vnitřní stavy.

2. **Reaktivita (Reactivity)**. Reaktivní agent vnímá interaktivně své okolní prostředí a reaguje včas na změny probíhající v něm, tak aby plnil cíle, pro něž byl navržen.

3. **Proaktivní přístup (Pro-activeness)**. Reakce agenta na podnět z okolí neurčuje pouze momentální stav, ale jeho chování je řízené cíli v tom smyslu, že agent si na základě své vlastní iniciativy vytváří cíl a může sám převzít iniciativu při řešení.

4. **Sociálnost (Social Ability)**. Agent je schopen interakce s ostatními agenty (případně i s lidmi) prostřednictvím společného jazyka (pro komunikaci agentů) a má tak i možnost s nimi kooperovat. To se někdy nazývá i *komunikační schopnost agenta*, umožňující, aby agent dostával potřebné informace z různých zdrojů.

5. **Schopnost spolupráce (Capacity for Cooperation)**. Inteligentní agent pro dosažení cíle spolupracuje s ostatními agenty.

6. **Schopnost zdůvodnění** (*Capacity for Reasoning*). Inteligentní agent může vykazovat schopnost vyvozovat další závěry z aktuálních znalostí a vlastních zkušeností.

7. **Adaptivní chování** (*Adaptive Behaviour*): tj. když se inteligentní agent učí nebo mění své chování na základě předchozích zkušeností.

8. **Důvěryhodnost** (*Trustworthiness*): Uživatel agenta (*user*) si musí být vysoce jistý, že agent bude působit k jeho prospěchu a bude mu vždy poskytovat pravdivé informace.

Wooldridge ve své zprávě [17] z roku 2002 zavádí některé další *občas diskutované vlastnosti a činnosti agentů* ve smyslu Minského definice. Jsou to:

- **Mobilita** (*Mobility*): schopnost agenta pohybovat se v rámci např. elektronické sítě (www včetně).

- **Věrohodnost** (*Veracity*): zda nebude její agent vědomě poskytovat nepravdivé informace.

- **Benevolence** (*Benevolence*): v posouzení, zda a jak mnoho je užitečné držet se i těch cílů (*conflicting goals*), s nimiž se agenti dostávají do konfliktu.

- **Racionálnost** (*Racionality*): zda agenti podřizují svoji aktivitu dosažení svých cílů a zda nebudou po rozvaze úmyslně zamezovat svým faktickým jednáním těm cílům, kterých se má (podle člověka) v dalším dosáhnout.

- **Učení/adaptace** (*Learning/Adaptation*): Jedná se o vlastnost, kdy agenti v průběhu času svoji výkonnost zlepšují (obvykle učením a adaptací).

Taková definice agenta sjednocuje širokou oblast zainteresovaných disciplín od umělé inteligence a inteligentní robotiky po evoluční biologii, etologii a sociologii.

## 4 Vytváření humanoidních robotů

### 4.1 Technické předpoklady realizace umělých forem života

1. První krok k vytváření nové generace umělých bytostí - (nazvějme je zatím ještě inteligentní roboti) spočívá v **návrhu nových supervýkonných procesorů** na objednávku, minimálně 10000 krát rychlejších, než jsou dnešní. Poté už bychom měli být schopni implementovat v reálném čase systémy počítačového vidění, sofistikované rozhodovací systémy, mechanismy výběru akcí (*Action Selection Mechanism*) a vyhodnocování nasnímaných dat z inteligentních senzorů různých typů. To je skutečně důležité.

2. Vystupuje **potřeba nových originálních látek a hmot**, obdobně progresivních, jako byly ty, vyvinuté původně pro kosmický průmysl (*Smart Fabrics*). Vyžadujeme pohodlně zpracovatelné, měkké a snadno udržovatelné struktury; vývoj směřuje k integraci elektronických modulů do textilních látek.<sup>11</sup> Jako nejlepší se jeví vetkání

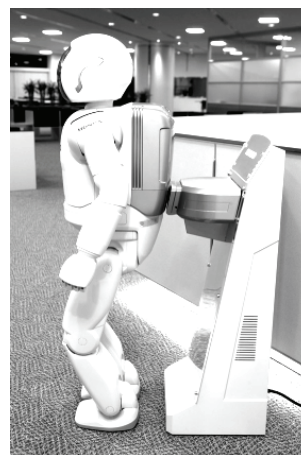
<sup>11</sup> Tento výzkum je další fází miniaturizace IT a VT s cílem učinit elektroniku lidmi i roboty „oblékatelnou“ (*wearable*). Jednotlivé funkční moduly elektronických systémů jsou na různých místech integrovány do povrchového oděvu těla. Odtud jsou přímo propojitelné do konkrétních komunikačních, počítačových a senzorových sítí (přes displej SMS, GPS, TK...).

vodičů a sběrných systémů přímo do textilie. Na jednotlivé vodivé dráhy se potom připojí mikroelektronické moduly, které musí být samozřejmě lehké, omyvatelné a podobně pružné, ohebné a trvanlivé jako běžné tkaniny. Takové vlastnosti má i nově vyvinutý velmi tenký transponderový modul s tkanou anténou, který se po zašití do obleku může i prát.

### 3. Hledáme *úspornější pohony* robotů.

*Umělé svaly* namísto energeticky velmi náročných motorů a převodovek pohybu se zdají být výhodnou cestou v budoucnosti. Ty současně však vhodné pro větší zátěže zdaleka ještě nejsou.

4. Je třeba se soustředit na otázku *napájení*, což je jeden z hlavních problémů v podstatě všech mobilních elektronických systémů (včetně notebooků). I zde bude asi nutné nastoupit nové cesty. Boty s vysokými, masivními podpatky se sice hodí dobře k umístění přídavných akumulátorů, jsou však příliš těžké i pro antropoidy. Slibné jsou zejména nové, velmi tenké a pružné tzv. *fóliové akumulátory*. I ty se dají integrovat přímo do povrchové vrstvy oděvu. Mohou se trvale dobíjet solárními články, umístěnými nejlépe v oblasti ramen,



Obr. 8. ASIMO – dobíjení.

kam dopadá nejvíce denního přímého světla. Aby bylo možné solární články pružně přimknout k rameni, jsou rozdělené do velmi malých segmentů, zapouzdřené a zapuštěné do pružného substrátu povrchové vrstvy oděvu<sup>12</sup>.

5. Zásadní význam má *vývoj kvalitní senzorky*, bez které je pokrok v oblasti kvalifikovaného rozhodování značně problematický. Množství stimulů, které je inteligentní robot schopen registrovat a efektivně je zpracovat, přitom silně determinuje jeho inteligenci (a naopak). Analogicky to platí u všech živých tvorů. Vedle inteligentních senzorů, zpřesňujících informaci o prostředí a o realizovaných akcích je důležité i kvalitní zpracování informací, které z těchto senzorů přichází. Člověk si z ohromného množství redundantních informací, zaznamenaných zrakem filtruje jen ty důležité a podle toho se rozhoduje. Obdobně je i úkolem inteligentního zpracování informací ze senzorů vybrat ty, které jsou pro další rozhodnutí podstatné. Robot by měl mít také možnost provázat a tím i verifikovat informace ze všech dostupných senzorů. Obdobně i člověk podle potřeby snímá a vyhodnocuje kromě zrakových i další např. sluchové, čichové, hmatové, případně i chuťové informace.

I zde je však nutno nejdříve závazně definovat jednotné, otevřené a na výrobcích nezávislé *komunikační rozhraní* pro inteligentní senzory. Při návrhu robotických celků tak bude snad již možné využívat senzory různých výrobců. Zatím je velký problém právě s jejich nezaměnitelností a vzájemnou nekompatibilitou [10].

<sup>12</sup> Vyvinuto pracovníky Frauenhoferova ústavu IZM (*Institut für Zuverlässigkeit und Mikrointegration*) v Berlíně v roce 2006.

## 4.2 Podvozky mobilních robotů

- *diferenciální podvozek* – dvě hnaná kola, rovnováha udržována opěrnými body, nebo pasivním kolem (koly),
- *synchrónní podvozek* – často 3 kola, každé se 2 stupni volnosti (může se otáčet i natáčet),
- *trojkolový podvozek* s řízeným předním kolem – 2 hnaná kola a jedno motoricky natáčené,
- *Ackermanův podvozek* – 4 kola, 2 hnaná a 2 natáčená kola (každé mírně jinak, protože každé při zatáčení opisuje jinou dráhu); tyto podvozky mají i běžné automobily,
- *trojúhelníkový podvozek* s třemi nezávisle poháněnými koly, jejichž osy procházejí těžištěm a jejichž povrch (složený obvykle z malých koleček) umožňuje volný skluz ve směru osy,
- *podvozky se všesměrovými koly*,
- *pásové podvozky*,
- *kráčející podvozky*.

## 4.3 Způsoby chůze u nohových mobilních robotů

**Kráčející roboty** jsou po kolových a pásových mechanismech třetí nejčastější variantou mobilních robotů a dominantní pro humanoidy. Na rozdíl od předchozích dvou typů je nespornou výhodou tohoto typu konstrukce schopnost překonávat větší překážky a nerovnosti povrchu. Výška překážek, které je robot schopen překonat, je omezena výškou světlosti podvozku a zdvihu nohy. Manévrovací schopnost robota je výrazně závislá na konstrukčním provedení robota, obecně však tyto typy robotů mají dostatečně dobré manévrovací schopnosti na úrovni kolových mechanismů.

*Nevýhodou nohové konstrukce* robota je vysoká energetická náročnost<sup>13</sup> na realizaci pohybu a dost značná výpočetní náročnost na řízení samotného pohybu. Oproti kolovým a pásovým konstrukcím dochází u humanoidů během jejich pohybu k dynamickému pohybu těžiště oproti opěrným bodům. Řídicí systém proto musí dostatečně řídit a stabilizovat polohu těžiště, aby se robot nepřeklopil. Důležitým kritériem použitého řízení je tedy *stabilita*. Navíc kráčivý podvozek je omezen v rychlosti přímého pohybu, jenž nadto není plynulý, ale probíhá ve vlnách. Vhodným řízením však lze zajistit přizpůsobení rychlosti a způsobu pohybu robota (typ chůze) aktuálnímu typu povrchu.

**Dvounožní roboti** patří mezi nejobtížnější typy chodících robotů jak z hlediska konstrukce robota, tak z hlediska jeho řízení. Kinematika tohoto robota odpovídá kinematice inverzního kyvadla. V klidovém stavu tento robot nemá staticky stabilní polohu. Pro zajištění rovnovážné polohy robota je nutné řídit polohu těžiště robota.

<sup>13</sup> Robotické “kráčedlo” bude vyžadovat okamžitý odběr až 15A, příkon až 2 kW.



### *Způsob chůze inspirovaný pohybem hmyzu*

Používá se pro malé jednoduché roboty bez, nebo se značně omezeným senzoričným systémem. Zavádí pojem *Gait State* jako množinu stavů - všech možných kombinací současných opěrných nohou (tedy těch, co nesou tělo robota ve stejném okamžiku).

Rovina je jak známo určena třemi body. To je také důvodem, proč jsou pružná nášlapná chodidla obou noh humanoida tak velká.

Vytváří se tři podmnožiny stavů:

**nestabilní stavy:** podmínka statické stability není nikdy dodržena

**částečně stabilní stavy:** záleží na vzájemné poloze chodidel nohou a těžiště; stav může být jak nestabilní, tak i stabilní

**absolutně stabilní stavy:** nezávisle na poloze jednotlivých chodidel, stojí-li robot na nohách odpovídající stavu z této podmnožiny, je podmínka statické stability vždy dodržena.

Chůze této kategorie využívá cyklické záměny absolutně stabilních stavů. Typickým příkladem je *Alternate Tripod*.

Cílem této kategorie je jednoduchý pohyb daným směrem – vpřed, případně včetně zatáčení. Není řízena poloha těla, způsob překonávání překážek – robot se pohybuje shodně po rovném i nerovném terénu, není řešena distribuce váhy robota mezi jednotlivé nohy, otázka dynamiky pohybu. Výhodou je značná jednoduchost a tím i výpočetní nenáročnost, celé řízení lze realizovat i s jedním osmibitovým procesorem, bývá realizována i jednoduchými neuronovými sítěmi;

### *Způsob chůze založený na silovém působení „noha robota – podložka“*

Zde je řízení chůze založeno na informaci o reakčních silách v chodidlech robota a hlavním silovém vektoru působícím na tělo robota, vyvozovaném jeho aktuátory. Tedy zatímco se v minulém bodě vychází z geometrických vztahů, které lze rozšířit o zpracování informací o silách v podvozku, zde je tomu obráceně.

### *Způsob chůze založený na geometrických pravidlech*

Takové řízení zjednodušeně vychází z faktu „pohybuje-li se noha, na níž robot stojí, vůči tělu doleva, pak takový pohyb posune robota doprava“. Řídicí systém si tedy udržuje informaci o aktuální poloze jednotlivých nohou a na základě požadovaného pohybu robota určuje směr a rychlost jejich pohybu. Často je vytvořen pro konkrétního robota, jeho kinematickou strukturu nohou, a řeší současně i úlohu zpětné kinematické transformace. U složitější kinematické struktury nohy, kdy má každá noha aspoň tři stupně volnosti, se využívá výhody možného oddělení řešení pohybu těla robota od řízení pohybu jednotlivých nohou. Na noze se definuje opěrný bod – chodidlo. Jeho pohyb vzhledem k robotu realizuje chůzi, zajištění tohoto pohybu často realizuje mikrořadič nohy (např. robot SILEX, 2005)

Řízení může být řešeno nejen z tohoto „geometrického“ pohledu, ale celý řídicí systém chůze může dále řešit např. distribuci sil v podvozku – nepřetěžovat nohy, dynamiku trupu. Pohyb humanoida po měkkém, nestabilním terénu vyřešil úspěšně Lehtinen (2000).

*Pozn.:* Japonci vyřešili úspěšně právě stabilní pohyb dvounohých robotů Honda, včetně jejich chůze po nerovném terénu, či po schodech nahoru a dolů (jednodušší). Zajímavá je u jejich humanoida ASIMO jeho **chůze do schodů a dolů**. Robot totiž zaměřuje svoji přesnou pozici vůči černobílým značkám před schody. Stejně značky se nachází i na plošině, ze které robot později sestupuje po schodech dolů. O použitém algoritmu na chůzi po schodech hodně napovídá, že fotografovat lze humanoida až při jeho sestupování. Robot nepoužívá kameru během chůze po schodech, ale pouze před jejím začátkem. Při výstupu nesmí být proto oslnován blesky. Rozměry schodů mu jsou zřejmě v případě jeho oslnění při fotografování dodány i alternativním způsobem.

#### *Generátor chůze dvounohého robota*

Zmíněné rozšíření vlastností chůze, jako jsou např. změna polohy těla či chůze po nerovném terénu, vyžaduje změnu kráčivého podvozku. Proto se používá složitější kinematika se třemi stupni volnosti na nohu. Geometrický pohled popíše řízení chůze pro potřeby řízení polohy jednotlivých chodidel v prostoru a čase, tak, aby se robot pohyboval - kráčel požadovaným způsobem. Chodidlem budeme rozumět pružný došlapový bod posledního článku nohy. Důležitým momentem je zajištění *průběžné statické stability postoje robota*. Není uvažována hmotnost robota a s ní související řešení zatížení jednotlivých nohou a dynamika pohybu. Důležitým rysem realizace řídicího systému je požadavek jen takové výpočetní a paměťové náročnosti, kterou lze zvládnout běžnou výpočetní technikou [12].

Generátor chůze je koncipován tak, že řízený robot, jeho velikost a tvar, stejně jako rozmístění, počet a kinematická struktura nohou, vystupuje spíše jako upřesňující mnohorozměrný parametr, než jako prvek, zásadní pro vlastní funkci generování chůze. Pak stejným generátorem chůze lze řídit robota při jejich různých typech.

Vstup generátoru chůze lze v obecné podobě zapsat jako šesti-prvkový vektor, který plně určuje pohyb a polohu těla robota v prostoru, resp. polohu těžiště těla robota. *Pohyb i polohu robota vyšetřujeme vždy vzhledem k jeho těžišti*. Takový vektor nazvěme

**vektor řízení robota:**

$$w_R = (v_{R_X}, v_{R_Y}, \omega_R, \alpha_R, \beta_R, h_R),$$

kde jednotlivé proměnné mají tento význam:

$v_{R_X}$  - rychlost pohybu vpřed,

$v_{R_Y}$  - rychlost pohybu stranou,

$\omega_R$  - úhlová rychlost otáčení – „zátáčení“,

$\alpha_R$  - náklon těla robota,

$\beta_R$  - úklon těla robota,

$h_R$  - „nadmořská“ výška těla robota, výška nad zvolenou vztažnou vodorovnou rovinou.

*Vektor řízení robota* můžeme formálně zapsat jako

$$w_R = (w_G | w_A),$$

kde  $w_G$  zahrnuje proměnné, které řídí chůzi:

$$w_G = (v_{R_x}, v_{R_y}, \omega_R),$$

a  $w_A$  zahrnuje proměnné, které řídí postoj robota:

$$w_A = (\alpha_R, \beta_R, h_R).$$

Vektor řízení těla robota nese informaci o pohybu, požadovaném nadřízeným systémem. Generátor chůze tento pohyb realizuje, je-li takový pohyb možný. V opačném případě ho neprovede. Pak vzniká rozdíl mezi požadovaným pohybem robota a jeho skutečným pohybem. Výstupem generátoru chůze je pohyb chodidel jednotlivých nohou robota tak, aby pokud možno byly splněny požadavky vstupu. Protože uvažujeme nohy s aspoň třemi stupni volnosti, lze realizovat nezávislý pohyb chodidla ve vodorovné rovině  $x-y$  a svisle dle složky  $z$ . To umožňuje i zde zavést *dvouvrstvou koncepci generátoru chůze*:

**Koordináční vrstva** generuje vlastní chůzi s využitím možností daných zvolenou kinematikou podvozku robota. Z hlediska pohybu představuje hlavní část generátoru chůze. Probíhá v ní vlastní generování chůze, především koordinace řízení jednotlivých nohou a plánování jejich budoucího pohybu, vyhodnocování povrchu terénu, po němž se robot pohybuje, řízení polohy těla robota, atd. Výstupem této vrstvy do vrstvy řízení nohy jsou již signály přímého řízení poloh chodidel jednotlivých nohou. Opačným směrem jsou distribuovány především informace z čidel jednotlivých nohou. Z koordinační vrstvy dostáváme navíc zpětnovazební informaci o skutečném pohybu robota, kterou lze využít pro zkvalitnění řízení.

**Vrstva řízení nohy** má za úkol řídit aktuátory<sup>14</sup> jednotlivých nohou na základě konkrétních požadavků koordinační vrstvy. Provádí *zpětnou kinematickou transformaci*, kdy vstupem jsou požadované souřadnice chodidla a výstupem odpovídající signály řízení aktuátorů nohy. Koordináční vrstva tedy pracuje pouze s polohou chodidel, nikoli se všemi kloubovými souřadnicemi nohou, které propočítává až vrstva řízení celé nohy. Tím je dosaženo výrazného zrychlení výpočtů polohy chodidla vůči ostatním chodidlům a těžišti robota. Zároveň lze i řídit chůzi a polohu těla nezávisle na sobě.

#### 4.4 Způsoby navigace mobilních humanoidních robotů

**Dead reckoning** – Matematická procedura pro určování současné pozice vozidla pomocí postupného přičítání díky známému kurzu a rychlosti v průběhu času (nejjednodušší implementací této metody je odometrie).

**Guide-path following** – Sledování vodící čáry – robot opticky či pomocí magnetometrů (či Hallových sond) sleduje vodící čáry; pro svou spolehlivost - v průmyslu to byla i stále je nejpoužívanější metoda.

<sup>14</sup> Aktuátory = zde akční členy (motory) jednotlivých řízených stupňů volnosti.

**Inerciální navigace** – využívá gyroskopů a akcelerometrů pro měření zrychlení a následně tak určení výsledné pozice.

inerciální snímače otočení

primárně snímající *úhlovou rychlost*

primárně snímající *úhlovou polohu*

mechanické *gyroskopy*

a) snímající jeden stupeň volnosti (*single-degree-of-freedom gyroscopes, SDFG*)

b) snímající dva stupně volnosti (*two axis, free gyros* – dvouosé, volné gyroskopy, 2DFG)

*optické gyroskopy* – využívající „Sagnacův efekt“ (rozdílná doba letu dvou paprsků světla v důsledku pohybuujícího se senzoru)

*akcelerometry* – využívající setrvačnosti hmoty

**Využívání GPS** (Global Position Systems)

**Navigace pomocí taktilních** (dotykových) + **proximitních** (bezdotykových) senzorů.

Pro získání informací o okolí využívají roboti nejrůznější senzory. Jejich popisem se zde ale - vzhledem k limitovanému rozsahu kapitoly zabývat nebudeme.

#### 4.5 Pohony - výkonná elektronika a motory

Pro pohon se v mobilní robotice užívají různé druhy elektromotorů. Patří mezi ně:

**Stejnoseměrný motor** (DC Motor) má nejjednodušší použití. Máme *Komutátorový*, *Bezkomutátorový* a *Krokový*, umožňující přesné natáčení s definovaným rozlišením.

**Střídavý motor** (AC Motor) – oproti DC motorům mají menší rozměry při dosažení stejného výkonu. Používají se jak *Synchronní*, tak *Asynchronní*.

**Servomotor** – integrována převodovka a inkrementální enkodér.

**Zdrojem energie pro mobilního robota** bývá nejčastěji **baterie** (*primární zdroj*, nelze jej nabít a znovu použít) nebo **akumulátor** (*sekundární zdroj* - lze nabít a znovu využít).

Nabídky různých firem z oblasti řízených pohonů bývají obecně označovány jako *Motion Control* – řízení pohybu. Ke dvěma řadám servozsilovačů jsou v naprosté většině řešení připojeny i vhodné střídavé synchronní servomotory s *otáčkovou* nebo *polohovou zpětnou vazbou*. Na hřídel motoru bývá vhodná převodovka, navržená na základě výpočtu a podle požadavků na pohon. Synchronní servopohon může být nahrazen pohonem s krokovým třífázovým motorem se zpětnou vazbou nebo bez ní. V některých úzce specializovaných soustavách se používá motor EC – *elektronicky komutovaný stejnosměrný synchronní motor*. Měníče frekvence lze konfigurovat pro funkci servozsilovače a používat k napájení servomotorů o výkonu až 500 W.

#### *Mechanické nadstavby*

Obecný rotační pohyb robota, vytvářený přímo na hřídeli motoru nebo na výstupu převodovky je často nutné převést na pohyb lineární, a to v jedné, dvou nebo třech osách. Typickým příkladem jsou manipulační systémy *pick-and-place*, jejichž úlohou je uchopit předmět a přenést jej na jiné místo. Pohyby musí být vedeny po

optimalizované trajektorii a operace vykonávány v co nejkratší době a synchronizovaně s okolními systémy zapojenými do výrobního procesu

**Převod rotačního pohybu na lineární** se obvykle řeší s využitím kuličkového šroubu nebo ozubeného řemenu. Typická délka vedení je od několika decimetrů do několika metrů. Lineární vedení pro pohyb vždy v jedné ose jsou dodávány jako celek, tedy včetně motorů s převodovkou a příslušným regulátorem. Jestliže úloha vyžaduje, aby bylo vedení upraveno pro větší zatížení, (maximálně do 200 kg), rozšíří se základní vedení o další. Obě paralelní vedení jsou spojena společnou hřídelí. Rotační pohyb s převodovkou s většími požadavky na přesnost nebo mechanickou odolnost lze řešit rotačními polohovacími moduly. V tomto případě je na servomotor nebo krokový motor namontována *harmonická bezvůlová převodovka* nebo *planetová převodovka* s vysokou účinností a převodovým poměrem až 1 : 50.

#### 4.6 Humanoidní robotika - shrnutí vývoje a její další cíle

Výzkum a vývoj robotů, zejména pak robotů humanoidních se rozvíjí překotně a často nekoordinovaně; ale my potřebujeme, aby pokrok v těchto disciplínách už pokročil do fáze jeho praktických aplikací v každodenním životě. Inteligentní servisní roboti se konečně začínají učením adaptovat pro práci i v předem nepřipraveném prostředí. A hlavně už začínají být schopni opravdu nezávislé, autonomní činnosti po dlouhou dobu provozu.

Jak již bylo výše uvedeno, jak vývoj, tak následně nákup inteligentních robotů (a to nejen humanoidního typu) je také finančně velmi náročná ekonomická záležitost. Průměrný člověk uvažuje: *„Jestliže je tak snadné najmout si důvěryhodného a spolehlivého člověka, aby uklidil dům, posekal trávník nebo přestěhoval nábytek, proč se zlobit s nákladným a nespolečenským humanoidem?“*

Roboti současnosti jsou stále ještě vhodnější pro použití především tam, kde mohou odevzdat lepší, kvalitnější práci nebo ušetřit náš čas a nikoho při tom neohrozit. Jsou vhodní např. pro precizní svařování, mikrochirurgickou operaci nebo tam, kam člověk nemůže nebo nechce jít, protože tato práce je pro něj příliš nepříjemná nebo nebezpečná<sup>15</sup>. Robot je ale nezastupitelný i v prostředí, která by naopak člověk mohl znečistit, například při výrobě krystalů křemíku nebo mikročipů [10].

K humanoidním robotům Isaaca Asimova<sup>16</sup> čeká lidstvo ještě dlouhá cesta několika desítek, ale možná i stovek let. Dát robotu lidské tělo stále přináší více nevýhod než výhod. Robotické servisní systémy jsou však již nyní běžnou součástí našeho denního života. Než humanoid budoucnosti nastoupí jako užitečný a vítaný pomocník do našich domácností, čeká na jeho tvůrce ještě mnoho práce. Zejména je třeba ještě dále *„zlidštit“ vzhled robota*, aby svému uživateli připomínal spíše člověka než složitý stroj – a aby ho nevylekal! Dále je ho třeba *vybavit senzory* a funkcemi, které mu umožní *rozpoznat mimiku a gesta* a *hlas* partnera. V každé situaci by je

<sup>15</sup> Práce při čištění jedovatého odpadu, neprůchodné kanalizace, likvidaci výbušnin, v radioaktivním prostředí, práce ve vesmíru (Mars, Měsíc) nebo pod vodou, v hlubinných dolech, atd.

<sup>16</sup> Humanoidé ve filmu: Já, robot, podle stejnojmenné povídky Isaaca Asimova z roku 1950.

takový humanoid měl umět správně interpretovat a využít k prohloubení své kooperace s člověkem - partnerem při náročných úkonech. Odborníci přitom musí mít stále na zřeteli, že humanoidní roboty neučiní pro společnost přijatelnějšími pouhé hodnoty samotných jejich technických parametrů, byť byly vynikající. Rozhodujícími pro jejich plné akceptování budou funkce, které dokážou humanoidi zítřka *spolehlivě plnit*, které mají *smysl* a především, které lidem život nekomplikují, ale naopak jim ho *usnadňují*. Mimořádně důležitým aspektem současně musí být *dokonalá bezpečnost*, protože riziko zranění technikou, která se vymkla kontrole, je obrovské.

Už v blízké budoucnosti dojde nesporně k fúzi biotechnologií, nanotechnologií<sup>17</sup> a informačních technologií do jedné jediné disciplíny. Umělé materiály s vysokou inteligencí budou kvalitnější a pohodlnější než ty přírodní. „*Nastoupila epocha syntetičnosti a všeobecné náhražkovitosti*“, říkal filosof K. Kosík: „*Vytváříme umělé látky, ale také umělé příběhy a umělé požitky i umělý život. A už i místo diference dobra a zla nastupuje rozlišení zástupné, náhražkovité: nikoli dobro a zlo, ale prioritá výkonu, úspěchu, profitu, rentability, rychlého vzestupu a obohacení!*“ Budiž toto, nám nejen technikům, ale všem zúčastněným v naší tvůrčí činnosti velkým mementem!

## 5 Úspěšné realizace robotů humanoidního typu - analýza

### 5.1 Vývoj humanoidních robotů v Japonsku

Japonsko je z hlediska robotického trhu na světě nespornou jedničkou. Japonsko bylo jedním z prvních států, který pochopil velký význam robotiky, a dnes jsou roboty v této zemi využívány v daleko větší míře než jinde. Na konci roku 2004 se v Japonsku používalo 356 500 robotů oproti 122 000 v USA. Japonsko je také největším světovým vývozcem robotů a země chce tento obor – svou „vlajkovou loď“, dále rozšiřovat prostřednictvím velkého počtu iniciativ, jako je program podporovaný ministerstvem hospodářství, *21<sup>st</sup> Century Robot Challenge*. Podle *Japan Robot Association* (JARA) má přibližně 130 japonských výrobců robotů velmi dobré výsledky ve třech hlavních oblastech: *průmyslová robotika, stavební robotika a servisní robotika*. Jen průmyslových robotů bylo počátkem 2008 na území Japonska více než 450 000, což je zhruba polovina celosvětového počtu. Na hannoverském veletrhu 2008 poutaly nejvíce pozornosti servisní roboty. Stánek s názvem „*Lidský život s humanoidy*“ představil dalšího úspěšného japonského **humanoidního robota HRP-2** pod přezdívkou **Promet** Robot o výšce 154 cm a hmotnosti 58 kg s 30 stupni volnosti obsahuje velmi kompaktní elektrický systém, takže nemusí mít na zádech „batoh“, jak lze vidět u mnohých jiných humanoidů, (viz dále ASIMO). Promet byl navržen tak, aby dokázal interaktivně spolupracovat s člověkem. Zatím chodí pouze po rovné ploše, má se ale brzy naučit zvládat i nerovný terén, zvednout se po pádu a začít s interaktivním chováním ve zcela volném prostoru.

<sup>17</sup> Nanotechnologie představuje perspektivní disciplinou, která si klade za cíl navrhovat a realizovat nejen nanoroboty a to v měřítcích  $10^{-9}$  metru, tedy velikostí srovnatelných s většinou molekul.

Japonci se však na *výzkum a realizaci humanoidních robotů* zaměřili již v 70. letech minulého století. První byl vyroben už v roce 1973, jmenoval se WABOT 1 a na rozdíl od těch současných ještě neuměl chodit - přenášel jen váhu z levé nohy na pravou. Byl tehdy vyvinut na tokijské univerzitě Waseda, kde výzkum humanoidních robotů probíhá úspěšně dodnes. *Waseda University Humanoid Robotics Institute (HRI)* má v rámci celosvětového vědeckého výzkumu robotů stále jednu z vedoucích pozic.<sup>18</sup> Na podzim 2008 byl světu představen nový humanoid s pracovním názvem WV-1. (*Waseda wheeled Vehicle-No.1*). Podrobnosti na Internetu, např. na stránce: <http://www.humanoid.waseda.ac.jp>

V roce 1986 ohlásila firma Honda start vývoje humanoidních robotů a již o sedm let později představila prvního humanoidního (již chodícího) robota P-1. V roce 1996 pak vyrobila robota P-2, který již nepotřeboval externí napájení a ovládání. Robot P-3 už uměl chodit do schodů. Konečně v roce 2000 spatřil světlo světa robot **ASIMO** (*Advanced Step in Innovative Mobility*), jenž je schopný s lidmi bazálně komunikovat. Robota ASIMO přivezl v r. 2003 do Prahy premiér Koizumi jako výraz úcty bratřím Čapkům. ASIMO tak mohl v budově Národního muzea v Praze „osobně“ položit květiny k pomníku Karla Čapka, spoluautora slova „robot“.

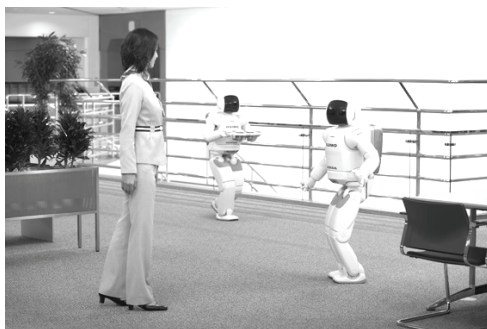
Prioritním cílem aplikovaného výzkumu robotiky jsou nyní v Japonsku *humanoidní roboti*. Kromě velkých firem typu Sony, Honda, Toyota apod. inteligentní roboty humanoidního typu dnes nabízí i mnoho dalších firem jako Matsushita Electric, Mitsubishi Heavy a Vstone. Humanoidní roboti mají pro Japonce velký význam, protože japonská populace rychle stárne, a právě roboti mají do budoucna převzít úlohu domácího společníka a pomocníka.

#### *Stručná charakteristika humanoida ASIMO*

Zmíněný první komerčně dostupný humanoidní robot ASIMO vznikl ve výzkumném středisku firmy Honda v japonském městě Wako. Ve své nejnovější 12. verzi má robot ASIMO celkem 34 stupňů volnosti (kloubů ovládaných servomotory), což mu umožňuje plynule se pohybovat a vykonávat v podstatě stejné činnosti jako člověk: chodí po rovině rychlostí až 6 km/h (Obr. 9.), běhá a tančí, chodí po schodech nahoru i dolů apod. Energii dodává 38.4V/10AH NiMH baterie, se kterou vydrží 30 minut. Je střední velikosti (výška 1,2 m) a vzhledu malého kosmonauta v bílém skafandru. Velmi dobře umí komunikovat s lidmi, dívat se jim do očí, pohybovat se vedle nich, podávat jim předměty denní potřeby, nápoje apod. Díky vestavěným vizuálním sensorům, laserům a infračidlům dokáže odhadovat vzdálenost překážek a patřičně přizpůsobovat své pohyby. K sensorové výbavě robota patří i dvě kamery, akcelerometr a gyroskop. Před překážkou stojící mu v cestě se zastaví nebo ji obejde. Mění volně směr pohybu, aniž by se před otočením musel zastavit. Dokáže rozpoznat obličej vybraných osob i rozlišit a interpretovat některá gesta (např. rozpozná podání ruky a opětuje je). Protože má sluch, otočí se ve směru, z něhož se na něj volá. Reaguje na vyslovení svého jména nebo na pozdrav, na povel se přemístí na určené místo a identifikuje určité známé

<sup>18</sup> Obdobnou celosvětovou prestiž má již v úvodu zmíněná Brooksova *Artificial Intelligence Lab, Massachusetts Institute of Technology (MIT)* - Humanoid Research Group.

hluky (pád předmětu, vzájemný náraz). Když do něj někdo strčí, umí obnovit svou rovnováhu, a je-li mu podán předmět, vezme si ho. V jeho batohu je ukryt výkonný počítač se složitým softwarem, který ovládá složitý mechatronický řídicí systém robota se zmíněnými 34 servomotory. Odborníci z Hondy nyní usilovně pracují na tom, aby vyvinuli pro robota ASIMO „mozek“, který by mu umožnil autonomně se rozhodovat. Za vzor si přitom berou uspořádání lidského mozku, jak se lze mj. dozvědět na webových stránkách firmy Honda: <http://ASIMO.honda.com>



Obr. 9. Servisní humanoidní roboti ASIMO.

## 5.2 Vývoj humanoidních robotů v Německu

Německo, které ve světě patří k předním výrobcům průmyslových a servisních robotů, nezůstává příliš pozadu ani ve vývoji robotů humanoidních. Prvním pracovištěm v Německu zabývajícím se vývojem humanoidních robotů byl Ústav pro technickou informatiku ITEC (*Institut für Technische Informatik*) univerzity v Karlsruhe. Zde byla již v roce 2000 vyrobena první demonstrační verze humanoidního robota s označením **Armar-I**, dosud používaná k ověřování nových funkcí robotů. Horní část těla (trup) robota Armar-I má celkem devatenáct stupňů volnosti a může se pohybovat na pojízdné plošině. Všechny jeho pohyby jsou generovány servomotory zapojenými v regulační smyčce s polohovou zpětnou vazbou.

Od roku 2001 jsou humanoidní roboti vyvíjeni v rámci výzkumného projektu SFB-588 *Humanoid Roboter – Lernende und kooperierende multimodale Roboter*, koordinovaného a řízeného Německou výzkumnou společností DFG (*Deutsche Forschungsgemeinschaft*) v Karlsruhe a naplánovaného až do roku 2012. Cílem je vyvinout humanoidního robota k všestrannému použití při zajišťování běžných činností v domácnosti. Na projektu interdisciplinárního charakteru spolupracují vedle třinácti ústavů univerzity v Karlsruhe také Fraunhoferův ústav pro informatiku a zpracování dat IITB (*Fraunhofer Institut für Informations- und Datenverarbeitung*) a Ústav pro vývoj produktů IPEK (*Institut für Produktentwicklung*) se sídlem tamtéž, Výzkumné středisko pro informatiku FZI (*Forschungszentrum Informatik*) a mnoho dalších pracovišť v Německu, přičemž každý z partnerů řeší určitý dílčí úkol. V první fázi výzkumného projektu bylo hlavním cílem dokončení vývoje další verze humanoidního



robota s označením **Armar-II**. Ten má, na rozdíl od svého předchůdce, již sensorovou hlavu vybavenou stereoskopickým kamerovým systémem, dvěma mikrofony a reproduktorem. Robot Armar-II je prostřednictvím kamerového systému schopen lokalizovat svého lidského uživatele a rozpoznávat objekty ve svém okolí. Mikrofony mu umožňují směrově slyšet a rozumět mluvené řeči. Přes reproduktory může dát najevo, že porozuměl povelu. Aby mohl manipulovat s předměty ve svém okolí, má Armar-II dvě ruce, každou s pěti ovládanými prsty.

#### *Zkušenosti uplatněné v nové generaci robotů*

Těžištěm projektu SFB 588 jsou ovšem zejména práce na vývoji nové generace demonstračního **humanoidního robota s označením Armar-III**, v níž jsou využity a zhodnoceny všechny dosud získané zkušenosti. Odpovědnost za vývoj kritických mechanických komponent nového robota převzal zmíněný IPEK. Pozornost je zde věnována nejprve paži robota, jednomu z jeho nejsložitějších pohybových systémů (každá paže má sedm stupňů volnosti). Cílem je zmenšit celkovou hmotnost paže a zlepšit její funkci. Klíčovou myšlenkou je realizovat paži robota bez akčních členů, tj. přemístit poměrně těžké pohonné jednotky do jeho trupu. Jde o účelné řešení, protože podobně jako u lidí jsou dynamické požadavky kladené na paže robota podstatně větší než požadavky na dynamiku trupu. Nakonec byla ale zvolena smíšená konstrukce s akčním členem pro ovládání zápěstního kloubu umístěným v předloktí. Vcelku se podařilo zmenšit hmotnost paže téměř o polovinu. Jedním z hlavních požadavků přitom bylo skloubit malou hmotnost s velkou přesností. Současně bylo z hlediska regulační techniky důležité zajistit, aby všechny pohyby bylo možné ovládat nezávisle na sobě, a navíc aby vedle polohy bylo vždy možné měřit i sílu a moment. Stejný princip byl použit také u nového krku, nyní se čtyřmi stupni volnosti, třemi ve spodním a jedním v horním segmentu krku. Kloub v horním segmentu umožňuje robotovi kývat „hlavou“, takže se robot Armar-III již dokáže dokonce i dívat dolů na vlastní „nohy“.

Robot Armar-III je také vybaven mnohem dokonalejší sensorovou hlavou, v níž jsou na místech očí nyní vloženy dva prostorové (stereoskopické) kamerové systémy, tedy celkem čtyři snímací kamery. Zatímco jeden kamerový systém je určen pro pohled do dálky, např. při lokalizaci člověka, druhý se dívá nablízko, tedy např. sleduje, jak jsou vykonávány požadované manipulační úkony. Uši robota tvoří soustava složená ze šesti mikrofonů umístěných nejenom na hlavě, ale i na trupu robota; to umožňuje spolehlivěji rozpoznávat polohu a rozlišovat různé zdroje zvuků, jako např. několik hovořících osob nebo vyzvánění telefonů. Tím byly také vytvořeny jednoznačně lepší výchozí podmínky pro rozpoznání řeči.

**Předpokladem úspěšného využití jakéhokoliv humanoida** jsou jeho zručnost a další univerzální dovednosti. Proto má robot Armar-III významně zdokonalené manipulační schopnosti. Jeho nové pětiprsté ruce s pneumatickým pohonem umožňují realizovat větší počet řízených hmatů. Jsou částečně potaženy citlivou „pokožkou“, která jim dovoluje vnímat informace o dotyku a úchopné síle a zajišťovat základní hmatové funkce robota. Pneumaticky poháněná umělá ruka se podobá lidské ruce jak svým tvarem a rozměry, tak i pohyblivostí, a díky jejímu měkkému povrchu je přímý kontakt mezi člověkem a robotem bez rizika. Aby humanoidní robot mohl úspěšně

vykonávat lidské činnosti, musí být dostatečně zručný a dovedný. K tomu ovšem nestačí jen dobrý hardware, ale nutný je také vhodný software. Jeho vypracování se ujali odborníci z Fraunhofer IITB.

### *Manipulace humanoida s velmi hladkými anebo křehkými předměty*

To je pro humanoida velmi náročný úkol. Aby takový úkol robot zvládl, musí mít nejenom určitý cit v konečcích prstů, ale současně je třeba programově zajistit vysoký stupeň koordinace pohybů obou jeho paží. Při přenášení hladkých předmětů, např. skleněné mísy, v obou rukou musí být přítlačná síla mezi prsty ruky robota a předmětem nastavena tak, aby předmět robotovi jednak nevyklouzl, ale také aby nebyl rozmačkán. K tomu účelu bylo třeba nejdříve vyvinout nový *senzor prokluzu*, který indikuje, zda ruka robota po držení předmětu neklouže. Podle údajů tohoto senzoru se reguluje síla působící na předmět. Jinou prověrkou dovedností robota je úkol spočívající v zasunutí klíče do otvoru v zámku nebo zástrčky elektrického spotřebiče do zásuvky. Podobně jako člověk, používá k tomu také robot své vizuální a hmatové (taktilní) senzory. Technicky jde o řešení problému typu *peg-in-hole* v několika fázích pohybu. Podle vizuálních informací se vkládaný objekt nejdříve navede do blízkosti otvoru. Následně se taktilním skenováním lokalizuje vstup do otvoru. Potom se za současného měření síly otáčí objektem tak dlouho, dokud ho nelze zasunout do otvoru při vynaložení minimální síly.

Sestavit pro řízení složitých pohybů robota klasický počítačový program vyžaduje velké programátorské znalosti a zkušenosti, které laický uživatel humanoidního robota nemůže mít. Proto chtějí odborníci z Fraunhoferova ústavu IITB ve spolupráci s pracovníky univerzity v Karlsruhe vypracovat *intuitivní poznávací (kognitivní) metodu programování robota na bázi učení*. Základní myšlenka spočívá v tom, že člověk většinou velmi dobře ví, jak by měl robot zadaný úkol splnit, protože to sám dokáže, a tudíž se jeví nejjednodušší, když mu to názorně předvede. V průběhu předvádění žádaného úkonu snímají senzory (stereokamery pro záznam obrazu, tzv. datové rukavice atd.) všechny pohyby člověka a předávají je jako sekvenci základních úkonů do počítače, kde jsou analyzovány a automaticky se z nich krok za krokem odvozuje flexibilní program pro řízení kinematiky robota. Celý proces není tak jednoduchý, jak se na první pohled zdá, a jeho dovedení do podoby použitelné v praxi bude ještě vyžadovat velmi mnoho programátorské práce.

K *dosažení nezbytné mobility* je demonstrační robot Armar-III upevněn na speciální mobilní plošině, která mu umožňuje pohybovat se lineárně v libovolném směru a při současném otáčení. K tomu účelu byl vyvinut nový kyčelní kloub robota se třemi stupni volnosti a velmi širokým rozsahem pohybu. Přimo nad ním se nachází trup robota, který má, stejně jako u lidí, za úkol spojit paže a hlavu s dolní částí těla. Podobně jako u člověka vnitřní orgány, je u robota v prostoru trupu umístěna řídicí elektronika. Nová mobilní plošina současně zásobuje celého robota energií a nese v sobě centrální řídicí počítač, který podle vloženého programu řídí všechny pohyby a další aktivity robota. Tuhá a ergonomicky uspořádaná struktura robota Armar-III byla navržena s použitím moderních optimalizačních postupů a je vyrobena z lehké slitiny na bázi hořčíku.

Výhledově se počítá s tím, že všesměrová mobilní plošina bude nahrazena dvounohým pohybovým systémem, který umožní robotovi chodit podobně jako člověk. Zde ovšem čekají na vývojáře zcela nové problémy spojené s řízením rovnováhy robota. Chůze na dvou nohách je v současnosti již technicky možná, ale složitá manipulace s objekty ještě není u dvounohých robotů SRN vyřešena. V tom mají němečtí vědci za svými kolegy hlavně z Japonska zatím zpoždění.

### 5.3 Humanoidní roboti společnosti PAL Robotics

Na podzim 2008 byl v Barceloně představen odborníkům a novinářům údajně *nejdokonalejší současný servisní robot - humanoid REEM-B* z dílny společnosti *PAL Robotics* ze Spojených arabských emirátů se sídlem v Barceloně. Robot REEM - B je zkonstruován s pomocí softwaru SolidWorks. PAL Robotics je pobočka společnosti PAL Technology Corporation zaměřená na vývoj autonomních robotů s plánovaným použitím jako *pomocníků v domácnostech, nemocnicích, nebezpečných prostředích* apod. PAL Robotics zvolili pro roboty – domácí sluhy humanoidní design, protože se prý tak údajně nejlépe přizpůsobují pro život lidí typickému prostředí domácností. Humanoidní design navíc umožňuje větší možnosti interakce robota s předměty a nástroji, které lidé běžně používají a mohou tak činit podobným způsobem jako oni.

#### *Stručná charakteristika humanoida REEM – B*

*Servisní roboti REEM-B* umí nejen rozvážně chodit, ale udrží rovnováhu i při nečekaném působení rušivých vnějších sil. Dokáží rozeznat rysy obličejů a zapamatovávat si důležité informace a situace, které se k dotyčným osobám vztahují. Se známými osobami si sami dokáží spojit jejich markantní charakteristiky, jednotlivá setkání a situace si zapamatovat apod. Roboti REEM si „uvědomují“ svoji polohu v rámci známého prostoru domácnosti a plně se zde orientují. Umí samostatně a velmi dobře hrát šachy. Model REEM – B dovede již průběžně vytvářet ale i nové mapy předem nepřipraveného okolí, zapamatovat si je a vyhýbat se pak i potmě překážkám. Mezi další schopnosti patří uchopení nákladu o hmotnosti do 13 kg, což představuje 20-25 % jejich vlastní hmotnosti. Model REEM-B je tak dnes jedním z nejsilnějších humanoidních robotů na světě, stojících na dvou nohách. V prostoru se orientují pomocí informací z kamer, infračervených čidel a sonarů. K manipulaci s nákladem jim slouží unikátní moduly článkových paží o velkém počtu volností a dálkové ovládání přes internet. Veškeré úkony robota mohou být řízeny hlasovými pokyny v angličtině. Dokáže sám také na ně slovně reagovat a komunikuje s okolím melodickým hlasem, vytvářeným syntetizátorem. Využívá předem vložené rozsáhlé slovní zásoby idiomů a dalších ustálených anglických frází.

Robot REEM-B má 40 stupňů volnosti, váží 60 kg a měří 140 cm. Na každé ruce má čtyři prsty a 10 motorů. To mu dává dobré motorické schopnosti. Současné baterie vydrží 3 hodiny plného provozu, plánuje se však mnohem déle. I tak je tato doba oproti současným robotům - humanoidům rekordní. Robot dokáže rukama jemně manipulovat a s nákladem 13 kg přitom kráčet rychlostí 1,5 km/h. Model REEM-B je dokonce schopen ad-hoc chodit do a ze schodů. Další jeho dovednosti se intenzivně vyvíjejí.

## 5.4 Nejmenší současný humanoidní robot světa

Humanoidní robot s názvem *i-Sobot*, kterého Guinnessova kniha rekordů označuje za nejmenšího vyráběného robota, byl na podzim 2008 na trhu dostupný jen za 99 amerických dolarů. Robot má předprogramovaných 180 samostatných pohybů, reaguje na slovní povely, vykonává až 240 pohybů v sledu za sebou s možností jejich naprogramování včetně pečlivého vyslovování slov, představování se, hraní na fiktivní kytaru či uctivých poklon publiku. Robot se programuje prostřednictvím dálkového ovládače, přičemž ho lze ovládat i pomocí joysticku. Humanoid *i-Sobot* rozpoznává 10 hlasových povelů, na které ví, jak správně reagovat. Jeho dílčí pohyby vytváří 17 krokových servomotorů. Gyroskop využívá na udržení rovnováhy. Robot umí stejně jako člověk hýbat hlavou, krkem, plecemi, rukama, nohama i chodidly. Vytváří tak přesvědčivý dojem přirozených lidských pohybů.

## 6 Budoucnost koexistence člověk – humanoid

### 6.1 Bezpečnostní rizika

*„Existuje pouze jediná okolnost, za které si dokážeme představit správce, jež nepotřebují podřízené, a pány, kteří nepotřebují otroky. Bylo by to pouze tehdy, pokud by každý stroj dokázal pracovat sám pouze na povel nebo chytře předvídat.“* Tato prorocká slova napsal Aristoteles v Obráně otroctví ve svém stěžejním díle „Politika“.

Umělý život usiluje o pochopení základních vlastností života. Jak tvořit živé systémy, jaké mají tyto systémy vlastnosti, zda se dají tyto vlastnosti aplikovat na některé inženýrské problémy. Umělý život by mohl objasnit průběh evoluce (*život-jaký-ho-známe*) a vznik inteligentního života na Zemi. Nesporně pozoruhodná jsou slova Karla Čapka pronesená při premiéře R.U.R.: *„Nesmíte si myslet, že vývoj, který dal vzniknout našemu životu byl jedinou vývojovou možností na této planetě.“*

Nevěřím, že umělá inteligence povede v důsledku svých aplikací, opírajících se o počítač, nutně k degradaci důležitosti člověka oproti výpočetní technice a sofistikovaným strojům obecně. Určitě se ale dočkáme protestů, někdy možná i násilných, proti technologiím, o nichž budou lidé přesvědčeni, že nás degradují. Že nás řadí do stejné kategorie jednoduchých, lidmi manipulovatelných výtvorů. Faktem je, že ve špičkových světových laboratořích, kde vědci sestavovali donedávna roboty především z „ocele a křemíku“, se už dnes sestavují první roboti z „ocele, křemíku a živých buněk“. Na předních pracovištích už umíme kultivovat svalové buňky, které pak v těchto zatím jednoduchých umělých mechanismech fungují coby ovládače a akční členy. Za několik málo let budou na stejném principu sériově vyráběny protézy, které budou zcela organickou součástí poškozených lidských těl. Také proces stále rychlejšího začleňování původně jen „křemíkových“ a „ocelových“ prvků do lidského těla už dávno začal. První pacienty, kteří si tyto technologie nechali implementovat, k tomu vedly klinické důvody. Snažili se prostě kompenzovat svá těžká tělesná postižení. Již dlouho máme kardiostimulátory a umělé kyčle (totální endoprotézy). Objevila se i umělá srdce, umělé ledviny, plíce, ale dnes se běžnou záležitostí stávají i

daleko složitější nervové protézy. Desetitisíce lidí s vážně poškozeným sluchem nosí v uchu (v části zvané kochlea – hlemýžď) natrvalo implantovaná zařízení. Ty generují až v 10 kmitočtových pásmech signály jakožto přímou nervovou stimulaci hlemýžďe, který tyto frekvence ve zdravém uchu normálně zachycuje. Tito lidé slyší prostřednictvím přímé elektronické stimulace svých periferních neuronů, přesněji prostřednictvím kombinace křemíkových a organických nervových spojení.

Existuje spousta dalších experimentů s implantováním „křemíkových a ocelových prvků“ lidem se zdravotními potížemi. Jsou vyráběny přístroje, určené k procvičování svalů pacientů po mozkové příhodě nebo s poškozenou páteří atd. Ukazuje se, že adaptabilita kritických oblastí lidského mozku je značná a to dává ochrnutým pacientům velkou naději do budoucna. Nepotrvá totiž dlouho a tyto zatím převážně klinické procedury se začnou uplatňovat v běžném životě. Rodney Brooks z MIT odhaduje, že již během příštích deseti až dvaceti let dojde k zásadnímu kulturně sociálnímu posunu. Lidé začnou postupně vnímat robotiku, informační technologie, křemík a ocel ve svém těle ne jako nucenou náhražku, ale jako cestu k lepšímu pochopení a dokonalejší interakci s okolním světem. Možná, že si pak i někteří vidomí, zcela zdraví lidé nechají do těla implantovat mechanismy, umožňující jim vidět i v infračerveném nebo ultrafialovém spektru. Existují futurologické představy, že svět bude jednou globálním superintelligentním kyberprostorem. Jeho zárodkem je dnešní Internet. Nevěřím, že by s ním mohl být člověk pro zvýšení svých intelektuálních schopností a znalostí již brzy bezdrátově propojen (přes rozhraní) přímo do mozku. Kdoví však, kam to posune *web sémantický*. Za zcela reálná pokládám obdobná vylepšování člověka biologického charakteru, profitující z očekávaného prudkého rozmachu genetického inženýrství, evoluční biologie a kvalitativního posunu v bio- a nanotechnologiích již v příštích desetiletích. Nárůst intelektuální výkonnosti naší civilizace bude pokračovat právě co nejužším spojením a vzájemným provázáním specifík inteligence člověka a umělé „inteligence stroje“ do výrazně efektivnějšího celku. Jsem proto přesvědčen, že lidská civilizace 21. století rozhodně nebude kopírovat jen zjednodušené vize autorů současných sci-fi příběhů a katastrofických filmů, kde dominantní postavení a nadvládu nad naším světem přebírají *kyborgové*, či nadlidsky výkonným umělým mozkem vybavení humanoidní roboti - k nerozeznání navenek podobní člověku. Tak chmurnou budoucnost líčí i prof. Warwick [15].

## 6.2 Ohrožení člověka inteligentními stroji

Samozřejmě nelze zcela vyloučit, že inteligentní stroje někdy ohrozí člověka. Z dnešního pohledu však neumíme objektivně posoudit, co by v budoucnu takovým „*ohrožením člověka*“ mohlo být. Vždyť lidé už tím, že proměňují své prostředí, mění také sami sebe a své postoje. Často se tak už nebojí toho, co dříve naši předci považovali za smrtelná nebezpečí, např. auta, letadla. A čím dál častěji své úhlavní nepřátele (divoké šelmy aj.) dokonce ochraňujeme. Tak jako jsme postupně novým dopravním prostředkům přizpůsobili legislativu i uspořádání sídlišť a krajiny, budeme si zřejmě zvykat i na mnohem komplikovanější a autonomnější stroje. Jejich vývoj prošel za posledních padesát let dlouhou cestu. Dnes již stavíme roboty s prvky umělého života. Roboti se vyvíjejí přibližně milionkrát rychleji, než jsme se vyvíjeli

my lidé. Ještě v polovině 90. let bylo jejich vědomí jen zhruba na úrovni hmyzu. To už je ale velký skok z pozice, kde byli před padesáti lety. Tehdy byli srovnatelní s jednobuněčnými organismy. Kdyby se i nadále vyvíjeli milionkrát rychleji než druh *Homo Sapiens* (a všechno tomu nasvědčuje), trvalo by jim to teoreticky už jen 50 dalších let, aby nás plně dostihli a byli nám svými schopnostmi ve všech oblastech rovni nebo dokonce nadřazení. Kevin Warwick říká, že „... *pro tyto inteligentní stroje budou lidé pak jen tím, čím jsou nyní zvířata pro mnohé z nás, tedy téměř bez významu*“ [15]. Nabízí se čapkovská otázka: Budou nás někdy roboti ovládat? Já si to nemyslím [10].

Přesto možnost, že tyto umělé organismy se stanou konkurenty lidí, zde jistě existuje, především díky tomu, že *každý organismus (i umělý) se snaží dosáhnout své cíle*. Evoluce umělých živočichů bude sledovat především cíle svého „živočišného druhu“ a ne cíle a očekávání svého „stvořitele“. Již dnes se pracuje na legislativě upravující pravidla komunikace s umělým světem<sup>19</sup>. Skutečně nebezpečné by pro nás mohly být zejména takové formy syntetického života, které s námi sdílí společný fyzický prostor (hardware, wetware aj.) a využívají ty samé zdroje, např. energetické (Havel, 2001). Dokonalé *kontrolní a bezpečnostní mechanismy* musíme mít proto k dispozici výrazně dříve, než vývoj ALife umožní takovým formám *samoreprodukci*, či dokonce *evoluci s otevřeným koncem*, což jsou vlastnosti všech živých systémů.

Ostatně už Norbert Wiener upozorňoval na možné riziko nevypnutelnosti budoucích kybernetických strojů. Zřejmě proto nikdy nebudou sestrojovány umělé bytosti, které by člověk nemohl sám vypnout. I když kupříkladu robot - voják, který by byl snadno vypnutelný by tím byl i velmi zranitelný. A jak by to dopadlo s jadernými elektrárnami, kdybychom připustili vypnutí jejich automatických řídicích systémů a všech systémů ochrany? Možná budou roboti nespolehliví právě proto, že po jejich vypnutí my, lidé, nebudeme už schopni plně nahradit jejich funkce [6,13,14]. Pomyslným klíčem od Pandořiny skříňky umělého života je nesporně jeho samostatný spontánní vývoj - *umělá evoluce*. S prof. Warwickem, s nímž nyní sedávám v jedné pracovně, souhlasím v tom, že „vyšlechtěním“ současného *Homo-Sapiens Sapiens* příroda evoluci zdaleka neskončila. S pochmurnou prognózou, vyjádřenou již titulem jeho knížky „Úsvit robotů – soumrak lidstva“ však souhlasit nemohu [15].

### 6.3 Prognóza dalšího vývoje inteligentní – humanoidní robotiky

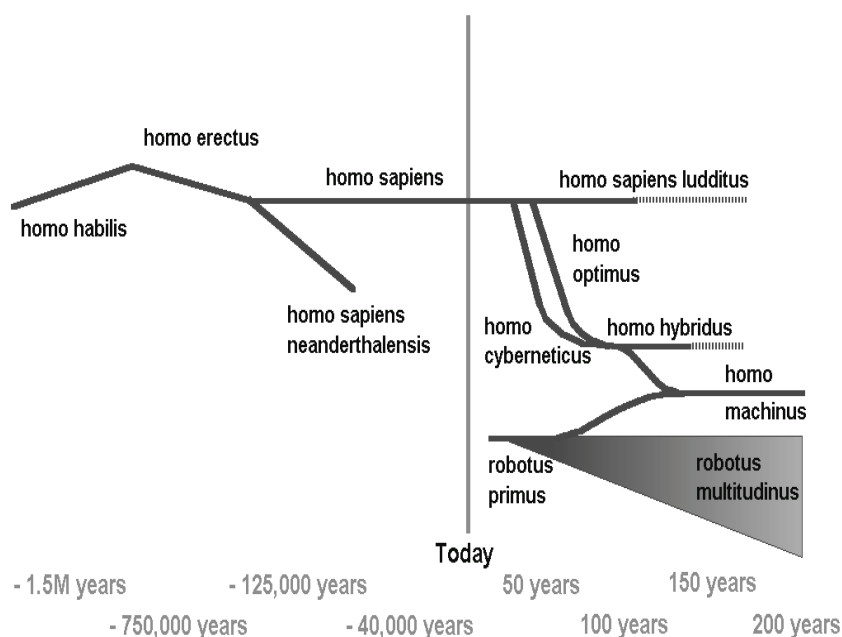
Prognostik Římského klubu profesor Ian Pearson ve své zprávě „The Future of Human Evolution“ pro Kongres USA odhadl (2005) vývoj v tomto směru v příštích 200 letech. Podle Pearsona (viz Obr. 10) je velmi pravděpodobné, že *nový biologický druh člověk/počítač* přežije pouze s podporou zkušeného staršího bratrance - člověka, jako mimořádně zdařilého produktu biologické evoluce.

V této souvislosti je ale na místě připomenout, že *vznik nového biologického druhu vůbec ještě neznamená, že tento druh bude v evoluci úspěšný*. Biologie zná mnoho příkladů: bílý žralok<sup>20</sup> je velmi starý a úspěšný produkt biologické evoluce,

<sup>19</sup> Zákaz výroby programů s implementovaným pudem sebezáchovy, tedy mj. nevypnutelných.

<sup>20</sup> Jeho evoluce se stejně jako u krokodýla téměř zastavila již před přibližně 130 miliony let.

který přežil ohromující počet mnohem mladších výstupů Darwinova mechanismu přirozeného vývoje. I v současnosti je však kolem nás řada příkladů, že inteligence a schopnost se úspěšně v neznámém prostředí adaptovat mohou být i v nepřímé úměrnosti. Rosomák je velmi inteligentní zvíře a je přitom na vymření, zatímco například vačice při testech není schopna realizovat ani některé jednoduché formy samoučení. Ale protože jejím základním rysem je žravost a vysoká plodnost, je rozšířena ve všech pro ni vhodných teritoriích.



**Obr. 10.** Evoluce inteligentních forem. (Podle Ian Pearson: “The Future of Human Evolution”).

Přesto, jak míní geniální prof. Stephen Hawking (2006), je prý již téměř jisté, že *budoucí vývoj lidstva budou určovat* ne dvě komponenty (člověk a robot), ale čtyři:

- 1) **člověk,**
- 2) **robot,**
- 3) **segment biosféry** ošetřený dodanou umělou inteligencí /pro komunikaci ap./ (vyšší savci, kytovci, hmyz a rostlinná říše) a
- 4) **umělý život** (např. právní systém, energetická soustava, telekomunikační a počítačové sítě nebo výroba potravin).

Je jasné, že *vytvoření dokonalejších lidí způsobí velké sociální a politické potíže*, protože zde stále budou lidé, kteří zdokonaleni nebudou.

I když jistě vesměs neobhajujeme genovou manipulaci lidí jakožto žádoucí vývoj, je vysoce pravděpodobné, že k ní postupně z různých důvodů dojde, ať ji již chceme či nikoli! Proto také nevěřím science fiction, jako jsou Hvězdné války (*Star Trek*), kde lidé v budoucnosti vzdálené 400 let jsou v podstatě zcela stejní, jako jsme my dnes.

Myslím si, že složitost lidského rodu a jeho DNA se bude poměrně rychle zvyšovat a především zrychlovat. Progrese posledních dvou století je toho důkazem.

Měli bychom proto připustit, že je velmi pravděpodobné, že k obdobnému vývoji dojde, a zvážit, co s tím uděláme [11].

Pokud si mají biologické systémy Homo Sapiens Sapiens udržet náskok před elektronickými - lidský rod potřebuje rychle a výrazně zkvalitnit své duševní a fyzické vlastnosti. Jen tak se člověk budoucnosti bude moct vyrovnávat se stále narůstající složitostí světa kolem sebe a plnit nestandardní zcela nové úkoly.

## 7 Závěr

Už v blízké budoucnosti dojde nesporně k fúzi biotechnologií, nanotechnologií a informačních technologií do jedné jediné disciplíny. Dá se očekávat masivní nasazování inteligentních robotů prakticky ve všech oblastech. Humanoidní roboti a jejich první účelné aplikace nejsou v současnosti již žádnou utopií, i když je jisté nezbytný ještě mnohaletý vývoj. Vedoucí v tomto směru jsou nyní pracoviště v Německu a Japonsku. V této souvislosti se uvažuje o nutnosti řešit i etické otázky týkající se zneužití robotů. Bylo by totiž naivní se domnívat, že nikdy nedojde ke zneužití robotů k útoku na jiného člověka. Vždyť člověk dokázal proti svému bližnímu obrátit snad každý svůj produkt – počínaje nástrojem a konče náboženstvím. A ve vyhledávání způsobů, jak použít to, co mělo být původně pro člověka a ne proti němu, je více než velmi kreativní. Dnes jsme svědky toho, jak armády ve válečných konfliktech nasazují „inteligentní“ robotické zbraně. Morálnost tohoto nasazení je odvozena pouze od morální oprávněnosti celého konfliktu. Ačkoliv dnešní inteligence robotů zdaleka nedostačuje pro posuzování morálních otázek jejich vlastního chování, pro likvidaci nepřítele již dostatečná je [14]. Nejhorší je, že armády jsou pouze „vrcholem ledovce“. Jak budou pronikat roboti do našeho života, budou vznikat další příležitosti k jejich zneužití. Už dnes nás začíná v této souvislosti zasahovat obecnější problém – jak si udržet dostatečnou úroveň soukromí a svobody života bez neustálého dohledu nejrůznějších plně automatizovaných a robotizovaných zařízení, kamerami a webem počínaje. Má to sice dnes ještě převahu výhod - znásilnění, loupežná přepadení a jiné zločiny jsou pro pachatele, snímané (pro ně nečekaně) kamerami v „přímém přenosu“ stále nebezpečnějším podnikem. Ale už brzy bude možné technicky sledovat člověka skutečně doslova všude, kam se hne<sup>21</sup>. Současný věk lze charakterizovat konstatováním, že *"zatímco stroje získávají stále více a více informací o lidech, mají lidé o strojích informací stále méně"*. Ano, takový je vývoj! Vždyť zatímco ještě v 70. letech měla přístup k výpočetní technice jen hrstka expertů, je nyní tato technika (PC, mobily atd.) plně v rukou uživatelů, kteří o principech jejího fungování skutečně již nepotřebují mít skoro žádné vědomosti. Nepotřebují tedy a zřejmě nikdy nebudou potřebovat velké množství detailních informací ani o robotech, které již dnes řídí. A

<sup>21</sup> Soukromí a svoboda bez obtěžování jsou základem „osvícených“ demokracií. Jsou-li tato práva porušena, záhy se objeví tyranie. Visí tak nad námi všemi stín Orwellova románu *1984*.



čím dál, tím více sofistikovaní roboti – (nejen humanoidi) budou prostupovat všemi oblastmi našeho žití.

To, že „*lidé mají o lidech díky používání strojů stále více informací*“ ... může být dobře i špatně. Záleží na způsobech využití takto nashromážděných informací.

Jsem ale přesvědčen, že to přispěje především ke zvýšení kvality našeho žití, včetně naší bezpečnosti, tolik žádoucí ve světě teroristických útoků a rasové a náboženské nesnášenlivosti. Věřím totiž, že se podaří nalézt *optimální kombinaci zákonů a zvyklostí*, která by účinně zabránila vládám i jedincům v nepřipustném nahlížení do našeho soukromí zneužitím techniky. Vždyť obecně vzato, sám robot nemá žádný primární důvod ublížit člověku. K tomu může dojít buď jen lidskou nedbalostí, nebo úmyslem. Skutečnou hrozbou lidstva nikdy nebyly stroje samy, ale lidská chamtivost, národnostní i náboženská nesnášenlivost a mocenský boj.

*Poděkování: Tato kapitola vznikla v rámci výzkumu a za podpory Projektu IM0567 CAK - Centra Aplikované kybernetiky a Katedry kybernetiky FEL ČVUT v Praze.*

## Literatura

- [1] Brooks, R. A.: Intelligence without Reason. In: L. Steels and R.A. Brooks, (eds) *The Artificial Life Route to AI*. Lawrence Erlbaum, Hillsdale, NJ, 1995, pp. 25–81.
- [2] Brooks, R. A.: Intelligence without Representation. In: *Artificial Intelligence* **47**, 1991, pp. 139–159.
- [3] CORDIS, Schaffung eines künstlichen Gehirns für Roboter. CORDIS Focus Newsletter, Brussel, September 2008, 282, p. 15.
- [4] Dierk, F.: *Etologie*. Karolinum, Praha: 2. vyd. 1996.
- [5] Franck, D.: *Verhaltensbiologie-Einführung in die Ethologie*. Thieme Verlag Stuttgart, 1985.
- [6] Hoffmann, K.: *Vorhersage und Erkenntnis*. Hogrefe: Göttingen, 1993.
- [7] Kadleček, D., Nahodil, P.: *New Hybrid Architecture in Artificial Life Simulation*. In: Kelemen, J., Sosík, P. (eds): *Advances in Artificial Life. ECAL 2001, LNCS Vol. 2159/2001*. Heidelberg, Springer-Verlag, 2001, s. 143–148.
- [8] Kelemen, J. a kol.: *Základy umelej inteligencie*. ALFA, Bratislava, 1993
- [9] Lorenz, K.: *Základy etologie*. 1. vydání, Academia, Praha 1993.
- [10] Nahodil, P.: Od jednoduchých robotů k anticipujícím stvořením. In: Mařík, V. a kol.: *Umělá inteligence 5*. Academia, Praha, 2007, s. 374 – 404.
- [11] Nahodil, P.: Svět umělého života a jeho vytváření. *Automatizace* **48** (5) (2005) 318–323.
- [12] Nahodil, P., Slavík, P., Kadleček, D., Řehoř, D.: *Dynamic Analysis of Agents Behaviour Combining ALife Visualization and AI - Invited Paper*. In: *LNCS. Volume 3071/2004*, Heidelberg: Springer Verlag. 2004, pp. 346–359.

- [13] Nahodil, P.: *From Slow Machines towards Anticipatory Creatures*. In: Mařík, V., Jacovkis, P., Štěpánková, O., Kléma, J. (eds): *Interdisciplinary Aspects of Human-Machine Co-existence and Co-operation*. - International Workshop „e-Golem“ to the World IFAC Congress, Prague: CTU Press, 2005, pp. 110–119.
- [14] Nahodil, P.: *From Intelligent Robots to Implementation of Artificial Life*. In: Proc. of CAL'02 – Cognition and Artificial Life, Milovy 2002, pp. 141–153.
- [15] Warwick, K.: *Úsvit robotů – soumrak lidstva*. Nakladatelství Vesmír, Praha 1999.
- [16] Wooldridge, M. J., Jennings, N. R.: *Intelligent Agents*. In *Lecture Notes in Artificial Intelligence*, LNAI 890/1995, Berlin: Springer-Verlag 1995, pp. 244–255.
- [17] Wooldridge, M.: *An Introduction to Multi-Agent Systems*. John Wiley & Sons, 2002.

# Evolučné algoritmy

Martin Pelikan<sup>1</sup>

**Abstrakt.** Evolučné algoritmy tvoria skupinu stochastických optimalizačných algoritmov, ktorých základný princíp je inšpirovaný evolúciou a genetikou. Asi najväčšia výhoda evolučných algoritmov v porovnaní so štandardnými metódami optimalizácie je to, že je pomerne jednoduché tieto metódy pochopiť, implementovať a používať v praxi, a napriek tomu sú schopné nájsť riešenia zložitých nelineárnych problémov v prijateľnom čase. Evolučné algoritmy tiež poskytujú priamočiary a účinný nástroj na simuláciu zložitých systémov. Cieľom tejto kapitoly je vysvetliť základné princípy evolučných algoritmov a ich použitia v praxi.

## 1 Úvod

Evolučné algoritmy používajú mechanizmy inšpirované evolúciou a genetikou na riešenie náročných problémov z oblastí optimalizácie, počítačového učenia, či simulácie zložitých systémov. V tejto kapitole sa budeme venovať hlavne ich použitiu v optimalizácii. Aj keď základy evolučných algoritmov boli položené už takmer pred pol storočím [11], najväčší rozmach v tomto odbore nastal až v posledných dvoch desaťročiach. Napriek tomu, že základný princíp evolučných algoritmov je pomerne jednoduchý, tieto algoritmy sú často schopné riešiť problémy prakticky neriešiteľné inými metódami. Aj preto sa čoraz častejšie objavujú aplikácie evolučných algoritmov v rôznych oblastiach od finančnictva [4] po inžiniersky dizajn [7,17], komponovanie hudby [14] a tvorbu portrétov podozrivých osôb v kriminalistike [13]. Cieľom tejto kapitoly je predstaviť základné princípy najznámejších druhov evolučných algoritmov, pričom najväčší priestor bude venovaný genetickým algoritmom.

Základná procedúra evolučného algoritmu je hlavným obsahom podkapitoly 2. Genetické algoritmy, ktoré sú v súčasnosti jednou z najpopulárnejších skupín evolučných algoritmov, sú predstavené v podkapitole 3. Evolučné stratégie sú náplňou podkapitoly 4. Genetické programovanie, ktoré umožňuje automatické programovanie pomocou evolučných algoritmov, je hlavnou témou podkapitoly 5. Niektoré z

---

<sup>1</sup> Missouri Estimation of Distribution Algorithms Laboratory (MEDAL), Department of Mathematics and Computer Science, 320 CCB, University of Missouri in St. Louis, St. Louis, MO 63121, USA, E-mail: pelikan@cs.umsl.edu

dôležitých aplikácií evolučných algoritmov sú predstavené v podkapitole 6. Podkapitola 7 prezentuje zhrnutie a záver.

## 2 Evolučný algoritmus

Pred vysvetlením základnej procedúry evolučného algoritmu je dôležité definovať problém, ktorý bude evolučným algoritmom riešený. Jeden spôsob definície optimalizačného problému je špecifikovať (1) reprezentáciu prípustných riešení problému a (2) metódu, ktorá umožní porovnávanie kvality dvoch alebo viacerých riešení. Cieľom je nájsť riešenie maximálnej kvality.

Množina prípustných riešení je zvyčajne definovaná ako priestor binárnych reťazcov alebo reálnych vektorov danej dĺžky. Je samozrejme možné použiť aj zložitejšie reprezentácie. Napríklad, v genetickom programovaní (viď podkapitola 5), riešenia reprezentujú programové kódy s danou syntaxou alebo matematické výrazy. Priestor všetkých možných riešení je zvyčajne príliš veľký na to, aby bolo možné ohodnotiť a porovnať všetky prípustné riešenia a preto je dôležité hľadať efektívne metódy, ktoré sú schopné nájsť najlepšie riešenia alebo riešenia im podobné po vyskúšaní len malého počtu potenciálnych riešení. Evolučné algoritmy predstavujú jeden z možných prístupov k riešeniu tohto problému.

Metóda na ohodnocovanie kvality riešení je zvyčajne založená na funkcii, ktorá pre každé potenciálne riešenie vráti reálnu hodnotu. Čím je táto hodnota vyššia, tým je riešenie považované za lepšie z hľadiska riešeného problému. Samozrejme, obdobne je možné považovať za lepšie riešenia tie, ktoré majú nižšiu funkčnú hodnotu. Pre mnohé evolučné algoritmy však nie je nevyhnutné, aby existovala vypočítateľná funkcia, ktorá umožní vyhodnocovanie kvality potenciálnych riešení. Riešenia môžu byť ohodnocované napríklad používateľom alebo skupinou používateľov, ktorí rozhodnú pre ponúkané dvojice alebo väčšie množiny potenciálnych riešení, ktoré z týchto riešení sú lepšie a ktoré horšie. Obdobne sa môže použiť experiment, v ktorom sú riešenia porovnané bez toho, aby kvalita týchto riešení musela byť vyjadrená konkrétnym reálnym číslom. Evolučné algoritmy sú tiež často aplikované na viackritériálne problémy (multiobjective problems), kde ohodnocujúca funkcia obsahuje niekoľko často až protichodných kritérií. Vo viackritériálnej optimalizácii je cieľom nájsť tzv. Pareto-optimálny front (Pareto-optimal front), ktorý pozostáva z riešení, ktoré sú navzájom neporovnateľné, lebo každé z týchto riešení je aspoň v jednom kritériu lepšie ako akékoľvek iné riešenie. Konečný výber riešenia je potom umožnený používateľovi, ktorý môže vyhodnotiť výsledky na základe ich konkrétnej podoby a iných výberových kritérií.

Terminológia evolučných algoritmov často čerpá v biológii. Potenciálne riešenia sú zvyčajne označované pojmom *jedinci* (individuals). Pre vektorové reprezentácie, jednotlivé premenné sú často označované ako *gény* (genes) a ich hodnoty ako *alely* (alleles). Hodnotiaca funkcia, ktorej hodnoty rastú s kvalitou riešení, je väčšinou označovaná pojmom *fitness funkcia* (fitness function) či *účelová funkcia*.

Evolučný algoritmus pracuje s populáciou jedincov. Táto populácia obsahuje obyčajne väčší počet jedincov, často stovky až tisícky. V tejto kapitole budeme veľkosť populácie označovať symbolom  $N$ .

Prvá populácia je zvyčajne vygenerovaná náhodne tak, aby pravdepodobnosť vygenerovania každého potenciálneho riešenia bola rovnaká. Táto populácia je následne obnovovaná pomocou troch základných operátorov:

**Výber.** Zo súčasnej populácie sa vyberie rodičovská populácia. Existuje niekoľko spôsobov výberu, ale základný princíp všetkých týchto metód je v tom, že riešenia vyššej kvality majú väčšiu pravdepodobnosť výberu a vyšší očakávaný počet kópií vo vybratej populácii rodičov. Jedna z populárnych metód je *výber založený na turnajoch* medzi jedincami (tournament selection). Na výber každého nového jedinca sa v tejto metóde najprv z populácie náhodne vyberie podmnožina  $k > 1$  jedincov a následne sa vyberie najlepší spomedzi nich. Procedúry výberu a turnaja sa opakujú až kým sa nevyberie dostatočný počet jedincov. Zvyčajne obsahuje populácia pred výberom a po výbere rovnaký počet jedincov. Hlavnou inšpiráciou pre operátor výberu je prirodzený výber podľa Darwina [6], kde najschopnejší jedinci prežívajú a majú potomkov s vyššou pravdepodobnosťou ako tí menej schopní.

**Variácia.** Variačné operátory (variation operators) sú následne použité na vytvorenie novej populácie jedincov (populácie detí) na základe populácie rodičov. Cieľom variačných operátorov je vytvoriť jedincov, ktorí kombinujú vlastnosti rodičov. Tieto operátory sú zvyčajne inšpirované genetickými operátormi odpozorovanými pri rozmnožovaní v prírode. Kríženie (crossover) môže byť použité na vytvorenie riešení, ktoré kombinujú časti riešení niekoľkých rodičov. Mutácia (mutation) môže byť použitá na náhodné zmeny rodičovských riešení.

**Nahradenie.** Na konci základnej iterácie evolučného algoritmu sa z pôvodnej populácie pred výberom a novovytvorenej populácie po aplikácii variačných operátorov vytvorí nová populácia riešení. Napríklad, pôvodná populácia môže byť úplne nahradená populáciou nových jedincov.

Kroky výberu, variácie a nahradenia sa opakujú s cieľom získať populácie neustále sa zlepšujúcich riešení. Na dosiahnutie tohto cieľa je dôležitá najmä kombinácia operátorov výberu a variácie. Operátor výberu upriami pozornosť na najlepšie riešenia zo súčasnej populácie. Variačné operátory vytvoria nové riešenia podobné týmto najlepším riešeniam. To samozrejme znamená, že operátory variácie musia byť schopné efektívneho využitia informácie v populácii rodičov na to, aby nové riešenia boli obdobne kvalitné ako rodičia, prípadne lepšie. Súčasne je dôležité, aby veľký počet nových riešení bol iný ako boli pôvodné riešenia. Cyklus výberu, variácie a nahradenia je zvyčajne označovaný ako jedna generácia alebo iterácia evolučného algoritmu. Evolučný algoritmus zopakuje množstvo takýchto generácií. V ideálnom prípade by mala populácia obsahovať riešenia postupne vyššej a vyššej kvality a výsledná populácia by mala obsahovať globálne optimum. Celkový počet generácií na dosiahnutie riešení požadovanej kvality sa mení v závislosti od konkrétneho algoritmu, problému, ako aj samotných požiadaviek na kvalitu dosiahnutých riešení, ale zvyčajne sa jedná o desiatky, stovky, tisícky, alebo aj viac generácií.

Cyklus obnovy populácie pomocou základných operátorov je prerušený po splnení stanovených podmienok. Napríklad, cyklus môže byť prerušený, keď bol dosiahnutý daný maximálny počet generácií alebo keď populácia neobsahuje dostatočne diverzifikovanú zbierku jedincov.

Základný evolučný algoritmus je zhrnutý v nasledovnom pseudokóde:

```

Evolučný algoritmus (EA)
  t := 0;
  náhodne vygeneruj prvú populáciu P(0);
  kým nie sú splnené kritéria pre zastavenie
  {
    S(t) :=vyber populáciu rodič. riešení z P(t);
    O(t) := aplikuj variačné operátory na S(t);
    P(t+1):= skombinuj P(t) a O(t);
    t := t+1;
  };

```

Rôzne evolučné algoritmy sa líšia najmä v reprezentácii a variačných operátoroch. Rozdiely medzi rôznymi skupinami evolučných algoritmov však často nie sú jasne definované a členenie evolučných algoritmov na podskupiny nie je vždy jednoznačné. Nasledovné podkapitoly sa venujú najpopulárnejším oblastiam evolučných algoritmov, a to genetickým algoritmom, genetickému programovaniu a evolučným stratégiám. Najviac priestoru bude venovaného genetickému algoritmu, ktorý bude slúžiť ako východisko pre zvyšok kapitoly.

### 3 Jednoduchý genetický algoritmus

V tejto podkapitole je predstavený jednoduchý genetický algoritmus (simple genetic algorithm) [22]. Podkapitola začína vysvetlením samotného jednoduchého genetického algoritmu. Činnosť jednoduchého genetického algoritmu bude potom znázornená na dvoch príkladoch. Podkapitola skončí načrtnutím teórie schém alebo šablón, ktorá je jedným z najpopulárnejších nástrojov na pochopenie činnosti genetických algoritmov a ich vlastností.

#### 3.1 Základný algoritmus

V jednoduchom genetickom algoritme [22] sú jedinci reprezentovaní binárnymi reťazcami danej dĺžky  $n > 0$ . Prvá populácia pozostáva z  $N$  náhodne vygenerovaných binárnych reťazcov.

Na výber populácie rodičov je použitý *proporcionálny výber* na základe hodnôt fitness funkcie (fitness proportionate selection). Táto metóda predpokladá, že hodnoty fitness funkcie sú pozitívne pre všetky potenciálne riešenia. Pravdepodobnosť výberu každého jedinca je priamoúmerná fitness hodnote tohto jedinca. Napríklad, ak má jedinec  $X$  fitness  $f(X)=6$  a jedinec  $Y$  má fitness  $f(Y)=3$ , pravdepodobnosť výberu jedinca

$X$ , ako aj očakávaný počet jeho kópií v populácii rodičov sú dvojnásobné v porovnaní s jedincom  $Y$ .

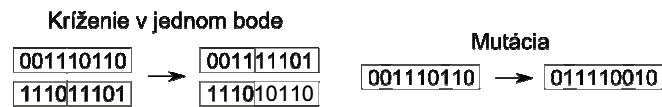
Na implementáciu proporcionálneho výberu je možné použiť algoritmus založený na analógii s ruletou. Pre každého člena  $X$  pôvodnej populácie  $P$  sa najprv vypočíta pravdepodobnosť jeho výberu,

$$p(X) = \frac{f(X)}{\sum_{Y \in P} f(Y)},$$

kde  $f(X)$  je fitness jedinca  $X$ . Každému jedincovi je potom pridelený úsek dĺžky  $p(X)$  na kružnici (rulete) s obvodom 1 tak, aby sa úseky pridelené rôznym jedincom neprekrývali. Keďže súčet pravdepodobností všetkých jedincov v  $P$  sa rovná 1, každý bod na obvode kružnice patrí jednému jedincovi z  $P$ . Jeden bod kružnice sa prehlási za počiatok (zvyčajne sa jedince nanášajú na kružnicu od počiatku v smere hodinových ručičiek). Pre výber každého nového jedinca sa najprv vygeneruje náhodné číslo  $r$  z intervalu  $[0, 1)$  s rovnomernou distribúciou na tomto intervale. Jediniec, ktorému prislúcha bod kružnice vzdialený  $r$  od počiatočného bodu v smere hodinových ručičiek, je potom pridaný do populácie rodičov. Náhodný výber sa opakuje, až kým je veľkosť populácie rodičov zhodná s veľkosťou  $N$  pôvodnej populácie. Očakávaný počet kópií každého jedinca je priamo úmerný jeho hodnote fitness a preto je pravdepodobné, že populácia rodičov obsahuje niekoľko kópií najlepších jedincov kým najhorší jedinci sa do tejto populácie nedostanú vôbec. Samozrejme, na výber rodičov je možné použiť aj iné metódy, ako napríklad vyššie uvedenú metódu založenú na turnajoch medzi jedincami.

Populácia nových jedincov je vytvorená z populácie rodičov pomocou operátorov kríženia a mutácie. Prvý krok pozostáva z kríženia, ktoré je aplikované na páry rodičov. Druhý krok pozostáva z mutácie, ktorá je aplikovaná na každého výsledného jedinca.

Z populácie rodičov obsahujúcej  $N$  jedincov sa náhodne vytvorí  $N/2$  rodičovských párov. Keďže ale proporcionálny výber používa tú istú distribúciu pre výber každého jedinca, páry rodičov môžeme v tomto prípade vyberať postupne, začínajúc s prvými dvoma jedincami a končiac s poslednými dvoma. Na každý z týchto párov sa s danou pravdepodobnosťou  $p_c$  (pravdepodobnosť kríženia) aplikuje operátor kríženia. Ako metódu kríženia dvoch jedincov, jednoduchý genetický algoritmus používa zvyčajne *kríženie v jednom bode* (one-point crossover). Kríženie v jednom bode začína náhodným výberom jednej z pozícií v reťazci reprezentujúcom jedincov. Rodičia si potom vymenia chvost svojich binárnych reťazcov začínajúci v tejto náhodne vygenerovanej pozícii. Keďže po výmene všetkých bitov zostávajú rodičia určite nezmenení, zvyčajne sa výber pozície zúži na všetky pozície okrem prvej. Príklad kríženia v jednom bode je znázornený na obrázku 1. Pravdepodobnosť kríženia,  $p_c$ , je jedným zo vstupných parametrov jednoduchého genetického algoritmu. Tento parameter je väčšinou v rozmedzí od 10% do 90% a jeho najčastejšie používaná hodnota je 60% na základe [10].



**Obr. 1.** Kříženie v jednom bode a mutácia na binárnych reťazcoch dĺžky 9.

Podobne ako v genetike, hlavná úloha kříženia je skombinovať informáciu obsiahnutú v rodičoch pomocou výmeny ich častí. Rodičia reprezentujú výber najkvalitnejších riešení daného problému. Ak je kříženie efektívne, malo by byť schopné s veľkou pravdepodobnosťou skombinovať tie vlastnosti rodičov, vďaka ktorým majú rodičia vysokú fitness.

Mutácia v jednoduchom genetickom algoritme zmení hodnotu každého bitu v danom reťazci s pravdepodobnosťou  $p_m$ . Pravdepodobnosť zmeny,  $p_m$ , je zvyčajne veľmi malá a preto je priemerný počet zmenených bitov zväčša 1 alebo iná malá konštanta nezávisle od konkrétnej dĺžky jedincov. Pre niektoré problémy je možné vypočítať optimálnu hodnotu tohto parametra [1] ale v praxi je najčastejším výberom  $1/n$ , kde  $n$  je dĺžka vektora (počet bitov). Príklad aplikácie mutácie je znázornený na obrázku 1.

### 3.2 Ručná simulácia jednoduchého genetického algoritmu

Pre jednoduchosť použijeme ako fitness funkciu tzv. *onemax*, kde je hodnota fitness vypočítaná ako súčet bitov v ohodnocovanom reťazci. Čím má reťazec viac bitov 1, tým je jeho fitness vyššia. Optimum (najlepšie možné riešenie) je reťazec s bitom 1 v každej pozícii. Inými slovami, fitness jedinca  $X = (X_1, X_2, \dots, X_n)$ , kde  $X_i$  označuje bit na  $i$ -tej pozícii tohto jedinca, je vypočítaná pomocou nasledovnej funkcie:

$$\text{onemax}(X_1, X_2, \dots, X_n) = \sum_{i=1}^n X_i.$$

Na výber rodičov použijeme metódu založenú na turnajoch medzi náhodne vybranými párami jedincov z pôvodnej populácie. Táto metóda je jednoduchšia ako proporcionálny výber a vo väčšine prípadov vedie k zvýšenej efektívnosti genetického algoritmu. Parametre budú v simulácii nastavené podľa tab. 1.

**Tabuľka 1.** Nastavenia parametrov v simulácii

parameter	vysvetlenie	hodnota
$n$	dĺžka reťazca	8
$N$	veľkosť populácie	10
$p_c$	pravdepodobnosť kříženia	0.6
$p_m$	pravdepodobnosť zmeny v mutácii	1/8

Prvý krok pozostáva z náhodného vygenerovania  $N=10$  binárnych reťazcov dĺžky  $n=8$ . Možným výsledkom tejto operácie je nasledovná populácia<sup>2</sup>:

<sup>2</sup> Ako zdroj náhodných čísel pre túto simuláciu bol použitý MATLAB.



	jedinec	fitness
1	10111001	5
2	00010010	2
3	01101001	4
4	00001001	2
5	00100110	3
6	00001010	2
7	11011000	4
8	11111000	5
9	11111000	5
10	00011001	3

Výber rodičov začína náhodným výberom 10 párov riešení z pôvodnej populácie. Z každého páru vyberieme najlepšieho jedinca na základe fitness funkcie (v prípade rovnakej kvality oboch riešení sa vyberie víťaz náhodne).

	turnaj	rodič (víťaz)	fitness
1	11111000, 00011001	11111000	5
2	00010010, 00011001	00011001	3
3	11011000, 10111001	10111001	5
4	01101001, 00001010	01101001	4
5	00100110, 11111000	11111000	5
6	00010010, 11111000	11111000	5
7	01101001, 10111001	10111001	5
8	00001001, 00010010	00010010	2
9	00100110, 11011000	11011000	4
10	10111001, 00011001	10111001	5

Už na prvý pohľad je zrejmé, že fitness hodnoty v populácii rodičov sú v priemere vyššie ako tie v pôvodnej populácii. Toto je následok tlaku na kvalitu jedincov.

V ďalšom kroku je na páry rodičov aplikované kríženie v jednom bode, pričom pravdepodobnosť kríženia je 60%. Keďže populácia desiatich jedincov obsahuje 5 párov, dá sa očakávať, že v priemere budú skrížené 3 páry jedincov. Na výsledných jedincov je potom aplikovaná mutácia, ktorá by mala v priemere zmeniť jeden bit, keďže  $n=8$  a  $p_m=1/8$ . Po krížení a mutácii vyzerá populácia nasledovne (bity vymenené krížením, ako aj bity zmenené mutáciou, sú zvýraznené hrubým písmom, v zátvorke je uvedená pozícia, od ktorej si rodičia vymenia hodnoty pri krížení):

	rodičia	po krížení	po mutácii	fitness
1	11111000 (3)	1 <b>1011001</b>	11111001	6
2	00011001	<b>00111000</b>	00111000	3
3	10111001	10111001	11111001	6
4	01101001	01101001	<b>00111101</b>	5
5	11111000 (2)	<b>11111000</b>	11110000	4

6	11111000	<b>11111000</b>	11111000	5
7	10111001	10111001	<b>10011011</b>	5
8	00010010	00010010	<b>10010010</b>	3
9	11011000 (6)	11011 <b>001</b>	1101 <b>0001</b>	4
10	10111001	10111 <b>000</b>	10111000	4

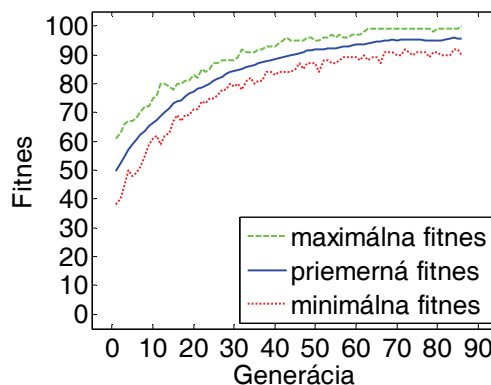
Po vytvorení nových jedincov pomocou výberu, kríženia a mutácie je pôvodná populácia nahradená populáciou nových riešení, ktorá slúži ako základ pre ďalšiu generáciu genetického algoritmu.

Ako by bolo možné kvantifikovať výsledky prvej generácie v našej simulácii? Jednou z možností je vypočítať priemernú fitness prvej populácie a porovnať ju s priemernou fitness populácie nových jedincov. Priemerná fitness pôvodnej populácie je 3,5. Priemerná fitness novej populácie je 4,5. Za pomoci výberu, kríženia a mutácie sme tak boli schopní vytvoriť populáciu lepších riešení ako bola tá pôvodná. Súčasne sa tieto riešenia vo väčšine prípadov odlišujú od pôvodných riešení. Boli sme preto schopní vytvoriť lepšie riešenia, ktoré sú síce podobné najlepším riešeniam v pôvodnej populácii, ale sú súčasne aj iné.

Simulácia ďalších generácií by mala viesť k ďalšiemu zvyšovaniu kvality našej populácie a po niekoľkých generáciách, populácia by mala obsahovať optimálny reťazec 11111111.

### 3.3 Simulácia genetického algoritmu na väčšom probléme

V tejto sekcii sú uvedené výsledky simulácie jednoduchého genetického algoritmu na funkcii onemax dĺžky  $n=100$ . Na výber jedincov je opäť použitý výber na základe turnajov veľkosti 2. Veľkosť populácie je  $N=100$ , pravdepodobnosť kríženia je  $p_c=0.6$  a pravdepodobnosť mutácie je  $p_m=0.01$ . Výsledky sú uvedené na obrázku 2.



**Obr. 2.** Výsledky simulácie genetického algoritmu na onemax dĺžky  $n=100$  bitov.

Výsledky ukazujú, že priemerná fitness populácie sa pomerne stabilne zvyšuje. Na začiatku je priemerná fitness blízko svojej očakávanej hodnote 50 (náhodne

vygenerované binárne reťazce dĺžky 100 by mali obsahovať v priemere 50 bitov hodnoty 1). Priemerná fitness potom rastie až na hodnotu okolo 95,7 v generácii 86. Postupný rast môže byť spozorovaný aj pre minimálnu a maximálnu fitness, aj keď tieto hodnoty sa v niektorých generáciách dočasne zhoršujú. V tejto simulácii, genetický algoritmus našiel optimálne riešenie s fitness 100 v 86. generácii. Ďalšie generácie by viedli k ďalšiemu zvyšovaniu kvality populácie, až kým by optimálni jedinci nezabrali takmer celú populáciu.

Keďže je v tomto príklade dĺžka jedincov  $n=100$ , celkový počet potenciálnych riešení je  $2^{100} \approx 1,27 \times 10^{30}$ . Optimálne riešenie bolo nájdené v 86. generácii, čiže celkovo bolo ohodnotených  $86 \times N = 8600$  jedincov (niektoré reťazce boli pravdepodobne ohodnotené niekoľkokrát). Celkovo bolo teda ohodnotených len  $6,78 \times 10^{-25}\%$  potenciálnych riešení a napriek tomu bolo nájdené globálne optimum.

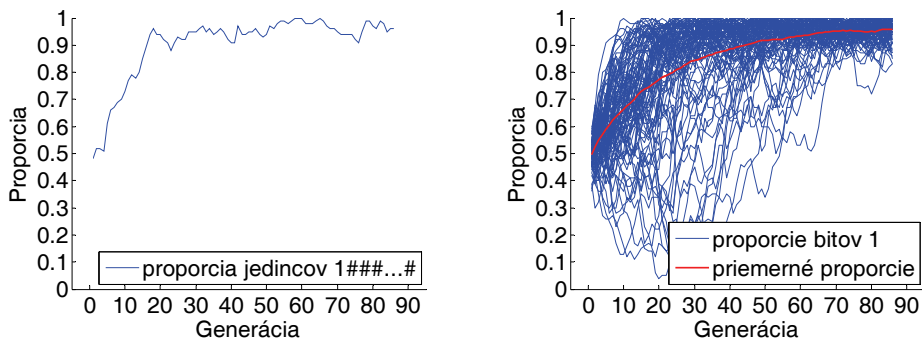
### 3.4 Schémy

Na vysvetlenie základných princípov fungovania genetického algoritmu sa často používa teória schém alebo šablón (schema theory) [11]. Pre binárne reťazce dĺžky  $n$ , schéma je definovaná ako reťazec tej istej dĺžky nad abecedou zloženou z troch symbolov:  $\{0, 1, \#\}$ . Každá schéma definuje podmnožinu binárnych reťazcov. Binárny reťazec  $X=(X_1, X_2, \dots, X_n)$  patrí do schémy  $H=(H_1, H_2, \dots, H_n)$  vtedy a len vtedy, keď je zhodný so schémou v každej pozícii, kde schéma obsahuje hodnotu 0 alebo 1. Napríklad, reťazce 01110 a 10111 patria oba do schémy ##11#, kým reťazce 10101 a 10000 do tejto schémy nepatria. Každý binárny reťazec dĺžky  $n$  patrí do  $2^n$  schém.

Jedným zo spôsobov charakterizácie populácie genetického algoritmu, ako aj dynamiky zmien v tejto populácii, je práve teória schém. Počet reprezentantov rôznych schém vyjadruje štruktúru populácie, kým zmeny v počte reprezentantov rôznych schém medzi jednotlivými generáciami vyjadrujú dynamiku zmien v populácii. Fitness hodnota schémy vyjadruje priemernú fitness hodnotu reprezentantov tejto schémy. Fitness schémy môže byť vypočítaná buď na základe rovnomernej distribúcie nad priestorom všetkých binárnych reťazcov danej dĺžky patriacich tejto schéme, na základe konkrétnej populácie takýchto reťazcov, alebo aj na základe inej distribúcie nad binárnymi reťazcami patriacimi tejto schéme.

Pre jednoduchý príklad použitia schém pri štúdiu genetických algoritmov sa vráťme k simulácii genetického algoritmu na funkcii *onemax*. Na pochopenie dynamiky zmien v populácii počas základného cyklu genetického algoritmu sa sústreďme na schémy 0###...# a 1###...#. Schéma 0###...# reprezentuje jedincov s 0 v prvej pozícii, kým schéma 1###...# reprezentuje jedincov s bitom 1 v prvej pozícii. Keďže počiatočná populácia bola vygenerovaná náhodne, predpokladaný počet reprezentantov každej z týchto schém je  $N/2$ , lebo očakávaný počet symbolov 0 a 1 na každej pozícii je  $N/2$ . Z definície výberu založeného na turnajoch je zrejmé, že jedinci s vyššou fitness by mali byť vybratí častejšie ako tí s nižšou fitness. Keďže jedinci patriaci schéme 1###...# by mali mať v priemere vyššiu fitness ako jedinci patriaci schéme 0###...# (lebo majú o jeden bit 1 viac a fitness je daná počtom bitov 1), počet reprezentantov schémy 1###...# by sa mal po výbere zvýšiť. Počas kríženia sa tento počet nezmení, kým počas mutácie sa tento počet zmení len málo. Preto môžeme

očakávať, že počet reprezentantov schémy 1###...# sa bude postupne zvyšovať. Analogická úvaha je možná aj pre ďalšie pozície v binárnych reťazcoch. Preto môžeme očakávať, že počet bitov 1 sa bude zvyšovať a s ním sa bude postupne zvyšovať aj fitness populácie, ako aj konkrétnych jedincov v populácii. Proporcie jedincov prislúchajúcim schéme 1###...# v našej simulácii genetického algoritmu na funkcii onemax dĺžky 100 bitov sú znázornené na obrázku 3. Dynamika zmien proporcií jednotlivých schém súhlasí s našou intuíciou a počet reprezentantov schém s hodnotou 1 na jednej pozícii sa postupne zvyšuje.



**Obr. 3.** Proporcía zástupcov schémy 1###...# na funkcii onemax dĺžky 100 bitov (naľavo) a obdobné proporcie pre ďalšie pozície v binárnych reťazcoch na tom istom probléme (napravo).

Existuje niekoľko teoretických modelov založených na schémach, ktoré sa snažia odhadnúť dynamiku zmien v počte reprezentantov rôznych schém v závislosti od vlastností konkrétnej fitness funkcie a operátorov [19]. Existuje tiež množstvo iných teoretických prístupov, založených napríklad na Markovových reťazcoch (Markov chains) [16], dimenzionálnych modeloch (dimensional models) [18].

### 3.5 Význam kríženia v genetickom algoritme

Sexuálna reprodukcia agentov populácie nie je jediný možný prístup k ich reprodukcii, existuje ešte možnosť tzv. asexuálnej reprodukcie, v ktorej len jeden agent - rodič vstupuje do procesu reprodukcie. Pomocou jednoduchého modelu ukážeme, za akých podmienok je sexuálna alebo asexuálna reprodukcia výhodná pre Darwinovu evolúciu. Predpokladajme, že funkcia  $f$  je separovateľná na dve nezávislé časti

$$f(\mathbf{x}) = f_1(x_1, x_2, \dots, x_j) + f_2(x_{j+1}, x_{j+2}, \dots, x_k)$$

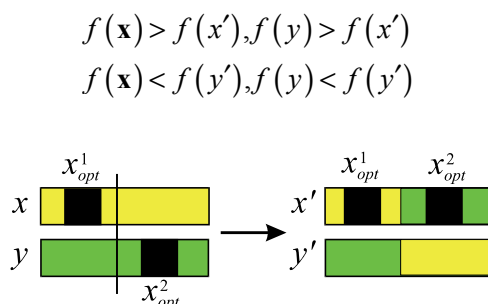
Optimalizačný problém pre funkciu  $f$  potom obsahuje dva nezávislé podproblémy

$$\mathbf{x}_{opt} = \mathbf{x}_{opt}^1 \oplus \mathbf{x}_{opt}^2$$

$$\mathbf{x}_{opt}^1 = \arg \min_{\mathbf{x}^1 \in \{a, b, c, \dots\}^j} f_1(\mathbf{x}^1)$$

$$\mathbf{x}_{opt}^2 = \arg \min_{\mathbf{x}^2 \in \{a, b, c, \dots\}^{k-j}} f_2(\mathbf{x}^2)$$

Význam kríženia pre takto separovateľný optimalizačný problém je znázornený na obrázku 4. Z tohto obrázku priamo vyplýva, že operácia kríženia môže podstatne urýchľovať evolúciu populácie, potenciálne môže spájať dve „podriešenia“ v jedno riešenie, ktoré obsahuje obe podriešenia.



**Obr. 4.** Význam kríženia pre separovateľnú účelovú funkciu. Operácia kríženia môže spojiť dve nezávislé riešenia – chromozómy v jedno riešenie, ktoré už je riešením oboch optimalizačných podproblémov.

V prípade, že účelová funkcia je separovateľná na dve alebo viac nezávislých častí, operácia kríženia zavádza do Darwinovej evolúcie *vnútorný paralelizmus*. Tento pojem, zavedený v teórii genetických algoritmov Hollandom [11], znamená, že v populácii chromozómov prebieha mnoho nezávislých optimalizácií, ktoré sú zamerané len na jednu časť účelovej funkcie. Operácia kríženia môže spájať tieto nezávislé „podriešenia“ na nové riešenia, ktoré už majú charakter riešení pokrývajúcich dva alebo viac optimalizačných podproblémov. Na základe týchto jednoduchých úvah môžeme pristúpiť k formulovaniu nasledujúceho zovšeobecnenia platného pre umelú (a snáď aj pre Darwinovu) evolúciu:

V prípade, ak sa gény v chromozóme dajú rozdeliť na dve susedné podskupiny, ktoré navzájom neinteragujú, potom *kríženie urýchľuje evolúciu*, t.j. hľadanie globálneho riešenia.

V opačnom prípade, ak gény v chromozóme nie je možné rozdeliť na dve susedné neinteragujúce skupiny, potom kríženie nie je dôležité pre urýchlenie evolúcie.

V 2. prípade (gény v chromozóme nie je možné rozdeliť na susedné neinteragujúce podskupiny) zahrnutie kríženia neurýchľuje evolúciu. Potom reprodukčný proces obsahuje len mutáciu pri kopírovaní otcovského chromozómu do chromozómu potomka.

V teórii evolučných algoritmov existujú dva „antagonistické“ tábory, ktoré sú charakterizované tým, či sú *za* alebo *proti* zahrnutiu kríženia do procesu reprodukcie. Prvý tábor reprezentovaný napr. Hollandom [11] (genetické algoritmy) verí, že kríženie patrí medzi základné črty evolučných algoritmov, jeho odstránenie redukuje evolučné algoritmy na slepé stochastické optimalizácie. Druhý tábor reprezentovaný napr. Fogelom [8] (evolučné programovanie) tvrdí, že zahrnutie kríženia do evolučných algoritmov nie je potrebné (kríženie je charakterizované ako gigantická mutácia). Podľa toho, čo sme si už povedali o význame kríženia v evolúcii, môžeme

povedať, že prvý (druhý) tábor má pravdu pre evolúciu definovanú s účelovou funkciou, ktorá je (nie je) separovateľná na neinteragujúce časti.

### 3.6 Iné reprezentácie a operátory v genetických algoritmoch

V tejto podkapitole sme použili binárne reťazce ako základnú reprezentáciu a predstavili sme niektoré základné operátory používané v prvých implementáciách genetických algoritmov. Samozrejme, existuje veľa problémov, kde potenciálne riešenia nie sú reprezentované binárnymi reťazcami. Jedným z najjednoduchších prístupov k riešeniu takýchto problémov je definovať zobrazenie medzi množinou potenciálnych riešení (prípadne podmnožinou týchto riešení) a priestorom binárnych reťazcov. Následne je možné použiť genetický algoritmus pre binárne reťazce, kde pred ohodnotením fitness každého jedinca sa tento jedinec najprv zobrazí do danej reprezentácie, a až potom sa na neho aplikuje fitness funkcia.

V súčasnosti však existuje množstvo verzií genetických algoritmov, ktoré sú schopné priamo pracovať s inými reprezentáciami, ako sú napríklad vektory reálnych premenných alebo reťazce premenlivej dĺžky. Bolo tiež navrhnutých množstvo rôznych operátorov výberu, kríženia, ako aj mutácie. Zvyšok tejto podkapitoly sa však bude venovať ďalším populárnym oblastiam evolučných algoritmov, a to genetickému programovaniu a evolučným stratégiám.

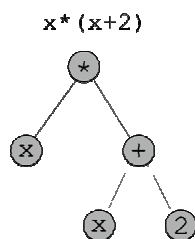
## 4 Genetické programovanie

Genetické programovanie [5,12,2] je v podstate genetický algoritmus aplikovaný na evolúciu *ohodnotených stromov* (labeled trees), ktoré zvyčajne reprezentujú programy s danou syntaxou alebo matematické výrazy nad danou množinou funkcií, premenných a konštánt. Základný rozdiel medzi jednoduchým genetickým algoritmom a genetickým programovaním teda spočíva v reprezentácii jedincov a v následných zmenách operátorov pracujúcich nad jedincami, konkrétne, v krížení a mutácii. V súčasnosti jestvuje množstvo rôznych prístupov ku genetickému programovaniu. V tejto kapitole sa však budeme venovať len tým najjednoduchším, podobne ako tomu bolo v prípade genetických algoritmov.

### 4.1 Reprezentácia jedincov

Jedinci sú v genetickom programovaní zvyčajne reprezentovaní stromami s ohodnotenými vrcholmi (labeled trees). Vnútorne vrcholy týchto stromov obsahujú funkcie z danej množiny prípustných funkcií, pričom podstromy definujú jednotlivé parametre týchto funkcií. Listy obsahujú tzv. terminálne symboly, ktoré zväčša reprezentujú premenné, konštanty alebo funkcie bez parametrov. Ako základný vstup je potrebné definovať množinu prípustných funkcií a terminálnych symbolov, ako aj ich interpretáciu. Syntax použitého jazyka a interpretácia jednotlivých ohodnotených stromov sú v prvom rade závislé od problému, ktorý chceme genetickým programovaním riešiť.

Ako jednoduchý príklad môžeme použiť aplikáciu genetického programovania pre optimalizáciu matematických výrazov s jednou reálnou premennou  $x$ . Jednou z možností pre výber funkcií je zvoliť množinu  $\{+, -, *, /\}$  a ako terminálne symboly použiť reálne čísla a symbol  $x$  pre túto vstupnú premennú. Príklad možného stromu vytvoreného z týchto symbolov je uvedený na obrázku 5.



**Obr. 5.** Príklad aritmetického výrazu reprezentovaného ohodnoteným stromom. Okrem samotného stromu je uvedený aj matematický výraz reprezentovaný týmto stromom.

Jednotlivé funkcie môžu zodpovedať konkrétnym príkazom v danom programovacom jazyku. Napríklad, funkcia môže reprezentovať príkaz *if* s tromi parametrami, pričom tieto tri parametre by mohli byť interpretované ako podmienka, výsledok v prípade splnenia tejto podmienky a výsledok v prípade nesplnenia tejto podmienky. Pre zložitejšie jazyky s možnosťami cyklov alebo podmienok je samozrejme dôležité, aby bolo možné akýkoľvek takýto program jednoducho interpretovať a otestovať pomocou fitness funkcie z hľadiska riešeného problému.

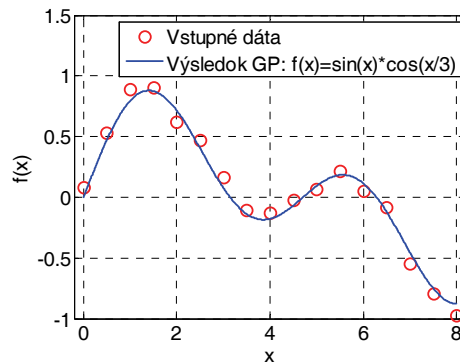
## 4.2 Fitness funkcia

Podobne, ako v genetických algoritmoch, aj v tomto prípade definuje fitness funkcia optimalizačný problém, ktorý by malo genetické programovanie riešiť. Pre každé potenciálne riešenie, ktoré je v tomto prípade reprezentované ohodnoteným stromom, vráti fitness funkcia hodnotu. Táto hodnota slúži ako číselné ohodnotenie kvality tohto stromu. Cieľom je nájsť ohodnotený strom najvyššej kvality.

Konkrétny tvar fitness funkcie závisí od problému, ktorý chceme riešiť. Jedným zo štandardných problémov riešených genetickým programovaním je *symbolická regresia*, kde jednotliví jedinci reprezentujú matematické výrazy podobné tomu, ktorý sme použili ako príklad ohodnoteného stromu v genetickom programovaní na obrázku 5. Na vstupe je daná množina dvojíc, kde každá dvojica špecifikuje jeden z možných vstupov, ako aj očakávaný výstup. Vstup väčšinou pozostáva z reálneho vektora a výstup pozostáva z jednej reálnej hodnoty. Cieľom je nájsť matematický výraz, definovaný na reálnych premenných zodpovedajúcim jednotlivým súradniciam vstupného vektora, ktorý čo najbližšie zodpovedá daným párom vstupov a výstupov.

Jeden zo spôsobov definície fitness funkcie pri riešení symbolickej regresie začína vyhodnotením matematického výrazu reprezentovaného ohodnocovaným jedincom pre každú vstupnú dvojicu. Fitness funkcia je potom vypočítaná na základe súčtu štvorcov

rozdielov medzi skutočnými a očakávanými výstupmi pre všetky tieto dvojice. Cieľom je minimalizácia tohto súčtu. Príklad výsledku dosiahnutého genetickým programovaním pre jednoduchý problém symbolickej regresie definovaný pre jednu vstupnú premennú je uvedený na obrázku 6.



**Obr. 6.** Príklad aplikácie genetického programovania na symbolickú regresiu.

Genetické programovanie sa často používa aj na riešenie problémov, kde stromy reprezentujú programové kódy v danom programovacom jazyku. Pre vypočítanie hodnoty fitness funkcie sa ohodnocovaný kód zvyčajne odsimuluje na množine daných príkladov a fitness funkcia je potom vypočítaná na základe dosiahnutých výsledkov a ich kvality vzhľadom na riešený problém. Každý program môže napríklad reprezentovať pravidlá, na základe ktorých sa môže agent rozhodovať, ako bude reagovať na rôzne senzorické vstupy konkrétnymi akciami a fitness funkcia môže byť vypočítaná na základe analýzy určitých modelových situácií poskytnutých používateľom. Ak napríklad chceme, aby sa agent vyhýbal prekážkam, fitness môže byť znížená pre každú kolíziu agenta s prekážkou počas simulácie.

Podobne, ako v iných evolučných algoritmoch, aj v genetickom programovaní môžu byť jedince ohodnocované vzájomným porovnaním na základe simulácie, či turnaja. Ak napríklad reprezentujú jedince stratégie na hru šachu, každá stratégia môže byť ohodnotená simuláciou partie s každou inou stratégiou v súčasnej populácii. Fitness hodnota stratégie potom môže byť určená ako počet vyhratých partii mínus počet prehratých partii.

### 4.3 Generovanie náhodných stromov

Generovanie náhodných stromov je potrebné na naplnenie prvej populácie stromov, ako aj na nahradenie náhodne vybratých podstromov pri mutácii. Stromy sa väčšinou generujú od koreňového vrcholu jednoduchým rekurzívnym algoritmom. Pre každý vrchol sa najprv náhodne vyberie jedna funkcia alebo terminál. Pokiaľ je výsledkom tohto výberu funkcia, jej parametre (podstromy) sú následne náhodne vygenerované pomocou rekurzívneho volania toho istého algoritmu. Generovanie končí v listových vrcholoch. Na obmedzenie zložitosti vygenerovaných stromov je možné určiť

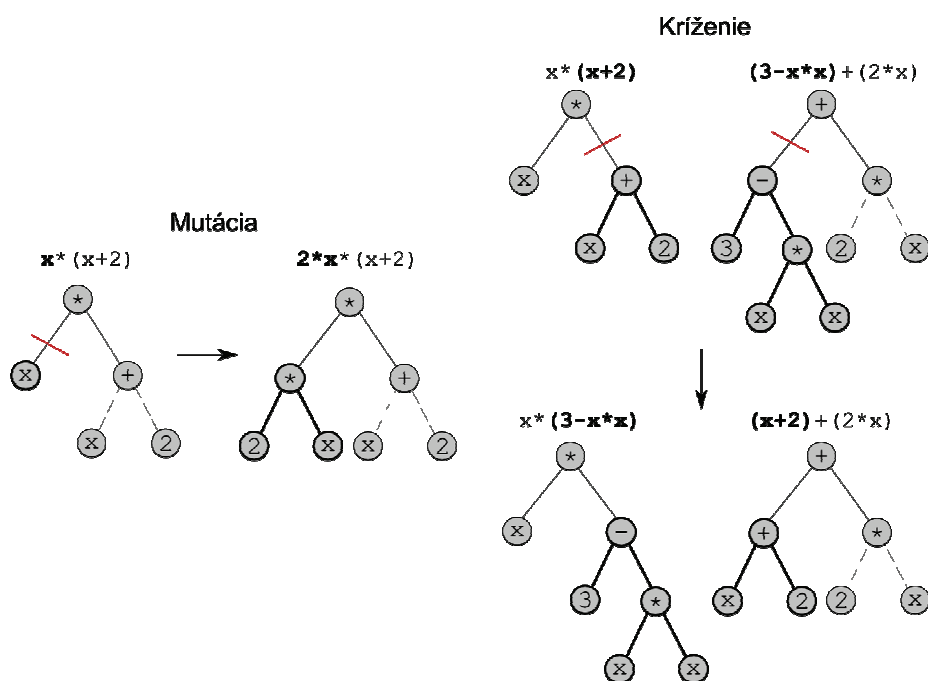


maximálnu hĺbku, čím sa každý vrchol v tejto maximálnej hĺbke obmedzí na terminálne symboly.

Pri generovaní prvej populácie stromov je nielen dôležité zabezpečiť, aby vygenerované stromy neboli príliš veľké (hlboké), ale aj aby počiatočná populácia obsahovala stromy rôznej hĺbky. Na toto je možné použiť variácie vyššie uvedeného rekurzívneho algoritmu.

#### 4.4 Kríženie a mutácia

Existuje množstvo prístupov k mutácii a kríženiu v genetickom programovaní. Princíp týchto operátorov je však rovnaký ako v tradičných genetických algoritmoch. Kríženie slúži na kombinovanie informácie obsiahnutej v rodičoch, kým mutácia slúži na menšie zmeny výsledných riešení. Obidva operátory sú zvyčajne aplikované s danou pravdepodobnosťou. Tu uvedieme azda najjednoduchšie z týchto prístupov [12].



**Obr. 7.** Kríženie a mutácia na matematických výrazoch v genetickom programovaní. Pre každý strom je uvedený aj matematický výraz, ktorý je týmto stromom reprezentovaný. Genetické programovanie samozrejme pracuje priamo so stromami a nie s výrazmi nimi reprezentovanými.

Jednobodové kríženie začína výberom náhodného vrcholu v každom z dvoch jedincov. Títo jedinci si následne vymenia podstromy začínajúce v týchto náhodne vybraných vrcholoch. Mutácia najprv vyberie náhodný vrchol stromu. Podstrom začínajúci v tomto vrchole je potom nahradený náhodne vygenerovaným stromom. Príklad kríženia a mutácie je uvedený na obrázku 7.

## 5 Evolučné stratégie

Genetické algoritmy a *evolučné stratégie* (evolution strategies) [20] sa líšia najmä v dvoch aspektoch. Kým v genetických algoritmoch sú jedinci zvyčajne reprezentovaní diskretnými vektormi pevnej dĺžky, evolučné stratégie pracujú zväčša s reálnymi vektormi. Ďalším rozdielom je to, že variácia v genetických algoritmoch väčšinou prikladá väčší dôraz na kríženie, kým v evolučných stratégiách je väčší dôraz kladený na mutáciu. Napriek tomu sú v súčasnosti pomerne bežné genetické algoritmy pracujúce s reálnymi vektormi a v oboch skupinách evolučných algoritmov sa bežne používa aj kríženie aj mutácia. Preto je delenie evolučných algoritmov na genetické algoritmy a evolučné stratégie hlavne výsledkom historického vývoja a hranice medzi týmito dvoma typmi evolučných algoritmov nie sú v súčasnosti jednoznačné. V tejto podkapitole predstavíme tzv.  $(1+\lambda)$ -evolučnú stratégiu alebo  $(1+\lambda)$ -ES, kde  $\lambda \geq 1$  je parameter určený používateľom.

$(1+\lambda)$ -ES pracuje s populáciou zloženou len z jedného jedinca reprezentovaného reálnym vektorom danej dĺžky. Počiatočný vektor je vygenerovaný náhodne. Každá súradnica počiatočného vektora je zvyčajne generovaná pomocou rovnomernej distribúcie na danom intervale.

Každá iterácia  $(1+\lambda)$ -ES začína vygenerovaním  $\lambda$  jedincov opakovaným použitím operátora mutácie na súčasné riešenie. Mutácia sa aplikuje na každú súradnicu osobitne. Pre každú súradnicu sa najprv vygeneruje náhodné číslo z normálnej distribúcie so strednou hodnotou 0 a štandardnou odchýlkou  $\sigma$ , kde parameter  $\sigma$  je vyjadruje *silu mutácie* (mutation strength). Toto náhodné číslo sa potom pripočíta k príslušnej súradnici. Každá súradnica prechádza mutáciou osobitne a nezávisle od ostatných súradníc. Po  $\lambda$ -násobnej aplikácii mutácie je vytvorených  $\lambda$  nových vektorov, ktoré sa síce líšia od pôvodného vektora, ale sú mu súčasne veľmi podobné, keďže sila mutácie je väčšinou pomerne malá.

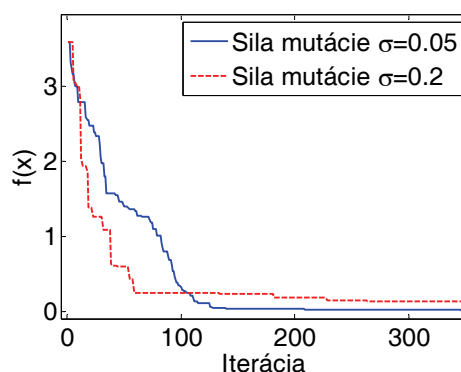
Po vytvorení  $\lambda$  nových vektorov porovnáme všetkých  $\lambda+1$  vektorov (pôvodný vektor a  $\lambda$  nových vektorov) na základe ich kvality z pohľadu riešenia daného optimalizačného problému. Ďalšia populácia je tvorená najlepším z týchto jedincov a iterácie sa opakujú, až kým nie sú splnené podmienky na ukončenie.

Jednou z najzaujímavejších oblastí výskumu v evolučných stratégiách je automatická adaptácia parametrov mutácie (adaptive mutation). Tu predstavíme jeden z prvých a súčasne najjednoduchších prístupov k adaptácii mutácie.

Základnou motiváciou pre adaptáciu parametrov mutácie je fakt, že optimálna veľkosť sily mutácie závisí najmä od vzdialenosti súčasnej populácie od optimálneho riešenia. Čím je táto vzdialenosť vyššia, tým by mali byť zmeny spôsobené mutáciou väčšie. Naopak, čím je populácia bližšie k optimu, tým by mali byť zmeny spôsobené mutáciou nižšie. Pre pochopenie tejto závislosti môžeme použiť jednoduchý príklad, kde je cieľom minimalizovať hodnotu nasledovnej kvadratickej funkcie nad  $n$  reálnymi premennými:

$$f(x_1, x_2, \dots, x_n) = \sum_{i=1}^n x_i^2 .$$

Výsledky simulácie (1+1)-ES na kvadratickej funkcii s  $n=10$  reálnymi premennými a dvoma hodnotami sily mutácie ( $\sigma=0.2$  a  $\sigma=0.05$ ) sú uvedené na obrázku 8. Z výsledkov je zrejmé, že silnejšia mutácia je výhodná najmä v prvej fáze evolučnej stratégie, keď je riešenie pomerne vzdialené od optima vo funkčnej hodnote 0 a zvýšená mutácia umožňuje rýchlejší postup k tomuto optimu. Naopak, ako náhle je riešenie bližšie k optimu, simulácia so silnejšou mutáciou začala stagnovať a slabšia mutácia bola schopná dosiahnuť riešenia vyššej kvality. Vo všeobecnosti by sa sila mutácie mala zvyšovať so vzdialenosťou od optima.



**Obr. 8.** Porovnanie (1+1)-ES na kvadratickej funkcii 10-tich reálnych premenných pre dve nastavenia sily mutácie.

Jednou z metód na automatické nastavovanie mutácie počas samotného behu evolučnej stratégie je 1/5 pravidlo (1/5 rule). 1/5 pravidlo zaznamenáva počet úspešných a neúspešných mutácií, pričom úspešná mutácia je tá, ktorá vedie k vylepšeniu riešenia. V určitých intervaloch (napríklad po každých desiatich iteráciách) sa vypočíta pomer  $p_s$  úspešných mutácií k celkovému počtu mutácií a na základe tohto pomeru sa upraví sila mutácie pomocou nasledovného pravidla:

$$\sigma' = \begin{cases} \sigma/c & \text{ak } p_s > 1/5 \\ \sigma \cdot c & \text{ak } p_s < 1/5 \\ \sigma & \text{ak } p_s = 1/5 \end{cases}$$

kde  $c < 1$  je reálna konštanta (zvyčajne sa používa  $c=0.8$  alebo  $c=0.9$ ). Toto pravidlo je úspešné najmä pre vyššie uvedenú kvadratickú funkciu, pre ktorú by mal byť optimálny pomer úspešných mutácií k celkovému počtu mutácií zhruba 1/5.

## 6 Niektoré úspešné aplikácie evolučných algoritmov

Existuje veľké množstvo zaujímavých aplikácií evolučných algoritmov, z ktorým spomenieme v tejto podkapitole iba niektoré.

Jedným z najznámejších príkladov úspešnej aplikácie genetických algoritmov v rozvrhovaní a plánovaní je firma Optimax, ktorej hlavným produktom bol program

určený na tvorbu plánov a rozvrhov založený na jednoduchom genetickom algoritme. Tento program bol napríklad použitý firmou John Deere, ktorá s jeho pomocou vyriešila rozvrhovanie výrobných liniek v piatich fabrikách. Firma Optimax bola v roku 1997 odkúpená firmou i2 za zhruba 60 miliónov amerických dolárov. Jestvuje množstvo ďalších úspešných aplikácií genetického algoritmu na rozvrhovanie. Napríklad, genetický algoritmus bol použitý aj na rozvrhovanie špeciálnych olympijských hier v Barcelone v roku 1992.

Vo finančnictve firma First Quadrant používa genetický algoritmus ako jeden z nástrojov na podporu riadenia investícií v hodnote zhruba 28 miliárd amerických dolárov.

Genetický algoritmus bol tiež použitý na zvýšenie efektivity prúdového motora známeho dopravného lietadla Boeing 777. Motor tohto lietadla bol vyrobený firmou General Electrics, ktorá je jedným z najstarších a najprominentnejších používateľov technológií založených na genetických algoritmoch v inžinierstve.

Evolučné algoritmy si našli svoju cestu aj do oblasti kriminalistiky, kde bolo navrhnuté použitie genetických algoritmov na náčrty osôb podozrivých z trestnej činnosti. Táto aplikácia je zaujímavá najmä tým, že kvalita jednotlivých náčrtov je ohodnocovaná svedkom, ktorý rozhoduje, ktoré z náčrtov vyzerajú presnejšie a ktoré naopak menej presne.

Evolučné algoritmy boli tiež použité pri tvorbe kompilátorov (Microsoft), návrhoch pevných a mobilných komunikačných sietí (Cox Associates), komponovaní džezových sól, či návrhu elektronických obvodov (napr. audio filtrov).

## 7 Záver

Evolučné algoritmy sú robustné stochastické optimalizačné algoritmy, ktoré s vysokou pravdepodobnosťou sú schopné nájsť globálne extrémny. Menovite v operačnom výskume a v diskretnej matematike poskytujú koncepčne jednoduché optimalizačné prístupy, ktoré nevyžadujú od riešiteľa hlboké preniknutie do matematických základov daného problému. Evolučné algoritmy sa stali v súčasnosti integrálnou súčasťou umelej inteligencie a kognitívnej vedy, kde reprezentujú jednoduchý numerický nástroj pre simuláciu Darwinovej evolúcie nielen v multiagentových systémoch, ale aj v teórii neurónových sietí pri evolučnom návrhu ich architektúry tak, aby čo najlepšie riešili požadované aktivity. V umelej inteligencii dokonca vznikol inžiniersky odbor nazývaný „výpočtová inteligencia“, ktorá je kombináciou troch „subsymbolických prístupov“, a to neurónových sietí, fuzzy logiky a evolučných algoritmov. Metódami výpočtovej inteligencie sme schopní riešiť inžinierske problémy v mnohých oblastiach aplikovanej umelej inteligencie prostredníctvom štandardných numerických metód. Tento nový vedný odbor v rámci umelej inteligencie sa stal alternatívou „klasickej“ umelej inteligencie, založenej na operáciách so symbolmi, ktorá dominovala do konca osemdesiatych rokov 20. storočia. V súčasnosti sa bežne používajú tak klasická umelá inteligencia, ako aj výpočtová inteligencia, často na riešenie rozdielnych typov problémov. Klasická umelá inteligencia vyniká pri hlbokom preniknutí do podstaty

problému spojeného s dlhodobým plánovaním, zatiaľ čo výpočtová inteligencia prejavuje svoje kvality pri riešení problémov v dynamicky sa meniacom prostredí.

## Literatúra

- [1] Back, T.: Optimal mutation rates in genetic search. In Forrest, S. (Ed.): *Proceedings of the 5th International Conference on Genetic Algorithms*. Morgan Kaufmann, 1993, pp. 2-8.
- [2] Banzhaf, W., Nordin, P., Keller, R. E., Francone, F. D.: *Genetic Programming – An Introduction*. Morgan Kaufmann, 1998.
- [3] Barricelli, N. A.: Esempi numerici di processi di evoluzione. *Methodos* **6** (21-22) (1954) 45-68.
- [4] Chen S.-H. (ed.): *Evolutionary Computation in Economics and Finance*. Physica-Verlag Heidelberg, 2002.
- [5] Cramer, N. A.: Representation for the Adaptive Generation of Simple Sequential Programs. In Grefenstette, J.: *Proceedings of the International Conference on Genetic Algorithms and their Applications*, Lawrence Erlbaum Associates, 1985, pp. 183-187.
- [6] Darwin, C.: On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life. London: John Murray, 1859 (Pôvod druhov. Kalligram, 2006).
- [7] Dasgupta, D., Michalewicz, Z. (Eds): *Evolutionary Algorithms in Engineering Applications*. Springer, 2001.
- [8] Fogel, L. J., Owens, A. J., and Walsh, M. J.: *Artificial Intelligence through Simulated Evolution*. New York: John Wiley, 1966.
- [9] Fraser, A. S.: Simulation of genetic systems by automatic digital computers. I. Introduction. *Aust. J. Biol. Sci.* **10** (1957) 484-491.
- [10] Goldberg, D. E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.
- [11] Holland, J. H.: *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI, 1975.
- [12] Koza, J. R.: *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.
- [13] Liu, J., Tang, X.: Evolutionary Search for Faces from Line Drawings. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **27** (6) (2005) 861 – 872.
- [14] Miranda E. R., Biles, J. A. (Eds): *Evolutionary Computer Music*. Springer, 2007.
- [15] Muhlenbein, H., Mahnig, Th.: Evolutionary algorithms: From recombination to search distributions. In Kallel, L., Naudts, B., and Rogers, A. (eds.): *Theoretical*

- Aspects of Evolutionary Computing*, Natural Computing, Springer Verlag, 2000, pp. 137-176.
- [16] Nix, A. E., Vose, M. D.: Modeling genetic algorithms with Markov chains. *Annals of Mathematics and Artificial Intelligence* **5** (1992) 79-88.
- [17] Parmee, I. C.: *Evolutionary and Adaptive Computing in Engineering Design*. Springer, 2001.
- [18] Pelikan, M., Sastry, K., Goldberg, D.E.: Sporadic model building for efficiency enhancement of hierarchical BOA. *Proceedings of the 2006 Genetic and Evolutionary Computation Conference*, 2006, 405—412.
- [19] Poli, R.: Recursive conditional schema theorem, convergence and population sizing in genetic algorithms. In Spears, W. M. and Martin, W. (eds.): *Proceedings of the Foundations of Genetic Algorithms Workshop (FOGA 6)*, Charlottesville, VA, USA, 2000, pp. 143-163.
- [20] Rechenberg, I.: *Evolutionsstrategie - Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Ph.D. thesis, Berlin Technical University, 1971.
- [21] Schwefel, H.-P.: *Numerische Optimierung von Computer-Modellen*. Ph.D. thesis, Berlin Technical University, 1974.
- [22] Vose, M. D.: *The Simple Genetic Algorithm: Foundations and Theory*. MIT Press, 1999.

# Funkční model mysli a jeho metodologický význam pro zkoumání operací lidského myšlení

Karel PSTRUŽINA<sup>1</sup>

**Abstrakt.** V kapitole se pokoušíme o vysvětlení toho, proč je přístup Top – down z metodologického hlediska výhodný pro zkoumání operací lidského myšlení. Speklativní funkční model mysli zahrnuje hlavní substrukтуры mysli a vztahy mezi nimi, z čehož vyplývají možné inspirace nejen pro vysvětlení operací lidského myšlení, ale také pro modelování těchto operací, jak se o to pokouší ti, kdož pracují v oblastech umělé inteligence.

## 1 Funkční model mysli a jeho metodologický význam

V mnoha ohledech je velmi obtížné formovat představu o celku mysli a o tom, jaké operace či funkce mysl a lidské myšlení vykonávají. Pokoušíme se o nárys tohoto modelu, neboť nás k tomu vede přesvědčení, že spekulace o mysli jako celku je jednou z vlastních úloh filosofie a že ostatně i všem ostatním pokusům o zmapování mysli musí již nějak spekulativní předtucha o celku mysli předcházet. Jen tak je možné interpretovat jednotlivá fakta zjištěná po způsobu vědy, nebo introspektivně vypořádaná; a jen tak je možné v mysli nalézat tušené nebo představované funkce, či nutnost tyto tušené nebo představované funkce opouštět jako chybné. Opačný (induktivní) přístup, tedy ten, kdy by byl realizován postup od jednotlivých faktických (experimentálně nebo statisticky) ověřitelných zjištění k rozpracování celku mysli, má sice své výhody v udržování se přísných faktů, avšak zároveň se zde naskytá mnoho úskalí. Především těch, kdy fakticita může zavádět k unáhlenostem a neopodstatněným generalizacím, nebo kdy se fakta stávají fikty. To se může projevit v soudobých snahách přístrojového zmocnění se mozku (prostřednictvím PET, nebo MgR metod), které ještě nemusí znamenat poznání lidského myšlení a mysli, případně i mnohých z počítačových simulací, které se mohou zabývat jen určitými operacemi, aniž by se dalších z operací lidského myšlení byt' jen dotkly. Tím však nechceme tyto metody a přístroje v žádném případě podceňovat. Domníváme se, že jsou neobyčejně významné a že nám poskytují velmi mnoho relevantních faktů. Filozofové však jimi nedisponují, a proto se pokoušíme o spekulativní a bezprostřední uchopení mysli a lidského myšlení.

---

<sup>1</sup> Vysoká škola ekonomická. W. Churchilla 4, 130 00 Praha 3, E-mail: Pstruzin@vse.cz

Uvědomujeme si také, že na počátku takového tázání po tom, co to mysl je a jak funguje, musí stát také otázka. Tedy to, jak se po něčem takovém jako je funkční model myslí tázat. Jde tedy nejen o průzkum myslí, ale také o možný způsob dotazování se na tento spekulativní průzkum.

Takováto otázka může být dvojaká.

Především taková, že poukazuje na možnou **techniku tázání** se tak, abychom dosahovali relevantních odpovědí. Druhá strana přístupu k tázání se po funkčním modelu myslí je tou, že tázáním se po myslí dopouštíme se toho, že chceme znát také odpověď na otázku, co to znamená mít fungující mysl; a proč zde něco takového jako je mysl vůbec jest. První z přístupů, tedy ten, který poukazuje na techniku tázání a vyúsťuje v popis, je tázání se po způsobu vědy (neboť si předem stanovuje hypotézu, kterou následně ověřujeme); druhý přístup je filosofujícím, nebo snad i filosofickým (protože tázání se po smyslu toho, co jest to mysl, je tázáním se po celku universa a postavení myslí v tomto celku).

Tázat se po funkčním modelu myslí je pak ještě komplikované tím, že se mysl táže sebe sama a na sebe sama, tedy že se chce dopátrat něčeho, co samo již předem a v sobě jest; co je nejen výsledkem, ale i předpokladem, tedy tím, co vůbec tázání umožňuje.<sup>2</sup>

Stojíme-li tedy na počátku našeho výkladu funkčního modelu myslí, musíme se nejprve omluvit za to, že tento model je jen verbální a nikoli graficky ztvárněnou představou, která by dávala snad názornější možnost náhledu. Doufáme však, že čtenář již přistupuje k jeho dotváření s jistou poučeností a že mu tudíž nebude zatěžko vytvářet duševní nárys souvztažností jednotlivých substruktur myslí. Pokud se budeme uchylovat k neumělým schématům, pak budou vždy jen dílčími a je zapotřebí vřadit je vždy do celku, v němž vévodí představa myslí. Na fragmentárnost těchto vizualizujících předtuch upozorňujeme také proto, že nedokážeme graficky zpřítomnit strukturnost myslí a pohybu lidského myšlení. Struktura myslí a také jejích částí (substruktur) je podle našeho názoru enkapsická a matrjožkovitá, a tudíž vícedimenzionální a dosud jsme nenalezli možnost, jak tuto mnohorozměrnost (překračující karteziánské souřadnice) ztvárnit.

Po této omluvě chceme také naznačit, jakým směrem se mohou ubírat naše úvahy o ontologickém statutu myslí, které vyplývají z naší koncepce funkčního modelu myslí. Jde sice o dvě různé otázky, neboť tážeme-li se po funkčním modelu myslí, ptáme se na to, co se v celku myslí děje; a otázka ontologického statutu zařazuje mysl do celku univerza, do celku všeho, co jest. Přes tyto otázky spolu souvisí. Tedy: **mysl je jsoucnem**. To je ontologický statut myslí. Je však takovým jsoucnem, které je médiem pohybu lidského myšlení a zároveň i nezbytnou podmínkou výskytovosti lidského myšlení. Mysl je něčím, co umožňuje pohyb lidského myšlení. Je něčím, kde jsou lidským myšlením konstituované intencionální obsahy ukládány; a zároveň je mysl i něčím, z čeho lidské myšlení vyzvedává mentály, když s nimi operuje.<sup>3</sup> Mysl tedy

<sup>2</sup> Toto je viděno mnohými (vědci) jako hlavní překážka možnosti myslí (nebo lidského myšlení) poznat sebe samu. Nedomníváme se však, že by to mělo v takovémto tázání a tím i úsilí po sebezpoznání zabránit.

<sup>3</sup> Ilustrativně si můžeme pomoci příkladem z čínského taoismu, kdy se ptali mudrce, co je to



nevymezujeme vůči mozku, ale ve vztahu k lidskému myšlení. Proto se nám také zdá, že tradiční Mind – Body problém je obtížně rozhodnutelný.<sup>4</sup> Z ontologicky zaměřeného pohledu tedy určujeme mysl jako jsoučno, a vše další již z toho vyplývá.

Zároveň je však možné říci, že lidské myšlení svým perpetuálním pohybem neustále dotváří konkrétní podobu jak mozku, tak mysli. Mozku tím, že každý pohyb lidského myšlení se v něm projeví a poněkud jej změní (tzv. plasticita mozku, zejména vytváření a změny synapsí); mysl tím, že ji lidské myšlení naplňuje intencionálními obsahy (mentály); a obou (mozku a mysli) tím, že lidské myšlení využívá specifiky jejich strukturního uspořádání pro své pohyby. Mysl a mozek tedy nejsou ve vzájemném vztahu kauzace nebo priority, ale obojí (mind i body) jsou ve vztahu k pohybu (tedy k lidskému myšlení). Tato jsoučna jsou tedy v závislosti na lidském myšlení stejně tak, jako lidské myšlení je závislé na nich. Lidské myšlení je tedy třetím členem roviny (rovnice) úvah o mind – body problému.<sup>5</sup>

Co to však znamená pro omezenost našich možností vizualizovat funkční model mysli, nebo jiným způsobem jej učinit blízkým pro uchopení?

Podle našeho plánu výkladu především to, že musíme výrazně rozlišovat mezi: operacemi lidského myšlení na jedné straně; a intencionálními obsahy mysli, s nimiž lidské myšlení operuje na straně druhé.

To však není jen jediná obtíž. Další je ta, kterou bychom mohli vymezit jako rozlišení mezi:

operacemi, kterými jsou získávané stimuly při projekcích lidského myšlení mezi jsoučna;

operacemi lidského myšlení při konstitucích věcí a mentálů; jakož i

operacemi, které lidské myšlení využívá při svých reakcích na již konstituované věci a mentály, tedy při konstruktivních pohybech.

Taktéž pak musíme začít rozlišovat mezi:

intencionálními obsahy mysli na úrovních konstitucí; a

na úrovních konstrukcí, tedy tehdy, kdy jsou splněny podmínky výskytovosti věcí a kdy z těchto věcí lidské myšlení

konstruuje předměty, nebo

z mentálů vytváří dílo.

Tím však výčet obtíží nekončí, neboť konstituce a konstrukce (tedy tvorba) mají další důsledek v tom, že při nich vznikají také epifenomény, které mají podobu sedimentů a ovlivňují další průběhy lidského myšlení. Tyto sedimenty nejsou předem zamýšlenými produkty, nejsou tedy záměrné. Nejsou také cílem konstruktivních

---

dům, a on odpověděl: „To nejsou zdi, ani dveře, ani okna, ..., ale dům to je to prázdno, které je obehnáno zdmi, okny, dveřmi.“ Obdobně je tomu i s myslí, i ona je tím prázdňem, v němž je možný pohyb lidského myšlení.

<sup>4</sup> Ostatně o tom se může každý čtenář přesvědčit ve spoustě odborných publikací. Za všechny uvádíme alespoň dvě. Srovnej:[1, 21]

<sup>5</sup> Domníváme se, že třetí člen je vždy zapotřebí hledat a nikoli se jen omezovat na protipóly.

pohybů lidského myšlení, a když tak až druhotným. To znamená tehdy, když lidské myšlení pochopí jejich užitečnost pro své další konstitutivní, nebo konstruktivní pohyby.

Rozvrh naší práce pak spočívá především v tom, že ústředním motivem celého modelu činíme lidské myšlení.<sup>6</sup> Nedospíváme tedy k němu, ale stavíme jej jako základ něčeho, co všemu ostatnímu v mysli dává vzniknout. Tím nechceme odvrhnout jednotlivé, pro lidské myšlení předpřipravené cesty, či strukturní moduly mysli, v nichž se pohybuje a které je nesnadné zjistit. Chceme jen zvýraznit to, že bez pohybu lidského myšlení by také vše ostatní, co se v mysli odehrává, bylo mrtvé, nepohyblivé. Operace lidského myšlení považujeme tedy za rozhodující, protože podle toho jaké operace jsou užívány, jaký strukturní pohyb lidské myšlení vykonává, podle toho dosahuje také odlišných intencionálních obsahů. Předem pak poznamenáváme, že operace lidského myšlení jsou převážně početními operacemi, neboť právě prostřednictvím výpočtu je umožněna výskytovost věcí (právě tak jako jsouceni).

A totéž, co zde proklamujeme pro lidské myšlení, je také zapotřebí uplatnit i vůči médiu, v němž se lidské myšlení jedinečně může pohybovat, tzn. vůči mysli. Také mysl je pro nás východiskem a nikoli něčím, co je druhotným. Obtížnější tedy než objasnit lidské myšlení nebo mysl je objasnit svět věcí (řekněme - přírodu). Lidské myšlení i mysl jsou východiskem, a tudíž předpokladem věcí (nikoli však jsouceni), my však upíráme pozornost na lidské myšlení a mysl, neboť se snažíme objasnit pravidla, zvláštnosti či odchylky, podle nichž je svět věcí a mentálů konstituován. Snažíme se přitom, abychom neopomenuli žádný poznatek, který je zjištěn o mozku a aby vše uvedené o lidském myšlení a mysli nebylo v rozporu s těmito poznatky.

Lidské myšlení definujeme jako čistý perpetuální pohyb, který operuje s počítky, vjemy, pojmy, představami, tóny, symboly, čísla atd., tzn. s intencionálními obsahy mysli, avšak vždy s něčím; a samo je **čistým pohybem**, který tyto počítky, vjemy, pojmy, představy, tóny, symboly, čísla atd., tedy intencionální obsahy mysli **unáší, smíchává je, nebo je rozděluje**, avšak je **za nimi** jako něco **samostatného**.

Snad není zapotřebí dodávat, že toto vše se odehrává v lidské mysli.

Domníváme se, že oproti filosofické tradici jsme zde dodali: *nebo je rozděluje*. Lidské myšlení se tradičně chápe jako něco syndodizujícího, nebo syntetizujícího, avšak domníváme se, že i diferenciacní pohyby ústící v rozlišení a rozdělení intencionálních obsahů mysli jsou významnými pohyby lidského myšlení.

<sup>6</sup> Zde je nutné zdůraznit, že hovoříme-li o funkčním modelu lidské mysli, nebo o lidském myšlení, máme na mysli naši mysl (mou osobní mysl), naše myšlení (mé osobní myšlení). Každý z nás je v tomto ohledu uzavřen v sobě a může vydávat svědectví jen o sobě. To je opět nevýhodou spekulativního, introspektivního přístupu k mysli, který je obtížně generalizovatelný. O ostatních lze soudit jen z našich projekcí mezi jsoucna, z toho jaké stimuly si přinášíme, na jaké reakce narážíme. Jen takto je možné pro nás vybudovat síť vztahů, z nichž vysuzujeme něco s obecnější platností. Teprve výklad o tomto (vždy mém) modelu mysli dává předpoklad k tomu, abychom o mysli mohli začít uvažovat jako o tom, co Husserl nazývá *intersubjektivně identickou věcí*, a kterou označuje jako *konstitutivní jednotu vyššího řádu*, která je vázána na *otevřenou pluralitu subjektů, které jsou navzájem v souladu*. Srovnej: [13] s. 315

Nejprve je snad nutné zvýraznit, jak pojmáme *onen čistý pohyb, který operuje vždy s něčím, avšak je za tím vším jako něco samostatného*. V této souvislosti by bylo možné uvažovat buď tak, že:

lidské myšlení není ničím jiným, než pohybujícími se tóny, představami, pojmy atd., které se vzájemně smíchávají a dávají další intencionální celky; nebo tak, že

lidské myšlení je jakousi mentální energií, která tóny, čísla, pojmy atd. unáší a syndodizuje je, nebo je syntetizuje (podle pravidel daných strukturou mysli a mozku) do intencionálních celků.

První z pojetí váže lidské myšlení na konkrétní singularity a nevymezuje mu nic, co by bylo jeho samostatností. V druhém pojetí je pak nutno uvažovat o tom, jak na sebe lidské myšlení může vázat všechny tyto rozmanité singularity, jakými jsou tóny, čísla, pojmy, představy atd., jak je k sobě nalepuje a jak je zase od sebe uvolňuje a jakou roli hraje při jejich smíchávání, či syntéze.

Domníváme se, že lidské myšlení je čistým proudem, avšak takovým, který může nabývat nejrůznějších proměn a být tóny, čísly, pojmy, představami atd., které jsou takto pohybujícími se a smíchávajícími se, nebo rozdělujícími se. Čistým proudem je tím, že je tím vším, co se takto pohybuje, a proto je za nimi jako něco samostatného, i když neustále prochází metamorfózami, tzn., že má vždy konkrétní podobu (není tedy ničí prázdným); a navíc, je samostatným, protože se pohybuje podle pravidel, kterými jsou operace a vzorce pohybu lidského myšlení. Tyto operace se mohou odlišovat podle toho, s čím právě lidské myšlení operuje, ale v zásadě je možné operace lidského myšlení charakterizovat tím, co při nich lidské myšlení mluví.

Další rozvržení našeho nynějšího pohledu na mysl jsme rozvažovali tak, aby zcela vyhovovalo různým možným představám a zdůvodněním toho, proč zde něco takového jako je lidská mysl vůbec jest (?), jak si lze vysvětlit její jestvování? Názorů a pohledů, které se snaží odpovědět na tyto otázky je nepřeborně, počínaje názory kreacionistů, a konče evolucionisty, nebo těmi, kteří mysl považují za nevyřešitelnou záhadu. Tuto otázku se nesnažíme odpovědět. Jen jsme brali v úvahu její položení a způsoby různých odpovědí na ni a snažili jsme se vytvořit si představu o mysli tak, aby nebyla v přílišném rozporu s nimi. V ohnisku našeho přístupu je tedy funkce mysli, nikoli její vznik. Ten je ostatně záležitostí myšlení, neboť to svým pohybem určuje jsoucná (a mysl je jsoucná).

Po těchto stručných vysvětleních našeho metodologického přístupu započneme další úvahy o funkčním modelu mysli.

## 2 Funkční model mysli - úroveň vnímání a konstitucí světa

Úvahy o funkčním modelu mysli započneme projekcemi lidského myšlení mezi jsoucná a vytrháváním stimulů ze jsoucen s následnými konstitucemi věcí na jsoucnech a vněmů v mysli.<sup>7</sup>

<sup>7</sup> V této souvislosti stojí za pozornost následující citát z díla Merleau-Pontyho: Reflexivní filosofie vychází z principu, který říká: má-li vněm být mým vněmem, musí být vždy již jednou z mých „představ“; jakožto „myšlení“ jsem ten, kdo uskutečňuje spojování různých aspektů, v nichž se mi prezentuje objekt a kdo je syntetizuje v určitý jednotný předmět. ... Já,

Jak již z předchozích pasáží vyplývá, naše představa je taková, že:

lidské myšlení se projikuje mezi jsoucná a rozprostírá mezi ně receptory, které vytrhují jednotlivé stimuly:  
 jsoucny je také biologický substrát, na který je lidské myšlení vázáno vazbou (tedy tělo, nebo mozek);  
 ze stimulů pak lidské myšlení na jsoucnech konstituuje věci a v myslí mentály (většinou jako vněmy, neboť jsou vnímány);  
 projekce lidského myšlení mezi jsoucná nejsou libovolné, ani nahodilé:  
 to je dáno především tím, že lidské myšlení je zatíženo svou vazbou na biologický substrát mozek a receptory, které libovolnost projekcí omezují; a dále na sociální prostředí, které se vždy, zejména skrze operace a vzorce operací promítá do pohybů lidského myšlení;  
 dále je to dáno tím, že lidské myšlení sebou při projekcích nese endocepty, které vychylují průběhy lidského myšlení směrem k očekávaným střetům se jsoucnými, z nichž již lidské myšlení umí vytrhnout stimuly a následně pak z nich konstituovat svět (tj. souhrn vzájemně vztažených věcí), nebo mentály.

Nejprve se pokusíme objasnit, proč jsme zavedli a přidržujeme se termínu **vněm**, když máme na myslí mentál, který je konstituován na základě ze jsoucných vytržených stimulů; a termín **vjem** pro mentál (třeba tentýž), když máme na myslí to, s čím operuje lidské myšlení, aniž by bylo stimulováno tím, co vytrhlo ze jsoucných, tedy, když jde např. o vzpomínku. Zdá se nám totiž, že je nutné rozlišovat mezi těmito dvěma druhy pohybů lidského myšlení a pro ulehčení v orientaci, o který pohyb se jedná a co je jeho výsledkem zavádíme toto rozlišení. Vněm je výsledkem pohybů lidského myšlení mezi jsoucnými, při nichž je velmi omezováno právě jsoucnými. Jedná se také vždy o zaměřenost na konstituce věcí na jsoucnech, tedy na něco, co poukazuje navenek, i když je to uvnitř myslí. Vněm chce být kopií věci tak, jak je tato konstituovaná a jak je vnímaná. Je současníkem (dvojčetem) a mentálním doprovodem každé konstituce a vytváří mentální reprezentaci světa. Vněm je však vždy jen něčím, co je aktuálním a navenek orientovaným. Oproti tomu je vjem něčím, s čím lidské myšlení operuje, aniž by bylo vnějškově stimulováno jsoucnými. Vjem je již pojetím, tzn. předchozím zmocněním se stimulů a jejich zpracováním a uchováním (např. jako paměťová stopa), v myslí. K vjemu přistupuje lidské myšlení a znovu s ním může operovat (fantazijně, snově, tvořivě), aniž by se vypravovalo mezi jsoucná, tzn. aniž by cokoli aktuálně ze jsoucných vnímalo. To však neznamená, že by vjem nemohl být také doprovodem aktuálního vněmu, avšak jen jako asociativní nebo analogické pozadí, které se neustále nabízí, při zpracování stimulů vytržených ze jsoucných a které může vystoupit do popředí v meditativním proudu lidského myšlení a změnit tak zaměřenost nebo orientační

---

ztracené ve vněmech, objevuje skrze reflexi znovu sebe sama, protože vněmy odhaluje jako myšlenky. Srovnej: [18] s. 52

pozornost lidského myšlení při vnímání a tedy i konstitucích věcí a vněmů. Vjem však hraje prim při reakcích lidského myšlení na mentály (které jsou tím, s čím lidské myšlení může dále operovat) a vytvářet tak dílo, tedy při tvorbě.

## 2.1 Skutečnost

Skutečnost je to, co konstituujeme jako věci nebo svět (tzn. shluky věcí, sjednocené do systémů nebo celků a vzájemně propojené vazbami a vztahy). Skutečnost je tedy dílem lidského myšlení budovaným na pozadí jsoucen a zároveň, z pro lidské myšlení dostupných částí jsoucen. Skutečnost je konstituovaná také na jsoucnech. Zároveň poznamenáváme, že etymologicky:

... české „skutečnost“, „skutek“ je odvozeno od „kutit“, „kutat“, „vykutat“, tj. vynášet, vynést ze skrytosti na světlo.<sup>8</sup>

Co to znamená? Podle našeho názoru lidské myšlení skutečnost neustále vytváří, konstituuje tím, jak vytrhává stimuly ze jsoucen a jak z nich konstituuje věci a svět. Neustále něco vykutáváme, vynášíme na povrch, vynášíme na světlo něco ze jsoucen (nikoli jsoucna), a z těchto stimulů pak konstituujeme věci. To, co vynášíme na povrch, tzn. do říše fenoménů, jsou jen ty stimuly, které jsme schopni zachytit, tzn. to, co nějak odpovídá možnostem našich sensorických orgánů a naší vnitřní endoceptivní struktuře. Skutečnost je tedy naší konstitucí z těchto stimulů. Naše smyslové orgány nejsou schopny vytrhnout jiné kvality, než ty, z nichž konstituujeme svět.<sup>9</sup>

To, co zde říkáme o skutečnosti, platí i o naší tělesnosti, která je v tomto funkčním modelu mysli reprezentovaná rovněž stimuly vytrhovanými smyslovými orgány. Rozdíl od tzv. *vnější skutečnosti* je právě jen v povaze receptorů zachycujících stimuly z našeho těla. Zatímco exteroceptory vytrhávají stimuly z *vnějších jsoucen*, interoceptory a propioceptory reprezentují v mysli naši tělesnost (včetně mozku). Můžeme sice v této souvislosti uvažovat i o diferencích mezi stimuly vytrhávanými exteroceptory a interoceptory, nebo propioceptory a také o tom, nakolik jsou pro konstituce intencionálních obsahů mysli důležité propiocepty a interocepty (tedy stimuly z našeho těla), avšak, domníváme se, že na podstatě těchto procesů to nic nemění, neboť i tělo je jen konstituováno (i když obsahuje mnoho dalšího obdobně tomu jak je tomu u každé z věcí). Je sice možné, že koincidence, či paralelita stimulů

<sup>8</sup> Srovnej pozn. překladatele in: [12] s. 40. Uvědomujeme si, že etymologie jen o mnohém napovídá, ale že je vždy na autorovi, zda takovouto nápovědu zesílí nebo zmírní-li její konotace. Výtku L. Kováče, kterou v této souvislosti přednesl na konferenci: *Kognícia a umelý život V.* ve Smolenicích v r. 2005 týkající se toho, že v angličtině je tomu jinak respektujeme a opakujeme to, co jsme na ni odpověděli, že se jedná o náš náhled a naše pojetí toho, co je skutečnost, a že nepovažujeme etymologii českého slova skutečnost za rozhodující argument, který by měl být všeobecně akceptován, ale jen za nám se hodící ilustraci, která nám pomáhá k objasnění.

<sup>9</sup> Jsoucna jsou jistě bohatší. Naše smyslové orgány nejsou schopny vytrhnout např. zvukové vlny, které nejsou v rozmezí mezi 20 – 20 000 Hz. Stejně tak nejsme schopni zachytit infračervené záření, vidět molekuly, atomy, nebo bakterie atd., i když při konstitucích světa na jsoucnech jsou všechny tyto vlastnosti přítomné (nebo snad lépe řečeno, jsou lomné).

vytržených exteroceptory a to, jak současně tyto stimuly působí na naše tělo (a co je tedy reprezentováno v mysli jako interocepty, či propiocepty), je tím, čím je ustanovována intence, avšak může to být jen jeden z příznaků tvorby intencí.<sup>10</sup> Intence je, podle našeho názoru, tvořena i komplexitou, adekvací a kontextem vytržených stimulů. Aktuální tělesný prožitek ji dopomáhá jen přesněji identifikovat a případně taxonomicky začlenit, nikoli však v plnosti konstituovat.<sup>11</sup> Intencionalita je vrstevnatá. Není to jen vědomí vztažené k něčemu, ale zároveň vždy mnohost významů, z nichž některé tvoří *jádro* intencionality a další významy jsou jakoby do něj zastrčené, jsou matrjožkovité a tvoří tak hierarchickou strukturu, která se může proměňovat v závislosti na aktuálních okolnostech prožitku. Tak se může stát, že tělesné se dostává v hierarchii někdy do popředí a jindy jsou tyto stimuly jen běžnými doprovodnými fázemi vztažnosti průběhu lidského myšlení k stimulům vytrženým ze jsoucen.<sup>12</sup>

Popisujeme-li skutečnost (věci a svět) jako něco, co je neustále lidským myšlením konstituováno, domníváme se, že nejde o solipsistický postoj, neboť zároveň tvrdíme, že zde jsou jsoucná (něco nás vždy stimuluje), z nichž jednak vykutáváme stimuly; a dále, na nichž konstituujeme věci a svět. Jde jen o to, že věci, (tj. zároveň i fenomény) jsou námi utvářeny, že jsou redukcí ze jsoucen, že se do nich promítá mnohé z nás samých, a že obsahují mnohé vlastnosti ze jsoucen, které nepřichází do uvědomění. Jsoucná a věci (fenomény) tedy tvoří **dvojsvětí** (dvojsvětí se však neomezuje jen na dvojici jsoucná svět). V této souvislosti poznamenáváme, že promýšlíme vztah naší koncepce dvojsvětí k Popperově ontologické koncepci tří světů<sup>13</sup> jak ji podává např. v práci: *The Self and its Brain*, případně ke koncepci N. Hartmanna,<sup>14</sup> jehož čtyři ontologické vrstvy jsou rovněž ve střetu s našimi úvahami.

To, co Popper pokládá za 3. svět, není podle našeho názoru ničím jiným než reakcemi lidského myšlení na mentály, které tvoří buď samostatné představy ve formě např. teorií pokoušejících se zachytit jsoucná a jejich poměrnosti, nebo jde o kulturní artefakty, která jsou rovněž výsledkem reakcí lidského myšlení na mentály. V obou případech jde o zachycení jiné, než fenomenální roviny jsoucen. Teorie a kulturní

<sup>10</sup> G.M. Edelman v této souvislosti píše: There is enough evidence to make the case that input from value systems can combine with modal sensory inputs to yield some earliest conscious experiences. [8] s. 133

<sup>11</sup> Jako příklad můžeme uvést to, co pociťujeme, když se bodneme špendlíkem. Prožitek bolesti se jistě bude vázat s intencí, kterou špendlíku přikládáme (a co je v mysli reprezentováno daným propioceptem), avšak k plnosti pochopení intence špendlíku takovýto souběh nestačí, neboť jeho intence je dána dále velikostí, tím, že slouží k šití, nebo vytahování třísky atd. Interocept v tomto případě reprezentuje jen kontext a to opět jen z části, v němž jsme špendlík identifikovali v souběhu exteroceptů a interoceptů.

<sup>12</sup> Přesto se domníváme, že je obtížné překročit jisté hranice mezi lidským myšlením a tělem, neboť jedno jest v prostoru a čase a druhé nikoli, neboť samo prostor a čas ustanovuje. Nelze tedy stanovit kauzaci bez překročení této hranice. To, že jde o vzájemné působení je nesporné. Dotvrzuje to např. fakt, že při ovlivnění mozkových, či jiných tělesných procesů drogou (např. alkoholem) se zcela mění průběh lidského myšlení a naopak, lékaři dosvědčí, že léčbě těla napomáhá dobrý mentální stav pacienta.

<sup>13</sup> Srovnej [23] s. 36 – 50

<sup>14</sup> Srovnej [9]

artefakty nejsou věcmi nebo předměty, nejsou tedy něčím podobným, jako jsou entity světa 1, i když jsou objektivními entitami (což Popper ilustruje na prvočíselných dvojčatech<sup>15</sup>). Matematické teorie, nebo i jiné kulturní artefakty jsou podle našeho názoru jen jiným zachycením jsoucen, než je tomu tehdy, kdy lidské myšlení vytrhává stimuly a konstituuje z nich svět a mentály. Matematické teorie a kulturní artefakty mají svůj základ ve vzájemných reakcích mezi jsoucny,<sup>16</sup> a jsou svébytným (na poznání orientovaným) střetem lidského myšlení se jsoucny. Nejsou však věcmi. A teprve při dalších reakcích lidského myšlení na mentály mohou být zpředmětněny ve formě díla. A nic na tom nemění, že (jak tvrdí Popper):

*Mají vnitřní strukturu, jež je zčásti autonomní. [23]*

Svět 3 je tedy podle našeho názoru jen něčím, co je odvozeno ze jsoucen a co tedy ještě předtím, než je petrifikováno jako matematická teorie nebo kulturní artefakt, obsahuje vztahy mezi jsoucny. Má-li být prohlášen za autonomní, pak tím jsou prohlášeny za autonomní vztahy mezi jsoucny. Rovněž ontologická koncepce Hartmannova se čtyřmi vrstvami (anorganická; organická; duševní; a duchovní) se nám zdá poněkud sofistikovaná, protože z hlediska lidského myšlení jsou fyzikální objekty a objekty biologické (živé), včetně našeho těla jen něčím, co je zprostředkováváno jako exterocepty, případně intero- a propriocepty. To že jde u biologické vrstvy o jinou strukturální organizaci, která umožňuje autoreplikaci, je jen záležitostí strukturální, nikoli ontologickou. O Hartmannově čtvrté vrstvě (duchu) pak platí totéž, k čemuž jsme se vymezili o odstavec výše, když jsme uvažovali o Popperově 3. světě.<sup>17</sup> Takže to, co zbývá je dvojsvětí.

I když nás především zajímá to, jaké jsou vzorce pohybu lidského myšlení při konstitucích a explanacích světa, nemůžeme ponechat stranou ani vnucující se otázku, po charakteru jsoucen.<sup>18</sup> Tato otázka je důležitá z toho důvodu, že věci a svět konstituujeme na jsoucnech a tudíž naše konstituce musí nějak respektovat charakter jsoucen, nemohou být v tomto ohledu libovolné. Důležitost pak shledáváme také v tom, že dvojsvětí (jsoucna – svět věcí) by mělo být v harmonii, v rovnováze, neboť bude-li distance dvou světů velká, pak to nutně musí vést k zborcení konstituovaného světa věcí. Napsali jsme, že lidské myšlení je úspěšné při konstitucích a objasněních světa, nebo také při konstrukcích předmětů tehdy, opakuje-li kosmický pohyb myšlení, tedy konstituuje-li věci stejně nebo obdobně jak jsou určena jsoucna (dává je do stejných vazeb a vztahů a respektuje poměrovost jsoucen). Dodáváme, že při tomto pohybu jde o participaci na řádu, který je mimo lidské myšlení, a který je lidským

<sup>15</sup> Srovnej [24] s. 84 – 85

<sup>16</sup> Lidské myšlení je jsoucnem mezi jsoucny

<sup>17</sup> Omlouváme se všem znalcům díla K. R. Poppera a N. Hartmanna za poněkud stručné vyjádření se ke koncepcím těchto filosofů. Chceme ujistit čtenáře, že naše promyšlení i diskuse s díly (v případě K. R. Poppera i s ním osobně) bylo dlouhodobé, avšak tato naše práce nechce být polemická, ale monotematická.

<sup>18</sup> Poznáváme, že obtížné není vysvětlit lidské myšlení, ale to, co to jsou jsoucna a jak jsou konstituovány věci a svět. Každý má zážitek z průběhů lidského myšlení, ale zážitek ze jsoucen je vždy již nějak zprostředkovaný průběhy lidského myšlení.

myšlením respektován jako nutnost, avšak, ke kterému se lidské myšlení úsilím o jeho poznání teprve přibližuje. Tím se také vyjadřujeme k charakteru jsoucen. Uvedeme příklad:

$$A^2 + B^2 = C^2$$

Tato rovnice je univerzalitou výskytovosti pravoúhlého trojúhelníka platící v eukleidovské geometrii, je nezávislá na čase a na kultuře atd. Platila i tehdy, když lidské myšlení ještě tyto souvislosti neodhalilo, právě tak jako bude platit, když lidstvo vymře (takto ostatně chápe K. R. Popper svět 3, i když jde o vztahy mezi jsoucný). Tímto příkladem však nechceme říci, že jsoucná jsou jen geometrické entity. Stejně tak, jako výše uvedená rovnice nutně podmínky výskytovosti pravoúhlého trojúhelníka platí, že směs  $H_2$  a  $O_2$  je někdy třaskavinou, jindy, tzn. při jiných poměrech mezi těmito chemickými prvky je vodou, která může být v pevném, kapalném nebo plynném skupenství a opět jindy, při jiných poměrech se vyskytuje jako peroxid vodíku a ničí bakterie, jindy může být také těžkou vodou (s jiným izotopem vodíku) atd. Poměry mezi C,  $H_2$  a  $O_2$  mohou dávat ocet, cukr, nebo alkohol a spoustu dalších věcí s velmi odlišnými vlastnostmi a vždy bude jejich výskyt záviset na poměrech mezi těmito třemi prvky a jejich strukturálními vazbami. Tytéž charakteristiky jsoucen a jejich poměrů výskytovosti platí v biologii, ale také i v sociologii můžeme nalézat některé stálé a universální charakteristiky výskytovosti v závislosti na poměrech a obdobně snad i v pohybech lidského myšlení, neboť i lidské myšlení je jsoucnem mezi jsoucný (a my se ostatně snažíme v této práci podhalit některé vzorce operací jeho pohybu). Také na úrovni lidského myšlení se tedy jedná o poměrovost, zobrazující se v obrazotvornosti, kdy se ze singularit konstituují věci a vněmy, nebo konstruují předměty a tvoří díla. Každopádně však musí lidské myšlení vyhledávat mezi jsoucný něco stálého, tedy být při poznávání v jiné rovině, než je tomu tehdy, když se pohybuje mezi věcmi.<sup>19</sup>

Pro Husserla je takovouto konstantní idealitou, skrz niž se lidské myšlení setkává s řádem jsoucen a na jejímž základě pak lze konstituovat věci, především to, co se objevuje v jazyku, tedy pojmy, protože jim náleží ideální objektivita a ideální identita, kterou nelze zaměňovat.<sup>20</sup> Nejsou to tedy jen geometrické struktury, ale idealitami

<sup>19</sup> Poznamenáváme, že ve fyzice jsou některé konstanty, které jsou opěrnými body při vysvětleních světa a tedy i při jeho konstitucích. Jsou jimi např. rychlost světla, Newtonova konstanta gravitace, nebo masa elektronu. ( $c = 299,792,458$ ;  $G = 6.673 \times 10^{-11}$ ;  $m_e = 9.10938188 \times 10^{-31}$ ). Lidské myšlení se k nim muselo dopracovat tím, že zachytilo při svých projekcích mezi jsoucná něco stálého. Tyto konstanty jsou sice někdy relativizovány tím, že někteří fyzici uvažují o možných světech, v nichž jejich stálost nemusí platit, ale to neodporuje tomu, že jsou pro lidské myšlení opěrnými body konstitucí a že tehdy, drží-li se jich lidské myšlení, opakuje něco, co je stálé. Zpochybnění těchto konstant jsme našli ve stati: [3]

<sup>20</sup> Srovnej: [7] Derrida zde cituje Husserla z jeho práce *Původ geometrie: Idealita dospívá ke své objektivitě „pomocí řeči, která jí jakoby propůjčuje její jazykové tělo“* (s. 65), takže nejde jen o idealitu, ale i o objektivitu. Dále pak Derrida poznamenává, že: *Absolutní jednoznačnost je vlastně představitelná jen ve dvou mezních případech. Za prvé tehdy, pokud označovaná*



může být vše, co lze nazvat eidos. Eidos je pak něčím, jehož uchopení a tedy i pochopení je mnohotvárným aktem. Nazření podstaty je originálně dávajícím aktem, a jako takové je analogem smyslového vnímání, a nikoli fiktivního zpřítomnění [13]. Husserlovské redukce jsou pak metodou jak se k těmto idealitám prodrat a tím také jak je poznat. To však neznamená, že bychom ignorovali to, že podstaty jsou nám přirozeně dány v intuici vždy, když v přirozeném světě cokoli děláme. Pro nás však vyvstává stále naléhavěji otázka, zda Husserlovy redukce nejsou jen jiným pojmenováním abstrakce. Lidské myšlení se musí přes věci, tzn. fenomény a dokonce i přes to, co je eidos, dopracovat k čistému vědomí a z něho pak začít. Je tedy zřejmé, že Husserlovi jde o prozkoumání (v naší terminologii řečeno) konstitutivních a konstruktivních mohutností lidského myšlení a tím zároveň i o možnosti poznání. V tomto směru je jeho fenomenologie zároveň i metodologií.

Vyslovujeme tedy domněnku, že **jsoucna jsou nečasové ideality**. K této domněnce nás vedou úvahy o možnostech, které toto pojetí nese sebou. Jednak je zde předpoklad malého množství basických kvalit (snad jen jediné kvality, např. čísla, nebo vibrace), které mohou právě poměrovými kombinacemi nabývat na pestrosti a různorodosti. A dále je to předpoklad, že výskytovost jsoucen je dána poměry mezi těmito kvalitami (nebo jedinou kvalitou) v jejich strukturním uspořádání. Ideality tedy nejsou určité, ale jsou jen určené pohybem (myšlením), který je kumuluje (dává k sobě) v určitých poměrech (kvantech), čímž se i tyto kvality (nebo tato kvalita) setřídí do určitých strukturních vzorců. **Poměry** a **struktury** jsou tedy tím rozhodujícím, co může lidské myšlení při svých projekcích mezi jsoucna rozpoznat. Ostatně, když lidské myšlení konstituuje věc a vněm musí také z jedné kvality sestavit strukturní vzorec, který teprve reprezentuje obdobnou kvalitu, jakou poznalo při svých projekcích. V mozku jsou veškeré stimuly převedeny na bioelektrické potenciály v neuronových sítích vytvářející neuronální koreláty vytržených stimulů. Tedy onou kvalitou, z níž lidské myšlení konstituuje různorodé kvality věcí a světa je bioelektrický impuls.<sup>21</sup> Domníváme se také, že především záleží na tom, jakou strukturu, tzn. jaký excitovaný neuronální okruh, struktura, či neuronální populace jsou aktivovány. Intenzita vnímání pak je asi dána frekvencí vstupů, tedy častostí vytržených stimulů. Jsoucna vystupují ve formě potencialit ve vztahu k realitám.

---

*věc není pouze absolutně soudržným, nehybným a přirozeným předmětem, ale též jsoucnem, jehož jednota, identita a objektivita by předcházela veškeré kultuře ... stejně je tomu i v případě, pokud – za druhé – máme na druhém pólu jazyka označit absolutně ideální předmět. Šanci na jednoznačnost by v tomto případě zajišťoval předmět nikoli předkulturní, nýbrž transkulturní, např. předmět geometrický. Tato šance by nepochybně odpovídala samotnému poslání vědy [7] (s. 55).*

<sup>21</sup> Rozdíl mezi sladkostí a slaností je jen, podle našeho názoru, ve strukturním uspořádání stimulů. Možná, že k tomu přispívají i neurotransmitery (nebo snad i aktivace peptidových vazeb, případně bílkovin v tělu neuronu), ale nic dalšího. Neurotransmitery nebo proteiny a bílkoviny by však byly jen pomocnými prostředky a za rozhodující považujeme to, v jaké struktuře, po jakých neuronálních spojích je stimul aktivován, abychom dostali jednu slanost a po druhé sladkost atd.

Reality (skutečnosti, tedy věci a přirozený svět) jsou na základě poměrů obdobně konstituovány lidským myšlením. Avšak to, zda se lidské myšlení dobírá adekvátních poměrů při konstitucích světa, závisí na jeho poznání poměrů, a struktury, v nichž jsou určena jsoucná. Pokud by zde byla jen libovůle, nebo upřednostnění vazby lidského myšlení na biologický nebo sociální substrát, pak by konstituce věci byly dočasné nebo by se vůbec nezdařily. Zjištění poměrů mezi jsoucnými mohou být lidským myšlením dosaženy při projekcích, nebo snad také reflexí a vyvozením toho, jak jsou v lidské mysli uspořádány mentály.

## 2.2 Konstituce věcí a mentálů

Nyní pokročíme dále ve výkladu funkčního modelu mysli, přičemž se zaměříme na základní pohyby lidského myšlení.

<p>a. na základě ze jsoucen vytržených stimulů konstituuje lidské myšlení svět na jsoucnech;</p>	<p>i. svět je konstituován na základě operací a vzorců, které buď vystihují to, jak jsou určena jsoucná, nebo jsou imaginacemi lidského myšlení;</p> <p>ii. nejčastěji jsou pro konstituce světa využívány ty operace a vzorce, které si lidské myšlení již osvojilo při předchozích projekcích mezi jsoucnými a které se lidskému myšlení při konstitucích osvědčily;</p>
<p>b. na základě ze jsoucen vytržených stimulů konstituuje lidské myšlení mentály;</p>	<p>i. to, co je konstituováno jako mentály umožňuje lidskému myšlení další konstituce ve formě kvazipozicionality, aniž by se lidské myšlení muselo projikovat mezi jsoucnými;</p> <p>ii. ne vše, co je lidským myšlením vytrženo ze jsoucen a ne vše, co je lidským myšlením konstituováno je jím také reflektováno (uvědoměno);</p> <p>iii. ne vše, co je lidským myšlením vytrženo ze jsoucen je jím také složeno ve věci nebo mentály; mnohé zůstává v singulární podobě; a mnohé je odsunuto do periferií mysli;</p>

Některé z výše uvažovaných momentů jsou natolik závažné, že vyžadují podrobnější výklad. O jiných, např. o tom, co zůstává v singulární podobě a co je odsunuto do periferií mysli (nebo co je ztraceno) může sice uvažovat, dokonce tyto stimuly mohou být důležité, avšak nemáme k nim přístup a proto také nebudeme o nich dále spekulovat.

### 2.3 Konstituce věcí a světa na jsoucnech

Prvním problémem, který chceme artikulovat je: co to znamená být konstituován na jsoucnech a to ze stimulů vytržených ze jsoucen?

Být konstituován na jsoucnech navozuje představu hnětení věcí z vytržených stimulů ze jsoucen a to někdy i na značnou dálku, neboť věci jsou jakoby od nás vzdálené, jakoby v prostoru již předem rozvržené a zároveň je navozena i představa, že konstituce věcí a mentálů se dějí postupně, jakoby v čase, tzn. že lidské myšlení se zdá být odděleno od jsoucen a v určitém teď se mezi jsoucna projikovalo a v jiném následném teď dopravilo vytržené stimuly do mysli a v dalším následném teď z něj konstitovalo v mysli mentál a na jsoucnu uhnětlo věc. To je představa mechanická a je jednou ze zatížeností naší obrazotvornosti, která je sice (jak jsme poznamenali) naší svobodou, ale zároveň má omezení vyplývající z pravidel pohybu lidského myšlení.

Uvažujeme však jinak a sice tak, že lidské myšlení je ve jsoucnech rozprostřené<sup>22</sup> a svou rozprostřeností ustanovuje prostor. Zároveň se domníváme, že konstituce věcí na jsoucnu probíhá mimo čas, neboť čas není ničím jiným než posobností jednotlivých projekcí<sup>23</sup> (konkrétní, jednotlivá projekce tedy na ustanovení času nedostačuje). Obvykle máme za to, že věci jsou neprostupné a že každá z konstitucí je spojováním částí v celek. Konstituci si však můžeme představit jako vidění (i když o konstituci půjde i u jiných stimulů vytržených jinými smyslovými orgány než zrakem). Pokusíme se to osvětlit na následujícím příkladu.

Co máme na mysli. Představme si jaký je rozdíl mezi tím, když se díváme na televizní obrazovku, kde jsou záběry z nějaké indické vesnice a tím, když jsme v této vesnici a díváme se na totéž. Rozdíl nebude jen v dvojrozměrnosti a trojrozměrnosti, ale např. také v tom, že nebudeme cítit aktuální indické pachy (a prý jsou značné), ale jen ty, které jsou u nás doma. A to je to, co chceme naznačit, hovoříme-li o konstitucích věcí a světa na jsoucnech. Vnímáme jen to, co je nám přístupné a tak, jak jsme schopni to vnímat. Vytrháváme, vybíráme mezi kvalitami jsoucen jen pro nás dostupné kvality a z nich a na jsoucnech konstituujeme věci. Ostatní kvality nevnímáme, nevšímáme si jich (jsou jen v Indii a nikoli u nás doma, kde sledujeme obrazovku; nebo to jsou takové kvality, pro něž nejsme vybaveni smyslovými orgány, např. ultrazvuk). Představme si, jaké by to bylo, kdybychom byli např. bakterie *Escherichia Coli*, nebo jaké by to bylo být netopýrem. Podrobně se nad těmito problémy zamýšlí L. Kováč [15] a T. Nagel [20]. Ale můžeme si představit i další variace. V takovýchto případech bychom na jsoucnech konstitovali jiné věci a

<sup>22</sup> Např. vzduch je jsoucnem a my jsme v něm. Není od nás vzdálen, je i v nás, není zde vnitřní a vnější, kůže není hranicí subjektu. Stejně je tomu se všemi jsoucny, a lidské myšlení je mezi nimi, je jedním z nich. Projekce se tedy neděje na vzdálenost, neboť je to lidské myšlení, které ustanovuje vzdálenost.

<sup>23</sup> Poznamenáváme, že termín projekce navozuje představu vzdálenosti, z níž se lidské myšlení vrhá mezi jsoucna, avšak nám jde při používání tohoto termínu především o to, že jsou to akty lidského myšlení které jsou orientované mezi jsoucna (na rozdíl od těch, kdy lidské myšlení operuje s vnitřními intencionálními obsahy mysli).

sestavili jiný svět. Nejde však jen o jiné světy, tak jak je chápal Leibniz a jak se tomu vysmívá Voltaire, ale i o to, jak můžeme další světy konstituovat na základě jiných předpokladů. Jde o to, jak si svět konstituuje barvoslepý, nebo slepý, nebo hluchoslepý. Pro každého z nich je svět odlišný a je takto odlišný i pro každého z nás, jen snad ne v takovéto míře. To však stále nejsme u toho, co to znamená konstituce na jsoucnech. Konstituovat věci na jsoucnech znamená, že ze stimulů, které jsou pro nás dostupné a které jsou vytrženy ze jsoucen sestavujeme věci. Máme tedy mezi jsoucný rozprostřeny své a svébytné senzorycké orgány, jejichž prostřednictvím vytrhujeme stimuly. Tyto **stimuly jsou jen části jsoucen**, která jsou transformována do forem (reprezentací), s nimiž může lidské myšlení nějak dále operovat a konstituovat tak svět a zároveň s tím i mentály.

Jaké jsou tyto formy (reprezentace)? V zásadě můžeme říci, že na úrovni mysli se jedná o singularity, které jsou sestavovány podle předpřipravených vzorců, nebo podle pravidel do struktur. Z neurofyziologického hlediska (tedy z pohledu biologického substrátu, na němž je lidské myšlení vázáno vazbou) bychom pak mohli říci, že se jedná o bioelektrické impulsy, jimiž jsou excitovány (předpřipravené) neuronální okruhy, struktury nebo populace. Domníváme se, že jistou roli zde hrají také biochemické procesy, ale dominující formou jsou bioelektrické impulsy vyměňované mezi jednotlivými neurony. Je tedy zřejmé, že v tom, co je věc, je obsažena jen část jsoucná, na němž je věc konstituovaná (ta část, která je v mohoucnosti smyslových orgánů a pravidel pohybu lidského myšlení). Z této části je věc konstituovaná v poměrech, které byly lidským myšlením zjištěny při projekcích a vytrhování stimulů.

Co to znamená být konstituován v poměrech? Lze si to představit jako míchání barev ze tří výchozích. O výsledku barevnosti pak rozhodují poměry (kvantity) mezi výchozími barvami. Naše představa je však ještě složitější v tom, že tyto výchozí kvality (tzn. v naše příkladě výchozí barvy) jsou ustanovovány až tím, že vytržené stimuly ze jsoucen (tzn. singularity – bioelektrické impulsy) jsou nejprve konkrétním způsobem strukturně uspořádávány, čímž vzniká barevnost a následně pak v poměrech smíchávány do konečné podoby, kterou má věc a vněm. Zároveň však znovu zdůrazňujeme, že to vše se odehrává na jsoucnech (nevíme, zda jsoucná jsou barevná, snad jen odráží konkrétní vlnové délky a jiné absorbují, nebo lámou) a že tedy pod tím, co vnímáme jako věc, jsou skryty další části jsoucen, které naše smyslové orgány nebyly schopny jako stimuly vytrhnout (např. jiné vlnové délky, radiace, atp.), nebo další poměrové uspořádání, které nebylo lidské myšlení schopno odhalit a tak si vypomohlo svým vlastním propočtem, ale které přesto jsou (pro nás nepoznanou) součástí věcí. To, co jsme ilustrovali na barvě, lze ilustrovat také na tvaru, hustotě, pevnosti, neprostupnosti, váze atd. Musíme přitom zdůraznit že konstituce věci je uspořádávání v poměrech velmi rozdílných kvalit, neboť každý z receptorů vytrhuje ze jsoucen naprosto odlišné kvality, které jsou pak poměrově uspořádávány. Dále znovu zdůrazňujeme, že lidské myšlení všechny kvality již předem očekává, takže vytrhuje

stimuly podle svých očekávání a má proto předpřipraveny vzorce a proto je konstituce věci na jsoucnu bezprostřední.<sup>24</sup>

Co to však znamená být konstituován; a být konstituován podle vzorce? Konstituce v původním významu (z lat. *constitutio*) znamená: postavení, stavba, uspořádání, zřízení, ustanovení. V našem pojetí pak je to především mluvení lidského myšlení: *je to tak*. Lidské myšlení tedy něco (stimuly vytržené ze jsoucna) uspořádává, vymezuje stimulům postavení a tím vytváří stavbu. Při vytváření stavby pak využívá pravidla (pravidelnosti) svého pohybu, z nichž základním je efektivnost, nebo to, co E. Mach označuje jako ekonomii myšlení.<sup>25</sup> Obdobné kritérium můžeme pak nalézt také u W. Jamese. Jde tedy o to, že lidské myšlení často využívá vzorce pro konstituce světa a mentálů, které se již osvědčily a tak vytváří zkratky a šetří tím svou energii. Lidské myšlení by se mohlo pokoušet sestavit ze singulárních stimulů něco jiného (tato odlišnost by ovšem nemohla být velká, neboť základem konstitucí jsou vždy ze jsoucna vytržené stimuly), avšak většinou participuje již na něčem známém, osvědčeném a stabilním (na něčem, co je již předpřipraveno v endoceptech), v němž se již umí pohybovat. Jak si takovou tendenci vysvětlit? Proč lidské myšlení spíše inklinuje k opakování již známého, než k využívání možného a tvořivosti? Proč jen v omezené podobě využívá imaginaci ke konstitucím? Proč neosmyslňuje svůj pobyt modem moci být?

Naše vysvětlení je následující. Jedním z důvodů může být to, že imaginace může ústít v neúspěšnou konstituci věci. V takovémto případě není korespondence mezi konstitucemi (fantazijních) mentálů a konstitucemi věcí. Zaměřenost imaginací v těchto případech nedovoluje vytržení adekvátních stimulů, neboť ty nejsou při projekcích k nalezení. Jiným případem je ten, kdy při syntetizující činnosti není imaginativní očekávání naplněno pro neschopnost lidského myšlení uvést jednotlivé stimuly do vzájemně harmonizujícího stavu a to buď mezi sebou, nebo mezi věcmi a jsoucnem. V tomto případě imaginace neparticipuje na možném a je příliš vzdálená od poměrů obsažených ve jsoucnech. Totéž platí nejen o konstitucích, ale i o objasňující funkci pohybu lidského myšlení. Imaginace, která není v souladu s poměrovostí jsoucna, neobjasňuje konstituované věci, což vyvolává nutnost dále hledat objasnění a stává se tudíž zátěží pro lidské myšlení.<sup>26</sup> Z těchto důvodů se patrně lidské myšlení uchyluje častěji k opakování při konstitucích než k tvorbě novostí.

<sup>24</sup> Opoždění uvědomění vněmu zde sice je, odhaduje se na setiny nebo desetiny sekundy, v závislosti na nutném převodu stimulu do formy reprezentace a předání vzruchu do patřičných mozkových center a jejich dalším zpracování. Kdyby se lidské myšlení projikovalo mezi jsoucna bez endoceptivních očekávání, byl by reakční čas ještě delší, což lze experimentálně ověřit.

<sup>25</sup> E. Mach používá termín ekonomie myšlení ve dvou různých významech: první je vztažen k vědeckému vysvětlení a druhý zakládá ontologické pojetí, tzn. genezi objektů a zákonů jejich vzniku z neutrálních zkušenostních elementů. Srovnej např.: [2]

<sup>26</sup> Příkladem zde může být něco, co třeba na dálku nerozpoznáme a hledáme pro to objasnění. Až teprve po přiblížení znovu konstituujeme věc v pro nás již známé a zařaditelné formě, nebo ji zpracujeme jako novost.

## 2.4 Konstituce mentálů

Další pozornost věnujeme konstitucím mentálů.

Mentály charakterizujeme především jako něco, co je ve formě kvazi-pozicionality, tzn. jako něco, co není (prostorově) umístitelné, avšak zároveň něco, co vzniká konstitutivní aktivitou lidského myšlení. Co to znamená, že něco je jen jakoby, že nezaujímá žádné místo v prostoru? Především to znamená, že mentály jsou jinak než věci, které mají svůj základ ve jsoucnech a proto je možné tyto věci vypátrávat jako topograficky vymezené, lze jim přiřadit karteziánské souřadnice a je možné se o ně opřít při jestvování i když např. nejsou aktuálně vnímány. Být však jen jakoby umístitelný, znamená, že to nemá přesně topograficky určitelné místo, a že je to tehdy a jen tehdy, jestliže je to generováno lidským myšlením a jím strháváno do proudu uvědomění. O některých mentálech bychom sice mohli uvažovat tak, že jsou také stabilní, např. paměťové vzpomínky, avšak to, co se zjevuje jako vzpomínka, musí být vždy lidským myšlením iniciováno tak, že je to vyzvednuto a vždy znovu konstituováno z mysli. Neuronální síť, či dokonce konkrétní neuronální okruh má jistě topograficky určité místo, ale samotný neuronální okruh není vzpomínkou. Tou je až zařazení zážitku do intencionální obsahové struktury mysli a její předvedení k uvědomění. Takto je možno započít uvažování o mentálech, jako o něčem, co je jako věc, ale co je vždy vázáno na pohyby lidského myšlení a to nikoli jen konstitutivně, ale i generativně, což znamená, že to musí být vždy lidským myšlením vybuzeno a uvědoměno.

Tím se dostáváme k tomu, co jsme označili ve funkčním modelu mysli tím, že: ne vše, co je lidským myšlením vytrženo ze jsoucen a ne vše, co je lidským myšlením konstituováno je jím také reflektováno (uvědoměno);

Uvědomění je mentálem. Uvědomění, jehož vznik charakterizujeme jako něco, co vzniká reflexí<sup>27</sup>, tzn., že vytržené stimuly jsou zaznamenány jako to, co lidské myšlení unáší a s čím dále operuje a co je zároveň sekundárním proudem lidského myšlení registrováno, zrcadleno. K tomu, co lidské myšlení nese sebou, ať již jako výsledek svých projekcí mezi jsoucná, nebo jako mentály z mysli, se neustále přičleňují asociativní a analogické intencionální obsahy, které jsou sekundárním proudem vyzvedávány z endoceptivní struktury a také zaznamenávány. Vedle tohoto procesu je

<sup>27</sup> Reflexí rozumíme zvrtnost, zakousnutí se do sebe, nebo zrcadlení toho, co se děje. Jde tedy o dvojitě zaznamenání téhož, jednou v bezprostřednosti a podruhé v zrcadle, jde o looping, rekurs, nebo reverberaci. Při trojitě reflexi pak je generována substruktura jáství, neboť ta je tou, která všechny reflexe drží v jednotě přes časové dimenze, tzn., že lidské myšlení reflektuje prožitky a zároveň i reflektuje to, že dochází k reflexi prožitků (třeba i byvších, nebo naopak k naplnění budoucích očekávaných prožitků). Obdobně o reflexi uvažuje i Husserl s tím, že píše o zvrstvení představ v noésis a noématu. Jako příklad uvádí návštěvu Drážďanské galerie a své zastavení před Teniersovým obrazem, který znázorňuje nějakou galerii s obrazy, takže jde o zvrstvení obrazů v obraze, případně vzpomínek ve vzpomínání (můžeme si vzpomínat, jak jsme na něco vzpomínali atd.). Srovnej: [13] s. 213. Obdobně se problematikou zabývá belgický malíř R. Magritte zejména v obraze: *Lidská situace číslo dvě*.

však také mnoho neuvědomovaných operací lidského myšlení, což znamená, že lidské myšlení něco unáší a operuje s tím, ale co sekundární proud nezaznamenává, nebo nezrcadlí, takže nedochází k uvědomění toho, s čím lidské myšlení aktuálně operuje.

Úroveň, na níž se odehrávají mentální procesy, jakým je utváření sebevědomí, je z našeho hlediska reflexí. Jde tedy o to, že je excitována taková substruktura mysli, která zaznamenává nejen to, co si uvědomujeme, ale také to, že jsem to já, kdo si něco uvědomuje. Sebevědomí tedy provádí proces ohýbání (tedy reflexe) a reverberace unášených stimulů a jejich propojení s asociacemi, či analogiemi. Na tomto procesu reflexe se významně podílí i jazykové substruktury mysli. Lze si to osvětlit následujícím způsobem. Prožitky jsou v mysli zaznamenány patrně ve formě pojmů. Abychom je mohli sdělit, musí být aktivovány jazykové substruktury, které přeloží pojem do slov a určí jim místa v gramatické struktuře věty. Pojem je takto vyzvednut a musí být podržen na potřebnou dobu, než jej lidské myšlení přetlumochí. Takovéto podržení umožňuje reflexi, neboť je neustále k dispozici po dobu hledání vhodného slova a vyjádření prožitku. Jazyková substruktura proto patrně vykonává při uvědomění a formování sebevědomí důležitou roli a to i tehdy, když tento prožitek nechceme sdělovat, ale jen si ho plně uvědomit. Zároveň takto identifikujeme i substrukturu tlumočnicka z pojmové formy (či z jazyka myšlení – mentalese) do formy přirozeného jazyka. A opět je při těchto procesech aktivována i substruktura jáství, která reflektuje, že je zde něco identického pro tyto prožité i sdělované prožitky.

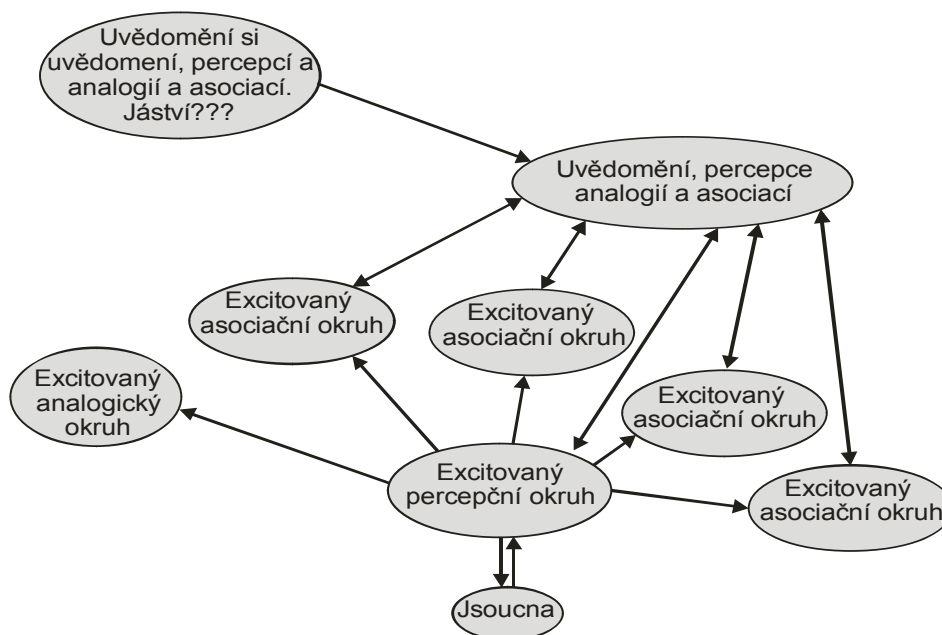
Sebeuvědomění, které ztotožňujeme s jástvím (self), je substrukturou mysli, v níž dochází k uvědomování si toho, že si něco uvědomujeme (že provádíme operace) a je také tím, co umožňuje určovat další průběhy lidského myšlení, tzn. koncentrovat se na určité intencionální obsahy, nebo pokoušet se připouštět do průběhů lidského myšlení další analogická, nebo asociativní spojení, atd. a co podržuje a tím dává všem těmto singularitám jednotu, sjednocuje je do celku.

Nejprve se pokusíme o zobrazení reflexe a následně pak o trojvrstevné operace vedoucí k sebevědomí, tj. k jáství. Musíme si však ujasnit, zda reflexe je operací lidského myšlení, tzn. určeným pohybem, při němž dochází k ohýbání a zrcadlení; nebo zda se nejedná o operaci, ale o doprovodný proces, který při jiných operacích (pohybech), (např. při abstrakci, komparaci a generalizaci) lidského myšlení je nutný, ale nikoli specifický, tj. takový, který by mohl být sám o sobě.

Přestože I. Kant považuje reflexi za rozhodující operaci při tvorbě pojmů, za operaci, která pod sebe zahrnuje abstrakci i komparaci; přestože E. Husserl, když analyzuje obecné struktury čistého vědomí, rovněž spatřuje jako rozhodující reflexi, což dokumentujeme: *Fenomenologická metoda se pohybuje výhradně v aktech reflexe* [13] (s. 150); přes výše uvedené se domníváme, že jde o doprovodný proces vznikající při operacích lidského myšlení, který je obrácí dovnitř mysli. Co nás k takovému tvrzení opravňuje? Jde o domněnku, že mnoho z toho, co se děje v naší mysli není uvědomováno a že mnoho z toho je také jen vedlejším produktem dominujících pohybů lidského myšlení. Co bychom reflektovali, když by o žádný průběh lidského myšlení nešlo, když by nebyly věci, ani mentály? Nejprve zde tedy musí být nějaké

konstitutivní akty a tím i intencionální obsahy mysli, které teprve následně mohou být reflexi uvědomovány. To jsou naše prozatímní argumentace.

Na obr. 1 se pokusíme schematicky znázornit, jak chápeme reflexi a vznik sebevědomí.



**Obr. 1.** Reflexe a vznik sebevědomí.

Ilustrace na obr. 1 znázorňuje jak se lidské myšlení projikuje mezi jsoucna, vytrhuje z nich stimuly, konstituuje z nich věci na jsoucnech a vněmy v mysli a přivádí vše k uvědomění. Zároveň s percepcí jsou (sekundárním proudem) generovány asociativní a analogické intencionální obsahy mysli a jsou tak rovněž přiváděny k uvědomění. A konečně je v mysli excitována další substruktura, která je uvědoměním si toho, že jak percepcie, tak i analogické a asociativní souvislosti jsou uvědomovány a tuto substrukturu mysli nazýváme jástvím, či sebevědomím. Domníváme se zároveň, že další cyklus, totiž uvědomění si toho, že si uvědomujeme to, že si něco uvědomujeme, nemá již smysl. Směřoval by totiž k něčemu, co si toto všechno uvědomuje (tedy jak vjemy, nebo stimuly vytržené ze jsoucnů a asociativní a analogické souvislosti těchto vjemů a stimulů; ale i to, že je zde někdo, nebo něco, kdo, nebo co si to vše uvědomuje, tedy nějaká další entita, v níž by se toto uvědomování toho všeho odehrávalo) a to by již nebylo spjata s konkrétní strukturou mysli, ale nějakým způsobem by ji překračovalo. Neshodujeme se tedy s Husserlem, který se domnívá, že: *Reflexe jsou opět prožitky a mohou se jako takové stávat substráty nových reflexí, a tak in infinitum, v principiální obecnosti.* [13] (s. 151). To, v čem se odlišujeme, je nikoli to, že reflexe jsou opět prožitkem, ale v tom, že by se tento proces uskutečňoval in infinitum. Lze si sice představit i nějakou další substrukturu mysli,



v níž by se to vše odehrávalo, ale taková by již nebyla uvědomovaná, ale nevědomá si sebe sama (neboť by se to v ní odehrávalo) a tudíž opět něčím, k čemu lidské myšlení již nemá přístup.

Dále chceme zdůraznit, to, že právě tuto dvojnásobnou reflexi (zrcadlení v zrcadle, nebo reverberaci) vytvářející sebevědomí lze považovat také za jáství proto, že do uvědomění a konstitucí mentálů přicházejí ony asociativní a analogické intencionální obsahy, tedy něco, co tvoří endoceptivní strukturu mysli. Takto se pak při konečné konstituci mentálu, nebo usměrnění pohybu lidského myšlení, či při případné jeho koncentraci na určitý intencionální obsah, projevuje celá související endoceptivní struktura mysli, tedy to, čím jáství disponuje. Jáství je tím, co udržuje různorodost v souvislostech, co udržuje minulost pohromadě s budoucněním a tedy i prostorovou perspektivu nacházení něčeho a vyjímání onoho z pozadí a tím i jeho umocňování. Jáství je důležitou substrukturou mysli z toho důvodu, že je to právě tato substruktura mysli, která **drží** všechny aktuální prožitky pohromadě s minulými i s očekávanými budoucími projekcemi a také se všemi asociativními a analogickými obsahy. To zároveň umožňuje syntézu i diferenciaci mezi konstituovanými věcmi i mezi mentály. Excitované asociativní a analogické prožitky se propojují se vším aktuálním a tak všechny konstituované věci a mentály jsou jakoby k ní vztaženy a mají tak své souvislosti, které jsou jak prostorové, tak i, a to především, časové. Jestliže jsme čas charakterizovali jako posobnost projekcí lidského myšlení mezi jsoucna s tím, že se v každém teď nachází něco, co bylo před tímto teď (i když je tento prožitek již částečně modifikován) a zároveň je v tomto teď i projektování příštích projekcí, pak jáství udržuje tuto kontinuitu a vytváří tím možnost zhodnocení minulého i očekávání budoucího. Časení je tedy nejen vertikála mezi jástvím a jsoucný, z nichž jsou vytrhávána jsoucna, ale také horizontála mezi retencemi a protencemi. Je to však také pohyb, který neustále vtahuje do souvislostí asociativní a analogické intencionální obsahy mysli, takže lidské myšlení se neustále pohybuje vícedimenzionálně. Jednotlivé stimuly a prožitky tak nejsou izolovány, ale mají souvislosti, které mohou napodobovat, nebo i reprodukovat souvislosti jsoucen, čímž i přibližovat to, aby pohyb lidského myšlení souvisel s pohybem myšlení.<sup>28</sup> Tímto držením různorodosti prožitků k jednomu (tj. k jáství) se zároveň ustanovuje jeden z důležitých momentů intencionality, kterou charakterizujeme jako komplexitu. Držení k jednomu se týká nejen prožitků, ale také i dvou proudů lidského myšlení jako dvou součástí téhož a zároveň i uvědomování aktuální činnosti obou proudů.

Mimoto samotné jáství, zdá se, jako kdyby bylo mimo čas. A to nejen proto, že

- samo čas ustanovuje posobnostmi svých projekcí mezi jsoucna, pohybem mezi retencemi a protencí a zároveň i propojováním s asociativními a analogickými intencionálními obsahy;
- tím, že je takto podmínkou výskytovostí (tzn. možností konstitucí věcí a mentálů);

<sup>28</sup> Zdá se, že pro vše vyjmenované je substruktura jáství velkou evoluční výhodou.

- ale především proto, že tuto kontinuitu a návaznost prožitků je schopno v sobě udržet.

Je to stále totéž jáství, které se ustanovuje výsledky projekcí, i když to, co lidské myšlení vytrhuje ze jsoucen, i to, co následně konstituuje, nebo konstruuje prochází změnami.<sup>29</sup>

Nyní po úvaze o uvědomění jako nutné podmínce konstituce mentálů, můžeme pokročit dále. Mentály jsou především vněmy, které jsou konstituovány lidským myšlením souběžně s věcmi jako jejich mentální dvojče, avšak jsou konstituovány nikoli na jsoucnech, ale jen v mysli. Konstituuje-li lidské myšlení věc, zároveň ji reflektuje, vnímá výsledek své činnosti. Tzn., že jde o reflexivní oblouk, zrcadlení konstitucí věcí v mysli, tedy jejich vnímání a uvědomění. Každá konstituce věci je zároveň vněmem. Konstituce mentálů se tedy příliš neliší od konstitucí věcí na jsoucnech. Vytržené stimuly ze jsoucen jsou skládány podle pravidel pohybů, tedy podle operací lidského myšlení do perceptů a dále pak jako paměťové stopy jsou uchovávány v substrukturách mysli. Odlišnosti konstitucí mentálů od konstitucí věcí spočívají v tom, že mentály jsou konstituovány nejen ze jsoucen vytržených stimulů, ale že se do tohoto procesu mísí také již dříve vytvořené intencionální obsahy mysli a také některé z dalších operací lidského myšlení. Tím dochází k tomu, že vněm není vždy naprosto totožný s věcí, i když stále je jejím dvojčetem v časové souběžnosti.

Kromě vněmů (vjemů) však můžeme identifikovat celou řadu dalších mentálů. O některých z nich jsme se již zmínili, jako např. vzpomínka, vůle, ale jde také o imaginace, plány, očekávání, účely, intence atd., zkrátka o naši vnitřní vybavenost, která se kryje s naší představou endoceptivní struktury osobnosti. Ale nejen toto jsou mentály. Je zde další uskupení mentálů např. ve formě znalostí, které je možné také nějakým způsobem zvěcnit a vytvořit z nich artefakt ve formě např. vědecké teorie, uměleckého díla atd. To však již patří do dalších částí funkčního modelu mysli, jak o tom pojednáme později.

Jak je však možné, že něco takového jako mentály, tedy něco, co je jakoby v nějaké pozici, vůbec jest? Tato otázka je znepokojující a proto se jí budeme věnovat. Naše odpověď je taková, že je nutné vycházet z povahy jsoucen. Jsoucná jsou určená myšlením, avšak jejich určení zároveň vytváří strukturu jejich uspořádání, vytváří vztahy mezi jednotlivými jsoucnými. Ve vztazích se pak vyjevují vlastnosti jsoucen, vyjevuje se to, jak se má co k čemu, jak si vzájemně vyhovuje, nebo jak je to k sobě antagonistické, případně až anihilační. Tím se také ustanovuje strukturní uspořádání jsoucen a potenciality jejich vzájemného působení. A stejně je tomu i s lidským myšlením, které je také jsoucnem. Ve vztahu k jiným jsoucnům, v našem případě k mozku a mysli se vyjevují jeho vlastnosti, které něco uspořádávají a vytváří. A tím, co je takto ve vztahu vytvářeno, jsou mentály. Protože však mozek a mysl jsou úžasnou

<sup>29</sup> Snad se čtenáři může zdát, že teze o tom, že jáství je jakoby mimo čas, odporuje tomu, že: *lidské myšlení něco očekává a představuje si nejen jsoucná, ale i sebe sama v čase a časením prostupuje k sobě samotnému*. Nedomníváme se, že jde o rozpor, neboť Jáství je stále identické (tedy jakoby mimo čas) i když se pozměňuje.

a snad obdoby nemající strukturou, mohou být neustále generovány další a další různorodé mentály. Mentály jsou neustále generovány a jsou jsoucími, pokud jsou ve vztahu s lidským myšlením. Pomine-li vztah, kterým byly mentály generovány, pominou i mentály, avšak pokud tento vztah trvá, potud jestvují i mentály a i když jsou v pozadí, lidské myšlení se k nim může dále vztahovat, vyzdvihovat je z pozadí, unášet je a vystavovat vztahům k jiným mentálům a vyjevovat tak jejich další možné souvislosti obdobně jak tomu je u jsoucen. To se však děje podle určených pravidelností pohybu lidského myšlení.

Základní pravidelností při konstitucích světa i mentálů je tedy efektivita pohybu lidského myšlení, využívající již osvědčené vzorce svých pohybů. Již jsme poznamenali, že učinit podrobný popis všech vzorců pohybu lidského myšlení je téměř nemožné, neboť jde vždy o něco, co se utváří nebo alespoň ověřuje prožitkovostí a že nám také o úplný výčet těchto pravidel nejde. Jistě existují některé obecné operace lidského myšlení, i vzorce, na které jsme se ilustrativně snažili poukázat.<sup>30</sup> Domníváme se však, že i tyto operace a vzorce vykazují individuální odchylky tím, jak jsou dotvářeny v individuální endoceptivní struktuře, případně jak jsou ovlivněny aktuálním emocionálním naladěním. Proto každá konstituce a objasnění je vždy střetem něčeho, co je lidskému myšlení vnější, s ním samým a s jeho endoceptivní výbavou. Je nutné také poznamenat, že ne vše, co je vytrženo ze jsoucen, je také složeno a tím konstituováno do podoby věcí a vněmů. Mnohé může v mysli zůstávat v singulární, nebo jen částečně složené, tzn. fragmentární podobě,<sup>31</sup> a protože tyto stimuly nenalezly své umístění v celku věcí, nestávají se součástí vněmů a dostávají se na periferie mysli (do nevědomých struktur), kde mohou čekat na dodatečné využití, např. při změně endoceptivního modelu světa. Mysl je v takovýchto případech obtěžkaná mentály, které pochází ze jsoucen, jsou neuvědomované, ale působí na další průběh lidského myšlení.

### 3 Operace lidského myšlení a intencionální obsahy mysli

#### 3.1 Operace lidského myšlení

Lidské myšlení je v neustálém pohybu, je pohybem. V našich úvahách o pohybech lidského myšlení se nebudeme přidržovat dnes již tradičního dělení na diskursivní a divergentní (někdy také označované jako laterální) formy, neboť podle našeho názoru je mezi těmito formami hlavní rozdíl v tom, že jde o uvědomované operace a operace, které nejsou plně uvědomované. To, zda se jedná o postup krok za krokem, nebo zda jde o paralelní operace, jejich zaměřenost na určitý cíl, nebo nezacílenost, nic nemění na tom, o jaké základní operace pohybu lidského myšlení jde. Proto je podle našeho

<sup>30</sup> Srovnej: [22]

<sup>31</sup> Obdobně tomuto Reiser tvrdí, že představy jsou rozložitelné na fragmenty a mohou být znovu přeuspořádány např. ve snech. (*It is known that individual images may be fragmented into parts and the parts then used separately in creating a dream image.*) [25] (s. 96)

názoru důležitější uvažovat o tom, co lidské myšlení při těchto operacích dělá, jak se pohybuje, co mluví. Pohyb lidského myšlení se řídí osvědčenými algoritmy nebo rozsáhlejšími vzorci, které jsou efektivními zkratkami šetřícími mentální energii. Basicke formy operací lidského myšlení jsme odvodili z vědeckých metod a nyní je shrneme v následující tabulce, v níž se pokazuje na průběh pohybu, který je mluvením.

<b>Operace lidského myšlení:</b>	<b>Popis:</b>
Negativita	<i>To není to</i>
Identita	<i>To je to</i>
Komparace	<i>To je jako to; nebo to není jako to</i>
Generalizace	<i>Tyto mají společné toto</i>
Abstrakce	<i>Toto charakterizuje to a toto necharakterizuje to; bez toho je toto stále tímto</i>
Symetrie	<i>To nastane proto, že takto je to v rovnováze k onomu</i>
Extrapolace	<i>Toto bude pokračovat takto, protože pokračování je funkcí onoho</i>
Eliminace	<i>Toto nepatří k oněm, protože se nevyznačuje tímto; nebo se vyznačuje tímto, což ony nemají</i>
Kombinatorika	<i>Tyto lze zkombinovat do tolika těchto; toto lze doplnit tímto, abychom získali ono;</i>
Trojčlenka	<i>Toto se má k tomu, jako ono se bude mít k onomu</i>
Sylogismus	<i>Jestliže to se má k onomu jako ono se má k tamtomu, pak to se má stejně k tamtomu</i>
???	???

Ve výše uvedeném souhrnu operací lidského myšlení jsou znázorněny tři úrovně, z nichž první zahrnuje operace negativy a identity. Při tomto pohybu je lidské myšlení nejvolnější a může unášet cokoli a operovat s tím tak, že postupně vytváří kategorizaci všeho, s čím se střetlo. Tato kategorizace mu umožňuje zefektivnit jeho pohyb, takže vše, co již bylo nějak zařazeno do struktury mentálů, je dále využíváno a lidské

myšlení nemusí každý ze stimulů poměřovat tím, že mluví: *to je to; a to není to*. Tento pohyb je však základním a lidské myšlení se k němu může kdykoli uchýlit.<sup>32</sup>

Další úroveň operací (komparace, abstrakce a generalizace) je úroveň vyjadřující tvorbu kategoriální struktury mentálů. Abstrakce a generalizace vytváří matrjožkovitou strukturu mentálních obsahů tím, že umožňuje pohyb na úrovni podstat a obecnin; a komparace naznačuje cestu pohybu lidského myšlení, které srovnáváním snadněji zhodnotí stimul, neboť je již zařaditelný podle podstaty a obecnosti. Lidské myšlení tak může se stimulem dále operovat a vytvářet složitější mentální celky. Těmito operacemi se také stimuly proměňují v pojmy, které již mají intence.<sup>33</sup>

Třetí úroveň je úroveň, při níž jsou syntetizovány složitější celky. Zároveň obsahuje i to, které z operací jsou k syntézám využívány. Lidské myšlení zde opouští basický pohyb identity a negativity a operuje s celými složitějšími strukturami operací najednou. Tyto operace jdou sice vyjádřit podle basických operací, avšak lidské myšlení je využívá jako celky. Tak např. sylogismus není ničím jiným než trojnásobnou implikací a jednou konjunkcí, což lze ztvárnit prostřednictvím identity a negativity.

Zdá se, že nad těmito operacemi jsou ještě další schémata nebo vzorce operací lidského myšlení, dále pak celé scénáře a v neposlední řadě dokonce celé narativní celky, které mají memetickou (nebo z části i archetypální) povahu a jsou opět komplexnější a umožňují lidskému myšlení, aby je najednou uchopilo a aplikovalo na situace, v nichž se ocitá, nebo na zpracování komplexnějších celků.

Vrcholnou fází operací lidského myšlení je ta, kterou obvykle nazýváme intuicí. V této operaci se najednou proloupe všechny předchozí. Jde o náhlou uchopení celku a jeho prostoupení s tím, že v náhlosti vše pochopíme. Intuici můžeme opět vysvětlit několika způsoby. Zdá se, že buď může jít o zkratky v průběhu lidského myšlení; nebo může jít o operace, které jsou velmi podobné tomu, co se děje v myslích autistických jedinců při jejich famózních výpočetních schopnostech; intuici lze také vysvětlit apriorními, či archetypálními základy myšlenkových procesů; další z vysvětlení nabízí teorie hologramu, kdy z jednoho poznatku docházíme k poznání celku; a konečně pátým způsobem vysvětlení intuice je koncepce transcendentní, při níž jakoby výsledek již byl předem dán a my se jen k němu dobíráme.

### 3.2 Intencionální obsahy lidského myšlení

Intencionalita je vždy zvýznamněním pojmu lidským myšlením, což znamená, že jde vždy o odkazování pojmu k pojmenované věci nebo mentálu a zároveň i o vřazení pojmu do struktury souvisejících pojmů tak, jak je tomu s věcmi, které mají vždy určitou pozici ve struktuře světa. Intencionalita zároveň vždy poukazuje i na to, jak lidské myšlení konstituuje věci skrze svou endoceptivní strukturu, v níž jsou obsaženy

<sup>32</sup> Právě v těchto operacích spočívá úspěšnost výpočetní techniky, která disponuje jen těmito operacemi; řečeno slovy počítačových odborníků: vše je jen relace: vypnuto – zapnuto.

<sup>33</sup> Intence, podle našeho názoru, je ustanovována až zařazením do struktury endoceptů. Nejedná se tedy o pouhý vztah vědomí k něčemu, který může být ustanovován již na úrovni vnímání, tedy o to, že jde o uvědomění něčeho.

plány, očekávání, ambice, starost, zkušenost, scénáře a schémata možných konstitucí a objasnění věcí, dlouhodobá paměť atd. Právě v této endoceptivní struktuře dochází k procesům zvýznamnění struktury věcí a tím také pojmů, které strukturu věcí kopírují v jazyce.

Intencionalita je však také úzce vázána na uvědomění toho, jaké místo pojem ve struktuře intencionálních obsahů myslí zaujímá. Uvědomění zde znamená, že lidské myšlení reflektuje, do jaké struktury pojem vřazuje a zároveň i to, co je pojmem pojmenováno a tedy také, zda struktura světa a struktura pojmů jsou v koincidenci. Musí tedy znovu jít o schopnost lidského myšlení rozštěpit se ve dva proudy, které současně operují na dvou úrovních.

Povšimněme si však blíže, jak je intencionalita pojmů vázána na:

- komplexitu
- adekváci (onomatopoeitiku) a
- kontexty.

To jsou pro nás tři rozhodující vrstvy vazeb, v nichž se intencionalita nejen vyjevuje, ale i konstituuje. Jde tedy o strukturu, do níž, teprve když je pojem vřazen, nabývá intencionální obsah. Tato struktura je také neustále oživována prožitky tak, že každý další prožitek potvrzuje nebo doplňuje (výjimečně i mění) intencionální zařazení pojmu.

**Komplexitou** rozumíme vztaženost každého pojmu k jiným pojmům a to:

- asociativně,
- analogicky, nebo
- logicko-syntakticky souvisejícím pojmům.

Ve velké většině případů jde především o asociativní vazby naplňující intencionální obsah pojmu. Každý pojem je v asociativní vazbě buď na základě styčnosti, podobnosti nebo kontrastu.<sup>34</sup>

Když lidské myšlení asociativně operuje s pojmy, pak přechází z původně myšleného pojmu k jinému, který má s původním intencionálním obsahem nějakou souvislost. Lidské myšlení však mohlo přejít úplně jinam, k jinému pojmu s jiným obsahem. Toto spojení však bylo inhibováno. To je způsobováno bohatostí matrižkovitě ohraničených pojmů. Asociace vřazuje pojem do struktury dalších

<sup>34</sup> Při **styčnosti** spojujeme třeba určitou osobu s jejím jménem, nebo spojujeme *blesk* a *hrom*, *řeku* a *lod'*, tedy vše, s čím jsme se ve svém životě setkali v bezprostředních, nebo i vzdálenějších souvislostech. Myšlenka nebo představa je vyvolána souvislostmi, které na sebe nějak navazují. Při **podobnosti** je spojení myšlenek dáno reálnou podobou. Tak může *děšť* vyvolat asociaci *sněhu*, *cigareta* vyvolá asociaci *doutníku*. Ale může dojít také k tomu, že vysloví-li někdo slovo *koleje* a má při tom na mysli *železniční koleje*, vyvolá v nás představu *studentských ubytoven*. Určitá myšlenka může vyvolat jinou také na základě **kontrastu**. To jsou případy vztahu mezi *černým* a *bílým*, nebo když slyšíme slovo *sever*, může to vyvolat kontrastní slovo *jih*, a podobně.

pojmu, které již s sebou nesou intencionální obsahy mysli, a tím spoluvytváří intencionální obsah daného pojmu. Tyto výkony jsou prováděny lidským myšlením automaticky, bez reflexe jednotlivých asociativních propojení. Lidské myšlení tedy myslí pojem vždy takto strukturně v komplexitě všech jeho možných stavů. A to, že pojem je myšlen v asociativních operacích, přičleňuje určitému pojmu intencionalitu, v níž je pojem k dispozici pro lidské myšlení a jeho následné operace. Pojem je takto zvýznamněn dalšími souvisejícími, podobnými nebo kontrastními pojmy. Vyjevuje se tedy ve své komplexitě.

Totéž, co platí o asociacích, platí také o analogiích, neboť i tyto začleňují pojem do strukturních vazeb, i když jinak, než je tomu při asociativních vazbách. Jestliže si asociace můžeme připodobnit zřetězení a průběžnost operací lidského myšlení v asociacích jako přechody po jednotlivých člancích řetězů (i s tím, že na těchto cestách je mnoho křížovatek), pak analogie je takový postup lidského myšlení, v němž obdobnost a vázanost pojmu má podobu celé plochy, pole, i když přece jen v mnohém ohraničenou. Tato ohraničenost může být ve vlastnostech a vztazích, ale také v nutné a opakovatelné následnosti, ve způsobech objasnění, nebo se může také týkat dynamiky vývoje toho, jaké stimuly lidské myšlení vytrhne ze jsoucen a jaké endoceptivní obsahy jsou uchovány v mysli. Při analogických myšlenkových operacích poznáme vlastnosti a vztahy určitých pojmů přenášíme i na jiné pojmy, které tím nabírají na významnosti, získávají intencionalitu. Analogie působí jako generátor intencionality, když zahrnuje co největší množství podstatných souvislostí. V takovém případě může určitá poznaná skutečnost ozvučit (reverberate) nereflektované, ale související operace lidského myšlení, a tím naplnit intencionální obsah pojmu. K tomu však je nutné, aby analog:<sup>35</sup>

- byl co nejčetnější a zároveň se týkal podstatných vlastností pojmu;
- z toho plyne také nutnost těsného spojení mezi vlastnostmi pojmu a analogu;
- je nutné brát v úvahu také vše, čím se pojem a analog odlišují.

Třetím faktorem komplexity, podílejícím se na intencionalitě pojmu, je jeho logicko-syntaktická vázanost. Zde jsou rozhodující souvislosti logické, kdy pojem může být jen v limitovaných vztazích k jiným pojmům a tato omezenost je dána logickými operacemi lidského myšlení. A totéž, co platí o omezenosti vazby konstituce intencionality při tvorbě pojmu v logických vztazích, platí rovněž i ve vazbách syntaktických. Také tyto vazby vymezují významovost pojmu. Lze to uvést mnoha příklady, které jsou založeny na homonymech, na jejichž základě je postaveno mnoho anekdot. Mnohdy však význam může být určen dokonce interpunkcí nebo intonací.<sup>36</sup>

Adekvacie vymezuje intencionalitu především shodností znění s obsahem daného slova, tedy souzvučností. Jde tedy o onomatopoesis. Souzvučností (libozvučností) se

<sup>35</sup> Analogem rozumíme něco, k čemu je pojem v analogii odkazován.

<sup>36</sup> Jedním z často uváděných příkladů je historka o tom, když král napsal jako odpověď na katovu otázku, zda mají odsouzeného popravít, nebo propustit následující rozsudek: *propustit ne popravít*.

zároveň zpřesňuje intencionální obsah pojmu. Adekváci lze ilustrovat řadou příkladů. Např. slovo *skomírání* vykazuje svůj intencionální obsah tím, že navozuje slova: *skoro* a *umírání*, a tak předurčuje silnější intencionální vazby. Toto vykazování není bezprostřední, musí být domyšleno a toto domyšlení a tím i propátrávání intencionálních obsahů myslí zároveň dokresluje intencionální obsah původního slova (místo *skomírání* bychom mohli použít jiné slovo, např. *únava*, avšak toto by již nabývalo přece jen trochu odlišného významu v určité větě). Obdobně je tomu i u mnoha dalších slov jakými jsou *stanovisko*, ve vazbě na *stanoviště*; *názor*, ve vazbě na *zření*, atd. Adekvacie navozuje hlavní asociativní vazby, a tím omezuje mlhavost asociativních a analogických vazeb intencionality.

Kontext se rovněž výrazným způsobem spolupodílí na tvorbě intencionality. Kontext zahrnuje nejen komplexitu při tvorbě pojmu, ale širší pole, v němž se na konstituci významu podílí situace, v níž se lidské myšlení nachází při použití konkrétního pojmu. Ve filosofické literatuře je tato problematika iniciovaná, a i když fragmentárně, přece jen na mnoha příkladech analyzovaná L. Wittgensteinem a nemáme k ní příliš co dodat. Lidské myšlení je vždy v nějaké situaci, neboť jeho pohybu něco předcházelo a něco vždy následuje a toto okolí se spolupodílí na intencích tohoto pohybu a přispívá k porozumění.

Intencionalitu je nutné brát nikoli jako neustálé přiřazování a pouhou strukturnost významu pojmů, ale spíše jako syntézy z navazujících pojmů. Tento výčet konstitutivních charakteristik intencionality pojmu není úplný. Intencionalita je velmi složitou problematikou a je obtížně stanovitelné, jakými prostředky se k ní lidské myšlení dopracovává. Její mapování však může napomoci také při formování např. sémantického webu.

#### 4 Tvorba novosti

O výsledcích reakcí lidského myšlení na mentály pojednáme v následujících pasážích. Již předem je však zřejmé, že výsledek těchto reakcí nemá jedinou podobu, ale že tvorba zaujímá široké a obtížně dohlédnutelné pole, neboť jde o tvorbu, která přináší neočekávané novosti a proto je velmi těžko předvídatelná. Přesto se pokusíme (s plným vědomím toho, že tvorbu nelze katalogizovat) o utřídění výsledků tvorby.

tvorba novostí je tvorbou nového světa ve formě kvazipozicionality i pozicionality ; tyto tvorby jsou účelově zaměřené  
výsledky tvorby mohou být fluxní, nebo petrifikované obsažené v dílech

*i konkrétní podoba fluxní tvorby může mít podobu:*

- rozhodnutí;
- řešení problémů; nebo
- konstrukce novosti (invenční nápad, obrazotvornost, atd.).



ii nejvýznamnějšími petrifikovanými formami tvorby jsou díla z oblastí:

- filosofie;
- náboženství;
- politiky;
- umění; a
- vědy.

Tvorba je pro lidské myšlení novým pohybem (pohybem po dosud neproběhnutých cestách), nebo uplatněním již osvědčeného pohybu na novou situaci, v níž se lidské myšlení ocitá. Takto je možné hovořit o individuálním základu tvorby. Lidské myšlení vždy disponuje intencionálními obsahy mysli. Pro tvorbu novostí jsou důležité nejen jejich množství, ale také a snad i především jejich vzájemné vztahy a souvislosti, neboť to tvoří základ pro další možné kombinace intencionálních obsahů a tedy i novostí, které jsou oním **moci být** jakožto nalézání smyslu.

Pokud lidské myšlení disponuje bohatou zásobou intencionálních obsahů a pokud tyto obsahy jsou ve stabilní struktuře a vzájemných vztazích uspořádány tak, že umožňují lidskému myšlení velké množství konstitucí a objasnění světa (pokud tedy lidské myšlení disponuje obsáhlými znalostmi), může na mnohé situace jen aplikovat již osvojené pohyby a nedostává se do situací, kdy potřebuje vyhledávat tvořivé pohyby a vytvářet nové kombinace a vztahy v endoceptivní struktuře intencionálních obsahů (i když i tyto pohyby jsou neustále prozkoušovány sekundárním proudem). Takto osvojování si znalostí vede paradoxně k situacím vyžadujícím malou míru tvořivosti, k situacím, kdy lidské myšlení šetří potenciál mentální energie a uchyluje se k algoritmickým, zkušenostním pohybům.

Jestliže si však lidské myšlení osvojilo jen určité algoritmické pohyby, které mu sice dostačují ke konstitucím světa a jeho objasnění, avšak v poměrně stabilním prostředí, když je tedy chudé na znalosti, pak každá nová situace, do níž se lidské myšlení dostane, vyžaduje nové pohyby, a tedy je lidské myšlení nuceno k tvořivosti.<sup>37</sup> Lidské myšlení se dostává do tlaku (tzn. presu, který je stresem) a musí se se situací vyrovnat.<sup>38</sup> Tvořivost však v tomto případě má jen chabou znalostní základnu. Vztahy mezi jednotlivými intencionálními obsahy se mnohdy dostávají do kontradikcí, a tak i

<sup>37</sup> Domníváme se, že tyto procesy jsou nejčastější v raném věku, kdy téměř vše je nové a algoritmické pohyby ještě neexistují. Postupně se však vytváří znalostní struktura, která dovoluje využívat již osvojené a tvořivost je vyhrazena jen pro další novosti. Tím chceme naznačit, že člověk s malou znalostní výbavou je tvořivější, avšak jeho tvořivost novostí je na poměrně nízké úrovni, neboť se zabývá jen malým počtem primitivních intencionálních obsahů. Tvořivost těch, kteří disponují hlubokou znalostní strukturou je namáhavější a také překonává historické bariéry, neboť obohacuje nejen jedince při jeho konstitutivních a objasňujících aktivitách, ale také celou komunitu těch, kteří se na novostech mohou podílet.

<sup>38</sup> Stres může působit jako generátor tvořivých procesů, ale může také zcela zablokovat invenční pohyby lidského myšlení a to pak upadá do apatie.

nově vznikající kombinace intencionálních obsahů nemusí dostačovat na řešení vzniklé situace.

Ideálním případem je ten, kdy je dostatečná znalostní základna a kdy pohyb lidského myšlení po nových drahách není jen vynucován situacemi, do nichž se lidské myšlení dostává (zejména při svých projekcích mezi jsoucna), ale kdy nové, invenční pohyby jsou **prozkoušováním možností jak moci být**. Takovýto pohyb může být iniciován sekundárním proudem, který vždy pozoruje i další možné souvislosti kombinací a donucuje primární proud k vychýlení od běžného průběhu.

Zdálo by se, že tvořivost a tím i nabývání nových tvarů struktury intencionálních obsahů myslí je jen záležitostí přeuspořádání již známého. K tomuto je nutné učinit poznámku, která se nutně vztahuje k ontologickým základům lidského myšlení.

Naše domněnka týkající se charakteru jsoucna je taková, že mají jednotný základ v tom, že jestvují a že kvalit jsoucna není mnoho, případně že je jen jedna a že jsoucna jsou určená (konkretizovaná) protoperacemi pohybu myšlení. Jestliže tedy vycházíme z tohoto ontologického základu, pak různorodost věcí konstituovaných lidským myšlením je nutně jen kombinací poměrů něčeho, co je jedno (kombinací poměrů ze jsoucna vytržených stimulů) a že tedy novosti jsou jen dalším strukturálním přeuspořádáním téhož, případně jen nalezením již něčeho, co jest. Problém je však v tom, že novosti jsou vždy dalším a dalším posunutím na strukturálním stromu. Z jednoho (z potenciality jsoucna, tedy z určité jedinečné kvality) se stane něco dalšího a odlišného svou strukturou a tedy i kvalitou, nejedná se tedy již o jsoucno, ale o věc. Tato kvalita je již složitější (nebo jiná) než všechny předchozí a na této složitější kvalitě je pak postavena další možnost tvorby tím, že ji přeuspořádáme. To se děje např. když z věci konstruujeme předmět. Tak máme původní kvalitu jsoucna a vedle ní něco dalšího (další složitější nebo jinou kvalitu) a jejich kombinacemi se dostáváme k dalším možnostem skutečnosti. Původní jednotný základ jsoucna pak již lidské myšlení nedohlží. Celý výše popsaný proces tvorby by šel znázornit obvyklou metaforou pyramidy obrácené na špičku, ilustrující rozrůstající se možnosti jak pohybu lidského myšlení, tak i tvorby novostí.

Proto se domníváme, že z hlediska pohybu lidského myšlení při tvorbě můžeme jen s obtížemi hovořit o pouhém přeuspořádání struktury, protože zde vyvstává nutnost nejen nově něco zkonstruovat, ale také syntetizovat (pro výsledek tvorby) z vhodných stimulů nebo endoceptů. Mnoho z těchto procesů tvorby je vykonáváno neuvědomovaně sekundárním proudem lidského myšlení, jako paralelní proces při konstitucích věcí a mentálů.

#### 4.1 Rozhodnutí

Jeden ze základních tvořivých pohybů lidského myšlení můžeme spatřovat v rozhodování. Na rozhodovací proces se můžeme dívat z několika hledisek, a sice podle toho:

- zda lidské myšlení při rozhodnutí vychází ze zkušenosti, kterou algoritmizuje; nebo
- zda je rozhodnutí založeno na okamžitých pocitech a emocích.

V prvním případě se lidské myšlení pohybuje tak, že vyhledává ve zkušenostní složce endoceptivní struktury obdobné situace a zvažuje, zda dřívější rozhodnutí bylo úspěšné, či zda vyžaduje nějakou úpravu. V případech, kdy minulostní pohyb lidského myšlení byl úspěšný, bude i aktuálně generován stejný, nebo obdobný pohyb lidského myšlení, neboť situace již byla jednou prožitá a aktuálně se vše jen opakuje. Rozhodnutí závisí především na vyhledávacích operacích sekundárního proudu lidského myšlení v endoceptivní struktuře, kde jsou obsaženy nejen obdobné situace v dlouhodobé paměti, ale i výsledky minulých rozhodnutí a jejich zhodnocení. V případech, že zkušenostní složka endoceptivní struktury nedisponuje v minulosti vykonaným úspěšným rozhodnutím, lidské myšlení sleduje, zda nenalezne nějaké rozhodnutí, které by šlo v aktuální situaci napodobit, zda tedy něco obdobného není k dispozici v rozhodnutích např. jiných subjektů. Lidské myšlení tedy zjišťuje, jaká rozhodnutí jsou generována v obdobných situacích jinými lidmi. Tím se přiklání k všeobecně a osvojuje si ho a uplatňuje ho v aktualizaci svého rozhodnutí. Všechny tyto případy mají svůj základ v důvěryhodnosti a v průměrnosti pochopené jako střed všech odchylek.

Jinou otázkou však je, zda by v dané situaci nemohlo být efektivnějším (jak z hlediska dosahování aktuálně sledovaného účelu, tak i z hlediska realizace absolutních hodnot) nějaké jiné rozhodnutí, zda by si lidské myšlení nemohlo osvojit jiné pohyby a tím dosáhnout cíl např. s menší náročností, nebo dosáhnout naplnění účelu v kratším čase (tzn. při menším počtu projekcí), atd.

Nutnost inovačního rozhodnutí souvisí s tím, je-li situace zcela nová, nebo je-li účel, jehož se má dosáhnout nějak modifikován. V těchto případech musí lidské myšlení vytvořit kalkul, kterým zhodnocuje jednotlivé kroky vedoucí k dosažení cíle. Evaluace si musí především stanovit, s čím lze danou situaci nebo dané postupy vedoucí k naplnění účelu porovnávat. V druhé fázi pak musí lidské myšlení tyto kroky vyhodnocovat. Lidské myšlení si tedy stanoví různé scénáře postupů, jak by situace mohly vypadat a vše další je záležitostí kalkulu, který upřednostňuje některý z předpřipravených scénářů.

Něco jiného ovšem je, když žádné z rozhodnutí není lidskému myšlení k dispozici a nějaké rozhodnutí musí být vykonáno. Pokud není lidské myšlení v tísní, může začít s experimentem typu pokus – omyl a může zjišťovat, nakolik bylo rozhodnutí adekvátní dané situaci a sledovanému cíli. Takovýto experiment může být realizován jako konstitutivní nebo jako čistě mentální akt. Lidské myšlení při tom jednotlivé kroky zhodnocuje. To se může dít tak, že pokusy a omyly jsou kvantifikovány a pak následně komparovány, čímž je získán předpoklad k propočítání nejnvýhodnější cesty k dosažení cíle. Někdy však nevíme, jaký pokus by měl být vykonán a zde přichází v úvahu axiomatická inference, což znamená, že si lidské myšlení stanoví některé z předpokladů a pomocí odvozování pak vytváří deduktivní vyvození cíle.

Jinak ovšem tomu je, když je lidské myšlení v tísní a musí učinit nějaké rozhodnutí. V takovýchto situacích patrně metoda pokusu a omylu nevystačí a kalkul nemůže být pro jeho zdoluhavost realizován. Přesto k nějakému rozhodnutí dochází a nás zde zajímá, na čem se takovéto rozhodnutí může zakládat. Chceme při této příležitosti upozornit na koncepci **somatických markerů**, které podle názoru A. Damasia [6] signalizují pro naši mysl nejen situaci, v níž se lidské myšlení aktuálně

nachází, ale které také téměř vždy adekvátně (a lépe než jakýkoli uvědomovaný kalkul) zhodnocují situaci a napovídají, jaké rozhodnutí musíme učinit.

V našem funkčním modelu myslí je tělo zastoupeno jen stimuly, které vytrhují interoceptory a propioceptory. Zdůrazňujeme, že tyto stimuly jsou lidským myšlením přednostně zhodnocovány, neboť se úzce váží k instinktivní a emocionální rovině pohybu lidského myšlení a proto jsou také v procesu rozhodování důraznější a rychlejší než intelektový zhodnocovací kalkul. Propočty pro rozhodnutí se mohou uplatnit vždy jen tak, že jsou buď zabudovány už jako vzorce v endoceptivní struktuře a tudíž předem zaměřují projekce lidského myšlení a předjímají výsledky jeho aktivit; nebo tak, že vetují instinktivní a emocionální zhodnocení mysli, čímž i vychylují pohyb lidského myšlení z této roviny.

Zdá se tedy, že somatické markery mohou být velmi dobrými navigátory při rozhodnutích a i když si neuvědomujeme všechny průběhy lidského myšlení při práci s interocepty a propiocepty, lidské myšlení je vždy zvažuje a přivádí, i když třeba jen v náznacích k uvědomění.<sup>39</sup> I když nevíme, co to vše znamená, je dobře si při rozhodnutích těchto somatických markerů všimnout a uvážít oč vlastně jde.

Při rozhodování jde o vytvoření křižovatky času, tedy směru budoucích projekcí a konstitucí. Jde tedy o mapování určitých možností být s ohledem na aktuální situace, v nichž se lidské myšlení nachází. Není to čas, který se rozštěpuje, ale jde o časový okamžik (o konkrétní projekci), který nasměruje další cesty při konstitucích světa a zaujímání místa v něm. Tak dochází k proměnám v samých základech toho, jak bude svět nahlížen, s jakými endocepty se bude lidské myšlení projikovat mezi jsoucna a vytrhovat z nich stimuly. Putování lidského myšlení se tedy při rozhodnutích dostává na křižovatky, které se nám zdají být křižovatkami v čase, a které vytváří další možnosti budoucích projekcí, tzn. zkusit jestvovat jinak, otevřít další ze svých modů a realizovat se v jeho rámci.

## 4.2 Řešení problémů

Jiným fluxním výsledkem tvorby je řešení problémů. Zde můžeme začít s úvahou o hádankách a o možnostech jak uplatnit pohyby lidského myšlení, které se objevují při řešení hádanek také při jiných příležitostech. Jednou z klasických hádanek byla otázka, kterou položila Sfinga Oidipovi, když přišel do Théb.<sup>40</sup> Tato hádanka symbolicky poukazuje na to, že rozřešení jakékoli hádanky, záhady, problému je velmi mocné, neboť je osvobozující (sfinga se zhroutí do propasti a Théby jsou osvobozené). Záhady mizí, když jim přijdeme na kloub, když si nějak pomůžeme, když jsme šikovní.<sup>41</sup>

<sup>39</sup> Např. tak, že máme zrychlený tep, nebo tlak ve střevech, nebo zčervenáme, nebo máme husí kůži atd. Všechny tyto tělesné znaky rozpoznáme a snad nám i napoví, jaké rozhodnutí je zapotřebí udělat. Zde se však již opět dostáváme do oblasti zkušenostních identifikací při rozhodnutích. Tentokrát jde o zkušenosti se zážitky vlastního těla.

<sup>40</sup> Kdo to je: ráno chodí o čtyřech, v poledne po dvou a večer o třech? Odpověď: člověk, který se v mládí batolí, v dospělosti chodí vzpřímeně a ve stáří potřebuje k chůzi hůl. Calasso uvádí následující modifikaci této hádanky: *Co je to za tvora, který má jediný hlas, ale nohy někdy dvě, někdy tři, někdy čtyři, a který je tím slabší, čím četnější jsou jeho nohy?* [5]

<sup>41</sup> T. S. Kuhn ostatně považuje činnost vědců v období normální vědy za řešení hádanek. Dovést

Co to však znamená. Hádanka je mnohdy metaforická a k jejímu vyluštění musíme odhalit správný a předem zamýšlený význam mnohoznačné metafory. Hádanky jsou lidskou tvorbou, ale s problémy obdobnými hádankám se můžeme setkat při projekcích mezi jsoucna, nebo při konstrukcích předmětů. Znamená to, že jsoucna jsou také metaforická (?), nebo jsou pro lidské myšlení jen zahalená jeho vlastními nedostatečnostmi.

Uvažujme např. o biologické vazbě lidského myšlení a o všem, co se odehrává ani ne v celém mozku, ale jen v jednotlivém neuronu, který předává stimul dalšímu neuronu. Jde o to, jak neuron uvolňuje neurotransmitéry (a navíc, kolik těchto neurotransmiterů existuje a jak každý z nich způsobuje jiné reakce), jak signál putuje synapsí a jak musí mít dendritický vstup do dalšího neuronu připraven receptory na určitý neurotransmitér, aby signál mohl být přepraven do těla buňky a jak zde mRNA pořádá peptidové vazby, aby se budovala příslušná bílkovina odpovídající vstupujícímu signálu a přepólovala plasma, atd. Mnozí badatele řešící tento problém (hádku) se dostávají ještě dále a uvažují o roli mikrotubulů a bioelektrickém excitačním potenciálu atd. To vše se odehrává v celých populacích neuronů nejen tehdy, když cokoli děláme, ale dokonce i tehdy, když jsme (zdánlivě) v klidu (neboť lidské myšlení není nikdy v klidu, je pohybem). A dát dohromady něco, co by vysvětlilo, alespoň nutné a dostatečné vztahy a reakce, alespoň v malé substruktuře mozku a zdůvodnilo tím, proč stimul excitoval konkrétní populaci neuronů a proč tato excitace měla za následek takové chování, je téměř neřešitelným problémem (hádkou). A to ještě nevíme, zda už známe všechny elementy této skládky, neboť nevíme, co je to vědomí a zda hraje kauzační roli vůči mozku. Proti tomu je hádanka Sfingy úlohou hravou a určenou pro děti.

Problém vymezujeme jako přesně ohraničený úsek neznámého. To znamená, že lidské myšlení se projikuje mezi jsoucna a vytrhuje stimuly, které:

- neumí uspořádat tak, jak je vytrhlo; nebo
- nezná jakou roli, či funkci má některý ze stimulů v celku věci; nebo
- dochází k tomu, že něco chybí, nebo něco nadbývá, nebo něco do sebe nezapadá a tak dané uspořádání při konstitucích (nebo konstrukcích) nedává očekávaný výsledek.

Pod řešením problému spatřujeme tedy přeuspořádání jednotlivých prvků tak, aby výsledkem byl požadovaný účel. Zdálo by se tedy, že jde jen o mechanickou činnost, která při konečném počtu elementů musí mít vždy nějaké řešení. Domníváme se však, že mnohdy jde o něco komplikovanějšího, neboť nejprve musíme některé elementy

---

problém normálního výzkumu k řešení znamená dosáhnout toho, co již bylo předjímano, novým způsobem. To vyžaduje vyřešení všemožných složitých přístrojových, pojmových a matematických hádanek. Člověk, kterému se to podaří, se sám prokazuje jako odborník na řešení hádanek a hádanky jsou výzvou, jež je důležitou částí pohnutek, které jej ženou vpřed.  
[16]

konstruovat tak, aby měly konkrétní představované vlastnosti a mohly plnit funkci, která je pro záměr nejen vhodná, ale i nutná. Mnohdy tedy jde o postupné skládání, při němž se nová vlastnost teprve vynoří, a ne vždy ji můžeme předem předpovídat.

Takovýto proces je v současnosti nejčastěji nazýván **emergencí**. Základem koncepce emergence z ontologického hlediska je teze vyjádřená slovy K. R. Poppera: *existují skutečné novosti*, to znamená, že jde o takové věci nebo předměty, které jsou samostatné a neredukovatelné na své části, ačkoli jsou jen z těchto částí složeny.<sup>42</sup>

Zdá se, že takovýto předstupěň (tzn. jistá míra seskupené složitosti) je mnohdy nezbytný z hlediska řešení problému a že ne vždy analýza jdoucí až k elementům (mnohdy by stačilo jít jen k entitám, které jsou prostředníkem mezi elementy a celkem) má vysvětlující povahu umožňující následné řešení problému. Další obtíž však nastává tehdy, kdy emergentní jev nemá požadovanou vlastnost a nemůže tedy plnit svou funkci v nově ustanoveném celku. Jestliže se tedy vynořuje nějaká nová vlastnost, je jen velmi obtížné predikovat, jaká tato vlastnost bude. Zde by snad mohl pomoci jeden z principů hegelovské dialektiky, kdy nově vznikající vlastnost by měla být protikladem předchozího stavu, avšak v emergentních procesech může jít o procesy zcela neočekávané, nezjevující jen očekávané nebo požadované kvality, ale i kvality, které vzhledem k účelu mohou být katastrofální.

Řešení problému je tvorbou, neboť vytváří nový postoj k něčemu, s čím se lidské myšlení setká a na co dosud nezná odpověď. Je rozlousknutím něčeho (problému) a odhalením jádra.<sup>43</sup>

<sup>42</sup> Emergenci jako pojem objasňuje I. M. Havel ve třech významech:

v evolučním významu, kdy vznik systému nemůže být predikován nebo vysvětlen z antecedentních podmínek;

v ne-redukčním významu vysvětlení jevů v ontologickém smyslu, kdy jevy se přirozeně vyskytují na určité úrovni, ale něco znemožňuje jejich redukci na příslušné nižší úrovni (v této souvislosti I. M. Havel uvádí příklad tekutosti vody, ačkoli molekula vody není tekutá; nebo demokracii ve společnosti, kdy jedinec není schopen vytvořit demokratický systém);

v ne-redukčním významu vysvětlení jevů v gnoseologickém smyslu, kdy vlastnost specifikované teorie  $T_1$  je emergentní, jestliže má reálný příklad, jestliže je spolu-případem s nějakou vlastností v redukované teorii  $T_2$ , ale nemůže být redukována k žádné vlastnosti definované v  $T_1$ .

Srovnej: [10]

I. M. Havel dále píše: Kolektivní fenomény se typicky vyskytují při velkém nahromadění individuálních jednotek, kdy chování každé z jednotky zčásti závisí na chování některých jiných sousedních jednotek. [11]

Také O. Lange se domnívá, že vznik nových vlastností je spojen nejen s nahromaděním a vzájemným ovlivňováním jednotek, ale vyplývá také ze struktury celku. Nové vlastnosti jsou pak funkcí systému jako celku. Tytéž prvky, s týmiž způsoby činnosti spojené do odlišných struktur vytvářejí vždy specifické systémy, které mají odlišné způsoby činnosti, neboli změna struktury vyvolává změnu způsobu činnosti celku. Srovnej: [17]

<sup>43</sup> Vzdáváme tímto hold českému jazyku. Lusk obsahuje uvnitř zabalená jádra a je nejprve nutné lusk rozlousknout, abychom jádra mohli vyjmout. Také problémy nebo hádanky se musí stejným způsobem rozlousknout.

### 4.3 Tvorba novostí

Novost je něco, co lidské myšlení dosud nekonstituovalo nebo nekonstruovalo. Takováto tvorba je vysloveně individuální záležitostí.<sup>44</sup> Může to však být něco, co již bylo konstituováno nebo konstruováno jiným lidským myšlením, ale přesto můžeme hovořit o tvorbě novosti, novém vyřešení problému, nebo o novosti v rozhodnutí. Takováto tvorba může mít nejrůznější formu. Může jít např. o pohybovou aktivitu.<sup>45</sup> Může jít o situační anekdotu, kterou ten, kdo neprožil tutéž situaci, někdy obtížně pochopí. Může jít o kuchařskou dovednost při harmonizaci pokrmu s různým kořením atd. Většina z těchto aktivit jsou tvůrčími počiny, které se bohatě vyskytují, mají svou velkou potenci, a proto jsou často napodobovány jinými, kteří pak mohou originalitu dovést k dokonalosti. Lidské myšlení vždy v těchto případech muselo nějak vybočit z obvyklých průběhů a z nápodoby, aby vytvořilo novost. Nejde tedy jen o osvojení si něčeho, co vytvořilo jiné lidské myšlení, ale co je individuální autentickou tvorbou.

Společensko – historická dimenze tvořivosti však tyto příklady překračuje, neboť se jedná o tvůrčí činnost, která je prototypem nabídnutým k nápodobě a dávajícím možnosti nových konstitucí a objasnění světa. Aby však k takové tvořivosti došlo, musí být postihnuty podstatné souvislosti ve jsoucnech, musí být postihnuty poměry, v nichž mohou být stimuly nově syntetizovány. Lidské myšlení v těchto případech také vybočuje z obvyklých průběhů. Jestliže např. někdo vymyslí nový kuchařský recept, nikterak se to netýká jeho názoru na svět (např. na oteplování zeměkoule), ale jen dílčího problému; avšak když někdo tvrdí, že čas se mění ve vztahu k rychlostem pohybu vzájemně k sobě vztažených těles, proměňuje tím celkový vysvětlující názor na to, co je to čas a jak jsme v čase, proměňuje tím celou objasňující strukturu světa. Proto takováto tvorba je mnohem obsáhlejší a podstatněji proměňující pohyby lidského myšlení.

Společensko – historická dimenze tvorby vyžaduje genialitu, která je měřena nikoli tím, kolik kdo zná, ale tím, kolik nového bylo přidáno k dosavadnímu poznání. Lidské myšlení je zde na dobrodružné výpravě, která nemá předem zajištěný úspěch. To, zda se povede dosáhnout novosti, nelze ničím zaručit. Nelze to vázat ani na množství prostředků vynakládaných na tvorbu, ani množstvím talentů, kteří budou v dané oblasti tvořit, ani ničím dalším. Je to záležitost projekcí lidského myšlení a toho, nakolik se mu podaří postihnout poměry, v nichž jsoucna jestvují a opakovat, případně modifikovat možnosti kombinací při konstitucích. Tento proces patrně spočívá v oproštění se od všeho, co zatěžuje lidské myšlení, tedy toho, co lze charakterizovat jako obvyklostí (zvyky) při projekcích a co lze charakterizovat jako zatíženost danou

<sup>44</sup> Domníváme se, že tvorba je vždy záležitostí individuální mysli. Názor, že mnohdy jde o týmovou práci a že se v týmech (vědeckých) emerguje novost obdobně tomu, co jsme výše popisovali je pro nás obtížně akceptovatelný. Týmová práce může k výskytu novosti významně přispívat, dokonce mnohdy může být i nezbytnou podmínkou, avšak novost se vždy, podle našeho názoru, vyskytne v individuální mysli. Je však obtížné rozhodnutí, nakolik jednotlivá lidské myšlení v týmu přispěla k tomu, že se novost vyjeví.

<sup>45</sup> Příkladem může být např. styl flop ve skoku vysokém, nebo hod koulí s otočkou, v gymnastice jsou dokonce některé cviky pojmenovávány po svých prvocvičiteli, obdobně je tomu u skoků v krasobruslení, atd.

vazbou na biologický substrát – mozek; a na společenské vztahy, z nichž plyne většina zaměření (ambicí) při konstitucích věcí a mentálů a jednostranně upřednostněných pragmatických hodnot. Případně může jít i o vizi sebe sama a o potvrzování této vize.

Jak však takového oproštění dosáhnout, jak prolomit bariéru individuálních ambicí? Žádný jednotný recept zřejmě nelze zformulovat. Stav, kdy se lidské myšlení blíží oproštění, nastávají snad tehdy, kdy je lidské myšlení na něco koncentrováno a to s plností. Především tedy jde o ztlumení projekcí mezi jsoucná (úplná izolace od projekcí není možná) a přenesení pohybu lidského myšlení do oblastí mentálů. Je-li lidské myšlení koncentrováno, pak to znamená, že se pokouší udržet pozornost jen na jediném intencionálním obsahu a nevybočuje z něj. Stále má v ohnisku pozornosti jedno a totéž (např. nějaký intencionální obsah). Ztlumit projekce je ovšem obtížné, protože lidské myšlení je rozprostřeno prostřednictvím receptorů mezi jsoucná a kromě exteroceptorů, které jsou ztlumeny, jsou zde také interoceptory a proprioceptory vytvářející neustále ataky tělesnosti na lidské myšlení a které v podobě stimulů volají po zpracování. Tělo jest<sup>46</sup> a jeho jestvování je velkou zkouškou, kterou musí lidské myšlení zvládnout, chce-li se koncentrovat pouze na jediný intencionální obsah. Koncentrace je velmi obtížná, protože lidské myšlení je pohybem, takže je nelze zastavit. Můžeme jen jeho pohyb omezit a převést z volného putování mezi mentály na neustále kroužení kolem daného intencionálního obsahu, můžeme tento okruh pohybu zužovat, ale zastavit lidské myšlení nelze. Teprve po dosažení tohoto stavu, při němž jsou také plně aktivovány oba proudy lidského myšlení,<sup>47</sup> se může lidské myšlení soustředit na svůj pohyb mezi mentály a na tvorbu možných světů v představách. Zde začíná to, co I. Kant nazval transcendentální obrazotvorností, z níž vyrůstají dva kmeny poznání: smyslovost a myšlení.

To znamená, že při udržení koncentrace a pod uvědomovanou kontrolou popouští lidské myšlení svůj pohyb a vybírá různé intencionální obsahy, které smíchává a kombinuje, při čemž nejenže opakuje na mentální úrovni cesty projekcí mezi jsoucná, ale pokouší se také o nové kombinace stimulů, které vytrhlo ze jsoucen a které vyzvedává z mysli jako vjemy. Zkouší, co by z nich bylo možno konstituovat tak, aby se svět zmnožil a měl větší objasňující potenci. Zkouší, jak by také mohl být. To vše se odehrává pouze v mysli a pouze s mentály. Vznikají tak nové představy.

Protože v těchto chvílích lidské myšlení ví, že jde pouze o tvorbu představ, je také zčásti oproštěno od vazby na jsoucná a obvyklostí svých projekcí mezi ně, popřípadě

<sup>46</sup> Každý může vnímat ataky tělesnosti, když si např. lehne a zavře oči. Po chvíli jej začne něco svědit, uslyší svůj dech, nebo tlukot srdce atd. Všechny tyto stimuly jej budou odvádět od koncentrace na konkrétní intencionální obsah mysli a teprve po ovládnutí těchto afikcí těla je mu umožněno plné soustředění.

<sup>47</sup> Domníváme se, že když se lidské myšlení koncentruje jen na jediný myšlenkový obsah, dochází k tomu, že primární proud má tendenci neustále odbíhat od tohoto konkrétního intencionálního obsahu k jiným, asociativně generovaným obsahům a že každý, kdo prochází tímto stádiem je překvapen obtížností úkolu koncentrace a neodbívání od toho, co je předmětem koncentrace na jedné straně, ale také bohatostí toho, co mysl neustále nabízí na straně druhé. Také se domníváme, že při takovýchto koncentracích si lze plně uvědomit oba proudy lidského myšlení. Jeden, který neustále odbíhá a druhý, který to pozoruje a dává příkaz vrátit se k původně určenému intencionálnímu obsahu.



také o navádění směrů pohybů, který je dán bytím jsoucen. Oproštěnost od vazby na biologický substrát zůstává, avšak inhibice zatížení sociálními vztahy, případně sebepotvrzením se může v těchto momentech proměnit a volnost pohybů lidského myšlení může být značná. Nejrelevantnější pokusy však nastávají tehdy, když se lidskému myšlení podaří v kontemplacích dosáhnout souběhu s kosmickým pohybem myšlení, tzn. tehdy, když opakuje bazické protoperace myšlení a napodobuje to, jak jsou konstituována jsoucná. I když jsou tyto operace prováděny jen s mentály, vznikají tak představy, které se mohou stát předlohou, na jejímž základě bude konstituován svět, nebo konstruovány předměty.

Stavy koncentrace lidského myšlení s následnými meditacemi (případně i kontemplacemi) jsou nutnou, avšak nikoli dostačující podmínkou pro tvorbu novostí. Jako další podmínku vidíme důsledné a dlouhodobé budování endoceptivní struktury s bohatými intencionálními obsahy mysli, jak jsme již na to poukázali.

#### 4.4 Tvorba jako dílo

Kromě novostí ve fluxní formě jsou výsledkem reakcí lidského myšlení na mentály také novosti petrifikované. Máme tím na mysli specifický průběh lidského myšlení, který vyúsťuje a zpředměňuje se v díle.

Dílo je fixací pohybů lidského myšlení, tím že tyto jsou zvnějšněny, že jsou nejen aktuálně přítomny v mysli, ale že se jim dostává také pozicionality. Je tomu tak většinou proto, že jejich obsažnost je značná a obtížně zachytitelná ve fluxní formě. Dílo však tím, že se stává artefaktem, obsahuje v sobě nutně strukturu umožňující lidskému myšlení při styku s ním kopírovat, nebo alespoň napodobovat pohyb, jaký mělo tvůrčovo myšlení. Toto opakování nemusí být přesné, neboť při setkání s dílem jde zároveň o přítomnost endoceptivní struktury vnímajícího a tudíž jde opět o vnímání skrze hodnotovou, intencionální strukturu vnímajícího, o jeho plány, očekávání atd. Při vnímání díla a tudíž i opakování pohybu lidského myšlení tvůrce (i když někdy je částečné) jde o jeho interpretaci.

Geniální dílo překračující historický stav oboru je lhostejné k okolí i ke svému autorovi. Žije svůj život v re-prezentacích a tudíž je stále nějak přítomno v dějinách. Jeho budoucnost spočívá v jeho interpretacích a jejich obměnách. Mnohokrát se stalo, že dílo bylo patřičně oceněno až po smrti autora, nebo po několika staletích, a že se k němu budoucnost vrací jako ke svému základu. Co to však znamená, že dílo předběhne svou dobu? V takovýchto případech jde především o odhalení idejí a jejich fixací v díle, které jsou univerzálně platnými. Jejich univerzalita spočívá v tom, že postihují něco ze jsoucen, tedy to, jak jsou jsoucná určena. Lidské myšlení však ve svém všeobecně ještě nepochytilo tyto poměry jsoucen a obvyklé konstituce a objasnění spočívají na jiných základech. Dílo a v něm obsažené ideje se stávají aktuálními až s ohledem na změnu v podmínkách, v nichž jsou interpretovány, tedy na všeobecném přijetí jiného názoru na svět. Nezávislost života geniálního díla pak spočívá v tom, že univerzalita díla v sobě skrývá iniciaci dalších možných interpretací (nabývá tedy smyslu moci být), z nichž některé mohou být (někdy i proti původnímu záměru autora) zvýrazněny a dílo se pak stává symbolem. Je tedy interpretováno v kontextech (z nichž

některé mohou být dány aktuálními souvislostmi) a jiné myšlenky (které mohly být v ohnisku autorova zájmu) jsou označovány jen jako kolorit doby.

Zdá se nám, že nejvýraznější projevy tvorby projevené v díle jsou z oborů filosofie, náboženství, politiky, umění a vědy. Jde o díla, která se soustředí na dobro (náboženství); spravedlnost (politika); krásu (umění); a pravdu (věda). Tedy na absolutní hodnoty. Nejprve je nutné poznamenat, že každá z těchto forem není nikterak specializovaná vazbou jen na výše přiřazenou hodnotu, ale že v každém z těchto specifických oborů lidské tvorby jsou obsaženy také další z absolutních hodnot i hodnoty pragmatické, avšak tyto jsou přítomny v daném oboru tvorby nikoli jako dominující.<sup>48</sup> Jediná filosofie v tomto ohledu snad pokrývá všechny obory tvorby v ryzosti určení toho, co je to hodnotová orientace a co jsou absolutní (transcendentální) hodnoty (tedy pátrá po tom, co je to být absolutním, být univerzálním, a co je to hodnota) a toto postavení jí v jistém smyslu zaručuje výlučnost oproti jiným oborům. Výlučnost filosofie však spočívá nejen v pátrání po základech absolutních hodnot, ale také toho, jak klademe otázky, které prozkoumávají to, co jest, včetně založení jednotlivých oborů lidské tvorby a tím také samotného lidského myšlení. Tázání se po tom, co jest, je zároveň i vždy tázáním se po smyslu jsoucího. Jde o reflexi, která je obdobná tomu jak se domníváme, že dochází k uvědomění. Také filosofie vždy něco zrcadlí (především bytí jsoucna; způsob, jímž se toto bytí jsoucna odkrývá a vytahuje na světlo uvědomění) a v tomto zrcadlení nachází filosofie sebe samu. Nejde tedy o zaměřenost na nějaký objekt, ale jde o samotnou filosofii, o filosofování, které tím, že se na něco zaměřuje, dává výpověď o sobě samotné. Vše další včetně hodnot (pravdy, dobra, krásy, spravedlnosti) je zde přítomno jako prostředek filosofie samotné, jako něco, na čem se filosofie osvědčuje.

Náboženství je prostředkem jak reflektovat Boha. Není tedy v tomto pojetí samoučelem, není sebereflexí, jak je tomu u filosofie. Jde o reflexi Boha. Právě tak je tomu i u dalších výše uvedených oborů. Věda se snaží podat svědectví o jsoucnech a jejich poměrech. Jsoucno je pro vědu objektem a vše (např. vědecké metody) směřuje k tomu, aby se věda dobrala pravdivého odkrytí jsoucna, případně aby dala návody k tomu, jak by mohly být věci přeuspořádány v předměty plnící předem stanovené účely. Totéž platí o umění, které chce zprostředkovat krásno; nebo pro politiku sjednávající spravedlnost (nebo se o to alespoň pokoušející). Ne vždy se těmto oborům podaří to, co si předsevzaly a jistě i uvnitř nich dochází k posunům v účelech i v tom, co zkoumají, avšak jde jim vždy o něco a teprve zprostředkovaně o sebe sama. V tom spatřujeme rozdíl mezi těmito obory a filosofií.

Filosofie je reakcí lidského myšlení na mentály, je snahou po takovém jejich uspořádání, které by mělo nejrozsáhlejší explanační mohutnost a umožňovalo tak stabilně konstituovat svět. Stabilní konstituce jsou velkou výhodou, a proto lidské

<sup>48</sup> Plně si uvědomujeme, že odborníci v těchto oborech mohou (a jistě i budou) protestovat proti redukci toho, co zkoumají, jen na absolutní hodnoty (i když by je to mělo potěšit) a navíc ještě jen na jednu z nich. Chceme zde jen poznamenat, že nám nejde o detailní postizení vymezených oborů, ale především o tvorbu samotnou, jakožto jednu z reakcí lidského myšlení na konstituované mentály. Chceme tedy ukázat, co z těchto reakcí plyne a jak dále petrifikované výsledky tvorby ovlivňují následné průběhy lidského myšlení.

myšlení bere při konstitutivních operacích v potaz vše, co mysl obsahuje, pokud je to lidskému myšlení přístupno. A i když to není aktuálně dostupné, je zde vždy vědomá snaha po co nejrozsáhlejší obsažnosti a sjednocení intencionálních obsahů mysli v jeden koherentní celek; takže při filosofování jde o neustálé operace lidského myšlení, které se snaží včleňovat stále se vynořující intencionální obsahy mysli a také nově vytrhované stimuly ze jsoucna do jednotného názoru. A filosofie je právě tvorbou takového rozsáhlého a jednotného názoru, je tvorbou vnitřního mentálního světa. Názorem je z hlediska lidského myšlení vnitřní mentální model světa, s nímž se uskutečňují projekce mezi jsoucna a následné konstituce světa a mentálů. Názor se však netýká jen konstituovaného světa a mentálů, ale zahrnuje také jsoucna, jejich uspořádání a jejich vzájemné vztahy a zahrnuje také otázku po bytí jako po smyslu. Názor je tedy tím, co orientuje lidské myšlení, co mu vykazuje a usnadňuje možnosti jeho pohybu. V tomto je názor něčím řídicím vůči pohybům lidského myšlení. Snad proto I. Kant dokonce podrobuje lidské myšlení službě názoru.

*At' již jsou způsoby a prostředky, jimiž se nějaké poznání vztahuje k předmětům, jakékoli, přece jen způsob, jímž se k nim vztahuje bezprostředně a jemuž veškeré myšlení slouží jako prostředek je názor. [14] (s. 53) <sup>49</sup>*

Domníváme se, že názor je dominantním jen vůči zaměřenosti pohybu lidského myšlení, nikoli však tak, že by určoval samotný pohyb, tzn. operace, které lidské myšlení používá při projekcích a konstitucích. Názor si jen vyžaduje stabilní konstituce světa, které jemu samotnému odpovídají. Budování tohoto názoru je tedy tím, co je dominující součástí filosofie. Názor je tušením bytí jsoucna a jejich uspořádání ztvárňujícím to, co je svět. Názor je tedy místem, kde se jsoucna stýkají s myslí, a tudíž je vždy východiskem vztahů ke jsoucňům. Vztahy pak jsou tím, co dále diferencuje rozlišovací potenci intelektu, neboť až tehdy, když je mysl ve vztahu ke jsoucnu může spatřit to, co by se mohlo stát cílem konstitucí světa, tedy toho, jak by měl svět vypadat.

Zároveň však jde vždy i o zhodnocování všeho s čím se lidské myšlení setkává a tak se ve filosofii objevuje snaha po sjednocení všech absolutních hodnot, po jejich vzájemném přiblížení a tím zároveň i o pochopení smyslu veškerenstva (celku). Tento smysl není účelovým dobíráním se, ale jde o samočinnost, při níž se prokazuje modus ponechavosti. Není zde zmocňující se úmysl, ale snaha k dopracování se, snaha po dosahování smyslu, která se tak stává orientační oporou. Smysl pramení z fundamentální filosofické otázky: ***Jak to, že je něco spíše nežli nic?*** Tato otázka je otázkou po bytí jsoucna a je to ona, která vyústí v údiv jakožto výživný pramen

<sup>49</sup> Poznamenáváme, že Kant však pod názorem rozumí (něco jako) souhrné smyslové vjemy, tedy to, co H. Bergson označuje jako obrazy. Srovnej: [4]. V textu uvedeném Kantovým citátem chceme jen poukázat na to, že Kant takto hovoří o dvou kmenech poznání (smyslovosti a rozvažování), které pramení z jednoho kořenu, kterým je transcendentální obrazotvornost jako schopnost syntézy. Tak je tomu v prvním vydání. V druhém se však mnohem více soustřeďuje na intelekt, který mu představuje pravidla; a také na schopnost dedukce. Pro nás je názor endoceptivní model světa, s nímž se myšlení projekuje mezi jsoucna. Není tedy tak úzce svázán jen se smyslovostí a v tomto pojetí také píšeme o tvorbě názoru jako o filosofii.

filosofického tázání. Jakýkoli pokus odpovědět na ni, vždy vyústí v tázání se po smyslu bytí.

Jak se však lze dobrat (či dobírat se) smyslu? A to smyslu bytí jsoucna a tím i sebe sama a svého postavení jako reflektujícího jsoucna mezi jsoucnými? K této otázce jsme se již z části vyjádřili, když jsme charakterizovali smysl jako realizaci modu **moci být**. Ale dobírání se k této charakteristice je již jinou otázkou. Je otázkou možností mysli a pohybů lidského myšlení, je to otázka týkající se překonávání vize setrvalého a algoritmicky se opakujícího stavu k dosahování dynamiky, v níž kritičnost na jedné straně a imaginace s intelektem na straně druhé, hrají dominující role.

Lidské myšlení většinou na bytí jako otázku zapomíná a uchyluje se k blízkým samozřejmostem jestvování světa. Lidské myšlení má velkou kapacitu při konstitucích věcí, ale bez odstupů od svých biologických a sociálních vazeb a bez prvotního názoru (tedy bez filosofie) nezahledá bytí a smysl. Teprve až vykoná specifickou (filosofickou) reflexi a reaguje na celé své počínání, je schopno **stát** před otázkou po smyslu. Také ostatní vyjmenované obory reakcí lidského myšlení na mentály (tedy, věda, umění, náboženství a politika) se potýkají s otázkami, které jsou pro ně dominující. Nemohou se však dopátrat svých základů, neboť ty jsou pro ně vždy již nějakým před-podkladem.

Tím, že jsme takto vykázali specifické místo filosofie oproti dalším oblastem tvorby (tzn. oproti vědě, politice, umění a náboženství), která vyúsťuje v petrifikovaných dílech, jsme nechtěli ani hanět, ani cokoli vynášet do výsostného postavení. Chtěli jsme jen prokázat, jak se to má podle našeho názoru s tvorbou jako reakcí pohybu lidského myšlení na mentály. Ohnisko našich poukazů pak bylo snad poněkud jednostranně vymezeno tím, co považujeme za absolutní hodnoty; a dále tím, jak jsou vyjasňovány základy tvorby jako jednoho z (rozhodujících) pohybů lidského myšlení, za což se omlouváme.

## 5 Závěr

Závěrečná poznámka je poděkováním editorům této práce. Ceníme si jejich iniciativního počínu a přejeme jejich projektu budoucí proslulost. Uvědomujeme si, kolik práce za tímto počínem stojí a zároveň i kolik rizika na sebe berou.

Děkujeme.

## Literatura

- [1] Armstrong, D.M.: *The Mind – Body Problem*. Colorado, Perseus Books Group 1999.
- [2] Banks, E. C.: *Ernst Mach's World Elements*. Dordrecht, Kluwer A.P. 2003, p. 123.
- [3] Barrow, J. D., and Webb, J. K.: Inconstant Constants. *Scientific American* 6 (2005) 33–39.
- [4] Bergson, H.: *Hmota a paměť*. Praha, Oikúmené, 2003.

- 
- [5] Calasso, R.: *Svatba Kadma s Harmonií*. Praha, Mladá fronta 2000, s. 235.
- [6] Damasio, A.: *Descartův omyl*. Praha, Mladá Fronta, 2000.
- [7] Derrida, J.: *Tradice vědy a skrývání smyslu*. Praha, Oikúmené 2003.
- [8] Edelman, G. M.: *Wider than the Sky*. New Haven and London, 2004.
- [9] Hartmann, N.: *Zur Grundlegung der Ontologie*. Berlin, Walter de Gruyter and co. 1965.
- [10] Havel, I. M.: *Artificial Thought and Emergent Mind*. Centrum pro teoretická studia. UK Praha 1993 - 11 The Emergence of Emergence
- [11] Havel, I. M.: *Artificial Thought and Emergent Mind*. Centrum pro teoretická studia. UK Praha 1993 - 12. Collective Phenomena
- [12] Heidegger, M.: *Věda, technika a zamyšlení*. Praha, Oikúmené 2004.
- [13] Husserl, E.: *Ideje k čisté fenomenologii a fenomenologické filosofii I*. Praha, Oikúmené 2004.
- [14] Kant, I.: *Kritika čistého rozumu*. Praha 2001.
- [15] Kováč, L.: Úvod do kognitivnej biológie. *Biologické listy* **51** (3) (1986) 172-190.
- [16] Kuhn, T. S.: *Struktura vědeckých revolucí*. Praha, Oikúmené 1997, s. 47.
- [17] Lange, O.: *Celek a vývoj ve světle kybernetiky*. Praha, Svoboda 1966, s. 39 – 40.
- [18] Merleau-Ponty, M.: *Viditelné a neviditelné*. Praha, Oikúmené 2004.
- [19] Nagel, T.: *Čo to všetko znamená*. Bratislava, Bradlo 1991.
- [20] Nagel, T.: Aké je byť netopierom? In: *Mysel' / telo / stroj*. Bratislava, Bradlo 1992 (editoři: Gál, E., Kelemen, J.) s. 12 a 37-50.
- [21] Nosek, J.: *Mysl a tělo v analytické filosofii*. Praha, Filosofia 1997.
- [22] *Pojednání o lidském myšlení*. Praha, Ekopress 2005. *Pojednání o lidském myšlení II*. Praha, Ekopress 2007.
- [23] Popper, K. R., Eccles, J. C.: *The Self and its Brain*. London, Routledge and Kegan Paul, 1986, chapt. 2.
- [24] Popper, K. R.: *Život je řešením problémů*. Praha, Mladá fronta 1998.
- [25] Reiser, M. F.: *Memory in Mind and Brain*. Yale University Press 1994.



# Konstruktivistický prístup k štúdiu kognície

Martin TAKÁČ<sup>1</sup>

**Abstrakt.** Schopnosť rozumieť – pripisovať význam – je kľúčovým aspektom kognície. Pre štúdium rôznych foriem kognície – živých aj umelých – je užitočné disponovať takým pojmovým a teoretickým aparátom, ktorý nie je založený na výlučnosti ľudského rozumenia. V tejto kapitole sa pokúsime takýto pojmový aparát vybudovať a budeme ho aplikovať na analýzu širokej škály kognitívnych systémov od jednoduchých predverbálnych organizmov až po ľudí a softvérové agenty, či roboty. Ukážeme, že nutnou podmienkou rozumenia je situovanosť v prostredí a schopnosť učiť sa. Vysvetlíme mechanizmy takéhoto učenia a uvedieme ich konkrétne výpočtové modely.

## 1 Úvod

Kľúčovou vlastnosťou kognitívnych, resp. inteligentných systémov, je schopnosť rozumieť, pripisovať význam [79]. Preto bude táto kapitola najmä o významoch. Štúdiom významov v rôznych typoch kognitívnych systémov – v živočíšnej ríši, u ľudí aj u umelých systémov sa môžeme naučiť veľa o kognícii ako takej. Aj preto, že významy sa často spájajú s komunikáciou a ľudský jazyk je najdokonalejší komunikačný systém, ktorého ovládanie je späté s mnohými kognitívnymi funkciami. Spôsob, akým budeme kogníciu a významy študovať, je konstruktivistický – aby sme pochopili, ako niektoré kognitívne mechanizmy fungujú, pokúsime sa ich (alebo aspoň ich zjednodušenú podobu) replikovať v počítačových simuláciách. Pôjde teda o „chápanie tvorením“ [60].

Slovo konstruktivistický v názve tejto kapitoly má však aj ďalší dôvod: ako ukážeme, schopnosť učiť sa a samostatne konštruovať významy na základe skúseností je pre kogníciu kľúčovou – významy, ktoré sú predprogramované, resp. dané zvonka, nikdy nebudú kognitívnemu systému vlastné a teda ťažko hovoriť o nejakom rozumení.

Rôzne typy agentov – baktérie, zvieratá, ľudia, niektoré počítačové programy a roboty – majú niečo spoločné: dosahujú ciele pomocou vnímania a konania vo svojom (reálnom či virtuálnom) prostredí [42]. Niektoré z nich dokážu komunikovať medzi sebou či dokonca s ľuďmi. Do akej miery rozumejú tomu, čo robia? Pokiaľ pripisujú

---

<sup>1</sup> Katedra aplikovanej informatiky FMFI UK, Mlynská dolina, 848 48 Bratislava, E-mail: takac@ii.fmph.uniba.sk

nejaký význam situáciám a udalostiam v prostredí, aká je povaha tohto významu? Používajú rovnaké významy aj keď komunikujú? Odkiaľ sa berú významy? Sú vrodené (predprogramované), alebo naučené? Takéto otázky nás budú zaujímať.

Niektorí autori odmietajú používať pojmy ako „rozumenie“ či „význam“ v súvislosti s inými ako ľudskými agentami. Často považujú za predpoklad rozumenia intencionalitu alebo vedomie [67]. Takýto prístup z definície vylučuje niektoré typy agentov, alebo aspoň celú vec ešte viac zahmlieva redukciou problému významu na ešte ťažší (keďže detekcia alebo dôkaz vedomia či intencionality u nie ľudských agentov sú problematické). My sa preto pokúsime o odlišný prístup. Budeme hľadať čo najneutrálnejšiu a najmenej antropocentrickú charakterizáciu rozumenia, aby bola použiteľná pre štúdium významov aj u predverbálnych živých organizmov a umelých systémov. Tento prístup je v súlade so snahami definovať život a vedomie všeobecnejším spôsobom tak, že ľudský život a ľudské vedomie sú len jednou z jeho možných inštancií [3, 38, 51].

Po krátkom úvode do formálnych teórií významu prejdeme lekciami z predverbálnych štádií fylogénzy a ontogénzy k štúdiu senzomotorickej inteligencie zvierat a malých detí. Počiatky významu totiž treba hľadať práve tu, omnoho skôr ako sa na evolučnej scéne objavil jazyk. Potom opíšeme problémy s významom u umelých systémov a budeme analyzovať niektoré príklady. Hlavne nás však bude zaujímať otázka pôvodu významov, preto preskúmame možnosti a limity konštruktivistického prístupu metodológiou výpočtového modelovania.

Prínos takéhoto prístupu je mnohoraký. Po prvé, žijeme v časoch, keď komunikácia a interakcia medzi človekom a strojom, či medzi strojmi navzájom nie je sci-fí, ale praktický inžiniersky problém. Potrebujeme navrhovať také reprezentačné formalizmy, ktoré nám umožnia vybaviť stroje/počítače ontológiami nutnými pre úspešné riešenie daných úloh a pre vzájomnú koordináciu. Reprezentácia musí byť dostatočne komplexná, aby postihla špecifiká materiálnych a sociálnych prostredí, vrátane ich dynamického charakteru. V otvorených prostrediach je schopnosť učiť sa autonómne konštruovať reprezentáciu relevantných znalostí kľúčovou. Po druhé, operacionalizácia sémantických teórií a budovanie relevantných výpočtových modelov pomáha vyjasniť pojem „rozumenia“, ktorý je zdrojom kontroverzií v umelej inteligencii od jej počiatku, a poskytuje mechanizmy ukotvenia symbolov. A nakoniec, konštruovanie výpočtových modelov nám môže pomôcť lepšie porozumieť sebe samým. Úspešné modely môžu mať spätný vplyv na teórie učenia a jazykového vývinu aj na kognitívnu vedu ako takú.

## 2 Teórie významu

Filozofi a lingvisti sa otázkou „čo to znamená *znamenat'* niečo“ zaoberajú už mnoho storočí. Dnes sa štúdiu významu venuje najmä sémantika a semiotika. V tejto časti predstavíme niekoľko vplyvných teórií, ktoré dávajú na otázku povahy významov a ich vzťahu k jazyku rôzne odpovede.



## 2.1 Funkcionalistická sémantika

Základnou otázkou je, či existujú nejaké objekty – materiálne, mentálne, ideálne, atď., ktoré konštituujú význam jazykových výrazov. Sémantické teórie, ktoré na túto otázku odpovedajú kladne, sú *denotačné*; tie, čo odpovedajú záporne, nazveme *nedenotačné*. Príkladom nedenotačného prístupu je funkcionalistická sémantika, ktorej základy položil Wittgenstein. Podľa (neskorého) Wittgensteina, významom slov nie je referencia na objekty alebo veci označené vo vonkajšom svete, ani žiadne idey či mentálne reprezentácie s nimi asociované, ale spôsob ich používania v komunikácii [101]. Jazyková výpoveď je v prvom rade aktom v reálnom svete, význam je preto viac vecou pragmatiky ako sémantiky. Wittgenstein prirovnáva použitie jazykového výrazu k ťahu v hre podľa nejakých pravidiel.

## 2.2 Realistická sémantika

V denotačných sémantikách sú jazykovými významami nejaké objekty. Podľa názoru na povahu týchto objektov môžeme rozlíšiť *realistické* a *kognitivistické* (alebo *konceptualistické*) prístupy. Podľa realistického prístupu sú významami entity „vo vonkajšom svete“ (porovnaj obr. 2a). Podľa kognitivistického prístupu sú významy mentálne entity „v hlave“ (obr. 2b).

V *extenzionálnych* realistických sémantikách sa význam definuje vzťahom k objektom vo svete, alebo formálnejšie k modelovej štruktúre  $M$ . Mená sa zobrazia na konkrétne objekty – prvky  $M$ , predikáty sa zobrazia na množiny objektov resp. relácie v  $M$ . Pomocou kompozície sa vety zobrazia na pravdivostné hodnoty. Dôsledkom tohto prístupu je, že významy sú objektívne, spoločné a nezávislé od pochopenia jednotlivých používateľov. Základy extenzionálnej sémantiky položil Tarski [87].

Čoskoro však boli objavené medzery množinovo-teoretického prístupu, pretože niektoré javy (kompozície predikátov, čas, modality) sa nedali napasovať na extenzionálnu definíciu významu. Pokusom o nápravu problémov extenzionálnej sémantiky bol návrh tzv. *intenzionálnej sémantiky* [15, 46, 55], v ktorej sa prvky jazyka zobrazujú na množiny možných svetov namiesto jediného. Tvrdenie – propozícia sa definuje zobrazením z možných svetov na pravdivostné hodnoty, určujúcim množinu svetov, v ktorých je dané tvrdenie pravdivé.

## 2.3 Kognitívna sémantika

Gärdenfors charakterizuje kognitívnu sémantiku nasledovne [28]:

1. Význam je konceptuálna štruktúra kognitívneho systému (nie podmienky pravdivosti v možných svetoch).
2. Konceptuálne štruktúry sú stelesnené (význam nie je nezávislý od percepcie a telesnej skúsenosti).
3. Sémantické elementy sa konštruujú z geometrických alebo topologických štruktúr (nie zo symbolov skladaných na základe nejakého systému pravidiel).

4. Kognitívne modely sú predovšetkým obrazovo-schematické (nie propozičné). Obrazové schémy sú transformované metaforickými a metonymickými operáciami [48].
5. Sémantika predchádza syntax a čiastočne ju determinuje (syntax nemožno opísať úplne nezávisle od sémantiky).
6. V kontraste s Aristotelovským poňatím kategórií založenom na nutných a postačujúcich podmienkach, koncepty vykazujú prototypové efekty [65].

Prvé dva body vymedzujú, že porozumenie jazyka je integrálnou súčasťou konceptuálneho systému, ktorý slúži aj na usudzovanie, orientáciu a konanie vo svete [47], teda nemôže byť zabezpečované nejakým izolovaným jazykovým modulom (v zmysle Fodora [25]). Na neurálnej úrovni koreluje s aktiváciou štruktúr zúčastňujúcich sa na vnímaní, konaní, predstavách, či rozprávaní o reprezentovanom obsahu [63, 64].

Geometrické štruktúry reprezentujúce sémantické elementy (bod 3) Gärdenfors umiestňuje do tzv. *konceptuálneho priestoru*. Konceptuálny priestor pozostáva z množstva dimenzií ako farba, výška tónu, teplota, váha, tri priestorové dimenzie, ktoré zodpovedajú vlastnostiam reprezentovaných objektov. Dimenzie dôležité pre naše prežitie sú pravdepodobne vrodene (vyvinuté v priebehu evolúcie) a vbudované do nášho nervového systému. Iné dimenzie sú naučené a niektoré z nich môžu byť kultúrne podmienené. Dimenzie sú organizované do domén (pre reprezentované objekty môžu byť relevantné dimenzie iba z niektorých domén). Základnou vlastnosťou konceptuálnych priestorov je to, že podobné objekty sú umiestnené blízko pri sebe. Konkrétny objekt je reprezentovaný ako bod v podpriestore zloženom z jednej alebo viacerých domén. Vektor súradníc bodu určuje hodnoty vlastností na jednotlivých dimenziách. Podobnosť dvoch objektov je nepriamo úmerná vzdialenosti ich bodových reprezentácií v konceptuálnom priestore. (Prirodzené) kategórie sú reprezentované (konvexnými) oblasťami v priestore. Geometrické centrá oblastí zodpovedajú najlepším, resp. typickým reprezentantom kategórií – prototypom.

Gärdenfors vybuďoval pomocou konceptuálnych priestorov vlastnú teóriu kognitívnej sémantiky [28]. Medzi iné vplyvné teórie patrí silová dynamika [86], rámcová sémantika [23], mentálne priestory [21] a kognitívna gramatika [49, 50].

Teraz sa bližšie pozrieme na Šefránkovu biologicky motivovanú kognitívnu teóriu reprezentácie [79]. Deklarovaným cieľom tejto teórie je vybudovať netriviálnu falzifikovateľnú úroveň analýzy kognície a rozumenia abstrahujúcu od mozgu a neurálnych procesov. Teória namiesto neurálnych procesov pracuje s obsahmi kognície – význammi. Jej kľúčovým predpokladom je, že významy sú oddeliteľné od jazyka, teda existujú aj u zvierat a detí v predverbálnom vývinovom štádiu. Všeobecne povedané, skúma významy u organizmov/agentov situovaných v nejakom prostredí. Organizmy majú potreby a ciele, ktoré naplňajú svojím správaním (vykonávaním akcií). Organizmy disponujú reprezentáciami zloženými z významov – *rozlišovacích kritérií*. Rozlišovacie kritériá sú abstrakciou schopnosti organizmu rozlišovať (rozpoznať, identifikovať) isté aspekty svojho (vnútorného či vonkajšieho) prostredia. Elementárne kritériá rozlišujú objekty (individuá), prirodzené triedy a prirodzené vlastnosti objektov, a prirodzené vzťahy medzi objektami. Kompozíciou a transformáciami sa dajú z elementárnych kritérií konštruovať komplexnejšie kritériá

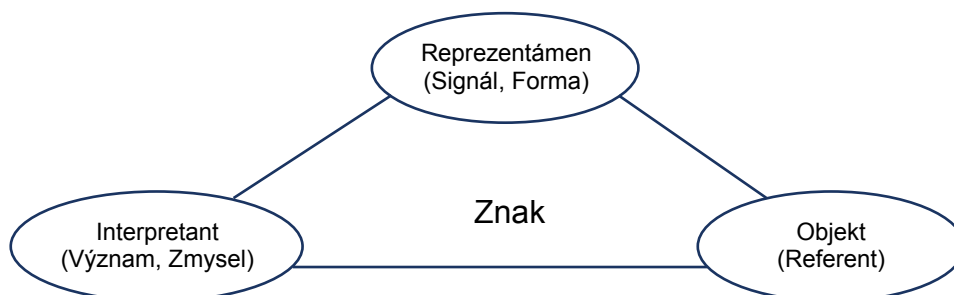
situácií, pravidiel (typov situácií), cieľov (želaných situácií), zmien v prostredí, plánov (projektovaných zmien), metód (osvedčených plánov), udalostí a typov udalostí.

Šefránek ďalej načrtáva cestu ako sa dostať od protosémantiky, protoinferencie a protokomunikácie jednoduchých organizmov cez dvojslovný protojazyk až k plnému ľudskému jazyku s propozičnou reprezentáciou a rekurzívnou syntaxou. Schopnosť rozumieť komplexnému jazyku je úzko prepojená s usudzovaním, presnejšie s hypotetickým (nemonotónnym) usudzovaním [34].

## 2.4 Semiotika

Podľa semiotických teórií vytvárame význam tvorbou a interpretáciou *znakov*. Znak môže mať podobu slov, obrázkov, zvukov, aktov, či objektov, ale stávajú sa znakmi len ak im pripíšeme nejaký význam. „Nič nie je znakom, pokiaľ to nie je interpretované ako znak“ [59]. A tiež, čokoľvek môže byť znakom, ak to niekto interpretuje ako *označujúce* niečo, teda odkazujúce na niečo alebo zastupujúce niečo iné ako samo seba [16]. Peirce definuje znak ako triádu – tri časti poprepájané reláciami:

- *reprezentámen* je forma, ktorú znak nadobúda,
- *interpretant* je zmysel, ktorý má znak pre nejakého interpretátora,
- *objekt* je to, na čo znak odkazuje.

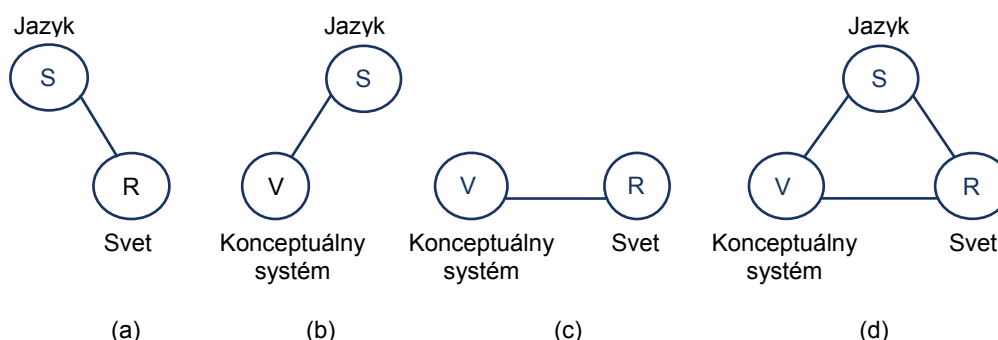


**Obr. 1.** Semiotický trojuholník

Triadické poňatie znaku sa niekedy označuje ako *semiotický trojuholník* (obr. 1). (Odteraz budeme namiesto pôvodnej terminológie *reprezentámen*, *interpretant*, *objekt* používať zaužívanéjšie označenia *signál*, *význam*, *referent*.)

Je dôležité zdôrazniť, že znak v Peirceovom poňatí nie je absolútna alebo ontologická vlastnosť veci, ale je to relačná situovaná interpretatívna rola, ktorú vec nadobúda v konkrétnom kontexte. Čo je pre jedného pozorovateľa znakom, môže byť pre iného len zbytočným alebo nevnímateľným šumom, v závislosti od pozorovateľovho stelesnenia (embodiment), spoločnosti a interakčnej histórie. Takýto pohľad má blízko k *enaktívnemu prístupu* [53]. Enaktívny prístup je asi najsilnejšie vyjadrený v dynamickej kognitívnej paradigme [89], ktorá nahliada na kogníciu ako na komplexný dynamický systém (resp. viacero spriahnutých dynamických systémov)

a „zdôrazňuje časový rozmer kognície a emergenciu samoorganizovaných vzorcov správania jedinca ako stabilných stavov v interakciách mozgu, tela a prostredia” [31]. Takáto perspektíva ide naprieč hranicami medzi mozgom, telom a svetom. Niektoré dynamické modely celkom odmietajú reprezentácie, iné ich považujú za dynamické entity (napr. momentálne stavy systému, trajektórie či atraktory a ich spádové oblasti). „Reprezentácie sú dočasné, kontextovo závislé stability uprostred zmeny a nie statické, nemenné a na kontexte nezávislé jednotky” [30].



**Obr. 2.** Interpretácia neúplných verzií semiotického trojuholníka „význam (V) – signál (S) – referent (R)”. a) Realistická sémantika umiestňuje významy do sveta. b) V „čisto” kognitívnej sémantike sú lexikálnymi významami interné reprezentácie. Bez vzťahu k svetu by však takáto sémantika trpela *problémom ukotvenia symbolov* [37]. c) Predverbálne konceptualizácie sveta môžu predchádzať jazyk. d) V ukotvenej kognitívnej sémantike sú vnútorné významy formované v interakciách so svetom.

### 3 Od predverbálnych významov k jazyku: evolučný pohľad

V rámci kognitívnej sémantiky sa významy považujú za súčasť konceptuálneho systému, ktorý bol tvarovaný skúsenosťou s okolitým svetom. Z toho vyplýva, že rozumenie nezačína až na úrovni jazyka: môžeme hovoriť o rozumení a významoch aj na predverbálnej úrovni (obr. 2c).

Pozrime sa teraz z pohľadu evolúcie na vznik rozumenia v živých organizmoch. Niektorí vedci postulujú elementárne formy kognície na veľmi hlbokých úrovniach fylogenetického stromu, na bakteriálnej, bunečnej, či dokonca molekulárnej úrovni [35, 44]. Z organizmov, ktoré sa objavili v priebehu evolúcie, pretrvali tie, ktorých štruktúra zodpovedala relevantným vlastnostiam ich prostredia. Biologická evolúcia spočíva v generovaní hypotéz o povahe prostredia a vo falzifikovaní týchto hypotéz [44]. V tomto zmysle organizmy, ktoré pretrvali, stelesňujú hypotézy, ktoré neboli zatiaľ falzifikované. Na evolučnú adaptáciu organizmov sa teda dá pozerat' ako na formu fylogenetického učenia, pričom znalosť sa uchováva v štruktúre (stelesnení) organizmov. Šance pretrvať sú vyššie u tých organizmov, ktoré aktívne exploraujú svoje prostredie a prispôbujú sa podmienkam, alebo ich dokážu aktívne pretvárať na

vhodnejšie svojim správaním. Najjednoduchšie kognitívne systémy pozostávajú z mechanizmov vnímania (senzie) a konania (akcie). U nich môžeme za najelementárnejší spôsob „pripisovania významu“ považovať to, že rozpoznávajú v prostredí pomocou svojich senzorov a aktuátorov informáciu relevantnú svojim cieľom resp. účelu [56]. Zložitejšie kognitívne mechanizmy – percepcia, afekcia a kognícia – sú postupne vkladané medzi senziu a akciu v procese interkalárnej evolúcie [45]. Myslenie sa objavuje na najvyšších stupňoch evolúcie ako abstraktná akcia – testovanie rôznych motorických akcií bez skutočnej realizácie svalmi. Deliberácia (zámer), krátkodobá anticipácia, proto-plánovanie, a konečne aj “čo-keby” myslenie (schopnosť mentálne simulovať rôzne scenáre a vyhodnotiť ich dôsledky bez nutnosti ich fyzickej realizácie) zvýšili šance organizmov na prežitie a poskytli im významnú evolučnú výhodu.

Vhodným teoretickým konštruktom na analýzu adaptívneho správania je pojem *reprezentácie*. Gärdenfors rozlišuje dva typy reprezentácie nutné pre evolúciu jazyka: *vyvolanú* (cued) a *oddelenú* (detached) [26]. Vyvolaná reprezentácia je vždy spustená nejakým externým stimulom prítomným v aktuálnej situácii organizmu. Organizmus, ktorý diferencovane reaguje na rôzne stavy svojho prostredia (napr. konzumuje objekty rozpoznané ako potravu a vyhýba sa objektom rozpoznávaným ako predátori) vykonáva (príjajmenšom implicitnú) kategorizáciu – hovoríme, že disponuje vyvolanými reprezentáciami príslušných kategórií. Vyvolanosť reprezentácií spočíva v tom, že sú vždy aktivované len v prítomnosti svojich referentov. Vyvolané reprezentácie sú deskriptívno-vysvetľujúcim konštruktom pre vrodené vôľou neovplyvniteľné diferencované behaviorálne reakcie, ktoré majú fylogenetický pôvod.

Dôležitým mechanizmom, ktorý rozširuje limitované pamäťové možnosti organizmov, je ukladanie externalizovaných pamäťových značiek do prostredia. Príkladom sú pachové stopy, ktoré pomáhajú zvieratám v orientácii, či u ľudí notoricky známy uzlík na vreckovke (a napokon vynález kníh a iných záznamových médií, ktoré výrazne akcelerovali ľudskú kultúru). Externalizovaná značka uložená do prostredia neskôr aktivuje príslušnú vyvolanú reprezentáciu.

*Oddelená* reprezentácia môže zastupovať objekty či udalosti, ktoré nie sú aktuálne prítomné ani nesúvisia so súčasťou situáciou v prostredí. Napríklad šimpanz, ktorý chce dočiahnuť banán a odíde preto hľadať dlhý konár, musí disponovať oddelenou reprezentáciou konára aj možnosti jeho použitia. Predpokladá sa, že oddelené reprezentácie sa vo fylogénéze objavujú približne s vývojom neokortexu [27]; v ontogenéze korešpondujú s vytvorením *stálosti predmetu* [62].

Disponovanie oddelenými reprezentáciami je nutnou podmienkou pre vyššie kognitívne funkcie ako plánovanie, klamanie, sebauvedomenie a jazyková komunikácia [26]. Plánovanie predpokladá schopnosť organizmu „mentálne“ (t.j. pomocou oddelených reprezentácií) vyhodnotiť dôsledky rôznych behaviorálnych scenárov a zvoliť postupnosť akcií, ktorá najviac vyhovuje aktuálnym cieľom. Dobré plánovanie by tiež malo zvažovať dôsledky akcií iných agentov. Predstieranie, resp. klamanie predpokladá schopnosť reprezentovať iné agenty aj s ich vlastnými reprezentáciami, zámermi, plánmi atď., teda akúsi „teóriu mysle“. Klamár musí taktiež disponovať reprezentáciou toho, ako ho budú vnímať klamané agenty. Týmto

vnoreným spôsobom vlastne disponuje (meta)reprezentáciou samého seba, čo je nutným prekursorom sebauvedomenia [26].

Za jednu z najvyšších kognitívnych schopností sa považuje jazyková kompetencia. Jazyk je symbolový znakový systém, ktorý umožňuje externalizáciu a komunikáciu oddelených reprezentácií. Práve oddelenosť reprezentácií umožňuje vyjadriť významy prekračujúce aktuálne „tu a teraz“, dokonca aj významy, ktorých referenty v reálnom svete neexistujú (napr. ježibaba, či trojuholník s tromi pravými uhlami). Podľa Gärdenforsa bola najsilnejším evolučným impulzom pre vznik jazyka potreba spoločenstva koordinovať sa pri spoločnom dosahovaní budúcich cieľov [29].

Koncepty, v ktorých je ukotvená sémantika jazyka, môžu mať rôzny pôvod. Tie, ktoré boli v priebehu evolúcie podstatné pre prežitie, môžu byť vrodené [24]. Iné sú konštruované v priebehu ontogenézy senzomotorickou interakciou s prostredím a pozorovaním dôsledkov vlastných akcií [2, 62]. Ešte iné koncepty sú prenášané kultúrne a ich osvojenie je stimulované samotným jazykom [99]. Vplyv jazyka na proces formovania významov v rámci detského lexikálneho vývinu má aj empirickú podporu [97].

## 4 Reprezentácia významov v umelých systémoch

### 4.1 Základné problémy

Vytvorenie prvých umelointeligentných systémov, ktoré do istej miery používali prirodzený jazyk (napr. ELIZA, SHRDLU) prinieslo so sebou otázky o povahe rozumenia u týchto systémov. Môžu umelo naprogramované systémy naozaj porozumieť jazyku, alebo iba manipulujú s (pre ne) nezmyselnými symbolmi? Tieto otázky vyvolali mnoho kontroverzií a emočne nabitých debát. Príkladmi pokusov o zodpovedanie, alebo aspoň reformuláciu, týchto otázok sú *Metafora čínskej izby* [67] a *Problém ukotvenia symbolov* [37].

John Searle tvrdí, že čisto na základe pozorovania správania (v tomto prípade komunikácie) nie je možné posúdiť, či systém naozaj rozumie tomu, o čom komunikuje [67]. Vymyslel myšlienkový experiment známy ako Čínska izba: predstavme si, že v izbe je zavretý človek, ktorý nevie po čínsky, a komunikuje s okolitým svetom pomocou lístkov. Ľudia zvonku mu posielajú otázky v čínštine. V izbe je škatuľa s čínskymi znakmi a kniha s pravidlami ako kombinovať znaky, ktorá umožňuje formulovať v čínštine odpovede na otázky. Človek v miestnosti skladá odpovede z čínskych znakov iba na základe formálnych pravidiel a porovnávaním tvaru znakov v knižke, na lístkoch a v škatuli. Predpokladajme, že sa v manipulácii so znakmi tak zdokonalí, že dokáže dávať na otázky korektné odpovede. Nikto mimo miestnosti sa nedozvie, že človek vo vnútri nerozumie ani slovo po čínsky.

Steven Harnad sformuloval svoju vlastnú verziu tohto problému, známu ako *problém ukotvenia symbolov*: „Ako možno urobiť sémantickú interpretáciu formálneho symbolového systému tak, aby bola systému vlastná, a nie len parazitovala na významoch v našich hlavách? Ako môžu byť významy nezmyselných symbolových

jednotiek, s ktorými sa manipuluje iba na základe ich (ľubovoľných) tvarov, ukotvené v niečom inom, ako len v ďalších nezmyselných symboloch?” [37] (pozri obr. 2b).

Harnad navrhol aj vlastné riešenie tohto problému – hybridný subsymbolovo-symbolový model mysle, v ktorom symbolická funkcia emerguje zdola nahor ako dôsledok ukotvenia názvov kategórií v ich senzoričných (subsymbolových) reprezentáciách dvoch typov:

1. *ikonické* reprezentácie, ktoré sú analogické lokálnym priemetom (projekciám) vzdialených objektov a udalostí,
2. *kategoriálne* reprezentácie – naučené alebo vrodene detektory vlastností, ktoré zo senzoričných projekcií vyberajú invariantné črty kategórií objektov a udalostí.

Vnútorne reprezentácie nemajú žiaden vlastný význam samy o sebe, ale získavajú ho štruktúrnym spriahnutím s prostredím [53]. Toto spriahnutie má dve zložky: individuálnu a sociálnu. Individuálna, ktorá sa nazýva aj *fyzickým ukotvením symbolov* [92], označuje schopnosť každého jedinca vytvoriť si vlastné prepojenie medzi entitami sveta a vnútornými reprezentáciami. Sociálna, nazývaná aj *sociálne* (alebo *externé*) *ukotvenie symbolov*, označuje kolektívne zjednávanie výberu zdieľaných symbolov a ich významov [12].

Významy sú teda individuálne vytvorené subjektívne konštrukty, ktoré sú kolektívne navzájom zladené. Fyzické ukotvenie symbolov po filozofickej stránke korešponduje s kognitívnym konštruktivizmom založeným na prácach Jeana Piageta [61] tým, že jedinca aktívne konštruujú svoje vlastné významy pomocou kognitívnych procesov na základe predchádzajúcej skúsenosti a interakcií s okolím. Sociálne ukotvenie symbolov je blízke sociálnemu konštruktivizmu [96] tým, že individuálne vytvorené významy sú motivované a formované sociálnym kontextom.

Vo zvyšku tejto časti urobíme stručný prehľad reprezentácie významov v niekoľkých umelých systémoch a výpočtových modeloch. Budeme analyzovať ich vyjadrovaciu silu a schopnosť vysporiadať sa s problémom ukotvenia symbolov.

## 4.2 Procedúry, pravidlá a predikátová logika

Začneme s niekoľkými systémami, ktoré sú považované za míľniky v ranej histórii umelej inteligencie. U prvého presláveného konverzačného programu ELIZA [98], ktorý simuloval nedirektívneho psychoterapeuta, je ťažko vôbec hovoriť o nejakých významoch. ELIZA nemala žiaden model vonkajšieho sveta, ani žiadnu sémantickú reprezentáciu prebiehajúcej konverzácie. Fungovala čisto reaktívnym spôsobom podľa niekoľkých šikovne naprogramovaných transformačných a dekompozičných pravidiel a nemala ani schopnosť učiť sa.

SHRDLU – systém pre porozumenie prirodzeného jazyka naprogramovaný Terryom Winogradom [100] umožňoval používateľovi komunikovať o jednoduchom simulovanom (vizualizovanom) „svete kociek“, pozostávajúcom z trojrozmerných útvarov rôznych farieb a veľkostí, v ktorom SHRDLU konal ako robot vybavený okom a magnetickou rukou. Užívateľ mohol dávať (textové) príkazy v prirodzenom jazyku, na základe ktorých SHRDLU manipuloval s objektami v simulovanom svete, alebo sa

mohol pýtať otázky o svete a o vykonaných akciách. Dôsledky akcií boli vizualizované na obrazovke. SHRDLU-ova znalosť o svete bola reprezentovaná z časti ako formuly predikátovej logiky a z časti ako procedúry operujúce na formulách. Atomické prvky reprezentácie boli symbolové a dané vopred. Obmedzená schopnosť SHRDLU učiť sa spočívala v pridávaní nových tvrdení do modelu sveta, na základe vstupu od užívateľa a predchádzajúcej histórie akcií v simulovanom svete. Keďže SHRDLU dokázal v simulovanom svete vykonávať akcie podľa pokynov v prirodzenom jazyku a dokázal správne odpovedať na otázky o stave sveta kociek, možno povedať, že jeho limitované významy boli ukotvené vo virtuálnom prostredí.

Shakey – prvý fyzicky realizovaný mobilný robot schopný usudzovať o svojom konaní bol vyvinutý v Stanfordskom výskumnom inštitúte (SRI) v rokoch 1966-72. Disponoval televíznou kamerou a triangulačným zariadením na určovanie vzdialeností, umiestnenými na pohyblivej hlave, nárazovými senzormi, dvomi krokovými motormi na ovládanie kolies a ďalšími motorčekmi na riadenie zaostrenia kamery a naklonenia hlavy [57]. Shakey sa dokázal pohybovať po rovných povrchoch medzi kanceláriami a vykonával jednoduché úlohy, ktoré si vyžadovali plánovanie, určovanie trasy a prekladanie jednoduchých objektov. Udržoval si axiomatický model sveta a využíval hierarchiu programov na percepciu (vrátane jednoduchého videnia), modelovanie sveta a konanie. Využíval plánovací systém STRIPS [22], ktorý dokázal zreťaziť akcie do plánov a vykonať ich na dosiahnutie cieľa zadaného používateľom. Ciele boli reprezentované ako formuly prvorádovej predikátovej logiky. STRIPS mal aj istú schopnosť učiť sa – dokázal zovšeobecňovať plány a uchovávať ich pre budúce použitie. Môžeme povedať, že Shakey „rozumel“ svojmu svetu do tej miery, že disponoval reprezentáciami dôsledkov svojich akcií a využíval ich na plánovanie akcií za účelom dosiahnutia cieľov pomocou modifikovania svojho prostredia. Jeho znalosť, z väčšej časti predprogramovaná a reprezentovaná symbolmi prvorádových logických formúl, bola ukotvená v reálnom svete.

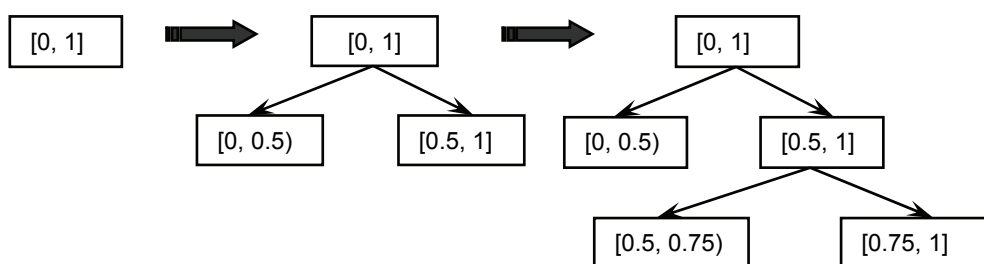
### 4.3 Diskriminačné stromy a prototypy

Teraz opíšeme niekoľko modelov, v ktorých sú významy reprezentované ako oblasti v nejakom geometrickom priestore obvykle definovanom možnými hodnotami senzorov, pomocou ktorých agenti vnímajú zdieľané prostredie: buď v jednorozmernom priestore pre každý senzor zvlášť, alebo vo viacrozmernom karteziánskom súčine oborov senzorických hodnôt. Významy v týchto modeloch nie sú dané vopred, ale každý agent si ich vytvára individuálne.

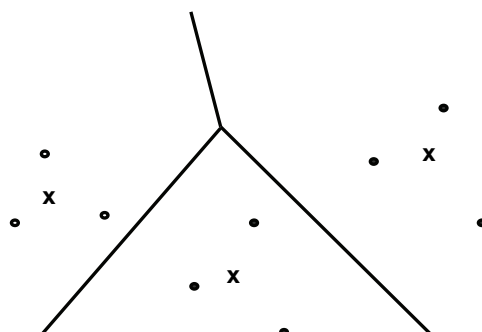
Senzorické priestory je možné rozdeliť na oblasti pomocou tzv. diskriminačných stromov [71], ktoré boli použité v mnohých modeloch emergencie jazyka, či už vo virtuálnych simuláciách [4, 69, 70] alebo v experimentoch s reálnymi robotmi [40, 72, 75, 76, 77]. Agenti konštruovali separátne diskriminačné stromy pre každý senzorický kanál tak, že vrcholy diskriminačného stromu zodpovedali podintervalom oboru hodnôt príslušného senzora. Tým zároveň determinovali granularitu reprezentácie: všetky hodnoty senzora, ktoré padli do nejakého intervalu, boli reprezentované príslušným vrcholom (teda boli navzájom nerozlišiteľné). Na začiatku každý strom pozostával iba z koreňa reprezentujúceho celý obor hodnôt daného senzora. Stromy sa



postupne adaptívne zjemňovali rozdeľovaním intervalov vrcholov na dve polovice, t.j. vytvorením dvoch synov daného vrchola (obr. 3), v tzv. diskriminačných hrách [75] zameraných na nájdenie významu (vrchola), ktorý jednoznačne odlišuje zvolený objekt (tému komunikácie) od všetkých ostatných súčasne prítomných objektov (kontextu). Užitočnosť každého rozdelenia sa monitorovala zaznamenávaním úspešnosti použitia v nasledujúcich diskriminačných hrách; irelevantné (nepoužívané alebo neúspešné) rozlišovania sa zahadzovali. Týmto spôsobom diskriminačné úlohy poskytli konštruovaným významom ukotvenie.



**Obr. 3.** Rozdelenie diskriminačného stromu. Obor hodnôt senzora sa opakovane delí na polovice (situácie, teda výsledný strom nemusí byť vyvážený).



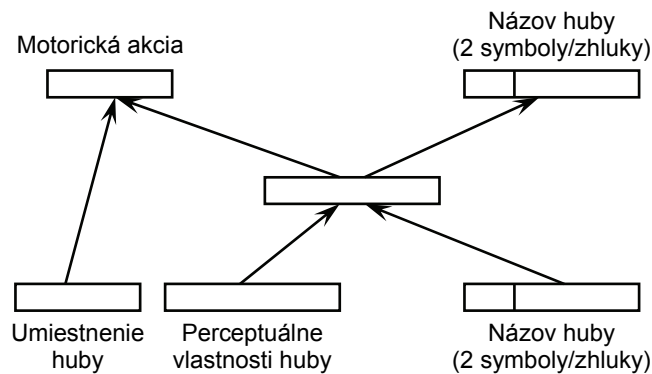
**Obr. 4.** Voronoiho mozaika priestoru kategórií generovaná prototypmi. Okrúhle body reprezentujú exempláre kategórií, krížiky (x) sú prototypy vypočítané ako ťažiská (stredy) príkladov.

Alternatívou diskriminačným stromom je prototypová reprezentácia významov, ktorá má svoj pôvod v empirickom zistení, že niektoré exempláre kategórií sú reprezentatívnejšie ako iné [65]. Takáto reprezentácia kategórií je efektívna, pretože si stačí pamätať najlepšie príklady (prototypy). Každý bod senzorickeho priestoru sa potom považuje za člena kategórie reprezentovanej priestorovo najbližším prototypom (obr. 4). Prototypy boli použité na reprezentáciu významov v experimentoch so

softvérovými agentami v simulovanom prostredí [93, 95] ako aj u reálnych robotov [91, 92]. My si použitie diskriminačných stromov a prototypov predvedieme vo výpočtových modeloch horizontálnej koordinácie lexikónu v časti 6.2.

#### 4.4 Dvojcestné neurónové siete

V konekcionistických modeloch ukotvenia symbolov sa často používa dvojcestná architektúra, ktorá obvykle obsahuje vizuálny vstup (napr. projekciu umelej siete) a jazykový vstup (napr. lokalistické alebo grafémové/fonémové kódovanie symbolov). Výstupná vrstva obsahuje symbolové jednotky na reprezentáciu slov a buď kategoriálnu reprezentáciu vstupných stimulov (napr. lokalistické výstupné jednotky pre každú kategóriu alebo vizuálnu reprezentáciu prototypov kategórií) alebo reprezentáciu želanej akcie (napr. hodnoty uhlov v kĺboch ramena). Všetky vstupné a výstupné vrstvy sú prepojené pomocou zdieľanej skrytej vrstvy. Cesta od vizuálneho vstupu k jazykovému výstupu sa používa na produkciu jazyka, napr. na pomenovanie (kategórie) objektu na vizuálnej scéne. Cesta od jazykového vstupu k vizuálnemu/kategoriálnemu/motorickému výstupu sa používa na porozumenie jazyka (s prípadnou akciou na základe jazykovej inštrukcie). Dve ďalšie cesty sa používajú na kategorizáciu, resp. akciu na základe vnemu (cesta od vizuálneho vstupu ku kategoriálnym/motorickým jednotkám) a na jazykovú imitáciu (cesta od jazykového vstupu k jazykovému výstupu) [11]. Dvojcestné neurónové siete boli použité napr. v modeloch [10, 13, 54].



**Obr. 5.** Príklad dvojcestnej neurónovej siete použitej v experimente Cangelosiho [10].

V modeli [10] žije populácia 80 organizmov vo virtuálnom prostredí obsahujúcom tri druhy jedlých a tri druhy jedovatých húb. Model využíva genetickú evolúciu organizmov. Organizmy získavajú fitness vtedy, ak sa vyhýbajú jedovatým hubám a vedú identifikovať druh jedlej huby a zjedia ju. Správanie každého organizmu je riadené dvojcestnou neurónovou sieťou (obr. 5) ktorá umožňuje vykonávať akcie na základe miesta výskytu a perceptuálnych vlastností huby a/alebo jazykovej inštrukcie od iných agentov, a tiež pomenovávať huby (buď na základe

miesta výskytu a perceptuálnych vlastností alebo ako imitácia jazykového vstupu iného agenta). Na začiatku je lexikón úplne náhodný a nezmyselný. Ku koncu evolúcie sú však agenty schopné vyvinúť si zdieľaný kompozičný jazyk. Hoci prezentovaný model je zámerne zjednodušený, predstavuje korektný konekcionistický prístup k ukotveniu symbolov. Významy, implicitne reprezentované vo váhach spojení siete, sú súkromné a vytvorené každým agentom samostatne. Osvojovanie si jazyka je spriahnuté so senzomotorickými aktivitami cez spoločnú skrytú vrstvu dvojcestnej architektúry.

#### 4.5 Korpusový prístup k významom

Hoci distribučné a kontextuálne charakteristiky slov nie sú postačujúce pre ukotvenie jazyka v reálnom svete (pozri Argument čínskej izby), predsa predstavujú dôležitú pomoc pre určenie významu slov [9, 52], čoho impresívnou demonstráciou sú konverzačné agenty (angl. chatter bots, chatbots) založené na kontextovom prehľadávaní korpusu. Chatbot je počítačový program, ktorý má simulovať inteligentnú konverzáciu s ľudskými užívateľmi. Mnoho chatbotov (ako ELIZA) je založených na rozpoznaní kľúčových slov v textovom vstupe a odpovedaní podľa predprogramovaných pravidiel. Chatbot Jabberwacky, ktorý naprogramoval Rolo Carpenter, je založený na odlišných princípoch: nie sú v ňom žiadne pevné predprogramované pravidlá a pracuje výhradne na základe interakcie s užívateľom [14]. Systém si uchováva obrovskú databázu všetkých predchádzajúcich rozhovorov a pokúša sa ju využiť na nájdenie najpriliehavejšej odpovede v aktuálnom kontexte. Prehľadávanie je založené na komplexnej viacvrstvovej sade heuristik, ktoré analyzujú konverzačný kontext a využívajú pozitívnu spätnú väzbu [39].

Úspešné budenie dojmu, že Jabberwacky naozaj rozmýšľa, spočíva vo veľkosti databázy. Jabberwacky bol on-line na webe od roku 1997 a zaznamenal viac ako 13 miliónov konverzácií. V komunikácii používa to, čo sa naučil od svojich konverzačných partnerov. Takto používa vtipy, idiómy, slovné hračky, slang a dokonca hovorí cudzími jazykmi. Dve nedávne inštancie Jabberwackyho – George a Joan vyhrali Loebnerovu cenu za rok 2005 a 2006. Loebnerova cena je každoročná súťaž, v ktorej cenu dostane ten chatbot, ktorého rozhodcovia považujú za všetkých zúčastnených za najbližšieho človeku. Formátom súťaže je štandardný Turingov test [90].

Hoci Jabberwackyho konverzácia môže byť ľuďmi interpretovaná ako zmysluplná – majúca nejaký sémantický obsah, tento obsah nie je vlastný a Jabberwacky o ňom nič nevie. Jazyková znalosť nie je ukotvená vo svete. Avšak z odlišného uhla pohľadu je Jabberwacky situovaný vo svete konverzačných postupností a v tomto svete sa naučí adekvátne reagovať v rôznych kontextoch. Znalosť o tom, čo je adekvátne, je zakódovaná v zaznamenaných odpovediach Jabberwackyho ľudských partnerov. V súlade s Brooksom [6, 7, 8] by sme mohli povedať, že Jabberwackyho inteligentné správanie je emergentným efektom jeho interakcií, pričom znalosť je distribuovaná z časti v jeho architektúre a z časti v prostredí.

## 4.6 Ďalšie modely

Z priestorových dôvodov nemôže byť náš prehľad modelov vyčerpávajúci. Medzi ďalšie zaujímavé prístupy k reprezentácii významov patria semiotické schémy [66], dynamické mapy [18], prepisy súčasných udalostí [19] a stelesnená konštrukčná gramatika ECG [1]. Čitateľa odkazujeme na pôvodné zdroje.

## 5 Konštrukcia významov

### 5.1 Individuálne významy

Predstavili sme základné teoretické prístupy, evolučný kontext rozumenia a tiež niektoré problémy s rozumením v umelých systémoch. Z toho všetkého možno vyvodit' ponaučenie, že na prítomnosť rozumenia či významov v živočíšnej ríši a v umelých systémoch treba nazerať skôr ako na kontinuum od veľmi primitívnych foriem až po zložité a nie ako na niečo, čo buď je v plnej forme alebo nie je vôbec. Pripravili sme si tak pôdu pre vyslovenie našej vlastnej predstavy o rozumení podľa možnosti čo v najneutrálnejších, nie antropocentrických (a dokonca nie biologických) pojmoch.

O rozumení budeme hovoriť iba u agentov (živých či umelých) s ohľadom na (reálne či virtuálne) prostredie, v ktorom sú situované. *Situovaný agent* je autonómna entita, ktorá v prostredí dosahuje svoje ciele pomocou vnímania a konania [42]. Všimnime si, že táto definícia neimplikuje ani nevyžaduje vedomie či intencionalitu: agent môže dosahovať nejaké ciele (alebo plniť nejaký účel) bez toho, aby o tom „vedel“.

Adaptívne agenty modifikujú samy seba, aby lepšie dosahovali svoje ciele. Budeme rozlišovať medzi fylogenetickými (dizajn-time, pri návrhu) a ontogenetickými adaptáciami (run-time, počas behu). Učenie je ontogenetická adaptácia založená na spätnoväzobnej slučke medzi percepciou a akciou (alebo všeobecnejšie, medzi vstupom a výstupom). Vplyv adaptácií na správanie agenta možno pohodlne opísať pomocou teoretických konštruktov „reprezentácia“ a „semiotický znak“ bez ohľadu na ich reálnosť, pôvod či implementáciu. Ak agent kategorizuje svet tým, že diferencovane reaguje na rôzne skupiny podnetov, hovoríme, že disponuje vyvolanými reprezentáciami, ktoré sú najelementárnejšou formou významov. Zložitejšie kognitívne schopnosti, ako rozhodovanie a plánovanie, si vyžadujú oddelené reprezentácie, ktoré môžu byť uchovávané, vyvolávané a spracovávané nezávisle od vonkajších environmentálnych spúšťačov.

### 5.2 Sociálne zdieľané významy

Doteraz sme hovorili o subjektívnych a súkromných významoch, ktoré záviseli od fylogenetickú a ontogenetickú históriu každého agenta. Kooperatívne plánovanie medzi agentami si vyžaduje externalizáciu a komunikáciu významov pomocou verejne vnímateľných signálov. Povedané v semiotických termínoch, každý agent interpretuje

signály ako znaky tým, že doplní dva zvyšné vrcholy a príslušné hrany semiotického trojuholníka (obr. 2d). Hovoríme, že agent pozorujúci signál ho chápe akoby mal sémantický obsah, ak sa správa v súlade s týmto obsahom [36]. Tento pohľad na rozumenie je interakcionistický: súkromné významy nemožno prenášať priamo; možno ich iba odvodiť z behaviorálnych interakcií. Komunikácia však môže byť úspešná, iba ak si budú súkromné významy dostatočne podobné. Na to sa dá spoľahnúť, ak sú asociácie medzi vrcholmi semiotického trojuholníka nearbitrárne a boli formované rovnakým mechanizmom, ako je to napr. v prípade vrodenných zvieracích signálnych systémov alebo u neadaptívnych umelých agentov riadených rovnakým predprogramovaným kódom. Avšak v *symbolických* komunikačných systémoch sú vzťahy medzi významami a signálmi arbitrárne z definície [16]. Ak boli významy formované alebo ovplyvnené individuálnou adaptáciou, vzájomná podobnosť významov u rôznych agentov nie je zaručená. Významy jednotlivých agentov je potrebné skoordinať a vzájomne zladit' adaptáciami v sociálnych interakciách. Tieto interakcie môžu byť jednosmerné (vertikálne), ak agent prispôsobí svoje vlastné reprezentácie existujúcemu komunikačnému systému bez toho, aby ho nejako ovplyvnil (napr. ak sa malé dieťa učí jazyk, ktorému je vystavené). Ak sú interakcie obojsmerné (horizontálne), t.j. vedú k adaptáciám na oboch stranách, celý jazyk sa správa ako dynamický systém, v ktorom „verejnú“ (zjednanú) významy emergujú ako (pohyblivé) ekvilibriá [28].

### 5.3 Princípy dizajnu rozumejúcich agentov

Z prezentovaného interakcionistického pohľadu na významy možno vyvodit' kritériá rozumenia u umelých agentov a princípy návrhu takýchto agentov:

- 1) Významy musia pochádzať z interakcií agentov s prostredím, v ktorom sú situované. Prostredie môže byť reálne, ale aj virtuálne – napr. softvérové agenty môžu vnímať prehľadávaním databáz a vykonávať akcie napr. uzatváraním elektronických obchodných transakcií.
- 2) Netriviálne prostredia sú dynamické, menia sa v čase. Preto musia byť agenty schopné okrem statických objektov a vzťahov zachytiť a reprezentovať dynamické vlastnosti sveta, ako sú zmeny, akcie a ich dôsledky, udalosti.
- 3) Keďže prostredie je otvorené, nie je možné všetky významy anticipovať a predprogramovať vopred (v čase návrhu), preto je potrebná ontogenetická adaptácia (učenie). Učenie – konštrukcia významov – by malo byť inkrementálne a permanentné.
- 4) Aby sa zabezpečilo vzájomné porozumenie v komunikácii, agenty musia byť vybavené mechanizmami na sociálnu koordináciu individuálne konštruovaných významov.

## 6 Výpočtové modely konštrukcie významov

V tejto časti predstavíme niekoľko výpočtových modelov ilustrujúcich horeuvedené princípy. Z didaktických dôvodov sme vybrali modely zamerané na jednotlivé princípy, či konkrétny spôsob ich implementácie, aj keď nespĺňajú všetky princípy zároveň.

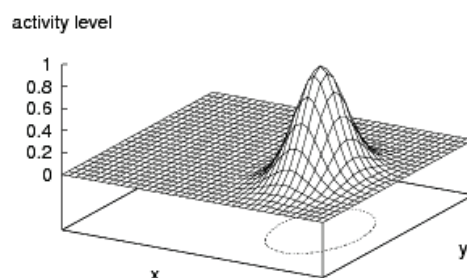
### 6.1 Individuálna senzomotorická explorácia

Ako prvý predstavíme výpočtový model individuálneho konštruovania významov ukotvených v senzomotorických interakciách [81]. Model pozostáva z jediného agenta situovaného v simulovanom prostredí – na dvojrozmernej mriežke s náhodne rozmiestnenými objektami štyroch typov: ovocie, hračky, nábytok a sám agent. Na začiatku agent nerozlišuje žiadne typy, len dostáva informáciu o vnímateľných vlastnostiach objektov okolo seba vo forme rámcov – množín dvojíc *(atribút: numerická hodnota)*. Atribúty kódujú vlastnosti ako pozícia na mriežke, veľkosť, hmotnosť, farba, tvar, atď. Čas je v modeli diskretný. V každom časovom kroku agent dostáva na vstupe jeden rámec pre každý vnímaný objekt. Agent si náhodne vyberá jeden z objektov a pokúša sa ním manipulovať – zdvihnúť ho alebo položiť. Výsledkom úspešných manipulácií sú zmeny hodnôt atribútov manipulovaného objektu (v prípade dvíhania a kladenia vertikálna poloha). Akcie sú parametrizované – agent si volí silu vynaloženú na dvíhanie a vertikálnu polohu vlastného ramena (tieto parametre sú zjednodušenou abstrakciou simulovanej motorickej akcie – v skutočnosti agent nemá žiadne rameno, ani motory, atď.). Maximálne hodnoty sily a polohy sú obmedzené „konštrukciou“ agenta.

Agent pozoruje dôsledky svojich akcií porovnaním hodnôt atribútov objektu v dvoch po sebe nasledujúcich časových krokoch. Skutočné dôsledky akcií sú vyhodnocované simulátorom prostredia podľa zjednodušených „fyzikálnych zákonov“, napr. ak je sila vynaložená na dvíhanie príliš malá v porovnaní s hmotnosťou dvíhaného objektu, objekt sa nepohne, v opačnom prípade sa dvihne priamo úmerne zvolenej vertikálnej polohe agentovho ramena.

Agent aktívne exploruje svoje prostredie vykonávaním náhodných akcií a postupne sa učí rozlišovať relevantné vlastnosti prostredia a buduje si kategórie objektov, akcií a zmien. Kategórie sú reprezentované tzv. *rozlišovacími kritériami* inšpirovanými už spomínanou kognitívnou sémantikou Šefránka [79]. Každé rozlišovacie kritérium je aktivačnou funkciou, ktorá pre svoj vstup vracia mieru príslušnosti vstupu v reprezentovanej kategórii (obr. 6). Medzi možné vstupy patrí perceptuálny rámec jedného objektu (u kritérií objektov a ich vlastností), „proprioceptívny“ rámec parametrov a typu akcie (u kritérií akcií), perceptuálne rámce viacerých objektov (u kritérií vzťahov), rámce toho istého objektu v rôznych okamihoch (u kritérií zmien) a výstupné aktivity iných kritérií (u kompozičných kritérií situácií a udalostí). Na začiatku agent nemá žiadne kritériá, tie si postupne konštruuje extrakciou spoločných štatistických vlastností exemplárov kategórií. Technické detaily čitateľ nájde v [82, 84]. Ak nejaká akcia vykonaná na rôznych objektoch vedie k dostatočne podobným dôsledkom (zmenám atribútov), tieto objekty budú

považované za členov tej istej kategórie (vzhľadom na danú akciu a dôsledok). V prípade príliš odlišných dôsledkov sa vytvoria nové kategórie. Rovnaký princíp sa používa aj na združovanie rôznych akcií vykonaných na tom istom objekte.



**Obr. 6.** Rozlišovacie kritérium s Gaussovskou krivkou aktivácie nad dvojrozmerným vstupným priestorom.

Propozičné znalosti o kauzálnych vzťahoch medzi akciami, objektami a zmenami sú reprezentované pomocou asociácií medzi príslušnými kategóriami (ako analógia asociálnych oblastí v mozgovej kôre). Takáto reprezentácia je v súlade s poznatkami o ľudskej organizácii znalostí. Podľa Gibsona dokážeme z prostredia priamo vnímať možnosti konania – tzv. *afordancie* [32]. Afordancie zahŕňajú spôsoby manipulovania s vecami, napr. dvere, ktoré sa dajú otvoriť, potrava, ktorá sa dá zjesť, povrchy, po ktorých sa dá kráčať, atď. V našom modeli sú afordancie objektu reprezentované ako všetky s ním asociované akčné kategórie. Objekty a zmeny asociované s akčnými kategóriami zase tvoria zárodok sémantickej reprezentácie centrovanej okolo slovíes, tzv. *slovesné ostrovy*, ktoré v organizácii detského lexikónu opísal Tomasello [88].

V počítačovej simulácii nášho modelu sme nechali agenta interagovať s prostredím počas 5000 krokov a merali sme užitočnosť vytvorenej reprezentácie ako schopnosť predikovať dôsledky vlastných akcií. Experimentálne výsledky potvrdili, že navrhnutý mechanizmus vytvárania kategórií na základe akcií vedie k ekologicky relevantným kategóriám. Technické detaily algoritmu a výsledky možno nájsť v [81].

Kvalitatívna analýza skonštruovaných významov vo vybranom simulačnom behu ukázala, že agent si vytvoril štyri objektové kategórie. Dve z nich by bolo možné vzhľadom na asociovanú kauzálnu znalosť interpretovať ako „veci príliš ťažké na to, aby sa dali zdvihnúť“ a „veci, ktoré sa nedajú položiť, pretože už sú na zemi“. Do tretej kategórie padla väčšina objektov typu ovocie a do štvrtej väčšina hračiek. V kritériách ostali prítomné aj iné atribúty ako hmotnosť a vertikálna poloha, čo znamená, že nejakým spôsobom pomáhali agentovi v klasifikácii (napr. ak by sa všetky ťažké objekty nachádzali v rovnakej časti mriežky alebo by mali rovnakú farbu). Konštruovaná reprezentácia bola teda situovaná a obsahovala aj subjektívny interakčný kontext agenta.

Tento experiment modeluje aktivity dieťaťa v senzomotorickom štádiu detského vývinu [62], v ktorom dieťa pomocou náhodných a spočiatku nekoordinovaných pohybov spoznáva objekty okolo seba, vlastné telo, a možnosti ich vzájomného

interagovania. Opakujúce sa kauzálne vzorce sú zaznamenané, čo neskôr umožní zámerné plánovanie akcií na splnenie vytýčených cieľov. Významným aspektom prezentovaného modelu je to, že agent je v čase návrhu vybavený iba učiacimi mechanizmami, a všetky znalosti o konkrétnom prostredí získa až počas behu. To umožňuje agentovi byť flexibilný a adaptovať sa na rôzne prostredia, ktoré nemuseli byť v čase návrhu známe, alebo sa mohli zmeniť počas agentovho „života“ (predstavme si analogickú úlohu naprogramovať robota, ktorý bude vyslaný na vzdialenú planétu s neznámymi podmienkami).

Z vytýčených princípov dizajnu rozumejúcich agentov tento model spĺňa princípy 1 až 3. Percepčno-akčné rozhranie modelu bolo (zámerne) značne abstraktné a zjednodušené. V stelesnených systémoch situovaných v reálnych prostrediach patria úlohy spojené s percepciou a spracovaním zašumeného analógového senzorickeho signálu a riadením efektorov (napr. servomotorov mechanického ramena) medzi tie ťažšie. Keďže model pozostával iba z jediného agenta, nebola v ňom žiadna sociálna koordinácia významov ani jazyk (princíp 4). Týmto aspektom sú venované nasledujúce časti.

## 6.2 Horizontálna koordinácia významov

V tejto časti predstavíme modelársky prístup skupiny Luca Steelsa ilustrujúci emergenciu koordinovaného komunikačného systému na základe horizontálnych kultúrnych interakcií [73]. Horizontálne interakcie sú obojsmerné, čo znamená, že každá z komunikujúcich strán sa snaží prispôsobiť tej druhej. Ak by sme si opäť predstavili semiotický trojuholník (obr. 1 a 2d), adaptácia prebieha na úrovni hrán aj vrcholov: agent mení svoje súkromné asociácie medzi externe pozorovateľnými signálmi, internými významami a ich príslušnými referentami vo svete, a tiež mení aj samotné interné významy.

Adaptácia je riadená spätnou väzbou o úspešnosti resp. zlyhaní komunikácie. Úspech v komunikácii nemusí nutne znamenať, že agenty majú s rovnakým signálom asociované identické významy: keďže interné významy nie sú viditeľné a nie je ich možné priamo porovnávať, kritériá úspešnosti sú pragmatické, teda znaky sa musia zhodnúť na úrovni referentov, čo je manifestované správaním agentov. Behaviorálne kritériá sú plauzibilné: ak napr. požiadam niekoho, aby mi podal pero, a on mi namiesto toho podá jablko, alebo otvorí dvere, môžem odôvodnene predpokladať, že ma neporozumel (príp. žartuje, je zlomyseľný, atď.).

V modeloch založených na tomto prístupe agenty (simulované alebo stelesnené v reálnych robotoch) hrajú rôzne typy tzv. *jazykových hier* s rôznymi pragmatickými cieľmi [75]. Hry sa hrajú v kolách: v každom kole sa z populácie náhodne vyberú dva agenty – jeden v role hovorcu a druhý v role poslucháča. Najprv zohrá hovorca tzv. *diskriminačnú hru*, v ktorej sa snaží jednoznačne identifikovať zvolený objekt na scéne (tému komunikácie) pomocou významov a tie následne lexikalizovať do signálov. Na základe týchto signálov sa potom poslucháč pokúša uhádnuť tému komunikácie v tzv. *hádacej hre*.

Agenty si ukladajú konceptuálne znalosti (významy) v podobe diskriminačných stromov, a jazykovú znalosť v podobe viacnásobných asociácií medzi vrcholmi



stromov a signálmi. Stromy aj asociácie sú konštruované a adaptované individuálne na základe úspešnosti v jazykových hrách (asociácie sa adaptujú tak, že sa mení ich číselne vyjadrená sila).

V prípade úspešnej diskriminačnej hry sa hovorca snaží lexikalizovať vybrané významy použitím tzv. stratégie *introspektívneho obvertera* [69]: zo všetkých signálov asociovaných s daným významom vyberie ten, ktorý by on sám s najväčšou pravdepodobnosťou interpretoval ako zvolený význam. Poslucháč sa potom pokúša interpretovať signál pomocou vlastných interne asociovaných významov a hľadá medzi objektami na scéne zodpovedajúci referent. Na konci poslucháč dostane (neverbálnu) spätnú väzbu o tom, či jeho tip na referenta bol správny a obaja si prislúšnym spôsobom adaptujú silu zúčastnených asociácií. V inom variante nazvanom *pozorovacia hra* [58] poslucháč nedostáva žiadnu spätnú väzbu. Namiesto toho sa množina možných referentov obmedzí sledovaním pohľadu (alebo ukazovania) hovorcu a poslucháč si upravuje asociácie vo svojom lexikóne Hebbovským učením (ak je viac možných kandidátov, vytvoria/posilnia sa asociácie s každým z nich). Nejednoznačnosti sa časom redukujú učením z viacerých situácií [68, 70].

Známou materiálnou realizáciou hádacích hier bol experiment *Hovoriace hlavy* – *Talking Heads* [72]. Na scéne boli inštalované dve kamery, do ktorých sa dali uploadovať viaceré agenty. Agenty pomocou kamier vnímali zdieľané prostredie – magnetickú tabuľu s geometrickými tvarmi rôznych farieb a veľkostí a komunikovali o nich.

Ešte sa zmienime o modeli hádacej hry, ktorý realizoval Vogt [92]. V tomto modeli boli agenty stelesnené v dvoch mobilných LOGO robotoch. Agenty používali prototypovú reprezentáciu významov. Cieľom jazykovej hry bolo komunikovať názov jedného zo svetelných zdrojov, ktoré roboty dokázali detekovať v prostredí.

V oboch zmienených modeloch sa agentom podarilo vytvoriť koordinovaný lexikón ukotvený v reálnych podmienkach. Význam týchto modelov je v tom, že demonštrujú, ako môže globálne koherentný komunikačný systém vzniknúť „z ničoho“ ako výsledok lokálnych interakcií jeho používateľov. Agenty začínajú bez významov (len s nediferencovanými senzorickými kanálmi) a s prázdny lexikónom – celý systém sa naštartuje vďaka mechanizmu invencie: v prípade, že agent nemá v lexikóne slovo na vyjadrenie nejakého významu, vytvorí si a asociuje nové náhodné slovo. Samozrejme, takémuto signálu spočiatku poslucháč neporozumie; pri adaptácii si však zapamätá asociáciu medzi signálom a významom a v budúcnosti ho už môže úspešne použiť. Týmto spôsobom sa signály šíria v populácii. Tendencia agentov používať tie signály a významy, ktoré sa v minulosti osvedčili, vytvára pozitívnu spätnoväzobnú slučku. Viaceré alternatívne signály a významy medzi sebou súťažia a víťazné sa posilňujú. To vedie k samoorganizácii a k emergencii koherentného komunikačného systému. Spontánna samoorganizácia jazyka v rámci jednej generácie a na základe čisto kultúrnych horizontálnych interakcií bola pozorovaná aj medzi ľuďmi – hluchonemými deťmi v prípade Nikaragujskej posunkovej reči [41].

Ak sa poohliadneme za vytýčenými princípmi dizajnu rozumejúcich agentov, tu prezentované modely spĺňajú kritériá 1, 3 a 4. Každý agent si individuálne vytvára a permanentne adaptuje významy na základe interakcií s prostredím. Významy sú interné – nedostupné iným agentom a potenciálne rôzne u rôznych agentov. Sú však

navzájom zladené, čo umožňuje agentom dosiahnuť ciele v jazykových hrách. Kritérium 2 je však porušené: agenti dokážu komunikovať iba o statických objektoch aktuálne prítomných v ich dohľade. Nemajú žiadne prostriedky ani na reprezentáciu, ani na komunikáciu dynamických aspektov prostredia. Komunikácia je založená iba na vyvolaných reprezentáciách.

### 6.3 Učenie na základe (vertikálnej) sociálnej inštrukcie

V tejto časti sa zameriame na jednosmerné interakcie medzi agentami a ich vplyv na proces adaptácie významov. Jednosmerný adaptačný proces je učením, preto budeme agenta, ktorý sa neadaptuje, nazývať *učiteľ* a agenta, ktorý sa adaptuje, *učiaci sa*. Tento typ učenia hrá významnú úlohu v procese osvojovania si jazyka u detí. Predstavíme výpočtový model inšpirovaný empirickým výskumom Sandry Waxman, ktorá skúmala vplyv verbálnej inštrukcie (pomenovania) na proces formovania kategórií [97]. Výsledky jej výskumu napovedajú, že konzistentné používanie rovnakého názvu pre rôzne objekty motivuje deti k hľadaniu podobností medzi nimi a k formovaniu kategórií.

Výpočtový model [85] pozostáva z dvoch agentov situovaných v simulovanom prostredí: učiteľ opisuje rôzne aspekty meniacej sa scény a učiaci sa agent si vytvára významy indukciou podobností medzi referentami vo viacerých situáciách. Simulované prostredie obsahuje plošné geometrické tvary charakterizované náhodne generovanými numerickými atribútmi ako počet vrcholov, poloha (súradnice stredu objektu) a veľkosť (opísaného obdĺžnika). Prostredie je dynamické – náhodne zvolené objekty môžu zmeniť veľkosť, polohu, alebo môžu byť odobraté z prostredia, tiež môžu byť pridané nové objekty (v jednom časovom kroku sa môže udiť aj viac zmien naraz).

Učiaci sa agent si vytvára reprezentáciu významov v podobe už zmienených rozlišovacích kritérií. Na začiatku nemá žiadne – tie si vytvára indukciou zo spoločných štatistických vlastností prezentovaných príkladov. Učiteľ v priebehu učenia pomenúva rôzne aspekty objektov, ich vlastností, vzťahov a dynamických zmien a učiaci sa agent musí nejako rozhodnúť, ku ktorým významom tieto pomenovania priradiť, kedy aktualizovať existujúce kritériá a kedy vytvárať nové. Indukcia významov sa riadi dvoma princípmi: predpoklad *žiadnej pravej synonymie* a *žiadnej pravej homonymie*. Aj keď prirodzené jazyky obsahujú slová s viacerými významami (homonymá) a aj viaceré výrazy pre ten istý význam (synonymá), v prípade tvorenia jazyka a významov z ničoho je užitočné začať s takýmito predpokladmi.

- *Žiadna pravá synonymia*: Rôzne slová majú rôzny význam, aj keď zdieľajú referent (v tom prípade označujú rozličné aspekty referenta). Tento predpoklad zodpovedá *Princípu kontrastu* [17], ktorý je známy z akvizície jazyka deťmi.
- *Žiadna pravá homonymia*: Jedno slovo má len jeden význam, aj keď označuje viacero referentov. Tento predpoklad je kľúčový pre riešenie nejednoznačnosti označenia učením sa z viacerých situácií: referenty toho istého slova tvoria vzorku – množinu inštancií kategórie označenej daným slovom.

**Príklad:** Majme agenta, ktorý je situovaný vo svete geometrických tvarov na mriežke 50×50 pričom bod so súradnicami (1, 1) je najviac vľavo dolu a bod so

súradnicami (50, 50) najviac vpravo hore. Ak agent vníma objekt daný rámcom  $f = \{ \text{počet\_vrcholov: } 3; \text{ veľkosť: } 18; \text{ farba: } 3; \text{ pozícia\_X: } 1; \text{ pozícia\_Y: } 23 \}$  pomenovaný slovami „vľavo“, „veľký“, „trojuholník“, vytvorí si tri rozlišovacie kritériá, ktoré sú na začiatku úplne identické a reprezentujú „momentku“ objektu danú rámcom  $f$ . Ako sú kritériá postupne aktualizované viacerými inštanciami, začínajú sa diferencovať. Napríklad rámce rôznych objektov, ktoré majú najrôznejšie farby, polohy, veľkosti a ďalšie vlastnosti, ale všetky majú 3 vrcholy, budú aktualizovať rozlišovacie kritérium asociované so slovom „trojuholník“. Atribúty, ktoré sa nevyskytujú vo všetkých inštanciách, sa z kritéria odstránia, a tie, ktoré mali vo vzorke veľký rozptyl, budú mať malú dôležitosť. Tým sa vlastnosť mať tri vrcholy (ktorá je vo vzorke invariantná – má nulový rozptyl) stane rozhodujúcou. Podobne slovo „vľavo“ bude označovať najrôznejšie objekty, ktorých spoločnou vlastnosťou bude malá hodnota atribútu *pozícia\_X*, atď. Čím viac kontextov použitia slova bude k dispozícii, tým je väčšia pravdepodobnosť, že irelevantné vlastnosti budú mať vo vzorke veľkú variabilitu. Avšak keby napríklad všetky trojuholníky, ktoré agent doteraz vnímal, boli veľké, vlastnosť byť veľký by sa stala súčasťou významu slova „trojuholník“. Takto vytvorená reprezentácia je preto vždy situovaná a kontextovo závislá.

Simulácia opísaného experimentu bežala 5000 krokov. V každom kroku učiteľ vybavený predprogramovanou ontológiou a lexikónom opísal aktuálnu scénu vrátane dynamických zmien. Učiteľov lexikón obsahoval 2 podstatné mená, 3 adjektíva a slovesá (štvorec, trojuholník, veľký, štíhly, malý, narásť, zmenšiť sa). Úlohou učiteľa bolo naštruktúrovať vzorku pre učiaceho sa agenta – rovnako dobre by mohol poslúžiť človek, ktorý by sledoval scénu a ručne dodával opisy toho, čo vníma. To by však bolo časovo neefektívne, preto bol pre urýchlenie experimentu použitý automatický učiteľ. Učiteľ zároveň ukázal na referent pomenovávanej kategórie, t.j. na objekt s danou vlastnosťou, zmenou alebo reláciou.

Toto zjednodušenie bolo použité zámerne, aby sme ukázali, že aj v neprítomnosti nejednoznačnosti *referenta* musí učiaci sa agent riešiť nejednoznačnosť *zmyslu*, pretože rôzne slová môžu opisovať rôzne aspekty toho istého (známeho) referenta. Povedané v termínoch semiotického trojuholníka, úlohou učiaceho sa agenta je na základe dvoch pozorovateľných vrcholov (signálu a referenta) skonštruovať alebo adaptovať tretí vrchol – svoj interný význam/zmysel a príslušné asociácie.

Za účelom merania kvality vytvorených významov učiaci sa agent tiež opisoval vnímanú scénu a opis sa porovnával s učiteľovým. Zaznamenával sa počet *chýbajúcich* slov (takých, ktoré na opis daného referenta použil učiteľ, ale nie učiaci sa) a *nesprávnych* slov (ktoré použil žiak, ale nie učiteľ). Okrem toho učiaci sa agent hral klasické hádacie hry a zaznamenávala sa úspešnosť v hádaní referenta opisu použitého učiteľom. Výsledky ukázali, že učenie sa z viacerých situácií je rýchle a spoľahlivé: učiaci sa agent si rýchlo vytvoril ontológiu a lexikón dostatočnej kvality, ktorý ostal stabilný po celý zvyšok simulácie. Detaily čitateľ nájde v [84, 85].

Významy skonštruované v tomto modeli neodrážajú priamo žiadne interakčné vlastnosti (afordancie) prostredia získané agentovou exploračiou. Namiesto toho sú determinované vopred existujúcou štruktúrou učiteľových kategórií sprostredkovanou pomocou jazyka. Experiment demonštruje ako možno kultúrne prenášať existujúci konceptuálny systém pomocou vertikálnych jazykových interakcií.

Jazyky však nie sú prenášané ako zakonzervované systémy, ale sú živé a v priebehu generácií podliehajú zmenám a vyvíjajú sa [20], napr. sa objavujú novotvary a miznú archaizmy, zjednodušuje a spravidelňuje sa syntax, atď. Emergentná štruktúra evoluujúceho sa jazyka je tvarovaná ontogenetickým procesom akvizície jazyka, ktorý je zase determinovaný (pováčšine vrozenými resp. predprogramovanými) mechanizmami učenia [5]. Medzigeneračné významové posuny v rámci iterovaného vertikálneho prenosu sme skúmali *metódou iterovaného učenia* [43] v experimente [83].

## 7 Záver

V tejto kapitole sme sa snažili zdôrazniť dôležitosť konštruktivistického prístupu k významom pre štúdium kognície. Rozpracovali sme pojmový aparát pre (inherentné) významy, ukázali sme, ako môžu umelé agenty individuálne konštruovať významy na základe interakcií s prostredím a ako je možné tieto významy kolektívne koordinovať. To je veľmi dôležité najmä v neznámych, dynamických a otvorených prostrediach, kde nie je možné anticipovať všetky možné významy v čase návrhu systému.

Čo sa týka komunikácie, poukázali sme na to, že identickosť interných významov nie je pre vzájomné rozumenie nutná; postačuje koordinácia na úrovni správania, čo vytvára dobrý základ pre našu komunikáciu (a koexistenciu) s umelo vytvorenými agentami. Inherentné významy odlišne skonštruovaných agentov s odlišnými cieľmi a ekologickými nikami budú odlišné od našich ľudských. Napriek tomu môžeme navzájom viesť zmysluplnú komunikáciu, ak si zjednáme spoločný komunikačný systém.

### 7.1 Možnosti ďalšieho výskumu

Na viacerých výpočtových modeloch sme ilustrovali individuálnu konštrukciu a kolektívnu koordináciu významov. Tieto modely boli v mnohých aspektoch zjednodušené, na demonštráciu realizovateľnosti navrhnutých ideí však postačujú. Z metodologických dôvodov sme skúmali opísané fenomény každý zvlášť: v modeli individuálnej konštrukcie významov chýbala komunikácia, v modeli koordinácie významov neboli žiadne mimojazykové aktivity, atď. Prirodzeným (a realistickejším) pokračovaním by mal byť model, ktorý integruje všetky spomínané aspekty.

V modeli senzomotorickej explorácie agent nemal žiadne ciele a vykonával akcie náhodne. Keďže si úspešne dokázal osvojiť a reprezentovať propozičné znalosti o dôsledkoch svojich akcií v prostredí, ďalším logickým krokom je rozšírenie modelu o potreby, potrebami generované ciele a mechanizmus plánovania akcií.

V modeli integrujúcom plánovanie, komunikačnú úroveň a mechanizmy horizontálnej a vertikálnej koordinácie bude možné experimentálne testovať, či autonómne agenty profitujú (efektívnejšie dosahujú ciele) z vzájomnej výmeny znalostí pomocou zdieľaného komunikačného systému.

Ďalší výskum by sa mal tiež zaoberať analýzou vyjadrovacej sily rôznych typov reprezentácie významov. Reprezentácia by mala zachytávať hierarchické

a taxonomické vzťahy medzi významami a podporovať sémantiku slovies. V reprezentácii pomocou diskriminačných stromov dynamické aspekty prostredia neboli zachytené vôbec. V sémantike rozlišovacích kritérií boli významy slovies konštruované z kritérií jedнокrokových zmien hodnôt atribútov. Pre niektoré slovesá môžu však byť potrebné dlhšie sekvencie, ešte iné možno vôbec nevystačia s diskrétnymi postupnosťami a budú potrebovať nejaký typ spojitaj reprezentácie dynamiky (napr. fázové portréty).

Jazyk bol vo všetkých prezentovaných modeloch viazaný na aktuálnu situáciu (vyvolaný). Primárnou evolučnou výhodou ľudského jazyka je však jeho oddelené používanie, t.j. komunikácia o veciach, ktoré presahujú „tu a teraz“. Oddelená komunikácia by bola príliš nejednoznačná bez použitia nejakej formy gramatiky [27]. Zakomponovanie gramatiky do modelov emergencie a akvizície jazyka predstavuje dôležitý smer budúceho výskumu. Prvé kroky v tomto smere už boli vykonané [74, 78].

Demonštrovali sme, ako sa dajú skonštruovať kategoriálne významy extrakciou spoločných štatistických vlastností z príkladov. Pre konštrukciu komplexnejších významov by bolo vhodné obohatiť model o analytické a syntetické operácie a nemonotónne usudzovanie [79].

Doteraz nespomenutým problémom je škálovateľnosť modelov. Tu by sme chceli upozorniť na ambiciózný projekt New Ties (angl. skratka pre „nové a emergentné modely sveta pomocou individuálneho, evolučného a sociálneho učenia“, <http://www.new-ties.org>). Cieľom projektu je vyvinúť veľké umelé spoločenstvo (rádovo tisíce agentov) schopné explorácie a chápania svojho prostredia pomocou kooperácie a interakcie [33, 94]. Rovnaké mechanizmy koordinácie jazyka a významov tak otvárajú dvere pre skúmanie zmiešaných ľudsko-strojových spoločenstiev.

*Pod'akovanie:* Táto kapitola vznikla za podpory grantovej agentúry VEGA v rámci grantovej úlohy 1/0361/08.

## Literatúra

- [1] Bergen, B., Chang, N.: Embodied construction grammar in simulation-based language understanding. In Ostman, J. O., Fried, M. (Eds.): *Construction grammar(s): Cognitive and cross-language dimensions*. Amsterdam, Netherlands: Johns Benjamins, 2003, pp. 147–190.
- [2] Bloom, P.: *How children learn the meanings of words*. Cambridge, MA: MIT Press 2000.
- [3] Blum, M., Williams, R., Juba, B., Humphrey, M.: *Toward a high-level definition of consciousness*. Invited talk presented at the Annual IEEE Computational Complexity Conference, San Jose, CA, June 2005.
- [4] Bodík, P., Takáč, M.: Formation of a common spatial lexicon and its change in a community of moving agents. In B. Tessem, P. Ala-Siuru, P. Doherty, B. Mayoh

- (Eds.): *Frontiers in AI: Proceedings of the Eighth Scandinavian Conference on Artificial Intelligence SCAI'03*. Amsterdam: IOS Press, 2003, pp. 37–46.
- [5] Briscoe, T. (Ed.): *Linguistic evolution through language acquisition: Formal and computational models*. Cambridge, U. K.: Cambridge University Press 2001.
- [6] Brooks, R. A.: Elephants don't play chess. *Robotics and Autonomous Systems* **6** (1990) 3–15.
- [7] Brooks, R. A.: Intelligence without reason. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence*. San Mateo, CA: Morgan Kaufmann, 1991, pp. 569–595.
- [8] Brooks, R. A.: Intelligence without representation. *Artificial Intelligence* **47** (1991) 139–159.
- [9] Bullinaria, J. A., Levy, J. P.: Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods* **39** (2007) 510–526.
- [10] Cangelosi, A.: Modeling the evolution of communication: From stimulus associations to grounded symbolic associations. In D. Floreano, J. Nicoud, F. Mondada (Eds.): *Proceedings of the 5th European Conference on Advances in Artificial Life*. Berlin, Germany: Springer, 1999, pp. 654–663.
- [11] Cangelosi, A.: Approaches to grounding symbols in perceptual and sensorimotor categories. In H. Cohen, C. Lefebvre (Eds.): *Handbook of categorization in cognitive science*. Amsterdam, Netherlands: Elsevier, 2005, pp. 719–737.
- [12] Cangelosi, A.: The grounding and sharing of symbols. *Pragmatics and Cognition* **14** (2006) 275–285.
- [13] Cangelosi, A., Parisi, D.: How nouns and verbs differentially affect the behavior of artificial organisms. In J. D. Moore, K. Stenning (Eds.): *Proceedings of the Twenty-third Annual Conference of the Cognitive Science Society*. London, U. K.: Lawrence Erlbaum Associates, 2001, pp. 170–175.
- [14] Carpenter, R.: *Jabberwacky – live chatbot*. Retrieved December 8, 2007, from <http://www.jabberwacky.com>
- [15] Carnap, R.: *Meaning and necessity: a study in semantics and modal logic*. Chicago, IL: University of Chicago Press 1947/1956.
- [16] Chandler, D.: *Semiotics: the basics* (Second ed.). London, New York: Routledge.
- [17] Clark, E.: The principle of contrast: A constraint on language acquisition. In B. MacWhinney (Ed.): *Mechanisms of language acquisition*. Hillsdale, NJ: Lawrence Erlbaum Associates, 1987, pp. 1–33
- [18] Cohen, P.: Dynamic maps as representations of verbs. In *Proceedings of the 13th Biennial European Conference on Artificial Intelligence*. New York, NY: John Wiley Sons, 1998, pp. 145–149.
- [19] Cohen, P., Oates, T., Atkin, M., Beal, C.: Building a baby. In G. W. Cottrell (Ed.): *Proceedings of the Eighteenth Annual Conference of the Cognitive Science Society*. Mahwah, NJ: Lawrence Erlbaum Associates, 1996, pp. 518–522.

- [20] Deacon, T. W.: *The symbolic species: The co-evolution of language and the brain*. New York, NY: W.W. Norton Co 1997.
- [21] Fauconnier, G.: *Mental spaces: Aspects of meaning construction in natural language*. Cambridge, MA: MIT Press 1985.
- [22] Fikes, R., Nilsson, N.: *STRIPS: A new approach to the application of theorem proving to problem solving* (Tech. Rep. No. 43r). Menlo Park, CA: AI Center, SRI International 1971.
- [23] Fillmore, C. J.: Frame semantics. In *Linguistics in the morning calm* (pp. 111–137). Seoul: Hanshin Pub. Co. 1982.
- [24] Fodor, J. A.: *Representations: Philosophical essays on the foundations of cognitive science*. Cambridge, MA: MIT Press 1981.
- [25] Fodor, J. A.: *The modularity of mind*. Bradford Books. Cambridge, MA: MIT Press 1983.
- [26] Gärdenfors, P.: Cued and detached representations in animal cognition. *Behavioral Processes* **35** (1996) 263–273.
- [27] Gärdenfors, P.: Language and the evolution of cognition. In V. Rialle, D. Fiset (Eds.): *Penser l'esprit: Des sciences de la cognition à une philosophie cognitive*. Grenoble: Presses Universitaires de Grenoble, 1996, pp. 151–172.
- [28] Gärdenfors, P.: *Conceptual spaces*. Cambridge, MA: MIT Press 2000.
- [29] Gärdenfors, P.: Cooperation and the evolution of symbolic communication. In K. Oller, U. Griebel (Eds.): *The evolution of communication systems*. Cambridge, MA: MIT Press, 2004, pp. 237–256.
- [30] Gelder, T. J.: Dynamic approaches to cognition. In R. A. Wilson, F. C. Keil (Eds.): *The MIT encyclopedia of the cognitive sciences*. Cambridge, MA: MIT Press, 1999, pp. 244–246.
- [31] Gibbs, R. W.: *Embodiment and cognitive science*. Cambridge, U. K.: Cambridge University Press 2006.
- [32] Gibson, J. J.: *The ecological approach to visual perception*. Boston: Houghton Mifflin 1979.
- [33] Gilbert, N., Besten, M. den, Bontovics, A., Craenen, B. G. W., Divina, F., Eiben, A. E., Griffioen, R., Hévízi, G., Lőrincz, A., Paechter, B., Schuster, S., Schut, M. C., Tzolov, C., Vogt, P., Yang, L.: Emerging Artificial Societies Through Learning [Electronic version]. *Journal of Artificial Societies and Social Simulation* **9** (2006) 9.
- [34] Ginsberg, M. (Ed.): *Readings in nonmonotonic reasoning*, San Mateo, CA: Morgan Kaufmann 1987.
- [35] Goodwin, B. C.: A cognitive view of biological process. *Journal of Social and Biological Structures* **1** (1978) 117–125.

- [36] Gulick, R. van: Consciousness, intrinsic intentionality and self-understanding machines. In A. J. Marcel, E. Bisiach (Eds.): *Consciousness in contemporary science*. Oxford, U. K.: Clarendon Press, 1988, pp. 78–100.
- [37] Harnad, S.: The symbol grounding problem. *Physica D* **42** (1990) 335–346.
- [38] Holland, O. (Ed.): Machine Consciousness [Special Issue]. *Journal of Consciousness Studies* **10** (4–5) (2003).
- [39] Icogno: *What AI techniques does Jabberwacky use?* Retrieved on December 8, 2007, from [http://www.icogno.com/what\\_ai\\_techniques.html](http://www.icogno.com/what_ai_techniques.html)
- [40] Jong, E. de, Vogt, P.: How should a robot discriminate between objects? A comparison between two methods. In *Proceedings of the Fifth International Conference on Simulation of Adaptive Behavior SAB'98*. Cambridge, MA: MIT Press, 1998, pp. 86–91.
- [41] Kegl, J., Senghas, A., Coppola, M.: Creation through contact: Sign language emergence and sign language change in Nicaragua. In M. DeGraff (Ed.): *Language creation and language change: creolization, diachrony, and development*. Cambridge, MA: MIT Press, 1999.
- [42] Kelemen, J.: The agent paradigm. *Computing and Informatics* **22** (2003) 513–519.
- [43] Kirby, S., Hurford, J.: The emergence of linguistic structure: an overview of the iterated learning model. In D. Parisi, A. Cangelosi (Eds.): *Computational approaches to the evolution of language and communication*. Berlin, Germany: Springer-Verlag, 2001, pp. 121–148.
- [44] Kováč, L.: Fundamental principles of cognitive biology. *Evolution and Cognition* **6** (2000) 51–69.
- [45] Kováč, L.: Ľudské vedomie je produktom evolučnej eskalácie emocionálneho výberu [Human consciousness is a product of evolutionary escalation of emotional selection]. In J. Kelemen (Ed.): *Kognície a umělý život III*. Opava, Czech Republic: Slezská univerzita, 2003, pp. 75–93.
- [46] Kripke, S. A.: A completeness theorem in modal logic. *Journal of Symbolic Logic* **24** (1959) 1–15.
- [47] Lakoff, G.: *Women, fire, and dangerous things: What categories reveal about the mind*. Chicago, IL: University of Chicago Press 1987.
- [48] Lakoff, G., Johnson, M.: *Metaphors we live by*. Chicago, IL: University of Chicago Press 1980.
- [49] Langacker, R. W.: *Foundations of cognitive grammar: Theoretical prerequisites* (Vol. 1). Stanford, CA: Stanford University Press 1987.
- [50] Langacker, R. W.: *Foundations of cognitive grammar: Descriptive applications* (Vol. 2). Stanford, CA: Stanford University Press 1991.
- [51] Langton, C. G. (Ed.): *Artificial life*. Reading, MA: Addison-Wesley 1989.
- [52] Li, P., Farkaš, I., MacWhinney, B.: Early lexical acquisition in a self-organizing neural network. *Neural Networks* **17** (2004) 1345–1362.



- [53] Maturana, H. R., Varela, F. J.: *The tree of knowledge: The biological roots of human understanding*. Boston, MA: Shambhala 1987.
- [54] Mirolli, M., Parisi, D.: Language as an aid to categorization: A neural network model of early language acquisition. In *Modelling language, cognition and action: Proceedings of the 9th Neural Computation and Psychology Workshop*. Singapore: World Scientific, 2005.
- [55] Montague, R.: *Formal philosophy: Selected papers of Richard Montague* (R. Thomason, Ed.). New Haven, CT: Yale University Press 1974.
- [56] Nehaniv, C.: The making of meaning in societies: Semiotic and information-theoretic background to the evolution of communication. In B. Edmonds, K. Dautenhahn (Eds.): *AISB Symposium: Starting from Society – the application of social analogies to computational systems*. Society for the Study of Artificial Intelligence and Adaptive Behaviour, 2000, pp. 73–84.
- [57] Nilsson, N. J.: *Shakey the robot* (Tech. Rep. No. 323). Menlo Park, CA: AI Center, SRI International 1984.
- [58] Oliphant, M.: *Formal approaches to innate and learned communication: Laying the foundation for language*. Unpublished doctoral dissertation, University of California, San Diego, CA 1997.
- [59] Peirce, C. S.: *Collected writings* (Hartshorne, C., Weiss, P., Burks, A. W., Eds.). Cambridge, MA: Harvard University Press 1931-58.
- [60] Pfeifer, R., Scheier, C.: *Understanding intelligence*. Cambridge, MA: MIT Press 1999.
- [61] Piaget, J.: *The child's construction of reality*. London: Routledge and Kegan Paul 1937/1955.
- [62] Piaget, J., Inhelder, B.: *La psychologie de l'enfant* [The psychology of the child]. Paris: PUF 1966.
- [63] Pulvermüller, F.: Words in the brain's language. *Behavioral and Brain Sciences* **22** (1999) 253–279.
- [64] Rizzolatti, G., et al.: Premotor cortex and the recognition of motor actions. *Cognitive Brain Research* **3** (1996) 131–141.
- [65] Rosch, E.: Principles of categorization. In E. Rosch, B. Lloyd (Eds.): *Cognition and categorization*. Hillsdale, NJ: Lawrence Erlbaum Associates, 1978, pp. 27–48.
- [66] Roy, D.: Semiotic schemas: a framework for grounding language in action and perception. *Artificial Intelligence* **167** (2005) 170–205.
- [67] Searle, J. R.: Minds, brains, and programs. *Behavioural and Brain Sciences* **3** (1980) 417–457.
- [68] Siskind, J. M.: A computational study of cross-situational techniques for learning word-to-meaning mappings. *Cognition* **61** (1996) 1–38.

- [69] Smith, A. D. M.: *Evolving communication through the inference of meaning*. Unpublished doctoral dissertation, Theoretical and Applied Linguistics, School of Philosophy, Psychology and Language Sciences, The University of Edinburgh 2003.
- [70] Smith, A. D. M.: The inferential transmission of language. *Adaptive Behavior* 13 (2005) 311–324.
- [71] Steels, L.: Constructing and sharing perceptual distinctions. In M. van Someren, G. Widmer (Eds.): *Proceedings of the European Conference on Machine Learning*. Berlin, Germany: Springer, 1997, pp. 4–13.
- [72] Steels, L.: *The Talking Heads experiment. Words and meanings*. (Vol. 1). Antwerpen: Laboratorium 1999.
- [73] Steels, L.: Language as a complex adaptive system. In M. Schoenauer (Ed.): *Proceedings of PPSN-VI*. Berlin, Germany: Springer, 2000, pp. 17–26.
- [74] Steels, L.: Constructivist development of grounded construction grammars. In D. Scott, W. Daelemans, M. Walker (Eds.): *Proceedings Annual Meeting Association for Computational Linguistic Conference*. Barcelona, 2004, pp. 9–19.
- [75] Steels, L., Kaplan, F.: Situated grounded word semantics. In T. Dean (Ed.): *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*. San Francisco, CA: Morgan Kaufmann, 1999, pp. 862–867.
- [76] Steels, L., Kaplan, F., McIntyre, A., Looveren, J. V.: Crucial factors in the origins of word-meaning. In A. Wray (Ed.): *The transition to language*. Oxford, U. K.: Oxford University Press, 2002, pp. 252–271.
- [77] Steels, L., Vogt, P.: Grounding adaptive language games in robotic agents. In I. Harvey, P. Husbands (Eds.): *Advances in artificial life. Proceedings of the Fourth European Conference on Artificial Life*. Cambridge, MA: MIT Press, 1997, pp. 474–482.
- [78] Steels, L., Wellens, P.: How grammar emerges to dampen combinatorial search in parsing. In P. Vogt, Y. Sugita, E. Tuci, C. Nehaniv (Eds.): *Symbol Grounding and Beyond: Proceedings of the Third International Workshop on the Emergence and Evolution of Linguistic Communication*. Berlin/Heidelberg, Germany: Springer, 2006, pp. 76–88.
- [79] Šefránek, J.: Kognícia bez mentálnych procesov [Cognition without mental processes]. In J. Rybár, L. Beňušková, V. Kvasnička (Eds.): *Kognitívne vedy*. Bratislava: Kalligram, 2002, pp. 200–256.
- [80] Šefránek, J., Takáč, M., Farkaš, I.: Vznik inteligencie v umelých systémoch. In D. Magdolen (Ed.): *Hmota, život, inteligencia: Vznik*. Bratislava: VEDA, 2008, pp. 245–270.
- [81] Takáč, M.: Categorization by sensory-motor interaction in artificial agents. In D. Fum, F. Del Missier, A. Stocco (Eds.): *Proceedings of the 7th International Conference on Cognitive Modeling*. Trieste, Italy: Edizioni Goliardiche, 2006, pp. 310–315.

- [82] Takáč, M.: Cognitive semantics for dynamic environments. In P. Hitzler, H. Schärfe, P. Øhrstrøm (Eds.): *Contributions to ICCS 2006 – 14th International Conference on Conceptual Structures*. Aalborg, Denmark: Aalborg University Press, 2006, pp. 202–215.
- [83] Takáč, M.: Konštrukcia významov a jej dynamika v procese iterovaného učenia [Construction of meanings and its dynamics in the iterated learning process]. In J. Kelemen, V. Kvasnička, J. Pospíchal (Eds.): *Kognice a umělý život VII*. Opava: Slezská univerzita, 2007, pp. 341–347.
- [84] Takáč, M.: *Construction of meanings in living and artificial agents*. Unpublished doctoral dissertation, Comenius University of Bratislava, Slovakia 2007.
- [85] Takáč, M.: Autonomous construction of ecologically and socially relevant semantics. *Cognitive Systems Research* **9** (2008) 293–311.
- [86] Talmy, L.: *Toward a cognitive semantics*. Cambridge, MA: MIT Press 2000.
- [87] Tarski, A.: Pojecie prawdy w jezykach nauk dedukcyjnych [The concept of truth in the languages of the deductive sciences]. *Prace Towarzystwa Naukowego Warszawskiego, Wydział III Nauk Matematyczno-Fizycznych* **34** (1933) 13–172.
- [88] Tomasello, M.: *First verbs: A case study of early grammatical development*. Cambridge: Cambridge University Press 1992.
- [89] Tschacher, W., Dauwalder, J.-P. (Eds.): *The dynamical systems approach to cognition: Concepts and empirical paradigms based on self-organization, embodiment, and coordination dynamics* (Vol. 10). Singapore: World Scientific, 1999.
- [90] Turing, A. M.: Computing machinery and intelligence. *Mind* **59** (1950) 433–460.
- [91] Vogt, P.: *Lexicon grounding on mobile robots*. Unpublished doctoral dissertation, Vrije Universiteit Brussel, Belgium 2000.
- [92] Vogt, P.: The physical symbol grounding problem. *Cognitive Systems Research* **3** (2002) 429–457.
- [93] Vogt, P.: The emergence of compositional structures in perceptually grounded language games. *Artificial Intelligence* **167** (2005) 206–242.
- [94] Vogt, P., Divina, F.: Language evolution in large populations of autonomous agents: issues in scaling. In *Proceedings of AISB 2005: Social Intelligence and Interaction in Animals, Robots and Agents*. 2005, pp. 80–87.
- [95] Vogt, P., Divina, F.: Social symbol grounding and language evolution. *Interaction Studies* **8** (2007) 31–52.
- [96] Vygotsky, L. S.: *Mind in society: The development of higher mental processes* (Rice, E., Ed. Trans.). Cambridge, MA: Harvard University Press 1930-35/1978.
- [97] Waxman, S. R.: Everything had a name, and each name gave birth to a new thought: Links between early word-learning and conceptual organization. In D. G. Hall, S. R. Waxman (Eds.): *Weaving a lexicon*. Cambridge, MA: MIT Press, 2004, pp. 295–335.

- [98] Weizenbaum, J.: Eliza – a computer program for the study of natural language communication between man and machine. *Communications of the ACM* **9** (1966) 36–45.
- [99] Whorf, B. L.: *Language, thought and reality: Selected writings of Benjamin Lee Whorf* (J. B. Carrol, Ed.). Cambridge, MA: MIT Press 1956.
- [100] Winograd, T.: *Procedures as a representation for data in a computer program for understanding natural language*. Unpublished doctoral dissertation, MIT, Cambridge, MA 1971.
- [101] Wittgenstein, L.: *Philosophical investigations*. New York: Macmillan 1953.

# Technologie audiovizuální syntézy řeči a její aplikace

Miloš ŽELEZNÝ<sup>1</sup>

**Abstrakt.** Příspěvek představuje technologii audiovizuální syntézy řeči. Podává výklad tohoto pojmu a definuje úlohu audiovizuální syntézy řeči jako takovou. V dalších částech pak uvádí požadavky kladené na datové korpusy a zkušenosti s jejich sběrem, představuje vlastní technologii audiovizuální syntézy řeči a technologii trojrozměrného snímání modelů hlavy nebo tváře používanou při vytváření umělých modelů na základě skutečných osob a konečně na ukázkách aplikací těchto technologií ilustruje jejich použití v reálných úlohách. V závěru pak shrnuje aktuální stav výzkumu, zhodnocuje dosahovanou srozumitelnost a přirozenost umělé řeči a uvádí příklady nasazení v praxi.

## 1 Úvod do audiovizuální syntézy řeči

Metody komunikace člověka se strojem se snaží stále více přibližovat komunikaci mezilidské. Kromě tradičních vstupů pomocí klávesnice a myši a grafického uživatelského rozhraní se stále častěji využívá komunikace mluvenou řečí [1]. Stále více se však v tomto oboru využívá dalších komunikačních modalit s řečí přímo či nepřímo souvisejících a využívajících kombinovaného vjemu. V této kapitole se budeme zabývat především metodami založenými na vizuálním vjemu doplňujícím akustickou řečovou komunikaci.

Již v roce 1976 prokázali McGurk a MacDonald [2] jev kombinovaného audiovizuálního vjemu, když posluchačům předložili záměrně zkombinovaný rozdílný vizuální a akustický záznam. Akustická složka představovala slabiku „ba“ a vizuální složka slabiku „ga“. Posluchači však často vnímali tento kombinovaný vjem jako slabiku „da“. Tento efekt je dostatečně robustní, čili přestože o něm víme, vjem to nezmění, na rozdíl například od určitých zrakových klamů, které, jakmile jednou prohlédneme, už přestanou být klamem. Zároveň výsledky tohoto a navazujících

---

<sup>1</sup> Katedra kybernetiky, Fakulta aplikovaných věd, Západočeská univerzita v Plzni, Univerzitní 8, 306 14 Plzeň, E-mail: zelezny@kky.zcu.cz

výzkumů prokazují, že člověk vnímá řeč jako kombinaci obou složek, vizuální a akustické.

Obě složky, akustická a vizuální se tedy vzájemně doplňují, aniž bychom si to za normálních okolností uvědomovali. Stačí ovšem zvýšený hluk prostředí, přenosový kanál se sníženou kvalitou nebo sluchová vada, a zjistíme, že akustická složka řeči není dostačující pro kvalitní porozumění. Se snižující se kvalitou vnímané akustické složky se zvyšuje význam vizuální složky pro lepší porozumění. Toho si musíme být vědomi při mezilidské komunikaci v rušném prostředí nebo při hovoru s osobou se zbytky sluchu. Také si toho musíme být vědomi při návrhu komunikace člověk-stroj.

Jednou z komponent systému komunikace člověk-stroj využívající komunikaci mluvenou řečí je modul řečového výstupu neboli syntézy řeči. V dnešní době takový modul pracuje většinou tak, že převádí libovolný text v daném jazyce na řečový signál. Pokud vezmeme v úvahu dosud prezentovaná fakta o významu vizuální složky, musíme pro její použití v komunikačním systému navrhnout modul syntézy řeči tak, aby poskytoval žádaný audiovizuální vjem. Výsledkem je tedy kombinovaná audiovizuální syntéza řeči (AVS).

Audiovizuální syntézou řeči [3] tedy rozumíme doplnění akustické syntézy řeči o vizuální složku, tj. animovaný 2D nebo 3D model lidské hlavy nebo tváře napodobující promlouvání řeči člověkem, často zjednodušeně nazývaný mluvící hlava. Nejjednodušším a také zpočátku nejčastěji využívaným způsobem vytvoření mluvčí hlavy je spojování dvojrozměrných klíčových snímků představujících jednotlivé vizémy (vizuální podoby hlásek) do syntetizovaných videí a interpolace mezi těmito klíčovými snímky. Složitějším, ale realističtějším a v poslední době častěji používaným způsobem vytvoření mluvčí hlavy je vytvoření kompletního trojrozměrného modelu hlavy a jeho animace pomocí řídicích bodů. Řídicí body jsou řízeny nějakou vhodnou parametrizací, která je opět pospojována z databáze vizémů a interpolována mezi jednotlivými vizémy.

V našem případě jsme se rozhodli právě pro realistickou parametricky řízenou animaci trojrozměrného modelu hlavy. Základem je tedy statický trojrozměrný model hlavy. Větší realističnosti dosáhneme, pokud je model získán snímáním hlavy skutečné osoby. Na takto vytvořeném modelu jsou pak stanoveny řídicí body, pomocí nichž ovlivňujeme tvar modelu v průběhu animace. Tyto body se typicky umísťují na konturu rtů, bradu, nos, tváře či obočí. Přitom je třeba vyřešit problém koartikulace, jevu, který je způsoben vzájemným ovlivněním sousedních vizémů.

Samotná akustická syntéza řeči probíhá podobným způsobem spojováním krátkých úseků řeči. Přehrávání celých vět namluvených člověkem nebudeme označovat za syntézu řeči. Vytváření různých textů je omezeno na namluvený korpus vět. Za nejjednodušší formu syntézy použitelnou pro velmi jednoduché aplikace lze označit syntézu spojováním slov, tak jak je používána na českých železničních stanicích. Korpus, který obsahuje všechna slova potřebná k vyjádření čísel vlaků a názvů stanic však nebyl namluven monotónně, a proto na přechodech slov vznikají rušivé jevy, které snižují přirozenost syntetizovaného textu. Tímto způsobem by ale pro skutečnou syntézu z textu bylo nutno namlouvat obrovské korpusy obsahující všechna slova. Z toho důvodu, tj. z důvodu nutnosti nahrát co nejmenší korpus pro syntézu řeči z libovolného textu, se při syntéze řeči používá spojování jednotek na

úrovni hlásek. Při spojování je pak třeba zajistit, aby výsledná syntetizovaná řeč vykazovala co nejméně poruch na přechodech jednotlivých hlásek. Vyšší přirozenosti akustické syntézy řeči je dosahováno modelováním prozodie, tedy melodie řeči.

Vzhledem k tomu, že akustická a vizuální syntéza jsou realizovány dvěma nezávislými moduly, je třeba je vzájemně synchronizovat. Při off-line syntéze postačí vytvořit celou promluvu nejprve akusticky, a poté vytvořit vizuální promluvu, tedy videozáznam, který je synchronizován poskytnutými tzv. notifikačními informacemi od modulu akustické syntézy. Při relativně krátkých promluvách je u aplikace audiovizuální syntézy akceptovatelný kombinovaný režim, kdy akustická syntéza je provedena off-line, a současně s přehráváním výsledného zvukového souboru je renderována (vykreslována) vizuální syntéza on-line. Pro skutečný online režim je však třeba zajistit on-line synchronizaci obou datových proudů.

Jiným režimem audiovizuální syntézy je režim, kdy vizuální složka je doplňována k existující akustické (většinou přirozené) řeči, kdy není textový přepis k dispozici. V takovém případě se používají dva režimy, buď s rozpoznáváním textu ze vstupní promluvy a poté generováním vizuální promluvy z takto získaného textu, nebo přímým mapováním akustických příznaků po nezbytném předzpracování přímo na příznaky vizuální. Obzvláště v prvním případě ale dochází ke zpoždění, takže akustický signál je třeba zobrazovat zpožděný, tak, aby byl synchronní s vykreslovanou vizuální syntézou řeči.

Výsledná audiovizuální syntéza je většinou zobrazována na monitoru počítače (nebo televizní obrazovce, příp. datovém projektoru) během jejího generování. V tom případě je nutný dostatečný výpočetní výkon pro vykreslování měnícího se 3D modelu v reálném čase. Přestože dnešní výpočetní technika nemá problémy s takovým zobrazením, je třeba mít na paměti, že syntéza coby komunikační prostředek není hlavním běžícím procesem. Z toho důvodu je potřeba navrhnout kvalitu vykreslovaného modelu tak, aby sice poskytoval žádanou přirozenost, ale aby nadbytečným počtem vrcholů sítě nezahlcovoval výpočetní techniku při generování syntézy a vykreslování výsledného 3D modelu. Proto se po snímání modelu provádí redukce dat, čímž se při zachování přirozenosti modelu sníží počet vrcholů sítě tak, aby zpracování a vykreslování modelu spotřebovalo méně strojového času procesoru.

Oblast použití audiovizuální syntézy je široká, od zábavných aplikací, přes kancelářské či domácí použití pro předčítání e-mailů nebo knížek, přes aplikace, kde se skutečně využívá vizuální složka ke zvýšení porozumění, jako například v zašuměném prostředí nebo v aplikacích pro osoby se zhoršeným sluchem až po aplikaci vizuální syntézy jako doplněk syntetizéru znakové řeči pro osoby hluché, kde je pro porozumění znakové řeči současná artikulace ústy nezbytná. Z dalších aplikací můžeme uvést například využití pro zasilání syntetizovaných zpráv do mobilních telefonů nebo jako aplikaci pro telefonování osob se zhoršeným sluchem, kdy zároveň s probíhajícím hovorem je promítána na obrazovku vizuální syntéza řeči získaná přímo z přicházejícího zvukového signálu.

Vzhledem k tomu, že modely pro audiovizuální syntézu řeči jsou získávány z dat naměřených na reálných osobách, je výsledek do značné míry závislý i na kvalitě návrhu korpusu a na kvalitě jeho nahrávek. V následující sekci budou představeny datové korpusy, které jsou potřebné pro úspěšný vývoj modulu audiovizuální syntézy

řeči. Ve stěžejní sekci této kapitoly pak představíme metody pro vlastní syntézu, další sekce bude věnována technologii trojrozměrného snímání statických modelů tváře a poté budou uvedeny některé příklady aplikací a reálně dosahované výsledky.

## 2 Datové korpusy

Současné metody návrhu technologií komunikace člověka se strojem jsou založeny na statistickém zpracování velkých objemů dat. Technologie rozpoznávání i syntézy jsou založeny na trénování klasifikátorů/modelů pomocí těchto dat. Obecně platí, že čím větší je množina dat, a tím pádem čím statisticky významnější je trénovací množina, tím kvalitnější model dostaneme. Na druhou stranu pořizování dat, často ve speciálních podmínkách je časově i finančně náročné a snaha je tedy použít co nejmenší korpus, který ale zaručí dostatečně kvalitně natrénované modely. Přitom hranice velikosti dat pro dostatečnou kvalitu natrénovaných modelů se liší podle úlohy a zvoleného přístupu k řešení.

V každém případě ale z uvedených informací vyplývá potřeba získání nebo pořizování datových korpusů pro jednotlivé úlohy z oblasti zpracování multimodální informace pro komunikaci člověk-stroj. Vzhledem k tomu, že při návrhu řečových aplikací jsou data závislá na jazyce, je třeba se omezit na jazyk, se kterým pracujeme, tedy v našem případě na češtinu. Pro řadu úloh komunikace člověk-stroj jsou k dispozici mnohdy i kvalitní korpusy, bohužel jsou však k dispozici převážně pro angličtinu (v menší míře také pro ostatní evropské jazyky, jako je francouzština, španělština nebo němčina či jazyky asijské, jako je čínština nebo japonština) a pro naše potřeby jsou nevyužitelné. To klade na vývojáře těchto technologií nároky pořídit vhodný korpus, většinou vlastními silami za pomoci dobrovolníků.

Přitom návrh a samotné pořízení jazykového korpusu není triviální záležitost. Jedním faktorem, na který je třeba se zaměřit, je splnění podmínek nahrávání vzhledem k cíli korpusu. Pro akustickou syntézu je zapotřebí nahrávat zvuková data v prostředí s co nejmenším šumem, ideálně ve zvukotěsné a bezodrazové místnosti. Dále například pro metodu s výběrem jedné reprezentativní instance každé základní řečové jednotky bude třeba nasbírat korpus monotónní řeči, zatímco pro metodu s výběrem z více instancí, kde je použita daty řízená prozodie je naopak nutný rozsáhlý korpus obsahující všechny možné prozodické varianty. Úplně jiné požadavky jsou kladeny na korpusy pro vizuální syntézu, i když i zde je základem monotónní osvětlení a kvalitní snímací podmínky.

Kromě hlediska akustické nebo vizuální kvality dat je třeba vzít v úvahu i vlastnosti jazykové. Korpusy jsou většinou čtené ze zadaného textu, který je třeba vhodně navrhnout, tak, aby obsahoval všechny jazykové/fonetické/vizuální jevy, které mají být syntetizovány. Této oblasti se věnuje celá oblast vědní disciplíny návrh korpusů. Typické zadání pro návrh korpusu zahrnuje informace jako je například jeho velikost, délka souvislých promluv (od jednotlivých slov až po celé věty) a požadavky na minimální výskyt fonémů/vizémů. Korpus je většinou navrhován jako foneticky balancovaný pro daný jazyk, tj. tak, aby statistické rozložení výskytu jednotlivých jevů (vizémů, trifónů v různých kontextech apod.) odpovídalo statistickému rozložení



těchto jevů v daném jazyce, a dále aby byl zajištěn minimální výskyt instancí pro řídké jevy. Výstupem výzkumu v této oblasti jsou například programy pro zpracování textu, které vybírají věty z velkých textových korpusů (získaných například zpracováním novinových článků) na základě statistické analýzy těchto textů. Výsledkem je potom text pro danou úlohu pro zadaný počet mluvčích, který je čten během nahrávání.

Kromě vlastních multimodálních dat (akustických, vizuálních, spektrálních popisů, předzpracovaných obrazových dat) musí korpus obsahovat informaci o základních řečových jednotkách. Korpus tedy musí být segmentován a anotován. Segmentace je většinou prováděna automaticky pomocí Viterbiho algoritmu na základě zpracování akustické nebo vizuální informace. Anotace je provedena na základě textu, který byl pro korpus vygenerován a předložen k přečtení. Vlivem lidského faktoru a chyb při zpracování dat je ale třeba celý korpus zkontrolovat a případně opravit, což je zdoluhavá rutinní práce. Výsledkem je označení každého úseku řeči v korpusu, který odpovídá jedné základní řečové jednotce a umožňuje tak jednoznačnou odvolávku na tuto jednotku, tak aby mohla být snadno vybrána v průběhu trénování modelu či vlastní syntézy.

## 2.1 Korpus pro akustickou syntézu řeči

Korpus pro akustickou část syntézy řeči je co do délky nahrávky nejobsáhlejší. Velikost korpusu je 5000 foneticky balancovaných vět, což činí přibližně 13 hodin záznamu řeči. Záznam byl získán od jedné osoby, neboť cílem je zde navrhnout syntézu řeči tak, aby kopírovala hlasové charakteristiky této osoby. Hlasové charakteristiky se tedy během nahrávání nesmí měnit, což klade nároky především na nahrávanou osobu. Například nachlazení, které by změnilo nahrávané osobě hlas, by znehodnotilo celý korpus. Požadavek na neměnné charakteristiky se vztahuje i na základní hlasivkový tón, čili korpus musí být namlouván maximálně monotónně. V našem případě byli najati profesionální mluvčí (konkrétně moderátoři rozhlasových stanic).

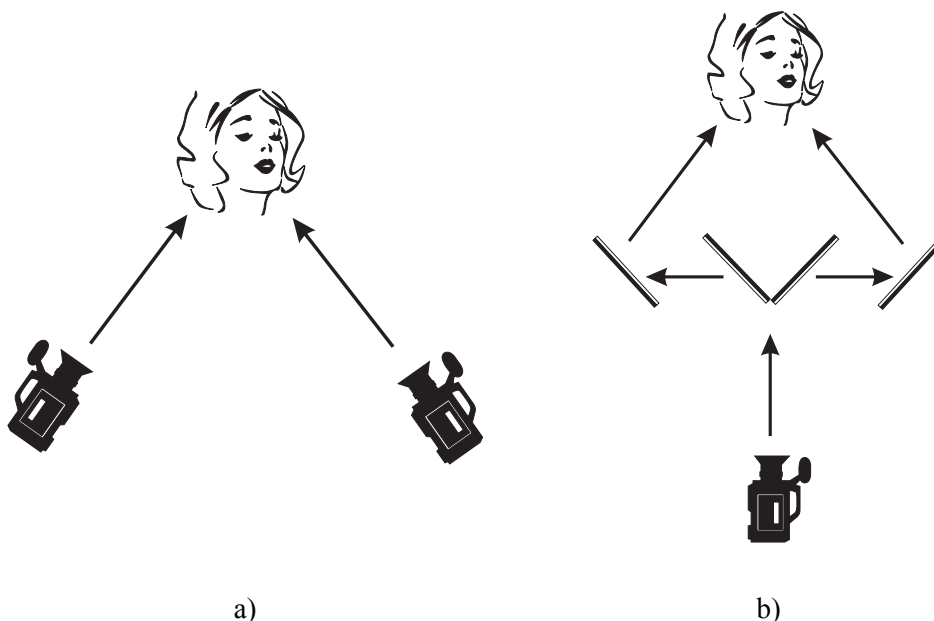
Každá věta je v korpusu zaznamenána jak v lingvistické podobě řeči, tak v podobě akustického signálu. Kromě vlastního akustického signálu z mikrofonu je zaznamenáván speciální signál popisující aktivitu hlasivek z laryngografu. Tento signál jednak umožňuje rozlišit znělé a neznělé zvuky, a jednak umožňuje přesně určit jednotlivé periody znělých zvuků. Jako lingvistický popis jsou použity ortografická a fonetická transkripce věty. Řečové signály jsou reprezentovány jak v časové oblasti, tj. ve zvukových souborech typu .wav, tak ve spektrální oblasti. Spektrální vlastnosti signálu jsou vyjádřeny pomocí Melovských frekvenčních spektrálních koeficientů (MFCC), které jsou počítány v oknech širokých 20 ms s posunem 4 ms. Systém používá vektor, který obsahuje 12 MFCC koeficientů + normalizovanou energii spolu s první, druhou a třetí diferencí (čili 52 koeficientů dohromady).

## 2.2 Korpusy pro vizuální syntézu řeči

Prvotním cílem pro návrh korpusu pro vizuální syntézu řeči je získání dat pro vytvoření modelu tvaru rtů (potažmo celé tváře či hlavy) tak, aby tyto bylo možno

použít pro zpětné generování vizuální podoby řeči. Přitom je žádoucí, aby tento model v co největší míře věrně kopíroval skutečnou osobu (ne nutně, ale často jednu konkrétní) a aby tato data bylo možno získat pokud možno automaticky. V prvním návrhu jsme se zaměřili na získání 3D modelu rtů a použití takových vhodných řídicích bodů, aby tento model bylo možno ovládat a animovat podle potřeb vizuální syntézy řeči.

Prvním krokem je návrh snímání tváře subjektu tak, aby bylo možno získat 3D charakteristiky. Uvažujme tak, že pro potřeby modelu pohybujících se rtů stačí snímat čelní obličejovou část. Pro získání 3D informace je pak možno využít princip stereovidění, čili složení pohledů dvou kamer z různých úhlů pohledu. Protože zaznamenáváme dynamická data pohybu rtů, je třeba, aby snímky obou kamer byly snímány synchronně, tj. se stejnou vzorkovací frekvencí a vždy ve stejných okamžicích. To je bez speciální techniky využívající externího synchronizačního zdroje a schopnosti dvou kamer být řízeny tímto synchronizačním zdrojem. S klasickou spotřební technikou je toto řešení nedosažitelné, neboť i když obě kamery poběží se stejnou vzorkovací frekvencí, pravděpodobnost, že budou snímat oba snímky ve stejný okamžik, je velice malá, naopak hrozí zde rozdíl v sejmutí odpovídajících snímků rovný až polovině vzorkovací periody.



**Obr. 1.** Ilustrace principu stereovidění: a) původní myšlenka principu stereovidění, b) upravený návrh pro odstranění problémů se synchronizací dvou kamer, využívající systému 4 zrcadel.

Stupněm k vyřešení problému synchronizace dvou kamer bylo uvědomění si, že poměr stran obdélníku opisujícího objekt zájmu, tedy hlavu, je spíše „na výšku“,

zatímco kamera snímá data spíše v pozici „na šířku“, a že pokud by se podařilo dostat do jednoho snímku kamery pohledy z obou směrů, nesnížilo by to významně rozlišení obrazu nutné pro co nejpřesnější model. Odtud byl již krůček k návrhu použití snímacího systému používajícího jednu kameru a systém zrcadel. Ilustrace systému stereovidění i úprava pro systém 4 zrcadel, který převádí pohledy na snímáný subjekt ze dvou stran do jednoho obrázku kamery a je používán i v dalších částech snímacího procesu jsou znázorněny na Obr. 1.

Druhým krokem je zpracování takto získaných dat. Pro výpočet trojrozměrné informace o tvaru rtů je zapotřebí získat korespondující dvojice bodů z obou polovin snímku. Toho je možné dosáhnout různými způsoby. V prvním korpusu nasbíraném pro vizuální syntézu byl subjekt snímán za běžného denního osvětlení. Oblast rtů byla detekována metodami počítačového vidění, pro zvětšení rozdílu barev rtů a okolních částí kůže byla použita modrá rtěnka. Po předzpracování byly detekovány řídicí body pro syntézu. Jako základní dva body byly detekovány levý a pravý vnější koutek rtů. Jako další byly detekovány horní a dolní okraj rtů na ose úsečky mezi prvními dvěma body a další dvě dvojice opět na přímkách mezi krajními body a osou. Tímto způsobem jsme dostali 8 řídicích bodů pro vnější konturu rtů. Dále byly získány dva body na středové ose, determinující výšku rtů a výšku otevřených úst a jeden bod na středu brady, který udává pohyb celé dolní čelisti. Pohyb těchto všech bodů je snímán ve vztahu k referenčnímu bodu, který je umístěn mezi očima a který kompenzuje pohyby celé hlavy. Rozmístění řídicích bodů prvního korpusu je ilustrováno na Obr. 2.



**Obr. 2.** Zobrazení rozmístění řídicích bodů prvního korpusu pro audiovizuální syntézu řeči. Body 1–4, 7, 9–11 reprezentují vnější konturu rtů, body 5, 6 udávají výšku otevřených úst a výšku horního a dolního rtu, bod 8 umístěný na středu brady reprezentuje pohyb celé dolní čelisti. R je referenční bod, který kompenzuje pohyby celé hlavy.

Vizuální syntéza řeči natrénovaná na tomto korpusu byla první funkční verzí vizuální syntézy. Později se ukázalo, že tento způsob získávání inventáře souřadnic řídicích bodů pro základní řečové jednotky není příliš přesný, jako koutek rtů byly totiž díky rozdílnému úhlu pohledu detekovány ve dvou pohledech různé body a navíc

středová osa byla detekována v každé polovině snímku jinak, díky vyšpulenému tvaru rtů ležel střed vždy poněkud stranou.

Proto při potřebě nahrát data dalšího mluvčího byl použit odlišný postup snímání. Systém zrcadel zůstal zachován, nově byly na tvář snímaného subjektu umístěny body ve tvaru malých kroužků vyrobených z reflexivního materiálu (používaného např. na dopravní značky). Tyto body byly umístěny do míst řídicích bodů na skutečné tváři. Snímání bylo prováděno v zatemněné místnosti s minimem denního (ale i umělého osvětlení) nutného jen pro organizaci nahrávání. Snímaná osoba byla navíc osvětlena infračerveným zdrojem světla, který je součástí kamery (používá se pro tzv. noční režim). Tento způsob snímání řídicích bodů i při použití jednoduchých algoritmů předzpracování obrazových dat dává velmi dobré a spolehlivé výsledky.

Třetí a zatím poslední nahraný korpus byl motivován cílem poskytnout co nejuvěrnější animaci pohybů úst vhodnou pro odezírání ze rtů. Do té doby vyvinuté algoritmy byly dobře hodnoceny slyšícími uživateli, ale při testu v základní škole pro neslyšící a nedoslýchavé se ukázalo, že pro osoby na odezírání ze rtů závislé je kvalita artikulace modelu nedostačující. Při návrhu nového korpusu bylo dbáno na kvalitní artikulaci snímaného subjektu, kterým byla logopedická pracovnice, zatímco druhá její kolegyně hlídala správnou artikulaci a v případě chyby navrhovala opravu. Tento korpus je z hlediska kvality artikulace nejkvalitnější, ale vzhledem k nahrávacím podmínkám a k nutnosti sledovat správnou artikulaci druhým odborníkem nebylo použito snímání v infračervené oblasti s nalepenými reflexivními značkami a korpus bylo nutno zpracovat ručně.

Celkem tedy byly nasbírány tři řečové korpusy použité pro trénování systémů audiovizuální syntézy řeči. První korpus obsahuje záznamy jednoho amatérského mluvčího se zvýrazněnou oblastí úst, druhý korpus obsahuje několik mluvčích zaznamenaných v podmínkách infračerveného osvětlení s nalepenými reflexivními značkami a konečně třetí korpus obsahuje data profesionální mluvčí s kontrolovanou artikulací.

Dalšími zdrojovými daty jsou data pro statické 3D modely. První model, se kterým byl vytvořen modul audiovizuální syntézy řeči, byl model virtuální hlavy, uměle vytvořený v 3D modelovacím programu. Později byl stanoven cíl vytvořit model podobající se reálné osobě. Výsledek vývoje snímacího postupu pro získání 3D modelu snímáním reálného subjektu je popsán v sekci 4. Snímací algoritmus používá již popsaný systém 4 zrcadel a řeší originálním způsobem problém korespondence bodů ve stereovidění. Pro ukázkovou aplikaci bylo vytvořeno několik modelů celé hlavy. Nejjednodušší způsob snímání je prezentován na popularizačních akcích pro veřejnost, při kterých bylo nasnímáno přibližně tisíc modelů obličejové části dobrovolníků, které pak byly prezentovány ozvučené syntetickým hlasem.

### 3 Technologie syntézy

#### 3.1 Akustická syntéza řeči

Přestože v této kapitole budeme klást důraz na vizuální složku syntézy řeči, uvedme nyní princip syntézy akustické složky. Moderní systémy převodu textu na řeč (TTS – text-to-speech) jsou založeny na principu tzv. konkatenční syntézy založené na korpusu, čili syntézy využívající zřetězování základních řečových jednotek vybíraných z řečového korpusu. Systém ARTIC (ARTificial Talker In Czech) [4] pro syntézu češtiny využívá dva základní přístupy, buď výběr jediné řečové jednotky (single unit instance system), nebo výběr z mnoha jednotek (multiple unit instance system nebo unit selection system). První přístup bere při požadavku na syntézu základní řečové jednotky pro každou základní jednotku pokaždé tutéž instanci z korpusu, zatímco druhý přístup vybírá vhodnou instanci základní řečové jednotky z mnoha možností v korpusu na základě posouzení jejich vhodnosti například porovnáním kontextu apod. Tím tento druhý způsob dosahuje lepší přirozenosti syntézy.

Jako základní řečová jednotka se volí obvykle jednotka na úrovni hlásky. Lze sice volit jednotku delší (jako slabika nebo celé slovo, vždyť i systém syntetizující nádražní hlášení poskládáním celých slov lze považovat za syntézu), ale pro syntézu řeči z libovolného textu je pak třeba mít uloženo až extrémně velké množství jednotek a náročnější je pak i řešení přechodů mezi jednotlivými jednotkami, kterých je tím pádem také velmi mnoho. V případě našeho systému byl zvolen jako základní řečová jednotka trifón, tedy fón (hláska) doplněný o levý a pravý kontext. Ten může na rozdíl od tradičně používaných difónů dobře modelovat jak statickou část hlásky, tak přechodové jevy z předchozí a na následující hlásku. Trifóny také mohou být velmi dobře modelovány pomocí skrytých Markovových modelů a usnadňují synchronizaci s vizuální složkou neboť trifóny jednoznačně korespondují s vizuálními základními řečovými jednotkami, vizémy.

Pro systém založený na výběru jediné reprezentativní instance pro každou řečovou jednotku byla navržena a použita metoda automatického vytváření trifónových inventářů (podobná jako byla použita v [5]). Tato metoda, založená na pečlivě vybraném korpusu řeči, používá statistický přístup, konkrétně skryté Markovovy modely jak k natrénování modelů, tak k vytváření inventáře akustických řečových jednotek, a to plně automaticky. Součástí tohoto přístupu je použití shlukování podobných skrytých Markovových modelů pro definici sady základních řečových jednotek, trifónů. Veškerá řeč dostupná v korpusu, byla segmentována na trifóny, zarovnáním skrytých Markovových modelů pomocí Viterbiho algoritmu [6]. Poté je off-line vybrána nejvhodnější instance pro každý trifón a použita jako reprezentativní jednotka pro syntézu.

Systém založený na výběru z mnoha instancí představuje alternativu k tomuto přístupu. Je založen na možnosti výběru z více instancí pro každou řečovou jednotku a vybírá nejvhodnější řečovou jednotku on-line při provádění syntézy. Pro segmentaci korpusu na základní řečové jednotky byl použit podobný přístup založený na skrytých Markovových modelech. Ukázalo se, že pro tento přístup jsou vhodnější difóny jako

základní řečové jednotky [7]. Bylo potvrzeno, že tento přístup dosahuje lepších výsledků, než přístup založený na off-line výběru jedné reprezentativní řečové jednotky. Na druhou stranu je ale tento systém mnohem náročnější na zdroje, tedy výpočetní a paměťové nároky a je tak omezen na výkonné počítače, na rozdíl od prvního přístupu, který je realizovatelný i v přenosných zařízeních typu PDA.

Pro syntézu řeči je na začátku třeba provést předzpracování textu. V našem případě provádíme detekci větných úseků založenou na interpunkci, dále jednoduchou normalizaci textu (přepis čísel a zkratek) a detailní fonetickou transkripci pomocí pravidel. Vlastní syntéza je pak v systému založeném na výběru jedné jednotky prováděna pomocí tzv. OLA algoritmu (overlap and add) jak ve frekvenční, tak i v časové oblasti.

Dalším krokem je generování prozodie, tedy melodie řeči, která je pro přirozenou syntézu řeči nezbytná. Modelovány jsou většinou tyto parametry a jejich průběh v rámci věty: výška základního hlasivkového tónu, amplituda (hlasitost řeči) a tempo (rychlost řeči). Pro generování prozodie byly navrženy dva algoritmy, přístup založený na pravidlech a přístup řízený daty. Přístup založený na pravidlech aplikuje sadu pravidel odvozených na základě fonetického výzkumu a popisu suprasegmentálních řečových jevů. Model pomocí těchto pravidel řídí prozodické parametry jako výška základního hlasivkového tónu, sklon celkové tendence melodie větného úseku, tvar všech klesavých nebo stoupavých intonací, modulace intenzity, změny trvání jednotlivých hlásek, vliv důrazu na jednotlivá slova. Nastavení těchto pravidel bylo provedeno experimentálně.

Protože kvalita prozodie založené na pravidlech je omezená co do přirozenosti syntetizované řeči, byl navržen nový přístup řízený daty [4], [8]. Hlavní myšlenka je nastavení parametrů modelu automaticky s využitím reálných dat z korpusu. Inventář intonací byl vytvořen automaticky metodou aglomerativního shlukování do  $n$  shluků, jejichž centroidy jsou považovány za intonace reprezentující variabilitu intonačních vzorů přes prozodická slova. Experimentovali jsme s hodnotou  $n$  mezi 10 a 200. V průběhu vlastní syntézy je každému prozodickému slovu přiřazena intonace z inventáře v závislosti na jeho příznakovém vektoru zahrnujícím určité kvantitativní charakteristiky (jako počet jeho hlásek nebo slabik, pořadí přízvučné samohlásky apod.) Daty řízený model produkuje přirozenější syntetickou řeč (daty řízený model byl upřednostněn 64% posluchačů pro ženský hlas a 74% pro mužský hlas [4], [9]).

### 3.2 Vizualní syntéza řeči

Syntéza vizuální složky může být založena na několika různých principech. Jedním ze základních způsobů je vytváření videozáznamu spojováním klíčových snímků, na kterých jsou statické dvojrozměrné obrázky obsahující hlavu, jejíž tvar rtů je nastaven do polohy odpovídající jednotlivým hláskám (ve vizuálním zpracování řeči se jako základní řečová jednotka uvažuje tzv. vizém, čili vizuální podoba hlásky). Snímky mezi každými dvěma klíčovými snímky je pak nutno interpolovat, a to nikoli lineárně, ale systémem morfingu, kdy hlava na obrázku postupně otevírá/zavírá ústa či přechází z jedné polohy (vizému) do druhé). Výhodou je relativní nenáročnost algoritmu a snadné použití, na druhé straně tento přístup má celou řadu nevýhod. Předně je třeba

zajistit nahrávání korpusu v normalizovaných podmínkách, tj. snímaná osoba se nesmí pokud možno vůbec pohybovat (což je ale velmi těžké zaručit, takže snímky výsledného korpusu je třeba normalizovat do stejné základní polohy hlavy), dále musí být zajištěny všechny ostatní podmínky (osvětlení apod.) konstantní po celou dobu snímání. Hlavní nevýhody jsou však z oblasti změn, změna postavy, ale i například její natočení do trochu jiné polohy znamená nové nahrávání a kompletní zpracování celého korpusu. Dvojměrným modelem hlavy není možno nijak pohybovat (otáčet apod.) Konečně úpravy vizémů pro výrazy tváře nebo pro jiný jazyk musí vždy být obsaženy v nahrávaném korpusu (dodatečné nahrávání není možné) nebo znamenají nutnost nahrát celý korpus znovu.

Druhou skupinu metod tvoří metody založené na parametricky ovládatelném trojrozměrném (3D) modelu hlavy. Základem syntézy je trojrozměrný model hlavy, který je možné upravovat do jiných poloh pomocí sady parametrů. V našem případě je jako sada parametrů použita množina tzv. řídicích bodů modelu, což jsou zvolené vrcholy modelu, změnou jejichž souřadnic lze ovlivnit celkový vzhled modelu. 3D model hlavy je cyklicky renderován (vykreslován), přičemž změnou parametrů jsou měněny souřadnice některých jeho vrcholů. Tyto změny jsou prováděny synchronně s akustickou syntézou řeči, od které musí dostávat informace o začátku a konci jednotlivých základních řečových jednotek. Vzhledem k principu 3D zobrazování ale lze modelem libovolně pohybovat (otáčet, přibližovat, posouvat) i během běžící syntézy. Syntéza pomocí 3D modelu je sice výpočetně náročnější, než je tomu u 2D modelu, výhodou je však její mnohem větší flexibilita. Vyšší přirozenosti 3D vizuální syntézy je možno dosáhnout, pokud model bude založený na konkrétní osobě. K tomu je možno využít principů 3D snímání.

Systém vlastní syntézy je, podobně jako v akustickém případě, založen na konkatenačním principu, tj. na principu zřetězování základních řečových jednotek (v tomto případě vizémů) či spíše jejich vektorů parametrů. Databáze vektorů řídicích bodů je vytvořena pro všechny vizémy a jejich varianty ve významných kontextech (tak například vizém „m“ se velmi liší v kontextech „eme“ a „umu“). Základním úkolem modulu vlastní syntézy je tedy zřetězovat tyto vektory řídicích bodů tak, aby v okamžiku vyslovování odpovídajícího fonému nastavil model tvar nastavený právě tímto vektorem parametrů, jinými slovy aby dosáhl stanoveného artikulačního cíle. Okamžik umístění vektoru je většinou ve středu doby odpovídající fonému, která je dodána modulem akustické syntézy. Pro plynulé vykreslování animace modelu je třeba interpolovat pozici v každém okamžiku vykreslování mezi těmito artikulačními cíli. Dále je třeba vyřešit problém koartikulace, tedy přechodového jevu, který vzniká na hranicích spojovaných základních řečových jednotek (v tomto případě vizémů).

Pro řešení problému koartikulace vizuální řeči bylo navrženo několik různých přístupů. Za jeden ze základních přístupů lze považovat systém dominantních funkcí navržených Massarem a Cohenem [10]. Tento přístup používá tzv. speciální funkce pro vzájemné ovlivňování sousedních artikulačních cílů. Tyto funkce nabývají maxima v místě umístění artikulačního cíle a exponenciálně klesají na obě strany, kde ovlivňují sousední vizémy. Výhodou těchto funkcí je, že specifický vizém může ovlivňovat ostatní v případě potřeby i do vzdálenosti několika sousedních vizémů, ovlivňování přes hranice slov nevyjímaje. Velkou nevýhodou této metody je však fakt, že tyto

dominanční funkce nezaručují dosažení artikulačního cíle. Tak například v kontextu „apa“ obě sousední samohlásky ovlivňují střední vizém „p“. Toto ovlivnění však způsobí nedovření úst vizému „p“ které je však u okluzivní souhlásky zásadní. Z toho důvodu byly hledány další možnosti řešení problému koartikulace.

Navržena byla metoda využívající podobného principu jako u akustické syntézy. Jak bylo zmíněno v předchozí sekci, jednou z možností je výběr z více instancí základních řečových jednotek. U akustické syntézy to zároveň řeší problém generování prozodie. Ve vizuálním případě je podobné metody využito k vyřešení problému koartikulace. Navržená metoda vizuálního výběru jednotek (VUS – visual unit selection) používá rozsáhlejší zdrojovou databázi a při generování syntézy vybírá vhodnou instanci podobně jako akustická syntéza s pomocí rozhodovacích stromů. Otázky v rozhodovacích stromech jsou však tentokrát založeny na vizuálních vlastnostech. Výsledkem je metoda, která lépe modeluje koartikulaci a zajišťuje dosažení artikulačních cílů, které by při použití dominančních funkcí dosaženy nebyly.

Pro dosažení plynulé animace je trojrozměrný model cyklicky vykreslován na zobrazovacím zařízení počítače. Druhou možností je offline vytvoření syntetizované věty ve formě videosouboru. Model je obvykle vykreslován s takovou frekvencí, při které uživatel nevnímá rozdíly mezi jednotlivými snímky, ale plynule animovaný model. Nejčastěji je používána frekvence použitá též v televizním vysílání (tzv. pulsničková frekvence) 50 Hz.

Statický model, ať už uměle vytvořený, či nasnímaný procesem popsaným v následující kapitole, je načten ze souboru, v našem případě ve formátu VRML. Tento model je uložen v paměti a v okamžiku generování každého snímku je upraven pomocí vektorů parametrů řídicích bodů a vykreslen. Řídicí body odpovídají určitým vrcholům 3D sítě modelu. Typicky jsou těmito body koutky úst, několik bodů po obvodu kontury rtů a několik dalších. Úprava způsobí posunutí řídicích bodů tak, jak to odpovídá požadovanému tvaru rtů. Hustota vrcholů v okolí rtů je však vysoká a počet vrcholů, které je třeba posunout je mnohem vyšší než jen několik řídicích bodů. Ostatní vrcholy je třeba posunout v souladu s posouvajícími řídicími body. K tomu je použita interpolační metoda, která posunuje vrcholy v nejbližším okolí řídicích bodů s využitím takzvaných zón vlivu (viz Obr. 3. ) které určují v závislosti na vzdálenosti a směru od jednotlivých řídicích bodů, jak bude který vrchol posunut.

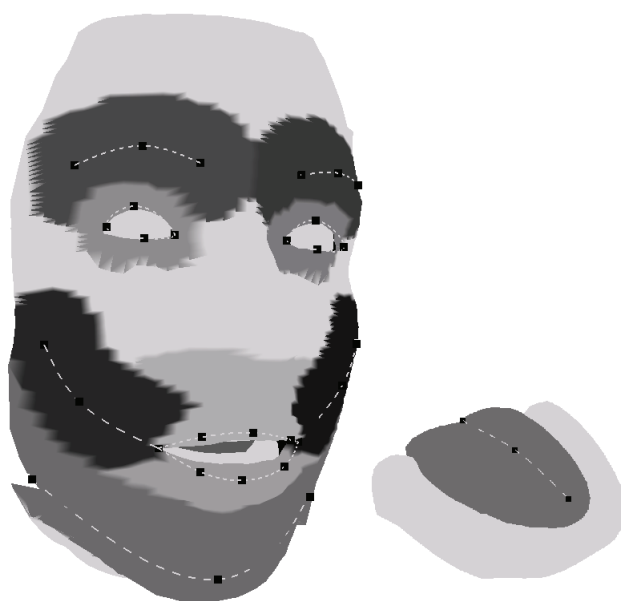
3D model je v reálném čase vykreslován s pomocí knihovny OpenGL. Vykreslovaný model sestává z množiny vrcholů trojúhelníkové sítě a jejich jednotlivých trojúhelníkových ploch. Pro každou elementární plošku je definována textura, která je u modelů získaných snímáním reálných osob získána v průběhu snímání. Vykreslovací část programu využívající knihovnu OpenGL pak využívá nastavení normál (kolmic k jednotlivým elementárním ploškám) a jejich vyhlazování na hranách jednotlivých plošek k tomu, aby vykreslovaný model neměl ostré hrany, které by narušovaly přirozený vzhled. Vykreslovací program umožňuje nastavení simulovaného zdroje osvětlení a právě ze vztahu mezi úhlem pohledu kamery a úhlem, pod kterým dopadá světlo ze světelného zdroje, právě k normále vykreslované plošky vypočítává jas jednotlivých bodů povrchu modelu.

Kromě pohybu řídicích bodů lze tedy, v průběhu vykreslování modelu, modelem různě pohybovat ať už rotačním nebo translačním pohybem v libovolném směru. Je už



jen záležitostí návrhu uživatelského rozhraní zabezpečit ovládání těchto funkcí (například pohyb myši při současném stisknutí některého jejího tlačítka). Dále je možné zapnout či vypnout vykreslování částí modelu.

Jak vyplynulo z diskuzí s odborníky na logopedii a odezírání ze rtů (lidské, nikoli strojové), pro správný vjem vizuálního je důležité vidět zuby a jazyk. Dále pro audiovizuální komunikaci je důležitý oční kontakt. Systém audiovizuální syntézy řeči by bez těchto prvků těžko dosahoval žádané přirozenosti. Proto je třeba modelovat i tyto součásti vizuální řeči, případně neverbální komunikace. Na druhou stranu získání 3D modelů těchto součástí obličeje je od reálných živých subjektů problematické. Tak například pohyby jazyka se dají sledovat buď pomocí speciálních elektrod umístěných na jazyku, nebo pomocí ultrasonografie, ale získání těchto dat vyžaduje náročnější vybavení.



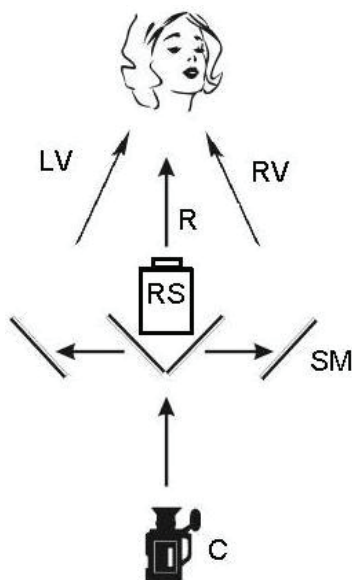
**Obr. 3.** Znárodnění umístění řídicích bodů pro animaci tváře a zón vlivu použitých pro interpolaci vrcholů tváře v oblastech přilehlých k řídicím bodům.

Z uvedených důvodů jsme přistoupili k modelování těchto součástí obličeje jako umělých, tedy uměle vytvořených a pouze adaptovaných na rozměry modelu hlavy konkrétního řečníka. Představovaný systém audiovizuální syntézy řeči tedy obsahuje umělý model jazyka, který je animován podle teoretických předpokladů tvaru jazyka, dále model zubů, kdy horní řada je statická vzhledem k modelu hlavy a spodní řada se pohybuje v souhlasu s pohybem dolní čelisti daným odpovídajícím řídicím bodem. Samostatnou kapitolou je umělý model očí, které jsou modelovány jako kulové objekty, přičemž barva duhovky je odvozena z detekované barvy duhovky snímané osoby. Kulové modely očí jsou animovány buď náhodným pohybem nebo jsou při komunikaci v dialogovém systému využívající odpovídající modul tzv. eye-trackingu

natáčeny podle detekované polohy uživatele. Do budoucnosti však uvažujeme o implementaci řízení očí na základě dat, získaných a zpracovávaných pomocí podobných postupů jako v případě prozodie.

#### 4 Trojrozměrné snímání

K získání trojrozměrného modelu hlavy skutečné osoby je použito snímací zařízení složené z kamery, systému 4 zrcadel a datového projektoru [3], viz Obr. 4. Zrcadla zde slouží ke složení dvou stereoskopických snímků do jednoho pohledu kamery. To nahrazuje klasickou sestavu pro stereovidění, která používá dvě kamery. Použití jedné kamery a zrcadel řeší problém synchronizace dvou kamer. Navíc vzhledem k faktu, že hlava má spíše poměr stran na výšku, zatímco snímek kamery na šířku, vejdou se do jednoho pole kamery oba pohledy bez významné ztráty rozlišení.



**Obr. 4.** Způsob stereoskopického snímání s využitím kamery C, systému 4 zrcadel SM a generátoru strukturovaného osvětlení RS (datapojektor). Generovaný svislý pruh R přejíždí po dobu snímání přes objekt zleva doprava a umožňuje získání 3D souřadnic z levého (LV) a pravého (RV) snímku.

Datový projektor je používán k vygenerování proužkového osvětlení. Pro správné vytvoření 3D modelu je třeba nalézt korespondující dvojice bodů v levém i pravém pohledu pro všechny vrcholy modelu. Nalezení korespondujících dvojic je však obzvláště u hladkých ploch (jako například plocha tváře) velmi těžké či nemožné. Proužkové osvětlení řeší problém nalezení korespondujících dvojic tím, že omezuje korespondující dvojice vždy jen na úzký svislý proužek. Nevýhodou tohoto snímacího systému je, že vyžaduje určitou dobu snímání, po kterou se snímaná osoba nesmí

pohnout. Ta je však cca 20 s, což je u spolupracujícího subjektu poměrně snadno dosažitelné.



**Obr. 5.** Obrázek hlavy snímané osoby a vytvořeného trojrozměrného modelu obličejové části metodou 3D scanování pomocí kamery, systému 4 zrcadel a dataprojektoru generujícího proužkové osvětlení. Obrázek ilustruje možnosti snímacího algoritmu pro snímání 3D modelů v reálných podmínkách. Snímání trvá cca 20 sekund a zpracování cca 5 minut

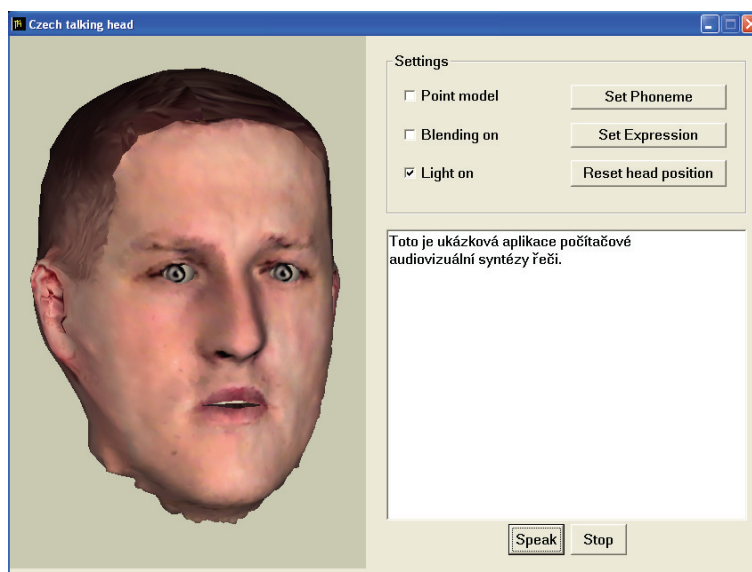
Výsledkem je trojrozměrný model tváře, u kterého jsou manuálně označeny řídicí body umístěné na kontuře rtů. Pro doplnění modelu na model celé hlavy je pak ještě prováděno snímání hlavy z ostatních úhlů pohledu.

## 5 Aplikace

Aplikací audiovizuální syntézy řeči se nabízí celá řada, od aplikací zábavných až po aplikace, které mohou pomoci postiženým osobám. V této sekci budou představeny některé z těchto aplikací, které jsou již realizovány, nebo jejichž realizace je v přípravě. Aplikace jsou seřazeny chronologicky podle doby jejich vzniku.

### 5.1 Prezentáční aplikace mluvící hlavy

První představovanou aplikací je ukázkový program pro prezentaci projektu audiovizuální syntézy řeči. Tento program vznikl nejprve jako testovací aplikace pro ověření funkčnosti audiovizuální syntézy řeči. Později bylo doplněno uživatelské rozhraní programu tak, aby ho bylo možno použít pro účely prezentace výsledků výzkumu odborné i laické veřejnosti.



**Obr. 6.** Obrazovka prezentační aplikace. V levé části vykreslovací okno, v pravé části zadávací textové pole a ovládací tlačítka.

Program je řešen jako dialogová aplikace pro operační systém MS Windows, napsaná v aplikaci MS Visual Studio. Skládá se z vykreslovací části okna, kde je zobrazována vykreslovaná audiovizuální syntéza, dále editačního boxu pro zadávání syntetizovaného textu a několika tlačítek pro ovládání programu. Vzhled uživatelského rozhraní je na Obr. 6. Ovládání pohybu modelu je prováděno pomocí myši. Jejím pohybem za současného stisku levého tlačítka je možno modelem otáčet, při stisknutém pravém tlačítku se model posouvá do stran, při stisknutých obou tlačítkách je možno model přibližovat nebo oddalovat. Pomocí jednotlivých ovládacích prvků lze ovládat tyto funkce:

- point model – zobrazení bodového (drátového) modelu. V tomto režimu není zobrazena plná plocha modelu, ale jen jeho trojúhelníková síť pro vizualizaci trojrozměrnosti modelu.
- blending on – zprůhlednění povrchové vrstvy (kůže). Umožní pohled „pod kůži“ a umožní tak lepší vizualizaci pohybu zubů, jazyka a očí.
- light on – zapnutí/vypnutí světla. Zapnutím světla je do virtuální scény umístěn zdroj osvětlení a při vykreslování modelu je brán v úvahu vztah sklonu povrchu vzhledem k poloze světelného zdroje a poloze kamery.
- set phoneme – nastavení konkrétního fonému/vizému. Po stisknutí tohoto tlačítka se otevře pomocné dialogové okno s výběrem fonému. Po výběru fonému je ve vykreslovacím okně zobrazen odpovídající vizém (artikulační cíl). Dialog dále umožňuje nastavení jednotlivých parametrů ovládajících animaci.
- set expression – nastavení konkrétního výrazu tváře. Podobně jako v předchozím případě se otevře dialogové okno pro výběr výrazu tváře.

Umožňuje nastavení několika základních emocionálních výrazů tváře (úsměv, smutek, zlost apod.) Dialog dále umožňuje nastavení jednotlivých parametrů ovládajících animaci.

- reset head position – nastavení původní polohy hlavy. Vrací model do originální polohy, například po různých rotacích nebo pohybech, kdy je model v nežádoucí poloze nebo dokonce mimo vykreslovací okno.
- play – syntéza a přehrání syntetizovaného textu.
- stop – zastavení přehrávání

Z hlediska programového tento program provádí syntézu vizuální složky řeči. Nejprve s daným textem zavolá modul akustické syntézy řeči, který zpět poskytne zvuková data a tzv. notifikační informace, udávající informaci o hranicích jednotlivých fonémů. Na základě těchto informací je postupně syntetizován vektor parametrů, který přímo v reálném čase ovládá vykreslování modelu tak, aby bylo synchronní se současně přehrávaným zvukem dodaným akustickou syntézou řeči.

## 5.2 Virtuální dvojník

Termín virtuální dvojník vznikl v souvislosti s popularizačními akcemi na podporu prezentace výsledků výzkumu a vývoje pro širokou laickou veřejnost. Poprvé byl tento termín použit ve chvíli, kdy byla poprvé použita data pro všechny tři složky audiovizuální syntézy řeči (statický model hlavy, dynamická data pro ovládání pohybu rtů, řečová data pro akustickou syntézu) od jedné osoby. Tak byly prezentovány výsledky projektu audiovizuální syntézy řeči v televizním pořadu České hlavy pod názvem „Zrození virtuálního dvojníka“.

Na tuto prezentaci pak navázala účast týmu řešitelů na popularizační akci „Věda v ulicích“, pro kterou byl vytvořen exponát „Virtuální dvojník“. V rámci této akce byl upraven postup snímání 3D modelů tváře na co nejjednodušší postup aplikovatelný v reálných podmínkách výstavy pro veřejnost. Tento exponát tak sestává ze dvou částí. První částí je již zmíněný program pro prezentaci/vykreslování audiovizuální syntézy řeči, upravený pro práci pouze s modely obličejové části modelu a zjednodušeným modelem rtů. Druhou částí je pak snímací program určený pro snímání 3D modelu obličejové části dobrovolníků – návštěvníků expozice s cílem získat 3D model jejich tváře a ten pak animovat jako promluvu libovolného textu.

Snímání modelů pro tuto aplikaci provádí v uzavřené kukani cca 3x4m, vybavené snímacím zařízením, která je shora zatemněna a umožňuje tak provádět snímání v téměř libovolných veřejných prostorách. Snímací zařízení je totožné se snímacím mechanismem popsáným v sekci 4, pouze je upraveno do mobilní podoby. Celé snímání 3D modelu tváře trvá cca 20 vteřin, poté následuje asi 5 minut zpracování dat, při kterém je vytvořen 3D model a na něm označeny rty. Následně je možno model zobrazit pro prezentaci a animovat s libovolným textem.

Řešitelský tým projektu audiovizuální syntézy řeči se již pravidelně účastní podobných popularizačních akcí (Věda v ulicích, Dny vědy a techniky, Vědecká kavárna) a během dvoudenní akce je typicky nasnímáno cca 250 modelů dobrovolníků. Výstupem je nejenom okamžitá prezentace mluvícího „virtuálního dvojníka“, ale také nasnímaný trojrozměrný model a v případě zájmu vygenerované video s požadovaným

textem. Tato aplikace se setkává s velkým úspěchem návštěvníků uvedených akcí všech věkových kategorií.

### 5.3 Odpovídající mluvicí hlava

Dalším vylepšením aplikace virtuálního dvojníka je použití jednoduchého modulu rozpoznávání řeči pro vytvoření velmi jednoduché simulace dialogového systému. Aplikace je založena na základu virtuálního dvojníka představeného v minulé podsekcí, s tím rozdílem, že tentokrát jsou použity dvě mluvicí hlavičky, jedna získaná snímáním a reprezentující aktuálního uživatele, druhá získaná předem a reprezentující odpovídající počítačový systém.

Po nasnímání 3D modelu dobrovolníka je tento model zobrazen současně ještě s další animovanou hlavou. Poté je zapnut vstup z mikrofону. Uživatel zadá jednoduchou početní úlohu (typu „Kolik je šest plus čtyři“) a jeho virtuální dvojník tento požadavek zopakuje. Použije se stejný, tedy přirozený zvukový záznam, ale k němu je doplněna vizuální syntéza řeči prezentovaná vytvořeným 3D modelem obličejové části hlavy téhož uživatele.

Toto zadání je rozpoznáno modulem rozpoznávání řeči, je vypočítán výsledek početního úkonu a tento je prezentován zpět uživateli pomocí audiovizuální syntézy řeči prezentované druhým, předem vytvořeným 3D modelem celé hlavy. Výsledkem je velmi jednoduchý dialogový systém, který demonstruje možnosti komunikace člověka se strojem, možné využití technologií a soustředí se na audiovizuální syntézu řeči. Přitom zároveň je dost jednoduchý, aby mohl být prezentován i laické veřejnosti na popularizačních akcích (které často navštěvují děti školního věku) a aby dosahoval potřebné robustnosti za reálných podmínek. Navíc přehráváním vysloveného požadavku uživatelem současně s vizuální syntézou tváře je odstraněn nedostatek, kdy virtuální model uživatele mluví jiným hlasem, neboť do vyvinutí robustní metody transformace hlasu je natrénování nového hlasu v takových podmínkách nereálné.

### 5.4 Úpravy pro jiné jazyky

Systém audiovizuální syntézy řeči byl navržen primárně pro češtinu. Byly však provedeny pokusy adaptovat vizuální syntézu na jiný jazyk, a to ve dvou případech. Prvním případem je adaptace na ruštinu v rámci společného projektu s institutem v Petrohradě, druhým je účast v soutěži mluvicích hlav LIPS2008: Visual Speech Synthesis Challenge, pořádanou v rámci prestižní mezinárodní konference v oblasti zpracování řeči INTERSPEECH 2008 v Brisbane, Austrálie.

V rámci řešeného projektu bilaterální spolupráce Česká republika – Ruská Federace s názvem DIMAS-CZ (Vývoj integrovaného multimodálního pomocného systému) je, mimo jiné úlohy, řešena úloha audiovizuální syntézy ruštiny. Modul akustické syntézy ruštiny byl získán ruskou stranou v rámci jejich předchozího společného projektu s běloruským pracovištěm. Naším cílem tedy je upravit modul vizuální syntézy tak, aby umožňoval syntézu ruských vizémů. Přestože se jedná o příbuzné, slovanské jazyky, sada vizémů se může lišit.

Pro první pokusy bylo třeba najít jednoduchou metodu, později bude proveden výzkum v oblasti definice ruských vizémů. V prvním kroku tedy byla provedena analýza sady ruských vizémů a bylo provedeno jejich mapování na české vizémy. Dále je prováděn výzkum a sběr dat pro vytvoření sady ruských vizémů. Výsledkem je aplikace audiovizuální syntézy řeči pro ruský dialogový systém. Ten navíc používá metodu sledování obličeje pro detekci přítomnosti uživatele a jeho polohy. Program byl proto upraven tak, že přijímá informace od modulu detekce polohy uživatele a provádí natočení hlavy a očí modelu směrem k uživateli.

Pro účast v soutěži Visual Lips Challenge 2008 bylo třeba systém audiovizuální syntézy řeči adaptovat na angličtinu, tak aby byl schopen vyhovět podmínkám soutěže. K tomuto účelu byla organizátory dodána trénovací databáze anglických audiovizuálních promluv. Na rozdíl od zmíněné úpravy pro ruštinu, kde bylo v první fázi provedeno pouze mapování vizémů, rozdíly mezi českými a anglickými vizémy jsou mnohem větší a mohly by způsobit menší srozumitelnost. Proto byla anglická verze našeho systému audiovizuální syntézy řeči natrénována pomocí dodaných dat. Pro úplnost dodejme, že pro testování byly použity originální anglické akustické promluvy namísto syntetizované řeči.

V testovací fázi soutěže, která probíhala až na místě konání konference v laboratoři Queensland University, byla dodána sada 42 vět, pro které bylo úkolem vygenerovat vizuální promluvy. Prvních 30 vět bylo doplněno silně zašuměným akustickým signálem pro test srozumitelnosti, zbylých 12 vět bylo uvažováno pro test audiovizuální konzistence a bylo proto doplněno čistým akustickým signálem. Je třeba dodat, že byly vytvořeny z anglických (význam dávajících) slov umělé věty, které smysl nedávaly pro zamezení efektu učení. Dobrovolníci z řad návštěvníků konference (rodilých anglofonních) pak hodnotili náhodně vybrané věty v těchto dvou kategoriích. Jak je podrobněji uvedeno v sekci 6, v této kapitole prezentovaný systém audiovizuální syntézy řeči adaptovaný pro anglické vizémy vyhrál kategorii srozumitelnosti mluvící hlavy.

## 5.5 Syntéza znakové řeči

Další úlohou, která je řešena na řešitelském pracovišti, je syntéza znakové řeči. Hlavní komponentou znakové řeči jsou manuální gesta, prezentovaná mluvcím (znakujícím). Pro syntézu znakové řeči je tedy třeba vytvořit 3D model celé postavy, znázorňující znakující osobu, který bude umožňovat animaci celé horní poloviny postavy, tj. především ruce. Každý znak je totiž specifikován polohou ruky (rukou), a to jak vzhledem k tělu, tak vzájemnou, dále tvarem ruky, její orientací a pohybem. Při syntéze znakové řeči je tedy třeba vyřešit problém animace rukou a zároveň řešit problémy inverzní kinematiky, aby nedocházelo k fyzikálně nerealistickým kombinacím polohy/orientace rukou.

Ve skutečnosti však nejsou manuální gesta jedinou složkou znakové řeči důležitou k porozumění. Znak a jeho význam v rámci věty je dále specifikován nemanuálními gesty, jako například výrazem obličeje, očním kontaktem, ale především doplňkovou artikulací ústy. I s diskuzí s odborníky na problematiku komunikace znakovou řečí vyplynulo, že artikulace ústy hraje rozhodující roli pro porozumění a

syntetizovaná znaková řeč pouze pomocí manuálních gest byla pro neslyšící nesrozumitelná.

Z uvedených důvodů je významnou součástí syntetizéru znakové řeči systém audiovizuální syntézy řeči. Pro přirozený vjem virtuální postavy (avatara) je samozřejmě syntéza manuálních gest i artikulace integrována do jednoho kombinovaného modelu postavy. Na druhou stranu vlastní syntéza obou součástí probíhá nezávisle. Zatímco základní jednotkou pro manuální gesta znakové řeči jsou znaky, které jsou definovány jejich strukturou složenou z popisu pozice, tvaru orientace a pohybu rukou, základní jednotkou pro artikulaci jsou vizémy reprezentované vektory parametrů pro jim odpovídající artikulační cíle.

Pro syntézu artikulace doplňující syntézu znakové řeči je třeba získat mírně odlišnou databázi pro natrénování databáze základních řečových jednotek. Artikulace používaná jako doplněk znakové řeči přenáší totiž informaci převážně vizuálně na rozdíl od artikulace doprovázející akustickou řeč. V případě znakové řeči je používána přehnaná artikulace, která však napomáhá dobře identifikovat specifické vizémy.

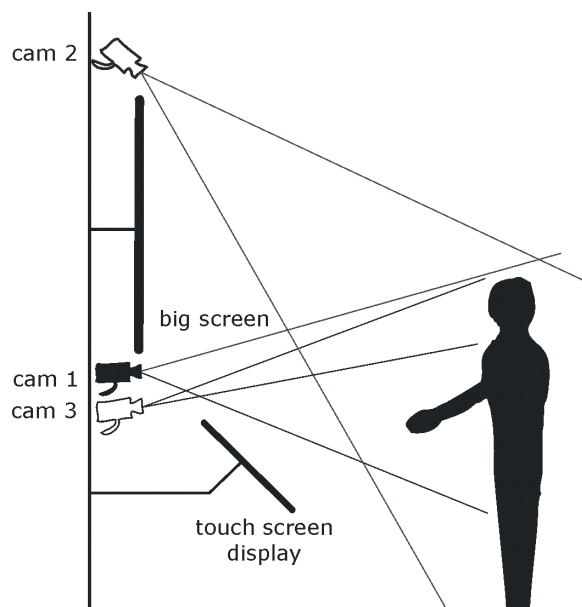
## 5.6 Informační kiosek založený na znakové řeči

Myšlenka informačního kiosku vychází z myšlenky dialogového systému využívajícího mluvené řeči jako základního informačního kanálu. Tato myšlenka je přenesena do světa znakové řeči a využívá ji jako základního komunikačního kanálu. Jenomže zatímco pro komunikaci s počítačem mluvenou řečí postačí mikrofon pro snímání signálu a reproduktor nebo sluchátko pro jeho generování, situace je u znakové řeči horší. Pro syntézu znakové řeči je zapotřebí zobrazovací zařízení umožňující dostatečné zobrazení celé postavy. Pro rozpoznávání znakové řeči je situace ještě obtížnější. Vzhledem k možným zákrytům rukou je pro správné určení, která ruka se nachází před/za tou druhou výhodné snímat trojrozměrnou informaci a k tomu je zapotřebí minimálně dvou kamer. Data z těchto kamer je pak třeba odpovídajícím způsobem zpracovat pro získání trojrozměrné informace.

Návrh prostorového uspořádání informačního kiosku je znázorněn na Obr. 7. Systém používá výkonnou výpočetní stanicí jako základ hardware. Jako vstupní a výstupní zařízení jsou připojeny následující součásti. Pro syntézu znakové řeči je to velkoplošný LCD displej otočený do polohy na výšku. Při poměru stran 9:16 současných obrazovek je takový displej vhodný pro vykreslování virtuální znakové postavy. Pro rozpoznávání znakové řeči je systém vybaven kamerami označenými na obrázku jako cam1 a cam2. Tyto kamery využívají principu stereovidění pro získání trojrozměrné informace o poloze rukou. Pro prostorově úsporné uspořádání kiosku výhodné pro umístění v reálném prostředí letiště, železniční stanice nebo úřadu bylo navrženo uspořádání kamer pro stereovidění nad sebou na rozdíl od mnohem častěji používaného uspořádání vedle sebe. Třetí kamera slouží pro snímání detailu obličeje a umožňuje využívat vizuální rozpoznávání řeči (odezírání ze rtů) pro zvyšování přesnosti rozpoznávání znakové řeči nebo akustické mluvené řeči. Pro rozpoznávání řeči je systém vybaven mikrofonem a pro hlasový výstup reproduktorem (nejsou na obrázku znázorněny). Pro alternativní vstup a pro průběžnou indikaci stavu



probíhajícího dialogu je systém vybaven dotykovým displejem, který je umístěn pod velkou LCD obrazovkou.



**Obr. 7.** Návrh prostorového uspořádání informačního kiosku založeného na zpracování znakové řeči. Kamery pro stereoskopické snímání manuálních gest znakové řeči jsou označeny cam1 a cam2. Kamera pro snímání detailu obličeje pro vizuální rozpoznávání řeči je označena cam3. Obrazovka LCD pro syntézu znakové řeči otočená o 90° do vertikální polohy je označena big screen. Dotykový displej pro zobrazení grafického uživatelského rozhraní a alternativní vstup je označen touch screen display.

Struktura aplikace je navržena tak, že systém je složen z jednotlivých programových modulů, které mezi sebou komunikují pomocí standardního TCP protokolu, což umožňuje případné rozdělení výpočetního výkonu mezi několik stanic. Srdcem je modul řízení dialogu, který ovládá ostatní moduly, čeká na vstupy z rozpoznávačů akustické a znakové řeči, generuje výzvy nebo odpovědi pro moduly syntézy akustické a znakové řeči, komunikuje s dotykovou obrazovkou a udržuje chod celého dialogu.

## 6 Závěr

### 6.1 Shrnutí

V této kapitole byla představena technologie audiovizuální syntézy řeči a její aplikace při komunikaci člověk-stroj. Na jejím počátku byl definován termín audiovizuální syntéza řeči jako doplnění akustické syntézy hlasu o syntézu pohybu rtů pro

maximálně přirozenou simulaci promluvy řeči virtuální postavou (avatarem, modelem hlavy, apod., většinou trojrozměrným), tedy o syntézu vizuální. V závislosti na požadavcích na funkčnost, složitost a přirozenost výsledného systému je možno vizuální syntézu řešit buď jako dvojrozměrnou, tzv. fotorealistickou, nebo založenou na animaci trojrozměrného modelu hlavy ovládaného sadou řídicích parametrů. Pro prezentovaný systém audiovizuální syntézy řeči byla zvolena druhá varianta, a proto je zbytek kapitoly věnován především této variantě.

Pro obě varianty je však třeba pro natrénování syntézy řeči získat poměrně rozsáhlé datové korpusy. Proto je samostatná sekce věnována návrhu a sběru korpusů jak pro akustickou, tak pro vizuální syntézu řeči. Množství a struktura dat jsou samozřejmě závislé na účelu korpusu. Tak pro akustickou syntézu je třeba získat velmi rozsáhlý soubor kvalitních akustických nahrávek včetně záznamů laryngografu, pro syntézu vizuální není tak důležitá kvalita akustického signálu, ale je třeba se zaměřit na vizuální složku dat a nahrávat data za speciálního osvětlení.

Ve stěžejní sekci této kapitoly je představena vlastní technologie jak akustické, tak vizuální syntézy řeči. Zvláštní pozornost je věnována termínu konkatenací syntéza, tedy syntéza spojováním základních řečových jednotek. Tento princip je totiž použit u akustické i vizuální syntézy. Liší se pouze reprezentace základních řečových jednotek. Zatímco u akustické syntézy jsou jako základní řečové jednotky používány trifóny a ukládána je jejich reprezentace ve formě akustického signálu v časové oblasti, u vizuální syntézy jsou základní řečovou jednotkou vizémy a ukládána je reprezentace ve formě řídicích bodů pro ovládání animace trojrozměrného modelu. Diskutována je i důležitá problematika koartikulace, tedy vyřešení přechodových jevů na hranicích jednotlivých spojovaných základních řečových jednotek.

Samostatná sekce je pak věnována systému trojrozměrného snímání statického modelu hlavy/obličeje. Popisuje systém trojrozměrného snímání modelu hlavy nebo obličeje, který používá kameru a systém 4 zrcadel. Tento systém snímání dále používá datový projektor pro generování svíslého proužku osvětlení snímaného subjektu. Podobný systém je použit i pro snímání trojrozměrných dynamických dat, navíc ale s použitím speciálních reflexivních značek nalepených na okolí rtů snímaného subjektu snímaných v infračerveném osvětlení.

Poslední sekce pak uvádí příklady aplikací audiovizuální syntézy řeči, od jednoduché aplikace pro prezentaci této technologie přes její další aplikace v systémech komunikace člověk-stroj až po využití v systémech syntézy znakové řeči. Uveden je i návrh informačního kiosku používajícího znakovou řeč jako hlavní komunikační prostředek.

Kapitola tedy podává přehled o vývoji technologie audiovizuální syntézy řeči, od nasbíraných řečových korpusů přes zpracování dat a trojrozměrné snímání až k aplikacím. Umožňuje tak čtenáři proniknout do tajů návrhu systémů multimodální komunikace člověk-stroj a jejich některých možných aplikací.

## 6.2 Zhodnocení

Vývoj prezentované technologie audiovizuální syntézy řeči (respektive její vizuální části) začal přibližně v roce 1999 nahráváním prvního audiovizuálního řečového

korpusu pro modelování pohybu rtů a vývojem prvních verzí metod vlastní syntézy a vykreslování trojrozměrného modelu. Úplně první testy byly provedeny pouze na velmi jednoduchém modelu rtů složeném z několika málo trojúhelníků 3D sítě. První funkční verze pak byla prezentována jako souhrn několika propojených diplomových prací a byla již založena na umělém modelu celé hlavy vytvořeném ve 3D modelovacím programu a importovaném do systému audiovizuální syntézy řeči.

První syntéza však vykazovala řadu nedostatků. Od té doby byl proto proveden výzkum mimo jiné v oblastech zlepšování modelování pohybů rtů pro jednotlivé vizémy, dále ve velmi důležité oblasti koartikulace, tedy vyřešení problému vizuálních nepřirozených jevů na hranicích jednotlivých spojovaných vizémů, v oblasti úpravy artikulace pro specifickou úlohu doplnění syntézy znakového jazyka a úpravy pro funkci s akustickou syntézou pro jiné jazyky (ruština, angličtina). Relativně samostatnou úlohou byl návrh trojrozměrného snímacího procesu pro získávání statických 3D modelů obličejů nebo celých hlav od libovolné osoby.

Produktem výzkumu je modul audiovizuální syntézy řeči, který byl mnohokrát ověřován při subjektivních percepčních testech (u akustické syntézy se většinou mluví o tzv. poslechových testech, zde je tento výraz vzhledem k důrazu na vizuální složku nahrazen slovem percepční – tedy na vjemu založený test). Na základě těchto testů byla technologie neustále zlepšována. Postupně bylo nahráno několik řečových databází s cílem upravit pohyby rtů tak, aby lépe odpovídaly konkrétnímu mluvčímu nebo aby přesněji artikulovaly pro potřeby uživatelů, kteří jsou více závislí na odezírání ze rtů. Hlavní část výzkumu je tématem disertační práce zpracované na řešitelském pracovišti [12].

Výsledky prezentovaného výzkumu se také ukázaly být zajímavé pro veřejnou prezentaci. Vzhledem k audiovizuální povaze technologie byly prezentace laické veřejnosti na popularizačních akcích kladně přijaty. To je způsobeno mimo jiné atraktivností procesu snímání a možnosti zobrazení vlastního mluvčího „virtuálního dvojníka“ na prezentační obrazovce. Tato aplikace zaznamenala na popularizačních akcích „Věda v ulicích“, „Vědecká kavárna“ nebo „Dny vědy a techniky“ úspěch, v největší míře u školní mládeže, ale i u dospělých, důchodce nevyjímaje. Během dvoudenní akce bylo vždy nasnímáno typicky cca 250 dobrovolníků. V současné době se již těžiště prezentací posouvá do ukázek komunikace pomocí gest a znakové řeči. Ukázky multimodální komunikace člověk-stroj však zůstávají trvalou složkou podobných akcí.

Ohlas na výsledky výzkumu jak na domácím tak mezinárodním poli, jak v odborných kruzích, tak v laické veřejnosti ukazuje, že byl proveden kvalitní výzkum s uznávanými dosaženými výsledky. Od prezentační aplikace pro veřejnost přes aplikaci pro doplnění znakové řeči až po úspěšně provedené úpravy pro jiné jazyky, výsledky prokazují dobrou aplikovatelnost. Výzkum vyplňuje mezeru mezi oblastmi mnohem více rozvinutých oborů zpracování mluvené řeči, jazyka a textu a počítačového vidění.

### 6.3 Ocenění ve Visual Speech Synthesis Challenge

Jednou ze zmíněných aplikací byla úprava systému audiovizuální syntézy řeči pro anglický jazyk. Cílem této úpravy byla účast v soutěži mluvčích hlav LIPS2008: Visual Speech Synthesis Challenge [12], [14], pořádané při příležitosti prestižní mezinárodní konference INTERSPEECH 2008, konané v Brisbane v Austrálii.

Cílem této soutěže bylo první mezinárodní srovnání výsledků systémů audiovizuální syntézy řeči vyvinutých různými týmy z celého světa. Aby byly testy co nejvíce srovnatelné, bylo podmínkou zadaptovat existující systém (pokud byl původně v jiném jazyce) na angličtinu. Za tím účelem byl organizátory poskytnut audiovizuální řečový korpus pro natrénování anglických vizemů. Ten byl dostupný všem zájemcům o účast v soutěži. Protože cílem bylo otestovat různé vizuální syntézy řeči a nikoli akustické, byly všechny vizuální systémy doplněny raději o původní přirozené než o syntetizované akustické promluvy.

Testovací texty k syntetizování a akustické promluvy byly dodány až těsně před začátkem konference v místě konání ve vyhrazené laboratoři v první den (tutorial day) konference. Na místě byly vygenerovány všemi účastníky testovací vizuální promluvy. Ty byly ořezány rozměrově pouze na oblast úst a doplněny akustickými promluvami. Tyto promluvy pak byly hodnoceny nezávislými dobrovolníky (rodilými anglofonními návštěvníky konference) naslepo, tedy bez označení testovaných systémů. Všechny promluvy tvořily věty složené z anglických slov, ale nedávající význam pro zamezení efektu učení.



**Obr. 8.** Ocenění „Golden Lips Award for Intelligibility“, získané v rámci soutěže mluvčích hlav LIPS2008: Visual Speech Synthesis Challenge, pořádané při příležitosti prestižní mezinárodní konference INTERSPEECH 2008.

První skupina vět byla doplněna o zašuměný akustický signál na hranici srozumitelnosti, kde byl předpoklad největšího zisku z doplnění syntetizované vizuální složky. Úkolem posuzovatelů bylo přepsat audiovizuálně vnímanou promluvu zpět do

textové podoby a hodnocena byla chybovost na úrovni slov v procentech. Řešitelský tým použil upravený systém, jak bylo popsáno v sekci 5.4 a s tímto systémem, který byl původně navržen pro češtinu a pro angličtinu pouze přetrénován pomocí dat dodaných organizátory, se úspěšně zúčastnil soutěže. V této první kategorii, nazvané srozumitelnost (intelligibility) získal za první místo ocenění „Golden Lips Award“, viz Obr. 8.

Druhá skupina vět byla doplněna o čistý akustický signál a úkolem posuzovatelů bylo posoudit kvalitu synchronizace akustického a vizuálního signálu. Jednotliví posuzovatelé hodnotili audiovizuální synchronizaci známkami na stupnici 1–5. Tuto kategorii vyhrál systém vyvinutý na Leibnizově univerzitě v Hannoveru v Německu.

Úspěch v mezinárodní soutěži, pořádané navíc při příležitosti patrně nejprestižnější mezinárodní konference v oblasti zpracování řeči INTERSPEECH, navštěvované více než 1000 návštěvníky potvrzuje kvalitu výzkumu provedeného na řešitelském pracovišti a srovnatelnost dosažených výsledků se světovou špičkou. Připomeňme ještě jednou, že systémy, původně navrhované pro jiný jazyk, než angličtinu byly nutností přetrénovat systém na rozdíl od původního návrhu pro cílový jazyk lehce handicapovány.

*Poděkování:* Tato kapitola vznikla za podpory Grantové agentury Akademie věd České republiky (GA AV ČR) v rámci projektu Informační společnost č. GA AV ČR 1ET101470416: MUSSLAP – „Multimodální zpracování lidské znakové a mluvené řeči počítačem pro komunikaci člověk-stroj“ a za podpory Ministerstva školství, mládeže a tělovýchovy České republiky (MŠMT) v rámci projektu KONTAKT č. ME08106: DIMAS-CZ – „Vývoj integrálního multimodálního pomocného systému“.

Autor by rád poděkoval všem spolupracovníkům, kteří přispěli ve větší či menší míře k provedení výzkumu v oblasti audiovizuální syntézy řeči a příbuzných oblastech audiovizuálního zpracování řeči nebo zpracování znakové řeči, jmenovitě především Ing. Zdeňku Krňoulovi, Ph.D., Ing. Pavlu Camprovi, Ing. Marku Hruzovi a Ing. Petru Císařovi, Ph.D., jakož i týmu pracujícímu na akustické syntéze řeči pod vedením Ing. Jindřicha Matouška, Ph.D.

## Literatura

- [1] Psutka, J., Müller, L., Matoušek, J., Radová, V.: *Mluvíme s počítačem česky*. Praha, Academia, 2006. 752 s. ISBN 80-200-1309-1.
- [2] McGurk, H., MacDonald, J.: Hearing lips and seeing voices. *Nature* **264**(5588), (1976) 746–748
- [3] Železný, M., Krňoul, Z., Císař, P., Matoušek, J.: Design, implementation and evaluation of the Czech realistic audio-visual speech synthesis. In *Signal Processing* **86**(12) (2006) 3657–3673.
- [4] Matoušek, J., Romportl, J., Tihelka, D., Tychtl, Z.: Recent improvements on ARTIC: Czech text-to-speech systém. In *Proceedings of ICSLP 2004*, vol. 3, Jeju, Korea, 2004, pp. 1933–1936.

- [5] Donovan, R. E., Woodland, P. C.: A hidden Markov-model-based trainable speech synthesizer. In *Comput. Speech Language* **13** (1999.0123) (1999) 223–241.
- [6] Matoušek, J., Tihelka, D., Psutka, J.: Automatic segmentation for Czech concatenative speech synthesis using statistical approach with boundary-specific correction. In *Proceedings of EUROSPEECH 2003*, Geneva, Switzerland, 2003, pp. 301–304.
- [7] Tihelka, D.: Symbolic prosody driven unit selection for highly natural synthetic speech. In *Proceedings of EUROSPEECH 2005*, Lisbon, Portugal, 2005, pp. 2525–2528.
- [8] Romportl, J., Matoušek, J.: Advanced prosody modelling. In *Proceedings of TSD 2004*, Bonn, Heidelberg, Springer, Berlin, 2004, pp. 441–447.
- [9] Tihelka, D., Matoušek, J.: The design of Czech language formal listening tests for the evaluation of TTS systems. In *Proceedings of LREC 2004*, Lisbon, Portugal, 2004, pp. 2099–2102.
- [10] Cohen, M. M., Massaro, D. W.: Modeling coarticulation in synthetic visual speech. In N. M. Thalmann & D. Thalmann (Eds.): *Models and Techniques in Computer Animation*. Tokyo: Springer-Verlag, 1993, pp. 139–156.
- [11] Krňoul, Z., Železný, M., Císař, P.: Face model reconstruction for Czech audio-visual speech synthesis. In *SPECOM'2004*. Saint-Petersburg : SPIIRAS, 2004. pp. 47–51. ISBN 5-7452-0110-x.
- [12] Krňoul, Z. *Automatická syntéza vizuální řeči - mluvící hlava*. Disertační práce. Plzeň: Západočeská univerzita v Plzni, 2008, 143 s.
- [13] Theobald, B., Fagel, S., Bailly, G., Elisei, F.: LIPS2008: Visual Speech Synthesis Challenge. In *Proceedings of Interspeech*, pp. 2310–2313, 2008.
- [14] <http://www.lips2008.org/>

# Index

## A

abstrakce 365, **376**, 377, 398  
adaptácia 79, 150, 162, 201, 219, 232,  
314, 350, 408  
afordancie 411, 415  
agent 31, 264, 395  
agent inteligentní 309, 314-315, 409-  
412  
– adaptívny 408  
– hybridní 313  
– intencionální 313  
– reaktivní 313  
– situovaný 408  
– sociální agent 307, 313, 412  
– technologie 312  
– vlastností 312  
AIBO 217, 221, 304, 307  
akustická syntéza řeči 426  
alfa–beta orezanie 107  
algoritmus OLA (overlap and add) 434  
algoritmus Viterbiho 429, 433  
android 305  
ARTIC (ARTificial Talker In Czech)  
433  
A–star 103  
atomická formula 22, 118  
atribút 1, 134-135, 410  
audiovizuální syntéza řeči 429, 434  
autokatalytický systém 248  
automatické dokazovanie 111, **116**,  
**119**  
autonómnosť 146, 159, 195, 304, 313

## C

CAPTCHA 88  
colliculus inferior 284  
corpus geniculatum mediale 284

časové postupnosti, modelovanie 38,  
61, 95, 206  
čínska izba 88, 402

## D

Darwinovská evolúcia 90, 231  
databáza faktov 108, 109  
databáza znalostí 108, 109-111  
datové korpusy pro akustickou syntézu  
429  
datové korpusy pro vizuální syntézu  
řeči 429  
Dawkinsova memetika 243, 377  
dedičnosť 130, 133, 242  
defaultné uvažovanie 131, 135  
defuzzifikácia 145, 153, 168, 170, 178,  
182, 185  
deklaratívne znalosti 92  
derivačný (odvodzovací) strom 127  
difón 433  
dimenzie fitness 208-209  
disjunkcia 9, 15-16, 20-21, 116, 121  
diskriminačný strom 404-405  
dôkaz priamy 120  
dôkaz sporom 121, 123  
dopredné reťazenie pravidiel 109  
dotazovací jazyk 133  
dráha (nervová) 283  
drift 234  
dvojcestná neurónová sieť **406**  
dynamické procesy 167, 219  
dynamika nenatrérovanej rekurentnej  
neurónovej siete **66**

## E

efektory 199, 412  
echo stavy, sieť 68-71

- Eigen, Manfred 248, 250  
ekosystémy – simulácia 213  
ELIZA 402, 403, 407  
Elmanova jednoduchá rekurentná  
neurónová sieť 45-48, 51-52, 61-64,  
73, 77, 79  
embedded systém 140-141  
emergencia 237, 262, 267, 386, 400,  
412  
– stratégia 267  
entropia genotypová a fenotypová 276  
epigenetická embryogenéza 242  
e-puck 196, 220, 221  
evolúcia in materio 202  
evolúcia interaktívna 210  
evolučná robotika 195  
evolučné algoritmy 195, **196**, 239, **335**  
evolučne stabilná stratégia 268-271  
evolučné stratégie 197, **350**  
existenčný kvantifikátor 117, 122  
expansion uzla 95  
expertné systémy 107, **110**, 162  
explicitná–implicitná dimenzia fitness  
209  
expresívna sila prepisu 203  
externá–interná dimenzia fitness 209
- F**
- fenotyp 198, 203-207, 242, 244-245,  
257-264  
filosofie 312, **355**, 390-391  
fitness (alt. fitness) funkcia 196, 233-  
235, 246, 254, 264, 336, 347  
FOAF – Friend Of A Friend ontológia  
136  
Fogel, Lawrence J. 345  
folding (skladanie) reťazcov **255**  
formalizovanie 165  
formálne dokazovanie 119  
formální teórie fuzzy množín 32-34  
formula 118  
FPGA (Field Programmable Gate  
Array) obvody 146, 156, 191, 202  
funkcia fitness 196, 233-235, 246, 254,  
264, 336, 347  
funkcionálna–behaviorálna dimenzia  
fitness 208  
funkčné symboly 118  
funkčný model mysli 355  
fuzzifikácia 145, 169  
fuzzy logika **1**  
fuzzy matematika 32-34  
fuzzy množiny 2, 151, 163-168  
fuzzy regulátor 145, 169, 171-184,  
189-191  
fuzzy systémy 90, 145-192  
fyzikálne prostredie – simulácia **214-  
215**
- G**
- ganglion spirála 283  
generalizace 24, 355, 371, 376-377  
genetické programovanie 162, 163,  
197, **202**, 335, **346-349**  
genetický algoritmus **186**, 197-198,  
**235-243**, **338-346**  
genotyp 198, 203-207, 233-235, 241-  
242, 244-245, 257-264, 275-276  
gény 204, 336, 345  
Gödelova logika 6, 8-10, 14, 16-19, 25,  
171  
Goguen, Joseph A. 4  
Goldberg, David E. 267,  
GOLEM 222-223  
graduální vlastnost 1-35  
gramatiky 61, 64, 204, 206, 240  
gynoid 305
- H**
- Hanojské veže 91, 94-96, 102-103  
heuristika 95, 101-102, 106, 245  
hill climbing 101,  
hlava, model 426, 435, 439, 442  
Holland, John 197, 235, 267, 310, 345  
hry 93-107, 264-266, 412-413  
humanoid 300-310, 314-332



– generátor chůze 316  
humanoid charakteristika 305  
– prognóza vývoje 330  
– umělý mozek (humanoidů) 308  
– způsoby chůze 316  
humanoidní roboti  
– cíle vývoje 321  
– interakce s člověkem– rizika 328  
– předpoklady realizace 314  
– úspěšné realizace 322  
– výchozí aparát vývoje 309  
hybridné systémy 142, 144, **151**, 163,  
181, 313, 403  
hypercyklus 248-249

## CH

chatbot 407,  
chemostat 248-262

## I

implikácia 8-9, 116, 171  
inferencia 111-114, 383, 399  
inferenčný (usudzovací) mechanizmus  
108, 132-133, 145, 152, 170-176,  
178, 189  
informační kiosek založený na znakové  
řeči 444  
inštanca 24, 132, 407, 414-415, 433  
inteligencia 87, 150, 160, 235, 302,  
309, 311-312, 331, 352, 396  
intencionalita 362, 377-380, 396  
intencionální obsahy 313, 356-359,  
370, **377-380**  
interaktivna evolúcia **210**  
interaurální časové zpoždění 282, 284  
interaurální fázový rozdíl 282, 295  
interaurální intenzitní rozdíl 282  
intuice 377  
intuicionistická logika 6, 12, 17, 25

## J

jádro (nervové) 283

jazyková hra 412-414  
– diskriminačná 412  
– hádacia 412  
– pozorovacia 413  
jsoucna 356-392

## K

Kalmanov filter 39, **54-60**, **82-84**  
Khepera 196, 202, 212, 216, 220, **221**  
kiosek, informační založený na  
znakové řeči 444  
klasická logika 1, 26  
klauzula 121, 123-124  
koartikulace 426, 435-436  
koevolúcia 191, **211-212**, **246-247**  
kognitivistika (kognitívna veda) 161  
kochleární jádro 283, 284-285  
kolektívna vina 273  
kompaktnosť prepisu 203  
komparace 376, 377  
kompetitívna koevolúcia 211, 246  
koncept 129, 130, 132, 397-398, 400,  
402  
konceptuálny priestor 398  
konflikt, riešenie 108, 265, 314  
konjunkcia 7, 9, 18, 33, 116  
konjunktívna normálna forma 123  
konkatenační princip 435  
konkatenační syntéza 433  
konštruktivizmus 403  
kontradikcia 13, 119, 121  
kontraktívne transformácie 67, 70  
kooperatívna koevolúcia 191, 211, 247,  
267  
kooperatívna štruktúra 179  
koordinácia 156, 199, 409, **412-414**  
korpus významov 407  
korpusy pro akustickou syntézu 426,  
428-429  
korpusy pro vizuální syntézu řeči 429-  
432  
Koza, John 310  
kříženie 197, 337, **339-340**, **344-346**,  
**349**

kvantifikátory 22-25, 117, 122-123  
kyborg 305

## L

Lamarckova evolučná teória 242-243  
LEGO 221-222  
ľudské myšlenie 161, 355-392, 401  
lingvistická premenná 151  
lingvistické modely 167-168  
listový uzol 106-107, 112-113  
literál 121  
logika BL 12-13  
L-systémy 204, 206  
Łukasiewiczova logika 6-14, 17-22, 29, 171

## M

Markovov model 68, 344, 433  
Markovov model s premenlivou dĺžkou pamäti 68  
Markovovská architekturná predispozícia 68  
mediální jádro corpus trapezoideum 284  
Melovské frekvenční keprální koeficienty MFCC 429  
metafora čínskej izby 88, 402  
minimax stratégia 106-107  
mluvící hlava 426, 439, 442  
mobilní roboty  
– historie 300  
– kráčejší roboty 300, 316-319  
– navigace 319  
podvozky 316  
– pohony 320  
model rtů 429-432  
modelovanie  
– časových postupností 38, 61  
modely hlavy 426, 435, 439, 442  
modus ponens 8, 13, 120  
Molecubes 215, 224-225  
molekulárny darwinizmus **248-263**  
monotónne uvažovanie 131

multiagentní systémy 264, 311,  
morfining 434  
morfológia 198, 200, 203, 206, 220, 222, 224, 232, 284-285  
mozkový kmen 284, 293, 295  
mutácia 197-198, 202, 232, 237, 250, 254, **340, 349-351**

## N

Nashova rovnováha 270  
nelineárny bezdrôtový komunikačný kanál 74  
nemonotónne uvažovanie 131, 399, 417  
neodarwinizmus 232-234,  
neredukovateľná zložitost' 240, 276, 386  
neurón 37-38, **39-40**, 151, 184, 200, 208  
neuron (biologický) 281, **283**, 385  
neurónová sieť **37**, 89, 146, 164, **200**, 312, 365, 370, 406  
neurónové riadenie 72-74, 147  
neurovedy 161, 281  
neutrálne mutácie 248, 262-263

## O

oddelený Kalmanov filter 55, **58**  
odvodzovanie znalostí 13, 31, 131-132, 145, 383  
ohodnotené stromy 106, 114, 202, 346-347  
OLA algoritmus (overlap and add) 434  
olivární jádro 283, 291  
ontológie 133, **135-137**, 356, 382  
operace lidského myšlení 358, **375-377**  
optimalizačný algoritmus 196, 235, 336  
oscilujúce funkcie na riadenie **200**  
ostrá vlastnosť 23, 168, 171  
otvorený svet 93, 131, 396

## P

paradox hromady 28-31  
paradox logické vševědoucnosti 31  
paradox Poincarého 31  
Pareto–optimální front 336  
plán 93, 95-100  
počítačová inteligencia 150, 163, 235, 352  
Poincarého paradox 31  
populácia 90, 186, 197, 231-238, 245, 264, 337-340, 349, 406  
povrch fitness 234  
pravdepodobnosť križenia 339-340  
pravdepodobnosť mutácie 253-254, 340  
pravdivostná tabuľka 5, 119  
pravidlo  
– fuzzy 184  
– generalizace 24  
– modus ponens 13, 120  
– necessitate 19  
– produkčné 107  
– rezolučné 121  
pravidlové systémy 108-110  
predikátová fuzzy logika 22  
predikátová logika 116-129, 403  
predikátové  
– konštanty 116  
– premenné 117  
– symboly 116, 118  
predikcia nasledujúceho symbolu 61  
preformačná embryogenéza 242  
prehľadávanie  
– do hĺbky 97, 99-100  
– do šírky 97  
priamy dôkaz 120, 123  
prírodný výber 232, 237, 243  
problém ukotvenia symbolov 402  
procedurálne znalosti 92, 135  
produkčné systémy 108-110  
produktová logika 8-12, 16, 21, 27  
projekce 359-360, 367  
proporcionálny výber 338-339

prototyp 398, 404-406, 413  
psychofyzika 283

## R

rámce 133, **134-135**, 410, 415  
reaktivita 159, 313  
reaktívne správanie 190, 199, 201, 313, 403  
Rechenberg, Ingo 197  
rekonfigurovateľné roboty 224  
rekurentná neurónová sieť **37-86**, 200-201  
rekurentné učenie v reálnom čase **54**  
rekurzívny limit 205  
replikácia 236, 243-244, 250, 254  
replikátor 235-279,  
reprezentácia významov 92, 311, 368, 398, **402-408**  
– vyvolaná 401  
– ikonická 403  
– kategoriálna 403  
– oddelená 401  
– prototypová 406  
reziduum t–normy 8-9  
rezolučný princíp **121-129**  
riadenie 142, 147, 156  
riadenie hybridného pohonu 72-74  
radiaca stratégia 95  
riešenie konfliktov 108, 265, 314  
robot 189  
– hustota instalácií 303  
– inteligentní 302, 307, 309, 313  
– průmyslový 302  
– rizika koexistence s ľuďmi  
– servisní pro osobní užití 304  
– servisní profesionální 302  
– zábavný 304, 323  
robotika evolučná 195-229  
roboty rekonfigurovateľné 224  
rozšírený Kalmanov filter 54, **56-60**, **82-84**  
ruleta 267, 339

## Ř

řeč, syntéza **425-450**  
řídící body 426, 431, 436

## S

selekcia rodičov 236, 338, 340  
selekčný tlak 196, 211, 240-241, 250, 254  
sémantická interoperabilita 135  
sémantická sieť **129-135**, 329, 380  
sémantické modelovanie znalostí **129**  
sémantika 396  
– denotačná 397  
– extenzionálna 397  
– funkcionalistická 397  
– intenzionálna 397  
– kognitívna 397  
– realistická 397  
semiotický trojuholník 399, 412  
semiotika **399-400**  
senzory robota 188-191, 199, 212, 310, 315, 404  
Shakey 404  
SHRDLU 402, 403, 404  
schémy  
– v genetických algoritmoch 197, **343-344**  
– myšlení 377  
– reprezentácie znalostí 130, 408  
schopnosť vývoja (z angl. termínu evolvability) 204  
Schuster, Peter 248, 275  
sieť s echo stavmi **68-71**  
sila mutácie 350-351  
silná konjunkce 9, 14, 15, 18, 20, 27, 33  
Sims, Karl 195, 196, 200, 204, 206, 209-212, 215, 222  
simulácia ekosystémov **213-214**  
simulácia fyzikálneho prostredia **214-215**  
simulácia emergencie stratégie **267-274**

skolemizácia 122  
Skolemova funkcia 122  
slabá konjunkce 9, 18  
slepé prehľadávanie 96, **98**, 101, 275, 345  
sluchová dráha 283-288, 290, 293-294  
slyšení prostorové 281-297  
sociálne dilemy **264-265**  
spätné reťazenie 99, 109-114  
spätné šírenie chybového signálu 37, **41-45**, **79-80**, 151, 181, 185  
– v čase **51-53**, **80-81**  
spätnoväzbové riadenie 146  
spätnoväzbový obvod 144  
spracovanie postupností symbolov **61-63**, 68  
stavový priestor 66-67, 93-95  
stereovidění 430-432, 438, 444  
stratégia alfa–beta orezania 107  
stratégia minimax 107  
strom 95-97, 105, 202, 246, 400, 404  
sub–symbolická umelá inteligencia 89-90  
sylogismus 376, 377  
symbolická regresia 347  
symbolická umelá inteligencia **87-93**  
syntaktické pravidlá 118  
syntetizér znakové řeči 427  
syntéza řeči 425-450  
systém  
– distribuovaný 73, 164  
– multiagentní – MAS **311-312**

## Š

šírenie aktivácie 132  
štandardizácia premenných 122, 124

## T

tautológia 12-13, 119  
teória 2, **119**, 123, 386  
t–konorma **21**, **166**, 185  
t–norma **8**, **167**  
tragédia spoločného **271-274**

trifón 428, **433**  
trojrozměrné snímání 426, 431, **438-439**  
TTS – text-to-speech 432  
Turingov test 88, 407  
turnajový výber 337, 340  
tvorba novosti 380

U

učenie 54, 81, 179, 185, 201, 314, 408-409, **414**  
umelá chémia 248  
umění 390  
unifikácia 124, 126  
unifikátor 124-125  
uzavretý svet 132

V

variácia 232, 337-338, 350  
väžňova dilema 264, 265, **266-267**, 272-273  
věda 390  
viacprúdový Kalmanov filter **59-60**  
viazané premenné 117  
virtuální dvojník **441-442**  
Viterbiho algoritmus 429, 433  
vizém 426, 433, 434-448  
vněmy 90, 359-361, 406  
vnímání 87, 283, 302, 359, 395, 401, 408  
vnorené systémy 139-193  
vnorený regulátor 145, 150  
vnútorný paralelizmus 345  
voľné premenné 117  
všeobecný kvantifikátor 117  
výpočtová inteligencia 150, 163, 235, 352  
vysvetľovací mechanizmus 111, **115**  
význam 133, 135-136, 362, 379, 380, 395, **396-400**, **408-409**, **410-416**, 443  
vznik komplexného orgánu 241

## W

Wright, Sewall 231-235, 243, 275

## Z

zabudované systémy 141  
zhluková analýza 177  
znaková řeč 443-445  
znalostné inžinierstvo 110  
znalostné technológie 135  
zried'ovací tok 249, 250