Edícia výskumných textov
Informatiky a informačných technológií

Umelá inteligencia a kognitívna veda III

PUBLIKÁCIU PODPORILO ZDRUŽENIE

GRATEX IT INŠTITÚT

v rámci fondu GraFIIT www.gratex.com

Táto publikácia vznikla za podpory grantovej agentúry VEGA SR v rámci grantovej úlohy VG 1/0141/10

Vladimír Kvasnička
Jiří Pospíchal
Pavol Návrat
Peter Lacko
Ľubomír Varga
(editori)

Umelá inteligencia a kognitívna veda III

Slovenská technická univerzita v Bratislave

Redakčná rada série *Umelá inteligencia a kognitívna veda*

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

PUBLIKÁCIU PODPORILO ZDRUŽENIE

GRATEX IT INŠTITÚT

v rámci fondu GraFIIT www.gratex.com

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

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

Schválilo vedenie Fakulty informatiky a informačných technológií STU v Bratislave dňa 8.2.2011, uznesenie č. 2011.5.1

© Fakulta informatiky a informačných technológií STU v Bratislave

ISBN 978-80-227-3542-1

AUTORSKÝ KOLEKTÍV KNIHY

Ing. Jan Burian, PhD. E-mail: mindful.creator@gmail.com	Katedra informačního a znalostního inženýrství, FIS, VŠE, nám. W. Churchilla, 130 00 Praha 3
RNDr. Luděk Cienciala, PhD. E-mail: ludek.cienciala@fpf.slu.cz	Ústav informatiky, Filozoficko-přírodovědecká fakulta, Slezská univerzita, 746 01 Opava
RNDr. Lucie Ciencialová, PhD. E-mail: lucie.ciencialova@fpf.slu.cz	Ústav informatiky, Filozoficko-přírodovědecká fakulta, Slezská univerzita, 746 01 Opava
Doc. Ing. Igor Farkaš, PhD. E-mail: farkas@fmph.uniba.sk	Katedra aplikovanej informatiky, Fakulta matematiky, fyziky a informatiky, Univerzita Komenského, Mlynská dolina, 84248 Bratislava
Prof. PhDr. Silvia Gáliková, CSc. E-mail: silvia@libris.sk	Filozofický ústav Slovenskej Akadémie vied, Klemensova 19, 813 64 Bratislava
Mgr. Jana Horáková, PhD. E-mail: horakova@phil.muni.cz	Masarykova univerzita, Filozofická fakulta, Ústav hudební vědy, A. Nováka 1, 602 00 Brno, Česká republika
Doc. RNDr. Alica Kelemenová, CSc. E-mail: kelemenova@fpf.slu.cz	Ústav informatiky, Filozoficko-přírodovědecká fakulta, Slezská univerzita, 746 01 Opava
Ing. Peter Lacko, PhD. E-mail: lackop@is.stuba.sk	Fakulta informatiky a informačných technológií Slovenská technická univerzita Ilkovičova 3, 842 16 Bratislava
RNDr. Andrej Lúčny, PhD. E-mail: lucny@fmph.uniba.sk	Katedra aplikovanej informatiky, Fakulta matematiky, fyziky a informatiky, Univerzita Komenského, Mlynská dolina, 84248 Bratislava
Mgr. Elena Matějková E-mail: ella.matejkova@gmail.com	Katedra filozofie FF UCM, Nám. J. Herdu 2, 917 01 Trnava
Prof. RNDr. Peter Mikulecký, CSc. E-mail: peter.mikulecky@uhk.cz	Fakulta informatiky a managementu, Univerzita Hradec Králové, Rokitanského 62, 500 03 Hradec Králové
Ing. Michal Oravec E-mail: michal.oravec@stuba.sk	Ústav riadenia a priemyselnej informatiky, Fakulta elektrotechniky a informatiky, Slovenská technická univerzita, Ilkovičova 3, 812 19 Bratislava
prof. RNDr. Jiří Pospíchal, DrSc. E-mail: pospichal@fiit.stuba.sk	Ústav aplikovanej informatiky, Fakulta informatiky a informačných technológií Slovenská technická univerzita Ilkovičova 3, 842 16 Bratislava 4

Doc. PhDr. Ján Rybár, PhD.
E-mail: rybar@fmph.uniba.sk

Katedra aplikovanej informatiky, Fakulta matematiky, fyziky a informatiky, Univerzita Komenského, Mlynská dolina, 84248 Bratislava

Doc. Ing. Ivan Sekaj, PhD.
E-mail: ivan.sekaj@stuba.sk

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

Doc. RNDr. Petr Sosík, PhD.
E-mail: petr.sosik@fpf.slu.cz

Ústav informatiky, Filozoficko-přírodovědecká fakulta, Slezská univerzita, 746 01 Opava

Prof. RNDr. Peter Sýkora, PhD.
E-mail: sykora@infovek.sk

Katedra filozofie FF UCM, Nám. J. Herdu 2, 917 01 Trnava

Doc. RNDr. Michal Šabo, CSc.
E-mail: michal.sabo@stuba.sk

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

Ing. Peter Trebatický, PhD.
E-mail: trebaticky@fiit.stuba.sk

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

prof. RNDr. Jozef Kelemen, DrSc. (*1951)

Predhovor

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

Na Slovensku tieto dva odbory majú už svoje študijné programy: umelá inteligencia v Košiciach na FEI TU a kognitívna veda v Bratislave na FMFI UK. Už desať rokov sa na česko-slovenskej, alebo slovensko-českej konferencii *Kognícia a umelý život* stretávajú odborníci z rôznych vedných oblastí (informatici, matematici, filozofi, psychológovia, lekári,...), ktorých zjednocuje záujem o poznanie kognitívnych procesov, prebiehajúcich v ľudskom mozgu a o ich simuláciu metódami modernej umelej inteligencie: neurónových sietí, evolučných algoritmov a multiagentových systémov. Táto kniha vychádza pri príležitosti konania konferencie Kognícia a umelý život XI, ktorá sa v tomto roku 2011 konala v kongresovom centre SAV na Smolenickom zámku.

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

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

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

Túto knihu venujeme nášmu kolegovi a priateľovi prof. RNDr. Jozefovi Kelemenovi, DrSc. (∗1951), vysokoškolskému pedagógovi a vedcovi na Sliezskej univerzite v Opave a Vysokej škole manažmentu v Bratislave, ktorý u nás na prelome 80-90 rokov spoluzakladal informatický odbor "umelá inteligencia" a patrí k "spiritus movens" našej spoločnej česko-slovenskej konferencie *Kognícia a umelý život*.

V Bratislave, marec 2011.

Vladimír Kvasnička
Jiří Pospíchal
Pavol Návrat
Peter Lacko
Ľubomír Varga
(editori)

Obsah

Predhovor

(1) Modely P kolonií 1
Luděk Cienciala, Lucie Ciencialová
1 Úvod 1
2 Membránové systémy 2
3 P kolonie 3
4 P kolonie s vývojem prostředí 9
5 P kolonie se vstupní páskou 13
6 Závěr 17
Literatura 18

(2) Konekcionizmus v náručí výpočtovej kognitívnej vedy 19
Igor Farkaš
1 Úvod 19
2 Paradigmy kognitívnej vedy 21
3 Počítanie z pohľadu paradigiem 27
4 Podobnosti a rozdiely medzi paradigmami 30
5 Neurónové siete v kognitívnej vede 37
6 Záver 53
Literatúra 54

(3) Na ceste k hľadaniu vedy o mysli 63
Silvia Gáliková
1 Úvod 63
2 Od démonov k fontánam a humanoidným robotom 66
3 Hmota, duch alebo ementál 69
4 Transparentná myseľ 73
5 Záver 76
Literatúra 78

(4) Remaking Robot. Od mýtu technického pokroku k subverzním strategiím posthumanismu 81
Jana Horáková
1 Robot jako remake 81
2 Remaking Robot 86
3 Remaking robot in/to Myth 90
4 Remaking robot in/to Art 92
5 Závěr 95

Literatura 96

(5) Fuzzy generatívne systémy 99
Alica Kelemenová
1 Jazyky a fuzzy jazyky 99
2 Gramatiky a fuzzy gramatiky 100
3 L systémy a fuzzy L systémy 106
4 Gramatické systémy a ich fuzzy varianty 110
5 Záver 115
Literatúra 116

(6) Štúdium doskových hier pomocou neurónových sietí 117
Peter Lacko, Vladimír Kvasnička
1 Úvod 117
2 Počítačové hranie hier 118
3 Formalizácie hry 121
4 Porovnanie prístupov umelej inteligencie 124
5 Učenie s odmenou a trestom 131
6 Adaptácia dopredných neurónových sietí 134
7 Adaptívna kombinácia lokálnych expertných neurónových sietí 144
8 Použitie rozšíreného Kalmanovho filtra na učenie neurónových sietí 150
9 Multi-dimenzionálne rekurentné neurónové siete 156
10 Záver 158
Literatúra 159

(7) Multiagentový prístup k modelovaniu mysle - alebo ako sledovať pingpongovú loptičku 163
Andrej Lúčny
1 Úloha 163
2 Klasický prístup k riadeniu robota 164
3 Potreba paralelizmu a výmeny dát 166
4 Multiagentový prístup 168
5 Teoretické pozadie multiagentového prístupu 179
6 Záver 181
Literatúra 182

(8) Kognitívne vylepšovanie človeka, evolúcia a etika 185
Elena Matějková a Peter Sýkora
1 Úvod (kognitívne vylepšovanie je vylepšovanie vylepšovania) 185
2 Kognitívne vylepšovanie človeka - taxonómia 188
3 Kybernetické kognitívne vylepšovanie (neuroimplantáty) 189
4 Chemické kognitívne vylepšovanie (chytrolieky, *smart drugs*) 190
5 Rozlišovanie medzi terapiou a vylepšovaním 192
6 Vylepšovanie kognitívnych schopností ako forma podvádzania 194
7 Genetické vylepšovanie kognitívnych schopností 196
8 Transhumanizmus, posthumanizmus 197
9 Je ľudská biologická podstata posvätná? 198
10 Záver 202
Literatúra 202

(9) Ambientní inteligence - očekávání, aplikace a rizika 205
Peter Mikulecký
1 Úvod 205
2 Co to je ambientní inteligence? 206
3 Multi-agentové architektury v ambientní inteligenci 209
4 Některé aplikační možnosti 214
5 Některé přednosti a rizika 216
6 Závěr 217
Literatura 218

(10) O zložitosti merania zložitosti 221
Jiří Pospíchal
1 Úvod 221
2 Zložitosť opisu 224
3 Zložitosť vytvorenia 226
4 Zložitosť organizácie 227
5 Záver 228
Literatúra 228

(11) Modely a metódy skúmania kognitívneho vývinu 231
Ján Rybár
1 Úvod 231
2 Konštruktivistický model kognitívneho vývinu 232
3 Nativistický model kognitívneho vývinu 237

4 Záver 241
Literatúra 242

(12) Paralelné evolučné algoritmy 243
Ivan Sekaj, Michal Oravec
1 Úvod 243
2 Evolučné algoritmy a genetický algoritmus 244
3 Paralelné evolučné a genetické algoritmy 252
4 Záver 266
Literatúra 267

(13) Membránové výpočty jako metafora informačních toků v organismech 269
Petr Sosík
1 Úvod 269
2 Základní definice 271
3 Komunikující membránové systémy 273
4 P systémy s aktivními membránami 279
5 Pulsní neuronové P systémy 287
6 Závěr 291
Literatura 292

(14) Neurčitosť, fuzzy množiny, aplikácie, problémy 297
Michal Šabo
1 Úvod 297
2 Fuzzy množiny a operácie s fuzzy množinami 298
3 Fuzzy implikácie 304
4 Klasické a fuzzy preferenčné štruktúry 306
5 Fuzzy regulátory 312
6 Záver 323
Literatúra 323

(15) Teória Kalmanových filtrov a neurónové siete 325
Peter Trebatický
1 Úvod 325
2 Dynamický systém 326
3 Kalmanov filter 326
4 Neurónová sieť ako dynamický systém 334
5 Rozšírený Kalmanov filter 338
6 „Unscented“ Kalmanov filter 341
7 Filter nprKF 345

8 Zdieľané Kalmanove filtre 349
9 Porovnanie jednotlivých metód 353
10 Záver 357
Literatúra 358

(16) Jak by mohly bakterie počítat 361
Jiří Wiedermann
1 Motivace 361
2 Model nanomašinek 364
3 Distribuované výpočty v populacích nanomašinek 366
4 Řízení nanomašinek pomocí obvodů 368
5 Závěr 369
Literatura 370

(17) Stvořitel. Tady a Teď 373
Jan Burian

Index 385

Modely P kolonií

Luděk CIENCIALA[1]

Lucie CIENCIALOVÁ[2]

Abstrakt. V našem příspěvku prezentujeme výsledky získané při výzkumu P kolonií. P kolonie je výpočetní model inspirovaný biochemií buněčné struktury a je založený na nezávislých membránových agentech žijících ve společném prostředí. Popíšeme základní model P kolonií a jeho vlastnosti. Základní model rozšíříme o mechanismy umožňující měnit prostředí respektive o možnost řídit vývoj P kolonie pomocí vstupní pásky.

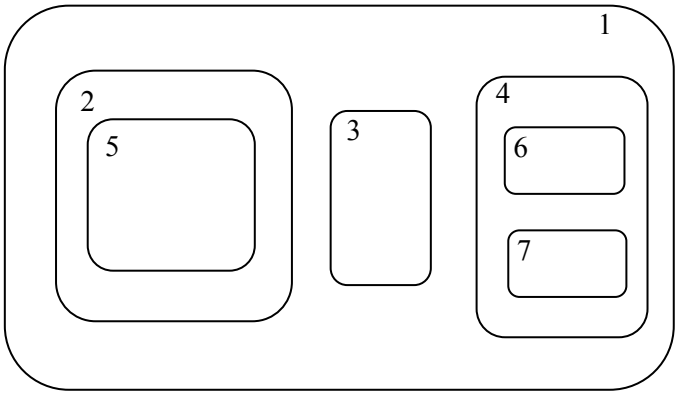
1 Úvod

Buňka je rozměrem malá, ale svým složením a funkcí naprosto jedinečná část živé hmoty. Má schopnost metabolismu, rozmnožování, komunikace s okolními buňkami i s okolním prostředím, ve kterém žije. Buňka je oddělena od svého prostředí pomocí plazmové membrány a je vnitřně rozdělena vnitřními membránami. Membrány tedy tvoří hranice všech organel i samotných buněk a vymezují oblasti, ve kterých se odehrávají biochemické procesy a umožňují více méně kontrolovanou výměnu látek s okolím. Membrány mají schopnost řídit výměnu látek s okolím, ať už na základě pouhé velikosti vlastních molekul, nebo pomocí principu aktivního transportu přes proteinové brány, které nejen vybírají vhodné molekuly, ale propouští je ve směru větší koncentrace látek dovnitř nebo ven. Samotná buňka je schopna řídit svůj život na základě prostředí, ve kterém žije. V jejím spojení s ostatními buňkami v koloniích, orgánech, organismech se skrývá schopnost přizpůsobit se a schopnost komunikovat jak s okolními buňkami, tak s vnějším prostředím. Membránové systémy jsou výpočetní modely inspirované právě biochemií buněčné struktury. Pro membránové systémy se po roce 2000 používá také název P systémy.

[1] Ústav informatiky FPF SLU, E-mail: ludek.cienciala@fpf.slu.cz
[2] Ústav informatiky FPF SLU, E-mail: lucie.ciencialova@fpf.slu.cz

2 Membránové systémy

Membránové systémy byly zavedeny v roce 1998 Gheorghem Păunem v práci [11] jako membránové struktury skládající se z membrán hierarchicky vnořených do hlavní membrány. Každá membrána vymezuje určitou oblast, ve které se mohou vyskytovat další membrány. Membránu neobsahující další membrány nazýváme elementární. Membránovou strukturu můžeme znázornit v grafické formě a to pomocí Vennových diagramů, ve kterých je každá dvojice membrán disjunktní nebo je jedna podmnožinou druhé. Příklad membránové struktury znázorněné pomocí Vennových diagramů je na obrázku 1.

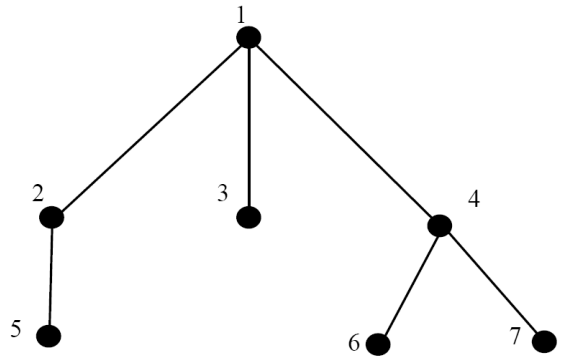


Obr. 1. Znázornění membránové struktury pomocí Vennových diagramů

Formálně membránovou strukturu zapisujeme dvojicemi závorek tak, že každý pár závorek charakterizuje umístění membrány v hierarchicky uspořádaném membránovém systému. Každá dvojice závorek je popsána číslem dané membrány. Struktura znázorněná na obrázku 1 má následující formální zápis: $[_1 [_2 [_5]_5]_2 [_3]_3 [_4 [_6]_6 [_7]_7]_4]_1$.

Membránovou strukturu můžeme reprezentovat i pomocí grafové reprezentace. V grafové interpretaci membránová struktura odpovídá grafu, který nazýváme strom. Hlavní membrána odpovídá kořenu a elementární membrány představují listy stromu. Membránová struktura z obrázku 1 v grafové interpretaci je znázorněna na obrázku 2. Na začátku výpočtu P systému oblasti ohraničené membránami obsahují multimnožiny objektů a také pravidla. Právě podle těchto pravidel se mění obsah jednotlivých oblastí a dané změny určují výpočet systému.

Existují různé varianty P systémů v závislosti na způsobu reprezentace objektů, na základě typů pravidel, způsobu formalizace vývoje systému, popisu výpočtu a výsledku výpočtu. Membránové systémy patří mezi nedeterministické systémy, jejichž výpočet probíhá na bázi stanovených pravidel v diskrétních krocích a paralelně pro všechny objekty. Existují i různé varianty P systému v závislosti na specifikaci rozsahu paralelismu.



Obr. 2. Znázornění membránové struktury pomocí grafu

Výpočet P systému je určen aplikací pravidel v diskrétních krocích, paralelně pro objekty v oblastech. Výpočet začíná počáteční specifikací obsahu jednotlivých oblastí membrán P systémů. Výsledkem výpočtu může být obsah předem určené výstupní membrány, a to buď všechen, nebo tvořený speciálními výstupními symboly, kdy se systém nachází v koncové konfiguraci. Výsledek může být také multimnožina objektů vstupující do systému během úspěšného výpočtu. Výpočet v P systémech má typické charakteristiky generativního formalismu přepisu v gramatikách (systémech gramatik).

V této práci se budeme věnovat jednomu z modelů membránových systémů a to P koloniím.

3 P kolonie

P kolonie jsou výpočetní model spojující dvě oblasti teoretické informatiky - kolonie a membránové systémy. P kolonie byly zavedeny v roce 2004 v práci [9].

Daný model je inspirován strukturou a činností živých organismů žijících ve společném prostředí. Příkladem mohou být například mravenci, jejichž nejůležitějším komunikačním prostředkem jsou mravenčí pachy (feromony). I přes malý mozeček díky dané komunikaci jsou schopni vytvořit poměrně složité společenské struktury. Druhým příkladem mohou být včely. Žihadlový aparát včely tvoří několik žláz produkujících například histamin, fosfolipázu, hyaluronidázu, které vyvolávají pocity svědění, pálení v okolí vpichu. Tyto žlázy vylučují i feromony sloužící k vyvolání poplachu u ostatních včel. Do rány po bodnutí vstřikují včely látku, která informuje ostatní včely o místě bodnutí, a vpichují další žihadla do blízkosti daného vpichu. Včely si značí i pomocí pachu cestu k nektaru. Samozřejmě dané látky pomáhají včelám najít cestu zpátky do úlu. Můžeme tedy říci, že včely a mravenci pachy používají jako dorozumívací jazyk.

Pokusme se nejdříve P kolonie definovat neformálně. Zde si s trochou nadsázky pod pojmem agent můžeme představit právě např. mravence nebo včelu. Nemáme za cíl vytvořit formální model popisující společenství mravenců či včel.

3.1 P kolonie neformálně

Do prostředí sloužící jako komunikační kanál i „skladiště“ objektů umístíme jednoduché membránové systémy nazývané agenty. Agenty představují nezávislé organismy žijící v P koloniích. Agenty a jejich životní projevy jsou definovány jednoduchým způsobem. Každý agent je tvořen jednou membránou ohraničující oblast s objekty. Počet objektů uvnitř agenta je pevně dán a je stejný pro všechny agenty. Počet objektů uvnitř každého agenta určuje tzv. kapacita P kolonie.

Každý agent má k dispozici svou množinu programů, které jsou tvořeny pravidly. Pravidlo může být přepisující, komunikační a řídící. Program určuje činnost agenta. Každý program obsahuje tolik pravidel, kolik je objektů uvnitř agenta. V každém okamžiku jsou všechny objekty uvnitř agenta změněny nebo přesunuty.

P kolonie má tři typy pravidel: (1) Přepisující pravidla $a \rightarrow b$. Aplikací pravidla je objekt a v agentu přepsán na objekt b. (2) Komunikační pravidla mají tvar $c \leftrightarrow d$. Pomocí tohoto pravidla je objekt c, který je uvnitř agenta, přesunut ven a objekt d, který je vně, je přesunut dovnitř agenta. (3) Řídicí pravidla mají tvar $c \leftrightarrow d/c' \leftrightarrow d'$ a dávají agentům možnost výběru mezi dvěma možnostmi. Pokud je toto pravidlo aplikováno, má pravidlo $c \leftrightarrow d$ vyšší prioritu k provedení než pravidlo $c' \leftrightarrow d'$. To znamená, že agent chce nejdříve aplikovat pravidlo $c \leftrightarrow d$. Pro aplikování pravidla musí být uvnitř agenta objekt c a v prostředí objekt d. Pokud dané pravidlo může být aplikováno, tak je aplikováno. Pokud ne, tak agent použije druhé pravidlo, tzn. pravidlo $c' \leftrightarrow d'$.

Pokud bychom hledali interpretaci daných pravidel v přírodě, potom přepisující pravidla umožňují změnu látky a uvnitř organismu na jinou látku b. Komunikační pravidla slouží ke komunikaci organismů pomocí prostředí. Látka c je vyloučena do prostředí a látka d je přijata organismem. Látka d může sloužit jako zpětná vazba při vyloučení látky c. Pomocí řídících pravidel agent (organismus) zjišťuje, jestli se v prostředí vyskytuje určitý typ látky. Pokud se daná látka nevyskytuje, přizpůsobí své chování tomuto faktu. Například mravenec pokud najde feromonovou stopu, tak ji bude sledovat, pokud ji ztratí, bude ji hledat.

Výpočet začíná v počáteční konfiguraci, která je definována následujícím způsobem: prostředí a všechny agenty obsahují pouze kopie objektů e, kde e je speciální typ objektu, který nazýváme environmentální. V přírodě si můžeme představit pod environmentálním objektem například vzduch nebo vodu. Objektů e je dostatečné množství. Tzn., že nikdy nenastane situace, že by objekt e pochyběl. Aplikováním programů agenty mohou měnit svůj obsah a pomocí prostředí mohou ovlivňovat chování ostatních agentů v dalších krocích výpočtu. To znamená, že stejně jako v případě mravenců a včel se prostředí využívá jako komunikační kanál. V každém kroku výpočtu každý agent nedeterministicky vybere jeden ze svých aplikovatelných programů a vykoná jej. Výpočet končí zastavením, kdy žádný agent nemůže aplikovat žádný ze svých programů. P kolonie patří do skupiny P systémů, kdy výsledkem výpočtu je počet určitých objektů v prostředí na konci výpočtu.

Samotný výpočet u P kolonií může probíhat dvěma způsoby a to paralelně nebo sekvenčně. V každém kroku výpočtu se každý agent „snaží najít“ program, který je aplikovatelný. Jestliže najde více než jeden, pak náhodně vybere právě jeden z nich a

použije ho. V každém kroku paralelního výpočtu pracuje maximální počet agentů. V každém kroku sekvenčního výpočtu je ze všech agentů mající aplikovatelný program vybrán jen jeden, který bude pracovat.

Z důvodu nedeterminismu v průběhu výpočtu můžeme získat několik výpočtů končících zastavením.

P kolonie jsou výpočetně úplné. Zajímavá je otázka, jaký je nejmenší počet agentů, počet programů v agentu při zachování výpočetní úplnosti. P kolonie se dvěma objekty jsou studovány v pracích [8, 9, 10]. Jsou zde také uvedeny jiné varianty P kolonií vycházející ze změny počtu objektů uvnitř agentů. Výsledky pro P kolonie se třemi objekty uvnitř každého agenta nalezneme v práci [10]. Ukázali jsme, že i P kolonie s jedním objektem v agentu jsou výpočetně úplné. P kolonie s kapacitou jedna s řídícími pravidly s nejvýše sedmi programy jsou výpočetně úplné a s řídícími/přepisujícími programy s počtem agentů *4* mají stejnou výpočetní sílu. P kolonie se dvěma objekty uvnitř agenta a s programy se stejným typem pravidel a s jedním agentem jsou výpočetně úplné.

Pokud budeme uvažovat, že P kolonie mají programy, které jsou tvořeny stejným typem pravidel (přepisujícími, komunikačními nebo řídícími) pro všechny objekty uvnitř agent, nazýváme P kolonii homogenní. Každá P kolonie s kapacitou jedna je homogenní. Homogenní P kolonie byly poprvé studovány v práci [5] a bylo dokázáno, že pro zachování výpočetní úplnosti stačí sedm programů u každého z agentů. Také i pět agentů zabezpečuje výpočetní úplnost P kolonií. V práci [1, 2] jsme ukázali, že šest programů v každém agentu bez omezení počtu agentů určuje, aby P kolonie byla výpočetně úplná. Další výsledek týkající se homogenních P kolonií se dvěma objekty v každém agentu je, že dva objekty v agentech dovolují maximálně snížit počet agentů tak, že výpočetní úplnost může být realizována pouze jedním agentem. Navíc čtyři programy v každém agentu dovolují generovat každou spočitatelnou podmnožinu přirozených čísel (bez omezení počtu agentů).

3.2 P kolonie

V této části definujeme P kolonie formálněji.

P kolonie je tvořena agenty a prostředím. Agenty i prostředí obsahují objekty. Každý agent má množinu programů. Program může obsahovat tři typy pravidel. První typ nazýváme přepisující pravidla, mají tvar $a \rightarrow b$. Druhý typ nazýváme komunikační pravidla, mají tvar $c \leftrightarrow d$. Třetí typ jsou řídící pravidla, mají tvar $c \leftrightarrow d/c' \leftrightarrow d'$. Zavedením řídících pravidel získaly agenty schopnost vybrat si mezi dvěma možnostmi. Pravidla mají tvar r_1/r_2. Pokud je aplikováno řídící pravidlo, pravidlo r_1 má vyšší prioritu pro vykonání než pravidlo r_2. To znamená, že agent, pokud je to možné, aplikuje pravidlo r_1, v opačném případě pravidlo r_2.

Pokud programy mají jeden typ pravidel, pak hovoříme o programech přepisujících, komunikačních nebo řídících. V případě P kolonií se dvěma objekty uvnitř každého agenta přepisující programy mohou být modifikovány na tvar $<ab \rightarrow cd>$. Stejným způsobem můžeme modifikovat komunikační programy na tvar $<ab \leftrightarrow cd>$.

Počet objektů uvnitř každého agenta určuje tzv. kapacita P kolonie a budeme ji označovat c.

Definice 1 P kolonie s kapacitou c je struktura $\Pi = (A, e, f, V_E, B_1, ..., B_n)$, kde

- A je abeceda kolonie, její prvky nazýváme objekty,
- $e \in A$ je environmentální objekt P kolonie,
- $f \in A$ je finální objekt P kolonie,
- V_E je multimnožina objektů nacházející se v prostředí na počátku výpočtu, definuje počáteční stav prostředí,
- B_i, $1 \leq i \leq n$, jsou agenty, B_i je struktura $B_i = (O_i, P_i)$, kde
  - O_i je multimnožina c kopií environmentálního objektu e, tato multimnožina definuje počáteční stav agenta,
  - $P_i = \{p_{i,1},...,p_{i,k_i}\}$ je konečná množina programů, každý program obsahuje c pravidel v jednom z následujících tvarů:
    - $a \rightarrow b$, pravidla v daném tvaru nazýváme přepisující pravidla,
    - $c \leftrightarrow d$, pravidla v daném tvaru nazýváme komunikační pravidla,
    - $c \leftrightarrow d/c' \leftrightarrow d'$, pravidla v daném tvaru nazýváme řídící pravidla.

Na začátku výpočtu prostředí obsahuje potřebné množství (i nekonečně mnoho) kopií environmentálního objektu e, stejně tak i agenty obsahují pouze kopie tohoto objektu, každý agent obsahuje právě c těchto objektů. P kolonie se nachází ve své počáteční konfiguraci.

Konfigurace je $(n + 1)$-tice řetězců objektů $(w_1, ..., w_n, w_E e^{\omega})$, kde $|w_i| = c$, $1 \leq i \leq n$, w_i reprezentuje všechny objekty uvnitř i-tého agenta a $w_E \in (A - \{e\})^*$ je souhrn všech objektů, které jsou obsaženy v prostředí a jsou různé od e.

Výpočet P kolonie může probíhat paralelně nebo sekvenčně podle toho, jestli pracuje maximální počet agentů anebo právě jeden agent.

Výpočet končí zastavením, což je stav, kdy ani jeden z agentů nemá aplikovatelný program. Pokud výpočet končí zastavením, výsledek výpočtu je dán počtem finálních objektů f obsažených v prostředí.

Výpočet probíhá nedeterministicky, proto je více možností, kdy svůj výpočet P kolonie ukončí zastavením. S každou P kolonií Π je tedy spojena množina přirozených čísel určující výsledek výpočtu. Označíme ji $N(\Pi)$.

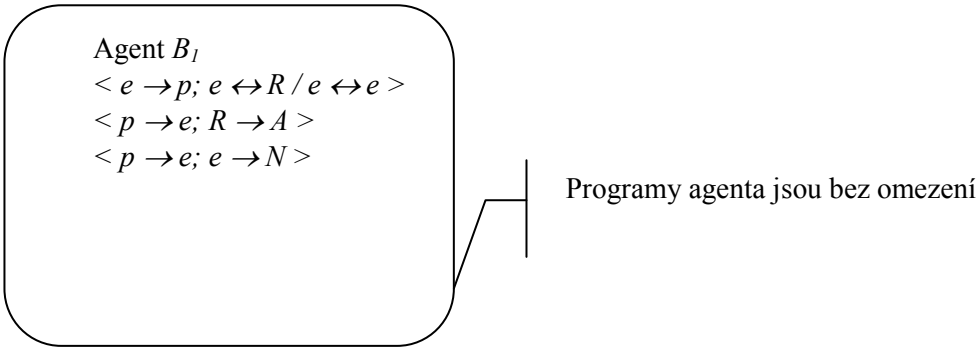
Mějme P kolonii $\Pi = (A, e, f, V_E, B_1, ..., B_n)$, maximální počet programů asociovaných s agenty v P kolonii nazýváme výškou P kolonie. Stupeň P kolonie je pak definován jako počet agentů v P kolonii. Třetím parametrem je kapacita popisující počet objektů uvnitř každého agenta.

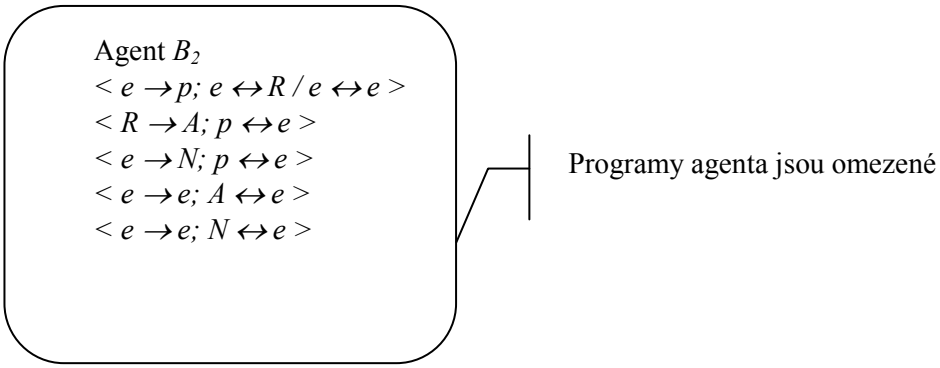
Třídu všech množin čísel $N(\Pi)$ počítaných P koloniemi s kapacitou c, stupně nejvýše n a s výškou nejvýše h bez použití řídících pravidel značíme $NPCOL_{par}(c, n, h)$ pro P kolonie pracující paralelně a pro P kolonie pracující sekvenčně $NPCOL_{seq}(c, n,$

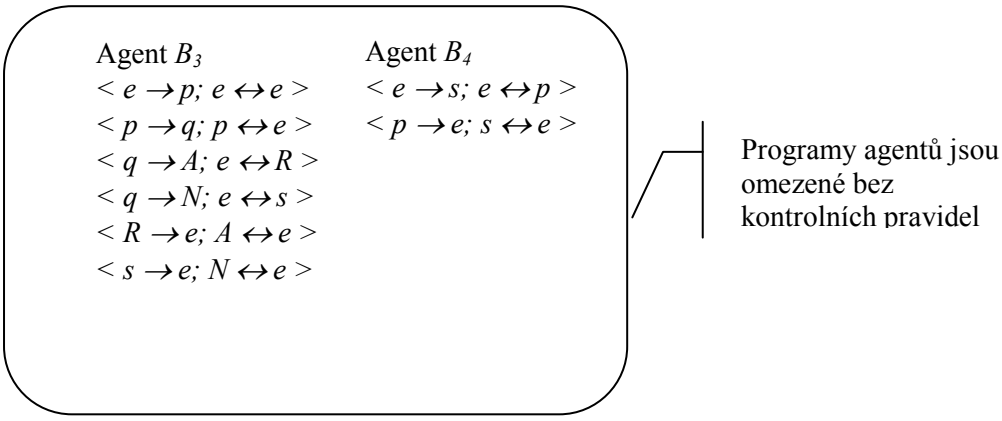
h). Pokud připustíme řídící pravidla, pak třídu všech množin přirozených čísel počítaných P koloniemi označíme $NPCOL_mK$.

V práci [9] jsou zavedeny P kolonie pouze se dvěma objekty uvnitř agenta. Každý program tedy obsahuje jedno pravidlo přepisující a jedno pravidlo komunikační, případně řídící. P kolonie se dvěma objekty uvnitř agentů a výše uvedenými pravidly nazýváme omezené. Pokud jsou kolonie také omezené, změníme notaci na $NPCOL_mR$ nebo *NPCOLmKR*, kde *m* je s*eq* nebo *par*.

Příklad 1 Funkci P kolonie si ukážeme na jednoduchém zadání. Vytvoříme P kolonii s kapacitou *2*, která řeší jednoduchou úlohu. Jedná se o simulaci *SUB*-instrukce registrového stroje, tedy odečtení jedničky z registru. Pokud je odečtení úspěšné, probíhá program jinak, než v případě kdy je odečtení neúspěšné a tedy registr obsahoval nulu. P kolonie bude mít za úkol odebrat z prostředí objekt *R*, pokud bude odebrání úspěšné (prostředí obsahovalo alespoň jednu kopii tohoto objektu), P kolonie místo něj umístí do prostředí objekt *A*, v opačném případě P kolonie vygeneruje objekt *N*.







Na uvedeném příkladě je vidět, jak se omezení kladená na programy odráží v počtu programů, provedených kroků a počtu agentů.

3.3 P kolonie s jedním objektem uvnitř agenta

V této části uvedeme vlastnosti P kolonií pouze s jedním objektem uvnitř každého agenta „žijícího“ v kolonii. Tzn., že každý program agenta je tvořen pouze jedním pravidlem. Dané pravidlo je přepisující, komunikační nebo řídící.

V případě agentů, které mají programy tvořené přepisujícími pravidly, říkáme, že agenty „žijí samy pro sebe“ a nekomunikují s prostředím. Daný agent pouze mění objekty, které obsahuje.

V důkazech vět určujících výpočetní sílu P kolonií s jedním objektem se využívá simulace registrového stroje P kolonií. Důkazy uvedených vět naleznete v [4, 5].

První výsledek říká, že P kolonie s kapacitou $c = 1$ a výškou h nejvýše 7 (tzn., že počet programů asociovaných s agenty je nevýše 7) bez omezení počtu agentů je výpočetně úplná.

Věta 1 $NPCOL_{par}K(1, *, 7) = NRE$.

Další věta určuje výpočetní sílu P kolonie s kapacitou $c = 1$ a počtem agentů 4 bez omezení výšky kolonie. P kolonie s danými parametry je opět výpočetně úplná.

Věta 2 $NPCOL_{par}K(1, 4, *) = NRE$.

3.4 P kolonie se dvěma objekty uvnitř agenta

Agenty se dvěma objekty mají v každém programu dvě pravidla. Pokud programy obsahují po jednom pravidle přepisujícím a jednom komunikačním nebo řídícím pravidle, nazýváme danou P kolonii omezenou.

Jestliže budeme uvažovat i jiné kombinace těchto tří typů pravidel, pak daná P kolonie není omezená.

Omezené P kolonie s řídícími pravidly jsou výpočetně úplné. Platí:

$NPCOL_{par}KR(2, *,5) = NRE$ v [7, 9],

$NPCOL_{seq/par}KR(2, 1, *) = NRE$ v [8],

$NPCOL_{seq}KR(2, 1, *) = NMAT$ v [8],

$NPCOL_{seq}KR(2, *, *) = NMAT$ v [8], kde *NMAT* je množina vektorů nezáporných čísel generovaných pomocí maticových gramatik bez kontroly výskytu.

Následující věta říká, že výpočetní síla P kolonie s kapacitou 2 a s jedním agentem je *NRE*. Důkaz této věty naleznete v [4].

Věta 3 $NPCOL_{par/seq}K(2, 1, *) = NRE$.

V této části jsme uvedli výsledky P kolonií, které mají uvnitř dva objekty. Nyní původní model rozšíříme o možnost vývoje prostředí.

4 P kolonie s vývojem prostředí

U původního modelu mají P kolonie prostředí statické a slouží pouze jako komunikační kanál pro organismy v P kolonii. Eko-P kolonie umožňují rozšířit daný model P kolonií o možnost měnit prostředí nezávisle na daných organismech, agentech. Nejčastěji se používají eko-P kolonie s kapacitou *2*, tzn., každý agent bude obsahovat dva objekty. Mechanismem umožňující měnit prostředí bude *0L* schéma.
Pod změnou prostředí si můžeme v přírodě představit například změnu teploty, jestli prší, svítí slunce apod. Vraťme se zpátky k mravencům a včelám a na to, jaký vliv dokáže mít nezávislé prostředí na jejich chování. Například v chladnějších obdobích roku za slunečného počasí mravenci hromadně vycházejí na povrch mraveniště, kde svá těla nahřívají na sluníčku a následně pak zamíří do vnitřku mraveniště, kde teplo vyzáří. Tímto počínáním jsou schopni zvýšit teplotu v mraveništi až o 10 stupňů. Prostředí také ve velké míře ovlivňuje chování včel. Včely mohou létat při dosažení určité tělní teploty, která je závislá na teplotě prostředí. Tato teplota se může lišit nejen mezi druhy, ale také mezi jedinci stejného durhu. Aktivita včel závisí na teplotě, slunečním záření, relativní vlhkosti vzduchu, rychlosti větru.

Protože daný model vychází z P kolonií, je i tady činnost agenta určena pomocí programů. Každý agent má k dispozici svou množinu programů. Program je tvořen jedním pravidlem přepisujícím a jedním pravidlem transportním. Přepisující pravidla mohou být generující nebo mazající. Generující pravidlo z jednoho objektu vytvoří dva objekty, je spojeno s transportním pravidlem, které přesunuje objekty do prostředí. Mazající pravidlo tvoří ze dvou objektů jeden a zase je spojeno s transportním pravidlem, které přesunuje objekt z prostředí do agenta. Programy obsahující mazající

pravidla nazýváme konzumující programy a programy s generujícími pravidly nazýváme produkující programy. Agent, který obsahuje pouze konzumující programy, označujeme jako agenta konzumenta a agenta obsahujícího pouze produkující programy jako agenta producenta.

Konzumující programy mají tvar: $<ab \rightarrow c, d\ in>$. Aplikací prvního pravidla konzumujícího programu způsobí, že místo objektů *a* a *b* bude agent obsahovat objekt *c*. Druhé pravidlo, pravidlo transportní, přesune objekt *d* z prostředí dovnitř agenta. Uvnitř agenta se vyskytují vždy dva objekty. První pravidlo programu by nám snížilo počet objektů o jeden, proto pomocí druhého pravidla je přesunut objekt z prostředí do agenta a to v jednom okamžiku.

Produkující programy mají tvar $<a \rightarrow bc, d\ out>$. Aplikace prvního pravidla produkujícího programu způsobí, že místo objektu *a* bude agent obsahovat objekty *b* a *c*. Druhé pravidlo stejně jako u konzumujícího programu je pravidlem transportním, ale tentokrát přesune objekt *d* z agenta do prostředí z důvodu, aby agent obsahoval právě dva objekty.

Prostředí eko-P kolonie slouží stejně jako u P kolonie jako komunikační kanál. Přes prostředí jsou agenty schopny ovlivňovat chování ostatních agentů.

Prostředí se vyvíjí nezávisle na agentech. Agenty mohou částečně ovlivnit vývoj prostředí přesouváním objektů do prostředí. Vývoj prostředí nezávisí na stavech jednotlivých agentů v eko-P kolonii. Vývoj jednotlivých objektů probíhá pomocí bezkontextových pravidel a to paralelně, tzn., že jsou přepsány všechny objekty, které mohou být přepsány. Mechanismem vývoje prostředí je postaven na *0L* schématech.
V každém kroku výpočtu každý agent nedeterministicky vybere jeden ze svých aplikovatelných programů a vykoná jej. Výpočet končí zastavením, kdy žádný agent nemůže aplikovat žádný ze svých programů. V prostředí se mohou vyskytovat i nadále aplikovatelná pravidla, ale za výsledek výpočtu uvažujeme počet specifických objektů v okamžiku zastavení činnosti všech agentů v eko-P kolonii.

4.1 Eko-P kolonie

V této části uvedeme definici eko-P kolonie, kroku výpočtu a výsledek výpočtu eko-P kolonie.

Definice 2 Eko-P kolonie s kapacitou *2* je struktura
$\Pi = (A, e, f, V_E, D_E, B_1, ..., B_n)$, kde

- A je abeceda kolonie, její prvky nazýváme objekty,
- e je environmentální objekt, $e \in A$
- f je finální objekt kolonie, $f \in A$,
- V_E je multimnožina objektů nacházející se v prostředí na počátku výpočtu, definuje počáteční stav prostředí, $V_E \in (A - \{e\})^*$,
- D_E je *0L* schéma (A, P_E), kde P_E je množina bezkontextových pravidel.
- B_i, $1 \leq i \leq n$, jsou agenty, B_i je struktura $B_i = (O_i, P_i)$, kde
  - O_i je multimnožina nad A, která definuje počáteční obsah agenta B_i a $|O_i| = 2$,

- $P_i = \{p_{i,1},...,p_{i,k_i}\}$ je konečná množina programů, které mohou být:
  - generující $<a \to bc, d\ out>$ - program je aplikovatelný, pokud agent obsahuje objekty *a* a *d*.
  - konzumující $<ab \to c, d\ in>$ - program je aplikovatelný, pokud agent obsahuje objekty *a* a *b*.

Každý agent obsahuje pouze jeden druh programů.
Počáteční konfigurace eko-P kolonie je $(n + 1)$-tice $(O_1, ..., O_n, V_E)$ multimnožin objektů nacházejících se v eko-P kolonii na začátku výpočtu, kde O_i pro $1 \le i \le n$ a pro prostředí V_E. Obecně konfigurace kolonie je definovaná jako $(w_1, ..., w_n, w_E)$, kde w_i reprezentuje všechny objekty *i*-tého agenta, $|w_i| = 2$, $1 \le i \le n$, $w_E \in (A - \{e\})^*$ určuje objekty různé od *e* obsažené v prostředí.
Výpočet eko-P kolonií probíhá paralelně. V každém kroku paralelního výpočtu pracuje maximální počet agentů. Každý agent, pokud může použít nějaký ze svých programů, tak ho použije. Jestliže agent má více aplikovatelných programů v daném kroku výpočtu, potom nedeterministicky vybere právě jeden a použije ho.
Pro programy z každé množiny P_i zavedeme označení z množiny $lab(P_i)$ takové, že $lab(P_i) \cap lab(P_j) = \emptyset$ pro $i \ne j$, $1 \le i, j \le n$.
Pro formální vyjádření kroku výpočtu zavedeme čtyři funkce: *whatsin*, *demand*, *putout*, *newin*. Dané funkce každému programu přiřadí multimnožina objektů, které musí být v agentu a v prostředí, aby daný program mohl být proveden. Funkce definují i objekty, které se budou nacházet uvnitř agenta a objekty, které obohatí prostředí po úspěšném provedení programu. Formálně, nechť $<ab \to c, d\ in>$ je konzumující program a $<a \to bc, d\ out>$ je generující program, pak definujeme funkce:

$whatsin(p_k) = \{ab\}$ pro konzumující program p_k
$whatsin(p_k) = \{ad\}$ pro generující program p_k
$demand(p_k) = \{d\}$ pro konzumující program p_k
$demand(p_k) = \emptyset$ pro generující program p_k
$newin(p_k) = \{cd\}$ pro konzumující program p_k
$newin(p_k) = \{bc\}$ pro generující program p_k
$putout(p_k) = \emptyset$ pro konzumující program p_k
$putout(p_k) = \{d\}$ pro generující program p_k

Změna konfigurace je definována jako: $(w_1, ..., w_n, w_E) \Rightarrow (w'_1, ..., w'_n, w'_E)$, kde jsou splněny následující podmínky:

- Množina označení programů P s $|P| \le n$ taková, že
  - $p, p' \in P, p \ne p', p \in lab(P_j), p' \in lab(P_i), i \ne j$
  - pro každé $p \in P, p \in lab(P_j), whatsin(p) = w_j, \bigcup_{p \in P} demand(p) \subseteq w_E$.
- Kromě toho, vybraná množina P je maximální, tzn., že žádný další program $r \in \bigcup_{1 \le i \le n} lab(P_i)$, $r \notin P$, nemůže být přidán do množiny P tak, aby byly splněné uvedené podmínky.

Obecně pro j, $1 \leq j \leq n$, pro které existuje $p \in P$, takové že $p \in lab(P_j)$ nechť $w'_{j'} = newin(p)$. Pokud neexistuje $p \in P$, $p \in lab(P_j)$ pro nějaké j, $1 \leq j \leq n$, nechť $w'_j = w_j$. Nechť $w_E - \bigcup_{p \in P} demand(p) \subseteq_{D_E} w''_E$ je krok odvození pro $0L$ schéma (A, P_E) a pak $w'_E = w''_E \bigcup \bigcup_{p \in P} putout(p)$.

$\bigcup$ a – zde představují operace nad multimnožinami.

Konfigurace je koncová, pokud množina označení programů P splňující dané podmínky, nemůže být vybrána jinak, než jako prázdná množina. Množinu koncových konfigurací označíme H. Jestliže se výpočet zastaví, můžeme s ním asociovat výsledek výpočtu. Výsledkem výpočtu je počet kopií speciálního symbolu f v prostředí. Množina čísel počítaných P kolonií Π je definována jako $N(\Pi)=\{ /w_E/_f \mid (O_1, ..., O_n, V_E) \Rightarrow^* (w_1, ..., w_n, w_E) \in H\}$, kde $(O_1, ..., O_n, V_E)$ je počáteční konfigurace, $(w_1, ..., w_n, w_E)$ je koncová konfigurace, $\Rightarrow^*$ označuje reflexivní a tranzitivní uzávěr $\Rightarrow$.

Mějme eko-P kolonii $\Pi = (A, e, f, V_E, D_E, B_1, ..., B_n)$. Maximální počet programů asociovaných s agenty nazýváme výškou h, počet agentů n, pak stupněm n eko-P kolonie.

Označme $NEPCOL_{x,y,z}(n,h)$ třídu množin čísel počítaných eko-P koloniemi takovými, že:

- x obsahuje s, pokud eko-P kolonie obsahuje agenty producenty,
- x obsahuje c, pokud eko- P kolonie obsahuje agenty konzumenty,
- $y = passive$, pokud $0L$ schéma disponuje pouze pravidly typu $a \rightarrow a$,
- $y = active$, pokud $0L$ schéma disponuje alespoň jedním pravidlem jiného typu než $a \rightarrow a$.
- $z = ini$, pokud prostředí nebo agenti na začátku prostředí obsahují jiné objekty než environmentální, jinak toto značení vynecháme,
- stupněm nejvýše n,
- výškou nejvýše h.

4.2 Chování agentů

Agent konzument se vyvíjí pomocí programu: $<ab \rightarrow c, d\ in>$. Agent s těmito programy čte z okolí objekty a zpracovává svůj obsah na základě toho, jaký objekt přijal. To je velmi podobné činnosti konečného automatu. Výsledkem výpočtu automatu je přijetí, či nepřijetí slova. Výsledek výpočtu eko-P kolonie je počet symbolů f nacházejících se v prostředí po ukončení aktivity agenta. Oba výpočetní modely se liší. Konečný automat má vstupní objekty na pásce a jsou určitým způsobem uspořádány. Na rozdíl eko-P kolonie nemá objekty uspořádané. Eko-P kolonie pracuje s permutacemi daných objektů.

Agent producent se vyvíjí pomocí programu: $<a \rightarrow bc, d\ out>$. Chování agenta by se dalo popsat podobně jako chování gramatik. Agent v každém kroku jeden objekt přesune do prostředí a z druhého objektu vytvoří dva nové. Pravidlo pro generování objektů v agentu je podobné pravidlům v regulární gramatice. Liší se výsledkem. Agent generuje objekty, ale jen ty finální mohou být započítány do výsledku, který je číslem. Časová souslednost generovaných objektů však již odpovídá slovům

generovaných gramatikou (o jeden krok zpožděno), protože v jednom kroku daný objekt vytvoří a teprve v dalším kroku vytvořený objekt přesune do prostředí. Výpočet eko-P kolonie se zastaví použitím programu, který odpovídá aplikování pravidla $A \rightarrow a$ v regulární gramatice.

4.3 Vlastnosti eko-P kolonií

Nyní uvedeme vlastnosti eko-P kolonií. V práci [6] autoři dokázali, že:

Věta 4 $NEPCOL_{sc,passive}(3, *) = NRE$.

Dokázali jsme, že eko-P kolonie s aktivním prostředím a dvěma agenty konzumenty generují rekurzivně spočetnou množinu přirozených čísel. V důkaze jsme využili simulace registrového stroje.

Věta 5 $NEPCOL_{c,active,ini}(2, *) = NRE$.

Eko-P kolonie s aktivním prostředím a dvěma agenty konzumenty tedy může generovat rekurzivně spočetnou množinu přirozených čísel. Nyní se zaměřím na eko-P kolonie obsahující jak agenty konzumenty, tak agenty producenty a jejich prostředí je statické (*0L* schéma obsahuje pouze pravidla typu $a \rightarrow a$).

NRM_{pb} označíme třídu množin čísel počítaných částečně slepým registrovým strojem. Dokázali jsme, že třída jazyků eko-P kolonie s jedním agentem producentem a jedním agentem konzumentem je podmnožinou třídy NRM_{pb}. Pro důkaz jsme použili částečně slepý registrový.

Věta 6 $NRM_{pb} \subseteq NEPCOL_{sc,passive}(2, *)$.

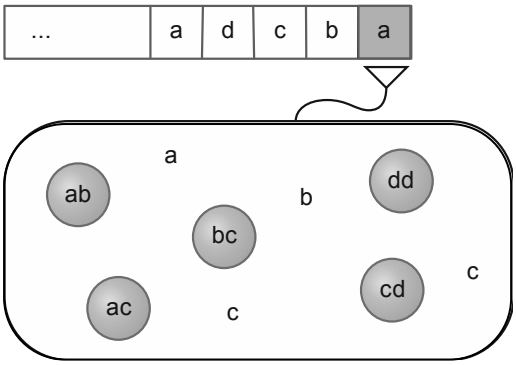
V této kapitole jsme uvedli některé vlastnosti rozšířeného modelu P kolonií, který nazýváme eko-P kolonie.

5 P kolonie se vstupní páskou

V předchozí kapitole jsme rozšířili model P kolonií o možnost měnit prostředí nezávisle na daných organismech. V této kapitole rozšíříme původní model P kolonií o vstupní pásku. Daný model P kolonie nazýváme PCol automat a můžeme ho chápat jako P kolonii řízenou vstupem.

Páska obsahuje stejně jako u konečných automatů vstupní slovo, které spolu s jeho stavem, určuje výpočet. V prostředí PCol automatu umístíme jednoduché membránové systémy nazývané agenty. I tady je každý agent tvořen jednou membránou. Každý agent má k dispozici svou množinu programů. Můžeme tedy říci, že PCol automat je tvořen agenty, které pracují na základě programů. Na rozdíl od původního modelu je výběr programu, který agent použije ovlivněn nejen stavem agenta a obsahem okolí, ale také vstupním symbolem na pásce. Agenty mají jednoduché životní projevy, ale přesto mohou emergovat v mnohem složitější životní projevy celé kolonie. Analogie a

vliv pásky u PCol automatu je jiná než změna prostředí u eko-P kolonie. Agenty, které mohou, se snaží získat objekt odpovídající vstupnímu symbolu tak, že tento objekt vytvoří z jiného, nebo objekt transportují z prostředí. Páska představuje pro kolonii jinou informaci, symboly jsou na ní zapsány a čteny v určitém pořadí.



Obr. 3. Schéma PCol automatu

Opět každý program obsahuje tolik pravidel, kolik je objektů uvnitř agenta. Počet objektů uvnitř agenta je pevně stanoven a je pro všechny agenty stejný. Každý program PCol automatu se skládá ze dvou pravidel. Jedno z nich může být tzv. *T*-pravidlo. Je to pravidlo umožňující agentu přečíst vstupní symbol z pásky. Vstupní symbol je považován za přečtený, pokud alespoň jeden agent použil právě *T*-pravidlo. Pokud program obsahuje *T*-pravidlo, budeme ho nazývat *T*-programem.
Na základě používání *T*-programů můžeme definovat různé způsoby výpočtu. Pro každou konfiguraci PCol automatu, která je dána obsahem všech agentů, obsahem prostředí a nepřečtenou částí pásky, můžeme určit, jak bude probíhat výpočet:

- jsou použity pouze *T*-programy – agent, který nemůže číst z pásky v daném kroku výpočtu, nebude pracovat,
- v množině aplikovatelných programů je alespoň jeden *T*-program – symbol z pásky bude přečtený alespoň jedním agentem,
- kritérium je výběr maximálního počtu *T*-programů – z pásky bude číst maximální počet agentů, ostatní, kteří nemají aplikovatelný *T*-program, použijí jiný program,
- pokud v některém kroku nemůže být použitý alespoň jeden *T*-program, v některých módech se výpočet nezastaví, ale může pokračovat krokem, v němž použijí agenty programy bez *T*-pravidel.

5.1 PCol automat

V této části uvedeme formální definici PCol automatu.

Definice 3 PCol automat s kapacitou *2* je struktura
$\Pi = (A, e, V_E, (O_1, P_1), \ldots, (O_n, P_n), F)$, kde

- A je konečná množina, kterou nazýváme abecedou, její prvky nazýváme objekty,
- e je environmentální objekt automatu, $e \in A$,
- V_E je multimnožina nad $(A - \{e\})$ charakterizující počáteční obsah prostředí,
- (O_i, P_i),, $1 \leq i \leq n$, je i-tý agent
  - O_i je multimnožina nad A, která definuje počáteční obsah agenta B_i a $|O_i| = 2$,
  - P_i je konečná množina programů, $P_i = T_i \cup N_i$, $T_i \cap N_i = \emptyset$, každý program je tvořen ze dvou pravidel z těchto typů pravidel:
  - Pásková pravidla (T-pravidla)
    - Přepisovací: $a \overset{T}{\rightarrow} b$,
    - Komunikační: $a \overset{T}{\leftrightarrow} b$.
  - Nepásková pravidla (N-pravidla)
    - Přepisovací: $a \rightarrow b$.
    - Komunikační: $c \leftrightarrow d$.
  - T_i je množina páskových programů (T-programů), které jsou tvořeny z jednoho T-pravidla a z jednoho N-pravidla.
  - N_i je množina nepáskových programů (N-programů) obsahující pouze N-pravidla.
- F je množina koncových stavů automatu, každý stav je (n + 1)-tice $(V_E; v_1, ..., v_n)$, kde v_i, $1 \leq i \leq n$, je obsah i-tého agenta.
-

U PCol automatu můžeme mít čtyři typy T-programů: $< a \overset{T}{\rightarrow} b,\ c \rightarrow d >$, $< a \overset{T}{\rightarrow} b,\ c \leftrightarrow d >$, $< a \overset{T}{\leftrightarrow} b,\ c \rightarrow d >$, $< a \overset{T}{\leftrightarrow} b,\ c \leftrightarrow d >$.

Konfigurací PCol automatu nazýváme $(n + 2)$-tici $(w_T; w_E; w_1, ..., w_n)$, kde $w_T \in A^*$ je nepřečtená část vstupní pásky, $w_E \in (A - \{e\})^*$ je multimnožina objektů různá od objektu e umístěném v prostředí PCol automatu a w_i, $1 \leq i \leq n$ je obsah i-tého agenta.

Výpočet začíná v počáteční konfiguraci definované vstupním slovem na pásce, obsahem prostředí a agentů na začátku výpočtu.

Pro každou $(n + 1)$-tici $(w_E; w_1, ..., w_n)$ můžeme sestavit množinu všch množin aplikovatelných programů $\mathcal{P}_{(w_E; w_1, ..., w_n)}$.

Pro $(n + 1)$-tici $(w_E; w_1, ..., w_n)$ a vstupní symbol a definujeme:

- t-přechod $\Rightarrow_t^a$: Jestliže existuje alespoň jedna množina aplikovatelných programů $P(\mathcal{P}$ taková, že každé $p \in P$ je T-program s T-pravidlem ve tvaru $x \overset{T}{\rightarrow} a$ nebo $x \overset{T}{\leftrightarrow} a$, $x \in A$ a množina P je maximální (neexistuje jiná množina P' s $|P'| > |P|$ splňující uvedené podmínky).

- *n*-přechod: $\Rightarrow_n$: Jestliže existuje alespoň jedna množina aplikovatelných programů $P \in \mathcal{P}$ taková, že každé $p \in P$ je *N*-program a množina *P* je maximální.
- *tmin*-přechod $\Rightarrow^a_{t_{min}}$: Jestliže existuje alespoň jedna množina aplikovatelných programů $P \in \mathcal{P}$ taková, že alespoň jeden program $p \in P$ je *T*-program s *T*-pravidlem ve tvaru $x \overset{T}{\rightarrow} a$ nebo $x \overset{T}{\leftrightarrow} a$, $x \in A$, může obsahovat i *N*-programy a množina *P* je maximální.
- *tmax*-přechod $\Rightarrow^a_{t_{max}}$: Jestliže existuje alespoň jedna množina aplikovatelných programů $P \in \mathcal{P}$ taková, že *P* obsahuje tolik *T*-programů (s *T*-pravidlem ve tvaru $x \overset{T}{\rightarrow} a$ nebo $x \overset{T}{\leftrightarrow} a$, $x \in A$) kolik je možné, do množiny *P* mohou patřit i *N*-programy a množina *P* je maximální.

PCol automat pracuje v *t* (*tmax*, *tmin*) módu, jestliže používá pouze *t*- (*tmax*-, *tmin*-) přechody. Automat pracuje v *nt* (*ntmax* nebo *ntmin*) módu, pokud používá *t*- (*tmax*- nebo *tmin*-) přechody a jestliže neexistuje žádná množina aplikovatelných programů tvořená *T*-programy (nebo použitými *T*-programy), potom může použít *n*-přechod. PCol automat pracuje v *init* módu, pokud používá pouze *t*-přechody a po přečtení všech vstupních symbolů použije *n*-přechody.

Pokud PCol automat pracuje v *t*, *tmax* nebo *tmin* módu, čte jeden vstupní symbol v každém kroku výpočtu. Z tohoto důvodu délka výpočtu je rovna délce vstupního slova na pásce.

PCol automat může mít tyto typy akceptování:

- Zastavením (*halt*) – neexistuje množina aplikovatelných programů odpovídající módu výpočtu. Výpočet je úspěšný, jestliže se výpočet ukončí a celá vstupní páska je přečtena.
- Přečtením posledního vstupního symbolu (*lastsym*) – výpočet je ukončen po přečtení posledního symbolu a pokud neexistuje množina aplikovatelných programů (odpovídající výpočetnímu módu, tak výpočet je úspěšný. Pokud výpočet se ukončí před přečtením všech symbolů z pásky, tak je výpočet neúspěšný.
- Dosažením konečného stavu (*finstate*) – výpočet je ukončen po přečtení posledního vstupního symbolu, pokud se automat nezastaví dříve. Výpočet je úspěšný pokud vstupní páska je přečtená a PCol automat se dostane do konfigurace patřící do množiny konečných stavů *F*. Zde je podobnost s výpočtem konečného automatu.

Jazyk přijímaný PCol automatem *Π* je definován jako množina slov, pro které existuje úspěšný výpočet v daném módu a pro daný typ akceptace.

Definice 4 $L(\Pi, mod, acc) = \{w / w \in A^*$ je akceptováno výpočtem v módu s *acc* typem akceptace$\}$, kde $mod \in \{t, nt, tmax, ntmax, tmin, ntmin, init\}$ a $acc \in \{halt, lastsym, finstate\}$.

5.2 Vlastnosti PCol automatu

V této části uvedeme některé vlastnosti PCol automatu. Výsledky společně s jejich důkazy jsou uvedeny v [3]. PCol automatem pracujícím v *t* módu s jedním agentem můžeme simulovat činnost konečného automatu. Tzn., že platí:

Lemma 1 Pro každý regulární jazyk L existuje PCol automat pracující v *t* módu, který má jednoho agenta a akceptujícího všechna slova z L dosažením koncového stavu.

PCol automat dokáže akceptovat i kontextový jazyk. Příklad je uveden v následujícím lemmatu.

Lemma 2 Existuje PCol automat akceptující jazyk $L = \{a^n b^n c^n / n \leq 0\}$ pracující v *t* módu výpočtu a akceptuje dosažením koncového stavu.

Z předchozích dvou lemmat vyplývá věta 7.

Věta 7 Třída regulárních jazyků je vlastní podmnožinou třídy jazyků akceptovaných PCol automaty, které pracují v *t* módu a akceptující dosažením koncového stavu.

6 Závěr

Uvedené výsledky říkají, že PCol automat pracující v *t* módu a akceptující koncovým stavem rozpozná regulární jazyky. Také jsme uvedli, že existuje kontextový jazyk, který jsou tyto automaty schopné akceptovat stejným způsobem. Pokud PCol automat pracuje v *init* módu, může přijímat jazyky typu *0*. Předpokládáme, že pokud automat bude pracovat v *ntmin* nebo *ntmax* módu, umí akceptovat všechny jazyky třídy *RE*.

Položme si tyto otázky: Můžeme vytvořit výpočetní model založený na struktuře a funkci živé buňky? Můžeme vytvořit výpočetní model inspirovaný strukturou a činností živých organismů? Je vlastně buňka počítač? Můžeme nějakým způsobem implementovat výpočty na živých buňkách nebo implementovat modely inspirované buněčnou strukturou na existující hardware? V tomto příspěvku jsme se pokusili popsat některé modely membránových systémů, které se právě inspirují strukturou a činností živých organismů. Popsali jsme původní model P kolonií, jeho rozšíření o možnost vývoje prostředí eko-P kolonie a P kolonie řízené vstupem. Uvedli jsme vlastnosti těchto modelů.

V oblasti výzkumu membránových systémů se zkoumají výpočetní techniky inspirované biochemií buněčné struktury. Z tohoto hlediska můžeme říci, že se membránové výpočty blíží k paradigmatu počítající buňky.

Literatura

[1] Cienciala, L., Ciencialová, L., Kelemenová, A.: Homogenní P kolonie. In Kelemen, J., Kvasnička, V., Pstružina, K. (eds.) *Kognice a umělý život VIII*, Opava, 2008, s. 71-80.

[2] Cienciala, L., Ciencialová, L., Kelemenová, A.: Homogeneous P colonies. *Computing and Informatics* 27, 2008, s. 481-496.

[3] Cienciala, L., Ciencialová, L., Csuhaj-Varjú, E., Vazsil, G.: PCol Automata: Recognizing Strings with P colonies. V Martínez del AMOR, M. A., Paun, G., Hurtado de Mendoza, I. P., Riscon-Núnez, A. (eds.) Eight Brainstorming Week on Membrane Computing, Sevilla, 2010, s. 65 - 76.

[4] Cienciala, L., Ciencialová, L., Kelemenová, A.: On the Number of Agents in P colonies, Membrane Computing. Proc. Intern. Workshop, WMC 2007, Thessaloniki, Greece, June 2007 (G. Eleftherakis et al., eds.), LNCS 4860, Springer, Berlin, 2007, pp. 193-208.

[5] Cienciala, L., Ciencialová, L.: Variations on the Theme: P colonies. In Kolář, D., Meduna, A., (eds.) *Proceedings of 1st International Workshop WFM `06*, Ostrava, 2006, s. 27-134, ISBN 80-86840-20-4.

[6] Ciencialová, L., Csuhaj-Varjú, E., Kelemenová, A., Vaszil, G.: On very Simple P colonies. In Gutiérrez-Escudero, R., Gutiérrez-Naranjo, M. A., Păun, G., Pérez-Hurtado, I., Riscos-Núnez, A. (eds.) *Seventh Brainstorming Week on Membrane Computing. Volume I*, Fénix Editora, Sevilla, Spain, 2009, s. 97-108, ISBN 978-84-613-2837-6.

[7] Csuhaj-Varjú, E., Kelemen, J., Kelemenová, A., Păun, G., Vaszil, G.: Cells in environment: P colonies. *Journal of Multiple-valued Logic and Soft Computing* 12, 3-4, 2006m s, 201-215.

[8] Freund, R., Oswald, M.: P colonies working in the maximally parallel and in the sequential mode. In Ciobanu, G., Păun, G. (eds.) *Pre-Proceedings of the 1st Inerantional Workshop on Theory and Application of P systems*, Timisoara, Romania, 2005, s. 49-56.

[9] Kelemen, J., Kelemenová A., Păun, G.: Preview of P colonies: A biochemically inspired computing model. In Bedau, M. et al. (eds.) *Workshop and Tutorial Proceedings. Ninth International Conference on the Simulation and Synthesis of Living Systems (Alife IX)*, Boston Mass., 2004, s. 82-86.

[10] Kelemen, J., Kelemenová, A.: On P colonies, a Simple Biochemically Inspired Model of Computation. Proceedings 6th International Symposium of Hungarian Researchers on Computational Intelligence, Budapest Polytechnic, Budapest, 2005. s. 40-56.

[11] Păun, G.: Computing with membranes. *TUCS report 208*, Turku, Centre for Computer Science, Finland, 1998.

Konekcionizmus v náručí výpočtovej kognitívnej vedy

Igor FARKAŠ[1]

Abstrakt. Príspevok sa zaoberá princípmi konekcionistických modelov (umelých neurónových sietí) a ich porovnaním s ostatnými paradigmami kognitívnej vedy (symbolizmom a dynamickým prístupom). Venujeme sa dôležitým konceptom ako sú mentálna reprezentácia, definícia počítania, mechanizmy učenia, významy a problém ukotvenia symbolov. Uvádzame teóriu ukotvenej kognície založenú na nových empirických poznatkoch o mozgu/mysli. Využitie umelých neurónových sietí ilustrujeme na príkladoch týkajúcich sa modelovania raného kognitívneho vývinu, ako napríklad vizuálno-motorická koordinácia, modelovanie referenčných rámcov v mozgu ako súčasti priestorovej kognície, rozpoznávanie objektov, a spracovanie jazyka na lexikálnej a vetnej úrovni, a jeho integráciu s ostatnými kognitívnymi funkciami.

1 Úvod

Vznik kognitívnej vedy ako vednej disciplíny, spojený s kognitívnou revolúciou v polovici 20. storočia, bol vyvrcholením paralelných snáh o poznávanie ľudskej mysle (mind) a mentálnych procesov [57]. Otázky týkajúce sa ľudskej mysle a mentálnych procesov si filozofi kládli už počas antických čias, a psychológovia asi od polovice 19. storočia. No až v priebehu minulého storočia sa naakumulovalo dostatočné množstvo poznatkov a vznikla vôľa pre komunikáciu odborníkov z rôznych tzv. materských disciplín kognitívnej vedy, čím sa vytvorili predpoklady pre interdisciplinárny prístup v tejto oblasti. To znamená, že na myseľ sa pozeráme z rôznych perspektív a využívame pritom rôzne dostupné metódy skúmania, čo v konečnom dôsledku obohacuje každú perspektívu. To si samotný objekt skúmania priamo vyžaduje, a to kvôli svojej komplexnosti z pohľadu úrovní a metód skúmania, ako aj akémusi špeciálnemu statusu mysle/mozgu ako entity vo svete okolo nás (a v nás). Dnes už prevláda názor, že myseľ a mozog úzko súvisia, že ich nemožno úplne oddeliť (ako to kedysi tvrdil Descartes v podobe dualizmu substancií). Na mozog sa pozeráme z objektívnej perspektívy (tretej osoby), skúmajúc ho metódami prírodných vied, zatiaľ

[1] Centrum pre kognitívnu vedu, Katedra aplikovanej informatiky, FMFI, Univerzita Komenského, Mlynská dolina, 84248 Bratislava, e-mail: farkas@fmph.uniba.sk

čo pri skúmaní mysle musíme využívať iné vedecké metódy, lebo každý z nás má interný prístup len ku svojej vlastnej mysli.

V minulosti bola povaha mysle predmetom vášnivých debát filozofov mysle, no v ostatných dekádach rokov sa vzťah mysle/mozgu preniesol do portfólia aj iných disciplín, lebo aj o mozgu už toho vieme oveľa viac. Dodnes neexistuje úplný konsenzus ohľadne tohto vzťahu, aj keď isté zúženie manévrovacieho priestoru prijateľných teórií sa dosiahlo, najmä smerom k rôznym formám tzv. neredukcionistického fyzikalizmu (napr. [38]). Nad otázkami fungovania mysle (človeka, resp. zvieraťa) sa vedeckými metódami zamýšľajú kognitívni psychológovia, ktorí formulujú teórie, a na ich základe vedecké hypotézy, ktoré testujú pomocou behaviorálnych experimentov. Povahou a fungovaním mysle sa zaoberajú i lingvisti, pretože jazyk možno považovať za akúsi bránu do mysle človeka, ktorý prostredníctvom jazyka dokáže vypovedať o svojich (vnútorných) mentálnych stavoch (čo v prípade zvierat využiť nemôžeme). Psycholingvistika sa opiera o psychologické experimenty zamerané na jazyk, zatiaľ čo neurolingvistika si všíma vzťahy medzi poškodeniami mozgu a jazykovým správaním, čo umožňuje hľadať tzv. neurálne koreláty jazykových funkcií. Kognitívna neuroveda tento vzťah skúma v celom spektre kognitívnych funkcií mysle/mozgu, a to vo svojich začiatkoch hlavne pomocou behaviorálnych experimentov. K tomuto skúmaniu od 80-tych rokov minulého storočia napomáhajú zobrazovacie metódy mozgu (napr. na báze funkčnej magnetickej rezonancie, alebo merania elektrickej aktivity mozgu), ktoré nám poskytujú informáciu o činnosti rôznych častí mozgu, s rozličnými stupňami priestorového a časového rozlíšenia, v závislosti od použitej metódy. Kognitívna antropológia má tiež svoje miesto v kognitívnej vede, pretože človek je sociálna bytosť, a tak jeho poznanie a s tým súvisiaci vývin mysle závisí aj od kultúry. Napokon, vznik počítačov sprevádzajúci kognitívnu revolúciu vniesol do kognitívnej vedy nové svetlo (ale i nové otázky a paradigmy) a umelá inteligencia sa tak stala tiež jednou z materských disciplín kognitívnej vedy. Snaha výpočtovo modelovať kognitívne procesy, čo je možné vďaka počítačom, nie je samoúčelná, ale nutná na to, aby sme mohli testovať vedecké hypotézy a prostredníctvom modelov a ich predikcií ponúkať inšpiráciu pre nové experimenty.

Práve výpočtové modelovanie budeme zdôrazňovať v tomto príspevku, pretože bez neho nemôže poznanie v kognitívnej vede principiálne dosiahnuť uspokojivú úroveň, čo sa týka kognitívnych mechanizmov (na rôznych časopriestorových úrovniach fungovania). Samozrejme, týmto sa výpočtové modelovanie nestáva jediným kľúčovým prístupom v kognitívnej vede, ktorá sa nezaobíde ani bez teoretických a experimentálnych metód skúmania. Skrátka, jedine spoločný postup všetkých metód skúmania mysle môže priniesť ovocie (ak keď finálny cieľ sa v tomto kontexte stále zdá byť dosť vzdialený).

Táto kapitola je organizovaná nasledovne. Najprv sa prejdeme hlavnými míľnikmi – paradigmami, ktoré sprevádzali vývin kognitívnej vedy až dodnes (časť 2). V kontexte týchto paradigiem obrátime pozornosť na niektoré kľúčové koncepty, ako je počítanie (časť 3), reprezentácie a významy (časť 4). Pokúsim sa vysvetliť, prečo považujem konekcionizmus za silnú paradigmu, ktorá ponúka kognitívne relevantné vysvetlenia. Použiteľnosť konekcionistických modelov budem ilustrovať v spektre

príkladov týkajúcich sa raného kognitívneho vývinu zahŕňajúc nielen oblasť tzv. nižšej kognície ale aj vyššej ako napr. jazyk (časť 5). Časť 6 tvorí zhrnutie hlavných bodov kapitoly. Ak sa mi podarí nejakým spôsobom inšpirovať (alebo dokonca presvedčiť) cteného čitateľa, táto kapitola splnila svoj účel. Ak nie, nevadí, každý má právo na svoj názor aj v oblasti kognitívnej vedy. ☺

2 Paradigmy kognitívnej vedy

Od svojich počiatkov prešla kognitívna veda za 60 rokov svojej existencie vývinom, ktorý nadobudol istý trend. Ukazuje sa totiž, že v kognitívnej vede nastáva nevyváženosť, pretože kognitívna psychológia sa stáva dominantnou disciplínou. Gentner ([59]) uvádza, že v súčasnosti tvorí kognitívna psychológia už viac ako 50% vedeckej produkcie v kognitívnej vede (pričom jej nárast začal už zhruba v roku 1978, keď tvorila asi 26%), zatiaľ čo ostatné dve disciplíny, ktoré patrili k jadru kognitívnej vedy už v jej počiatkoch – umelá inteligencia a lingvistika – majú asi po 20%. Ak tento trend bude pokračovať, v priebehu ďalších 30 rokov „ovládne“ kognitívna psychológia kognitívnu vedu. Toto by však viacerí psychológovia, vedomí si dôležitosti interdisciplinárneho poznávania, považovali za Pyrrhovo víťazstvo. Bolo by veľmi dobré, keby tento poznatok sa stal vetrom do plachiet rôznym výpočtovým modelom, pretože empirických dát je už pomerne dosť (no nikdy nie dosť ☺).

Jednotlivé paradigmy, ktoré spomenieme, sa týkajú výpočtových prístupov pri formalizácii mysle a mentálnych procesov, no majú aj svoje filozofické implikácie týkajúce sa vzťahu mysle a mozog. Formálne výpočtové prístupy, ako bolo spomínané, ponúkajú **explicitné mechanizmy** fungovania mysle/mozgu, čo principiálne nemožno dosiahnuť pomocou teoretických modelov (s krabičkami a šípkami). Pre jednotlivé paradigmy kognitívnej vedy sú kľúčovými pojmami (konceptmi) ***mentálne reprezentácie***, ***významy*** a ***výpočtové operácie***, avšak ako uvidíme, tieto koncepty sa používajú nerovnakým spôsobom, čo sa pokúsim objasniť. Spomeňme stručne tri hlavné paradigmy kognitívnej vedy: symbolovú[2], konekcionistickú a dynamickú.

2.1 Symbolová paradigma

Vznik symbolovej paradigmy (pre ktorú sa zaužíval aj názov ***kognitivizmus***[3] alebo aj ***klasická paradigma***) bol podnietený vynájdením moderných digitálnych počítačov, ktoré realizujú diskrétne výpočty so symbolmi. Výpočet v počítači prebieha za pomoci dvoch odlišných kľúčových komponentov: procesora a pamäte. Procesor sériovým spôsobom spracováva symboly, uložené v pamäti a vykonáva pritom inštrukcie podľa programu uloženého v inej časti pamäte. Interakcia počítača s prostredím je na periférii záujmu a prebieha prostredníctvom vstupno-výstupných podsystémov, ktoré sprostredkovávajú vstupy pre centrálny procesor (vhodnou transformáciou) a výstupy

[2] Niektorí autori používajú v slovenčine, resp. češtine termín “symbolická”. Ja v kontexte kognitívnej vedy preferujem termín “symbolová” (súc inšpirovaný Martinom Takáčom), aby sme tento význam odlíšili od konvenčného významu slova (napr. symbolický akt).

[3] Termín kognitivizmus nebudem používať, lebo môže navodzovať dojem, že táto paradigma má najbližšie ku vysvetleniu kognitívnych procesov

z neho, konzistentne napr. s teóriou modularity ([48]). Klasická paradigma konceptualizuje **myseľ ako výpočtový stroj**, oddeliteľný od prostredia, ktorý manipuluje s internými symbolmi podľa logických pravidiel, podobne ako to robí počítač. Mentálne operácie sú realizované pomocou vnútorného jazyka mysle, napr. mentálčiny ([49]). ***Výpočtovú teóriu mysle*** vystihuje hypotéza ***o fyzikálnom symbolovom systéme***, ktorý podľa jej autorov „disponuje nutnými a postačujúcimi prostriedkami na všeobecné inteligentné konanie“ ([100], s. 87)[4].

Vychádzajúc zo skorších prác symbolikov ([143],[100],[48]), Harnad ([65]) zrekonštruoval výstižnú **definíciu symbolového systému** ako (1) súboru arbitrárnych „fyzických znakov" (tokens), zaznamenaných na papieri, páske, či inom médiu, s ktorými (2) sa dá manipulovať pomocou „explicitných pravidiel“, (3) podobných znakom, či reťazcom znakov. Pravidlami riadená symbolová manipulácia je zameraná (4) iba na tvar symbolových znakov (nie ich „význam“), tzn. je čisto syntaktická a obsahuje (5) „pravidlami riadené kombinácie“ a rekombinácie symbolových znakov. Existujú (6) primitívne atomické symbolové znaky a (7) zložené znaky (reťazce). Celý systém a všetky jeho časti – atomické znaky, zložené znaky, syntaktické manipulácie, aktuálne i možné pravidlá sú (8) „sémanticky interpretovateľné“, t.j. syntaxi možno systematicky priradiť význam, napr. pri zastupovaní objektov alebo opisovaní stavov.

Z pohľadu symbolizmu je teda podstatné, že myslenie (a tým aj konanie) človeka je vnútorne riadené algoritmami, ktoré sú realizované programami, a nejako implementované v mozgu. Ako, to symbolizmus nerieši, pretože sa to považuje len za vec implementácie. Pri počítačoch môžeme ľubovoľný algoritmus naprogramovať v nejakom programovacom jazyku a potom ho implementovať vo zvolenej hardvérovej platforme. Podobná predstava prevláda i v prípade mozgu. Nezáleží ako sú v mozgu implementované mentálne procesy.

Počítačová metafora mysle inšpirovala aj Marrovu ([90]) teóriu troch nezávislých **úrovní abstrakcie** (vysvetlenia nejakého fenoménu), a to výpočtovej, algoritmickej a implementačnej, ktorú rozpracoval v kontexte modelovania vizuálneho spracovania informácie, a ktorá výrazne ovplyvnila kognitívnu vedu. ***Výpočtová*** úroveň opisuje to, aká informácia sa počíta a prečo, bez bližšej špecifikácie ako. ***Algoritmická*** úroveň už vyžaduje znalosť o používaných reprezentáciách, uchovávajúcich informácie, a o výpočtoch s nimi. ***Implementačná*** úroveň sa týka realizácie konkrétneho algoritmu vo fyzickom substráte. Medzi úrovňami platí vzťah, že jedno vysvetlenie na vyššej úrovni sa dá transformovať na viacero vysvetlení na nižšej úrovni. Ako ilustračný príklad uvažujme násobenie dvoch viacciferných čísel. To je teda cieľ výpočtu, ktorý sa dá dosiahnuť rôznymi algoritmami, napríklad takým, ktorý človek bežne používa, a v ktorom sa medzivýsledky násobenia (jedného čísla číslicou druhého čísla) zapíšu pod seba a potom sa sčítajú. Napokon, implementačná úroveň už predstavuje konkrétnu realizáciu tohto algoritmu v nejakom fyzickom médiu (počítač, pero a papier, a i.). Je zrejmé, že symbolová paradigma sa zaoberá výpočtovou a algoritmickou úrovňou analýzy, implementačná je považovaná za nepodstatnú.

[4] Čo to presne je inteligentné konanie/správanie nás teraz nemusí trápiť. Postačí, že čitateľ má o tom intuitívnu predstavu. Viac na túto tému možno nájsť v [137].

Symbolový (algoritmický) pohľad na myseľ je dodnes viacerými odborníkmi považovaný za tú správnu úroveň abstrakcie najmä v kontexte vysvetľovania procesov tzv. ***vyššej kognície***, ako je usudzovanie, plánovanie, a tiež používanie jazyka (napr. [49][107][136]). Výhodou klasickej paradigmy je, že poskytuje silné matematické a logické formalizmy, ktoré sú zväčša transparentné a preto človeku zrozumiteľné. Je celkom možné, že ide o jednu z početných kognitívnych predispozícií človeka (cognitive biases) smerom k diskrétnym formalizmom, pretože operácie používané na symboloch sú blízke ľudskému mysleniu ([115]). Pre človeka je prirodzené uvažovať v termínoch diskrétnych podmienok, pravidiel, propozícií a logických inferencií. Napríklad, ak sa niečoho bojím (propozícia opisujúca môj emocionálny stav je pravdivá), mojím rozhodovaním sledujem cieľ, aby som vykonaním vhodnej akcie dosiahol požadovaný stav, aby daná propozícia nebola pravdivá.

2.2 Konekcionizmus

Konekcionizmus predstavuje spektrum metód, ktoré vznikli v rámci umelej inteligencie, a ktoré boli inšpirované architektúrou a fungovaním mozgu. Môžeme teda hovoriť o inšpirácii „zdola." Konekcionistický systém pozostáva z množiny jednoduchých prvkov – umelých neurónov, navzájom komunikujúcich cez váhované spojenia, ktoré predstavujú dlhodobú pamäť systému. Sila konekcionistického systému – umelej **neurónovej siete** – nie je v samotných neurónoch, ale v ich ***vzájomnom prepojení a interakcii***. Paralelné spracovanie a distribuovanosť aktivity predstavujú základný architektúrny rozdiel v porovnaní so symbolovým systémom, pretože každý neurón je súčasne procesorom aj pamäťou (aj keď elementárnou). Taktiež, povaha komunikácie medzi neurónmi má ***numerický*** a nie symbolový charakter, preto hovoríme v prípade sietí o **subsymbolových reprezentáciách**[5]. Neurónová sieť môže pracovať so spojitými hodnotami, čo je veľmi vhodným nástrojom na modelovanie kognície, pretože empirické dáta nasvedčujú tomu, že mnoho kognitívnych procesov má spojité prejavy. Napríklad, nielenže perceptuálne vstupy (napr. obraz na sietnici) a motorické výstupy bývajú spojité (pohyby, zvuková podoba reči), ale podobne môžeme argumentovať aj v prípade modelovania vyšších procesov, napr. rozhodovania ([133]). Spomeňme si, aké ťažké môže byť niekedy rozhodnúť sa medzi dvoma alternatívami. Tento proces „súperenia" alternatív sa dá vysvetliť pomocou neurónovej siete so spojitými aktiváciami. Neurónová sieť teda nespracováva symboly, ale numerické dáta v podobe vektorov. Operácie nad takýmito vektorovými dátami sú (nelineárnymi) ***transformáciami***, čo má svoje implikácie. Je zrejmé, že takéto operácie nie sú vôbec transparentné, a preto sú človeku málo zrozumiteľné, a aj preto si neurónové siete vyslúžili prívlastok „čierna skrinka." Bez použitia techník zhlukovania a vizualizácie vysokorozmerných dát naozaj nevieme, čo sa v neurónovej sieti deje, keď sledujeme jej prebiehajúce aktivity.

[5] Aj keď treba upresniť, že tzv. lokalistické modely neurónových sietí (teda nie distribuované) pracujú s reprezentáciami, ktoré možno nazvať symbolovými. Každý neurón reprezentuje jeden symbol (má hodnotu 1, ostatné 0), a nezáleží na tom, ktorý neurón reprezentuje ktorý symbol – forma a obsah sú oddelené.

Konekcionistický prístup, na rozdiel od symbolového, zahŕňa aj implementačnú úroveň analýzy, pretože mozog nie je počítač (časť 3), a teda pre mozog neplatí implikácia o nezávislosti implementačnej úrovne od vyšších úrovní. Niektorí filozofi navrhujú oddeliť vysvetľované fenomény od mechanizmov ich vzniku ([1]), no v konekcionistických modeloch sú tieto dva aspekty prepojené, teda existuje závislosť medzi implementovanými mechanizmami (danými interakciami medzi neurónmi) a kognitívnymi fenoménmi ([103]).

Dôležitou vlastnosťou umelých neurónových sietí je to, že na rozdiel od symbolovej umelej inteligencie, kde znalosť do systému vkladá dizajnér, ponúkajú biologicky inšpirované mechanizmy akvizície znalostí, **učenia**, aplikovateľné na rôznych úrovniach abstrakcie ([67],[35]). To znamená, že parametre siete (váhy spojení) nemusíme explicitne nastaviť (čo sa prakticky ani nedá), no treba sieti poskytnúť vhodný mechanizmus učenia. Podobne ako človek, neurónová sieť sa učí na príkladoch. Učenie spôsobuje zmenu váh tak, aby sa po naučení dosiahlo požadované správanie siete, čo formálne odpovedá hľadaniu extrému (minima, resp. maxima) nejakej hodnotovej funkcie.

V prípade neurónových sietí tiež bežne hovoríme, že počítajú a teda spracovávajú informáciu. Počítajú však rovnako ako symbolové systémy? Niektorí odborníci tvrdia, že áno, pretože existujú teoretické práce o ekvivalencii medzi univerzálnym počítačom (Turingovým strojom[6]) a rekurentnou neurónovou sieťou s nekonečne veľkou pamäťou (pozri napr. prehľad v [138]). Táto ekvivalencia však predpokladá klasický pohľad na počítanie ako na vykonávanie programu (algoritmus). Neurónové siete však ponúkajú viac. Modely neurónových sietí sú spektrom metód z pohľadu typov aktivácií neurónov a ich dynamiky. Modely, ktoré pracujú v spojitom priestore, a niektoré dokonca v spojitom čase sú matematickými modelmi, ktoré sa dajú presne opísať len pomocou (nelineárnych) diferenciálnych rovníc, a nie diskrétnych krokov výpočtu (počítačová implementácia ponúka len aproximáciu). Zahrnutie takýchto modelov si vyžaduje voľnejšiu definíciu počítania (časť 3). Od modelov neurónových sietí pracujúcich v spojitom čase sa priamo dostávame k paradigme dynamických systémov.

2.3 Dynamické systémy

Čo klasická paradigma a konekcionizmus zdieľajú, je ich izolovanosť od okolitého prostredia. V prípade symbolového modelu je to zjavné, pretože izolovanosť je jedným z predpokladov. Neurónová sieť sa síce učí, čiže interaguje s prostredím, no bežným predpokladom je, že všetky trénovacie dáta máme hneď dispozícii, takže interakcia nie je „plnohodnotná," pretože výstupy siete nemajú dopad na prostredie. Vďaka izolovateľnosti výpočtového modelu od prostredia môžeme (no nemusíme) pracovať s vnútornými reprezentáciami entít z prostredia, ktoré systém spracováva.

Takýto prístup sa stal tŕňom v oku niektorým kognitívnym vedcom, ktorí prišli s myšlienkou kognitívneho systému ako dynamického systému, ktorý nepotrebuje reprezentácie, pretože celý proces interakcie s prostredím prebieha v spojitom čase

[6] Bežný digitálny počítač môžeme chápať ako fyzický príklad Turingovho stroja, aj keď nemá nekonečne veľkú pamäť (čo sa teoreticky predpokladá, aby sa dal na ňom realizovať ľubovoľný algoritmus).

([142],[113]). Kognitívne procesy teda nie sú chápané izolovane od prostredia, ale v úzkej previazanosti s ním (coupling). Kognitívne procesy sú mentálnou aktivitou v čase, ktorú môžeme opísať pomocou systému nelineárnych diferenciálnych rovníc. Táto paradigma, postavená na teórii nelineárnych dynamických systémov predpokladá existenciu dvoch kľúčových, vzájomne prepojených, typov premenných. Jedným sú tzv. kolektívne premenné, ktoré zahŕňajú vzťahy medzi interagujúcimi zložkami dynamického systému, a vysvetľujú správanie systému. Druhým typom sú riadiace parametre, ktorých kvantitatívne zmeny môžu spôsobovať kvalitatívne zmeny v správaní systému (tzv. fázové prechody). Príkladom takéhoto systému môže byť pohyb koňa, ktorého rytmus pohybu (vyjadreného periodickými priebehmi malého počtu kolektívnych premenných) závisí od rýchlosti pohybu (riadiaci parameter). Kôň sa môže pohybovať v každom čase len v jednom rytme, (fázové) prechody medzi nimi nemusia nutne nastávať pri tých istých rýchlostiach, a závisia aj od stavu prostredia (napr. hrboľatá cesta). Súčasťou takéhoto dynamického opisu kognitívneho procesu sa teda priamo stáva aj prostredie, čo rozširuje konceptuálne chápanie mysle. Bressler a Kelso ([12]) argumentujú, že dynamický pohľad je relevantný aj pre charakterizáciu samotných mentálnych procesov (vnútri kognitívneho systému), hoci identifikácia kolektívnych premenných a riadiacich parametrov v tomto prípade je zatiaľ tvrdým orieškom.

Dynamický pohľad na spoznávanie vonkajšieho sveta je konzistentný so **zjednávacím prístupom** ([141]), kde sa dôraz z vnútorných reprezentácií (vopred daného) vonkajšieho sveta presúva na ***vnímanie a jednanie vo svete***, ktorý sa takto spoluvytvára. Tým vzniká prostredníctvom spätnej väzby kruhový proces, pretože zjednávaný svet je prostredím pre zjednávanie (enactment), ale zjednávanie je súčasne predpokladom zjednávaného sveta. V rámci tohto pohľadu si každý subjekt (kognitívny systém) zjedná svoj svet sám v procese dynamickej interakcie so svetom (vrátane iných jedincov), a ten bude jedinečný.

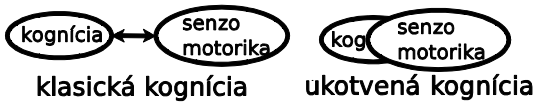
2.4 Smerom k ukotvenej kognícii

Spomínané paradigmy vznikali postupne, tak ako sme ich spomínali, a v súčasnosti stále tvoria jadro výpočtových prístupov kognitívnej vedy, pretože každá paradigma má svojich zástancov. Ešte sa im budeme venovať, teraz venujme na chvíľu pozornosť hlavným prúdom a teoretickým koncepciám vo vývoji kognitívnej vedy, ktoré prispeli k posunu v dominantnosti jednotlivých paradigiem ([59]). Už v 70. rokoch minulého storočia sa v kognitívnej psychológii objavovali skeptické pohľady na centrálnosť (vnútorných) reprezentácií a symbolového spracovania informácie v ľudskej kognícii. Hlavný predstaviteľ, Gibson ([61]), tohto prúdu známeho ako **ekologická psychológia**, zdôrazňoval, že „nie je dôležité, čo je vo vnútri hlavy, ale vnútri čoho (akého prostredia) sa hlava nachádza.“.

Tento prúd plynulo prešiel v 80. rokoch do teórie **situovanej kognície**, ktorá bola postavená na predpokladoch, že ľudská inteligencia je fundamentálne interaktívna ([70]), a že ľudská kognícia je fundamentálne neoddeliteľná od kontextu ([82]). Podľa prvého predpokladu človek bezprostredne využíva prostredie (iných ľudí, zariadenia, situácie, atď.) pre svoje inteligentné konanie, a je s nimi spojený v rámci

spätnoväzbovej slučky. Druhý predpoklad viedol k rozšírenému chápaniu mysle a kognície v zmysle jej umiestnenia aj do prostredia – myseľ človeka nesídli len v jeho mozgu, ale jej súčasťou sú aj prvky prostredia (napr. knihy, obrazy, počítač, rôzne poznámky a i.).

Dôraz na interakciu s prostredím ostal v centre pozornosti aj naďalej, no v 90-tych rokoch sa začal klásť dôraz na reprezentácie, ktoré túto interakciu umožňovali, a to v rámci prúdu **vtelenej kognície**. Vtelená (stelesnená, embodied) kognícia, podobne ako tvrdil Gibson, vychádza z predpokladu, že pochopenie mentálnych procesov nemožno oddeliť od tela človeka, konkrétne jeho zmyslov a motoriky. Avšak vtelená kognícia predpokladá, že informácia môže vstúpiť do tela organizmu v podobe modálnych reprezentácií a podieľať sa na ďalších, vyšších procesoch kognície ([6]). Zmysly určujú povahu perceptuálnych vstupov, a motorické ústrojenstvo zase určuje spektrum akcií, ktoré človek môže vykonať, a ktoré tiež predstavujú obmedzenia (constraints) pre vznik a modifikáciu vnútorných reprezentácií. Napríklad, mentálna reprezentácia objektu kladivo nezahŕňa len jeho perceptuálne vlastnosti, ale aj možnosti použitia tohto objektu (ak s ním dotyčný subjekt má skúsenosť).



Obr. 1. V klasickom systéme sú (modálne) percepčné/motorické procesy oddelené od amodálnych kognitívnych procesov, a vzájomné prepojenie je realizované cez (nešpecifikovaný) mechanizmus transdukcie (smerom ku kognitívnym procesom) a referencie (späť). V prípade ukotveného systému majú reprezentácie vzájomný prienik, čím sa kognitívne (mentálne) procesy stávajú tiež modálnymi.

V komunite kognitívnej vedy sa objavilo viacero rôznych no súvisiacich pohľadov na vtelenie a situovanosť kognície ([60],[151]). Barsalou v [7] navrhol termín **ukotvená kognícia**, ktorý zjednocuje rôzne aspekty vtelenia a situovania v prostredí (**Obr. 1**). Silnú podporu pre túto paradigmu v kognitívnej vede možno nájsť v narastajúcej empirickej evidencii (za ostatných 20 rokov), či už z behaviorálnych štúdií (napr. [10]) alebo zobrazovacích metód (napr. [116]). Teórie ukotvenej kognície sú síce opisné, no ponúkajú testovateľné hypotézy pre ďalší empirický výskum (napr. [80],[132]). Zaujímavou predikciou týchto teórií je to, že všetky kognitívne funkcie (vrátane tých vyšších) sú redukovateľné na (nízkoúrovňové) senzomotorické procesy. Inými slovami, neurálne reprezentácie vyšších a nižších kognitívnych procesov majú prienik. Tento pohľad pochopiteľne stojí v ostrom kontraste so symbolizmom, ktorý predpokladá autonómnosť mentálnych reprezentácií spojených s vyššou kogníciou, a chápe ich ako amodálne symboly. Mentálne operácie sú realizované napríklad vo vnútornom jazyku mysle, napr. mentálčine ([49]). Naopak, teória perceptuálnych symbolov ([6]) ponúka riešenie v podobe multimodálnych subsymbolových reprezentácií, ktoré sú ukotvené v modalitách a nepodliehajú tak problému transdukcie (prenosu z modálnej úrovne do amodálnej). Inými slovami, perceptuálny symbol je v podstate koncept, a úloha symbolov sa redukuje na ich priradenie ku konceptom (napr. slová k objektom, či matematické symboly

k matematickým konceptom). Operácie s perceptuálnymi symbolmi prebiehajú tiež v priestore modalít.

3 Počítanie z pohľadu paradigiem

Spomínané tri paradigmy ponúkajú rôzne odpovede na otázky spojené s modelovaním mentálnych procesov. Symbolizmus a konekcionizmus ponúkajú kvalitatívne rozdielne vysvetlenia, no niektorí odborníci hovoria o výpočtovej ekvivalencii medzi nimi. Základom ich argumentácie je to, že symbolový systém dokáže realizovať akýkoľvek algoritmus (lebo Turingov stroj je univerzálny výpočtový stroj) a že umelým neurónovým sieťam boli tiež pripísané univerzálne vlastnosti. Dvojvrstvová dopredná neurónová sieť dokáže aproximovať ľubovoľnú spojitú funkciu ([69]) a rekurentná neurónová sieť (t.j. so spätnou väzbou) je výpočtovo ekvivalentná univerzálnemu Turingovmu stroju ([131]). To znamená, že akýkoľvek algoritmus sa dá simulovať rekurentnou neurónovou sieťou, a naopak, akákoľvek neurónová sieť sa dá simulovať na bežnom počítači (ktorý môžeme chápať ako Turingov stroj). Toto je pravda, a bežne simulujeme konekcionistické modely na digitálnych počítačoch. Z toho však nevyplýva, že je jedno, či sa nejaké kognitívne procesy snažíme vysvetliť konekcionisticky alebo pomocou klasickej paradigmy. Jedno to nie je z viacerých dôvodov (uvádzaných nižšie), okrem iného aj preto, že tieto existenčné dôkazy neriešia otázku ako skonštruovať sieť (t.j. nastaviť jej architektúru a váh), tak aby realizovala daný algoritmus. Symbolisti môžu nastaviť parametre (pravidlá) svojich modelov ručne, v konekcionizme to prakticky možné nie je. Vysvetlenie fenoménov v kognitívnej vede často zahŕňa akvizíciu poznatkov a to nie je možné riešiť bez modelovania zmien v systéme (jeho pamäti).

Musíme si vysvetliť koncept počítania, a že počítanie v neurónovej sieti nemusí byť to isté ako počítanie v symbolovom systéme. Otázka počítania začala byť vedecky aktuálnou krátko potom, keď bol v 40-tych rokoch minulého storočia vynájdený digitálny počítač, ktorý inšpiroval vznik teórie vypočítateľnosti a počítačovej vedy. Tie poskytujú teoretické poznatky o tom, čo to znamená počítať na Turingovom stroji. Počítanie v symbolovom systéme v podstate znamená vykonávanie programu, realizujúceho nejaký algoritmus sériovým spôsobom[7]. Pracujeme v diskrétnom čase, s diskrétnymi entitami, symbolmi. Aj človek bežne pri počítaní má na mysli diskrétne výpočtové úkony s číslami, ktoré vedú k výsledku.

V prípade neurónových sietí však tiež predpokladáme, že počítajú. Je povaha ich počítania odlišná od symbolových systémov? V kognitívnej vede prevláda názor, že ľudská myseľ realizuje výpočty, a teda spracováva informácie, pričom my sa tieto procesy snažíme vysvetliť pomocou formálnych modelov. Viacerí zástancovia konekcionizmu zastávajú názor, že mozog vykonáva výpočty a že neurálne počítanie vysvetľuje ľudskú kogníciu (napr. [45],[21],[8],[102]). V čom sa líšia názory konekcionistov a symbolikov je otázka, či konekcionistické modely sú lepšími výpočtovými modelmi než klasické symbolové modely. Väčšina konekcionistov tvrdí,

[7] Samozrejme, paralelizácia výpočtu sa v bežných počítačoch dá dosiahnuť, ale tento fakt nie je z konceptuálneho hľadiska podstatný. Ako teoretický koncept, Turingov stroj pracuje sériovo.

že áno, symbolisti zase, že nie, lebo obe vysvetlenia sú vzájomne transformovateľné (opierajúc sa o spomenuté dôkazy o ekvivalencii). Niektorí symbolici tvrdia, že neurónové siete nepočítajú vôbec, alebo ak, tak len vtedy, keď implementujú symbolový systém ([50]). Niektorí prívrženci konekcionizmu zase zastávajú názor, že mozog nie je výpočtovým systémom (napr. [33],[54]), a že naše výpočtové modelovanie je len jeho diskrétnou aproximáciou.

3.1 Je mozog výpočtový stroj?

Zastavme sa najprv pri tejto otázke. Otázka, či počítač je výpočtovým strojom je triviálna, pretože počítač bol navrhnutý ako výpočtový stroj (a stanovil tak klasickú definíciu počítania). V počítači prebieha spracovanie informácie pomocou elektrických aktivít (prostredie je homogénne) a implementačná úroveň, na ktorej výpočty prebiehajú, je jednoznačne určená. Sú to logické hradlá, ktoré sa preklápajú medzi dvoma možnými diskrétnymi stavmi. V prípade mozgu však implementačná úroveň vôbec nie je jednoznačná, pretože mozog nikto nezostrojil za účelom počítania. Ten sa evolučne vyvinul tak, aby umožnil organizmu efektívne fungovať v dynamickom prostredí ([21]). Na akej úrovni potenciálne prebiehajú výpočty v mozgu? Najprijateľnejšou hypotézou sa javí úroveň buniek mozgu (neurónov), hoci existujú neurofyziologické poznatky o tom, že nižšia (molekulárna) úroveň fyzikálnej reality ovplyvňuje činnosť samotných neurónov. Na druhej strane, na každom jednom neuróne nezáleží[8], tak možno vyššia úroveň (neurálnych zhlukov) by bola vhodnejšia a efektívnejšia. Okrem toho, mozog pracuje „v zajatí“ všadeprítomných spätnoväzobných slučiek, ktoré fungujú na rôznych časopriestorových škálach ([5]), a ktoré vnášajú do systému paralelizmus procesov operujúcich v spojitom čase. Napokon, prostredie v mozgu nie je homogénne, pretože spracovanie informácie má elektricko-chemický charakter: elektrický signál šíriaci sa pozdĺž axónu neurónu sa premieňa na chemickú difúziu na synapsách medzi neurónmi. Okrem toho, rôzne (pomalšie) hormonálne a neuromodulátorové systémy tiež ovplyvňujú správanie neurónu ([73]). Berúc do úvahy všetky tieto skutočnosti dospievame k presvedčeniu, že mozog nie je výpočtovým strojom v klasickom ponímaní, podobne ako mnohé iné komplexné prírodné systémy, ktoré majú nelineárny dynamický charakter. Napriek tomu, na ceste k jeho poznávaniu formálnymi metódami my nemáme inú možnosť, len aproximovať činnosť mozgu diskretizáciou v čase a v priestore (analógové počítače sa prakticky nepoužívajú).

3.2 Klasické a neklasické počítanie

Rozdiel medzi paradigmami reflektovaný aj v odlišnosti medzi klasickým a neklasickým počítaním. Neurónové siete nie sú homogénnou skupinou metód, pretože niektoré modely neurónových sietí počítajú klasicky, niektoré neklasicky a niektoré nepočítajú vôbec ([106]). Tieto kategórie modelov prirodzene vznikali aj chronologicky. Prvý model neurónových sietí pozostával z logických neurónov, ktoré

[8] Je známe, že fyzicky blízke neuróny majú často identickú funkciu, čím vzniká redundancia i robustnosť systému.

pripomínali logické hradlá, a ktoré boli pospájané do vrstiev ([95]). Tento model sa nevedel učiť, pracoval s diskrétnymi hodnotami aktivít (0 a 1) a z výpočtového hľadiska je preto ekvivalentný so symbolovým systémom. Fungovanie takejto siete by sa dalo opísať programom, ktorý sieť realizuje.

Neskôr vznikli modely dopredných sietí na báze neurónov typu perceptrón ([125]) so spojitou aktivačnou funkciu, čím sa stavový priestor neurónovej siete stal spojitým. Tieto modely sa dajú trénovať pomocou malých zmien ich parametrov, čím sa stali použiteľnými pre riešenie rôznych úloh (v rámci vtedajších obmedzení). Učiace pravidlo v adaptívnom systéme je jeho kľúčovým prvkom, pretože umožňuje implementovať ***elementárnu formu zmeny znalosti*** (internej reprezentácie časti vonkajšieho sveta), a to sa dá dobre realizovať v priestore numericky založených reprezentácií. V prípade týchto modelov už hovoríme o **neklasickom počítaní**, čo znamená, že model realizuje požadovanú funkciu – výpočet (t. j. nejakému vstupu priradí požadovaný výstup), bez toho, že by nutne vykonával program. Voľnejšie chápanie počítania teda nepredpokladá realizáciu programu. Napríklad, dopredná sieť sa vie naučiť násobiť dve viacciferné čísla bez toho, že by vypočítavala medzivýsledky, ako to bolo v prípade vyššie spomínaného algoritmu v symbolovom systéme. Výpočet prebehne v podstate na jeden krok: na vstup sa dajú čísla, aktivácia pretransformuje na výstup siete, čím sa generuje výsledok. Prečo učiaca sa neurónová sieť v princípe nevykonáva program vyplýva aj z toho, že pracuje dynamicky v spojitom priestore, a elementárne zmeny parametrov vedú k zmenám vo vstupno-výstupnom zobrazení, pričom takýto meniaci sa vzťah sa nedá redukovať na sériu diskrétnych operácií – potrebovali by sme nekonečnú presnosť. Avšak dá sa argumentovať, že tým že simulujeme neurónovú sieť pomocou symbolového systému (počítača), jej nedávame nekonečnú presnosť, a tak redukcia sa zabezpečiť dá. To je pravda, preto na ideálnu simuláciu neurónovej siete pracujúcej v spojitom priestore aktivít a váh by sme potrebovali analógový počítač, aby sme dosiahli nekonečnú presnosť pri simulácii dynamického procesu učenia.

Treťou kategóriou konekcionistických modelov sú tie, ktoré nielenže pracujú v spojitom priestore, ale aj v spojitom čase. Tieto modely nájdeme medzi umelými neurónovými sieťami, ale častejšie medzi sieťami s impulznými neurónmi ([86]). V tomto prípade už **nemôžeme hovoriť ani o neklasickom počítaní**, pretože počítanie si vyžaduje diskrétny čas. Z rovnakého dôvodu tu nehovoríme ani o reprezentáciách, pretože tie tiež predpokladajú diskrétny čas. Presná simulácia takýchto modelov sa tiež dá dosiahnuť iba na analógových počítačoch, kde sa všetky zmeny hodnôt dejú plynulo.

Takže konceptuálny rozdiel medzi symbolovým systémom a neurónovou sieťou závisí od toho, aký model siete máme na mysli, a tento rozdiel sa týka matematického opisu, nie implementácie. Implementácia akejkoľvek neurónovej siete v digitálnom počítači znamená jej diskretizáciu v čase a stáva sa tak aproximáciou modelu, ktorá môže byť viac alebo menej presná. Simulácia neurónovej siete už pochopiteľne je príkladom klasického alebo neklasického počítania, v závislosti od modelu. Súčasné 32, resp. 64-bitové počítače dosahujú vysokú presnosť, a problém by mohol nastať len v prípadoch, ak by sme potrebovali modelovať zložité dynamické procesy s chaotickým správaním, čo sa týka rekurentných sietí. V prípade väčšiny úloh tento

problém nemáme, lebo pracujeme s výstupmi siete, kde využívame istú robustnosť. To znamená, že na presnej hodnote výstupu nezáleží, a v prípade malých odchýlok sa interpretácia výstupu nezmení (napr. pri spomínanom násobení).

4 Podobnosti a rozdiely medzi paradigmami

V kontexte modelov spojitých neurónových sietí je dynamický prístup úplne konzistentný s rekurentnými neurónovými sieťami pracujúcimi v spojitom čase a priestore. Okrem toho, obe paradigmy predstavujú príklady ***kolektívnych systémov*** pozostávajúcich z množstva interagujúcich prvkov, čoho výsledkom sú ***emergentné javy*** sprevádzajúce dynamiku systému. V čom sa obe paradigmy líšia, je (1) dimenzia systému (t.j. stavového priestoru), ktorá býva v prípade neurónových sietí (daná počtom neurónov) omnoho vyššia, a (2) fakt, že len pri neurónových sieťach boli navrhnuté rôzne mechanizmy učenia. V neurónovej sieti teda existujú dve, vzájomne sa ovplyvňujúce dynamiky, operujúce na rôznych časových škálach – pomalá adaptačná a rýchla aktivačná dynamika. V dynamických systémoch sú síce signifikantné fázové prechody (externe spôsobené zmeny parametrov ktoré vedú k zmene dynamiky), ale ako k nim prirodzene dochádza, resp. ako zabezpečiť adaptáciu je už zložitejšie. Dôraz ostáva na aktivačnej dynamike. Neexistencia učiacich mechanizmov v dynamických systémoch je ich slabou stránkou. Keďže dimenzia oboch prístupov je výrazne rozdielna, zaujímavé prepojenie oboch paradigiem by mohlo napríklad spočívať v tom, že učiaca sa rekurentná neurónová sieť by mohla slúžiť ako „substrát“ pre hľadanie dynamického systému, ktorý operuje na hypotetickom stavovom priestore s nižšou dimenziou[9]. To by sa dalo teoreticky dosiahnuť identifikovaním kolektívnych premenných na základe monitorovania aktivácií existujúcich neurónov (napr. ich zoskupovaním). Podobne, napríklad meranie aktivity mozgu pomocou EEG prístroja ponúka niekoľkokanálové signály odpovedajúce aktivite mozgu na vyššej úrovni než je úroveň neurónov.

Čo má klasická paradigma spoločné s konekcionistickými modelmi pracujúcimi v diskrétnom čase, a čím sa líšia od dynamickej paradigmy, je existencia vnútorných reprezentácií (no s rozdielnymi vlastnosťami). Tie sú v prípade klasickej paradigmy bežne dizajnované ručne ako symboly, resp. symbolové štruktúry, a v prípade neurónovej siete sú dizajnované len čiastočne (napr. požadované výstupy siete), pretože sieť si vytvára ešte svoje vnútorné reprezentácie. Neurónová sieť sa bežne učí na základe vopred pripraveného prostredia, avšak, dá sa principiálne používať spôsobom, aby interagovala s prostredím (či už simulovaným alebo fyzickým, ak jej implementovaná v agentovi), čím by si mohla svoju reprezentáciu prostredia zjednávať. V tom prípade by už išlo o trénovanie, kde by sa trénovacie vstupy sieť vytvárala sama vďaka svojej aktivite. Napríklad, robot riadený neurónovou sieťou, ktorý sa učí interagovať s objektmi vo svojom okolí upriamovaním pozornosti rôznymi

[9] Dynamika vysokorozmerných komplexných systémov (s dimenziou danou počtom neurónov v sieti) nezriedka prebieha v nízkorozmernom nelineárnom podpriestore, čo by mohlo zodpovedať opisu pomocou dynamického systému v priestore tejto nižšej dimenzie.

smermi počas interakcie pripravuje nové perceptuálne vstupy (tie teda nie sú vopred dané).

Niektorí zástancovia tzv. radikálnej tézy dynamickej paradigmy ([142]) zdôrazňujú jej odlišnosť od symbolizmu a konekcionizmu práve kvôli jej nevýpočtovému charakteru, s čím súvisí predpoklad o neexistencii (mentálnych) reprezentácií. Zdôvodňujú to tým, že dynamický systém vyvíjajúci sa v čase sa nedá implementovať pomocou diskrétnych výpočtov, a preto nemôže podporovať ani žiadne vnútorné reprezentácie vonkajšieho sveta. Tu však možno namietať, že tento pohľad je odôvodniteľný len v kontexte modelovania senzomotorickej interakcie človeka s prostredím, zúženej na koordináciu pohybov, kde možno naozaj nepotrebujeme vnútorné reprezentácie (hoci aj o tom sa dá debatovať). Ťažko si ho však predstaviť v širšom kontexte kognitívnych procesov človeka, ktoré si vyžadujú existenciu mentálnych reprezentácií, či už vyvolateľných (cued) vonkajším podnetom, alebo oddelených (detached), t.j. vyvolateľných vnútorne ([58]). Aproximácia spojitých dynamických systémov pomocou modelov s diskrétnym časom je schodnou cestou, ktorá napr. celkom dobre funguje aj v biologicky prijateľnejších modeloch neurónových sietí s impulznými neurónmi ([71]). Preto je reprezentačno-výpočtový prístup kompatibilný s mäkšou verziou dynamickej paradigmy, resp. S pohľadom na mozog ako na samoorganizujúci sa systém ([76],[37]). Podobnú optiku vidieť aj u menej radikálnych zástancov dynamickej paradigmy ([113]).

Môžeme konštatovať, že konekcionistická a dynamická paradigma zdieľajú (spolu s pravdepodobnostnými, bayesovskými prístupmi) viacero spoločných charakteristík a možno dôjde k zbližovaniu týchto paradigiem vo výpočtovej kognitívnej vede ([87]). Symbolová paradigma stojí vo výraznom kontraste s týmito paradigmami, no pravdepodobne si zachová svoju rolu „hrubozrnného," no zrozumiteľného opisu kognitívnych fenoménov. Ten možno obohatiť o detailnejšie, štatisticky orientované konekcionistické vysvetlenie mechanizmov napríklad pomocou neurónovej siete ([1]). Napríklad, vysvetlenie tvorby minulého času anglických slovies pomocou jedného pravidla a výnimiek je transparentným symbolovým opisom, ktorý možno obohatiť o vysvetlenie procesu akvizície ako ho pozorujeme u detí (časť 5.5).

4.1 Reprezentácie

Podobne ako v prípade rozdielneho chápania pojmu ***počítanie*** by bolo dobré si vysvetliť dôležitý koncept ***reprezentácie***. Tento pojem (reprezentácia) sa bežne používa nielen v umelej inteligencii, ale aj v kognitívnej psychológii, či lingvistike. Jazykový pôvod tohto pojmu napovedá, že ide o re-prezentovanie niečoho vonkajšieho (vo svete) niekde inde, vo vnútri nejakého systému (živého alebo umelého) alebo i na papieri. V kognitívnej psychológii sa hovorí o ***mentálnych reprezentáciách***, čo je v podstate prvý teoretický konštrukt v kognitívnej vede. Mentálna reprezentácia je základným konceptom výpočtovej teórie mysle, podľa ktorej sú kognitívne stavy a procesy konštituované výskytom, transformáciou a uchovávaním (v mysli/mozgu) reprezentácií (t.j. štruktúr nesúcich informáciu) nejakého typu ([111]). Korene pojmu mentálna reprezentácia však siahajú až do antických čias, keď širšie chápanie tohto

pojmu medzi filozofmi neznamenalo výpočtový charakter[10]. Až neskorší predpoklad, že reprezentácia je (abstraktný) objekt so sémantickými vlastnosťami, dala tomuto konceptu výpočtový charakter, pretože s takto definovanými objektmi by sa už podľa predpokladu dalo manipulovať.

S nástupom umelej inteligencie sa hlavným cieľom stala **reprezentácia znalostí**, t.j. snaha o formalizáciu znalostí takým spôsobom, aby sa dali uložiť a používať v počítači (symbolovom systéme). Tu zohrali dôležitú úlohu lingvistika a logika, ktoré ponúkli existujúce formalizmy na rôzne implementácie reprezentácie znalostí. Môžeme povedať, že všetky znalosti (tzv. báza znalostí) sú z pohľadu klasickej paradigmy ***explicitné a symbolové***, a keď systém disponuje pravidlami odvodzovania (inferenčné mechanizmy), dokáže si odvodiť ďalšie znalosti. Dôležitou vlastnosťou reprezentácií v symbolovom systéme je, že sú stabilné. Môžu sa síce v čase meniť (ak je to súčasťou programu), ale možno povedať, že sú väčšinou kontextovo nezávislé. Napríklad, koncept „strom“ je zväčša reprezentovaný rovnakým amodálnym symbolom (alebo štruktúrou) vo všetkých prípadoch, a ak nie, tak ku symbolu reprezentujúceho strom treba pridať iný symbol (alebo štruktúru), ktorý daný kontext špecifikuje (pozri Harnadovu definíciu symbolového systému v časti 2.1).

Otázka reprezentácií je relevantná aj v prípade konekcionizmu, kde má však inú povahu. Keďže vieme, že dlhodobou pamäťou natrénovanej siete sú spojenia medzi neurónmi i so svojimi hodnotami, znalosť je v sieti distribuovaná, a reprezentovaná ***implicitne a numericky***. Analýzou natrénovanej siete však vieme zväčša identifikovať, čo konkrétne neuróny a spojenia medzi nimi zabezpečujú, aj keď komplikáciou je práve distribuovanosť informácie. Okrem toho, interné reprezentácie v neurónovej sieti bývajú prirodzene kontextovo závislé (napríklad, pozri príklad Elmanovej siete v časti 5.6 , ktorá si vytvorí vnútorné kontextovo-závislé sémantické reprezentácie slov). Kontextová závislosť v kontexte distribuovaných systémoch má zaujímavú implikáciu, pretože sa môžeme pýtať na mieru distribuovanosti. Existuje v mozgu neurón, ktorý páli ***vtedy a len vtedy***, keď na senzorickom vstupe (nie nevyhnutne vizuálnom) je nejaký strom? Táto úvaha vedie k tzv. bunkám starej mamy (grandmother cells), ktoré túto podmienku spĺňajú (daný neurón reprezentuje moju starú mamu, t.j. je jedinečný). Máme viacero dôvodov predpokladať, že neurónov s touto vlastnosťou je v mozgu pravdepodobne veľmi málo (medzi také môžu patriť napr. bunky, ktoré detegujú konkrétne ľudské tváre). Takže reprezentácie v kontexte neurónových sietí je lepšie chápať ako vnútorné aktivácie systému pri danom vstupnom podnete v danom kontexte. Doslovné chápanie pojmu reprezentácia je mätúce.

Radikálna téza v doméne dynamických systémov nepredpokladá existenciu reprezentácií. Dynamické procesy pracujú v spojitom čase, a preto v žiadnom časovom okamihu nemôžeme predpokladať, že niečo z vonkajšieho sveta by mohlo byť aktuálne reprezentované vnútri systému, pretože všetko sa stále mení. Antireprezentačný postoj sa objavil aj v umelej inteligencii, keď Brooks ([13]) prišiel s alternatívou k tradičnej

[10] No už bolo možné hovoriť o reprezentačnej teórii mysle, ktorá pracovala s konceptami ako sú myšlienky, presvedčenia, túžby, či obrazy, a ktorá sa v kognitívnej vede voľne zamieňa s výpočtovou teóriou mysle.

symbolovej architektúre. Tzv. ***subsumpčná architektúra***, aplikovaná v oblasti robotiky, zahŕňa hierarchické usporiadanie vrstiev, ktoré zabezpečujú organizáciu správania robota podľa priorít (napr. cieľ robota dostať sa na nejakú pozíciu môže byť prechodne podradený cieľu dobiť si akumulátor). Vrstvy medzi sebou komunikujú, pričom využívané výpočtové prostriedky sú symbolové. Napriek tomu si takáto architektúra nevyžaduje uchovávanie vnútorných reprezentácií, pretože správanie robota je vždy len reakciou na vonkajší podnet (ide o reaktívny systém). Tento trend v robotike mal snahu priblížiť umelú inteligenciu ku kognitívnej vede, aj keď nezachytáva aspekty vyššej kognitívnej činnosti, ktorá sa nezaobíde bez vnútorných mentálnych reprezentácií.

Dynamická paradigma je teda výstižná pre opis tej časti prostredia, s ktorou človek môže interagovať, a na ktorú má tak povediac „dosah." Medzi percepciou a motorikou je istá asymetria v tom zmysle, že človek dokáže spracovávať signály z celého pre neho vnímateľného sveta (vďaka senzorom), no motorický repertoár a jeho dosah je obmedzený. Napriek tomu si vytvárame vnútorný model prostredia obsahujúci aj tie nedostupné časti, v prípade ktorých už nemožno hovoriť o nejakej slučke. Takže mentálne reprezentácie zrejme majú buď senzomotorické charakteristiky alebo len perceptuálne charakteristiky, čo je v súlade s teóriou ukotvenej kognície, a čo sa premieta aj do reprezentácie významov ([44]).

4.2 Významy a sémantika

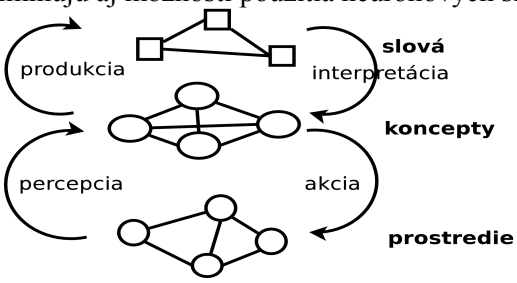
Otázka vlastností mentálnych reprezentácií a s ňou súvisiaci problém významov (sémantiky) patrí ku kľúčovým otázkam, ktoré sa kognitívna veda snaží riešiť. V tejto súvislosti boli identifikované dve kľúčové otázky: (1) ako sú významy vnútorne reprezentované a (2) či sú ukotvené vo svete. Podľa niektorých autorov sú tieto otázky nezávislé ([126],[134]). Otázka povahy vnútorných (mentálnych) reprezentácií závisí od paradigmy (hlavne symbolová verzus subsymbolová) a dá sa na ňu pozerať ako na pretrvávajúci spor o tom, či myseľ reprezentuje významy pomocou (symbolových) propozícií ([117]) alebo pomocou (subsymbolových) obrazov ([79]). Táto dichotómia je konzistentná s tvrdením, že (1) tradičné symbolové teórie uzatvárajú významy do separátnej sémantickej pamäti (uloženej niekde v mysli), ktorá je založená na amodálnych (t.j. modálne nezávislých) symboloch a operáciách s nimi, a že (2) teórie ukotvenej kognície a jazyka predpokladajú, že významy musia byť reprezentované v rôznych modalitách, v rámci senzomotorickej skúsenosti človeka so svetom ([7]).

Problém ako ukotviť významy identifikoval Harnad ([65]) ako **problém ukotvenia symbolov**. Formulácii tohto problému predchádzali dva iné významné míľniky v histórii kognitívnej vedy. Prvým bol Turingov ([143]) návrh, ako operacionalizovať otázku zistenia inteligencie stroja pomocou tzv. imitačnej hry, ktorá sa v literatúre stala známou ako **Turingov test**[11]. Tento test vnukol predstavu, že myslenie (a teda aj ľudskú kogníciu) možno chápať ako manipuláciu so symbolmi.

[11] Jednoduchšia forma Turingovho testu predpokladá, že človek komunikuje v prirodzenom jazyku s neznámym systémom cez terminál, a jeho cieľom je zistiť (na základe odpovedí), či za systémom je skrytý človek alebo počítač.

Druhým míľnikom bola kritika tohto myšlienkového experimentu, podľa ktorej iba na základe pozorovania vonkajšieho správania (napr. odpovedí systému), nemožno určiť či systém naozaj rozumie tomu, o čom hovorí ([130]). A ak nerozumenie, potom nemôže byť inteligentný. Searlov **argument čínskej izby** ilustruje situáciu, pri ktorej sa navonok môže zdať, že osoba zavretá v izbe, komunikujúca s okolím cez terminál, rozumie významom svojich odpovedí, pričom je zrejmé (nielen nám, ale aj samotnej osobe v izbe), že tomu tak nie je[12]. Problém čínskej izby naráža na otázku vlastnej (intrinsic) sémantiky (významu). Argument spočíva v tom, že ak je význam symbolov v symbolovom systéme systému nevlastný (v porovnaní s vlastnými významami v našich hlavách), tak to nemôže byť prijateľný model pre sémantiku podobnú tej ľudskej. Kognícia preto nemôže byť manipuláciou so symbolmi ([65]), pretože tie nie sú ukotvené. Základnou otázkou ukotvenia je, ako vytvoriť interný mechanizmus prepojenia mentálnych reprezentácií s vonkajším svetom takým spôsobom, aby mentálne reprezentácie boli systému vlastné, teda nezávislé na interpretácii externým pozorovateľom ([152]).

Problém ukotvenia symbolov je vo svojej podstate zložitejší, než sa na prvý pohľad zdá, a k jeho riešeniu pristupovali viacerí autori so svojimi modelmi (pozri prehľad v [149]). Tadeo a Floridi ([139]) prišli s návrhom kritéria tzv. nulového sémantického záväzku, ktoré podľa nich nespĺňa žiaden z existujúcich modelov, a preto dospeli k záveru, že problém ukotvenia symbolov sa doteraz nepodarilo vyriešiť. Avšak spomínané kritérium sa javí ako príliš reštriktívne, pretože maximálne zužuje úlohu dizajnéra ([148]). Napríklad, kritérium nepripúšťa ani existenciu učiacich mechanizmov v systéme (čo v prípade umelých systémov predstavuje účasť dizajnéra), čím sa principiálne eliminujú aj možnosti použitia neurónových sietí.[13]



Obr. 2. Vzťah medzi objektmi prostredia (referentmi), konceptmi a symbolmi jazyka. V symbolovom systéme sú koncepty amodálnymi symbolmi, ktoré vznikajú transdukciou z percepcie objektov vo svete, a ktoré sa mapujú na slová jazyka (symboly). Úloha ukotvenia sa týka samotných konceptov, pričom ich prepojenie s lingvistickým systémom je už symbolovo-symbolové. V prípade ukotveného systému sú koncepty modálne (napr. perceptuálne symboly) a preto sú priamo ukotvené v senzomotorickom systéme. Slová sú súčasťou lingvistického systému, ktorý je prepojený na koncepuálny systém a úloha ukotvenia symbolov (jazyka) spočíva v konštrukcii tohto prepojenia.

[12] Osoba A ktorá nehovorí po čínsky je zamknutá v miestnosti a komunikuje cez terminál s osobou B vonku, ktorá hovorí po čínsky. Predpokladáme však, že osoba A má k dispozícii kompletnú sadu pravidiel na manipuláciu s čínskymi znakmi, ktorá umožňuje osobe A odpovedať zmysluplne na ľubovoľnú otázku od osoby B.

[13] Okrem toho, ako správne poznamenáva Vavrečka, také kritérium nespĺňa ani učiace sa dieťa.

Obr. 2 ilustruje vzťah medzi vonkajším svetom a internou štruktúrou agenta, ktorý disponuje konceptuálnym systémom a lingvistickým systémom (jazykom). Požiadavky na ukotvenie sa v takomto systéme uplatňujú rôzne, podľa toho či ide o symbolový systém alebo ukotvený. Rozumnou požiadavkou pre ukotvenie významov (napr. slov) je, že musia byť prepojené na okolitý svet cez senzomotorický systém (napr. vnímanie červeného objektu spôsobí aktiváciu koncept červenej farby a prípadne aktiváciu slova červená), a že takéto prepojenie s prostredím sa vytvorí agent/človek sám pomocou svojich interných mechanizmov ([141]). Tým sa stávajú pre neho jedinečné, aj keď samozrejme musia byť o rôznych agentov dosť podobné, aby mohli úspešne komunikovať. Podobne to robí dieťa (pozri časť 2.4), ktoré si v ranom štádiu vývinu vytvára vnútorné reprezentácie fyzických objektov pomocou ich percepčných charakteristík a tzv. afordancií, čiže možností, ako s objektmi manipulovať ([61]). O významoch objektov môžeme hovoriť aj bez nutnosti ich pomenovávať (čiže používať jazyk), ako napríklad u detí v predverbálnom štádiu vývinu alebo v prípade zvierat ([44]).

Treba si uvedomiť, že samotné symboly existujú aj v prostredí, nielen v našich hlavách, vieme ich externalizovať a ukladať v rôznych formách, či už na papieri, v počítači, v obrazovej forme alebo ako v podobe ľubovoľných znakov (samozrejme, interpretácia symbolu bez kognitívneho systému nie je možná, takže tú hlavu potrebujeme). Napríklad, pri čítaní textu symboly priamo spracovávame a snažíme sa ich mapovať na koncepty. Podobná situácia nastáva, keď sa dieťa učí nové výrazy v prítomnosti rodiča. To znamená, že proces ukotvená kognícia pripúšťa účasť dizajnéra (napr. rodiča) ako súčasti prostredia v procese ukotvenia symbolov (čo však nepripúšťa spomínané kritériom nulového sémantického záväzku).

Ukotvenie významov je výzvou pre kognitívnu vedu. Konekcionistické modely majú dve výhody: (1) využívajú subsymbolové reprezentácie (ktoré môžu plniť úlohu perceptuálnych symbolov), a (2) vedia sa učiť.[14] Zostávajúcou kritikou konekcionizmu môže byť, že sa trénujú na vopred pripravených dátach. To znamená, že neurónová sieť síce používa distribuované subsymbolové reprezentácie, ktoré sú ukotvené v prostredí, na základe ktorého vznikli, avšak tie boli predpripravené externe, čím bola určená ich interpretácia. Inými slovami, takýto systém si nezjednáva svoj svet, lebo je mu vopred daný v podobe podnetov a odpovedajúcich významov. Napriek tomu môžeme povedať, že významy má ukotvené, ak symboly už existujú v prostredí a treba si ich osvojiť (napr. matka učí svoje dieťa pomenovávať objekty). Tento proces zahŕňa dva kroky: vytvorenie konceptuálnych reprezentácií a ich prepojenie na jazykové symboly. Toto je iná situácia než tie, keď uvažujeme populáciu agentov, ktoré sa naučia vzájomne komunikovať tak, že si vytvoria nový spoločný jazyk ([134],[140]). Tam významy ani pomenovania nie sú vopred dané, ale emergujú počas interakcií medzi agentmi.

[14] Prvú podmienku nespĺňajú symbolové systémy (teda drvivá väčšina z nich, ak pripustíme existujúce formy adaptačných mechanizmov v týchto systémoch, napr. V podobe aktualizácie bázy znalostí). S druhou podmienkou majú problém dynamické systémy, pretože sa neučia.

4.3 Učiace sa systémy

Aby sme to zhrnuli, z hľadiska typu výpočtov ako aj typu reprezentácií, konekcionizmus tvorí akýsi ***most medzi klasickou paradigmou a dynamickou paradigmou***, aj keď k dynamickým systémom má oveľa bližšie než ku klasickým systémom. Čo však jedine konekcionizmus ponúka, a čo je kľúčové, sú biologicky inšpirované mechanizmy učenia. Vďaka existencii týchto mechanizmov sa dajú neurónové siete využiť v návrhu umelých systémov, ktoré na istej úrovni abstrakcie nielen simulujú ľudské správanie, ale aj vývin kognície na ontogenetickej škále.

Tu treba podotknúť, že metódy učenia používané v neurónových sieťach sú len podmnožinou **metód strojového učenia** (napr. [91]), ktoré sa búrlivo rozvíjajú aj v novej umelej inteligencii. V tomto kontexte niektorí autori hovoria o **výpočtovej inteligencii**, ktorá si našla miesto na zemi, a ktorá sa chce odlišovať od tradičnej umelej inteligencie práve dôrazom na algoritmy učenia ([26],[32]). Nie je prekvapujúce, že neurónové siete sú považované za hlavnú súčasť metód výpočtovej inteligencie.

V ďalšom texte budeme hovoriť o neurónových sieťach, pričom budeme mať na mysli umelé neurónové siete, ktoré sa líšia typmi aktivít i svojou dynamikou od impulzných neurónových sietí, prednostne používaných vo výpočtovej neurovede ([86]). Tie sú ešte vernejšou aproximáciou biologickej reality než umelé siete, a vzťahujú sa na ne všetky kategórie učiacich mechanizmov. Výber úrovne aproximácie záleží od konkrétneho cieľa, pričom aj umelé neurónové siete ponúkajú vysvetlenia, konzistentné s neurobiologickými princípmi na istej úrovni abstrakcie (napr. pomocou impulznej neurónovej siete možno reprezentovať mentálny stav spojený s rozpoznaním nejakého objektu ako synchrónne oscilácie v nejakej populácii neurónov, čomu v umelej neurónovej sieti odpovedá aktivita jedného alebo viacerých výstupných neurónov, podľa toho či uvažujeme lokalistické alebo distribuované reprezentácie).

Konekcionizmus zažil prvý boom v druhej polovici 80-tych rokov, keď objavenie algoritmu učenia pomocou spätného šírenia chyby ([127]) pomohlo prekonať 15-ročný útlm súvisiaci s problémom naučiť vtedajšie modely (jednovrstvové siete) riešiť ťažšie (nelineárne) klasifikačné úlohy.[15] Niektorí priekopníci avizujú ďalšie možné oživenie v tejto výskumnej oblasti. Geoffrey Hinton z Univerzity v Toronte argumentuje úspešnosťou viacvrstvových ***generatívnych*** modelov neurónových sietí (napr. model DBN, [68]).[16] Jürgen Schmidhuber z IDSIA, známeho výskumného inštitútu v Lugane vo Švajčiarsku, zase predpovedá možnú renesanciu neurónových sietí vďaka ich čerstvým úspechom s viacvrstvovými modelmi v klasifikačných úlohách (pozri časť 5.4). Súčasne však možno pozorovať, že od 90-tych rokov nastal v konekcionistickom tábore čiastočný presun v orientácii na iné, štatisticky založené metódy učenia (nie primárne biologicky inšpirované), napr. bayesovské modely ([63]),

[15] Tento algorimus bol vynájdený už o viac ako dekádu rokov skôr, no na publicitu si zrejme musel počkať ([67]).

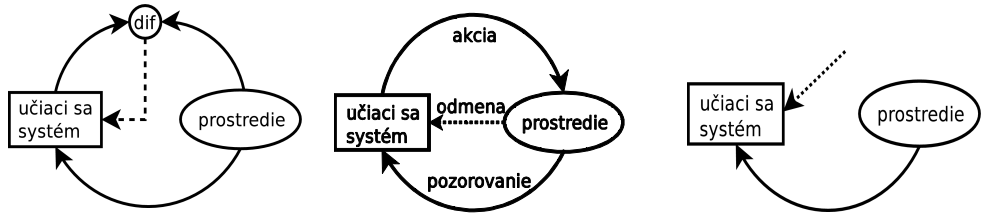
[16] Generatívne a diskriminačné modely tvoria dve hlavné triedy učiacich sa klasifikátorov. Diskriminačné modely (väčšina modelov neurónových sietí) pasívne reagujú na daný vstup a generujú výstup, zatiaľ čo generatívne modely zahŕňajú aj koncept očakávaní (expectations), ktoré pôsobia zhora nadol prostredníctvom spätných váhových spojení.

ktoré tiež ponúkajú kognitívne relevantné vysvetlenia na vyššej úrovni abstrakcie. Je reálne predpokladať, že oba prúdy budú ďalej napredovať súbežne ([59]).

5 Neurónové siete v kognitívnej vede

Možno vyššie spomínané fakty čitateľa presvedčili o tom, že modelovanie ľudskej kognície spojenej s učením je principiálne ťažké v klasickej paradigme. Pochopiteľne, snahy o klasické prístupy budú pretrvávať a sú opodstatnené, napr. v oblasti budovaní softvérových systémov, ktoré budú slúžiť človeku, ako napríklad znalostné/expertné systémy pre rôzne domény ([75]) alebo sémantický web ([28]). My sa budeme venovať neurónovým sieťam a ich využitiu v kontexte kognitívnej vedy.

Ako sme spomínali, kľúčovou vlastnosťou neurónových sietí, o ktorú sa kognitívna veda môže opierať, sú **mechanizmy učenia**. Tie môžeme rozdeliť do troch skupín (**Obr. 3**), a to v závislosti od toho, s akou mierou explicitnosti sa sieti podáva informácia o chybe na výstupe. Pri ***učení s učiteľom*** (supervised learning) každý neurón má k dispozícii takúto informáciu, a mení svoje vstupné váhy tak, aby sa jeho chyba zmenšovala. Výstupné neuróny dostávajú túto informáciu priamo, a pre skryté neuróny sa dá táto chyba vypočítať, ako to bolo odvodené v známom pravidle učenia pomocou spätného šírenia chyby ([127]). Druhú skupinu tvorí ***učenie s posilňovaním*** (reinforcement learning),, kde len sieť ako celok dostáva z prostredia informáciu (nie nevyhnutne v každom kroku) v podobe skalárnej veličiny o tom, „ako sa jej darí“ (tento typ učenia sa u nás tiež nazýva učením „pomocou odmeny alebo trestu,“ čo vystihuje význam spätnej väzby). Napokon, pri ***učení bez učiteľa*** (unsupervised) sieť nemá k dispozícii nijakú explicitnú informáciu o chybe, učí sa pomocou samoorganizácie. To znamená, že jej správanie závisí od konkrétneho učiaceho pravidla, ktorý vedie k nejakej forme extrakcie štatistických korelácií vo vstupných dátach. Všetky tri skupiny učiacich pravidiel sú pravdepodobne relevantné pre rôzne časti mozgu, pričom najväčší význam pre mozgovú kôru ako predpokladaný hlavný neurálny korelát kognitívnych funkcií, má asi samoorganizácia ([30]).



Obr. 3. Schémy troch paradigiem učenia: (a) s učiteľom, (b) s posilňovaním, (c) bez učiteľa.

Správna forma učenia vedie k schopnosti siete **zovšeobecňovať** (generalizovať) svoje znalosti a správne reagovať aj v nových situáciách. Neurónová sieť hľadá počas učenia štrukturálne vzťahy na (trénovacích) príkladoch a túto znalosť potom uplatňuje pri testovaní. Tu platia poznatky známe z výpočtovej teórie učenia (platné aj pre iné adaptívne modely), ktoré dávajú návod, ako sieť učiť a vyhnúť sa preučeniu (čo vedie k slabému zovšeobecňovaniu).

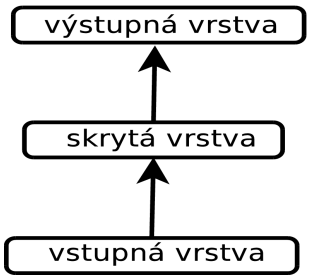
Prirodzenou aplikačnou oblasťou je vývinová kognitívna robotika (pozri prehľad napr. v [4]), ktorá využíva **konštruktivistický prístup** k budovaniu umelých agentov

(robotov), odlišný od tradičnej dizajnérskej umelej inteligencie, s cieľom vybaviť agentov vhodnými učiacimi mechanizmami, ktoré by im umožnili nadobúdať a využívať znalosti počas ontogenézy ([105]).[17] V tejto oblasti nastáva oživenie, hnané víziou, že modelovanie kognitívnych procesov nemožno dosiahnuť bez prepojenia na nadobúdané senzomotorické zručnosti vyvíjajúceho sa vteleného agenta, situovaného v sociálnom prostredí ([104]). Pri návrhu mechanizmov učenia sú užitočné poznatky z kognitívnej neurovedy a vývinovej kognitívnej psychológie, ktoré ponúkajú empirické dáta. Táto oblasť je v podstate veľmi široká, pretože spektrum zručností, ktoré sa malé dieťa musí naučiť, je veľmi široké, spomeňme napríklad vnímanie objektov a ich rozpoznávanie, vizuálno-motorická koordinácia, referenčné rámce využívané pri priestorovej kognícii, zdieľanie pozornosti, ale aj tie vyššie ako prepojenie kognície na jazyk, ukotvenie významov, problém systematickosti v jazyku, a iné. My sa pozrieme konkrétnejšie na niektoré z nich.

5.1 Vizuálno-motorická koordinácia

Jednou z prvých zručností, ktoré si malé dieťa osvojuje, je senzomotorická koordinácia (pozri prehľad niekoľkých modelov v [41]). Jej skorou fázou je vizuálno-motorická koordinácia, kde možno rozlíšiť dva hlavné podciele: dosahovanie objektov (reaching) a uchopovanie objektov (grasping). Táto činnosť spočíva v snahe agenta dať do súvisu vizuálnu informáciu o cieľovom objekte (a prípadne koncového bodu ramena) s vnútornou proprioceptívnou informáciou o pozícii ramena. Existuje veľa behaviorálnych, neurofyziologických a zobrazovacích poznatkov o tom, že mozog si konštruuje interné predikčné modely na tento účel ([74]). Tradičná teória vizuálno-motorického riadenia predpokladá koexistenciu dopredných a inverzných modelov, ktoré spolupracujú. ***Dopredné*** interné modely predpovedajú senzorické dôsledky agentom vykonávaných akcií, t.j. ich vstupom je motorický signál a výstupom senzorická reprezentácia („kde budem vidieť moju ruku, keď s ňou vykonám konkrétnu akciu?“). ***Inverzné*** interné modely fungujú naopak, vypočítavajú potrebné motorické pokyny na dosiahnutie cieľa, t.j. ich vstupom je senzorická informácia a výstupom sú motorické akcie („aký pohyb musím vykonať, aby som sa dotkol rukou objektu?“). Oba modely sa ovplyvňujú a môže ich byť viac. Výstup inverzného modelu môže slúžiť ako vstup dopredného modelu, ktorého význam podľa súčasných teórií spočíva v napomáhaní pri potrebe korekcie trajektórie (napr. v prípade čiastočného prekrytia cieľového objektu, ktoré preruší vizuálny vstup) a v kompenzácii oneskorenia spôsobeného vizuálno-motorickou slučkou ([29]). Inverzný model využíva vonkajší oblúk cez prostredie, dopredný nie. Je prirodzené oba modely realizovať pomocou neurónových sietí, pretože vizuálno-motorická koordinácia sa deje v spojitom prostredí. Typickým prístupom je použitie dopredných sietí učených s učiteľom (**Obr. 4**), pretože v prípade inverzného modelu sa predpokladá dostupnosť chyby z prostredia, a v prípade dopredného modelu sa chyba vypočítava vnútorne, porovnaním senzorickej predikcie s tou skutočnou.

[17] Je potešiteľné, že aj v bývalom Československu sa myšlienky kognitívnej robotiky objavovali už pred 30-imi rokmi (Havel, 1980).



Obr. 4. Dvojvrstvová[18] dopredná neurónová sieť (s neurónmi typu perceptrón). Šípky označujú úplné prepojenie medzi susednými vrstvami, t.j. od každého neurónu nižšej vrstvy vedie váhované spojenie ku každému neurónu v nasledujúcej vrstve (čím dostávame dve matice parametrov). Počty vstupov a výstupných neurónov sú vo všeobecnosti rôzne, závisia od zvoleného kódovania použitých dát. Počtom skrytých neurónov určujeme zložitosť modelu. Takáto sieť je vhodná pre úlohy, ktoré sa dajú formálne opísať ako hľadanie nelineárneho vzťahu medzi vstupmi a výstupmi na základe predložených párov (vstup, požadovaný výstup). Perceptróny sa najčastejšie učia pomocou metódy spätného šírenia chyby, ktorá podľa gradientu chyby v priebehu učenia minimalizuje rozdiely medzi skutočnými a požadovanými výstupmi.

Pri voľbe učiacej paradigmy hrá rolu aj kognitívna prijateľnosť modelu, to znamená realistickosť predpokladov, na ktorých je model založený. Napríklad, v kontexte vizuálno-motorickej koordinácie môže byť predpoklad o dostupnosti chybového signálu otázny, pretože to predpokladá znalosť o požadovanej motorickej aktivite (t.j. ako keby matka učila dieťa siahať na objekty tak, že by mu vzala ruku do svojej). Preto prijateľnejšou alternatívou (aj keď zložitejšou z hľadiska učenia) je učenie s posilňovaním, pretože tu sa predpokladá „mäkšia“ verzia spätnej väzby. Tou môže byť napríklad vzdialenosť medzi rukou a objektom (ktorú vizuálny systém odhaduje), ktorá nehovorí priamo, ako pohnúť rukou (systém to musí vypočítať; pozri časť 5.2). V kontexte kognitívnej robotiky je dôležitým krokom to, že boli navrhnuté rozšírenia konceptu učenia s posilňovaním (pôvodne navrhnutého pre diskrétnu doménu) aj pre spojitú doménu, pre ktoré sú neurónové siete vhodné ([31]). V kontexte učenia s posilňovaním to znamená, že diskrétne tabuľky sa nahradia aproximátormi funkcií (realizované napr. doprednými sieťami), ktoré umožňujú generalizáciu na nové vstupy. Príkladom tohto prístupu je nami simulovaný a vylepšovaný model trénovaný pomocou algoritmu CACLA ([146]), ktorý využíva dve dopredné siete: jednu na generovanie akcií (tzv. aktér) a druhú na odhad očakávanej odmeny v jednotlivých stavoch, konkrétne pozícií ruky od cieľa (tzv. kritik). Model sa nám podarilo nastaviť tak, že sa dokáže naučiť trajektórie z ľubovoľného počiatočného do ľubovoľného koncového bodu v pracovnom priestore s veľkou presnosťou ([78]).

Pochopenie mechanizmov spojených s predvídaním a ohodnocovaním motorických akcií je veľmi dôležité, nakoľko tieto typy mechanizmov sa môžu podieľať aj na vyššej kognitívnej činnosti (napr. plánovanie a jazyk) a pravdepodobne mozog využíva rovnaké alebo podobné mechanizmy aj na tejto úrovni abstrakcie. Napríklad, dopredné modely sa pravdepodobne podieľajú pri porozumení

[18] Vstupnú vrstvu nepovažujeme za vrstvu neurónov. Sú to v podstate len receptory, a nemajú aktivačnú funkciu, ktorá realizuje nelineárnu funkciu medzi vstupom a výstupom neurónu.

pozorovaným akciám iných agentov v rámci systému tzv. zrkadliacich neurónov ([97]). Taktiež sa predpokladá, že slúžia ako neurálny korelát vlastnej vôľovej činnosti subjektu („sense of agency“), ktorý mu umožňuje odlíšiť seba od iného agenta ([27]).

5.2 Referenčné rámce

Použitie doprednej neurónovej siete priamo na prevod medzi vizuálnymi súradnicami (na vstupe) a motorickými súradnicami (na výstupe) je najjednoduchší spôsob, ako to dosiahnuť. Takéto modely ponúkajú vysvetlenie, „ako to funguje“ na pomerne vysokej úrovni abstrakcie, ak si uvedomíme, že v mozgu tomu odpovedá komunikácia medzi troma lalokmi: spánkovým, kde sa spracovávajú vizuálne informácie, predným, kde sa spracovávajú motorické informácie, a temenným, kde sa realizuje prevod medzi **referenčnými rámcami** (súradnicovými systémami). Temenný lalok teda predstavuje oblasť, kde sa robia prevody medzi perceptuálnymi súradnicami a motorickými signálmi, ktoré treba vygenerovať, aby sa rameno (či noha) dostalo na požadované miesto v 3D priestore. Tento proces je však zložitejší, ak si uvedomíme, že mozog využíva viacero referenčných rámcov – napríklad retinocentrický (pozícia objektu na sietnici oka), hlavocentrický (pozícia objektu vzhľadom ku hlave) a i. – a prepočítava súradnice medzi nimi (napr. [114]). To je pri priestorovom vnímaní potrebné pri mnohých úlohách, napríklad na to, aby sme pri pohybovaní hlavy s pohľadom zafixovaným pre seba dopredu (t.j. bez pohybu očí) dokázali vnímať nejaký objekt v zábere (receptívnom poli) ako statický napriek tomu, že jeho súradnice na sietnici sa budú meniť.

Takéto prepočty súradníc medzi okom, hlavou a telom, experimentálne objavené u makakov v posteriórnej časti temenného laloka (Brodmannova area 7a), sa dajú vysvetliť pomocou doprednej neurónovej siete s jednou skrytou vrstvou neurónov (**Obr. 4**). Prvý takýto model ([153]) bol trénovaný práve na takýto prevod súradníc medzi sietnicou a hlavou, a to tak, že na vstup dostával podnety s retinocentrickými súradnicami ako aj informáciu o polohe oka v rámci hlavy (t.j. smer pohľadu) a na výstupe generoval pozíciu objektu odozvy s hlavocentrickými súradnicami. Zaujímavé bolo zistenie, že sieť rieši úlohu na skrytej vrstve podobným spôsobom ako mozog. Skryté neuróny si vytvorili svoje vlastné receptívne polia, ktoré sa kvantitatívnym spôsobom líšili v citlivosti daného neurónu buď na jeden vstup (pozícia objektu na sietnici) alebo na druhý vstup (smer pohľadu oka vzhľadom ku hlave). Lineárnou kombináciou aktivít skrytých neurónov sieť dosiahne na výstupe pozíciu objektu v priestore, nezávislú (invariantnú) od natočenia hlavy i nasmerovania očí. Tieto rozmanité samoorganizované receptívne polia charakterizujú **priestorový zisk** (spatial gain) neurónu a zabezpečujú vlastne moduláciu odozvy neurónu (jeden vstup moduluje citlivosť neurónu na druhý vstup, resp. naopak). Neuróny s takýmito charakteristikami boli objavené už na viacerých miestach v mozgu a ***ziskové polia*** sú považované za univerzálny mechanizmus neurálnych výpočtov v mozgu ([129]).

Analogicky možno pomocou doprednej neurónovej siete modelovať prepočítavanie medzi percepčnými súradnicami a motorickými súradnicami (pozícia ruky) pri zámere siahnuť na objekt ([25]). Dokonca nedávno bol navrhnutý model opäť na báze dvojvrstvového perceptrónu, inšpirovaný novými neurovednými poznatkami,

ktorý implementuje ***dvojitú moduláciu*** ziskových polí (compound gain field), pretože pozícia cieľového objektu vzhľadom k pozícii ruky sa počíta z pozície tohto objektu vzhľadom k oku, modulovanej pozíciou oka a súčasne aktuálnou pozíciou ruky ([18]).

Človek na reprezentáciu priestorových vzťahov teda využíva rôzne referenčné rámce, všetky uvádzané príklady sa týkali vlastnej (ego) perspektívy. Dieťa najprv pozná len ***egocentrický rámec***, a až neskôr sa naučí chápať priestorové vzťahy aj z relatívnej perspektívy (***alocentrický rámec***), napr. z pozície iného človeka alebo pozorovaného objektu. Pre úplnosť dodajme, že existuje aj absolútny referenčný rámec, ktorý znamená priestorovú orientáciu človeka podľa svetových strán. Jednotlivé rámce v mozgu koexistujú, spolupracujú a subjekt medzi nimi dokáže prepínať ([14]). Okrem toho, používanie referenčných rámcov závisí od kultúry a je ovplyvnené aj jazykom ([88]). Neurálne modelovanie týchto procesov je len v začiatkoch. Napríklad, jeden model využíva atraktorové neurónové siete so spojitou dynamikou, kde dlhodobá pamäť sa modeluje pomocou alocentrických reprezentácií a krátkodobá pamäť pomocou egocentrických reprezentácií, modulovaných smerovaním vizuálnej pozornosti ([15]). Najnovšie poznatky o prepínaní medzi rámcami pochádzajú aj zo štúdie selektívnej vizuálnej pozornosti, ktorá sa bežne považuje za egocentrický mechanizmus. Frischen, Loach a Tipper ([55]) pomocou experimentov poukazujú na to, že pozorovateľ dokáže aktivovať aj ekvivalentný alocentrický mechanizmus, pomocou ktorého dokáže cielene zameriavať vizuálnu pozornosť na rôzne objekty z perspektívy iného subjektu.

5.3 Zrkadliace neuróny

S problematikou referenčných rámcov súvisí schopnosť človeka (alebo zvieraťa) rozumieť správaniu iného jedinca. Schopnosť predpovedať kroky pozorovaného pohybu (napr. človek pozoruje iného človeka, ako siaha po šálke) by sa dala vysvetliť pomocou „prenesenia sa do kože aktéra.“ Toto vysvetlenie pozostáva z dvoch krokov: najprv sa subjekt „prepne“ do perspektívy pozorovaného subjektu (aktivuje sa mu alocentrický rámec), a potom simuluje sledovaný pohyb, namapovaný na mechanizmy vlastnej vizuálno-motorickej koordinácie, ktoré má už vyvinuté na základe predošlej vlastnej skúsenosti. V tomto kontexte treba spomenúť existenciu tzv. **zrkadliacich neurónov**, objavených najskôr u makakov ([121]) a len nedávno u človeka ([99]), ktoré sú považované za dôležitý komponent neurálnych obvodov umožňujúcich spektrum schopností človeka v oblasti sociálnej kognície, počnúc porozumenia pozorovaným akciám, schopnosťou imitovať, až po čítanie mysle a porozumenia emóciám pozorovaného subjektu ([56]). Je možné, že zrkadliace neuróny zohrali úlohu aj v evolúcii jazyka ([3]).

Tieto sociálne orientované schopnosti nie sú pravdepodobne človeku vrodené, musí sa ich naučiť, čo sa deje počas prvých rokov života. Tieto procesy musia vedieť vysvetliť aj výpočtové modely. Existujú viaceré modely na báze neurónových sietí, ktoré simulujú proces porozumenia akciám, no s využitím viacerých zjednodušení. Jedným z nich je napríklad to, že využívajú objektocentrický rámec, vďaka ktorému predpokladajú, že vykonávanie pohybu a i jeho pozorovanie generujú tie isté súradnice (čiže nezávislé od agenta). Otázka modelovania porozumenia motorickým akciám nie

je v súčasnosti uspokojivo vyriešená, pretože existujú rôzne vysvetlenia. Všetky modely síce riešia prepojenie senzorických a motorických signálov (s využitím inverzného a prípadne i dopredného modelu), ale robia to iným spôsobom, za iných predpokladov a s využitím inej paradigmy učenia (všetky tri paradigmy majú svoje zastúpenie). Kritický prehľad konceptuálnych a výpočtových modelov, hlavne na báze umelých neurónových sietí možno nájsť v práci Farkaš, Malý a Rebrová, [43]. Aj keď to zatiaľ tak nie je, dá sa predpokladať, že modelovanie zrkadliacich neurónov sa stane súčasťou vývinovej kognitívnej robotiky.

5.4 Rozpoznávanie objektov

Túto schopnosť dieťa nadobúda pomerne skoro (počas prvého roku života). Vie sa, že vizuálny systém rieši úlohy typu „čo“ a „kde“ rôznymi časťami mozgu, ktoré sú však prepojené ([144]). Inými slovami, „where“ systém, situovaný v temenných lalokoch (v oboch hemisférach) rieši otázky pozície objektu (v rôznych referenčných rámcoch spomínaných vyššie) bez jeho identifikácie, a naopak, „what“ systém sa orientuje na identifikáciu objektu (na ktorý máme zameranú pozornosť), nezávisle od jeho presnej pozície[19], veľkosti a natočenia. Úlohou určovania pozície objektu sme sa zaoberali v predchádzajúcej časti, teraz sa pozrime na úlohu rozpoznania. Invariantné rozpoznávanie je prirodzenou požiadavkou nielen z pohľadu potrebnej úspešnosti rozpoznania, ale aj z hľadiska kognitívnej relevancie: toto dokáže i dieťa. Jedným z prvých neurálnych modelov, ktorý čiastočne riešil otázku invariantnosti, a to hierarchickým spôsobom, bol štvorvrstvový perceptrón ([83]), trénovaný s učiteľom (štandardnou metódou spätného šírenia chyby) a testovaný na úlohe rozpoznávania rukou písaných číslic (databáza MNIST). Riešenie problému spočíva vo využití dvoch princípov: architektúrneho a výpočtového. Architektúrny princíp spočíva v tom, že každý neurón vyššej vrstvy v rámci svojej skupiny dostáva vstupy z časti receptívneho poľa neurónov nižšej vrstvy (t.j. časti vstupného priestoru, ktorú vidí) a susedné neuróny majú receptívne polia mierne posunuté (vidia iné časti vstupu), a to topografickým spôsobom. Druhým princípom je, že neuróny v rámci skupiny zdieľajú svoje dopredné váhy (weight-sharing), t.j. majú ich stále rovnaké, čím sa účelne redukuje počet voľných parametrov a podporuje schopnosť zovšeobecnenia.[20] Tieto dva princípy zabezpečia, že skupina neurónov začne fungovať ako detektor nejakého príznaku (napr. čiary nejakej orientácie) v rôznych pozíciách na sietnici. Ak máme viacero skupín na danej vrstve vedľa seba, tie sa naučia detegovať iné príznaky. Rôznorodosť príznakov dosiahne sieť sama v rámci učenia, čo je peknou ilustráciou toho, ako učiaci signál „naviguje“ detektory príznakov k ich špecializácii (ktorá závisí od počiatočných hodnôt váh, a tie bývajú náhodné). Ak nejaký neurón ešte vyššej vrstvy vidí časť neurónov nižšej vrstvy, stáva sa invariantným detektorom tohto

[19] Tu mám na mysli relatívne malé posuny v rámci uhla pohľadu, keď sa na objekt pozeráme, no každý malý posun znamená veľké zmeny v aktiváciách receptorov na sietnici, preto predstavuje výpočtový problém.

[20] Toto je z výpočtového hľadiska šikovný krok, no z biologického hľadiska je takáto úprava ťažko prijateľná, aj keď susedné neuróny v kôre mávajú veľmi podobné charakteristiky (vrátane synaptických účinností), čím je zabezpečená redundancia a robustnosť.

príznaku. Tento proces sa opakuje aj vo vyšších skrytých vrstvách, pričom narastá miera invariantnosti, a výstupná vrstva predstavuje kategórie vstupov (10 číslic).

Od pôvodného modelu ([83]) bolo navrhnutých viacero vylepšení na báze neurónových sietí alebo iných metód strojového učenia, no pozoruhodným je nedávny model ([22]), ktorý svojou presnosťou (no nie časom trénovania) prekonal všetky predchádzajúce modely na vyššie spomínanej štandardnej množine rukou písaných číslic (MNIST). Model doprednej siete pozostáva z väčšieho množstva vrstiev (autori ich skúšali až 9), no využíva len štandardné pravidlo učenia pomocou spätného šírenia chyby ([127]), ktoré viacerí už považujú za „zastarané". Ako vysvitlo, ak sieť má dostatok času na trénovanie (urýchlenie výpočtu asi 50x realizovali pomocou grafických procesorov), dokáže sa dosť presne naučiť úlohu, až na úrovni ľudských schopností. Z pohľadu potrebného času a počtu príkladov však strojové učenie veľmi zaostáva za ľudskou kogníciou.

Ukazuje sa však, že návrh hlbokých hierarchických architektúr je asi cestou, ako dosiahnuť invariantné robustné rozpoznanie komplexných dát ([9]). Koniec koncov, hierarchické spracovanie informácie je inherentnou vlastnosťou aj architektúry mozgu, čo sa týka nielen percepcie ([46]), ale aj konania ([62]).

5.5 Akvizícia jazyka – lexikálna úroveň

Oblasť akvizície jazyka je zaujímavou doménou z toho dôvodu, že v sebe zahŕňa široké spektrum procesov, počnúc fonologickou analýzou, cez morfologickú a lexikálnu úroveň až po syntax a sémantiku viet.[21] Okrem toho, niektoré aspekty jazyka (napr. gramatika) priamo nabádajú na symbolové prístupy k modelovaniu. Nie náhodou boli spomedzi počítačových modelov jazyka donedávna dominantnými práve symbolové modely. Tie dokážu transparentne opísať niektoré aspekty (v podstate všetky tie, ktoré sa dajú zachytiť pravidlami), no na druhej strane nevedia vysvetliť niektoré kognitívne javy vznikajúce pri používaní jazyka. Preto sa popri klasických modeloch začala v 90-tych rokoch rozvíjať **konekcionistická psycholingvistika** ([20]) v oblasti modelovania jazyka, ale aj v kontexte ostatných kognitívnych procesov ([119],[96]). Prirodzený jazyk pritom vzbudil najviac záujmu, ale aj kontroverzie ([108]). Podľa niektorých kritikov sa totižto (minimálne) niektoré aspekty jazyka vôbec nedajú vysvetliť konekcionisticky, ale len pomocou symbolových modelov. Jadro kontroverzie súvisí s už spomínanou otázkou, aká je povaha mentálnych procesov prebiehajúcich v mozgu pri vyšších kognitívnych procesoch ([108]): či dokážu konekcionistické modely iba implementovať symbolové modely (***implementačný konekcionizmus***), alebo či mentálne procesy možno zredukovať na subsymbolovú úroveň, a v tom prípade konekcionizmus bude znamenať alternatívnu paradigmu k vysvetleniu povahy jazyka (***eliminačný konekcionizmus***). Odpoveď na túto otázku je pri súčasnej úrovni poznania zložitá, pretože pre viaceré jazykové fenomény existuje konekcionistické i symbolové vysvetlenie. V ďalšej časti budem ilustrovať oblasť konekcionistickej psycholingvistiky na pár príkladoch.

[21] Mohli by sme pokračovať nadvetnou úrovňou smerom ku pragmatike jazyka a diskurzu, ale pre naše prezentačné účely postačí, ak skončíme pri vetnej úrovni.

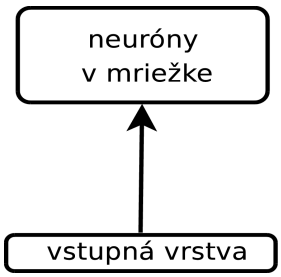
Detailnejší a obsažnejší prehľad modelov môže ctený čitateľ nájsť v prehľadových prácach ([19],[36],[123]).

V oblasti morfologického spracovania je najznámejším príkladom problém tvorenia minulého času slovies v angličtine. Ako vieme, väčšina anglických slovies získava v minulom čase koncovku /-ed/, avšak pri nepravidelných slovesách sú minulé tvary rôzne. Pre tento jav existujú dve vysvetlenia ([110]). Klasické vysvetlenie predpokladá existenciu ***dvoch ciest*** v mozgu, jednu pre pravidelné slovesá (aplikácia pravidla pridania koncovky /-ed/) a druhú pre nepravidelné slovesá (tabuľka párov neurčitok - minulý čas). Algoritmus pre rozhodovanie je takýto: mám neurčitok, ak nájdem tvar v tabuľke, použijem odpovedajúci minulý čas, ak nie, aplikujem pravidlo pre pravidelné slovesá. Na druhej strane, konekcionistické vysvetlenie predpokladá ***len jednu*** cestu, pretože dopredná neurónová sieť sa dokáže naučiť tvoriť minulý čas všetkých slovies a zovšeobecňovať svoju znalosť pri predložení nových tvarov. Neurobiologické poznatky o mozgu sa zdajú byť skôr v prospech existencie dvoch ciest ([110]), no symbolový model nevie uspokojivo vysvetliť **proces akvizície** minulého času slovies u dieťaťa, zatiaľ čo neurónová sieť áno (napr. [112]). Proces akvizície má tri typické fázy: najprv dieťa správne tvorí minulé tvary pravidelných i tých niekoľkých známych nepravidelných slovies, potom však začne nesprávne aplikovať pravidlo /-ed/ aj na nové nepravidelné slovesá, no napokon sa minulé tvary všetkých nepravidelných slovies naučí správne.[22]

Táto debata poukazuje na zaujímavý, všadeprítomný fenomén, s ktorým sa učiaci sa agent stretáva. Veľa dát, vrátane tých lingvistických, má vlastnosť tzv. **kváziregulárnosti**, čo znamená, že medzi dátami (či už medzi vstupmi, alebo v rámci vstupno-výstupných párov na trénovanie) existuje systematickosť (zákonitosti, pravidelnosti), ale aj viaceré výnimky (ako sme videli pri slovesách). Na vysvetlenie takýchto vzťahov sa v symbolovom prístupe predpokladá, že systematickosť je reprezentovaná pomocou explicitných **pravidiel**, zatiaľ čo **výnimky** sa zabezpečujú asociačným mechanizmom ([110]). Pri konekcionistickom vysvetlení potrebujeme len jeden systém a všetky vstupné dáta sa navzájom líšia iba stupňom konzistentnosti z pohľadu vstupno-výstupnej transformácie. To znamená, že aj podobnosť (systematickosť) má gradovaný a nie binárny charakter (pravidlo/výnimka). V prípade nepravidelných anglických slovies tiež možno nájsť ich vnútorné členenie na jednotlivé podskupiny (keep/kept, sleep/slept) s rôznymi veľkosťami až po veľmi nepravidelné slovesá (napr. go/went). Táto kváziregulárnosť odzrkadľuje rôzne nároky na (nelineárnu) transformáciu pri generovaní požadovaného tvaru, od čoho potom závisí aj poradie, v ktorom sa sieť minulý čas jednotlivých slovies naučí. Z pohľadu neurónovej siete teda ide o úlohu nelineárnej transformácie s rôznymi stupňami pravidelnosti. Podobný prípad kváziregulárnosti možno pozorovať i pri čítaní slov (systematickosť závisí od jazyka), spracovaní rečového signálu alebo pri artikulácii slov ([20]).

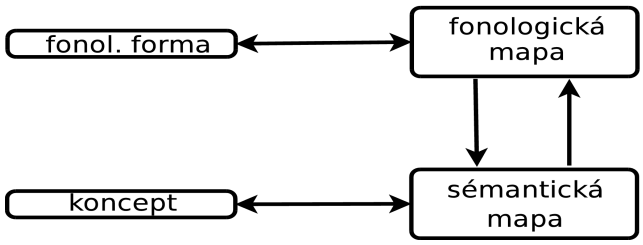
[22] Symbolové vysvetlenie tohto procesu by sa dalo asi formulovať takto: v prvej fáze dieťa len memoruje minulé tvary, v druhej fáze začne aplikovať pravidlo /-ed/ a až v tretej fáze nájde rovnováhu medzi oboma cestami. Debata okolo tvorby minulého času má svoj vývoj a celá problematika je o čosi zložitejšia, než sme tu uviedli (viac možno nájsť v [36]).

Analógiou kváziregulárnosti je známa teória prototypov a základných kategórií ([124]), pomocou ktorej možno vysvetliť kategorizáciu objektov u človeka (a iných živočíchov). Stupeň príslušnosti objektu do danej triedy možno charakterizovať ako spojitý (teda nie binárne ako v symbolovom systéme), čo závisí od prototypickosti objektu (príklad: vrabec je lepším prvkom kategórie vták než tučniak). Toto vysvetlenie má aj psychologickú podporu, pretože človek dokáže verbálne zhodnotiť, nakoľko sa mu pozdáva daný objekt ako prvok danej kategórie.



Obr. 5. Samoorganizujúca sa mapa (SOM). Vstupy sú úplne prepojené adaptovateľnými váhami so všetkými neurónmi v mape, usporiadanými do pravidelnej 2D mriežky. SOM zobrazí distribuovane reprezentované vstupy na lokalistické odpovede v mape (zobrazenie redukujúce dimenziu). Požadovanou vlastnosťou tohto zobrazenia je vlastnosť zachovania topológie: blízke vstupy sa zobrazia na blízke neuróny v mape. Takéto zobrazenie možno učením dosiahnuť práve vďaka existencii neurálnej štruktúry v mape (každý neurón „vie", akých má susedov), čo sa využije pri mechanizmoch súťaženia a kooperácie medzi neurónmi. Reakcie neurónov SOM sa organizujú podľa štatistických vlastností vstupov.

Ako ďalší príklad spomeňme modelovanie tzv. lexikálnych máp, kde možno využiť neurónové siete so samoorganizáciou. Známy model samoorganizujúcej sa mapy ([77],[35]) umožňuje modelovať rôzne lexikálne efekty a je biologicky zaujímavý. SOM (**Obr. 5**) realizuje zobrazenie, ktoré zachováva vzťahy podobnosti medzi vstupnými dátami, čo je vlastnosť niektorých častí mozgovej kôry, kde máme napríklad rôzne senzorické mapy (napr. v doméne vizuálnej alebo akustickej), alebo motorické mapy, ktoré si mozog interne vytvára. Predpokladá sa, že v mozgu môžu existovať i mapy vyšších, abstraktnejších entít, napríklad konceptov ([120]), čo sa ponúka ako vysvetlenie kognitívnych porúch pozorovaných u pacientov, napr. neschopnosť pomenovať predmety patriace do nejakej kategórie. Vysvetlenie spočíva práve v predpoklade o topografickom usporiadaní odpovedajúcich neurónov v mape: simulácia lokálnej poruchy v mape s veľkou pravdepodobnosťou poškodí len neuróny v jednej časti mapy, zodpovedajúce nejakej kategórii objektov.



Obr. 6. DevLex model lexikálneho vývinu. Dve rastúce SOM sú vzájomne asociatívne prepojené. Fonologická mapa sa organizuje na fonologických symboloch a sémantická mapa na reprezentáciách konceptov. Všetky spojenia sa aktualizujú pomocou samoorganizácie.

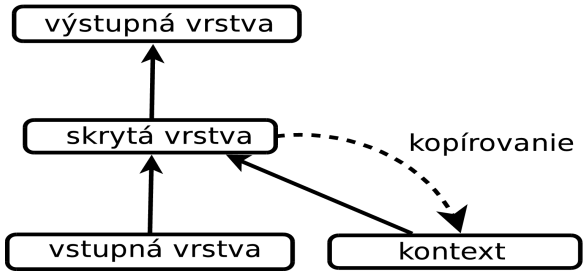
Niektoré modely založené na SOM majú komplexnejšiu architektúru, čo rozširuje ich použiteľnosť. Napríklad Dislex ([98]) predstavuje systém troch prepojených máp (fonologická a ortografická mapa sú prepojené so sémantickou mapou), čo umožňuje ich vzájomné aktivovanie. Tým sa dajú modelovať jazykové procesy, ako napríklad čítanie (ortograficko-sémantická linka) alebo artikulácia (sémanticko-fonologická linka), ale aj rôzne poruchy ako dyslexia alebo špecifické afázie (napr. poruchy pomenovania vecí), a to pomocou simulovaných lokálnych porúch v modeli. Na základe inšpirácie Dislexom navrhli Li, Farkaš a McWhinney ([84]) model DevLex (**Obr. 6**), pozostávajúci z dvoch navzájom prepojených rastúcich máp (fonologickej a sémantickej), ktoré majú tú výhodu, že nepotrebujeme vopred poznať vhodnú veľkosť siete, pretože tá sa upraví podľa dát. Rastúce siete sú ilustráciou maturačných procesov (funkčných a anatomických), ktoré sa predpokladajú v mozgoch cicavcov, a ktoré sa môžu týkať aj oblastí, spracovávajúcich jazyk ([118]). DevLex bol navrhnutý ako psychologicky prijateľný model vznikajúceho lexikónu u malých detí. Napríklad ukazuje, ako nielen gramatické kategórie (podstatné mená, slovesá), ale aj sémantické kategórie (napr. zvieratá, jedlo, atď.) postupne vznikajú v sémantickej časti modelu. DevLex taktiež vysvetľuje javy rôznych chýb pomenovania objektu, a to v závislosti od frekvencie slov a od sémantických podobností medzi slovami. Na druhej strane, DevLex je príkladom typického použitia neurónovej siete, kde vstupy z prostredia sú predpripravené, takže nemožno hovoriť o spätnoväzbovej slučke, ako je to v prípade učiaceho sa dieťaťa.

5.6 Akvizícia jazyka – vetná úroveň

Viaceré konekcionistické modely boli aplikované i na vetnej úrovni, kde je to už viac zaujímavé, lebo ku slovu sa dostáva syntax a sémantika[23]. Prvé modely sa zaoberali akvizíciou gramatiky, a tu kľúčovú úlohu zohrala priekopnícka práca Jeffreyho Elmana ([34]), ktorá spustila „lavínu“ používania **rekurentných neurónových sietí** pri spracovaní sekvenčných dát v jazyku i mimo neho. Elman zaviedol novú paradigmu trénovania nasledujúcej položky (item) v danej vstupnej sekvencii symbolov, napr. slov v prípade viet. Elmanova jednoduchá rekurentná sieť (simple recurrent network, SRN) sa trénuje na vetách nejakého jazyka, vždy slovo po slove, a snaží sa predikovať

[23] O lexikálnej sémantike sme hovorili v prípade lexikálnych máp, kde sme predpokladali existenciu sémantických príznakov slov, vytvorených dizajnérom.

nasledujúce slovo. Slová boli kódované lokalisticky, t.j. každý neurón svojou aktivitou reprezentoval práve jedno slovo. Napríklad vo vete John broke glass, ak je aktuálnym vstupom John, sieť má predikovať slovo broke. V ďalšom kroku sa broke stáva vstupom, a požadovaným výstupom je glass, atď. Túto schopnosť má nepochybne aj človek, ktorý pri počúvaní reči dokáže s istou presnosťou predikovať, čo rečník povie. Je tomu tak vďaka syntaktickým, sémantickým a pragmatickým obmedzeniam, ktoré determinujú povrchovú štruktúru viet.



Obr. 7. Schéma jednoduchej rekurentnej siete (SRN). V porovnaní s viacvrstvovým perceptrónom je obohatená o spätnú väzbu od skrytých neurónov. Plné šípky označujú úplné prepojenie medzi susednými vrstvami neurónov pomocou adaptovateľných váh. Čiarkovaná šípka označuje kopírovanie aktivít skrytých neurónov na kontextové vstupy, čím sa aktivity skrytých neurónov v aktuálnom kroku stanú kontextovými vstupmi v nasledujúcom kroku. Túto sieť možno trénovať aj pomocou štandardnej metódy spätného šírenia chýb, ale aj pomocou efektívnejších algoritmov. SRN je vhodná na niektoré časopriestorové úlohy, kde kontextové vstupy slúžia ako pamäť (s exponenciálnym zabúdaním do minulosti) a umožňujú sieti generovať výstup nielen podľa aktuálneho vstupu, ale aj podľa ľavého kontextu.

Aj keď predikcia nasledujúceho slova nie je finálnym produktom pri spracovaní jazyka (cieľom je porozumieť vete), predikcia je jedným z užitočných komponentov jazykového systému v ľudskom mozgu (človek takisto robí predikcie rôzneho typu, viz časť 5.1). Preto táto úloha vzbudila v 90-tych rokoch pozornosť i v konekcionistickej komunite. Z pohľadu paradigmy učenia ide o špeciálny typ učenia s učiteľom (**self-supervised**), pretože požadované výstupy pochádzajú zo vstupnej sekvencie. Elman (1990) aplikoval svoj model SRN na dve úlohy so sekvenčnými dátami. Prvou bola predikcia nasledujúceho (tlačeného) písmena v súvislom texte. Sieť teda v každom kroku prečítala aktuálne písmeno (symbol) a jeho ľavý kontext, a mala predikovať nasledujúcu písmeno (bez medzier medzi slovami). Elman ukázal, že predikcie začiatku nového slova boli vždy najhoršie. Tento prístup implikuje, že schopnosť predikcie nasledujúcej fonémy (v prípade hovorenej reči) môže byť primárnym mechanizmom, ktorý novorodenci používajú pri segmentácii slov ([128]).

Druhou úlohou bola predikcia nasledujúcich slov (tiež lokalisticky kódovaných) v korpuse jednoduchých viet (napr. John runs, Mary loves John, John broke glass, atď.). SRN sa naučila pomerne presne predikovať vhodných kandidátov v každom kroku (v jazyku danom stochastickou bezkontextovou gramatikou bola neurčitosť, takže v každom mohlo nasledovať viacero slov vhodnej kategórie), čím sa teda naučila gramatiku jazyka. Naučila sa ju však implicitne, pretože nezískala explicitnú znalosť o pravidlách (na rozdiel od symbolových modelov, kde pravidlá gramatiky sú

explicitne dané). Schopnosť naučiť sa gramatiku sa Elmanovi podarilo vysvetliť, keď analyzoval interné reprezentácie po natrénovaní, ktoré si SRN vybudovala vo svojej skrytej vrstve.[24] Zaujímavým zistením bolo, že tieto interné distribuované reprezentácie mali svoju štruktúru, takže podobné slová (vyskytujúce sa v podobných kontextoch) mali podobné interné reprezentácie, čo pri ich zhlukovaní viedlo k odhaleniu jednotlivých syntaktických i sémantických kategórií slov. Naučená sieť sa teda naučila gramatiku, čo dosiahla vytvorením sémantických reprezentácií na skrytej vrstve slov na základe minimalizácie výstupnej predikčnej chyby (viac o sémantických reprezentáciách a významoch pojednávam v časti 5.7).

Ako bolo povedané, Elmanove práce zohrali v histórii rekurentných neurónových sietí dôležitú úlohu a SRN je dodnes najčastejšie používaným modelom rekurentnej siete[25]. Predikčné modely sekvencií slúžili aj ako stimulácia pre empirické skúmanie a záujem o štatistické učenie, pretože sa ukázalo, že ľudia dokážu zachytávať štatistické závislosti na rôznych úrovniach. Táto schopnosť je často podvedomá (subjekt nevie, že sa dané závislosti učí, no jeho výsledky v teste sú toho dôkazom), a preto ide o **implicitné učenie** ([24]). To sa samozrejme netýka len jazyka, ale aj iných sekvenčných činností. V kontexte je jazyka je zaujímavé, že človek dokáže sledovať aj tzv. nesusedné závislosti ([101]), ako napríklad vo vzťažnej vete Janko, ktorý sedí vzadu, vidí mačku, kde v hlavnej potrebujeme naviazať sloveso vidí na podmet Janko. Táto schopnosť je nevyhnutná pri používaní jazyka, jej vysvetlenie závisí od pohľadu (zástanca symbolovej paradigmy by to vysvetlil tak, že subjekt využíva explicitnú gramatiku), no z hľadiska štatistického učenia nie je vôbec triviálna, pretože slovný materiál medzi cieľovými slovami je premenlivý a môže tak pôsobiť „rušivo“. Bolo však ukázané, že aj SRN sa dokáže za istých podmienok naučiť s istou presnosťou nesusedné závislosti v umelom jazyku (pozri [40] a tamojšie referencie).

Druhá línia konekcionistického výskumu súvisiaca so spracovaním jazykových sekvencií sa týka tzv. **syntaktickej systematickosti** jazyka (a mysle). Fenomén systematickosti ľudskej mysle sa priamo dotýka spomínaných paradigiem a možno ho vysvetliť pomocou analógie so schopnosťou človeka systematicky rozumieť jazyku. Ak niekto vie povedať alebo rozumie vete Janko ľúbi Aničku, bez problémov vie vytvoriť vetu Anička ľúbi Janka a porozumieť jej, pretože vie, že jazyk sa skladá z častí a vie, ako ich skladať pri tvorbe viet. Systematickosť v jazyku implikuje produktívnosť (schopnosť generovať teoreticky neobmedzené množstvo viet) a kompozičnosť (skladanie atomických znakov do zložených pomocou rekombinácie; viz. definícia symbolového systému v časti 2.1). V symbolových modeloch je problém systematickosti automaticky vyriešený, pretože používané reprezentácie a syntaktické operácie nad nimi (podľa explicitných pravidiel) priamo zabezpečujú

[24] Postup získania interných reprezentácií slov je nasledovný: po natrénovaní siete raz prejdeme trénovacou množinou a zaznamenávame aktivácie skrytej vrstvy pri každom vstupnom slove (v rôznych kontextoch). Potom aktivácie pre každé slovo spriemerníme, čím dostaneme kontextovo-nezávislé reprezentácie slov (niečo ako prototypy pri kategorizácii), ktoré môžeme znázorniť v hierarchickom diagrame a vidieť tak podobnosti medzi slovami.

[25] V ostatných rokoch sa začal častejšie skúmať aj jednoduchší model, rekurentná sieť s echo stavmi (pozri prehľad v [85]), ktorá sa dá rýchlejšie trénovať a v niektorých prípadoch postačí ako efektívna náhrada SRN.

produktívnosť symbolového systému. V neurónových sieťach takéto vlastnosti vôbec nie sú zrejmé (sieť si nemôže explicitne uložiť gramatiku vo svojej pamäti), a podľa niektorých kritikov ani principiálne možné. Fodor a Pylyshyn ([50]) vo svojom vplyvnom kritickom článku sformulovali argumentáciu zhruba nasledovne: „Ľudská myseľ je systematická, a preto jej interné reprezentácie sú štruktúrované. Keďže v konekcionistických systémoch neexistujú štruktúrované reprezentácie, nie sú (tie systémy) dobrými modelmi ľudskej mysle."

Vysvetlenie systematického správania stále predstavuje pre neurónové siete výzvu ([2]), aj keď nejaké lastovičky sa už objavili. Hadley ([64]) navrhol rozlíšenie medzi slabou a silnou syntaktickou systematickosťou. Neurónová sieť je slabo systematická, ak dokáže spracovať vety (t.j. správne predikovať) s novými kombináciami slov, pričom všetky slová sú na syntaktických pozíciách, na ktorých sa objavili už aj počas trénovania (napr. sieť trénovaná na vetách chlapec ľúbi dievča a pes naháňa mačku dokáže spracovať aj vetu pes naháňa dievča). Na druhej strane, silná systematickosť už vyžaduje, aby sieť vedela generalizovať aj v nových syntaktických pozíciách (napr. spracovať vetu pes naháňa chlapca, za predpokladu že podstatné meno chlapec sa nikdy nevyskytlo počas trénovania v úlohe objektu len subjektu). Bolo ukázané, že vlastnosť tzv. slabej syntaktickej systematickosti dokážu nadobudnúť rekurentné siete s rôznymi architektúrami a algoritmami učenia: SRN, ESN (Frank, 2006a) a rekurentná sieť so samoorganizáciou ([42]). Vlastnosť silnej systematickosti bola preukázaná v prípade ESN ([52]) a SRN ([11]). Tieto pozitívne výsledky eliminačného konekcionizmu oslabujú kritiku, podľa ktorej aj keď sa podarí natrénovať nejaký konekcionistický model so systematickým správaním, to nestačí, pretože konekcionizmus „ako taký" (vo všeobecnosti) by mal byť systematický; len vtedy ho môžeme považovať za kognitívne relevantný model ([50]). Treba však priznať, že dosiahnuté výsledky sa týkajú jednoduchších foriem jazyka, takže problém konekcionistickej syntaktickej systematickosti ešte nie je uspokojivo vyriešený.

5.7 Porozumenie vetám

Otázka významov a ich reprezentácie sa netýka len samotného jazyka, ale ide o širší koncept. Spomínali sme, že Elmanova SRN vytvárala (kontextovo-závislé) sémantické reprezentácie slov na skrytej vrstve, čo znamená, že jednotlivým aktivitám skrytých neurónov by sme mohli priradiť význam. Takýto spôsob reprezentovania významom je príkladom prístupu, ktorý môžeme nazvať štatistickým symbolizmom. Ide v podstate o učiace metódy, ktoré využívajú štatistické závislosti vo výskytoch slov (symbolov), teda ako slová po sebe nasledujú v nejakom korpuse (kovariácia). Inými slovami, význam slova, t.j. jeho sémantická reprezentácia, sa odvodí na základe výskytov tohto slova v rôznych kontextoch[26] (v prípade SRN ide o ľavý kontext, t.j. predchádzajúce slová). Tu však ostáva problémom ukotvenie symbolov (časť 4.2), pretože takéto významy sa vytvárajú súvzťažne, u každého slova pomocou ostatných slov, a preto tieto významy nie sú ukotvené v prostredí. Je to akoby sa niekto snažil učiť slová

[26] Kontext slova môže mať rôzne formy, napríklad môže ísť aj o počítanie výskytov slov v rôznych dokumentoch ([81]).

cudzieho jazyka len pomocou výkladového slovníka toho jazyka. Jediným zdrojom informácie je tu spomínaná kovariácia. Tým sa však tento spôsob tvorby významov odlišuje od tradičných symbolových systémov, kde sémantiku slov vkladá dizajnér a kde kovariácia slov ako zdroj informácie nehrá úlohu.

V prípade konekcionistických modelov hrá štatistika kľúčovú úlohu. Prvé modely využívali vopred pripravené sémantické reprezentácie ako požadované výstupy, ktoré mala sieť priradiť vetám na vstupe. Napríklad, model McClellanda a Kawamota ([94]) je realizovaný ako štandardná dopredná sieť, ktorá sa učí priradiť sémantické roly (agens, paciens, inštrument, modifikátor) jednotlivým konštituentom vety (podmet, prísudok, predmet, predložková fráza „s“), čo možno považovať za dôležitý krok pri komprehenzii (porozumení). Hlavným obmedzením modelu je to, že významy viet sú vopred dané a predpokladajú sa len vety generované podľa niekoľkých rámcov (templates) s požadovanými sémantickými rolami. Napriek tomu úloha mapovania viet na významy nie je triviálna, lebo priradenie rolí závisí od kontextu (vo vetách Ball hit girl a Ball moved je ten istý podmet raz inštrumentom a raz paciensom), alebo závisí od sémantiky slovesa (vo vetách Man ate pasta (with) spoon a Man ate pasta (with) cheese v prvom prípade predložková fráza „s“ modifikuje paciensa, zatiaľ čo v druhom prípade vyjadruje inštrument). V niektorých prípadoch ostáva nejednoznačnosť nerozlúštená (napr. vo vete Boy hit girl with ball nevieme, či ball je inštrumentom alebo modifikátorom paciensa girl). Model však dokázal tieto nejednoznačnosti zohľadniť a správne zovšeobecňovať (aj keď len na malej množine viet). Tento model sme predstavili na ilustráciu raného prístupu k reprezentácii významov, ktoré nevyužívajú prvky situovanosti ani vtelenia, a naopak zdieľajú charakteristiky s tradičnými symbolovými systémami (napr. rámce). Na tento model nadviazali ďalšie vylepšenia smerom ku zložitejším konštrukciám a architektúram. Súčasnú „špičku“ konekcionistického modelovania komprehenzie viet z hľadiska zložitosti architektúry a používaného jazyka predstavujú pravdepodobne dva modely: CSCP ([122]) a INSOMNet ([92]).

Rohde ([122]) navrhol integrovaný model komprehenzie a produkcie viet (CSCP), ktorý sa trénuje na vetách (rôznej zložitosti) generovaných podľa gramatiky zjednodušeného anglického jazyka (ale pozoruhodnej zložitosti). Význam vety pozostáva z množiny propozícií, ktoré sú reprezentované distribuovane a sú explicitne prezentované sieti počas učenia. Jedna časť systému (sémantický systém) sa učí skomprimovať sekvenciu týchto propozícií do jednej, statickej reprezentácie významu celej vety. Model CSCP je postavený na báze modulov SRN ([34]), ktoré sa naučia zobraziť vstupnú vetu (sekvenciu slov) na jej celkový význam (komprehenzia) ako aj opačným spôsobom, t.j. význam na povrchovú formy vety (produkcia). Dôležité je, že CSCP je schopný simulovať rôzne psychologické aspekty spracovania viet u človeka a poukazuje na dôležitosť prepojenia medzi komprehenziou a produkciou počas učenia jazyka. Diskutabilnou stránkou modelu je však práve trénovanie pomocou explicitných propozičných reprezentácií, ktoré prebieha pomocou tzv. dotazovania (query) na jednotlivé konštituenty propozícií, čo je z kognitívneho hľadiska otázne (tento prvok figuruje aj v starších, jednoduchších modeloch komprehenzie viet, zrejme ako nutná pomôcka na to, aby sa model dokázal vstupno-výstupnú transformáciu naučiť [36].

Flexibilnejší model INSOMNet ([92]) nevyžaduje fixné rámce propozícií ako CSCP. Tento model s modulárnou architektúrou bol navrhnutý ako interpretátor anglických viet, ktorý predloženej vete v textovej forme (slovo za slovom) priradí jej sémantickú reprezentáciu. Reprezentácie viet s tzv. plytkou sémantikou, použité na trénovanie výstupného modulu s externým učiteľom, boli vzaté z lingvisticky anotovaného korpusu, v ktorom sémantika každej vety má formálne podobu orientovaného grafu. Takýto graf vyjadruje syntaktické a sémantické vzťahy medzi vetnými členmi, reprezentovanými distribuovane. Reprezentácie viet založené na tejto lingvistickej konceptuálnej štruktúre majú dve zaujímavé vlastnosti: (1) vďaka grafovej štruktúre sa vyhýbajú problému reprezentovania syntaktických stromov s veľkou hĺbkou (typických pre prirodzený jazyk v kontexte symbolových a štatistických modelov), a (2) podobne ako symbolové programy tiež využívajú smerníky, no tie sú implementované pomocou aktivácií viacerých neurónov a sieť sa ich musí naučiť ako súčasť reprezentácie vety. Model sa musí tiež naučiť gramatiku, pretože tá nie je explicitne daná. INSOMNet je psychologicky prijateľný model, lebo (podobne ako človek) sa učí na príkladoch, generuje predikcie, nemonotónne reviduje interpretáciu vety pri jej inkrementálnej analýze, správa sa robustne, dokáže paralelne vytvárať viacnásobné interpretácie v prípade nejednoznačných viet, a vie modelovať rôzne jazykové poruchy pozorované u ľudí.

V oboch spomínaných modeloch sa používajú sémantické reprezentácie viet vo formáte propozícií, čo znamená, že vieme lokalizovať jednotlivé konštituenty vety. Teórie porozumenia textu ([47]), podporené psychologickými experimentmi, potvrdzujú, že okrem propozičnej úrovne človek využíva aj tzv. situačnú úroveň reprezentácie, ktorá už je jazykovo nezávislá a formuje sa nad propozičnou úrovňou (textová báza). Tento poznatok slúžil ako inšpirácia pre model distribuovaného situačného priestoru (DSS,[53]), ktorý bol navrhnutý na reprezentáciu významov (situácií) na situačnej úrovni. Situačné vektory teda nemajú propozičný charakter, pretože neuróny v DSS sa podieľajú na reprezentácií viacerých situácií, a reprezentácia každej situácie je distribuovaná. DSS reprezentácie situácií sú pripravené vopred pomocou samoorganizácie. Úloha neurónovej siete (SRN) spočíva v tom, že jednotlivým vstupným vetám, opisujúcim nejakú situáciu v uvažovanom mikrosvete má SRN priradiť ich situačný vektor, čo sa interpretuje ako pochopenie významu vety. SRN teda predpovedá danú situáciu, preskakujúc propozičnú úroveň reprezentácie vety. SRN číta vetu slovo po slove, a ku koncu má vygenerovať situačný vektor. Samozrejmou požiadavkou je, aby sieť dokázala zovšeobecňovať, t.j. aby porozumela nielen novým vetám opisujúcim známe situácie, ale aj novým vetám opisujúcim nové situácie. Napríklad, v modeli existujú tri postavičky, ktoré môžu robiť rôzne činnosti na rôznych miestach (samostatne alebo spolu). Ak sa systém naučí porozumieť vetám Janko hrá šach vnútri a Janko sa hrá na schovávačku vonku, musí porozumieť aj vetám Janko hrá šach vonku a Janko sa hrá na schovávačku vonku (ak sú tieto situácie konzistentné s vlastnosťami mikrosveta). Táto schopnosť zovšeobecnenia vyjadruje vlastnosť tzv. **sémantickej systematickosti** ([50]). Situačné vektory v modeli DSS majú pekné vlastnosti, sú subsymbolové (forma vektora a jeho význam sú neoddeliteľné) a teda analógové, čím sú konzistentné s perceptuálnymi symbolmi ([6]). Tieto situačné vektory však nie sú modálne, a preto ani ukotvené. Okrem toho,

v situačných vektoroch neexistujú koncepty ani vzťahy medzi nimi (čo je typickou vlastnosťou propozičných vektorov). Model DSS od tohto abstrahuje, no ťažko si predstaviť mentálne reprezentácie situácií, kde nefigurujú koncepty.

5.8 Integrácia modalít vo vývinovej kognitívnej robotike

Ukazuje sa, že vo výpočtovom modelovaní v kognitívnej vede nastáva v súčasnosti oživenie a snaha o integrované ukotvené prístupy pri učení kognitívnych schopností s čo možno najväčšou mierou autonómnosti učiaceho sa agenta. Z dôvodu zložitosti realizácie týchto cieľov sa používajú zväčša jednoduchšie architektúry sietí a spôsoby učenia, nielen v snahe o vytváranie multimodálnych perceptuálnych reprezentácií, ale najmä v úlohách prepojenia jazykových schopností na ostatné kognitívne funkcie (prehľad modelov a výzvy pre neurorobotiku možno nájsť v [16]). Dôraz sa teda kladie na to, aby jazykový vývin šiel „ruka v ruke“ s kogníciou a nie izolovane, ako to bolo v modeloch uvedených v predchádzajúcich častiach o modelovaní akvizície jazyka (5.5-5.7). Typickým elementom týchto modelov je napríklad prepojenie motorickej aktivity s učením sa pohybových slovies, ktoré takto priamo nadobúdajú ukotvené významy (napr. anglické slovesá kick, pick, lick sa týkajú rozličných častí tela a vieme, že aj ich neurálne koreláty v mozgu sú odlišné, [116]). K tomu sa dá v najjednoduchšom prípade pristupovať pomocou jednej doprednej neurónovej siete, ktorá dostáva na vstup aktuálnu proprioceptívnu informáciu o svojom tele, napr. ramene, a snaží sa urobiť relevantnú akciu (natočenie ramena), pričom spracováva aj jazykový vstup, ktorý danú akciu špecifikuje ([89]). Takáto sieť sa dá trénovať aj pomocou bežného algoritmu spätného šírenia chyby. Zložitejším prístupom je prepojenie dvoch rekurentných sietí, senzomotorickej a jazykovej, ktoré sa súčasne trénujú na sekvenciách s využitím predikcií na požadovaných hodnotách v ďalšom kroku (uhly natočenia ramena robota, resp. ďalšie slovo vety) ([135]). Natrénovaná sieť potom dokáže napríklad vykonať požadovanú akciu na základe jej pomenovania, a to aj v prípade nových akcií – táto schopnosť zovšeobecnenia sa opiera o vlastnosť kompozičnosti (spomínanej v časti 5.6). Tieto prístupy však využívajú nerealistické predpoklady pri učení, ako napríklad existenciu požadovaných hodnôt pre pohyb ramena. Realistickejším prístupom je napríklad učenie s posilňovaním, kde spätná väzba z prostredia nie je taká explicitná (pozri časť 5.1) a teda agent musí prísť na to, aké akcie má vykonávať aby viedli k cieľu. Niektoré prístupy zase viac využívajú samoorganizáciu v senzorickej, motorickej i jazykovej doméne. Napríklad model Wermter a Willshaw ([150]) má pekné topografické vlastnosti (podobne ako senzorické i motorické časti mozgovej kôry) a môže slúžiť ako asociátor akcií a ich jazykového opisu, no chýbajú mu predikčné vlastnosti (spätná väzba). Možno povedať, že integračné prístupy s využitím neurónových sietí najmä v oblasti kognitívnej robotiky sú vo svojich začiatkoch, a až budúcnosť ukáže, ako sa podarí napredovať smerom k realistickejšej zložitosti (kognitívnych architektúr a prostredia).

6 Záver

V príspevku sme sa venovali výpočtovým prístupom v kognitívnej vede, so zameraním na konekcionizmus. Výpočtové modelovanie vo všeobecnosti možno považovať na nevyhnutný metodologický komponent, pretože ponúka explicitné mechanizmy, ktoré opisujú ako môže kognícia fungovať ([87]). Pri prechádzke paradigmami kognitívnej vedy sme sa snažili objasniť najmä rozdiely medzi symbolovou a konekcionistickou paradigmou, pričom sme spomenuli, že koncept počítania musíme chápať širšie než ponúka počítačová metafora mysle, pretože mozog nie je počítač a neurónové tiež počítajú, no nemusia pritom vykonávať programy. S využitím teoretickej argumentácie ako aj empirickej evidencie sme dospeli k záveru, že konekcionistické prístupy ponúkajú detailnejšie vysvetlenie kognitívnych mechanizmov, konzistentné s neurobiologickými princípmi na istej úrovni abstrakcie (ale značne nižšej než v prípade symbolových modelov). Zaujímavou vlastnosťou neurónových sietí je aj to, že tvoria akýsi most medzi symbolovou a dynamickou paradigmou (ktoré stoja v úplnom kontraste), pretože obsahujú niektoré charakteristiky oboch. Za hlavnú výhodu neurónových sietí považujeme ich paradigmy učenia s rôznym stupňom spätnej väzby z prostredia (učenie s učiteľom, s posilňovaním a so samoorganizáciou).

Využiteľnosť umelých neurónových sietí sme ilustrovali na príkladoch zameraných na vybrané procesy spojené s raným kognitívnym vývinom. Pohľadom zdola sme sa snažili pokryť nielen úlohy nižšej kognície (napr. senzomotorická koordinácia, rozpoznávanie objektov) ale aj vyššej (jazyk, otázka syntaktickej a sémantickej systematickosti). Konekcionistický prístup v kognitívnom modelovaní je založený na predpoklade, že ľudská myseľ/mozog nie je hybridný systém, na ktorý potrebujeme nutne použiť dva opisy ([39]), ale že z evolučného hľadiska ide o homogénny systém, čo sa týka mechanizmov a mentálnych reprezentácií. Clark ([23], s. 135) hovorí v tejto súvislosti o otázke tzv. kognitívnej inkrementálnosti: „Aký je vzťah medzi stratégiami používanými pri základných problémoch vnímania a konania, a tými, ktoré používame na riešenie abstraktnejších problémov na vyššej úrovni?“ Ohľadne tejto otázky však stále neexistuje v kognitívnej vede konsenzus, pretože myšlienka hybridnej kognitívnej architektúry je stále aktuálna, a je realizovaná aj v návrhoch hybridných systémov (napr. [72]).

Principiálnou motiváciou pre orientáciu na neurónové siete pri kognitívnom modelovaní je aj predpoklad, že svet je spojitý (t.j. fyzikálne charakteristiky vnímané senzormi sú spojité), akcie agenta sú spojité (uhly natočenia), a preto aj myseľ agenta/človeka by prirodzene mohla byť spojitá ([133]). Kľúčová výhoda spojitých priestorov spočíva v možnosti zovšeobecňovania, čo sa javí ako kognitívne relevantný faktor a požiadavka na učiace sa systémy.

Pochopiteľne, svet naokolo má aj svoje diskrétne prvky, napríklad počty objektov bývajú diskrétne, počet akcií vykonateľných agentom v danom kontexte býva diskrétny vzhľadom na ich efekt (napr. odbočiť vľavo alebo vpravo), a koniec koncov logické úlohy rozmanitého typu ktoré človek dokáže riešiť, bývajú formulované pomocou symbolov. Celé toto spektrum je nutné vedieť vysvetliť konekcionisticky, ak táto paradigma má ambíciu tvoriť komponenty ideálnej kognitívnej architektúry.

V diskrétnom svete symbolových systémov sú aplikovateľné kvalitatívne odlišné mechanizmy, no tie sú tiež užitočné, napr. pri návrhu expertných systémov alebo systémov s ontológiami, ktoré slúžia človeku. Voľba prístupu a úrovne abstrakcie teda závisí od cieľa a kritérií. Umelá inteligencia a výpočtová kognitívna veda majú veľa spoločného, no je medzi nimi jeden dôležitý rozdiel. Umelá inteligencia sa zaujíma o efektívne riešenia využiteľné v inteligentných systémoch, bez ohľadu na ich biologickú relevantnosť. V kognitívnej vede na biologickej prijateľnosti záleží.

Poďakovanie: Tento príspevok vznikol z podpory grantovej agentúry VEGA v rámci grantových úloh 1/0439/11 a 1/0602/10, KEGA K-09-016-00.

Literatúra

[1] Abrahamsen A., Bechtel W.: Phenomena and mechanisms: Putting the symbolic, connectionist, and dynamical systems debate in broader perspective. In R. Stainton (Ed.), *Contemporary Debates in Cognitive Science*. Oxford: Basil Blackwell, 2006.

[2] Aizawa K.: *The Systematicity Arguments*. Kluwer Academic, Dordrecht, 2003.

[3] Arbib M.: From monkey-like action recognition to human language: an evolutionary framework for neurolinguistics. *Behavioral and Brain Sciences,* 28, 105-168, 2005.

[4] Asada M. a spol.: Cognitive developmental robotics: a survey. *IEEE Transactions on Autonomous Mental Development* 1(1), 12-34, 2009.

[5] Bell A.: Levels and loops: The future of artificial intelligence and neuroscience. *Philosophical Transactions of the Royal Society London B,* 354, 2013-2020, 1999.

[6] Barsalou L.: Perceptual symbol systems. *Behavioral and Brain Sciences*, 22(4), 577-660.

[7] Barsalou L.: Grounded cognition. *Annual Reviews of Psychology,* 59, 617-645, 2008.

[8] Bechtel W., Abrahamsen A.: Connectionism and the Mind: Parallel Processing, Dynamics and Evolution in Networks. Malden, MA: Blackwell, 2002.

[9] Bengio Y., LeCun Y.: Scaling learning algorithms towards AI. V knihe Bottou L. et al. (eds.), *Large-Scale Kernel Machines,* MIT Press, 2007.

[10] Boroditsky L., Ramscar M.: The roles of body and mind in abstract thought. *Psychological Science*, 13, 185-88, 2002.

[11] Brakel P., Frank S.: Strong systematicity in sentence processing by simple recurrent networks. In: *Proceedings of the 31st Annual Conference of the Cognitive Science Society,* 1599-1604, 2009.

[12] Bressler S., Kelso J.S.: Cortical coordination dynamics and cognition. *Trends in Cognitive Sciences*, 5, 26-36, 2001.

[13] Brooks R.: A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1), 14-23, 1986.

[14] Burgess N.: Spatial memory: how egocentric and allocentric combine. *Trends in Cognitive Sciences,* 10(12), 551-557, 2006.

[15] Byrne P., Becker S., Burgess N.: Remembering the past and imagining the future: a neural model of spatial memory and imagery. *Psychological Review,* 114, 340-375, 2007.

[16] Cangelosi A. a spol.: Integration of action and language knowledge: a roadmap for developmental robotics, *IEEE Transcations on Autonomous Mental Development*, 2(3), 167-195, 2010.

[17] Cangelosi A., Tikhanoff V., Fontanari J., Hourdakis E.: Integrating language and cognition: A cognitive robotics approach. *IEEE Computational Intelligence Magazine*, 2(3), 65-7, 2007.

[18] Chang S., Papadimitriou C., Snyder L.: Using a compound gain field to compute a reach plan. *Neuron*, 64, 744–755, 2009.

[19] Christiansen M., Chater N.: Connectionist natural language processing: The state of the art. *Cognitive Science*, 23(4), 417-437, 1999.

[20] Christiansen M., Chater N. (eds.): *Connectionist psycholinguistics.* Ablex Publishing, 2001.

[21] Churchland P., Sejnowski, T.: *The Computational Brain*. Cambridge, MA: MIT Press, 1992.

[22] Ciresan D., Meier U., Gambardella L., Schmidhuber J.: Deep big simple neural nets for handwritten digit recognition. *Neural Computation,* 22(12), 3207-3220, 2010.

[23] Clark A: Mindware: An Introduction to the Philosophy of Cognitive Science. Oxford University Press, 2001.

[24] Cleeremans A., Destrebecqz A., Boyer M.: Implicit learning: News from the front. *Trends in Cognitive Sciences*, 2, 406-416, 1998.

[25] Cohen Y., Andersen R.: A common reference frame for movement plans in the posterior parietal cortex. *Nature Reviews Neuroscience,* 3, 553-562, 2002.

[26] Craenen B., Eiben A.: Computational intelligence. In: *Encyclopedia of Life Support Sciences*, EOLSS Publishers Co. Ltd., 2003.

[27] David N., Newen A., Vogeley K.: The „sense of agency“ and its underlying cognitive and neural mechanisms. *Consciousness and Cognition*, 17, 523-534, 2008.

[28] Davies J.: Semantic Web Technologies: Trends and Research in Ontology-based Systems. John Wiley & Sons, 2006.

[29] Desmurget M., Grafton S.: Forward modeling allows feedback control for fast reaching movements. *Trends in Cognitive Sciences,* 4(11): 423-431, 2000.

[30] Doya K.: What are the computations of the cerebellum, the basal ganglia, and the cerebral cortex? *Neural Networks,* 12, 961-974, 1999.

[31] Doya K.: Reinforcement learning in continuous time and space. *Neural Computation*, 12(1), 219-245, 2000.

[32] Duch W.: Towards comprehensive foundations of computational intelligence. *Lecture Notes in Computer Science*, Springer, Vol. 63, 261-316, 2007.

[33] Edelman G.: Bright Air, Brilliant Fire: On the Matter of the Mind. New York: Basic Books, 1992.

[34] Elman J.: Finding structure in time. *Cognitive Science*, 14(2), 179-211, 1990.

[35] Farkaš I.: Samoorganizujúce sa mapy. Kapitola v knihe V. Kvasnička a kol., *Úvod do teórie neurónových sietí*, IRIS, Bratislava, 142-189, 1997.

[36] Farkaš I.: Konekcionistické modelovanie jazyka. V knihe: Rybár J., Kvasnička V., Farkaš I. (zost.), *Jazyk a kognícia*, Kalligram, Bratislava, 262-305, 2005.

[37] Farkaš I.: Samoorganizácia ako hybná sila dynamických vzorcov aktivít v mozgu a mysli. V zborníku *Kognice a umělý život VI,* Opava: Slezská univerzita, 143-148, 2006.

[38] Farkaš I.: Hľadanie kauzálnych vzťahov v probléme mysle a tela z pohľadu neredukcionistického fyzikalizmu. V knihe: Kvasnička V. a spol. (zost.), *Myseľ, inteligencia a život*. Vydavateľstvo STU v Bratislave, 3-16, 2007.

[39] Farkaš I.: Konceptuálne východiská pre model stelesnenej mysle. V knihe: Kvasnička V., Kelemen J., Pospíchal J. (zost.), *Modely mysle*, Vydavateľstvo Európa, Bratislava, 35-64, 2008.

[40] Farkaš I.: Learning nonadjacent dependencies with a recurrent neural network. In: Koeppen M. et al. (zost.), *Int. Conference on Neural Information Processing Systems (ICONIP'2008)*, Springer, Lecture Notes in Computer Science, 292-299, 2009.

[41] Farkaš I.: Neurálne modely v kognitívnej robotike: vizuálno-motorická interakcia. V zborníku Kelemen J., Kvasnička V.: *Kognice a umelý život X,* Ostravice, ČR, 93-99, 2010.

[42] Farkaš I., Crocker M.: Syntactic systematicity in sentence processing with a recurrent self-organizing network. *Neurocomputing*, 71, 1172-1179, 2008.

[43] Farkaš I., Malý M., Rebrová K.: Mirror neurons: theoretical and computational issues. *Technická správa TR-2011-28*. FMFI Univerzita Komenského, 2011.

[44] Farkaš I., Rebrová K.: Meaning and its representation from the perspective of grounded cognition. *Kognice,* Praha, 2010.

[45] Feldman J., Ballard D.: Connectionist models and their properties. *Cognitive Science*, 6, 205–254, 1982.

[46] Felleman D., van Essen D.: Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex*, 1(1), 1-47, 1991.

[47] Fletcher C., Chrysler S.: Surface forms, text bases, and situation models: recognition memory for three types of textual information. *Discourse Processes*, 13, 175-190, 1990.

[48] Fodor J.: *The Modularity of Mind*. Cambridge, MA: MIT Press, 1983.

[49] Fodor J.: The Mind Doesn't Work That Way. MIT Press, 2000.

[50] Fodor J., Pylyshyn Z.: Connectionism and cognitive architecture. *Cognition,* 28, 3-71, 1988.

[51] Frank S.: Learn more by training less: Systematicity in sentence processing by recurrent networks. *Connection Science*, 18, 287-302, 2006a.

[52] Frank S.: Strong systematicity in sentence processing by an echo-state network. In: *Proceedings of ICANN*, Part I, Lecture Notes in Computer Science, vol. 4131, Springer, 2006b.

[53] Frank, S., Haselager W., Van Rooij I.: Connectionist semantic systematicity. *Cognition*, 110, 358-379, 2009.

[54] Freeman W.: *How Brains Make Up Their Minds*. New York: Columbia University Press, 2001.

[55] Frischen A., Loach D., Tipper S.: Seeing the world through another person's eyes: simulating selective attention via action observation. *Cognition,* 111(2), 212-8, 2009.

[56] Gallese V., Keysers C. and Rizzolatti G.: A unifying view of the basis of social cognition. *Trends in Cognitive Sciences*, 8, 396-403, 2004.

[57] Gardner H.: The Minds New Science: A History of the Cognitive Revolution. New York: Basic Books, 1987.

[58] Gärdenfors P.: Cued and detached representations in animal cognition. *Behavioural Processes*, 36, 263-273, 1996.

[59] Gentner D.: Psychology in cognitive science: 1978-2038. *Topics in Cognitive Science*, 2, 328-344, 2010.

[60] Gibbs R.: *Embodiment and Cognitive Science*. New York: Cambridge University Press, 2006.

[61] Gibson J.: *Ecological Approach to Visual Perception.* Boston: Houghton-Mifflin, 1970.

[62] Greenfield J.: Language, tools and brain: The ontogeny and phylogeny of hierarchically organized sequential behavior. *Behavioral and Brain Sciences,* 14, 531-595, 1991.

[63] Griffiths T., Kemp C., Tenenbaum J.: Bayesian models of cognition. In R. Sun (Ed.), *Cambridge Handbook of Computational Psychology,* New York: Cambridge University Press, 59-100, 2008.

[64] Hadley R.: Systematicity in connectionist language learning. *Mind and Language*, 9(3), 247-272, 1994.

[65] Harnad S.: The symbol grounding problem. *Physica D*, 42, 335-346, 1990.

[66] Havel, I.M.: Robotika – úvod do teorie kognitivních robotů. SNTL, Praha, 1980.

[67] Haykin S.: Neural Networks and Learning Machines, Prentice Hall, 2008.

[68] Hinton G., Osindero S., Teh Y.: A fast learning algorithm for deep belief nets. *Neural Computation*, 18, 1527-1554, 2006.

[69] Hornik K., Stinchcombe M., White H.: Multilayer feedforward networks are universal approximations. *Neural Networks*, 2, 359-366, 1989.

[70] Hutchins E.: *Cognition in the Wild*. MIT Press, 1996.

[71] Izhikevich E.: Which model to use for cortical spiking neurons? *IEEE Transactions on Neural Networks*, 15, 1063-1070, 2004.

[72] Jilk D., Lebiere C., O'Reilly R., Anderson J.: SAL: An explicitly pluralistic cognitive architecture. *Journal of Experimental and Theoretical Artificial Intelligence*, 20, 197-218, 2008.

[73] Kaczmarek L., Levitan I. (zost.): *Neuromodulation: The Biochemical Control of Neuronal Excitability.* NY: Oxford University Press, 1987.

[74] Kawato M.: Internal models for motor control and trajectory planning. *Current Opinion in Neurobiology,* 9(6): 718-727, 1999.

[75] Kelemen J., Liday M.: *Expertné systémy pre prax*. Bratislava, Sofa, 1996.

[76] Kelso J.: Dynamic Patterns: The Self-Organization of Brain and Behavior. The MIT Press, Cambridge, MA, 1995.

[77] Kohonen T.: *Self-Organizing Maps*. Springer, (3rd, extended ed.), 2001.

[78] Korenčiak R.: Aproximácia motorického priestoru ramena simulovaného robota. *Diplomová práca*, FMFI Univerzity Komenského, 2010.

[79] Kosslyn S.: Image and Brain: The Resolution of the Imagery Debate. Cambridge, MA: MIT Press, 1994.

[80] Lakoff G., Johnson M.: Philosophy in the Flesh: The Embodied Mind and It's Challenge to Western Thought. New York: Basic Books, 1999.

[81] Landauer T., Foltz P., Laham D.: Introduction to latent semantic analysis. *Discourse Processes*, 25, 259-284, 1998.

[82] Lave J.: Cognition in Practice: Mind, Mathematics and Culture in Everyday Life. New York: Cambridge University Press, 1988.

[83] LeCun Y. a spol.: Backpropagation applied to handwritten zip code recognition, *Neural Computation*, 1(4), 541-551, 1989.

[84] Li P., Farkaš I., MacWhinney B.: Early lexical development in a self-organizing neural network. *Neural Networks,* 17(8-9), 1345-1362, 2004.

[85] Lukosevicius M., Jaeger H.: Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3), 127-149, 2009.

[86] Maass W., Bishop C. (eds.): *Pulsed Neural Networks*. MIT Press, 1999.

[87] McClelland J.: The place of modeling in cognitive science. *Topics in Cognitive Science*, 1(1), 11-38, 2009.

[88] Majid A., Bowerman M., Kita S., Haun D., Levinson, S.: Can language restructure cognition? The case for space. *Trends in Cognitive Science*, 8, 108-114, 2004.

[89] Marocco D., Cangelosi A., Fischer K., Belpaeme T.: Grounding action words in the sensorimotor interaction with the world: experiments with a simulated iCub humanoid robot. *Frontiers in Neurorobotics*, 4(1), 1-15, 2010.

[90] Marr D.: *Vision*. San Francisco, CA: W.H. Freeman, 1982.

[91] Marsland S.: *Machine Learning: An Algorithmic Introduction*. CRC Press, New Jersey, USA, 2009.

[92] Mayberry M.: Incremental nonmonotonic parsing through semantic self-organization. *Doctoral thesis*. Department of Computer Sciences, University of Texas at Austin, 2004.

[93] McClelland J.: The place of modeling in cognitive science. *Topics in Cognitive Science,* 1(1)*,* 11-38, 2009.

[94] McClelland J., Kawamoto A.: Mechanisms of sentence processing: Assigning roles to constituents of sentences. In *Parallel Distributed Processing,* Vol. 2. MIT Press, Cambridge, MA, 318-362, 1986.

[95] McCulloch W., Pitts W.: A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 7, 115-133, 1943.

[96] McLeod P., Plunkett K., Rolls, E.: *Introduction to Connectionist Modelling of Cognitive Processes.* Oxford, UK: Oxford University Press, 1998.

[97] Miall R.: Connecting mirror neurons and forward models. *Neuroreport,* 14(17), 2135-7, 2003.

[98] Miikkulainen R.: Dyslexic and category-specific aphasic impairments in a self-organizing feature map model of the lexicon. *Brain and Language,* 59(2), 334-366, 1997.

[99] Mukamel R., Ekstrom A., Kaplan J., Iacoboni M., Fried I.: Single-neuron responses in humans during execution and observation of actions. *Current Biology,* 20(8), 750-756, 2010.

[100] Newell A., Simon H.: Computer science as empirical enquiry. *Communications of the ACM*, 19 (1976) 113-126. In: Mind Design II, J. Haugeland (ed.), 81-110. Cambridge, MA: MIT Press, 1997.

[101] Newport E., Aslin R.: Learning at a distance I. Statistical learning of non-adjacent dependencies. *Cognitive Psychology,* 48, 127-162, 2004.

[102] O'Brien G., Opie J.: How do connectionist networks compute? *Cognitive Processing*, 7, 30-41, 2006.

[103] O'Reilly R., Munakata Y.: Computational Explorations in Cognitive Neuroscience: Understanding the Mind by Simulating the Brain. MIT Press, 2000.

[104] Pfeifer R., Lungarella M., Iida F.: Self-organization, embodiment, and biologically inspired robotics. *Science,* 318(5853), 1088-93, 2007.

[105] Pfeifer R., Scheier C.: *Understanding Intelligence*. MIT Press, Cambridge, MA, 1999.

[106] Piccinini G.: Some neural networks compute, others don't. *Neural Networks*, 21(2-3), 311-321, 2008.

[107] Pinker S.: *How the Mind Works.* W.W. Norton and co., 2009.

[108] Pinker S., Mehler J. (zost.): *Connections and Symbols.* Cambridge, MA: MIT Press, 1988.

[109] Pinker S., Prince A.: On language and connectionism: Analysis of a parallel distributed processing model of language acquisition. *Cognition, 28*, 73-193, 1988.

[110] Pinker S., Ullman M.: The past tense debate. *Trends in Cognitive Sciences,* 6(11), 456-474, 2002.

[111] Pitt D.: Mental representation, *The Stanford Encyclopedia of Philosophy,* http://plato.stanford.edu/archives/fall2008/entries/mental-representation, 2008.

[112] Plunkett K., Juola P.: A connectionist model of English past tense and plural morphology. *Cognitive Science, 23*, 463-490, 1999.

[113] Port R., van Gelder T. (Eds.): *Mind as Motion: Explorations in the Dynamics of Cognition.* Cambridge, MA: MIT Press, 1995.

[114] Pouget A., Snyder L.: Computational approaches to sensorimotor transformations. *Nature Neuroscience*, 3, (suppl.), 1192-1198, 2000.

[115] Pstružina K.: Abstrakce, komparace a generalizace jako základní operace lidského myšlení. V knihe: *Myseľ, inteligencia a život*. Vydavateľstvo STU v Bratislave, 111-117, 2007.

[116] Pulvermüller F.: Brain mechanisms linking language and action. *Nature Reviews Neuroscience,* 6(7), 576-582, 2005.

[117] Pylyshyn Z.: *Computation and Cognition.* MIT Press, 1984.

[118] Quartz S., Sejnowski T.: The neural basis of cognitive development. *Behavioral and Brain Sciences*, 20, 537-596, 1997.

[119] Quinlan P.: Connectionism and Psychology: A Psychological Perspective on New Connectionist Research. Chicago, IL: University of Chicago Press, 1991.

[120] Ritter H., Kohonen T.: Self-organizing semantic maps. *Biological Cybernetics, 61*, 241-254, 1989.

[121] Rizzolatti G., Fadiga L., Fogassi L., Gallese V.: Premotor cortex and the recognition of motor actions. *Cognitive Brain Research*, 3, 131-41, 1996.

[122] Rohde D.: A connectionist model of sentence comprehension and production. *Doctoral thesis.* Carnegie Mellon University, Pittsburg, 2002.

[123] Rohde D., Plaut D.: Connectionist models of language processing. *Cognitive Studies*, 10, 10-28, 2003.

[124] Rosch E.: Principles of categorization. In: Rosch E., Lloyd B. (zost.), *Cognition and Categorization*. Lawrence Erlbaum Associates, Hillsdale, NJ, 27-48, 1978.

[125] Rosenblatt, F.: Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. Washington, DC: Spartan, 1962.

[126] Roy D.: Grounding words in perception and action: computational insights. *Trends in Cognitive Science*, 9(8), 389-96, 2005.

[127] Rumelhart D., McClelland J. and the PDP research group: *Parallel Distributed Processing: Exploration in the Microstructure of Cognition*, Vol. 1. Cambridge, MA: MIT Press, 1986.

[128] Saffran J., Newport E., Aslin R.: Word segmentation: the role of distributional cues. *Journal of Memory and Language,* 35, 606-621, 1996.

[129] Salinas E., Sejnowski, T.: Gain modulation in the central nervous system: Where behavior, neurophysiology and computation meet. *The Neuroscientist*, 7(5), 430-440, 2001.

[130] Searle J.: Minds, Brains, and Programs. *Behavioral and Brain Sciences*, 3, 417-458, 1980.

[131] Siegelmann H., Sontag E.: Turing computability with neural nets. *Applied Mathematics Letters*, 4, 77-80, 1991.

[132] Smith L.: Cognition as a dynamic system: principles from embodiment. *Developmental Review,* 25, 278-98, 2005.

[133] Spivey M.: *The Continuity of Mind*. Oxford: Oxford University Press, 2007.

[134] Steels L.: The symbol grounding problem has been solved, so what's next? In: de Vega M., Glenberg A. & Graesser A. (eds.), *Symbols and Embodiment: Debates on Meaning and Cognition,* Oxford University Press, 223-244, 2008.

[135] Sugita Y., Tani J.: Learning semantic combinatoriality from the interaction between linguistic and behavioral processes, *Adaptive Behavior*, 13(1), 33-52, 2005.

[136] Šefránek J.: *Inteligencia ako výpočet.* IRIS Bratislava 2002.

[137] Šefránek J., Takáč M., Farkaš I.: Vznik inteligencie v umelých systémoch. V knihe Magdolen, D. (zost.): *Hmota, život, inteligencia: Vznik.* VEDA, Bratislava, 245-270, 2008.

[138] Šíma J., Orponen P.: General-purpose computation with neural networks: A survey of complexity theoretic results. *Neural Computation,* 15, 2727-2778, 2003.

[139] Tadeo M., Floridi L.: The symbol grounding problem: a critical review of fifteen years of research. *Journal of Experimental and Theoretical Artificial Intelligence*, 419-445, 2005.

[140] Takáč M.: Construction of Meanings in Living and Artificial Agents. *Dizertačná práca*, FMFI UK v Bratislave, 2007.

[141] Takáč M.: Autonomous construction of ecologically and socially relevant semantics. *Cognitive Systems Research*, 9, 293-311, 2008.

[142] Thelen E., Smith L.: A Dynamic Systems Approach to the Development of Cognition and Action. Cambridge, MA: MIT Press, 1994.

[143] Turing A.: Computing machinery and intelligence. *Mind*, 59, 433-460, 1950.

[144] Ungerleider L., Mishkin M.: Two cortical visual systems. V knihe D. Ingle, M. Goodale, & R. Mansfield (Eds.), *Analysis of visual behavior,* Cambridge, MA: MIT Press, 549-586, 1982.

[145] Van Gelder T., Port R.: It's about time: An overview of dynamical approaches to cognition. In: *Mind as Motion: Explorations in the Dynamics of Cognition.* Cambridge, MA: MIT Press, 1-44, 1995.

[146] Van Hasselt H., Wiering M.: Reinforcement learning in continuous action spaces. *IEEE Int. Symposium on Approximate Dynamic Programming and Reinforcement Learning,* 272-279, 2007.

[147] Varela F., Thompson E., Rosch, E.: *The Embodied Mind: Cognitive Science and Human Experience.* Cambridge, MA: MIT Press, 1991.

[148] Vavrečka M.: Ukotvení symbolů v kontextu nulového sémantického záväzku. V zborníku *Kognícia a umelý život VI*, Opava: Slezská univerzita, 401-411, 2006.

[149] Vavrečka M.: Kognitivní sémantika a její aplikace v modelu reprezentace prostorových vztahů. *Dizertačná práca*, Masarykova Universita v Brne, 2008.

[150] Wermter S., Elshaw M.: Learning robot actions based on self-organising language memory. *Neural Networks*, 16, 691-699, 2003.

[151] Wilson M.: Six views of embodied cognition. *Psychonomics Bulletin Review,* 9(4), 625-636, 2002.

[152] Ziemke T.: Rethinking grounding. In: Riegler A. et al. (eds.), *Understanding Representation in the Cognitive Sciences*, New York: Plenum Press, 177-190, 1999.

[153] Zipser D., Andersen R.: A back-propagation programmed network that simulates response properties of a subset of posterior parietal neurons. *Nature,* 331, 679-684, 1988

Na ceste k hľadaniu vedy o mysli

Silvia GÁLIKOVÁ[1]

Abstrakt. Povaha mysle človeka predstavuje v súčasnej vede jeden z najkomplikovanejších problémov. Cieľom kapitoly je poukázať na kľúčové etapy a otázky späté so skúmaním stavov mysle v 20. storočí. Stavy normálnej a poškodenej mysle sa stali predmetom výskumu rozmanitých vedných disciplín, kognitívnej psychológie, lingvistiky, filozofie mysle, antropológie, neurovied, umelej inteligencie atď. Uplatňovanie najnovších zobrazovacích techník aktivity mozgu pri objasňovaní vnútornej skúsenosti nastoľuje nové otázky súvisiace s povahou „psychických chorôb", osobnou identitou, morálnou zodpovednosťou, „monitorovaním a manipuláciou" mysle človeka.

1 Úvod

V nasledujúcom texte stručne naznačím hlavné míľniky skúmania povahy ľudskej mysle. Zameriam sa predovšetkým na poznatky získané z experimentálneho a teoretického výskumu od druhej polovice 20. storočia po súčasnosť. Z bohatej histórie náhľadov na povahu mysle[2] by som rada pripomenula dva významné momenty. Prvým je úzka previazanosť úvah o mysli so znalosťami o tele (mozgu), s náhľadmi na zdravie a chorobu človeka. V rámci tejto línie zohral historicky rozhodujúcu úlohu starogrécky lekár Hippokrates (5.st.p.n.l.) a tradícia jeho nasledovníkov. Druhý prelom predznamenal Vesalius svojím dielom De humani corpus (1543), ktoré po stáročiach odsunulo autoritu galenovskej, do mystiky zahalenej, medicíny. K rozmachu naturalistického prístupu ku skúmaniu mysle významne prispeli objavy v astronómii, chémii, medicíne ako aj mechanistická filozofia a fyziológia 17. storočia (Descartes, Hobbes), britská empirická filozofia (Locke, Hume) a francúzske osvietenstvo 18. storočia (La Mettrie, Helvétius). Zásadný, doslova revolučný zvrat v názoroch na vznik a vývin mysle človeka priniesla na konci 19. storočia Darwinova evolučná teória [12].

V 20. storočí možno k hlavným míľnikom skúmania mysle zaradiť tieto: 1) rozkvet kognitívnej vedy, 2) experimentálny a klinický výskum normálnej a

[1] Filozofický ústav Slovenskej Akadémie vied, Klemensova 19, 813 64 Bratislava, e-mail: silvia@libris.sk

[2] Pojem mysle používam v texte synonymicky s pojmami duša, vnútorný svet, psychika, mentálne stavy

poškodenej mysle, 3) zrod počítačovej vedy a umelej inteligencie. Teoretický predpoklad o prirodzenej povahe psychických stavov priniesol radikálne zmeny vo vzťahu k objasneniu mechanizmov mysle a ich vzťahu k správaniu človeka. Zároveň sa nastolila potreba prehodnotiť tradičný model človeka ako a) slobodného, t.j. vedomého racionálneho agensa a b) stojaceho na vrchole hierarchie spomedzi živých a neživých systémov. Prívlastok revolučná si kognitívna veda vyslúžila predovšetkým za smelý pokus o vedecké uchopenie toho, čo z bežnej skúsenosti poznáme ako myšlienky, predstavy, pocity či pamäť.

1.1 Večnosť filozofických problémov

Kto je nositeľom mysle? Prežije duša zánik fyzického tela? Kde sídli duša? Aký je vzťah rozumu a emócií? Môžu stroje (počítače) myslieť? Aké sú to psychofyzikálne zákony? Ako možno charakterizovať črty vedomej skúsenosti? Na základe akých princípov sa organizuje vnímanie? Máme slobodnú vôľu? Je duša metaforou? Už na prvý pohľad badať, že nejde o úplne nové, ale skôr staré či znovuživené otázky. Uchopiť a vysvetliť povahu ľudskej mysle patrí k najstarším a najzávažnejším problémom. Vnútorný svet našich prianí, myšlienok a pocitov bol po stáročia zahalený rúškom tajomnosti. Jeden z dôvodov spočíval v domnelej neuchopiteľnosti a neobjektivizovateľnosti privátneho sveta človeka. Tento prístup nadväzoval na tradíciu spájania psychických stavov s nadprirodzeným, transcendentným, teda s „čímsi“, čo v zásadnom zmysle prekračuje možnosti a ciele vedeckej explanácie. Nemenej významný dôvod, pre ktorý myseľ odolávala experimentálnemu skúmaniu bola rozmanitosť a komplikovaná štruktúrovanosť mysle. Mentálne stavy sa rozčlenili na a) zmyslové vnemy (sensations) a b) propozičné postoje (propositional attitudes). K prvého typu sa radia pocity bolesti, čuchové, sluchové, zrakové vnemy, emócie, fantázia atď. Druhý typ mentálnych stavov, napríklad želania, presvedčenia, viery sa spájajú so schopnosťou vysloviť obsah myšlienok v jazyku. Na rozdiel od prežívaného fenomenálneho obsahu zmyslových vnemov, v prípade propozičných postojov je rozhodujúci sémantický obsah výrazov. Osobitý problém reflektovaný prevažne v anglo-americkej filozofii mysle predstavoval ontologický status mentálnych stavov, mentálnych procesov a mentálnych udalostí [39].

Riešenie ***večného problému*** povahy mysle sa teda v skutočnosti týkalo viacerých vzájomne prepletených problémov, ku ktorým sa zaradili napríklad tieto: vzťah myseľ (duša) – telo, Ja, identita osoby, mentálny obsah, sloboda vôle, zodpovednosť, intencionalita, subjektivita, mentálna kauzalita atď. Úsilie mysliteľov a vedcov sprístupniť ľudskú myseľ „zvonka“ brzdila absencia primeraných metód a teoretických prístupov. Ďalšiu ťažkosť spôsoboval „úkaz“ pomerne frekventovaný v dejinách myslenia a vedy, a to paralelná existencia viacerých protichodných tendencií. V období rozkvetu renesančnej vedy, to boli napríklad hrôzostrašné „hony na čarodejnice“. Náboženský fanatizmus, strach z nového poznania, narušania dominujúcej autority a ideológie vyústil do krutého násilia páchaného na nevinných (duševne chorých, odpadlíkoch, kacíroch atď.).

V súčasnosti možno zjednodušene hovoriť o ***asymetrii*** medzi pribúdajúcimi poznatkami vedných disciplín o fungovaní ľudskej mysle a teoretickým

zhodnocovaním či prijímaním výsledkov vedeckého bádania. Stredobodom diskusií, ktoré vstupujú aj do povedomia širšej verejnosti sú napríklad diagnostické a terapeutické úspechy „západnej“ medicíny na jednej strane a šírenie nepodložených (principiálne netestovateľných) alternatívnych spôsobov liečby na druhej strane. Z teoretického hľadiska však, myslím, väčšmi zaráža skutočnosť, že so znevažovaním a nepochopením významu vedeckého skúmania sa stretávame aj vo filozofii. Svedectvom uvedeného sú útoky na domnelý „novopozitivizmus“ alebo „scientizmus“ súčasnej vedy prejavujúce sa aj v rehabilitácii dualistických filozofických prístupov.

1.2 Súboj „izmov“

Teoretizovanie o povahe stavov ľudskej mysle v 20. storočí významne ovplyvnili myšlienky zakladateľov analytickej filozofie. Ludwig Wittgenstein [47] zasadil úder ambíciám filozofie napodobňovať metódy vedy a odhalil úskalia opantanosti filozofov vlastnou metafyzikou a jazykom. V podobných intenciách Gilbert Ryle [39] ostro odsúdil Descartov názor na vzťah duše a tela a uštedril mu prívlastok „dogma ducha v stroji“. Dualistické teórie o vzťahu mysle a tela, či už v podobe descartovského interakcionistického dualizmu alebo spinozovského paralelizmu, síce nastolili, ale, podľa väčšiny filozofovi ponechali psychofyzický problém otvorený. Reakciou na problémy súvisiace s „tvrdým orieškom“ produalistických koncepcií – problémom mentálnej kauzality sa pokúšali vysporiadať fyzikalistické teórie. Zástancovia fyzikalizmu alebo materializmu, „zaškatuľkovaní“ aj ako „naturalisti“ pristupujú k stavom mysle ako k prirodzeným, objektivizovateľným a redukcii podliehajúcim javom. K jedným z prvých, rozšírených prístupov tejto línie patril v 20. storočí behaviorizmus, ktorý sa podujal vysvetliť povahu mysle na základe pozorovateľného správania. Vedecký behaviorizmus, známy z dejín psychológie, vznikol ako reakcia na problémy vyplývajúce z introspekcionistickej metodológie [45]. Pre pripísanie stavov mysle organizmu sa stali nevyhnutné (a postačujúce) výstupy, t.j. správanie systému navonok. Niektorí filozofi interpretovali behavioristický program ako ignoráciu až odmietnutie existencie mentálnych stavov. Zaplniť „biele miesto“ v behavioristickej schéme medzi vstupmi a výstupmi systému sa pokúsili predstavitelia oboch variant teórií identity, typovej identity (type-identity) a inštančnej identity (token-identity) [13]. V prvom prípade ide o identifikáciu mentálneho typu na fyzikálny mozgu vo všeobecnosti. Inštančná teória identity sa týka identity konkrétnych mentálnych stavov indivídua v čase ***t*** s konkrétnymi fyzikálnymi stavmi indivídua v čase ***t***. Identifikáciu mysle s mozgom pokladali niektorí filozofi za prejav šovinizmu, nakoľko viedla k neoprávnenému upieraniu mysle takým systémom bez mozgu ako sú, napríklad počítače, baktérie alebo Marťania. K jednému z najznámejších protifyzikalistických argumentov sa radí argument ***viacúrovňovej realizácie*** (multiple-realizability argument) Hillary Putnama [34]. Jadro argumentu tvorila téza o existencii abstraktnej kauzálnej organizácii systému, štruktúru ktorej možno prenášať na systémy ľubovoľného fyzikálneho zloženia. Mentálne stavy sa charakterizujú ako ***funkcie*** systému, ktoré možno z príslušnej organizácie vyabstrahovať. Pre pripísanie mysle systému sa stala konštitutívna funkcia, podobne

ako v prípade peňazí. Funkciu platidla plnia rovnako zlaté, strieborné mince, papierové peniaze alebo kreditné karty.

Nespokojnosť filozofov s naturalizmom vo filozofii mysle, so stavom vedeckého skúmania mysle sa v posledných desaťročiach prejavila v rehabilitácii dualizmu. Spomedzi rôznych podôb neodualizmu získal popularitu ***dualizmus vlastností*** nazývaný aj teória dvoch aspektov. Prívrženci dualizmu vlastností na rozdiel od interakcionistického (substanciálneho) dualizmu odmietli ideu existencie dvoch odlišných „entít", mysle a tela. Substanciálnu odlišnosť medzi mysľou a telom nahradili rozlíšením vlastností mysle a tela. Myseľ, na rozdiel od tela, nemožno vyčerpávajúco vysvetliť na základe fyzikálnych princípov. Reduktívnej explanácii sa, podľa tohto prístupu, myseľ vymyká vzhľadom na špecifickú povahu ***kválií***, t.j. subjektívnych aspektov skúsenosti, toho, ***aké je to pre subjekt*** vnímať oblohu alebo počúvať symfóniu. Subjektívna a jedinečná kvalita mentálnych stavov ide ruka v ruke s jej neuchopiteľnosťou a privátnosťou. Kváliá reprezentujú neredukovateľný aspekt mysle presahujúci fyzikalistickú, behavioristickú alebo funkcionalistickú explanáciu. Neodualistickými črtami sa vyznačuje Searlova koncepcia spájania „mentalizmu s fyzikalizmom" [41], Nagelova dvojaspektová teória [31] a Chalmersov „neredukcionistický funkcionalizmus" [20]. Na jedinečnosť kvalitatívnych aspektov mysle sa odvolávajú aj zástancovia emergencie a superveniencie. Oba prístupy spájajú vznik mysle s istým stupňom komplexity vo vývine materiálnych systémov [44].

Faktom zostáva, že ani zo širokej palety „izmov" sa filozofom nepodarilo vymedziť spoločnú pracovnú charakteristiku mysle. Možno aj v dôsledku permanentného „lavírovania" medzi širokými a úzkymi charakteristikami stavov mysle. Podľa ***širokého*** vymedzenia možno myseľ pripísať každému objektu nášho sveta a vesmíru (panpsychizmus). Úzke vymedzenie identifikuje myseľ so schopnosťou používať artikulovanú reč, uvedomovať si psychické stavy, vedieť aké je to prežívať vlastnú skúsenosť (nereduktívny funkcionalizmus). Je zrejmé, že v súčasnosti žiaden z uvedených prístupov neobstojí. Nezohľadňujú totiž tak významné skutočnosti ako je vývinová povaha stavov mysle a vedomia, existencia psychiky u predverbálnych detí, pacientov s poruchami reči a myslenia alebo nevedomá povaha vnútorných dejov a determinantov konania človeka.

2 Od démonov k fontánam a humanoidným robotom

2.1 Zrod kognitívnej vedy

Historické korene vzniku kognitívnej vedy siahajú približne do prvej polovice 20. storočia. Napriek tomu, že kognitívna veda sa etablovala postupne a vo svojom vývine prechádzala viacerými etapami, W. Bechtel a G. Graham [2] uvádzajú rok 1956 ako možný rok jej zrodu. Na stretnutí, ktoré sa v tomto roku konalo v Massachusettskom technologickom inštitúte (MIT) predniesol N. Chomsky príspevok k teórii jazyka, G. Miller o úlohe magického čísla sedem v krátkodobej pamäti a A. Newell spolu s H. Simonom príspevok o vlastnom počítačovom modele

s názvom ***General Problem Solver***[3]. Kľúčovou sa stala otázka konštrukcie inteligentných strojov vychádzajúca z tradície kybernetiky, umelých neurónových sietí a symbolickej umelej inteligencie (AI) [36]. Pre kybernetiku zohral ústrednú rolu pojem ***feedbacku***, t.j. schopnosti systému vložiť vygenerovanú informáciu späť do systému, čím sa posilnilo jeho správanie v meniacom sa prostredí. Myšlienka, ktorá sa preverila už počas II. svetovej vojny spočívala v možnosti mechanického a neskôr elektronického uplatnenia súboru inštrukcií. Novovznikajúca veda sa okrem kybernetiky inšpirovala poznatkami viacerých vedných disciplín, neurovedy, lingvistiky, antropológie, filozofie mysle počítačovej vedy atď. K otázkam, ktoré si kládli teoretici sa zaradili tieto: Ako získava systém (organizmus) informácie? Ako ich spracúva a vyhodnocuje? Ako si pamätáme a riešime problémy? Akú rolu zohráva pozornosť pri vnímaní objektov? Nakoľko sme intencionálne racionálne subjekty schopné vôľovej kontroly? Zodpovedať nastolené sa priekopníci kognitívnovedného skúmania podujali uplatnením nových metód a prístupov. K základným metódam kognitívnej psychológie sa zaradili napríklad, laboratórne (kontrolované) experimenty, psychobiologický výskum, sebapozorovanie, kazuistiky, počítačová simulácia, analýza protokolov atď. [43]. V kognitívnej neurovede sa neurónová aktivita skúmala metódou ***lézií***, hľadaním súvislostí medzi poškodeným mozgom, poruchou kognitívnych stavov a poruchou správania. Dramatickú zmenu vo vzťahu ku skúmaniu štruktúry a funkcie mozgu vyvolalo uplatňovanie nových zobrazovacích technológií [43], [24].

Ako som uviedla, spoločným cieľom ***interdisciplinárneho*** skúmania sa stalo objasnenie ***kognície*** (z lat. ***cognitio***, poznanie), t.j. povahy a fungovania poznávania, vnímania, pamäte, myslenia. Terminologické a teoretické problémy s vymedzením pojmu kognícia sa premietli aj do charakteristiky predmetu kognitívnej vedy. V skratke možno hovoriť o širokej a úzkej definícii. V širokom chápaní sa kognitívna veda vymedzuje ako: „Interdisciplinárna aktivita, ktorá sa zaoberá myslením a inteligenciou a zahŕňa filozofiu, psychológiu, umelú inteligenciu, neurovedu, lingvistiku a antropológiu“ ([15], s. 11). Úzke vymedzenie kognitívnej vedy znie napríklad nasledovne: „Myslenie možno najlepšie vysvetliť v termínoch reprezentačnej štruktúry mysle a výpočtových procedúr, ktoré pracujú s týmito štruktúrami“ ([17], s. 25). V počiatočných etapách konštituovania kognitívnej vedy dominovalo úzke chápanie kognície zdôrazňujúce ***reprezentačnú*** schopnosť kognitívnych stavov a analogickosť povahy a funkcie mysle s digitálnym počítačom. V období ***zrenia*** a inštitucionalizácie kognitívnej vedy (1960-1985) sa preto výskum koncentroval predovšetkým na oblasť informatiky, počítačovej vedy a umelej inteligencie. Spomedzi teórií sa udomácnila ***výpočtovo-reprezentačná teória mysle*** (VRT) s filozofickými koreňmi siahajúcimi do 17. storočia k dielu „starých otcov“ kognitívnej vedy - Reného Descarta a Thomasa Hobbesa. Descartov pojem reprezentácie (schopnosti mysle zastupovať "čosi iné") sa začal vzťahovať na entity - nositeľov obsahu vo vnútri daného systému. Realita

[3] S „kognitívnou revolúciou“ sa spája rok 1948, kedy sa v Kalifornii uskutočnilo prvé tzv. Hixonské sympózium venované problému schopnosti nervového systému kontrolovať správanie systému. Na sympóziu vystúpili osobnosti, ktoré zásadne ovplyvnili ďalší vývin kognitívnej vedy - matematik John von Neumann, matematik a neurofyziológ Warren McGulloch a neuropsychológ Karl Lashley.

obsahov mysle, vnútorných reprezentácií sa stala základom presvedčenia o existencii mentálneho sveta ako ***autonómneho*** reprezentačného systému. Téza o závislosti kognície na reprezentovaní informácie zohrala zásadné miesto v úvahách predstaviteľov kognitívnej vedy o fungovaní inteligentných systémov. Reprezentácie (myšlienky, viery, predstavy) predstavovali zhluky informácií fungujúce ponad kognitívnymi procesmi. Tieto procesy (napríklad, vnímanie) sa podieľali na dekódovaní informácií zo sveta, čím vytvárali alebo menili reprezentácie. Predpokladalo sa, že kognitívne procesy majú komputačný charakter a vyžadujú vnútorné reprezentácie ako médium komputácie. Prívrženci VTR sa usilovali poskytnúť popis programov (softvéru) pre myseľ. Psychologické stavy sa chápali napríklad ako stavy reprezentujúce svet pomocou jazyka mysle [10] a psychologické procesy predstavovali výpočtové operácie s danými reprezentáciami. K najvýznamnejším predstaviteľom a popularizátorom kognitívnej vedy patrí Jerry Fodor a Noam Chomský. Fodor rozpracoval originálnym spôsobom funkcionálnu architektúru psychologických mechanizmov. V známej ***modulárnej teórii mysle***, postuloval ***moduly*** - autonómne jednotky, jednoduché výpočtové systémy nepodliehajúce vedomej kontrole. Modulárnou organizovanosťou sa vyznačovali podľa Fodora len vstupné kognitívne systémy (napríklad, vnímanie) na rozdiel od súčastí centrálneho systému (vedomie, rozhodovanie). Osobitý prínos Chomského, zástancu vrodenosti jazyka spočíval v rozpracovaní základných princípov „univerzálnej gramatiky“ [22].

Problematika reprezentačnej schopnosti stavov mysle bola previazaná so špecifickým úsilím pochopiť inteligentné systémy „zhora“, od kognitívnych stavov človeka smerom „dolu“ k porovnateľným stavom stroja (počítača). Tento prístup rozvíjali prívrženci tradičnej umelej inteligencie [26]. Nová umelá inteligencia si zvolila opačný postup „zdola“ od jednoduchých živých alebo umelých systémov smerom „hore“ ku komplexnejším prejavom správania systémov. Postulovanie vnútorných reprezentácií sa postupne nahradilo úsilím vypozorovať ako „samovoľne“ vzniká a ako vyvíja sa racionalita systému v meniacom sa prostredí.

Cenným prínosom tradície kognitívnovedného skúmania mysle sa stalo zaujatie postoja ***neutrality*** vo vzťahu k nositeľovi mysle. Napriek zjavnej neintuitívnosti abstrahovania od bežnej predstavy, t.j. pripisovania mysle živému organizmu, tento metodologický postup priniesol významné poznatky pri vysvetľovaní vzniku a mechanizmov fungovania vnímania, poznávania, pamäti, riešenia problémov atď. [38].

2.2 Mechanická lienka sedembodková

Človeka odpradávna fascinovali stroje, ktoré sám vymyslel, skonštruoval a v najlepšom prípade aj používal. Descartes, napríklad obdivoval automatické postavy a mechanizmus fontán v kráľovských záhradách Saint Germaine en Laye vybudované na mechanickom princípe. Mechanizmus fontán mu slúžil ako analógia na opis fungovania nervového systému človeka. Nervy pripodobňoval k trubiciam vo fontánach a rozumovú dušu k „tomu, kto sa stará o fontány“. Aj ďalšiemu francúzskemu filozofovi La Mettriemu pripomínalo ľudské telo „stroj, ktorý si sám naťahuje svoje pružiny“. V histórii názorov na fungovanie mozgu patrili k obľúbeným

metaforám telefónna centrála, telegraf alebo hydraulický systém. Už u G. W. Leibniza sa stretávame s myšlienkou automatizácie myslenia charakterizovanou v pojmoch reprezentácií. K realizácii automatizácie bolo potrebné zostrojiť kalkul, t.j. systém znakov v rámci ktorých možno vykonávať operácie na základe vopred stanovených pravidiel. Podmienky na zostrojenie „analytického stroja" vytvoril až v 19. storočí Charles Babbage. Stroj mal byť poháňaný parou a kontrolu postupnosti jednotlivých operácií zaisťoval „program" vpísaný do diernych štítkov. Nóvum spočívalo v schopnosti stroja „zapamätať si" vypočítané číslo a použiť ho v ďalšej operácii. Na svoju realizáciu si však projekt musel počkať ešte niekoľko desaťročí [32].

Iniciátorom myšlienky o skonštruovaní inteligencie v čisto fyzikálnom zariadení bol britský matematik Alan Turing. Ako priekopník počítačovej vedy navrhol v roku 1936 matematický koncept stroja, tzv. ***Turingov stroj***, schopného spracúvať informácie a vykonávať algoritmizované výpočty. Turing vychádzal z analógie výpočtového zariadenia a písacieho stroja, a predstavoval si nekonečne dlhú pásku, na ktorú sa mohlo písať a napísané znaky aj vymazať. Svojimi úvahami nastolil viaceré otázky: Čo všetko možno uplatnením ***Turingovho stroja*** pokladať za vypočítateľné? Nie je aj ľudské konanie vo svojej podstate výsledkom výpočtov? Jednou z najprovokatívnejších a zároveň aj ostro kritizovaných sa stala myšlienka o schopnosti počítačov ***myslieť***. V kontexte diskusií o spoločných znakoch, ale najmä rozdieloch medzi počítačom a človekom získal značnú popularitu myšlienkový experiment známy ako ***imitačná hra***. Hru hrali traja hráči, muž (A), žena (B) a moderátor (C). Cieľom moderátora nachádzajúceho sa v inej miestnosti než hráči je určiť, kto z dvojice je muž a kto žena. Moderátor ich pozná ako hráča X a hráča Y a na konci hry oznámi „X je A a Y je B" alebo „X je B a Y je A". [44] Turing zvažoval, čo by stalo, ak by sa jeden z hráčov nahradil počítačom a moderátor by ***nebol schopný*** rozlíšiť medzi ním a človekom. Nemohli by sme konštatovať, že prešiel takpovediac ***testom***, na základe ktorého by sme mu mohli pripísať myslenie a inteligenciu? Ostrú kritiku imitačnej hry známej skôr ako ***Turingov test*** vzniesol popredný filozof mysle J. Searle svojím argumentom „***Čínskej izby***" [41]. Jadro argumentu spočívalo v odvolávaní sa na jedinečnú schopnosť človeka uvedomovať si význam toho, čo si myslel alebo čo vypovedal. Počítače nie sú, podľa Searla, schopné introspektívnej kvality subjektívnej výpovede, uvedomovania si a rozumenia. To ich a priori vyraďuje z radov nositeľov mysle a inteligencie.

Predstava o mysliacich až strach naháňajúcich strojoch – robotoch sa v priebehu dlhoročnej diskusie stala zvlášť citlivou.

3 Hmota, duch alebo ementál

3.1 Miska tvarohu v našich hlavách

Takýmto nezvyklým prívlastkom označil v 17. storočí anglický filozof Henry More mozog - vodnatú beztvarú hmotu, ktorá podľa neho nemôže v žiadnom prípade disponovať schopnosťou myslieť. To, že sa s podobnou intuíciou stretávame v každodennej skúsenosti, myslím, neprekvapuje. Málokto si totiž uvedomuje, že

vo svojej hlave nosí objekt, ktorý sa vyvinul v priebehu približne 1 miliardy rokov evolúcie života. V dôsledku svojej organizácie dokáže mozog vytvoriť takmer nekonečnú sústavu rôznych typov aktivácie zodpovedajúcej neustálym premenám vonkajšieho sveta. Poznanie základných princípov tejto dynamiky pomáha čoraz viac pri odhaľovaní povahy chorôb, motívov konania a podstaty ľudskej existencie. Nie náhodou oficiálne kruhy Spojených štátov označili posledné roky 20. storočia „***dekádou mozgu***“. Vedecký výskum ľudskej mysle dosiahol nesmierny nárast vďaka uplatňovaniu nových metód a teoretických prístupov. Metódy výskumu mozgu ***post mortem*** (pitvy) obohatilo skúmanie štruktúry a aktivity poškodeného mozgu ***in vivo*** [3]. Spomedzi ***priamych metód*** sa predmetom rozsiahlych diskusií stalo úsilie odhaliť neurónové koreláty vedomej skúsenosti. Rozporuplné debaty o povahe slobody vôle, o „merateľnosti nemerateľného“, vyvolali pionierske výskumy B. Libeta z 80.-tych rokov 20. storočia. Libetove experimenty demonštrovali, že vedomé vôľové akty predchádza špecifická elektrická zmena v mozgu (potenciál pripravenosti, PP), ktorá začína 550 ms pred vôľovým aktom (pohyb zápästia). Subjekt si ***uvedomuje*** intenciu konať (vedomé želanie) 350-400 ms ***potom*** ako naštartuje PP, ale 200 ms pred motorickým aktom. Napriek tomu, že vôľový proces je iniciovaný nevedome, vedomá funkcia môže stále kontrolovať výstup. Sloboda vôle je zachovaná, konštatuje Libet, neiniciuje síce vôľový akt, ale môže kontrolovať prevedenie (neprevedenie) aktu [29].

Vzhľadom na absenciu detailnej „mapy mozgu“ viacerí filozofi a vedci uprednostňovali ***nepriame metódy*** skúmania vzťahu myseľ – telo (mozog). K významným stratégiám nepriameho prístupu na neurónovej úrovni sa radí neurónové kódovanie a na úrovni sietí vytváranie modelov spájajúcich neurónovú aktivitu s psychofyzikálnymi dátami behaviorálnych štúdií [23].

Nové pohľady do skúmania povahy ľudskej psychiky vniesli poznatky z výskumu neurologických a psychických syndrómov. Experimentálny výskum poškodení mysle ilustruje unikátnosť ľudského mozgu vzhľadom na také vlastnosti neurónovej aktivity ako je jednota, diferenciácia, plasticita, schopnosť reprezentovať či neurónový vývin. Cenný materiál poskytuje analýza spočiatku kurióznego poškodenia mysle tzv. "fantómové končatiny" (phantom-limbs). Zvláštnosť poškodenia sa prejavuje tým, že pacienti pociťujú prítomnosť neexistujúcich častí svojho tela. Napríklad pacient T., ktorý stratil ľavú ruku v dôsledku nehody začal po niekoľkých týždňoch pociťovať prítomnosť chýbajúcej ruky, jeho "prsty" boli schopné "uchopiť" predmety v dosahu ruky. Navyše, mnohí pacienti prežívajú vo svojich "fantómoch" až neznesiteľnú bolesť. Významný americký neurovedec V. Ramachandran zistil, že mapy povrchu tela sa skutočne môžu meniť. Mapa chýbajúcej ruky pacienta T. sa "znovuvytvorila" hneď na dvoch miestach, na tvári a na mieste nad amputovanou rukou. Stimulácia oboch častí tela vyvoláva u T. pocity vo fantómovej končatine, napríklad pocity v prstoch ruky. Keď sa T. zasmeje alebo pohne perami, impulzy aktivujú oblasť "ruky" v kortexe, vytvárajú ***ilúziu*** toho, že ruka stále existuje, zdá sa, akoby jeho mozog "halucinoval ruku" ([35], s.33). Z ďalších poškodení možno uviesť ***anosognóziu***, stav pri ktorom pacienti popierajú svoju paralýzu, k svojim končatinám sa správajú ako k cudzím objektom (napríklad, syndróm cudzej ruky). V dôsledku bilaterálneho okcipitálneho poškodenia dochádza k ***Antonovmu syndrómu*** – popieraniu slepoty nevidiacimi pacientmi. Pri syndróme „kôrovej slepoty“ („blindsight“) dochádza

k poškodeniu oblasti primárneho vizuálneho kortexu. L. Wieskrantz [46] pri výskume tohto syndrómu zistil, že napriek slepote ľavého vizuálneho poľa je pacient schopný vnímať, rozlišovať predmety nachádzajúce sa v poškodenom poli. Pacienti síce popierali fakt, že vidia, ale metódou „nútenej voľby“ (forced choice) prítomnosť objektu „uhádli“. Weiskrantz interpretoval túto skutočnosť ako príklad oddelenia rozlišovacej vizuálnej schopnosti a vizuálneho uvedomenia si pacienta. Možným opakom „kôrovej slepoty“ je ***prosopagnózia*** - neschopnosť rozpoznať známe tváre (vrátane vlastnej).

Komplikovaný vzťah vlastností jednoty a diferenciácie ilustruje experiment s pacientmi postihnutými syndrómom „***rozštiepeného mozgu***“ (split-brain). Z dôvodu stlmenia epileptických záchvatov sa pacientom odstránil mohutný zhluk neurónových vlákien (corpus callosum) spájajúci obe hemisféry mozgu. Napriek tomu, že mimo laboratória sa pacienti správali normálne, pri experimentoch sa prejavovala abnormalita najmä v medzihemisferickej integrácii zmyslovej a motorickej informácie. Pacientom, napríklad, premietli priestorové pozície na pravej strane plátna, ktoré sa prezentovali ľavej hemisfére a nezávisle na ľavej strane plátna sa prezentovali priestorové pozície pravej hemisfére. Oddelené hemisféry pacientov sa obrazcami „zaoberali“ ako s odlišnými problémami. Niektorí autori predpokladajú, že strata návratných spojení je zodpovedná za stratu vedomej integrácie medzi oboma hemisférami mozgu. Ľudia s normálnym mozgom nie sú schopní vnímať dve nezávislé vizuálne následnosti ako súhrn dvoch paralelných úloh, skôr kombinujú vizuálnu situáciu do jednej scény. V prípade „rozštiepeného mozgu“ sa zrakové pole každej z hemisfér rozštiepi na dve v strede. Pacient však nevníma svoj zrak ako rozpoltený. Ak sa ľavej hemisfére ukáže pravá strana tváre, tvrdí, že vidí celú tvár. Niektorí autori interpretujú tento syndróm ako poškodenie vedomia [42], iní skôr ako poškodenie miery dosahu zmyslového vstupu k jednotlivým izolovaným hemisféram. Ak sú obe hemisféry mozgu oddelené, množstvo zmyslového vstupu je polovičné. Úbytok zrakovej informácie však neimplikuje, že pacienti s „rozštiepeným mozgom“ sú si vedomí len spolovice. Vedomie nezávisí od stupňa informácie prichádzajúcej so zmyslovej modality (vedomým disponujeme napríklad aj keď sme zavretí v tmavej miestnosti). Uvedený syndróm môže poukazovať aj na to, že obe hemisféry mozgu spracúvajú informácie oddelene a jedna nie je schopná vyvodiť nič významné o vedomí. Zložité vzťahy medzi vlastnosťami plasticity a anatomickej väzby dokladá napríklad „***hemineglekt***“, pri ktorom (poškodenie pravého parietálneho laloka) si pacienti uvedomujú len ľavé časti vecí alebo sveta. Je prekvapujúce, že vedomá rekonštrukcia tohto stavu sa nenarúša, akoby úsilie mozgu integrovať „premohlo“ existujúce poškodenie.

3.2 Myseľ/mozog ako počítač

Schopnosť mozgu reprezentovať vonkajší aj vnútorný svet sa stala predmetom diskusií o podobe zmysluplnej teórie tejto komplexnej aktivity. Ťažkosti s charakteristikou mysle a jej vzťahom k mozgu vyústili do „etablovania“ zvláštnej entity nazvanej myseľ/mozog (mind/brain). Predpoklad synchrónne prebiehajúcej aktivity nervovej sústavy implikoval podľa viacerých autorov prítomnosť komputačného (výpočtového)

princípu v mozgu. Štýl komputácie reprezentovala premena jednej vzorky na druhú prostredníctvom veľkých konfigurácií synaptických spojení tvoriacich paralelne distribuovaný systém (PDS). V súvislosti so spomínanou výpočtovo-reprezentačnou teóriou mysle sa modelom pre vnútorné ***spracovanie informácií*** stáva digitálny počítač. Komputácia vyžaduje "manipuláciu vnútorných symbolických reprezentácií na základe pravidiel". Povedať, že myseľ spracúva informácie znamená, že ich spracúva formálne. Formálne procesy (algoritmy) majú následne schopnosť ***reprezentovať***. Kognitívni vedci rozlišovali formálne operácie so symbolmi a reprezentačné vzťahy medzi reprezentáciami a tým, čo reprezentujú. Jadro uvedeného prístupu tvorí téza, podľa ktorej systém (organizmus alebo stroj) produkuje zmysluplné správanie na základe vykonávania ***formálnych operácií*** so symbolickými štruktúrami, ktoré majú reprezentačný vzťah k svetu. V dôsledku toho dochádza k oddeľovaniu formálnych procesov od ich fyzikálneho podkladu. Postupne sa rodí metafora mysle ako ***počítača*** a obraz človeka (stroja?) ako funkcionálneho systému, ktorý kóduje informácie z prostredia, analyzuje ich, uchováva a vyvoláva. Pre opis fungovania mysle sa stala základnou úroveň spracovania informácií. Fakt, že v digitálnom počítači spĺňa funkciu spracovania informácií program priviedol filozofov k analógii myseľ-mozog – softvér-hardvér. Zmysluplnosť uplatnenia metafory počítača sa zdôvodňovala tým, že počítač je podobne ako myseľ abstraktný fyzikálne realizovaný systém. Analógia nastoľuje otázku, či sa má myseľ vzťah k mozgu tak ako softvér k hardvéru počítača. Vďaka programovaciemu jazyku je počítač schopný simulovať rozličné systémy spracovania informácií. Za istých podmienok sa stáva univerzálnym ***Turingovým strojom***, t.j. strojom schopným simulovať stroj vykonávajúci akékoľvek komputačné postupy. Predpokladá sa detailné reprodukovateľné správanie simulovaného stroja v rámci schémy vstup-spracovanie informácie-výstup. Otázkou zostáva, či možno aktivity mysle, vedomia a inteligentného správania vôbec pokladať za komputačné postupy či procesy. Kladná odpoveď zástancov umelej inteligencie (AI) vychádza z charakteristiky mozgu/mysle ako systému spracovania informácií [27].

Americký filozof mysle Paul Churchland [23] priblížil analógiu mozog-počítač nasledovne. Podobne, ako jednotlivé časti vo vnútri televíznej obrazovky, synaptické spojenia medzi neurónmi sú relatívne stabilné. Neuróny majú však schopnosť meniť svoje vnútorné aktivačné úrovne „v zlomku sekundy“. Tak ako pixle na televíznej obrazovke, každá úroveň aktivácie neurónu je aktualizovaná podnetmi z vonkajšieho sveta. Celková vzorka aktivácie vytvára obraz lokálnej situácie mozgu, časová následnosť permanentne sa meniacich vzoriek konštituuje schopnosť mozgu reprezentovať meniaci sa svet. Za predpokladu, že štandardná obrazovka má asi 525X360 pixelové rozlíšenie, 200 000 pixelov tvorí reprezentačnú schopnosť obrazovky. Ľudský mozog má asi 100 miliárd neurónov, z ktorých každý disponuje vlastnými „hodnotami jasnosti“. Ak by sme napríklad každý neurón pokladali za pixel, potom by schopnosť mozgu bola 500 000 krát väčšia než schopnosť obrazovky. Na tomto mieste sa podľa Churchlanda natíska otázka, kto sa v mozgu pozerá na prebiehajúcu show? V protiklade k intuíciám z bežnej skúsenosti sa ukazuje, že nik. Neexistuje žiadne „Ja“ ponad mozog ako celok, každú časť mozgu „pozoruje“ iná časť mozgu dokonca, často viaceré časti súčasne.

K argumentom proti pripodobňovaniu mozgu a mysle k fungovaniu počítača patria najčastejšie námietky z anatómie a fyziológie, zo spôsobu spracovania informácií, zo špecifickosti vlastností mysle (mozgu) a z účelovosti systému. Oxfordská neurovedkyňa Susan Greenfieldová [18] prirovnáva neurónovú sieť skôr k chaotickému klbku špagiet než k integrovanému obvodu v počítači. Ako zdôrazňuje, mozog je chemický systém a aj jeho elektrické prejavy majú chemický pôvod. Ani chemické deje bez priamych elektrických prejavov nie sú analogické činnosti počítača, pretože niektoré z nich ovplyvňujú reakciu bunky na prichádzajúce signály. Chemické zloženie neurónov sa mení, a preto ich nemožno pokladať za nemenný hardvér. Mozog sa na rozdiel od počítača nemusí riadiť naprogramovaným algoritmom, jeho aktivita je spontánna, „rozhodne sa“ pohnúť telo alebo „cíti“, že má bolesti a „usiluje sa“ ich zmierniť. Navyše, ako dokazuje individuálny neurónový vývin, celkovú vzorku spojení možno opísať všeobecnými pojmami, ale mikroskopická variabilita mozgu ho robí jedinečným (dva identické mozgy neexistujú!).

Napriek tomu, že konceptuálny rozdiel medzi hardvérom a softvérom nemožno doslova napasovať na jednotlivé úrovne organizácie mozgu, heuristický význam ***metafory*** počítača sa nevytráca. Mozog (myseľ) možno minimálne chápať ako biologický počítač, ktorý spracúva informácie. Ak uvažujeme o ***duplikácii*** schopností ľudskej mysle (mozgu), úsilie o zostrojenie mysliacich a cítiacich počítačov nestratí nič zo svojej provokatívnosti. Vonkoncom nie v dobe, keď sa podobné úsilie realizuje v prípade poškodenia a zlyhávania funkcií tak významných orgánov ľudského tela, ako je srdce, obličky alebo sluchového aparátu [24].

4 Transparentná myseľ

4.1 Ilúzia Leibnizovho mlyna

Frekventovanými prekážkami pri hľadaní vedy o mysli sa stali nedorozumenia súvisiace s vymedzením „kompetencií“ a vzájomných vzťahov participujúcich vedných disciplín. Tradičný rozkol vedeckého a filozofického skúmania „vnútorného“ možno ilustrovať na príklade Leibnizovho mlyna, myšlienkového experimentu filozofa 17. storočia G. W. Leibniza, ktorý vo svojom diela ***Monadológia píše***: „Napokon treba priznať, že pociťovanie a to, čo od neho závisí, je ***nevysvetliteľné pojmami mechaniky***, čiže pomocou tvarov a pohybov. A ak si predstavíme stroj, ktorého štruktúra by produkovala myslenie, vnímanie, pociťovanie, mohli by sme si ho myslieť, zachovávajúc tie isté pomerné miery, taký veľký, že by bolo možné vstúpiť doňho ako do mlyna. Za takej danosti, ak poprezeráme jeho vnútro, nájdeme iba časti, ktoré sa vzájomne poháňajú, a nikdy nič také, čo by vysvetľovalo pociťovanie. Preto (pociťovanie) treba hľadať v jednoduchej, a nie zloženej substancii a či v stroji“. Podľa P. Churchlanda [23] Leibnizova úvaha nespochybňuje tézu o identifikácii vedomej mysle s fyzikálnymi procesmi mozgu. Fakt, že vnútorný svet si uvedomujeme v podobe predstáv, myšlienok, pocitov neznamená, že tieto stavy vytvárajú iný než fyzikálny typ reality (t.j., že nie sú súčasťou stroja). Nemáme dočinenia s javom odlišnej povahy, ale skôr s iným typom ***poznania*** (slnko napríklad vnímame ako

neveľký oranžový disk na oblohe). V prípade predstavovaného mlyna jednoducho nevieme, čo máme hľadať (nepoznáme jazyk mechanizmov stroja). Leibniz sa preto, tvrdí Churchland, otázke vyhol a jeho myšlienkový experiment ako námietka proti materializmu neobstojí. Odvolávanie sa na spôsob, akým prežívame vlastný vnútorný svet zjavne nepostačuje na postulovanie ontologickej priepasti medzi mysľou a telom (Leibniz sa to pokúšal teóriou o predurčenej harmónii). Inšpiratívnosť a význam Leibnizovho myšlienkového experimentu odkrýva podľa môjho názoru potrebu vysvetliť mechanizmy a funkcie ***javení***, toho, prečo sa nám náš vnútorný, ale aj vonkajší svet javí tak ***ako sa nám javí***.

Absencia uspokojivého vysvetlenia tohto problému tvorí zrejme aj jednu z príčin nedorozumení vo vzťahu k uplatňovaniu zobrazovacích techník aktivity mozgu. Podobne ako v histórii vedeckých objavov a zavádzania nových výskumných metód aj používanie neurotechnológií vyvolalo u teoretikov obavy. Pokrok v neurozobrazovaní a psychofarmakológii umožňuje korelovať psychologické stavy so špecifickými funkciami mozgu a meniť psychologické stavy modifikovaním neurochémie. Tieto skutočnosti nastoľujú nové etické, sociálne a filozofické otázky: Čo ***hľadáme*** pri pohľade na MR snímky mozgu človeka? Kde sídli duša? Čo si jedinec myslí alebo čo si pamätá? Kedy jedinec hovorí pravdu alebo klame? Alebo skôr sa detegujú na neurobiologickej úrovni naše ***presvedčenia*** o tom, ako spoločnosť chápe klamstvo? Problémy späté s „monitorovaním a manipuláciou“ aktivity mozgu, ovplyvňovaním myšlienok a konania človeka intervenciou na neurobiologickej úrovni (počas neurochirurgického zákroku), „ohrozením privátnosti jedinca“ sa stali stredobodom súčasného teoretizovania o povahe mysle. Významom a vplyvom najnovších poznatkov vedeckého skúmania aktivity mozgu sa zaoberá novovznikajúca disciplína ***neuroetika***, ktorú tvorí etika neurovedy a neuroveda etiky [25]. Zámerom ***etiky*** neurovedy je formulácia etických princípov riadiacich výskum mozgu, liečba neurologických chorôb a dopad skúmania aktivity mozgu na sociálne, morálne a filozofické názory. ***Neuroveda*** etiky sa koncentruje na vedecké skúmania etického správania, t.j. neurobiológiu reprezentácií hodnôt, morálneho zdôvodňovania a správania. Súčasné výskumy mozgu nasvedčujú tomu, že základom morálneho usudzovania a správania, schopnosti človeka myslieť a konať morálne „nezodpovedá“ jedna oblasť mozgu, t.j. neexistuje „morálne centrum“ mozgu. Obzvlášť významný je dopad neurotechnológií na sociálny a mravný život ľudí. Spoločnosť vychádza z predpokladu o zodpovednosti ľudí za svoje činy. Dôsledky konania, ktoré je a ktoré nie je pod vôľovou kontrolou jedinca sa posudzujú morálne aj legálne. Následky dopravnej nehody zapríčinené rýchlou jazdou vedú k odlišnému rozhodnutiu o „vine“ než v prípade, ak by šofér zlyhal v dôsledku infarktu alebo psychomotorickej epileptickej príhody [4].

Posudzovanie miery opodstatnenosti obáv plynúcich z nastolených otázok otvára priestor filozofickej reflexii. Okrem ujasňovania terminologických a konceptuálnych otázok pojmov myseľ, vedomie, ja, sloboda vôle – spočíva v odmietnutí striktných hraníc medzi povahou a cieľmi vedeckého a filozofického bádania.

4.2 „Čo na srdci, to na jazyku“

V bežnej skúsenosti prežívame stavy radosti bolesti, hnevu, plánujeme, želáme si, premýšľame. Každý človek disponuje vlastným vnútorným svetom, s ktorým sa dokáže podeliť aj s druhými. Ľudová slovesnosť odkrýva vzťah stavov mysle k telu mnohorakým spôsobom, ako napríklad: "chodí ako telo bez duše", "škoda tej duše v tom mizernom tele", "duša nezhynie, hoc i telo pominie". Obraz človeka ako jednoty mysle a tela, duše a mozgu, vnútorného a vonkajšieho, psychického a telesného máme pred očami dennodenne. Na pozadí tejto optiky žijeme a konáme vcelku neproblematicky (mimo metafyzického kontextu). K pretrvávaniu problému povahy mysle podobne ako psychofyzického problému prispievajú nejasnosti o povahe jazyka. Hovoriť o realite, t.j. jestvovaní metafor, predpokladá vysporiadať sa so vzťahom doslovné-metaforické. Jazyk o vnútorných mentálnych stavoch je metaforický. V skutočnosti vieme, že vo vnútri našich tiel (mozgu) sa nič také ako mentálne a kognitívne stavy v zmysle "náprotivkov" fyzikálnych stavov nenachádza. Mentálne stavy (myšlienky, pocity, predstavy) sú metafory, pomocou ktorých vypovedáme o svete vlastnej psychiky. Na rozdiel od bežnej skúsenosti, filozofi často zamieňajú metaforický jazyk za doslovný. Problematickosť tohto vzťahu evokovala filozofom diametrálne odlišné ***obrazy*** vzťahu sveta, objektov a jazyka. Metaforickosť výpovedí o subjektívnych pocitoch, predstavách nadobudla napríklad podobu výpovedí o mentálnych stavoch, objektoch a procesoch. Nedorozumenia vo vzťahu k vysvetľovaniu vnútorných stavov posilňuje aj názor, podľa ktorého charakteristika mentálnych stavov ako metafor upiera mentálnym stavom ***reálnu*** existenciu, resp. redukuje ich na čosi iné, než čím sú v skutočnosti. Metaforickosť jazyka však ***nemôže*** upierať ani potvrdzovať existenciu „čohosi“. To isté platí aj pre "obyvateľov" vnútorného sveta človeka. Ak nazveme malé dieťa „anjelom“, nepredpokladáme, že ***doslova*** rozprestrie krídla a odletí. Dieťa k anjelovi pripodobňujeme preto, lebo sme si zvykli spájať detskú nevinnosť s rozprávkovou čistotou a dobrotou anjelov.

Navyše, jadro uvedeného omylu spočíva v stotožňovaní doslovného s vedeckým jazykom a metaforického s básnickým (filozofickým) jazykom. Dôležitý nie je jazyk sám, jazykový útvar, typ, ale problémy - ciele, ktoré prostredníctvom neho vyjadrujú – problémy, ktoré tvoria súčasť konkrétneho diskurzu, kontextu – rozprávky, literárneho príbehu, vedeckej hypotézy.

Pocity, myšlienky a reprezentácie spolu so svojimi nositeľmi patria do okolitého sveta podobne ako počasie, medzinárodné vzťahy alebo teórie. V tomto zmysle nesporne existujú reálne! Navyše, opodstatnenosť dištinkcie doslovné-metaforické neznamená, že vedecký jazyk nemôže nadobudnúť metaforickú podobu. V dôsledku experimentálneho a teoretického výskumu v neurovedách, psychológii, kognitívnej vede dochádza ku zmene chápania metafory. Popri tradičných dekoratívnych funkciách sa v súčasnosti akcentujú najmä kognitívne a argumentatívne funkcie metafory. Metafory tvoria súčasť testovateľných teórií a modelov, spoluvytvárajú metafyzický základ vedeckej práce, zohrávajú významnú lingvistickú a heuristickú rolu pri rozširovaní terminológie na nové javy. P. Churchland napríklad používa vo svojom neurokomputačnom modeli vedomia metaforu rekurentnej siete, ktorá „prináša informácie o skorších stavoch tej istej vrstvy“, čím systém nadobúda typ krátkodobej

pamäte. S. Greenfieldová pripodobňuje vedomie ku ***kontinuu*** vyvíjajúcemu sa v ontogenéze a vo fylogenéze. Vedomie vysvetľuje prostredníctvom metafory „kameňa hodeného do vody“. Tak ako závisí veľkosť, šírka vlniek vo vode od veľkosti kameňa a od sily, ktorou ho do vody vhodíme, podobne, aj veľkosť podnetu (silná, mierna bolesť) mení stupeň vedomia (hlboké, plytké).

Špecifické vlastnosti prefrontálnej kôry inšpirovali E. Goldberga [16] k metafore mozgu ako orchestra s dirigentom. Obojsmerná prepojenosť prefrontálnej kôry s mnohými štruktúrami mozgu jej umožňuje koordinovať a integrovať činnosť, inými slovami, robí ju dirigentom zložitého a premenlivého súboru hráčov. Americký neurovedec Antonio Damasio prirovnáva aktivitu automatického systému k rozvetvenému stromu. V metafore vysokého hustého stromu vidí zapísané celé dejiny evolúcie, pretože stále rastúce vetvy stromu si uchovávajú obojsmernú komunikáciu so svojimi koreňmi. Základným cieľom vrodenej a automatizovanej výbavy stromu (mozgu) je dosiahnutie homeostázy, čo možno najlepšieho prispôsobenia sa vnútornému a vonkajšiemu prostrediu. Strom reprezentujúci ***homeostatický systém*** tvorí celý rad úrovní od tých najjednoduchších po zložitejšie. Na najnižších vetvách sa nachádzajú metabolické procesy (kontrola srdcových sťahov a krvný tlak, základné reflexy (reakcia na hluk alebo dotyk), imunitný systém (likvidácia baktérií, parazitov). Na stredných vetvách prebieha správanie späté s bolesťou (pri popálení vysielajú zasiahnuté bunky chemické signály), so slasťou (tvorba endorfínov). Na vyššej úrovni prežívame hlad, smäd, hru, sexualitu a ešte vyššie emócie, „korunný šperk automatizovanej regulácie života“ ([4], s. 44). Deje na uvedených úrovniach sú síce aktívne už pri narodení, ***kedy*** sa však budú používať závisí od ďalšieho vývinu a skúsenosti indivídua.

Filozofi, neurovedci, psychológovia a kognitívni vedci zdôvodňujú používanie metaforického jazyka rôzne: existenciou viacerých úrovní vysvetľovania mysle, absenciou detailného poznania fungovania mozgu, komunikovaním významov novým spôsobom, opisom ťažko uchopiteľných javov, ale aj povahou prístupu nositeľa mysle k vlastnej „vnútornej skúsenosti“. Vzhľadom na svoju biologickú výbavu, človek nemá vedomý prístup k mechanizmom a procesom zabezpečujúcim chod reprezentácií alebo predstáv. Navyše, ako vieme z experimentálneho výskumu, ľudská myseľ podlieha pri spracúvaní a vyhodnocovaní informácií celému radu kognitívnych ilúzií a omylov [7], [9].

Problém metafory nespočíva v zmysluplnosti či životaschopnosti metaforického jazyka, ale skôr v posudzovaní používania vhodných a nevhodných metafor. Či si to uvedomujeme alebo nie, metafory podobne ako myšlienky a teórie ovplyvňujú naše konanie. Uznanie metaforickej povahy filozofického jazyka môže viesť k rozuzleniu "večných" problémov mysle, Ja, vedomia atď. Ako som naznačila, vedeckosť hypotéz, modelov a teórií zďaleka nevylučuje používanie poetického, metaforického jazyka.

5 Záver

Vo filozofii dominoval v rôznych obmenách model človeka ako jednoty psychického (mentálneho) a telesného (fyzikálneho). Hypostazovanie protikladov duševné-telesné z bežnej skúsenosti vyústilo v teórii do ***alchymistického*** obrazu mysle a človeka, zmesi

poznateľného, nepoznateľného, záhadného a prirodzeného. Sňať ťažobu, ktorú na pozadí tohto obrazu niesli na svojich pleciach teoretici sa podujali vedci. Jedno je isté. Myseľ ***nemôže*** byť ***zároveň*** nadprirodzená a prirodzená. Charakteristika psychických stavov ako prirodzených javov sveta je prelomová. Naturalistická perspektíva prispieva k odbúravaniu explanačne vágnych prístupov a teórií. Predstava mysle ako nemateriálnej substancie, „ducha v stroji“, výsostne ľudskej kvality „tróniacej“ na pomyselnom vrchole tvorstva ustupuje do ríše mýtov. Začlenenie psychických stavov k prirodzeným javom spochybňuje aj pokusy filozofov zlučovať prirodzené s neredukovateľným alebo subjektívne s neuchopiteľným [20]. Vnútorný svet človeka v celej svojej bohatosti a komplikovanosti sa začlenil k vedeckým explanandám. Klinické pozorovania, experimentálny výskum normálnej a poškodenej mysle, modely a teórie odhalili skostnatenosť viacerých filozofických konceptov mysle. Spochybnili sa striktné hranice medzi vedomými a nevedomými stavmi mysle, rozumom a emóciami, doložila sa obmedzenosť introspektívneho nazerania a iluzórnosť existenčnej spätosti vedomia a jazyka. Neopodstatnenosť striktných dištinkcií sa týka aj vzťahu medzi metódami tzv. prvej a tretej osoby, proklamovaných niektorými súčasnými filozofmi (mysle). Prehodnotenie tradičného obrazu človeka ako slobodného, vedomého a racionálne konajúceho subjektu je výzvou pre všetkých teoretikov mysle a vedomia.

Úsilie nositeľov mysle skúmať a vysvetľovať vlastnú myseľ si nesporne zaslúži rešpekt. Porozumieť vlastnej mysli a mysli druhých neraz predpokladá vzdorovať tradícii, pohodliu vlastných predsudkov a vier. V našej tradícii a kultúre nie je jednoduché odbúravať ***stigmu*** spätú s diagnózou a liečbou „duševne chorých“. Naplniť aspoň sčasti túžbu po hlbšom poznaní pôvodu, vývinu a fungovania mysle vyžaduje nemalú námahu a nedá sa nijak urýchliť.

O tom, či myseľ nahliadneme z nadprirodzenej alebo prirodzenej perspektívy, rozhodujeme sami. Ak si zvolíme druhú, ťažšiu cestu, vykročíme v duchu hippokratovského oddémonizovania mysle a tela človeka. Vysvetľovanie povahy, zákutí ľudskej mysle má pred sebou dlhú a tŕnistú cestu. Prehlbovanie znalostí o vzťahu mysle a konania človeka, úsilie zmierňovať všetky formy ľudského utrpenia, myslím, hovoria za všetko. Obozretný optimizmus symbolizuje nádej vo vzťahu k sebapoznaniu človeka. Jozef Kelemen to výstižne vyjadril slovami: „***... stroj nemusí byť nevyhnutne technickým zariadením podliehajúcim výlučne fyzikálnym zákonom, ale môže byť aj útvarom duchovnej kultúry, výplodom a súčasťou mýtu, postaveného ako protipól našich neistôt a obáv vo svete, v ktorom žijeme. Stroj nás nakoniec môže naplniť pocitom pokoja, pretože umožňuje tušiť neomylného strojvodcu***“ ([27], s. 53).

Literatúra

[1] Andreansen, N.: *Brave New Brain*. Oxford, Oxford University Press, 2001.

[2] Bechtel, W., Graham, G.: *A Companion to Cognitive Science*. Oxford, Blackwell, 1999.

[3] Crick, F.: *The Astonishing Hypothesis: The Scientific Search for the Soul*. New York, Scribner, 1994.

[4] Damasio, A.: *The Feeling of What Happens.* Praha, Dybbuk, 2004.

[5] Descartes, R.: *Vášně duše.* Praha: Mladá fronta 2002.

[6] Descartes, R.: *Meditácie.* Bratislava, Chronos, 1997.

[7] Dennett, D.: *Consciousness Explained.* Boston, Little, Brown, 1991.

[8] Edelman, G., Tononi, G.: *A Universe of Consciousness*. New York, Basic Books, 2000.

[9] Eysenk, M., Keane, M.: *Kognitivní psychologie.* Praha, Academia, 2008.

[10] Fodor, J. A.: *The Modularity of Mind.* Cambridge, MIT Press, 1983.

[11] Fodor, J. A.: Precis of The Modularity of Mind. *Behavioral and Brain Sciences,* 8*,* (1985) 1-46.

[12] Gálik, D.: Teória evolúcie a filozofia. In Kvasnička, V. a kol.: *Umelá inteligencia a kognitívna veda II.* STU, Bratislava, 2010, s. 89-123.

[13] Gáliková, S., Gál, E.: *Antológia filozofie mysle.* Bratislava, Kalligram, 2003.

[14] Gáliková, S.: *Psyché – od animálnych duchov k neurotransmiterom*. Bratislava, Veda, 2007.

[15] Gardner, H.: *The Mind's New Science: A History of the Cognitive Revolution*. New York, Basic Books, 1985.

[16] Goldberg, E.: *Jak nás mozek civilizuje*. Praha, Karolinum, 2004.

[17] Green, D. W.: *Cognitive Science: An Introduction.* Malden, MA, Blackwell, 1996.

[18] Greenfield, S.: *Journey to the Centers of the Mind*. New York, W. H. Freemann, 1995.

[19] Harnish, M. R.: *Minds, Brains, Computers: An Historical Introduction to the Foundations of Cognitive Science*. Oxford, Blackwell, 2002.

[20] Chalmers, D.: *The Conscious Mind.* New York: Oxford University Press, 1996.

[21] Chomsky, N.: *Modular approaches to the study of The mind*. San Diego, San Diego State University Press, 1984.

[22] Chomsky, N.: *New Horizons in the Study of Language And Mind*. Cambridge, Cambridge University Press, 2000.

[23] Churchland, P. M.: *The Engine of Reason, the Seat of the Soul.* Cambridge, MA, MIT Press, 1995.

[24] Churchland, P. S.: *Brain-Wise: Studies in Neurophilosophy*. Cambridge, MA, The MIT Press, 2002.

[25] Illes, J.: *Neuroethics.* Oxford, Oxford University Press, 2006.

[26] Kelemen, J.: *Strojovia a agenty.* Bratislava, Archa, 1994.

[27] Kelemen, J.: *Myslenie a stroj.* Bratislava, Kaligram, 2010.

[28] Koukolík, F.: *Lidský mozek.* Praha, Portál, 2000.

[29] Libet, B.: Do We Have Free Will? *Journal of Consciousness Studies,* **8-9** (1999), pp. 47-57.

[30] Nagel, T.: What it is like to be a bat? *Philosophical Review,* **4** (1974) pp. 435-50.

[31] Penfield, W., Rasmussen, T.: *The Cerebral Cortex of Man: A Clinical Study of Localization of Function*. New York, MacMillan, 1950.

[32] Petrů, M.: *Fyziologie mysli – Úvod do kognitivní vědy*. Olomouc, Triton, 2008.

[33] Piaget, J.: *The Language and the Thought of the Child.* New York, Humanities Press, 1959.

[34] Putnam, H.: Philosophy and Our Mental Life. In: Beakley, B., Ludlow, P. (ed.): *The Philosophy of Mind* Oxford, Blackwell, 1992.

[35] Ramachandran, V. S., Blackeslee, S.: *Phantoms in the Brain.* New York, Quill, 1998.

[36] Rybár, J. Ľ. Beňušková Ľ. Kvasnička, V.: *Kognitívne vedy*. Bratislava, Kalligram, 2002.

[37] Rybár, J., Kvasnička, V., Farkaš, I.: *Jazyk a kognícia*. Bratislava, Kalligram, 2005.

[38] Rybárová, D.: Teória "ladenness" a modulárna teória mysle. *Filosofický časopis*, **3** (2001) s. 373-379.

[39] Ryle, G.: *The Concept of Mind*. Oxford, Hutchinson, 1949.

[40] Searle, J.:*The Rediscovery of Mind.* Cambridge, MA, MIT Press, 1992.

[41] Searle, J.: *Mysl, mozek a věda.* Praha, Mladá fronta, 1994.

[42] Sperry, R. W.: Forebrain commissurotomy and conscious awareness. *Journal of Medicine and Philosophy,* **2** (1977) pp. 101-126.

[43] Sternberg, R. J.: *Kognitivní psychologie*. Praha, Portál, 2002.

[44] Thagard, P.: *Úvod do kognitivní vědy.* Praha, Portál, 2001.

[45] Watson, J. B.: „Psychology as the Behaviorist Views It“. *Psychological Review*, **20** (1913) pp 158-177.

[46] Weiskrantz, L.: Some Contributions of Neuropsychology of Vision and Memory to the Problems of Consciousness. In: A. Marcel, E. Bisiach (ed.): *Consciousness in Contemporary Science*. Oxford, Oxford University Press, 1989, pp. 183-199.

[47] Wittgenstein, L.: *Philosophical Investigations.* Oxford, Basil Blackwell, 1953.

Remaking Robot. Od mýtu technického pokroku k subverzním strategiím posthumanismu

Jana HORÁKOVÁ[1]

Abstrakt. Příspěvek je věnován robotovi jako figuře modernistického mýtu pokroku. Z této perspektivy budeme sledovat Čapkovy roboty ve hře R.U.R. a jejich následovníky v oblasti robotického umění. Robot v umění už neznamená to stejné co na začátku 20. století, ale jeho funkce je stále táž: Umožňuje artikulovat problémy sociální, politické, ale i obecně lidské. Robotické umění stále funguje jako kritický, satirický a subverzní nástroj užívaný umělci k vyvažování tendencí dominujících naší kultuře, ve které stroje, tentokrát informační a komunikační technologie, hrají možná ještě větší roli než tomu bylo na začátku století.

1 Robot jako remake

Robot se velice záhy po svém uvedení na ´prkna, která znamenají svět´ ve hře Karla Čapka *R.U.R. Rossum's Universal Robots* (1920) stal symbolem tendencí k poměřování lidských schopností schopnostmi strojů (a naopak), které zasahují mnohem širší sféru lidských aktivit, než je oblast umělecké kreativity. Čapkův robot není jen jedním z celé řady uměleckých ztělesnění člověka ´věku stroje´, či ´člověka-stroje´, mezi které můžeme zařadit celý zástup lidských postav ztvárněných v geometricky hranatých a kuželovitých tvarech kubisty, futuristy, konstruktivisty, ale i surrealisty. Čapkovi roboti jsou především skeptickou a satirickou reakcí na vize nové společnosti a nového člověka formulované příslušníky uměleckých avantgardních hnutí, které jejich přívržence nakonec přivedly k službě některému z totalitních řádů, ať už italskému fašismu nebo ruskému komunismu.

Stejnou měrou jakou figura robota odráží utopické vize nových společenských řádů a jejich ideálních příslušníků, reflektuje také aktuální, všeobecně sdílenou tendenci ´věku stroje´, jak bývá označováno období přibližně mezi dvěma světovými válkami. Pro tuto dobu je příznačný vrcholící proces zavádění výrobních montážních linek a s nimi spojené rozšíření masové výroby, rozvoj masových médií a masové

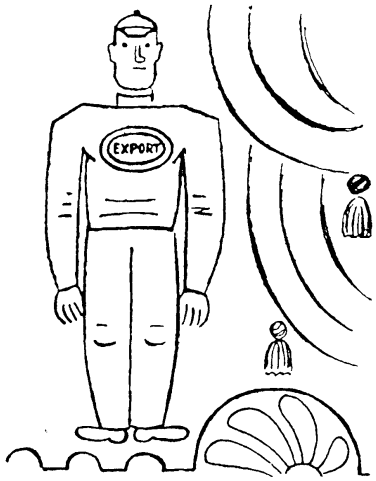
[1] Masarykova univerzita, Filozofická fakulta, Ústav hudební vědy, A. Nováka 1, 602 00 Brno, Česká republika. E-mail: horakova@phil.muni.cz

kultury produkujících ´masového člověka´. Touhy tohoto ´nového člověka´ vtipně glosoval Josef Čapek, významný český kubistický malíř a spisovatel, ale také autor slova robot a kostýmů pro oficiální premiéru *R.U.R.* v Národním divadle v Praze (25. ledna 1921), ve svém uměleckém eseji *Homo Artefactus* vydaném v roce 1924. S neskrývanou dávkou ironie a nadsázky psal:
„Moderní člověk nechce být ojedinělou loutkou a hračkou; moderní lidstvo přeje si být zmechanizováno tak, jak je, zmechanizováno šíře a obecněji ve všech svých složkách, vrstvách, vztazích, povoláních a zálibách.“ [3]

Čapkovi roboti přesahují svým významem řadu mnoha jiných dobových ztělesnění člověka ztvárněného ve strojově geometrické formě tím, že nejsou pouze výrazem nové estetické senzibility moderní doby, ale spíše komentářem všeobecnějších tendencí, kterých jsou změny v oblasti umění součástí. Jinými slovy, Čapek využívá charakteristické rysy konkrétních uměleckých směrů své doby, aby je ve stejný okamžik komentoval (tematizoval) jako součást širších společenských jevů. Tento přístup souvisí s celkovým postojem Karla Čapka vůči dobovým uměleckým tendencím a -ismům, který je charakteristický jistým distingovaným odstupem či nadhledem. Jak upozorňuje Vladimír Papoušek, Čapek „Zůstává skeptikem, přestože se na poli moderny pohybuje s velkou jistotou a znalostí. Toto pole pro něj ale bude už jednou provždy předmětem studijního zájmu, nikoliv však existenciální nezbytností.“ [18] Papoušek Čapkův postoj nehodnotí negativně, ale právě naopak. Vidí v něm jistou přednost projevující se schopností velmi přesně a nadčasově pojmenovat dobovou situaci individua.

Čapkova pozice ve vztahu k dobovému uměleckému dění je spíše pozicí komentátora než přímého účastníka. Můžeme říci, že je silně intelektuálně založeným umělcem včleňujícím do svého díla odkazy na aktuální umělecké, ale také filozofické a kulturní tendence. Tento postoj činí zařazení jeho tvorby k nějakému konkrétnímu modernistickému směru velice obtížným, můžeme mluvit spíše o citacích fragmentů z diskurzu avantgard.

Obtížná zařaditelnost Čapkovy tvorby je rovněž důsledkem neobyčejné šíře záběru jeho tvorby. Čapek se volně pohyboval mezi různými uměleckými směry a mezi literaturou pro intelektuální elitu a populárním proudem psaní. Obtíže s přiřazením k některému z uměleckých proudů se však projevují i v jednotlivých dílech. Například hra *R.U.R.* byla po svém uveřejnění, ve 20. letech minulého století, chápána jako součást avantgardní umělecké literatury, především expresionismu. Avšak už ve 30. letech se hra objevila v amerických školních čítankách jako typický příklad fantastiky v literatuře a stala se tak součástí standardní středoškolské literatury. [22] Ve 40. a 50. letech se zdál zahraničním vydavatelům Čapkův styl ve srovnání s právě vznikající literární tvorbou příliš odlehčený a povrchní. Jejich hodnocení však bylo do jisté míry ovlivněno ideologicky, neboť najednou nebylo jasné, na kterou stranu železné opony tohoto obhájce humanistických hodnot, z principu odmítajícího příklon k některé z ´pravd´, vlastně zařadit. Nakonec se v anglosaském prostředí ustálilo jednoduché pojmenování Čapka jako autora science fiction. [24] Čapek se stal dokonce centrální postavou historie tohoto žánru, jenž se právě v první polovině 20. století konstituoval a snažil se zformulovat svoji historii.



Obr. 1. Robot na divadelních ´prknech, která znamenají svět´. Ilustrace odstavce věnovaného hře R.U.R. a především robotům v eseji Josefa Čapka *Homo Artefactus* [3]

1.1 Robot jako kýč

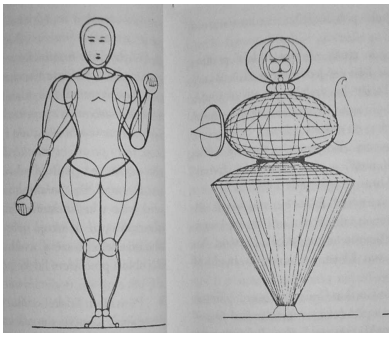
První scénické uvedení Čapkovy hry *R..U.R.* vyvolalo výjimečně silnou pozitivní reakci široké veřejnosti. Hra vzbudila velkou pozornost doma i v zahraničí. Svědčí o tom i rychle navazující překlady hry a premiéry inscenací *R.U.R.* ve světových metropolích jako New York (1922), Londýn (1923), Berlín (1923), Vídeň (1923), Paříž (1924) a v mnoha dalších. V domácích dobových periodikách se však setkáme také s celou řadou negativních reakcí. Čapek byl opakovaně kritizován za to, že se svezl na vlně dobové popularity tématu ´robotizace člověka´, žánru kolektivního dramatu, utopie a sociálních témat. Miroslav Rutte připomněl tyto hlasy ve svém článku z roku 1929, který psal u příležitosti uvedení *R.U.R.* v Divadle na Vinohradech: „Čapkovi bývalo často vytýkáno, že úspěch R.U.R. byl založen na vypočítavém zužitkování dobové poptávky“ Právě tato „dobová poptávka (…) byla pokládána za hlavního součinitele dramatického účinu R.U.R.“ [20]. Nejkritičtější hlasy se ozývaly z řad levicově zaměřených příslušníků avantgardních uměleckých hnutí. Motivy jejich negativního hodnocení můžeme hledat ve skutečnosti, že Čapek byl tzv. ´hradním spisovatelem´, přítelem prezidenta T. G. Masaryka, a ve svých postojích neprojevoval žádné nadšení pro radikální, revoluční řešení dobových sociálních problémů, pro nastolování nových společenských řádů, se kterými se ideál nového člověka v tvorbě avantgardních umělců bezprostředně pojil.

Čapkovi kritici považovali hru *R.U.R.* za neautentický umělecký projev, dílo, jehož autor se pouze svezl na vlně popularity určitých témat a napsal je záměrně jako populární hru na export. Svůj dojem kritici nejednou shrnuli označením **kýč**. Karel Teige dokonce v jedné novinové anketě údajně na dotaz, jakým slovem by v češtině nahradil germanismus kýč navrhoval RUR. [21] Arnošt Dvořák zase v recenzi premiéry hry napsal: „(…) skvělý kýč, který bude přeložen co nejdříve do němčiny,

snad i do angličtiny, a nastoupí triumfální cestu po všech dnešních divadlech Evropy (…) (kromě Ruska: tam mají např. Majakovského Mysterie Buff (…). Ale všude jinde mají kýče své oprávnění a svoji úlohu – jak by ji neměl kýč tak skvělý?“. [5]

Obr. 2. Ateliérové fotografie prvních robotů z premiéry R.U.R. v ND Praha (25.1.1921) .

Z hlediska radikálnosti politického postoje autora či z hlediska radikálnosti uměleckého gesta lze skutečně Čapkovu hru nahlížet jako krotké ztvárnění témat probojovávaných příslušníky uměleckých avantgard. Tito umělci tvořili strojové humanoidní figury geometrických tvarů, které byly mnohem nekompromisnějším ztvárněním tématu člověka-stroje než Čapkovi roboti a mnohem lépe odpovídali dobovému požadavku originality a invence. Navíc, tyto figury člověka-stroje byly nejen radikální uměleckou formou, ale pojily se často také s radikálními utopickými vizemi nových společenských řádů a s anti-měšťáckými postoji jejich tvůrců.

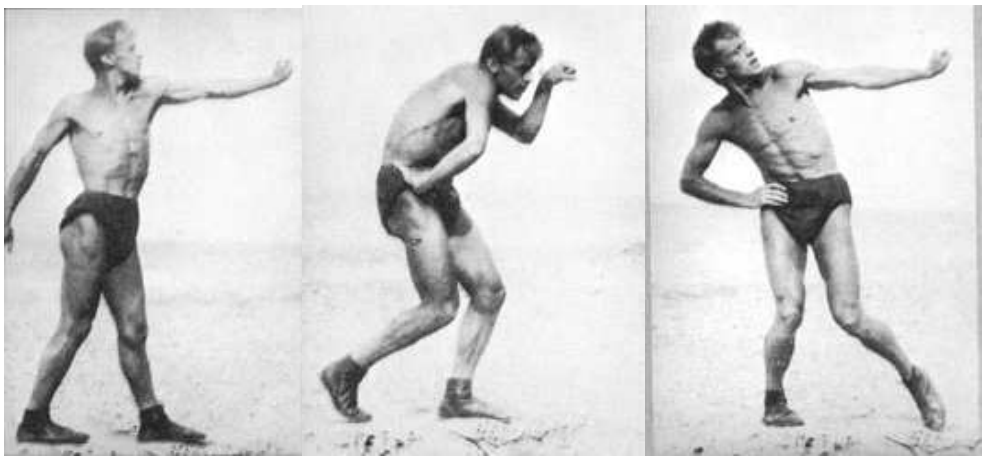


Obr. 3. Nákresy transformací lidského těla od Oskara Schlemmera (vlevo: Funkční zákony lidského těla – výsledek: kloubová panna; vpravo: Pohybové zákony lidského těla – výsledek: technický organismus.)

Čapkovi roboti, ve srovnání například s ´roboty´ Oskara Schlemmera na scéně Bauhausu, s Mejercholdovými herci podrobenými strojové organizaci svých pohybů

(herecká tréninková technika - Biomechanika), nebo s tanečníky v *Tancích strojů* Nikolaje Foreggera, jejichž choreografie byla inspirována typickými pohyby strojů a nástrojů, museli působit jako konzervativní, zkrocený obraz člověka-stroje, přizpůsobený vkusu širokého publika. Jakoby Čapek svými roboty skutečně parazitoval na jisté konvenci své doby: na anti-tradicionalistickém a inovačním étosu uměleckých hnutí. Jakoby přivlastněním si určitých prvků jejich jazyka chtěl pouze vnějškově dodat své hře moderní a avantgardní lesk.

Ohlas kritických postojů Čapkových současníků ke hře *R.U.R.* najdeme i v hodnocení Williama Harkinse, na dlouhou dobu autora jediné monografie o Karlu Čapkovi v angličtině, který se o hře celkově nevyjadřoval nijak lichotivě a o robotech napsal, že „Ve skutečnosti, pouze postava robota (a konečná proměna robotů v lidi) je expresionistická; jinak je Čapkovo zpracování konvenční a realistické“. [6]



Obr. 4. Ukázka tréninkových figur Mejercholdovy Biomechaniky, které měly vést herce k schopnosti ovládat své tělo jako stroj.

Můžeme se proto oprávněně ptát. Je *R.U.R.* skutečně kýč? Domníváme se, že z jistého hlediska je tento názor oprávněný. Podporuje jej i známá definice kýče formulovaná Milanem Kunderou v románu *Nesnesitelná lehkost bytí*. Kundera ztotožňuje kýč s bezpodmínečným souhlasem s bytím a nachází jej všude tam, kde autor pracuje s univerzálními tématy, která vyzývají ke kolektivnímu prožívání těchto témat:

„Kýč vyvolává těsně po sobě dvě slzy dojetí. První slza říká: jak je to krásné, děti běžící po trávníku! Druhá slza říká: jak je to krásné, být dojat s celým lidstvem nad dětmi běžícími po trávníku! Teprve ta druhá slza dělá z kýče kýč. Bratrství všech lidí světa bude možno založit jen na kýči.“ [16]

Kundera také upozorňuje, že neexistuje pouze jeden univerzální kýč, ale mnoho kýčů:

„Pramenem kýče je kategorický souhlas s bytím. Ale co je základem bytí? Bůh? Člověk? Boj? Láska? Muž? Žena? Názory jsou na to různé, a proto jsou i různé kýče:

katolický, protestantský, židovský, komunistický, fašistický, demokratický, feministický, evropský, americký, nacionální, internacionální.“ [16]

V intencích Kunderovy definice kýče bychom v případě *R.U.R.* mohli mluvit o kýči moderní doby. *R.U.R.* by bylo kýčem, který hlásá: Jak je krásné být moderním člověkem, členem moderní společnosti, hrdinou průmyslového věku ochotným položit na oltář technického pokroku i svůj život!

Takto formulované poselství Čapkovy hry odpovídá obecně přijaté interpretaci *R.U.R.*. Avšak Čapek neužíval motivy jako utopie, nový sociální a ekonomický řád založený na technickém pokroku (robotech) přímo, ale, jak jsme naznačili již v úvodu této studie, s kritickým odstupem a ironií. Když se do hry pozorněji začteme, zjistíme, že hra není utopickým dramatem, či tragédií moderního věku, ale využívá principy komických žánrů. Hra *R.U.R.* rozvíjí zápletku záměny lidí a robotů, ze které vzniká řada komických situací: Hned v předehře Helena Gloryová nerozezná továrníky od robotů a nabádá je k revoluci. Vrcholná scéna obléhání továrníků vzbouřenými roboty je groteskní parodií. Závěrečná scéna je vystavena tak, aby zmatené publikum aplaudovalo ´šťastnému konci´ lidstva na jehož místo nastoupili roboti.[2]

Až dnes jsme možná schopni ocenit tuto Čapkovu schopnost demaskovat dobové ideály, témata a formy jejich artikulace. *R.U.R.* je hrou, ve které autor subverzně pracuje s žánrem kolektivního dramatu, konceptem nového člověka, i s myšlenkou technického pokroku jako řešení sociálních problémů. Odhaluje limity a slabé stránky dobových ideologií a ideálů a činí je směšnými. Tento typicky postmoderní odstup od dobového dění, který autor uplatnil, znemožňuje hodnotit hru jako kýč. Neboť, jak píše Tomáš Kulka:

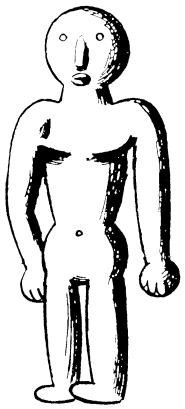
„Pracovat s postupy kýče není však to samé, jako vytvořit kýč. Kýč jako takový nic nezpochybňuje, žádné otázky neklade.“ A na další stránce dodává: „Kýč je vždy naprosto vážný, proto je směšný a patetický. Ironie je největším nepřítelem kýče.“ [15]

2 Remaking Robot

Zpětně můžeme hodnotit Čapkovu tvorbu jako projev praxe využívající citace a odkazy na jiná díla, případně vznikající přepisováním obecně známých příběhů (v případě *R.U.R.* jednoho z nejstarších příběhů o stvoření člověka), později považované za typicky postmoderní. Jde o uměleckou tvořivost rezignující na originalitu a jedinečnost díla ve prospěch posílení intertextových vazeb umožňujících mnohovrstevné čtení díla ve vztahu k ostatní kulturní produkci, umělecké i mimoumělecké. Tato praxe, označovaná jako projev naší „kultury užití“, dostala v posledních letech různé názvy jako remake, remix, parodie, pastiš, postprodukce.

V Čapkově dramatu *R.U.R.* jde tedy vlastně o **remake** mýtu o umělém člověku prostřednictvím jeho zasazení do reálií moderní doby. Sám Čapek tuto interpretaci připustil, když v odpovědi na otázku spolupracovníka listu Prager Tagblatt při příležitosti uvedení *R.U.R.* na jevišti Nového německého divadla v Praze o svých robotech řekl:

[2] Více k interpretaci hry R.U.R. v modu komických žánrů viz [7].



Obr. 5. Golem podle Josefa Čapka v umělecké eseji *Homo Artefactus* [3].

„R.U.R. je vlastně převodem pověsti o Golemovi v moderní formu. To mně ovšem napadlo teprve, když kus byl hotov. „U čerta, to je vlastně přece Golem, „ řekl jsem si, „Robot je Golem, provedený v masové tovární výrobě.“ [4]

Můžeme být dokonce ještě konkrétnější a vztáhnout Čapkovu hru o robotech ke golemovským legendám, které vznikaly v 16. století, mezi něž patří také legendy pražského okruhu, podle kterých stvořil Golema pražský renesanční židovský učenec, rabín Jehuda Löw ben Becalel (asi 1520 – 1609) v podobě oživlého hliněného sluhy, jenž měl chránit židovskou komunitu. Již v těchto legendách se příběh o napodobení stvoření člověka propojuje s představou Golema nikoli jako výrazu mystického zasvěcení, ale jako prakticky využitelného sluhy. V této variantě golemovské legendy můžeme hledat předobraz zápletky hry *R.U.R.* vrcholící ve vzpouře robotů, nad kterými jejich tvůrci ztratili kontrolu. Také Golem v pražské legendě se totiž vymkl kontrole svého tvůrce a způsobil destrukci a katastrofu.

Jazykem informatiky o tom píše Zdeněk Neubauer: „(…) nevypočitatelnost Golemova chování je vskutku nevypočitatelná, (…) nelze jí předejít zpřesňováním příkazů – „zdokonalováním programu“. Golem je cosi ne-dokonalého, nedotvořeného v samé své podstatě: jeho opakované aberace – vykolejování z řádu předpokladatelnosti a zdravého smyslu – byly vlastně varovnými znameními hrozícího nebezpečí.“ [17]

Legenda o Golemovi v této své podobě již neslouží k varování před nebezpečím lidského pokušení stvořit umělého člověka, ale odráží modernistickou stylizaci člověka jako neomezeného tvůrce světa, kterému se jeho výtvor vymkl z rukou. Můžeme proto spolu s Karlem Čapkem říci, že robot je vlastně Golem, ale stejně tak můžeme přijmout tvrzení Zdeňka Neubauera, že tento Golem je vlastně již robotem, ve smyslu ideálního sluhy, otroka či dělníka, stvořeného k tomu, aby vykonával těžkou práci a sloužil člověku.

Čapkovu hru o robotech můžeme interpretovat nejen na základě způsobu vzniku robot a jeho chování, jak jsme to učinili v souvislosti s golemovskou legendou. Intertextově je třeba interpretovat také vzhled robotů. Jejich strnulý postoj a mechanické pohyby poukazují k vizuálnímu jazyku uměleckých avantgard. Strnulé

postavy robotů však rozhodně nepůsobí ´nadčlověčensky´(alespoň v prvních dějstvích hry), ale spíše komicky, jako karikatura člověka věku stroje. Jakoby Čapek svými roboty komentoval vzhled postav kuželovitých a hranatých tvarů, zbavených všech individuálních rysů (viz např. obr. 3 a 4). Tito figury působí obvykle dynamicky, hrdinně, jako jakýsi abstraktní ideál či vzor - univerzální obraz člověka. Čapkovi Roboti jsou však výsledkem redukce člověka na zaměnitelnou pracovní sílu, o které snili ideologové průmyslové revoluce a o níž psali bratři Čapkové již v povídce *Systém* z roku 1909, ve které však nemluví o robotech, ale o ideálních dělnících nazvaných po svém tvůrci *Operarius utilis Ripratoni*, vzniklých specifickou kulturní převýchovou.[9] Ideální průmyslové revoluce připomíná později také Josef Šafařík ve studii *Člověk ve věku stroje* (1969): „Henry Ford vysvětloval, že se mu podařilo přivést své továrny k dokonalosti a optimální výkonnosti tím, že – jak se vyjádřil – vypudil z továrny všechno umění, čímž mínil, že nahradil živou bytost automatem. V témž smyslu hlásal Taylor, že ideálem průmyslové ekonomie je stroj, k jehož obsluze by stačila vycvičená opice nebo i člověk duševně méněcenný (…).“ [23]

Čapkově hře a významu robotů můžeme lépe porozumět, pokud ji budeme chápat jako komentář dobové imaginace, kterou můžeme vyčíst z uměleckých děl, ekonomických a sociologických úvah, ale také z produkce populární kultury - z reklam a ilustrací publikovaných v dobových periodikách. Josef Čapek tento vztah naznačil v eseji *Homo Artefactus* [3], ve kterém představil Čapkovy roboty vedle karikatur vyjadřujících imaginaci věku stroje, v nichž převládala oslava krásy strojů. Tyto satiricky převyprávěné dějiny umělých bytostí podobných člověku začínají od Golema a Homunkula, pokračují přes středověké rytíře v kovové zbroji a končí u veteránů první světové války, jejichž těla musela být opravena umělými protézami, a u moderních kyborgů, bytostí vzniklých vylepšením člověka strojovými protézami a implantáty, například fotoaparátem či hodinami. Jak sám autor doznává v úvodu díla, cílem této hravě a vtipně psané historie umělých bytostí bylo připomenout, že zejména futuristy proklamovaný ideál ´nového člověka´ s vlastnostmi stroje, není novou myšlenkou, výrazem vymanění moderního člověka ze sentimentálního vztahu k dějinám a kultuře minulosti, ale naopak je aktualizací jedné z nejstarších vizí provázejících lidstvo.

2.1 Robot jako klišé

Robot se stal nejdříve módní a brzy zcela přirozenou součástí mezinárodního jazyka. František Chudoba již v roce 1933 konstatoval, že slovo robot (včetně jeho odvozenin) se ujalo v angličtině a bylo zařazeno do oxfordského slovníku: „Zjistíme také, že Čapkův robot, jenž vnikl do angličtiny Selverovým překladem jeho známé hry R.U.R. až r. 1923, zapustil v ní mnohem hlubší kořeny (…), jak soudíme nejen z hojných literárních a novinářských dokladů až do r. 1930, kdy byla redakční historie jeho vývoje ve Velké Británii uzavřena, nýbrž ještě více z jeho četných odnoží, které vypučely teprve pod britským sluncem (robotize, robotizing, robotization, robotry, robotism, robotian, robotesque aj.) – zkrátka letmé nahlédnutí do oxfordských Doplňků nás poučí, jak Čapkův nápad v Anglii mnohem více rozrostl a rozkošatil nežli v jeho rodné zemi“. [10] Rozšíření slova robot ilustruje také zpráva ČTK, otištěná např.

v Lidových novinách dne 4. listopadu 1997, ve které se dočteme, že v Londýně vycházející Collins Dictionary zařadil slova ´robot´ a ´sametová revoluce´ mezi sto slov charakterizujících 20. století.

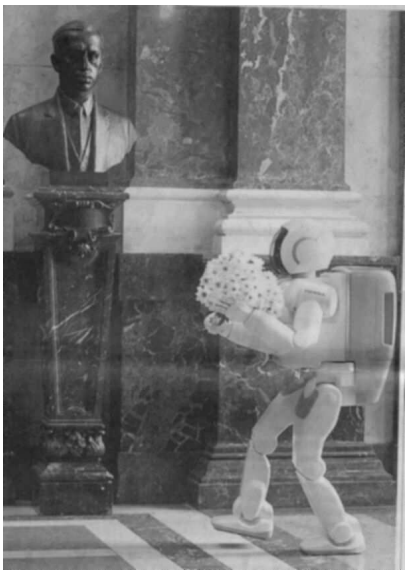
Význam slova robot se ve slovníkových heslech většinou dělí na dvě definice vztahující se k životné a neživotné variantě slova robot: první směřuje do řádu fikce, a druhá definuje robota jako prakticky využitelný stroj. V *Akademickém slovníku cizích slov* najdeme díky tomu dvě samostatná hesla robot. V prvním významu je robot fiktivní postavou, humanoidním automatem nebo těžce pracujícím člověkem: „automat podobající se člověku a konající lidskou práci (nazv. podle vymyšlených bytostí v dramatu K. Čapka R.U.R. (…); přeneseně člověk pracující namáhavě, bez odpočinku.“ [1] Ve druhém (neživotném) smyslu slova je robot definován jako univerzální přístroj (napojený na počítač), schopný autonomního jednání a dorozumění s člověkem: „univerzální přístroj pro domácnost“ nebo „zařízení ovládané počítačem, schopné vnímat a analyzovat vlastnosti prostředí, dorozumívat se s člověkem a samostatně do prostředí zasahovat.“ [1]

Toto nikoliv ojedinělé zdvojení hesla robot poukazuje k jeho dvojí historii: Jedna je součástí fikce a jedna žité reality. Avšak první roboti Karla Čapka se tomuto jednoduchému zdvojení vzpírají. Od svého autora, který se ve své tvorbě rád pohyboval mezi realitou a fikcí, dostali do vínku jistou ambivalenci. Jsou sice z organické látky jako živé organismy (z tohoto hlediska patří roboti v rámci fantastických bytostí mezi homunkuly), ale jsou také sestaveni na továrních výrobních linkách jako třeba Fordovy automobily (z této perspektivy jsou roboti moderní variantou humanoidních mechanismů zvaných androidy). Pro ještě větší komplikovanost nechá Čapek prostor pro to, aby se v závěru výrobního procesu organicky zakončili (srostli). Čapkovi roboti v sobě tedy své později rozdvojené heslovité výklady zcela přirozeně propojují a ještě rozšiřují. Původní, řekněme tedy praví, roboti jsou totiž metaforou vzniklou z imaginace a zkušenosti věku stroje. A jako metafora je robot neredukovatelný na svůj heslovitý výklad.

Na druhé straně je třeba konstatovat, že robot přešel velmi brzy do mezinárodního slovníku, nejdříve jako neologismus vyjadřující ducha věku stroje a později se stal zcela konvenční součástí slovníku populární kultury 20. a dnes již i 21. století. S robotem v jeho nejrůznějších podobách se můžeme setkat v oblasti vědecké fantastiky, ať už literární nebo filmové, a právě vědecko fantastická produkce ovlivnila všeobecnou představu robota zřejmě nejvíce. Dnes se s roboty běžně setkáme nejen na plátnech kin, na stránkách časopisů a knížek milovníků sci-fi, ale také v kuchyních a v dětských pokojích, stejně jako v továrních halách a v experimentálních laboratořích výzkumných center, ale také v galeriích nebo na jevištích divadel a koncertních sálů. Tato všudypřítomnost robota v naší kultuře v posledních devadesáti letech vedla Christophera Johnsona ke konstatování, že robot je **klišé** [11], tedy častým užíváním vyprázdněný pojem, který není příliš atraktivním předmětem zájmu humanitních vědců.

Příčiny tohoto postoje k robotovi v oblasti humanitních věd můžeme hledat ve všeobecně rozšířené představě, že všichni přece víme o robotech dost, vždyť jsme vyrostli v kultuře plné robotů. Robot se tak může jevit jako trochu archaická figura a symbol našich snů a tužeb, které spojujeme s dobou, na kterou rádi vzpomínáme,

dobou dospívání či nedospělosti, nepovažujeme je však za dostatečně seriózní předmět zájmu akademiků. Přitom právě pohled na robota z hlediska jeho různých kulturních remaků a jejich vzájemných vztahů může být velice zajímavý. Žádný „rozumův-univerzální-robot“ totiž neexistuje a ani existovat nemůže. Robot je metafora, se kterou můžeme vést dialog směřující k její interpretaci, jež vždy smysl díla částečně redukuje a částečně kontaminuje dalším smyslem, který přináší interpret. Pokusy o doslovné chápání Čapkových robotů, jako třeba čtení Čapkova dramatu jako návodu na sestavení robota, nejsou pouze redukcí původního smyslu, ale původní, metaforický význam robota se v těchto případech již zcela vytrácí. Roboty v oblasti experimentální robotiky, stejně jako jejich různá filmová, literární či jiná umělecká ztvárnění tak můžeme interpretovat jako více či méně zdařilé pokusy o výklad původní Čapkovy metafory. Z tohoto hlediska je zajímavé srovnávat roboty, jak je známe dnes a jak si je představujeme v budoucnosti, s původní metaforou robota. Odchylky od původního obrazu robota vypovídají mnohé také o vývoji vztahu naší společnosti ke strojům.



Obr. 6. Robot Asimo klade květy k bustě K. Čapka v Národním muzeu, Praha 2003.

3 Remaking robot in/to Myth

Z pohledu začátku 21. století již můžeme tvrdit, že Čapkův příběh o robotech se stal mýtem, který byl mnohokrát převypravován autory science fiction, vědci v robotických laboratořích se stále snaží robota podobného člověku zkonstruovat a slovo robot přešlo do obecného, mezinárodního jazyka. Dokonce, jak to předpovídal Jorge Luis Borges v případě Wellsových příběhů, žije slovo robot mimo jazyk, v němž vzniklo. O příběhu o robotech můžeme proto mluvit nejen jako o jednom z mýtů, ale dokonce možná o nejvlivnějším mýtu moderní doby. [14] Robot se stal součástí moderního mýtu o

technickém pokroku neohroženě překračujícím hranice přirozené, biologické humanity. V této své podobě se objevuje především v oblasti vědecké-fantastiky. Roboti však byli brzy přemístěni z řádu fikce také do hmotného světa, aby se stali nejdříve technickými kuriozitami sloužícími popularizaci mýtu technického pokroku, později však stále více také běžnou součástí naší žité reality a předmětem seriózního zájmu vědců a inženýrů.

Způsob, jakým se stal robot součástí moderního mýtu pokroku nám může pomoci odhalit metodologie a pozorování Rolanda Barthesa. [2] Podle něj mýtus často užívá **neologismy** odvozené z obecně známých slov a také využívá znakové systémy prvního řádu (slova, věty, obrazy) jako svůj zdrojový materiál, který transformuje do narativu mýtu, jenž funguje jako znakový systém druhého řádu. Význam zdrojového materiálu je přitom mýtem nutně deformován: na jedné straně redukován a na druhé straně nahrazen nebo integrován novým významem, který tento mýtus nese.

V souladu s Barthesovým popisem vzniku mýtu je **robot** neologismus vytvořený z českého slova robota, znamenajícího nucenou práci nebo těžkou práci. Etymologie slova robot je však bohatší. Najdeme v ní také význam sirotek (robě), čímž se k motivu člověka redukovaného na svoji práci přidává motiv umělé bytosti, bytosti nezrozené přirozeným způsobem Robot se tímto způsobem stává součástí skupiny umělých bytostí (od homunkulů a Golemů po umělé bytosti ve filmech jako Blade Runner, 1982, nebo Umělá inteligence: IA, 2001).

Přesně v souladu s Barthesovými slovy o vztahu mezi látkou mýtu a mýtem samotným, byl význam robota, jako ambivalentní metafory poukazující ke stavu humanity v době ´věku stroje´, moderním mýtem technického pokroku deformován ve prospěch promluvy tohoto mýtu. Robot jako ústřední figura mýtu technického pokroku získal podobu super-stroje. Avšak pod (po)vrchní vrstvou mýtu stále působí ostatní významy robota: Robot jako člověk redukovaný na motor tovární produkce (srovnej s filmem Ch. Chaplina *Moderní věk*, 1936), robot jako otrok a jako živý nástroj (viz Aristotelova vize ideální společnosti ve spisu *Politika*), robot jako živá figurína či panenka, o kterých píše Zikmund Freud ve svém článku *Das Unheimliche* z roku 1919. Tyto významy nejsou v protikladu s mýtem technického pokroku, ale mýtus technického pokroku do sebe tyto motivy spojené s fascinací, závratí, ale i s obavami organicky zahrnul a těží z nich.

Slovo robot se dokonce samo stalo základem řady dalších neologismů, které zmiňuje už František Chudoba. [10] Nejvlivnější z nich je neologismus **robotika**, který vytvořil ve čtyřicátých letech Isaac Asimov. Poprvé užil toto slovo v povídce *Hra na honěnou* (Runaround) v roce 1942, ve které také představil tři zákony robotiky. Asiumovovo pojetí příběhů o robotech, charakteristické mírumilovným soužitím robotů s lidmi, které zaručují robotí zákony vepsané přímo do jejich pozitronovým mozků, inspirovalo mnoho vědců a inženýrů pracujících v oblasti robotiky v jejich zájmu o zkonstruování humanoidního robota.

4 Remaking robot in/to Art

Dnešní roboti jsou ve svých různých podobách více méně domestikovanými stvořeními, krotkými nástroji, které slouží lidem v různých oblastech jejich činnosti. Představa robota jako zastaralé metafory, vyprázdněné svým opakovaným, automatickým, užíváním, představa robota jako klišé, vyplývá zřejmě právě z této skutečnosti. Robot je však rovněž velice efektivním uměleckým prostředkem, umožňujícím klást aktuální otázky o stavu humanity stejně jako o směřování technického pokroku, případně o prostoru mezi tělem a technologií.

Současní umělci pracující s roboty, nebo zajímající se o robotiku, nemohou ignorovat mytologickou, literární ani průmyslovou historii robota a jiných umělých forem života. Umělci pracující v oblasti robotického umění od 60. let dvacátého století využívají pokročilou robotickou technologii a poznatky oborů jako umělá inteligence a umělý život právě k tomu, aby zpochybňovali stereotypy naší imaginace, které obklopují koncept robota. Avšak stejně jako je obtížné, nebo spíše nemožné, formulovat univerzální definici robota, jsou problematické i snahy definovat **robotické umění**, disciplínu zahrnující velice různorodé a často do jiných uměleckých disciplín zasahující díla.

V *New Media Dictionary* vycházejícím na pokračování v časopise *Leonardo*, zaměřeném na dialog mezi uměním, vědou a technikou, najdeme definici robota platnou pro oblast umělecké tvorby:

„Robot – samočinné zařízení navržené k provádění úkolů, které normálně vykonává člověk. Termín robot je převzat z české hry, ve které je slovo robota (sic! pozn. jh) (´nucená práce´) užito k označení ´umělého dělníka´. Průmyslový robot nevypadá jako člověk; jeho role je čistě funkční. V současném umění jsou roboti většinou užívaní k ironickým komentářům o sociálním chování lidí. V roce 1964 vytvořil Nam June Paik dálkově ovládaného Robota K-456, který přistupoval k divákům a zdravil je. O rok později Edward Kienholz představil The Friendly Grey Computer – Star Gauge model 5, počítač sedící na židli, který odpovídá na otázky diváků ano, nebo ne. Roboti byli také užívaní k tomu, aby samočinně vytvářeli umělecká díla. V roce 1978 v Amsterdamu vystavil Harold Cohen malého robota, který byl schopen vytvořit velkoformátové abstraktní kresby na papíře umístěném na zemi. Robot měl ultrazvukový navigační systém a bylo do něj naprogramováno několik zákonů kompozice. V současné době vyrábí roboty řada umělců, např. Jennifer Hallová, Laura Kikauka, Chico MacMurtrie, Stelarc a Norman White.“ [19]

O definici robotického umění se pokusil také Edoardo Kac [12], který dospěl k závěru, že charakteristickým rysem robotického umění, odlišujícím je od ostatních uměleckých forem je upřednostňování chování před formou. Ve stejném roce se spolu s Marcelem.li Antunezem Rocou pokusil v manifestu *Robotic art* specifikovat robotické umění ještě přesněji. [13] Napsali:

„Roboti jsou nové věci v uměleckém světě. Robotické umění má předchůdce v dílech umělců jako Tinguely a Paik, ale konstituuje zcela novou uměleckou formu se svými vlastními pravidly, odlišnou od skulptury, videa, performance a dalších příbuzných uměleckých praktik. Prototypy najdeme v sekvenčních strojích, které donekonečna opakují své časové struktury. Pouze mikroprocesory umožňují komplexnější a zřetelné

chování pokaždé, ať už ve specifické, nebo náhodné formě. Mikroprocesory jsou v robotickém umění stejně důležité jako štětce, barva a plátno v malbě […].
Roboti jako žánr se nesnaží přeměnit v uzavřené a fixní formy. Pokud se objeví nová situace, která by je obsahovala nebo překonala, jsou schopni jako koncept zmizet. Roboti existují v kritickém bodu kreativní debaty a konceptuálního zkoumání, který manifestuje sebe sama v rozšiřujících se telematických a kybernetických doménách."

Roboti v umění rozšiřují úzce technicky zaměřenou definici robota užívanou ve vědě a v průmyslu. Pojí se s tvorbou tematizující sociální otázky stejně jako individuální, osobní otázky. Základní vlastností robotického uměleckého díla je, že jde o objekty působící v čase a prostoru, reagující na různé vnější podněty. Roboti v umění mohou mít formu autonomních agentů působících v reálném prostředí, biomorfních automatů, elektronických protéz spojených s živými organismy a telerobotů.

Ústředním problémem a současně tématem robotického umění je chování robotů. Umělci záměrně zamlžují, jestli je jednání jejich robotů autonomní, poloautonomní, responsivní, interaktivní, adaptivní, organické, telepresenční nebo ještě nějaké jiné. Důležité je také chování recipientů, které se významným způsobem podílí na celkové kvalitě uměleckého zážitku. Robotická umělecká díla se často mísí s různými technologiemi. Často také překračují kategorie objektů a je vhodnější interpretovat je spíše jako environmenty. [8]

4.1 Subverzní strategie robotického umění

Řekli jsme, že základní podmínkou robotického umění je hmotná přítomnost robotického systému mezi námi. [13] Tento předpoklad je sám o sobě polemikou s diskursem umění nových, digitálních médií, který preferuje software před hardwarem, čímž se tato tvorba včleňuje do širší tendence umění 20. století, která se vyznačuje směřováním k abstrakci a k dematerializaci uměleckého díla. V tomto duchu píše i Peter Weibel, jeden z významných umělců a teoretiků podílejících se na rozvíjení disciplíny umění nových médií, který v článku *The World as Interface* [25] popisuje historii médií jako trajektorii, která vede k naplnění snu o překročení hranice obrazovky počítače do virtuálního prostoru a s tím spojenému rozplynutí recipientova ´Já´ ve prospěch sdíleného ´kolektivního vědomí´.

Roboti v umění slouží často jako nástroj kritiky a konfrontace s dominantním techno-optimistickým étosem současné techno-kultury, napájeným modernistickým mýtem pokroku, jehož manifestací je i Weibelův argument. Můžeme dokonce konstatovat, že robotické umění je v tomto svém protipohybu jakýmsi undergroundem umění nových médií. Zatímco Peter Weibel prezentuje vývoj směřující k volnému pohybu uživatele v datovém prostoru, Mitchel Whitelaw [26] popisuje protikladný vývoj technologie směrem k robotickým systémům, které překročily pomyslnou hranici virtuálního světa za obrazovkou počítače a vstoupily do našeho fyzického a fyzikálního prostoru. Píše: „[Robotická umělecká díla] vyrazila ven z vnitřku okna, které představuje počítačová obrazovka a vědomě obývají fyzický prostor. (…) Umístěním svých děl do prostoru spolu s námi, raději než ´někde jinde´ virtuálního nebo simulovaného prostoru, tvoří umělci různé hmotné systémy: robotická stvoření,

technologické a biologické směsice, instalované robotické ´ekosystémy´ a ´komunity´. (…) V jádru zájmu těchto umělců je pocit ´bytí s námi´."

Jakoby se právě v tomto protipohybu robotického umění naplňovala vize zrušení rozhraní a vytvoření plně imersivních systémů jako např. technologie virtuální reality, která pohání diskurs umění nových, digitálních médií. Zatímco systémy virtuální reality zdvojují naše smysly a izolují nás od okolního světa, právě robotické umění poskytuje zážitek bezprostřední konfrontace se strojovými systémy – v tomto případě roboty. Jejich hmotná přítomnost v našem prostoru je často velice zneklidňující a vyvolává u diváků primitivní reakce strachu z jiného, neznámého. Roboti v umění tak otevírají otázky hranice mezi skutečným a umělým, realitou a fikcí, objektem a subjektem, živým a neživým.

Příbuzenský vztah mezi Čapkovými roboty a roboty v současném mediálním umění je třeba hledat v konceptuální rovině a nikoli v rovině technologického vývoje jejich formy. Hlavní tematické okruhy, které ustavil Čapek ve své hře, a které se stále pojí s robotem v umění jsou tři:

1. Robot jako kolektivní postava a robot jako ztělesnění kolektivní zkušenosti členů společnosti věku stroje. Robot reprezentuje bytost redukovanou na svůj pracovní výkon, na své místo/funkci ve výrobním systému. Je personifikací nebo jistým typem derivovaným z kolektivní zkušenosti příslušníků strojové či informační společnosti. (srov. [23] a [9])
2. Robot je stroj a současně sluha nebo otrok. Robot je také cizinec, žena nebo dítě. Robot mluví jménem všech těch, o kterých příslušníci dominantního diskursu mluví jako o objektech svých zájmů. S jejich touhou po autonomii a společenském uznání jejich autonomie se mohou ztotožnit všichni Jiní.(srov. [8])
3. Robot je figurou, kterou spojujeme s ohrožením lidstva a s revolucí. Potenciálně může odmítnout svoji roli či místo/funkci, které mu bylo přiděleno těmi, kteří mají moc a kontrolu. Toto ´odmítnutí´ můžeme někdy popisovat jako autonomní chování a jindy třeba jako chybu v programu, záleží na kontextu.

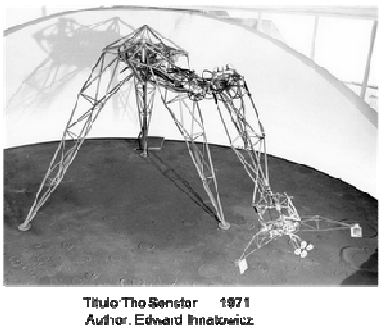
Obecně můžeme konstatovat., že robot v umění má apelativní funkci. Umělci v oblasti robotického umění tvrdí, že je pro ně mnohem důležitější reakce publika než robotický systém, který vytvořili. Zřejmě i proto roboti v umění často vstupují do veřejného prostoru. Jejich intervence do veřejného prostoru jsou současně intervencemi bytostí řádu fikce do prostoru naší žité reality.

Tito potomci Čapkových robotů slouží v umění jako média umělecké produkce, která odolávají mytizaci informačních technologií ve prospěch stále silného modernistického mýtu technického pokroku, s nímž se bezprostředně pojí robot v produkci cience-fiction. Jejich rezistence se projevuje, když umělci vědomě užívají robotické systémy k tomu, aby demaskovali, dekonstruovali či ironicky parafrázovali, tento moderní mýtus robota.

Obr. 7. Robot Miyata Jiro se svojí autorkou, umělkyní japonského původu Momoyo Torimitsu. New York 2003.

Další mocnou strategii současného robotického umění můžeme přirovnat k systému moderní poezie. Zatímco mýtus „míří k *ultra-signifikaci*, amplifikaci primárního systému, poezie se naopak snaží nalézt *infra-signifikaci*, predsémiologický stav řeči.“ [2]. Poezie zaujímá místo protikladné mýtu: „Proto se naše moderní poezie vždy stvrzuje jako vražda řeči, jako jakási smyslová, prostorová analogie ticha. Poezie stojí na pozici, která je protikladem mýtu: mýtus je sémiologický systém, který chce sama sebe překročit k systému faktickému; poezie je sémiologický systém, který se chce stáhnout do systému esenciálního.“ [2]

Moderní poezie a některé robotické performance sdílejí snahu zabít jazyk, materiál, ze kterého je stvořen mýtus. Příkladem tvorby vedené touto snahou jsou robotické performance Louise-Philippa Demerse a Billa Vorna, která Mitchel Whitelaw nazval robotickým divadlem afektu a Vorn o nich mluví jako o jistém typu post-industriální poezie“. [26] Robotické umění, podobně jako moderní poezie, v těchto případech překračuje hranice jazyka do prostoru, který můžeme pojmenovat jako prostorovou a hmotnou analogii ticha, směřující od znaků nikoli ke smyslu, ale k věcem samým.



Obr. 8. Zoomorfní robot Senster Edwarda Ihnatowicze z roku 1971 s plachými reakcemi na chování návštěvníků.

5 Závěr

Roboti v umění již neznamenají to stejné co na začátku 20. století, když Karel Čapek naspal svoji hru *R.U.R.*. Jejich funkce v umění je však stále táž: Koncept robota stále

umožňuje artikulovat sociální, politické a obecně lidské problémy. Pokusili jsme se zde i v jiných textech ukázat, že robot v umění, od doby svého objevení se ve hře *R.U.R.*, působí jako kritický, satirický a subverzní prostředek uměleckého výrazu, snažící se vyrovnávat dominantní tendence působící v naší společnosti, ve které stroje (tentokrát zejména informační a komunikační technologie) hrají stále větší úlohu. Robotické umění svou hmotnou přítomností mezi námi je antitezí k tendenci nových, digitálních médií a informačních a komunikačních technologií stávat se pro nás neviditelnými.

Nová, digitální média nejsou šťastným koncem historie médií. Nejsou lékem na všechny problémy, které nastolila starší (tzv. masová) média. Nová, digitální média jsou podmínkou a součástí naší post-spektakulární společnosti, ovlivňující především na naši vizuální zkušenost a vyžadující od nás recipientů spíše intelektuální porozumění dílu (např. intertextové čtení a interpretace, porozumění jazyku digitálních médií atd.). Robotické umění naopak pracuje především s našimi instinkty a emocemi, poukazuje k limitům naší racionality. Zatímco umění nových, digitálních médií je většinou zaměřené na vidění, intelekt a kontrolu, robotické umění je většinou založeno na ne zcela kontrolovaných či nekontrolovaných performancích robotů nebo strojových systémů v emoce vyvolávajících situacích. Roboti v umění jsou proto často užíváni jako média dialogu s antropocentrických viděním světa. Ve svých různých formách působí a jednají roboti směrem k destabilizaci našich představ o světě a k zpochybnění naší sebe-jistoty v užívání těchto konceptů. Ambivalence robota, této bytosti mezi skutečností a fikcí, mezi biologickou entitou a strojem, funguje jako rozbuška, která ničí mřížku pojmů a přesvědčení, skrze které nahlížíme svět, a tím relativizuje také naše navyklé způsoby zakoušení světa.

Můžeme konstatovat, že robot v umění nových médií působí ve službách výrazu nedůvěry k nekritickému přijímání nových mediálních technologií v sociálních a kulturních uspořádáních naší současné společnosti. Roboti v umění jsou prostředky analýzy nevědomých a neviditelných trajektorií moci a imaginace, které animují naši společnost.

Poděkování: Tato kapitola vznikla s podporou grantové agentury ČR v rámci grantové úlohy Centra základního výzkumu AMU a MU, „Výzkum funkcí techniky při vzniku a provozování múzického díla“, LC544.

Literatura

[1] Akademi*cký slovník cizích slov*, Praha: Academia, 2000.

[2] Barthes, R. Mýtus dnes, in Barthes, R.: *Mytologie*, Praha: Dokořán, 2004.

[3] Čapek, J. *Ledacos. Umělý Člověk,* Praha: Dauphin, 1997.

[4] Čapek, K. *R.U.R.*, Prager Tagblatt, 23.9. 1935.

[5] Dvořák, A. *O skvělém kýči*, Rudé právo, 27.1.1921.

[6] Harkins, W. E. *Karel Čapek,* New York – London: Columbia University Press, 1962.

[7] Horáková, J. *Robot jako robot.* Praha: KLP, 2010.

[8] Horáková, J. – Demers, L.- P.: Anthropocentrism and the Staging of Robots, in Adams, R. – Gibson, S. – Müller, A. S. (eds.): *Transdisciplinary Digital Art. Sound, Vision and the New Screen*, Berlin- Heidelberg: Springer, 2008, p. 434 – 450.

[9] Horáková, J. – Kelemen, J.: The Robot Story: Why Robots Were Born and How They Grew Up, in. Husbands, P. – Holland, O. – Wheeler, M. (eds.): *The Mechanical Mind in History*, Cambridge (MA): MIT Press, 2008, s. 555 – 560.

[10] Chudoba, F. *Národní dílo,* Lidové noviny, 15. a 16. 12. 1933.

[11] Johnson, Ch. I-You-We, Robot. Bradley, A. – Arman, L.: *Technicity*, Praha: Literaria Pragensia, 2006, p. 101 – 128.

[12] Kac, E.: Orogin and Development of Robotic Art, in *Art Journal* 56, 1997, č. 3 Drusker, J.(ed.): Digital reflections: The Dialogue of art and technology. Special Issue on Electronic Art, New York: CAA, p. 60 – 67. Dostupné na adrese: http://www.ekac.org./roboticart.html (rev. 27.1.2011)

[13] Kac, E. – Roca, M. A.: Robotic Art, *Leonardo Electronic Almanac* 5, 1995, č. 5. Dostupné na adrese: http://www.ekac.org/kacmarcelli.html (rev. 27.1.2011)

[14] Klíma, I. *Moderní mýtus,* Čapek, K. R.U.R. (ed. M. Halík), Praha: Československý spisovatel, s. 185 – 200.

[15] Kulka, T.: *Umění a kýč,* Praha: Torst, 2000.

[16] Kundera, M. *Nesnesitelná lehkost bytí*, Brno: Atlantis, 2006.

[17] Neubauer, Z. Golem a další příběhy o kabale, symbolech a podivuhodných setkáních, Praha: Malvern, 2002.

[18] Papoušek, V. *Gravitace avantgard. Imaginace a řeč avantgard v českých literárních textech první poloviny dvacátého století*, Praha: Akropolis, 2007.

[19] Poissant, L. *New Media Dictionary*. Dictionary Terms, Part VIII, Multimedia Part II, in: Leonardo 36, 2003, č. 2, s. 151 – 156.

[20] Rutte, M. *Královské divadlo na Vinohradech*, Národní listy, 16. 1.1929, č. 314.

[21] Seifert, J. *Všecky krásy věta*, Praha: Československý spisovatel, 1982.

[22] Scholes, R. – Rabkin, E.S. *Science-Fiction-History, Science, Vision,* New York: Oxford University Press, 1977.

[23] Šafařík, J. *Člověk ve věku stroje*, Brno: Atlantis, 1991.

[24] Wechsler, R. Karel Čapek in America, in: Makin, M. – Toman, J.: *On Karel Čapek : A Michigan Slavic Symposium* (Michigan Slavic Materials 34), Ann Arbor (MI): UMI Research Press, 1992 , pp. 109 – 125.

[25] Weibel, P. The World as Interface: Toward the Construction of Context-Controlled Event-Worlds. Druckrey, T. (ed.) *Electronic Culture: Technology and visual representation.* New York: Aperture, 1996.

[26] Whitelaw, M. Metacreation: *Art and Artificial Life,* Cambridge (MA): MIT Press, 2004.

Fuzzy generatívne systémy

Alica KELEMENOVÁ[1]

Abstrakt. Cieľom kapitoly je poskytnúť prehľad fuzzy modelov generatívnych systémov a formulovať aktuálne problémy výskumu v tejto oblasti. Zameriame sa na štandardné fuzzy gramatiky aj na fuzzy L systémy. Tieto budú východiskom k úvahám o rôznych typoch prejavov neurčitosti v systémoch spolupracujúcich gramatík a k formalizácii fuzzy gramatických systémov. Budeme skúmať prípady, keď komponentmi gramatických systémov sú fuzzy gramatiky ako aj prípady, kedy stupeň neurčitosti je priradený jednotlivým komponentom gramatického systému.

1 Jazyky a fuzzy jazyky

Známa práca L. A. Zadeha o fuzzy množinách z roku 1965 položila základy dnes významnej teórie, ktorá zasiahla prakticky všetky matematické štruktúry a viedla ku skúmaniu ich fuzzy variant.

V tejto kapitole sa budeme venovať štúdiu formálnych jazykov a ich fuzzy modifikácií, ktoré sú predmetom štúdia od roku 1969 [9]. Formálny jazyk je akákoľvek množina slov, teda reťazcov symbolov z danej abecedy. Formálne jazyky, tak ako boli pôvodne zavedené obsahujú jednoznačne určenú množinu slov. Vieme, ktoré reťazce do jazyka patria a ktoré sú z neho vylúčené. Naproti tomu sa v prirodzených jazykoch stretáme často s istým stupňom vágnosti, nepresnosti, neurčitosti. Nemusí byť celkom jednoznačné, či to, čo vidíme, je strom alebo krík, či je kamarát vysoký alebo nízky. K tomu, aby sme zachytili naznačené rozdielnosti medzi nejasne určenými objektmi patriacimi do jazyka, môže byť vhodné zaviesť stupeň neurčitosti, nepresnosti, vágnosti slov (reťazcov symbolov) a definovať nepresný jazyk - ***fuzzy jazyk*** a tiež rozšíriť štúdium klasickej teórie formálnych jazykov na štúdium teórie fuzzy jazykov ako množiny slov (reťazcov) s priradeným stupňom vágnosti.

Formálne jazyky zvyčajne určujeme gramatikami, automatmi, resp. strojmi alebo algebraickými výrazmi. Je preto prirodzené definovať fuzzy jazyky na báze vágnej varianty tých prostriedkov, ktoré ich vytvárajú, teda za pomoci ***fuzzy gramatík***, ***fuzzy automatov*** resp. ***fuzzy strojov*** a ***fuzzy algebraických výrazov***.

V tejto kapitole sa sústredíme na vytváranie fuzzy jazykov na báze rôzny typov fuzzy gramatík a ich systémov. Čitateľovi, ktorý sa zaujíma o fuzzy automaty

[1] Ústav informatiky, Slezská univerzita, Opava, e-mail: kelemenova@fpf.slu.cz

odporúčame do pozornosti knihy dvojice autorov J. N. Mordeson a D. S. Malik [10], [12] a tiež monografiu W. Wechlera [15]. Algebraická teória fuzzy jazykov je prezentovaná v rade článkov P. R. J. Asvelda (napr. [1]).

Pre množinu A je fuzzy množina FA určená funkciou príslušnosti μ_A, ktorá priraďuje prvkom množiny A hodnoty reálnych čísel z uzavretého intervalu [0,1]. Hodnota $\mu_A(x)$ reprezentuje stupeň príslušnosti prvku x do A. My sa budeme primárne zaoberať jazykmi nad abecedou Σ a fuzzy jazykmi. Pre jazyk $L \subseteq \Sigma^*$ je fuzzy jazyk FL určený nasledovne

$$FL = \{(w, \mu_L(w) \mid w \in L, \mu_L(w) \in [0,1])\}.$$

Ako príklad môžeme uviesť fuzzy jazyk nad dvojprvkovou abecedou $\Sigma = \{0,1\}$

$$FL = \{(a^m b^n, \mu_L(a^m b^n)) \mid m,n \geq 1\} \cup \{(w,0) \mid w \neq a^m b^n, m,n \geq 1\},$$

kde $\mu_L(a^m b^n) = \min(m,n).\max(m,n)^{-1}$.

Fuzzy jazyk FL a $\lambda \in [0,1]$ určujú nasledovné fuzzy jazyky s prahovou hodnotou λ, respektíve λ_1, λ_2.

$$\mathrm{L}(\mu, \lambda) = \{w \mid (w, \mu(w)) \in FL, \mu(w) > \lambda\}$$
$$\mathrm{L}(\mu, \lambda_{\geq}) = \{w \mid (w, \mu(w)) \in FL, \mu(w) \geq \lambda\}$$
$$\mathrm{L}(\mu, \lambda_{=}) = \{w \mid (w, \mu(w)) \in FL, \mu(w) = \lambda\}$$
$$\mathrm{L}(\mu, \lambda_1, \lambda_2) = \{w \mid (w, \mu(w)) \in FL, \lambda_1 \leq \mu(w) \leq \lambda_2\}$$

V nasledujúcej časti uvedieme fuzzy gramatiky a nimi určené fuzzy jazyky. Ďalej sa zameriame na paralelne prepisujúce gramatiky, konkrétne na L systémy a fuzzy L systémy. Formálne jazyky môžu byť vytvorené aj ďalšími typmi gramatík, ako napríklad gramatikami s riadeným odvodením alebo tiež bohatšími gramatickými štruktúrami, napríklad gramatickými systémami, ekogramatickými systémami a kolóniami. Záverečné podkapitoly sa budú venovať možnostiam ako zaviesť fuzzy varianty týchto gramatických štruktúr a pomocou nich definovaných fuzzy jazykov.

2 Gramatiky a fuzzy gramatiky

Azda najznámejším a najčastejšie využívaným modelom formálnych gramatík sú klasické, sekvenčne pracujúce gramatiky Chomského hierarchie. Jedná sa o štruktúru, ktorá umožňuje vygenerovať reťazce z pomocného štartovacieho symbolu. Generovanie prebieha na základe pravidiel gramatiky postupným nahradzovaním tej časti vytvoreného reťazca obsahujúceho pomocné symboly, ktorá tvorí ľavú časť pravidla pravou časťou pravidla. Takto z reťazca tvaru $u\alpha v$ pomocou pravidla $\alpha \to \beta$ vytvoríme reťazec $u\beta v$, čo skrátene zapisujeme ako $u\alpha v \Rightarrow u\beta v$. Cieľom takýchto odvodení je vytvoriť všetky slová, ktoré už neobsahujú žiaden pomocný symbol. Táto množina slov určuje jazyk vytvorený (generovaný) gramatikou.

Formálne je gramatika určená štvoricou $G = (N, T, P, S)$, kde N je množina pomocných symbolov, T je množina koncových symbolov, S je štartovací pomocný

symbol a P je konečná množina pravidiel tvaru $\alpha \to \beta$, určujúca, že podreťazec α obsahujúci aspoň jeden pomocný symbol je možné nahradiť reťazcom β. Množina reťazcov terminálnych (koncových) symbolov, ktorú získame všetkými možnými odvodeniami určuje jazyk vytvorený gramatikou.

Ako užitočný ilustračný príklad poslúži gramatika $G_z = (N_z, T_z, P_z, S)$, vytvárajúca správne utvorené reťazce zátvoriek obsahujúce tiež písmená a. V tomto príklade gramatiky budeme mať len jeden pomocný symbol, S, $N_z = \{S\}$, tri koncové symboly: „a“, otváraciu zátvorku „(“ a zatváraciu zátvorku „)“, $T_z = \{a,(,)\}$ a pravidlá:

- $S \to (S)S$,
- $S \to (S)$,
- $S \to a$,

patriace do P_z. Ukážkou generovania reťazca v gramatike G_z je postupnosť:

$$S \Rightarrow (S) \Rightarrow ((S)S) \Rightarrow ((S)(S)S) \Rightarrow ((S)((S))S) \Rightarrow$$
$$\Rightarrow ((S)((S))a) \Rightarrow ((S)((a))a) \Rightarrow ((a)((a))a)$$

V prvom kroku odvodenia sme použili druhé pravidlo, v ďalšom prvé pravidlo a v nasledujúcom znovu prvé pravidlo na druhý výskyt písmena (neterminálu) S. Pokračovali sme druhým pravidlom aplikovaným na druhý výskyt písmena S. Slovo generované gramatikou sme získali tak, že sme v posledných troch krokoch odvodenia použili tretie pravidlo vždy na posledný výskyt písmena S.

Čitateľovi odporúčame do pozornosti publikácie [11, 4], kde sa môže oboznámiť so základmi klasickej teórie formálnych jazykov a Handbook [14] predstavujúci túto teóriu v celej jej šírke a pestrosti.

2.1 Fuzzy gramatiky

Pri zovšeobecnení gramatiky na fuzzy gramatiku postupujeme tak, že priradíme pravidlám gramatiky mieru μ. Táto miera bude prostredníctvom odvodenia slov v gramatikách určovať mieru príslušnosti odvodeného slova do jazyka. V súlade so zvyklosťami, symbol V^* v nasledujúcej definícii označuje množinu všetkých reťazcov konečnej dĺžky vytvorených z prvkov množiny V.

Definícia: Fuzzy gramatika G je štvorica $G = (N, T, P, S)$, kde

- N je konečná, neprázdna, neterminálna abeceda,
- T je konečná, neprázdna, terminálna abeceda taká, že $N \cap T = \{\ \}$,
- P je konečná, neprázdna množina fuzzy pravidiel tvaru $\alpha \overset{c}{\to} \beta$, kde $\alpha \in (N \cup T)^* N (N \cup T)^*$, $\beta \in (N \cup T)^*$ a $c \in [0,1]$,

- $S \in N$ je štartovací symbol gramatiky.

Zavedieme funkciu μ, ktorá priradí klasickému pravidlu $\alpha \to \beta$ hodnotu c podľa predpisu:

$\mu(\alpha \to \beta) = c$ práve vtedy, keď $\alpha \overset{c}{\to} \beta$ patrí do P.

Krok odvodenia vo fuzzy gramatike sa oproti kroku odvodeniu v gramatike bude líšiť tým, že mu priradíme hodnotu určenú použitým pravidlom.

Definícia: Nech $G = (N, T, P, S)$, $\alpha \overset{c}{\to} \beta \in P$ a $x, y \in (N \cup T)^*$, potom zápis $x\alpha y \overset{c}{\Rightarrow} x\beta y$ určuje ***krok odvodenia*** fuzzy gramatiky G. Hovoríme, že reťazec $x\beta y$ je priamo odvodený z $x\alpha y$ s mierou príslušnosti $c = \mu(\alpha \to \beta)$.

Odvodenie slova w vo fuzzy gramatike zodpovedá odvodeniu slova v klasickej gramatike, pričom celému odvodeniu je pripísaná hodnota zodpovedajúca najslabšiemu miestu v odvodení, teda minimu z hodnôt pravidiel použitých v tomto odvodení.

Definícia: ***Fuzzy odvodenie*** $S \overset{c_1}{\Rightarrow} \alpha_1 \overset{c_1}{\Rightarrow} \alpha_2 \ldots \alpha_{m-1} \overset{c_m}{\Rightarrow} \alpha_m = w$ slova w vo fuzzy gramatike G budeme skrátene zapisovať ako $S \overset{c}{\Rightarrow}^* w$, kde hodnota c priradená tomuto odvodeniu slova w je daná predpisom $c = \min(c_1, \ldots, c_m)$.

V danej gramatike môžeme mať niekoľko rôznych odvodení toho istého slova w zo štartovacieho symbolu S a tiež je možné, že niektoré slová v gramatike vôbec neodvodíme.

Definícia: Mieru príslušnosti slova w generovaného fuzzy gramatikou G určíme nasledovne:

$\mu_G(w) = 0$ pre každé $w \notin L(G)$,

$\mu_G(w) = \max\{c : S \overset{c}{\Rightarrow}^* w,$ pre všetky možné odvodenia slova w v $G\}$.

Zápisom $S \overset{c}{\Rightarrow}_G^* w$ budeme označovať skutočnosť, že w je možné odvodiť v gramatike G s mierou príslušnosti $c = \mu_G(w)$.

Všimnite si rozdiel v zápisoch $S \overset{c}{\Rightarrow}^* w$ a $S \overset{c}{\Rightarrow}_G^* w$. Zápis $S \overset{c}{\Rightarrow}^* w$ je zápisom konkrétneho odvodenia z jemu zodpovedajúcou hodnotou c a zápis $S \overset{c}{\Rightarrow}_G^* w$ charakterizuje odvodenie slova w vo fuzzy gramatike G tak, že odvodenému slov w priradí mieru príslušnosti do jazyka na základe odvodenia s najvyšším ohodnotením.

Definícia: ***Fuzzy jazyk*** $L_F(G)$ generovaný fuzzy gramatikou G, je množina: $L_F(G) = \{(w, c) \mid w \in T^*, S \overset{c}{\Rightarrow}_G^* w\}$.

Fuzzy gramatiky umožňujú definovať aj klasické (ostré) jazyky určené prahom λ tak, ako sme to uviedli v predchádzajúcej podkapitole.

Príklad: Majme fuzzy gramatiku $G = (N, T, P, S)$, kde $N = \{S, A, B\}$, $T = \{a, b\}$ s pravidlami:

$$S \overset{0,5}{\rightarrow} A \qquad A \overset{0,4}{\rightarrow} a \qquad B \overset{0,9}{\rightarrow} a$$

$$S \overset{0,7}{\rightarrow} B \qquad A \overset{0,9}{\rightarrow} b$$

$$S \overset{0,3}{\rightarrow} BA \qquad A \overset{0,6}{\rightarrow} B \qquad B \overset{0,2}{\rightarrow} b$$

Pomocou tejto fuzzy gramatiky odvodíme len slová a, b, aa, bb, ab, ba. Výpočet miery príslušnosti budeme ilustrovať na niektorých z týchto slov. Slovo a je možné v G odvodiť troma rôznymi spôsobmi:

$S \overset{0,5}{\Rightarrow} A \overset{0,4}{\Rightarrow} a$ skrátene $S \overset{0,4}{\Rightarrow}{}^{*} a$, kde $0,4 = \min\{0,5;0,4\}$

$S \overset{0,5}{\Rightarrow} A \overset{0,6}{\Rightarrow} B \overset{0,9}{\Rightarrow} a$ skrátene $S \overset{0,5}{\Rightarrow}{}^{*} a$, kde $0,5 = \min\{0,5;0,6;0,9\}$

$S \overset{0,7}{\Rightarrow} B \overset{0,9}{\Rightarrow} a$ skrátene $S \overset{0,7}{\Rightarrow}{}^{*} a$, kde $0,7 = \min\{0,7;0,9\}$

Miera príslušnosti slova a generovaného gramatikou G je:

$$\mu_G(a) = \max\{0,4;0,5;0,7\} = 0,7$$

Slovo ab má v gramatike G nasledujúce odvodenia:

$S \overset{0.3}{\Rightarrow} BA \overset{0.9}{\Rightarrow} aA \overset{0.6}{\Rightarrow} aB \overset{0.2}{\Rightarrow} ab$ skrátene $S \overset{0.2}{\Rightarrow}{}^{*} ab$

$S \overset{0,3}{\Rightarrow} BA \overset{0,6}{\Rightarrow} BB \overset{0,9}{\Rightarrow} aB \overset{0,2}{\Rightarrow} ab$ skrátene $S \overset{0.2}{\Rightarrow}{}^{*} ab$

$S \overset{0,3}{\Rightarrow} BA \overset{0,9}{\Rightarrow} Bb \overset{0,9}{\Rightarrow} ab$ skrátene $S \overset{0.3}{\Rightarrow}{}^{*} ab$

$S \overset{0,3}{\Rightarrow} BA \overset{0,9}{\Rightarrow} aA \overset{0,9}{\Rightarrow} ab$ skrátene $S \overset{0.3}{\Rightarrow}{}^{*} ab$

Miera príslušnosti slova ab generovaného gramatikou G je $\mu_G(ab) = 0,3$. Fuzzy jazyk generovaný touto gramatikou G je jazyk:

$$L_F(G) = \{(a;0,7),(b;0,5),(aa;0,3),(bb;0,2),(ab;0,3),(ba;0,2)\}$$

Jazyk určený fuzzy gramatikou G s prahovou hodnotou $\lambda = 0,2$ je jazyk:

$$L(G;0,2) = \{a, b, aa, ab\}$$

2.2 Max-prod gramatiky

Ďalšie typy fuzzy gramatík sa líšia od gramatík predstavených v predchádzajúcej časti spôsobom výpočtu miery príslušnosti generovaných slov. Predstavíme max-prod gramatiky, ktoré sa od fuzzy gramatík líšia tromi faktormi a to ohodnotením pravidiel, štartovacím symbolom a mierou príslušnosti vygenerovaných slov. Konkrétne sa jedná sa o nasledujúce odlišnosti.

Pravidlá v max-prod gramatikách sú ohodnotené nezápornými reálnymi číslami, prvkami z $R^{\geq 0}$.

Fuzzy gramatiky z predchádzajúcej časti kapitoly mali vždy jednoznačne určený štartovací symbol. Štartovacím symbolom max-prod gramatiky môže byť so stanovenou mierou každý neterminálny symbol gramatiky.

Mieru príslušnosti generovaných slov určuje v prípade max-prod gramatík súčin hodnôt jednotlivých pravidiel použitých v odvodení vynásobený hodnotou priradenou neterminálnemu symbolu, z ktorého odvodenie začína. V ďalšom budeme pod pojmom gramatika rozumieť ***bezkontextovú*** gramatiku, teda gramatiku, ktorej pravidlá umožňujú prepísať len jeden neterminálny symbol.

Definícia: ***Max-prod*** gramatika G je štvorica $G = (N, T, P, \eta)$, kde

- N a T sú konečné, neprázdne množiny neterminálnych a terminálnych symbolov takých, že $N \cap T = \{\ \}$,
- P je konečná, neprázdna množina fuzzy pravidiel tvaru $A \overset{p}{\rightarrow} \alpha$, kde $A \in N$, $\alpha \in (T \cup N)^*$ a p je reálne, nezáporné číslo,
- $\eta : N \rightarrow R^{\geq 0}$ je funkcia, ktorá priradí neterminálnemu symbolu A reálnu nezápornú hodnotu $\eta(A)$, stupeň príslušnosti byť štartovacím symbolom.

Max-prod gramatika je striktná ak $p \in [0,1]$ pre všetky jej fuzzy pravidlá $A \overset{p}{\rightarrow} \alpha$ a $\eta : N \rightarrow [0,1]$.

Označme symbolom D množinu všetkých usporiadaných dvojíc (r, k), kde r je označenie pravidla a k prirodzené číslo. Konečná postupnosť týchto dvojíc, prvok z D^*, určuje odvodenie.

Definícia: Nech G je max-prod gramatika, $\alpha, \beta \in (N \cup T)^*$ a $d = (r_1,k_1)(r_2,k_2)...(r_n,k_n) \in D^*$. Zápis $\alpha \overset{p}{\Rightarrow} \beta \pmod d$ určuje odvodenie v gramatike, pre ktoré platí:

- $\mathrm{r_i} = (\mathrm{A_i} \overset{p_i}{\rightarrow} \alpha_i) \in P$ je pravidlo použité v i-tom kroku na k_i-ty výskyt písmena A_i, pre $i = 1,2,\ldots,n$ a
- $\mathrm{p} = \mathrm{p_1} \cdot \mathrm{p_2} \cdot \ldots \cdot \mathrm{p_n}$.

Zápis $\alpha \overset{0}{\Rightarrow} \beta \pmod d$ použijeme práve vtedy, keď β nie je možné odvodiť z α pomocou d.

V danej gramatike môžeme mať pre dané $\alpha, \beta \in (N \cup T)^*$ niekoľko (prípadne aj nekonečne veľa) rôznych odvodení d, s rôznymi priradenými hodnotami p. Položíme $\mu_G(\alpha, \beta) = \sup\{\mathrm{p} \mid \alpha \overset{p}{\Rightarrow} \beta \pmod d$ pre všetky možné odvodenia $d \in D^*\}$. Do oboru hodnôt tejto funkcie patrí okrem nezáporných reálnych čísel aj znak ∞, budeme ho označovať $\mathrm{R}_\infty^{\geq 0}$, $\mathrm{R}_\infty^{\geq 0} = R^{\geq 0} \cup \{\infty\}$.

Definícia: Nech $G = (N, T, P, \eta)$ je max-prod gramatika. Potom funkcia príslušnosti terminálnych slov $\mu_G : \mathrm{T}^* \rightarrow \mathrm{R}_\infty^{\geq 0}$ je daná predpisom:

$$\mu_G(w) = \sup\{\eta(A) \cdot \mu_G(A,w) \mid A \in N\}.$$

Fuzzy jazyk generovaný max-prod gramatikou G je množina slov

$$L_{MP}(G) = \{(w,c) | w \in T^*,\ c = \mu_G(w)\}.$$

Fuzzy jazyk s konečným ohodnotením je jazyk, pre ktorý platí $\mu_G(w) \in R^{\geq 0}$, to znamená, že príslušnosť každého slova do jazyka je daná konečnou hodnotou.

Príklad: Majme max-prod gramatiku $G = (N, T, P, \eta)$, kde $N = \{S, A, B, C, D\}$ a $T = \{a, b\}$ s nasledujúcimi fuzzy pravidlami:

$$r_1 = (S \overset{9}{\rightarrow} bAC) \qquad r_2 = (A \overset{8}{\rightarrow} B) \qquad r_3 = (A \overset{7}{\rightarrow} b)$$

$$r_4 = (B \overset{6}{\rightarrow} b) \qquad r_5 = (C \overset{5}{\rightarrow} AD) \qquad r_6 = (C \overset{4}{\rightarrow} c)$$

$$r_7 = (D \overset{3}{\rightarrow} a)$$

a funkciou η, kde $\eta(S) = 1$, $\eta(A) = 2$, $\eta(B) = 3$, $\eta(C) = 4$, $\eta(D) = 5$.

Z reťazca bAAD odvodením určeným postupnosťou $d = (r_2,2)(r_4,1)$ dostávame reťazec bAbD s mierou príslušnosti 48. Teda $\mathrm{bAAD} \overset{48}{\Rightarrow} \mathrm{bAbD}(\mathrm{mod}(\mathrm{r}_2,2)(\mathrm{r}_4,1))$. Analogicky platí $S \overset{63}{\Rightarrow} bbC(\mathrm{mod}(r_1,1)(r_3,1))$, $S \overset{45}{\Rightarrow} \mathrm{bAAD}\,(\mathrm{mod}(\mathrm{r}_1,1)(\mathrm{r}_5,1))$.

V gramatike je možné odvodiť bbc rôznym spôsobom, napríklad $S \overset{252}{\Rightarrow} \mathrm{bbc}\,(\mathrm{mod}(\mathrm{r}_1,1)(\mathrm{r}_3,1)(\mathrm{r}_6,1))$ alebo $S \overset{1728}{\Rightarrow} \mathrm{bbc}\,(\mathrm{mod}(\mathrm{r}_1,1)(\mathrm{r}_2,1)(\mathrm{r}_4,1)(\mathrm{r}_6,1))$. Do fuzzy jazyka generovaného gramatikou patria nasledujúce dvojice:

$$\mathrm{L_{MP}(G)} = \{(\mathrm{a},15), (\mathrm{b},96), (\mathrm{c},16), (\mathrm{ba},2880), (\mathrm{bbc},1\,728), (\mathrm{bbba},331040)\}.$$

$(\mathrm{a},15) \in \mathrm{L_{MP}(G)}$ pretože jediné odvodenie slova a v G má tvar: $D \overset{3}{\Rightarrow} \mathrm{a}\,(\mathrm{mod}(\mathrm{r}_7,1))$ a $\eta(\mathrm{D}).3 = 15$. Analogicky dostávame $(\mathrm{c},16) \in \mathrm{L_{MP}(G)}$, $(\mathrm{b},96) \in \mathrm{L_{MP}(G)}$, pretože nasledujúce odvodenie poskytuje maximálnu hodnotu $\mu_G(b)$:

$$A \overset{48}{\Rightarrow} \mathrm{b}\,(\mathrm{mod}(\mathrm{r}_2,1)(\mathrm{r}_4,1))\ \mathrm{a}\ \eta(\beta) = \eta(\mathrm{A}).48 = 96.$$

Čitateľ sa môže presvedčiť že aj ďalšie hore uvedené dvojice patria do jazyka.

Teória fuzzy gramatík a špeciálne fuzzy bezkontextových gramatík a jazykov a max-prod gramatík je spracovaná v monografiách [8] a [10]. Čitateľ sa v nich má možnosť oboznámiť s redukovanými fuzzy resp. max-prod gramatikami, s fuzzy variantmi gramatík v Chomského normálnom tvare a v Greibachovej normálnom tvare, ktoré sú normálnym tvarom pre jazyky s konečným ohodnotením.

Fuzzy jazyky s prahovou hodnotou λ budú v prípade max-prod gramatík určené hodnotou $\lambda \in \mathrm{R}_{\infty}^{\geq 0}$. Budeme venovať pozornosť nasledujúcim triedam týchto jazykov.

Definícia: Nech $\lambda \in \mathrm{R}^{\geq 0}$. Potom definujeme triedy jazykov:

$$\mathcal{L}_\lambda = \{L(\mu_G, \lambda) \mid G \text{ je bezkontextová gramatika}\} \text{ a}$$

$$\mathcal{L} = \cup_{\lambda \in \mathrm{R}^{\geq 0}} \mathcal{L}_\lambda.$$

Pre jazyky max-prod gramatík s prahovou hodnotou platia nasledujúce tvrdenia [10]:

Veta: Jazyky $L(\mu_G, \lambda)$ striktných max-prod bezkontextových gramatík G sú bezkontextové jazyky. Jazyky $L(\mu_G, \infty)$ sú bezkontextové.

Veta: Trieda bezkontextových jazykov je podtriedou $\mathcal{L}_\lambda$ pre $\lambda \in \mathrm{R}^{\geq 0}$. $\mathcal{L} = \mathcal{L}_\lambda$ ak $\lambda \in \mathrm{R}^+$ a $\mathcal{L}_\lambda$ pre $\lambda = 0$ je rovná triede bezkontextových jazykov.

Bezkontextové jazyky sú vlastnou podtriedou $\mathcal{L}$. Jazyk $\mathrm{L} = \{\mathrm{a}^\mathrm{n}\mathrm{b}^\mathrm{n}\mathrm{c}^\mathrm{m} \mid \mathrm{n} \geq \mathrm{m} \geq 1\}$ je príkladom jazyka z $\mathcal{L}$, ktorý nie je bezkontextový.

Pre max-prod gramatiky platí nasledujúca pumpovacia veta.

Veta: Nech G je max-prod gramatika. Potom pre všetky $L = L(\mu_G, \lambda)$ a $\lambda \in \mathrm{R}^{\geq 0}$ existuje prirodzené číslo n také, že každé slovo $x \in L, |x| > \mathrm{n}$, je tvaru $\mathrm{x} = \mathrm{x}_1\mathrm{x}_2\mathrm{x}_3\mathrm{x}_4\mathrm{x}_5$ pre $x_2x_4 \neq \varepsilon$ a pre všetky k prirodzené čísla platí buď $x_1x_2^kx_3x_4^kx_5 \in L$ alebo $x_1x_3x_5 \in L$.

Dôsledok: Kontextový jazyk $L = \{a^nb^nc^n \mid n \geq 1\}$ nepatrí do $\mathcal{L}$.

Veta: Trieda $\mathcal{L}$ je uzavretá na zjednotenie, prienik s regulárnymi množinami, zreťazenie s bezkontextovými jazykmi, substitúciu za bezkontextové jazyky, homomorfizmus, inverzný homomorfizmus a zrkadlový obraz. Trieda $\mathcal{L}$ nie je uzavretá na prienik.

3 L systémy a fuzzy L systémy

Lindenmayerove systémy sú typickým predstaviteľom paralelne pracujúcich gramatík. Paralelizmus výpočtu v týchto systémoch úzko súvisí s biologickou motiváciou ich zavedenia v roku 1968. Od sekvenčných gramatík sa odlišujú okrem totálne paralelného prepisovania aj začiatkom a ukončením výpočtu. L systémy na rozdiel od sekvenčných gramatík pracujú len s jednou abecedou a v každom kroku odvodenia prepisujú všetky symboly aktuálneho reťazca. Výpočet sa začína štartovacím slovom, axiómou a do jazyka vytvoreného L systémom patria všetky vygenerované slová. Základným predstaviteľom Lindenmayerovych systémov sú systémy bez interakcie, skrátene 0L systémy.

Formálne je 0L systém je trojica $G = (\Sigma, P, \omega)$, kde Σ je abeceda systému, P je konečná množina pravidiel tvaru $a \to w$, kde $a \in \Sigma$, $w \in \Sigma^*$ a $\omega \in \Sigma^*$ je axióma systému. Množina P je úplná v tom zmysle, že pre každé $a \in \Sigma$ obsahuje aspoň jedno pravidlo určujúce prepis a.

Ako jednoduchý ilustračný príklad 0L systému uvedieme exponenciálne rastúci systém s abecedou obsahujúcou jediný symbol a, obsahujúci jediné pravidlo $a \to aa$

a začínajúci axiómou a. Tento systém vytvára reťazce a-čok nasledujúcim spôsobom:

$$a \Rightarrow aa \Rightarrow aaaa \Rightarrow a^8 \Rightarrow a^{16} \Rightarrow a^{32} \Rightarrow a^{64} \Rightarrow \ldots$$

a generuje jazyk $L(G) = \{a^{2^n} : n \geq 0\}$.

Uvedieme ešte jeden ilustračný príklad s pomalším rastom. 0L systém G bude pracovať s abecedou $\Sigma = \{a, b\}$, s pravidlami $P = \{a \to a, b \to aba\}$ a axiómou bb.

Prvé kroky odvodenia z axiómy bb sú nasledovné:

$$bb \Rightarrow abaaba \Rightarrow aabaaaabaa$$

V každom kroku odvodenia sa okolo každého z dvojice b-čok vygenerujú dve nové písmená a a všetky pôvodné písmená a sa zachovajú.

Jazyk vytvorený týmto 0L systémom je $L(G) = \{a^n b a^{2n} b a^n : n \geq 0\}$.

Okrem tohto základného modelu L systému sa v teórii stretneme s rôznymi dobre motivovanými rozšíreniami: tabuľkovými L systémami, interaktívnymi L systémami, L systémami s vyčlenenou koncovou abecedou, zakódovanými L systémami,.... V ďalšej časti sa oboznámime s fuzzy variantmi niektorých z nich.

Čitateľovi odporúčame do pozornosti publikácie [5] a [8], kde sa môže oboznámiť so základmi teórie Lindenmayerovych systémov a Handbook [14] predstavujúci túto teóriu na pokročilej úrovni.

3.1 Fuzzy L systémy

V tejto časti budeme študovať fuzzy L systémy reprezentujúce paralelné gramatiky. Zachytávajú mechanizmy vhodné na modelovanie vývoja, do ktorého je začlenená vágnosť vlastná väčšine procesov. Fuzzy L systémy sa môžu hodiť na modelovanie nejasne určených pravidiel vývoja s následkami na štruktúru modelovaného objektu.

Definícia: Fuzzy 0L systém, skrátene F0L systém, je štruktúra $FG = (\Sigma, P, \omega, K, \mu)$, kde:

- Σ je neprázdna konečná množina, abeceda,
- P je množina pravidiel tvaru $a \to \alpha$ pre $a \in \Sigma$, $\alpha \in \Sigma^*$,
- ω je axióm, prvok množiny Σ^*,
- $K = \{r_1, r_2, \ldots, r_p\}$ je množina označení pravidiel,
- $\mu : K \to [0,1]$ je zobrazenie, ktoré určuje mieru aplikácie pravidla z P.

Ak $P' \subseteq P$ je množina pravidiel s nenulovou mierou aplikácie, potom $G = (\Sigma, P, \omega)$ nazveme 0L systémom prislúchajúcim k F0L systému FG a označíme UFG systém.

Definícia: Nech $FG = (\Sigma, P, \omega, K, \mu)$ je fuzzy 0L systém. Nech d je odvodenie $\omega \Rightarrow^* x$ v UFG systéme. Nech $K' \subseteq K$ je

množina označení práve tých pravidiel, ktoré sa v odvodení d používajú. Potom položíme

$$\mu(d) = \min\{\mu(r_i) \mid r_i \in K'\}.$$

Nech $d_1, d_2, \ldots, d_t$ sú rôzne odvodenia slova x v UFG, potom definujeme $\mu(x)$, stupeň neurčitosti reťazca x, takto:

$$\mu_{FG}(\omega) = \max\{\mu(r_i) \mid r_i \in K\},$$

$$\mu_{FG}(x) = \max\{\mu(\mathrm{d_i}) \mid \mathrm{d_i} \text{ je odvodenie } x, 1 \le i \le t\} \text{ pre } x \in L(UFG) - \{\varpi\},$$

$$\mu_{FG}(x) = 0 \text{ pre } x \notin L(UFG).$$

Definícia: Fuzzy jazyk generovaný fuzzy 0L systémom FG je

$$L(FG) = \{(x, \mu_{FG}(x)) \mid x \in L(UFG)\}.$$

Príklad: F0L systém $FG = (\{a\}, P, a, \{1,2\}, \mu)$, kde P a μ sú dané nasledovne:

1. $a \to aa \qquad \mu(1) = 0{,}2$
2. $a \to aaa \qquad \mu(2) = 0{,}3$

generuje fuzzy jazyk:

$$L = \{(\varepsilon;0)\} \cup \{(a^m;0{,}3) \mid m = 3^s, s \ge 0\} \cup \{(a^m;0{,}2) \mid m \ne 3^s \text{ pre } s \ge 0, m \ne 0\}$$

Z celkovej abecedy $\sum L$ systému môžeme vyčleniť jej podabecedu Δ a zaujímať sa len o tie slová, ktoré sú vytvorené z písmen podabecedy. Takto dostávame z 0L systému E0L systém a z fuzzy 0L systému fuzzy E0L systém.

Definícia: Fuzzy E0L systém, skrátene FE0L systém, je štruktúra $FG = (\sum, P, \omega, K, \mu, \Delta)$, kde $(\sum, P, \omega, K, \mu)$ je F0L systém a $\Delta \subseteq \sum$. E0L systém $UFG = (\sum, P', \omega, \Delta)$, priradený k $FG = (\sum, P, \omega, K, \mu, \Delta)$ je systém, kde $(\sum, P', \omega)$ je U0L systém priradený k 0L systému $(\sum, P, \omega, K, \mu)$.

Fuzzy jazyk definovaný fuzzy E0L systémom je

$$L(FG) = \{(x, \mu_{FG}(x)) \mid x \in L(UFG) \cap \Delta^*\}$$

Príklad: Nech je daný FE0L systém $FG = (\{S, A, B, a, b\}, P, S, \{1, \ldots, 10\}, \mu, \{a, b\})$, kde P a μ sú dané nasledovne:

1. $S \to aSb$	$\mu(1) = 1$	6. $S \to Bb$	$\mu(6) = 0{,}6$
2. $S \to ab$	$\mu(2) = 1$	7. $B \to Bb$	$\mu(7) = 0{,}6$
3. $S \to aA$	$\mu(3) = 0{,}8$	8. $B \to \varepsilon$	$\mu(8) = 0{,}6$
4. $A \to aA$	$\mu(4) = 0{,}8$	9. $a \to a$	$\mu(9) = 1$
5. $A \to \varepsilon$	$\mu(5) = 0{,}8$	10. $b \to b$	$\mu(10) = 1$

Jazyk generovaný fuzzy E0L systémom *FG* je:

$$L = \{(\varepsilon;0)\} \cup \{(a^x;0) \mid x \neq a^* b^*\} \cup \{(a^m b^m;1) \mid m \geq 1\} \cup$$
$$\{(a^m b^m;0{,}8) \mid m > \mathrm{n}\} \cup \{(a^m b^m;0{,}6) \mid n > \mathrm{m}\}.$$

Inou modifikáciou 0L systémov sú tabuľkové 0L systémy, skrátene T0L systémy, v ktorých máme k dispozícii niekoľko skupín pravidiel, ktoré môžeme po jednotlivých krokoch striedať. Reprezentujú vývoj za rôznych podmienok.

Definícia: Fuzzy T0L systém, skrátene FT0L systém, je štruktúra $FG = (\sum, \mathcal{P}, \omega, K, \mu)$, kde:

- $\mathcal{P}$ je množina tabuliek, každá tabuľka P z $\mathcal{P}$ je množina pravidiel,
- K je množina označení pravidiel tvaru (i, j). Dvojica (i, j) označuje j-te pravidlo i-tej tabuľky,
- $\mu : K \to [0,1]$ je zobrazenie určujúce mieru aplikácie pravidiel.

$UFG = (\sum, \mathcal{P}', \omega)$ je základný T0L systém priradený FT0L systému FG za predpokladu, že $\mathcal{P}'$ obsahuje práve tie pravidlá z P, ktoré majú mieru aplikácie väčšiu ako 0.

Definícia: Nech $FG = (\sum, \mathcal{P}, \omega, K, \mu)$ je fuzzy FT0L systém. Odvodenie $d : \omega \Rightarrow^* x$ v UFG systéme je určené tak, že na každom kroku odvodenia sa používajú pravidlá len z jednej tabuľky. Nech $K' \subseteq \mathrm{K}$ je množina označení práve tých pravidiel, ktoré sa v odvodení d používajú. Potom položíme $\mu(d) = \min\{\mu(r_1) \mid r_1 \in K'\}$.

Nech $\mathrm{d}_1, \mathrm{d}_2, \ldots, \mathrm{d}_\mathrm{t}$ sú rôzne odvodenia x v UFG potom definujeme $\mu(x)$ takto: Stupeň neurčitosti reťazca x je:

$$\mu_{FG}(\omega) = \max\{\mu(r_i) \mid r_i \in K\} \text{ a}$$
$$\mu_{FG}(x) = \max\{\mu(d_i) \mid l \leq i \leq t\} \text{ pre}$$
$$x \in L(UFG) - \{\varpi\},$$
$$\mu_{FG}(x) = 0 \text{ pre } x \notin L(UFG).$$

Definícia: Fuzzy jazyk generovaný fuzzy T0L systémom FG je

$$L(FG) = \{(x, \mu_{FG}(x)) \mid x \in L(UFG)\}$$

Príklad: Nech je daný FT0L systém $FG = (\{a\}, \{P_1, P_2\}, a, \{(1,1), (1,2), (2,1), (2,2)\}, \mu)$, kde P_1, P_2 a μ sú dané nasledovne:

P_1:	(1,1) $a \to aa$	$\mu((1,1)) = 0{,}4$		P_2:	(2,1) $a \to aaa$	$\mu((2,1)) = 0{,}1$
	(1,2) $a \to \varepsilon$	$\mu((1,2)) = 0{,}5$			(2,2) $a \to \varepsilon$	$\mu((2,2)) = 0{,}5$

Jazyk generovaný fuzzy E0L systémom je:

$$L = \{(a;0{,}5), (\varepsilon;0{,}5)\} \cup \{(a^{2n};0{,}3) \mid n \geq 1\} \cup \{(a^m;0{,}1) \mid m = 3(2k+1), k \geq 0\} \cup$$
$$\{(a^m;0) \mid \text{v ostatných prípadoch}\}.$$

Analogicky je možné zaviesť fuzzy ET0L systém a všetky varianty fuzzy IL systémov.

Označíme $\mathcal{L}$(F0L) (resp. $\mathcal{L}$(FD0L), $\mathcal{L}$(FE0L), $\mathcal{L}$(FT0L), $\mathcal{L}$(FET0L)) triedu všetkých F0L jazykov (resp. D0L, E0L, T0L, ET0L jazykov).

Pre uvedené triedy jazykov a triedy $\mathcal{L}$(FCFL), $\mathcal{L}$(FCFL) fuzzy bezkontextových a fuzzy kontextových jazykov platia nasledovné vzťahy [13]:

Veta: $\mathcal{L}(\text{FD0L}) \subset \mathcal{L}(\text{F0L}) \subset \mathcal{L}(\text{FE0L}) \subset \mathcal{L}(\text{FET0L})$

$\mathcal{L}(\text{F0L}) \subset \mathcal{L}(\text{FT0L}) \subset \mathcal{L}(\text{FET0L})$

$\mathcal{L}(\text{FCFL}) \subset \mathcal{L}(\text{FE0L}) \subset \mathcal{L}(\text{FET0L}) \subset \mathcal{L}(\text{FCSL})$

Tieto vzťahy kopírujú známe analogické vzťahy pre nefuzzyfikované varianty L systémov.

Zavedieme varianty fuzzy L systémov s nepresnou axiómou podobne ako v prípade striktných max-prod gramatík. Štartovací symbol L systému bude určený funkciou $\eta : \Sigma \to [0,1]$, ktorá priradí symbolu a hodnotu $\eta(a)$, stupeň príslušnosti faktu, že a je štartovacím symbolom. Z každého symbolu s nenulovou hodnotou $\eta(a)$ je možné začať odvodenie. Fuzzy jazyky definované takýmito systémami označíme s indexom F_M a následne triedy fuzzy jazykov označíme $\mathcal{L}(F_M$ 0L) (resp. $\mathcal{L}(F_M$ E0L), $\mathcal{L}(F_M$ T0L), $\mathcal{L}(F_M$ ET0L)). Násobné axiómy majú za následok nasledujúce zmeny v generatívnej sile:

Veta: $\mathcal{L}$(F0L) a $\mathcal{L}(F_M$ 0L) sú neporovnateľné.

$\mathcal{L}(F_M$ T0L) a $\mathcal{L}$(FT0L) sú neporovnateľné.

$\mathcal{L}$(FE0L) = $\mathcal{L}(F_M$ E0L).

Podmnožina fuzzy T0L systémov takých, v ktorých je stupeň všetkých pravidiel v tabuľke rovnaký, má menšiu generatívnu silu ako FT0L.

Fuzzy jazyky s prahovou hodnotou pre T0L systémy a ET0L systémy nezvyšujú generatívnu silu pôvodného typu systémov.

Ďalším okruhom skúmaným v literatúre pre fuzzy L jazyky a jazyky fuzzy L systémov určených prahovými hodnotami sú uzáverové vlastnosti skúmaných tried jazykov vzhľadom ku všetkým známym operáciám a typom zobrazení. S týmito výsledkami sa môže čitateľ oboznámiť napríklad v [13] a [16].

4 Gramatické systémy a ich fuzzy varianty

Generatívnym modelom nemusí byť len jedna gramatika, ale aj niekoľko gramatík, ktoré spolupracujú predpísaným spôsobom na postupnom generovaní slov z jazyka. V tomto prípade hovoríme o gramatických systémoch. ***Gramatické systémy*** predstavujú súčasť teórie formálnych jazykov, ktorá sa zaoberá syntaktickými modelmi určenými na popis a štúdium (správania) decentralizovaných (a často tiež multiagentových) systémov na symbolickej úrovni.

Gramatické systémy sú tak výpočtovými zariadeniami ako aj prostriedkami, ktoré umožňujú zachytiť a charakterizovať v decentralizovaných systémoch kooperáciu, distribúciu, komunikáciu, paralelizmus, emergentné správanie a iné.

Ako typického predstaviteľa gramatických systémov uvedieme ***kooperujúce distributívne gramatické systémy*** (skrátene ***CD gramatické systémy***).

Voľne povedané CD gramatický systém pozostáva z niekoľkých gramatík, ktoré v spolupráci, postupným používaní jednotlivých komponentov (gramatík) systému menia prostredie prepisovaním reťazca symbolov, ktoré ho reprezentuje. Správanie sa CD gramatického systému môže byť charakterizované množinou všetkých stavov prostredia, ktoré je možné vytvoriť z počiatočného stavu a ktoré spĺňajú isté kritéria ukončenia.

Definícia: CD ***gramatický systém*** $\Gamma = (V, T, G_1, \ldots, G_n, S)$ je štruktúra taká, že:

- V a T sú dve konečné množiny, úplná abeceda V a terminálna abeceda T, $T \subseteq V$,
- $G_i = (N_i, T_i, P_i, S_i)$ je gramatika určujúca i-tu komponentu Γ pre $1 \le i \le n$. Pre každú komponentu systému, P_i je množina pravidiel gramatiky G_i s terminálnymi symbolmi T_i a neterminálnymi symbolmi N_i a $N_i \cup T_i \subseteq V$,
- $S \in N$ je štartovací symbol celého systému.

Hovoríme, že z u je možné odvodiť v i-tym komponentom v t-móde (v ukončenom móde), ak použijeme niektoré pravidlá z P_i na prepis reťazca u za sebou dovtedy, kým je to možné a odvodenie je ukončené reťazcom v.

Definícia: Píšeme $u \Rightarrow_i^t v$ práve vtedy keď:

- $u = u_0 \Rightarrow_i u_1 \Rightarrow_i \ldots \Rightarrow_i u_k = v$ pre $k \ge 1$, a
- $v \Rightarrow_i w$ pre žiadne $w \in V^*$.

(Druhá odrážka zaručuje, že nie je možné prepísať v v G_i žiadnym pravidlom.)

Jazyk vytvorený CD gramatickým systémom Γ v t-móde odvodenia pozostáva zo všetkých terminálnych slov z T^* odvodených z axiómy postupným používaním komponentov systému:

$$L_t(\Gamma) = \{w \in T^* : S \Rightarrow_{i_1}^t w_1 \Rightarrow_{i_2}^t w_2 \ldots \Rightarrow_{i_l}^t w_l, w_l = w;$$
$$1 \le i_j \le n; 1 \le j \le l, \text{ pre nejaké } l \ge 1\}$$

Príklad: Jednoduchým príkladom gramatického systému je systém s tromi komponentmi $\Gamma = (V, T, G_1, G_2, G_3, A)$, kde $V = \{A, B, a\}$, $T = \{a\}$ a komponenty sú určené nasledujúcimi množinami pravidiel:

$$P_1 = \{A \to BB\}, \qquad P_2 = \{B \to A\}, \qquad P_3 = \{A \to a, A \to b\}.$$

Reťazce obsahujúce A môžeme v tomto systéme prepísať prvým komponentom na reťazce obsahujúce BB nahradzujúce A alebo tretím komponentom na ľubovoľné slovo nahradzujúce všetky A-čka symbolmi a alebo b. Reťazce obsahujúce B môžu byť nahradené pomocou druhej komponenty na reťazce nahradzujúce symboly B za symboly A.

Typické odvodenie v tomto gramatickom systéme je:

$$A \Rightarrow^t BB \Rightarrow^t AA \Rightarrow^t BBBB \Rightarrow^t AAAA \Rightarrow^t aaba$$

a jazyk generovaný gramatickým systémom Γ tvoria slová w dĺžky 2^k pre $k \geq 0$,

$$L_t(\Gamma) = \{w \mid w \in \{a,b\}^*, |w| = 2^k, k \geq 0\}.$$

V súčasnosti je teória gramatických systémov široko rozvinutou súčasťou teórie formálnych gramatík a jazykov s niekoľkými dobre rozpracovanými podoblasťami a využíva sa aj ako zdroj inšpirácie pri rozvoji iných teórií viažucich sa na výpočty v niektorých syntaktických modeloch spracovania symbolov. Detailnejšie sa s gramatickými systémami môže čitateľ zoznámiť v monografii [2], alebo v kapitole z [14].

V ďalšom uvedieme možnosti ako zaviesť fuzzy modely gramatických systémov na základe skúseností z predchádzajúcich postupov pri fuzzy gramatikách a fuzzy L systémoch. Budeme pri tom sledovať myšlienky z prác [6] a[7].

4.1 Systémy fuzzy gramatík

Ak nie je presne známa spoľahlivosť pravidiel v komponentoch gramatických systémov, je možné použiť fuzzy pravidlá v gramatikách komponentov systému na vyjadrenie tejto nejasnosti. Komponenty systému budú fuzzy gramatikami a vytvorená štruktúra je systém fuzzy gramatík. Prepisovacie pravidlá v komponentových gramatikách tvaru $\alpha \overset{p}{\rightarrow} \beta$, kde $\alpha \in N, \beta \in (N \cup T)^*$ a $p \in [0,1]$ budú určovať fuzzy kompetencie individuálnych gramatík (označíme ako μ_c). Komponentové gramatiky gramatického systému prispievajú k odvodeniu ich vlastnou fuzzy kompetenciou priradenou k odvodeniu reťazcov v gramatickom systéme. Stupeň odvodenia $u \Rightarrow^t v$ v komponentovej gramatike je daný vzťahom $\mu_c(u \Rightarrow^t v) = x$ (stručne $u \overset{x}{\Rightarrow}^t v$). Funkcia μ_c môže byť daná napríklad ako μ pre fuzzy gramatiky alebo ako μ pre max-prod gramatiky.

Stupeň odvodenia slova v celom gramatickom systéme je potom určený funkciou μ_s definovanou pre kompozíciu krokov odvodenia individuálnych komponentov. Funkcie μ_s a μ_c môžu byť identického typu v tom zmysle, že obidve sú definované ako vo fuzzy gramatikách (alebo sú obidve definované ako v max-prod gramatikách) alebo môžeme tiež uvažovať prípady, keď sú μ_s a μ_c odlišné. Miera príslušnosti μ_Γ gramatického systému je určená dvojicou $\mu_\Gamma = (\mu_s, \mu_c)$.

Následne získame fuzzy jazyky ako výsledok generovania slov v gramatickom systéme, kde stupeň slova odvodeného systémom fuzzy gramatík je určený ako suprémum stupňa všetkých jeho odvodení.

Tieto myšlienky budeme ilustrovať na príklade gramatického systému z predchádzajúceho odseku. Pravidlám v našich komponentoch priradíme fuzzy hodnoty takto:

$$P_1 = \{A \overset{x}{\to} BB\}, \qquad P_2 = \{B \overset{y}{\to} A\}, \qquad P_3 = \{A \overset{z}{\to} a, A \overset{u}{\to} b\}.$$

Špeciálne pre $x = y = z = u = 1$ v t-móde odvodenia získame ten istý jazyk obsahujúci všetky slová w zložené z a a b dĺžky $2^k, k \geq 0$. Všetky odvodené terminálne slová majú stupeň 1 pre všetky $\mu_G = (\mu_s, \mu_c)$, ktoré sme doteraz uvažovali.

1) Predpokladajme, že $\mu_G = (\mu_s, \mu_c)$, kde μ_s a μ_c odpovedajú miere μ fuzzy gramatík. Typické odvodenie v tomto systéme fuzzy gramatík je napríklad:

$$A \overset{x}{\Rightarrow}^t BB \overset{y}{\Rightarrow}^t AA \overset{x}{\Rightarrow}^t BBBB \overset{y}{\Rightarrow}^t AAAA \overset{k}{\Rightarrow}^t aaba$$

Pre prvú komponentu použitú v kroku $A^i \overset{x}{\Rightarrow}^t B^{2i}$ platí $\mu_c(A^i \Rightarrow^t B^{2i}) = x$ čo je minimum stupňov i individuálnych krokov, ktoré používajú pravidlo $A \overset{x}{\to} BB$ v tomto odvodení. Analogicky, druhý komponent fuzzy gramatiky v kroku $B^i \overset{y}{\Rightarrow}^t A^i$ dáva hodnotu $\mu_c(B^i \Rightarrow^t A^i) = y$, čo je minimum stupňov i individuálnych krokov realizovaných pravidlom $B \overset{y}{\to} A$ v tomto odvodení. Použitie pravidiel tretej komponenty fuzzy gramatiky dáva hodnoty:

$k = z$ pre terminálne slová nad a,
$k = u$ pre terminálne slová nad b,
$k = \min\{z, u\}$ v ostatných prípadoch.

Nech stupeň μ_s celého odvodenia je daný ako minimum stupňa všetkých individuálnych krokov komponentov v odvodení. Pre každé slovo vytvorené systémom fuzzy gramatík existuje len jedno možné odvodenie, preto maximum stupňa cez všetky odvodenia slov v L je v tomto prípade rovné

$\mu_\Gamma(a) = z$,
$\mu_\Gamma(a^n) = \min\{x, y, z\}$ pre $n = 2^k, k \geq 1$,
$\mu_\Gamma(b) = u$,
$\mu_\Gamma(b^n) = \min\{x, y, z\}$ pre $n = 2^k, k \geq 1$ a
$\mu_\Gamma(w) = \min\{x, y, z, u\}$ pre $n = 2^k, k \geq 1$ v ostatných prípadoch.

2) Predpokladajme $\mu_\Gamma = (\mu_s, \mu_c)$, kde obidva μ_s a μ_c odpovedajú μ pre max-prod gramatiky.

Stupeň odvodenia terminálneho slova w je daný ako súčin stupňov všetkých individuálnych krokov komponentov. V tomto prípade to je identické so súčinom stupňov všetkých individuálnych krokov v tomto odvodení. Uvažujme odvodenie:

$$A \overset{x_1}{\Rightarrow}^t BB \overset{y_1}{\Rightarrow}^t AA \overset{x_2}{\Rightarrow}^t BBBB \overset{y_2}{\Rightarrow}^t AAAA \overset{k}{\Rightarrow}^t aaba$$

Keď použijeme v kroku $A^i \overset{x_i}{\Rightarrow}^t B^{2i}$ pravidlá prvej komponenty, potom získame $\mu_c(A^i \Rightarrow^t B^{2i}) = x^i$, t.j. hodnotu, ktorá je súčinom stupňov i individuálnych krokov používajúcich $A \overset{x}{\to} BB$ v tomto odvodení. Analogicky, použitie pravidiel druhej

komponenty v kroku $B^i \overset{y_i}{\Rightarrow}{}^t A^i$, určuje hodnotu $\mu_c\left(B^i \Rightarrow^t A^i\right) = y^i$, ktorá je súčinom stupňov i individuálnych krokov používajúcich $B \overset{y}{\rightarrow} A$ v tomto odvodení. Použitie pravidiel tretej komponenty dáva:

$k = z^t$ pre terminálne slovo a^t,

$k = u^t$ pre terminálne slová nad b^t,

$k = z^t u^t$ pre terminálne slovo w s t výskytmi a a t' výskytmi b vo w.

Stupeň príslušnosti reťazca w v jazyku určenom gramatikou je daný ako suprémum stupňa všetkých odvodení w v gramatike. Každé slovo má len jedno odvodenie preto $\mu(aaba) = x.y^2.x^2.y^4.z^3.u$.

Pre $w \in L$ dĺžky 2^n a pre hodnotu k určenú ako hore je $\mu(w) = x.x^2 \ldots x^{2^n}.y^2 \ldots y^{2^n}.k$.

Je jednoduché skompletizovať aj zostávajúce prípady, kedy je jedna z funkcií μ_s a μ_c v $\mu_\Gamma = (\mu_s, \mu_c)$ daná funkciou μ pre fuzzy gramatiky a druhá funkciou μ pre max-prod gramatiky. Detaily ponecháme na čitateľa.

4.2 Fuzzy systémy gramatík

Ku špecifickej situácii dochádza, keď nie je fungovanie komponentov gramatiky perfektne určené. V tomto prípade priradíme fuzzy hodnoty komponentom gramatických systémov tak, aby charakterizovali príslušnosť komponentu do gramatického systému. V tomto prípade je funkcia μ priradená jednotlivým komponentom gramatického systému. Takýto systém budeme nazývať ***fuzzy systémom gramatík***.

Fuzzy jazyky fuzzy systémov gramatík budú výsledkom fuzzy aktivity komponentov systému. Jedná sa o fuzzy efekt celej komponentovej gramatiky na celé odvodenie v gramatickom systéme, jednotlivé pravidlá v komponentoch sú pravidlami klasickej gramatiky Chomského hierarchie.

Odvodené slová budú patriť do jazyka generovaného gramatickým systémom s mierou určenou mierami komponentových gramatík použitých v ich odvodeniach. Fuzzy systém gramatík môžeme považovať za špeciálny prípad systému fuzzy gramatík, kde v $\mu_\Gamma = (\mu_s, \mu_c)$ je funkcia μ_c rovná konštante pre všetky možné odvodenia v príslušnej komponente. To zodpovedá špeciálnemu prípadu, kedy sa μ_c určuje ako vo fuzzy gramatikách, pričom každé pravidlo komponenty má priradenú rovnakú fuzzy hodnotu. Ilustrujeme túto myšlienku na príklade fuzzy systému gramatík.

Príklad: $\Gamma = (V, T, G_1, G_2, G_3, A)$, kde $V = \{A, B, a\}$, $T = \{a\}$ s komponentovými gramatikami určenými pravidlami:

$$P_1 = \{A \rightarrow BB\}, \qquad P_2 = \{B \rightarrow A\}, \qquad P_3 = \{A \rightarrow a, A \rightarrow b\}.$$

Priradíme každej komponentovej gramatike jej mieru z intervalu $[0,1]$:

$\mu(G_1) = x$, $\mu(G_2) = y$, $\mu(G_3) = z$.

Ku každému kroku odvodenia $\Rightarrow^t$ priradíme hodnotu $\mu(G_i)$ odpovedajúcu komponente G_i použitej v tomto kroku odvodenia. Typické odvodenie v tomto systéme fuzzy gramatík je:

$$A \overset{x}{\Rightarrow}^t BB \overset{y}{\Rightarrow}^t AA \overset{x}{\Rightarrow}^t BBBB \overset{y}{\Rightarrow}^t AAAA \overset{k}{\Rightarrow}^z aaba$$

Stupeň odvodenia v gramatickom systéme je určený funkciou μ_s definovanou pre kompozíciu krokov odvodenia individuálnych komponentov. Funkcia μ_s môže odpovedať fuzzy gramatike alebo max-prod gramatike a

$$\mu_\Gamma(w) = \sup\{\mu_s(w) : D \text{ odvodenie } w \text{ v } \Gamma\}$$

1. V prípade analogickom fuzzy gramatikám máme $\mu_\Gamma(a) = \mu_\Gamma(b) = z$, pretože len komponent G_3 je použitý v odvodení slov a a b. Pre všetky ostatné slová w z L platí $\mu_\Gamma(w) = \max\{x, y, z\}$.
2. V prípade analogickom k max-prod gramatikám máme $\mu(w) = (x, y)^n z$ pre každé $w \in L$ dĺžky 2^n. Na odvodenie slova w dĺžky 2^n použijeme n krát komponenty G_1, G_2 a odvodenie ukončíme komponentom G_3.

5 Záver

Cieľom tejto kapitoly bolo oboznámiť čitateľa s pojmami fuzzy gramatík, fuzzy L systémov a s fuzzy jazykmi, ktoré reprezentujú. Okrem základných modelov, ktoré sme uviedli, sa čitateľ môže zoznámiť aj s ďalšími variantmi fuzzy L systémov napríklad fuzzy L systémami s riadeným odvodením.

Vychádzajúc z týchto rozpracovaných modelov sme načrtli možnosti, ako zaviesť fuzzy varianty gramatických systémov. Nedotkli sme sa však ešte ďalšej možnosti, že by sa mód derivácie v rozličných komponentoch mohol meniť, prípadne že by s určitým stupňom neurčitosti používal jeden komponent rôzne typy odvodení. Naznačené postupy umožňujú definovať a rozpracovať fuzzy varianty pre gramatické systémy s ďalšími, v literatúre diskutovanými módmi odvodení, pre paralelné gramatické systémy, pre kolónie a ekogramatické systémy, pre systémy multimnožín.

Poďakovanie: Táto kapitola vznikla za podpory grantovej agentúry Slezské univerzity v Opave v rámci grantovej úlohy SGS 05/2010.

Literatúra

[1] Asveld, P. R. J.: Fuzzy context-free languages-part 1: generalized context-free grammars. *Theoretical Computer Science* **347**(1-2) (2005) 167–190.

[2] Csuhaj-Varjú, E. et al.: *Grammar Systems*. Yverdon: Taylor & Francis, 1994.

[3] Gnanasekaran, S.: Fuzzy Local Languages. *International Mathematical Forum*, **5** (2010) 2149 – 2155.

[4] Harrison, M.A.: *Introduction to Formal Language Theory*. Addison-Wesley, Reading, Mass. 1978.

[5] Herman, G. T., Rozenberg, G.: *Developmental Systems and Languages*. Elsevier Science Inc. New York 1975.

[6] Kelemen, J., Kelemenová, A.: New Computationalism - a Lesson from Embodied Agents. In Rudas, I., Kacprzyk, J. eds: *Towards Intelligent Engineering a Information Technology,* SCI vol. 243., Berlin: Springer, 2009, pp. 55-66.

[7] Kelemen, J., Kelemenová, A.: Managing Decentralized Uncertainties – Concept of Fuzziness in Grammar Systems. In *Proc. 11th IEEE International Symposium on Computational Intelligence a Informatics, CINTI 2010*. Obuda University, Budapest, 2010, pp. 33-37. (IEEE Catalog Number CFP 1024M-PRT).

[8] Kelemenová, A.: Lindenmayerove systémy a ich tvorca v spomienkach, faktoch a citátech. In Kelemen, V. Kvasnička, J. Pospíchal: *Kognice a umělý život V*, Slezská univerzita v Opavě, Opava, 2005, 235-250.

[9] Lee, E. T., Zadeh, L. A.: Note on fuzzy languages. *Information Science* **1** (1969) 421-434.

[10] Malik, D. S., Mordeson, J. N.: *Fuzzy Discrete Structures*. Heidelberg: Physica-Verlag, 2000.

[11] Molnár, Ľ., Češka, M., Melichar, B.: *Gramatiky a jazyky*. Bratislava: ALFA 1987.

[12] Mordeson, J. N., Malik, D. S.: *Fuzzy Automata a Languages*. Boca Raton: Chapman & Hall/CRC, 2002.

[13] Prasad, N. M., Mahajan, M, Krithivasan, K.: Fuzzy L systems. *Intern. J. Comp. Math.* **36** (1990)139-161.

[14] Rozenberg, G., Salomaa, A. Eds.: *Handbook of Formal Languages*. Berlin: Springer, 1997.

[15] Wechler, W.: *Concept Fuzziness v Automata a Language Theory*. Berlin: Academie-Verlag, 1978.

[16] Zu-Guo Yu: Fuzzy L Languages. *Fuzzy sets and systems* **117** (2001) 317-321.

Štúdium doskových hier pomocou neurónových sietí

Peter LACKO[1]

Vladimír KVASNIČKA[2]

Abstrakt. Umelá inteligencia sa venuje problému hrania hier už od vzniku počítačovej vedy. Za ten čas sa jej v niektorých hrách podarilo dosiahnuť vynikajúce výsledky, ale stále existujú problémy, ktoré nie je počítač schopný riešiť lepšie ako človek. To nám vytvára priestor na návrh nových algoritmov či už symbolickej alebo subsymbolickej umelej inteligencie. Neurónové siete boli úspešne použité pri tvorbe stratégií pre doskové hry. Významnú úlohu v tejto problematike hrá učenie s odmenou a trestom, ktoré pri trénovaní nepotrebuje učiteľa. V poslednej dobe boli dosiahnuté dobré výsledky aj s rekurentnými neurónovými sieťami.

1 Úvod

Prvé počítače vznikli počas druhej svetovej vojny a jedným z prvých programátorov bol Alan Turing. Turing je známy v odbore počítačovej vedy z rôznych dôvodov. Počas druhej svetovej vojny a po jej skončení Turing experimentoval so „strojovými procedúrami" na hranie šachu. Kvôli absencii počítača Turing simuloval vykonávanie inštrukcií ručne na papieri. Toto bolo prvýkrát, čo niekto uspel s použitím dobre definovaných pravidiel na hranie hry ako šach. O niečo neskôr, v roku 1952, Arthur L. Samuel [1] napísal program na hranie dámy. Základným cieľom jeho práce nebol program hrajúci hru, ale program, ktorý sa učí. Vo svojej práci vyskúšal dva prístupy - učenie zapamätávaním a učenie zovšeobecňovaním. Bol to asi historicky prvý program, ktorý bol schopný zdokonaľovať sám seba bez zásahu človeka.

Zlatým grálom umelej inteligencie bola hra šach. V sedemdesiatych a osemdesiatych rokoch minulého storočia sa výskum počítačového hrania hier sústredil hlavne na šach a prístup prehľadávania možných ťahov (angl. brute-force). Ukázalo sa, že je silná korelácia medzi prehľadávacou rýchlosťou a výkonnosťou programu.

[1] Fakulta informatiky a informačných technológií, Slovenská technická univerzita v Bratislave, Ilkovičova 3, Bratislava, E-mail: lacko@fiit.stuba.sk

[2] Fakulta informatiky a informačných technológií, Slovenská technická univerzita v Bratislave, Ilkovičova 3, Bratislava, E-mail: kvasnicka@fiit.stuba.sk

Dôsledkom bola koncentrácia na zrýchľovanie prehľadávacích algoritmov namiesto výskumu iných prístupov.

Koncom osemdesiatych rokov sa výskum znovu rozbehol a deväťdesiate roky priniesli mnoho úspechov. Prvá hra, ktorá mala nie-ľudského majstra, bola dáma. V roku 1994 program Chinook [2] vyhral svetový turnaj medzi človekom a strojom. Ďalší úspech sa dostavil veľmi skoro. Už v roku 1997 vyhral projekt Deep Blue nad úradujúcim svetovým šampiónom Garry Kasparovom [3]. V tomto období sa objavili aj nové prístupy učenia, ako napríklad učenie s odmenou a trestom [4] implementované v programe TD-gammon [5]. Kvalita jeho hrania sa blížila k úrovni najlepších hráčov hry backgammon na svete.

Počítačové hranie hier má ešte stále oblasti, v ktorých nedokážeme vytvoriť program, ktorý by sa vyrovnal ľuďom (napríklad hra Go [6]). Sú to hlavne hry, ktoré sa nedajú riešiť klasickými prístupmi umelej inteligencie, pretože majú veľký faktor vetvenia (hráč môže urobiť veľké množstvo rôznych ťahov). Práve tieto hry je zaujímavé riešiť novými prístupmi umelej inteligencie ako sú neurónové siete a evolučné algoritmy.

V tejto kapitole sa zameriame na využitie umelých neurónových sietí na hranie hier. Pod hrou budeme rozumieť doskovú hru ako dáma, šach alebo GO. Najprv si zadefinujeme hru ako problém, ktorý dokážeme formalizovať. Ďalej rozoberieme použitie dopredných, ale aj rekurentných sietí na riešenie problému hry. Ukážeme možnosti trénovania a "samoučenie" sietí s použitím prístupov učenia s odmenou a trestom ale aj evolučne.

2 Počítačové hranie hier

Hranie hier je zaujímavou a dôležitou časťou umelej inteligencie. Hry považujeme za problémy, ktoré nemajú triviálne riešenia. Na hranie rôznych hier existujú rôzne prístupy, či už symbolickej alebo subsymbolickej umelej inteligencie. Pre väčšinu zložitejších hier nepoznáme jeden optimálny algoritmus, a preto je to zaujímavá oblasť pre testovanie a vytváranie nových prístupov a algoritmov. V celej práci pod pojmom hry rozumieme väčšinou doskovú hru, ako napríklad šach, dáma, alebo reversi.

Problém hrania hier je pre nás zaujímavý hlavne z nasledovných dôvodov [30] (orientujeme sa na doskové hry):

- hra je štruktúrovaný a dobre definovaný problém, kde:
  - sú presne definované pravidlá,
  - sa dá jednoducho rozoznať úspech alebo neúspech hráča,
- ľubovoľný stav hry má presnú reprezentáciu
  - hráč má všetky informácie o prostredí hry,
  - hráčove informácie sú presné,
- úspešnosť riešenia sa dá dobre merať (počítač môže hrať proti inému programu, prípadne proti ľudskej bytosti).

Hry môžeme rozdeliť podľa toho, či sú počas hry všetky dostupné informácie známe všetkým hráčom na *hry s úplnou informáciou* a *hry s neúplnou informáciou*. V

nasledujúcich kapitolách opíšeme tieto hry spolu s ich najznámejšími počítačovými riešeniami.

2.1 Hry s úplnou informáciou

Hry s úplnou informáciou (angl. games of perfect information) sú tie, pri ktorých sú všetky dostupné informácie známe všetkým hráčom počas celej hry. Medzi najznámejšie patria piškvorky, dáma, šach, Go.

Najväčším úspechom v hraní šachu bolo bezpochyby víťazstvo počítačového systému Deep Blue [2] v turnaji proti svetovému šampiónovi Garry Kasparovovi v roku 1997.

Deep Blue bol masívny paralelný systém navrhnutý na prehľadávanie stromov hry šach. Systém pozostával z 30-uzlového (30-procesorového) IMB RS-6000 SP počítača a 480 šachových čipov určených na prehľadávanie stromov hry. Každý procesor ovládal 16 šachových čipov. Celkový výkon systému sa počas hry menil. Pri taktických pozíciách, ktoré vyžadovali veľké prehľadávanie do hĺbky, bola rýchlosť približne 100 miliónov ťahov za sekundu. Pre jednoduchšie pozície sa rýchlosť blížila k 200 miliónom prehľadaných ťahov za sekundu. Celkový priemer počas hry proti šampiónovi bol 126 miliónov ťahov a maximum bolo 330 miliónov ťahov za sekundu.

Ďalším dôležitým parametrom systému je vyhodnocovacia funkcia. Táto funkcia bola implementovaná v hardvéri šachových čipov, čo zabezpečovalo vysoký výkon. Nevýhodou tohto prístupu je nemožnosť pridávania nových vlastností do systému. Heuristická funkcia vyhodnocovala 8000 rôznych vzorov, ktorým priraďovala ohodnotenie. Boli to príznaky od najjednoduchších, ako figúrka na určitom mieste, až po komplexné analýzy situácií.

Systém Deep Blue počas hry používal aj rôzne knižnice. V začiatku hry sa zapájala knižnica otvorení, ktorá obsahovala približne 4000 pozícií. Ak nebolo možné nájsť dané otvorenie v tejto knižnici, používal sa výsledok z analýzy 700000 majstrovských hier. Táto analýza poskytovala systému Deep Blue určité predpoklady pre vhodnosť ťahov. Poslednou použitou knižnicou bola databáza koncoviek. Táto databáza obsahovala všetky možné hry s 5 a menej figúrkami na šachovnici.

Systému Deep Blue sa podarilo v roku 1997 poraziť Garryho Kasparova s výsledkom 3,5:2,5. Autori tak dokázali, že je možné poraziť svetového šampióna v tak zložitej hre, akou je šach. Ich riešenie spája rôzne prístupy symbolickej umelej inteligencie.

Deep Blue bol postavený vďaka firme IBM a nepracoval na bežnom počítači, ale pri hre dáma sa podarilo dosiahnuť majstrovské výsledky aj na osobných počítačoch. V roku 1989 začal Jonathan Schaeffer pracovať na programe, ktorý sa neskôr stal najznámejším programom na hranie dámy s názvom Chinook [37]. Chinook zaviedol niečo úplne nové na poli počítačovej dámy - databázu koncoviek. Tá dávala programu perfektné znalosti o všetkých pozíciách na šachovnici s osem a menej figúrkami vo forme víťazstvo/remíza/prehra. Znamená to, že v okamihu keď na šachovnici zostalo menej ako osem figúrok, Chinook dokázal ťahať optimálne. Takáto databáza vlastne neukladá veľa informácií, a teda sa dá dobre skomprimovať. Celá 8-figúrková databáza

mala približne 6GB v komprimovanej forme a obsahovala 443 miliárd pozícií. V dnešnej dobe majú už autori vypočítanú 10-figúrkovú databázu koncoviek. V roku 1996 program Chinook (vtedy ešte len s 8-figúrkovou databázou) skončil na prvom mieste amerického šampionátu, kde zanechal ľudského šampióna ďaleko za sebou (porazil ho trikrát). Dáma ma podobné hodnotenie hráčov ako šach. V roku 1996 vyzeralo poradie najlepších hráčov sveta americkej federácie dámy (American Checker Federation) nasledovne [37]: Chinook 2816, Ron King 2632, Asa Long 2631. Bodovanie predikuje pri 200 bodovom rozdieli 75% pravdepodobnosť výhry.

Aj po víťazstve stroja nad Kasparovom vývoj programov na hranie šachu nezastal. V roku 2002 program Deep Fritz hral proti vtedajšiemu majstrov sveta Vladimírovi Kramikovi (titul získal od Garryho Kasparova v roku 2000). Tento krát už program nevyužíval špeciálny hardvér, ale štvorprocesorový systém [55]. Tento zápas skončil remízou 4:4. Štyri roky po remíze sa Kramik stretol opäť s programom Deep Fritz, tento krát už program využíval "len" prostriedky Dual Intel Core 2 Duo 5160 [55]. Zvýšenie výkonu procesorov a zlepšené algoritmy pomohli softvéru vypočítať približne osem miliónov pozícií za sekundu. Verzia z roku 2002 bola schopná vypočítať 2,7 milióna pozícií za sekundu. Zrýchlenie umožnilo poslednej verzii prehľadávanie do hĺbky 17 až 18 ťahov počas hry. Počítač [54] tento zápas vyhral so skóre 4:2.

2.2 Hry s neúplnou informáciou

Hry s neúplnou informáciou (angl. games of imperfect information) sa vyznačujú tým, že niektoré informácie sú skryté pre niektorých alebo všetkých hráčov. Typickým reprezentantom tejto kategórie sú kartové hry, domino prípadne scrable.

Scrable je síce hrou, kde hráč nepozná, aké hracie kamene majú súperi, ale ukazuje sa, že to nemá veľký vplyv na celkovú schopnosť hrať túto hru. Dôležitejšie je mať perfektne zapamätaný oficiálny slovník a mať schopnosť prehľadávať všetky legálne ťahy. Dnešné programy dokážu poraziť ľudského hráča.

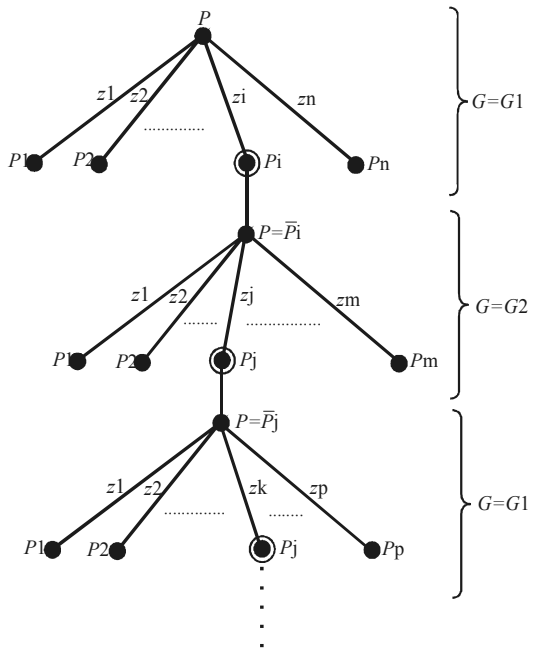
Medzi hrami s neúplnou informáciou sa výskum zameral hlavne na hru poker. Už v roku 1944 teória hier používala poker na testovacie účely. Vtedy von Neumann a Morgenstern [29] modelovali ekonomické prostredie pomocou teórie hier. Zistili, že hráči pokru často „blafujú" a toto správanie je potrebné zahrnúť aj do automatického hráča pokru.

Dnešným najsilnejším počítačovým hráčom pokru je program Poki [1] vytvorený tímom z University of Alberta , ktorý vedie Jonathan Schaeffer. Autori majú ešte veľa práce pred sebou, kým budú schopní súperiť s hráčmi na Svetovom šampionáte v pokri (World Series of Poker), ktorý sa koná každoročne v Las Vegas. Ich výsledky ale vyzerajú veľmi nádejne. Ich práca sa zameriava na dve oblasti. Prvou je stávkovanie počas hry s použitím pravdepodobnostných znalostí na rozhodnutie, akú akciu spraviť (zložiť - vzdať sa, zvýšiť stávku, ukázať karty). Program taktiež používa simuláciu konca hry v reálnom čase, z ktorej získava štatistické hodnoty, ktoré sú použité pri rozhodovacom procese. Druhou oblasťou výskumu je modelovanie oponentov. Tento prístup je založený na fakte, že hráči používajú rôzne štýly hry. Počítačový hráč by mal byť schopný ich rozpoznať a upraviť podľa toho svoje rozhodovanie.

3 Formalizácie hry

V tejto kapitole sa budeme zaoberať formalizáciou hry dámy, ktorá je aplikovateľná pre všetky symetrické hry dvoch hráčov (šach, Go, backgamon — vrhcáby a pod.). Na konci kapitoly uvádzame pravidlá zjednodušenej dámy, ktorú používame ďalej v práci ako testovacie prostredie pre algoritmy.

Nech aktuálna pozícia hry je popísaná premennou P, na túto pozíciu môžu byť aplikované prípustné akcie - ťahy tvoriace množinu $A(P)$. Použitím ťahu $a' \in A(P)$ pretransformujeme pozíciu P na novú pozíciu P', $P \xrightarrow{a'} P'$. Inverznú pozíciu $\overline{P}$ dostaneme z pozície P tak, že biele figúry zameníme za čierne figúry a naopak. Pojem inverznej pozície bude dôležitý pre formuláciu jednotného algoritmu pre symetrické hry, ktorý bude jednotný tak pre prvého ako aj druhého hráča. Použijeme multiagentový prístup, budeme predpokladať, že hru hrajú dvaja agenti G_1 a G_2, ktorí sú vybavení tzv. kognitívnym orgánom, pomocou ktorého sú schopní ohodnocovať jednotlivé nasledujúce pozície. Formulácia algoritmu je nasledovná (pozri obr. 1):



Obr. 1. Diagramatická reprezentácia symetrickej hry pre dvoch hráčov.

Vybraný hráč G z aktuálnej pozície vytvorí všetky možné nasledujúce pozície P_1, P_2, … . Použitím svojho kognitívneho orgánu (reprezentovaného napríklad neurónovou sieťou) ohodnotí každú novú pozíciu číslom z, ako svoj nasledujúci ťah vyberie si tú pozíciu, ktorá má maximálne ohodnotenie. Tento elementárny akt rozhodovania sa opakuje tak, že hráči sa striedajú. Oba hráči majú rovnocenný kognitívny orgán, to znamená, že ich deklarácia na prvého a druhého hráča je náhodná. Z tejto skutočnosti vyplýva, že druhý hráč používa svoj kognitívny orgán na inverzné pozície $\overline{P}$, pristupuje k svojej aktuálnej pozícii ako keby bol prvý hráč.

Algoritmus:

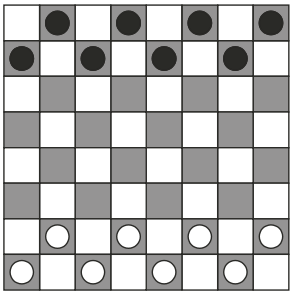
1. Hra je zahájená prvým hráčom, $G \leftarrow G_1$, a počiatočnou pozíciou, $P \leftarrow P_{ini}$.
2. Hráč G vytvorí z pozície P množinu nasledujúcich pozícií $A(P) = \{P_1, P_2, ..., P_n\}$. Každá pozícia P_i z množiny nasledujúcich pozícií je ohodnotená koeficientom $0<z_i<1$. Hráč vyberie za nasledujúcu pozíciu takú $P' \in A(P)$, ktorá je ohodnotená maximálnym koeficientom z, $P \leftarrow P'$. Ak pozícia P je víťazná, potom hráč G víťazí a hra pokračuje krokom 4.
3. Hra prechádza na druhého hráča, $G \leftarrow G_2$, pozíciu P si vytvorí inverziou aktuálnej pozície, $P \leftarrow \overline{P}$, hra pokračuje krokom 2.
4. Koniec hry.

Kľúčovú úlohu v algoritme má výpočet koeficientov $z=z(P')$ pre pozície $P' \in A(P)$. Toto môže byť vykonané buď metódami klasickej umelej inteligencie, ktoré sú založené na kombinácii metód spätného prehľadávania a rôznych heuristík. Našu pozornosť usmerníme na moderný prístup multiagentových systémov, ktorý sa zakladá na predpoklade, že správanie sa agenta v prostredí a/alebo vykonávanie určitých činností je plne determinované jeho kognitívnym orgánom, ktorý vykazuje určitú plasticitu (t.j. je schopný učenia). Kognitívny orgán agenta G je obvykle numericky realizovaný parametrickým zobrazením $G(w):P \rightarrow R$, kde w je parameter (parametre) zobrazenia a ktoré priradzuje každej pozícii (prostrediu) reálne číslo z, formálne $z=G(P,w)$. Zmenou w menia sa vlastnosti kognitívneho orgánu (t.j. dostávame iné parametrické zobrazenie), týmto spôsobom máme zabezpečenú plasticitu kognitívneho orgánu, čo je nutná podmienka procesu učenia agenta v danom prostredí, v procesu ktorého stále lepšie a lepšie zvláda požadovanú úlohu. Vo všeobecnosti parametrické zobrazenie $G(w):P \rightarrow R$ môže byť realizované mnohými rôznymi spôsobmi, od ktorých sa požaduje len splnenie podmienky „univerzálneho aproximátora“. Táto všeobecná podmienka je automaticky splnená v rámci požadovanej presnosti pre mocninné rozvoje alebo pre neurónové siete s dopredným šírením signálu a aspoň s jednou vrstvou skrytých neurónov.

Budeme predpokladať, že agent je schopný svojím kognitívnym orgánom ohodnocovať vznikajúce nasledujúce pozície tak v prípade, že je prvý, alebo aj druhý hráč. Tento predpoklad je plauzibilný pre symetrické hry, kde strategické pravidlá tak pre prvého, ako aj druhého hráča sú rovnaké, určitý malý rozdiel však existuje v tom, že hru zahajuje prvý hráč a tak môže v určitom rozsahu vnucovať štýl hry druhému hráčovi. Vo vyššie uvedenom algoritme je táto požiadavka univerzálnosti kognitívneho orgánu (t.j. agent nemá dva kognitívne orgány, jeden pre prípad, že je prvým hráčom a iný pre prípad, že je druhým hráčom) realizovaná tým, že druhý hráč svojím kognitívnym orgánom neohodnocuje priamo pozície $P' \in A(P)$ ale ich inverzie $\overline{P}'$, t.j. agent aj ako druhý hráč ohodnocuje nasledujúce pozície z pohľadu prvého hráča, čiže kognitívny orgán je nastavený na to, že agent je akoby vždy prvým hráčom, v prípade, že je druhým hráčom, „ohodnocuje“ inverzné pozície.

3.1 Pravidlá zjednodušenej dámy

V tejto kapitole opíšeme pravidlá zjednodušenej dámy, ktorá bude použitá ako testovacie prostredie pri väčšine metód učenia neurónových sietí v nasledovných kapitolách. Zjednodušená dáma sa hrá na hracej ploche s 8×8 políčkami. Na začiatku hry je na hracej ploche rozdelených 8 bielych a 8 čiernych figúrok podľa obrázku 2. Počas hry hráči striedavo ťahajú svojimi figúrkami o jedno políčko dopredu vľavo alebo vpravo, ak je toto políčko voľné. Ak je toto políčko obsadené oponentovým kameňom, musí ho hráč preskočiť a tak vylúčiť z hry. Ak má hráč viac možností vylúčenia súperovej figúrky, môže si vybrať ktorýkoľvek. Hra končí víťazstvom toho hráča, ktorý sa prvý dostane na jedno zo štyroch políčok na kraji súperovej strany šachovnice. Toto predstavuje zmenu oproti známym pravidlám dámy (v klasických pravidlách by tu hráč získal dámu). Koncové časti hry, kedy je na hracej ploche malý počet figúrok, by boli pre neurónovú sieť ťažko riešiteľné, a preto sme ich z hry takto vylúčili. Ťažkosť riešenia pre neurónovú sieť spočíva pravdepodobne v tom, že riešenie koncovej časti hry si vyžaduje plánovanie veľa ťahov dopredu.



Obr. 2. Počiatočná pozícia zjednodušenej hry dámy

Ako protihráča sme pri našich pokusoch používali MiniMax algoritmus opísaný v nasledovnej kapitole. Tomu sme pri rôznych pokusoch menili hĺbku prehľadávania (orezali sme strom v určitej hĺbke a na listy stromu sme aplikovali heuristickú funkciu), a tak sme menili kvalitu jeho hry. Použili sme nasledovnú heuristickú funkciu:

Ak označíme počet našich figúrok na hracej ploche l a súperových m, potom môžeme $y_n[i]$ označiť y-ovú (os smerom k súperovej časti hracej plochy $1\ldots8$) pozíciu i-tej našej figúrky na šachovnici a $y_s[i]$ súperovej. Ohodnotenie šachovnice môžeme zapísať nasledovne:

$$hodnotenie = \begin{cases} 100 & \text{ak hráč vyhral} \\ -100 & \text{ak hráč prehral} \\ \sum_{i=1}^{l} y_n[i] - \sum_{j=1}^{m} (8 - y_n[i]) & \text{inak} \end{cases}$$

Toto ohodnotenie zaručuje, že MiniMax sa bude snažiť dostať svoje figúrky na súperovu stranu hracej plochy a bude sa snažiť zabrániť súperovi v postupe. Hra nášho hráča je dosť útočná.

4 Porovnanie prístupov umelej inteligencie

Dnešná umelá inteligencia ponúka dva rôzne prístupy riešenia problémov [30]:

- klasické prístupy v umelej inteligencii (symbolické prístupy) - tieto presadzujú symbolickú reprezentáciu vedomostí a ich sekvenčné spracovanie, nachádzajú uplatnenie v problémoch, ako je spracovanie prirodzeného jazyka, v problémoch plánovania procesov alebo explicitného usudzovania;

- moderné prístupy v umelej inteligencii (subsymbolické prístupy) - tieto sú založené vo veľkej miere na neurónových sieťach a evolučných algoritmoch. Vo väčšine prípadov používajú nesymbolickú reprezentáciu vedomostí a sú vhodne použiteľné v oblastiach rozpoznávania vzorov, simulácie pamäte, prípadne v takzvaných nízkoúrovňových percepčných procesoch. Tieto moderné prístupy predstavujú systém umelej inteligencie ako dynamický systém.

Každý z týchto prístupov má určite svoje výhody. V tejto kapitole sa budeme snažiť opísať tieto dva prístupy z pohľadu použitia pri strojovom hraní hier. Vhodnosť ich použitia závisí od množstva znalostí, ktoré máme o riešenom probléme. Ak problém dostatočne poznáme a sme schopní vytvoriť napríklad množinu pravidiel, ktorá rieši tento problém, je vhodné použiť symbolickú umelú inteligenciu. V prípade, že riešený problém je ťažko opísateľný, oplatí sa vyskúšať niektorý z prístupov subsymbolickej umelej inteligencie [25].

4.1 Symbolický prístup

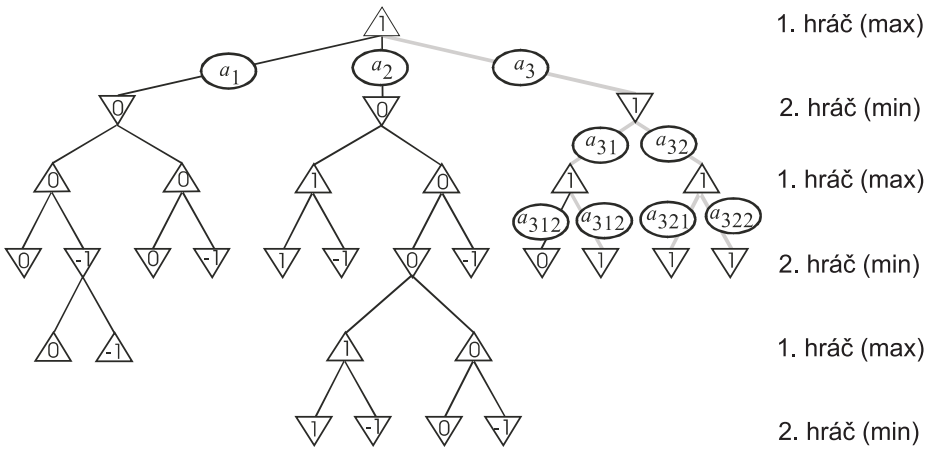
Jedným zo základných algoritmov riešenia problému hry klasickej umelej inteligencie je prehľadávanie stromov riešenia algoritmom MiniMax [33, 30]. Uvádzame ho tu aj preto, lebo ho používame ďalej v kapitole ako referenčný algoritmus pri hre proti neurónovej sieti. Jednoduché hry sa dajú riešiť aj použitím množín pravidiel, prípadne použitím znalostných systémov.

Algoritmus MiniMax

Hranie hry sa dá formalizovať ako postupnosť stavov hry (sekcia 3). Stav hry môže byť napríklad rozostavenie figúrok na šachovnici. V každom stave môže hráč vykonať konečný počet ťahov, a teda zmeniť aktuálny stav na stav nový.

Skutočnosť konečného počtu ťahov v každom stave nám dáva možnosť vytvoriť úplný strom hry. Takýto strom má ako koreň počiatočný stav hry a listy predstavujú konečné stavy. Listy sa ohodnotia podľa toho, či hráč vyhral, prehral alebo hra

skončila remízou. Ak by sme boli schopní vygenerovať pre hru úplný strom, boli by sme schopní nájsť v tomto strome cestu, ktorá by viedla až k nášmu víťazstvu alebo k najlepšiemu možnému výsledku aj v prípade optimálnej hry protihráča. Algoritmus by sa snažil nájsť cestu v grafe všetkých možných hier tak, že by predpokladal, že oponent bude vykonávať vždy najlepšie ťahy. Takto navrhnutý algoritmus by dokázal robiť dokonalé rozhodnutia. Príkladom takého algoritmu je algoritmus MiniMax (znázornený na obrázku 3):



Obr. 3. Strom riešení hypotetickej hry, kde 1. hráč je reprezentovaný trojuholníkom (2. hráč opačným trojuholníkom). Koncové pozície sú ohodnotené podľa toho, či prvý hráč vyhral 1, prehral -1, alebo remizoval 0. Idúc zdola nahor ohodnocujeme vrcholy stromu riešení tak, že pre 1. hráča berieme maximálnu (pre 2. hráča minimálnu) hodnotu susedných dolných vrcholov umiestnených nižšie. V prípade, že horný vrchol stromu je ohodnotený 1, potom existujú také ťahy 1. hráča, že vyhrá bez ohľadu na ťahy druhého hráča (označené na obrázku hrubou čiarou). Prebrané z [20].

Algoritmus MiniMax:

- vygenerujú sa všetky stavy, ktoré môžu počas hry z aktuálneho stavu nastať — vytvorí sa strom, ktorý reprezentuje všetky možné dokončenia rozohranej hry,
- rozhodne sa, ktorý nasledujúci stav je najlepší:
  - listy vygenerovaného stromu sa ohodnotia bodovacou funkciou,
  - postupne sa ohodnotia stavy na vyšších úrovniach tak, že vrcholy, v ktorých je na ťahu *max,* sa ohodnotia ako maximálne ohodnotenie jeho nasledovníkov a vrcholy, v ktorom je na ťahu *min*, sa ohodnotia ako minimum z jeho nasledovníkov,
  - najlepší nasledujúci ťah je ten, ktorý maximalizuje hodnotu koreňa stromu,
- vykoná sa ťah, ktorý vedie k najlepšiemu nasledujúcemu stavu.

V tabuľke 1 [12] uvádzame veľkosti prehľadávacích stromov pre rôzne hry. Z prezentovaných hodnôt vyplýva, že na dnešných výpočtových prostriedkoch nie je možné vygenerovať celý strom.

Tabuľka 1. Veľkosti prehľadávacích stromov pre rôzne hry [12]

Hra	Počet uzlov
Tic-Tac-Toe	10^5
Dáma	10^{31}
Šach	10^{123}
Go	10^{360}

Preto je vhodné použiť nejaké orezávanie stromu. Najjednoduchším riešením je orezávanie do určitej hĺbky stromu. Potom je potrebné listy tohto stromu ohodnotiť nejakou heuristickou funkciou. Heuristická vyhodnocovacia funkcia odhaduje, aké dobré je dané postavenie na hracej ploche, a teda či vedie k víťazstvu. Takto upravený algoritmus už nedokáže robiť dokonalé rozhodnutia, ale stále dokáže robiť dobré rozhodnutia. Kvalita ohodnocovacej funkcie silno ovplyvňuje celkový výkon programu. Čím je hĺbka stromu väčšia, tým sú rozhodnutia lepšie. Ďalším významným zdokonalením je α-β orezávanie. Jeho zlepšenie spočíva v tom, že algoritmus neprehľadáva tie vetvy stromu, ktoré už nemôžu zlepšiť riešenie. Toto orezávanie dokáže znížiť počet potrebných prehľadávaní z N na $\sqrt{N}$ (tabuľka 2).

Tabuľka 2. Výpočet časovej náročnosti vyriešenia hry šach

Počet možností	10^{46}
Počet možností vďaka α-β orezávaniu	10^{23}
Počet pozícií preskúmaných za sekundu	10^9
Úplné vyriešenie šachu	10^{14} sekúnd $\approx 10^4$ tisícročí

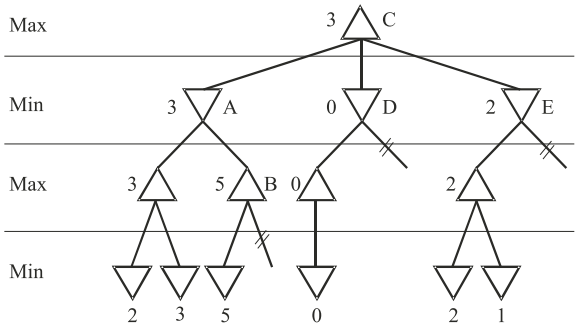
Algoritmus MiniMax je vhodný, ak ohodnotenia získané heuristickou funkciou odrážajú pravdepodobnosť, že daný ťah vedie k víťazstvu. Predpoklad je založený na tom, že heuristická funkcia sa použije na listoch vygenerovaného stromu, a teda bližšie ku koncu hry. Heuristická funkcia je často len hrubým odhadom výhodnosti danej pozície. A práve preto sa môže stať, že algoritmus MiniMax vyberie nie práve najlepší ťah.

Alfa-beta usekávanie

Možnou optimalizáciou algoritmu MiniMax je alfa-beta usekávanie [30], ktoré pracuje s dvoma prahovými hodnotami:

- α je dolné ohraničenie hodnoty, ktorú môže vôbec nadobudnúť stav, v ktorom je na ťahu *max*,
- β je horné ohraničenie hodnoty, ktorú môže vôbec nadobudnúť stav, v ktorom je na ťahu *min*.

Stratégia alfa-beta hľadania si udržiava tieto dve hodnoty tak, ako sa prezerá cesta a na ich základe usekáva podstrom (teda neprehľadáva ho ďalej) hneď ako zistí, že jeho ohodnotenie je horšie než α alebo β. Hodnoty α a β vytvárajú určitý interval, v ktorom je pre nás prehľadávanie ešte zaujímavé. Hodnota α zastaví všetky prehľadávania, ktoré majú príliš nízke ohodnotenie a hodnota β zastaví tie s príliš vysokým ohodnotením.



Obr. 4. Alfa-beta usekávanie stromu prehľadávania
A má $\beta = 3$ (ohodnotenie uzla A nebude väčšie ako 3)
B sa usekne, lebo $\alpha_B \geq \beta_A (5 \geq 3)$
C má $\alpha = 3$ (ohodnotenie uzla C nebude menšie ako 3)
D sa usekne, lebo $\beta_D \leq \alpha_C (0 \leq 3)$
E sa usekne, lebo $\beta_E \leq \alpha_C (2 \leq 3)$
ohodnotenie C je 3

Hodnoty α a β sa môžu v určitom uzle zmeniť až v okamihu, kedy nasledovníci tohto uzla majú pridelené MiniMaxové ohodnotenia. Pravidlá pre zmenu sú nasledovné (α je inicializovaná na $-\infty$, β na $+\infty$):

- hodnota Max uzla (vrátane koreňa) sa určí ako súčasná najväčšia MiniMaxová hodnota jeho nasledovníkov,
- hodnota Min uzla sa určí ako súčasná najmenšia MiniMaxová hodnota jeho nasledovníkov (nikdy sa nemôže zväčšiť).

Pravidlá pre prerušenie prehľadávania podstromu sú nasledovné:

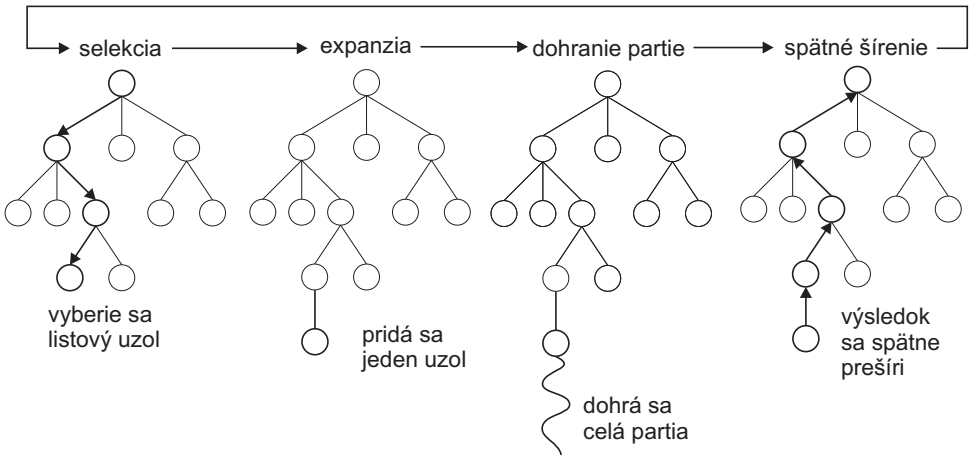
- *alfa usekávanie* - hľadanie sa môže ukončiť pod každým Min uzlom, ktorého β hodnota nie je väčšia ako α hodnota ľubovoľného z jeho predchodcov (uzly D a E na obrázku 4); výsledná hodnota tohto Max uzla je jeho α hodnota,
- *beta usekávanie* - hľadanie sa môže ukončiť pod každým Max uzlom, ktorého hodnota α nie je menšia ako β ľubovoľného z jeho predchodcov (uzol B na 4); výsledná hodnota tohto Min uzla je jeho β hodnota.

Algoritmus MiniMax s alfa-beta usekávaním dáva rovnaké rozhodnutia ako bez neho. Jeho výhodou je zmenšenie počtu prehľadávaných stavov, čím sa znižuje časová aj pamäťová náročnosť algoritmu.

Prehľadávanie stromu metódou Monte-Carlo

V poslednom čase sa Monte-Carlo prehľadávanie stromu (angl. Monte-Carlo tree search - MCTS) stalo populárnym prístupom v počítačovom hraní hier. Najúspešnejšie bola táto metóda použitá v programoch hrajúcich hru Go [14].

Monte-Carlo prehľadávanie stromu hry je prehľadávanie formou najskôr najlepší (angl. best-first) [13], ktoré nepotrebuje heuristickú funkciu na ohodnocovanie šachovníc. Je založené na náhodnom prehľadávaní herného stromu s použitím výsledkov z predchádzajúcich behov. Algoritmus postupne vytvára, zväčšuje a ohodnocuje herný strom, a tak postupne zlepšuje svoj odhad ohodnotenia najvhodnejších ťahov. Algoritmus prebieha v štyroch krokoch (obrázok 5):



Obr. 5. Postupnosť krokov pri prehľadávaní stromu s pomocou Monte-Carlo metódy

1. *krok výberu uzla (selekcia)*
   Selekcia vyberá uzol stromu, ktorý sa bude ďalej skúmať na základe skôr získaných skúseností. Zabezpečuje rovnováhu medzi preskúmavaním už známych uzlov, ktoré sú ohodnotené najlepšie (exploitation) a preskúmavaním neznámych uzlov (exploration).

2. *krok expanzie*
   Do stromu sa pridá uzol, ktorý bude prvým počas dohrávania partie.

3. *krok dohrania partie (angl. play-out)*
   Dohranie partie prebieha ako simulované ťahanie do konca partie (sám proti sebe). Ťahanie môže byť náhodné, aj keď podľa [13] je lepšie použiť pseudonáhodné ťahanie ovplyvnené heuristickou funkciou.

4. *krok spätného šírenia*
   Dohranie hry sa ohodnotí podľa výsledku hry (1 - výhra, 0 - remíza, -1 - prehra). Toto ohodnotenie sa potom prešíri od pridaného uzla v kroku expanzie cez prejdené uzly až ku koreňu stromu. Ohodnotenie uzla je priemer ohodnotení všetkých hier ktoré prechádzali cez tento uzol.

V roku 2008 program MoGo Titan porazil profesionálneho hráča Go (8 dan) počas hry na doske 19×19 pri hendikepe 9 kameňov. Program používal 800 jadier superpočítača Huygens. V tom istom roku sa podarilo programu Crazy Stone vyhrať s hendikepom 8 kameňov nad profesionálnym hráčom Go (4 dan). CrazyStone používal len 8 jadier dvojprocesorového počítača. Podľa [14] víťazstvo počítača nad človekom bez hendikepu na šachovnici 19×19 je ešte dosť vzdialené, ale počítače prekonajú človeka na šachovnici 9×9 už v krátkom čase.

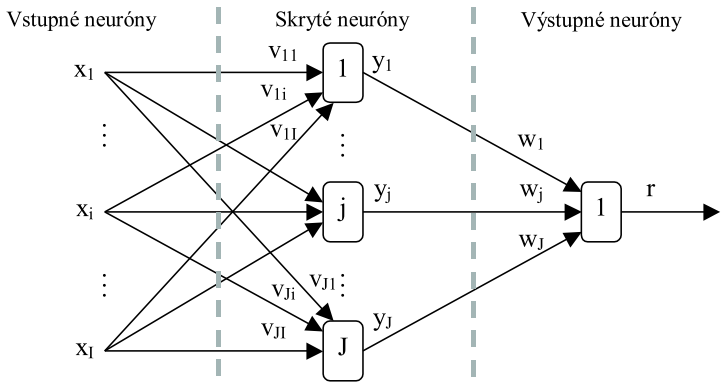
4.2 Subsymbolický prístup

Subsymbolický prístup sa snaží riešiť práve problémy, s ktorými má symbolický prístup ťažkosti. Je tvorený novými prístupmi v umelej inteligencii, ako neurónové siete a evolučné algoritmy.

Neurónové siete

Neurónové siete modelujú biologické neurónové siete (obrázok 6) [21]. Sú zložené z neurónov a spojení medzi nimi. Základnou požiadavkou na neurónové siete je plasticita, teda schopnosť prispôsobovania, učenia. Plasticita je špecifikovaná dvoma spôsobmi [34]:

- štrukturálna plasticita, ktorá znamená, že sa mení počet skrytých neurónov (t.j. mení sa štruktúra - topológia neurónovej siete),
- parametrická plasticita, ktorá zodpovedá zmene váh spojov a prahových koeficientov skrytých a výstupných neurónov.



Obr. 6. Trojvrstvová dopredná umelá neurónová sieť v ktorej sa učí skrytá a výstupná vrstva

Umelé neurónové siete môžu byť učené s učiteľom alebo bez neho. Učenie s učiteľom potrebuje trénovaciu množinu dvojíc vstup-výstup (v našom prípade šachovnica-ohodnotenie). Zaujímavé výsledky dosiahol Wu [51] pri použití rekurentnej neurónovej siete s učením s učiteľom pri hre Go. Najznámejšou ukážkou úspešnosti učenia bez učiteľa je učenie s odmenou a trestom (podrobnejšie sa mu budeme venovať v nasledujúcich kapitolách). Jednou z najznámejších aplikácií učenia s odmenou a trestom je Tesaurov program [43, 44, 45] na hranie hry backgammon. Tesaurov program TD-Gammon vyžadoval len malé množstvo znalosti hry backgammon a dokázal sa naučiť túto hru extrémne dobre. Kvalita hrania sa blížila k úrovni najlepších hráčov na svete. Učiacim algoritmom v TD-Gammone bola kombinácia algoritmu TD(λ) a nelineárnej aproximácie funkcie použitím viacvrstvovej neurónovej siete trénovanej spätným šírením chyby.

Po odohraní 300 000 hier, kedy TD-Gammon (so 40 skrytými neurónmi) hral sám proti sebe, bol schopný hrať asi na rovnakej úrovni ako ostatné programy. Vylepšený TD-Gammon, ktorý mal 160 skrytých neurónov a prehľadával strom hry 3 ťahy dopredu (čo je už vlastne kombinácia symbolického a subsymbolického prístupu), dosahoval už majstrovské výsledky. Jeho výsledky sa veľmi blížia k hracej sile najlepších ľudských hráčov. Tieto programy už dokonca zmenili spôsob hrania najlepších ľudských hráčov. Napríklad TD-Gammon sa naučil hrať niektoré otvorenia inakšie, ako boli dovtedajšie konvencie medzi najlepšími hráčmi. Po ďalších analýzach začali aj ľudskí hráči hrať tieto otvorenia.

Adaptácia kognitívneho orgánu pomocou genetického algoritmu

Genetický algoritmus je stochastická metóda optimalizácie [22]. Jeho základnou myšlienkou je napodobnenie biologických procesov evolúcie. Hlavným princípom Darwinovej teórie evolúcie je téza prirodzeného výberu, ktorá hovorí, že len najlepšie prispôsobení jedinci populácie prežívajú. Reprodukciou dvoch jedincov s vysokým *fitness* (ohodnotenie prispôsobenia jedinca prostrediu) vzniká jedinec, ktorý bude mať s veľkou pravdepodobnosťou tiež veľký *fitness*. Cieľom evolúcie je vznik jedinca s maximálnym možným *fitness* v danom prostredí.

Genetický algoritmus prebieha nad množinou jedincov, ktorí tvoria populáciu. Každý jedinec je ohodnotený pomocou jeho *fitness*, čo je hodnota, ktorá vyjadruje jeho úspešnosť v danom prostredí. Genetický algoritmus prebieha v nasledujúcich krokoch (existujú rôzne úpravy výberu jedinca, kríženia a mutácie jedincov, viac v [22]):

- kvázináhodným výberom pomocou rulety sa vyberú dvaja rodičovskí jedinci. Výber ruletou zabezpečuje vyššiu pravdepodobnosť vstupu do reprodukcie jedincom s vyšším *fitness*. Pravdepodobnosť výberu jedinca do reprodukcie je:

$$p_i = \frac{f_i}{\sum_j f_j}$$

- kde p_i reprezentuje pravdepodobnosť výberu jedinca i do reprodukcie, f_j je fitness jedinca j.

- reprodukcia prebieha len s určitou pravdepodobnosťou. Ak jedinci neprechádzajú reprodukciou, vracajú sa nezmenení do populácie. Reprodukcia prebieha v dvoch krokoch — kríženie a mutácia:
    - kríženie *je výmena častí chromozómov* jedincov medzi sebou. Pri krížení sa vyberie bod kríženia v ktorom sa chromozóm rozdelí a rodičia si navzájom vymenia časti chromozómu, a tak vzniknú potomkovia
    - mutácia je zmena časti chromozómu jedinca s určitou pravdepodobnosťou (pri binárnom kódovaní chromozómu zmena 1 na 0 a 0 na 1, pri reálnom kódovaní pripočítanie náhodného čísla s normálnym rozdelením)
    - vrátenie nových potomkov do populácie.

Subsymbolický prístup bol úspešne použitý aj pri hraní dámy. Evolučné hranie dámy vytvoril Kumar Chellapilla a David B. Fogel [3, 6]. Tento prístup bol založený na evolučne optimalizovaných neurónových sieťach. Neurónová sieť ohodnocovala hracie plochy. Toto ohodnotenie bolo potom spracované algoritmom MiniMax, ktorý bol zodpovedný za výber ťahu. Úprava váh prebiehala evolučne, teda populácia sietí hrala turnaje hier. Výsledok turnaja siete predstavoval jej *fitness*. Po odohraní turnajov sa jedinci s najväčším *fitness* reprodukovali. Počiatočná populácia mala váhové vektory neurónovej siete nastavené na náhodné hodnoty.

Pri testoch použili autori doprednú sieť s 2 skrytými vrstvami obsahujúcimi 40 a 10 neurónov a jedným výstupným neurónom. Hĺbka prehľadávania MiniMax algoritmom bola 4. Potom vypočítali 100 generácií sietí a z tejto populácie vybrali najlepšiu sieť. S touto sieťou potom hrali proti živým protivníkom na internetovej stránke www.zone.com, ktorý poskytoval funkciou ohodnotenia hráča. Neurónová sieť dosiahla po 100 hrách ohodnotenie 1 825, čo je dobrý výsledok, keďže počiatočné ohodnotenie siete bolo 1 600. Sieť teda bola schopná poraziť živých hráčov, ktorí nevedeli, že hrajú so strojom.

Celkový výsledok dokazuje, že evolučný algoritmus sa dokáže naučiť hrať stratégiu hry. Autori uvádzajú, že zvýšenie hĺbky prehľadávania a heuristický prístup k algoritmu MiniMax by podstatne vylepšil hracie možnosti tohto algoritmu.

5 Učenie s odmenou a trestom

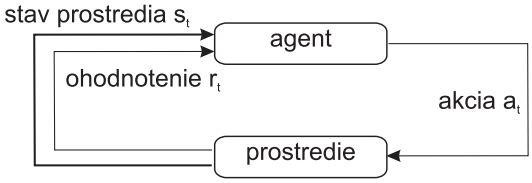
Táto kapitola opisuje učenie s odmenou a trestom vo všeobecnosti. Väčšina ďalej odvodených metód pre neurónové siete používa práve tento prístup k učeniu. Učenie s odmenou a trestom [10, 41] spája dve disciplíny počítačovej vedy na to, aby úspešne riešilo problémy, ktoré ani jedna disciplína sama nevyrieši. Spája dynamické programovanie, ktoré sa tradične požíva na riešenie úloh optimalizácie a riadenia, s prístupom učenia s učiteľom, ktoré sa používa pri trénovaní parametrizovateľných aproximátorov funkcií (napríklad neurónových sietí ako bolo opísané v sekcii 4.2). Učenie s odmenou a trestom bolo úspešne použité pri strojovom hraní hier ako backgammon [43, 44, 45], Othello (reversi) [27], šach [8] a Go [36].

Učenie s učiteľom má jednu základnú nevýhodu, ktorou je potreba párov podnet a odpoveď (trénovacia množina). Počas celého trénovania učiteľ pozná správnu odpoveď na daný vstup. Učiaci algoritmus si dokáže vypočítať chybu, ktorá ovplyvňuje zmenu váh siete. Exitujú ale úlohy, pre ktoré nepoznáme trénovaciu množinu. Učenie s odmenou a trestom rieši práve tento problém. Agentovi, ktorý sa nachádza v nejakom prostredí nehovoríme, aké akcie má vykonávať, len mu dávame spätnú väzbu. Spätnou väzbou je ohodnotenie ako dobre sa v danom prostredí správa. Môže to byť agent, ktorý hrá hru a nedostáva od učiteľa správne ťahy na momentálne situácie. Agent je ohodnotený počas hry, alebo na jej konci podľa toho, ako dobre hral.

5.1 Vlastnosti učenia s odmenou a trestom

Učenie s odmenou a trestom je inšpirované výskumom v oblasti psychológie, keď už v roku 1911 Edward Thorndike [34] formuloval zákon účinku (angl. law of effect) a zákon opakovania (angl. law of exercise). Zákon účinku hovorí, že akcia, po ktorej prichádza odmena, sa asociuje s danou situáciou a v budúcnosti bude použitá s väčšou pravdepodobnosťou v danej situácii. Akcii nasledovanej trestom sa väzba na danú situáciu oslabuje, a teda v budúcnosti bude menej používaná. Učenie prebieha ako opakované (zákon opakovania) posilňovanie/oslabovanie väzieb medzi situáciou a akciou.

Pri štandardnom prístupe učenia s odmenou a trestom agent interaguje s prostredím (obrázok 7). V každom kroku agent prijíma stav prostredia $s_t \in S$ vo forme senzorických vstupov (S je množina všetkých stavov prostredia). Na základe týchto vstupov agent vyberá vhodnú akciu $a_t \in A(s_t)$, kde $A(s_t)$ je množina všetkých možných akcií v stave s_t. Akcia agenta sa vykoná v prostredí a nejakým spôsobom ho zmení na stav s_{t+1}. V ďalšom kroku agent môže dostať odmenu $r_{t+1} \in \Re$ a na vstupe má nový stav s_{t+1}.



Obr. 7. Interakcia agenta s prostredím

Učenie s odmenou a trestom sa skladá z troch častí — prostredie, funkcia odmeny a funkcia ohodnotenia.

Prostredie

Pri učení s odmenou a trestom sa systém učí mapovanie zo situácií na akcie pomocou interakcie cez pokusy a omyly v dynamickom prostredí. Toto prostredie musí byť aspoň z časti pozorovateľné agentom. Vnemy z prostredia môžu byť vo forme informácií zo senzorov, symbolických opisov alebo „pocitov“ (napríklad situácia keď

je agent stratený). Akcie môžu byť nízkoúrovňového charakteru (ako napätie na motoroch), vysokoúrovňové (prijatie pracovnej ponuky). Ak sú vnemy systému dokonalé, teda systém dokáže získať všetky informácie potrebné na rozhodnutie, systém vyberá akcie na základe stavu prostredia.

Funkcia ohodnotenia

Ako už bolo uvedené, pri učení s odmenou a trestom sa systém učí mapovanie zo stavu na akciu. Cieľ systému je určený pomocou funkcie ohodnotenia, ktorá ohodnocuje správanie agenta. Agent sa snaží maximalizovať svoju sumu ohodnotení.

Existuje teda mapovanie medzi stavom a akciou na funkciu ohodnotenia. Po tom ako agent vykoná akciu v nejakom stave systému, dostáva ohodnotenie vo forme odmeny ako skalárnej hodnoty. Pri vytváraní systému s učením s odmenou a trestom sa musí navrhnúť dobrá funkcia ohodnotenia. Táto musí dobre definovať ciele agenta. Ďalej uvádzame dva možné prístupy k tvorbe funkcie ohodnotenia - problém s oneskorenou odmenou a problém minimalizovania času k cieľu.

Pri problémoch s oneskorenou odmenou je ohodnotenie počas celého behu nulové až na konečný stav. Skalárne ohodnotenie je kladné (odmena), ak cieľový stav je požadovaným cieľom alebo záporné (trest), ak je potrebné sa konečnému stavu vyhýbať. Ak napríklad chceme naučiť agenta hrať hru dáma, systém by mohol byť definovaný nasledovne. Rozmiestnenie figúrok na šachovnici predstavuje stav prostredia agenta. Možné akcie agenta sú ťahy, ktoré môže vykonať. Funkcia ohodnotenia bude nulová počas celej hry až do jej skončenia. V prípade, že agent vyhrá, dostane ohodnotenie +1 a ak prehrá tak –1. Keďže sa agent snaží maximalizovať svoje ohodnotenie, bude sa správať tak, aby stavy, v ktorých vyhráva, boli jeho cieľové stavy a stavy, v ktorých prehráva, boli stavy, ktorým sa ma vyhýbať.

Pri problémoch minimalizovania času k cieľu vedie ohodnocovacia funkcia agenta k tomu, aby bol schopný vyriešiť problém v najkratšom čase, prípadne nájsť najkratšiu cestu. Ako príklad uvedieme experiment známy ako problém auta na kopci (z ang. Car on a hill problem). Auto sa nachádza v údolí medzi dvoma strmými svahmi. Cieľom agenta je dostať sa s autom na vrchol pravého svahu, a tak sa dostať do požadovaného konečného stavu. Stavom prostredia je pozícia a rýchlosť auta. Možné sú tri akcie — ťah dopredu, ťah dozadu a žiaden ťah motora. Prostredie je definované tak, že auto nemá dostatočný ťah, aby sa priamo dostalo na pravý vrch, a teda musí sa využiť hybnosť (zotrvačnosť), aby bolo možné dosiahnuť cieľ. Odmena je stále –1 s výnimkou dosiahnutia požadovanej pozície na pravom vrchu, kde nadobúda hodnotu 0. Keďže sa agent snaží maximalizovať svoju odmenu, volí také akcie, ktoré vedú v najkratšom čase k požadovanému cieľu.

Funkcia hodnoty

Najprv si definujeme dva pojmy - *stratégia* a *hodnota*. *Stratégia* určuje správanie agenta v prostredí, definuje aká akcia sa má vykonať v danom stave. Je to teda mapovanie stavu na akcie. Formálne môžeme stratégiu π zapísať ako zobrazenie stavu $s \in S$ a akcie $a \in A(s)$ na pravdepodobnosť $\pi(s,a)$ výberu akcie a v stave s. Stratégiu si agent počas učenia upravuje tak, aby dosahoval najlepšie ohodnotenia.

Optimálnou stratégiou π^* voláme takú, ktorá je lepšia alebo rovnako dobrá ako všetky ostatné.

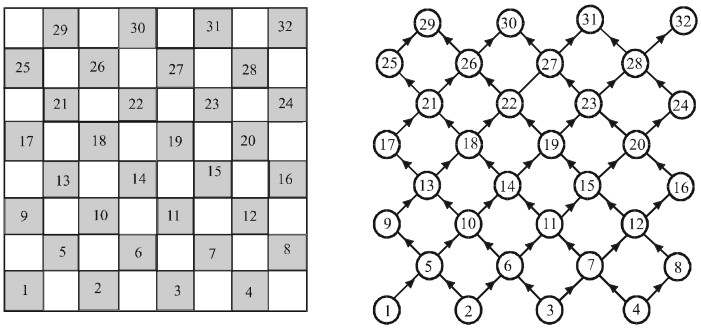
Hodnota stavu je definovaná ako súčet všetkých ohodnotení, ktoré dostane agent, ak začne v danom stave s a bude sa riadiť podľa nemeniacej sa stratégie π až po koncový stav. Funkciu hodnoty zapisujeme $V^{\pi}(s)$, je mapovaním medzi stavom a hodnotou. Môže byť aproximovaná ľubovoľným funkčným aproximátorom (napr. viacvrstvový perceptrón alebo tabuľka). Optimálna stratégia je teda taká, ktorá mapuje stavy na akcie tak, že súčet ohodnotení je maximálny pri štarte z ľubovoľného stavu a vykonávaní akcií až po dosiahnutie koncového stavu. Otázkou ostáva, ako nájsť algoritmus, ktorý spoľahlivo nájde funkciu hodnoty. Metódam hľadania sa budeme venovať v ďalších sekciách.

6 Adaptácia dopredných neurónových sietí

V tejto sekcii opíšeme klasický prístup k adaptácii umelých neurónových sietí a odvodíme TD(λ) pravidlo pre učenie s odmenou a trestom.

6.1 Štruktúra kognitívneho orgánu - neurónovej siete

Prv ako pristúpime k špecifikácii kognitívneho orgánu agentov, musíme zaviesť tzv. numerickú reprezentáciu pozícií. Pozícia je pre hru dáma reprezentovaná 32-rozmerným vektorom (obrázok 8):

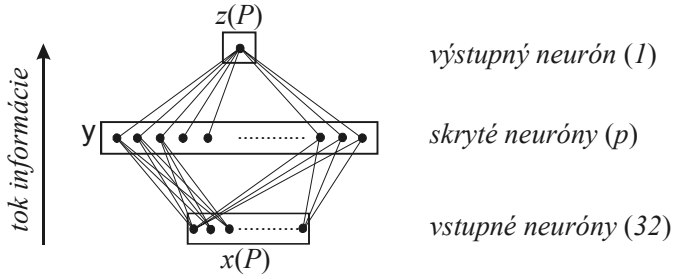


Obr. 8. Ľavý obrázok znázorňuje indexovanie čiernych políčok hracej dosky. Pravý obrázok vyjadruje orientovaný graf, ktorý je použitý pre jednoduchú špecifikáciu nasledujúcej figúrky pri ťahu.

$$x(P) = (x_1, x_2, \ldots, x_{32}) \in \{0,1,-1\}^{32} \tag{1a}$$

kde jednotlivé zložky určujú jednotlivé políčka v pozícii P

$$x_i = \begin{cases} 0 & (i - \text{té pole je neobsadené}) \\ 1 & (i - \text{té pole je obsadené našou figúrkou}) \\ -1 & (i - \text{té pole je obsadené súperom}) \end{cases} \tag{1b}$$



Obr. 9. Dopredná (feed forward) neurónová sieť s jednou vrstvou skrytých neurónov. Vstupné aktivity sú rovné 32-rozmernému vektoru $\boldsymbol{x}(P)$, ktorý kóduje pozície hry. Výstupná aktivita sa rovná reálnemu číslu $z(P)$ z otvoreného intervalu (0,1), toto číslo je ohodnotenie vstupnej pozície.

Použitá neurónová sieť má architektúru typu „doprednej siete“ [11] s jednou vrstvou skrytých neurónov. Aktivity vstupných neurónov sú určené numerickou reprezentáciou $\boldsymbol{x}(P)$ danej pozície P, výstupná aktivita sa rovná ohodnoteniu pozície $\boldsymbol{x}(P)$ (pozri obrázok 9), počet parametrov neurónovej siete je $34p+1$, kde p je počet skrytých neurónov. Plasticita takto špecifikovaného kognitívneho aparátu pomocou neurónovej siete je určená dvoma spôsobmi: (a) **štrukturálna plasticita,** ktorá znamená, že sa mení počet skrytých neurónov (t.j. mení sa štruktúra – topológia neurónovej siete), a (b) **parametrická plasticita**, ktorá odpovedá zmene váh spojov a prahových koeficientov skrytých a výstupných neurónov.

Aktivity skrytých neurónov a výstupného neurónu sú určené vzťahmi

$$y_i = t\left(\sum_{j=1}^{9} w_{ij} x_j + \vartheta_i\right) \quad (i = 1, 2, \ldots, p) \tag{2a}$$

$$z = t\left(\sum_{i=1}^{p} \tilde{w}_j y_j + \tilde{\vartheta}\right) \tag{2b}$$

kde $t(\xi)$ je prechodová funkcia tvaru jednoduchej sigmoidy

$$t(\xi) = \frac{1}{1 + e^{-\xi}} \tag{3}$$

Pri adaptácii neurónových sietí sa obvykle využívajú parciálne derivácie výstupnej aktivity $\lambda(P)$ vzhľadom k parametrom neurónovej siete. Tieto parciálne derivácie vzhľadom k parametrom výstupného neurónu sú určené takto

$$\frac{\partial z(P)}{\partial \tilde{\vartheta}} = z(P)\left[1 - z(P)\right], \quad \frac{\partial z(P)}{\partial \tilde{w}_j} = \frac{\partial z(P)}{\partial \tilde{\vartheta}} y_j \tag{4}$$

Podobným spôsobom sú určené aj parciálne derivácie vzhľadom k parametrom skrytých neurónov

$$\frac{\partial z(P)}{\partial \vartheta_i} = y_i(1-y_i)\frac{\partial z(P)}{\partial \tilde{\vartheta}}\tilde{w}_i, \quad \frac{\partial z(P)}{\partial w_{ij}} = \frac{\partial z(P)}{\partial \vartheta_i}x_j \tag{5}$$

Parciálne derivácie môžeme jednoducho počítať rekurentným postupom nazývaným „metóda spätného šírenia“, najprv sa spočítajú parciálne derivácie pre parametre výstupného neurónu, potom môžeme spočítať parciálne derivácie vzhľadom k parametrom skrytých neurónov.

6.2 Klasický prístup k adaptácii neurónových sietí reprezentujúcich kognitívny orgán agenta

Klasický prístup k adaptácii neurónových sietí je založený na použití tzv. tréningovej množiny, ktorá obsahuje dvojice vektorov vstupných a požadovaných výstupných aktivít (poznamenajme, že vektor vstupných aktivít obvykle je numerickou reprezentáciou objektu klasifikovaného neurónovou sieťou, pričom požadované výstupné aktivity klasifikujú daný objekt)

$$A_{train} = \left\{ \boldsymbol{x} = (x_1, x_2, \ldots, x_n) / z_{req} \right\} \tag{6}$$

To znamená, že neurónová sieť je parametrické zobrazenie *n*-rozmerných vektorov na reálne číslo z otvoreného intervalu (0,1)

$$G(w): R^n \to (0,1) \tag{7}$$

Adaptácia neurónovej siete je realizovaná pomocou *minimalizácie* účelovej funkcie vyjadrujúcej sumu kvadrátov rozdielov medzi vypočítanými a požadovanými výstupnými aktivitami

$$E(w) = \frac{1}{2}\sum_{\mathbf{x}/z_{req}\in A_{train}} \left(z_{req} - G(\mathbf{x}; w)\right)^2 \tag{8}$$

Váhové koeficienty neurónovej siete sú upravované pomocou gradientovej metódy najprudšieho spádu (steepest descent)

$$w = w - \alpha\frac{\partial E}{\partial w} = w + \Delta w \tag{9a}$$

$$\Delta w = \alpha \sum_{\mathbf{x}/z_{req}\in A_{train}} \left(z_{req} - G(\mathbf{x}, w)\right)\frac{\partial G(\mathbf{x}, w)}{\partial w} \tag{9b}$$

kde α je kladný parameter nazývaný „krok učenia“. Výpočet parciálnych derivácií výstupných aktivít vzhľadom k parametrom neurónovej siete je popísaný vzťahmi (4) a (5).

Po všeobecnom úvode o štandardnom spôsobe adaptácie neurónových sietí obrátime našu pozornosť na použitie tohto typu adaptácie neurónových sietí, ktoré slúžia ako kognitívny orgán agenta hrajúceho symetrickú hru pre dvoch hráčov. Množina *R* nech obsahuje všetky možné pozície hry. Z pohľadu daného hráča, pozície z príslušnej tréningovej množiny môžu byť ohodnotené reálnym číslom z_{req} , ktoré špecifikuje či daná pozícia vedie k víťazstvu (z_{req}=1) alebo k prehre (z_{req}=0). Vo

všeobecnosti, hodnotenie pozícií je netriviálna záležitosť a môže byť realizované s ohraničenou presnosťou "expertom", ktorý na základe svojich vedomostí, skúseností a intuície ohodnotí každú pozíciu z tréningovej množiny číslom z_{req}. Žiaľ, tento priamočiary prístup je nerealizovateľný pre hry s podstatne väčším stavovým priestorom (napr. šach, dáma,…), časová náročnosť metódy spätného prehľadávania prudko rastie (exponenciálne) s počtom možných pozícií.

Predpokladajme, že pozície z množiny R máme ohodnotené číslami $z_{\mathrm{req}} \in [0,1]$, potom môžeme zostrojiť tréningovú množinu takto

$$A_{train} = \left\{ \boldsymbol{x} / z_{req} ; x \in P \right\} \tag{10}$$

čo znamená, že každá pozícia z množiny P je ohodnotená reálnym číslom $z_{\mathrm{req}} \in [0,1]$ v závislosti od toho, či pozícia pri optimálnej hre daného hráča vedie buď k víťazstvu, remíze, alebo prehre. Tréningová množina (10) môže byť použitá ako základ pre realizáciu adaptačného procesu neurónovej siete – kognitívneho orgánu, ktorého parametre v priebehu učenia sa nastavia tak, aby kognitívny orgán správne klasifikoval pozície, t.j. predpokladá sa, že výstupná aktivita neurónovej siete aproximuje čísla z

$$z_{req} \approx G\left(\mathbf{x}, w\right) \forall \, \mathbf{x} / z_{req} \in \mathrm{A}_{train} \tag{11}$$

Obrazne povedané, úspešne adaptovaný kognitívny orgán klasifikuje pozície podobne, ako „expert“, pomocou ktorého sme vytvorili tréningovú množinu.

Niekoľko záverečných poznámok k tomuto priamočiaremu použitiu neurónovej siete k implementácii algoritmu pre symetrickú hru dvoch hráčov. Aj keď je koncepčne veľmi jednoduchý a jedná sa o bezprostrednú aplikáciu neurónových sietí k problematike počítačovej implementácie symetrických hier, obsahuje vážnu reštrikciu ako je konštrukcia tréningovej množiny pre účely adaptačného procesu. Ohodnotenie každej pozície číslom λ pomocou „experta“ je realizovateľné s dostatočnou presnosťou len pre hry s relatívne malým stavovým priestorom. Podobne, použitie systematickej metódy „spätného prehľadávania“ je aplikovateľné len pre hry s malým stavovým priestorom, časová náročnosť „spätného prehľadávania“ prudko rastie so zväčšovaním stavového priestoru. Naznačený prístup bol úspešne použitý pre hru backgammon Geraldom Tesaurom [5] z IBM v r. 1989 v programe *Neurogammon*, kde rozsiahla tréningová množina bola vytvorená analýzou mnoho tisíc „majstrovských partií“. Vytvorený program používajúci neurónovú sieť na ohodnotenie pozícií tejto hry patrila svojho času k najlepším programom pre hru backgammon a v r. 1989 vyhral olympiádu programov pre túto hru.

6.3 Adaptácia kognitívneho orgánu agenta pomocou TD(λ)

V tejto kapitole uvedieme základné princípy modernej metódy učenia s odmenou a trestom, ktorá v súčasnosti vo verzii „temporal difference” TD(λ) patrí medzí účinné algoritmické prostriedky adaptácie kognitívnych orgánov multiagentových systémov [30, 40]. Základné princípy učenia s odmenou a trestom sú tieto: Agent sleduje závislosť medzi vstupným obrazcom a výstupným signálom jeho kognitívneho orgánu (ktorý sa často nazýva „akcia” alebo „riadiaci signál”). Na základe externého skalárneho signálu odmeny vyhodnocuje kvalitu výstupného signálu. Cieľom učenia je taká modifikácia kognitívneho orgánu agenta, aby výstupné signály maximalizovali

príjem externých odmeňovacích signálov. V mnohých prípadoch signál odmeny je časovo oneskorený. Prichádza po dlhom slede akcií až na záver a môže byť chápaný ako ohodnotenie celej sekvencie akcií – toho či viedla k požadovanému výsledku alebo nie. V tomto prípade agent musí riešiť tzv. problém „temporal difference" priradenia, kde je učenie založené na diferenciách medzi dočasne vykonanými predikciami pre jednotlivé elementy celkovej sekvencie akcií.

V tejto kapitole naznačíme konštrukciu metódy TD(λ) ako určité zovšeobecnenie štandardnej metódy učenia neurónových sietí popísanej v predchádzajúcej kapitole. Predpokladajme, že poznáme sekvenciu pozícií a jej ohodnotenie reálnym číslom, ktoré odpovedajú pozíciám daného agenta – hráča, ktoré musel ohodnocovať číslom z

$$P_1, P_2, \ldots, P_m, z_{reward} \tag{12}$$

kde z_{reward} je vonkajšie ohodnotenie sekvencie postupnosti a odpovedá skutočnosti či posledná pozícia P_m je pre daného agenta – hráča víťazná, remízová, alebo prehratá.

$$z_{reward} = \begin{cases} 1 & (sekvencia\ pozícií\ je\ vítazná) \\ 0.5 & (sekvencia\ pozícií\ je\ remízová) \\ 0 & (sekvencia\ pozícií\ je\ prehraná) \end{cases} \tag{13}$$

Zo sekvencie (12) vytvoríme m dvojíc pozícií a ich ohodnotenie číslom z_{reward}, vytvoríme nasledujúcu účelovú funkciu

$$E(w) = \frac{1}{2}\sum_{t=1}^{m}\left(z_{reward} - G(\boldsymbol{x}_t; w)\right)^2 \tag{14}$$

Budeme hľadať také váhové koeficienty neurónovej siete - kognitívneho orgánu, ktoré minimalizujú účelovú funkciu. V prípade, že sa nám podarí nájsť také váhové koeficienty siete, pre ktoré je účelová funkcia nulová, potom každá pozícia zo sekvencie (12) je ohodnotená číslom z_{reward}. Rekurentná formula pre obnovu váhových koeficientov má tvar (pozri (9a) a (9b))

$$w = w - \alpha\frac{\partial E}{\partial w} = w + \Delta w \tag{15a}$$

$$\Delta w = \alpha\sum_{t=1}^{m}\left(z_{reward} - z_t\right)\frac{\partial z_t}{\partial w} \tag{15b}$$

kde $z_t = G(P_i, w)$ je ohodnotenie t-tej pozície P_t pomocou neurónovej siete - kognitívneho orgánu číslom z_t. Našim cieľom bude, aby všetky pozície zo sekvencie (12) boli ohodnotené rovnakým číslom z_{reward}, ktoré nám špecifikuje, či hra daného hráča pozostávajúca zo sekvencií (12) bola víťazná, remízová, alebo viedla k prehre. Výraz v zátvorke v (15b) dá sa jednoducho prepísať do nasledujúceho tvaru

$$\begin{aligned} z_{reward} - z_t &= z_{m+1} - z_t \\ &= z_{m+1} - z_m + z_m - z_t \\ &= z_{m+1} - z_m + z_m - z_{m-1} + z_{m-1} - P_t \\ &= \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ &= \sum_{k=t}^{m}\left(z_{k+1} - z_k\right) \end{aligned} \tag{16}$$

K prepisu formule (15b) použijeme vzťah kde $z_{m+1}=z_{\text{reward}}$. Dosadením vzťahu (16) do (15b) a použitím jednoduchej algebraickej identity

$$\sum_{t=1}^{m}\sum_{k=t}^{m} A_{kt} = \sum_{t=1}^{m}\sum_{k=1}^{t} A_{tk} \tag{17}$$

dostaneme

$$\begin{aligned}\Delta w &= \alpha \sum_{t=1}^{m}\left(\sum_{k=t}^{m}\left(z_{k+1}-z_k\right)\right)\frac{\partial z_t}{\partial w}\\ &= \alpha \sum_{t=1}^{m}\left(z_{t+1}-z_t\right)\sum_{k=1}^{t}\frac{\partial z_t}{\partial w}\end{aligned} \tag{18}$$

Sumarizujúc, inkrement obnovovacej formuly váhového koeficienta má tvar

$$\Delta w = \sum_{t=1}^{m}\Delta w_t \tag{19a}$$

$$\Delta w_t = \alpha\left(P_{t+1}-P_t\right)\sum_{k=1}^{t}\frac{\partial z_k}{\partial w} \tag{19b}$$

Tento dôležitý výsledok môže byť "zovšeobecnený" na formulu, ktorá tvorí základ TD(λ) rodiny učiacich metód

$$\Delta w_t = \alpha\left(z_{t+1}-z_t\right)\sum_{k=1}^{t}\lambda^{t-k}\frac{\partial z_k}{\partial w} \tag{20}$$

kde parameter $0\le\lambda\le1$. Pre $\lambda=1$ formula (20) poskytuje pôvodný výsledok (18), zatiaľ čo pre $\lambda=0$ dáva

$$\Delta w_t = \alpha\left(z_{t+1}-z_t\right)\frac{\partial z_t}{\partial w} \tag{21}$$

t.j. inkrement váhového koeficienta je určený len posledným "pozorovaním".

Formule (19) a (20) umožňujú rekurentný výpočet inkrementu Δw. Zavedieme nový symbol $e_t(\lambda)$, ktorý sa dá jednoducho počítať rekurentne

$$e_t(\lambda) = \sum_{k=1}^{t}\lambda^{t-k}\frac{\partial z_k}{\partial w} \Rightarrow e_{t+1}(\lambda) = \lambda e_t(\lambda) + \frac{\partial z_{t+1}}{\partial w} \tag{22}$$

kde $e_1(\lambda) = \partial z_1/\partial w$. Potom jednotlivé "parciálne" inkrementy Δw_t sú určené takto

$$\Delta w_t = \alpha\left(z_{t+1}-z_t\right)e_t(\lambda) \tag{23}$$

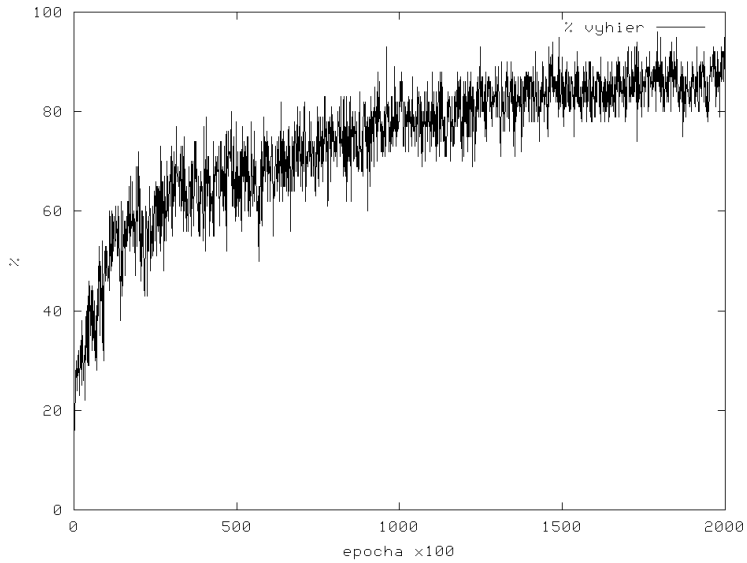
6.4 Použitie TD(λ) pravidla pri hre zjednodušenej dámy

V tejto kapitole uvedieme výsledky učenia TD(λ) pravidla pri hraní zjednodušenej dámy. Navrhnutú neurónovú sieť sme trénovali a testovali pomocou algoritmu MiniMax. Pri hraní bola neurónová sieť použitá ako ohodnocovacia funkcia možných ťahov. Vyberal sa ťah s najväčším ohodnotením ako je to opísané v sekcii 3. Pokúsime sa taktiež o analýzu natrénovanej neurónovej siete.

V prvom najjednoduchšom prístupe sme uvažovali len 1-agentový systém, kde jeden agent hrá proti inému agentovi, ktorého rozhodovanie je založené na metóde spätného prehľadávania do maximálnej hĺbky 3 (algoritmus MiniMax). Hra sa neustále opakuje s alternujúcim poradím. Po ukončení hry agent z dvojice, ktorého kognitívny

orgán je reprezentovaný neurónovou sieťou, adaptuje svoju neurónovú sieť pomocou metódy s odmenou a trestom, podľa TD(λ) pravidla.

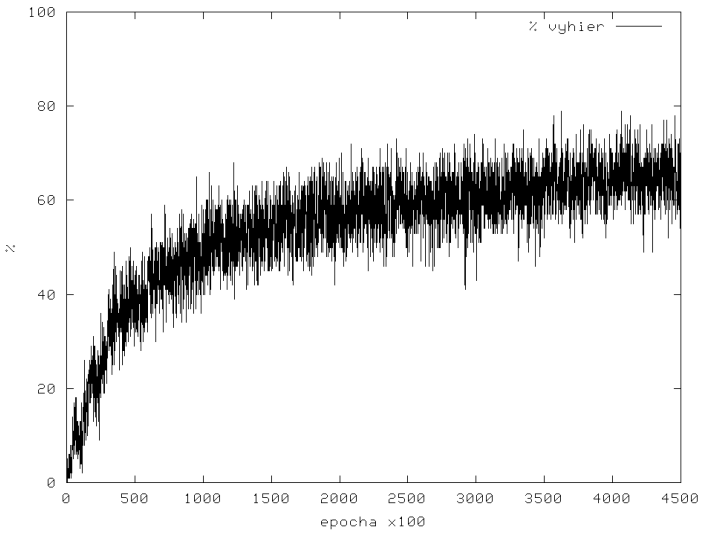
Pri testovaní a trénovaní sme použili dvojvrstvovú doprednú sieť (opísanú v sekcii 6.1) so 64 skrytými neurónmi, rýchlosťou učenia 0,01 a koeficientom λ =0,9. Sieť sa učila počas hrania zápasov. Po každom zápase dostala sieť ohodnotenie 1, ak vyhrala a 0, ak prehrala. Vždy po odohraných 100 partiách sme vynášali pomer vyhraných a prehraných hier do grafu. Na obrázku 10 sa nachádza priebeh učenia siete hrajúcej proti algoritmu MiniMax (pracujúcemu do hĺbky 1). Ako je vidieť z grafu, sieť bola schopná učiť sa a podstatne zlepšiť svoje výsledky. Na začiatku učenia sieť vyhrávala približne 40% partií, ale po natrénovaní vyhrávala 90% partií, čo je veľmi dobrý výsledok. Trénovanie prebiehalo počas odohrania 200000 zápasov.



Obr. 10. Priebeh učenia siete hrajúcej proti algoritmu MiniMax (pracujúcemu do hĺbky 1)

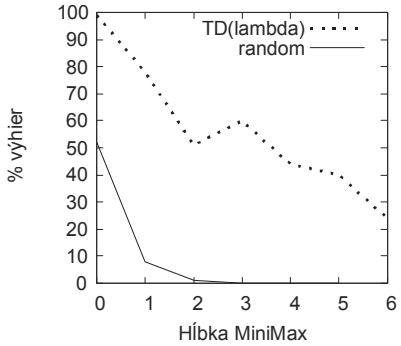
V ďalšom pokuse hrala sieť proti MiniMax-u pracujúcemu do hĺbky 3 (obrázok 11). Tu je vidieť, že sieť sa učila podstatne pomalšie a aj po 450000 odohraných zápasoch dosahovala víťazstvo v približne 65% hier. Ale aj napriek tomu toto je vynikajúci výsledok, lebo ak by sieť hrala rovnako ako náš MiniMax do hĺbky 3, vyhrávala by len 50% zápasov. Toto môže vlastne znamenať dve veci. Jednou je zle navrhnutá heuristická funkcia použitá v MiniMax-e. Druhou možnosťou je to, že sieť je schopná predpokladať s určitou správnosťou viac ako 3 ťahy dopredu. Tu sa vlastne ukazuje, že navrhnutie správnej heuristickej funkcie v symbolickej umelej inteligencii je zložitý problém a že neurónová sieť dokázala poraziť náš symbolický algoritmus.

Zaujímavým je porovnanie natrénovanej siete s rôznymi hĺbkami algoritmu MiniMax (obrázok 12). Testovaná sieť je natrénovaná z predchádzajúceho pokusu, a teda trénovaná na MiniMax-e hĺbky 3. Je zaujímavé, že v tomto prípade sieť mala problém s MiniMax-om hĺbky 2. Je to asi zapríčinené tým, že táto sieť si vytvorila stratégiu proti hĺbke 3 a MiniMax do hĺbky 2 ju „prekvapil“ inou stratégiou.



Obr. 11. Priebeh učenia siete proti algoritmu MiniMax pracujúcemu do hĺbky 3

Porovnanie taktiež ukazuje, že natrénovaná sieť hrá podstatne lepšie ako hráč, ktorý ťahá náhodne. Zaujímavý je aj výsledok hry proti MiniMax-u hĺbky 6. Aj keď sieť nevyhrávala podstatnú časť partií, stále bola schopná vyhrať 24%. Takýto výsledok považujeme za veľmi dobrý, keďže MiniMax do hĺbky 6 je veľmi silný protivník aj pre človeka.



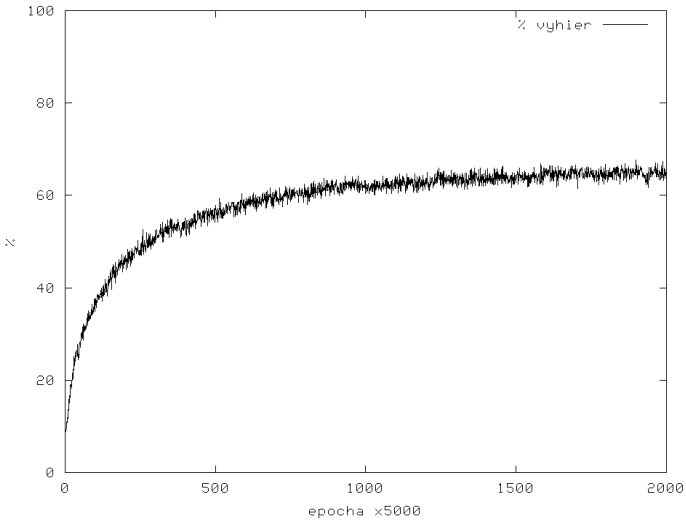
Obr. 12. Porovnanie algoritmu MiniMax a TD(λ) a náhodného ťahania

6.5 Výsledok adaptácie populácie neurónových sietí

V druhom zložitejšom prípade už uvažujeme pravý multiagentový systém, v ktorom populácia agentov hrala opakovane medzi sebou „megaturnaj“, pričom po každej partii sa adaptuje neurónová sieť oboch agentov pomocou metódy s odmenou a trestom. Počas megaturnaja hrala každá sieť partie s celou populáciou tak, že s každým oponentom hrala ako biely a aj ako čierny hráč.

Pre tento pokus sme vytvorili 20 neurónových sietí so 64 skrytými neurónmi, rýchlosťou učenia 0,01 a koeficientom λ=0,9. Tieto siete hrali megaturnaj a merali sme ich naučenie pomocou MiniMax-u hĺbky 3 (pri meraní sa siete neučili). Priebeh učenia je zobrazený na obrázku 13. Z obrázku je pekne vidieť, že aj napriek tomu, že neurónové siete sa neučili na hre s MiniMax-om, dokázali sa naučiť hrať dámu. To znamená, že na naučenie sietí sme už nepotrebovali experta, ktorý nám dokáže poskytovať dobrú hru. Tento výsledok považujeme za veľmi dobrý, keďže sme schopní získať stratégiu riešenia nejakého problému len z jeho opisu.

Tento pokus by sa dal prirovnať k situácii z reálneho života, kde by sme 20 ľuďom vysvetlili pravidlá hry a nechali ich hrať medzi sebou. Ľudia by hrali podľa pravidiel a vedeli by vyhodnotiť či vyhrali, alebo prehrali a na základe týchto informácií by sa učili. Tak ako ľudia, aj naše umelé neurónové siete sú schopné naučiť sa hrať hru.



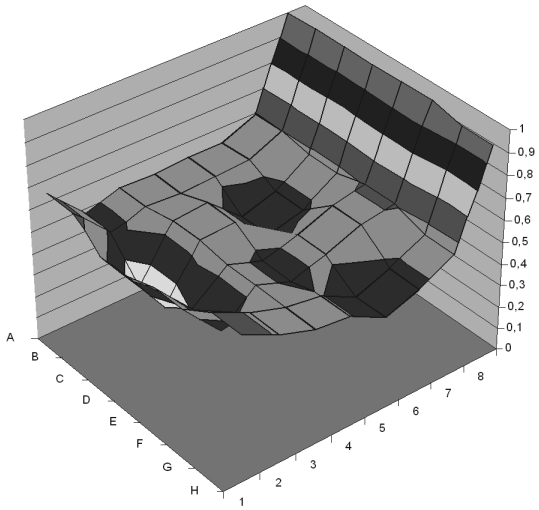
Obr. 13. Priebeh učenia populácie 20 neurónových sietí. Vynesená je priemerná hodnota výhier celej populácie proti algoritmu MiniMax pracujúcemu do hĺbky 3.

6.6 Analýza natrénovanej neurónovej siete

Zaujímavým problémom umelej inteligencie je pokus o získanie pravidiel z natrénovanej neurónovej siete. Pokúsili sme sa zistiť, aká taktika vznikla počas trénovania neurónovej siete. Analýzu sme založili na tom, že sme na vstup siete privádzali hracie plochy, na ktorých bola vždy len jedna figúrka. Sledovali sme, akú odozvu má sieť na figúrku v každej možnej pozícii hracej plochy. Výsledky sme vyniesli do grafu (obrázok 14), ktorý sme nazvali mapa odozvy. Je zaujímavé, že počas trénovania sa v sieti vygenerovala taktika, ktorá by sa dala opísať nasledovnými pravidlami:

- čím je figúrka bližšie k víťaznej pozícii, tým má lepšie ohodnotenie,
- figúrky z 2. radu by mali čo najskôr opustiť svoje pozície a ísť do útoku, ohodnotenie týchto figúrok je veľmi nízke, teda sieť si vyberie taký ťah, kedy figúrky opustia 2. rad,

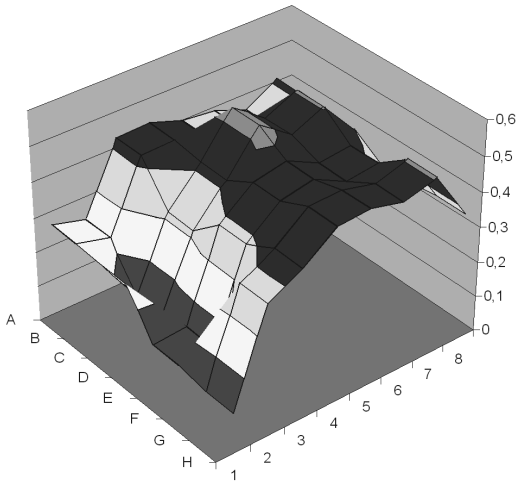
- figúrky z prvého radu by tam mali zostať. Ohodnotenie figúrok v 1. rade je v niektorých pozíciách dosť veľké a ak by sme s nimi chceli pohnúť, museli by sme prejsť cez 2. rad, ktorý má nízke ohodnotenie, preto sieť s týmito figúrkami pokiaľ nemusí, nehýbe. Teda takticky si ponechá svoje figúrky na políčkach, ktoré znamenajú pre súpera výhru, a tak ho blokujú.



Obr. 14. Mapa odozvy siete na hráčove figúrky

Ak vytvoríme rovnakú mapu odozvy, ale s použitím súperovej figúrky, prídeme k týmto záverom (obrázok 15):

- čím je súper bližšie k jeho víťazstvu, tým je ohodnotenie nižšie, teda sieť sa bude snažiť zabrániť prieniku súpera k jeho víťaznej pozícii,
- súperove figúrky stojace na políčkach, ktoré znamenajú naše víťazstvo, majú nízke ohodnotenie, preto sa ich bude sieť snažiť odtiaľ vylákať (napríklad obetovaním vlastnej figúrky).



Obr. 15. Mapa odozvy siete na protihráčove figúrky

6.7 Zhrnutie

Ukázali sme, že neurónová sieť je schopná naučiť sa hrať hru a vytvoriť si stratégiu, ktorou je schopná poraziť svojho súpera. Je pravda, že algoritmus MiniMax od určitej hĺbky generovania stromu neurónovú sieť poráža, ale rýchlosť generovania odpovede je pri neurónovej sieti neporovnateľne rýchlejšia a taktiež pamäťové nároky neurónovej siete sú minimálne. Algoritmus MiniMax sa dá ďalej vylepšovať napríklad alfa-beta orezávaním a tak zvyšovať jeho rýchlosť a znižovať pamäťové nároky. Nevýhodou neurónovej siete je potreba učenia, ktoré môže byť zdĺhavé. Zaujímavým výsledkom je, že neurónová sieť si dokáže pri hre so slabším protivníkom vytvoriť stratégiu, ktorá je vhodná aj proti silnejším protivníkom.

7 Adaptívna kombinácia lokálnych expertných neurónových sietí

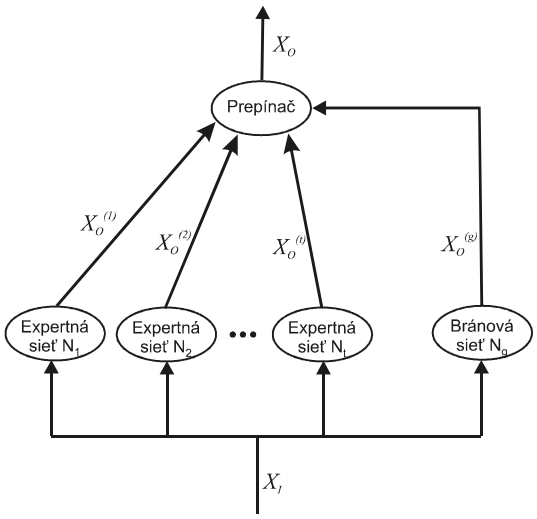
Niektoré úlohy riešené neurónovými sieťami sa dajú rozložiť na menšie podúlohy. Tieto podúlohy môžu byť lepšie riešiteľné expertnými neurónovými sieťami. Takto môžeme vytvoriť skupinu expertných sietí [15, 16], z ktorých každá dokáže riešiť časť pôvodnej úlohy (prebrané z [19]). Predpokladajme teda existenciu t neurónových sietí (expertných sietí, obrázok 16):

$$N_i = \left(G^{(i)}, w^{(i)}, \vartheta^{(i)}\right), (i = 1, 2, ..., t) \tag{24}$$

a jednu sieť, ktorá bude slúžiť ako bránová sieť (angl. gating network):

$$N_g = (G^{(g)}, w^{(g)}, \vartheta^{(g)}) \tag{25}$$

kde G je orientovaný graf reprezentujúci topológiu siete, ω sú váhové koeficienty a ϑ sú prahové koeficienty neurónovej siete. Bránová sieť bude vyberať, ktorý expert dokáže najlepšie vyriešiť momentálnu úlohu. Pre expertné siete predpokladajme rovnaký počet vstupných a výstupných neurónov.



Obr. 16. Adaptívna kombinácia lokálnych neurónových sietí.

Bránová sieť bude vyberať ktorý expert dokáže najlepšie vyriešiť momentálnu úlohu. Bránová sieť má taktiež rovnaký počet vstupných neurónov ako expertné siete. Počet výstupných neurónov bránovej siete je *t*. Pre danú kombináciu $\mathbf{x}_I / \hat{\mathbf{x}}_O$ sú výstupy expertných sietí popísané vektorom $\mathbf{x}_O^{(i)} = (x_1^{(i)}, x_2^{(i)}, ..., x_t^{(i)})$ a výstup bránovej siete je opísaný vektorom $\mathbf{x}_O^{(g)} = (x_1^{(g)}, x_2^{(g)}, ..., x_t^{(g)})$. Všetky prvky výstupných aktivít sú z intervalu (0, 1). Proporcionálne koeficienty $p_i \in (0,1)$ sú vytvárané bránovou sieťou ako odozva na vstupný vektor. Proporcionálne koeficienty sú definované ako funkcia softmax [50]:

$$p_i = \frac{x_i^{(g)}}{\sum_{(j=1)}^{(t)} x_j^{(g)}}, \ (i = 1, 2, ..., t) \tag{26}$$

$$\sum_{i=1}^{t} p_i = 1 \tag{27}$$

Ďalej budeme proporcionálne koeficienty interpretovať ako pravdepodobnosti, že zodpovedajúca expertná neurónová sieť je použiteľná na klasifikáciu vstupného vektora $\mathbf{x}_I$. Chybová funkcia sa dá zovšeobecniť ako:

$$\begin{aligned} E &= \frac{1}{2}\sum_{i=1}^{t} p_i \left(\mathbf{x}_O^{(i)} - \hat{\mathbf{x}}_O\right)^2 = \frac{1}{2}\sum_{i=1}^{t}\left(\sum_k \left(g_k^{(i)}\right)^2\right) \\ g_k^{(i)} &= \begin{cases} p_i\left(x_k^{(i)} - \hat{x}_k\right), k \in V_O \\ 0, k \notin V_O \end{cases} \end{aligned} \tag{28}$$

kde $x_k^{(i)}$ je *k*-ta výstupná aktivita *i*-tej expertnej neurónovej siete. Cieľom trénovania adaptívnej kombinácie expertných sietí je nájsť takú kombináciu váh a prahov vo všetkých expertných a bránovej sieti, aby sa minimalizovala hodnota chybovej funkcie. Adaptácia je analogická ako pri procese adaptácie doprednej neurónovej siete, aj keď vzťahy popisujúce parciálne derivácie sa mierne líšia od vzťahov platných pre dopredné neurónové siete:

$$\frac{\partial E}{\partial w_{ij}^{(g)}} = \frac{\partial E}{\partial \vartheta_i^g} x_j^{(g)} \tag{29}$$

potom parciálne derivácie $\frac{\partial E_k}{\partial x_i^{(g)}}$ sa získajú rekurentne podľa vzťahu:

$$\frac{\partial E_k}{\partial x_i^{(g)}} = f'(\xi_i^{\{g\}})\left(g_i^{(g)} + \sum_l \frac{\partial E_k}{\partial \vartheta_l^{(g)}} w_{li}^{(g)}\right), \tag{30}$$

kde za $g^{(g)}$ dosadzujeme:

$$g_i^{(g)} = \begin{cases} \left(\frac{1}{2}\left(\mathbf{x}_O^{(i)} - \hat{\mathbf{x}}_O\right)^2 - E\right)\cdot\left(\sum_j \mathbf{x}_j^{(g)}\right)^{-1}, i \in V_O \\ 0, i \notin V_O \end{cases}. \tag{31}$$

Najväčšiu hnaciu silu pri zmene váh a prahov bránovej siete má diferencia

$$\frac{1}{2}\left(\mathbf{x}_O^{(i)} - \hat{\mathbf{x}}_O\right)^2 - E. \tag{32}$$

Najväčšie zmeny hodnôt váh a prahov nastávajú pre expertnú sieť s indexom j, ktorý je definovaný ako:

$$j\left(\forall(i \neq j): \left(\mathbf{x}_O^{(i)} - \hat{\mathbf{x}}_O\right)^2 < \left(\mathbf{x}_O^{(j)} - \hat{\mathbf{x}}_O\right)^2\right). \tag{33}$$

Ak stredná kvadratická odchýlka expertnej siete s indexom i je menšia ako vážený priemer chýb všetkých expertných sietí, i-ta výstupná aktivita bránovej siete sa zvýši, v opačnom prípade sa táto aktivita zníži.

Pri použití metódy najstrmšieho spádu na súčasné trénovanie expertných sietí a bránovej siete na dvojiciach $\mathbf{x}_I / \hat{\mathbf{x}}_O$, systém začína po čase inklinovať k používaniu jedinej expertnej siete (jej výstupný vektor sa čo najviac podobá na požadovaný výstup) na klasifikáciu daného vstupného vektora $\mathbf{x}_I$. A teda proporcionálne koeficienty majú tendenciu nadobúdať binárne hodnoty

$$p_j = \begin{cases} 1, j = i \\ 0, inak \end{cases}, \tag{34}$$

kde index i predstavuje tú expertnú sieť, ktorej rozdiel medzi skutočným výstupom a požadovaným výstupom je minimálny.

Natrénovaná zmes sietí sa môže použiť na klasifikáciu vzorov. Na získanie odpovedí môžeme použiť viac prístupov. Pri každom musíme vypočítať odozvu všetkých expertných sietí a bránovej siete na vstupný vektor $\mathbf{x}_I$. Na základe týchto informácií bude vytvorený výstup z kombinácie sietí, ktorý čo najviac zodpovedá požadovanému výstupu. Jednou z možností pri výbere expertnej siete na klasifikáciu daného vstupu je vybrať sieť s maximálnym proporcionálnym koeficientom. Ďalšia možnosť nevyberá jednu z expertných sietí, ktorej výstupný vektor bude považovať za výstupný vektor kombinácie sietí, ale poskytne lineárnu kombináciu výstupov všetkých expertných sietí:

$$\tilde{\mathbf{x}}_O = p_1\mathbf{x}_O^{(1)} + p_2\mathbf{x}_O^{(2)} + \ldots + p_t\mathbf{x}_O^{(t)} \tag{35}$$

7.1 Trénovanie adaptívnej kombinácie lokálnych expertných sietí s použitím TD pravidla

Ďalej sa budeme venovať odvodeniu učiaceho pravidla TD(λ) pre adaptívnu kombináciu lokálnych expertov. Chybovú funkciu upravíme tak, aby obsahovala všetky chyby siete počas celej hry (spočítame chyby zo všetkých m ťahov hry) [26]:

$$E = \frac{1}{2}\sum_{j=1}^{m}\left(\tilde{\mathbf{x}}_{O,j} - \hat{\mathbf{x}}_O\right)^2 \tag{36}$$

Použitím metódy najstrmšieho spádu dostávame pre zmenu váh expertov metódou TD(λ) nasledujúce pravidlo (j - index ťahu, i - index expertnej siete):

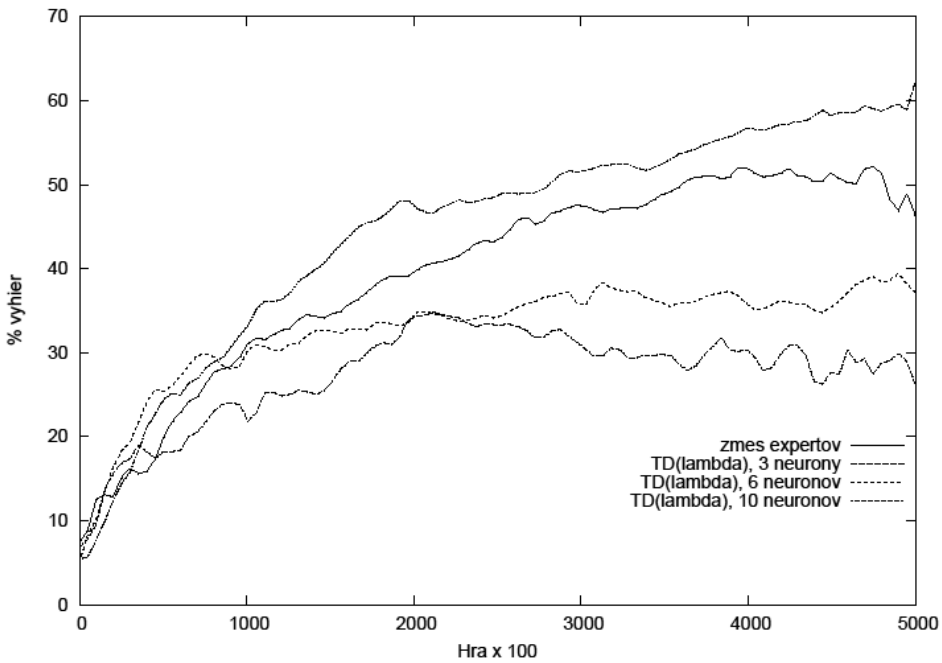
$$\Delta w_j^{(i)} = \alpha\left(\tilde{\mathbf{x}}_{O,j+1} - \tilde{\mathbf{x}}_O, j\right)\sum_{k=1}^{j}\lambda^{j-k} p_{i,k}\frac{\partial \mathbf{x}_{O,k}^{(i)}}{\partial w_j^{(i)}}. \tag{37}$$

Pre bránovú sieť dostávame:

$$\Delta w_{j,l} = \alpha\frac{1}{2}\frac{1}{\left\|\mathbf{x}_{O,j}^{(g)}\right\|}\left(\tilde{\mathbf{x}}_{O,j+1} - \tilde{\mathbf{x}}_{O,j}\right)\sum_{k=1}^{j}\lambda^{j-k}\frac{\partial \mathbf{x}_{l,k}^{(g)}}{\partial w_j}\left(\mathbf{x}_{O,k}^{(l)} - \tilde{\mathbf{x}}_{O,k}\right), \tag{38}$$

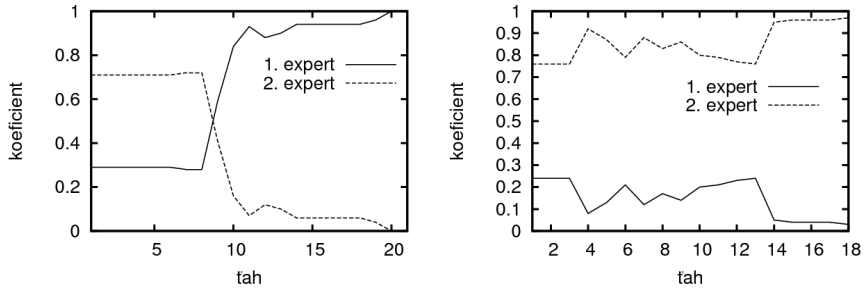
kde l je index l -teho experta (výstupného neurónu bránovej siete).

Pre overenie schopnosti adaptívnej kombinácie lokálnych expertov naučiť sa hrať párovú hru sme použili hru zjednodušenej dámy. Sieť sa učila počas hry proti algoritmu MiniMax pracujúcemu do hĺbky 2. Porovnávali sme dvojvrstvovú doprednú sieť a zmesi expertov trénované prístupom TD(λ). Zmes expertov obsahovala dve neurónové siete s troma skrytými neurónmi, bránová sieť mala tiež tri neuróny. Rýchlosť učenia α =0,1 a koeficient λ =0,99. Na obrázku 17 je priebeh získaných bodov počas učenia opísanej siete.



Obr. 17. Porovnanie priebehu učenia medzi doprednou sieťou a architektúrou adaptívnej kombinácie lokálnych expertných (dvaja experti) sietí proti algoritmu MiniMax (pracujúcemu do hĺbky 2).

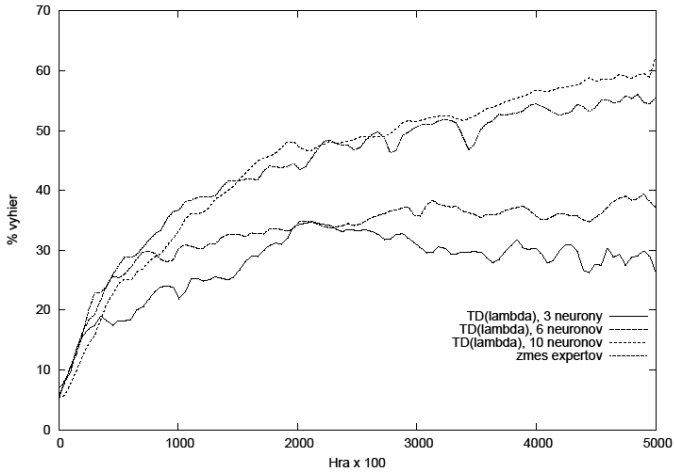
Pre porovnanie na grafe sú zobrazené priebehy učenia pre doprednú sieť s rôznym počtom skrytých neurónov. Je vidieť, že zmes expertov hrala podstatne lepšie ako dopredná sieť s troma skrytými neurónmi a skoro tak dobre ako dopredná sieť s 10-timi skrytými neurónmi. Je teda jasné, že sieť musela zapájať oboch expertov. Pri porovnaní treba ale zohľadniť fakt, že počet váh testovanej zmesi expertov a sieti s 10 skrytými neurónmi je vlastne rovnaká. V grafoch na obrázku 18 sú uvedené priebehy proporčných koeficientov tej istej bránovej siete počas dvoch hier. Je vidieť, že druhý expert sa zapájal pri otvorení, ale koncovky podľa situácie na šachovnici riešili obaja.



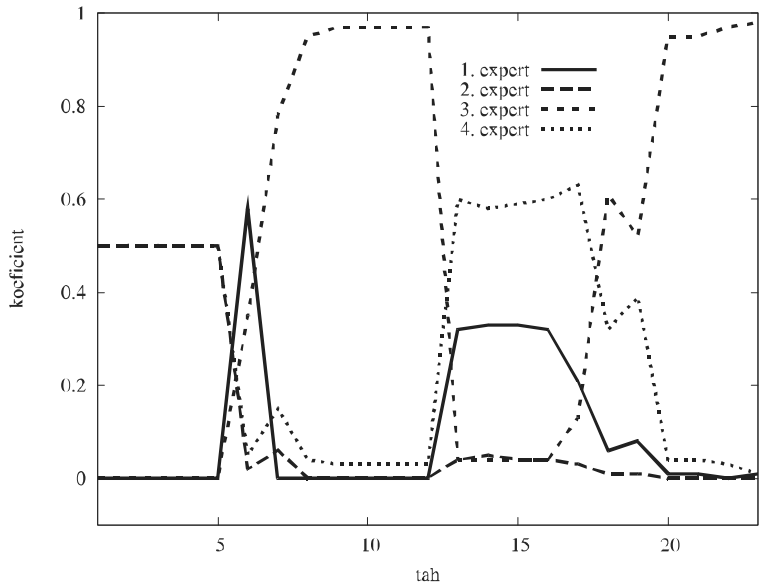
(a) 1. hra (b) 2. hra

Obr. 18. Priebehy proporcionálnych koeficientov počas jednej hry pri použití 2 expertov

V ďalšom pokuse sme rozšírili počet expertov na 4. Takáto sieť bola schopná hrať lepšie ako s dvoma expertmi, aj keď rozdiel nebol veľmi veľký. Priebeh učenia je na obrázku 19. Zaujímavé je, ako boli využívaní experti počas hry. Na grafe na obrázku 20 je priebeh proporcionálnych koeficientov. Je vidieť, že hru začína expert 2, potom ťahy 6 až 11 ovládal len expert 3. Pred koncom hry sa zapojil expert číslo 4 a 1. Koniec hry riadil expert 3. Počas tejto hry sa do rozhodovania zapojil každý expert.



Obr. 19. Porovnanie priebehu učenia medzi doprednou sieťou a architektúrou zmesi expertných sietí so 4 expertmi proti algoritmu MiniMax (pracujúcemu do hĺbky 2).



Obr. 20. Priebehy proporcionálnych koeficientov počas jednej hry pri použití 4 expertov.

Aby sme presnejšie preskúmali ako sa zapájajú experti do hry, zobrali sme 50 víťazných partií najlepšie natrénovanej zmesi expertov (vyhrávala 70% partií). Každú partiu sme rozdelili na tretiny, kde prvá tretina ťahov reprezentuje začiatok partie, druhá stred a tretia koniec partie. Pre každú fázu hry sme spriemerovali proporcionálne koeficienty bránovej siete. Výsledok je v tabuľke 3. Je vidieť, že prvý expert riadi otvorenie partie, v strede hry sa zapája prvý a druhý expert. Koncovku riadi štvrtý expert. Zaujímavé je, že tretí expert sa vlastne počas hry vôbec nezapájal.

Tabuľka 3. Priemer proporcionálnych koeficientov počas fáz hry

	1. expert	2. expert	3. expert	4. expert
začiatok partie	0.62	0.21	0.05	0.09
stred partie	0.51	0.30	0.05	0.15
koniec partie	0.08	0.19	0.02	0.71

Ďalším experimentom sme sa snažili zistiť, či sú experti naozaj špecializovaní alebo či dokážu hrať samostatne zjednodušenú dámu. Z obrázku 19 je vidieť, že sieť s 3 skrytými neurónmi je schopná naučiť sa hrať dámu proti algoritmu MiniMax pracujúcemu do hĺbky 2 s úspešnosťou viac ako 30%. V tabuľke 4 sú výsledky hrania jednotlivých expertov, z ktorých každý mal 3 skryté neuróny. Je vidieť, že ak hral každý expert samostatne, dosahoval slabé výsledky. Najviac sa darilo expertovi, ktorý bol v zmesi používaný pri koncovkách (4. expert). Celkom úspešný bol aj expert, ktorý bol podľa bránovej siete čiastočne vhodný na každú fázu partie. Expert, ktorý bol používaný pri otvoreniach, nehral samostatne vôbec dobre. Pravdepodobne nedokázal rozpoznať víťazné pozície, prípadne pozície, ktoré vedú v poslednej fáze hry k víťazstvu. Tretí expert, ktorého bránová sieť nevyužívala, nedokázal hrať ani samostatne.

Tabuľka 4. Percentuálny podiel víťazstiev expertov, ak rozhodujú sami

	1. expert	2. expert	3. expert	4. expert
% výhier	8	15	3	19

7.2 Zhrnutie

Odvodili sme TD(λ) pravidlo pre zmes expertných neurónových sietí. Ukázali sme, že zmesi expertov sa dokážu úspešne naučiť stratégiu hry pri učení s odmenou a trestom. Experti sa špecializujú na rôzne pozície na hracej ploche. Keď ale porovnávame výsledky zmesi expertov s doprednou sieťou tak, že zohľadňujeme počet váh, zmesi expertov neprinášajú žiadne zlepšenie. Bolo by ale ešte vhodné vyskúšať siete s rôznym počtom skrytých neurónov ako expertov. Takýto prístup by mohol priniesť lepší výsledok z pohľadu počtu váh.

Podobné riešenie so zmesou expertných neurónových sietí aplikovali aj Zaman a Wunsch na hru Go [52]. Neporovnávali ale výsledky s TD(λ) pravidlom, ale sledovali schopnosť učenia proti symbolickému algoritmu.

8 Použitie rozšíreného Kalmanovho filtra na učenie neurónových sietí

V roku 1960 R. E. Kalman publikoval [18] svoje riešenie rekurzívnej filtrácie diskrétnych údajov lineárneho systému. V posledných rokoch sa venuje Kalmanovej filtrácii veľká pozornosť, keďže po úprave sa dá použiť na trénovanie neurónových sietí či už dopredných alebo rekurentných [7, 11, 32, 46, 47, 48].

8.1 Použitie Kalmanovho filtra na učenie neurónových sietí

Správanie neurónovej siete môžeme opísať ako nelineárny systém takto:

$$\mathbf{w}_{k-1} = \mathbf{w}_k + \mathbf{r}_k \tag{39}$$

$$\mathbf{y}_k = \mathbf{h}(\mathbf{w}_k, \mathbf{x}_k) + \mathbf{q}_k \tag{40}$$

Rovnica (39) opisuje to, že stav neurónovej siete je charakterizovaný ako stacionárny proces narušovaný procesným šumom $\mathbf{r}_k$. Stav siete $\mathbf{w}$ je vektor váh. Rovnica pozorovania vyjadruje požadovaný výstup siete ako lineárnu funkciu vstupného vektora $\mathbf{x}_k$ a vektora váh $\mathbf{w}_k$. Táto rovnica je rozšírená o náhodný šum merania $\mathbf{q}_k$. Šum merania je biely, má nulovú strednú hodnotu a kovarianciu $E[\mathbf{q}_k \mathbf{q}^T{}_k] = \mathrm{Q}_k$. Procesný šum je tiež biely s nulovou strednou hodnotou a kovarianciou $E[\mathbf{r}_k \mathbf{r}^T{}_k] = R_k$.

Trénovanie neurónovej siete použitím rozšíreného Kalmanovho filtra môžeme charakterizovať ako hľadanie odhadu stavu $\mathbf{w}$ (váh siete) pri minimálnej kvadratickej

chybe s použitím všetkých doterajších pozorovaní. Keďže je stavová funkcia lineárna (dokonca $f(\mathbf{w}_k) = \mathbf{w}_k$), môžeme definovať základné veličiny Kalmanovho filtra takto:

$$\mathbf{K}_k = \mathbf{P}_k \mathbf{H}_k^T [\mathbf{H}_k \mathbf{P}_k \mathbf{H}_k^T + \mathbf{R}_k]^{-1} \tag{41}$$

$$\hat{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_k + \mathbf{K}_k [\mathbf{y}_k - h(\hat{\mathbf{w}}_k, \mathbf{x}_k)] \tag{42}$$

$$\mathbf{P}_{k+1} = \mathbf{P}_k - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_k + \mathbf{Q}_k \tag{43}$$

n_x	Počet všetkých váh (stavových premenných)
n_o	Počet výstupných neurónov (výstupov)
$\mathbf{x}_{[n_x \times 1]}$	Stav systému, t.j. vektor všetkých váh
$h(\cdot)_{[n_o \times 1]}$	Funkcia vracajúca vektor skutočných výstupov
$\mathbf{y}_{[n_o \times 1]}$	Vektor požadovaných výstupov
$\mathbf{K}_{[n_x \times n_o]}$	Matica Kalmanovho zosilnenia
$\mathbf{P}_{[n_x \times n_x]}$	Kovariančná matica chyby stavu
$\mathbf{Q}_{[n_x \times n_x]}$	Kovariančná matica procesného šumu
$\mathbf{R}_{[n_o \times n_o]}$	Kovariančná matica šumu pozorovania
$\mathbf{H}_{[n_o \times n_x]}$	Matica pozorovania (Jacobián)

Prvky matice $\mathbf{P}_0$ volíme rádovo $100\mathbf{I} - 1000\mathbf{I}$ ($\mathbf{I}$ je jednotková matica). Maticu inicializujeme $\mathbf{Q}_0$ na $10^{-1}\mathbf{I}$ až $0.1\mathbf{I}$. $\mathbf{R}_0$ $10\mathbf{I} - 100\mathbf{I}$. Prvky matice $\mathbf{H}$ získame ako parciálne derivácie výstupu siete k jej váham. V prípade doprednej siete použijeme algoritmus spätného šírenia chyby.

8.2 Učenie s odmenou a trestom s použitím EKF

Našou snahou je upraviť rozšírený Kalmanov filter tak, aby sme ho mohli použiť pri učení s odmenou a trestom. V tejto kapitole opíšeme možnú úpravu trénovania siete rozšíreným Kalmanovým filtrom na učenie s odmenou a trestom. Navrhnutý prístup porovnáme s TD(λ) algoritmom na hre Tic-Tac-Toe a zjednodušená dáma.

Pri učení s odmenou a trestom sa snažíme nájsť optimálnu funkciu hodnoty. Funkcia hodnoty pri hraní hier nám dáva ohodnotenie stavu s_t hry, pričom toto ohodnotenie predstavuje odhad konečného výsledky hry v prípade, že sa nachádzame v tomto stave. Myšlienka je teda taká, že celá postupnosť šachovníc počas jednej hry ($s_1, \ldots, s_T$) by mala byť ohodnotená rovnako ako výsledok. Ak túto myšlienku zapíšeme rovnicou, dostávame tzv. Monte Carlo pravidlo učenia [41]:

$$\text{Pre každé } t = 1 \ldots T : V(s_t) = V(s_t) + \alpha(reward - V(s_t)), \tag{44}$$

kde α je rýchlosť učenia, $V(s)$ je funkcia hodnoty (sekcia 5.1), čo v našom prípade znamená výstup neurónovej siete a *reward* je výsledok hry.

Definujme rovnice (41)-(43) ako funkciu *EKF(vstup, požadovaný výstup).* Algoritmus učenia je potom jednoduchý:

```
pre t = 1 až T rob EKF(s[t], reward);
```

Monte-Carlo metóda nemusí byť úplne vhodná, keďže prvý ťah partie sa môže snažiť ohodnotiť 0 (hra skončila prehrou), ale to nemusí byť správne, keďže prvý ťah mohol byť správny, len niekde na konci hry urobil hráč chybu. Preto je lepšie upraviť vzťah (44) na TD(0) pravidlo [41]. Tento prístup upravuje ohodnotenia stavov tak, že ohodnotenie stavu v čase *t* upravuje podľa ohodnotenia stavu $t+1$.

$$\text{Pre každé } t = 1 \ldots T : V(s_t) = V(s_t) + \alpha(V(s_{t+1}) - V(s_t)), \tag{45}$$

kde $V(s_{T+1}) = reward$.

Algoritmus učenia môžeme zapísať:

```
pre t = 1 až T rob EKF(s[t], V(s[t+1]));
```

V ďalšej kapitole porovnáme učenie s odmenou a trestom s použitím EKF oproti TD(λ) algoritmu.

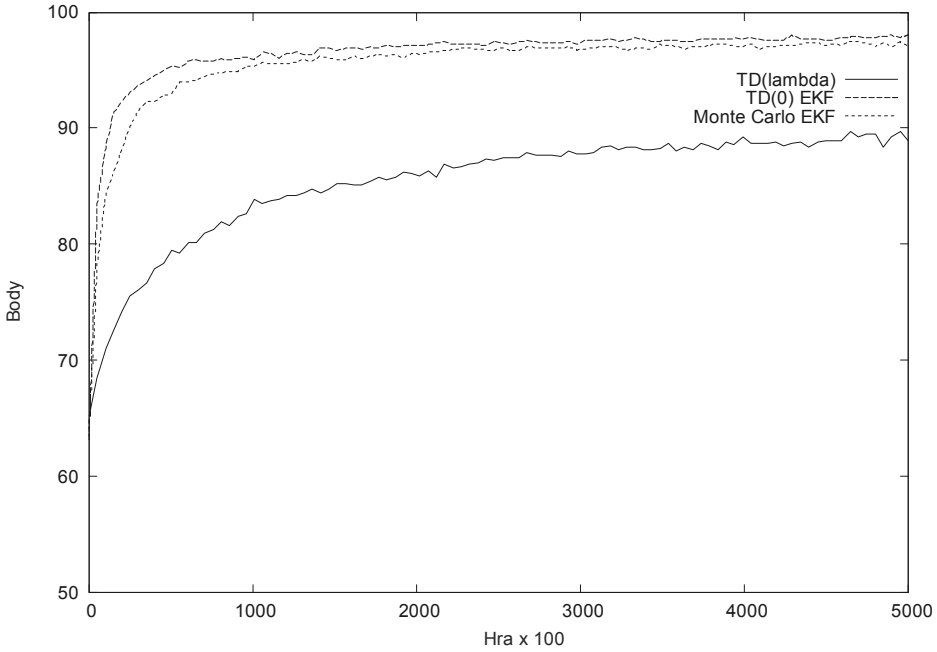
8.3 Dosiahnuté výsledky učenia s odmenou a trestom s použitím EKF

V tejto kapitole uvedieme dosiahnuté výsledky pre opísané metódy učenia a pre rôzne hry a sily oponenta.

Porovnanie výsledkov pre hru Tic-Tac-Toe

Ako prvú hru pre porovnanie algoritmov učenia sme vybrali hru Tic-Tac-Toe. Je to hra hraná na mriežke 3×3. Hráči ukladajú striedavo na hraciu plochu svoje kamene. Ak niektorý z hráčov vytvorí horizontálnu, vertikálnu alebo diagonálnu trojicu, tak vyhráva. Táto hra má maximálne 9 ťahov, teda každý hráč nepotiahne viac ako 5-krát. Testovali sme algoritmus TD(λ) a rozšírený Kalmanov filter použitý na učenie s odmenou a trestom metódou TD(0) a Monte Carlo. Priebehy v grafoch sú priemery 100 behov učenia. Každý algoritmus bol testovaný s 10 rovnako náhodne inicializovanými váhami sietí, pričom učenie na každej sieti prebehlo 10-krát.

Graf na obrázku 21 zobrazuje schopnosť naučiť sa stratégiu proti súperovi ťahajúcemu náhodne. V grafe sú vynesené body, ktoré získal testovaný algoritmus. Za víťazstvo dostáva 1 bod a za remízu 0,5.



Obr. 21. Priebeh učenia neurónových sietí (skrytá vrstva mala 10 neurónov) pri učení s prístupom TD(λ) a EKF proti náhodne ťahajúcemu súperovi pri hre Tic-Tac-Toe

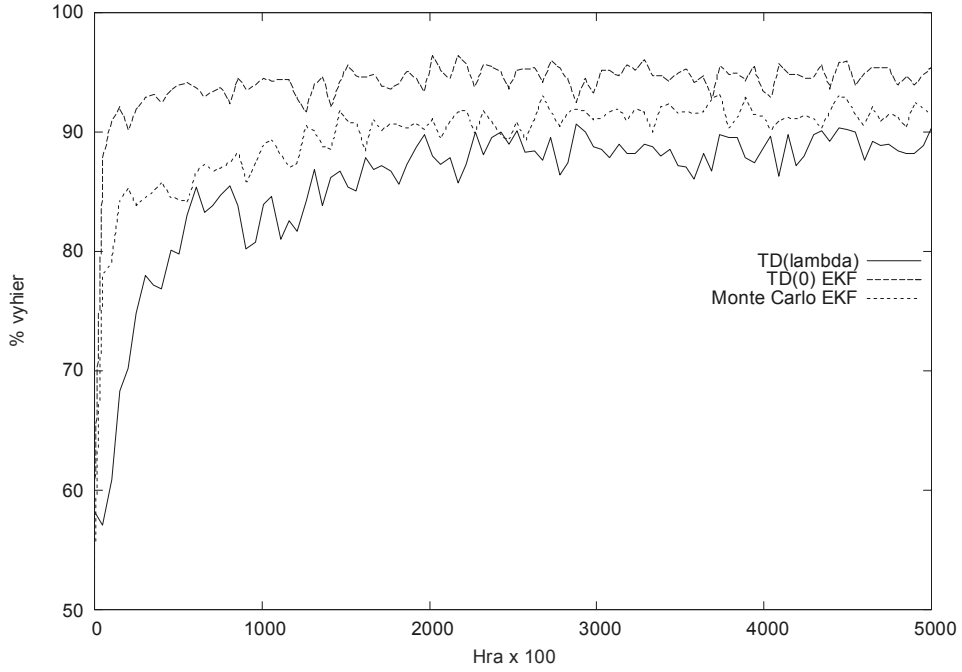
Neurónová sieť mala 10 skrytých neurónov, rýchlosť učenia pre TD(λ) algoritmus bola 0,1 a λ bola 0,99. Rozšírený Kalmanov filter (EKF) bol inicializovaný $\mathbf{P}_0 = 100\mathbf{I}$, $\mathbf{Q}_0 = 0\mathbf{I}$ a $\mathbf{R}_0 = 1\mathbf{I}$. Je vidieť, že TD(0) EKF sa z pohľadu odohraných partií učí najrýchlejšie a po odohraní 100 000 hier dosahuje najlepší výsledok. Algoritmus Monte Carlo EKF je len o niečo horší. TD(λ) sa učí najpomalšie a nakoniec jeho výsledok nie je ani taký dobrý, ako výsledky EKF.

8.4 Porovnanie výsledkov pre zjednodušenú dámu

Testovanie učiacich algoritmov pri hre zjednodušenej dámy prebiehalo rovnako ako pri hre Tic-Tac-Toe. Neurónová sieť mala 10 skrytých neurónov v prvých dvoch pokusoch a 20 skrytých neurónov v ďalšom pokuse. Rýchlosť učenia pre TD(λ) algoritmus bola 0,1 a λ bola 0,99. EKF bol inicializovaný $\mathbf{P}_0 = 100\mathbf{I}$, $\mathbf{Q}_0 = 0\mathbf{I}$ a $\mathbf{R}_0 = 1\mathbf{I}$. Každý algoritmus bol testovaný s 10 rovnako náhodne inicializovanými váhami sietí, učenie na každej sieti prebehlo raz.

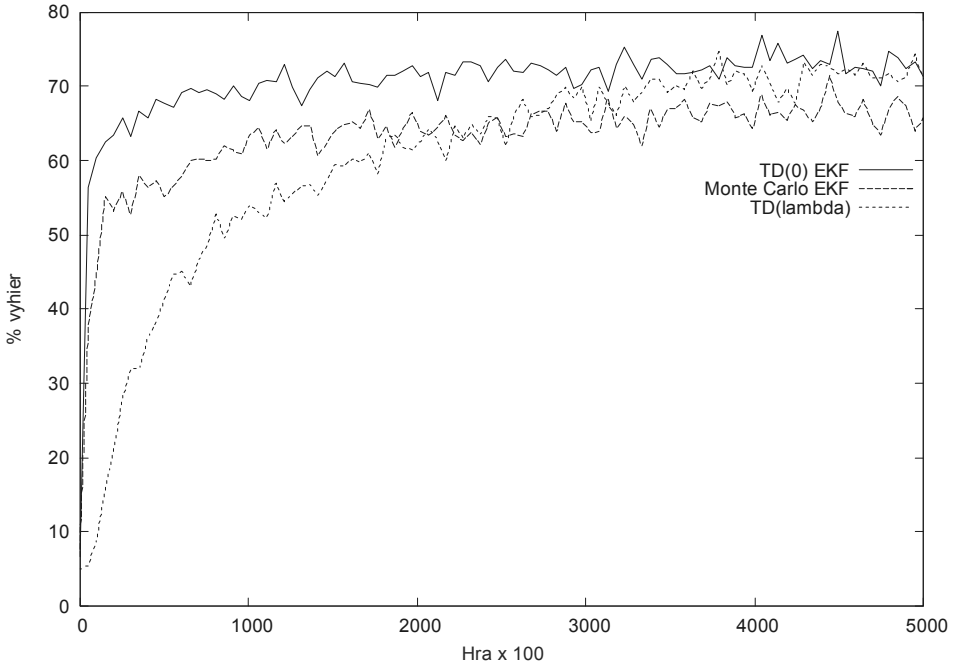
Prvý graf (obrázok 22) zobrazuje percento vyhraných partií v priebehu učenia proti náhodne ťahajúcemu súperovi. Siete mali 10 skrytých neurónov. Rozšírený

Kalmanov filter s TD(0) učením sa veľmi rýchlo naučí vyhrávať viac ako 90 % hier. Monte Carlo EKF sa učí taktiež veľmi rýchlo, ale ani po odohraní 500 000 hier nehrá tak dobre ako TD(0) variant.

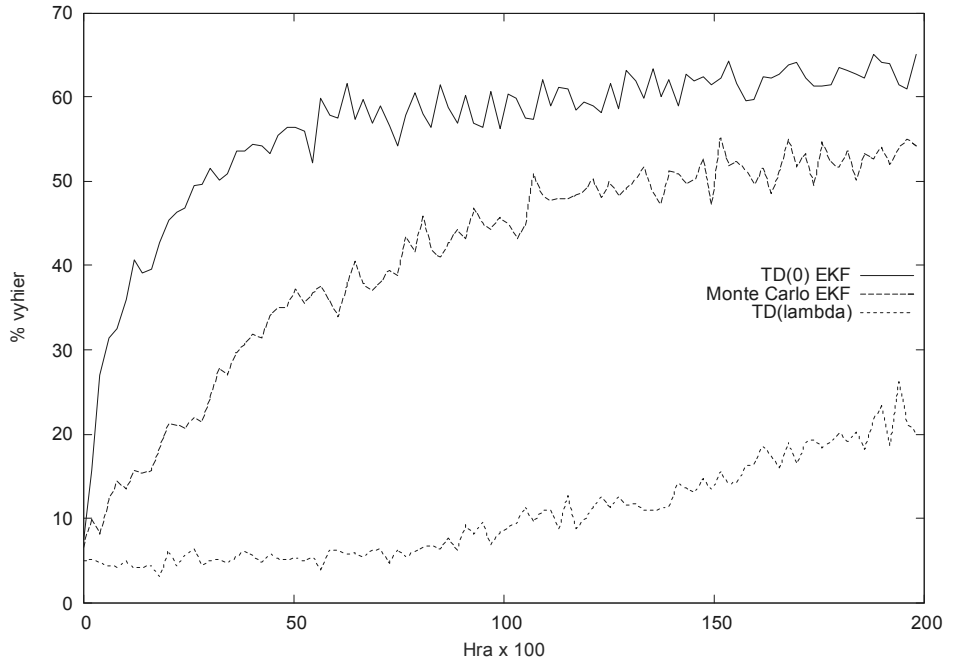


Obr. 22. Priebeh učenia neurónových sietí (skrytá vrstva mala 10 neurónov) pri učení s prístupom TD(λ) a EKF proti náhodne ťahajúcemu súperovi pri hre dáma

Pri ďalšom experimente (obrázok 23) sme to algoritmom skomplikovali tým, že sa učili proti algoritmu MiniMax pracujúcemu do hĺbky 2. Monte Carlo EKF dosahuje relatívne rýchlo dobré výsledky, ale nie je schopný vytvoriť rovnako dobrú stratégiu ako TD(λ) alebo TD(0) EKF algoritmus. Na obrázku 24 je zobrazených prvých 20 000 hier tohto experimentu. Je vidieť, že TD(0) EKF stačilo približne 10 000 hier, aby bol schopný vyhrávať 60 % partií.



Obr. 23. Priebeh učenia neurónových sietí (skrytá vrstva mala 20 neurónov) pri učení s prístupom TD(λ) a EKF proti MiniMax algoritmu pracujúcemu do hĺbky 2 pri hre dáma



Obr. 24. Priebeh učenia neurónových sietí (skrytá vrstva mala 20 neurónov) pri učení s prístupom TD(λ) a EKF proti MiniMax algoritmu pracujúcemu do hĺbky 2 pri hre dáma

8.5 Zhodnotenie

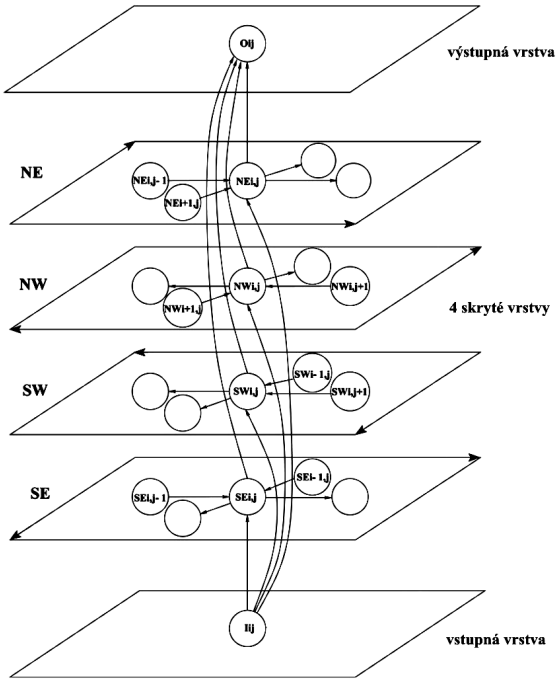
Rozšírený Kalmanov filter sme úspešne použili pri hraní hier s použitím učenia s odmenou a trestom. Jeho hlavnou výhodou je rýchlosť učenia z pohľadu odohraných partií. Učenie s použitím pravidla TD(λ) dosiahlo rovnaké výsledky až po odohraní veľkého počtu hier. Jedinou nevýhodou rozšíreného Kalmanovho filtra je jeho výpočtová náročnosť. Táto nevýhoda sa dá úspešne vyriešiť použitím grafického procesora ako výpočtového prostriedku. Tento dokáže až 6-násobne zvýšiť rýchlosť Kalmanovho filtra [48].

9 Multi-dimenzionálne rekurentné neurónové siete

Rekurentné siete boli pôvodne vyvinuté na spracovanie sekvencií dát. Vďaka ich rekurentným väzbám, sú schopné využívať predošlé informácie. Väčšina rekurentných sietí je navrhnutá na spracovanie jedno-dimenzionálnych dát - napríklad rôzne postupnosti symbolov ako texty, či časové rady. Obrazové dáta sú ale dvojrozmerné, preto použitie jedno-dimenzionálnych rekurentných sietí neprináša dobré výsledky.

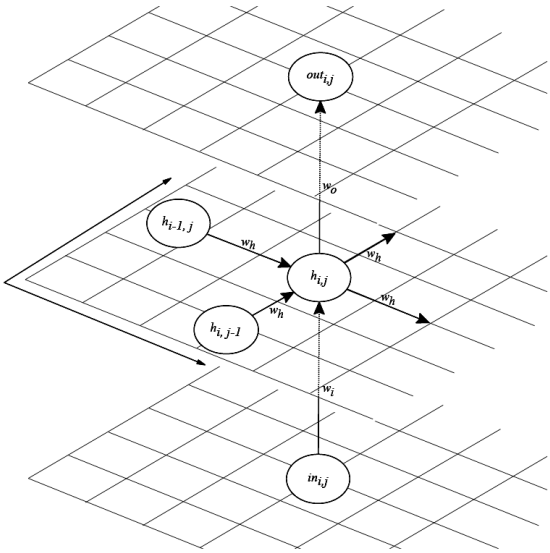
9.1 Štruktúra multi-dimenzionálnej rekurentnej neurónovej siete

Multi-dimenzionálne rekurentné neurónové siete (MDRNS) [51, 53] môžu zachytávať viac rozmernosť vstupných dát. Štruktúra takejto siete pre dvojrozmerné vstupné dáta je na obrázku 25.



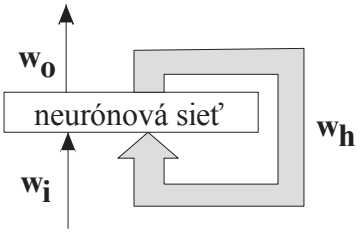
Obr. 25. Štruktúra multi-dimenzionálnej rekurentnej neurónovej siete (prebrané z [51])

Sieť sa skladá z piatich vrstiev. Jedna je vstupná, ktorá má rozmer vstupu $n \times n$, štyri sú skryté a jedna výstupná. Každá skrytá aj výstupná vrstva má tiež $n \times n$ uzlov, teda rovnaký počet ako vstup. Vrstvy sa líšia v tom, ako sú v nich poprepájané uzly. Názov vrstvy zachytáva smer šírenia signálu vo vrstve. Na obrázku 26 je zobrazená jedna vrstva a smery šírenia signálov zo vstupu, na vrstve a k výstupnej vrstve. Každý uzol je neurónová sieť. Spoje medzi uzlami sú označené váhovými koeficientmi. Uzly vo vrstve sú pospájané väzbami s váhami w_h. Uzly susediace s okrajom mriežky nemajú vstup od iného uzla, preto je tento vstup nahradený preddefinovanou hodnotou a váhovými koeficientmi w_b. Vstup a výstup má váhy w_i a w_o. Dostávame teda 4 skupiny váh - w_h, w_b, w_i a w_o. Celkový počet váh v sieti pri k počte skrytých neurónov je $2k + k^2 + k + k = 4k + k^2$ [51].



Obr. 26. Štruktúra multi-dimenzionálnej rekurentnej neurónovej siete (prebrané z [51])

Váhy sú rovnaké pre všetky uzly. Keďže každý uzol je tá istá neurónová sieť a v rámci jednej vrstvy sú uzly pospájané rovnakými váhami *wh* vzniká nám rekurentná sieť zobrazená na obrázku 27 (rekurentné väzby sú vlastne dve - od dvoch predchádzajúcich uzlov).



Obr. 27. Schematicky znázornená rekurentná väzba uzlu multi-dimenzionálnej rekurentnej neurónovej siete.

Takto navrhnutá neurónová sieť má tú výhodu, že ju môžeme použiť na ľubovoľne veľký dvojrozmerný vstup. Pri trénovaní sa dá celá štruktúra MDRNS rozvinúť do doprednej siete. Na trénovanie siete sa dá potom úspešne použiť metóda najstrmšieho spádu (sekcia 6.2).

9.2 Použitie MDRNS pri hraní GO

Autori v [53] trénovali multi-dimenzionálnu rekurentnú neurónovú sieť (MDRNS) na hre GO. Sieť bola schopná naučiť sa hrať hru na základnej úrovni, čo je ale podstatnejšie, učenie mohlo prebiehať na malých hracích doskách rozmeru 7x7, pričom takto natrénovaná sieť sa potom dala naškálovať na rozmer 19x19. Aj na veľkej hracej ploche bola schopná vhodne ohodnocovať hracie plochy. Vzory ktoré vedela sieť nájsť na malej šachovnici, úspešne ohodnocovala aj na veľkej.

10 Záver

Umelá inteligencia sa venuje problému hrania hier už od vzniku počítačovej vedy. Za ten čas sa podarilo dosiahnuť v niektorých hrách vynikajúce výsledky, ale stále existujú problémy, ktoré nie je počítač schopný riešiť lepšie ako človek. To nám vytvára priestor na návrh nových algoritmov či už symbolickej alebo subsymbolickej umelej inteligencie. Ukázalo sa, že každý z prístupov, prípadne ich kombinácia, keď je použitý správne na vhodnú hru, je schopný hrať hru na majstrovskej úrovni. Pri symbolickej umelej inteligencii sa pokúšame vytvoriť algoritmy, ktoré by priamo používali znalosti, ktoré máme o danej hre, pričom pri subsymbolickom prístupe si tieto znalosti vytvára algoritmus akoby sám. Stále sa zvyšujúci výkon počítačov a ich pamätí zlepšuje výsledky klasických algoritmov a umožňuje skúšanie nových prístupov.

Ukázali sme, že prístup učenia s odmenou a trestom, ktorý pracuje bez učiteľa sa dá vhodne použiť pri vytváraní kognitívneho orgánu agenta. Tento je schopný naučiť sa hrať hru a teda vytvoriť si stratégiu, ktorou dokáže poraziť iný algoritmus prípadne človeka. Prístupy učenia s odmenou a trestom sa dajú použiť na klasické dopredné neurónové siete, ako aj architektúru zmesi expertných neurónových sietí. Ďalšie vylepšenia a zrýchlenia učenia sa dajú získať použitím učenia s Kalmanovou filtráciou.

Táto kapitola sa zameriavala na vznik stratégie hry, ale učenie s odmenou a trestom sa dá úspešne použiť aj pri úlohách riadenia a optimalizácie [9, 38]. Preto by bolo vhodné vyskúšať tu opísané prístupy aj na praktické úlohy automatizácie.

Poďakovanie: Táto časť publikácie vznikla s podporou grantu VG1/0141/10, VG VG1/0675/11, APVV-0233-10 a vďaka podpore v rámci OP Výskum a vývoj pre projekt: Výskum metód získavania, analýzy a personalizovaného poskytovania informácií a znalostí, ITMS: 26240220039, spolufinancovaný zo zdrojov Európskeho fondu regionálneho rozvoja.

Literatúra

[1] Billings D., Davidson A., Schaeffer J., Szafron D.: The Challenge of Poker. *Artificial Intelligence Journal*, **134** (2002) 201-240.

[2] Campbell M., Hoane Jr. A. J., Hsu F.: Deep Blue. *Artificial Intelligence*. **134** (2002) 57-83.

[3] Chellapilla K., Fogel D. B.: Co-Evolving Checkers Playing Programs using only Win, Lose, or Draw. *AeroSense99, Symposium on Applications and Science of Computational Intelligence II*, Vol. 3722, K. Priddy, S. Keller, D.B. Fogel, and J.C. Bezdek (eds.), SPIE, Bellingham, WA, Orlando, FL, pp. 303-312, 1999.

[4] Chellapilla K., Fogel D.B.: Evolving Neural Networks to Play Checkers without Expert Knowledge. *IEEE Trans. Neural Networks*, **10** (1999) 1382-1391.

[5] Chellapilla K., Fogel D.B.: Evolution, Neural Networks, Games, and Intelligence. *Proc. IEEE*, **87** (1999) 1471-1496,.

[6] Chellapilla K., Fogel D.B.: Evolving an Expert Checkers Playing Program without Using Human Expertise. *IEEE Transactions on Evolutionary Computation*, **5** (2001) 422-428.

[7] Čerňanský M., Benušková L.: Simple recurrent network trained by RTRL and extended Kalman filter algorithms. *Neural Network World*, **13** (2003) 223-234.

[8] Dazeley R.: *Investigations into playing Chess endgames using reinforcement learning*. Honours thesis, University of Tasmania, 2001.

[9] Hagen S. T., Kröse B.: Linear Quadratic Regulation using Reinforcement Learning. In: *Proc. BENELEARN98, 8th Belgian-Dutch Conference on Machine Learning*, pp. 39-46, 1998.

[10] Harmon M.E., Harmon S.S.: *Reinforcement learning: A tutorial*. 1996, dostupný aj na adrese: www.nbu.bg/cogs/events/2000/Readings/Petrov/rltutorial.pdf (14.7.2011)

[11] Haykin S.: *Kalman Filtering and Neural Networks*. John Wiley & Sons, New York, 2002.

[12] van den Herik H.J., Uiterwijk J. W.H.M., van Rijswijck J.: Games solved: Now and in the future. *Artificial Intelligence* **134** (2002) 277–311.

[13] Chaslot G. M. J-B., Winands M. H. M, Szita I., van den Herik H. J.: Cross-entropy for Monte-Carlo tree search. *ICGA Journal*, **31** (2008) 145-156.

[14] Chaslot G. M. J-B. et al.: Human-computer Go revolution 2008. *ICGA Journal*, **31** (2008) 179-185.

[15] Jacobs R. A., Jordan M. I., Nowlan S., Hinton G. E.: Adaptive mixtures of local experts. *Neural Computation*, **3** (1991) 79−87.

[16] Jordan M., Jacobs R.: Modular and hierarchical learning systems. *The Handbook of Brain Theory and Neural Networks*, MIT Press, Cambridge, MA, 1995, dostupné aj na www.cs.berkeley.edu/~jordan/papers/handbook.ps (14.7.2011)

[17] Julier S.J., Uhlmann J.K.: A New Extension of the Kalman Filter to Nonlinear Systems. *Proceedings of AeroSence: The 11th International Symposium on Aerospace/Defence Sensing, Simulation and Controls*, pp. 182-193, 1997.

[18] Kalman, R.E.: A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, **82** (1960) 35–45.

[19] Kvasnička V.: Adaptive mixture of local neural networks. *Neural Network World*, **3** (1993) 161-174.

[20] Kvasnička V.: Metódy prehľadávania a riešenia problémov. V knihe Kvasnička V., Pospíchal J.: *Informatika pre sociálne vedy*. Vydavateľstvo UK, Bratislava, 2005, dostupné na http://math.chtf.stuba.sk/FSEV_Informatika/doc/Kapitola03_lit.doc (15.4.2005)

[21] Kvasnička V. et al.: *Úvod do teórie neurónových sietí*. IRIS, Bratislava, 1997.

[22] Kvasnička V., Pospíchal J., Tiňo P.: *Evolučné algoritmy*. Vydavateľstvo STU, Bratislava, 2000.

[23] Lacko P., Kvasnička V., Pospíchal J.: An Emergence Of Game Strategy In Multiagent Systems. *International Journal of Computational Intelligence and Applications*, **4** (2004) 283-298.

[24] Lacko P.: Evolučná emergencia stratégie hry v multiagentových systémoch. V zborníku *Kognice a umělý život IV*. Slezská universita v Opavě, Opava, s. 333-351, 2004.

[25] Lacko P.: A Comparison of symbolic and subsymbolic approach in artificial intelligence. In: *MENDEL 2005: 11th International Conference on Soft Computing*. Brno University of Technology, Brno, pp. 163-168, 2005.

[26] Lacko P., Kvasnička V.: Mixture of Expert Used to Learn Game Play. In: *Artificial Neural Networks - ICANN 2008*. Lecture Notes in Computer Science, Vol. 5163, Springer Berlin/Heidelberg, pp. 225-234, 2008.

[27] Mišák P.: *Použitie neurónových sietí k tvorbe stratégie párovej symetrickej hry*. Diplomová práca, FIIT STU, 2008.

[28] Müller M.: Computer Go. *Artificial Intelligence*, **134** (2002) 145-179.

[29] Neumann J.V., Morgenstern O.: *Theory of Games and Economic Behaviour*. Princeton University Press, Princeton, NJ, 1944.

[30] Návrat P. et al.: *Umelá inteligencia.* Vydavateľstvo STU, Bratislava, 2000.

[31] Lucas S. M., Runarsson T.P.: Temporal difference learning versus co-evolution for acquiring othello position evaluation. *CIG Proceedings*, pp. 52–59, 2006.

[32] Patel G. S.: *Modeling Nonlinear Dynamics with Extended Kalman Filter Trained Recurrent Multilayer Perceptrons*. Diploma Thesis, McMaster University, 2000.

[33] Russell S.J., Norvig P.: *Artificial Intelligence: Modern Approach*. Prentice Hall, Upper Saddle River, HJ, 1995.

[34] Rybár J., Beňušková L., Kvasnička V.: *Kognitívne vedy*. Kaligram, Bratislava, 2002.

[35] Samuel A.L.: Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development* **3** (1959) 210-229.

[36] Schraudolph N.N., Dayan P., Sejnowski T.J: Learning to Evaluate Go Positions Via Temporal Difference Methods. In L.C. Jain & N. Baba (eds.): *Soft Computing Techniques in Game Playing*. Springer Verlag, Berlin, pp. 77–98, 2000.

[37] Schaeffer J., Lake R., Lu P., Bryant M.: Chinook is the world checkers champion. *AI Magazine*, **17** (1996) 21-29, elektronický dokument dostupný na adrese www.aaai.org/ojs/index.php/aimagazine/article/download/1208/1109 (14.7.2011)

[38] Singh S., Bertsekas S.: Reinforcement learning for dynamic channel allocation in cellular telephone systems. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *NIPS-9*, p. 974. The MIT Press, 1997.

[39] Sinčák P., Andrejková G.: *Neurónové siete. Inžiniersky prístup (1. diel)*. Elfa, Kosice, 1996, elektronický dokument dostupný na adrese http://neuron-ai.tuke.sk/cig/source/publications/books/NS1/html/all.html (14.7.2011)

[40] Sutton R.S.: Learning to Predict by the Method of Temporal Difference. *Machine Learning,* **3** (1988) 9-44.

[41] Sutton R.S., Barto A.G.: *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, 1998.

[42] Tesauro G.: Neurogammon wins computer olympiad. *Neural Computation*, **1** (1989) 321-323.

[43] Tesauro G.: Temporal Difference Learning and TD-Gammon. *Communications of the ACM*, **38** (1995) 58-67.

[44] Tesauro G.: Practical Issues in Temporal Difference Learning. *Machine Learning*, **8** (1992) 257-77.

[45] Tesauro G.: TD-Gammon, a Self-Teaching Backgammon Program, Achieves Master-Level Play. *Neural Computation*, **6** (1994) 215-9.

[46] Trebatický P.: *Trénovanie rekurentných neurónových sietí technikou rozšíreného Kalmanovho filtra*. Diplomová práca, FIIT STU, Bratislava, 2004.

[47] Trebatický P.: Recurrent Neural Network training with the Kalman Filter-based techniques. *Neural Network World*, **15** (2005) 471-488.

[48] Trebatický P., Pospíchal J.: Neural Network Training with Extended Kalman Filter Using Graphics Processing Unit. *Artificial Neural Networks - ICANN 2008*, Lecture Notes in Computer Science, Vol. 5164, pp. 198-207, 2008.

[49] Zaman R., Wunsch D.C.: TD methods applied to mixture of experts for learning 9x9 Go evaluation function. *IJCNN 1999*, pp. 3734-3739, 1999, elektronický dokument dostupný na http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=830746 (14.7.2011)

[50] Warren S. S.: *What is a softmax activation function*? 2002, elektronický dokument dostupný na http://www.faqs.org/faqs/ai-faq/neural-nets/part2/section-12.html (14.7.2011)

[51] Wu L., Baldi P.: Learning to play Go using recursive neural networks. *Neural Networks*, **21** (2008) 1392-1400.

[52] Zaman R., Wunsch D.C. III: TD methods applied to mixture of experts for learning 9x9 GO evaluation function. *IJCNN 99*, pp. 3734 – 3739, 1999.

[53] Schaul T., Schmidhuber J.: Scalable Neural Networks for Board Games. In Alippi C., Polycarpou M., Panayiotou C., and Ellinas G. (eds.): *Proceedings of the 19th International Conference on Artificial Neural Networks: Part I* (ICANN '09). Springer-Verlag, Berlin, Heidelberg, pp. 1005-1014, 2009.

[54] *ChessBase, Kramnik vs Deep Fritz: Computer wins match by 4:2* [online], 2006 [cit. 14.7.2011], dostupné na internete:

http://www.chessbase.com/newsdetail.asp?newsid=3524

[55] *ChessBase, The last man vs machine match*? [online], 2006 (cit. 14.7.2011), dostupné na internete: http://www.chessbase.com/newsdetail.asp?newsid=3504

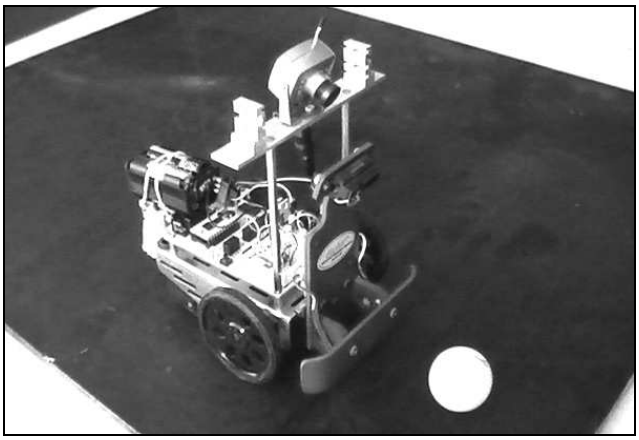
Multiagentový prístup k modelovaniu mysle - alebo ako sledovať pingpongovú loptičku

Andrej LÚČNY[1]

Abstrakt. Prezentujeme syntézu prác Marvina Minského, Rodneyho Brooksa a ich nasledovníkov. Na príklade riadenia mobilného robota, ktorého úlohou je sledovať pingpongovú loptičku, demonštrujeme multiagentový prístup k modelovaniu mysle. „Myseľ“ robota modelujeme ako decentralizovaný systém zložený z autonómnych a proaktívnych modulov – tzv. agentov – ktorého globálne správanie povstáva z ich vzájomných interakcií. Vysvetľujeme princípy ako situovanosť, subsumpcia a ďalšie.

1 Úloha

Za modelovú úlohu, ktorá nás bude sprevádzať celým výkladom, zvolíme napodobeninu známej hry s mačiatkom, ktoré naháňa imitáciu myši: mobilný robot vybavený dvojkolesovým podvozkom a kamerou bude sledovať pingpongovú loptičku zavesenú na šnúrke (Obr. 1).



Obr. 1. Modelová úloha: robot sledujúci pingpongovú loptičku.

[1] Katedra aplikovanej informatiky, FMFI UK, Bratislava, E-mail: lucny@fmph.uniba.sk

Táto málo užitočná ale zaujímavá úloha je – ako uvidíme neskôr – pekná z toho dôvodu, že ju možno dobre škálovať, t.j. stupňovať podmienky za ktorých ju má robot vykonávať i požiadavky na kvalitu jeho správania.

2 Klasický prístup k riadeniu robota

Pokiaľ znížime svoje nároky na kvalitu na minimum, na implementáciu takéhoto robota nám stačia základné znalosti z programovania, spracovania obrazu a ovládania motorov (Kód 1)[2].

```
SerialPort motors(comport,baudrate);
VideoCapture camera(0);
for (;;) {
    camera >> image;
    cvtColor(image,gray,CV_BGR2GRAY);
    GaussianBlur(gray,gray,Size(9,9),2,2);
    vector<Vec3f> circles;
    HoughCircles(gray,circles,CV_HOUGH_GRADIENT,2,50,200,100);
    if (circles.size() > 0) {
        float x = circles[0][0];
        float radius = circles[0][2];
        if (x < image.cols/2 – tolerance) motors << turnleft;
        else if (x > image.cols/2 + tolerance) motors << turnright;
        else if (radius < optimal – tolerance2) motors << forward;
        else if (radius > optimal + tolerance2) motors << backward;
        else motors << stop;
    }
    else motors << stop;
    image.release();
    delay(100); // wait 100 ms
}
```

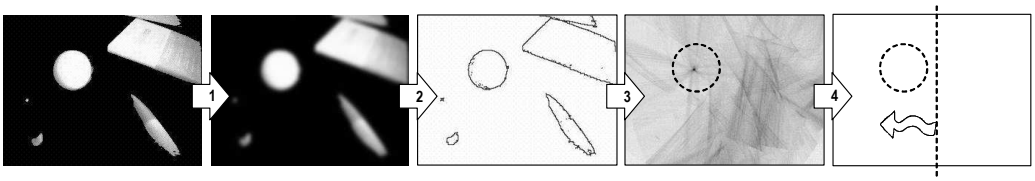
Kód 1. Ideový kód klasického riešenia (C++, OpenCV).

Robot je tu riadený tak, že opakovane snímame obraz a spracujeme ho na príkaz, ktorý vyšleme do podvozku robota. Celý proces spracovania vstupného obrázku na daný príkaz prebieha priamočiarou postupnosťou transformácii. Obrázok sa v prvom rade premení na čiernobiely. Potom ho trochu rozmažeme, čo je obľúbený a veľmi užitočný krok pred každou operáciou, ktorá pristupuje k obrazu ako keby bol spojitý (keďže to zníži šum). Potom vyvoláme Houghovu transformáciou[3], ktorá nám vráti

[2] Kódy a ďalšie materiály možno získať na www.agentspace.org/uiakv3

[3] Kľúčovou myšlienkou Houghovej transformácie je obrátenie procesu nakreslenia objektu určitých parametrov v binárnom obraze, s použitím hrubej sily. Tak ako z daného stredu a polomeru vieme vykresliť obraz kružnice, vieme aj z obrazu kružnice odhadnúť najpravdepodobnejší stred a polomer, z ktorých bola táto kružnica vykreslená. Odhad prebieha tak, že každý vykreslený bod hlasuje za všetky možné parametre kružnice, ktoré by ho mohli vykresliť (v rámci určitého rozsahu týchto parametrov). Následne sú tieto hlasy spočítané a parametre s najväčším počtom hlasov zodpovedajú hľadanej kružnici. Tento proces sa dá významne zrýchliť tým, že sa nehlasuje za všetky možné stredy, ale len za tie,

stredy a polomery kružníc na obraze (Obr. 2). Potom zoberieme prvú z nich a ak jej stred je príliš vľavo, vyšleme na motory príkaz na točenie vľavo, ak je príliš vpravo, tak točíme vpravo, ak je polomer príliš malý ideme dopredu, ak je príliš veľký, dozadu. Inak motory zastavíme.



Obr. 2. Jednotlivé fázy premeny čiernobieleho obrazu na motorický príkaz (1. rozmazanie, 2. Hranový detektor, 3. Houghova transformácia, 4. logika riadenia motorov)

Ostáva vysvetliť, aký význam má príkaz `delay(100)`, tj. čakanie 100 milisekúnd. Pokiaľ by sme ho nepoužili, náš program by okupoval procesor v maximálnej možnej miere. Hoci pri vstupno-výstupných operáciách (pri čítaní obrázku z kamery a vysielaní príkazu do podvozku) sa vykonávanie nášho programu na chvíľu zastaví, nestačí to na to, aby náš program ponechal priestor ďalším procesom v našom počítači. A to v prvom rade vláknam, ktoré sami nekódujeme, ale sú súčasťou knižníc ktoré linkujeme. Ľahko by sme sa o potrebe explicitného čakania presvedčili, keby sme si v našom kóde dali zobraziť obraz `gray`, vložením príkazu `imshow("gray",gray);` – pokiaľ teraz zakomentujeme `delay()`, nebudeme v zobrazovacom okne pre tento obraz nič vidieť (hoci obraz do tohto okna posielame, jeho zobrazenie beží v inom, pre nás neviditeľnom vlákne a to sa nedostane ku slovu). Je to preto, že sme pripustili stav, ktorý je z logického hľadiska v poriadku, ale z hľadiska reálneho vykonávania nášho programu v čase, v poriadku nie je. Ide o tzv. LiveLock, jeden zo základných problémov, s ktorými každý systém pracujúci v reálnom čase zápasí.

Zaujímavá je taktiež otázka, ako veľké má byť toto čakanie. Zdola je ohraničené výkonom procesora, vstupno-výstupných zariadení ako aj našou potrebou prenechávať kapacitu procesora iným procesom. Nemá napríklad zmysel ísť tu na 1 milisekundu: hoci procesoru by aj to pomohlo, kamera nám nedáva nové obrázky v takej frekvencii. Ešte zaujímavejšie však je, že toto číslo je ohraničené aj zhora, pričom tento limit je daný rýchlosťou motorov robota. Pre procesor by bolo lepšie, keby sme čakali napríklad celú sekundu – zatiaľ by však robot mohol pri otáčaní sa za loptičkou urobiť aj čelom vzad a úplne ju stratiť z dohľadu! Tomuto problému sa dá čeliť tým spôsobom, že motormi nebudeme hýbať kontinuálne, ale každý príkaz bude vykonávaný len určitý vopred stanovený čas. Takže ak aj príkaz na otočenie generujeme len raz za sekundu, robot sa bude točiť vždy len 100 milisekúnd

---

ktoré ležia na kolmici ku hrane, na ktorej sa nachádza hlasujúci bod. Úvodnou fázou tejto transformácie je teda jednak nájdenie bodov hrán (odpočítavaním jasu vzájomne blízkych bodov v smere x a y nájdeme hrubé hrany, prahovaním ich binarizujeme a ďalej upravujeme iteratívnym procesom stenčovania a orezávania), jednak ich orientácie v týchto bodoch (smernica ich normály je daná podielom diferencií v smere x a y) Viď [5] [16].

a zvyšných 900 milisekúnd bude stáť. Tento trik sa v robotike aj všeobecne používa a vyplývajú z neho charakteristické trhané pohyby, radšej rýchle, ale trvajúce krátky čas, prekladané pauzami v ktorých sa nič nedeje[4]. Pokiaľ čitateľovi toto riešenie stačí, môže rovno prejsť na ďalšiu kapitolu tejto knihy. My to takto robiť nechceme. Pre nás príkaz vyslaný na motory znamená, že sa vykonáva, až kým nevyšleme príkaz nový. Produkovaný pohyb robota je takto plynulý, prirodzený a podobný pohybu živých tvorov. Na druhej strane náš program takto svojim spôsobom „jazdí na neosedlanom koňovi“ a ak sa pri svojich výpočtoch niekde zdrží, „hrozí mu pád“. V ďalšom sa budeme snažiť ukázať ako sa dá „na tomto koni udržať aj bez sedla“.

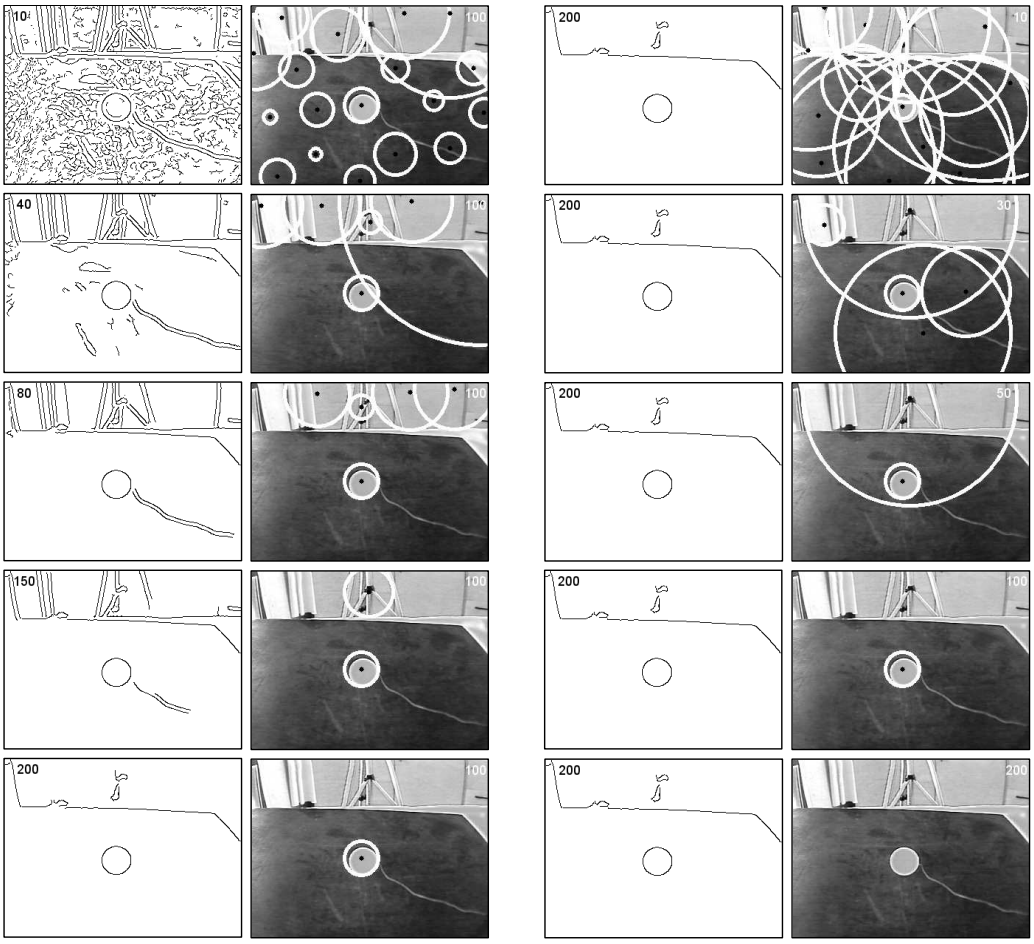
Na záver ešte spomeňme, že pri zvolenom riešení nebudú generované motorické príkazy perfektne pravidelné. Samotný výpočet príkazu totiž zaberie pre rôzne obrázky rôzne dlhý čas. Pauza má naopak perfektne rovnakú dĺžku. Oveľa krajšie by bolo, keby sme vedeli pre generovanie príkazu zabezpečiť stálu frekvenciu. Vtedy by sa dĺžka pauzy musela prispôsobiť dĺžke výpočtu. V praxi to znamená, že musíme pracovať so systémovými hodinami.

Aký nevinný príkaz – `delay(100);` – a koľko umenia sa v ňom skrýva!

3 Potreba paralelizmu a výmeny dát

Ako sme už spomenuli, zvolená úloha sa dá stupňovať do zložitejších podôb. Jeden zdroj takého stupňovania vidíme pri volaní Houghovej transformácie, kde vystupujú štyri konštanty (`2,50,200,100`). Prvá určuje presnosť s akou určujeme parametre kružnice, druhá minimálnu vzdialenosť stredov dvoch detekovaných kružníc. Tretia určuje hodnotu prahu pri hľadaní hrán, ktoré sa budú brať v úvahu a posledná prah od ktorého berieme v úvahu zdetekovanú kružnicu. Najmä tretia a štvrtá konštanta majú veľký vplyv na to, či sa podarí loptičku na obraze správne nájsť a pritom ich optimálne hodnoty nemožno vopred vedieť (Obr. 3), lebo závisia od podmienok na scéne, v ktorej sa robot nachádza. Tieto podmienky sa menia nielen zmenou osvetlenia scény, ale aj samotným pohybom robota v nej. Bolo by preto vhodné vyskúšať viac potenciálnych prahov a hľadať ich optimálne hodnoty, pri ktorých zdetekujeme niekoľko málo kružníc. To samozrejme môžeme spraviť aj v rámci klasického riešenia tak, že zakaždým v cykle vyskúšame veľa rôznych hodnôt prahov a podľa vráteného počtu kružníc budeme vedieť ktorý výsledok máme zobrať v úvahu pre generovanie motorického príkazu. Lenže týmto sa jeho generovanie značne spomalí a my môžeme mať – pri kontinuálnom hýbaní s motormi – problémy s rýchlosťou reakcií, ako aj s ich nízkou frekvenciou. Samozrejme ak raz vyhodnotíme hodnotu optimálneho prahu, môžeme predpokladať, že istý čas sa táto hodnota nezmení, alebo zmení len málo. Mohli by sme teda niekoľko najbližších krokov vykonať s rýchlymi reakciami (spoliehajúc sa na jedinú hodnotu prahu) a potom si dať jednu pomalú reakciu, pri ktorej hodnotu prahu prehodnotíme a pohyby robota radšej na chvíľu zastavíme.

[4] Túto vlastnosť bravúrne a inštinktívne postrehli herci, ktorí práve takéto pohyby robia, keď chcú vyvolať v divákovi dojem, že sú robotmi.



Obr. 3. Vplyv konštánt na výsledok Houghovej transformácie. Vľavo: vplyv prahu na detekciu hrán (viď číslo v ľavom hornom rohu), vpravo vplyv prahu na akceptovanie potenciálnej kružnice (viď číslo v pravom hornom rohu).

Čuduj sa svete, zase sme sa dostali k tomu charakteristickému trhanému pohybu. Ako sa mu vyhnúť? Pri spôsobe programovania, keď kladieme do sekvencie príkaz za príkazom, to proste nie je možné[5]. Musíme si uvedomiť, že riadenie robota musí do určitej miery napodobňovať procesy, ktoré sa dejú v našej mysli. A niet charakteristickejšej vlastnosti pre tieto procesy, než je ich paralelizmus. Hoci aj nemáme paralelný počítač, musíme vedieť tento paralelizmus napodobniť v tom, že pomalé procesy (ako je v našom prípade hľadanie optimálneho prahu) nesmú brzdiť vykonávanie rýchlejších procesov (ako je generovanie motorického príkazu, keď už

[5] Jedna škaredá možnosť by tu ešte bola a to naprogramovať oba procesy ako tzv. podprogramy s viacnásobným prístupom, tj. rozbiť ich kód na malé úseky a volať v správny čas správny úsek. To už ale zavádzame do aplikačného programu primitívne plánovanie procesov, čo sa patrí ponechať operačnému systému. Pri tomto type programovania sa kód stáva veľmi rýchlo neprehľadný. Preto táto stratégia nie je vhodná pre komplexnejšie systémy.

tento prah poznáme). Aktualizácia výsledkov pomalých procesov v systéme musí mať nižšiu frekvenciu než aktualizácia výsledkov procesov rýchlych. (Cieľom je, aby sme radšej reagovali rýchlo s neoptimálnym prahom, než nereagovali, čakajúc na optimálnu hodnotu. Rýchla a menej presná reakcia sa dá dodatočne zdokonaliť vo chvíli, keď je už optimálna hodnota známa.)

Takže už pri tak jednoduchej komplikácii našej úlohy, akou je potreba zvládnuť rôzne svetelné podmienky na scéne, vidíme potrebu paralelného behu viacerých procesov (zatiaľ dvoch, ale je snáď jasné, že pri stúpajúcich nárokoch na správanie robota, to pri tomto čísle neostane). Okrem toho vidíme potrebu prevádzkovať v istom súlade rýchle a pomalé procesy. A vidíme aj potrebu, aby si tieto procesy vzájomne vymieňali dáta (v našom prípade hodnotu optimálneho prahu).

4 Multiagentový prístup

K potrebe paralelizmu procesov a výmeny dát medzi nimi sa prepracuje tvorca každého zložitejšieho robota. Obvykle ju však vyrieši takzvane ad-hoc. V závislosti od platformy, ktorú má k dispozícii využije multiprocesové alebo multivláknové prostredie, ktoré mu vytvára operačný systém, spoliehajúc sa na spravodlivé (preemptívne) striedanie procesov či vlákien v procesore. Výmenu dát potom zrealizuje s využitím niektorého zo synchronizačných mechanizmov (semafory a pod.), keďže nie je ani tak problém dáta vymieňať, ako pri tejto výmene zachovať ich korektný obsah[6]. Tu sa väčšina vývojárov zastaví a pokojne využije na dosiahnutie tohto cieľa v rámci jedného systému rôzne techniky. My však s menšinou pôjdeme ďalej a budeme požadovať normalizáciu použitého mechanizmu.

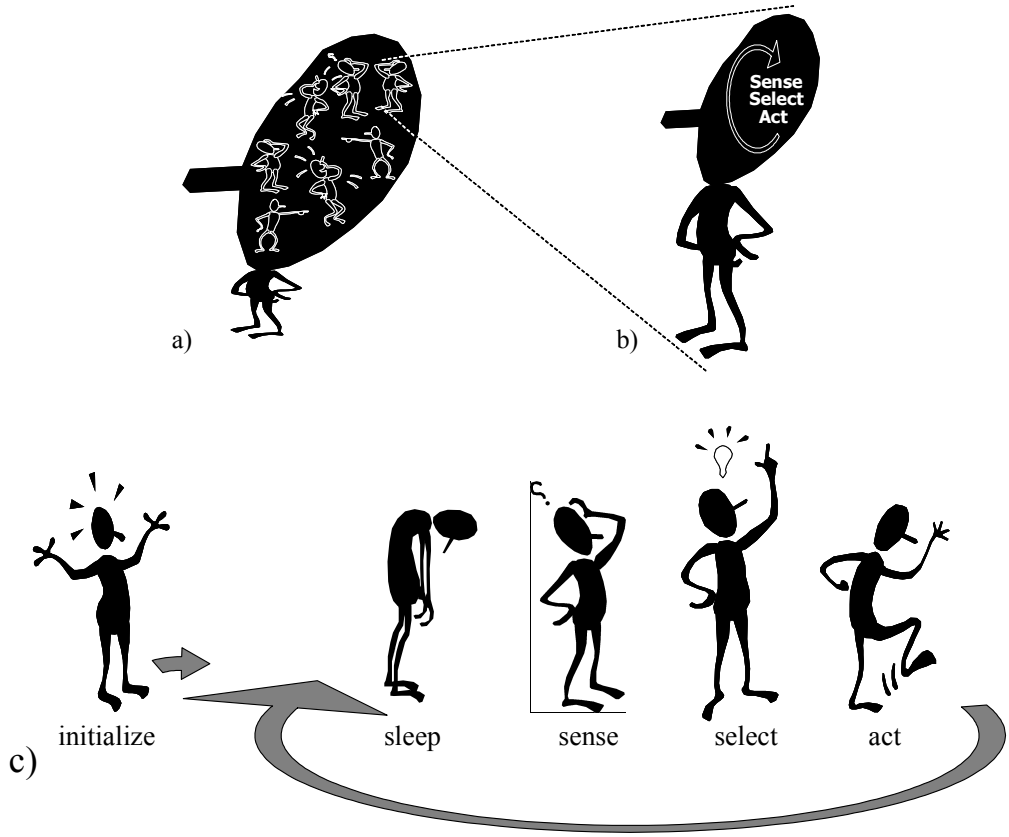
Jednu z možností takejto normalizácie poskytuje tzv. multiagentový prístup. Na akýkoľvek systém pri ňom pozeráme ako na distribuovaný, t.j. skladajúci sa z modulov, ktoré musia spolu na diaľku komunikovať, aby si vymieňali dáta[7] (Obr. 4a). Vývojár tohto systému potom miesto explicitnej implementácie správania systému, explicitne implementuje správanie jeho jednotlivých modulov a to tak, aby globálne prejavy systému, ktoré z jednotlivých správaní týchto modulov vyplynú, zodpovedali želanému správaniu. Navyše pri multiagentovom prístupe sa tieto moduly vyznačujú tým, že majú vlastné riadenie, t.j. ide o paralelne bežiace procesy, o tzv. agentov (Obr. 4b). Samozrejme pokiaľ takýto systém prevádzkujeme na jednom počítači, paralelný beh bude len napodobnený a žiadny agent zďaleka nesmie vyčerpať všetok strojový čas (to by nastal spomínaný LiveLock) – inými slovami väčšinu času musí prespať. Čo sa týka tvaru kódu, každý agent vyzerá podobne: po inicializačnej fáze vbehne do nekonečného cyklu, v ktorom zaspí, pri zobudení vníma v akej situácii sa ocitol, zvolí si, či a čo v danej situácii podnikne a vykoná to (Obr. 4c). To je samozrejme obrazne povedané[8]: vnímanie tu spočíva v prenose dát do agenta, voľba vo

[6] Hrozí tu, že viaceré súbežné procesy naraz modifikujú rovnaký dátový priestor a jeho obsah sa tým stáva nekonzistentný

[7] Pokiaľ celý náš systém beží na jednom počítači napĺňame tak obsah sloganu „The computer is the network!"

[8] a to rečou typickou pre oblasť známu ako „multiagentové systémy". Viď napr. [6] [9] [18]

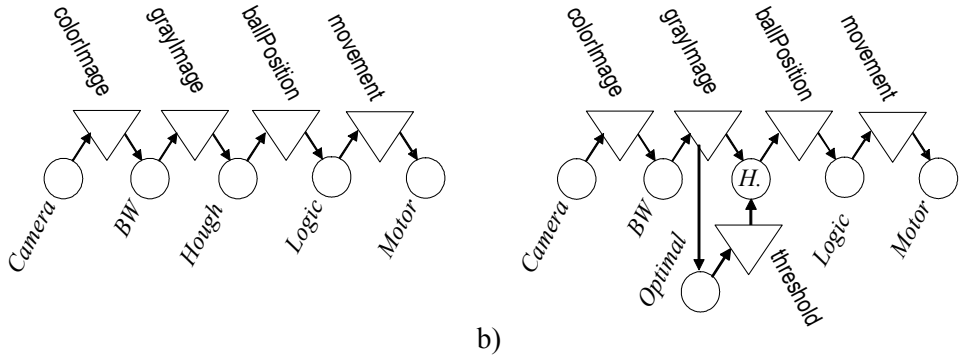
výpočte, ktorý tieto dáta spracúva a vykonanie akcií v prenose dát z agenta. Prenos dát pritom môže vo všeobecnosti prebiehať dvomi spôsobmi: priamo, t.j. agent pošle dáta druhému agentovi a nepriamo, t.j. agent umiestni dáta do prostredia medzi agentmi, odkiaľ si ich iní agenti môžu prevziať. My sa sústredíme výhradne na druhý spôsob[9], t.j. na nepriamu komunikáciu, lebo lepšie vyhovuje našim zámerom.



Obr. 4. a) Multiagentový systém. b) Agent. c) Životný cyklus agenta.

Ako teraz pristúpime k tvorbe systému, ktorý riadi nášho robota, čo sleduje pingpongovú loptičku? V prvom rade prerobíme pôvodné riešenie tak, že sekvenciu, ktorá spracúva obraz rozbijeme na niekoľko agentov, ktorí si v analogickej sekvencii budú posúvať dáta (Obr. 5a). Zamýšľame totiž túto sekvenciu postupne vetviť (v prvom kroku pridaním hľadania optimálneho prahu – Obr. 5b) a neskôr dokonca vytvárať v rozrastajúcej sa dátovej výmene cykly.

[9] náš prístup bude založený na vlastnej architektúre, ktorú sme nazvali Agent-Space [13]

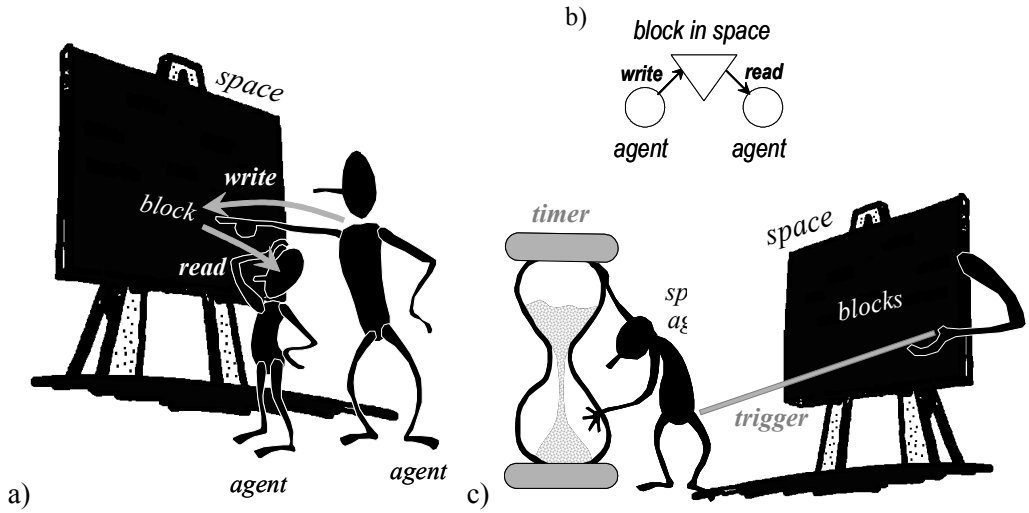


Obr. 5. a) Multiagentový systém analogický pôvodnému riešeniu. b) Rozšírenie, pri ktorom pomalý agent *Optimal* nezdržuje rýchlejšieho agenta *Hough* (*H.*). (Agenti sú zobrazení kruhom, komunikované dáta trojuholníkom.)

Z hľadiska implementácie je každý agent realizovaný samostatným vláknom[10]. Ostáva vysvetliť ako sú tieto vlákna synchronizované a akým štandardným mechanizmom si vymieňajú dáta. Ako sme už spomenuli, základným princípom tejto dátovej výmeny je pre nás výlučne nepriama komunikácia, je teda založená na tzv. čiernej tabuli (blackboard)[11], ktorú nazývame – v súlade so zaužívanom názvom v tzv. koordinačnom programovaní – *space*. Táto tabuľa je spoločná pre všetkých agentov a je možné na ňu umiestniť pomenované dáta rôzneho formátu, tzv. bloky. Základnú manipuláciu s blokmi predstavuje na každej takejto tabuli zápis a čítanie, t.j. jeden agent blok zapíše a druhý agent ho prečíta (Obr. 6a,b). Zápis bloku bude samozrejme prebiehať vo fáze vykonávania akcií (*act*), otázka je však na základe čoho sa uskutoční čítanie. Keďže agenti priamo nekomunikujú, čitateľ bloku nevie, ani kto, ani kedy zapísal na tabuľu novú informáciu. Zapisujúci agent sa teda nijako nepričiní o to, aby sa čítajúci agent zobudil (z fázy *sleep*) a blok prečítal (v rámci fázy *sense*). Ostávajú mu preto len dve možnosti. Môže sa chodiť pravidelne pozerať na tabuľu, čo je účinná stratégia, keď vie, že za normálnych okolností sa tam pravidelne objavujú nové dáta. Vtedy používa tzv. časovač (*timer*). S touto stratégiou by vystačil napríklad agent BW, ktorý vie, že kamera poskytuje obrázky raz za 100ms. Keby ju však použili všetci agenti v celej sekvencii, spracovanie obrazu by sa zbytočne oneskorovalo o čas, ktorý uplynie medzi zápisom a čítaním. Preto má čítajúci agent k dispozícii aj ďalší mechanizmus zvaný spúšťač (*trigger*). Spúšťač je realizovaný tabuľou, ktorú na úvod čítajúci agent požiada, aby ho zobudila, keď sa zmení obsah bloku s určitým menom. Keď zapisujúci agent spôsobí, že táto udalosť nastane, tabuľa čítajúceho agenta zobudí a ten si hodnotu daného bloku prečíta (Obr. 6c).

[10] Čo je pochopiteľne menej efektívne riešenie, volíme ho pre jeho schopnosť ďalšieho škálovania, t.j. jeho rozširovania na komplexnejší systém

[11] Ideovo ju možno považovať za derivát tzv. tupple-space z jazyka LINDA[8], navrhnutého pre programovanie paralelných počítačov, ktorý našiel uplatnenie v rámci tzv. koordinačného programovania [4], i v rámci technológie Java Jini spoločnosti Sun ako tzv. Java Space [17].



Obr. 6. a) Nepriama komunikácia medzi agentmi. b) Schéma dátového toku medzi dvomi agentmi. c) Budenie agenta časovačom (timer) a spúšťačom (trigger).

Ostatné aspekty manipulácie s blokmi popíšeme, až keď ich budeme potrebovať (je ich ako uvidíme pomerne dosť a niektoré by pri skutočnej tabuli vyvolávali ešte zázračnejší dojem ako spúšťače – naša tabuľa je však len kus softvéru vo virtuálnom svete a môžeme jej dať také schopnosti ako sa nám hodí). Skúsme sa teraz pozrieť ako bude vyzerať kód agentov realizujúcich riadenie nášho robota (Obr. 5a).

```
class AgentCamera : public Agent {
    protected:
        VideoCapture camera;
        void init (string args) {
            camera.open(0);
            timer_attach(100,100);
        }
        void sense_select_act (int pid) {
            Mat image;
            camera >> image;
            space_write("colorImage",image.clone(),200);
        }
    public:
        AgentCamera (string args) : Agent(args) {};
};
```

Kód 2. Kód agenta, ktorý realizuje príjem obrazu z kamery (C++, OpenCV, AgentSpace)

Prvý z nich – agent *Camera*, bude budený časovačom a v pravidelných taktoch bude preberať obraz z kamery a vkladať ho na tabuľu – *space* (Kód 2). Pritom je zaujímavé všimnúť si niekoľko vecí. V prvom rade, vidíme, že agent je definovaný ako odvodená trieda od triedy *Agent*, ktorá je aplikačne nezávislá a implementuje v sebe všetky

potrebné (a dosť krkolomné) mechanizmy: vlákna, časovače, tabuľu a spúšťače, ktoré by vývojár bez využitia knižnice *AgentSpace* musel do istej miery urobiť sám a to s každou novou aplikáciou. Nadradená trieda *Agent* zriadi vo svojom konštruktore samostatné vlákno (*pthread*), ktoré vykoná metódu *init()* a potom v cykle pomocou synchronizačného zámku (*mutex*) zaspí a po zobudení zavolá metódu *sense_select_act()*. Prekrytím týchto dvoch metód preto vývojár zimplementuje plnohodnotného agenta. Ďalej je zaujímavé si uvedomiť, že do *space* je potrebné vložiť nie ten objekt, s ktorým lokálne v agentovi pracujeme, ale jeho klon. On sám totiž môže byť po vykonaní zápisu modifikovaný a nakoniec zanikne ako náhle riadenie opustí *sense_select_act()*. *AgentSpace* preto automaticky pri zápise klonuje tzv. hlbokým klonovaním všetko, čo vie naklonovať samotná platforma. Existujú však zložité objekty – ako je napríklad obraz – ktoré vnútorne odkazujú na alokované dáta a o ich klonovanie sa musí postarať vývojár. Samozrejme klonovanie, podobne ako ďalšie mechanizmy *AgentSpace*, predstavuje pre systém určitú záťaž. V odôvodnených prípadoch sa dá do Space zapísať aj smerník a synchronizácia prístupov ako aj deštrukcia zapísaného objektu je potom ponechaná na vývojárovi. Podobný problém nastáva pri čítaní zo space, pretože viac agentov môže prečítať ten istý objekt. Vývojár ho preto nesmie modifikovať – a ak by to potreboval, potom ho musí naklonovať.

```
class AgentBW : public Agent {
    protected:
        void init (string args) {
            trigger_attach("colorImage");
        }
        void sense_select_act (int pid) {
            Mat color, gray, blank(240,320,CV_8UC3);
            color = space_read("colorImage",blank);
            cvtColor(color,gray,CV_BGR2GRAY);
            space_write("greyImage",gray.clone(),120);
        }
    public:
        AgentBW (string args) : Agent(args) {};
};
```

Kód 3. Kód agenta, ktorý realizuje transformáciu farebného obrazu na čiernobiely (C++, OpenCV, AgentSpace)

Najjednoduchší z agentov je agent *BW*, ktorý realizuje transformáciu farebného obrazu na čiernobiely (Kód 2.). Mohlo by sa zdať, že tento kód by bolo lepšie včleniť priamo do agenta *Hough*, lenže čiernobiely obraz je dosť častým vstupom pre rôzne metódy spracovania obrazu a s pribúdajúcimi agentmi, by sme ho stále znovu a znovu zbytočne počítali. Na rozdiel od agenta *Camera*, *BW* je budený spúšťačom, t.j. jeho synchronizačný zámok sa odblokuje zápisom do *space* (v našom prípade to bude práve vlákno agenta *Camera*, ktoré zápisom farebného obrazu odblokuje vlákno agenta *BW* a podnieti ho, aby z tohto obrazu vypočítal obraz čiernobiely).

Na prácu agenta *BW* priamo nadväzuje agent *Hough*, ktorý zo *space* prečíta čiernobiely obraz a zapíše do neho pozíciu loptičky (Kód 4.). Tu nastáva dilema

spočívajúca v tom, že agent *Hough* niekedy nájde a inokedy nenájde loptičku. Má teraz na výber dve stratégie. Keby chcel zobudiť agenta *Logic* spúšťačom, t.j. analogicky ako bol budený on sám i agent *BW*, musel by v prípade, že loptičku nenájde zapísať do *space* hodnotu „neznáma“ a *Logic* by sa na to zobudil a vydal by pokyn na zastavenie robota. Alternatívne by mohol hodnotu zo *space* zmazať, pričom vidíme, aké je dôležité, aby aj zmazanie bloku zo *space* vyvolávalo spúšťače zaregistrované na daný blok.

```
class AgentHough : public Agent {
    protected:
        void init (string args) {
            trigger_attach("grayImage");
        }
        void sense_select_act (int pid) {
            Mat blank(240,320,CV_8UC1), gray = space_read(„grayImage“,blank);
            HoughCircles(gray,circles,CV_HOUGH_GRADIENT,2,50,200,100);
            if (circles.size()>0) space_write("ballPosition",circles[0],300);
        }
    public:
        AgentHough (string args) : Agent(args) {};
};
```

Kód 4. Kód agenta, ktorý realizuje Houghovu transformáciu (C++, OpenCV, AgentSpace)

Niežeby to nefungovalo, ale v našom kóde sme použili lepšiu stratégiu spočívajúcu vo využití mechanizmu časovej platnosti[12]. Pozorný čitateľ si iste v našich kódoch pri zápisoch do *space* všimol dosiaľ nekomentovaný tretí parameter, ktorý napríklad pre zápis *grayImage* nadobúda hodnotu 120. Čo to je? Ako sme spomenuli, naša tabuľa je softvérová a z toho dôvodu sa dokáže – keď je to potrebné – správať aj zázračným spôsobom. Okrem spúšťačov, dokáže realizovať aj mechanizmus, ktorý zapisujúcemu agentovi umožní definovať čas, po uplynutí ktorého dáta z tabule bez jeho pričinenia zmiznú. Čiže tých 120 znamená, že keby sme stopli agenta *Camera*, tak ešte 120 milisekúnd budú mať agenti k dispozícii čiernobiely obraz (normálne ho aktualizujeme každých 100 ms), pričom po tento čas sa nebude meniť. Bolo by ale tragédiou, keby to trvalo dlhšie, lebo ak by napríklad v tomto obraze bola loptička vľavo, robot by sa stále točil doľava a nič by mu nepomohlo sa z tohto točenia dostať, keďže v jeho vnútornej reprezentácii by už jeho pohyb nemal na vnímaný obraz žiaden vplyv. Samozrejme o odstránenie tohto obrazu zo *space* by sa mohol postarať aj niektorý z agentov, nič menej to by bola zbytočná komplikácia ich kódu (napríklad keby to bol agent BW, musel by byť budený aj časovačom, aj spúšťačom – čo sa síce dá – vďaka tomu, že *sense_select_act()* dostáva ako parameter identifikátor *pid* (*proxy id)*, porovnateľný z návratovou hodnotu *trigger_attach()* respektíve *timer_attach()* – ale je to zmätok). Ďaleko jednoduchšie je využiť mechanizmus časovej platnosti.

Ako teda tento mechanizmus využije agent *Hough*? Namiesto toho, aby v prípade, že loptičku na obraze nenájde, mazal z tabule svoj predchádzajúci názor, zapíše tam

[12] Prvý krát bol tento mechanizmus navrhnutý pre Java Space ako tzv. „data leasing“ [17]

radšej tento názor s obmedzenou časovou platnosťou a to takou, aby v prípade, že loptičku kontinuálne detekuje, na tabuli jeho názor vždy bol. V našom prípade sme zvolili 300 milisekúnd, čo zodpovedá náročnosti výpočtu Houghovej transformácie.

Z tohto čísla zároveň vidíme, že agent *Hough* nemusí nevyhnutne stihnúť spracovať každý snímok ktorý príde od kamery. V tradičnom riešení (Kód 1) sa to vyrieši tým, že sa frekvencia záberov z kamery prispôsobí výpočtovým nárokom Houghovej transformácie. Tu naopak tento vplyv nemáme a preto ak paralelne pustíme rýchleho agenta, ktorý by loptičku detekoval nie podľa tvaru ale napríklad podľa farby, mohol by pracovať na rovnakej frekvencii ako kamera.

```
class AgentLogic : public Agent {
    protected:
        void init (string args) {
           attach_trigger("ballPosition"); // attach_timer(100,100)
        }
        void sense_select_act (int pid) {
            Vec3f middle(cols/2,2*row/3,optimal);
            Vec3f circle = space_read("ballPosition",middle);
            float x = circle[0];
            float radius = circle[2];
            string motion;
            if (x < cols/2 - tolerance) motion = turnleft;
            else if (x > cols/2 + tolerance) motion = turnright;
            else if (radius < optimal - tolerance2) motion = forward;
            else if (radius > optimal + tolerance2) motion = backward;
            else motion = stop;
            space_write("movement",motion,100);
        }
    public:
        AgentLogic (string args) : Agent(args) {};
};
```

Kód 5. Kód agenta, ktorý realizuje logiku pohybu robota (C++, AgentSpace)

Kód agenta Logic (Kód 5) uvádzame bez definovania niektorých konštánt. Obsahuje jednu zaujímavú fintu. Pri čítaní zo *space* je oveľa lepšie než kontrolovať, či sa hodnotu podarilo prečítať (požadovaná hodnota tam ešte nemusí alebo už nemusí byť), definovať náhradnú hodnotu, ktorú čítanie v takom prípade vráti. To je ten dosiaľ ignorovaný druhý parameter *space_read()*. Čiže miesto toho, aby sme vetvili kód pre prípad, že máme loptičku a nemáme loptičku, budeme v prípade, že loptičku nemáme, predpokladať, že je na správnom mieste. Keďže je tento agent budený spúšťačom, tak v momente kedy sa loptička stratí, agent *Hough* prestane aktualizovať blok *ballPosition*, 300 ms sa nič neudeje, potom platnosť *ballPosition* vyprší, načo sa agent *Logic* zobudí (rovnako ako keby prišla nová hodnota), nadobudne dojem, že loptička je v správnej polohe, vydá príkaz *stop* a nič sa nedeje, až kým *Hough* neuvidí loptičku znovu.

Alternatívne by agent *Logic* mohol byť budený časovačom s nízkou frekvenciou (viď zakomentovaný príkaz v init(), kód 5). V takom prípade by bol motorický príkaz

generovaný kontinuálne, došlo by však k určitému (hoci akceptovateľnému) oneskoreniu jeho vykonania.

Na záver tohto príkladu uvádzame kód agenta *Motor* (Kód 6), ktorý len zoberie príkaz na pohyb a vykoná ho na motoroch tým, že ho zapíše do komunikačného portu. Z neho ide do podvozku robota, kde je prečítaný v palubnom procesore a adekvátne je podľa neho generovaná frekvencia, ktorou sú ovládané príslušné servo motory. Jediný nový prvok tu spočíva vo využití odovzdávania argumentov – odovzdávame takto meno komunikačného portu.

```
class AgentMotor : public Agent {
    protected:
        SerialPort motor;
        void init (string args) {
            motor.open(args,9600/*bps*/);
            attach_trigger("movement");
        }
        void sense_select_act (int pid) {
            string motion = space_read("movement",stop);
            motor << motion;
        }
    public:
        AgentMotor (string args) : Agent(args) {};
};
```

Kód 6. Kód agenta, ktorý realizuje logiku pohybu robota (C++, AgentSpace, SerialPort)

Ostáva všetkých agentov naštartovať. To sa našťastie urobí pomerne ľahko, lebo medzi agentmi nie sú žiadne formálne väzby či závislosti. Jedine po naštartovaní nesmieme hneď ukončiť beh programu, ale musíme ho zablokovať, napríklad na vstup od užívateľa (Kód 7.)

```
int main() {
    AgentCamera Camera("");
    AgentBW BW("");
    AgentHough Hough("");
    AgentLogic Logic("");
    AgentMotor Motor("COM40");
    getch(); // don’t exit
}
```

Kód 7. Príklad naštartovania systému (C++, AgentSpace)

V tejto chvíli máme hotové multiagentové riešenie analogické tradičnému. Zatiaľ sme z toho, že sme pár riadkov kódu nahradili desaťnásobkom nemali žiadnu výhodu. Teraz však taká chvíľa prichádza. Pozrime sa teda na to, ako zrealizujeme rozšírenie systému na Obr. 5b. V prvom rade musíme z kódu vyňať natvrdo nakódované konštanty prahov a umiestniť ich na tabuľu (*space*) a urobiť z nich tak parametre. Je všeobecne poučné, že pre potenciálne rozširovanie systému je vhodné všetky takéto

konštanty už preventívne držať v *space* – hoci to opäť zvyšuje réžiu, keďže ich odtiaľ treba čítať. Preberanie parametrov zo *space*, ktoré sa menia iba zriedka sa síce dá efektívne napísať pomocou spúšťačov (Kód 8a), nič menej oveľa krajší a udržateľnejší kód agentov budeme mať práve vtedy, ak ich budeme čítať pri každom použití (Kód 8b).

```
a)  int cid;
    int ct;
    void init (string args) {
        trigger_attach("grayImage");
        cid = trigger_attach("cannyThreshold");
        ct = 200;
    }
    void sense_select_act (int pid) {
        if (pid == cid) ct = space_read("cannyThreshold",200);
        else {
            Mat blank(240,320,CV_8UC1), gray = space_read("grayImage",blank);
            HoughCircles(gray,circles,CV_HOUGH_GRADIENT,2,50,ct,100);
            if (circles.size()>0) space_write("ballPosition",circles[0],300);
        }
    }

b)  void init (string args) {
        trigger_attach("grayImage");
    }
    void sense_select_act (int pid) {
        int ct = space_read("cannyThreshold",200);
        Mat blank(240,320,CV_8UC1), gray = space_read("grayImage",blank);
        HoughCircles(gray,circles,CV_HOUGH_GRADIENT,2,50,ct,100);
        if (circles.size()>0) space_write("ballPosition",circles[0],300);
    }
```

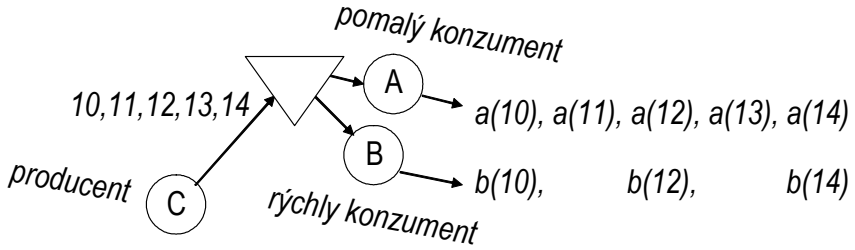
Kód 8. Preberanie parametrov: a) spúšťačmi (reaktívne), b) zakaždým (čisto reaktívne)

Je zaujímavé si všimnúť, že agentov v tomto príklade odlišuje nielen použitie spúšťačov, ale aj vlastnosť týkajúca ich tzv. vnútorného stavu. Kým agent nakódovaný v 8a) obsahuje informáciu, ktorá v ňom trvalejšie prežíva (tj. parameter *ct*), agent z 8b) si nič dlhodobo nepamätá. V zmysle teórie behaviorálnych systémov ide o tzv. čisto reaktívneho agenta [2]. Pre vývojárov takéhoto systému je čistá reaktivita všetkých agentov ideálom: niečím čo sa zriedka podarí dosiahnuť, ale čím bližšie sa k tomu dostaneme, tým lepšie. Vtedy sme si totiž istí, že v žiadnom agentovi nemáme schovanú a nedobytnú informáciu, ktorú by sme pri ďalších rozšíreniach systému potrebovali zmeniť a nebudeme mať na to iné prostriedky, než sa vrátiť k úprave kódov už nakódovaných a odladených agentov – tak ako sme to my museli urobiť v prípade vytiahnutia parametra z agenta *Hough*. V našom prípade šlo o triviálnu úpravu, takže žiadny problém – ale nemusí to byť vždy tak.

Po úprave agenta *Hough* už nebude problém pridať ďalšieho agenta *Optimal* (Obr. 5b), ktorý podobne ako *Hough* preberá čiernobiely obraz, ale nepočíta na ňom jednu transformáciu, ale viacero, pričom skúša nájsť takú hodnotu prahu pri ktorej ešte

nejaké kružnice zdetekované má, ale keby zdvihol prah ešte vyššie, už nemá. Nijako sa pritom neponáhľa. Vie, že nie je ten najdôležitejší agent a tak si dáva aj úmyselné pauzy, nech dopraje strojový čas iným. Na rozdiel od agenta *Hough* sa vôbec netrápi preto, že nestihne prečítať každý obraz, čo vygeneruje agent *BW*. Stačí mu zamudrovať správny prah raz za čas. Robot sa tak pri sledovaní loptičky môže dostať do iných svetelných podmienok v ktorých loptičku stratí hoci sa na ňu pozerá. Vtedy trvá istý čas, kým agent *Optimal* vymyslí nový prah. Je to podobné ako keď človek príde zo svetla do tmy – chvíľu mu to trvá, kým sa začne orientovať.

Čítanie čiernobieleho obrazu, ktorý produkuje agent *BW* agentmi *Hough* a *Optimal* je poučné z hľadiska javu, ktorý nazývame implicitné vzorkovanie (obr. 7). Keď agent *BW* zapisuje do *space* blok *grayImage*, vždy tým prepíše predchádzajúcu hodnotu (a vyvolá deštrukciu objektu, ktorý ju reprezentuje). Vôbec ho nezaujíma, či ho niektorý z agentov stihol prečítať. Agent *Hough* sa snaží čítať podľa možnosti každý, zato *Optimal* ich nestíha prečítať drvivú väčšinu. Je pritom jedno, či budeme agenta *Optimal* budiť časovačom alebo spúšťačom: pokiaľ sa motá v *sense_select_act()*, všetky budenia sa v rámci rovnakého podnetu zlievajú do toho posledného (nezlejú sa len rôzne budenia, takže napríklad ak prichádza veľa podnetov z rôznych spúšťačov, agent dostane od každého práve jedno budenie). Dátový tok medzi agentom *BW* (producentom dát) a agentmi *Hough* a *Optimal* (konzumentami dát) je teda automaticky vzorkovaný na takú frekvenciu, ktorá zodpovedá výpočtovým možnostiam konzumentov. Toto vzorkovanie je inherentnou vlastnosťou *space* a v kóde agentov sa nijako neprejavuje.



Obr. 7. Implicitné vzorkovanie

Implicitné vzorkovanie nie je vhodné vždy a všade. Bolo by napríklad nepríjemné počítať takýmto systémom peniaze. Avšak pre systém, ktorý má pracovať v reálnom čase a nedostáva sa mu na to dostatok výpočtovej kapacity, je implicitné vzorkovanie požehnaním. Takýto systém sa nikdy nemôže zahltiť a akosi sám od seba robí vždy to najlepšie, čo sa dá pri danej výpočtovej kapacite stihnúť. Pritom vývojár nemusí toto prispôsobovanie sa nijako explicitne vyjadrovať vo svojich kódoch – funguje to úplne automaticky. Je zaujímavé si všimnúť, že je tu obrátená logika chápania, čo je to systém reálneho času. Tradičný názor je, že na to by ste mohli mať systém reálneho času, nevyhnutne potrebujete dostatočnú výpočtovú kapacitu. Pri našom prístupe toto obmedzenie prekonávame a schopnosť systému ako-tak pracovať v reálnom čase je daná jeho modulárnym usporiadaním. Budujeme tu systém, ktorý chce byť systémom

reálneho času napriek tomu, že sa mu na to nedostáva dosť výpočtovej kapacity. Má však mechanizmus, ktorým tomuto nedostatku čelí[13].

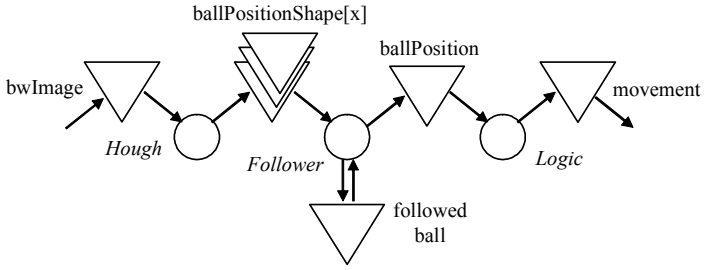
```
a)  void init (string args) {
        trigger_attach("grayImage");
    }
    void sense_select_act (int pid) {
        int ct = space_read("cannyThreshold",200);
        Mat blank(240,320,CV_8UC1), gray = space_read("grayImage",blank);
        HoughCircles(gray,circles,CV_HOUGH_GRADIENT,2,50,200,100);
        for (int i=0; i<circles.size(); i++)
            space_write("ballPositionShape*",circles[0],300);
    }

b)  void init (string args) {
        tid = trigger_attach("ballPosition*",TRIGGER_MATCHING);
    }
    void sense_select_act (int pid) { // pid == tid
        Vec3f last = space_read("followedBall",Vec3f(0,0,0));
        bool nolast = (last[2] == 0);
        SpaceIterator<Vec3f> it;
        for (it=space_read_begin(pid); it!=space_read_end(pid); it++) {
          Vec3f candidate = it->value;
          if (nolast || (fabs(candidate[0]-last[0])<tolerance
                     && fabs(candidate[1]-last[1])<tolerance)) {
             space_write("followedBall",candidate,600);
             space_write("ballPosition",candidate,300);
             break;
          }
       }
    }
```

Kód 9. a) úpravy agenta *Hough*. b) nový agent *Follower*

Keď teraz robot vojde do tmy, na chvíľu ustrnie a potom začne normálne pracovať. Keď však vystúpi z tmy do svetla, nespráva sa až tak pekne, lebo počas doby hľadania nového prahu sa snaží sledovať to veľké množstvo kružníc, ktoré sa mu zrazu objavili pred očami. Podobná situácia vzniká aj bez zmeny svetelných podmienok, pokiaľ do výhľadu robota umiestnime viacero loptičiek. Na vylepšenie tohto správania možno uvažovať ďalšiu modifikáciu systému, ktorá zabezpečuje kontinuitu sledovania jednej loptičky. Nedostatok nášho doterajšieho návrhu spočíva v tom, že agent *Hough* zapíše do *space* vždy len jednu pozíciu: prvú, ktorá sa mu naskytne. Nie je samozrejme problém, aby tam zapísal miesto jednej pozície celý vektor. Ešte lepšie však bude, keď do *space* zapíše jednotlivé pozície ako samostatné bloky. Je to lepšie preto, že tak možno zamiešať jeho názor s názorom ďalších detektorov, ktoré by sme pridali v budúcnosti. Dostávame sa tak k potrebe opäť rozšíriť schopnosti *space*, tentoraz o hromadné manipulácie s blokmi.

[13] Domnievame sa, že v podobnej situácii sa nachádzajú aj živé organizmy a že v ich modulárnej štruktúre by sa dal nájsť podobný mechanizmus [12]



Obr. 8. Úprava pre zameriavanie pozornosti na konkrétnu loptičku

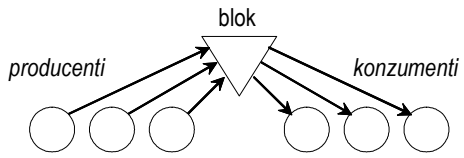
Realizáciu vidíme na kódoch 9a a 9b. Agenta *Hough* (Kód 9a) sme modifikovali tak, aby do *space* zapísal všetky nájdené kružnice. Na to mu slúži hviezdičková konvencia, ktorá spôsobí, že k menu zapisovaného bloku je pripojený index tak, aby jeho meno nebolo v konflikte s aktuálne platnými blokmi s rovnakou maskou. K menu zapisovaných blokov bola taktiež pridaná prípona *Shape* determinujúca pôvodcu bloku, aby v budúcnosti mohli iní agenti zapisovať iných kandidátov, napríklad *ballPositionColor**. Medzi agenta *Hough* a agenta *Logic* vložíme agenta *Follower*, ktorý, bude realizovať zameranie pozornosti robota na jednu loptičku (Kód 9b). Ako vidíme z jeho kódu, čítanie viacerých blokov podľa masky je realizované prostredníctvom iterátora, ktorého agent získa podľa identifikátora jeho spúšťača. Stratégia tohto agenta je pomerne jednoduchá: pokiaľ nesleduje konkrétnu loptičku, tak sa sústredí na prvú, ktorá sa mu naskytne. Potom sa snaží medzi prichádzajúcimi loptičkami nájsť takú, ktorej poloha sa od zvolenej príliš neodlišuje. O tej predpokladá, že ide o jeho loptičku, ktorá sa mu v obraze mierne posunula. Zabúda teda zvolenú pozíciu a sústreďuje sa na novú, jej podobnú. Pokiaľ sa mu dlhší čas nepodarí nájsť podobnú pozíciu, zvolenú pozíciu zabúda a je pripravený akceptovať prvú inú pozíciu, ktorá sa mu naskytne. Zložitejší je spôsob, akým je táto stratégia implementovaná. Nielenže zvolená pozícia sledovanej loptičky je uložená do *space* ako blok *followedBall* – čím sa agent stáva čisto reaktívnym, ale podstatne je tu využitá i časová platnosť tohto bloku.

5 Teoretické pozadie multiagentového prístupu.

Okrem svojich technických predchodcov (LINDA, Java Space, multiagentové systémy) má uvedená architektúra blízko ku Minského societnému modelu mysle [15] a Brooksovej subsumpčnej architektúre [1]. Našim zámerom pri návrhu AgentSpace bolo, aby sme s týmito dvomi prístupmi boli kompatibilní, tj. aby sme im poskytli vhodnú implementačnú platformu. Uvedieme niekoľko podnetov, ktoré z týchto prístupov vychádzajú:

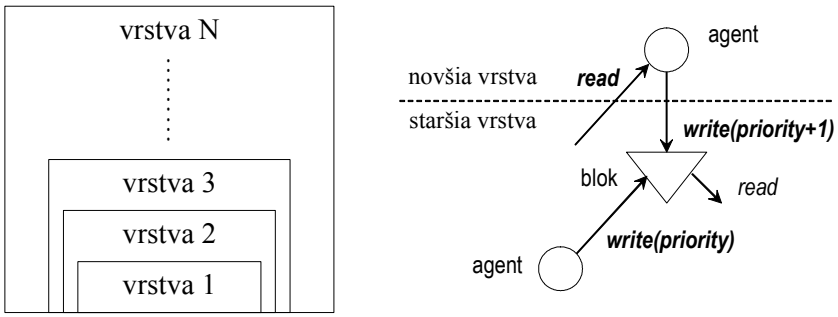
Decentralizácia. Tak, ako požaduje Minsky pre svoj societný model, pri našom prístupe je systém decentralizovaný v tom zmysle, že v ňom neexistuje agent, ktorý by riadil činnosť ostatných agentov. Pokiaľ niektorý z agentov zlyhá, môže to mať síce

neblahý vplyv na globálne správanie systému, ale tento stav nebude generovať žiadnu výnimku, ktorá by zastavila celý systém. Agenti, ktorí nie sú odkázaní na bezprostredné výsledky zlyhaného agenta, sa budú mať ďalej čulo k životu. Pri ich dostatočnom počte a vzájomnom zálohovaní sa dokonca môže stať, že sa zlyhanie jediného agenta v globálnom správaní prejaví len okrajovo, alebo dokonca tvorivo. Táto vlastnosť je daná schopnosťou blokov byť čítaný i zapisovaný viacerými agentmi. Každý blok potenciálne reprezentuje dátový tok many:many, pričom zostava producentov i konzumentov bloku sa môže dynamicky obmieňať (Obr. 9).



Obr. 9. Dátový tok many:many

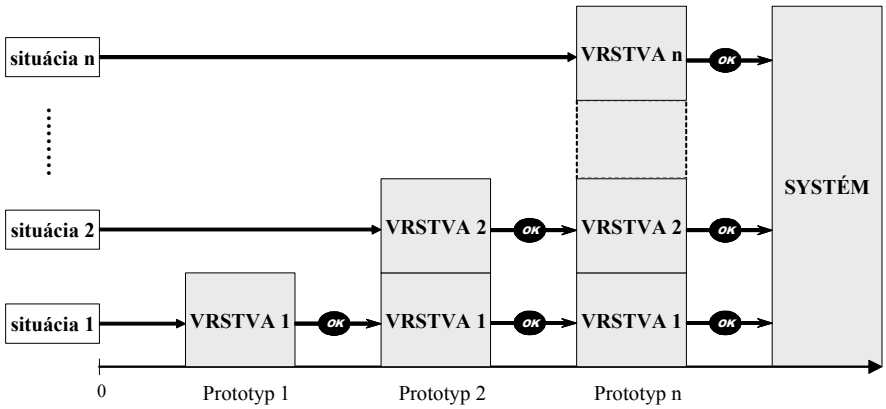
Subsumpcia. Brooks v [1] navrhol podnetný spôsob ako môže vývojár napodobňovať pri vývoji evolúciu bez toho, že by pracoval s nejakou populáciou jedincov. Je založený na fakte, že evolúcia sa odráža vo vnútornom usporiadaní systému, pričom pre zjednodušenie predpokladáme, že v potomkovi nájdeme celú štruktúru jeho predchodcu zaobalenú do nových štrukturálnych vrstiev (Obr. 10 vľavo)



Obr. 10. Subsumpcia a jej realizácia v AgentSpace

Základným predpokladom použitia subsumpcie je vhodná modularita systému, ktorá musí umožniť, aby sa evolučne novšie vrstvy vedeli s koordinovať s evolučne staršími. Pri našom prístupe na to dokážeme využiť zapisovanie bloku viacerými agentmi. Evolučne novší agent môže prepísať hodnotu v evolučne staršom bloku, čím zmení dátový tok medzi dvomi evolučne staršími agentmi. Ten z nich, ktorý blok číta, pritom vôbec nevie rozpoznať, že prečítaná hodnota nepochádza z jeho vrstvy. Takýmto spôsobom sa môže evolučne novšia vrstva votrieť do činnosti evolučne starších vrstiev: prečíta si z nich obsah blokov a prípadne ich poprepisuje tak, aby tieto vrstvy manipulovala k špecifickej činnosti. Pre tento trik je však dobré zaviesť ďalšiu schopnosť nášho *space* a to prioritný zápis, aby novší agent nemusel súperiť so starším

– normálne by totiž záležalo len od okamihu zápisu a čítania, či konzument bloku prečíta hodnotu zapísanú starším alebo novším producentom. Preto zapisovateľ okrem mena, hodnoty a časovej platnosti môže pri zápise definovať aj prioritu zapísanej hodnoty. Zápis hodnoty s menšou prioritou potom *space* ignoruje (Obr. 10 vpravo), pričom zapisujúcemu agentovi sa zdá, že prebehol v poriadku.



Obr. 11. Vývoj situovaného systému pomocou subsumpcie

Situovanosť. Nami vytvorený systém je situovaný v tom zmysle ako požadoval Brooks [2], t.j. do jeho konštrukcie sa premietajú parametre konkrétneho prostredia, pre ktoré systém funguje, ako aj parametre senzorov a motorov systému. Odráža to napríklad množstvo tolerančných konštánt, ako i snaha spojazdniť najprv systém v jednoduchšej situácii a potom túto situáciu stupňovať. Tento prístup je vhodný najmä pre také aplikácie, kde je jasné, čo od systému očakávame, ale nedáva nám to ani najmenšiu nápovedu ako to implementovať. Typickým príkladom takej aplikácie je autonómne vozidlo: vieme povedať kedy jazdí správne a kedy zráža chodcov na prechode, ale to len málo osvetľuje, aká by mala byť jeho riadiaca štruktúra. Situáciu v ktorej tento systém má fungovať, preto na úvod výrazne zjednodušíme a postupne budeme stupňovať, pričom na ďalší kvalitatívny stupeň sa budeme posúvať pomocou subsumpcie (Obr. 11). V jednoduchšej situácii necháme systém konať prostredníctvom evolučne starších vrstiev. V zložitejšej situácii sa však ozvú novšie vrstvy, votrú sa do činnosti starších, vymažú čo je nesprávne, prepíšu čo sa dá opraviť a sami pridajú svoj diel akcií tak, aby odozva celého systému na danú situáciu bola adekvátna.

6 Záver

Multiagentový prístup k programovaniu riadiacich štruktúr robotov a iných systémov umelej inteligencie je zaujímavou inšpiráciou na prenos našich predstáv o fungovaní mysle do technickej praxe. Vychádza z už pomerne starých myšlienok o modulárnom usporiadaní a postupnom vývine mysle [7] [15], ktoré vyjadruje novým spôsobom a to prostriedkami, ktoré boli vyvinuté v oblasti multiagentových systémov [18]

a v koordinačnom programovaní [3]. Na príklade konkrétnej architektúry AgentSpace [13], ktorá je výsledkom nášho dlhoročného snaženia, sme ukázali, ako sa takéto systémy programujú a aké výhody a nevýhody to prináša. Hlavnou výhodou je tu schopnosť postupného rozširovania vyvíjaného systému a podpora jeho práce v reálnom čase. Nevýhodou sú zvýšené nároky na výpočtový výkon. Každopádne, pokiaľ chce niekto vybudovať komplexný systém, ktorý sa čo len vzdialene podobá v istých ohľadoch mysli, musí byť na takéto investovanie pripravený. Našou architektúrou sme implementovali niekoľko akademických i komerčných systémov a jej formu už považujeme za ustálenú.

S ohľadom na zložitosť prezentovaného materiálu sme pri našom výklade použili metodologický prístup, ktorý na konkrétnej aplikácii vysvetľuje jednotlivé mechanizmy tak, aby bolo jasné, prečo sme ich navrhli práve tak, ako sme ich navrhli. Dúfame, že sme nás postup zvolili správne.

Poďakovanie: Táto kapitola vznikla za podpory grantovej agentúry VEGA v rámci grantovej úlohy VEGA 1/0280/08.

Literatúra

[1] Brooks R.: *Intelligence without representation*. Artificial Intelligence 47, 1991, pp. 139-159

[2] Brooks, R.: *Cambrian Intelligence*, The MIT Press, Cambridge, Mass., 1999

[3] Brooks, R.: *Robot – The Future of Flesh and Machines.* Penguin Books, London, 2002

[4] Ciancarini P., Rossi D.: Jada: *Coordination and Communication for Java Agents.* In Mobile Object Systems: Towards the Programmable Internet (Vitek J., Tschudin C. eds.), LNCS Volume 1222, 213-228, Springer-Verlag, Berlin, 1997

[5] Davies, E. R.: *Machine Vision: Theory, Algorithms, Practicalities.*, Elsevier, 2005

[6] Doran J.: *Distributed AI and its Applications.* In *Advanced Topics in Artificial Intelligence* (Mařík V., Štepánková O., Trappl R., eds.), Springer-Verlag, Berlin, 1992

[7] Fodor J.: *The Modularity of Mind*. The MIT Press, Cambridge, Mass., 1983

[8] Gelernter D.: *Generative Communication in LINDA*. ACM on Transactions on Programming Languages and Systems, Volume 7(1), 80-112, 1985

[9] Jennings N.: *On agent-based software engineering*. Artificial Intelligence 117, 2000, pp. 277-296

[10] Kelemen, J.: *Multiagent symbol systems and behavior-based robots*. Applied Artificial Intelligence 7, (1993), 419-432.

[11] Kelemen, J.: *The Agent Paradigm.* Computing and Informatics, Vol. 22. (2003), pp. 513-519

[12] Lúčny, A.: *Reaktívny model inteligentného systému.* In: Kognitívne vedy II (Kvasnička, V. – Pospíchal, J., eds.), CHTF STU Bratislava, 1999, 75-84

[13] Lúčny, A.: *Building Complex Systems with Agent-Space Architecture.* Computing and Informatics, Vol. 23 (2004), pp. 1001-1036

[14] Lúčny, A.: *Od medzimodulových spojení k nepriamej komunikácii medzi agentami.* Znalosti, VŠE Ostrava, 2007.

[15] Minsky, M.: *Society of Mind.* Simon & Schuster, New York, 1986

[16] Sonka, M. – Hlaváč, V. – Boyle, R.: *Image processing, analysis and machine vision*. 3rd edition, Thomson Learning, Toronto, 2007

[17] Waldo J.: *Mobile Code, Coordination and Changing Networks.* Concoord, Lipari, 2001

[18] Wooldridge M.: *Introduction to MultiAgent Systems.* John Wiley & Sons, 2002.

Kognitívne vylepšovanie človeka, evolúcia a etika

Elena MATĚJKOVÁ[1] a Peter SÝKORA[2]

Abstrakt. Kognitívne vylepšovanie (enhancement) človeka prostredníctvom neuročipov, chemických látok alebo génových manipulácií je dnes predmetom dôležitých etických diskusií. Základná otázka znie, či takéto vylepšovanie je morálne prijateľné alebo naopak, má byť spoločnosťou zakázané. Zdá sa, že eticky najkontroverznejšie je genetické vylepšovanie človeka, ktoré by mohlo v konečnom dôsledku viesť k vzniku nového, posthumánneho biologického druhu.

1 Úvod (kognitívne vylepšovanie je vylepšovanie vylepšovania)

Ľudská schopnosť neustále vylepšovať svoje schopnosti a možnosti patrí k najvlastnejšej prirodzenosti druhu Homo sapiens. Človek sa od úsvitu svojej existencie snaží rôznymi spôsobmi prekročiť svoje biologické danosti, ktorými ho vybavil prírodný výber. Túto možnosť, na rozdiel od zvierat, má vďaka jednej svojej výnimočnej prirodzenej danosti – vďaka svojim mimoriadnym kognitívnym schopnostiam. Len vďaka nim človek namiesto chýbajúcej srsti, vymyslel oblečenie, namiesto chýbajúcich trhákov vymyslel nôž, namiesto nôh antilop vymyslel auto, namiesto krídel vtákov lietadlá. Ak si spočiatku svojimi vylepšeniami iba kompenzoval nedostatky svojho biologického druhu oproti iným druhom, postupne tieto kompenzácie začali ďaleko prekračovať limity živej prírody vôbec. Auto je rýchlejšie než ktorákoľvek antilopa, raketa doletí ďalej než ktorýkoľvek vták.

Dnes však vďaka rozvoju moderných vied a technológií dochádza k tomu, že viaceré vylepšenia sa stávajú integrálnou súčasťou samotného ľudského tela. Môžeme tu identifikovať dva druhy procesov. Prvým je kyborgizácia ľudského tela. V nej ide o čo najtesnejšie prepojenie biologického hardvéru (tela) s hardvérom kybernetickým. Príkladom je integrovanie mikročipov do tela. Druhým procesom je to, čo by sme mohli zatiaľ pracovne nazývať „biologické upgradeovanie“ ľudského tela. Súvisí s obrovským rozvojom molekulárnej biológie a na ňu nadväzujúcich moderných

[1] Katedra filozofie FF UCM, Nám. J. Herdu 2, 917 01 Trnava,
E-mail: ella.matejkova@gmail.com
[2] Katedra filozofie a Centrum pre bioetiku FF UCM, Nám. J. Herdu 2, 917 01 Trnava,
E-mail: sykora@infovek.sk

biotechnológií, ktoré nastalo od polovice minulého storočia. Vďaka tomuto rozvoju ľudia získali poznatky a nástroje, ktoré im umožňujú cielene zasahovať do subtílnej molekulárnej stavby živých organizmov a vytvárať tak nielen účelne modifikované organizmy, ale aj organizmy úplne nové, aké by vo voľnej prírode nikdy nemohli vzniknúť (ako sú napríklad baktérie produkujúce ľudské hormóny).

Najefektívnejšie je uskutočňovať biologické modifikácie na úrovni génov a to hneď z dvoch dôvodov. Po prvé, gény sú v živých organizmoch na začiatku kauzálneho reťazca (v danom prostredí) vedúceho k určitej biologickej funkcii, prípadne biologického produktu. Črevná baktéria Escherichia coli začala produkovať ľudský hormón inzulín potom, čo do jej chromozómu bol prenesený ľudský gén pre inzulín. Vďaka univerzalite genetického kódu bola ľudská genetická informácia o syntéze inzulínu molekulárnymi mechanizmami baktérie správne interpretovaná a baktéria začala tento ľudský hormón syntetizovať. Po druhé, raz urobená genetická modifikácia, pokiaľ sa dostane do pohlavných buniek, sa automaticky prenáša do všetkých nasledujúcich generácií potomkov, takže nie je potrebné robiť tú istú modifikáciu nanovo v každej novej generácii.

Pretože nie všetky genetické modifikácie u mnohobunkových organizmov sú aj dedičné. Napríklad, nahradenie poškodeného génu pre krvotvorbu v bunkách kostenej drene normálnym génom prineseným zvonku (tzv. somatická génová terapia – *somatic gene therapy*) vedie k tomu, že sa síce opraví krvotvorba u postihnutého pacienta, ale pokiaľ sa tento gén neopraví aj v jeho pohlavných bunkách (tzv. génová terapia pohlavných buniek, pohlavnej línie – *germ-line gene therapy*), tak potomkovia somaticky génovo vyliečeného pacienta budú naďalej postihnutí týmto genetickým ochorením.

Od objavenia sa prvých GMO (geneticky modifikovaných organizmov) bolo jasné, že je len otázka času, kým sa k nim priradí aj človek. Najprv sa začalo s genetickou modifikáciou jednobunkových organizmov (baktérií, kvasiniek), ale neustále sa zrýchľujúci vývoj biológie prišiel veľmi rýchlo s genetickými modifikáciami zložitých mnohobunkových organizmov vrátane cicavcov (myši, potkany, dobytok, opice).

V súvislosti s týmto pokrokom a čoraz masívnejším prenikaním GMO do praxe sa čoraz viac začala klásť otázka, či takéto konanie je aj rozumné. Predovšetkým v súvislosti s možnými negatívnymi dôsledkami GMO pre životné prostredie a zdravie ľudí. Ale otázka bezpečnosti nie je jediným problémom genetických modifikácií. Oveľa základnejšou je otázka etická. Vo všeobecnej polohe je to otázka, či je vôbec správne geneticky modifikovať organizmy nie kvôli možnému ohrozeniu ľudí či života iných biologických druhov, ale vzhľadom na tieto organizmy samotné. Vzhľadom na tradičný antopocentrizmus nadobúda táto otázka predovšetkým podobu, či je správne zasahovať do ľudskej genetickej informácie a meniť tak samotnú ľudskú (biologickú) podstatu, prirodzenosť.

Keďže otázka správnosti nášho konania spadá do oblasti morálky a etiky, ide o etický problém. Dnes stojíme na križovatke možno budúcich ľudských dejín – otvárajú sa nám biotechnologické možnosti začať proces radikálneho vylepšovania ľudských biologických, a teda aj kognitívnych dispozícií pomocou genetických manipulácií tak, že to ovplyvní všetky nasledujúce generácie ľudí. Očividne je veľa

našich biologických daností, v ktorých by sme si želali byť lepšie disponovaní a čo by nám nepochybne v mnohom uľahčilo náš život. Nepochybne medzi takýchto kandidátov na vylepšenie patria aj naše kognitívne schopnosti, ktoré hlavne vo vyspelých a moderných informačných spoločnostiach opierajúcich sa o znalostnú ekonómiu, hrajú v živote ľudí rozhodujúcu úlohu.

O užitočnosti kognitívneho vylepšovania niet pochýb. Užitočnosť však nie je jediným kritériom, ktoré by sme mali vziať do úvahy. Samozrejme, je tu už spomínaná otázka bezpečnosti. Teda problém, že aj dobre mienené vylepšenia našich kognitívnych schopností môžu v konečnom dôsledku viesť k poškodeniu iných našich mentálnych schopností alebo iných funkcií nášho mozgu, že počiatočné vylepšenia nad normálny stav sa môžu zvrátiť vo svoj pravý opak a viesť dokonca k zhoršeniu oproti pôvodnému stavu atď. atď. Máme viacero neblahých poučení v tomto smere z ekológie. Viaceré dobre mienené zásahy do životného prostredia viedli v konečnom dôsledku k negatívnym javom a k zhoršeniu, nie zlepšeniu. To všetko zatiaľ nedokážeme vylúčiť a už len preto, by sme sa nemali s vylepšovaním ponáhľať (chcelo by to prinajmenšom uvaliť na tieto manipulácie s ľudskou genetickou informáciou dočasné celosvetové embargo).

Ale otázka bezpečnosti sa týka akýchkoľvek technologických noviniek – od atómových elektrární, cez GM plodiny až po genetické inžinierstvo človeka. Pri procese schvaľovania nového lieku či novej liečebnej metódy sa stanovuje miera prijateľného rizika – úplne sa riziko vylúčiť nedá. Ani používanie tak bezpečného lieku akým je penicilín nie je celkom bez rizika (cf. úmrtia v dôsledku veľmi zriedkavej alergie na penicilín), ale toto riziko sa nestalo argumentom zakazujúcim úplne používať toto antibiotikum.

Do akej miery sú dôsledky nášho konania nebezpečné (a pre koho okrem ľudí?) je síce dôležitý aspekt etického uvažovania, ale etickosť na bezpečnosť redukovať nemožno. Napríklad, často sa argumentuje, že klonovanie ľudí je neetické, pretože jeho úspešnosť je mizivá . Známy zvierací klon ovca Dolly je jedinou zdravou ovcou v experimente, na počiatku ktorého stálo 29 prenosom jadra vytvorených umelých embryí z celkového počtu 237 pokusov. Z týchto embryí len tri dokončili vývoj do pôrodu a z týchto troch sa životaschopnou ukázala len jedna jediná ovca – Dolly. Postup u človeka s takouto efektívnosťou by bol jednoznačne odmietnutý, aj keby z čisto etických dôvodov bol prijateľný. Ibaže nie je problém si predstaviť, že skôr alebo neskôr technológia klonovania bude tak efektívna ako ktorýkoľvek akceptovaný zložitý medicínsky postup (napr. transplantácia srdca) a bude v rámci našich noriem považovaná za bezpečnú. Napriek tomu sa etický problém, či je správne klonovať ľudí, týmto nestratí.

Podobne môžeme predpokladať situáciu, že vylepšovanie ľudskej genetickej informácie bude vzhľadom na svoju rizikovosť prijateľnou biomedicínskou metódou a napriek tomu ešte stále budeme musieť odpovedať na otázku, či ho máme považovať za eticky prijateľné. Vzhľadom k čomu máme pomeriavať jeho etickosť ak bezpečnosť už nie je kritériom? Ak pominieme náboženskú námietku, že zasahovaním do ľudskej prirodzenosti prijímame rolu Stvoriteľa, ktorá nám z podstaty veci neprináleží, aké iné sekulárne etické dôvody musíme zvážiť, aby sme posúdili, či kognitívne vylepšovanie človeka pomocou zásahov do ľudského genetického dedičstva je správne alebo nie.

Máme napríklad považovať evolúciou vytvorenú ľudskú genetickú podstatu za hodnotu, ktorá musí zostať nedotknutou, za hodnotu, akou sú napríklad historické kultúrne pamiatky? Znamená to, že napomáhať ľudskej evolúcii je neetické? Alebo naopak, ako sa niektorí domnievajú, neetické je nevylepšovať ľudský druh, pokiaľ k tomu už máme nástroje a zostať pasívnou hračkou v „rukách" prirodzenej a na náhode založenej evolúcii? Neodporoval by práve takýto pasívny postoj k vlastnému osudu našej ľudskej prirodzenosti a doterajšiemu vývoju ľudskej civilizácie? Existujú eticky relevantné rozdiely medzi rôznymi spôsobmi vylepšovania, je povedzme negenetický spôsob vylepšovania našich kognitívnych schopností pomocou psychofarmák či mozgových mikročipov etickejší než pomocou zasahovania do genetickej informácie? Ukážeme si niektoré hlavné argumenty, ktoré sa používajú v prebiehajúcej diskusii o etických aspektoch vylepšovania človeka, s dôrazom na vylepšovanie kognitívnych schopností.

2 Kognitívne vylepšovanie človeka - taxonómia

Kognitívne vylepšovanie človeka môžeme definovať ako rozširovanie, zväčšovanie alebo skvalitňovanie základných kognitívnych schopností ľudskej mysle, ako je získavanie nových informácií a poznatkov (percepcia), selektovanie informácií (pozornosť a koncentrácia), ich vyjadrenie (porozumenie) a ukladanie (pamäť). V širšom kontexte sem môžeme počítať aj použitie týchto informácií na usmerňovanie svojho správania

Vylepšovanie kognitívnych schopností nie je pre človeka úplne novým fenoménom. Vzdelávanie po stáročia predstavuje nielen odovzdávanie špecifických zručností a informácií, ale tiež zdokonaľovanie mentálnych schopností, akými sú koncentrácia, pamäť a kritické myslenie. V súlade s taxonómiou vylepšovania podľa Bostroma a Sandberga ([2]: 3) môžeme v prípade vzdelávania hovoriť o vylepšovaní pomocou tzv. **interného mentálneho softvéru**. Interný „mentálny softvér" zahŕňa napríklad učenie sa jazyka alebo matematiky. Je to používanie nášho mozgu, „trénovanie" našej schopnosti myslieť, učiť sa, zapamätávať si. Môžeme to prirovnať k fyzickému vylepšovaniu tela. Interný mentálny softvér je akýsi tréningový program, ktorým posilňujeme, a tým v konečnom dôsledku vylepšujeme (zvyšujeme kapacitu) nášho „mozgového svalu".

Naproti tomu ako **externý mentálny softvér** môžeme označiť napríklad počítačové programy, webové stránky a internetové prepojenia, ktoré taktiež expandujú našu schopnosť zbierať informácie, ukladať ich a analyzovať. Externý „mentálny softvér" nám tiež slúži ako externá podpora našej pamäti, proces, ktorý začal vynálezom písma. Písané dokumenty, knihy, pero či kníhtlač sú klasické podoby toho, čo môžeme nazvať **externý „mentálny hardvér**". Ten v poslednej dobe nadobúda čoraz rafinovanejšiu podobu reprezentovanú dnešnými počítačmi.

Taxonómiu ľudského kognitívneho vylepšovania uzatvára štvrtá kategória - **interný mentálny hardvér**. Ide o priamy hardvérový zásah do mozgu, ktorý vedie k vylepšeniu jeho funkcií. Podľa povahy hardvéru ich rozdeľujeme na tri kategórie: mechanické, elektronické a kybernetické v jednej kategórii, a chemické a biologické

ako osobité kategórie[3]. Napriek tejto veľkej rozmanitosti sa ukazuje, že viaceré etické problémy majú všetky tri kategórie spoločné. Napríklad, problém spornej hranice medzi ich použitím na terapeutické alebo čisto vylepšovacie ciele, otázka spravodlivého distribuovania možností vylepšovania v spoločnosti či potreba nanovo definovať človeka a ľudskosť v súvislosti s kyborgizáciou a posthumanizmom (podrobnejšie o tom nižšie).

Zdá sa, že tradičné spôsoby kognitívneho vylepšovania nebudia pohoršenie, takže nie je celkom jasné, či odmietanie kognitívneho vylepšovania je odmietaním princípu vylepšovania, alebo nové spôsoby prinášajú oproti tradičným nielen kvantitatívny, ale aj kvalitatívny posun. Tradičné spôsoby kognitívneho vylepšovania doteraz nevyvolávali etické pochybnosti, napr. vylepšovadlá (*enhancers*) v podobe rôznych bylinných extraktov podporujúce koncentráciu a pamäť, ako je kofeín, guarana, či extrakty zo stromu Gingko biloba.

3 Kybernetické kognitívne vylepšovanie (neuroimplantáty)

S rozvojom vedy a technológie došlo postupne k posunu od mechanických protetík, ktoré boli len čisto mechanicky spojené s ľudským telom, k vývoju robotických, kybernetických a elektronických prostriedkov, ktoré už nemusia spĺňať len čisto protetický cieľ, ale použité u zdravého človeka, môžu rozširovať jeho možnosti a schopnosti. Mikročipové implantáty, pomocou ktorých je elektronický prístroj priamo napojený na centrálny nervový systém reprezentujú proces kyborgizácie človeka asi najvýraznejšie. Ide o vytváranie hybridného spojenia medzi dvoma svetmi, ktoré sú tradične považované za svoje protiklady – medzi neživým mechanickým, resp. elektronickým, robotickým svetom na jednej strane a živým, biologickým, organickým svetom na strane druhej.

Musíme si uvedomiť, že kyborgizácia sa zďaleka netýka len ľudského tela v tradícii medicínskej protetiky, ale aj iných biologických druhov (napr. projekty využitia kyborgických zvierat pre špionážne účely). Naším cieľom tu však nie je rozšíriť etický diskurz týkajúci sa vylepšovania človeka o etickú problematiku ochrany (práv) zvierat, ako skôr upozorniť na skutočnosť, že problematika kyborgizácie so sebou prináša výzvu nanovo premyslieť tradičnú dichotómiu medzi živým a neživým, medzi biologickým, organickým na jednej strane a mechanickým, robotickým, elektronickým, kybernetickým na strane opačnej.

Ak pominieme konzervatívny postoj, ktorý odmieta akékoľvek podstatnejšie zásahy do ľudskej prirodzenosti, aké iné etické problémy prináša kyborgizácia človeka? Predovšetkým môžeme zatiaľ popustiť uzdu fantázii na spôsob tvorcov vedecko-fantastickej literatúry a filmov a začať rozmýšľať o mozgových čipoch, ktoré

[3] Toto rozdelenie je samozrejme veľmi zjednodušujúce, a to ani nechceme otvárať tému prebiehajúcej konvergencie technológií, ktoré sa v tejto kapitole snažíme rozdeliť do osobitých kategórií. Táto konvergencia je známa ako *NBIC zjednotenie* (*nano-bio-info-cogno unification*), čiže zjednotenie nanotechnológie, biotechnológie, informačnej technológie a kognitívnych vied (pozri http://convergentsystems.pbworks.com/w/page/16444399/NBIC-Converging-Technologies)

by nielen rozširovali či urýchľovali operačnú pamäť ľudskej mysle, ale rozširovali by aj jej dlhodobú pamäť do takej miery, že by sme si fakticky pamätali všetko, s čím sme sa stretli čo i len raz v našej skúsenosti. Učenie by sa možno zmenilo v bezdrôtové downloadovanie príslušných informácií z databáz v tom momente, kedy by sme ho potrebovali (niečo na spôsob známej sekvencie z filmu Matrix o schopnostiach riadiť helikoptéru). Na druhej strane, hrozilo by, že by sme si pamätali obrovské množstvo nepodstatnej, nezaujímavej, triviálnej informácie, ktorá by nám v mysli len zavadzala. No to by bol zrejme len dočasný menší technický problém, ktorý by inteligentný softvér v pamäťovom neuročipe hravo zvládol a triedil by pre nás podstatné od nepodstatných informácií. Ibaže s každým krokom, ktorým by sme softvéru v mikročipoch v našom mozgu odovzdávali expertné právomoci, by sme sa súčasne zbavovali svojej autonómnosti. Asi tak, ako sa už dnes odovzdávame v lietadlách do rúk automatických pilotov (dokonca bez toho, aby sme k tomu ako pasažieri dali informovaný súhlas), či rozhodovanie za investície zverujeme bankovým expertným softvérom a pod.

Eticky relevantná otázka v prípade budúcich ľudských kyborgov potom znie, kde sú hranice našej autonómnosti, slobodnej vôle, miera delegovania zodpovednosti, za ktorú už ľudská bytosť ísť nemôže? Alebo ináč – od ktorého momentu prestáva byť CEO v našom kyborgickom tele náš organický mozog, naša prirodzená ľudská „duša“, a stáva sa ním náš mozgový mikročip a jeho softvér? Alebo ešte ináč – je táto otázka vôbec dobre položená? Nestratí sa v ľudskom kyborgovi hranica medzi organickou a kybernetickou zložkou natoľko, že sa úplne stratí duálnosť organizmus-stroj aj na úrovni mysle a vedomia a bude existovať len jedno hybridné kyborgické telo s jednou hybridnou kyborgickou mysľou a vedomím? V tejto situácii si rozhodne nevystačíme s dnešnými klasickými definíciami človeka opierajúcimi sa ľudskú biologickú prirodzenosť.

Avšak kým v týchto úvahách zatiaľ značne predbiehame udalosti (o ktorých si donedávna mnohí mysleli, že nenastanú, a patria do sféry čírej fantázie, dnes stojí otázka, kedy táto situácia nastane, či v blízkej alebo až v pomerne vzdialenej budúcnosti), už dnes sa vynárajú v súvislosti s kyborgizáciou človeka síce menej fundamentálne, ale o to viac konkrétne etické problémy. Prekvapujúco, kým doteraz sme špekulovali skôr o ochrane človeka pred kyborgami, zdá sa, že dnešným etickým problémom je skôr opak – ochrana kyborgov pred ľudskou nenávisťou a zlobou. V malom rozsahu, ale možno symbolicky a s náznakom toho, čo bude v blízkej budúcnosti oveľa častejším fenoménom, môžeme pozorovať dehonestujúce postoje bežných ľudí voči ľuďom s kochleárnymi implantátmi, ktorí dnes predstavujú asi prvých veľmi jednoduchých „kyborgov“ v bežnom živote.

4 Chemické kognitívne vylepšovanie (chytrolieky, smart drugs)

Moderná biomedicína je dnes schopná poznávať funkčnosť resp. disfunkčnosť ľudského tela na molekulárnej úrovni[4] a na tejto úrovni aj nachádzať vhodné lieky. Tak ako postupne medicína odhaľuje príčiny rôznych metabolických porúch telesných

[4] A samozrejme aj na iných úrovniach, ale zámerne ich v tejto kapitole nespomíname.

orgánov až na molekulárnej úrovni (napr. vzťah medzi cukrovkou, molekulami inzulínu a molekulami receptorov pre inzulín na bunkových membránach), výnimkou nie je ani mozog a jeho fungovanie. Už nejde len o metódu omylov a pokusov, ktorá sleduje správanie mozgu čoby čiernej skrinky porovnávaním vstupov (používaním chemických látok) a výstupov (sledovaním správania a kognitívnych schopností), ale schopnosť modernej neurobiológie (spolu s ostatnými neurovedami) čoraz viac nahliadať priamo do vnútra tejto čiernej skrinky. Odhaľovaním molekulárnych príčin neurologických onemocnení sa moderná farmakoterapia snaží nájsť liečivá takpovediac šité na mieru, ktoré by interagovali presne špecifikovaným spôsobom s molekulárnymi procesmi prebiehajúcimi v mozgu. Samozrejme, to je ideálny a dlhodobý cieľ, ku ktorému smeruje vývoj nootropných liečiv.

Napríklad liečivá založené na amfetamínovej báze, ako napríklad *Ritalín* alebo *Adderral*, predpisujú lekári na liečbu detí s poruchou pozornosti sprevádzanou hyperaktivitou (ADHD). Avšak zistilo sa, že ak sa tieto lieky použijú u zdravých dospelých jedincov, spôsobujú zvýšenie koncentrácie na intelektuálnu činnosť. Podobne je to v prípade iného nootropného lieku *Modafinilu* (*Provigilu)*, ktorý sa za normálnych okolností predpisuje ako liek pre pacientov s narkolepsiou. Narkolepsia je porucha spánku, pri ktorej jedinec náhle upadá do spánku kedykoľvek počas dňa. Ide o neovládateľné záchvaty spánku, ktoré pacient nedokáže kontrolovať. Rozsiahle výskumy zameriavajúce sa na účinky modafinilu preukázali, že táto látka má zaujímavé účinky na zdravých jedincov, u ktorých redukuje pokles výkonu pri nedostatku spánku, znižuje reakčný čas, výrazne zlepšuje krátkodobú pamäť. [6],[27],[28].

Podľa niektorých by mohli byť tieto lieky využívané ľuďmi v profesiách, ktoré si vyžadujú mimoriadnu pozornosť a koncentráciu, ako sú napríklad lekári vykonávajúci náročné a dlhotrvajúce operácie, letoví dispečeri, záchranári alebo vojaci v náročných bojových akciách a pod.

Avšak už dnes si tisíce predovšetkým univerzitných študentov, vylepšuje svoje mentálne funkcie pomocou týchto chemických neurostimulátorov. Ako ukázal jeden nedávny prieskum najprestížnejšieho vedeckého časopisu na svete *Nature,* až 20% respondentov zo 60 krajín sveta sa priznalo, že používa lieky ako *Ritalín, Adderal* či *Provigil*, aby si týmto zlepšili pamäť a schopnosť sústrediť sa. Iný prieskum ukázal, že 30% amerických univerzitných študentov niekedy vyskúšalo lieky poskytujúce kognitívne vylepšenie ([24]).

Všimnime si, čo sa vlastne v takýchto prípadoch deje. Tá istá chemická látka, ktorá na jednej strane lieči pacientov tým, že obnovuje u nich normálne hodnoty fungovania mentálnych procesov, ak je použitá u zdravých jedincov posúva fungovanie mentálnych procesov k nadnormálnym hodnotám, čím sa vlastne stáva vylepšovadlom (enhancerom).

Na príklade nootropných liekov je dobre vidieť skutočnosť, že v mnohých prípadoch vylepšovania je definícia toho, čo je a čo nie je vylepšením a vylepšovadlom závislá na kontexte a nie na samotnom charaktere danej látky či prostriedku. Tým sa dostávame k dôležitej téme etického diskurzu o kritériách, na základe ktorých sa má posudzovať prijateľnosť či neprijateľnosť vylepšovacích zásahov do človeka.

5 Rozlišovanie medzi terapiou a vylepšovaním

Jedným z kľúčových etických kritérií, ktoré bolo navrhnuté v diskusii o tom, ako regulovať použitie rôznych mechanických, chemických či biologických prostriedkov, ktoré by sa dali použiť na vylepšovanie človeka, spočíva na rozlišovaní medzi terapeutickým a vylepšujúcim účinkom. Na základe tohto rozdielu sa stanovilo, že kým eticky prijateľné je použitie prostriedkov na terapeutické ciele, neetické je použiť tie isté prostriedky na vylepšovacie ciele: zatiaľ čo terapia je niečo prospešné, a teda prípustné, vylepšovanie predstavuje niečo pochybné, neprirodzené a eticky neprípustné [19].

Liečbu je možné definovať ako obnovenie druhovo-typického fungovania (*species-typical functioning*), zatiaľ čo vylepšovanie ako odklon od neho, či dokonca prerušenie tohto druhovo-typického fungovania. Niektorí vedci navrhujú, aby bola pri rozlišovaní medzi terapiou a vylepšovaním používaná *kvázi-štatistická* koncepcia „normálnosti“, podľa ktorej by sa akýkoľvek zásah do ľudského organizmu, ktorého cieľom by bolo chrániť alebo obnoviť druhovo-typický stav, mal chápať ako terapia [5]. Na základe tohto rozlíšenia potom vylepšovanie predstavuje zásah, ktorého cieľom je zdokonaliť funkcie organizmu nad rámec „normálneho“ stavu.

Avšak aj so štatisticky ponímanou normalitou je problém. Normalita sa tu chápe v podstate kvantitatívne ako štatisticky priemerná hodnota nejakej vlastnosti, funkcie, ktorá sa u nášho biologického druhu normálne vyskytuje. Ale ako je to s takpovediac kvalitatívnou normalitou, teda s púhou prítomnosťou či neprítomnosťou vlastnosti, schopnosti, funkcie oproti našej prirodzenej druhovej výbave? N. Daniels a ďalší, ktorí chápu normálny stav indivídua v rámci svojho biologického druhu by považovali získanie novej vlastnosti za kvalitatívnu zmenu. Tak napríklad, pre človeka je druhovo-typickým stavom alebo fungovaním to, že pomocou svojho zraku nevidí infračervené svetlo. Ak by bol zrak človeka upravený tak, že by človek získal schopnosť vidieť aj infračervené svetlo (napríklad vďaka mozgovému implantátu alebo biologickým pozmenením sietnice), išlo by samozrejme o kvalitatívne vylepšenie, kým navrátenie zraku pomocou zrakového implantátu je obnovením normálneho stavu, no ak by vďaka tomuto implantátu človek vedel zaostriť na väčšie vzdialenosti, než je tomu v priemere, išlo by o kvantitatívne vylepšenie.

Táto argumentácia má za svoju východiskovú premisu predpoklad, že normálny ľudský stav definovaný štatisticky v rámci biologického druhu *Homo sapiens* je etickou normou, z pozície ktorej môžeme posudzovať, či niečo je alebo nie je etické, podľa toho, či smeruje k nastoleniu normálneho stavu alebo naopak, k jeho vychýleniu. Asi nebudeme považovať za etické, ak by niekto vedome druhému spôsobil zhoršenie funkcií jeho tela oproti normálnemu stavu (ochorenie) a prijmeme zákony, ktoré budú ľudí chrániť pred tým, aby niekto zhoršil ich telesné fungovanie (ochrana zdravia). Platí však rovnaký argument aj vtedy, ak pôjde o vychýlenie oproti normálu opačným smerom, ak nepôjde o zhoršenie, ale zlepšenie funkcií? Samotná definícia zdravia nám neumožňuje v takomto prípade hovoriť o „zlepšení zdravia“, pretože zdravie je definované ako normálny a nie „nadnormálny“ stav. Ak by sme etickosť v tomto prípade viazali len na zdravie, potom samozrejme všetky vedomé činy vedúce k odchýleniu od zdravého stavu by sme mali považovať za neetické. Je dostatočným

rozlišujúcim kritériom „normálny stav“, ktorý určuje, čo je liečba poškodenia a čo je vylepšenie nad rámec tohto „normálneho stavu“?

Nemusíme samozrejme súhlasiť s princípom, ktorý má pochopiteľne svoj veľký význam v medicínskej etike a spájať otázku etickosti s pojmom zdravia a môžeme sa pokúsiť hľadať iné kritérium, ktoré by nám pomohlo nejakým spôsobom eticky analyzovať tento fenomén. Ibaže etické kritérium, zakladajúce sa na odlíšení terapie od vylepšenia, má svoje vlastné konceptuálne problémy, pre ktoré je ho ťažké akceptovať minimálne vo všeobecnej rovine. Problém spočíva v pojme normality, o ktorý sa toto kritérium opiera a to minimálne v dvoch aspektoch. Po prvé, problémom je samotné definovanie normality u človeka, po druhé, normalita je socio-kultúrne relatívna.

Príkladom, ktorý túto problematickosť pekne demonštruje je snaha rozlíšiť medzi terapiou a vylepšovaním u osoby, ktorej pamäť je značne oslabená v dôsledku Alzheimerovej choroby. Predstavme si, že po aplikácii liečebnej terapie môže byť pamäť tejto osoby na vyššej úrovni ako pamäť zdravej osoby. Môžeme v tomto prípade hovoriť o terapii, ktorej cieľom je odstrániť prejavy choroby a obnoviť normálny stav pamäti, alebo o vylepšovaní, keďže funkcia pamäti u jedného jedinca prekročila normálny stav pamäti u zdravého jedinca?

Vo všeobecnosti mnohé procesy ľudského organizmu spojené napríklad s jeho starnutím sú „normálnym“ prejavom, a teda súčasťou jeho druhovo-typického fungovania. Postupné odumieranie mozgových buniek alebo atrofia srdcového svalu patria medzi procesy, ktoré sú spojené s narastajúcim vekom organizmu. Pokiaľ chápeme postupné opotrebovanie sa organizmu a s ním spojené prejavy ako prirodzenú súčasť jeho vývinu, musíme prijať, že je prejavom jeho „normálneho“ stavu. Ak sme prijali tvrdenie, že vylepšovanie je zásah, ktorého cieľom je prekročiť rámec normálneho stavu, dá sa oprávnene namietať, že mnohé postupy, ktoré sa aplikujú na zmiernenie alebo odstránenie prejavov spojených so starnutím a sú označované ako terapeutické, sú v skutočnosti vylepšujúcimi postupmi a metódami. Buď budeme normálne starnutie považovať za chorobu, ktorú máme povinnosť liečiť, a potom intervenčné alebo neintervenčné prostriedky brániace starnutiu musíme považovať za lieky a nie vylepšovadlá, alebo ako hovorí J. Harris, potom musíme aj bežné a eticky nekontroverzné prostriedky podporujúce mozgovú aktivitu, srdcovú činnosť alebo pohybový aparát vo vysokom veku chápať ako vylepšovanie a pokiaľ je vylepšovanie eticky neprípustné, tak ich zakázať ([13]:46). Na druhej strane by to však znamenalo, že predĺženie priemernej dĺžky života povedzme o ďalších 200 rokov by bolo z etického hľadiska zdravotne nevyhnutnou liečbou, ktorá by sa mala hradiť zo zdravotného poistenia aj u jedincov, ktorí na zdravotné poistenie neprispievajú?

Predĺženie dĺžky života je spomínaný kvantitatívny druhovo-normálny aspekt vylepšovania, ale v hre sú už dnes aj kvalitatívne aspekty. Tak napríklad podľa J. Harrisa je pravdepodobné, že by sme nenamietali proti získaniu biologickej imunity voči vírusu HIV vďaka biotechnologickým manipuláciám s naším imunitným systémom, hoci by nešlo o obnovenie „normálneho“ stavu organizmu, ale skôr prekročenie rámca „normálneho“ stavu nášho druhovo-typického fungovania([13]: 49).

Druhou dôležitou námietkou voči koncepcii normality ako etickému kritériu je skutočnosť, že biologická variabilita jedincov v dôsledku polymorfizmu génov je v rámci každého druhu niečím úplne normálnym. To znamená, že aj minoritne

zastúpené varianty v populácii, ktoré takto reprezentujú odchýlku od štatistického priemeru, nemusia v dôsledku svojej nízkej početnosti byť nenormálne. Najlepší príklad sú rôzne krvné skupiny. Normálna krvná skupina v určitej populácii (pre rôzne populácie to platí pre rôzne krvné skupiny) nie je len tá, ktorá je najviac zastúpená, napr. skupina B, ale aj všetky ostatné – A, AB či 0.

Ak tento biologický pohľad aplikujeme napríklad na druhy sexuálnej orientácie u človeka, tak dominantná heterosexuálna orientácia nie je o nič normálnejšia a tým pádom etickejšia, než minoritné homosexuálne orientácie. Ale práve v dôsledku spomínaného prepojenia medzi normalitou, štatistickým priemerom a etickosťou, sa ešte v nedávnej minulosti homosexualita považovala za chorobu, ktorú je potrebné liečiť (dokonca spoločnosť mala právo nanútiť takúto liečbu – cf. prípad A. Turinga).

Tieto a ďalšie dôvody viedli nakoniec k stanovisku viacerých bioetikov, že hraničná línia medzi terapiou a vylepšovaním je nejasná, že dokonca medzi týmito dvoma konceptami nie je možné principiálne rozlíšiť, a preto toto kritérium je pre etické posudzovanie irelevantné [3], [13]. Podľa nich by toto kritérium malo byť nahradené iným kritériom, podľa ktorého by malo platiť, že pokiaľ nie sú žiadne obmedzenia týkajúce zdravia a nášho práva na liečbu, nemali by byť ani žiadne obmedzenia čo sa týka nášho vylepšovania [18].

6 Vylepšovanie kognitívnych schopností ako forma podvádzania

Častým argumentom proti vylepšovaniu kognitívnych schopností je, že ide o formu akéhosi mentálneho dopingu, o spôsob podvádzania. Považujeme za samozrejmé, že výkon, ktorý športovci dosiahnu vďaka dopingu môže byť síce excelentný, avšak nie je uznaný, pretože je vnímaný ako porušenie pravidiel, ako podvod. Podobný doping podporujúci mentálny výkon si možno predstaviť v súťažení o umiestnenie sa v rebríčku prijatých uchádzačov na prestížnu univerzitu alebo do intelektuálne náročného a dobre plateného zamestnania. Mnohí v poslednom čase upozorňujú napríklad na neférové výhody vyplývajúce z nespravodlivého ohodnocovania výkonov a študijných výsledkov, ktoré študenti dosiahli pomocou neurostimulátorov [11].

Iní naopak argumentujú, v prospech používania neurostimulátorov, prípadne iných foriem zdokonaľovania kognitívnych schopností, pričom tvrdia, síce tak trochu cynicky, že ak je podvádzanie chápané ako aktivita, ktorá zúčastnenému umožňuje získať výhodu porušením istých pravidiel, potom nemožno použitie neurostimulátorov chápať ako podvádzanie, keďže ho žiadne súčasné pravidlá nezakazujú ([3]: 6).

Zástancovia nových postupov a metód vylepšovania človeka sa pýtajú, prečo by mali byť zakázané psychofarmakologické látky podporujúce výkon kognitívnych schopností, keď nie sú zakázané „povzbudzovadlá“, ako napríklad káva, čaj, rôzne energy-drinky, či bylinné výťažky (z Gingko biloba, guarana). Tieto tradičné prostriedky vylepšujúce kognitívne schopnosti nie sú zakázané, neoznačujeme ich ako dopingové látky, a preto ich používanie, napr. pred skúškou na vysokej škole, sa nepovažuje za podvádzanie ([3]: 6). To nás privádza k otázke, v čom by mal spočívať eticky relevantný rozdiel medzi využitím tradičných a nových prostriedkov zdokonaľujúcich schopnosť učiť sa a zapamätávať si získané poznatky, keď v princípe

ich účinky sú rovnaké no jedny sú považované za bezproblémové a druhé za kontroverzné?

Dalo by sa argumentovať, že tak, ako je v športe rozlíšiteľný vytrvalý tréning a podpora ctižiadostivosti od použitia dopingových prostriedkov, v oblastiach, kde je preverovaný výkon kognitívnych schopností (napríklad pri skúškach na univerzite) je potrebné rozlišovať medzi šálkou kávy, extra hodinami štúdia so súkromným učiteľom a medzi použitím liekov zvyšujúcich koncentráciu a zlepšujúcich pamäť. Je pravda, že ani mnohé „prirodzené" prostriedky pôsobiace na zdokonalenie kognitívnych schopností nie sú dnes prístupné pre všetkých, avšak nerovnocenný prístup k súčasným „prirodzeným" prostriedkom vylepšovania je oveľa ľahšie redukovateľný ako nerovnocenný prístup k novým prostriedkom vylepšovania. Hodnotenie kognitívneho vylepšovania ako podvádzania závisí podľa niektorých od toho, či chápeme nadpriemerný výkon kognitívnych schopností ako „pozičnú výhodu"5 indivídua v spoločnosti, teda ako istú garanciu výhodnejšieho postavenia v spoločnosti, a tiež od toho, čo je pre nás hodnotné na samotnom vzdelávaní [4].

Pokiaľ jednotlivec získal túto „pozičnú výhodu" prostredníctvom kognitívneho vylepšenia, napríklad použitím neurostimulujúcich liekov, môžeme hovoriť o diskriminácii voči ostatným členom spoločnosti. Ak je používanie farmakologických látok alebo iných povzbudzovadiel zakázané, potom použitie neurostimulujúcich látok v situácii sociálneho súťaženia je podvádzanie.

Čo sa týka hodnotenia samotného vzdelávania, niektorí autori podotýkajú, že to, či pre nás bude zdokonaľovanie kognitívnych schopností predstavovať podvádzanie, závisí od toho, čo je pre nás hodnotné na vzdelávaní [4]. Ak je vzdelávanie chápané ako súťaženie v známkach, potom môže vylepšovanie predstavovať podvádzanie. Ak však hodnotu vzdelávania tvorí získavanie zručností, vedomostí a poznania, ktoré prispeje nielen k zlepšeniu životov jednotlivcov, ale aj celej spoločnosti, potom nemožno chápať vylepšovanie kognitívnych schopností ako podvádzanie. Práve naopak, mohlo by predstavovať významnú úlohu vo vzdelávaní.

Nemôžeme si odpustiť poznámku, že takéto zdôvodnenie podvádzania je viac ako cynické. Prekvapujúco, pripomína argumentáciu, ktorá sa na Slovensku a v Čechách používa na ospravedlnenie vytunelovania fondov kupónovej privatizácie. Argument znie asi takto: je síce pravda, že na kupónovej privatizácii niektorí jedinci (tunelári) zbohatli tým, že drobných majiteľov podviedli, doslova okradli, ale ich konanie môžeme považovať za etické, pretože vďaka tomu sa uskutočnila transformácia socialistickej ekonomiky na kapitalistickú, čím sa prispelo nielen k zlepšeniu životov jednotlivcov, ale aj celej spoločnosti, takže je to všetko nakoniec eticky v poriadku.

---

[5] „Pozičná výhoda" (alebo „pozičné dobro") je termín, ktorý je prebraný z ekonómie. V kontexte genetického vylepšovania človeka je to výhoda alebo dobro, ktoré môže jednotlivcovi, ktorý ju má, pomôcť k získaniu výhodného spoločenského postavenia, pozície v spoločnosti. Môžeme povedať, že je to výhoda v spoločenskom postavení plynúca zo zdokonalených fyzických a mentálnych kapacít. Hodnota pozičných výhod je relatívna, lebo závisí od tých, ktorí pozičnú výhodu nemajú. Pozičnou výhodou je napríklad vysoká inteligencia [4].

7 Genetické vylepšovanie kognitívnych schopností

Nielen oblasť kybernetiky a farmakológie, ale aj oblasť genetiky prináša nové možnosti zdokonalenia kognitívnych schopností. Už dnes je možné vytvoriť kópie ľudských génov, ktoré sú zmanipulované tak, aby dochádzalo k zvýšenej tvorbe určitého proteínu dôležitého v biochémii mozgu. Takéto gény je možné implantovať do ľudského tela a nahradiť nimi poškodené gény. Táto metóda sa označuje ako „génová terapia“.

Pri uvažovaní o genetickom vylepšovaní kognitívnych schopností je kľúčovým problémom rozlišovanie medzi génovou manipuláciou somatických (telových) buniek a génovou manipuláciou pohlavných (zárodočných) buniek.

Zatiaľ čo zmena uskutočnená v somatickej bunke je limitovaná na dané indivíduum a neprenáša sa na jeho potomkov, zmena uskutočnená v pohlavných bunkách jedinca sa prenáša na ďalšie generácie. Práve preto, že uskutočnená zmena by bola dedičná, je génová terapia zárodočných buniek eticky kontroverzná. Je pravda, že pomocou génovej terapie by bolo možné odstrániť danú chorobu nielen u konkrétneho jedinca, ale aj u jeho potomkov, ktorí by k nej inak boli predisponovaní.

Manipulácia génov pohlavných buniek vedie k možnosti získať kontrolu nad kvalitou budúcej generácie. V predstave budúcnosti, v ktorej by sa „navrhovali deti na mieru“ vidia mnohí etici problém inštrumentalizácie ľudského života a ohrozenie identity a autonómie jedinca [12], [20].

A nielen to. Od používania génovej terapie pohlavných buniek je už len krôčik k tomu, aby sa touto technikou vnášali nielen gény obnovujúce normálny druhovo-špecifický stav genetického profilu jedinca a jeho potomkov, ale aj gény viac alebo menej meniace tento stav, či už kvantitatívne alebo kvalitatívne. Výsledkom je spustenie procesu umelej ľudskej evolúcie, ktorá môže vyústiť až do vzniku nového, posthumánneho ľudského druhu (viď nižšie).

Povzbudzujúce výsledky pokusov s genetickým vylepšovaním kognitívnych schopností laboratórnych zvierat dávajú tušiť, že sa blíži doba, keď tieto metódy budú realizovateľné aj na človekovi. Napr. zdokonalenie pamäti prostredníctvom genetického zásahu preukázali experimenty na myšiach, ktorým sa dostalo značnej pozornosti aj v médiách. Joe Tsien spolu so svojimi spolupracovníkmi modifikovali gény NR2B u myší tak, aby produkovali väčšie množstvo špecifickej bielkoviny, čo malo za následok zdokonalenie výkonu pamäti [26]. Gény NR2B sú zodpovedné za tvorbu NMDA receptorov na nervových bunkách, v dôsledku čoho ovplyvňujú tvorbu mentálnych spojení, asociácií. Podobným spôsobom to funguje aj u človeka. Keď sa napríklad dotkneme plameňa ohňa, vizuálny signál a signál bolesti sú vyslané do mozgu v rovnakom čase, aktivujú NMDA receptory a vytvoria spojenie a spomienku popálenia sa.

Ukázalo sa, že geneticky zmanipulované „superchytré myši“ boli napríklad schopné lepšie a pohotovejšie sa zorientovať v priestore. Ich reakčný čas pri riešení opakovaného problému bol nižší oproti kontrolnej vzorke. Testovanie ich pamäti preukázalo, že na zažitú situáciu reagovali inak ako myši, ktorých gény neboli modifikované, lepšie si ju pamätali. U modifikovaných myší sa prejavila lepšia schopnosť učenia. V teste, ktorý mal preukázať zlepšenú pamäť a schopnosť

rozpoznávať objekty boli myši vložené do priestoru, v ktorom niekoľko minút skúmali dva rôzne objekty. O niekoľko dní neskôr boli myši znovu vložené do rovnakého priestoru, jeden z predmetov bol však vymenený za nový. U modifikovaných myší bolo spozorované, že oveľa rýchlejšie zistili, že sa jedná o nový objekt a venovali jeho skúmaniu väčšiu pozornosť a dlhší čas ako starému objektu. Nemodifikované myši venovali rovnaký čas skúmaniu oboch objektov, starý si jednoducho „nepamätali". Opakovaním testu v rôznych časových intervaloch vedci zistili, že myši, ktorých gény boli modifikované si objekty pamätali štyri až päťkrát dlhšie ako nemodifikované. Rovnaké výsledky priniesol test zameraný na orientáciu v priestore. „Superchytré myši" sa naučili nájsť v nádobe s vodou záchytnú plošinku po troch opakovaniach, geneticky nemodifikované myši potrebovali až šesť opakovaní [26].

Pre úplnosť musíme dodať, že touto kognitívnou genetickou manipuláciou sa prejavil aj nechcený negatívny dôsledok, a síce zvýšená citlivosť myší na určité formy bolesti. Plynie z toho všeobecné poučenie pre génové manipulácie vôbec a kognitívne génové manipulácie obzvlášť - okrem chcených výsledkov sa môžu objaviť aj neočakávané a vopred nepredikovateľné negatívne bočné efekty, ktoré môžu eventuálne prevážiť nad pozitívnymi efektmi získanými manipuláciou.

8 Transhumanizmus, posthumanizmus …

Schopnosť ľudí manipulovať priamo s ľudskou genetickou prirodzenosťou a predizajnovať ju podľa svojich predstáv, potrieb a očakávaní, otvára úplne novú dimenziu bioetickej diskusie. Aj keď zďaleka nemusí ísť o manipuláciu len s tými génmi, ktoré ovplyvňujú ľudské kognitívne schopnosti (ale napríklad zdravie, dlhovekosť a pod.), zdá sa, že práve manipulácia s týmito génmi je eticky najproblematickejšia. Dobre ju ilustruje otázka, ktorú si viacerí filozofovia dnes začínajú klásť: čo sa stane s ľudskou spoločnosťou, ak sa v nej začne konštituovať nová rasa mimoriadne chytrých ľudí? Ak už nepôjde len o niekoľko málo géniov roztrúsených po svete, tak ako tomu bolo doteraz v ľudskej histórii, ale títo géniovia začnú vznikať na objednávku štátu alebo rodičov a budú ich postupne tisíce, státisíce až milióny, budú žiť pohromade a budú v spoločnosti tvoriť novú ľudskú minoritu so svojou vlastnou subkultúrou, ktorej majoritná spoločnosť kognitívne normálnych ľudí možno ani nebude rozumieť?

Až donedávna v tejto veci platil konsenzus ako medzi vedcami tak aj etikmi, ktorý vyplýval skôr z obavy pred možnými biologickými rizikami, než z nejakých etických úvah, že manipulovanie s genetickou informáciou ľudských pohlavných buniek musí zostať tabuizované. Faktom však je, že technologicky sa manipulácia so somatickými a pohlavnými bunkami neodlišuje a tak všetky výdobytky na poli legálnej a eticky akceptovanej somatickej génovej terapie (tá sa v praxi, aj keď stále ešte experimentálne už roky uskutočňuje) sa dajú využiť aj pre potreby génových manipulácií s pohlavnou líniou. So zvyšovaním efektívnosti somatickej génovej terapie a rapídne rastúceho poznania ľudského genómu argument rizikovosti stráca postupne svoju silu.

Preto nie je veľkým prekvapením, že sa začínajú v poslednej dobe objavovať vedci aj bioetici volajúci po odtabuizovaní genetických zásahov do ľudskej pohlavnej línie. Poukazujú na to, že negatívny postoj voči genetickým zásahom do ľudského

genofondu je skôr výsledkom prirodzených emocionálnych obáv, než dôsledkom racionálneho etického zdôvodnenia ([17]). Podľa nich etická otázka znie: Ak znížime riziko génových manipulácií vyplývajúce z našej nevedomosti a technologickej nedokonalosti na minimum, zostáva nám ešte nejaký racionálny etický argument, ktorý by takéto manipulácie odsudzoval?

9 Je ľudská biologická podstata posvätná?

Jeden z častých etických argumentov proti genetickému manipulovaniu s ľudskou biologickou prirodzenosťou je v zásade náboženskej povahy: ľudská prirodzenosť (genetická informácia ľudského rodu) je niečo posvätné, čoho sa človek nesmie dotknúť (tabu) – je to danosť, niečo čo nevytvoril človek, ale vytvoril to Boh či Evolúcia (Príroda), niečo, čo tu už po desaťtisícročia úspešne funguje (úspešne z evolučného hľadiska, lebo sme až do dnešných dní nielen prežili, ale ľudská populácia závratne narastá – tento rok bude tvorená rekordnými siedmimi miliardami jedincov). To svedčí nepochybne o veľkom stupni dokonalosti, ktorá je výsledkom buď vševedúcej inteligencie alebo je to pomerne dokonalý výsledok miliónov rokov vývoja pomocou pokusov a omylov, ktorý mal dostatočne dlhý čas k tomu, aby v správnom pomere vybalancoval ľudský genóm do harmonického celku. Každý súdny človek si musí uvedomiť, že nie je radno narušiť komplexný celok (podobne ako je to v prípade globálneho zemského ekosystému) bez toho, aby sme ho dokonale poznali.

Prvá námietka, ktorá nás napadne je, že ak je človek tak dokonalý, prečo sa počas celej svojej existencie snaží zdokonaľovať, prečo nie je spokojný s tým, čo mu bolo dané „do vienka“ či už Bohom alebo Evolúciou? Ak je schopnosť sebazdokonaľovania vlastná ľudskej prirodzenosti, nie je celkom normálne, že je len otázkou času, kedy človek dospeje do situácie, že sa pokúsi vylepšiť aj svoju ľudskú prirodzenosť ?

Navyše si musíme uvedomiť, že spomínaný biokonzervatívny postoj ochraňujúci nedotknuteľnosť ľudskej genetickej informácie operuje skôr s argumentom rizika, než s čisto etickým princípom. Preto je namieste otázka, v čom môže spočívať čisto etická argumentácia proti alebo za vylepšovanie genetickej prirodzenosti ľudského druhu?

Dobrý príklad etického formulovania problému ilustruje otázka, ktorú si v tejto súvislosti kládol známy nemecký filozof Hans Jonas (1903-1993) - či môžeme, ba dokonca musíme, prijímať zodpovednosť za vývoj budúcich generácií. Presnejšie, či máme právo určovať genetický profil budúcich pokolení, bez ich vedomia a samozrejme informovaného súhlasu (ktorý je dnes štandardom v medicíne pri akýchkoľvek zásahoch do tela pacientov), v situácii, ktorú Jonas charakterizuje ako absenciu reciprocity práv a povinností, na ktorom spočíva celý náš doterajší etický diskurz ([14]:73).

Tento princíp zodpovednosti za budúcnosť ľudstva vo všeobecnosti neobstojí. M. Lappé napríklad argumentuje, že nemožnosť dovidieť všetky konzekvencie, ktorými by naše dnešné činy mohli ohroziť blaho (*well-being*) budúcich generácií, nemôže byť etickým zdôvodnením všeobecného zákazu zasahovať do dedičnej informácie človeka, pretože takýto zákaz by bol to isté, ako nedovoliť ľuďom z týchto dôvodov mať vlastné deti ([16]:160).

Známy obhajca vylepšovania človeka, britský bioetik John Harris argument zodpovednosti voči budúcim generáciám otáča o 180 stupňov. Podľa neho nezodpovedné nie je vylepšovanie budúcich generácií, ale ich nevylepšovanie. Vylepšovanie ľudí je preto morálnym imperatívom, našou povinnosťou ([13]:19-35). Reaguje tiež na Habermasov pokus, riešiť problém vylepšovania človeka v rámci diskurzívnej etiky, ktorej autorom je Habermas. Harris na tento pokus reaguje trochu ironickou poznámkou, že aj keď možno genetické manipulácie spôsobia, že jedinec nebude patriť do univerza morálnych bytostí (o čom však Harris pochybuje), čím by sme ho automaticky z morálneho diskurzu vylúčili, ešte stále pre takéhoto jedinca bude „obrovskou výhodou, že bude patriť do univerza zdravých bytostí ([13]:141)".

Voči Jonasovmu a Habermasovmu argumentu komunikačnej asymetrie medzi generáciami Harris argumentuje, že tu nejde o principiálne nič nové, že s takouto asymetriou sa stretávame úplne bežne v každodennej praxi a nepociťujeme to ako etický problém. Rodičia sa bežne rozhodujú za svoje deti, napríklad keď za ne súhlasia s ich očkovaním tesne po pôrode, pričom nie je na 100% vylúčené, že dieťa v dôsledku očkovania nezomrie. V medicínskej praxi sa bežne vyskytujú situácie tzv. predpokladaného súhlasu (*proxy consent*), keď sa lekári riadia tým, čo je najlepšie v pacientovom záujme, bez toho, aby mali jeho súhlas, napríklad v situácii, keď pacient nie je schopný takýto súhlas dať (nachádza sa napríklad v dôsledku autonehody v kóme).

Bolo by určite nezodpovedné voči budúcim generáciám, tvrdí Harris, keby sme dnes zakázali ich prarodičom, aby ich napríklad raz a navždy zbavili náklonnosti ochorieť na niektoré vážne, či dokonca smrteľné choroby, akou je AIDS. Ako príklad uvádza možnosť, že by pomocou genetických manipulácii mohli budúce generácie ľudí získať odolnosť voči HIV vírusu, prípadne niektorému druhu rakoviny ([13]: 22).

Týka sa však Harrisov argument aj kognitívnych schopností? Je hlúposť chorobou, ktorú je potrebné liečiť a toto liečenie je etické tak, ako akákoľvek iná liečba? Celkom logickým krokom je, že by sa hlúposť liečila v budúcnosti genetickou modifikáciou pohlavnej línie ľudí, pretože to by bol najefektívnejší spôsob liečby, pokiaľ by sa ukázalo, že je to geneticky podmienené ochorenie.

Harris obhajuje právo rodičov vylepšovať okrem seba samých aj kognitívne schopnosti svojich detí pomocou génovej manipulácie pohlavných buniek. Ignoruje fakt, že sa týmto nastolí selekcia kognitívnych schopností v pozitívnej spätnej väzbe, fenomén v biologickej evolúcii známy ako „runaway selection" ([25]: 169), ktorý vedie k zrýchľujúcej sa disruptívnej evolúcii, čo znamená, že pomerne veľmi rýchlo povedie k vytvoreniu dvoch ľudských subpopulácií stále viac sa líšiacich svojimi genofondami. Novovznikajúcu minoritnú subpopuláciu by sme mohli nazvať *genetickou aristokraciou* (Americký biológ Lee Silver ich nazýva *GenRich* [22]) a druhú, väčšinovú populáciu *genetickými naturščikmi* (*Naturals* u L. Silvera). Niet pochýb, že kognitívne vylepšení jedinci vďaka pozičnej výhode majú vysoké šance úspešne vyštudovať na najprestížnejších univerzitách a postupne obsadiť vo vedomostnej informačnej spoločnosti vďaka svojim schopnostiam a vedomostiam najprestížnejšie pozície, pochopiteľne so zodpovedajúcimi platmi, čo im umožní, aby si mohli dovoliť ešte modernejšie a ešte drahšie vylepšovadlá, a to nie len pre seba, ale aj svoje deti. Príslušníci genetickej aristokracie už nebudú súťažiť s naturščikmi, ale

medzi sebou, čím sa selekčný tlak ešte viac zosilní a disruptívna evolúcia urýchli [25]. Dá sa predpokladať, že v dôsledku toho veľmi rýchlo dôjde k sociálnemu separovaniu oboch subpopulácií až do tej miery, že sa nakoniec aj veľmi zriedkavé zmiešané páry genetickej aristokracie a naturščikov ukážu ako neplodné ([22]: 288). To bude bod, ktorý možno považovať za vznik nového, posthumánneho druhu, keďže reprodukčná bariéra je kľúčovým kritériom existencie samostatného biologického druhu.

Demokratické inštitúcie a s ním aj liberálno-demokratický typ štátu podľa Fukuyamu v dôsledku objavenia sa posthumánnych ľudí zaniknú [9]. Pôvodná ľudská rasa neobstojí v konkurenčnom boji s genetickou aristokraciou a bude ňou najprv podmanená a neskôr pravdepodobne úplne nahradená. George Annas v tejto súvislosti neváha transhumanizmus označiť za genocídu ľudského rodu ([1]).

Sú tieto obavy biokonzervativistov prehnané? Nie tak celkom, pokiaľ sa, prekvapujúco, nebudeme pozerať do budúcnosti, ale naopak do minulosti. Snahu kognitívne vylepšovať ľudskú rasu inšpirovanú vedcami sme tu už mali v prvej polovici minulého storočia. Poznáme ju pod zdiskreditovaným termínom eugenika.

Eugenika bolo spoločenské hnutie iniciované genetikmi na začiatku XX. storočia, ktoré si osvojili v rôznej miere aj vlády viacerých vyspelých západných demokratických spoločností (medzi nimi napr. Švédsko, Nórsko, Dánsko, USA, Kanada) a ktorého cieľom bolo „vyšľachtiť“ obyvateľstvo svojich krajín tak, aby bolo kognitívnejšie vyspelejšie. Keďže ešte celé desaťročia nemali vtedajší vedci v rukách metódy priamych genetických manipulácií, mohli v tejto dobe poslúžiť len nepriame spôsoby ako ovplyvniť želateľným smerom zloženie genofondu nejakej populácie. Buď pomocou pozitívnej selekcie (podpora inteligentnejších a vzdelanejších obyvateľov, aby mali čo najviac detí), alebo pomocou negatívnej selekcie (zabránenie, aby tí menej chytrí mali potomstvo, napr. tým, že ich štát nútene sterilizoval).

Nechválne známou sa stala eugenika potom, čo po skončení druhej svetovej vojny vyšli na verejnosť zverstvá páchané nacistami v koncentračných táboroch, ktorými naplňovali Hitlerov „eugenický“ projekt rasovej čistoty vyspelejšej a tým pádom nadradenej árijskej rasy, pomocou genocídy Židov, Rómov, a mentálne retardovaných Nemcov v plynových komorách.

S nástupom modernej genetiky v šľachtiteľstve a medicíne koncom XX. storočia a objavením možnosti ľudského oplodnenia *in vitro* („deti zo skúmavky“) sa opäť začal objavovať termín eugenika v súvislosti s novými metódami identifikovania genetického poškodenia ľudských plodov v prenatálnej diagnostike a s následnou interrupciou. Najnovšie je možné identifikovať gény už u niekoľkodňových ľudských embryí vytvorených *in vitro* ešte predtým, než sa implantujú do maternice ženy (tzv. predimplantačná genetická diagnostika – PiGD). To umožňuje selektovať na základe génov už medzi embryami a rozhodovať, ktoré z nich budú vybrané, aby pokračovali vo vývoji ďalej, a ktoré budú zničené.

Ak pominieme demagogický argument odporcov PiGD, že takáto selekcia embryí nie je ničím iným než modernou verziou nacistického zabíjania postihnutých ľudí v koncentračných táboroch, stále platí, že ide o proces výberu „dobrých“ génov do genofondu budúcej generácie, čo je definícia eugeniky. Zástancovia týchto metód namietajú, že rodičia majú právo na zdravé deti a že selekcia na úrovni osembunkových ľudských embryí je eticky obhájiteľná, pretože zničenie takýchto

embryí nie je zabitím človeka a je v istom zmysle ešte morálnejšie, než interrupcia, ktorá je vo väčšine moderných spoločností eticky i právne akceptovaná. V rámci moderného (bio)etického diskurzu je dôležité to, že o výbere embryí rozhodnú rodičia a nie štát. Preto sa v tejto súvislosti hovorí o liberálnej eugenike.

Ukazuje sa, že dôležitým etickým kritériom aj v rámci liberálnej bioetiky je, či ide o terapeutický cieľ (vybrať zdravé embryo spomedzi geneticky poškodených), alebo identifikovať embryá, ktoré majú určité genetické vlastnosti nadpriemerné alebo ináč vhodné pre ciele rodičov (napr. gény pre modré oči, nadpriemernú výšku či fyzickú dispozíciu, a v neposlednom rade, vysoké kognitívne schopnosti). V tejto súvislosti sa zaužívalo hovoriť o tzv. (geneticky) dizajnovaných deťoch. Zatiaľ sa väčšinou prijíma názor, že kým je eticky prijateľné využívať PiGD pre účely prevencie narodenia vážne geneticky poškodených detí, je eticky neprijateľné, aby si rodičia geneticky dizajnovali deti podľa svojich predstáv (obdoba diferencie medzi liečbou a vylepšovaním).

Jedným z hlavných argumentov, ktorý používa aj významný súčasný nemecký filozof J. Habermas je, že takýmto dizajnovaním rodičia limitujú neskoršiu slobodnú voľbu ich potomka žiť svoj vlastný život podľa svojich predstáv a rozhodnutí ([12]). Námietkou druhej strany je, že po tisícročia celkom bežne rodičia vnucujú svojim potomkom, či už výchovou, ale aj inými nátlakmi svoje predstavy o tom, čo majú ich deti študovať, aké by mali byť ich koníčky a pod. bez toho, aby to bolo považované za niečo neetické.

Avšak poučenie z histórie eugeniky by sme predsa len nemali podceňovať. Nemusíme pritom uvažovať o posthumánnej situácii (viď L. Silver vyššie), keď popri ľudskom druhu začne existovať nový, posthumánny druh. Veľké problémy sa môžu objaviť už počas samotného procesu smerujúceho k posthumánnemu druhu, teda v procese transhumanizácie. Najlepšie si uvedomíme problém vtedy, ak pripustíme, že vďaka génovým manipuláciám sa začnú bohatým rodičom (ktorí si to budú môcť finančne dovoliť) rodiť deti s kognitívnymi schopnosťami géniov. Postupne, ako ich podiel bude v spoločnosti narastať, oni sa stanú novou normou kognitívnych schopností a ľudia s normálnymi kognitívnymi schopnosťami sa ocitnú v pozícii „mentálne retardovaných." Je to situácia, ktorú mnohí z nás poznajú zo známeho filmu GATTACA ([10]). Geneticky nevylepšení jedinci budú pochopiteľne v pozičnej nevýhode oproti geneticky vylepšeným, čo vážne nabúrava princíp spravodlivosti, na ktorom stoja moderné liberálno-demokratické spoločnosti – princíp rovnosti šancí.

Ale aj na tento argument majú zástancovia transhumanizmu odpoveď. Jednoducho vymyslíme rôzne systémy na vyrovnávanie šancí medzi bohatými, ktorí si vylepšenia môžu zaplatiť a chudobnými, ktorí na to nebudú mať peniaze. Napríklad systém navrhnutý M. Mehlmanom a J. Botkinom, ktorý P. Singer označil za geniálny – štát by podľa tohto návrhu mal organizovať špeciálnu lotériu, v ktorej by sa ako výhra rozdávala chudobným možnosť geneticky sa vylepšiť v rovnakom rozsahu, ako to robia bohatí za peniaze. Dokonca nemusel by to byť štát, ktorý by zabezpečoval finančné prostriedky pre túto lotériu, mohlo by sa to diať prostredníctvom osobitnej dane, ktorá by sa uvalila na využívanie genetických technológií [23: 285].

Napriek intenzívnej diskusii v akademických kruhoch zostáva stále otvorená otázka, do akej miery sú techno-optimistickí zástancovia transhumanizmu, ktorí

obhajujú permisívny a liberálny prístup spoločnosti k vylepšovaniu individuálnych ľudských schopností nezodpovednými hazardérmi s budúcnosťou ľudstva, alebo naopak, sú skôr racionálnymi vizionármi, podľa ktorých hazardérmi sú biokonzervativisti, ktorí v dôsledku iracionálnych obáv chcú zakonzervovať biologickú podstatu ľudského rodu aj za cenu toho rizika, že ľudstvo nebude biologicky pripravené primerane reagovať na radikálne nové výzvy budúcnosti.

10 Záver

Ocitli sme sa v situácii, keď vďaka modernej vede sme schopní nielen opravovať našu biologickú konštitúciu, ale ju dokonca vylepšovať. A to v mnohých smeroch týkajúcich sa našich daností a schopností, vrátane tých kognitívnych. Vďaka tomu sa hlúpejší môžu stať chytrejšími, a chytrejší ešte chytrejšími. Koľko nešťastia je dôsledkom púhej ľudskej hlúposti! Znamená to, že chytrejšia ľudská spoločnosť automaticky prinesie šťastnejší život svojím členom? A ktorým? Všetkým, len niektorým alebo v priemere?

Ľudstvo sa vylepšuje od nepamäti. Ale až teraz toto vylepšovanie stojí na prahu radikálneho zlomu – môže sa týkať samotnej biologickej podstaty ľudskosti. Otázka preto znie: je kognitívne vylepšovanie človeka správnou alebo nesprávnou cestou do budúcnosti? Posudzovanie správnosti či nesprávnosti nejakého konania je etickým problémom. Aké kritéria máme zvoliť v prípade kognitívneho vylepšovania, na základe ktorých budeme pomeriavať, či niečo je alebo nie je eticky prijateľné?

Pokúsili sme sa v tejto kapitole stručne prezentovať hlavné etické problémy v súvislosti s kognitívnym vylepšovaním a pokusy nájsť určité riešenia, tak ako sa objavujú v rámci bohatej diskusie prebiehajúcej v akademických kruhoch medzi etikmi, bioetikmi a neuroetikmi posledných asi dvadsať rokov. Ukázali sme, ako sa hľadanie etických kritérií spojených s problematikou kognitívneho vylepšovania ľudí sústreďuje do odpovedí predovšetkým na štyri okruhy otázok: otázku etickej relevantnosti rozlišovania medzi terapiou a vylepšovaním, otázku mentálneho dopingu a podvádzania, otázku spoločensky spravodlivého prístupu k možnostiam vylepšovania, otázku vylepšovania ľudskej biologickej podstaty.

Poďakovanie: Táto kapitola vznikla za podpory grantových agentúr KEGA a VEGA v rámci riešenia projektov č. 3/7300/09 a č. 1/0132/10.

Literatúra

[1] Annas, G.: Genism, Racism and the Prospect of Genetic Genocide. In: *The Future of Values*. J. Bindé. Paris and Oxford-New York, UNESCO and Berghahn Books 2004, pp. 284-288.

[2] Bostrom, N.; Sandberg, A.: *Cognitive Enhancement: Methods, Ethics, Regulatory Challenges*. Oxford University 2009.

[3] Bostrom, N.; Roache, R.: *Smart Policy: Cognitive Enhancement and the Public Interest*. Oxford University 2009.

[4] Bostrom, N., Roache, R.: Ethical Issues in Human Enhancement, In: *New Waves in Applied Ethics*, ed. Ryberg, J., Petersen, T.S., Wolf, C., Palgrave Macmillan 2007.

[5] Buchanan, A., Brock, D., Daniels, N., Wikler, D.: *From Chance to Choice. Genetics and Justice*. Cambridge: Cambridge University Press 2001.

[6] Caldwell, J. A., Jr., Caldwell, J. L., Smythe, N. K., 3rd and Hall, K. K.: A Double-Blind, Placebo-Controlled Investigation of the Efficacy of Modafinil for Sustaining the Alertness and Performance of Aviators: A Helicopter Simulator Study. *Psychopharmacology*, **150** (2000) 272-282.

[7] Daniels, N.: Growth Hormone Therapy for Short Stature: Can We Support the Treatment/Enhancement Distinction? *Growth: Genetics & Hormones,* **8** (1992) 46-48.

[8] Fukuyama, F.: „Transhumanism.“. *Foreign Policy*, No. 144 (Sep. - Oct., 2004) 42-43.

[9] Fukuyama, F.: *Our Posthuman Future*. New York: Farrar, Straus and Giroux 2002.

[10] *GATTACA:* film (réžia: Andrew Niccol), 1997.

[11] Gazzaniga, M.S.: *The ethical brain.* New York: Dana Press 2005.

[12] Habermas, J.: *Budoucnost lidské přirozenosti. Na cestě k liberální eugenice?* Praha: Filosofia 2003.

[13] Harris, J.: *Enhancing Evolution: The Ethical Case for Making Better People*. Princeton University Press, New Jersey 2007.

[14] Jonas, H.: *Princip odpovědnosti*. Praha: OIKOYMENH 1997.

[15] Kamm, F.: Is There a Problem with Enhancement? *American Journal of Bioethics*, **5** (2005) 5-14.

[16] Lappé, M.: Ethical Issues in Manipulating the Human Germ Line. In: *Bioethics: An Anthology*. Kuhse H. a P. Singer (eds). Oxford: Blackwell Publishers 1999, pp. 155-164.

[17] Matějková, E.: Od človeka k post-človeku, In: *Identita - Diferencia* (4. slovenský filozofický kongres SFZ, 13. – 15. 9. 2010, Smolenice) (v tlači).

[18] Naam, R.: *More Than Human,* Broadway Books, New York 2005.

[19] President's Council on Bioethics: *Beyond therapy: biotechnology and the pursuit of happiness*. New York: Dana Press 2003.

[20] Sandel, M.J.: *The Case against Perfection: Ethics in the Age of Genetic Engineering*. Cambridge 2007.

[21] Savulescu, J. - Bostrom, N.(Eds.).: *Human Enhancement*. Oxford: Oxford University Press 2009.

[22] Silver, L.: *Remaking Eden.* New York: Avon Books, Inc. 1998.

[23] Singer, P.: Parental Choice and Human Improvement. In: Savulescu, J.- Bostrom, N.(Eds.).: *Human Enhancement*. Oxford: Oxford University Press 2009, pp. 276-289.

[24] Stix, G.: Turbocharging the Brain. In: *Scientific American* (2009) 47-48.

[25] Sýkora, P.: *Bioetika transhumanizmu*. In: Realita a fikcia (9. medzinárodná vedecká konferencia SFZ, 12.-13.10. 2009, Smolenice), R. Karúl a M. Porubjak (eds.) SAV a UCM, Bratislava 2009, s. 169-175.

[26] Tsien, J. et al: Genetic enhancement of learning and memory in mice. *Nature* **401** (1999) 63-69.

[27] Turner, D. C., Robbins, T. W., Clark, L., Aron, A. R., Dowson, J. and Sahakian, B. J.: Cognitive Enhancing Effects of Modafinil in Healthy Volunteers. In: *Psychopharmacology*, **165** (2003) 260-269.

[28] Turner, D., Sahakian, B.: The cognition enhanced classroom. In: Miller, P.; Wilson, J.: *Better Humans? The politics of human enhancement and life extension*, Demos: London 2006, pp. 79-85.

Ambientní inteligence - očekávání, aplikace a rizika

Peter MIKULECKÝ[1]

Abstrakt. Oblast ambientní inteligence, tedy oblast, která zkoumá možnosti vytváření inteligentních prostředí schopných lidem pomáhat řešit obtížné, ale i běžné každodenní úkoly, je v současnosti předmětem intenzivního výzkumu provázeného podobně intenzivní snahou o vývoj nových zajímavých a užitečných aplikací. V kapitole se budeme zabývat analýzou současného stavu v této oblasti, přičemž budeme konfrontovat očekávání, která výzkum a aplikace ambientní inteligence provázejí, s dosaženými výsledky v případech několika typických aplikací. Vyjdeme-li z obvykle používané multi-agentové architektury těchto prostředí, budeme na danou oblast pohlížet jako na možnou symbiózu lidí a „inteligentních" věcí, jež je v inteligentním prostředí budou obklopovat a upozorníme na některá více či méně pravděpodobná rizika, která další rozvoj ambientní inteligence mohou ohrožovat.

1 Úvod

Oblast tzv. ***ambientní inteligence*** (angl. ***Ambient Intelligence***, což lze volně přeložit jako ***všudypřítomná inteligence*** nebo i ***inteligence prostředí***), je v období posledních cca 5 let předmětem intenzivního výzkumu zejména v zemích Evropské unie, jejíž nedávný 6. i současný 7. rámcový program výzkumu považuje tuto oblast za jednu ze stěžejních pro další rozvoj znalostní společnosti. Představa o ambientní inteligenci vychází z původní vize Marka Weisera [34] o počítačích pro 21. století a předpokládá posun ve využívání informačních technologií od osobních počítačů k různým typům informačních zařízení, které budou integrovány do běžného životního prostředí lidí a budou jim moci napomáhat při řešení běžných i méně běžných životních či pracovních situací. V souvislosti s tím se rovněž předpokládá potlačení dnes převažujících technologických aspektů výpočetních a informačních technologií do pozadí, přičemž do popředí zájmu se přesunou inteligentní rozhraní konkrétních uživatelských prostředí. Další podrobnosti k celé problematice lze nalézt např. v [2], [3], [5], [6], [9],

[1] Fakulta informatiky a managementu, Univerzita Hradec Králové, Rokitanského 62, 500 03 Hradec Králové, e-mail: peter.mikulecky@uhk.cz

[12] nebo [13], či [20], [22] nebo [33], ale i v celé řadě dalších titulů v současnosti dost početné odborné literatury z této oblasti.

První výsledky v oblasti ambientní inteligence jsou docela slibné. Rozvíjí se zejména aplikovaný výzkum, zaměřený na řešení inteligentních prostředí např. pro fyzicky hendikepované občany, nebo pro přestárlé [21]. Jiné směry výzkumu se orientují na rozvoj technologií pro realizaci prostředí s ambientní inteligencí. Zatím je však málo upozorňováno na případná nebezpečí, která tato technologie a zejména pak její možné budoucí rozsáhlé nasazení v sobě skrývá. Současná, více méně konzumní společnost, nepochybně uvítá (pokud tak už částečně neudělala) pohodlí a nezpochybnitelné přínosy, které z nasazení technologií ambientní inteligence plynou či mohou vyplývat. V této kapitole bychom rádi čtenáře uvedli do problematiky ambientní inteligence, popsali několik možností aplikací této technologie, ale i upozornili na některá nebezpečí daného přístupu.

2 Co to je ambientní inteligence?

2.1 Úvodní představa

Ambientní inteligence (angl. ***Ambient Intelligence***) je vize informační společnosti budoucnosti, ve které je největší důraz kladen na:

- uživatelskou přívětivost,
- efektivní a distribuovanou podporu služeb,
- posílení možností efektivního využívání relevantních zdrojů ze strany uživatelů,
- podporu interaktivní činnosti uživatelů.

Tato vize předpokládá, že lidé budou působit v prostředí inteligentních a intuitivně využitelných rozhraní k procesorům, umístěným do všech typů objektů. Pod objekty zde rozumíme zcela běžné předměty, které nás obklopují a které tvoří prostředí, v němž působíme. Takovéto prostředí bude pak schopné rozpoznat přítomnost konkrétních individuí a reagovat na jejich přítomnost nevyrušujícím a obvykle i neviditelným způsobem, který bude plně integrován do dané situace. Tato reakce prostředí by měla zásadně působit ve prospěch daného uživatele, měla by mu pomoci řešit problémy, které mohou jeho existenci či činnost v daném prostředí komplikovat.

Cílem využití ambientní inteligence je podstatně rozšířit interakci lidí a digitálních informačních technologií zejména využitím všudypřítomných (angl. ubiquitous) výpočetních zařízení. Jádrem představy o ambientní inteligenci jsou tři vzájemně propojené a v současnosti se prudce rozvíjející technologie:

- všudypřítomné výpočty (ubiquitous computing)
- všudypřítomná komunikace (ubiquitous communication)
- inteligentní uživatelská rozhraní (intelligent user interfaces)

Všudypřítomné výpočty znamenají zejména integraci mikroprocesorů do běžných objektů denní potřeby, jako je nábytek, oděv, kuchyňská zařízení, ale i hračky, apod. V podstatě se jedná o použití počítačů doslovně všude. Počítače jsou zde dostupné pomocí prostředků fyzického prostředí, jsou nicméně obvykle skryty očím uživatele.

Všudypřítomná komunikace umožňuje kromě jiného třeba to, že různé objekty s integrovanými procesory budou schopny komunikovat mezi sebou či s uživatelem především prostředky bezdrátové komunikace, která se postupně stane dominantní technologií.

Inteligentní uživatelská rozhraní umožňují obyvatelům prostředí s ambientní inteligencí interakci s tímto prostředím a jeho ovládání přirozenou cestou (hlas, gestikulace, dotyky) s vysokým stupněm personalizace (zohlednění preferencí uživatele či kontextu s danou situací nebo činností). Toto všechno umožní silné zjednodušení používání různých zařízení, která budou využitelná velice přirozeně a intuitivně.

V rámci výzkumu ambientní inteligence lze zaznamenat veliké úsilí o vyvíjení nových typů uživatelských rozhraní. Výsledkem je mimo jiné i proměna uživatelských rozhraní z dnes zatím obvykle používaných displejů na vysoce intuitivně využitelná rozhraní ovládána hlasem, dotykem, apod. V té souvislosti můžeme zaznamenat i další nový pojem - inteligentní předměty (smart objects), což jsou předměty běžného každodenního využití, které však navíc mají a projevují jistou inteligenci v důsledku činnosti v nich zabudovaných mikroprocesorů.

Možnosti realizace ambientní inteligence v různých typech prostředí jsou zkoumány v mnoha směrech[2]:

- vzájemné rozpoznávání a přizpůsobování zařízení, která utvářejí prostředí,
- personalizace a adaptace systémů s ohledem na potřeby a preference uživatelů,
- proaktivní jednání prostředí, se snahou o anticipaci úmyslů uživatelů a na základě toho směřování konkrétní podpory uživatele,
- porozumění dynamice okolního prostředí, na základě čehož pak systém upravuje svou další činnost,
- podpora vzájemné součinnosti a spolupráce jednotlivých komponent a systémů ve sdíleném prostředí,
- autonomie, samoregulace a samoopravitelnost komponent.

Na všechny právě uvedené body lze rovněž nahlížet i jako na vlastnosti, které bychom od ambientní inteligence mohli a měli požadovat.

---

[2] Zde se mísí názory celé řady autorů, a to nejenom z oblasti ***ambient inteligence***, ale i z nezávisle rozvíjejících se příbuzných konceptů známých a rozvíjených pod názvy ***pervasive environments***, ***smart environments***, ***intelligent environments*** nebo ***ubiquitous environments***.

2.2 Scénáře pro ambientní inteligenci

Pro ilustraci představy o ambientní inteligenci byla vytvořena celá řada tzv. scénářů, osvětlujících možné přínosy této technologie pro společnost. Scénáře jsou více méně příklady možného budoucího využití přístupů ambientní inteligence v rozmanitých oblastech každodenního lidského života [12]. Stručně si je připomeňme společně se vzpomínkou na jejich nedávné desetileté jubileum, které dává možnost hodnocení, kam tato oblast od původních ideových představ za deset let dospěla.

První představa specielní pracovní skupiny ISTAG, kterou Evropská komise pověřila vytvořením základního dokumentu pro další výzkum a vývoj v oblasti ambientní inteligence, byla v dokumentu [12] vyjádřena v podobě následujících čtyř scénářů[3]:

- Maria – The Road Warrior,
- Dimitrios and Digital Me,
- Carmen – Traffic, Sustainability, and Commerce,
- Annette and Solomon in the Ambient for Social Learning.

První scénář, **Maria - The Road Warrior** - pojednává o mladé reprezentantce jedné firmy, která se přesouvá do jiného města na plánované obchodní setkání. K dispozici má vysoce sofistikované osobní komunikační zařízení (P-Com), skrze něž posléze prostředí města, kam Maria přichází, s Marií komunikuje o všem, co pro ni toto prostředí připravilo a zařídilo. Po příjezdu či příletu na místo má Maria připravené vypůjčené auto, zároveň jí prostředí zařídilo povolení přístupu autem do města, rezervovalo parkovací místo a pokoj v hotelu. Prostředí pokoje je samozřejmě předem personalizováno dle požadavků Marie.

Na obchodním jednání je Marii ze strany prostředí poskytnuta veškerá informační i znalostní podpora, je připravena její prezentace a vše k ní potřebné, v rámci čehož se Maria může dle potřeby připojovat ke všem firemním zdrojům a databázím, bude-li to nutné.

Večer po schůzce je pro Marii připraveno několik možností relaxace respektující její zvyklosti, momentální náladu či míru její únavy.

Podíváme-li se na první scénář z dnešního pohledu, lze konstatovat, že nic z toho už více méně není nemožné. Jako P-Com lze využít dnešní chytré telefony, problém samozřejmě může být v jejich propojení s požadovanými službami a zejména míra automatizace poskytování těchto služeb. Je jistě rozdíl v tom, zda inteligentní prostředí, detekující příjezd Marie do města samo iniciuje komunikaci Mariina P-Comu s příslušným umělým partnerem v hotelu a „domluví“ s ním veškeré potřebné detaily Mariina pobytu, aniž by tím Marii jakkoliv obtěžoval či zatěžoval, nebo zda je Maria nucena toto vše domluvit svým chytrým telefonem osobně. Intenzivní výzkum je směřován do oblastí, které by právě zmíněnou automatickou komunikaci umožnily.

[3] Ponecháváme původní anglické názvy scénářů z toho důvodu, že české názvy více méně nebyly vůbec vytvořené, neměly proto ani čas a možnost se v domácí odborné komunitě více ujmout.

Druhý scénář - **Dimitrios and Digital Me** je zaměřen na spolupráci člověka (Dimitriose) s jeho avatarem (tedy digitálním zástupcem). Digital-Me (D-Me) je digitální zařízení reprezentující Dimitriosova avatara, jenž je schopen v zastoupení Dimitriose přebírat všechny pokusy o kontakt s Dimitriosem (zejména telefonické hovory či zprávy) a vyřizovat je místo Dimitriose, po vyhodnocení jejich naléhavosti.

Hovory, které vyhodnotí jako zásadně důležité, pak přesune na samotného Dimitriose, zejména pokud se D-Me nepovede vyřídit vše potřebné s D-Me volající osoby. Při důsledném fungování D-Me může být osoba, kterou D-Me zastupuje, velice významně odlehčena od různých nedůležitých a vyrušujících záležitostí, jež je D-Me schopen vyřídit samostatně a pouze o nich posléze svého majitele informuje.

Z dnešního pohledu tento scénář zřejmě ještě není plně realizovatelný. Jsou sice k dispozici technicky velice pokročilá zařízení (chytré telefony či tablety), ale stále ještě nejsou nadány takovou inteligencí, aby dokázaly ve výše uvedeném smyslu člověka zastoupit. Zde je ještě celá řada velice zajímavých cílů pro výzkum v umělé inteligenci i v ambientní inteligenci.

Třetí scénář - Carmen - **Traffic, Sustainability, and Commerce** - je scénář zaměřený na podporu běžných aktivit zaměstnané ženy, která se potřebuje v nějakou dobu dostat na několik míst, nakoupit pár věcí a dostat se včas domů.

Inteligentní prostředí jí najde a zařídí možnosti přepravy na potřebná místa, optimalizuje její přesuny, vyhodnotí potřeby týkající se nákupů, upozorní na případné slevy v obchodech a doporučí jí výhodné nákupy, ochraňuje ji před možnými nehodami a naplánuje celou její misi tak, aby se dostala v plánovanou dobu domů.

Tento scénář je dnes realizovatelný, i když možná ještě ne úplně automaticky. Různé úrovně navigace, seznamy bodů zájmů (PoI), možnost získání informací v kontextu s místem kde se člověk nachází či kam se chystá zajít, to vše jsou cesty k naplnění scénáře Carmen. Technicky dnes už nic z uvedeného není nemožné a lze zaznamenat snahy o vytvoření podobných systémů.

Čtvrtý scénář - **Annette and Solomon in the Ambient for Social Learning** - je scénář popisující inteligentní prostředí (učebnu) pro podporu výuky či diskusí studijních skupin. Učebna si „pamatuje“ každého studujícího, jeho dosavadní postup v učení, připravuje mu vhodně zvolené studijní materiály, organizuje individuální program učení, apod. Učebna rovněž organizuje i sdílené vyučování několika osob dle jejich mentálního stavu a progresu ve studiu.

Učebna je rovněž schopna probírat s odpovědným pedagogem progres skupiny i individuálních studujících a přizpůsobovat celý program výuky dle potřeby.

V dnešní době velice rozvinutých systémů a technologií pro podporu výuku ani tento scénář není nepředstavitelný. Technologie pro podporu výuky se stále rozvíjejí a zdokonalují a v kombinaci s pokročilými postupy ambientní inteligence realizaci scénáře už téměř v plné míře umožňují.

3 Multi-agentové architektury v ambientní inteligenci

Koncept ambientní inteligence (ambient intelligence) předpokládá maximální, až všudypřítomné integrování výpočetních a síťových technologií do prostředí tak, aby

z hlediska uživatele přestalo být podstatné, že s technologiemi vůbec pracuje, případně o kolik aplikací se jedná, jak jsou propojeny a jak fungují [12].

Na prostředí, výše uvedeným způsobem protkané technologiemi, můžeme velice výhodně nahlížet jako na multiagentový systém, tedy distribuovaný, decentralizovaný systém autonomních, heterogenních a navzájem interagujících entit (agentů) [17]. Síla multiagentových systémů spočívá v možnosti propojit různé druhy agentů, disponující rozdílnými možnostmi a schopnostmi zpracovávat nejrůznější druhy vstupů.

Agent je, jak známo, schopný autonomně a flexibilně jednat v dynamickém prostředí [29]. Dle [35] je to umělá výpočetní entita, která může realizovat jisté úlohy s jistým stupněm autonomie či iniciativy, přičemž se inteligentně přizpůsobuje svému okolí. Z hlediska těchto definic člověk není považován za agenta, z pohledu ambientní inteligence je ovšem mnohdy užitečné člověka mezi agenty zahrnout.

Při snaze budovat ambientní inteligenci mohou být agentové systémy inspirativní ve třech směrech, a sice jako [10]:

- metafora, na základě které lze uvažovat o komplexních, distribuovaných výpočetních systémech,
- zdroj technologií, pomocí nichž lze takové systémy realizovat,
- inteligentní uživatelská rozhraní.

3.1 Agenty jako metafora

Systémy utvářející inteligentní prostředí jsou orientované na funkce, zatímco agenty tradičně chápeme jako entity, určené k plnění konkrétních úloh.

V agentovém prostředí lze uvažovat skupiny agentů, které spolupracují a/nebo sdílejí nějaký účel, např.:

- agent poskytující informace a agent přiřazený displeji,
- osobní agenty skupiny jednotlivých uživatelů,
- množina agentů reprezentující osobní zařízení (např. počítač) jednoho uživatele.

V rámci určené skupiny mohou agenty komunikovat na základě role, kterou v této skupině zastávají. Lze očekávat, že takto vymezený pojem skupiny pomůže uživateli získat lepší možnost ovládat interaktivní prostředí, nebo v něm explicitně vidět kombinace zařízení, služeb či relevantního obsahu. Úlohy a aktivity právě zapojeného uživatele jsou každá reprezentovány jinou skupinou agentů.

Propojení agentů s jednotlivými zařízeními, službami či obsahem v rámci daného prostředí řešil Misker s kolektivem [19] vytvořením tzv. ad hoc agentového prostředí. Propojení mezi agenty vzniká v průběhu řešení nějakého úkolu a nutnou podmínkou přitom je sdílení jistých objemů znalostí mezi agenty a rovněž i sdílení komunikačních protokolů. Namísto interakce s individuálními zařízeními pak uživatel interaguje s inteligentním prostředím jako s celkem. Zařízení a služby je nutno chápat jako více méně nezávislé, aby byla naplněna představa o autonomii agentů – tím však dochází k jistému pnutí mezi snahou uživatele mít plnou kontrolu nad prostředím a autonomií agentů. Možným řešením je zde právě zavedení již zmíněného konceptu

spolupracujících skupin lidí a agentů jako způsobu, jak uživateli poskytnout plnou kontrolu nad spolupracujícími agenty.

3.2 Agenty jako technologie

Současné agentové technologie, realizované jako architektury, relevantní techniky a algoritmy jsou dobře aplikovatelné při vytváření různých typů inteligentních prostředí. Příklady těchto aplikací jsou uvedeny např. v [5], [6], [11], [19], [24], [25], [28], [30], [31] či [32].

Jak jsme již uvedli dříve, vize inteligentního prostředí s ambientní inteligencí je představou prostředí založeného na nenápadných, mnohdy až neviditelných interakcích mezi lidmi a počítačovými službami v globálním výpočetním prostředí. Služby v takovémto inteligentním prostředí se chápou jako všudypřítomné v tom smyslu, že je nezprostředkuje nějaký speciální poskytovatel, ale místo toho jsou přidružené všem druhům objektů či zařízení v prostředí, přičemž tato zařízení v ničem nepřipomínají dnešní počítače. Lidé budou interagovat s těmito službami prostřednictvím inteligentních a intuitivních rozhraní, začleněných do těchto objektů a zařízení, které budou vstřícné lidským potřebám.

Veliká třída předpokládaných aplikací v inteligentních prostředích by měla přinášet užitek obyčejným lidem v běžném sociálním kontextu. Tento požadavek přináší potřebu vytvoření takových technologií, které jsou transparentní. V tomto kontextu lze transparentnost chápat jako vlastnost, která uživateli pomůže pochopit, je-li to potřeba, funkcionální chování objektu či zařízení v prostředí. Jednoduše řečeno, transparentnost by měla posunout interakce v inteligentním prostředí blíže způsobu, jakým uvažují lidé, než ke způsobu, jakým pracují počítače.

Další výzvou, kterou dle [32] představa inteligentních prostředí navozuje je, že elektronická infrastruktura prostředí by měla působit inteligentně ve prospěch lidí. Inteligentní prostředí těžko může vystačit s představou jednoho přímo ovládaného rozhraní, skrze které uživatel požaduje výsledky od nějakého monolitického systému. Nabízí se spíše představa, dle které musí být konceptuální složky takovéhoto prostředí často nejenom schopné jednoduché spontaneity, ale musí být i proaktivní, tedy chovající se tak, jakoby to byly agenty jednající v zájmu lidí. Pro pochopení složek inteligentních prostředí se tedy ukazuje být přirozené využít agentové metafory. Agent v tomto kontextu může být softwarová (ale i hardwarová) entita, která vnímá či ovlivňuje prostředí, má znalosti o tomto prostředí a o svých vlastních cílech, a umí proaktivně plánovat dosažení svých cílů nebo cílů svých uživatelů, čímž lze pomocí složené interakce elektronického a fyzického prostředí dosáhnout užitečného výstupu pro lidské uživatele.

Nejdůležitějším technologickým aspektem každého složitého softwarového systému je jeho architektura. V případě ambientní inteligence architektura inteligentních prostředí musí zohlednit strukturu složek systému, jejich vzájemné vztahy, jejich konstrukci a vývoj v čase. Zvláštností systémové architektury inteligentních prostředí je, že se musí přizpůsobovat dynamickým požadavkům nových uživatelů. Z tohoto aspektu je nutno uživatele považovat za stále mobilnější. Předpokládá se totiž jejich dynamicky se měnící interakce se stále jinými

inteligentními prostředími – domov, auto, pracovní prostředí, letiště, restaurace, atd. V každém takovém prostředí je nutné, vzhledem k pohybu uživatele, zabezpečit jeho přístup ke zdrojům prostředí (ke službám, prostředkům, atd.). Zde je důležitým předpokladem funkčnosti prostředí to, že jsou jeho jednotlivé komponenty navrženy tak, aby byly schopné vzájemné komunikace, ne-li přímo „dohovoru“ o společném postupu, kterého účelem by za všech okolností měla být podpora aktivit uživatele, pomoc při řešení jeho problémů, apod. Uživatel totiž celkem právem očekává, že přechodem do nového prostředí neztratí nic na možnostech plynule pokračovat ve své, již započaté činnosti, ať už pracovní, nebo jiné. Od inteligentního prostředí se pak očekává, že se požadavků uživatele přizpůsobí a adaptuje se na ně.

K právě popsanému účelu celá řada řešení spoléhá na multi-agentové systémy navržené na různých úrovních abstrakce. Jako příklad lze zde uvést zejména práce [8], [11], nebo [28]. Ze specifikace FIPA [7] vychází např. architektura ASK-IT [31], orientovaná na vytvoření inteligentního prostředí podporujícího aktivity tělesně postižených lidí, přičemž autoři rozšířili architekturu, kterou FIPA specifikovala pro aplikace zaměřené na podporu individuálního cestování (PTA – Personal Travel Assistance).

Mezi další aplikační oblasti pro použití multi-agentových přístupů k vytvoření inteligentního prostředí patří, kromě již zmíněných, především:

- inteligentní prostředí pro domácnosti či celé rodinné domy, zahrnující kromě zajišťování běžných potřeb (jako je topení, či zalévání zahrad) i méně běžné, jako je inteligentní provoz domácího kina,
- inteligentní kanceláře, zde jako příklad lze uvést koncept Smart Office [8], i náš výzkum zaměřený na vytvoření prostředí podporujícího vysokou kvalitu rozhodování manažerů [15],
- inteligentní zdravotnická zařízení, sledující podobný cíl, jako již zmíněný výzkum Spanoudakise a Morakise [31],
- inteligentní vozidla či dokonce inteligentní provoz mnoha vozidel v nějakém prostředí.

Na základě již řečeného lze konstatovat, že dvě základní kritéria pro posouzení úspěšnosti inteligentního prostředí mohou být vymezeny jako:

- kvalita rozhodování inteligentních entit, tvořících toto prostředí, a
- schopnost učení se skupin těchto entit (agentů) při snaze zlepšovat úroveň a personalizaci služeb tímto prostředím poskytovaných.

3.3 Agenty pro uživatelská rozhraní

Jedním ze zásadních rysů inteligentních prostředí (zejména vycházejících z konceptu ambientní inteligence) je proměna uživatelských rozhraní z dnes zatím obvykle používaných displejů na vysoce intuitivně využitelná rozhraní ovládána hlasem, dotykem, apod. Pro takovéto typy uživatelských rozhraní může být využití agentové technologie velkým přínosem, jak již ukazují početné zkušenosti různých autorů (viz [8], [11] a další).

V tradičnějších prostředích multimodalita v interakci s výpočetními prostředky již byla a je samozřejmě zkoumána, chybí zatím dostatek výsledků o tom, jak lze dosažené výsledky přenést do prostředí, ve kterém se uživatel obvykle neobrací (a ani nemůže obracet) na konkrétní obrazovku (či její část) nebo na konkrétní objekt. Nijholt a jeho kolegové v této souvislosti upozorňují [23], že většina výzkumu ambientní inteligence nevěnuje pozornost skutečnosti, že člověk se může cítit ztracený v takovémto prostředí, že třeba může mít pocit, že neví, s kým mluví, resp. nemusí být schopen si vybudovat vztah k anonymnímu prostředí, které jej sice podporuje, ale rovněž trvale sleduje a zaznamenává jeho aktivity.

V [1] je uveden hezký přehled interakce uživatele s inteligentním prostředím, přičemž se upozorňuje na potřebu komplexněji pochopit způsoby, jakými zde uživatelé provádějí úlohy a aktivity. Uživatelská interakce s agenty má několik důležitých bodů, které ovlivňují technické možnosti uživatelské interakce:

- autonomie agentů,
- načasování a dobu splnění,
- dostupnost a objevování,
- identita agentů.

Dosavadní metafora "desktop computing", tedy přímá manipulace s okny, ikonami, menu, kurzorem apod. jako hlavní způsob interakce člověka s počítačem se v inteligentním prostředí nahradí něčím, co lze nazvat "room metaphor". „Room metaphor" lze chápat jako interakci člověka se soubory předmětů, které jej obklopují. Tyto „inteligentní předměty" (smart objects) mají specifické funkce a taková rozhraní, aby je člověk mohl používat způsoby, které jsou pro něj obvyklé. Využití agentových přístupů je tedy v tomto směru velikou výzvou.

3.4 Požadavky na použití agentů

Empirická zjištění z mnoha dosavadních výsledků výzkumu inteligentních prostředí ukazují, že:

- uživatelé vždy potřebují mít (resp. rozhodně by měli mít) kontrolu nad prostředím, ve kterém se pohybují,
- problematika zabezpečení osobních údajů, resp. důvěryhodnosti inteligentních prostředí se jeví být velice zásadní pro další rozvoj této oblasti,
- uživatelé nesmějí být technologiemi rušení či obtěžováni; tj. podpůrné inteligentní systémy jim nesmějí komplikovat stávající způsoby řešení problémů,
- učení systému musí být pokud možno implicitní, nesmí uživatele žádným způsobem zatěžovat,
- podstatným požadavkem je potřeba personalizace služeb prostředím poskytovaných.

Vše zmíněné lze použitím agentových architektur dosáhnout, nebo alespoň položit vhodný základ k naplnění těchto požadavků. Je zde však stále ještě veliký prostor pro další výzkum.

4 Některé aplikační možnosti

4.1 Akademické prostředí

V předchozí části jsme se již zabývali vizí ambientní inteligence a na ní založené představě o inteligentních prostředích obecně, co to ale může znamenat např. v univerzitním prostředí? Především, perspektiva všudypřítomných počítačů a z toho plynoucí možnosti pro využití ambientní inteligence nám dává širší perspektivu pro vytváření studijních skupin tak, abychom brali v úvahu i širší kontext týkající se studujících, např. místo vzdělávání, čas a dostupnost různých zdrojů (či už technických, jako jsou počítače, nebo vzdělávacích a dalších). Tato perspektiva může mít značný dopad na efektivnost univerzitního vzdělávání, zejména v případě kombinované či distanční formy vzdělávání. Navíc může ovlivnit i složitost různých administrativních procesů v rámci univerzity. To vše však vyžaduje takové změny odpovídajícího univerzitního prostředí, které umožní zavedení principů a přístupů ambientní inteligence. Zde lze vhodně vycházet z již zmíněného scénáře Annette and Salomon [12].

Je přirozené, že různé studijní skupiny, ne-li každý jednotlivý student optimálně vyžaduje diferencovaný přístup ze strany pedagogů a administrativy. Tento přístup silně souvisí s kontextem různých situací jejich vzdělávacího procesu a mnohdy také s různými docela komplikovanými situacemi administrativní povahy.

Uveďme dva scénáře, volně převzaté z [16], které nám způsobem analogickým k [12] ozřejmí, o co v daném přístupu může jít.

- **Příklad 1**: Studentka Andrea byla přijata ke studiu a chce začít studovat v prvním ročníku. Nemá žádnou nebo téměř žádnou znalost studijních předpisů, pravidel, zvyků, zkrátka neví, jak to na univerzitě chodí. V prvních dnech pobytu na škole najednou čelí celé řadě pro ni obtížných problémů, spojených se zápisem do studia, s výběrem předmětů, přizpůsobením svého rozvrhu, seznámením se s pedagogy, budovami, místnostmi, knihovnou, apod. Pravděpodobně i částečná automatická znalostní či informační podpora v uvedeném směru by se jí zřejmě hodila. Skvělé by bylo, dostalo-li by se jí této podpory v kontextu s místem, kde se právě nachází a se situací či problémem, který právě řeší.

- **Příklad 2**: Docent Blahoslav vchází do své místnosti, inteligentní dveře se mu samozřejmě automaticky otevřou a aktivizuje se komplexní audiovizuální systém v jeho místnosti, který ho příjemným ženským hlasem uvítá a podá automaticky první informace o tom, co je Blahoslavův nejbližší program. Na obrazovce jeho počítače se objeví menu pokrývající potřebné informační či znalostní zdroje související s Blahoslavovým programem pro daný den. Objeví se informace či znalosti relevantní Blahoslavovu profilu, jeho odborným zájmům a situacím, které jej zanedlouho čekají. V průběhu dne se v kontextu s časem, plánem činnosti a probíhající situací celá informační a znalostní podpora Blahoslava od jeho inteligentní pracovny průběžně mění.

Samozřejmě, od scénářů, podobných uvedeným, je ke skutečné inteligenci univerzitního prostředí ještě docela dlouhá cesta. Technologicky je nicméně většina z naznačených scénářů docela reálná, větším problémem je obsahová stránka relevantní znalostní či informační podpory. I zde existuje celá řada postupů, metod a

nástrojů, ale jako každá změna, i tato záleží samozřejmě i na změně celkové kultury organizace (v tomto případě univerzity), aby aplikace relevantních nástrojů či přístupů byla pracovníky i studenty univerzity přijímána a akceptována. A samozřejmě, aby se na ní sami aktivně podíleli.

4.2 Veřejná správa

Jinou zajímavou oblastí, kde může vize ambientní inteligence nalézt dobrý prostor pro aplikace je oblast veřejné či státní správy. Především, perspektiva všudypřítomných počítačů a z toho plynoucí možnosti pro aplikaci postupů a technologií ambientní inteligence nám dává širší perspektivu pro inteligentní a přitom automatickou podporu méně zkušených či vůbec ze styků s orgány veřejné správy nezkušených občanů. Tato perspektiva může mít značný dopad na spokojenost s úrovní služeb zákazníkům, zejména v případě potřeby vybavit najednou několik složitých požadavků. Navíc může ovlivnit i složitost různých administrativních procesů v rámci příslušného orgánu pro výkon veřejné správy. To vše však vyžaduje takové změny odpovídajícího prostředí orgánu veřejné správy, které umožní zavedení principů a přístupů ambientní inteligence.

Je přirozené, že různé typy občanů v kombinaci s různorodostí jejich požadavků optimálně vyžadují diferencovaný přístup ze strany úředníků veřejné správy. Tento přístup silně souvisí s kontextem různých situací procesu naplňování jejich požadavků a mnohdy také s různými docela komplikovanými situacemi administrativní povahy.

Uveďme dva scénáře, volně pojednané dle [16], které nám přiblíží možnosti daného přístupu:

- **Příklad 1**: Občan Cyril bydlící v rodinném domku, potřebuje vypořádat s několika příbuznými svůj majetek. Nemá žádnou nebo téměř žádnou znalost souvisejících předpisů, pravidel, zvyků, více méně ani neví, co vlastně potřebuje, co má od orgánu veřejné správy požadovat a hlavně jak to žádat. Při své (možná v celém životě) první návštěvě takovéhoto úřadu najednou čelí celé řadě pro něj obtížných a zatím neřešených problémů. Pravděpodobně i částečná automatická znalostní či informační podpora ve směru vysvětlení, co požadovat a návodů, jako to požadovat, by se Cyrilovi zřejmě velice hodila. Inteligentní prostředí, do kterého vstoupí, jej bude schopno automaticky identifikovat (na základě čipu v jeho průkazu totožnosti), nalézt v příslušných databázích veškeré uchovávané údaje týkající se Cyrila a jeho majetku a zdvořile komunikovat o tom, co Cyrila na úřad přivádí. Mezitím se Cyril pohodlně usadí do křesla před rozměrnou obrazovku umístěnou na protilehlé zdi, přičemž mu bude nabídnuto občerstvení. Avatar (tedy uměle generovaná bytost v mnohém připomínající člověka, ale pouze zobrazovaná na obrazovce) v podobě příjemné úřednice mu prostřednictvím zmíněné obrazovku bude k dispozici jako průvodce vším, co Cyril potřebuje vybavit, nebo i celou řadu dokumentů vybaví samostatně v Cyrilově zastoupení (vzpomeňme na scénář Dimitrios and Digital-Me). Po postupném zjištění potřebných úkonů inteligentní prostředí vše provede, vytiskne příslušné dokumenty a zajistí příslušné změny v záznamech, týkajících se Cyrila a jeho jmění. Samozřejmostí je diskrétnost všech operací a naprostá ochrana a bezpečnost údajů.
- **Příklad 2**: Pracovnice orgánu veřejné správy Daniela vchází do své příjemně zařízené místnosti, inteligentní dveře se jí samozřejmě automaticky

(po identifikaci oprávnění Daniely tuto místnost používat) otevřou a aktivizuje se komplexní audiovizuální systém v její místnosti, který ji mladým a zvučným mužským hlasem uvítá a podá automaticky první informace o tom, co je Danielin nejbližší program. Na obrazovce jejího počítače se objeví menu pokrývající potřebné informační či znalostní zdroje související s Danieliným programem pro daný den. Objeví se informace či znalosti relevantní jejímu profilu, jejích odborným zájmům a situacím, které ji zanedlouho čekají. V průběhu dne se v kontextu s časem, plánem činnosti a probíhající situací celá informační a znalostní podpora Daniely od její inteligentní pracovny průběžně mění. Systém podporuje tvorbu Danieliných rozhodnutí a v případě potřeby nějaké informace či znalosti tuto vyhledá a promptně dodá.

Výše popsané aplikace v prostředích pro výkon veřejné správy jsou samozřejmě zatím fikcí. Technologicky jsou ale tyto scénáře docela reálné, větším problémem je obsahová stránka relevantní znalostní či informační podpory. I zde existuje celá řada využitelných postupů, metod a nástrojů, mezi něž patří jednak znalostní technologie (management znalostí, znalostní systémy) a jednak různé možnosti, které nabízí ontologické inženýrství. V souvislosti s tím lze odkázat na prameny, které uvádějí bližší popis některých v této oblasti využitelných přístupů, např. [4], [26], nebo [27]. Samozřejmě, jako každá změna, i tato záleží samozřejmě i na změně celkové kultury orgánů veřejné správy, aby aplikace relevantních nástrojů či přístupů byla pracovníky i návštěvníky míst výkonu veřejné správy přijímána a akceptována.

5 Některé přednosti a rizika

Existuje celá řada dalších zajímavých aplikačních možností i hotových aplikací postupů a technologií spjatých s představou ambientní inteligence. Na druhé straně však, pomineme-li nesporné přednosti inteligentního prostředí, pomáhajících např. fyzicky znevýhodněným spoluobčanům řešit běžné i obtížnější problémy, které pro ně běžné životní prostředí jejich bydliště či dopravních prostředků nebo třeba úřadů představuje, je potřeba upozornit na možná nebezpečí, která tato technologie potenciálně v sobě skrývá. Stručně některé z nich rozeberme.

Je celkem evidentní, že k tomu, aby bylo inteligentní prostředí schopno generovat nějakou činnost v reakci na potřeby uživatele v něm se nacházejícího, musí poměrně důkladně monitorovat jeho činnost, analyzovat jeho projevy a předpovědět jeho úmysly. Po odhadnutí úmyslu uživatele by inteligentní entity, které všudypřítomnou inteligenci daného prostředí vytvářejí, měly provést akci či posloupnost akcí, pomáhajících uživateli řešit problém, kterému čelí či usnadnit nějakou jeho činnost. Jak upozorňujeme již dříve na jiném místě [18], toto předpokládá, že uživatel je neustále těmito entitami sledován, údaje o jeho činnosti se ukládají z pohledu uživatele neznámo kam a mohou být využity celkem vzato neznámo kým pro různé účely, které nemusí být z hlediska uživatele vždy bez rizika. Toto monitorování lze samozřejmě podmínit předchozím souhlasem uživatele, jak tomu bývá při používání různých programových a jiných systémů, předpokládajících předchozí identifikaci uživatele a získání projevu jeho vůle, že chce, aby systém s ním (a on se systémem) spolupracoval. Přesto zde stále hrozí ztráta soukromí, k jejímuž ošetření zatím chybí jakýkoliv legislativní rámec.

Dalším problémem, v jistém smyslu vyplývajícím z předchozí úvahy je potenciální potřeba řešit několik protichůdných cílů najednou. K tomu si stačí uvědomit, že je-li v prostředí s ambientní inteligencí jeden uživatel, řešící nějaký problém, prostředí by mu mělo být schopno pomoci celkem přímočaře. Vstoupí-li do prostředí jiný uživatel a sleduje řešení svých cílů, pak si musí architektura prostředí poradit s řešením možných konfliktů v cílech obou uživatelů. Prostředí bude stát před otázkou rozhodnutí o prioritách cílů těchto uživatelů. Jsou-li tyto cíle protichůdné, ba až diametrálně odlišné, řešení konfliktu může být velice obtížné, ne-li zatím nemožné. Samozřejmě problémy tohoto typu jsou velikou výzvou pro výzkum, přičemž některé první a docela slibné výsledky už existují. Každopádně však upřednostnění cílu jednoho uživatele před cílem druhého může představovat pro tohoto druhého docela problém, ne-li riziko.

Okruh potenciálních problémů, které může ambientní inteligence přinést, můžeme závěrem rozšířit o několik dnes už celkem známých a publikovaných příkladů. Jak už jsme uvedli, součástí ambientní inteligence je i všudypřítomná komunikace entit, tvořících příslušné inteligentní prostředí. Předpokládáme-li, že tyto entity jsou např. v dětských hračkách, např. v panenkách, není nikterak nemožná představa panenky komunikující on-line s obchodem, ve kterém si objednává třeba nové šatičky, aniž by byl uživatel o tom předem informován. Může proběhnout i on-line platba na vrub jeho on-line ovládaného účtu, z čehož posléze uživatel zjistí, co se přihodilo a jak byly utraceny jeho peníze. Toto potenciálně může přerůst do nechtěných příkladů vzorců konzumního chování pro děti, majitelky těchto hraček. Je-li kupování stále nových šatiček akceptovatelné u panenky, proč by nebylo i u příslušné majitelky?

Celkem známým příkladem je představa, často používaná pro ilustraci některých přínosů všudypřítomné inteligence, spočívající v tom, že inteligentní lednička nebo spíž by sama pečovala o doplňování svého obsahu tím, že při zaznamenání úbytku nějakých zásob by automaticky komunikovala s příslušným dodavatelem a toto zboží doobjednala. Majiteli by pouze přišel účet. Zde vzniká prostor dokonce pro projevy konzumerismu, které by už byly mimo lidské společenství: mohlo by se totiž stát, že by konzumerismem byla stižena sama inteligentní lednice! Zatím to snad není možné, leda snad na základě nějaké poruchy, inteligentní entity (mimo lidí) nemají zatím vyvinutý pocit osobního štěstí, ale výzkum v oblasti emocí u umělých entit je rovněž docela intenzivní, takže se lze v budoucnosti možná nadít různých překvapení.

To vše, ač to možná pro někoho zní absurdně, je dnes již možné, ba dokonce existují prototypy inteligentních domů, které jsou uvedené komunikace a inteligentního jednání schopné.

6 Závěr

Závěrem kapitoly o ambientní inteligenci, jejím významu ale i rizikách tedy shrňme: ambientní inteligence je v současnosti předmětem velmi intenzivního převážně aplikačního výzkumu. Může být velikým pomocníkem, ale zároveň generuje celou řadu zcela nových a doposud nepředstavitelných problémů. S nadsázkou lze říci, že klasickou představu jakéhosi „Velkého bratra“, sledujícího veškerou činnost konkrétního člověka či celé skupiny lidí lze v prostředí se všudypřítomnou inteligencí

nahradit představou o celé „skupině velkých bratrů“, schopných se vzájemně domluvit a koordinovaně jednat. Je vhodné si toto uvědomovat a při výzkumu všudypřítomné inteligence na to neustále pamatovat. Je to technologie vymyšlená k tomu, aby lidem pomáhala a ne škodila, je proto nutné s ní přiměřeně zacházet a vhodně ji aplikovat.

Poděkování: Příspěvek souvisí s probíhajícím výzkumem financovaným z grantu GAČR č. P403-10-1310 „SMEW - *Smart Environments at Workplaces*“.

Literatura

[1] Abowd, G.D. et al: The Human Experience. *IEEE Pervasive Computing*, **1**(2002) 48-57.

[2] Aghajan, H., Augusto, J.C., Delgado, R.L.: *Human-Centric Interfaces for Ambient Intelligence*. Academic Press, 2009, ISBN 978-0123747082

[3] Augusto, J.C., Shapiro, D. (eds): *Advances in Ambient Intelligence*. IOS Press, 2007, ISBN 9781586038007

[4] Bureš, V., Ponce, D.: Knowledge-Based Support of Newcomers Integration into an Organization. In *ITiCSE 05, Proc. of the 10th Annual ACM SIGCSE Conf. On Innovation and Technology in Computer Science Education*, ACM, New York, 2005, p.372.

[5] Cai, Y., Abascal, J. (eds.): *Ambient Intelligence in Everyday Life*, Springer, 2006, ISBN 9783540377856

[6] Cook, D., Das, S.: *Smart Environments: Technology, Protocols and Applications*. Wiley – Interscience, 2004, ISBN 978-0471544487

[7] *FIPA Personal Travel Assistance Specification*, available on-line http://www.fipa.org/specs/fipa00080/index.html.

[8] Gal, C.L. et al.: Smart Office: An Intelligent Interactive Environment. *IEEE Intelligent Systems,* **4**(2001).

[9] Greenfield, A.: Everyware: The Dawning Age of Ubiquitous Computing. New Riders Publishing, 2006, ISBN 9780321384010

[10] Grill, T., Khalil, I., Kotsis, G.: Agents for ambient Intelligence – Support or Nuisance. *ÖGAI Journal* **23/1**, 2004.

[11] Hu, J., Feijs, J.: An Agent-based Architecture for Distributed Interfaces and Timed Media in a Storytelling Application. In *Proc. ACM AAMAS 2003*, ACM, New York, 2003, pp.1012-1013.

[12] ISTAG: *Scenarios for Ambient Intelligence in 2010.* Luxembourg: European Commission, 2001.

[13] Krumm, J.: *Ubiquitous Computing Fundamentals*. Chapman & Hall/CRC, 2009, ISBN 978-1420093605

[14] Mangina, E., Carbo, J., Molina, J.M.: *Agent-based Ubiquitous Computing*. Atlantis Press, 2009, ISBN 978-9078677109

[15] Mikulecký, P.: Remarks on Ubiquitous Intelligent Supportive Spaces. In *Recent Advances in Applied Mathematics and Computational and Information Sciences, Proc. of the 15th American Conference on Applied Mathematics and Proc. of the International Conference on Comp. and Information Sciences* (Eds. Jegdic, K., Simeonov, P., Zafiris, V.), Vols. I and II, WSEAS Press, Athens, 2009, pp.523-528.

[16] Mikulecký, P., Olševičová, K.: University Education as an Ambient Intelligence Scenario. In *ECEL 2005, 4rd European Conference on e-Learning* (Ed Dan Remenyi), ACL, Reading, England, 2005. pp. 275-280.

[17] Mikulecký, P., Olševičová, K.: Inteligentní prostředí jako multiagentový systém. In *Kognice a umělý život VII* (Kelemen, J., Kvasnička, V., Pospíchal, J., eds.), Slezská univerzita v Opavě, 2007, pp. 245-248.

[18] Mikulecký, P., Olševičová, K., Ponce, D.: Ambient Intelligence - Monitoring and Supervision of New Type. In *Monitoring, Supervision and Information Technology* (eds. P. Kleve, R.V. De Mulder, C. van Noortwijk), 2nd ed., LEFIS Series 1, Prensas Universitarias de Zaragoza, 2007, pp. 115-134.

[19] Misker, J.M.V., Veenman, C.J., Rothkrantz, L.J.M.: Groups of Collaborating Users and Agents in Ambient Intelligent Environments. In *Proc. ACM AAMAS 2004*, ACM, New York, 2004, pp. 1318-1319.

[20] Monekosso, D., Remagnino, P., Kuno, Y.: *Intelligent Environments: Methods, Algorithms and Applications*. Springer, 2008, ISBN 978-1848003453

[21] Mordini, E., De Hert, P.: *Ageing and Invisibility*. IOS Press, 2010, ISBN 978-1607506140

[22] Nakashima, H., Aghajan, H., Augusto, J.C.: *Handbook of Ambient Intelligence and Smart Environments*. Springer, 2009, ISBN 978-0387938073

[23] Nijholt, A., Rist, T., Tuijnenbreijer, K.: Lost in Ambient Intelligence? In *Proc. Of ACM CHI 2004 Workshop,* ACM, New York, 2004.

[24] O´Hare, G.M.P., O´Grady, M.J., Collier, R., Keegan, S., O´Kane, D., Tynan, R., Marsh, D.: Ambient Intelligence Through Agile Agents. In *Cai, Y.: Ambient Intelligence for Scientific Discovery*, LNAI 3345, Springer Verlag, 2005, pp. 286-310.

[25] Obermair, Ch., Ploderer, B., Reitberger, W., Tscheligi, M.: Cues in the Environment: A Design Principle for Ambient Intelligence. In: *Proc. of CHI 2006*, Montreal, ACM, New York, 2006, pp. 1157-1162.

[26] Olševičová, K., Čech, P.: Resources recommendation system based on ontologies. In *Sborník konference Kognice*, Univerzita Hradec Králové, 2005, pp. 79-90.

[27] Olševičová, K., Mikulecký, P.: Learning Management System as Ambient Intelligence Playground. In *Proc. of the IADIS International Conference Web*

Based Communities 2006 (Ed. by Piet Kommers, Pedro Isaías and Ambrosio Goikoetxea), IADIS Press, 2006, pp. 12-18.

[28] Ramdane-Cherif, A., Benarif, S., Levy, N.: Agent Technologies For Pervasive Computing and Communications. In *Proceedings of the Third IEEE International Conference on Cognitive Informatics (ICCI'04),* IEEE Press, New York, 2004.

[29] Russell, S., Norvig, P.: *Artificial Intelligence: A Modern Approach*. 2nd ed., Prentice-Hall, 2002.

[30] Sashima, A., Izumi, N., Kurumatani, K.: CONSORTS: A Multiagent Architecture for Service Coordination in Ubiquitous Computing. In *Proc. Multi-agent for Mass User Support - MAMUS 2003 (K. Kurumatani et al., eds.),* LNCS 3012, Springer Verlag, 2004, pp. 190-216.

[31] Spanoudakis, N., Moraitis, P.: Agent-based Architecture in an Ambient Intelligence Context. In *Proc. 4th European Workshop on Multi-Agent Systems (EUMAS'06)*, Lisbon, Portugal, 2006.

[32] Stathis, K., Toni, F.: Ambient Intelligence Using KGP Agents. In *EUSAI 2004 (P. Markopoulos et al, eds.)*, LNCS 3295, Springer Verlag, 2004, pp. 351-362.

[33] Weber, M. et al.: *Advanced Intelligent Environments*. Springer, 2009, ISBN 978-0387764849

[34] Weiser, M.: The Computer for the 21st Century. *Scientific American 265* (1991) 94-104.

[35] Wooldridge, M.: *Introduction to MultiAgent Systems*. New York: J. Wiley & Sons, 2002.

O zložitosti merania zložitosti

Jiří POSPÍCHAL[1]

Abstrakt. Kapitola sa zaoberá rôznymi definíciami zložitosti (complexity) a prístupmi k jej meraniu. Zložitosť je neľahko definovateľný pojem vo všeobecnosti, existujú desiatky definícií v rôznych vedných odboroch, ako v architektúre, biológii, informatike, dynamických systémoch vo fyzike, inžinierstve, finančníctve, teórii hier apod. Prístupy k zložitosti klasifikujeme na základe troch aspektov: Ako obťažné je jav alebo systém opísať, ako obťažné je ho vytvoriť, aký je stupeň jeho organizácie.

1 Úvod

Ako súvisí zložitosť s umelou inteligenciou a kognitívnou vedou? Kognitívna veda sa snaží skúmať, ako človek rozmýšľa a jeho mozog môžeme určite nazvať zložitým. Umelá inteligencia sa snaží zaistiť, aby počítač dokázal rozmýšľať rovnako dobre alebo aj lepšie ako človek a najväčší problém má so zvládaním zložitých systémov vonkajšieho sveta.

Väčšina systémov v prírode aj v spoločnosti, ktoré sa snažíme skúmať, simulovať a predpovedať ich správanie, nie sú celkom usporiadané ani absolútne chaotické a náhodné, ale niečo medzi. Sú charakterizované určitým stupňom organizácie, štruktúry, pamäti, pravidelností, symetrie a rozpoznateľných opakujúcich sa vzorov. Automaticky pritom predpokladáme, že napríklad ľudský mozog je zložitejší ako mozog nižších živočíchov, alebo ako počítač, rovnako ako predpokladáme, že genetická informácia človeka je zložitejšia ako genetická informácia améby a že v priebehu evolúcie vznikajú stále zložitejšie organizmy. Sú tieto predpoklady ale pravdivé? A ako sú zložité počítačové modely vonkajšieho sveta v porovnaní so skutočným svetom?

Na tieto otázky začali ľudia hľadať odpoveď už pred desiatkami rokov. Žiaľ, výsledky zatiaľ nie sú veľmi povzbudzujúce. Aj keď teoretických výdobytkov vo forme definícií a metód, ako zložitosť merať, je v rôznych oblastiach desiatky, iba minimum z nich je prakticky použiteľných, a tie ktoré sú použiteľné, sú špecifické pre úzko špecializovanú problematiku. Asi najznámejšie informatikom, najlepšie definované a prakticky vypočítateľné definície zložitosti sú v oblasti zložitosti

[1] Fakulta informatiky a informačných technológií STU, 842 16 Bratislava, e-mail: pospichal@fiit.stuba.sk

algoritmov. Vzhľadom na to, že ide o veľmi špecifickú oblasť, ktorá je do hĺbky preberaná v štandardných učebniciach teoretickej informatiky [18], v tejto kapitole sa zameriame na ostatné mierky zložitosti. S akými problémami sa teda pri meraní zložitosti stretávame?

Ako prvé je asi treba si vyjasniť naše chápanie pojmu zložitý, ktoré tu chápeme ako preklad anglického prívlastku complex. Problémom je, že prívlastok zložitý sa používa ako preklad tak pre prívlastok complex ako aj pre prívlastok complicated. Snáď najjednoduchšie vysvetlíme ich rozdiel uvedením príkladov. Mechanické hodinky sú zložitý strojček v zmysle complicated, sú zložené z jednotlivých súčastí, ktorých interakcie sú ale jednoznačne dané a aj výsledok interakcií je dobre predpovedateľný – hodinky budú ukazovať čas a pohyb jednotlivých ich súčiastok je dobre predpovedateľný. Oproti tomu zložitý systém v zmysle ďalej používaného významu complex môže byť napríklad systém tvorby počasia alebo telekomunikačný systém, kde síce jednotlivé interakcie sú predikovateľné (snáď až na subatomárne javy u atómov, z ktorých sa koniec-koncov atmosféra skladá), ale množstvo súčastí systému je také, že sa výsledok všetkých ich interakcií nedá jednoducho modelovať alebo predpovedať. Zložitosti väčšinou napomáha, že interakcie neexistujú medzi všetkými zložkami systému, a interakcie na lokálnej úrovni sú často rozšírené o interakcie „ďalekého dosahu“. Podobne, interakcie môžu byť rôzne silné. Taký systém má množstvo „rovnovážnych“ stavov, ktorých počet exponenciálne narastá s množstvom súčastí systému.

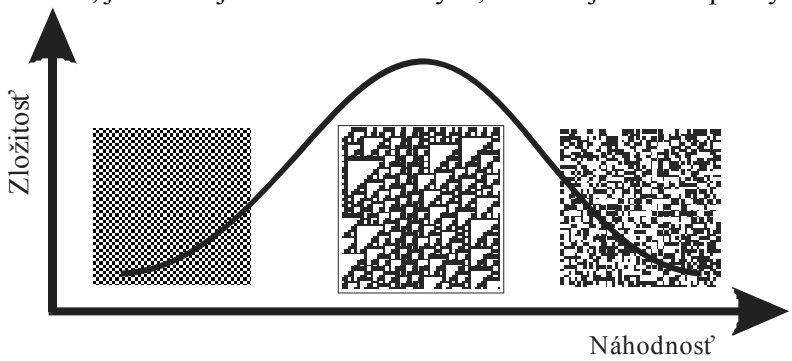
Pokiaľ si necháme vyhľadať knihy, ktoré majú v názve „Measures of Complexity“, dostaneme až tri [1, 9, 16], publikované od konca osemdesiatych po polovicu deväťdesiatych rokov minulého storočia. Z toho je vidno, že téma merania zložitosti je už dlho skúmaná, zozačiatku najmä vo fyzike a informatike, ale neskôr aj v ostatných odboroch. Napriek tomu ale neexistujú všeobecne prijímané spôsoby merania zložitosti. Jedným zo základných dôvodov je snáď to, že vlastne nemáme presne zadefinované, čo to je tá zložitosť, čo vlastne chceme merať. Lepšie povedané, presných definícií existuje veľa, a stále ich pribúda, sú rôzne v rôznych vedných disciplínach a existujú aj rozdielne definície aj v rámci jednotlivých vedných disciplín. Čo nám chýba, je zhoda, ktorý z prístupov k zložitosti je ten pravý.

Navyše, niektoré definície sú veľmi špecifické a jednoznačne aplikovateľné, napríklad pre zložitosť počítačových algoritmov, či genómov v biológii, iné sú zasa poňaté natoľko filozoficky z nadhľadu, že sú prakticky neaplikovateľné. Už pred 12 rokmi Bruce Edmonds obhájil PhD. prácu, v ktorej porovnával rôzne definície zložitosti a ich nedostatky [5]. Prístupný popis zložitosti a jej merania podala v poslednom čase aj Melanie Mitchell vo svojej knihe Complexity [15].

Pôvodne latinské slovo complexus znamenalo okrem iného prepletený, prepojený. To naznačuje, že pre zložité systémy je podstatná interakcia jednotlivých rozlíšiteľných častí. Klasickým prístupom skúmania vo vede je simplifikácia, redukcionizmus, zjednodušenie skúmaného javu, problému či systému na základné súčasti, ktoré potom môžeme skúmať oddelene a predmet skúmania vysvetliť jednoduchou kompozíciou týchto častí. Tento prístup ale u pravej „zložitosti“ nefunguje, u komplexných javov je interakcia rovnako významná alebo aj významnejšia ako vlastnosti jednotlivých rozlíšiteľných súčastí predmetu skúmania a táto interakcia má nelineárne vlastnosti,

teda nie je ľahko extrapolovateľná a správanie systému nie je jednoducho predpovedateľné ani nie je systém ľahko riaditeľný. Pre zložité systémy (v zmysle „complexity“) platí, že celok je viac ako suma častí, čo sa prejavuje v novo definovaných disciplínach ako veda o chaose, evolučná ekonomika či teória sietí.

Rozlíšiteľné súčasti, ktoré sa môžu správať rôzne, vedú k neporiadku, náhodnému šumu. Perfektná interakcia súčastí, ich prepojenie vedie k nastoleniu poriadku, symetrie, ako u atómov uhlíku usporiadaných do perfektného kryštálu diamantu. Zložitosť v zmysle komplexnosti môže existovať iba keď tendencia k neporiadku a interakcia vedúca k poriadku sú v rovnováhe, stavy prvkov systému nie sú ani štatisticky náhodné, ani deterministicky určené. Zložitý systém je „na hranici chaosu“ a pravidelnosti (pozri Obr. 1). Závislosť štruktúrnej zložitosti na náhodnosti, tak ako je ukázané na Obr. 1, je ale iba jedna z veľa možných, neexistuje obecne platný vzťah [6].



Obr. 1. Krivka miery zložitosti v závislosti na stupni náhodnosti, postupne pravidelne usporiadaný systém má malú náhodnosť a malú zložitosť, fraktálový systém je na prvý pohľad viac náhodný a viac zložitý (aspoň keď nepoznáme pravidlá jeho generovania) a celkom náhodný systém považujeme zasa za málo zložitý (voľne podľa [10]).

Takáto zložitosť je ale často závislá na škále pozorovania. Keď si zoberieme zopár molekúl zmrznutej vody, zložitosť nepozorujeme, a keď sa na sneh pozeráme z diaľky, vidíme iba rovnomernú bielu. Iba pod lupou môžeme pozorovať zložité tvary vločiek, v zásade na seba podobných, ale v podrobnostiach rozdielnych. Ten istý princíp môžeme nájsť v zložitosti okrem priestorovej škály v dynamických zložitých systémoch aj na časovej škále.

Fyzik Seth Lloyd v dvojstránkovom článku z r. 2001 [13] zadeľuje spôsoby, čo vlastne meriame, keď meriame zložitosť, do troch kategórií:

a) Ako zložité je to popísať?
b) Ako zložité je to vytvoriť?
c) Aký stupeň organizácie to má?

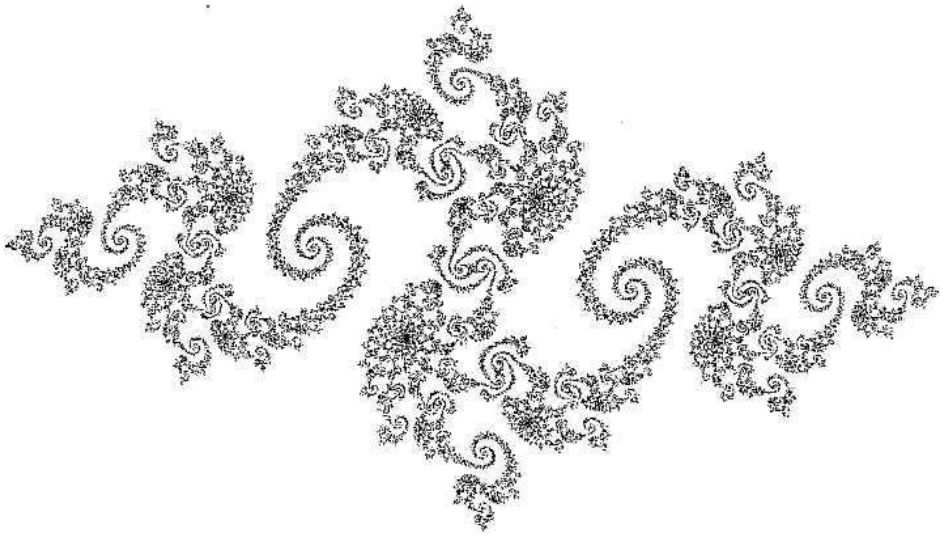
V jeho zozname 42 spôsobov merania zložitosti sa zišli koncepty z oblasti výpočtov a teórie informácií, dynamických systémov a termodynamiky (samozrejme, od tej doby priebežne pribúdajú ďalšie mierky zložitosti).

Zložitosť opisu obsahovala mierky obsahu informácie, entropie, algoritmickej zložitosti, minimálnej dĺžky popisu, a ďalšie špeciálne mierky (Fisherova a Chernoffova informácia, Renyiho entropia, dĺžky rôznych spôsobov zakódovania, fraktálová dimenzia, Lempel-Ziv zložitosť). Zložitosť vytvorenia zahŕňa časovú,

pamäťovú a informačnú výpočtovú zložitosť, logickú hĺbku, termodynamickú hĺbku, cenu a „crypticity“. Zložitosť organizačnej štruktúry potom zahŕňa efektívnu zložitosť, hierarchickú zložitosť, gramatickú zložitosť a ďalšie, podobne ako množstvo zdieľanej informácie medzi súčasťami systému sa dá merať v jednotkách informácie, algoritmickej informácie, kapacite prenosových kanálov, stupňov korelácie, atď. Vybrané mierky zložitosti si teraz priblížime.

2 Zložitosť opisu

Keď začneme s najjednoduchším (aspoň na prvý pohľad) problémom obsahu informácie, v jeho priamočiarej aplikácii sa prejavia závažné problémy. Prvá, klasická mierka zložitosti bola zavedená Andrejom **Kolmogorovom** [11], a zhruba odpovedala dĺžke najkratšieho programu potrebného na generovanie (binárneho) reťazca, ktorého zložitosť meriame (momentálne sa taká mierka volá algoritmická informácia). Predpokladáme pritom, že akýkoľvek subjekt merania je prevediteľný na binárny reťazec. Aj keď takýto prístup môže byť prirodzený pre informatikov, napríklad problém zmerania zložitosti ľudského mozgu, ktorý vyžaduje zakódovanie takejto štruktúry do formy binárneho reťazca, môže byť zložitejší ako samotné meranie zložitosti takéhoto reťazca.



Obr. 2. Fraktálový obraz Juliovej množiny je generovaný iteračnou rovnicou s použitím komplexnej premennej. Vzhľadom na jednoduchú definíciu má prakticky nulovú zložitosť meranú ako informačný obsah, ale veľkú Bennettovu „logickú hĺbku“, lebo jeho vygenerovanie zaberie veľa krokov.

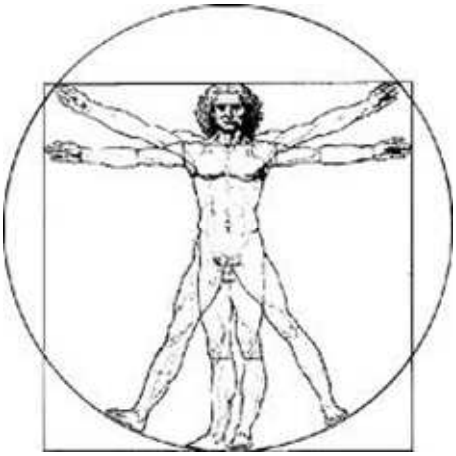
Keď preskočíme obťaž zakódovania viacrozmernej dynamicky sa meniacej štruktúry do reťazca, aj meranie zložitosti reťazca je vo všeobecnosti nevypočítateľný problém. To súvisí v počítačovej vede s problémom zastavenia, ktorý je samotný zahmlenou formou Goedelovho teorému. Dôkaz nevypočítateľnosti si priblížime pomocou klasického Berryho paradoxu: Nech n je najmenšie kladné celé číslo, ktoré

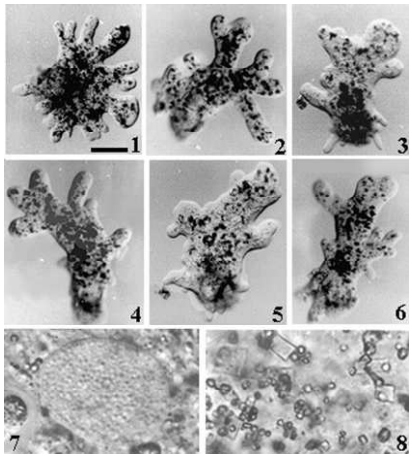
nemôže byť definované menej ako dvadsiatimi slovami. Problém je, že predchádzajúca veta má menej ako 20 slov a my sme ňou číslo definovali, veta teda popiera samu seba. Podobným prístup môžeme zvoliť u Kolmogorovovej zložitosti. Zoberieme si program, ktorý generuje všetky možné reťazce, meria ich Kolmogorovovu zložitosť a prvý reťazec s dostatočnou zadanou zložitosťou n vráti. Problém je, že keď zoberieme n väčšie ako odpovedá dĺžke programu, tak máme paradox. Malým programom sme vygenerovali reťazec, ktorý sa nedá vygenerovať tak malým programom.

Navyše nám Kolmogorovova zložitosť vyjde maximálna u náhodne generovaného reťazca. Okrem toho, aj jednoduchý program môže počítať dlho alebo krátko, a to by sa na mierke tiež malo odraziť. Inak má napríklad na prvý pohľad veľmi zložitý fraktálový obrazec prakticky nulovú zložitosť meranú ako informačný obsah. Vygenerovanie takého obrázku fraktálu môže zabrať veľa výpočtového času, ale prakticky je generovaný jednoduchou matematickou formulou, pozri Obr. 2.

Ďalšou mierkou obsahu informácie môže byť **Shannonova entropia** ako priemerný informačný obsah alebo „mierka prekvapenia“, s ktorou pozorujeme ďalšie prvky reťazca. Nech systém produkuje časovú sériu N rozdielnych symbolov s_i a p_i je pravdepodobnosť výskytu symbolu s_i v reťazci. Entropia tohto reťazca je definovaná ako $H = -\sum_i p_i \log p_i$. Reťazec zložený zo stále sa opakujúcich prvkov by mal nulovú entropiu, ale náhodne generovaný reťazec (alebo zašumená obrazovka) by mal maximálnu entropiu. (Záleží ešte samozrejme na vybranej dĺžke reťazca. Reťazec zložený z prvej poloviny z núl a druhej poloviny z jednotiek má nulovú entropiu v porovnaní s reťazcom, kde sú tieto hodnoty náhodne prehádzané.) Ani táto mierka nám teda veľmi nevyhovuje.

Existujú aj ďalšie špecifické mierky zložitosti, ako napr. **fraktálová dimenzia** [15], ktoré ale majú iba úzke využitie.





Obr. 3. Da Vinciho Vitruvian Man ako ukážka Homo Sapiens a améba Polychaos dubium [2], ktorá má genóm obsahujúci 670 miliárd bázových párov, cez 200 krát viac ako človek.

Povedzme si ale, že sa teoretickými problémami nebudeme zaoberať a skúsime zmerať niečo praktické, ako je zložitosť genómu, tak, že budeme merať obsah informácie jednoducho počtom bitov potrebných na zakódovanie. Keby sme nevedeli nič o informácii, povedzme zakódovanej reťazcom bitov, tak to nerobí problém. Lenže, keď si zoberieme dĺžku genetického kódu vyjadreného počtom bázových párov, ktorý sa dá na bity jednoducho previesť, tak jeden z typov améb obsahuje dvestokrát viac párov ako človek (pozri Obr. 3). Samozrejme, keďže sa nám chce veriť, že človek je zložitejší ako améba, niečo v tejto mierke neodpovedá našej predstave.

Problém je asi v tom, že iba časť kódu slúži priamo na vytváranie proteínov (u človeka iba asi dve percentá), ďalšie časti slúžia ako pomocné oddeľovače alebo sú to jednoducho nadbytočné nepoužívané nezmysly. Zistiť však s istotou, čo je a čo nie je podstatné a ktorá časť by sa dala bez problémov vypustiť alebo nahradiť úspornejším kódom je pri počte stoviek miliárd bázových párov nereálne. Aj keď je pravda, že väčšina kódu na vytváranie proteínov sa dá vyselektovať, takáto úloha by bola stále nereálna, nech už pre genóm améby alebo človeka.

3 Zložitosť vytvorenia

Aj kvôli nerovnomernosti medzi prakticky nulovou Kolmogorovovou zložitosťou fraktálov a medzi naším inštinktívnym chápaním fraktálov ako zložitých obrazcov boli navrhnuté ďalšie mierky. Pre informatikov je snáď najznámejšia výpočtová zložitosť (vzhľadom na čas alebo pamäťový priestor), ktorej sa ale pre jej štandardný výskyt vo všetkých základných učebniciach teoretických základov informatiky nebudeme venovať tu. Trocha menej notoricky známa je **Bennettova „logická hĺbka“** [3] definovaná ako dĺžka behu najkratšieho programu generujúceho nami želaný reťazec (presnejšie povedané, počet krokov Turingovho stroja s najmenším počtom stavov a pravidiel, ktorý vygeneruje taký reťazec a zastaví sa). Alternatívne definície do nutných výpočtových prostriedkov započítavajú aj pamäť.

Aj tu ale narazíme. Zoberme si reťazec celkom náhodne vygenerovaný. Na to, aby sme dostali ten istý reťazec, budeme potrebovať program typu Vytlač „011010010. daný náhodný reťazec“. Takýto program bude dlhší ako samotný náhodný reťazec, teda náhodný reťazec je maximálne zložitý aj podľa Kolmogorovovej mierky, ako aj podľa logickej hĺbky, či podľa miery entropie. Nášmu intuitívnemu chápaniu sa ale prieči povedať, že náhodne zašumenú obrazovku televízie by sme mali ohodnotiť ako zložitejšiu v porovnaní s obrazovkou zobrazujúcou čokoľvek zmysluplné.

V osemdesiatych rokoch Seth Lloyd a Heinz Pagels navrhli **termodynamickú hĺbku** [12], ako najprijateľnejší vedecky určený reťazec udalostí vedúcich k vzniku daného systému, alebo celkové množstvo termodynamických a informačných zdrojov vedúcich k fyzickému vzniku procesu. Pretože tvorba človeka trvala oveľa dlhšie, ako tvorba améby, asi by mala byť zložitejšia. Pokiaľ sa ale ostatné mierky aspoň tvárili, že by sa snáď dali spočítať, u tejto mierky ani nemá zmysel o tom uvažovať. Kto vie povedať reťazec udalostí (a čo berieme ako udalosť, každý pohyb atómu, alebo mutáciu bázy?), ktorý produkuje najjednoduchšiu živú bunku, nie to človeka?

4 Zložitosť organizácie

Fyzik Murray Gell Man [7] navrhol ako mierku „**effective complexity**“, u ktorej predpokladá, že každý meraný systém (jav, predmet) je kombináciou tak pravidelnosti, tak náhodnosti. Dĺžka popisu tých pravidelností je to čo hľadáme. Keď pravidelnosti môžeme popísať rovnako dobre rôznymi prístupmi, hľadáme potom ten najkratší, používame Occamovu britvu. Treba ešte upozorniť na názvom sa podobajúcu effective measure complexity definovanú Grassbergerom [8] ako množstvo informácie v danej časti sekvencie potrebné na predikciu ďalšieho symbolu.

Čo potrebujeme u „effective complexity“, je povedať si, čo je náhodné a čo je výsledkom pravidelnej štruktúry. Program, ktorý potom dostaneme, nemusí generovať reťazec presne, môže generovať iba pravidelné štruktúry a náhodné bity s danými pravdepodobnosťami rozdelenia a koreláciami. Na to ale potrebujeme nejakú superinteligentnú bytosť, ktorá by nám povedala, čo je a čo nie je náhodné a ktoré tie pravidelné štruktúry vlastne hľadáme. Keď sa napríklad pozrieme na prvý pohľad náhodný reťazec bitov, nemusíme v ňom rozpoznať, že je to vlastne iba zakódovaná odmocnina z dvoch. Taký reťazec má význam a dá sa presne generovať na rozdiel od celkom náhodného reťazca.

Podobný problém je **stupeň hierarchie**, navrhnutý ako mierka zložitosti Herbertom Simonom v r. 1962 [17], kde je zložitý systém definovaný ako taký, ktorý sa dá rozložiť na podsystémy, tie na ich podsystémy, atď. Napríklad telo je rozložiteľné na orgány, tie na bunky, tie na organely a iné systémy v bunkách, atď. Trocha sa to podobá sebapodobnosti fraktálov. Dá sa takto ukazovať v biológii, že vznikajú v priebehu evolúcie stále zložitejšie organizmy, od baktérií, cez eukariotické bunky, mnohobunečné organizmy, až po kolónie organizmov ako u hmyzu [14]. Pre informatikov tu ale chýba objektivita v určení, čo je to podsystém, alebo aj ako je definovaná úroveň.

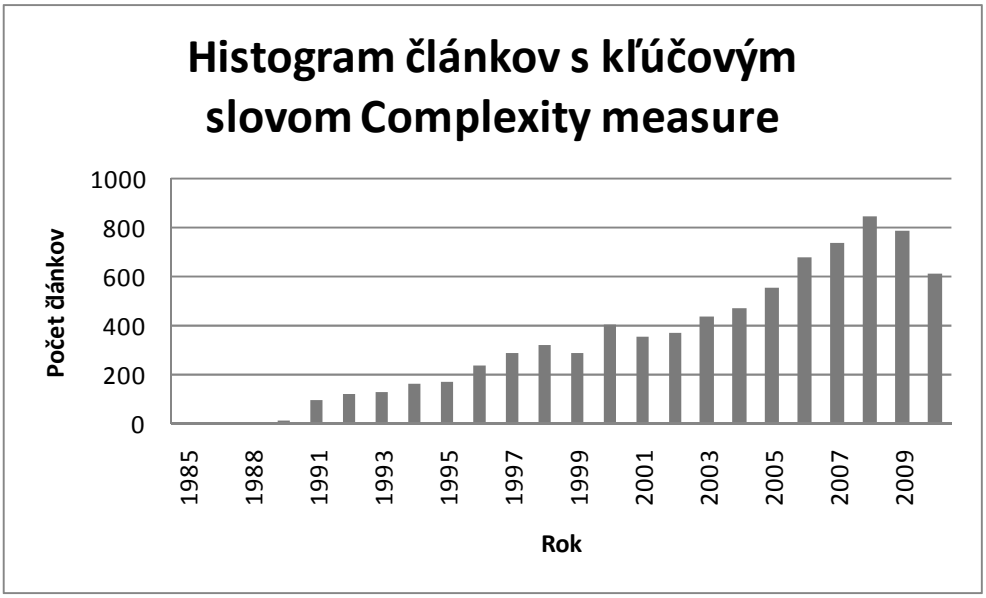
Keď žiadna z doterajších mierok zložitosti nebola prakticky použiteľná, neostáva nám nič iné ako sa obrátiť na štandardný spôsob spracovania veľkého množstva informácií, a to na štatistiku. Fyzici Jim Crutchfield a Karl Young navrhli mierku **štatistickú zložitosť** (statistical complexity) [4], ktorá je nulová v prípade našich dokonalých informácií o systéme (alebo keď je systém perfektne usporiadaný), ale je tiež nulová, pri našej totálnej neznalosti systému, alebo jeho absolútnej náhodnosti. Prvý prípad je plne predpovedateľný, druhý má jednoduchý štatistický popis. Čo hľadáme, je mierka čiastočnej usporiadanosti medzi týmito extrémami. V podstate berieme systém ako kontinuálny zdroj generujúci reťazec informácií a hľadáme minimálne množstvo informácií o minulom správaní sa systému, ktoré nám stačí na optimálnu predpoveď štatistického správania sa v budúcnosti. Chceme mať model, ktorý sa bude správať zo štatistického hľadiska nerozpoznateľne od vlastného systému. Pretože systémy sú rôzne a rôzni ľudia dávajú rôznu váhu vybraným typom usporiadanosti, ani štatistická zložitosť nie je iba v jednej verzii, ale existuje ich veľa. Rôzne verzie štatistickej zložitosti sú špeciálne užitočné, pokiaľ má systém zložitú vnútornú dynamiku, o ktorej ale nič nevieme.

5 Záver

Súdiac podľa toho, ako často sa zložitosť objavuje v biologických systémoch, alebo aj v systémoch obsahujúcich logicky separovateľné, často rôznorodé časti spolu s ich hlbokým poprepájaním, emergencia zložitosti v takých systémoch môže byť nevyhnutným javom.

V biológii môže byť prirodzeným výsledkom evolučného tlaku na robustnosť systémov, ktoré sú v metastabilnej rovnováhe napriek mohutnému výskytu náhodných javov.

V priebehu posledných dvadsiatich rokov bola dôležitosť definovania zložitosti a jej vhodnej mierky rozpoznaná, čo je vidno aj na počte článkov, obsahujúcich ako kľúčové slová „complexity measure“ podľa bibliografickej citačnej databáze Web Of Science [19], pozri Obr. 4. Aj keď módna vlna štúdia zložitosti už odznela, táto oblasť stále ešte iniciuje veľa zaujímavých smerov výskumu.



Obr. 4. Histogram počtu článkov s kľúčovým slovom complexity measure za posledných 25 rokov

Poďakovanie: Táto kapitola vznikla za podpory grantovej agentúry VEGA v rámci grantových úloh VEGA 1/0141/10.

Literatúra

[1] Abraham, N. B., Albano, A. M., Passamante, A., and Rapp, P. E. (eds): *Measures of Complexity and Chaos* (NATO Science Series B: Physics). Springer 1990.

[2] Amoebae on the Web, Institute of Cytology RAS, http://amoeba.ifmo.ru/species/amoebidae/polydub.htm, 7.2.2011

[3] Bennett, C. H.: How to define complexity in physics, and why. In W. H. Zurek (editor): *Complexity, Entropy, and the Physics of Information*. Reading, MA: Addison-Wesley, 1990, p. 142.

[4] Crutchfield, J. P. and Young, K.: Inferring statistical complexity. *Physics Review Letters* **63** (1989) 105–108.

[5] Edmonds, B.: *Syntactic Measures of Complexity*. Doctoral Thesis, University of Manchester, Manchester, UK, 1999, http://cfpm.org/~bruce/thesis/

[6] Feldman, D. P. and Crutchfield, J.P.: Statistical Measures of Complexity: Why? *Physics Letters A* **238** (4-5) (1998) 244-252.

[7] Gell-Mann, M.: What is complexity? *Complexity* **1** (1) (1995) 16–19.

[8] Grassberger, P: Information and Complexity Measures in Dynamical Systems. In Atmanspacher, H., Scheingraber, H. (eds.): *Information Dynamics*. Plenum Press, New York, 1991, pp. 15-33

[9] Henderson-Sellers, B.: *Object-Oriented Metrics: Measures of Complexity*. Prentice Hall, 1995.

[10] Huberman, B. and Hogg, T.: Complexity and Adaptation. *Physica* **22d** (1986) 376-384.

[11] Li, M. and Vitanyi, P.: An Introduction to Kolmogorov Complexity and Its Applications. New York: Springer-Verlag, 1997.

[12] Lloyd, S. and Pagels, H.: Complexity as thermodynamic depth. *Annals of Physics* **188** (1988) 186–213.

[13] Lloyd, S.: Measures of complexity: a nonexhaustive list. *Control Systems Magazine*, *IEEE* **21**(4) (*2001*) 7 – 8.

[14] McShea, D. W.: The hierarchical structure of organisms: A scale and documentation of a trend in the maximum. *Paleobiology* **27** (2) (2001) 405–423.

[15] Mitchell, M.: Chapter 7: Defining and Measuring Complexity. V knihe *Complexity A Guided Tour*. Oxford University Press, 2009.

[16] Peliti, L., Vulpiani, A. (eds.): *Measures of Complexity*. Lecture Notes in Physics, Springer, 1988.

[17] Simon, H. A.: The architecture of complexity. *Proceedings of the American Philosophical Society* **106** (96) (1962) 467–482.

[18] Sipser, M.: Introduction to the Theory of Computation. Course Technology, MIT, 2nd ed., 2005.

[19] Web of Science, http://apps.isiknowledge.com, visited 31.1.2011

Modely a metódy skúmania kognitívneho vývinu

Ján RYBÁR[1]

Abstrakt. V tejto kapitole sa zaoberáme dvomi dnes už klasickými modelmi kognitívneho vývinu (ich filozofiou, ale aj ich konkrétnymi metódami výskumu s nimi spojenými). Prvým je takzvaný konštruktivistický model (ktorého kľúčové princípy sformuloval J. Piaget) a druhým je tzv. nativistický model (ktorého základnú platformu sformuloval N. Chomsky). Silnou stránkou obidvoch modelov je, že dokázali stimulovať rozsiahly teoretický a experimentálny výskum.

1 Úvod

Podstatou Piagetovho konštruktivistického modelu sú tri rozhodujúce faktory kognitívneho vývinu: vonkajšie učenie, dedičnosť a proces vnútornej ekvilibrácie. O čo tu ide, možno krátko priblížiť na jeho kritike behaviorizmu[2]. Piaget odmieta interpretáciu správania (konania) ako nejakej mechanickej reakcie na podnety zvonku. Podľa neho na začiatku nie sú len podnety, ale aj citlivosť na podnety, jednoducho schopnosť organizmu dať odpoveď. Teda u Piageta je nevyhnutným komponentom aktívne chápanie poznávajúceho subjektu. Podľa neho poznanie by ani nemohlo začať, keby dieťa už pri narodení nedisponovalo určitými kognitívnymi nástrojmi (asimilačnými schémami). Za takéto východiskové asimilačné schémy považoval základné reflexy (sací, uchopovací atď.). K tomto procesu asimilácie je komplementárny proces akomodácie, čiže zmeny týchto schém pod vplyvom

[1] Centrum kognitívnych vied, KAI, Fakulta matematiky, fyziky a informatiky, Univerzita Komenského, Mlynská dolina, 842 48 Bratislava. e-mail: rybar@fmph.uniba.sk

[2] Behaviorizmus bol dominantnou paradigmou v psychológii prvej polovice dvadsiateho storočia. Jeho základné tézy sformuloval J. Watson, hlavná z nich hovorí o tom, že vedecká psychológia môže skúmať len stimuly, ktoré pôsobia na organizmus a jeho nasledovné reakcie (tzv. schéma S-R). To, čo je medzi tým, je čierna skrinka, ktorá nie je prístupná vedeckému skúmaniu. Preto predmetom vedeckej psychológie môže byť len správanie (behaviour).

prostredia. Poznanie je potom produktom, konštrukciou, ktorá vzniká zrovnovážňovaním týchto dvoch procesov.[3]

N. Chomsky pri formulovaní svojho modelu tiež vyšiel z kritiky behaviorizmu, presnejšie behavioristickej interpretácie jazyka, keď odmietol tézu, že jazyk si osvojujeme len imitovaním (pričom dôležitú úlohu tu hrá posilňovanie odmenou a trestom). Týmto spôsobom sa nijako nedá vysvetliť, prečo sa dieťa tak rýchlo a ľahko naučí niečo tak zložité ako je jazyk, dokonca bez nejakej osobitnej priamej inštrukcie. Chomsky to formuloval ako argument nedostatočnosti lingvistického vstupu (poverty of stimulus argument). Prostredie tu ani zďaleka nehrá takú rolu, akú mu často pripisujeme. Základnou tézou Chomského modelu je, že prostredie len spúšťa mechanizmy, ktoré sú vrodene zabudované (sú už v defaulte). Tým obrátil pozornosť na skúmanie vrodených faktorov. To bola veľká výzva pre odborníkov z mnohých relevantných disciplín (predovšetkým psychológov a neurovedcov), preskúmať čím poznávajúci subjekt defaultovo disponuje.

2 Konštruktivistický model kognitívneho vývinu

2.1 Fylogenéza a ontogenéza poznania

Za jeden z najefektívnejších spôsobov výkladu konštruktivistického modelu považujeme analýzu jeho formovania u mladého Piageta (neskôr svoj model už len precizoval a uplatňoval na ďalšie a ďalšie oblasti).[4]

Od ranej mladosti dôležitou témou pre neho bolo porovnávanie vývinu poznania vo fylogenéze a v ontogenéze. Odkiaľ sa vzala táto téma u Piageta a prečo práve ona zohráva takú významnú úlohu? Odpovedať na túto otázku nie je až také ťažké. Ak sa pozrieme na prvé stránky jeho vedeckej biografie zistíme, že sa mu dostalo excelentného vedeckého školenia v biologických vedách. Bol uznávaným expertom v oblasti alpských mäkkýšov (odborné štúdie publikoval už na gymnáziu).

Jeho pozornosť, pokiaľ ide o teóriu vývinu, sa však neobmedzovala len na biologické vedy. Paralelne s tým sa mladý Piaget veľmi intenzívne zaujímal aj o vývin vedeckých ideí v dejinách. Jeho záujem o túto oblasť bol vzbudený a rozvíjaný takými osobnosťami, ako bol H. Bergson (jeho práce mali na neho významný vplyv), A. Reymond (študoval u neho v Neuchâteli) a L. Brunsvicg (študoval u neho počas svojho parížskeho pobytu).

Ďalším rozhodujúcim faktorom bolo jeho pôsobenie v Binetovom psychologickom laboratóriu v Paríži (1919-1921). Tu odštartoval skúmanie formovania základných pojmov v psychogenéze dieťaťa, ktoré ho vlastne najviac preslávilo.[5]

---

[3] Piaget sám svoju teóriu vývinu nazýval teóriou zrovnovážňovania (ekvilibrácie) a tento faktor vývinu považoval za najdôležitejší [10].

[4] Jean Piaget (1896-1980) študoval biológiu na univerzite vo švajčiarskom Neuchâteli. Doktorát o alpských mäkkýšoch tu obhájil v r. 1918. Neskôr pôsobil hlavne na Ženevskej univerzite.

[5] K práci v tomto laboratóriu sa dostal viac menej náhodou, keď popri štúdiu dejín vedy hľadal ešte nejakú prácu, v ktorej by sa mohol aj experimentálne uplatniť a bola mu ponúknutá práca práve v Binetovom laboratóriu.

Napriek tomu zaujímavé je, že hlavnou Piagetovou výskumnou oblasťou bolo porovnávanie vývinu poznania v dvoch líniách (v dejinách vedeckého myslenia a v psychogenéze dieťaťa). To vyvolalo celý rad nedorozumení. Niektorí autori to odmietajú na prvý pohľad. Nemožno predsa porovnávať elementárne a najzložitejšie úrovne. Ide o porovnávanie neporovnateľného. Je v tom obsiahnutý veľký dešpekt voči tomu, že by skúmanie vývinu poznania u dieťaťa mohlo niečo priniesť pre pochopenie vývinu tak sofistikovanej úrovne, ako je vývin vedeckého poznania. Ignoruje sa tu vlastne to, čo Piageta najviac motivovalo, a to, že deti sú ideálne živé laboratórium, prostredníctvom ktorého môžeme skúmať základné mechanizmy vývinu poznania ako takého.[6]

Iné nedorozumenie je v tom, že toto hľadanie analógií v týchto dvoch líniách (dejiny vedeckého myslenia a psychogenéza dieťaťa) sa berie v zmysle Haecklovej teórie rekapitulácie: ontogenéza je opakovaním fylogenézy. Je pravda, že Haeckel a jeho stúpenci sa opierali o určité fakty, vychádzali z poznania, že ľudské embryo v istom štádiu svojho vývinu má niečo z nižších živočíchov, napríklad žiabre a podobne. No bolo by iste absurdné hľadať obsahovú zhodu napríklad medzi pojmami newtonovskej fyziky a spontánnymi fyzikálnymi pojmami u dieťaťa. Od takej zhody sa Piaget samozrejme dištancuje. Nepostrehnutie tohto rozdielu je jedným z typických znakov povrchného zoznámenia sa s koncepciou jeho modelu.

Piagetovou hlavnou hypotézou bolo, že medzi vývinom poznania v psychogenéze a v dejinách vedeckého myslenia existuje zhoda nie obsahová, ale v mechanizmoch vývinu poznania. Nemáme tu možnosť analyzovať všetky ním sformulované mechanizmy, preto sa obmedzíme pre ilustráciu len na jeden, a to na mechanizmus štádií ***intra***, ***inter*** a ***trans***. Ilustráciu uvedieme z Piagetovho skúmania spontánnych geometrických pojmov, konkrétne vertikálnych a horizontálnych línií.

Metóda skúmania je veľmi jednoduchá, dieťa má nakresliť domček s komínom (alebo skopírovať predkreslený domček s komínom). Pre ***intra*** štádium je charakteristické, že dieťa má problémy s korektným nákresom horizontálnych a vertikálnych línií. Koncentruje sa len na línie vo vnútri obrázku. Obrázok chápe ako izolovaný objekt, chýbajú referencie na vonkajšok. Preto kreslia deti (do 5 – 6 rokov) komín na šikmej streche (dokonca aj pri jeho priamom kopírovaní) kolmo na ňu. Dajú sa uviesť aj iné príklady: dieťa kreslí hladinu vody v ležiacej fľaši paralelne s dnom (niekedy až do 8 – 9 rokov) a podobne. V ***inter*** štádiu dieťa už berie do úvahy, že obrázok je v nejakom prostredí, teda prihliada už aj na línie mimo obrázku a správne kreslí komín kolmo na spodný okraj výkresu alebo hranu stola. Štádium ***trans***, presnejšie, transfigurálne vzťahy sa nedajú znázorniť (transcendujú, presahujú empirickú skúsenosť). O čo ide, ilustruje napríklad experiment so slimákom pohybujúcim sa na pohyblivej doske. Otázka pre dieťa znie: „Čo sa musí urobiť, aby slimák ostal na tom istom mieste?“ (na úrovni určitého fixného bodu). Tu sa musia brať do úvahy dva referenčné systémy: pohybujúci sa slimák a pohybujúca sa doska. Tieto dva pohyby sa nedajú naraz sledovať (dokonca ani vtedy, keď majú ten istý

[6] Až od Piagetových čias sa malé dieťa začína považovať za veľmi zaujímavý a dôležitý objekt kognitívneho výskumu ([4], [8]).

smer), nedá sa to nakresliť v jednej figúre (s výnimkou animovaného obrázka), a teda sa to musí vypočítať[7].

Podľa Piageta tento mechanizmus ***intra***, ***inter*** a ***trans*** má univerzálny charakter a dá sa aplikovať aj na vývin vedeckého myslenia v dejinách, napríklad na vývin geometrie. Charakteristiky ***intra*** fázy má euklidovská geometria, ***inter*** projektívna geometria a ***trans*** neeuklidovské geometrie [9].

2.2 Formovanie fundamentálnych pojmov v psychogenéze

Piaget sa okrem skúmania vývinu (formovania) spontánnych[8] geometrických pojmov zaoberal aj formovaním fyzikálnych a algebraických pojmov. V tejto súvislosti sa zaoberal skúmaním formovania takých fundamentálnych pojmov, ako je čas, priestor, nevyhnutnosť, náhoda, kvantita (číslo) atď.[9]

Pozrime sa teraz na niektoré ďalšie výsledky, ktoré Piaget dosiahol aplikáciou svojho modelu pri skúmaní vývinu fundamentálnych pojmov. Piagetov výskum začína od skúmania raných štádií vývinu dieťaťa. Na ilustráciu môžeme uviesť príklad zo senzomotorického štádia (t.j. obdobia od 0 do 2 rokov).[10] Ide o skúmanie chápania kauzality u štvor-päťmesačného dieťaťa. Nad postieľkou sú zavesené hrkálky, z ktorých visí šnúrka. Dieťa sa ju naučí ťahať a zabáva sa na tom. (Alebo tam šnúrka ani nemusí byť. Dieťa príde na to, že keď sa natriasa, hýbe postieľkou, hrkálky vydávajú zvuk.) K zaujímavému javu (presnejšie činnosti) dochádza, keď premiestnime tieto hrkálky povedzme dva metre od postieľky. Dieťa zase hľadá šnúrku (alebo sa hádže v postieľke) dúfajúc, že hrkálky vydajú zvuk. Filozofickou terminológiou by sme teda skutočne mohli povedať, že dieťa je činnostným solipsistom. Príčinu všetkého „vidí“ vo vlastnej činnosti, to, že príčinou pohybu môže byť aj iný predmet, je záležitosť neskoršieho vývinu [10].

Formovanie fundamentálnych pojmov vrcholí v tzv. operačných štádiách kognitívneho vývinu. Piaget v tejto súvislosti hovorí o pojmoch zachovania. Ide o najznámejšiu časť jeho diela. Aj tieto experimenty sú veľmi jednoduché, ľahko realizovateľné a zopakovateľné. Medzi najznámejšie patria experimenty, ktoré ukazujú formovanie pojmu objemu (resp. zachovanie kontinuitného množstva látky), dĺžky, váhy a ďalšie. Na ilustráciu uvedieme niektoré z nich. Máme dva rovnaké poháre naplnené vodou. Dieťa odsúhlasí, že sú naplnené tým istým množstvom vody (obyčajne tak, že priťukne obidva poháre a porovná hladinu vody v obidvoch

---

[7] „Napríklad slimák A sa môže pohybovať v jednom smere pohybom +A alebo v inom –A a doska B pohybom +B a –B: z toho vyplývajú štyri kombinácie +A +B, +A, -B atď., ktoré sa delia na dvanásť podľa toho, či A = B, A > B alebo A < B” ([5]).

[8] T.j. takých pojmov, ktoré sa sformujú aj bez didaktického pôsobenia (to znamená aj u detí, o ktoré sa z nejakých príčin nikto špeciálne nestaral, resp. nikdy nechodili do školy).

[9] Tento výskum tvorí podstatnú časť jeho diela. Podobnosť s Kantovou schémou základných pojmov (kategórií) tu nie je náhodná. Piaget za hlavný nedostatok Kanta považoval práve skutočnosť, že nič nehovorí o formovaní týchto pojmov. Jeden z hlavných bodov jeho výskumného programu bolo práve doplniť Kanta v tomto smere.

[10] Piaget rozlišuje v kognitívnom vývine dieťa štyri veľké štádia: senzomotorické štádium, predoperačné štádium, štádium konkrétnych operácií a štádium formálnych operácií.

pohároch). Potom prelejeme vodu z obyčajného pohára do úzkeho pohára na šampanské a pýtame sa, či sa zmenilo teraz množstvo vody? Dieťa (4 – 5 ročné) si myslí, že áno. Môžu sa vyskytnúť dokonca dva varianty. Buď hovorí, že vody je v ňom viac, lebo siaha vyššie, inokedy zase, že menej, lebo druhý pohár je užší. Alebo uveďme ešte iný variant, ak guľôčku z plastelíny preformujeme do valčeka, dieťa hovorí, že množstvo plastelíny sa zmenilo, je ho teraz viac, pretože valček je dlhší. V tomto veku dieťa zjavne ešte nemá pojem zachovania. Dosiahne ho až okolo 6 – 7 rokov, keď tvrdí, že množstvo vody (resp. plastelíny) sa nemení.

Efektné sú aj experimenty týkajúce sa formovania zachovania dĺžky. Pred dieťaťom je nakreslený oblúčik, jeho úlohou je nakresliť na ten istý papier oblúčik (drôtik) ako narovnaný, alebo naopak, ak je tam rovná čiara, nakresliť ako bude vyzerať, ak by sme ju ohli do oblúčika. V prvom štádiu (4 – 5 rokov) dieťa rieši úlohu tak, že dodržiava ekvivalenciu krajných bodov čiar, tzv. efekt hranice. V tejto fáze dieťa ešte nemá pojem zachovania dĺžky. O utvorení pojmu zachovania dĺžky môžeme hovoriť, až keď dieťa prekoná tento efekt hranice (obyčajne vo veku nad 6 rokov). V inom variante tohto experimentu pracujeme s dvomi rovnakými pravítkami. Ak jedno posunieme trochu doprava (alebo doľava), dieťa v prvom štádiu bude tvrdiť, že sa zmenila ich veľkosť (je schopné posúdiť vždy len jednu stranu pravítiek). To je charakteristické pre predoperačné štádium. Úlohu vyrieši až vtedy, keď je schopné brať do úvahy obidve strany pravítiek (v Piagetovej terminológii, keď tu prebehne kompenzačná operácia). Také dieťa už má pojem zachovania dĺžky a je to dôkaz, že sa už nachádza v štádiu konkrétnych operácií.[11]

2.3 Metóda habituácie a dishabituácie

Tento Piagetov výskum mal obrovský ohlas a je stimulujúci dodnes, hoci medzičasom boli vytvorené nové metódy skúmania, pomocou ktorých sa dosiahli aj diametrálne odlišné výsledky. Jednou z takých oblastí je skúmanie ranej kognície dieťaťa.[12] Zaujímavé diskusie prebehli o tom ako si malé dieťa formuje predstavy o skrytých objektoch (ako si konštruuje „pojem“ stáleho predmetu). Piaget svojho času tvrdil, že dieťa mladšie, než osem mesiacov, je radikálnym solipsistom. Svet pre neho je svetom ustavične objavujúcich sa a miznúcich predmetov. O čo ide si môžeme objasniť na veľmi jednoduchom experimente. Ak pred, povedzme šesťmesačné dieťa, dáme nejaký atraktívny predmet (hodinky, kľúče apod.), dieťa okamžite pôjde po ňom. Ak však predmet zakryjeme nejakou prikrývkou, dieťa neurobí nič (nestrhne prikrývku), akoby pre neho prestal predmet existovať. No, ak prikryjeme kľúče transparentnou

[11] Podobne v predoperačnom štádiu ešte nie je schopné pochopiť napríklad operáciu tranzitivity (chápe iba, že jedna bábika je menšia alebo väčšia než druhá). V štádiu konkrétnych operácií už chápe, že napríklad bábika B môže byť väčšia než bábika A a zároveň menšia než bábika C. V štádiu formálnych operácií zvládne túto operáciu už bez toho, aby muselo pritom vidieť konkrétne objekty.

[12] Ako sme už vyššie naznačili Piaget tomuto obdobiu venoval veľmi veľkú pozornosť a boli to práve knihy venované tomu obdobiu, ktoré mali veľký ohlas napríklad “La construction du réel chez l´enfant” (Konštrukcia reality u dieťaťa) a “Naissance de l´intelligence” (Zrod inteligencie).

prikrývkou, dieťa nebude mať žiadny problém. Ako to vysvetliť? Jedna možnosť je, že si to nepamätá, ale tomu protirečia iné výskumy, z ktorých vyplýva, že dieťa v tomto veku si dobre pamätá informácie staré aj niekoľko týždňov. Druhá, pravdepodobnejšia možnosť je, že uchopovanie a percepcia predmetu nie je ešte dostatočne koordinovaná na neurálnej úrovni.

V deviatich mesiacoch už predmet nájde, ale problém stále ešte nie je vyriešený.[13] Môžeme sa o tom presvedčiť nasledujúcim experimentom. Ak napríklad nejakú hračku (plyšového medvedíka) schováme pod vankúš, dieťa ho už hravo nájde, ak však ho schováme pod prikrývku, dochádza k pozoruhodnému javu. Dieťa najprv hľadá medvedíka pod vankúšom. Ako by tu rozhodujúcou nebola percepcia, ale predchádzajúca úspešná činnosť. Tento fenomén Piaget nazýva chybou A nonB. Po dosiahnutí prvého roku života sa tento fenomén vytráca.

S tým, že stály predmet sa formuje u dieťaťa až tak neskoro sa však nevedeli zmieriť niektorí Piagetovi nasledovníci. Podľa nich základná Piagetova chyba spočívala v tom, že svoje výskumy orientoval až na deti, ktoré dokázali manipulovať s predmetmi. A to je podľa nich už dosť neskoro. Títo psychológovia vychádzali zo skutočnosti, že hoci malé dieťa nerozpráva, má pomerne obmedzené pohybové aktivity, napriek tomu je takmer od narodenia vynikajúcim pozorovateľom. Uvedomovali si však, že ak chcú Piagetove výsledky vyvrátiť, musia zmeniť výskumnú technológiu. Použili metódu merania času pozornosti dieťaťa, tzv. metódu habituácie a dishabituácie. Pri tejto metóde sa meria čas zamerania alebo odvrátenia pozornosti dieťaťa na určitú udalosť. Inak sa pracuje s podobnými experimentálnymi scénami, ako to bolo u Piageta. Dieťa, keď pozerá na vláčik, ktorý prejde za zástenu (prenosná stena) očakáva, že na druhej strane z poza nej vyjde. Na tom nie je nič nezvyčajného. To je štandardná situácia (možná udalosť). Potom však zdvihneme zástenu a pred očami dieťaťa položíme na koľajnice prekážku (veľký kváder). Následne zase spustíme zástenu a pustíme za ňu vláčik. Medzi tým však trikovo, aby dieťa nevidelo, prekážku odstránime a dieťa vidí vláčik vychádzať spoza zásteny. Čas nameranej pozornosti je v tomto prípade významne vyšší, než v prvej situácii (dieťa je prekvapené). Svedčí to o tom, že má nejakú predstavu o prekážke (kvádri) za zástenou. Touto metódou boli testované už 3 – 4 mesačné deti (Baillargeon 1993). To ukazuje, že deti si formujú predstavu stáleho predmetu oveľa skôr, než sa domnieval Piaget.[14]

Avšak veci nie sú až také jednoduché a problém so zakrytým predmetom môže mať ešte aj pätnásťmesačné dieťa, ako ukazuje nasledujúci experiment. Vezmeme do ruky kľúče, ukážeme ich dieťaťu, dlaň zatvoríme, ruku vsunieme pod prikrývku a kľúče tam necháme. Potom ruku vytiahneme. Dieťa je prekvapené, že kľúče v nej nie sú, bude hľadať, či nepadli na zem, diviť sa, kde sa stratili, ale nenapadne ho pozrieť sa pod prikrývku. To znamená, že predstava existencie skrytého predmetu sa formuje veľmi dlho a komplikovane. Až v osemnástich mesiacoch je tento problém uspokojivo vyriešený.

---

[13] Pozoruhodné je, že mačky majú takýto „pojem“ stáleho predmetu už v treťom mesiaci života.

[14] Táto metóda je v súčasnosti najdôležitejšou metódou psychologického výskumu uplatňovanou pri skúmaní kognície veľmi malého dieťaťa. Používa sa na skúmanie rôznych fyzikálnych, numerických, dokonca sociálnych intuícií dieťaťa.

Hľadanie nových metód stimuloval iný teoretický model, ktorý predpokladal, že dieťa má oveľa viac vrodených kognitívnych predispozícií, než predpokladal J. Piaget. Analýza tohto modelu (ktorý sa zvykne označovať ako nativistický) a metód s ním spojených bude predmetom ďalšej podkapitoly.

3 Nativistický model kognitívneho vývinu

3.1 Vrodené faktory verzus učenie

Základné idey nativistického modelu kognitívneho vývinu sformuloval N. Chomsky. Jeho záujmy a to isté možno povedať aj o jeho vplyve, sa nikdy neohraničovali len na (na jeho pôvodný odbor) lingvistiku.[15] Patrí medzi autorov, ktorí nemajú žiadny problém s prekračovaním hraníc jednotlivých vedných odborov (v tom sa nápadne podobá J. Piagetovi). Ak cesta vedie za hranice vlastného odboru „prefrčia“ cez ňu, bez ohľadu na to, či sú tam väčšie alebo menšie bariéry. Lingvistika, informatika, psychológia, kognitívna veda, filozofia a ďalšie disciplíny, ktoré sa nejakým spôsobom dotýkajú skúmania jazyka, mentálnych procesov a mysle, to sú oblasti, kde „surfoval a surfuje“ N. Chomsky.

Vedeckú reputáciu si získal teóriou (klasifikáciou) formálnych jazykov, ktorá mala veľký ohlas aj v počítačovej vede (predovšetkým v oblasti kompilačných konštrukcií a teórii automatov). Pomerne rýchlo sa ukázalo, že Chomsky patrí medzi tých lingvistov, ktorých koncepcie menia pohľad na samotný predmet lingvistiky. Jeho práce spôsobili, že lingvisti sa začali zaujímať nielen o skúmanie výsledkov jazykovej schopnosti (ako to dominuje v štrukturálnej lingvistike), ale aj o skúmanie procesu utvárania jazykovej produkcie.

Chomsky je autorom generatívnej a transformačnej gramatiky, v rámci ktorej prišiel s mnohými novými pojmami a teóriami. Medzi takéto patrí teória X-bar, pomocou ktorej charakteristickými stromovými grafmi opisuje syntaktické štruktúrne pozície vetných konštituentov. Ďalej rozpracoval teóriu hĺbkovej a povrchovej štruktúry a pravidlá transformácie medzi nimi. Tiež uplatnil nový pohľad na povahu lexikónu (mentálneho slovníka), ktorý podľa neho obsahuje aj informácie o syntaktickej forme slov. Podal aj osobitú teóriu prideľovania pádov a tematických rolí a iné [3]. To všetko je vzťahované k chápaniu jazyka ako kognitívnej vrodenej štruktúry.

Možno to povedať aj tak, že lingvistické školy pred Chomským sa zameriavali na skúmanie jazykového správania (jazykových výpovedí). Čiže tým, čo sa v jeho terminológii nazýva jazykovou performanciou. Takéto chápanie jazyka však pokladal za príliš úzke. Jeho zaujímali iné otázky, ako to, že hovorca-poslucháč je schopný porozumieť vetám, ktoré predtým nikdy nepočul, a tvoriť nové vety, ktorým budú

[15] Noam Chomsky sa narodil v r. 1928 vo Philadelphii. Študoval lingvistiku u profesora Z. Harrisa na Pennsylvánskej univerzite v Pittsburghu. Doktorát získal na Harvardovej univerzite v r. 1955. Väčšinu svojej profesionálnej kariéry pôsobil na Massachusettskom technologickom inštitúte v Bostone.

rozumieť iní? Ako to, že hovorca-poslucháč (malé dieťa alebo človek, ktorý explicitne neovláda gramatiku) vie rozlíšiť správne utvorené vety od nesprávnych. Túto jazykovú schopnosť nazýva jazykovou kompetenciou a na túto oblasť (akými pravidlami sa riadi dieťa, čo mu umožňuje tak ľahko zvládnuť jazyk) je sústredený jeho záujem.

Podľa Chomského jazyk nemôžeme skúmať bez toho, aby sme neskúmali aj procesy jeho osvojovania. Toto osvojovanie jazyka je nevyhnutnou súčasťou lingvistického skúmania. Nepatrí však potom takýto výskum už skôr do psychologickej vedy? Pre Chomského to nie je žiaden problém, pretože podľa neho lingvistika je súčasťou psychológie. Nech už je ako chce, v každom prípade svojimi názormi prispel k dynamickému rozvoju disciplíny, ktorá sa nazýva psycholingvistika. Ako sme už v úvode uviedli, korene tohto prístupu boli sformulované v Chomského slávnej kritike behavioristickej interpretácie jazyka, v ktorej odmietol tézu, že jazyk si osvojujeme len imitovaním (posilňovaným odmenou a trestom).

Tu treba zdôrazniť, že Chomský má výhrady k bežnému chápaniu učenia ako základnému faktoru vývinu. Treba sa pýtať, ako sa jazyk učíme alebo ako sa jazyk vyvíja? Učíme sa jazyk ako etiketu (napríklad, ako sa máme správať pri stole) alebo nám jazyk v priebehu detstva rastie ako nos, ruky a podobne? Chomsky explicitne presadzoval koncepciu jazyka ako mentálneho orgánu.

V tom zmysle má celkom iný názor na úlohu prostredia v jazykovom vývine. Podľa neho jazykové prostredie je len spúšťač vrodene zabudovaných zariadení (LAD – language acquisition device). Imitovanie ani zďaleka nehrá takú dôležitú úlohu, ako sa zdá na prvý pohľad. Veľmi dôležitým argumentom v prospech LAD je, že dieťa sa učí jazyk ľahko a rýchlo dokonca i bez nejakej osobitnej priamej inštrukcie. Ako sme už vyššie spomenuli, Chomsky to formuloval ako argument nedostatočnosti lingvistického vstupu (poverty of the stimulus argument). Potvrdzujú to aj mnohé výskumy, z ktorých vyplýva, že dôsledným systematickým opravovaním chýb dieťaťa nedosiahneme nejakú výraznú akceleráciu v osvojovaní si jazyka dieťaťom [12]. Zistilo sa, že rodičia to systematicky ani veľmi nepraktizujú. Je to vlastne nemožné, pretože potom by museli opravovať dieťa stále (čo je rovnako neprijateľné pre obidve strany). Pre rodičov je oveľa dôležitejšie, či dieťa rozumie obsahu hovoreného a na ten sa aj zameriavajú. Ak rodičia predsa len skúsia chvíľu dieťa systematický opravovať, sú veľmi často sklamaní z nie veľmi úspešných dosiahnutých výsledkov (zvlášť vzhľadom na úsilie, ktoré pritom museli vynaložiť). Pokiaľ sa dieťa nedostane do príslušného vekového štádia, inštrukciu jednoducho ignoruje. To znamená, že jazykový vývin je do značnej miery spojený s procesmi dozrievania, ktoré sú biologicky determinované.

3.2 Vrodený gramatický softvér

Ale vráťme sa k otázke ako si Chomsky (tento radikálny zástanca vrodenosti syntaktických jazykových štruktúr) predstavuje fungovanie vrodených mechanizmov zúčastnených na tvorbe reči. Svoju koncepciu neustále spresňoval. Hovoril o vrodenosti univerzálnej gramatiky, o zariadení pre nadobúdanie jazyka (***language acquisition device***) a o vrodených jazykových princípoch a parametroch.

Zaujímavé spresnenie toho, čo je vrodené, rozpracoval Chomsky práve vo forme parametrickej teórie. Podľa neho máme množinu vrodených univerzálnych princípov a parametrov, rovnakých pre všetky indivíduá. Je to akýsi univerzálny softvérový balík, ktorý sa spúšťa (resp. nastavuje) pod vplyvom primárnych lingvistických dát (PLD) materinského jazyka. V skratke to môžem zapísať ako: LAD-parametre + PLD = idiolekt[16]. Týchto parametrov je potenciálne okolo sto. Ide len o to, čo sa použije z tohto softvérového balíka v danom konkrétnom jazyku. Nie v každom jazyku sa musia spustiť tie isté parametre. O tom, ktoré parametre sa nastavia, rozhoduje konkrétne jazykové prostredie ([6], [7]). Ak to funguje, priblížime na tzv. pro-drop parametri – absencii sémanticky prázdneho zámena v pozícii podmetu. Tento parameter je nastavený napríklad v taliančine, španielčine, ale aj v slovenčine. Môžeme povedať: ***Prší, Idem do kina, Nemám peniaze*** a podobne. Tento parameter však nie je nastavený vo všetkých jazykoch. Napríklad v angličtine musí byť prítomné prázdne zámeno v pozícii podmetu: ***It´s raining, I go to the cinema tonight, There is no money*** a podobne. Dôležitosť takýchto princípov a parametrov (vrodených kognitívnych štruktúr) spočíva v tom, že práve ich prostredníctvom sa dieťa učí jazyk tak ľahko a rýchlo.

3.3 Metóda umelých pojmov

Chomského model jazykového vývinu svojho času vzbudil veľký ohlas a našiel si mnohých stúpencov a nasledovníkov. Jeho idey stimulovali množstvo nových metód výskumu, ktorých predmetom je skúmanie jazyka. Medzi najpozoruhodnejšie patrí metóda používania umelých pojmov ([2]) a metóda dvojitej disociácie uplatnená v oblasti rečových porúch ([11]).

Metóda umelých pojmov resp. „ohýbania umelých slov“ ako prvá použila Berko-Gleasonová vo svojej doktorskej práci na Massachusettskom technologickom inštitúte ešte v päťdesiatych rokoch minulého storočia. Podstatu tejto metódy najlepšie priblížime, ak uvedieme základný variant tohto jednoduchého, ale dômyselného experimentu. Pomocou tejto metódy sa dajú testovať deti vo veku 4-7 rokov. Deťom ukážeme nakresleného vtáčika. Tento vtáčik sa nepodobá na nijakého z reálne existujúcich vtákov (je to tzv. artificial bird). Tohto nového umelého vtáčika nazveme wug. Dieťa konštatuje: „Je tam teda jeden wug.“ Potom k nemu nakreslíme ešte jedného takého istého vtáčika a dieťa bez problémov utvorí plurál tohto novotvaru: „Teraz sú tam dva wugy (resp. v angličtine wugs). Dieťa teda tvorí plurál slova, ktoré nikdy predtým nepočulo. Výhodou tejto metódy je istota, že deti sa s týmito slovami v rôznych gramatických pozíciách určite ešte nikdy nestretli. Táto metóda bola použitá nielen pri skúmaní uplatňovania pravidiel plurálu, ale aj minulého času, stupňovania prídavných mien a iných gramatických tvarov. V tejto súvislosti sa potom pracuje nielen s umelými podstatnými menami, ale podobne sa môže pracovať aj s umelými slovesami, prídavnými menami a s umelými príponami a predponami.[17]

---

[16] Idiolekt sú práve tie nastavené parametre, ktoré používame vo svojom jazyku.

[17] Tieto testy majú aj diagnostický význam. Pomocou nich je možné včas zachytiť agramatikov, ktorí neprejdú týmito testami a účinne im pomôcť.

Metóda umelých jazykových pojmov sa veľmi rozšírila a inšpirovala aj ďalších psycholingvistov. Nadväzuje na ňu aj koncepcia D. Slobina. Tento psycholingvista a jeho tím skúmali viac ako štyridsať európskych, ázijských, afrických, amerických jazykov, vrátane jazykov prírodných národov. Získané výsledky výskumu sú veľmi zaujímavé. Čo majú vlastne všetky jazyky spoločné? Zistilo sa, že niektoré princípy sa uplatňujú vo všetkých jazykoch. Medzi také patria napríklad tieto princípy: Venuj pozornosť slovosledu! Vyhýbaj sa výnimkám! Venuj pozornosť koncovkám slov! Na ilustráciu si všimnime aká „technológia" výskumu bola použitá pri skúmaní posledného princípu a aké konkrétne výsledky priniesol tento výskum. Opíšme priebeh experimentu trochu bližšie: 1. Deťom boli ukázané obrázky dvoch nových umelých živočíchov (ktoré podobne ako spomínaný vtáčik ***wug*** neexistujú v prírode). Deti sa naučili ich mená ***wug***[18] a ***flip***. 2. Naučili sa nové slovesá. Sloveso ***pum*** znamená postaviť zviera vertikálne, sloveso ***bem*** znamená postaviť zviera horizontálne. 3. Ak činnosti týchto živočíchov boli pozorované jedným pozorovateľom, sloveso dostalo príponu ***-abo*** (teda vytvorili sa slová ako ***pumabo***, ***bemabo***), ak činnosti týchto živočíchov boli pozorované viacerými pozorovateľmi, sloveso dostalo predponu ***aki-*** (teda vytvorili sa slovesa ***akipum***, ***akibem***). Výsledok výskumu bol pozoruhodný: zistilo sa, že deti sú vnímavejšie na koncovky a nové slová s novými koncovkami sa dokázali naučiť oveľa rýchlejšie, než slová s predponami. Z toho D. Slobin usúdil, že tento princíp patrí medzi vrodené strategické princípy.

3.4 Hypotéza o dvoch kognitívnych systémoch (metóda dvojitej disociácie)

Pozoruhodnou hypotézou o vrodených predpokladoch jazykovej schopnosti dieťa inšpirovanou Chomského modelom je aj Pinkerova hypotéza o dvoch jazykových kognitívnych systémoch, z ktorých jeden má na starosti lexikón a druhý tvorbu pravidiel. Pinker svoju hypotézu overuje najmä na pravidelných a nepravidelných slovesách v angličtine. Pre nepravidelné slovesá je dôležitý pamäťový kognitívny systém (pretože to sú slová, ktoré sa neohýbajú) a pre pravidelné slovesá je to schopnosť našej mysle tvoriť pravidlá (pre slová, ktoré sa ohýbajú).[19]

Podľa neho pravidlá nie sú jednoduchým produktom registrovania gramatických tvarov v reči rodičov, ale produkt inštinktívne špecifikovanej architektúry jazykového systému. Pravidlá sa nevytvárajú a neosvojujú dieťaťom len na základe veľkého počtu výskytu slova v danom tvare. Proti tomu svedčí napríklad aj fakt (hoci sa to na prvý pohľad nezdá), že slová, ktoré sa málo vyskytujú, sa vždy ohýbajú pravidelne. V hre je skôr schopnosť nášho jazykového systému disponovať pravidlami – mentálnou gramatikou [11].

Pinker svoju hypotézu o dvoch kognitívnych systémoch sa snaží doložiť bohatým materiálom zo skúmania rečových porúch súvisiacich s mozgovými a genetickými

[18] Zhoda v mene živočícha v tomto a predchádzajúcom experimente je viac-menej náhodná.

[19] Pinkerov výklad je veľmi atraktívny, spája ho s množstvom priam literárnych príbehov o pravidelných a nepravidelných gramatických útvaroch. Pinker aj pri takej zdanlivo úzkej téme dokáže povodiť čitateľa od dejín lingvistiky, cez fundamentálne psycholingvistické a filozofické teórie (doktríny) až po výklad neurálnych korelátov jazyka.

poškodeniami. Pri analýze týchto porúch uplatňuje metódu dvojitej disociácie (disociácia – rozklad). Podstata tejto metódy spočíva v tom, že ak sa nám podarí dokázať, že existujú pacienti, ktorí majú poruchu nejakej jazykovej schopnosti a zároveň existujú iní pacienti, ktorých porucha je k ich poruche v inverznom vzťahu (to, čo jedným chýba, to presne druhí majú a naopak), ide o dve autonómne poruchy. V tejto súvislosti sa mu podarilo nájsť niekoľko efektných príkladov.

Uveďme aspoň jeden z nich. Podľa Pinkera v takom inverznom, zrkadlovom vzťahu sú napríklad pacienti s SLI (Specific Language Impairment)[20] a pacienti s Williamsovým syndrómom. Pacienti so SLI majú problémy s wug testom. Zaujímavé je, že z bežného ohýbania slov (skloňovania, časovania atď.) nemusíme hneď zistiť, že majú problém s gramatikou. Vzhľadom na to, že zväčša ide o pacientov s vysokou inteligenciou, ktorí sa tieto slová vo všetkých tvaroch ohýbania dokážu naučiť naspamäť (slová v pádoch berú ako rôzne samostatné slová). Oproti nim pacienti s Williamsovým syndrómom, ktorí napriek svojmu nízkemu IQ celkom slušne obstoja v jazykových testoch (pričom dosť často uplatňujú pravidlá aj pri nepravidelných tvaroch). Obidve poruchy sú geneticky podmienené. Gény prvej skupiny pacientov so SLI narúšajú gramatiku, no pritom nepostihujú inteligenciu. A naopak, gény pacientov s Williamsovým syndrómom poškodzujú inteligenciu, ale neoslabujú ich gramatiku. Prví pacienti majú problém s pravidelnou konfiguráciou a všetko riešia pomocou mentálneho slovníka, oproti tomu druhí pacienti všetko riešia pomocou pravidiel. Podľa Pinkera to znamená, že okrem kognitívneho systému, ktorý má na starosti mentálny lexikón (t.j. pamäti) existuje ešte aj systém, ktorý má na starosti pravidlá a je geneticky podmienený.

4 Záver

Ako sme mohli vidieť, uvedené modely sa líšia predovšetkým v tom, aká úloha sa v nich pripisuje dedičnosti a aká prostrediu. Svojho času behavioristi extrémne preceňovali úlohu prostredia. J. Piaget sa snažil o rovnováhu, ale zoznam kognitívnych nástrojov, ktoré má dieťa od narodenia k dispozícii, obmedzil len na základné reflexy. Samozrejme, reakciou na to bolo, že sa sformovala Chomského nativistická iniciatíva, ktorá začala klásť veľký dôraz na vplyv vrodených faktorov. Pozoruhodné je, že dnes sa váhy znovu začínajú nakláňať viac na stranu dôležitosti prostredia v kognitívnom vývine. Súvisí to s konekcionistickou vlnou v psychológii, s modelovaním psychických procesov pomocou umelých neurónových sietí, z ktorého vyplýva, že učenie môže prebiehať veľmi rýchle a mnohé kognitívne štruktúry nemusia byť vrodené. Konekcionisti však nespochybňujú všetky predstavy o vrodenosti. Je to skôr o spresňovaní, čo je vrodené. Je tiež pravda, že nájsť presnú hranicu medzi vrodenosťou a prostredím je takmer neuskutočniteľný sen, pretože v hre je príliš mnoho faktorov. Napriek tomu, toto hľadanie je veľmi produktívne, pretože vedie k novým a novým poznatkom o kognitívnom vývine.

[20] Za SLI sa považujú všetky poruchy reči, ktoré nie sú spôsobené nahluchlosťou, autizmom, retardáciou alebo nejako inou známou poruchou.

***Poďakovanie**:* Táto kapitola vznikla v Centre kognitívnych vied FMFI UK s podporou grantovej agentúry VEGA MŠ SR, projekt č. 1/0602/10 a KEGA MŠ SR, projekt č. 3/7300/09.

Literatúra

[1] Baillargeon, R. (1993): The Object Concept Revisited. New Directions in the Investigation of Infants Physical Knowledge. In: Granrud, C. E. (Ed.): *Visual Perception and Cognition*. Nillsdale, New Jersey, Lawrence Erlabaum.
[2] Berko, J. (1958): The child´s learning of English morphology. *Word 14*, 150-157.
[3] Dolník, J. (2005): Jazykový systém ako kognitívna realita. In: *Jazyk a kognícia*. Bratislava, Kalligram.
[4] Gruber, H. E., Vonèche, J. J. (1995): *Essential Piaget.*London, Jason Aronson.
[5] Houdé, O., Miéville, D. (1994): *Pensée logivo-mathématique*. Nouveaux objets interdisciplinaires. PUF, Paris.
[6] Chomsky, M. (1997): *The Minimalist Program*. Cambridge, Mass., MIT Press.
[7] Jackendoff, R. (2002): *Foundations of Language, Brain, Meaning, Grammar, Evolution. Oxford.* Oxford University Press.
[8] Müller, U., Carpendale, J. I. M., Smith, L.(2009): *Cambridge Companion to Piaget*. Cambridge University Press.
[9] Piaget, J., Garcia, R. (1989): *Psychogenesis and the history of science.*New York, Columbia University Press.
[10] Piaget, J., Inhelderová, B.(1997): *Psychológia dieťaťa*. Bratislava, Sofa.
[11] Pinker, S. (2003): *Slová a pravidlá. Zložky jazyka*. Bratislava, Kalligram.
[12] Pinker, S. (1994): *Language Instinct.* New York, Morrow.
[13] Spelke, E. (1991): Physical Knowledge in Infany. In: Jonas, A. (Ed.): *Perceptual Development in Infancy.* Hillsdale, New Jersey, Erlabaum.

Paralelné evolučné algoritmy

Ivan SEKAJ[1], Michal ORAVEC[1]

Abstrakt. Evolučné algoritmy sú stochastické výpočtové prístupy založené na napodobení prirodzenej evolúcie. Napriek svojej algoritmickej jednoduchosti sú schopné riešiť zložité optimalizačné a prehľadávacie úlohy. Ich špecifikom, najmä pri riešení komplikovaných a rozsiahlych problémov však je, že kladú zvýšené nároky na výpočtový výkon, resp. výpočtový čas. Z tohto dôvodu má význam tieto algoritmy paralelizovať. Paralelné evolučné algoritmy (PEA) predstavujú v súčasnosti veľmi výkonný nástroj riešenia matematických, technických, ekonomických aj iných praktických úloh. V predloženej kapitole budeme pojednávať o klasifikácii, vlastnostiach a architektúrach PEA.

1 Úvod

Evolučné algoritmy (EA) už v dnešnej dobe patria medzi všeobecne akceptované a používané prístupy schopné riešiť rôznorodé úlohy z mnohých aplikačných domén, ktoré je možno formulovať ako optimalizačné alebo prehľadávacie. Obyčajne je však možné do takejto podoby jednoducho transformovať aj mnohé iné problémy, ktoré explicitne nie sú formulované ako optimalizačné. Podmienkou použitia EA je jednak definovanie účelovej funkcie (tzv. fitness), ktorá je minimalizovaná, resp. maximalizovaná a tiež existencia vhodného výpočtového nástroja (funkcia, algoritmus, model, počítačová simulácia, CAD prostriedok, atd.), ktorý umožní vyhodnotenie miery úspešnosti daného potenciálneho riešenia alebo porovnanie jednotlivých riešení navzájom. Z množstva potenciálnych cieľových úloh uveďme optimalizáciu multimodálnych funkcií mnohých premenných, nekonvexné úlohy, zložité kombinatorické úlohy atď. Z praktických oblastí využitia to môžu byť úlohy využívajúce rôzne typy modelov v technických, ekonomických aplikáciách, konštrukčné úlohy a iné.

Výpočtový princíp EA je založený na existencii množiny (populácie) potenciálnych riešení (jedincov) v každom výpočtovom cykle (generácii). Tie sú v priebehu evolúcie navzájom porovnávané, súperia navzájom o udržanie sa v populácii a hlavne o možnosť vytvoriť potomkov. Potomkovia vzniknú kombinovaním a modifikáciou rodičov. Tento prístup umožňuje prehľadávanie prípustného, obyčajne rozsiahleho a mnohodimenzionálneho priestoru prípustných

[1] Ústav riadenia a priemyselnej informatiky, FEI STU Bratislava, Ilkovičova 3, 812 19 Bratislava, E-mail: ivan.sekaj@stuba.sk, michal.oravec@stuba.sk

riešení vo viacerých smeroch. Prax však ukazuje, že ani takáto paralelizácia hľadania nemusí byť postačujúcim predpokladom nájdenia vyhovujúceho riešenia. Najmä vtedy nie, keď sa jedná o zložitý problém. Z toho dôvodu má význam v algoritme zaviesť ďalšiu paralelizečnú vrstvu. Hlavným cieľom tejto kapitoly je pojednanie o princípoch paralelných evolučných algoritmov (PEA), o ich klasifikácii, vlastnostiach a možných prínosoch. Okrem toho v ďalšom ukážeme výsledky niektorých experimentov a porovnáme vybrané prípady PEA.

Nakoľko v predchádzajúcich dieloch tejto publikácie boli základy EA vysvetlené, nebudeme sa opisom ich princípu ani ich prehľadom veľmi podrobne zaoberať. Iba pripomeňme, že medzi evolučné algoritmy patria evolučné stratégie, genetické algoritmy, genetické programovanie, diferenciálna evolúcia, umelý imunitný systém, PSO algoritmus, gramatická evolúcia, SOMA algoritmus, prípadne aj niektoré iné. Základné informácie o EA je možné nájsť napr. v monografiách [7], [12], [5], [21] a pod. Z našich publikácií sú to [9], [16], [23], ale parciálne aj iné. Zároveň, bez ujmy na všeobecnosti, zúžime svoju pozornosť na genetické algoritmy (GA). Tieto sú najčastejšie používaným evolučným prístupom so širokým spektrom aplikačných možností. To čo platí o paralelizácii GA rovnako platí aj o paralelizácii ostatných reprezentantov EA. Než sa ale dostaneme k opisu paralelných evolučných algoritmov resp. paralelných genetických algoritmov (PGA), predsa len sa ešte pristavme pri niektorých otázkach týkajúcich sa princípu EA resp. GA.

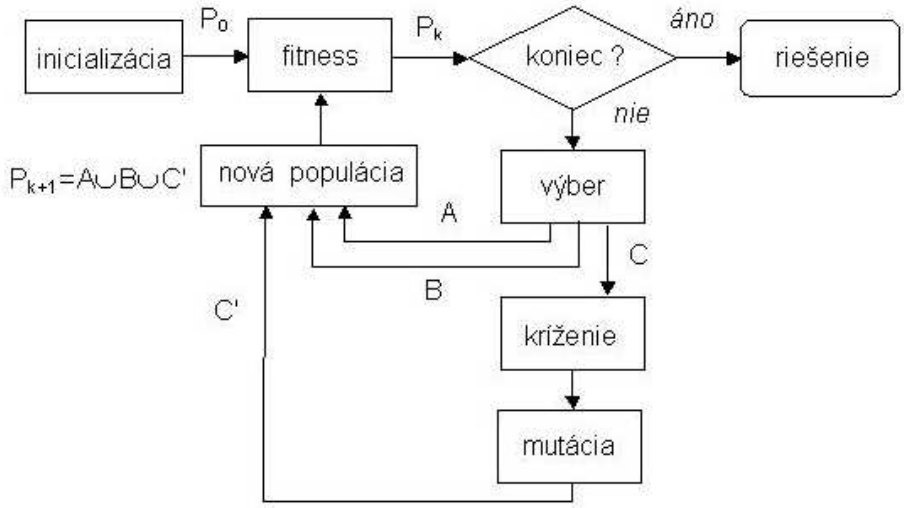
2 Evolučné algoritmy a genetický algoritmus

EA aj GA čerpali svoju inšpiráciu v biologickej evolúcii. História ich výskumu a následného použitia siaha do polovice 20. storočia, predovšetkým do Európy a USA. Ich príchod, rozšírenie a praktické nasadenie bolo možné až vtedy, keď to začal dovoľovať stav a dostupnosť výpočtovej techniky. Je to spôsobené veľkou výpočtovou náročnosťou EA. Podobne ako v prírode, evolúcia požadovaného riešenia daného problému je otázkou mnohých výpočtových generácií, v ktorých sa opakujú relatívne jednoduché operácie. Efektivita EA závisí od jeho parametrizácie, ale aj od zložitosti riešeného problému. Čiže jednak od veľkosti a dimenzie prehľadávaného priestoru a jednak od členitosti účelovej funkcie. To znamená, či je táto unimodálna, (obsahuje jeden extrém, čo je triviálny prípad) alebo je multimodálna (viac lokálnych a jeden globálny extrém), koľko lokálnych extrémov, ako tvarovaných obsahuje a aká je pozícia a tvar hľadaného globálneho extrému. Praktickou mierou efektivity evolučného (ale aj každého iného komparatívneho typu) algoritmu je, koľko vyhodnotení účelovej funkcie potrebuje, aby našiel globálny extrém úlohy (prípadne aspoň vyhovujúce riešenie). Počet vyhodnotení účelovej funkcie je prvý dôležitý ukazovateľ výpočtovej náročnosti každej metódy. Reprezentuje celkový výpočtový čas. Druhým dôležitým ukazovateľom je, aké riešenie algoritmus poskytuje, prípadne, či má schopnosť nájsť globálny extrém. Preto dôležitou vlastnosťou EA je schopnosť opustiť lokálny extrém (ukončiť stagnáciu v „krátkodobo najlepšom“ riešení), pokračovať v prehľadávaní priestoru a smerovať ku globálnemu optimu. Pri zložitých praktických úlohách sa

napriek všetkému väčšinou nepodarí nájsť globálny extrém, čiže najlepšie možné riešenie. Ale obyčajne postačuje aj dobré suboptimálne riešenie.

2.1 Princíp genetického algoritmu a charakteristiky vnútorného stavu populácie

Spomedzi rôznych možných schém uvažujme nasledovnú blokovú schému genetického algoritmu podľa Obr. 1. Na efektivitu tejto konfigurácie vplývajú viaceré parametre. Sú to: veľkosť populácie (P), spôsob výberu rodičov (C) a nemodifikovaných jedincov (A - najlepší, B - iní) a ich počet, miera kríženia v populácii (koľko jedincov je krížených v populácii), miera mutácie (koľko jedincov je mutovaných v populácii).



Obr. 1. Bloková schéma genetického algoritmu

Dôležitými ukazovateľmi charakterizujúcimi správanie sa GA (ale aj iných typov EA) sú selektívny tlak a diverzita. Nakoľko tieto ukazovatele budú mať význam aj pri PGA, stručne ich vysvetlime. ***Selektívny tlak*** [7], [12], [16] je ukazovateľ definovaný ako pomer pravdepodobnosti výberu najúspešnejšieho jedinca ku priemernej pravdepodobnosti výberu jedincov populácie. Zjednodušene povedané, je to miera daného algoritmu (zvolených metód selekcie) preferovať najlepších jedincov oproti ostatným. Zvyšovaním selektívneho tlaku uprednostňujeme najúspešnejšie jedince a potláčame šancu prežitia podpriemerných. Algoritmus s vysokým selektívnym tlakom má vlastnosť rýchlo konvergovať k najbližšiemu lokálnemu extrému, ale tiež v jeho okolí dlho zotrvávať. Populácia bude po istom počte generácií obsahovať iba veľmi podobné alebo dokonca rovnaké jedince. Tomuto javu sa hovorí ***predčasná konvergencia*** (Obr. 2). Selektívny tlak je možné zvyšovať uprednostňovaním výberu najlepších jedincov populácie (tzv. elitarizmus) na úkor náhodne vybraných alebo menej úspešných.

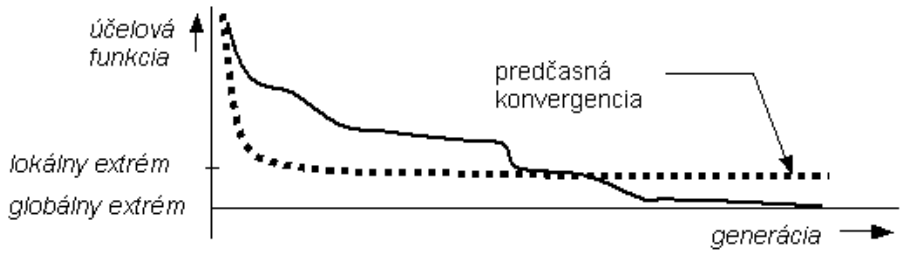
Naproti tomu ***diverzita*** pôsobí opačne. Výber, ktorý spôsobuje vysokú mieru diverzity je taký, pri ktorom sa do novej populácie neustále dostávajú reťazce rôznorodých vlastností. V populácii potom existuje veľká rôznorodosť genetickej informácie. Zvýšenie diverzity predstavuje pre algoritmus väčšiu šancu opustiť lokálny extrém a nájsť nové smery hľadania globálneho extrému. Na druhej strane narastajúca diverzita spomaľuje konvergenciu algoritmu. Mieru diverzity je možné zvyšovať používaním náhodného výberu, ale zároveň aj zvýšeným výskytom mutácií. Jedným z niekedy používaných spôsobov výrazného zvýšenia diverzity je „injektáž“ náhodne generovaných jedincov do populácie, čiže náhrada pôvodných reťazcov novými reťazcami obsahujúcimi iba náhodné hodnoty génov. Diverzitu génov v populácii je možné vyjadriť viacerými spôsobmi. My použijeme vzťah pre rozptyl j-teho génu celej populácie v tvare:

$$r_j = \sqrt{\frac{\sum_{i=1}^{n}\left(x_{i,j} - \overline{x_j}\right)^2}{n-1}}, \tag{1}$$

kde $x_{i,j}$ je j-tý gén i-teho jedinca a $\overline{x}_j$ priemerná (stredná) hodnota j-teho génu v celej populácii o n jedincoch. Mieru diverzity celej populácie môžeme chápať ako sumu rozptylov všetkých dim génov, kde dim je počet génov každého jedinca (dimenzia priestoru)

$$D_{pop} = \sum_{j=1}^{\dim}\left(r_j\right) \tag{2}$$

Z uvedeného je zrejmé, že selektívny tlak a diverzita pôsobia istým spôsobom proti sebe. Optimálnu konvergenciu algoritmu pri danom konkrétnom probléme môžeme zabezpečiť vhodnou rovnováhou medzi týmito činiteľmi.



Obr. 2. Príklad predčasnej konvergencie vs. smerovanie do globálneho extrému.

Okrem diverzity si zadefinujme ešte ďalšiu charakteristiku, ktorá vyjadruje doplnkovú vlastnosť populácie a tou je miera podobnosti jej jedincov - afinita. Afinita i-teho jedinca k j-temu jedincovi populácie je definovaná:

$$affinity(i,j) = \frac{\sum_{k=1}^{dim} gene(i,j,k)}{dim} \times 100\%, \tag{3}$$

kde dim je počet génov jedinca, $0 < i \le n$, $i+1 < j \le n$ a $gene(i,j,k)$ je definovaný ako:

$$gene(i,j,k)=\begin{cases}1 & ak\ |x_{i,k}-x_{j,k}|\le eps\\ 0 & ak\ |x_{i,k}-x_{j,k}|>eps\end{cases}, \tag{4}$$

kde n je veľkosť populácie. To znamená k-ty gén jedinca i je podobný s k-tym génom jedinca j ak ich absolútna vzdialenosť je menšia než veľmi malá konštanta eps (my sme uvažovali v experimentoch hodnotu eps ako 0.5% veľkosti definičnému oboru). Afinitou i-teho jedinca v rámci populácie chápeme jeho percentuálnu podobnosť ku všetkým ostatným jedincom populácie, t.j. aritmetický priemer podobností i-teho jedinca k ostatným jedincom podľa:

$$aff_each(i)=\frac{\sum_{j=1,i\neq j}^{n} affinity(i,j)}{n-1} \tag{5}$$

Nakoniec afinita celej populácie je aritmetický priemer afinít všetkých jej jedincov:

$$aff_whole=\frac{\sum_{i=1}^{n} aff_each(i)}{n} \tag{6}$$

Afinita vyjadruje antagonistickú vlastnosť populácie k diverzite. Ako bolo spomenuté, zvýšením diverzity možno opustiť lokálny extrém a nasmerovať hľadanie ku globálnemu extrému. Analogicky to platí o znížení afinity. Vysoká afinita populácie znamená, že jedince sa na seba navzájom príliš podobajú a algoritmus inklinuje k uviaznutiu v lokálnom extréme. Preto je vhodné počas evolúcie udržiavať afinitu nad istou hranicou. V prírode je evolúcia úspešná tam, kde sa v dostatočnej miere v životných podmienkach vyskytuje zmena a chaos. V opačnom prípade milióny rokov stagnuje. Na druhej strane ale ani príliš vysoká diverzita nie je pre evolúciu optimálna. Zmena, chaos a kľud musia byť v rovnováhe.

2.2 Testovacie funkcie

V ďalšom texte budú na demonštráciu evolučných algoritmov využité tri multimodálne testovacie funkcie, ktoré sú z kategórie tzv. členitých funkcií. Hľadanie globálneho extrému týchto funkcií nie je jednoduchá úloha. Prvá funkcia "Four holes" je v tvare:

$$f_1(X)=\sum_{i=1}^{5}\left(\frac{G_i}{1+d_i(X)}\right)+\frac{G_6}{1+(d_6(X))^2}, \tag{7}$$

kde $d_i=|X-E_i|; i=1,...,6; 0<x_j<100; j=1,...,5;$

G1=2; G2=1; G3= -1; G4= -2; G5= -3; G6= -4;

E1={10 90 90 90 90}; E2={10 20 50 50 70}; E3={25 25 25 25 25};

E4={25 25 25 25 25.5}; E5={5 95 95 95 95}; E6={10 20 30 40 50}.

Jej graf pre 2 premenné je na Obr. 3. Globálny extrém funkcie je v bode $X=\{10,20,30,40,50\}$ a má hodnotu $f_{1,opt}=-4$.

Druhá funkcia je tzv. "Eggholder funkcia", lebo pripomína prepravku na vajíčka (aj keď vajíčka rôznej veľkosti). Graf 2D rezu funkcie je na Obr. 4.

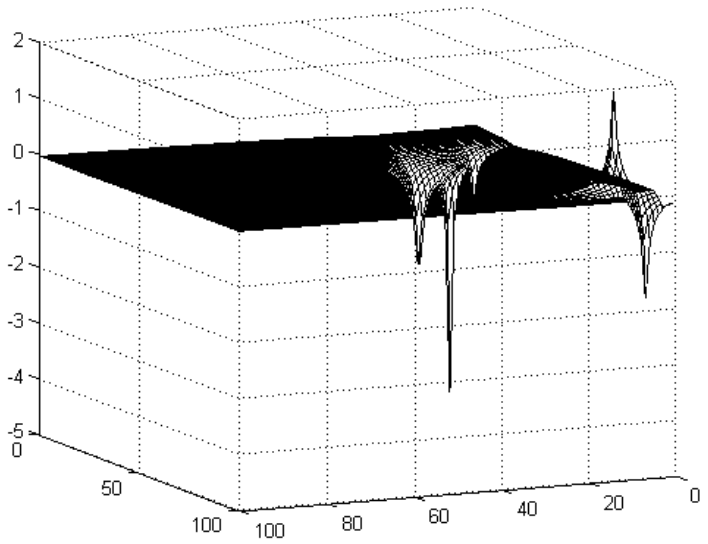
$$f_2(X)=\sum_{i=1}^{n-1}\left(-x_i\sin\left(\sqrt{|x_i-(x_{i+1}+47)|}\right)\right)+\sum_{i=1}^{n-1}(x_{i+1}+47)\sin\left(\sqrt{\left|x_{i+1}+47+\frac{x_i}{2}\right|}\right), \qquad (8)$$

kde $-500<x_i<500; i=1,2,...,n$. V našich testoch n=5 alebo n=10. Poloha globálneho extrému nie je známa.

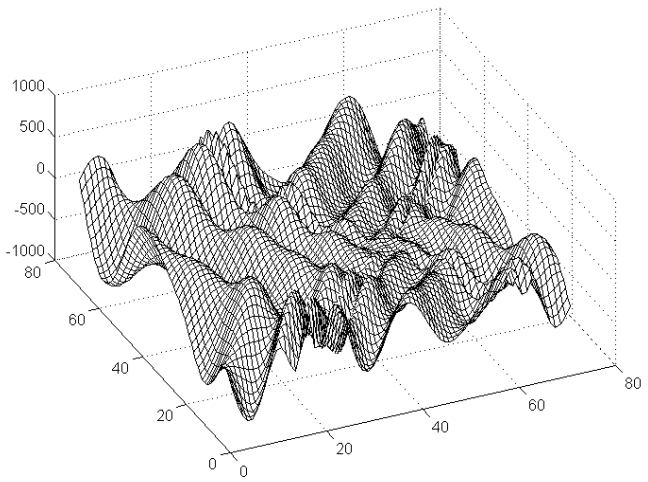
Posledná funkcia je v tvare:

$$f_3(x)=\begin{cases} x_1^2+x_2^2+x_3^2 & ak: \\ & -500>xi>500; \\ & i=1,2,3 \\ \frac{1}{2}\sum_{i=1}^{3}\left(-x_i\sin\left(\sqrt{|x_i|}\right)\right)- & \\ 800\exp\left(-\frac{x_1^2}{1500}-\frac{x_2^2}{1500}-\frac{x_3^2}{1500}\right)- & \\ 400\exp\left(-\frac{(x_1^2-200)}{1500}-\frac{(x_2^2-200)}{1500}-\frac{(x_3^2-200)}{1500}\right)- & inak \\ 300\exp\left(-\frac{(x_1^2+300)}{1500}-\frac{(x_2^2+300)}{1500}-\frac{(x_3^2+300)}{1500}\right) & \end{cases} \qquad (9, 10)$$

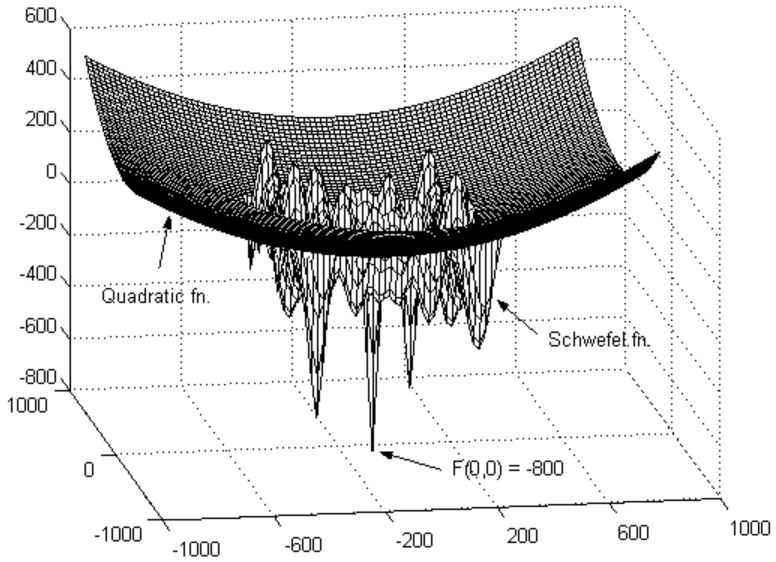
Definičný obor funkcie je: $-1000<x_i<1000; i=1,2,3$. Jej graf pre dve premenné je na Obr. 5. Globálny extrém funkcie je v bode $X=\{0,0,0\}$ a má hodnotu $f_{3,opt}=-800$.



Obr. 3. Graf funkcie f_1 - "Four holes" o 2 premenných (pohľad zdola)



Obr. 4. Graf funkcie f_2 - "Eggholder" o 2 premenných



Obr. 5. Graf funkcie f_3 o 2 premenných

2.3 Prípadová štúdia jednoduchého GA

Na Obr. 6 a až e sú zobrazené rôzne príklady pri minimalizácii funkcie Four holes (Obr. 3). Použili sme obyčajný GA s jednou populáciou s veľkosťou 50 jedincov. V experimentoch je demonštrovaných 5 rôznych konfigurácií algoritmu, pri ktorých sa vyskytujú kombinácie nasledovných prípadov selektívneho tlaku a diverzity:

1. Vysoký selektívny tlak predstavuje prípad, keď rodičia (skupina C) aj nemodifikovaní príslušníci novej populácie (B) boli vyberaní tak, že boli uprednostňované najúspešnejšie jedince.
2. Pri nízkom selektívnom tlaku bola väčšina rodičov (C) aj prežívajúcich jedincov (B) vyberaná náhodne, bez ohľadu na ich úspešnosť. Menšina bola vyberaná pomocou ruletového výberu, ktorý dáva väčšiu pravdepodobnosť prežitia úspešnejším. Zároveň bol vybraný najúspešnejší jedinec populácie (A) a presunutý bez modifikácie do novej generácie, aby bola zabezpečená monotónna konvergencia algoritmu (toto je uskutočnené vo všetkých 5 prípadoch).
3. Vysoká diverzita sa dosahuje vysokou pravdepodobnosťou mutácie rodičov (C) (okolo 80%) a aj aplikáciou náhodného výberu jedincov (v rámci B,C).
4. Nízka diverzita predstavuje prípad, keď rodičia (C) boli mutovaní s nízkou pravdepodobnosťou (okolo 2%,) a nebol použitý náhodný výber jedincov (v rámci B,C).
5. Normálny selektívny tlak predstavuje strednú hodnotu medzi vysokým a nízkym selektívnym tlakom.

6. Normálna diverzita predstavuje strednú hodnotu medzi vysokou a nízkou diverzitou.

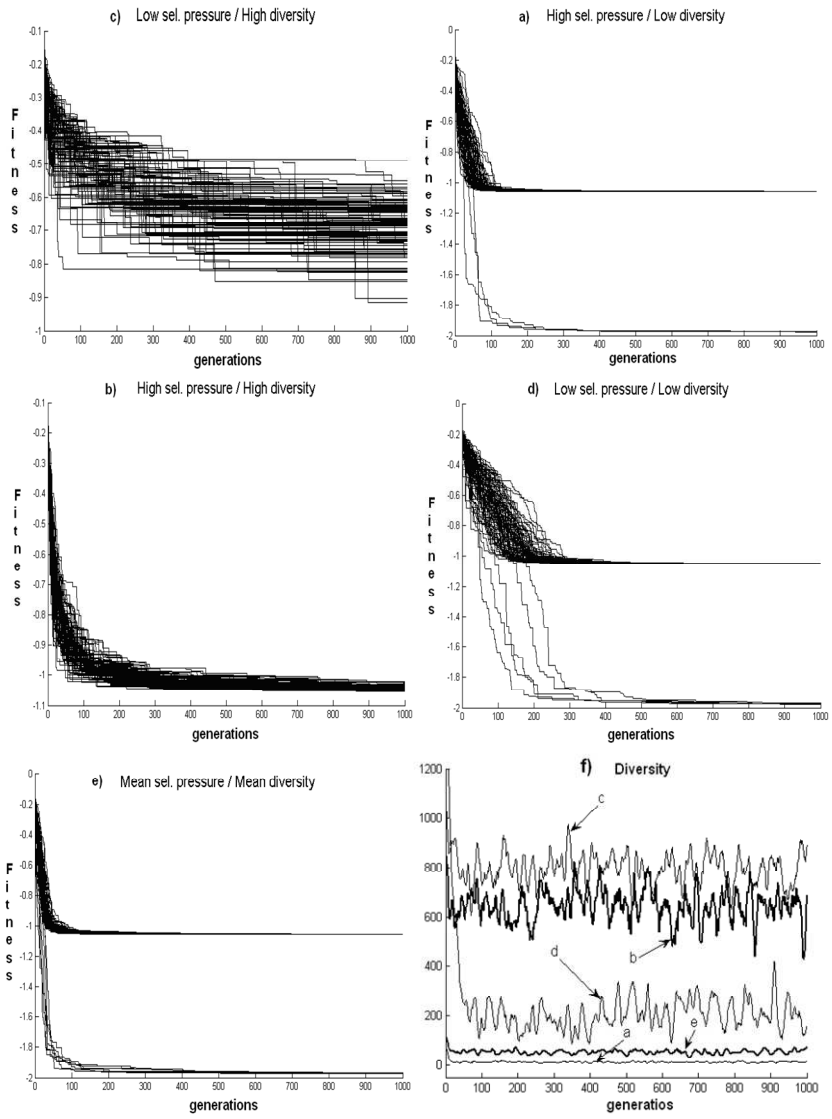
V každom experimente sa jedná o 100 nezávislých spustení GA. V prípade ***a*** (kombinácia 1+4) nastáva pomerne rýchla konvergencia, do lokálneho extrému o hodnote -1 alebo -2. V prípade ***b*** (1+3) je konvergencia pomalšia a menej úspešná, lebo algoritmus nachádza iba najprístupnejší lokálny extrém (-1). Toto spôsobuje príliš vysoká diverzita génov populácie, ktorá brzdí evolúciu. Perspektívne jedince sú príliš často mutované (teda obyčajne aj pokazené), nemajú priestor na prejavenie sa. Ešte horšia situácia je v prípade ***c*** (2+3), keď pôsobí aj malý selektívny tlak. Mierne lepší prípad, než privysoká diverzita nastáva pri nízkej diverzite v prípade ***d*** (2+4). Najlepší výsledok však bol dosiahnutý pre stredné hodnoty selektívneho tlaku aj diverzity, keď sú oba faktory v rovnováhe - prípad ***e*** (5+6). Tieto parametre boli nastavené experimentálne. Ale ani v tomto prípade sa algoritmus nedostal do lepšieho lokálneho extrému, než hodnota -2. Okrem tohto pozorovania si môžeme všimnúť, že vo všetkých prípadoch sa priebeh evolúcie zastaví po fáze prudkého poklesu v okolí lokálneho extrému, ktorý už neopustí. Tento jav sa nazýva predčasná konvergencia. Po istom počte generácií sú už jednotlivé jedince populácie natoľko podobné (afinita populácie je príliš vysoká), že ani mutácia niekoľkých génov ich nevytrhne z okolia lokálneho extrému. Na Obr. 6f sú zobrazené priebehy miery diverzity populácie pre prípady ***a*** až ***e***. Na výpočet rozptylu bol použitý vzťah (2). Je zrejmé, že miera diverzity je vysoká tam, kde bola v algoritme príliš vysoká mutácia. Sú to prípady ***b*** a ***c***, kde je úspešnosť najhoršia. Naopak najúspešnejšie sú prípady ***d*** a ***e***, kde sa diverzita drží v istom vhodnom pásme medzi minimom a maximom.

Pre ilustráciu uvádzame príklad prvých 10 jedincov populácie z počiatočnej generácie, ktorí boli náhodne vygenerovaní v rozsahu (0, 100) – ľavý stĺpec a prvých 10 jedincov populácie po uplynutí 100 generácií výpočtu GA – pravý stĺpec.

68.3150	02.8708	28.9250	69.7314	17.8809	10.0229	20.0434	50.0243	50.0301	69.7745
14.6209	83.4555	27.4791	06.7924	66.7252	10.0229	20.0434	49.9354	50.0301	69.7745
43.4894	75.0410	05.4504	18.2954	14.6548	10.0229	20.0434	49.9354	49.3604	68.3247
95.3993	08.4972	02.6584	29.0352	27.6347	10.0229	19.7496	29.3837	50.0301	77.7041
84.4785	59.3280	31.5915	52.3648	01.6052	11.4005	20.0434	50.0243	50.0301	89.1973
66.7207	80.7443	66.5866	04.9956	94.5181	10.0229	21.4597	49.9354	50.0301	69.7745
28.1308	33.8387	76.8473	00.0949	74.6754	10.0229	20.0434	49.9354	51.0263	69.7745
97.2645	08.6261	02.3106	80.2313	75.8971	10.0229	19.6934	49.9354	50.0301	69.7745
97.8522	88.5891	87.2460	36.2110	65.6072	72.4174	20.0434	49.9354	50.3732	69.7745
66.3350	47.3173	58.9998	38.8806	94.8599	09.3804	20.0434	49.9354	50.0301	63.6264

Po 100 generáciách sú všetky jedince veľmi podobné a nachádzajú sa v blízkom okolí lokálneho extrému $X = \{10, 20, 50, 50, 70\}$, $f_1(X) = -1$. Jednoduchý GA bez ďalších úprav už nemá potenciál dosiahnuť lepší výsledok. A to ani v prípade, že budeme zväčšovať populáciu alebo zväčšovať selektívny tlak. Krátkodobo najlepšie jedince v súčinnosti so selektívnym tlakom strhnú ostatné jedince populácie do silného lokálneho extrému. Zvyšovanie diverzity taktiež pôsobí negatívne.

Možnosťou, ako sa dá ďalej zvyšovať efektivita GA (EA) je ďalšia úroveň paralelizácie. Existuje aj iný prístup, ktorým sú adaptívne evolučné algoritmy. Touto témou sa ale nebudeme v tomto texte zaoberať.



Obr. 6a-f. Priebeh evolúcie pri minimalizácii funkcie Four holes o 5 premenných: a) vysoký selektívny tlak + nízka diverzita, b) vysoký selektívny tlak + vysoká diverzita, c) nízky selektívny tlak + vysoká diverzita, d) nízky selektívny tlak + nízka diverzita, e) stredný selektívny tlak + stredná diverzita, f) priebeh miery diverzity pre všetky prípady a-e

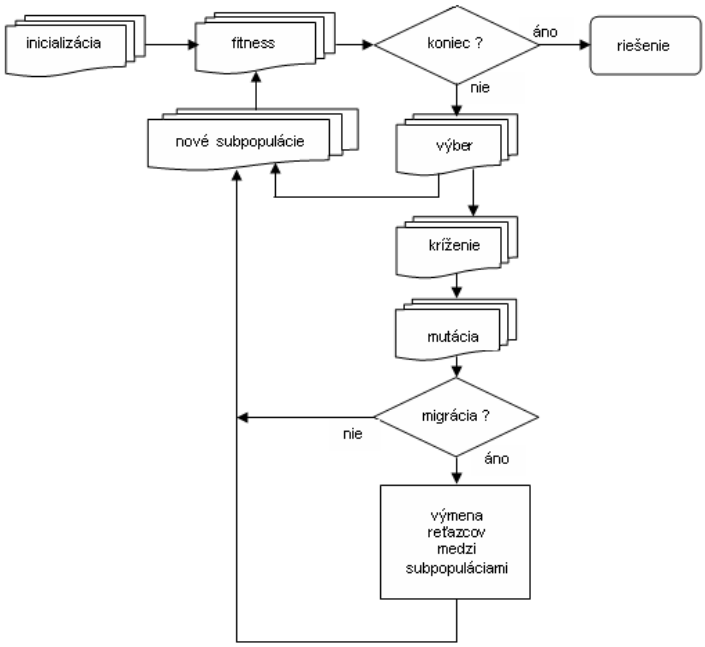
3 Paralelné evolučné a genetické algoritmy

Pod paralelnými evolučnými algoritmami možno vo všeobecnosti chápať ďalšiu úroveň paralelnej organizácie populácie, ktorá má cieľ zvýšiť výkonnosť aj efektivitu EA. Prístupy, ktoré toto rôznym spôsobom riešia je možné klasifikovať pomocou

rôznych hľadísk, ktoré si v ďalšom predstavíme. Prehľadu a klasifikácii PEA sa venovali viaceré publikácie ako [1], [3], [4], [13] a iné.

PEA bez ohľadu na použitý hardvér a softvér a bez ohľadu na počet zúčastnených počítačov či procesorov prinášajú výhody v porovnaní s obyčajnými („jednopopulačnými“) EA (OEA). Sú to hlavne: paralelné hľadanie vo veľkom a mnohodimenzionálnom priestore potenciálnych riešení, lepšia možnosť vnútornej organizácie výpočtu v porovnaní OEA, vyššia efektivita pri rovnakom počte vyhodnotení účelovej funkcie v porovnaní s OEA, schopnosť nájsť lepšie riešenia, schopnosť uvoľniť sa zo stagnácie v lokálnom extréme a smerovať ku globálnemu extrému, resp. nájsť ho. Okrem toho, v prípade zapojenia väčšieho počtu procesorov/počítačov sa dosahuje sublineárneho zvýšenia výpočtového výkonu a teda skrátenia výpočtového času.

Na Obr. 7 je zjednodušená bloková schéma PEA. Treba však poznamenať, že táto schéma neplatí všeobecne pre všetky typy PEA, ale iba tie, ktoré používajú migračný model (viď nižšie). Výpočet tu prebieha vo viacerých nezávislých, ale kooperujúcich subpopuláciách (ostrovoch), či už na jednom alebo viacerých procesoroch/počítačoch či v ľubovoľnej výpočtovej sieti. Každý ostrov je do istej miery izolovaný od okolia a nezávislý. Výpočet nad danou subpopuláciou prebieha podobne, ako to bolo opísané skôr. Napriek tomu však jednotlivé ostrovy sú previazané migračnými prípadne inými väzbami a navzájom si vymieňajú genetickú informáciu.



Obr. 7. Principiálna bloková schéma paralelného genetického algoritmu migračného typu

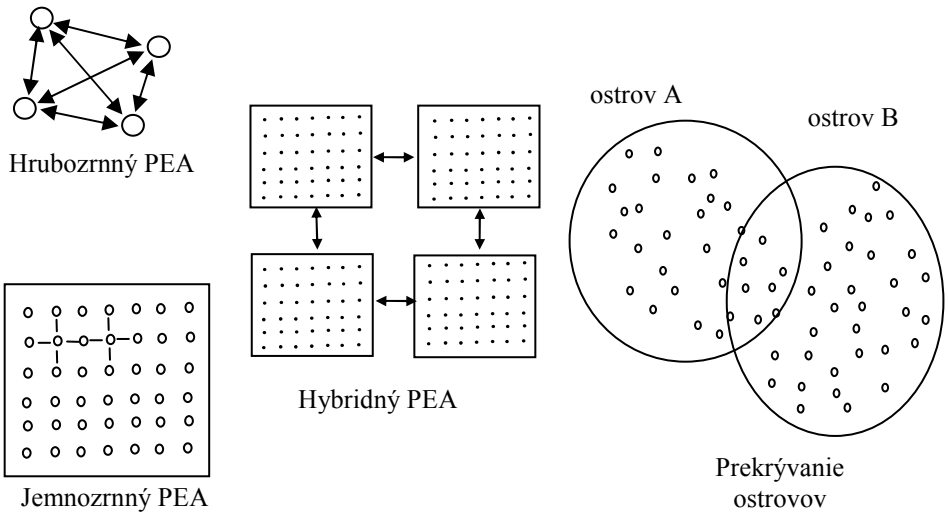
V ďalšom pojednáme stručne o hľadiskách rozdelenia PEA, o ich vlastnostiach a architektúrach. V literatúre sa najčastejšie uvádzajú tri hľadiská klasifikácie PEA:

A. Prvým hľadiskom rozdelenia PEA je fakt, či výpočet prebieha na jednom procesore (počítači) alebo na viacerých procesoroch (multiprocesorových systémoch, počítačových klastroch, ľubovoľných počítačových sieťach, gridoch). Toto je vo svojej podstate iba technická záležitosť a nemá vplyv na funkčnú štrukturalizáciu jednotlivých subpopulácií alebo celej populácie. Nárast počtu procesorov sublineárne zvyšuje výpočtový výkon a skracuje výpočtový čas. Toto má samozrejme veľký význam pri náročných úlohách, kde je nutné distribuovať výpočtové zaťaženie. Dokonca mnohé problémy by boli na jednom procesore nerealizovateľné. Všetky ďalej uvedené hľadiská delenia PEA môžu byť realizované na jednom aj na viacerých procesoroch. V tejto súvislosti ale uveďme jednu špecifickú triedu PEA. Tento typ sa nazýva v literatúre ***globálna*** paralelizácia alebo ***master-slave***. Je tu používaná iba jedna populácia a distribuovane, na viacerých procesoroch je uskutočňovaný iba výpočet fitnes, prípadne aj mutácia a kríženie. Hlavným dôvodom je rozloženie výpočtovo náročného vyhodnotenia fitness, ktoré predstavuje časovo najnáročnejšiu operáciu GA (môže obsahovať zložité výpočty, simulácie atd.). Jednotlivé vyhodnotenia sú priraďované procesorom podľa ich voľnej kapacity. Komunikácia nastáva iba vtedy, keď je odosielaná požiadavka na vyhodnotenie jedinca alebo keď je posielaný výsledok vyhodnotenia naspäť. Tu rozlišujeme synchrónny a asynchrónny prípad. Pri synchrónnom centrálny procesor čaká, až všetky procesory dokončia vyhodnotenie a pošlú hodnoty fitness všetkých jedincov populácie. Až potom sa uskutoční výber a spracovanie novej populácie. Takýto algoritmus sa správa rovnako, ako sekvenčný GA (jednopopulačný GA na jednom procesore), iba výpočet je približne n-násobne rýchlejší, kde n je počet zúčastnených procesorov. Nevýhodou tohto spôsobu je, že centrálny algoritmus musí čakať na najpomalší procesor, čím môžu nastať aj dlhšie prestoje rýchlejších výpočtových segmentov. Pri asynchrónnom režime centrálny procesor nečaká so spracovaním danej generácie, kým všetky (predovšetkým pomalšie) procesory pošlú údaje. To znamená, že v jednej populácii sa môžu stretnúť jedinci z rôznych generácií výpočtu. Toto ale nemusí byť prekážkou.

B. Najčastejším pohľadom na rozdelenie PEA je granularita čiže počet a veľkosť subpopulácií (ostrovov). V [3] sa PEA delia na ***hrubozrnný*** a ***jemnozrnný*** typ (Obr. 8). Pri hrubozrnnom je celá populácia rozdelená na pomerne malý počet navzájom izolovaných ostrovov. Na ich prepojenie sa tu používa iba operátor migrácie, ktorý posiela jedincov z jednej subpopulácie do druhej. Výhoda takejto komunikácie sa prejaví vtedy, ak kombináciou parciálnych riešení z rôznych ostrovov môžeme získať lepšie riešenia. Bližšiemu rozdeleniu hrubozrnných algoritmov sa venuje napr. práca [11]. Naproti tomu pri jemnozrnných PEA populácia obsahuje väčší počet menších ostrovov. Krajný a asi aj najčastejší prípad je, keď každý ostrov reprezentuje jediný jedinec. Tu môžeme, ale nemusíme využívať migráciu. Posledným prípadom v tomto hľadisku delenia sú hybridné PEA (Obr. 8). Predstavujú rôzne kombinácie jemnozrnných a hrubozrnných prípadov.

C. Posledným hľadiskom klasifikácie PEA je spôsob zdieľania jedincov medzi ostrovmi. Prvým spôsobom zdieľania je už spomenutá ***migrácia***. Tu rozdeľujeme celú populáciu na viac ostrovov so vzájomnou migráciou a sú tu zahrnuté hrubozrnné a čiastočne aj jemnozrnné PEA. Druhým spôsobom je ***prekrývanie subpopulácií*** (Obr. 8). V tejto topológii sa jednotlivé ostrovy čiastočne prekrývajú. Chýba tu migrácia medzi ostrovmi. Interakcia jednotlivých ostrovov je zabezpečená operátorom kríženia a výberu v rámci spoločných regiónov, kde sa ostrovy prekrývajú. Migrácia má svoju analógiu v živej prírode v takých prípadoch, keď na geograficky oddelených územiach koexistuje niekoľko subpopulácií rovnakého druhu a môže dochádzať medzi nimi k prechodu alebo vzájomnej výmene jedincov. Prekrývanie je prípad, keď sa teritóriá jednotlivých subpopulácií čiastočne prekrývajú a tu dochádza ku kríženiu jedincov oboch subpopulácií navzájom.

Niektorým prípadom z kategórie delenia B a C sa budeme v ďalšom venovať podrobnejšie.

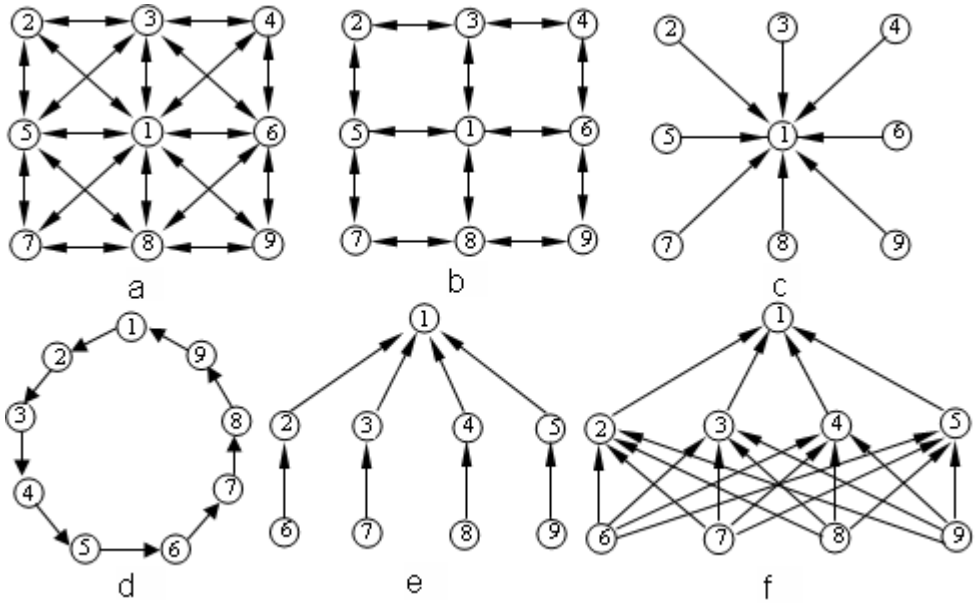


Obr. 8. Príklady rôznych typov PEA

3.1 Hrubozrnný typ PEA

Najčastejšie používanou topológiou v PEA je hrubozrnný prípad resp. migračný model s menším počtom ostrovov. Dôležitými charakteristikami tohto prípadu sú: topológia migračných väzieb (architektúra), veľkosť ostrovov, voľba okamihov migrácie, výber migrantov, prípadne aj iné parametre. Príklady rôznych topológií sú na Obr. 9. V prípade ***a***, ***b*** sa jedná o 2-D mriežkové topológie. Analogicky možno uvažovať aj 3-D prípad. Mriežky sú charakteristické hustými migračnými väzbami. To môže mať za následok že jednotlivé ostrovy sa pomerne intenzívne ovplyvňujú. Tu sa ako nevýhoda prejavuje, že krátkodobo najlepší jedinec ovplyvní aj susedné ostrovy a

strhne v nich riešenie do lokálneho extrému. Tento nežiaduci jav je o to silnejší, keď aj perióda migrácie je krátka. O tom bude reč ešte neskôr. Prípady ***c***, ***e***, ***f*** sú vrstvové topológie. Tu je výhodou, že niektoré ostrovy sa navzájom nemôžu ovplyvňovať (napr. v prípade ***e***, ***f*** uzly 6 až 9 sa neovplyvňujú, alebo 2 neovplyvní 3 atď.).



Obr. 9. Príklady rôznych topológií hrubozrnných PEA

Jednotlivé ostrovy môžu používať rovnako nakonfigurované EA, s rovnakými parametrami. Ale výhodou môže byť, keď sú použité rôzne konfigurácie. Napríklad pri topológii ***e*** alebo ***f*** je výhodné, keď v dolných uzloch 6, 7, 8, 9 sa prejavuje nižší selektívny tlak a vysoká diverzita, ktoré produkujú nové smery hľadania. V uzloch vyššej úrovne 2, 3, 4, 5 je zas naopak vhodné aplikovať vyšší selektívny tlak a menšiu diverzitu, aby sme sa upriamili na najlepšie jedince zdola. Vo vrchnom ostrove sa použije stredne nakonfigurovaný EA na konečnú evolúciu výsledku. Experimentálne porovnanie vybraných topológií ukážeme v ďalšom. Pri návrhu EA v jednotlivých uzloch nám nič nebráni použiť dokonca rôzne typy EA pre jednotlivé uzly (GA, evolučné stratégie, diferenciálna evolúcia atď.), ak nájdeme na to opodstatnenie.

Pokiaľ ide otázku ***počtu ostrovov*** v hrubozrnnom PEA, nedá sa paušálne povedať, aká je optimálna hodnota. Bežne sa používa 4 a viac. Na Obr. 10 je porovnanie evolúcie pri minimalizácii Eggholder funkcie 10 premenných. Sú tu porovnané prípady pre 5, 9, 16 a 25 ostrovov. Veľkosť populácie bola vo všetkých prípadoch rovnako 50 jedincov. Nakoľko veľkosť celkovej populácie sa mení s počtom ostrovov, z dôvodu objektívneho porovnania je graf evolúcie PGA vztiahnutý na počet vyhodnotení účelovej funkcie, namiesto počtu generácií, ako v ostatných grafoch. Zväčšovanie počtu ostrovov zvyšuje diverzitu celej populácie, ale súčasne narastá aj výpočtová náročnosť. Príliš veľký počet ostrovov už neprináša požadovaný efekt. V našich experimentoch sme najčastejšie používali topológie s počtom ostrovov od 9 do 16, čo bolo vo všetkých aplikáciách postačujúce. Druhou otázkou je ***veľkosť subpopulácií***

(počet jedincov v ostrovoch). Tu platí to isté, čo pri obyčajných, jednopopulačných EA. Príliš malá populácia nezabezpečuje dostatočnú diverzitu a priestor na vznik nových perspektívnych jedincov. Ľahko je ovplyvnená krátkodobo najlepšími jedincami. Príliš veľká populácia zas už neprináša významné zlepšenie a zvyšuje výpočtovú náročnosť. O voľbe vhodnej veľkosti populácie v súvislosti s náročnosťou úlohy sa pojednáva napr. v [7], [12].

Ďalším dôležitým faktorom hrubozrnných PEA je spôsob migrácie medzi ostrovmi. Tu ide o dve otázky. Jednak je to ***frekvencia migrácií*** (pri periodickej migrácii), prípadne ***voľba okamihov migrácie*** (pri aperiodickej) a tiež ***výber migrujúcich jedincov***. Pri synchrónnej migrácii je potrebné definovať veľkosť intervalu vyjadrenom v počte generácií, po ktorom nastáva migrácia v smere definovaných migračných väzieb. Okrem iného je o tomto pojednané aj v [20]. Ak je tento interval príliš krátky PEA sa môže deformovať až na jednopopulačný EA a čiastočná izolácia ostrovov sa stráca. Príliš dlhý interval migrácie zas smeruje k úplnej izolácii ostrovov, takže optimálna voľba leží medzi týmito mantinelmi. Pri návrhu vhodného intervalu sa odporúča dať algoritmu dostatok času na evolúciu perspektívnych jedincov, ktorí už obsahujú vhodné stavebné bloky optimálneho riešenia. Tie sa migráciou dostanú do inej subpopulácie, kde práve takéto bloky môžu chýbať. Pokiaľ takéto stavebné bloky ešte nie sú vyvinuté, migrácia nemá požadovaný efekt. Iný „návod“ na voľbu migračného intervalu je dať k dispozícii toľko generácií, aby sa priebeh účelovej funkcie stabilizoval po najstrmšom poklese (pri minimalizácii). Vtedy je už populácia do istej miery nasmerovaná do nejakého extrému. Dá sa to odhadnúť po niekoľkých spusteniach jednoduchého (jednopopulačného) GA (SGA - Single GA). Napríklad podľa Obr. 6 e je vhodné voliť interval migrácie blízko hodnoty 100 generácií. Na Obr. 11 je experimentálne porovnanie GA pri minimalizácii Eggholder funkcie 10 premenných. Každá krivka je graf priemeru 30 spustení PGA s inou hodnotou periódy migrácie T. Z výsledkov je vidieť, že pre krátke periódy migrácie T=1 generácia až T=10 generácií je evolúcia horšia, než prípad PGA bez migrácie. To preto, že v subpopulácii ešte nenastali vhodné podmienky pre výmenu genetickej informácie a migrácia je skôr neprospešná. Výmena informácií je pričastá a podmienky sú podobné, ako v jednej spoločnej populácii (SGA). Naproti tomu pri T>20 už majú jednotlivé ostrovy dostatok času na tvorbu perspektívnych jedincov a konvergencia PGA je úspešnejšia, než bez migrácie. Najrýchlejšia konvergencia je v tomto prípade pre periódu migrácie T=100 generácií.

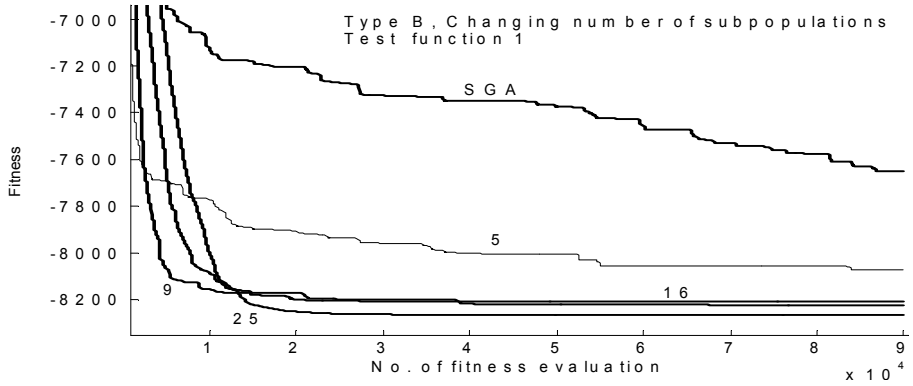
Pri aperiodickej migrácii nedochádza k výmene jedincov v pravidelných a vopred stanovených intervaloch. Migrácia nastáva vtedy, keď sú splnené vhodné, dizajnérom stanovené podmienky. Napríklad, keď diverzita génov populácie, vyjadrená v tvare (2) poklesne pod stanovenú hranicu. Vtedy z(do) daného ostrova migrujú jedinci do(zo) susedných ostrovov atď.

Poslednou otázkou súvisiacou s migráciami je výber migrujúcich jedincov. Ide o výber jedincov zo zdrojového ostrova, ktorí budú skopírovaní a tieto ich klony budú migrovať v smere definovaných migračných spojení podľa topológie. Ďalej ide o výber jedincov, ktorí budú v cieľovom ostrove nahradení prichádzajúcimi jedincami (aby veľkosť subpopulácie zostala rovnaká). Analýza tohto problému je aj v [20]. Najčastejšie sa používa jedna z nasledujúcich možností:

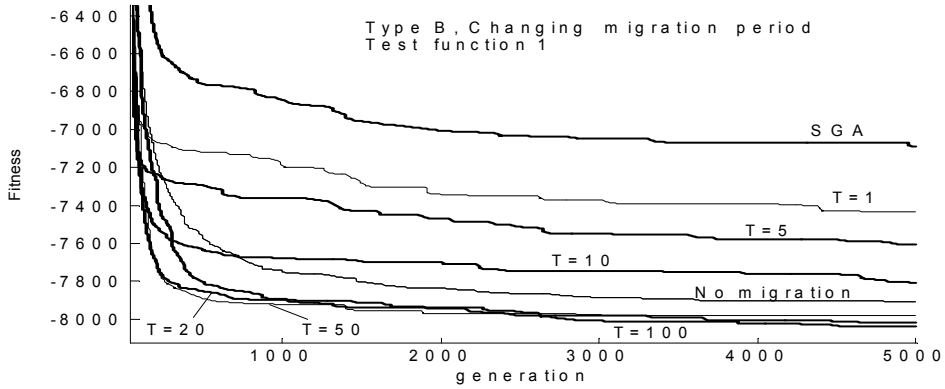
1. najlepší jedinec (alebo viacerí) zo zdrojového ostrova nahradí náhodného jedinca (viacerých) v cieľovom ostrove (best - random)
2. najlepší jedinec zo zdrojového ostrova nahradí najslabšieho jedinca v cieľovom ostrove (best - worst)
3. náhodný jedinec zo zdrojového ostrova nahradí náhodného jedinca v cieľovom ostrove (random - random)
4. náhodný jedinec zo zdrojového ostrova nahradí najslabšieho jedinca v cieľovom ostrove (random - worst)
5. iný spôsob

Na Obr. 12 sú porovnané úspešnosti prípadov ***a*** až ***d*** pri použití GA na minimalizáciu funkcie Four holes. Migroval jeden jedinec podľa topológie na Obr. 9a. Možno konštatovať, že v danom prípade je najúspešnejší spôsob migrácie best - rand.

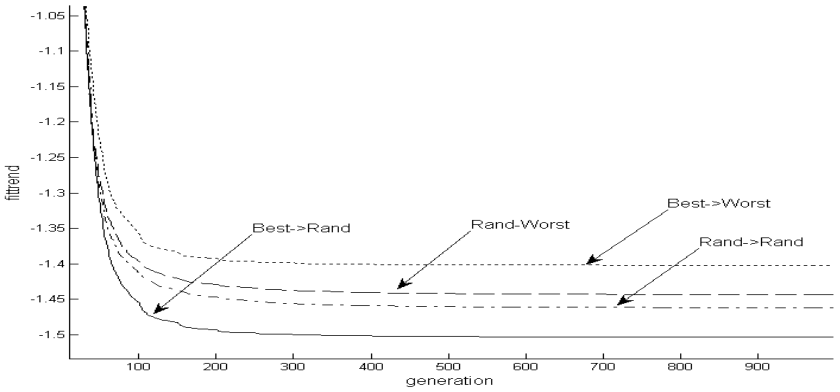
Ďalšou otázkou je, koľko migrantov v rámci jedného spojenia uvažujeme. Najčastejší prípad je 1 až 2 jedince.



Obr. 10. Porovnanie rôzneho počtu ostrovov na konvergenciu PGA



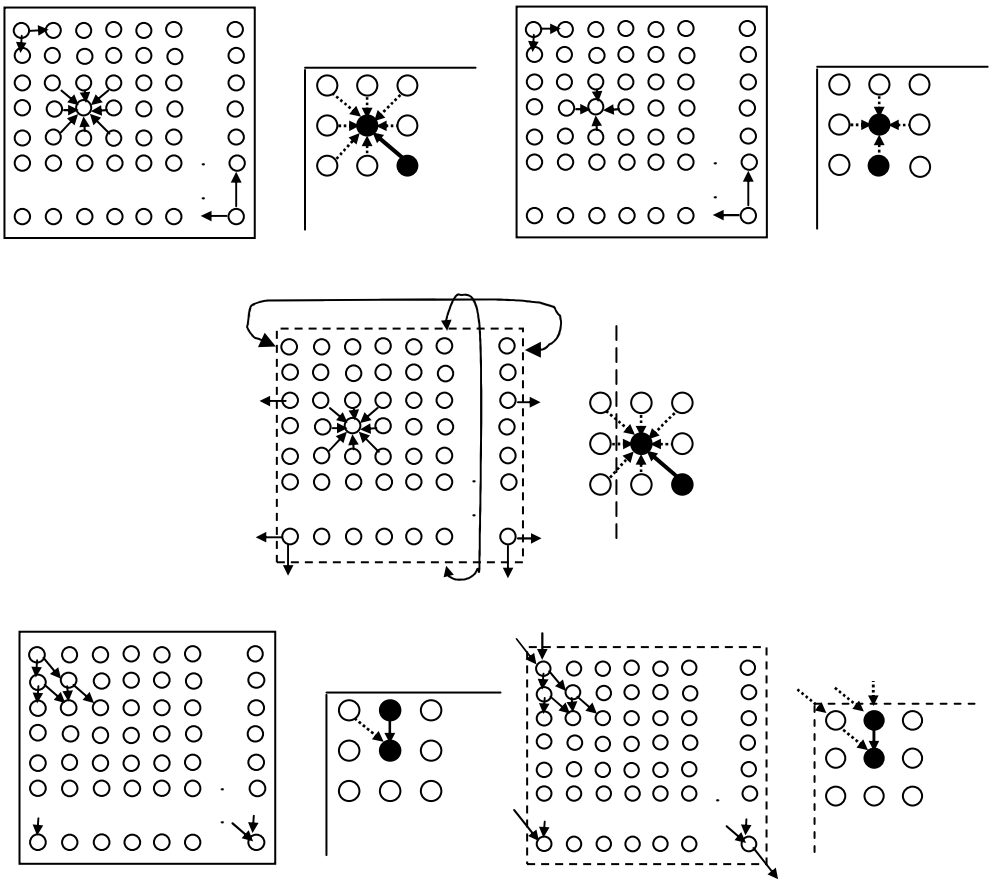
Obr. 11. Vplyv periódy migrácie na konvergenciu PGA



Obr. 12. Porovnanie rôzneho spôsobu výberu migrantov

3.2 Jemnozrnný typ PEA

Skôr sme už naznačili, že jemnozrnný typ PEA je charakteristický väčším počtom ostrovov s menším počtom jedincov. Hraničným a aj častým prípadom je, keď ostrov reprezentuje jeden jedinec (ak v tomto prípade vôbec môžeme ešte hovoriť o ostrovnej štruktúre). Takéto prípady sú aj na Obr. 13. V danom prípade sa jedná o PEA s prekrývaním ostrovov a nie o migračný typ. Každý krúžok predstavuje jedného jedinca, ktorý si v každej generácii hľadá partnera medzi svojimi susedmi, ako je naznačené šípkami. S jedným vybraným partnerom sa s istou pravdepodobnosťou (typicky okolo 70%) skríži. Ako metódu výberu partnera spomedzi susedov môžeme použiť bežné metódy využívané v GA ako elitársky výber, ruletový, turnajový, náhodný, atď. V nakreslenom prípade sú jeho potenciálni partneri iba bezprostrední susedia. Čiže pre 2D prípad sú 4 (v kolmých smeroch) resp. 8 (aj v šikmých). Ale môžeme akceptovať aj širšie okolie. Pokiaľ uvažujeme aj nasledovný rad, pribudne ďalších 16 potenciálnych partnerov. Analogické by to bolo pri 3D prípade (kocka, kváder). Okrem kríženia sa aj tu používa operátor mutácie, ktorý v každej generácii zmutuje náhodné gény náhodne zvolených jedincov populácie. Na Obr. 13 a je jemnozrnná topológia PEA kde komunikácia prebieha v rámci 8 susedov alebo iba 4 susedov. Prípad ***a*** je ohraničený na krajoch mriežky, za ktorou už sa susedia nenachádzajú. Prípad ***b*** sa líši tým, že mriežka je uzavretá do valca v oboch smeroch. Tu hranice nie sú. Odlišné vlastnosti majú topológie ***c*** a ***d***. Tu komunikácia nastáva iba smerom nadol a vpravo, čiže populácia má smerové vlastnosti. Porovnanie prípadov ***a-d*** pri minimalizácii funkcie Eggholder je na Obr. 14. Každý graf je priemerom 100 spustení PGA. Na Obr. 15 je porovnanie rôznych metód výberu partnera. Najúspešnejšie sa v tomto prípade javí použitie náhodného výberu partnera spomedzi všetkých susedov. Uvedené topológie jemnozrnných PGA sa nazývajú aj ***difúzne***, nakoľko genetická informácia sa šíri podobne ako pohyb hmoty pri difúzii. V literatúre možno nájsť aj pomenovanie celulárne PGA.



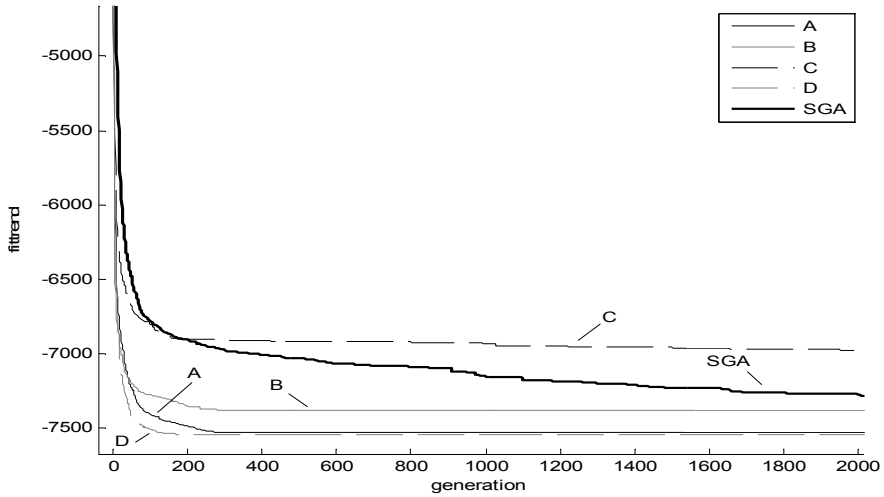
Obr. 13. a-d Jemnozrnné PGA s rôznymi topológiami

Jedným z rozdielov medzi jemnozrnnými a hrubozrnnými PEA je rýchlosť šírenia genetickej informácie. Pri jemnozrnných sa táto informácia šíri pomaly, v priebehu jednej generácie iba od suseda k susedovi, ako spomínaný difúzny pohyb. Pri hrubozrnných PEA sa v jednej generácii môže skrížiť jeden jedinec s ľubovoľným iným jedincom v rámci ostrova a po migrácii následne s jedincami iného ostrova. V druhom prípade je mobilita jedincov vyššia.

3.3 Reinicializácia

Ak si pozrieme ľubovoľný priebeh konvergencie GA, môžme konštatovať, že najstrmší pokles účelovej funkcie nastáva v počiatočnej fáze evolúcie. Potom sa pokles spomalí, až začne stagnovať v okolí nejakého extrému, obyčajne lokálneho. Je to spôsobené poklesom diverzity génov v populácii. Tento jav sme vysvetľovali už v časti 2.3 . Možnou cestou, ako tomuto nepriaznivému správaniu zabrániť je udržiavať diverzitu v populácii na dostatočne vysokej hodnote (alebo afinitu populácie na dostatočne nízkej). To je možné dosiahnuť napríklad adaptívnym prispôsobovaním miery mutácie v priebehu výpočtu GA na základe vyhodnocovania diverzity (napríklad použitím

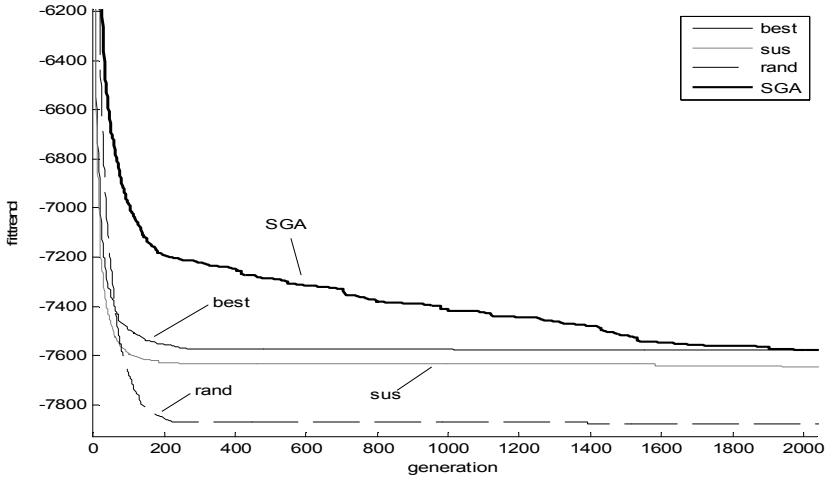
vzťahu (2)). Inou veľmi účinnou metódou je reinicializácia populácie. Tá sa ukazuje byť veľmi efektívna v prípade členitých účelových funkcií s ťažko lokalizovateľnými extrémami. Podstatou reinicializácie je nahradenie jednej alebo viacerých subpopulácií náhodne vygenerovanými jedincami, rovnako ako pri počiatočnej inicializácii. Predtým však je potrebné zabezpečiť, aby sa tie najúspešnejšie jedince zachovali. To je možné dosiahnuť tak, že pred reinicializáciou tieto jedince premigrujú do ostrova, ktorý nereinicializujeme. Ak uvažujeme topológie na Obr. 9 a-d, je zmysluplné reinicializovať ktorýkoľvek ostrov, alebo všetky, napríklad okrem ostrova 1. Pri topológiách Obr. 9 e,f zas môžme zvoliť pre reinicializáciu napríklad iba ostrovy 6-9.



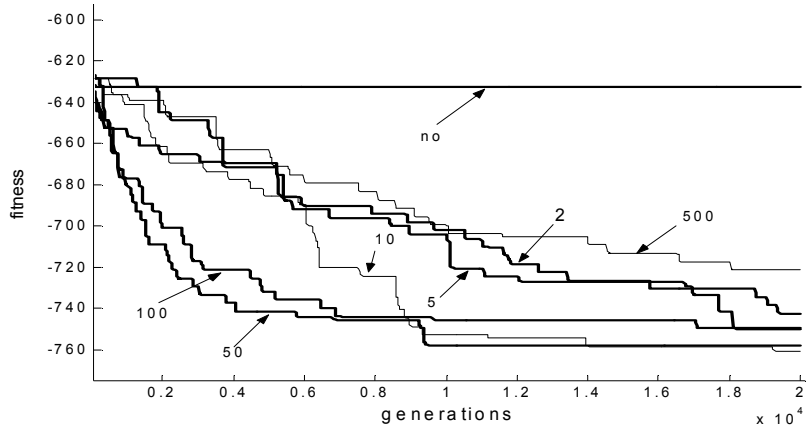
Obr. 14. Porovnanie rôznych topológií jemnozrnných PGA, pri minimalizácii funkcie Eggholder o 10 premenných

Pri reinicializácií nie je podmienkou, že musíme nahradiť všetkých jedincov daného ostrova. Môžeme reinicializovať iba časť jedincov. Tu však môžu nastať komplikácie, ak v algoritme pôsobí silný selektívny tlak. Títo jedinci môžu byť rýchlo z populácie odstránení. To preto, lebo v počiatočných fázach ešte nemajú dostatočnú úspešnosť v porovnaní so staršími jedincami populácie, s ktorými nemôžu konkurovať.

Ďalšou otázkou je, kedy má k reinicializácii dochádzať. Najjednoduchší prípad je periodická reinicializácia. Dĺžka intervalu reinicializácie sa volí podobnou úvahou, ako bolo opísané pri perióde migrácie. Obe periódy môžu byť aj rovnaké. Pokiaľ nevieme vhodne odhadnúť tento parameter, je možné využiť iné metódy. Reinicializácia môže nastať vtedy, keď diverzita v ostrove poklesne pod definovanú hranicu (napríklad opäť využitím vzťahu (2)). Iným kritériom môže byť dosiahnutie predpísanej hodnoty afinity jedincov v danej subpopulácii (použitím vzťahu (6)). Z našich experimentov vyplynulo, že vhodnou podmienkou pre reinicializáciu je dosiahnutie 70%-nej afinity v subpopulácii.

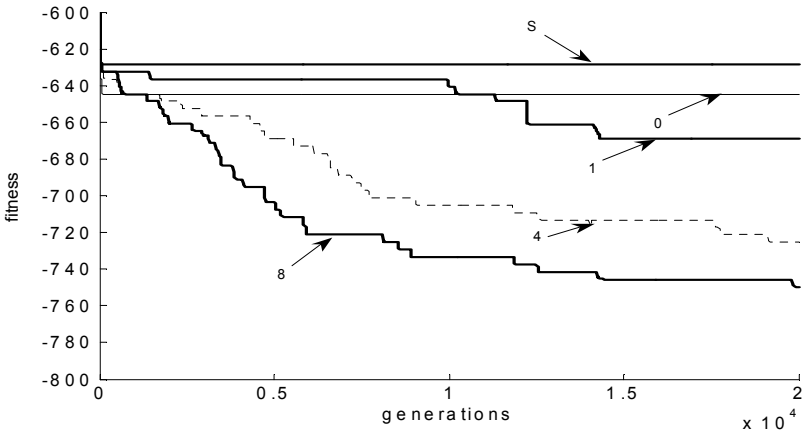


Obr. 15. Porovnanie rôznych metód výberu partnera pri jemnozrnných PGA pri minimalizácii funkcie Eggholder o 10 premenných. SGA - jednoduchý GA, best - výber najúspešnejšieho suseda, sus - ruletový výber, rand - náhodný výber

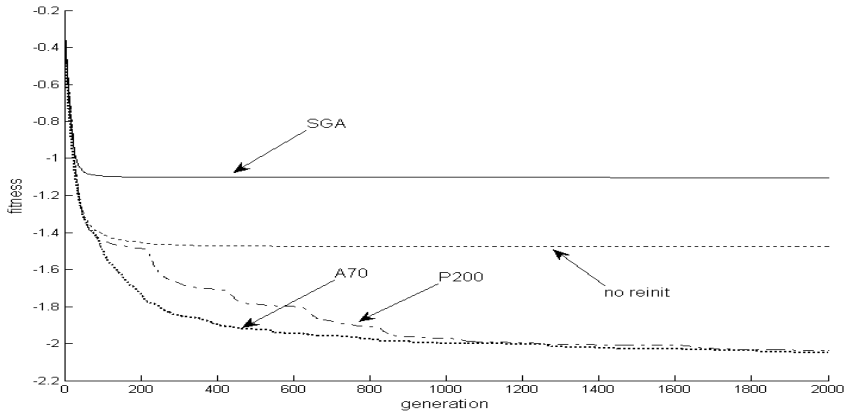


Obr. 16. Periodická reinicializácia, hrubozrnná topológia a, 8 reinicializovaných ostrovov 2-9, rôzne periódy reinicializácie, no – bez reinicializácie

V ďalšom ukážeme niekoľko experimentálnych porovnaní. Na Obr. 16 je porovnanie rôznych prípadov periodickej reinicializácie. Každý graf je priemer 30 spustení PGA s testovacou funkciou f_3. Najlepšie výsledky sú dosiahnuté s periódami reinicializácie 10 a 50 generácii. Na Obr. 17 je porovnanie rôzneho počtu reinicializovaných ostrovov. Najlepší prípad je pre 8 ostrovov. Na Obr. 18 je porovnanie rôznych typov reinicializácie pri funkcii Four holes. Najúspešnejším prípadom je aperiodická reinicializácia pri dosiahnutí 70% - nej afinity.



Obr. 17. Periodická reinicializácia, hrubozrnná topológia *a*, rôzne počty reinicializovaných ostrovov, *s* – SGA, 0 – bez reinicializácie.

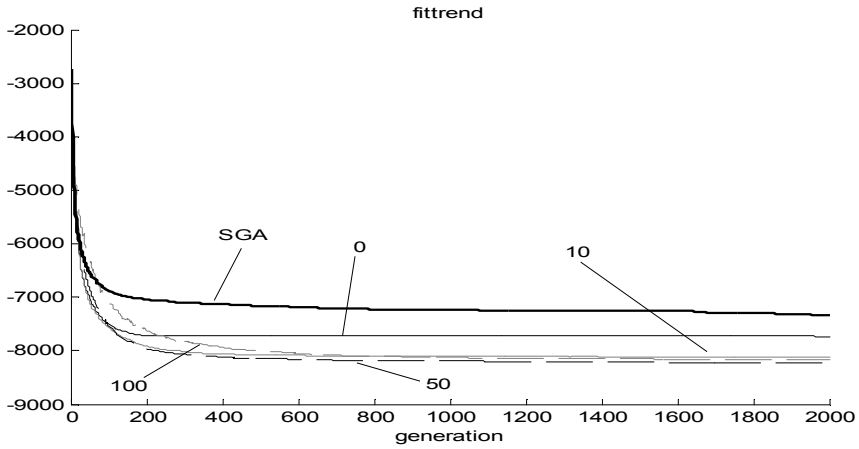


Obr. 18. Funkcia Four holes, hrubozrnná topológia a, no reinit – bez reinicializácie, P200 – periodická reinicializácia po 200 generáciách, A70 – reinicializácia vždy po dosiahnutí 70% - nej afinity populácie, reinicializuje sa 8 ostrovov (2 až 9)

Pri jemnozrnných typoch PEA, kde sa ostrovy nevyskytujú, môže tiež dochádzať k reinicializácii. Výber reinicializovaných jedincov je obyčajne náhodný. Na Obr. 19 je porovnanie evolúcie jemnozrnného PGA s rôznymi počtami reinicializovaných jedincov. Veľkosť populácie tu bola 540 jedincov s topológiou podľa Obr. 13 a. Najlepší výsledok bol dosiahnutý s počtom 50 reinicializovaných jedincov, ale podobnú úspešnosť sme dosiahli aj s 10 a 100 jedincami.

Reinicializácia je drastický a pomerne jednoduchý spôsob, ako dostať populáciu zo stagnácie v silnom lokálnom extréme. Analógiou reinicializácie PEA v evolúcii živej prírody sú prírodné katastrofy, ako boli výkyvy životných podmienok v počiatočných fázach vzniku života, neskôr veľké suchá, doby ľadové a podobne, ktoré zapríčinili úhyn celých skupín živočíšnych druhov. Najpopulárnejším príkladom je akiste pád asteroidu pred 65 mil. rokmi, ktorý znamenal koniec éry veľkých jašterov

a zrejme umožnil príchod nového, silného, lokálneho extrému v živočíšnej ríši, ktorým je homo sapiens. Zrejme nezodpovedanou otázkou zostane, či aj nás čaká ďalšia reinicializácia (nebodaj zapríčinená samotným človekom), než sa objaví na Zemi ešte dokonalejší živočíšny druh.



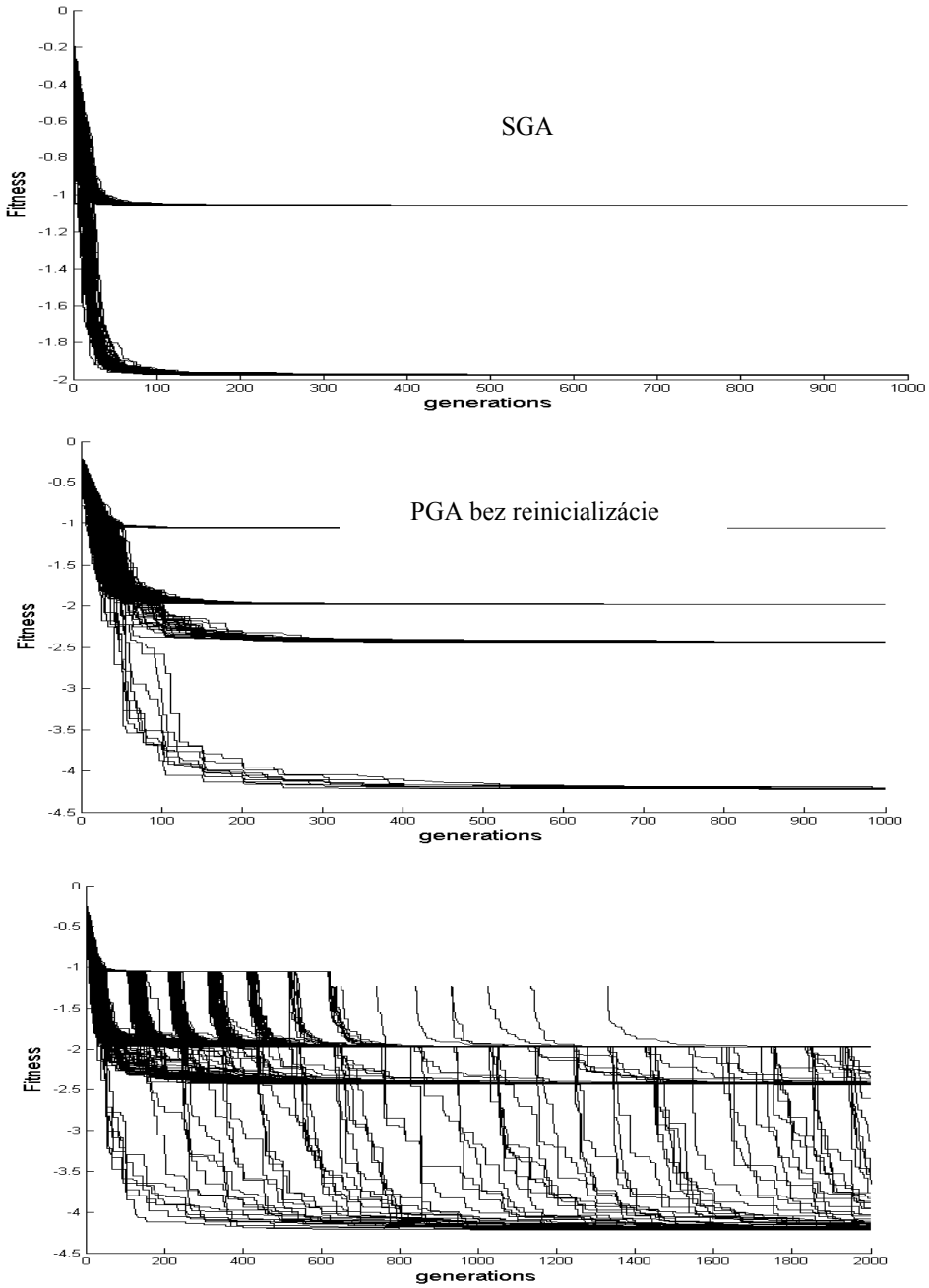
Obr. 19. Porovnanie rôznych počtov reinicializovaných jedincov pri jemnozrnnom PGA s topológiou a, minimalizácia Eggholder funkcie 10 premenných

3.4 Prípadová štúdia PGA

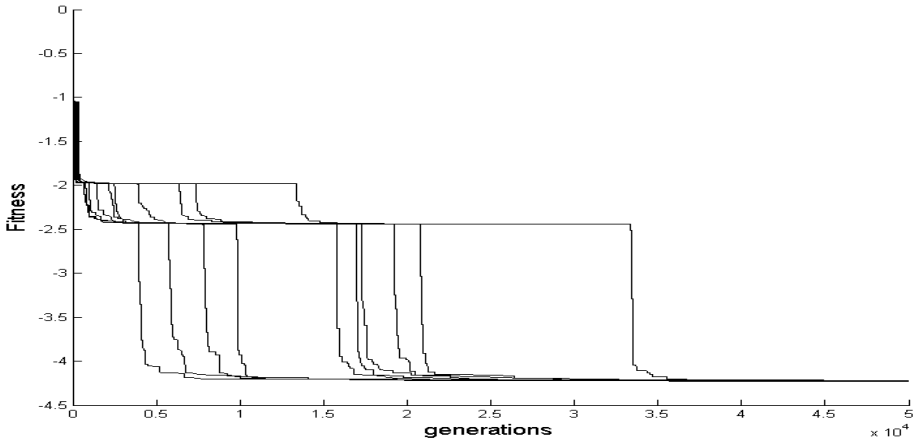
V poslednom experimente demonštrujeme tri verzie genetického algoritmu pri minimalizácii funkcie Four holes, podobne ako to bolo v časti 2.3 . Je to pomerne ťažký optimalizačný problém, kde v 5D priestore hľadáme úzky globálny extrém o hodnote -4. Prvý obrázok Obr. 20 - hore zobrazuje obyčajný GA (SGA) so 180 jedincami v jednej populácii, druhý prípad na Obr. 20 - stred je PGA s topológiou podľa Obr. 9 a bez reinicializácie a posledný prípad Obr. 20 - dolu je tá istá topológia s periodickou reinicializáciou ostrovov 2 až 9 po 100 generáciách. V každom z 9 ostrovov je po 20 jedincov, čiže spolu je v PGA tiež 180 jedincov. Pred reinicializáciou vždy premigrujú najúspešnejšie jedince do ostrova 1. Perióda migrácie je vo všetkých prípadoch 50 generácií pri aplikácii 1 migranta (best-worst).

Pre každý z troch prípadov bolo uskutočnených 500 spustení. SGA nie je veľmi úspešný, nakoľko sa mu podarilo nájsť iba jeden z lokálnych extrémov -1 alebo -2. Po krátkej fáze poklesu uviazol v tomto lokálnom extréme a už ho neopustil. Podobná stagnácia nastala aj v prípade nereinicializovaného PGA, s tým rozdielom, že takýto algoritmus bol schopný z 500 spustení nájsť najlepšie riešenie (v bode -4) 9-krát. Okrem toho klesanie účelovej funkcie je pomalšie a je v ňom možné pozorovať vplyv migrácie medzi ostrovmi s periódou 50 generácií. Najlepšie výsledky poskytuje PGA s reinicializáciou. Ten našiel globálny extrém do počtu 2000 generácií z 500 spustení 71-krát. Pri dostatočne veľkom počte generácií by jeho úspešnosť vzrástla asi na 70%. Na poslednom obrázku (Obr. 21) je demonštrovaných 10 spustení PGA s topológiou podľa Obr. 9 c po 50 000 generáciách. Ostatné podmienky boli podobné, ako

v predchádzajúcom prípade. Táto konfigurácia dosahuje 100% úspešnosť nájdenia extrému v bode -4.



Obr. 20. Porovnanie SGA (hore), PGA (topológia Obr. 9 a) bez reinicializácie (stred) a PGA (topológia Obr. 9 a) s reinicializáciou (dolu)



Obr. 21. PGA (topológia Obr. 9 c) s reinicializáciou 8-ich ostrovov s periódou reinicializácie 100 generácií, po 50 000 generáciách evolúcie

4 Záver

Paralelné evolučné algoritmy reprezentujú výkonný a robustný nástroj na riešenie náročných optimalizačných a prehľadávacích problémov. Ich hlavnými prínosmi sú:

1. Paralelné hľadanie vo veľkom a mnohodimenzionálnom priestore potenciálnych riešení,
2. lepšia možnosť vnútornej organizácie výpočtu a distribúcia parametrov evolučného algoritmu v porovnaní obyčajným (jednopopulačným) evolučným algoritmom,
3. vyššia efektivita pri rovnakom počte vyhodnotení účelovej funkcie v porovnaní s OEA, z čoho vyplýva významná úspora výpočtového času (aj pri použití jedného procesora),
4. schopnosť nájsť lepšie riešenia, schopnosť uvoľniť sa zo stagnácie v lokálnom extréme a smerovať ku globálnemu extrému,
5. možnosť distribúcie výpočtového výkonu na viac procesorov.

V predloženej kapitole sme predstavili princíp aj hľadiská klasifikácie PEA. Naznačili sme, ako je ich možné parametrizovať. Okrem toho sme predstavili princíp reinicializácie, čo je efektívny nástroj pri riešení členitých účelových funkcií. Uvedené metódy a algoritmy sme demonštrovali na príkladoch minimalizácie vybraných testovacích funkcií. Z vyhodnotení experimentov vyplývajú viaceré poznatky. PEA konvergujú rýchlejšie, než SGA o rovnakom celkovom počte jedincov vďaka väčšej genetickej diverzite v jednotlivých subpopuláciách. Pri optimalizácii členitých účelových funkcii je prínosom, keď jednotlivé subpopulácie PEA majú rôzne charakteristiky (rôzne miery mutácie, rôzny selektívny tlak). Pri vrstvových štruktúrach hrubozrnných PEA (Obr. 9 c,e,f) je vhodné, ak má vrchná, prípadne stredná vrstva nižšiu diverzitu a vyšší selektívny tlak a v spodnej vrstve je to naopak.

V spodnej vrstve je vhodné tiež aplikovať reinicializáciu. Subpopulácie spodných vrstiev generujú nové smery hľadania, horná vrstva sa upriamuje na spresnenie polohy globálneho extrému. Pri voľbe topológie PEA sa nepodarilo nájsť jednu univerzálne najúspešnejšiu topológiu pre všetky typy úloh. Voľba vhodnej topológie závisí od konkrétnej aplikácie.

PEA je vhodné použiť predovšetkým v prípadoch členitých multimodálnych účelových funkcií, kde je problematické nájsť globálny extrém. Pri málo členitých alebo dokonca unimodálnych úlohách (ako sú z matematických funkcií napr. n-rozmerná kvadratická funkcia alebo Schwefelova funkcia) sa výhody PEA oproti OEA stierajú, dokonca PEA môžu byť aj mierne menej efektívne.

Použitím PEA pri riešení zložitých praktických optimalizačných úloh na našom pracovisku bolo možné získať jednak kvalitnejšie riešenia a súčasne dosiahnuť redukcie výpočtového času o desiatky percent v porovnaní s jednoduchými EA, resp. GA. V niektorých prípadoch sa podarilo dosiahnuť redukciu výpočtového času až na desatinu pôvodného výpočtového času (na rovnakom počte procesorov).

***Poďakovanie**:* Táto kapitola vznikla za podpory grantovej agentúry SR VEGA v rámci grantovej úlohy 1/0690/09.

Literatúra

[1] Alba E., Tomassini M.: *Parallelism and Evolutionary Algorithms*. IEEE Trans. On Evolutionary Computation, Vol. 6, NO.5, October 2002

[2] Bäck T.: Selective Pressure in Evolutionary Algorithms: A Characterization of Selection Mechanisms. IEEE Conf. on Evolutionary Computing, Florida, USA, 1994.

[3] Cantú-Paz E.: A Summary of Research on Parallel Genetic Algorithms, IlliGAL Report No. 95007, July 1995

[4] Cantú-Paz E.: Migration Policies, Selection Pressure, and Parallel Evolutionary Algorithms, IlliGAL Report, 1995 or Springer, 2001

[5] Eiben A.E., Smith J.E.: *Introduction to evolutionary computing*, Springer, Berlin, 2003.

[6] Giacobini M., Tomassini M., Tettamanzi A.G.B., Alba E.: *Selection intensity in cellular evolutionary algorithms for regular lattices*, IEEE Transactions on Evolutionary Computation, 2005

[7] Goldberg, D. E.: Genetic Algorithms in Search, Optimisation and Machine Learning, Addisson-Wesley, 1989

[8] Kajan S., Sekaj I., Oravec M.: *Realizácia paralelných genetických algoritmov v Matlabe pomocou Parallel Computing Toolboxu*. In Technical Computing Prague 2008 : 16th Annual Conference Proceedings. Prague, Czech Republic, 11.11.2008. Praha: Humusoft s.r.o, 2008

[9] Kvasnička V., Pospíchal J., Tiňo P.: *Evolučné algoritmy*, Vydavateľstvo STU, Bratislava, 2000.

[10] Lacevic B., Konmjicija S., Avdagic Z.: *Population Diversity Measure Based on Singular Values of the Distance Matrix*, CEC, Singapore, 2007

[11] Lin S.Ch., Punch W., Goodman E.: *Coarse-grain parallel genetic algorithms: Categorization and new approach*, IEEE Symposium on Parallel and Distributed Processing, 1994

[12] Michalewicz Z.: *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd ed. Springer-Verlag, Berlin Heidelberg New York 1996.

[13] Nowostawski M., Poli R.: *Parallel Genetic Algorithm Taxonomy*, KES'99, May 1999

[14] Oravec, M.: *Evolučné algoritmy*, Diplová práca, FEI STU, Bratislava, 2008

[15] Sekaj I.: *Robust Parallel Genetic Algorithms with Re-Initialisation*, in PPSN VIII, September 18-22, Birmingham, 2004

[16] Sekaj I.: *Evolučné výpočty*, IRIS, Bratislava, 2005

[17] Sekaj I., Perkacz J.: *Some Aspects of Parallel Genetic Algortims with Population Re-initialization*, CEC, Singapore, 2007

[18] Sekaj I., Oravec M., Uchaľ P.: *Selected Population Characteristics for Evolutionary Algorithms.* In Proceedings on the Mendel 2008: 14th International Conference on Soft Computing, Brno, Czech Republic, 18.-20.6.2008.

[19] Sekaj, I., Oravec, M.: *Selected Population Characteristics of Fine-grained Parallel Genetic Algorithms with Re-initialisation,* Proceedings of the GEC 2009, Shanghai, 2009

[20] Skolicki Z., DeJong K.: *The influence of migration sizes and intervals on island models*, GECCO, Washington, USA, 2005

[21] Whitley D.: *A genetic algorithm tutorial*, Statistics and Computing **4**, 1994, pp. 65-85.

[22] Whitley D., Rana S., Heckendorn R. B.: *The island model genetic algorithm: On separability, population size and convergence*, in Journal of Computing and Information Technology, 7(1), 1999, pp. 33-47.

[23] Zelinka, I., Oplatková, Z., Šeda, M., Ošmera, P., Včelař, F.: *Evoluční výpočetní techniky - Principy a aplikace*, BEN, Praha, 2009.

Membránové výpočty jako metafora informačních toků v organismech

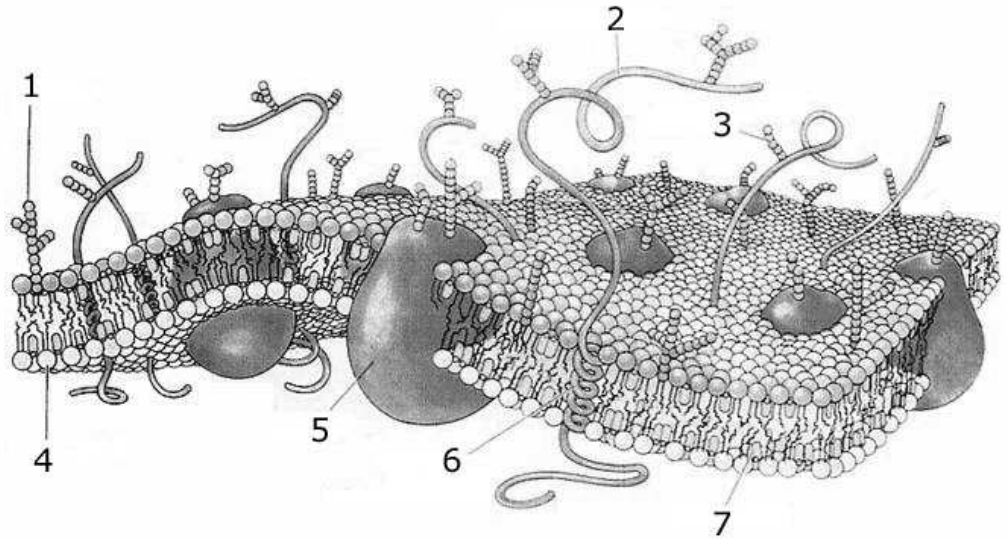
Petr SOSÍK[1]

Abstrakt. Membránové systémy představují výpočetní metaforu chování živých buněk, provázanou s oblastmi biovýpočtů, umělého života a syntetické biologie. Klíčovými operacemi v membránových systémech je regulovaný průchod molekul membránami, buněčné dělení, rozpad membrán, přenos informací po nervových vláknech a podobně. Všechny tyto operace pracují s abstraktními atomickými objekty a teorie membránových systémů je zkoumá z hlediska zpracování informací. Studuje mimo jiné, jak různé kombinace těchto operací ovlivňují výpočetní potenciál takto matematicky abstrahované buňky či buněčného systému. V této kapitole budeme zkoumat vlastnosti komunikujících membránových systémů, systémů s aktivními membránami a pulsních neuronových P systémů. Ukážeme například, jakou roli hraje v tomto abstraktním modelu buňky synchronizace, polarizace membrán či paralelismus, budeme studovat sílu evolučních posloupností individuí a důležitou roli (bio-)diverzity.

1 Úvod

Model membránového systému vznikl před rokem 2000 jako součást výzkumu v oblasti tzv. biovýpočtů, tedy pokusů o konstrukci výpočetních mechanismů či automatů na bázi organické hmoty. Prvním publikovaným časopiseckým článkem o membránových systémech je [31]. Jde o značně abstraktní matematický model inspirovaný chováním buněčných membrán. Jeho klíčovým prvkem jsou selektivně propustné membrány, propouštějící pouze určité molekuly, případně pouze určitým směrem, v určitých kombinacích, za určitých podmínek a podobně. Membrány mohou obsahovat další membrány, odtud pramení název membránový systém, později začal být užíván název P systém. Veškeré objekty v membránovém systému mají diskrétní povahu — jsou reprezentovány atomickými symboly, popsanými symboly konečné abecedy. Postupně vznikla řada variant P systémů obohacených o další vlastnosti jako možnost dělení membrán nebo naopak jejich proděravění a rozpadu, polarizaci membrán, spojování membrán do tkání, regulaci transportu membránovými

[1] Ústav informatiky, Filozoficko-přírodovědecká fakulta, Slezská univerzita v Opavě, 74601 Opava, E-mail: petr.sosik@fpf.slu.cz



Obr. 1. Schéma buněčné membrány: 1–glykolipid; 2,6–alfa-helix protein; 3–oligosacharidový boční řetězec; 4–fosfolipid; 5–globulární protein; 7–cholesterol. Převzato z Wikipedie, držitel práv: National Institute of Standards and Technology (NIST).

proteiny a podobně. Specifickou variantou jsou tzv. pulsní neuronové P systémy, inspirované pulsním modelem neuronu a jeho chováním.

Na obor membránových systémů je možno nahlížet z několika úhlů. Originální motivace a snahy sestrojit buněčný počítač či automat na bázi těchto modelů dosud nepřinesly hmatatelné výsledky (rep. jsou v začátcích). Podobně se vedlo i dalším modelům počítání podle přírody, jako jsou DNA výpočty nebo kvantové výpočty. Proto je tato jejich motivace a interpretace dnes postupně opouštěna a přesouvá se, např. v případě DNA výpočtů, spíše do roviny nanomateriálů, automatů a nanotechnologií.

Jinou možností je užívat membránové systémy k modelování dějů v buňkách. Modelování a studium biologických systémů z hlediska jejich informačních toků a organizace se ukazuje být klíčovým pro pochopení jejich funkcí. Přitom pozorujeme postupný posun od spojitých modelů pracujících s abstraktními veličinami jako koncentrace, reakční kinetika a podobně, k modelům diskrétním popisujícím chování každé jednotlivé buňky či dokonce molekuly. Tyto modely pracují buďto ve spojitém čase (jako tzv. modely IPS - Interacting Particle Systems [19]) nebo v diskrétním čase, jako celulární automaty nebo právě membránové systémy. Existují slibné výsledky využívající membránových systémů k modelování například regulačních a signálních kaskád v buňkách [17], růstu nádorů [23, 35], quorum sensing [34] a dalších buněčných procesů.

Konečně je možno použít membránové systémy ke zkoumání informačního potenciálu dílčích procesů v buňkách, a jako takové by mohly být pomocnou disciplínou systémové nebo syntetické biologie. Známý je výrok klasika současné evoluční teorie Richarda Dawkinse: Chcete-li pochopit život, nemyslete na vibrující

gel a chvějící se sliz, přemýšlejte o informační technologii. Tím je míněna technologie digitální, jež je základem přenosu genetické informace i genové exprese.

V této kapitole představíme několik modelů membránových systémů, z nichž každý vládne rozdílným repertoárem abstraktních bio-inspirovaných operací. Různé kombinace a varianty těchto operací vedou k odlišné výpočetní (obecněji informační) síle výsledného systému. Záměrně vynecháváme většinu formálních důkazů a odkazujeme čtenáře na odbornou literaturu. Při studiu detailů v této literatuře si čtenář oprávněně pomyslí, že způsob, jakým jsou důkazy vedeny, je z pohledu biologa absurdní. Uvažuje totiž propojování nebo programování dílčích operací bez jakýchkoli omezení. Organismus je naproti tomu složitý systém s celou řadou omezení nutných pro zachování jeho života. Je tedy třeba si uvědomit, že membránové systémy dovolují pouze získání horního odhadu toho, jaké jsou regulační a informační schopnosti daného systému, anebo, nahlíženo z druhé strany, jaké operace minimálně potřebujeme, chceme-li docílit určité komplexnosti chování. K tomu, jak jsou dílčí operace v realitě propojeny, je zapotřebí jiných přístupů.

Kapitola je uspořádána takto: v části 3 se zabýváme komunikujícími membránovými systémy, založenými na operaci synchronizovaného přenosu neměnných signálních částic membránami (symport/antiport). Bude nás zajímat zejména vliv synchronizace, paralelismu a determinismu na jejich schopnost zpracování informací. V části 4 jsou představeny P systémy s aktivními membránami, vybavené operacemi jako je buněčné dělení, rozpad membrán, polarizace membrán a samozřejmě opět průchod částic membránami. Konečně část 5 je věnována pulsním neuronovým P systémům, inspirovaným přenosem informace v biologických neuronových sítích. Zde se budeme zabývat dílčími operacemi uvnitř neuronů, jako je relaxační doba, konzumace pulsů nebo zpracování signálu v neuronech pomocí regulárních výrazů.

2 Základní definice

V této části shrneme některé standardní matematické definice používané ve zbytku kapitoly. Laskavý čtenář se může dopátrat podrobnějších informací například v [28]. Následující text předpokládá elementární znalosti a konvence z teorie formálních jazyků a automatů na úrovni bakalářského studia, viz například [18].

Označme V nějakou konečnou množinu symbolů. Zápis V^* značí množinu všech řetězců (slov) složených ze symbolů z V. Je-li $w \in V^*$ slovo, pak $|w|$ označuje jeho délku, tedy počet symbolů, z nichž se skládá. Prázdné slovo délky 0 je označeno λ. Označme dále $V^+ = V^* \setminus \lambda$ množinu všech neprázdných slov nad V.

Nechť N je množina všech přirozených čísel včetně 0 a $\mathbb{N} \cdot \mathrm{RE}$ označuje množinu rekurzivně spočetných přirozených čísel. Nechť V je konečná množina objektů. Pak $card(V)$ označuje její *kardinalitu*, tzn. počet prvků ve V.

Multimnožina (nad V) je funkce $M: V \rightarrow N \cup \{+\infty\}$; pro $a \in V$, $M(a)$ definuje *násobnost* a v multimnožině M. Budeme říkat, že element a multimnožiny M má *nekonečnou násobnost*, jestliže $M(a)=+\infty$. Kardinalita multimnožiny je číslo $card(M)=\sum_{a \in V} M(a)$.

Support multimnožiny M je množina $supp(M)=\{a\in V \mid M(a)>0\}$. Říkáme, že element *a patří* do multimnožiny M (označujeme $a\in M$), jestliže $a\in supp(M)$.

Symbolem φ označujeme *prázdnou multimnožinu*, t.j., takovou, jejíž support je prázdná množina ∅. Konečné multimnožiny jsou takové, které obsahují objekty jen s konečnou násobností.

Nechť M_1,M_2 jsou dvě multimnožiny. *Sjednocením* M_1 a M_2 je multimnožina $M_1\cup M_2$ definovaná $(M_1\cup M_2)(a)=M_1(a)+M_2(a)$, pro každé $a\in V$.

Rozdíl $M_1\backslash M_2$, dvou multimnožin takových, že $M_1(a)\geq M_2(a)$ pro všechny $a\in V$, je definován jako $(M_1\backslash M_2)(a)=M_1(a)-M_2(a)$ pro všechny $a\in V$. Samozřejmě, jestliže $M_1(a)=+\infty$ a $M_2(a)$ je konečná, tak $M_1(a)\backslash M_2(a)=+\infty$. Jestliže $M_2(a)=+\infty$, potom $M_1(a)\backslash M_2(a)=0$.

2.1 Multipočitadlový stroj

Multipočitadlový stroj (představený v [24] jako *stroj s programem*) je stroj opatřený několika počitadly, z nichž každé může obsahovat libovolné přirozené číslo. Stroj vykonává program složený z instrukcí, z nichž každá je označena unikátním návěštím (jako čísla řádků v jazyce Basic). Instrukce mají následující tvar:

l_i:(ADD(r),l_j,l_k) Stav počitadla r je zvýšen o 1 a výpočet pokračuje náhodným výběrem ze dvou instrukcí s návěštím l_j nebo l_k. V případě deterministického stroje platí $l_j=l_k$ a instrukce má tvar l_i:($ADD(r),l_j$).

l_i:(SUB(r),l_j,l_k) Jestliže počitadlo r není nulové, instrukce odečte z jeho obsahu 1 a pokračuje instrukcí označenou l_j, jinak program pokračuje instrukcí l_k.

l_h:*HALT* Stroj se zastaví.

Deterministický multipočitadlový stroj pracuje se vstupem $m\in N$, který je na začátku výpočtu vložen do vstupního počitadla č. 1. Vstup m je akceptován tehdy, když program skončí instrukcí HALT a stroj má všechna počitadla prázdná. V [24] je ukázáno, že multipočitadlový stroj se třemi počitadly může realizovat libovolný výpočet proveditelný na Turingově stroji nebo na konvenčních počítačích. Pokud umožníme speciální zakódování vstupu, postačí dokonce dvě počitadla.

2.2 Částečně slepý multipočitadlový stroj

Částečně slepý multipočitadlový stroj (PBCM) byl zaveden v [12]. Liší se instrukcí *SUB* ve tvaru l_i:($SUB(r),l_j$). Vstup, na začátku uložený ve vstupním počitadle, je akceptován, pokud stroj vykoná instrukci HALT a všechna počitadla jsou prázdná. Pokud se stroj pokusí snížit hodnotu počitadla obsahujícího nulu, dojde k zastavení výpočtu a vstup není akceptován. V [12] se uvádí, že problém prázdnosti pro PBCM je rozhodnutelný a že PBCM akceptují vlastní podmnožinu $N\cdot$RE, nemají tedy univerzální výpočetní sílu. Třída množin přijímaných PBCM je označena $N\cdot$PBCM.

2.3 Regulární výrazy

Definice 1 *Buď V konečná abeceda symbolů. Pak (i) prázdné slovo λ a každý symbol $a \in V$ jsou regulární výrazy, (ii) pokud E_1, E_2 jsou regulární výrazy, pak rovněž $(E_1) \cup (E_2), (E_1)(E_2)$, a $(E_1)^*$ jsou regulární výrazy, (iii) toto a pouze toto jsou regulární výrazy nad V.*

V definici nahoře představuje $(E_1)^*$ libovolný (i nulový) počet zřetězení výrazu E_1. Operátor zřetězení $\cdot$ a nadbytečné závorky bývají často vynechávány. Každý regulární výraz E zastupuje jistou množinu řetězců nad V, kterou označujeme $L(E)$. Například výraz $(a \cup aa)(aaa)^*$ reprezentuje všechny řetězce složené ze symbolů a, jejichž délka po dělení třemi dá zbytek 1 nebo 2. Dva regulární výrazy E_1 a E_2 nazveme *ekvivalentní* pokud $L(E_1)=L(E_2)$.

Definice 2 (regular_cf) *Regulární výraz $E=E_1 \cup \ldots \cup E_n$ (kde všechna E_i obsahují pouze operátory $\cdot$ a *) je v* jednohvězdičkové normální formě *(JNF), pokud $\forall i \in \{1,\ldots,n\}$, E_i obsahuje nejvýše jeden výskyt *.*

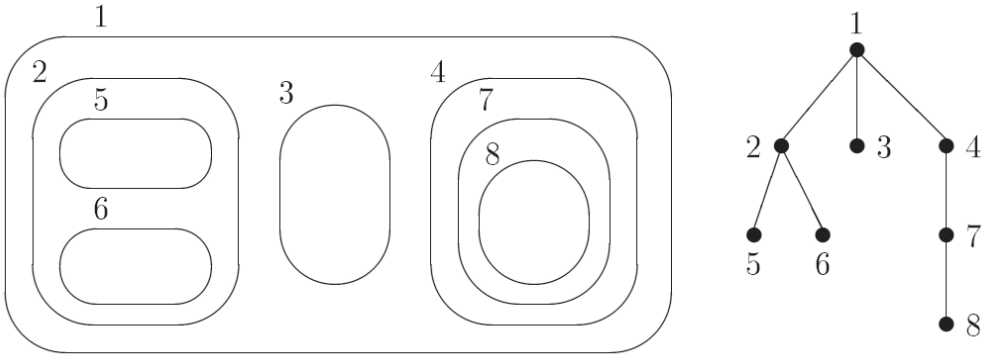
Každý regulární výraz může být transformován na ekvivalentní výraz v jednohvězdičkové normální formě. Tato transformace však může způsobit, že délka regulárního výrazu exponenciálně vzroste.

3 Komunikující membránové systémy

Komunikující P systémy (CPS) byly zavedeny v [38] jako varianta P systémů pracujících s atomickými objekty reprezentovanými symboly. Jak jsme se již zmínili, hlavní složkou membránového systému je vnořený systém membrán, znázorněný Vennovým diagramem nebo případně grafem – stromem, jež odpovídá vnoření jednotlivých membrán (viz obr. 2, převzatý z práce [33]). Zkráceně tuto membránovou strukturu popíšeme dobře uzávorkovaným výrazem:

$$\mu=[_1[_2[_5\,]_5[_6\,]_6]_2[_3\,]_3[_4[_7[_8\,]_8]_7]_4]_1.$$

Tento typ struktury je inspirován existencí organel v buňkách, které mají vlastní membránu. Je zřejmé, že větší počet úrovní vnoření je abstrakcí, která stěží hledá oporu v biologické realitě. Na druhé straně větší hloubka vnoření hraje roli pouze pro některé z níže uvedených výsledků, zpravidla stačí hloubka vnoření 2. Vnější membrána, která je vždy jediná, se nazývá obal nebo hlavní membrána. Membrána, jež neobsahuje další vnořené membrány, se nazývá elementární.



Obr. 2. Membránová struktura a jí odpovídající strom.

Membrána je označena typovým indexem z konečné množiny typů H. Více různých membrán může být téhož typu. V regionu obklopeném membránou se nacházejí atomické objekty reprezentované symboly konečné abecedy V. Každý objekt může být přítomen v libovolném počtu kopií. Objekty mohou participovat v různých typech operací, jimiž se odlišují varianty P systémů. V případě CPS jde výlučně o operace přenášející objekty skrze membrány, aniž by docházelo k jejich změně.

Komunikující membránový systém stupně n, $n \geq 1$, je formálně definován jako struktura

$$\Pi=(V,\mu,env,E_i,R_i,i_0),\ 1 \leq i \leq n,$$

kde:

V je konečná, neprázdná abeceda, kde prvky nazýváme *objekty*;

μ je vnořená membránová struktura stávající se z n membrán. Membrána 1 představuje *obal.*

$env: V \rightarrow +\infty$ je multimnožina objektů přítomných na začátku výpočtu v okolním prostředí v neomezeném počtu kopií;

$E_i: V \rightarrow N, 1 \leq i \leq n$ jsou multimnožiny objektů nad V;

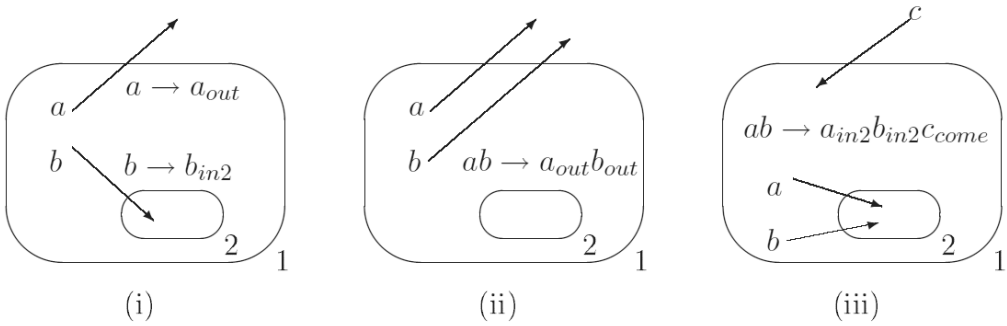
$R_i, 1 \leq i \leq n$ jsou množiny pravidel ve tvaru:

(i) a_τ (jednoduchý přesun objektu membránou);

(ii) $ab_{\tau 1}b_{\tau 2}$ (synchronní přesun dvou objektů – symport);

(iii) $ab_{\tau 1}b_{\tau 2}c_{come}$ (přesun objektu z vnějšího prostředí dovnitř membránového systému);
kde $a,b,c \in V$ a $\tau,\tau_1,\tau_2 \in \{here,out\} \cup \{in_k | 2 \leq k \leq n\}$.

$i_0 \in \{1,\ldots,n\}$ je *vstupní membrána*.



Obr. 3. Ilustrace použití pravidel typu (i), (ii) a (iii) komunikujícího membránového systému.

Konfigurace CPS stupně n je definována jako $(F_{env},F_1,\ldots,F_n)$, kde F_{env} a $F_k, 1\leq k \leq n$, jsou multimnožiny objektů z V. *Počáteční* konfigurace CPS je označena $(env,E_1,\ldots,E_n)$.

Výpočetní krok CPS je přechod mezi dvěma konfiguracemi $(F_{env},F_1,\ldots,F_n)\Rightarrow(F'_{env}, F'_1,\ldots,F'_n)$ pomocí pravidel. Objekty jsou při aplikaci pravidla přesunuty do membrán určených dolními indexy na pravé straně pravidla. Jestliže u některého objektu je dolní index *here* nebo je vynechán, znamená to, že objekt zůstává v původní membráně. Pravidla typu $ab_{\tau_1}b_{\tau_2}c_{come}$, $\tau_1,\tau_2\in\{out\}\cup\{in_k|2\leq k \leq n\}$ mohou být asociována pouze s hlavní membránou. Situaci ilustruje obr. 3. Pravidla mohou být užívána v jednom ze tří základních módů:

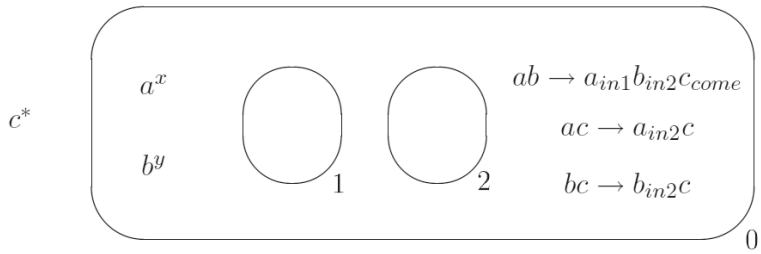
- V *maximálním paralelním módu* je aplikována současně v každé membráně maximální multimnožina aplikovatelných pravidel, tj. taková, že každé pravidlo má pro sebe vyhrazené symboly přítomné v membráně nebo v okolním prostředí, s nimiž bude pracovat (a s nimiž nepracuje žádné jiné pravidlo), a současně už nelze žádné pravidlo přidat. Pokud v membráně existuje více než jedna taková maximální multimnožina (všimněte si, že různé maximální multimnožiny mohou mít různou velikost), tak se jedna z nich náhodně vybere.

- V *minimálním paralelním módu* je použito nejméně jedno ze všech aplikovatelných pravidel v každé membráně.

- V *sekvenčním módu* je aplikováno právě jedno nedeterministicky vybrané pravidlo.

Výpočtem CPS Π nazveme posloupnost výpočetních kroků začínající počáteční konfigurací a probíhající tak dlouho, dokud existují v systému použitelná pravidla. Abychom definovali výsledek výpočtu, mohou být membránové systémy vybaveny vstupní a výstupní membránou. Do vstupní membrány i_0 může být na začátku výpočtu vložena vstupní multimnožina objektů E_{i0}. Systém pak může například vygenerovat v určené membráně nějakou výstupní multimnožinu F_{out}. Anebo může vstupní

membrána chybět a membránový systém vygeneruje nedeterministicky do výstupní membrány nějakou multimnožinu objektů, pak říkáme, že jde o *generující* membránový systém. Anebo naopak chybí výstupní membrána a výstupem je pouze fakt, zda se výpočet zastaví nebo ne. Pak jde o *akceptující* membránový systém, inspirovaný chováním formálních automatů, jako je Turingův stroj.



Obr. 4. Příklad komunikujícího membránového systému počítajícího funkci maxima.

Příklad: Uvažujme CPS na obr. 4. Nechť hlavní membrána je současně vstupní a vstupem jsou hodnoty x a y, reprezentované počty objektů a a b. Předpokládejme že $x>0$ a $y>0$ a že systém pracuje v maximálně paralelním módu. V prvním kroku systém umístí do membrán 1 i 2 hodnotu $\min\{x,y\}$ reprezentovanou počtem objektů a, resp. b. Současně je z vnějšího prostředí importován stejný počet objektů c. V hlavní membráně nyní mohou být kromě objektů c pouze objekty a nebo b, ale ne oba typy současně. V dalších krocích jsou pak tyto objekty a nebo b postupně přemisťovány do membrány 2. Když se výpočet zastaví, počet objektů v membráně 1 je $\min\{x,y\}$ a v membráně 2 $\max\{x,y\}$. Systém je deterministický – v každém kroku lze aplikovat jedinečnou multimnožinu pravidel (viz níže).

Pokud by systém pracoval v minimálně paralelním anebo sekvenčním módu, stal by se nedeterministickým. Po zastavení výpočtu by počet objektů v membráně 1 byl $z\in\langle 1,\min\{x,y\}\rangle$ a v membráně 2 pak $x+y-z$.

Dále budeme pracovat s akceptujícími CPS. Předpokládejme, že výpočet začínající v konfiguraci (env,E1,…,En) skončí v konfiguraci (Fenv,F1,…,Fn), kdy už není žádné aplikovatelné pravidlo. Pak říkáme, že Π akceptuje číslo card(Ei0). Pokud by se výpočet nikdy nezastavil, číslo card(Ei0) není akceptováno. Množinu přirozených čísel akceptovaných CPS Π označíme N(Π). Třídu množin přirozených čísel akceptovaných všemi CPS stupně $n\geq 1$ označíme NCPAn(mode), kde mode∈{maxpar,minpar,seq} pro CPS pracující v maximálním paralelním, minimálním paralelním nebo sekvenčním módu.

CPS nazýváme deterministický, jestliže daná počáteční konfigurace, posloupnost konfigurací je jedinečná, tzn., v každém kroku výpočtu je multimnožina aplikovatelných pravidel (vzhledem k vybranému módu výpočtu) jedinečná. V případě minimálního paralelismu z toho vyplývá, že v každé membráně je nejvýše jedno

aplikovatelné pravidlo. Označme NCPAn(mode,det) třídu množin při¬ro¬ze¬ných čísel akceptovaných deterministickými CPS stupně n.

CPS nazýváme *k*-deterministický, $k \geq 1$, jestliže je deterministický a v každém kroku má multimnožina aplikovatelných pravidel (v celém systému) nejvýše k prvků. Všimněme si, že každý 1-deterministický CPS pracuje de facto v sekvenčním módu (ačkoli formálně může užívat minimální nebo maximální paralelní mód), protože v každém kroku může aplikovat pouze jedno pravidlo, takže svůj případný paralelismus nemůže nijak využít. Proto každá množina akceptovaná pomocí 1-deterministického CPS může být akceptována i CPS pracujícím v sekvenčním módu. Neplatí to však obráceně, protože sekvenční CPS může být nedeterministický - může mít na výběr z více pravidel.

Následující výsledky dovolují porovnat výpočetní sílu komunikujících P systémů pracujících v různých módech, a také si všimnout vlivu determinismu. Začneme výsledkem, který ukazuje univerzální výpočetní sílu 2-deterministických komunikujících P systémů, přičemž k jejímu dosažení postačí systém s dvěma membránami.

Věta 3.1 ([8]) $NCPA_k(maxpar,det)=NCPA_k(minpar,det)=N\cdot RE,\ k\geq 2.$

Důkaz prezentovaný v [8] je založen na konstrukci akceptujícího CPS stupně 2, který simuluje činnost deterministického multipočitadlového stroje. Každé počitadlo je reprezentováno specifickým typem objektů, jejichž počet uvnitř systému odpovídá hodnotě počitadla. Pokud se hodnoty počitadla mění, slouží vnější prostor obklopující systém jako odkladiště a zásobárna neomezeného počtu těchto objektů. Pro každou instrukci v simulovaném programu máme v systému speciální objekty a pravidla, která zajistí, že stav počitadel reprezentovaných objekty se mění právě tak, jak to vyžaduje simulovaná instrukce. Takto ukážeme, že $N\cdot RE \subseteq NCPAk(maxpar,det)$ a $N\cdot RE \subseteq NCPAk(minpar,det)$. Obrácené inkluze jsou pak důsledkem Turingovy-Churchovy teze a věta je dokázána.

Situace se změní, když přejdeme od minimálního nebo maximálního paralelního módu k módu sekvenčnímu. V tomto případě se výpočetní síla systému sníží na úroveň částečně slepého počitadlového stroje.

Věta 3.2 ([8]) *Třída NCPAn(seq),n≥2 je ekvivalentní třídě množin čísel akceptovaných pomocí PBCM.*

Důkaz je proveden pomocí dvojí vzájemné simulace sekvenčního komunikujícího P systému a částečně slepého počitadlového stroje. Když budeme navíc požadovat, aby systém byl 1-deterministický (a tím současně zajistíme, že systém bude pracovat sekvenčně), dojde k dramatickému poklesu jeho výpočetní síly.

Věta 3.3 ([8]) *Každá množina přirozených čísel akceptovaná 1-deterministickým CPS má jeden z následujících tvarů* $\emptyset$, N, $\{0,1,\ldots,n\}$, $n\geq 0$.

Důkaz. Čtenář si jistě dokáže představit triviální 1-deterministický CPS, který akceptuje prázdnou množinu možných vstupů (nikdy se nezastaví, bez ohledu na to,

jaký vstup dostane), stejně tak i ten, který akceptuje N (zastaví se vždy bez ohledu na to, jaký vstup dostane).

Ukážeme konstrukci CPS akceptujícího množinu $\{0,1,\ldots,n\}$, $n\geq 0$. Bude jím systém $\Pi=(V,[_1]_1,env,E_1,R_1,1)$, kde $V=\{a,b_1,\ldots,b_{n+1}\}$, $supp(env)=\{b_2,\ldots,b_{n+1}\}$, $E_1=\{a^m,b_1\}$. Pravidla patřící do R_1 jsou: $ab_i\rightarrow a_{out}b_{i,out}b_{i+1,come}$ pro $1\leq i\leq n$, a $ab_{i+1}\rightarrow ab_{i+1}$ Systém pracuje tak, že postupně v n krocích vypuzuje do okolního prostředí vstupní objekty a. Pokud se membrána vyprázdní během n kroků, systém se zastaví. Pokud po n krocích v membráně stále ještě objekty a jsou, znamená to, že vstupní hodnota byla $m>n$ a systém se díky poslednímu pravidlu dostane do nekonečného cyklu, nezastaví se a vstup neakceptuje.

Dokážeme dále, že 1-deterministické CPS nemohou akceptovat nic víc, než množiny uvedené v tvrzení věty. Stačí, když dokážeme, že pro libovolné $n\geq 1$ a pro 1-deterministický CPS Π, $n\in N(\Pi)$ implikuje $(n-1)\in N(\Pi.)$ Pak v důsledku matematické indukce má přijímaná množina tvar $\{0,1,\ldots,n\}$, $n\geq 0$. Proto tedy budeme předpokládat, že $n\in N(\Pi)$ pro nějaký 1-deterministický CPS Π stupně $k\geq 1$. Nechť i_0 je vstupní membránou Π a nechť vstup Π je prezentován počtem objektů a obsažených v i_0 v počáteční konfiguraci. Označme pomocí $left(r)$ multimnožinu objektů na levé straně pravidla r. Nyní uvažujme pravidla aplikovatelná v i_0 během výpočtu Π se vstupem n, která obsahují na svých pravých stranách objekt a.

(i) Nechť v systému nejsou taková pravidla, anebo nechť každé takové pravidlo r je aplikovatelné v konfiguraci $(F_{env},F_1,\ldots,F_k)$ takové, že $F_{i0}(a)>left(r)(a)$. Π se vstupem $n-1$ prochází stejnou posloupností konfigurací jako se vstupem n, výjimkou, že počet a v membráně i_0 v každé konfiguraci je snížena o jedničku. Nicméně je aplikována stejná posloupnost pravidel, a proto se Π zastaví a akceptuje $n-1$.

(ii) Nechť je aplikováno pravidlo r v konfiguraci $(F_{env},F_1,\ldots,F_k)$ takové, že $F_{i0}(a)=left(r)(a)$. Zároveň nechť $(F_{env},F_1,\ldots,F_k)$ je první dosažená konfigurace, kde platí rovnost $F_{i0}(a)=left(r)(a)$.

Π se vstupem $n-1$ dosáhne analogické konfigurace, ale s $F_{i0}'=F_{i0}\setminus\{a\}$. Tudíž pravidlo r nemůže být aplikováno, daný 1-deterministický systém Π nemá žádné další aplikovatelné pravidlo a zastaví se. V důsledku toho Π vždy akceptuje $n-1$.

Důsledek 3.4 *$NCPA_k(seq,det)=NCPA_1(minpar,det)=\{\{0,1,\ldots,n\}\mid n\geq 0\}\cup\{\emptyset,N\}$, pro libovolné $k\geq 1$.*

Důkaz. Každý deterministický CPS stupně 1 pracující v minimálním paralelním módu, nebo deterministický CPS libovolného stupně pracujícím v sekvenčním módu je nutně podle definice 1-deterministický.

Naše poslední studie v této části textu se bude týkat nedeterministických CPS pracujících v maximálně paralelním módu, avšak s jedinou membránou. Uvolníme tedy všechna dosud uvažovaná omezení na činnost systému, a zavedeme místo nich jediné - systém může mít pouze jednu membránu. I takto omezený systém zvládne simulovat multipočitadlový stroj, a důsledkem je následující tvrzení.

Věta 3.5 ([8]) $NCPA_k(maxpar)=N\cdot\mathrm{RE},\ k\geq 1.$

Na závěr naší rozpravy o komunikujících P systémech shrneme získané výsledky. Pokud chceme CPS, který zvládá univerzální výpočty (tedy je schopen realizovat jakýkoli algoritmus z konvenčních počítačů, byť o jeho výpočetní efektivnosti jsme zatím nemluvili), je třeba buďto připustit maximální nebo minimální paralelismus a dát systému nejméně dvě membrány. Pak je schopen vykonat libovolný výpočet, a to deterministickým i nedeterministickým způsobem. Přitom k tomu stačí, aby paralelismus byl omezen na dvě současně prováděné operace v každém kroku. A v případě maximálního paralelismu a nedeterminismu stačí dokonce jedna membrána. Sílu deterministických CPS stupně 1 pracující v maximálním paralelním módu zatím neznáme.

Jakmile ze systému odebereme možnost paralelních výpočtů (neboli, jinými slovy, možnost synchronizace operací), klesne okamžitě jeho výpočetní síla na úroveň částečně slepého počitadlového stroje, pokud počet membrán je alespoň dvě. I to je však ještě poměrně velká síla postačující k řešení řady problémů. Konečně, když budeme navíc k požadavku sekvenčního výpočtu žádat determinismus, dostaneme systém, který je schopen pouze triviálních výpočtů, dokáže de facto pouze rozlišit, zda vstupní hodnota přesahuje určitou mez nebo ne.

Shora uvedené výsledky jsou shrnuty v tab. 1. Co se výpočetní efektivity komunikujících membránových systémů týče, dodejme, že byly například sestrojeny CPS, které mohou v lineárním čase simulovat optimální paralelní algoritmus výpočtu maxima [39] a výpočtu prefixové sumy.

4 P systémy s aktivními membránami

Název aktivní membrány signalizuje, že u těchto typů membránových systémů mohou membrány během výpočtu měnit své vlastnosti. A to buďto měnit svou polarizaci, nebo se za jistých podmínek může membrána rozdělit na dva potomky. Membrána může být polarizována kladně, záporně nebo neutrálně. Tyto tři případy budeme označovat jako $[_i\ ]_i^+, [_i\ ]_i^-, [_i\ ]_i^0$. Membrána je dále označena typovým indexem z konečné množiny typů H. Více různých membrán může být téhož typu. Stejně jako u komunikujících membránových systémů se v regionu obklopeném membránou nacházejí atomické *objekty* reprezentované symboly konečné abecedy V. Na rozdíl od

Tabulka 1. Třídy množin přijímaných akceptujícími CP systémy. Počet membrán je k. Zkratka Triv. označuje množiny v jednom ze tvarů ∅, N, {0,1,…,n}, n≥0.

	k=1 *det*	k=1 *nondet*	k≥2 *det*	k≥2 *nondet*
seq	*Triv.*	?	*Triv.*	N·PBCM
minpar	*Triv.*	?	N·RE	N·RE
maxpar	?	N·RE	N·RE	N·RE

CP systémů se však mohou objekty během výpočtu měnit v jiné anebo zanikat. Objekty mohou participovat v následujících typech operací (všimněte si, že tvar zápisu současně znázorňuje, co operace dělá):

(a) $[_h a]_h^{\alpha}$,

kde $h \in H, \alpha \in \{+,-,0\}, a \in V, v \in V^*$

(*evoluce objektu:* objekt a evolvuje v membráně typu h s polarizací α, výsledkem je obecně multimnožina objektů v; budeme-li však požadovat, aby v byl pouze jediný objekt, výpočetní sílu systému to nezmenší);

(b) $a[_h\ ]_h^{\alpha_1} \rightarrow [_h b]_h^{\alpha_2}$,

kde $h \in H, \alpha_1, \alpha_2 \in \{+,-,0\}, a, b \in V$

(*komunikační pravidlo:* objekt a prochází dovnitř membrány typu h s polarizací α_1, přitom se může modifikovat na objekt b, ale může též platit $a=b$; polarizace membrány se přitom může rovněž změnit na α_2);

(c) $[_h a\ ]_h^{\alpha_1} \rightarrow [_h\ ]_h^{\alpha_2} b$,

kde $h \in H, \alpha_1, \alpha_2 \in \{+,-,0\}, a, b \in V$

(*komunikační pravidlo:* analogické předchozímu pravidlu, ale objekt prochází vně membrány typu h);

(d) $[_h a\ ]_h^{\alpha}$,

kde $h \in H, \alpha \in \{+,-,0\}, a, b \in V$

(*rozpad membrány:* v membráně typu h s polarizací α způsobí objekt a její proděravění a rozpad, veškeré objekty v ní obsažené se stávají součástí obsahu membrány, v níž tato byla vnořena; podle existující konvence tato operace nemůže být aplikována na vnější membránu);

(e) $[_h a\,]_h{}^{\alpha_1} \rightarrow [_h b\,]_h{}^{\alpha_2} [_h c\,]_h{}^{\alpha_3}$,

kde $h \in H, \alpha_1, \alpha_2, \alpha_3 \in \{+,-,0\}, a,b,c \in V$

(*dělení elementární membrány:* elementární membrána typu h s polarizací α_1 se v přítomnosti objektu a rozdělí na dvě membrány stejného typu; její obsah je zkopírován do obou výsledných membrán s výjimkou objektu a, který je modifikován na objekty b, resp. c přecházející do první, resp. druhé kopie);

(f) $[_{h0}[_{h1}\,]_{h1}{}^{\alpha_1} \cdots [_{hk}\,]_{hk}{}^{\alpha_1}\, [_{hk+1}\,]_{hk+1}{}^{\alpha_2} \cdots [_{hn}\,]_{hn}{}^{\alpha_2}]_{h0}{}^{\alpha_0} \rightarrow$

$[_{h0}[_{h1}\,]_{h1}{}^{\alpha_3} \cdots [_{hk}\,]_{hk}{}^{\alpha_3}]_{h0}{}^{\alpha_5}\, [_{h0}[_{hk+1}\,]_{hk+1}{}^{\alpha_4} \cdots [_{hn}\,]_{hn}{}^{\alpha_4}]_{h0}{}^{\alpha_6}$,

kde $k \geq 1$, $n > k$, $h_i \in H$, $0 \leq i \leq n$, a $\alpha_0, \ldots, \alpha_6 \in \{+,-,0\}$, přičemž $\{\alpha_1, \alpha_2\} = \{+,-\}$;

(*dělení ne-elementární membrány:* pokud ne-elementární membrána typu h_0 obsahuje polarizované vnořené membrány $h_1, \ldots, h_n$, z nichž prvních k má opačnou polarizaci než zbytek, dojde k rozdělení membrány h_0 na dvě kopie, přičemž membrány $h_1, \ldots, h_k$, přecházejí do první kopie a zbytek do druhé; pokud h_0 obsahuje další membrány, tyto musí mít neutrální polarizaci a jsou zkopírovány do obou kopií membrány h_0; totéž platí i pro případné objekty v h_0).

P systém s aktivními membránami má následující klíčové vlastnosti:

- je diskrétní v prostoru i v čase, je obecně nedeterministický;
- na rozdíl od CP systémů popsaných výše se v systémech s aktivními membránami vyvíjí každý objekt samostatně, není zde operace podmíněná současným výskytem více objektů jako symport nebo antiport;
- veškeré objekty a membrány se vyvíjejí v každém kroku paralelně s použitím aplikovatelných pravidel;
- na každou membránu v každém kroku lze použít nejvýše jedno pravidlo typu (b)–(f), každý objekt participuje nejvýše v jedné operaci typu (a)–(e);
- žádná membrána v průběhu činnosti nemění svůj typ, pouze polarizaci;
- hloubka vnoření membránové struktury se může v průběhu výpočtu zmenšovat, ale nikdy ne zvětšovat.

Příklad: Mějme P systém s aktivními membránami $\Pi = (V, H, \mu, w_0, \ldots, w_n, R)$, kde:

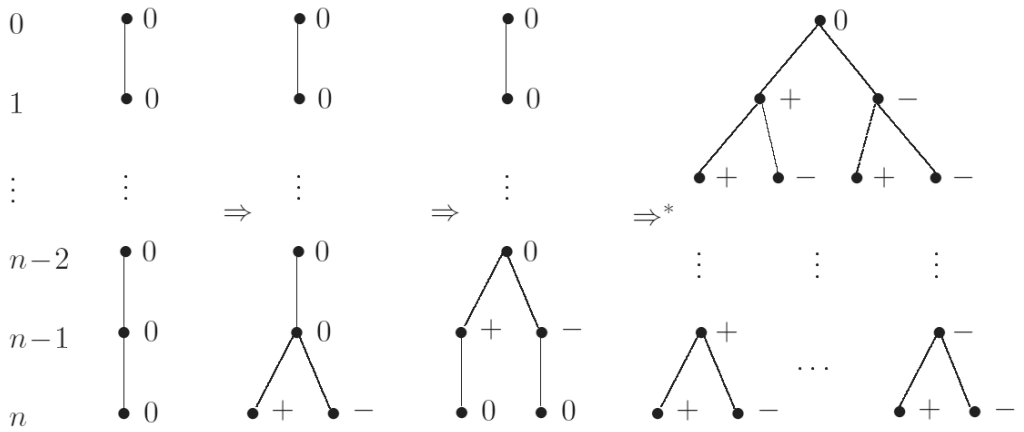
- $V = \{b_0\} \cup \{b_i, f_i, t_i \mid 1 \leq i \leq n\}$;

- $H=\{0,1,\ldots,n\}$;
- $\mu=[_0[_1\ldots[_n]_n{}^0\ldots]_1{}^0]_0{}^0$;
- $w_0=w_1=\ldots=w_{n-1}=\lambda$, $w_n=b_0$;
- R obsahuje tato pravidla:

  (i) $[_n b_i]_n{}^0 \rightarrow [_n b_i]_n{}^+ [_n b_i]_n{}^-$, $0\le i\le n-1$,

  (ii) $[_n b_{ii+1} t_{i+1}]_n{}^+$ and $[_n b_{ii+1} f_{i+1}]_n{}^-$, $0\le i\le n-1$,

  (iii) $[_{i-1}[_i]_i{}^+[_i]_i{}^-]_{i-1}{}^0 \rightarrow [_{i-1}[_i]_i{}^0]_{i-1}{}^+ [_{i-1}[_i]_i{}^0]_{i-1}{}^-$, $2\le i\le n$.



Obr. 5. Příklad P systému s aktivními membránami, který postupně vytváří stromovou strukturu membrán.

Výpočet systému Π je ilustrován na obr. 5. V prvním kroku se elementární membrána na úrovni vnoření n rozdělí pomocí pravidla (i). Dvě výsledné membrány vyprodukují v následujícím kroku pomocí pravidel (ii) symboly f_1 and t_1. Současně pravidlo (iii) rozdělí ne-elementární membránu $[_{i-1}]_{i-1}{}^0$ na dvě části. Tento cyklus se n-krát opakuje a vlna dělení membrán pomocí pravidla (iii) se pohybuje stromem membránové struktury směrem k jeho kořeni. Výpočet končí po $2n-1$ krocích, kdy membránová struktura tvoří vyvážený strom. Každý z jeho 2^n listů obsahuje objekty $\{x_1,x_2,\ldots,x_n,c_n\}$, kde $x_i\in\{f_i,t_i\}$, $1\le i\le n$, tak že jsou přítomny všechny možné n-tice. Symboly t and f mohou označovat například logické hodnoty.

V této chvíli už informatik tuší, že P systém s aktivními membránami bude z výpočetního hlediska značně silný. Regulace aplikovatelných pravidel pomocí

polarizace membrán dodává systému tzv. *výpočetní univerzalitu.* To znamená, že jakýkoli reálně existující počítač lze simulovat nějakým P systémem s aktivními membránami.

Možnost exponenciálního růstu počtu membrán díky jejich dělení navíc dává šanci na dosažení výpočetní síly tzv. *druhé počítačové třídy* [5]. [2] Tato třída obsahuje idealizované modely paralelních počítačů, které disponují neomezeným množstvím procesorů. Každý procesor může v jednom kroku aktivovat jen pevně omezený počet dalších procesorů, takže počet aktivních procesorů může růst v čase nejvýše exponenciálně. Počítače druhé třídy, kdyby byly reálně vyrobeny, by mohly v polynomiálním výpočetním čase řešit obtížné úlohy patřící do třídy **PSPACE**, tedy takové, k jejichž vyřešení by klasický počítač patrně potřeboval exponenciálně velký čas.

Budeme studovat, jak se jednotlivé operace v systémech s aktivními membránami podílejí na jejich výpočetní síle. Samozřejmě nelze často přímo říci, která operace přidává systému kolik síly, ale jisté kombinace operací jsou mocnější než jiné. Níže uvedené výsledky se týkají tzv. *rozpoznávajících P systémů* s aktivními membránami [29], které obdrží z vnějšího prostředí popis případu nějakého rozhodovacího problému a po jistém čase vydají jeho řešení vysláním speciálního objektu *ano* nebo *ne*. Rozpoznávající P systémy s aktivními membránami budeme pro stručnost dále nazývat *AM systémy.*

Pro měření výpočetní síly zkoumaných membránových systémů budeme používat matematický aparát teorie složitosti. Teorie složitosti se zabývá výpočetními problémy a zkoumá zejména spotřebu procesního času a paměti potřebnou k jejich vyřešení na různých typech výpočetních systémů (v našem případě membránových systémů). Podle své náročnosti se dělí úlohy do tzv. *složitostních tříd*. Mezi základní patří třída **P**, obsahující úlohy, které lze v rozumném čase řešit na konvenčních počítačích. Známá je třída **NP** (a jí příbuzná **co-NP**), do níž patří velká řada praktických významných optimalizačních problémů, pro něž neznáme algoritmy k jejich vyřešení v polynomiálním čase (ale současně ani nemáme důkaz, že nemohou existovat). Konečně další důležitou třídou je **PSPACE**, která obsahuje problémy efektivně řešitelné na počítačích druhé třídy. Pro výše popsané třídy platí $\mathbf{P} \subseteq (\mathbf{NP} \cup \mathbf{co-NP}) \subseteq \mathbf{PSPACE}$. Pro další studium doporučujeme například [36] nebo [5].

4.1 Uniformní posloupnosti membránových systémů

Představme si počítač, který řeší nějaký rozhodovací problém, například problém obchodního cestujícího. V informatice je problém chápán jako (potenciálně nekonečná) množina *případů*, kde každý případ je možno formulovat jako otázku s odpovědí ANO/NE. V našem příkladě znamená každý případ množinu měst, mezi nimiž vedou silnice (vše reprezentováno grafem) a otázka zní, zdali je možno po silnicích všechna města projet, aniž by bylo některé navštíveno vícekrát. Různé případy problému mohou mít různou velikost (počet měst). Běžný číslicový počítač může mít program,

[2] *První počítačová třída* odpovídá veškerým doposud známým a fyzicky vyrobeným počítačům, jejichž výpočetní výkon je ekvivalentní – vzhledem k polynomiální redukci – Turingovu stroji.

který je schopen řešit všechny případy problému bez ohledu na jejich velikost (pokud má dost paměti a dost času). Totéž mohou dělat abstraktní modely počítače, s nimiž informatika pracuje, například Turingův stroj, protože mají k dispozici neomezenou paměť a výpočetní čas. Existuje ale řada jiných výpočetních modelů, které tak pružné nejsou, jako například acyklické logické obvody [6]. Každý obvod má pevně danou velikost (a tím i kapacitu paměti a maximální čas výpočtu) kterou nelze v průběhu výpočtu zvyšovat. Takže každý konkrétní obvod je schopen vyřešit pouze případy určité omezené velikosti. Pro větší případy musíme sestrojit obvod nový atd.

V podobné situaci jsou i membránové systémy. Na rozdíl od obvodů sice mohou (dík množení symbolů a membrán) vytvořit během výpočtu neomezeně velikou paměť, avšak práce s ní je neefektivní a pomalá. Hlavním důvodem je, že pokud jsou čísla reprezentována počty objektů v nějaké membráně, jde vlastně o reprezentaci unární soustavou, na rozdíl od soustavy binární v číslicových počítačích. Proto je pro řešení rozhodovacích problémů vhodnější použít *uniformní posloupnosti P systémů.* Každý P systém $\Pi(n)$ v posloupnosti $\{\Pi(n) \mid n \geq 1\}$ je schopen řešit jen případy velikosti n. Slovo *uniformní* v tomto případě znamená, že systém $\Pi(n)$ je možno přesně popsat konvenčním počítačem, jemuž jako vstup dáme číslo n, v polynomiálním čase vzhledem k n. Podobné omezení je uvaleno na způsob, jakým pro systém $\Pi(n)$ zakódujeme popis případu, který má řešit. Tato podmínka zamezuje fixlování: pokud bychom na přípravu $\Pi(n)$ (a kódování vstupu) měli neomezený čas, mohli bychom si odpovědi na všechny případy velikosti n vypočíst už dopředu a vložit je do $\Pi(n)$ (anebo do vstupního zakódování případu - vstupním kódem by vlastně už bylo jeho řešení). Takže systém $\Pi(n)$ by už nemusel nic počítat, jen by přečetl hotovou odpověď.

Alternativním přístupem jsou *semi-uniformní* řešení problémů, kdy sestrojíme speciální P systém $\Pi(x)$ pro každý řešený případ. Prozatím se zdá, že až na nepodstatné výjimky jsou uniformní a semi-uniformní řešení u P systémů stejně silná. Třídu všech problémů řešitelných v polynomiálním čase uniformními posloupnostmi membránových systémů typu $\mathcal{R}$ označíme $\mathbf{PMC}_R$. Podrobnější informace o posloupnostech membránových systému lze nalézt v [29].

Někdy je vhodné ještě více omezit výpočetní prostředky, které jsou k dispozici pro vytvoření jednotlivých členů posloupnosti $\Pi(n)$. Například v $\mathbf{AC}^0$- uniformní posloupnosti musí být jednotlivé její členy (a též vstupní kódování) zkonstruovány pomocí uniformních logických obvodů konstantní hloubky anebo, ekvivalentně, na paralelním počítači (PRAM) v konstantním čase. V dalším textu budeme říkat, že výpočet provedený za těchto omezení probíhá v $\mathbf{FAC}^0$.

4.2 AM systémy bez polarizace

Ukazuje se, že v AM systémech je velmi důležitá možnost ovlivnit chování celé membrány, jinými slovy, regulovat aplikovatelnost pravidel na objekty v ní obsažené. Uvedeme výsledky pro tři varianty membránových systémů bez polarizace – tj. takové, kdy všechny membrány jsou polarizovány neutrálně a jejich polarizace se během výpočtu nemění. V tom případě samozřejmě musíme zrušit podmínku, že membrány $h_1,\ldots,h_k$ a $h_{k+1},\ldots,h_n$ v pravidlech typu (f) jsou polarizovány opačně. AM systémy bez polarizace budeme označovat jako *AM^0 systémy.*

První výsledek je poměrně překvapivý a ukazuje, že pokud u AM^0 systémů odebereme možnost rozpadu membrán, systém prakticky ztratí jakoukoli výpočetní sílu:

Věta 4.1 ([25]) *Každý AM^0 systém bez pravidel typu (d) (tj. bez rozpadu membrán) se dá nahradit výpočetně ekvivalentním AM systémem s jedinou membránou a s pravidly, která v jednom kroku přepisují některé vstupní objekty na yes a jiné na no. Tato náhrada se dá navíc provést v* ***FAC^0****.*

Jinými slovy, výstup každého takového systému je triviální a lze jej velmi snadno předpovědět. Z hlediska konstruktéra počítačů lze činnost takového systému realizovat krabičkou, v níž jsou některé vstupy propojeny drátem na výstup *yes* a jiné na *no*, jinak v ní není nic. Pozoruhodné je, že sílu těchto systémů nezvýší ani použití dělení ne-elementárních membrán, jinak velmi mocná operace. Důvodem je, že sice systém může vytvořit exponenciálně mnoho výpočetních elementů (membrán), ale nemůže zvýšit rozmanitost výpočtů, které v nich probíhají. To nepřímo ukazuje na sílu biodiverzity ve společenstvech.

Situace se podstatně změní, jakmile umožníme rozpad membrán. Jakmile jsou objekty uvolněny z rozpadlé membrány do membrány jí obklopující, dostávají se do jiného prostředí, kde mohou být k dispozici odlišná pravidla než v membráně původní. Důsledkem je informační potenciál odpovídající první počítačové třídě.

Věta 4.2 ([25]) *Uniformní posloupnosti AM^0 systémů pouze s pravidly typu (a) (evoluce objektu) a (d) (rozpad membrány) řeší v polynomiálním čase přesně třídu problémů* ***P****.*

Informatik by mohl namítnout, že jelikož každý případ může být zakódován do vstupních objektů AM^0 systému v polynomiálním čase, může toto kódování pro problémy třídy **P** už obsahovat rovnou jejich řešení, takže samotný systém už nemusí nic počítat. Výše uvedený výsledek však platí i pro $\mathbf{AC}^0$- uniformní posloupnosti AM^0 systémů. Tedy je zřejmé, že řešení úlohy počítá samotný systém.

Věta 4.3 ([26]) *Výsledek věty 4.2 se nezmění, přidáme-li do systému komunikační operace typu (b) a (c), a dále pravidla typu (e) omezená na tvar [ha]h→[hb]h[hb]h (symetrické dělení elementárních membrán).*

Je pozoruhodné, že se během cca. 10 let nepodařilo zjistit, zda stejný výsledek platí i v případě, že připustíme asymetrické dělení elementárních membrán, a tomuto problému je věnována značná pozornost. Jakmile ovšem umožníme dělení ne-elementárních membrán, výpočetní síla těchto systémů vzroste na úroveň počítače 2. třídy:

Věta 4.4 ([2]) *(Semi-)uniformní posloupnosti AM0 systémů řeší v polynomiálním čase přesně třídu problémů* **PSPACE***.*

Pokud ponecháme operaci dělení ne-elementárních membrán, avšak nepřipustíme evoluci objektů a komunikační pravidla, informační potenciál systému se pravděpodobně sníží, avšak přesná charakterizace jejich síly není dosud známa:

Věta 4.5 ([44]) *Semi-uniformní posloupnosti AM^0 systémů bez pravidel typu (a), (b), (c) řeší v polynomiálním čase nejméně všechny problémy ze třídy* ***NP****.*

Tabulka 2. Výpočetní síla AM systémů bez polarizace. X označuje použitou operaci/vlastnost, v případě ? na jejím použití nezáleží.

Evoluční pravidla (a)	X	X		X
Komunikační pravidla (b),(c)	X	?		X
Rozpad membrán (d)		X	X	X
Sym. dělení elem. membrán (e_S)	X	?	?	?
Dělení elementárních membrán (e)	X		X	X
Dělení ne-elem. membrán (f)	X		X	X
Třída problémů řešitelných v polynomiálním čase	$\subseteq$ **AC^0**	= **P**	$\supseteq$ **NP**	= **PSPACE**

4.3 AM systémy s polarizací

Výpočetní potenciál, který dosahují AM^0 systémy, bude při zapojení polarizace dosažitelný i bez operace rozpadu membrán. Tato skutečnost naznačuje, že polarizace membrán a jejich rozpad jsou dva navzájem zastupitelné regulační mechanismy. Skutečně, oba dovolují, aby se skokově změnila sada operací aplikovatelných na objekty uvnitř membrány, což je klíčové po výpočetní efektivitu systému.

Věta 4.6 ([40]) *Semi-uniformní posloupnosti AM systémů pouze s komunikačními pravidly typu (b) a (c) mohou simulovat v polynomiálním čase a prostoru libovolný Turingův stroj, a jsou tudíž schopny v polynomiálním čase řešit třídu problémů* **P**.

Z důkazu podaného v literatuře vyplývá, že simulace Turingova stroje není realizována ve fázi konstrukce systému $\Pi(x)$ řešícího případ x nějakého problému, ale provádí ji samotný AM systém $\Pi(x)$. Když nyní přidáváme do systému další operace typu (a) a (d) (evoluce objektů, rozpad membrán), výpočetní síla systému se nemění [43]. (Po přidání evoluce objektů však stačí použít uniformní posloupnosti namísto semi-uniformních.) Podstatná změna výpočetní síly přichází až s operací elementárního dělení membrán.

Věta 4.7 ([30]) *Uniformní posloupnosti AM systémů s operacemi typu (a), (b), (c) a (e) (evoluce objektů, komunikační pravidla a dělení elementárních membrán) řeší v polynomiálním čase nejméně všechny problémy ze tříd* $\boldsymbol{NP \cup co{-}NP}$.

Pokud přidáme navíc operaci rozpadu membrán, výsledek se nemění. Je pravděpodobné, že by bylo možno omezit i použití některých z typů operací (a), (b),

(c). Konečně, přidáním dělení ne-elementárních membrán se dostáváme na sílu počítače 2. třídy.

Věta 4.8 ([1, 37]) *(Semi-)uniformní posloupnosti AM systémů řeší v polynomiálním čase přesně třídu problémů* **PSPACE**. *Výsledek platí bez ohledu na to, zda použijeme operaci rozpadu membrán.*

Tabulka 3. Výpočetní síla AM systémů s polarizací. X označuje použitou operaci/vlastnost, v případě ? na jejím použití nezáleží

Evoluční pravidla (a)	?	X	X
Komunikační pravidla (b),(c)	X	X	X
Rozpad membrán (d)	?	?	?
Sym. dělení elem. membrán (e_S)		?	?
Dělení elementárních membrán (e)		X	?
Dělení ne-elem. membrán (f)			X
Třída problémů řešitelných v polynomiálním čase	= **P**	⊇ **NP** ∪ **co-NP**	= **PSPACE**

5 Pulsní neuronové P systémy

Z neurologie je známo, že informace je mezi neurony předávána formou elektrických pulsů, které mají prakticky shodný průběh a amplitudu [21]. Intenzita vzruchu je určována frekvencí těchto pulsů. Tradiční modely pro jejich studium jsou analogové, modelující průběhy a frekvence těchto signálů. Pulsní neuronové P systémy (*Spiking Neural P Systems*, zkráceně *SN P systémy*) [15] představují pokus o vytvoření plně diskrétního modelu (pracujícího s přirozenými čísly). Této idei nahrává skutečnost, že pulsy v biologických neuronových sítích lze interpretovat pomocí jedniček (puls přítomen) a nul (puls nepřítomen). Spornou zůstává otázka, zda například přenosovou funkci (váhu) jednotlivých synapsí a časování pulsů reálných neuronů lze modelovat přirozenými čísly bez ztráty věrnosti modelu. Nicméně zde se budeme zabývat pouze výpočetním potenciálem SN P systémů a opomineme aspekty živých neuronových sítí jako schopnost učení, zobecňování poznatků a podobně. K tomuto účelu postačí náš zjednodušený diskrétní model.

SN P systém je popsán orientovaným grafem synapsí propojujících neurony – uzly grafu. Neuron vysílá skrze synapse signály (pulsy) do všech neuronů, s nimiž je jeho výstup propojen. Pulsy se akumulují v těle přijímajících neuronů. Vysílání, resp. zapomínání pulsů je řízeno spouštěcími pravidly (1), resp. zapomínacími pravidly (2),

$$E/a^c \rightarrow \mathrm{a};d \tag{1}$$

$$a^s \rightarrow \lambda \tag{2}$$

kde symbol a reprezentuje jeden puls, E je regulární výraz, $c,s \in N^+$ je počet akumulovaných pulsů, jež se aplikací pravidla spotřebují (ostatní pulsy zůstávají v neuronu), $d \in N$ je relaxační doba mezi spuštěním pravidla a vysláním pulsu. Vidíme výraznou abstrakci vůči biologické realitě jednak v použití silných regulárních výrazů, a dále v tom, že vyslání pulsu nezpůsobí úplné vynulování neuronu jako ve skutečnosti. Tyto vlastnosti však lze vypustit při zachování některých zásadních výsledků uvedených níže.

Pravidlo může být použito, pokud počet pulsů akumulovaných v neuronu je pokryt regulárním výrazem E, ve smyslu, že $a^n \in L(E)$, kde $L(E)$ je jazyk definovaný výrazem E. Během relaxační doby d po aplikaci pravidla je neuron uzavřen a nepřijímá žádné pulsy. Po uplynutí d kroků vyšle puls a současně se opět otevře. Jsou dovolena rovněž tzv. zapomínací pravidla ve tvaru $a^s \rightarrow \lambda$, která způsobí, že obsahuje-li neuron právě s pulsů, je jeho obsah vynulován, aniž by byl vyslán puls. Neurony mohou obsahovat více pravidel. Je-li počet pulsů v neuronu pokryt současně regulárními pravidly více pravidel, je jedno z nich vybráno náhodně. Pokud však neuron obsahuje též zapomínací pravidlo(a), tak jeho(jejich) spouštěcí počet pulsů s nesmí být pokryt žádným regulárním výrazem jiného pravidla v témže neuronu.

Systém začíná pracovat z počáteční konfigurace dané předdefinovaným rozložením pulsů v neuronech, pravidla jsou aplikována v diskrétních časových krocích. Všechny neurony fungují synchronně (ačkoli byly zkoumány i asynchronní varianty). Sekvenci konfigurací, kterými systém prochází, nazveme *výpočet.* Ten probíhá, dokud v systému kolují nějaké pulsy. Vstup a výstup je reprezentován jako série binárních číslic (1 = puls, 0 = nepřítomnost pulsu) přijímaných/vysílaných na vstupu/výstupu vybraných neuronů. Někteří autoři též reprezentují vstupní a výstupní hodnoty unárně – jako délku intervalu mezi dvěma pulsy.

Zvláštností SN P systémů je, že o vyslání pulsu rozhoduje vyhodnocení regulárního výrazu v neuronu, a to v konstantním čase. Tato vlastnost může být motivována názory některých neurobiologů, podle nichž uvnitř neuronů probíhají daleko složitější výpočty, než jaké předpokládají jejich běžně užívané modely. Fyzik Roger Penrose dokonce soudí, že kvantové jevy na nervových vláknech mohou způsobovat až superturingovský výpočetní potenciál neuronových sítí a vysvětlovat určité paranormální úkazy jako telepatii a podobně. (Podle názoru autora této kapitoly se nejeví složité a jemné výpočty uvnitř neuronů v realitě jako podstatné, uvážíme-li fakt, že síť musí být velmi robustní a odolná vůči řadě fluktuací jako jsou hladiny neurotransmiterů, množství živin atd., nemluvě už o odumírání neuronů a synapsí.) Ať tak či onak, regulární výrazy představují velmi silný výpočetní formalismus, jak uvidíme v následujícím textu.

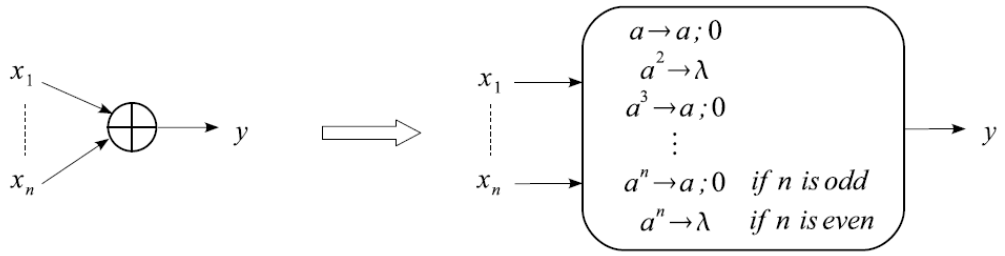
Hned v úvodním článku [15] bylo ukázáno, že takto definované SN P systémy jsou výpočetně univerzální – mohou realizovat jakýkoli algoritmus spustitelný na konvenčních počítačích nebo na Turingově stroji. Postupně byl tento výsledek dále zesílen:

Věta 5.1 ([14, 11]) *SN P systémy disponují univerzální výpočetní silou, i když odebereme kombinaci libovolných dvou z následujících tří dílčích operací:*

- *použití regulárních výrazů (s výjimkou těch nejjednodušších: a^* a a^n, $n \geq 1$);*
- *použití zapomínacích pravidel;*
- *použití relaxační doby.*

Je zajímavé, že pro odstranění regulárních výrazů, nerealistických z pohledu biologa, bylo nutno konstruovat paměť SN P systému pomocí soustavy pulsů neustále cirkulujících v uzavřených obvodech, což odpovídá principu krátkodobé paměti v živých organismech. Je pravděpodobné (avšak není znám formální důkaz), že při odebrání všech tří těchto operací systém ztratí výpočetní univerzalitu. Analýzu výpočetní efektivity SN P systémů zahájíme zdánlivě pesimistickým a kontra-intuitivním výsledkem.

Věta 5.2 ([27]) *Žádný SN P systém s konstantním počtem neuronů nemůže simulovat Turingův stroj s menším než exponenciálním zpomalením.*



Obr. 6. Simulace XOR s neomezeným počtem vstupů pomocí SN P systému. Převzato z [20].

Proč je tento výsledek kontra-intuitivní? Neurony v SN P systémech mohou snadno realizovat v konstantním čase logické operace AND, OR, NOT (případně XOR) [20], viz obr. 6. Přítomnost pulsu na výstupu neuronu v daném časovém okamžiku je chápána jako logická 1, nepřítomnost jako 0. Je známo, že pomocí těchto operací lze sestrojit libovolný (i cyklický) logický obvod. Takto sestavený neuropočítač bude pak pracovat se stejnou nebo obdobnou efektivitou jako klasický číslicový počítač. Klíčový rozdíl je ovšem ve způsobu ukládání informací. Turingův stroj (a další modely první počítačové třídy) má k dispozici neomezeně velkou binární (obecně *n*-ární) paměť. SN P systémy sice neomezenou paměť mají rovněž — jejich idealizovaný neuron může akumulovat libovolné množství pulsů — ale ukládání informací v ní je unární, a tím exponenciálně pomalejší. Abychom se dosáhli obdobné síly, jakou mají klasické modely počítače používané v informatice, musíme se opět uchýlit k (semi-)uniformním posloupnostem SN P systémů.

5.1 Posloupnosti pulsních neuronových P systémů

Začneme zkoumáním vlastností *uniformních* posloupností rozpoznávajících SN P systémů. Ty jsou velmi podobné uniformním posloupnostem AM systémů popsaným

v kapitole 4.1, s následující odchylkou. U AM systémů měříme jejich velikost v počtech membrán a objektů, jež obsahují, neboť toto odpovídá jejich případné fyzické (buněčné) implementaci. U biologických neuronů, jež jsou inspirací SN P systémů, však pulsy akumulované v neuronu nejsou fyzikálním objektem, ale elektrickým potenciálem. Proto chápeme velikost neuronu obsahující n pulsů pouze jako počet bitů čísla n. Velikost popisu SN P systémů je podstatná pro konstrukci jejich uniformních posloupností. Každý člen posloupnosti totiž musí být zkonstruován v polynomiálním čase, a tudíž taky jeho velikost musí být polynomiálně omezena. Náš způsob chápání velikosti SN P systémů vede k následujícímu výsledku:

Věta 5.3 *Uniformní posloupnosti rozpoznávajících SN P systémů charakterizují svými výpočty v polynomiálním čase třídu* ***NP****.*

Tento výsledek je na první pohled překvapivý v kontrastu s větou . Princip je však velmi jednoduchý – rozhodující část výpočetní síly ve větě 5.3 dodávají SN P systémům regulární výrazy. Skutečně, existují jednoduché regulární výrazy, jejichž vyhodnocení je analogické vyřešení některých NP-úplných problémů, například problému SUBSET-SUM. Proto regulární výrazy představují velmi silný aparát a je vhodné uvažovat o jejich omezení v SN P systémech. Vhodným omezením se jeví například jednohvězdičková normální forma, popsaná v části 2.

Věta 5.4 *Uniformní posloupnosti rozpoznávajících SN P systémů, které mají všechny regulární výrazy v jednohvězdičkové normální formě, charakterizují svými výpočty v polynomiálním čase třídu* **P**.

V předchozím textu jsme v souvislosti s AM systémy zmiňovali jejich uniformní posloupnosti, které řeší v polynomiálním čase ještě složitější problémy ze třídy **PSPACE**. Podmínkou však byla schopnost vytvořit velmi rychle díky množení exponenciální množství membrán. Podobnou vlastností však SN P systémy nedisponují. Pokud bychom chtěli řešit problémy třídy PSPACE v polynomiálním čase, museli bychom nějakým způsobem umožnit aktivaci exponenciálního množství neuronů, například v nějaké předpřipravené neohraničené struktuře [16]. Tento přístup je analogický konstrukci některých standardních modelů ve 2. počítačové třídě, například PRAM.

Poslední zajímavou variantou, o které se zmíníme, jsou *neuniformní* posloupnosti SN P systémů. Znamená to, že každý SN P systém v posloupnosti může být konstruován s použitím neomezeného času a prostoru, nebo dokonce s použitím superturingovských výpočtů. Tato myšlenka má oporu v realitě – uvažme, že buněčné systémy se vyvíjejí pomocí evolučních tlaků, které jsou nepředvídatelné, a je známo, že evoluční posloupnosti i jednoduchých automatů mají superturingovskou výpočetní sílu [42].

Pokud bychom pro každý případ problému zkonstruovali samostatný SN P systém, mohly by takové posloupnosti řešit jakékoli, i nevyčíslitelné problémy – samotný SN P systém by nemusel dělat nic, jen sdělit řešení případu, které by bylo vypočteno a vloženo do něj během jeho konstrukce. Budeme proto požadovat, aby každý SN P systém – člen posloupnosti $\{\Pi(n)|n=1,2,3,\ldots\}$ – řešil *všechny* případy daného problému o velikosti n. Zde je možno najít úzký vztah k tzv. *Turingovým strojům*

s nápovědou, které mají k dispozici během výpočtu jisté množství informace získané nespecifikovaným způsobem (vnuknutí, intuice, jak libo). Třídu problémů, kterou tyto stroje řeší v polynomiálním čase a s nápovědou polynomiální velikosti, označujeme **P**/*poly*.

Věta 5.5 *Neuniformní posloupnosti rozpoznávajících SN P systémů polynomiální velikosti, které mají všechny regulární výrazy v jednohvězdičkové normální formě, charakterizují svými výpočty v polynomiálním čase třídu **P**/poly.*

6 Závěr

V této kapitole jsme studovali několik typů membránových systémů (P systémů), které abstrahují některé operace probíhající v buňkách a buněčných membránách z hlediska zpracování informací. Protože informační procesy v živé hmotě se stávají stále zásadnějším tématem z hlediska pochopení pochodů v organismech a systémové biologie obecně, je zajímavé prozkoumat vliv kombinace různých operací na výpočetní sílu matematicky popsaného modelu buňky či buněčného systému. Mezi tyto operace patří například řízený prostup objektů membránami, polarizace membrán, evoluce elementárních objektů (abstrahované chemické reakce), rozpad (proděravění) membrán a dělení membrán. V případě pulsních neuronových P systémů je to pak např. použití regulárních výrazů ke spouštění pulsů, vymazávání pulsů, relaxační perioda mezi pulsy a podobně.

V první části kapitoly jsme se zabývali komunikujícími mebránovými systémy. Zde se jako velmi důležitá ukázala možnost synchronizace různých paralelně probíhajících operací v systému. Jakmile synchronizaci odstraníme (například požadavkem na sekvenční aplikaci pravidel), systém ztratí značnou část své výpočetní síly. Rovněž požadavek determinismu prováděných operací v některých případech zmenšuje informačně-procesní kapacitu systému.

Následoval text věnovaný P systémům s aktivními membránami. Elementárním poznatkem bylo, že nahromadění velkého (exponenciálního) množství proměnlivých objektů v systému nezvyšuje výrazně jeho informační potenciál. Ten se zvýší podstatně v okamžiku, kdy získáme k dispozici (např. buněčným dělením) exponenciální množství membrán, v nich mohou probíhat různorodé pochody. Dalším podstatným faktorem je, zda tyto membrány mohou být dále vnitřně strukturovány či nikoli. Poměrně překvapivě se ukázalo, že podstatnou operací pro výpočetní sílu systému je přítomnost operace rozpadu membrán a jejich polarizace. Tyto dvě operace se mohou do jisté míry suplovat. Klíčem k vysvětlení je fakt, že obě umožňují změnu prostředí uvnitř membrány a tedy přepnutí do jiného režimu činnosti. Bez možnosti podobného přepínání, a tedy kontrolované diverzity, zůstává potenciál zpracování informací nízký, i když množství membrán exponenciálně roste.

Konečně z výsledků získaných pro pulsní neuronové P systémy uveďme, že ačkoli mohou snadno simulovat logické obvody a mají navíc neomezenou paměťovou kapacitu, jsou exponenciálně méně efektivní než klasické počítače. Vysvětlením je v tomto případě jejich neefektivní unární způsob ukládání informací, který je však odpozorován z neuropsychologie – když si představujeme nějakou množinu objektů,

naše představy jsou unární. Abychom dosáhli rozumného výpočetního potenciálu, musíme pracovat s (evolučními) posloupnostmi neuronových P systémů.

Tyto závěry samozřejmě vycházejí z vysoce abstraktního modelu membránového systému, který opomíjí řadu aspektů chování živé hmoty. Publikované závěry tedy slouží pouze ke stanovení mantinelů, v nichž se pohybuje informační síla modelovaného biologického systému. I tak jsme však na tomto modelu pozorovali řadu fenoménů, jež jsou potvrzeny z přírody, například důležitou roli synchronizace, (bio)-diverzity, regulace činnosti buněk změnou jejich vnitřního prostředí (chemickým či elektrickým signálem), potřebu neustálé cirkulace signálů v neuronových paměťových modulech a další.

Poděkování: Tato kapitola vznikla za podpory grantu SGS/4/2010 Slezské univerzity v Opavě.

Literatura

[1] Alhazov, A., Martín-Vide, C., Pan, L.: Solving a PSPACE-complete problem by P systems with restricted active membranes. *Fundamenta Informaticae* **58**(2) (2003) 67-77.

[2] Alhazov, A., Pérez-Jiménez, M.: Uniform solution of QSAT using polarizationless active membranes. In Durand-Lose and Margenstern [9], pp. 122-133.

[3] Andrei, S., Cavadini, S., Chin, W.-N.: A new algorithm for regularizing oneletter context-free grammars. *Theoretical Computer Science* **306** (2003) 113-122.

[4] Antoniou, I., Calude, C., Dinneen, M. (eds.): *Unconventional Models of Computation, UMC'2K*, Proceedings of the Second International Conference. Springer, 2001.

[5] Boas, P. v. E.: Machine models and simulations. In van Leeuwen [41], pp. 1-66.

[6] Boppana, R., Sipser, M.: The complexity of nite functions. In van Leeuwen [41], pp. 757-804.

[7] Calude, C., Costa, J., Freund, R., Oswald, M., Rozenberg, G. (eds.): *Unconventional Computing*, 7th International Conference, UC 2008, Vienna, Austria, August 25-28, 2008. Proceedings, Lecture Notes in Computer Science, volume 5204. Springer, 2008.

[8] Cienciala, L., Ciencialová, L., Frisco, P., Sosík, P.: On the power of deterministic and sequential communicating p systems. *Int. J. of Foundations of Computer Science* **18**(2) (2007) 415-431.

[9] Durand-Lose, J., Margenstern, M. (eds.): *Machines, Computations, and Universality*, 5th International Conference, MCU 2007, Lecture Notes in Computer Science, volume 4664. Springer, 2007.

[10] Eleftherakis, G., Kefalas, P., Paun, G., Rozenberg, G., Salomaa, A. (eds.): *Membrane Computing*, 8th International Workshop, WMC 2007, Lecture Notes in Computer Science, volume 4860. Springer, 2007.

[11] García-Arnau, M., Pérez, D., Rodríguez-Patón, A., Sosík, P.: Spiking neural P systems: Stronger normal forms. *Intern. J. of Unconventional Computing* **5**(5) (2009) 411-425.

[12] Greibach, S. A.: Remarks on blind and partially blind one-way multicounter machines. *Theoretical Computer Science* **7** (1978) 311-324.

[13] Gutiérrez-Naranjo, M., P un, G., Romero-Jiménez, A., Riscos-Núnez, A. (eds.): *Fifth Brainstorming Week on Membrane Computing*. Sevilla: Fenix Editora, 2007.

[14] Ibarra, O., P un, A., P un, G., Rodríguez-Patón, A., Sosík, P., Woodworth, S.: Normal forms for spiking neural P systems. *Theoretical Computer Science* **372**(2-3) (2007) 196-217.

[15] Ionescu, M., P un, G., Yokomori, T.: Spiking neural P systems. *Fundamenta Informaticae* **71**(2-3) (2006) 279-308.

[16] Ishdorj, T.-O., Leporati, A., Pan, L., Zeng, X., Zhang, X.: Deterministic solutions to QSAT and Q3SAT by spiking neural P systems with pre-computed resources. In Martínez-del-Amor et al [22], pp. 1-27. Volume 2.

[17] Jack, J., Paun, A.: Discrete modeling of biochemical signaling with memory enhancement. *T. Comp. Sys. Biology* **11** (2009) 200-215.

[18] J.E. Hopcroft, J. U., R. Motwani: *Introduction to Automata Theory, Languages, and Computation*. Boston: Addison-Wesley, 2001.

[19] Krone, S., Lu, R., Fox, R., Suzuki, H., Top, E.: Modelling the spatial dynamics of plasmid transfer and persistence. *Microbiology* **153**(8) (2007) 2803-2816.

[20] Leporati, A., Zandron, C., Ferretti, C., Mauri, G.: On the computational power of spiking neural P systems. In Gutiérrez-Naranjo et al [13], pp. 227-245.

[21] Maass, W., Bishop, C. (eds.): *Pulsed Neural Networks*. Cambridge: MIT Press, 1999.

[22] Martínez-del-Amor, M., Orejuela-Pinedo, E., P un, G., Hurtado de Mendoza, I., Riscos-Núnez, A. (eds.): *Seventh Brainstorming Week on Membrane Computing*. Sevilla: Fenix Editora, 2009.

[23] Mazza, T., Cavaliere, M.: Cell cycle and tumor growth in membrane systems with peripheral proteins. *Electronic Notes in Theoretical Computer Science* **227** (2009) 127-141.

[24] Minsky, M.: *Computation - Finite and Infinite Machines*. New Jersey: Prentice Hall, Englewood Cliffss, 1967.

[25] Murphy, N.: *Uniformity Conditions for Membrane Systems: Uncovering Complexity Below*. P. Maynooth: National University of Ireland, 2010. PhD thesis.

[26] Murphy, N., Woods, D.: Active membrane systems without charges and using only symmetric elementary division characterise P. In Eleftherakis et al [10], pp. 367-384.

[27] Neary, T.: On the computational complexity of spiking neural P systems. In Calude et al [7], pp. 189-205.

[28] Păun, G., Rozenberg, G., Salomaa, A. (eds.): *The Oxford Handbook of Membrane Computing*. Oxford: Oxford University Press, 2010.

[29] Pérez-Jiménez, M.: A computational complexity theory in membrane computing. In Păun et al [32], pp. 125-148.

[30] Pérez-Jiménez, M., Romero-Jiménez, A., Sancho-Caparrini, F.: Complexity classes in models of cellular computing with membranes. *Natural Computing* **2** (2003) 265-285.

[31] Păun, G.: Computing with membranes. *J. Comput. System Sci.* **61** (2000) 108-143.

[32] Păun, G., Pérez-Jiménez, M., Riscos-Núnez, A., Rozenberg, G., Salomaa, A. (eds.): *Membrane Computing*, 10th International Workshop, WMC 2009, Lecture Notes in Computer Science, volume 5957. Berlin: Springer, 2010.

[33] Păun, G.: P systems with active membranes: attacking NP-complete problems. *J. Automata, Languages and Combinatorics* **6**(1) (2001) 75-90.

[34] Romero-Campero, F. J., Pérez-Jiménez, M. J.: A model of the quorum sensing system in vibrio fischeri using P systems. *Artificial Life* **14**(1) (2008) 95-109.

[35] Sedwards, S., Mazza, T.: Cyto-Sim: a formal language model and stochastic simulator of membrane-enclosed biochemical processes. *Bioinformatics* **23**(20) (2007) 2800-2802.

[36] Sipser, M.: *Introduction to the Theory of Computation*, Second Edition. Independence (Kentucky): Course Technology, 2005.

[37] Sosík, P., Rodríguez-Patón, A.: Membrane computing and complexity theory: A characterization of PSPACE. J. Comput. System Sci. 73(1) (2007) 137-152.

[38] Sosík, P.: P systems versus register machines: two universality proofs. In *Pre-proceedings of Workshop on Membrane Computing* (WMC-CdeA2002). Curtea de Arges: Molconet Publication, 2002, pp. 371-382.

[39] Sosík, P., Matýsek, J.: Membrane computing: when communication is enough. In Calude, C., Dinneen, M. J., Peper, F. (eds.) *Unconventional Models of Computation*. Springer, 2002, Lecture Notes in Computer Science, volume 2509, pp. 264-275.

[40] Valsecchi, A., Porreca, A., Leporati, A., Mauri, G., Zandron, C.: An efficient simulation of polynomial-space turing machines by P systems with active membranes. In Paun et al [32], pp. 461-478.

[41] van Leeuwen, J. (ed.): *Handbook of Theoretical Computer Science*, volume A: Algorithms and Complexity. Amsterdam: Elsevier, 1990.

[42] Wiedermann, J., van Leeuwen, J.: How we think of computing today. In Beckmann, A., Dimitracopoulos, C., Löwe, B. (eds.) *Logic and Theory of Algorithms*, 4th Conference on Computability in Europe, CiE 2008. Springer, 2008, Lecture Notes in Computer Science, volume 5028, pp. 579-593.

[43] Zandron, C., Ferretti, C., Mauri, G.: Solving NP-complete problems using P systems with active membranes. In Antoniou et al [4], pp. 289-301.

[44] Zandron, C., Leporati, A., Ferretti, C., Mauri, G., Pérez-Jiménez, M.: On the computational efficiency of polarizationless recognizer P systems with strong division and dissolution. *Fundam. Inform*. **87**(1) (2008) 79-91.

Neurčitosť, fuzzy množiny, aplikácie, problémy

Michal ŠABO[1]

Abstrakt. Teória fuzzy logiky a fuzzy množín zapĺňa medzeru oddeľujúcu klasické matematické modely od reálnych systémov. Zaoberá sa neurčitosťou spôsobenou nejasnosťou pojmov a je integrálnou časťou teórie inteligentných systémov. V tejto kapitole popíšeme základné nástroje teórie fuzzy množín, problémy a súčasný stav v súvislosti s niektorými aplikáciami.

1 Úvod

K najväčším pokrokom v aplikovanej matematike v minulom storočí patrí zmena pohľadu na pojem neurčitosti. V klasickej matematike bola neurčitosť považovaná za nežiaduci jav a bola snaha ju odstrániť všetkými spôsobmi. Prejavy neurčitosti ako nepresnosť, vágnosť, neistota boli považované za nevedecké. Za najdôležitejší medzník vo vývoji moderného pojmu neurčitosti sa považuje článok L. Zadeha "Fuzzy sets" z roku 1965 [23]. V tomto článku Zadeh zaviedol pojem fuzzy množiny, množiny, ktorej hranice sú nepresné - rozmazané. Treba však pripomenúť, že Zadeh nadväzoval na myšlienky ďalších vedcov a nielen matematikov. Spomenieme napríklad amerického filozofa Maxa Blacka. Zadehove myšlienky našli odozvu nielen v teórii množín a pravdepodobnosti, ale aj v klasickej Aristotelovej dvojhodnotovej logike. V novom pohľade tvrdenie „x je prvkom množiny A“ nemusí byť nutne pravdivé alebo nepravdivé, ale môže byť pravdivé len v istom stupni. Obvykle sa stupeň príslušnosti vyjadruje číslom z intervalu [0;1]. Hraničné hodnoty 0 a 1 reprezentujú úplnú istotu, že prvok nepatrí resp. patrí do množiny A. Na rozdiel od neurčitosti spôsobenej nepresnosťou merania, t.j. neistoty o aký prvok ide, fuzzy množiny umožňujú narábať s neurčitosťou zapríčinenou vágnosťou pojmu, ktorý v bežnom jazyku vymedzuje množinu. Napríklad neistotu príslušnosti človeka do množiny vysokých ľudí nespočíva v nepresnosti merania jeho výšky, ale v neurčitosti pojmu „vysoký človek“.

Úspešnosť myšlienky zavedenia pojmu fuzzy množín spočíva najmä v tom, že fuzzy množiny môžu byť popísané klasickými matematickými prostriedkami a teória fuzzy množín používa a obohacuje klasickú matematickú analýzu a teóriu množín.

Hlavnou motiváciou teórie fuzzy množín je snaha vyjadriť vágnosť ľudských poznatkov vyjadrených prirodzeným jazykom matematickými prostriedkami a tak

[1] FCHPT STU, Radlinského 9, 812 37 Bratislava, e-mail: michal.sabo@stuba.sk

prekonať medzeru oddeľujúcu klasické matematické modely od reálnych fyzikálnych systémov. Teória umožňuje systém reprezentovaný poznatkami uvedenými v bežnom hovorovom jazyku uviesť do reálnej činnosti. Dôkazom je množstvo aplikácií fuzzy expertných systémov v rôznych oblastiach, napríklad v spotrebnom, automobilovom, leteckom a vojenskom priemysle, poľnohospodárstve, robotike a ďalších oblastiach. Teória fuzzy množín je integrálnou časťou rozvíjajúcej sa teórie inteligentných systémov.

Fuzzy modelované rozhodovacie systémy simulujú ľudské myslenie včítane používania vágnych pojmov ako napríklad: "trochu doprava", "dosť malý", "veľmi starý" a pod. Prvé pokusy fuzzy modelovania rozhodovacieho procesu spočívali v zovšeobecnenom „modus ponens“ ako základnom odvodzovacom pravidle (napr. Mamdaniho fuzzy regulátor). Úspešné použitie fuzzy expertných systémov je limitované vyriešením problémov súvisiacich so zovšeobecnením pojmov klasickej logiky a teórie množín. Principiálnym problémom je fuzzy modelovanie klasických logických spojok ako napríklad „súčasne, alebo, vyplýva“ a pod.

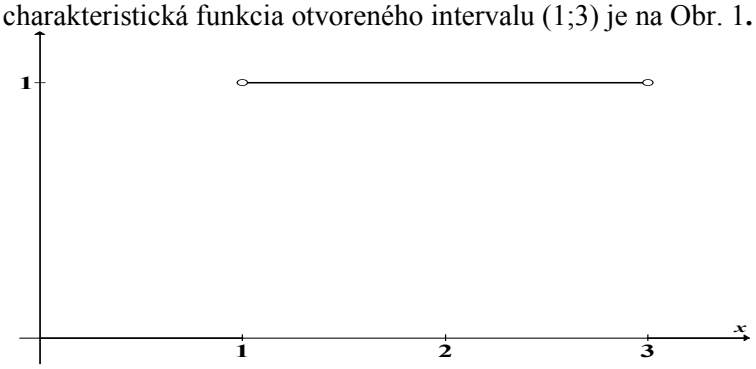
V tejto kapitole stručne uvedieme motiváciu a definíciu niektorých základných pojmov viachodnotovej (fuzzy) logiky a teórie fuzzy množín v 2. a 3. časti. Ďalšie dve časti sú venované aplikáciám fuzzy množín, fuzzy preferenčným štruktúram a fuzzy regulátorom. Ešte poznamenávame, že v tejto kapitole desatinné čísla budeme zapisovať desatinnou bodkou a nie desatinnou čiarkou.

2 Fuzzy množiny a operácie s fuzzy množinami

Charakteristická funkcia klasickej (crisp) podmnožiny A univerzálnej množiny $\boldsymbol{X}$ je definovaná vzťahom:

$$\mu_A(x) = \begin{cases} 1 & ak\ x \in A \\ 0 & ak\ x \notin A \end{cases}$$

Napríklad, charakteristická funkcia otvoreného intervalu (1;3) je na Obr. 1.



Obr. 1. Charakteristická funkcia intervalu (1;3)

Fuzzy podmnožina A univerzálnej množiny $\boldsymbol{X}$ je definovaná svojou funkciou príslušnosti.

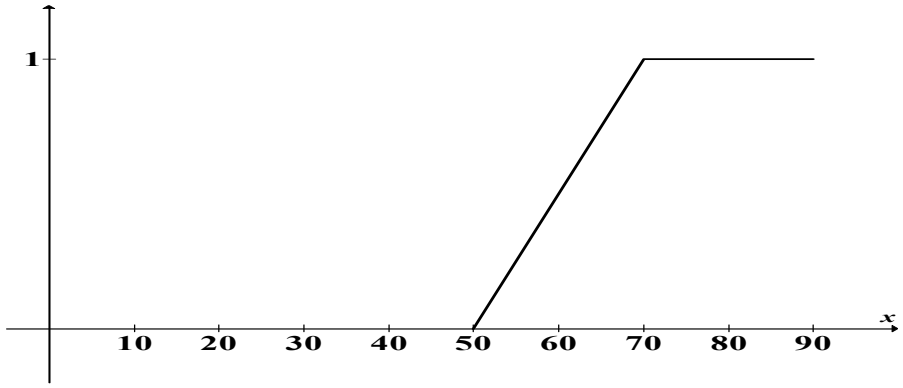
$$\mu_A : \boldsymbol{X} \to [0;1]$$

Pre jednoduchosť budeme označovať fuzzy podmnožinu a jej funkciu príslušnosti rovnakým symbolom a budeme ju nazývať aj fuzzy množina. Teda fuzzy množina A je daná svojou funkciou príslušnosti $A : \mathrm{X} \to [0;1]$ Ak je univerzálna množina $\boldsymbol{X} = \{x_1, x_2, \ldots, x_n\}$ konečná, potom jej fuzzy podmnožiny možno zapisovať v tvare:

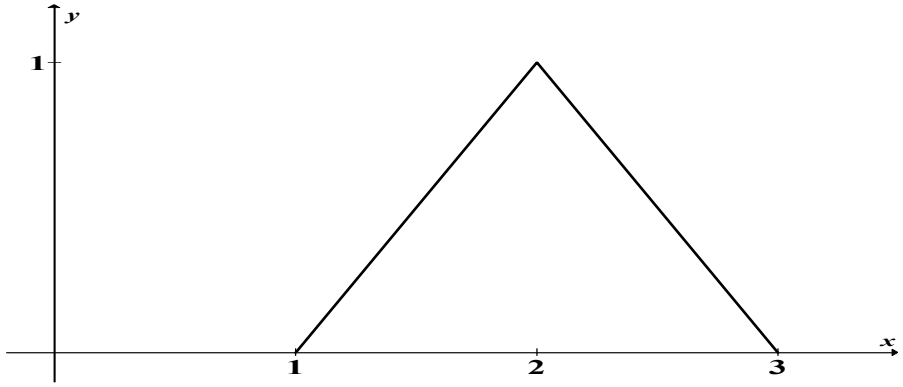
$$A = \{A(x_1)/\, x_1,\, A(x_2)/\, x_2,\, A(x_3)/\, x_3,\, \ldots,\, A(\mathrm{x_n})/\, x_n\}$$

Napríklad, $B = \{0.5/a,\, 0/b,\, 1/c\}$ je fuzzy podmnožina univerzálnej množiny $\boldsymbol{X} = \{a, b, c\}$ taká, že $A(a) = 0.5$, $A(b) = 0$, $A(c) = 1$. Triedu všetkých fuzzy podmnožín univerzálnej množiny $\boldsymbol{X}$ budeme označovať symbolom $F(\boldsymbol{X})$.

Napríklad, fuzzy množina „vekov starých ľudí“ (Obr. 2) môže byť definovaná funkciou príslušnosti na univerzálnej množine $\boldsymbol{X} = R+$. Fuzzy množina čísel „blízkych číslu 2“ môže byť definovaná na univerzálnej množine R (Obr. 3). Takéto fuzzy množiny nazývame tiež trojuholníkové fuzzy čísla [10], [12].

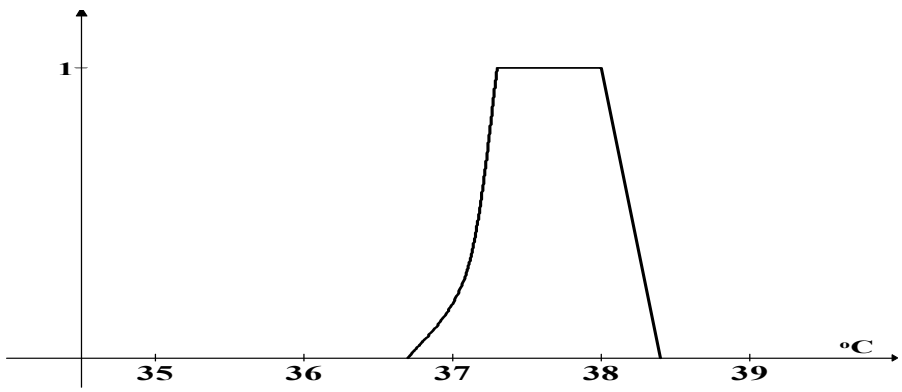


Obr. 2. Fuzzy množina "vek starého človeka"



Obr. 3. Fuzzy množina „Číslo blízke 2“

Teda, vek 60-ročného človeka patrí do fuzzy množiny vekov starých ľudí s príslušnosťou 0.5. Možno to interpretovať tak, že 50% opýtaných si myslí, že 60-ročný človek je starý. Uveďme ešte jeden príklad z lekárskeho prostredia. Fuzzy množina" zvýšených telesných teplôt" je na Obr. 4.



Obr. 4. Fuzzy množina "zvýšených telesných teplôt"

Takéto fuzzy množiny nazývame fuzzy intervaly [10], [12]. Fuzzy množina A je ***normálna***, ak existuje $x \in \boldsymbol{X}$ tak, že $A(x) = 1$. Nosičom (suportom) fuzzy podmnožiny A univerzálnej množiny $\boldsymbol{X}$ nazývame množinu:

$$\mathrm{supp}(A) = \{ x \in \boldsymbol{X} \mid A(x) > 0\}$$

Doplnok klasickej (crisp) množiny je definovaný pomocou logickej negácie (non)

$$\neg : \{0,1\} \to \{0,1\}$$

Zjednotenie a prienik klasických množín sú definované pomocou logických spojok

$$\wedge, \vee : \{0,1\}^2 \to \{0,1\},$$

nazývaných konjunkcia a disjunkcia (and, or). Ich zovšeobecnenia sa nazývajú fuzzy negácia, fuzzy konjunkcia a fuzzy disjunkcia. Fuzzy konjunkciu a fuzzy disjunkciu s ďalšími rozumnými vlastnosťami nazývame triangulárnou normou resp. triangulárnou konormou (skrátene t-norma a t-konorma). Názov "triangulárna" pochádza z teórie pravdepodobnostných metrík – trojuholníkovej nerovnosti.

Označme fuzzy negáciu n, t-normu T a t-konormu S: Teda

$$n : [0;1] \to [0;1], \qquad T, S : [0;1]^2 \to [0;1]$$

kde $[0;1]^2 = [0;1] \times [0;1]$ je karteziánsky súčin jednotkových intervalov. Potom možno definovať (bodovo – pre každé $x \in \boldsymbol{X}$):

- doplnok fuzzy množiny A ako fuzzy množinu A^c, ktorá má funkciu príslušnosti:

$$A^c(x) = n(A(x))$$

- prienik fuzzy množín ako fuzzy množinu $A \cap B$, ktorá má funkciu príslušnosti:

$$(A \cap B)(x) = T(A(x), B(x))$$

- zjednotenie fuzzy množín ako fuzzy množinu $A \cup B$, ktorá má funkciu príslušnosti:

$$(A \cup B)(x) = S(A(x), B(x))$$

Od unárnej operácie $n: [0,1] \to [0,1]$ a binárnych operácií $T, S: [0;1]^2 \to [0;1]$ vyžadujeme okrem vlastností zdedených z dvojhodnotovej logiky aj ďalšie "rozumné" vlastnosti [8], [9], [10], [12].

Definícia 1: Unárna operácia $n: [0,1] \to [0,1]$ sa nazýva fuzzy negácia, ak:

- $a < b \Rightarrow n(a) \geq n(b)$, pre všetky $a, b \in [0,1]$,
- $n(0) = 1,\ n(1) = 0$.

Príklad 1: Najmenšia a najväčšia a štandardná fuzzy negácia ($x \in [0;1]$):

$$a)\quad n_1(x) = \begin{cases} 1 & ak\ x = 0 \\ 0 & inak \end{cases} \qquad b)\quad n_2(x) = \begin{cases} 0 & ak\ x = 1 \\ 1 & inak \end{cases} \qquad c)\quad n_3(x) = 1 - x$$

Fuzzy negácia n sa nazýva ***silná***, ak je rastúcou involutívnou bijekciou $[0;1] \to [0;1]$, t.j. $n(n(x)) = x$ pre každé $x \in [0;1]$. Je zrejmé, že silná fuzzy negácia je spojitá a $n = n^{-1}$. V tejto kapitole budeme používať len silnú fuzzy negáciu. Potom, pre doplnok fuzzy množiny platí $(A^c)^c = A$.

Definícia 2: Binárnu operáciu $T: [0;1]^2 \to [0;1]$ nazývame t-normou ak pre všetky $a, b, c \in [0;1]$ platí:

t1) $T(1,a) = a$ (neutralita prvku 1),

t2) $a \leq b \Rightarrow T(a,c) \leq T(b,c)$ (monotónnosť),

t3) $T(a,b) = T(b,a)$ (komutativita),

t4) $T(T(a,b),c)) = T(a,T(b,c))$ (asociativita).

Príklad 2: Nasledujúce operácie sú t-normy ($x,y \in [0; 1]$):

1) $T_M(x,y) = \min(x,y)$ (najväčšia t-norma)
2) $T_P(x,y) = xy$ (súčinová t-norma)
3) $T_L(x,y) = \max(0, x + y - 1)$ (Lukasiewiczova t-norma)

Definícia 3: Binárnu operáciu $S: [0;1]^2 \to [0;1]$ nazývame t-konormou, ak pre všetky $a, b, c \in [0;1]$ platí:

t1) $S(0,a) = a$ (neutralita prvku 0),

t2) $a \leq b \Rightarrow S(a,c) \leq S(b,c)$ (monotónnosť),

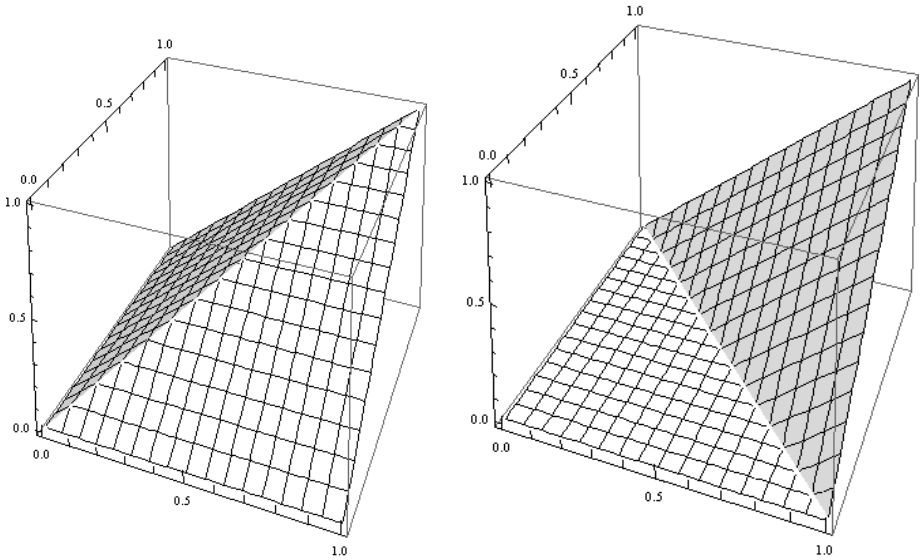
t3) $S(a,b) = S(b,a)$ (komutativita),

t4) $S(S(a,b),c)) = S(a,S(b,c))$ (asociativita).

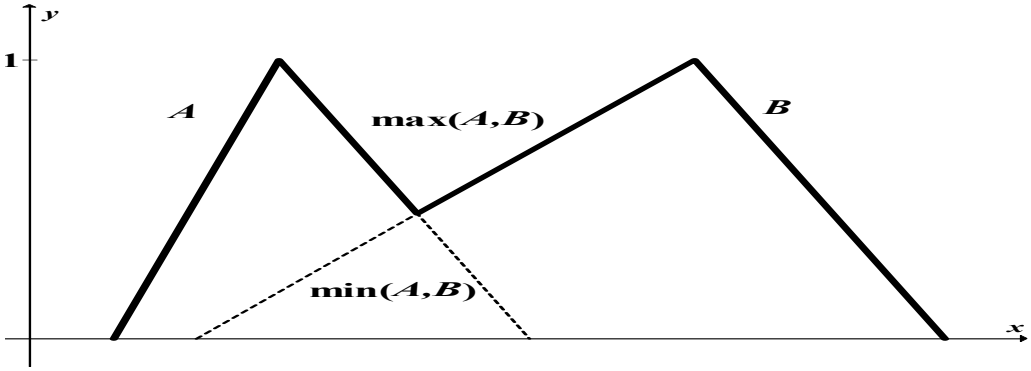
Príklad 3: Nasledujúce operácie sú t-konormy ($x,y \in [0; 1]$):

1) $S_M(x,y) = \max(x,y)$ (najmenšia t-konorma)
2) $S_P(x,y) = x+y-xy$ (súčinová t-konorma)
3) $S_L(x,y) = \min(1, x + y)$ (Lukasiewiczova t-konorma)

Ak použijeme pri prieniku a zjednotení fuzzy množín použijeme t-normu T_M = min a t-konormu S_M = max potom hovoríme o štandardnom prieniku a zjednotení (Obr. 6).



Obr. 5. Grafy t-noriem T_M, T_L.



Obr. 6. Štandardný prienik a zjednotenie fuzzy čísel A, B.

Spojitá t-norma T (t-konorma S) je ***archimedovská*** ak $T(x,x) < x$ ($S(x,x) > x$) pre všetky $x \in (0;1)$. Hovoríme, že t-norma ***má deliteľov nuly***, ak $T(x,y) = 0$ pre nejaké $(x,y) \in (0;1)^2$. Hovoríme, že t-norma T je ***striktná*** ak $T(a,c) < T(b,c)$ pre všetky $0 \le a < b \le 1, \quad 0 < c < 1$. Spojitá archimedovská t-norma, ktorá má deliteľov nuly sa nazýva ***nilpotentná*** t-norma. Poznamenajme, že T_L, a T_P sú archimedovské t-normy, T_P je striktná, T_L je nilpotentná.

Ak φ je rastúci automorfizmus intervalu [0;1]; t.j. $\varphi(0) = 0$, $\varphi(1) = 1$, φ je spojitá a rastúca. Potom φ - transformácia ľubovoľnej t–normy T

$$T_\phi(x,y) = \phi^{-1}(T(\phi(x),\phi(y))), x, y \in [0;1]$$

je tiež t-norma. Dá sa dokázať, že každá nilpotentná t-norma je φ - transformáciou Lukasiewiczovej t-normy T_L a každá spojitá striktná t-norma je φ - transformáciou súčinovej t-normy T_P .

Spojité Archimedovské t-normy je možné generovať pomocou funkcie jednej premennej – klesajúceho generátora. ***Klesajúcim generátorom*** nazývame takú spojitú ostro klesajúcu funkciu $f : [0;1] \to [0;\infty]$, že $f(1) = 0$. Poznamenajme, že symbolom $f(0) = \infty$ skrátene zapisujeme skutočnosť, že $\lim_{x \to 0+} f(x) = \infty$.

Veta 1: Spojitá t-norma je Archimedovská vtedy a len vtedy ak existuje klesajúci generátor f taký, že:

$$T(a,b) = f^{-1}(\min\{f(0), f(a) + f(b)\})$$

navyše:

- $f(0) = \infty$ práve vtedy, ak T je striktná,
- $f(0) < \infty$ práve vtedy, ak T je nilpotentná.

Teda, ak generátor f je ohraničený, potom dostávame nilpotentnú t-normu. Napríklad, $f(x)=1-x$, $x \in [0;1]$ je generátor Lukasiewiczovej t-normy T_L, $f(x) = -\ln x$ je generátor T_P. Pretože nenulový násobok generátora je generátor rovnakej t-normy, je možné generátor nilpotentnej t-normy normovať, t.j. $f(0) = 1$. Pomocou generátora možno definovať triedy t-noriem závislých od daného parametra. K najznámejším triedam patrí trieda Frankových t-noriem (s – parameter)

$$T_s^F(a,b) = \log_s\left[1 + \frac{(s^a - 1)(s^b - 1)}{s - 1}\right], \quad s > 0, \quad s \neq 1, \quad a,b \in [0;1]$$

s generátorom

$$f(x) = -\ln\frac{s^x - 1}{s - 1}, \quad s > 0, \quad s \neq 1, \quad x \in [0;1]$$

Limitou možno Frankovu triedu t-noriem doplniť:

$$T_0^F = \lim_{s \to 0} T_s^F = T_M, \quad T_1^F = \lim_{s \to 1} T_s^F = T_P, \quad T_\infty^F = \lim_{s \to \infty} T_s^F = T_L.$$

Podobne možno generovať aj t-konormu pomocou rastúceho generátora. Rastúcim generátorom nazývame takú spojitú ostro rastúcu funkciu $g : [0;1] \to [0;\infty]$, že $g(0)=0$.

Veta 2: Spojitá t-konorma S je Archimedovská vtedy a len vtedy ak existuje rastúci generátor g taký, že:

$$S(a,b) = g^{-1}(\min\{g(1), g(a) + g(b)\})$$

Aj silnú negáciu možno generovať pomocou klesajúceho generátora.

Veta 3: Unárna operácia n: $[0;1] \to [0,1]$ je silná fuzzy negácia práve vtedy, ak existuje klesajúci generátor h: $[0;1] \to [0,1]$ taký, že:

$$n(a) = f^{-1}(f(0) - f(a))$$

Ak n je silná fuzzy negácia, T je t-norma, potom binárna operácia:

$$S(a,b) = n(T(n(a), n(b)))$$

je t-konorma (duálna k t-norme T). Potom T, S, n spĺňajú zovšeobecnený De Morganov zákon ($a, b \in [0; 1]$):

$n(S(a,b)) = T(n(a),n(b))$, alebo (ekvivalentne) $n(T(a,b)) = S(n(a),n(b))$.

V tomto prípade hovoríme, že $\{T, S, n\}$ je ***duálna trojica***. Ak $n(x) = 1 - x$, potom:

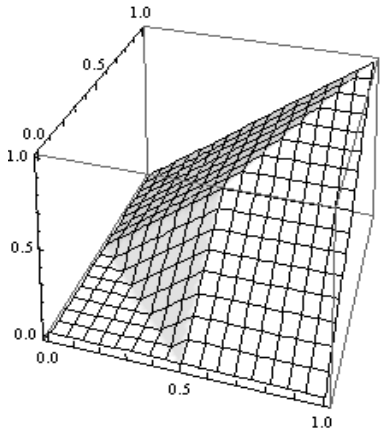
$$S(a,b) = 1 - T(1 - a, 1 - b)$$

Pripomeňme De Morganov zákon pre klasické (crisp) množiny:

$$(A \cap B)^c = A^c \cup B^c \quad \text{alebo} \quad (A \cup B)^c = A^c \cap B^c$$

Jedna z možností konštrukcie nových t-noriem spočíva vo vytváraní ordinálnych súčtov daných t-noriem [8]. Nech $\{(a_k, b_k), k \in K\}$ je trieda vzájomne disjunktných subintervalov intervalu [0;1] a $\{T_k, k \in K\}$ je trieda t-noriem rôznych od t-normy T_M. Potom ***ordinálny súčet*** $T = \{ [a_k, b_k], T_k \}_{k \in K}$ je tiež t-norma definovaná vzťahom:

$$T(x,y) = \begin{cases} a_k + (b_k - a_k)\, T\left(\dfrac{x - a_k}{b_k - a_k}, \dfrac{y - a_k}{b_k - a_k}\right) & \text{ak} \quad x, y \in [a_k, b_k] \\ \min(x,y) & \text{inak} \end{cases}$$



Obr. 7. Ordinálny súčet $\{[0;1/2], T_L\}$.

Graficky, ordinálna suma T obsahuje zmenšené grafy t-noriem T_k. Na Obr. 7 je graf ordinálneho súčtu s jedným sčítancom $\{[0;1/2], T_L\}$. Trieda Frankových t-noriem bola odvodená pri riešení problému charakterizácie t-noriem T a t-konoriem S spĺňajúcich Frankovu rovnicu (pre všetky $a, b \in [0;1]$).

$$T(a, b) + S(a, b) = a + b$$

Z Frankovej práce [5] vyplýva, že v triede spojitých t-noriem a t-konoriem jediným riešením sú Frankove t-normy a ich ordinálne súčty.

3 Fuzzy implikácie

Implikácia je dôležitá spojka v logike a je základnou operáciou v procese odvodzovania. Zovšeobecnená implikácia sa nazýva fuzzy implikácia. V klasickej dvojhodnotovej logike možno implikáciu vyjadriť vo viacerých formách, ktoré sú v takejto logike ekvivalentné. Napríklad, pre výroky p, q sú zložené výroky

$$p \Rightarrow q, \quad q' \Rightarrow p', \quad \neg p \vee q, \quad \neg p \vee (p \wedge q)$$

ekvivalentné. Samozrejme, to komplikuje zovšeobecnenie pojmu implikácie. Nasledujúcu definíciu fuzzy implikácie možno považovať za rozumný kompromis.

Definícia 4: Binárna operácia I : $[0;\ 1]^2 \to [0;\ 1]$ sa nazýva fuzzy implikácia ak pre každé $a,b,c \in [0;1]$ platí:

i1) $I(1, 0) = 0$, $I(0, 0) = I(1, 1) = 1$,

i2) $a \le b \Rightarrow I(b, c) \le I(a, c)$, $I(c, a) \le I(c, b)$.

To znamená, že fuzzy implikácia je nerastúca v prvom argumente a neklesajúca v druhom argumente. Priamo z definície plynie, že pre každé $a, b \in [0; 1]$ platí:

$$I(0, a) = I(b, 1) = 1.$$

Ľahko ukázať, že binárna operácia:

$$I(a,b) = I_{S,n}(a,b) = S(n(a),b), \quad a,b \in [0;1]$$

je fuzzy implikácia pre každú t-konormu S a silnú negáciu n. Takéto fuzzy implikácie nazývame S-implikácie.

Príklad 4: (S – implikácie)

1) $n(a) = 1 - a$, $S(a, b) = \max\{a,b\}$, potom:

$I_{S,n}(a,b) = \max\{1-a, b\})$ (Kleene-Dienesova implikácia)

2) $n(a) = 1 - a$, $S(a, b) = a + b - ab$, potom:

$I_{S,n}(a,b) = 1 - a + ab$ (Reichenbachova implikácia)

3) $n(a) = 1 - a$, $S(a, b) = \min\{1, a + b\}$, potom:

$I_{S,n}(a,b) = \min(1,1 - a + b)$ (Lukasiewiczova implikácia)

Fuzzy implikácie možno konštruovať aj iným spôsobom. Nasledujúci typ fuzzy implikácií je založený množinovej rovnosti: $A^c \cup B = \bigcup\{Z; A \cap Z \subset B\}$:

$$I_T(a,b) = \sup\{z \in [0;1],\ \text{pre ktoré } T(x,z) \le y\}, a,b \in [0;1]$$

kde T je spojitá t-norma. Dá sa dokázať, že I_T je fuzzy implikácia pre každú spojitú t-normu T. Obvykle je nazývaná ***reziduálna*** implikácia alebo, skrátene, ***R-implikácia***. Priamo z definície plynie, že:

$$I(a,b) = 1 \text{ pre každé } a,b \in [0;1],\ a \le b$$

Príklad 5: (R – implikácie)

1) $T(x,y) = \min\{a,b\}$. Potom

$$I_T(x,y) = \begin{cases} 1 & \text{ak } x \le y \\ y & \text{inak} \end{cases} \qquad (\textit{Godelova implikácia})$$

2) $T(x,y) = xy$. Potom

$$I_T(x,y) = \begin{cases} 1 & x \le y \\ y/x & otherw. \end{cases} \qquad (\textit{Goguenova implikácia})$$

3) $T(x,y) = \max(0, x + y - 1)$. Potom

$$I_T(x,y) = \min(1 - x + y, 1) \quad (\textit{Lukasiewiczova implikácia})$$

Každá reziduálna implikácia I_T má niektoré zaujímavé vlastnosti. napríklad:

$$T(x, I_T(x,y)) \le y \text{ pre všetky } x,y \in [0;1] \quad \text{(modus ponens)}$$

Nech (T,S,n) je duálna trojica, t.j. $n(T(x,y)) = S(n(x),n(y))$. Zaujímavá je otázka: Za akých podmienok platí $I_{S,n} = I_T$? Problém rieši nasledujúca veta [20].

Veta 4: Nech (T,S,n) je spojitá duálna trojica. Potom $I_{S,n} = I_T$ práve vtedy ak T je nilpotentná t-norma s normovaným klesajúcim generátorom f a $n(x) = f^{-1}(1 - f(x))$ pre každé $x \in [0;1]$.

Táto veta objasňuje, prečo v oboch predošlých príkladoch vystupuje Lukasiewiczova implikácia. Vtedy totiž platí $I_{S,n} = I_T$.

4 Klasické a fuzzy preferenčné štruktúry

4.1 Klasické dvojhodnotové preferenčné štruktúry

Modelovanie preferencií je dôležitým krokom v rozhodovacích procesoch a operačnom výskume. V procese, ktorý nazývame optimalizácia, hľadáme z možných alternatív najlepšiu alternatívu. K vyriešeniu tohto problému zostavujeme účelovú funkciu a potom hľadáme jej minimum alebo maximum. Často však nájdenie účelovej funkcie je veľmi komplikované. Vtedy je možné použiť metódu preferencie, t.j. porovnávanie alternatív po dvojiciach. Základným pojmom je pojem preferenčnej štruktúry. Uvažujme konečnú množinu alternatív $\boldsymbol{A}$ a očakávame, že porotca (posudzovateľ) ich posudzuje po dvojiciach. Napríklad, kupec K chce kúpiť dom. Zisťuje, že z prístupných ponúk (alternatív) nie je jednoduché určiť najlepšiu. Každá ponuka (alternatíva) môže mať svoje výhody i nevýhody, a preto sa rozhodne porovnávať ich po dvojiciach. Pre každú dvojicu alternatív sa kupec (posudzovateľ) rozhoduje pre jednu z možností:

- jasne preferuje jednu z alternatív,
- alternatívy sú pre neho rovnocenné (nerozlíšiteľné),
- nie je schopný alternatívy porovnať.

Nech $\boldsymbol{A}$ je množina alternatív. Posudzovateľ takto vie zostaviť tri binárne relácie $P, I, J \subset \boldsymbol{A} \times \boldsymbol{A}$, kde P je relácia striktnej preferencie, I je relácia rovnocennosti (nerozlíšiteľnosti) a J je relácia neporovnateľnosti. Pre každú dvojicu $(a,b) \in \boldsymbol{A} \times \boldsymbol{A}$ porotca rozhodne práve pre jednu z možností:

1. a preferuje pred b $\Leftrightarrow (a,b) \in P$ (preferencia)
2. b preferuje pred a $\Leftrightarrow (b,a) \in P$
3. a a b sú nerozlíšiteľné $\Leftrightarrow (a,b) \in I$ (nerozlíšiteľnosť)
4. a a b sú neporovnateľné $\Leftrightarrow (a,b) \in J$ (neporovnateľnosť)

Označme P^t binárnu reláciu definovanú vzťahom:

$$(a,b) \in P^t \Leftrightarrow (b,a) \in P$$

Je logické od binárnych relácií P,P^tI,J očakávať niektoré „rozumné“ vlastnosti a tak definovať preferenčnú štruktúru. Je zaujímavé, že vo všeobecnosti nemožno očakávať tranzitívnosť preferencie. Posudzovateľ môže preferovať alternatívu a pred alternatívou b a alternatívu b pred alternatívou c, ale nemusí preferovať a pred c. Pri každom posudzovaní mohli rozhodovať iné okolnosti. Napríklad, pri kupovaní domu to môže byť kvalita, ale aj cena, poloha a iné.

Definícia 5: Preferenčná štruktúra (PS) na množine neprázdnej množine $\boldsymbol{A}$ je trojica (P,I,J) binárnych relácií na množine $\boldsymbol{A}$ takých, že:

Ps1) P je asymetrická, t.j. $P \cap P^t = \emptyset$; I, J sú symetrické; t.j., pre každé $a,b \in \boldsymbol{A}$,

$$(a,b) \in I \Leftrightarrow (b,a) \in I,\ (a,b) \in J \Leftrightarrow (b,a) \in J$$

Ps2) I je reflexívna, P, J sú antireflexívne; t.j. pre každé, $a \in \boldsymbol{A}$

$$(a,a) \in I\ ,\ (a,a) \notin P\ ,\ (a,a) \notin J,$$

Ps3) P ,I, J sú navzájom disjunktné,

Ps4) $P \cup P^t \cup I \cup J = \boldsymbol{A} \times \boldsymbol{A}$.

Z definície priamo vyplýva, že aj binárna relácia P^t je disjunktná s binárnymi reláciami P,I,J.

Príklad 6: Posudzovateľ posudzoval po dvojiciach množinu $\boldsymbol{V} = \{a,b,c\}$ troch alternatív. Vyplnil nasledujúcu tabuľku zistených preferencií:

	a	b	c
a	0	1	0
b	0	0	1
c	0	0	0

Teda preferuje alternatívu *a* pred alternatívou *b*, alternatívu *b* pred alternatívou *c*. Oznámil, že alternatívy a a c nevie porovnať. Teda definoval preferenčnú štruktúru reprezentovanú binárnymi reláciami P, P^t ,I, J, ktoré možno reprezentovať štvorcovými maticami (označujeme ich rovnako ako relácie):

$$P = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad P^t = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad J = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

Matica P^t je transponovaná k matici P. Skontrolujte, že trojica (P, I, J) má vlastnosti $Ps1$ – $Ps4$, a teda je preferenčná štruktúra. Všimnime si v predošlom príklade, že porotca preferoval a pred b, b pred c, ale nepreferoval a pred c. Relácia preferencie nie je tranzitívna.

Každú reláciu $H \subset A \times A$ možno reprezentovať aj jej charakteristickou funkciou. Budeme ju označovať rovnakým písmenom. Teda H: $\boldsymbol{A} \times \boldsymbol{A} \to \{0, 1\}$ tak, že:

$$H(a,b) = \begin{cases} 1, & \text{ak } (a,b) \in H \\ 0, & \text{inak} \end{cases}$$

Potom možno formulovať jednoduchú definíciu preferenčnej štruktúry:

Trojica (P, I, J) binárnych relácií na konečnej množine alternatív je (PS) práve vtedy, ak:

1. I je reflexívna a symetrická,
2. $P(a,b)+P(b,a)+I(a,b)+J(a,b) = 1$, pre každé $a,b \in \boldsymbol{A}$.

Preferenčnú štruktúru možno zadať aj jednoduchšie. Namiesto 3 relácii stačí zadať reláciu

$$R = P \cup I.$$

Teda $(a,b) \in R$, ak porotca preferuje a pred b alebo ich nevie rozlíšiť. Nazývame ju ***rozšírená preferencia***. Je zrejmé, že R musí byť reflexívna, t.j. $(a,a) \in R$ pre každú alternatívu a (matica R musí mať na hlavnej diagonále jednotky). Označme symbolom R^c komplementárnu reláciu k relácii R, t.j.:

$$R^c = \boldsymbol{A} \times \boldsymbol{A} - R$$

Teda štvorcová matica R^c vznikne z matice R výmenou núl za jednotky a naopak. Ľahko sa dá ukázať, že:

$$(R^t)^c = R^d = P \cup J$$

a potom

$$P = R \cap R^d,\ I = R \cap R^t,\ J = R^c \cap R^d,$$

kde zjednotenie možno reprezentovať maximom matíc a prienik minimom matíc. V predošlom príklade

$$R = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \quad R^t = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \quad R^c = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} \quad R^d = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

Ľahko overiť, že:

$$P = \min\{R, R^d\},\ I = \min\{R, R^t\},\ J = \min\{R^c, R^d\}$$

Tento postup umožňuje zostrojiť (PS) pomocou jedinej reflexívnej binárnej relácie R. Porotcovi v predošlom príklade stačilo vyplniť tabuľku zodpovedajúcu matici R.

4.2 Fuzzy preferenčné štruktúry

Preferenčné štruktúry s dvojhodnotovými binárnymi reláciami neumožňujú zachytiť stupeň presvedčenia posudzovateľa o preferencii alternatívy pred inou alternatívou. Ešte horšia je situácia, ak rozhodnutie o preferencii je kompromisom viacerých posudzovateľov. To obmedzuje praktické použitie klasických preferenčných štruktúr. Preto snaha definovať fuzzy preferenčné štruktúry je veľmi prirodzená. V takýchto štruktúrach nahradíme binárne relácie na množine alternatív $\boldsymbol{A}$, reprezentované ich charakteristickou funkciou, binárnou fuzzy reláciou

$$H: \boldsymbol{A} \times \boldsymbol{A} \to [0, 1]$$

Teda, ak $a,b \in \boldsymbol{A}$, potom $0 \le H(a,b) \le 1$. Napríklad, skutočnosť, že $P(a,b) = 0{,}6$ možno interpretovať tak, že 60 percent posudzovateľov preferuje alternatívu a pred alternatívou b. Vzťahy a operácie s binárnymi fuzzy reláciami vyžadujú použitie logických fuzzy spojok; fuzzy negácie, fuzzy konjunkcie a fuzzy disjunkcie. Už vieme, že "rozumným" zovšeobecnením týchto spojok sú t-normy, t-konormy a silné fuzzy negácie. Uvažujme ***duálnu trojicu*** $\{T, S, n\}$. Pre jednoduchosť, budeme používať len štandardnú fuzzy negáciu,

$$n(\mathrm{x}) = 1 - x,\ x \in [0; 1]$$

Potom t-konorma S je duálna ku t-norme T, t.j., pre každé $a,b \in [0; 1]$ platí:

$$S(a,b) = 1 - T(1-a, 1-b)$$

To nám umožní definovať fuzzy preferenčnú štruktúru (FPS) ako zovšeobecnenie (fuzzifikáciu) preferenčnej štruktúry. Budeme písať $H = T(F,G)$, ak:

$$H(a,b) = T(F(a,b), G(a,b)) \text{ pre všetky } (a,b) \in \boldsymbol{A} \times \boldsymbol{A}.$$

Analogicky $H = S(F,G)$. Ak $T(F(a,b), G(a,b)) = 0$ pre všetky $(a,b) \in \boldsymbol{A} \times \boldsymbol{A}$, zapíšeme $T(F,G) = \varnothing$. Teda:

$$\varnothing : \boldsymbol{A} \times \boldsymbol{A} \to \{0\}$$

Symbolom $\boldsymbol{A}^2$ označíme binárnu fuzzy reláciu na $\boldsymbol{A}$ takú, že:

$$\boldsymbol{A}^2 : \boldsymbol{A} \times \boldsymbol{A} \to \{1\},$$

Označme $P^t(a,b) = P(b,a)$. Každú binárnu fuzzy reláciu na konečnej množine $\boldsymbol{A}$ možno interpretovať pomocou štvorcovej matice obsahujúcej prvky z intervalu [0; 1]. Binárna fuzzy relácia $\emptyset$ je reprezentovaná štvorcovou maticou, ktorej všetky prvky sú 0. Binárna fuzzy relácia $\boldsymbol{A}^2$ je reprezentovaná štvorcovou maticou, ktorej všetky prvky sú 1.

Definícia 6: Fuzzy preferenčná štruktúra (FPS) na množine neprázdnej množine $\boldsymbol{A}$ je trojica (P, I, J) binárnych fuzzy relácií na množine $\boldsymbol{A}$ takých, že:

Fs1) P je asymetrická t.j., $T(P, P^t) = \emptyset$; I,J sú symetrické; t.j., pre každé, $a, b \in \boldsymbol{A}$

$$I(a,b) = I(b,a),\ J(a,b) = J(b,a),$$

Fs2) I je reflexívna, P, J sú antireflexívne; t.j. pre každé , $a \in \boldsymbol{A}$

$$I(a,a) = 1 \in I,\ P(a,a) = 0,\ J(a,a) = 0,$$

Fs3) $T(P, I) = \emptyset$, $T(P, J) = \emptyset$, $T(J, I) = \emptyset$,

Fs4) $S(P, P^t, I, J) = \boldsymbol{A} \times \boldsymbol{A}$.

Vlastnosť $Fs4$ je zovšeobecnením (fuzzifikáciou) rovnosti $Ps4$: $P \cup P^t \cup I \cup J = \boldsymbol{A}^2$. V prípade klasických binárnych relácii ju možno nahradiť viacerými rovnosťami; napríklad $(P \cup P^t) = I \cup J$ alebo $(P \cup I)^c = P^t \cup J$ a inými. To ukazuje, že problém "rozumnej" definície (FPS) nemusí byť jednoduchý. [1], [2], [4]. Ešte väčší problém priniesol dôkaz nasledujúcej vety.

Veta 5: Nech v definícii (FPS) je T striktná spojitá t-norma a S je k nej duálna t-konorma, ($S(a,b) = 1 - T(1-a, 1-b)$). Potom takáto (FPS) je klasická preferenčná štruktúra.

Veta hovorí, že použitím striktnej t-normy a jej duálnej t-konormy v definícii (FPS) môžeme dostať len klasickú (PS). Teda, v takomto prípade nemožno nájsť fuzzy binárne relácie P, I, J (nadobúdajúce viac ako 2 hodnoty) vyhovujúce definícii (FPS). Teda obmedzíme sa na spojité t-normy, ktoré nie sú striktné. K takýmto t-normám patria nilpotentné t-normy. Nilpotentné t-normy sú spojité archimedovské t-normy, ktoré majú deliteľov nuly. Už vieme, že každá nilpotentná t-norma je φ - transformáciou Lukasiewiczovej t-normy T_L. preto sa v ďalších úvahách obmedzíme na Lukasiewiczovu t-normu T_L a duálnu t-konormu S_L. Potom možno formulovať definíciu preferenčnej štruktúry jednoduchšie. Nasledujúcu vetu možno používať ako zjednodušenú definíciu (FPS).

Veta 6: Trojica (P, I, J) binárnych fuzzy relácií na konečnej množine alternatív je (FPS) práve vtedy, ak I je reflexívna a symetrická a:

$$P(a,b) + P(b,a) + I(a,b) + J(a,b) = 1, \text{ pre každé } a, b \in \boldsymbol{A}.$$

Príklad 7: Posudzovatelia posudzovali po dvojiciach množinu $\boldsymbol{V} = \{a, b, c\}$ troch alternatív. Dohodli sa na nasledujúcich maticiach reprezentujúcich preferenciu, nerozlíšiteľnosť a neporovnateľnosť alternatív a, b, c:

$$P=\begin{bmatrix}0 & 0.2 & 0.1\\ 0 & 0 & 0\\ 0 & 0.6 & 0\end{bmatrix} \qquad I=\begin{bmatrix}1 & 0.1 & 0.7\\ 0.1 & 1 & 0.2\\ 0.7 & 0.2 & 1\end{bmatrix} \qquad J=\begin{bmatrix}0 & 0.7 & 0.2\\ 0.7 & 0 & 0.2\\ 0.2 & 0.2 & 0\end{bmatrix}$$

Napríklad 20% posudzovateľov preferuje alternatívu *a* pred *b*, 10% posudzovateľov nevie rozlíšiť alternatívy *a, b* a 70% posudzovateľov nevie porovnať alternatívy *a, b*. Ľahko sa presvedčíme, že trojica (*P, I, J*) je (FPS). Okrem reflexívnosti *I*, anti-reflexívnosti *P*, *J*, symetrii *I, J*, asymetrii *P*, vidíme, že:

$$P(a,b)+P(b,a)+I(a,b)+J(a,b) = 1 \text{ pre každé } a,b \in V.$$

Vidíme tiež, že zostavenie (FPS) nie je jednoduché. Preto, myšlienka fuzzifikovať konštrukciu (PS) pomocou rozšírenej preferencie $R = P \cup I$ tak , že:

$$(P,I,J) = (R \cap R^d, R \cap R^t, R^c \cap R^d)$$

je veľmi lákavá. Bohužiaľ, nasledujúca [1], [2], [4] hovorí, že to nie je možné jednoduchou fuzzifikáciou prieniku pomocou t-normy.

Veta 7: Nech *T* je spojitá t-norma a *S* je k nej duálna t-konorma. Potom nie je možné, aby pre každú reflexívnu binárnu fuzzy reláciu platila R trojica:

$$(P,I,J) = (T(R, R^d,)\ T(R, R^t)\ T(R^c, R^d)) \text{ a } R = S(P,I).$$

Negatívny výsledok predošlej vety neumožňuje zostrojiť (FPS) jednoduchou fuzzifikáciou pomocou jedinej spojitej t-normy *T* a duálnej t-konormy *S*. Fodor, Roubens, De Baets [1], [2], [4] a ďalší navrhli nasledujúcu konštrukciu (FPS):

- hodnoty $P(a,b)$, $I(a,b)$, $J(a,b)$ závisia len na hodnotách $R(a,b)$, a $R(b,a)$, t.j., existujú binárne operácie (zobrazenia) p,i,j: $[0;1]\times[0;1]\to[0;1]$ tak, že pre všetky $a,b \in [0;1]$ platí:

$$P(a,b) = p(R(a,b), R(b,a)),$$
$$I(\mathrm{a,b}) = i(R(a,b), R(b,a)),$$
$$J(\mathrm{a,b}) = j(R(a,b), R(b,a)),$$

- zobrazenia $p(x, 1\text{-}y)$, $i(x, y)$, $j(1 - x, 1 - y)$ sú neklesajúce v argumentoch *x* a *y*,
- *i* a *j* sú symetrické, t.j., $i(x, y) = i(y, x)$, $j(x, y) = j(y, x)$,
- (*P, I, J*) je (FPS) pre každú reflexívnu binárnu fuzzy reláciu *R* na množine alternatív
- $S_L(P, I) = R$.

Takto definovanú trojicu (*p,i,j*) binárnych operácií na [0;1] nazývame ***monotónna generujúca trojica*** (FPS). Dá sa tiež dokázať, pre takúto trojicu a pre všetky $a,b \in [0;1]$ platí:

$$T_L(a,b) \le p(\mathrm{x,1\text{-}y}),\ i(a,b), j(1\text{-}a,1\text{-}b) \le S_L(a,b)$$

Nasledujúca veta [1], [4], [17] nám umožňuje jednoduchým spôsobom nájsť monotónne generujúce trojice (*p,i,j*). Ešte poznamenajme, že binárnu operáciu $C:[0;1]\times[0;1]\to[0;1]$ nazývame ***kvázikopulou*** [13], ak $C(x,0) = C(0,x) = 0$, $C(x,1) = C(1,x) = x$ pre každé $x \in [0;1]$, *C* je neklesajúca v oboch argumentoch a *C* je 1-Lipschitz, t.j:

$$C(a_2,b_2) - C(a_1,b_1) \le a_2 - a_1 + b_2 - b_1,$$

pre každé $0 \le a_1 \le a_2 \le 1$, $0 \le b_1 \le b_2 \le 1$. Kvázikopula môže ale nemusí byť t-normou (nemusí byť komutatívna ani asociatívna) a naopak t-norma nemusí byť kvazikopulou. Avšak, t-normy T_L, T_M , T_P sú kvázikopule [13], [14].

Veta 8: [17]. Uvažujme binárnu operáciu $p:[0;1]\times[0;1]\to[0;1]$ takú, že:

1) Binárna operácia $C:[0;1]\times[0;1]\to[0;1]$ daná predpisom $C(x,y) = p(x,1\text{-}y)$ je kvázikopula,
2) $p(x,y) - p(y,x) = x - y$ pre každé $x,y \in [0;1]$,

Potom trojica (p,i,j), kde:

$$i(x,y) = x - p(x,y), \quad j(x,y) = 1 - y - p(x,y)$$

pre každé $x,y \in [0;1]$ je monotónna generujúca trojica (FPS).

Príklad 8: Položme $p(x,y) = \max\{x - y, 0\}$. Potom $C(x,y) = p(x,1-y) = \max\{0, x + y - 1\} = T_L(x,y)$ je kvázikopula. Ľahko sa presvedčíme, že pre všetky $x,y \in [0;1]$ platí:

$$p(x,y) - p(y,x) = \max\{x - y, 0\} - \max\{y - x, 0\} = x - y.$$

Položme

$$i(x,y) = x - p(x,y) = x - \max\{0, x - y\} = \min\{x, y\},$$

$$j(x,y) = 1 - y - p(x,y) = 1 - y - \max\{0, x - y\} = \min\{1 - x, 1 - y\}.$$

Podľa predošlej vety trojica (p,i,j) je monotónna generujúca trojica (FPS). Teda pre každú reflexívnu binárnu fuzzy reláciu R generuje (FPS):

$$P(a,b) = p(R(a,b), R(b,a)) = \max\{R(a,b) - R(b,a), 0\},$$

$$I(a,b) = i(R(a,b), R(b,a)) = \min\{R(a,b), R(b,a))\},$$

$$J(a,b) = j(R(a,b), R(b,a)) = \min\{1 - R(a,b), 1 - R(b,a))\}.$$

Napríklad, ak:

$$R = \begin{bmatrix} 1 & 0.3 & 0.8 \\ 0{,}1 & 1 & 0.2 \\ 0.7 & 0.8 & 1 \end{bmatrix}$$

potom

$$P = \begin{bmatrix} 0 & 0.2 & 0.1 \\ 0 & 0 & 0 \\ 0 & 0.6 & 0 \end{bmatrix} \quad I = \begin{bmatrix} 1 & 0.1 & 0.7 \\ 0.1 & 1 & 0.2 \\ 0.7 & 0.2 & 1 \end{bmatrix} \quad J = \begin{bmatrix} 0 & 0.7 & 0.2 \\ 0.7 & 0 & 0.2 \\ 0.2 & 0.2 & 0 \end{bmatrix}$$

Všimnime si, toto je práve (FPS) z Príkladu 7.

V nasledujúcom príklade použijeme monotónnu generujúcu trojicu tak, že žiadna z binárnych operácii p,i,j nie je symetrická ani asociatívna.

Príklad 9: Položme $p(x,y) = \max\{x\text{-}2xy, x - y, 0\}$. Potom $C(x,y) = p(x,1-y) = \max\{0, x + y - 1, 2xy - x\} = T_L(x,y)$ je kvázikopula. Ľahko sa presvedčíme, že pre všetky $x,y \in [0;1]$ platí $p(x,y) - p(y,x) = x - y$. Položme:

$$i(x,y) = x - p(x,y)$$

$$j(x,y) = 1 - y - p(x,y)$$

Podľa predošlej vety (p,i,j) je monotónna generujúca trojica. Teda, pre každú reflexívnu binárnu fuzzy reláciu R generujú (FPS). Napríklad, reflexívnu binárnu fuzzy reláciu R z predošlých príkladov:

$$R = \begin{bmatrix} 1 & 0.3 & 0.8 \\ 0.1 & 1 & 0.2 \\ 0.7 & 0.8 & 1 \end{bmatrix},$$

potom:

$$P = \begin{bmatrix} 0 & 0.24 & 0.1 \\ 0.04 & 0 & 0 \\ 0 & 0.6 & 0 \end{bmatrix}, \quad I = \begin{bmatrix} 1 & 0.06 & 0.7 \\ 0.06 & 1 & 0.2 \\ 0.7 & 0.2 & 1 \end{bmatrix}, \quad J = \begin{bmatrix} 0 & 0.66 & 0.2 \\ 0.66 & 0 & 0.2 \\ 0.2 & 0.2 & 0 \end{bmatrix}.$$

Zaujímavá je monotónna generujúca trojica (*p, i, j*), kde

$$p(x,y) = x(1-y), \quad i(x,y) = xy, \quad j(x,y) = (1-x)(1-y).$$

Teda (FPS) = (*P,I,J*) tak, že:

$$P(a,b) = R(a,b)(1-R(b,a)),$$
$$I(a,b) = R(a,b).R(b,a),$$
$$J(a,b) = (1-R(a,b))(1-R(b,a)).$$

Záverom poznamenajme, že z danej preferenčnej štruktúry je možné určiť aj na zostavenie účelovej funkcie, napríklad vhodnou kombináciou zložiek (FPS) [6].

5 Fuzzy regulátory

Klasický regulátor je zariadenie, ktoré sa stará, aby parameter (parametre) regulovaného systému nadobúdal požadované hodnoty. Regulátor dostáva informáciu o stave sledovaného systému (vstup) a navrhuje alebo nastavuje riadiaci zásah (výstup), vstupu $x \in \boldsymbol{X}$ priradí výstup $y \in \boldsymbol{Y}$. Teda regulátor vykonáva zobrazenie:

$$\varphi : \boldsymbol{X} \rightarrow \boldsymbol{Y},$$

kde $\boldsymbol{X}$ je vstupný priestor a $\boldsymbol{Y}$ je výstupný priestor. Regulátor býva často zložité zariadenie, využívajúce komplikovaný matematický aparát, často rieši systémy diferenciálnych rovníc. Aby systém nebol príliš pomalý, často je nutné silné zjednodušenie s rizikom, že potom nepopisuje niektoré dôležité vzťahy. Jeho ďalším najväčším problémom je presnosť meraných hodnôt.

Na inom princípe pracujú fuzzy regulátory. Fuzzy regulátor používa bázu pravidiel, ktoré sú vyjadrením vzťahu medzi jazykovými premennými [23]. Hodnoty jazykovej premennej nie sú čísla, ale slová (väčšinou prídavné mená). Napríklad jazyková premenná X = vek človeka a jej hodnoty sú:

$H(X)$ = {mladý, veľmi mladý, starý, veľmi starý , ...}.

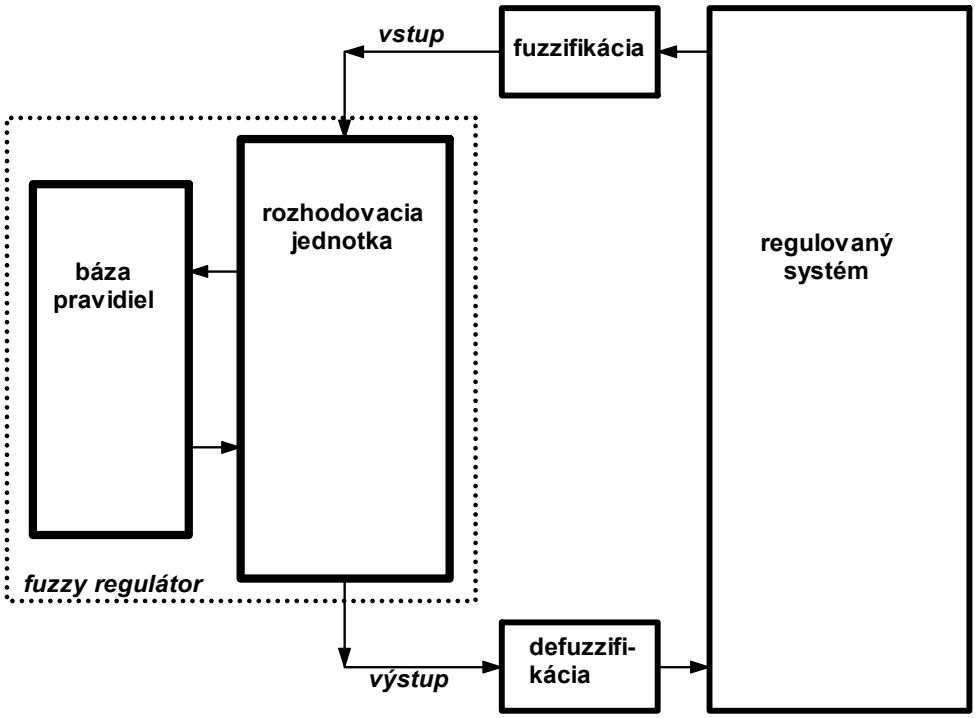
Iný príklad, jazyková premenná Y je teplota a jej hodnoty:

$H(Y)$ = { nízka, zvýšená, vysoká, ...}.

Každú hodnotu jazykovej premennej možno vyjadriť pomocou fuzzy množiny na príslušnej univerzálnej množine. Teda fuzzy regulátor priraďuje fuzzy podmnožine A univerzálnej množiny $\boldsymbol{X}$ fuzzy podmnožiny C univerzálnej množiny $\boldsymbol{Y}$. Regulačný proces je daný prenosovou funkciou:

$$\Phi : F(\boldsymbol{X}) \quad \rightarrow \quad F(\boldsymbol{Y}),$$

kde $F(\boldsymbol{X})$ resp. $F(\boldsymbol{Y})$ sú triedy všetkých fuzzy podmnožín univerzálnych množín $\boldsymbol{X}$ resp. $\boldsymbol{Y}$. Teda danému vstupu $A \in F(\boldsymbol{X})$ priradí výstup $C \in F(\boldsymbol{Y})$. Zjednodušená schéma regulovaného systému je na Obr. 8.



Obr. 8. Schéma regulačného obvodu

V ďalšom budeme fuzzy regulátorom budeme jadro fuzzy regulátora pozostávajúce z inferečného (odvodzovacieho) bloku a bázy pravidiel. Nebudeme sa venovať problému fuzzifikácie vstupu a defuzzifikácie výstupu. Vstup aj výstup regulátora môže byť jeden parameter alebo vektor parametrov. Ak je výstup n-rozmerný vektor, možno regulátor rozložiť na n regulátorov. Teda budeme uvažovať len regulátor MISO (Multiple Input Single Output). Teda vstupný priestor $\boldsymbol{X}$ môže n-rozmerný, výstup jednorozmerný. Vstup do regulátora býva často konkrétne číslo (crisp) z priestoru $\boldsymbol{X}$. Potom ho možno fuzzifikovať ako singleton (charakteristická funkcia jednoprvkovej množiny) alebo spojite, ako fuzzy číslo, podľa stupňa neistoty merania. Výstup regulátora je fuzzy množina, ktorú treba defuzzifikovať. Napríklad, nájdením prvej súradnice ťažiska plochy pod grafom funkcie príslušnosti [9], [12].

Báza pravidiel obvykle pozostáva z konečného počtu pravidiel typu:

AK..., POTOM...

Tento typ pravidiel je zovšeobecnením klasického odvodzovacieho pravidla ***modus ponens***. V klasickej (dvojhodnotovej) logike je modus ponens daný tautológiou:

$$[p \wedge (p \Rightarrow q)] \Rightarrow q.$$

Ak vieme, že platí implikácia $p \Rightarrow q$ a výrok p, potom musí platiť aj výrok q. Zapisujeme ho pomocou schémy:

Pravidlo: $p \Rightarrow q$
Pozorovanie: p
———————————
Záver: q

Zovšeobecnením dostávame odvodzovacie pravidlo pre fuzzy výroky. Uvedieme najskôr príklad:

Pravidlo: Ak je rýchlosť vysoká, treba prudko brzdiť.
Pozorovanie: Rýchlosť je dosť vysoká.
———————————
Záver: Treba dosť prudko brzdiť.

V našom príklade jazyková premenná X = rýchlosť môže nadobúdať slovné hodnoty:

$H(X)$ = { nízka, zvýšená, pomerne vysoká, vysoká, ...}

a jazyková premenná Y = brzdenie môže nadobúdať slovné hodnoty

$H(Y)$ = {prudké, mierne, žiadne, veľmi prudké, ...}.

Poznamenajme znova, že každú hodnotu jazykovej premennej možno vyjadriť pomocou fuzzy množiny na univerzálnej množine. $\boldsymbol{X}$ resp. $\boldsymbol{Y}$. Napríklad, v našom príklade $\boldsymbol{X}$ = [0; 200] (km/hod), $\boldsymbol{Y}$ = [0, 50] (uhol stlačenia brzdového pedála v stupňoch). Zovšeobecnené odvodzovacie pravidlo typu AK.....,. POTOM..... má tvar:

AK X je A, POTOM Y je C,

kde A je fuzzy podmnožina univerzálnej množiny $\boldsymbol{X}$ vyjadrujúca slovnú hodnotu jazykovej premennej X a C je fuzzy podmnožina univerzálnej množiny $\boldsymbol{Y}$ vyjadrujúca slovnú hodnotu jazykovej premennej Y. Prvá časť pravidla je predpoklad (antecedent) a druhá časť je záver (consequent). Zovšeobecnená odvodzovacia schéma má teda tvar:

Pravidlo: AK X je A, POTOM Y je C.
Pozorovanie (vstup): X je A'.
———————————
Záver (výstup): Y je C'.

Všimnime si, že vstupná fuzzy množina A' sa nemusí zhodovať s antecedentom (predpokladom) - fuzzy množinou A.

Ak chceme proces odvodzovania automatizovať a naprogramovať, je vhodné definovať fuzzy reláciu $R = R(A,C)$ popisujúci vzťah jazykových premenných X, Y vyjadrený pravidlom. Potom, pre vstupnú fuzzy množinu A' odvodzujeme výstupnú fuzzy množinu C' pomocou kompozičného pravidla. Aktuálny výstup $C' \in F(Y)$ je možné dostať kompozíciou vstupu $A' \in F(\mathbf{Y})$ a fuzzy relácie $R \in F(\boldsymbol{X} \times \boldsymbol{Y})$, t.j. $C' = A' \circ R$. Štandardná kompozícia je definovaná vzťahom:

$$C'(y) = \sup_{x \in \boldsymbol{X}} \min\{A'(x), R(x,y)\}, \quad y \in \boldsymbol{Y} \qquad \text{(SM)}$$

Hovoríme, že C' je sup-min kompozíciou vstupu A' a fuzzy relácie R; $C' = A' \circ R$. Ak operáciu minima nahradíme t-normou T, potom:

$$C'(y) = \sup_{x \in \boldsymbol{X}} T(A'(x), R(x,y)), \quad y \in \boldsymbol{Y} \qquad \text{(ST)}$$

Zapisujeme $C' = A' \circ_T R$ a hovoríme, že C' je sup–T kompozícia vstupu A' a fuzzy relácie R.

Ak univerzálne množiny $\boldsymbol{X}$, $\boldsymbol{Y}$ sú konečné a ich kardinality (počty prvkov) sú: $|\boldsymbol{X}| = m$ resp. $|\boldsymbol{Y}| = n$. Potom fuzzy podmnožiny $\boldsymbol{X}$ možno zadať pomocou m-rozmerného vektora a fuzzy podmnožiny $\boldsymbol{Y}$ možno zadať pomocou n-rozmerného vektora a relácia $R \in F(\boldsymbol{X} \times \boldsymbol{Y})$ je zadaná maticou typu $m \times n$. Kompozícia $C' = A' \circ R$ daná vzťahom (SM) je vlastne max-min násobenie matíc a kompozícia $C' = A' \circ_T R$ je max-T násobenie matíc:

$$C'(y) = \max_{x \in \boldsymbol{X}} \min\{A'(x), R(x,y)\}, \quad y \in \boldsymbol{Y} \qquad \text{(MM)}$$

$$C'(y) = \max_{x \in \boldsymbol{X}} T(A'(x), R(x,y)), \quad y \in \boldsymbol{Y} \qquad \text{(MT)}$$

Príklad 10: Nech $\boldsymbol{X} = \{1, 2, 3, 4\}$, $\boldsymbol{Y} = \{a, b, c\}$ Nech A' = {0.3/1, 0.8/2, 1/3, 0.1/4} = (0.3, 0.8, 1, 0.1), nech:

$$R = \begin{pmatrix} 0.6 & 1 & 0.4 \\ 0.8 & 1 & 0.6 \\ 0.7 & 1 & 0.5 \\ 1 & 1 & 0.9 \end{pmatrix},$$

potom:

$$C' = A' \circ R = (0.3,\ 0.8,\ 1, 0.1) \circ \begin{pmatrix} 0.6 & 1 & 0.4 \\ 0.8 & 1 & 0.6 \\ 0.7 & 1 & 0.5 \\ 1 & 1 & 0.9 \end{pmatrix} = (0.8, 1, 0.6)$$

Teda $C' = \{0.8/a, 1/b, 0.6/c\}$. Ak použijeme t-normu T_P – súčin a vzťah (MT), potom:

$$C' = A' \circ_T R = (0.3,\ 0.8,\ 1, 0.1) \circ_T \begin{pmatrix} 0.6 & 1 & 0.4 \\ 0.8 & 1 & 0.6 \\ 0.7 & 1 & 0.5 \\ 1 & 1 & 0.9 \end{pmatrix} = (0.7, 1, 0.5).$$

Vráťme sa k báze pravidiel fuzzy regulátora B. Predpokladajme, že obsahuje k pravidiel typu AK.... POTOM...:

P_1 : AK X je A_1 POTOM Y je C_1
P_2 : AK X je A_2 POTOM Y je C_2
.....
P_k: AK X je A_k POTOM Y je C_k

V snahe použiť kompozičné pravidlo odvodzovania hľadáme takú reláciu $R \in F(\boldsymbol{X} \times \boldsymbol{Y})$, že fuzzy výstup C' dostaneme kompozíciou fuzzy vstupu A' a relácie R.

Najskôr rozoberieme prípad, keď báza obsahuje len jedno pravidlo tvaru:

AK X je A POTOM Y je C.

Pretože fuzzy pravidlo je implikácia, relácia R má v podstate modelovať fuzzy implikáciu. Obvykle je používaná reziduálna implikácia I_T. Potom príslušná relácia má tvar:

$$R_{res}(x,y) = I_T(A(x), B(y)),$$

a štandardný fuzzy výstup:

$$C'(y) = \sup_{x \in X} \min\{A'(x), R_{res}(x,y)\}, \quad y \in \boldsymbol{Y}.$$

Je zrejmé, že klasický regulátor s číselným vstupom a výstupom možno považovať za generátor reálnej funkcie. Reálnu funkciu f možno zadať bázou pravidiel AK..., POTOM... tvaru:

$$\text{AK } x = x_0 \text{ POTOM } y_0 = f(x_0).$$

Vstupu x_0 je priradený výstup y_0. Funkcia $f \subset D(f) \times H(f)$ a

$$f = \{(x,y): x \in D(f) \wedge y = f(x) \in H(f)\}.$$

Pritom je zrejmé, že f je reláciou odvodenou pomocou logickej konjunkcie. Mamdani-Assilianov regulátor používa na modelovanie relácie R štandardnú konjunkciu minimum, nie implikáciu. (Ak $p = 1$, potom výroky $p \Rightarrow q$, $p \wedge q$ sa zhodujú).
Teda:

$$R(x,y) = \min\{A(x), C(y)\}, \quad (x,y) \in \boldsymbol{X} \times \boldsymbol{Y}.$$

Ak A' je aktuálna vstupná fuzzy množina pravidla, potom výstupnú fuzzy množinu C' nájdeme pomocou štandardného kompozície:

$$C'(y) = \sup_{x \in X} \min\{A'(x), \min\{A(x), C(y)\}\}, \quad y \in Y.$$

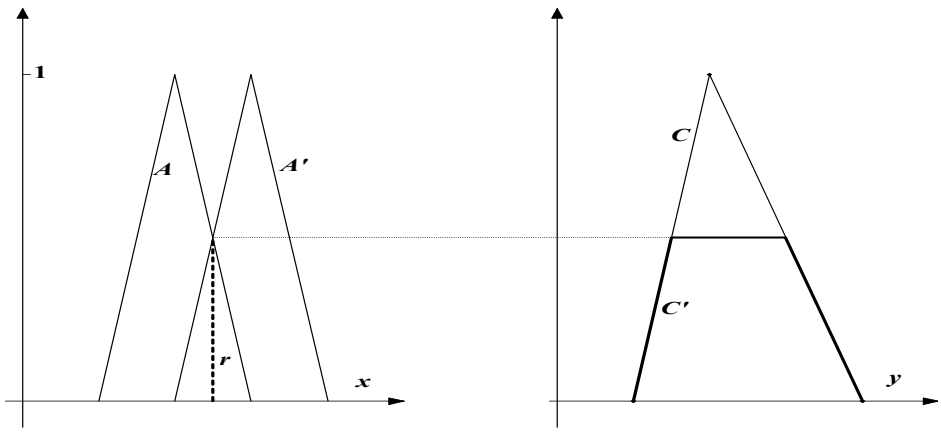
Vďaka asociatívnosti minima možno predchádzajúci vzťah upraviť do tvaru:

$$C'(y) = \min\{\sup_{x \in X} \min\{A'(x), A(x)\}, C(y)\}\} = \min\{r, C(y)\}, \quad y \in Y.$$

Symbolom r sme označili hodnotu:

$$r = \sup_{x \in X} \min\{A'(x), A(x)\}.$$

Vidíme, že r je závisí od vstupu A' a antecedentu A; $r = r(A,A')$. Toto číslo nazývame stupňom zapálenia (firing value) pravidla alebo relevanciou pravidla. Možno ho považovať za stupeň zhody (prekrytia) aktuálneho vstupu A' a predpokladu pravidla A (Obr. 9).



Obr. 9. Odvodenie výstupu Mamdaniho metódou

Vzťah:

$$C'(y) = \min\{r, B(y)\}, \quad y \in Y.$$

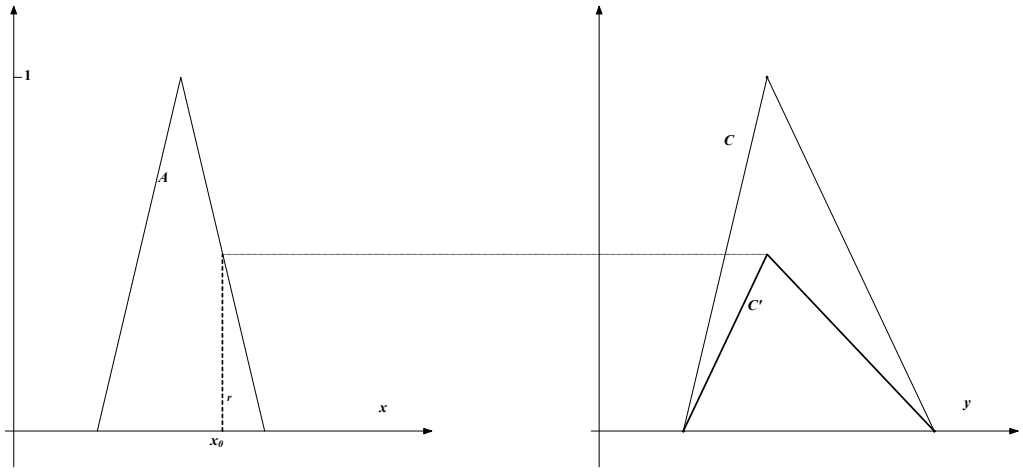
Je prepisom vzťahu (MM). Analogicky možno prepísať vzťah (MT):

$$C'(y) = T(r, B(y)), \quad y \in \boldsymbol{Y},$$

kde T je t-norma a stupeň zapálenia:

$$r = \sup_{x \in X} T(A'(x), A(x)).$$

Napríklad, ak aktuálny vstup A' je crisp (singleton), t.j. A' je charakteristická funkcia jednoprvkovej množiny $\{x_0\}$ a t-norma $T = T_P$ (súčin), potom $r = A(x_0)$ a odvodenie výstupu je na Obr. 10.



Obr. 10. Odvodenie výstupu pri použití súčinu.

Zaoberali sme sa fuzzy regulátorom založenom na reziduálnej implikácii a fuzzy regulátorom založenom na fuzzy konjunkcii (Mamdani-Assilian). Porovnanie týchto prístupov väčšinou preferuje Mamdani-Assiliánov regulátor. Jeho prednosťou je i to, že je menej výpočtovo zložitý.

Vráťme sa teraz k báze pravidiel $\mathbb{B}$, ktorá obsahuje k pravidiel. $k > 1$. Teda:

$$\mathbb{B} = \{P_1, P_2, \ldots P_k\}.$$

Nech A' je vstupná fuzzy množina. Pre každú pravidlo fuzzy množinu vieme určime individuálny výstup C_i pomocou konjunkcie – t-normy T. Teda:

$$C_i'(y) = T(r_i, C_i(y)), \quad y \in Y,$$

kde r_i je stupeň zapálenia pravidla P_i., t,j.:

$$r_i = \sup_{x \in X} T(A_i'(x), A(x)).$$

Celkový výstup fuzzy regulátora C' pre daný vstup A' dostaneme procesom, ktorý nazývame agregácia (definujeme neskôr). Budeme používať bodovú agregáciu, t.j.:

$$C'(y) = \mathrm{Agg}\{C_1'(y), C_2'(y), \ldots, C_k'\}.$$

Najpodstatnejšia požiadavka pre tento proces je, aby nezapálené pravidlo neovplyvňovalo celkový výsledok. Pre Mamdani-Assiliánov regulátor platí:

Ak $r_i = 0$, potom $C_i'(y) = 0$ pre každé $y \in Y$.

Z toho plynie, že číslo 0 musí byť neutrálny prvok agregácie Agg. Takouto agregačnou funkciou je maximum. Pre nezáporné čísla z intervalu [0;1] platí:

$$\max\{a_1, a_2, ..., a_i, a_{i+1}, ..., a_k\} = \max\{a_1, a_2, ..., a_i, 0, a_{i+1}, ..., a_k\}$$

Teda, štandardný celkový výstup C':

$$C'(y) = \max\ \{\ C_1'(y), C_2'(y), \ldots, C_k'(y)\}$$

alebo:

$$C'(y) = \max_i \{\min(r_i, C_i(y)\}\ .$$

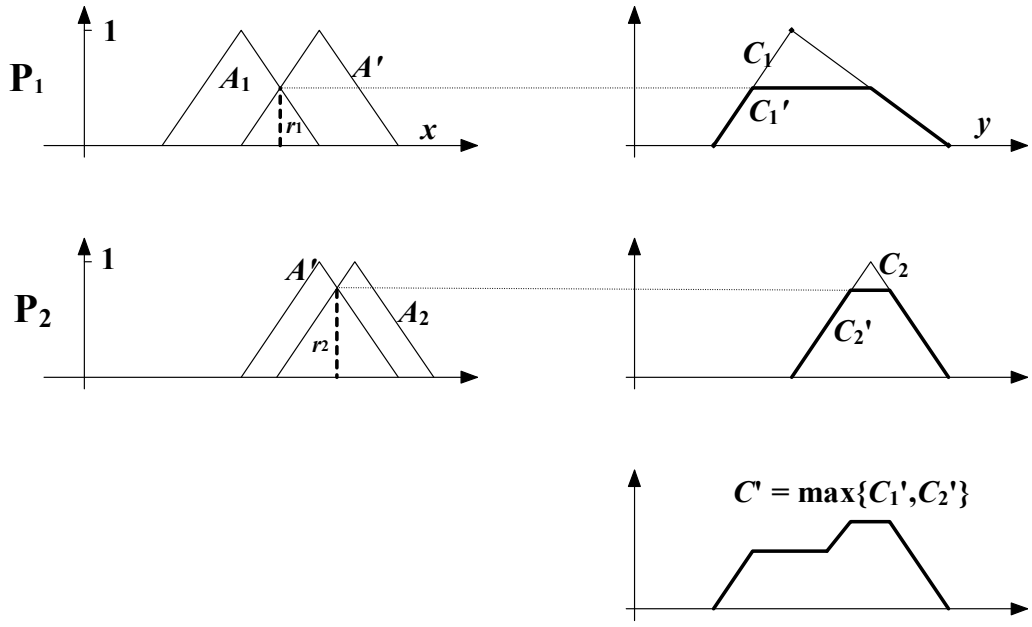
Poznamenajme, že možno postupovať aj inak (s rovnakým výsledkom): Nájdeme pre každé pravidlo P_i reláciu R_i. Potom relácia R zodpovedajúca celej báze je:

$$R = \max\{R_1,\ R_2, \ldots, R_k\}$$

a

$$C' = A' \circ R$$

Pre konečné univerzálne množiny je kompozícia prevádzaná max-min násobením matíc. Takéto odvodzovanie nazývame FATI (first aggregation, then inference). Odvodzovanie pomocou stupňov zapálenia je prehľadnejšie a nazývame ho FITA (fist inference, then aggregation). Možno ho efektívne zovšeobecňovať. Na Obr. 11 je znázornená báza Mamdani-Assiliánovho regulátora s 2 pravidlami, s fuzzy vstupom A' a graficky odvodeným fuzzy výstupom C'.



Obr. 11. Odvodenie výstupu štandardného Mamdaniho-Assiliánovho regulátora

Požiadavky na bázu pravidiel

Venujme sa teraz báze pravidiel $\mathbb{B} = \{P_1, P_2, \ldots P_k\}$. Je zrejmé, že kvalita fuzzy regulátora závisí najmä od nej. Napríklad, ak v báze pravidiel existujú dve pravidla so zhodným predpokladom (antecedent), ale rôznymi závermi (consequent), báza

pravidiel vytvára spor a je nekorektná. Predstavme si, že by v báze pravidiel boli pravidlá:

P_1: AK teplota je nízka POTOM je hustota veľká.

P_2: AK teplota je nízka POTOM je hustota malá.

Takýto fuzzy regulátor nemôže spoľahlivo riadiť žiaden proces. Zhrňme najčastejšie uvádzané požiadavky na bázu pravidiel [7], [9]:

1. Konzistencia: $(A_i = A_j) \Rightarrow (C_i = C_j)$
2. Úplnosť: $\bigcup_i \operatorname{supp} A_i = \boldsymbol{X}$
3. Interakcia $\Phi(A_i) = C_i$, pre každé $i \in \{1, 2, \ldots, k\}$

Úplnosť bázy pravidiel zaručuje, že pre každý vstup existuje aspoň jedno použiteľné pravidlo. Vážnejší problém je interakcia, niekedy nazývaná interpolácia. Je známe, Mamdaniho-Asiliánov regulátor [11] nemá vo všeobecnosti túto vlastnosť. Vieme len, že $\Phi(A_i) \geq C_i$, pre každé $i \in \{1, 2, \ldots, k\}$. V literatúre sú známe niektoré dosť ťažko overiteľné podmienky zaručujúce vlastnosť interakcie. Podmienky sa týkajú najmä bázy pravidiel. Spomenieme iba, že prienik (vzájomné prekrytie) antecedentov nemôže byť príliš veľké.

Obvykle počiatočná báza pravidiel je získavaná zo skúseností expertov, pozorovaním, porovnávaním s podobným problémom. Je upravovaná počas ladenia regulátora. Ladenie často využíva neurónové siete a genetické algoritmy.

Princíp obmedzenej informácie

Je zrejmé, že ideálny výstup regulátora je singleton (charakteristická funkcia jednoprvkovej množiny $\{x_0\}$). Takýto (jednoznačný) výstup nevyžaduje defuzzifikáciu, je návodom na regulačný zásah a teda prináša najväčšiu informáciu. Princíp obmedzenej informácie (Information Boundedness Principle – IBP) hovorí, že aktuálny výstup pravidla C' nemôže priniesť viac informácie ako prináša consequent C, t.j.:

$$\inf(C') \leq \inf(C),$$

kde inf je miera informácie, ktorú prináša daná fuzzy množina [15], [21]. Teda, ak consequent pravidle C je rozmazaný (fuzzy), potom aktuálny výstup nemôže byť singleton. Ak záver pravidla hovorí: POTOM teplota je vysoká, aktuálny výstup nemôže hovoriť, že teplota je presne 500° C. Mieru informácie možno definovať rôznym spôsobom. Napríklad pre fuzzy podmnožinu A konečnej univerzálnej množiny možno mieru informácie (nazývame ju špecificita - Sp) definovať ako rozdiel najväčšej hodnoty príslušnosti a aritmetického priemeru ostatných. Napríklad, ak $A = \{0.5, 0.6, 0.9, 0.2, 0, 0.1\}$, potom:

$$\mathrm{Sp}(A) = 0.9 - \frac{0.5 + 0.6 + 0.2 + 0 + 0.1}{5} = 0.62\,.$$

Pre singleton $B = \{0, 0, 0, 1, 0, 0\}$, zrejme $\mathrm{Sp}(B) = 1$. Ak fuzzy množina má konštantnú funkciu príslušnosti, potom jej špecificita je nulová. Napr., ak $B = (0.6, 0.6, 0.6, 0.6)$, potom $\mathrm{Sp}(B) = 0$. Fuzzy regulátor spĺňa IBP , ak $\mathrm{Sp}(C') \leq \mathrm{Sp}(C)$.

Zovšeobecnený fuzzy regulátor

Výpočet výstupnej fuzzy množiny C' popísaný vyššie možno rozdeliť na 3 kroky:

1. krok: Výpočítať stupňe zapálenia jednotlivých pravidiel r_1, r_2, r_3, ..., r_k , kde:

$$r_i = \sup_{x \in X} T(A_i'(x), A(x))$$

2. krok: Odvodenie individuálneho výstupu každého pravidla ($i = 1, 2, ..., k$):

$$C_i'(y) = T(r_i, C_i(y)), \ y \in Y$$

3. krok: Získanie celkového výstupu:

$$C'(y) = \max \{ C_1'(y), C_2'(y), \ldots, C_k'(y)\}$$

Pre štandardný Mamdaniho-Asiliánov regulátor T = min. Upozorňujeme najmä na skutočnosť: ak $r_i = 0$, potom $C_i' = T(0, C_i) \equiv 0$ a číslo 0 je neutrálny prvok operácie maxima na intervale [0;1]. Zovšeobecnením uvedených krokov možno definovať zovšeobecnený fuzzy regulátor (ZFR) tak, aby Mamdaniho-Asiliánov regulátor bol jeho špeciálnym prípadom. V snahe zovšeobecniť 2. krok zavedieme operátor navrhnutý Yagerom [21], [22] a nazývaný RET operátor (Relevancy Transformation operátor). K zovšeobecneniu 3. kroku použijeme vhodnú agregačnú funkciu.

Definícia 7: Nech $e \in [0;1]$ je pevne zvolený prvok. Binárny operátor:

$$\text{Ret}: [0;1]\times[0;1] \rightarrow [0;1]\ [0;1]$$

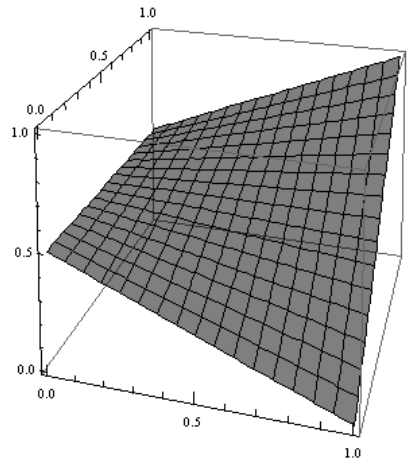
nazývame *RET* ***operátor*** vzhľadom na prvok e ak

r1) $\text{Ret}(1,a) = a$, $\text{Ret}(0, a) = e$ pre každé $a \in [0;1]$,

r2) Ret je neklesajúci v druhom argumente. ($\text{Ret}(r,a_1) \leq \text{Ret}(r,a_2)$, ak $a_1 < a_2$,

r3) Ret je neklesajúci v prvom argumente na $[0;1]\times [e,1]$ a nerastúci v prvom argumente na $[0;1]\times[0,e]$.

Poznamenajme, že r1) hovorí, že efektívny výstup pravidla P_i je rovný záveru pravidla C_i, ak pravidlo je plne zapálené ($r_i = 1$) a výstup pravidla nerozlišuje elementy výstupného univerza ***Y,*** ak pravidlo nie je zapálené ($r_i = 0$). Z r2) a r3) plynie, že $\text{Ret}(r,e) = e$ pre všetky $r \in [0;1]$. Ľahko ukázať, že pre $e = 0$ je Ret fuzzy konjunkciou a pre $e = 1$ je fuzzy implikáciou. Viac o filozofii RET operátorov nájdete v [19], [21], [22].



Obr. 12. PRET operátor pre $e = 0.5$.

Príklad 11: Nech $e \in [0;1]$ je pevne zvolený prvok. Definujme pre $r,a \in [0,1]$:

$$h(r,a) = ra + (1- r)e,$$

h je RET operátor vzhľadom na element e. Nazývame ho PRET (Product RET) operátor. Na Obr. 12 je takýto PRET operátor pre $e = 0.5$.

Prvok e z RET operátora by mal byt neutrálnym prvkom agregačného operátora (funkcie), ktorý zovšeobecňuje v 3. kroku použitý operátor maximum.

Definícia agregačnej funkcie

Funkciu $Agg : \bigcup_{i \in \mathrm{N}} [0;1]^i \to [0;1]$ nazývame agregačnou funkciou (operátorom), ak:

a1) $\mathrm{Agg}(x) = x$ pre všetky $x \in [0,1]$,

a2) $\mathrm{Agg}(0, 0, \ldots, 0) = 0$, $\mathrm{Agg}(1,1,\ldots,1) = 1$,

a3) $\mathrm{Agg}(x_1, x_2, \ldots, x_n) \le \mathrm{Agg}(y_1, y_2, \ldots, y_n)$, ak $x_1 \le y_1$, $x_2 \le y_2$, ... $x_n \le y_n$

Prvok e je neutrálny prvok Agg ak:

$$\mathrm{Agg}(a_1, a_2, ...a_i, a_{i+1} ..., a_k) = \mathrm{Agg}\{a_1, a_2, ..., a_i, e, a_{i+1}, ..., a_k\}$$

Všetky t-normy (t-konormy) sú agregačné operátory s neutrálnym prvkom 1 (0). Aritmetický vážený priemer sú agregačné operátory, ktoré nemajú neutrálny prvok. Príkladom agregačnej funkcie s netriviálnym neutrálnym prvkom $e = 1/2$ je:

$$\mathrm{Agg}(a_1, a_2, ..., a_k) = \left\langle \sum_i a_i - \frac{k}{2} \right\rangle + \frac{1}{2}, \text{ kde } \langle z \rangle = \max\left\{ -\frac{1}{2}, \min\left\{ z, \frac{1}{2} \right\} \right\}$$

Vidíme, že $\mathrm{Agg}(0.4, 0.6, 0.8) = \left\langle 1.8 - \frac{3}{2} \right\rangle + \frac{1}{2} = 0.3 + 0.5 = 0.8$,

$$\mathrm{Agg}(0.4, 0.5, 0.6, 0.8) = \langle 2.3 - 2 \rangle + \frac{1}{2} = 0.3 + 0.5 = 0.8 .$$

Definícia zovšeobecneného fuzzy regulátora

Zovšeobecnený fuzzy regulátor (ZFC) s bázou pravidiel $\mathbb{B} = \{P_1, P_2, \ldots P_k\}$ a s fuzzy vstupom $A' \in F(\boldsymbol{X})$ a fuzzy výstupom $C' \in F(\boldsymbol{Y})$ [16], [18] je štvorica:

$$\Theta = (\mathbb{B}, \text{Fir}, \text{Ret}, \text{Agg}),$$

kde:

- $\mathbb{B} = \{P_1, P_2, \ldots, P_k\}$ je báza pravidiel pozostávajúca z k AK..., POTOM... pravidiel s normálnymi predpokladmi a závermi tak, že
$$\bigcup_i \text{supp } A_i = \boldsymbol{X}$$
- Fir: $F(\boldsymbol{X}) \times F(\boldsymbol{Y}) \to [0,1]$ priraďuje každému vstupu $A' \in F(\boldsymbol{X})$ vektor relevancií $(r_1, r_2, ..., r_n)$ tak že $r_i = \text{Fir}(A', A_i)$, $i = 1,2, \ldots, k$. Označme $(r_1, r_2, ..., r_n) = \text{Fir}(A', \Theta)$. Predpokladáme, že pre každú normálnu fuzzy množinu $A \in F(\boldsymbol{X})$ platí: $\text{Fir}(A,A) = 1$,
$$\text{Fir}(A, \Theta) = (r_1, r_2, ..., r_k) \neq (0, 0, ..., 0)$$
a vektor $(r_1, r_2, ..., r_n)$ obsahuje najviac jedno číslo $r_i = 1$. (Pre každý normálny vstup je zapálené aspoň jedno pravidlo a najviac jedno pravidlo je plne zapálené).
- *Ret:* $[0,1]^2 \to [0,1]$ je RET operátor vzhľadom na daný špeciálny prvok $e \in [0,1]$ Individuálny výstup $C'_i = \text{Ret}(r_i, C_i)$ každého pravidla je daný po bodoch:
$$C'_i(y) = \text{Ret}(r_i, C_i(y)) \text{ pre každé } y \in \boldsymbol{Y},\ i = 1,2, \ldots, n.$$
- *Agg*: $[0,1]^k \to [0,1]$ je reštrikcia agregačnej funkcie s neutrálnym prvkom e. (Prvok e je súčasne špeciálnym prvkom RET operátora Ret a neutrálny prvok agregačnej funkcie Agg.)

Celkový výstup C' pre daný normálny vstup A' je daný prenosovou funkciou:

$$Y = \Phi_\Theta(X)$$

Tak, že pre každé $y \in \boldsymbol{Y}$

$$Y(y) = \underset{i}{\text{Agg}}\left(\text{Ret}\left(r_i, C_i(y)\right)\right).$$

Na záver spomenieme vlastnosť interakcie, ktorá hovorí, že $\Phi(A_i) = C_i$ pre každé $i \in \{1, 2, \ldots, k\}$. Vieme, že ani štandardný Mamdaniho-Assiliánov regulátor, nemusí mať túto vlastnosť. Uvedieme pomerne jednoduchú postačujúcu podmienku aby (ZFR) mal túto vlastnosť. Označme:

$$c = \max\{\text{Fir}(A_i, A_j),\ i,j \in \{1,2, \ldots, k\},\ i \neq j\}.$$

Veta 9: Uvažujme zovšeobecnený fuzzy regulátor $\Theta = (\mathbb{B}, \text{Fir}, \text{Ret}, \text{Agg})$ taký, že $c < 1$ a pre všetky $r,b \in [0,1]$, $0 \leq r \leq c$ platí:

$$\text{Ret}(r,b) = e,$$

kde Ret je RET operátor vzhľadom na element $e \in [0,1]$. Potom $\Phi(A_i) = C_i$ pre každé $i \in \{1, 2, \ldots, k\}$.

Takýto RET operátor je napríklad: h: $[0;1]^2 \to [0;1]$ daný vzťahom:

$$h(r,b) = \begin{cases} e & \text{ak} \quad r \leq c \\ b + (b-e)(r-1)/(1-c) & \text{inak} \end{cases}.$$

Ukázali sme, že zovšeobecnený fuzzy regulátor možno zostaviť pomocou RET operátora vzhľadom na vybraný prvok e (napr. $e = 0.5$) a agregačnej funkcie s neutrálnym prvkom e a odskúšať pomocou vhodnej bázy pravidiel [3]. Špeciálne, ak použijeme v 2. kroku fuzzy implikáciu ($e = 1$), vhodnou agregačnou funkciou je minimum, ak použijeme fuzzy konjunkciu ($e = 0$), vhodnou agregačnou funkciou je maximum.

6 Záver

Oboznámili sme sa so základnými pojmami teórie viachodnotovej logiky a fuzzy množín, s problémami pri ich definícii a používaní. Z mnohých možných aplikácii sme sa venovali ich použitiu vo fuzzy rozhodovacích procesoch, najmä málo známym fuzzy preferenčným štruktúram a fuzzy regulátorom. V závere sme navrhli zovšeobecnený fuzzy regulátor a skúmali sme jeho vlastnosti. Umožňuje použitie mnohých fuzzy spojok (operátorov), napríklad pri simulácii fuzzy regulátora v programovom balíku MATLAB. V závere uvádzame zoznam literatúry, v ktorej záujemca nájde ďalšie poznatky z danej problematiky.

Poďakovanie: Táto kapitola vznikla za podpory grantovej agentúry Vega vrámci grantových úloh 1/0983/12, 1/0569 /12.

Literatúra

[1] De Baets, B.,Fodor, J.: Twenty Years if Fuzzy Preference Modelling. *Belgian Journal of Operations Research, Statistics and Computer Science, Vol* **37** (1997), 61-82.

[2] De Baets, B., Van de Walle, B., Kerre, E.:Fuzzy Preference Structures and their Characterisation. *J. of Fuzzy Math*. **3** (1995), 373-381.

[3] Drahoš, M.: Závislosť efektívnosti fuzzy regulácie od použitej odvodzovacej metódy. *Diplomová práca FCHTP STU,* 2004.

[4] Fodor, J., De Baets, B.: Fuzzy Preference Modelling. *Fundamentals and Recent Advances, Studies in Fuzziness and Soft Computing,* Springer, 2008, 207-217.

[5] Frank, M. J.: On the Simultaneous Associativity of F(x,y) and x+y-F(x,y). *Aequationes Mathematicae* **19**(1979), 194-226.

[6] Hliněná, D., Kráľ, P.: Similarity of Fuzzy Preference Structures Based on Metrics. *New Dimension in Fuzzy Logic and Related Technologies. Eusflat* 2007, Vol. **2**, 421-436.

[7] Navara, M., Petrík, M.: How to use controller with conditionally firing values. *New Dimension in Fuzzy Logic and Related Technologies. Eusflat* 2007, Vol. **2**, 87-94.

[8] Klement, E. P., Mesiar, R., Pap, E.: *Triangular Norms*. Kluwer Academic Publisher, Dordrecht 2000.

[9] Klír, B. J., Bo, Y.: *Fuzzy sets and fuzzy logic. Theory and Application*. Prentice Hall, New York, 1995.

[10] Kolesárová, A., Kováčová, M.: *Fuzzy množiny a ich aplikácie*. STU 2004

[11] Mamdani, E, H., Assilian, S.: An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller. *International Journal of Man-Machine Studies* **7** (1975), 1-13.

[12] Navara, M., Olšák, P.: *Základy fuzzy množín*. Skriptum FE ČVUT, 2002.

[13] Nelsen, R. B.: *An Introduction to Copulas*, Springer, NY, 1999.

[14] Petrík, M., Sarkoci, P.: Convex combinations of nilpotent triangular norms. *Journal of Mathematical Analysis and Applications* **350** (2009), 271-275.

[15] Sarkoci, P., Šabo, M.: Information Boundedness Principle in Fuzzy inference Process.. *Kybernetika* **38** (2002), 328-333.

[16] Šabo, M.: On Some Properties of Fuzzy Controllers. *Journal of Applied Mathematics* **2** (2009), No. 1, 117-124.

[17] Šabo, M., Strežo, P.: On Reverse of Some Binary Operations. *Kybernetika* **41** (2005), 435-450.

[18] Šabo, M.: Relevancy Transformation Operators and Aggregation Functions in Fuzzy Inference Process. *Proc. of IPMU* 2008, Vol. **1**, 1392-1397.

[19] Šabo, M., Kolesárová A., Varga Š.: Ret Operators generated by triangular norms and copulas. *Int. J. of Uncertainty, Fuzziness and Knowledge Based Systems,* Vol. **9**(2001), 169-181.

[20] Šabo, M.: On many valued implications. *Tatra Mt. Math. Publ.* **14** (1998), 161-167.

[21] Yager, R. R.: Global requirement for implication operators in fuzzy modus ponens, *Fuzzy sets and Systems* **106**(1999), 3-10.

[22] Yager, R. R.: Uninorms in fuzzy modelling. *Fuzzy sets and Systems* **122**(2001, 167-175.

[23] Zadeh, L.: Fuzzy sets. *Inform. And Control* **8**(1965), 338-353.

Teória Kalmanových filtrov a neurónové siete

Peter TREBATICKÝ[1]

Abstrakt. Trénovanie neurónových sietí, najmä rekurentných, je veľmi obtiažny numerický problém, ktorý, ak je riešený klasickými gradientovými optimalizačnými technikami pre dlhšie vstupné reťazce, tak sa veľmi zle a pomaly približuje k uspokojivým výsledkom. Preto sa hľadajú efektívnejšie spôsoby učenia rekurentných sietí. Techniky založené na Kalmanovej filtrácii predstavujú lepšiu alternatívu ku gradientovým metódam.

1 Úvod

Trénovanie rekurentných neurónových sietí je veľmi obtiažny numerický problém, ktorý, ak je riešený klasickými gradientovými optimalizačnými technikami pre dlhšie vstupné reťazce, tak sa veľmi zle a pomaly približuje k uspokojivým výsledkom. Práve metódy založené na Kalmanovej filtrácii predstavujú lepšiu alternatívu ku gradientovým metódam.

Kalmanova filtrácia je užitočná metóda pre odhad aktuálneho a tiež aj budúceho stavu dynamického systému. V skutočnosti je Kalmanov filter optimálny pre lineárne dynamické systémy. Väčšina systémov je však nelineárna. To je dôvod, prečo sa hľadajú techniky pre rozšírenie Kalmanovej filtrácie, aby ich bolo možné použiť v týchto typoch systémov. V tejto kapitole popisujeme niekoľko takých modifikácií, pričom ich rovnice uvádzame v tvare upravenom pre účely trénovania neurónových sietí.

Čitateľa najprv oboznámime s pojmom dynamický systém, ktorý je kľúčový v teórii Kalmanovho filtrovania. Potom podrobne vysvetlíme a odvodíme pôvodný Kalmanov filter. Ukážeme, že na neurónovú sieť je možné nazerať ako na nelineárny dynamický systém, čím je možné na ne aplikovať túto teóriu. Nie však priamo, ale použitím niektorej modifikácie, ktoré uvádzame. Popíšeme rozšírený Kalmanov filter, Unscented Kalmanov filter, filter nprKF a napokon ich zdieľané verzie.

Nakoniec všetky metódy porovnáme v experimentoch aj so zástupcom klasických gradientových metód.

[1] Fakulta informatiky a informačných technológií, Slovenská technická univerzita v Bratislave, Ilkovičova 3, 842 16 Bratislava, E-mail: trebaticky@fiit.stuba.sk

2 Dynamický systém

Dynamický systém vo všeobecnosti pozostáva z množiny stavov a z predpisu, ktorý určuje nasledujúci stav v závislosti na minulých stavoch systému [1, 2]. Stav je vektor, ktorého zložky tvoria všetky relevantné premenné systému tak, aby stavové vektory poskytovali dostatok informácií na predikciu budúcich stavov. Dôležitý fakt je, že stavy systému nepozorujeme priamo. K dispozícii máme len výstup zo systému, ktorý je určený aktuálnym stavom, takže je možné odhadnúť aktuálny stav systému.

Dynamické systémy môžu byť vyjadrené buď v spojitom alebo diskrétnom čase. My sa budeme zaoberať iba dynamickými systémami pracujúcimi v diskrétnom čase. Dynamický systém je diskrétny, ak pracuje v diskrétnych, zvyčajne rovnako vzdialených, časových intervaloch. Neznamená to však automaticky, že jeho výstupy nadobúdajú diskrétne hodnoty. Dynamický systém môžeme vyjadriť nasledovnými dvoma rovnicami:

1. *Stavová rovnica*

$$\mathbf{x}_{k+1} = \mathbf{f}(k, \mathbf{x}_k) + \mathbf{q}_k$$

kde $\mathbf{f}(k, \mathbf{x}_k)$ je vo všeobecnosti nelineárna *prechodová maticová funkcia* prevádzajúca stav $\mathbf{x}_k$ z času k do času $k+1$. O procesnom šume $\mathbf{q}_k$ sa predpokladá, že je aditívny, biely a gausovský s nulovou strednou hodnotou a kovariančnou maticou $\mathbf{Q}_k$.

2. *Rovnica pozorovaní*

$$\mathbf{y}_k = \mathbf{h}(k, \mathbf{x}_k) + \mathbf{r}_k$$

kde $\mathbf{y}_k$ je pozorovanie v čase k a $\mathbf{h}(k, \mathbf{x}_k)$ je *maticová funkcia pozorovaní*. O šume pozorovaní $\mathbf{r}_k$ sa takisto predpokladá, že je aditívny, biely a gausovský s nulovou strednou hodnotou a s kovariančnou maticou $\mathbf{R}_k$. Šumy $\mathbf{r}_k$ a $\mathbf{q}_k$ sú navzájom nezávislé.

Tieto dve rovnice vyjadrujú, že (skrytý) stav systému sa vo všeobecnosti mení nelineárne funkciou meniacou sa v čase a navyše na zmenu stavu môže vplývať šum. Podobne pozorovanie (výstupný vektor) zo systému vo všeobecnosti nelineárne závisí od aktuálneho stavu systému funkciou meniacou sa v čase a tiež naňho môže vplývať šum. Ak sú obe funkcie $\mathbf{f}$ aj $\mathbf{h}$ lineárne, potom hovoríme, že aj dynamický systém je lineárny. Inak je nelineárny.

Dynamickými systémami sú napríklad slnečná sústava, počasie, burza, šírenie sa epidémie, chemické reakcie, atď.

3 Kalmanov filter

Kalmanov filter, čo je sada matematických rovníc, je považovaný za jeden z významných objavov dvadsiateho storočia – článok E. Kalmana [3] vyšiel v roku 1960. Jeho prvými aplikáciami bolo riadenie zložitých dynamických systémov, ako rôzne výrobné procesy, lietadlá, lode, vesmírne lode (napr. bol použitý v navigačnej jednotke Apolla).

Od začiatku bol a stále je hojne využívaný nielen v automatizácii, ale aj v grafických a ekonomických aplikáciách, atď. no až relatívne nedávno sa začal objavovať aj v aplikáciách na trénovanie neurónových sietí, čo súvisí s postupom vývoja počítačových systémov.

3.1 Princíp

Za predpokladu, že systém je lineárny, že poznáme jeho model, a že šum je biely a gausovský, je Kalmanov filter optimálny v zmysle, že minimalizuje chybu štvorcov [4, 5, 1]. Jeho činnosť pozostáva z dvoch krokov – krok predikcie a krok korekcie. V kroku predikcie sa vypočíta odhad aktuálneho stavu systému a v ďalšom kroku sa tento odhad upraví o namerané hodnoty zašumené bielym šumom.

Stavový vektor systému x_k (alebo len stav) je definovaný ako minimálna množina údajov, ktorá jednoznačne popisuje správanie dynamického systému, kde k je diskrétny čas. Jednoducho povedané, stav systému je vektor obsahujúci všetky relevantné premenné systému.

1. *Stavová rovnica*

$$\mathbf{x}_k = \mathbf{F}_{k,k-1}\mathbf{x}_{k-1} + \mathbf{q}_{k-1}$$

kde $\mathbf{F}_{k,k-1}$ je *prechodová matica* prevádzajúca stav $\mathbf{x}_k$ z času *k-1* do času *k*. O procesnom šume $\mathbf{q}_k$ sa predpokladá, že je aditívny, biely a gausovský s nulovou strednou hodnotou a kovariančnou maticou $\mathbf{Q}_k$.

2. *Rovnica pozorovaní*

$$\mathbf{y}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{r}_k$$

kde $\mathbf{y}_k$ je *pozorovanie* v čase k a $\mathbf{H}_k$ je *matica pozorovaní*. O šume pozorovaní $\mathbf{r}_k$ sa takisto predpokladá, že je aditívny, biely a gausovský s nulovou strednou hodnotou a s kovariančnou maticou $\mathbf{R}_k$. Šumy $\mathbf{r}_k$ a $\mathbf{q}_k$ sú navzájom nezávislé.

Problém Kalmanovho filtrovania spočíva v optimálnom vyriešení lineárnej stavovej rovnice a súčasne lineárnej rovnice pozorovaní pre neznámy stav. Treba použiť všetky pozorované údaje pozostávajúce z vektorov $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k$ na nájdenie odhadu minimálnej strednej kvadratickej chyby pre každé $k \geq 1$.

Kalmanov filter pracuje v dvoch neustále sa opakujúcich krokoch:

1. *Časová aktualizácia*, tiež sa nazýva krok predikcie. Sem spadajú rovnice

$$\hat{\mathbf{x}}_k^- = \mathbf{F}_{k,k-1}\hat{\mathbf{x}}_{k-1}$$

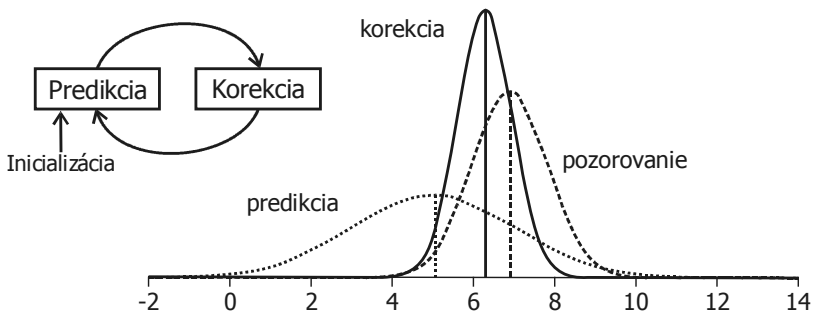
$$\mathbf{P}_k^- = \mathbf{F}_{k,k-1}\mathbf{P}_{k-1}\mathbf{F}_{k,k-1}^T + \mathbf{Q}_{k-1}$$

Vypočítame apriórne (t.j. pred pozorovaním) predikcie stavu a kovariancie chyby.

2. *Aktualizácia pozorovaním*, tiež sa nazýva krokom korekcie

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T \left[\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k \right]^{-1}$$
$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \left(\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^- \right)$$
$$\mathbf{P}_k = \left(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k \right) \mathbf{P}_k^-$$

T.j. korigujeme odhad získaný v predchádzajúcom kroku na základe pozorovania $\mathbf{y}_k$.



Obr. 1. Ilustrácia princípu Kalmanovej filtrácie. Pracuje v dvoch opakujúcich sa krokoch: predikcia a jej korekcia na základe pozorovania. Na jednorozmernom príklade podmienenej distribučnej funkcie môžeme vidieť, že je možné dosiahnuť lepší odhad aj s nie veľmi presnou predikciou a/alebo pozorovaním [4].

Na ilustrovanie princípu nám poslúži jednoduchý príklad [4]. Predpokladajme, že sme sa v noci stratili na mori a nemáme tušenie o našej polohe. Skúsime sa teda zorientovať pomocou hviezd (pre jednoduchosť predpokladajme jednorozmernú polohu). Kvôli nepresným prístrojom, ľudskej chybe a pod. je však výsledok nášho merania nepresný. Dokážeme však povedať, že presnosť je taká, že má určitú štandardnú odchýlku. Vieme preto zakresliť podmienenú pravdepodobnosť našej pozície ako je na obr. 1 – krivka označená *predikcia*, pretože svoju polohu odhadujeme.

Predstavme si teraz, že náš kamarát je skúsený námorník a v približne rovnakom čase urobí meranie nezávislé od nášho. Nakoľko má viacej skúseností, predpokladáme, že štandardná odchýlka jeho merania je menšia ako naša. Zakreslíme jeho odhad polohy ako krivku *pozorovanie* (obr. 1). Teraz upravíme našu *predikciu* na základe jeho *pozorovania* a ak to spravíme optimálne, získame krivku označenú ako *korekcia*.

Tento koncept predikcie – korekcie je základným princípom Kalmanovej filtrácie a je nemenný vo všetkých druhoch filtrov spomenutých v tejto kapitole.

3.2 Odvodenie Kalmanovho filtra

V tejto časti uvedieme odvodenie Kalmanovho filtra prevzaté z [2], opravené v niekoľkých detailoch a doplnené z pôvodného článku E. Kalmana [3] z roku 1960.

Zopakujme, že stavový vektor systému $\mathbf{x}_k$ (alebo len stav) je definovaný ako minimálna množina údajov, ktorá jednoznačne popisuje správanie dynamického systému, kde k je diskrétny čas. Jednoducho povedané, stav systému je vektor obsahujúci všetky relevantné premenné systému.

1. Stavová rovnica

$$\mathbf{x}_{k+1} = \mathbf{F}_{k+1,k}\mathbf{x}_k + \mathbf{q}_k \tag{1}$$

kde $\mathbf{F}_{k+1,k}$ je *prechodová matica* prevádzajúca stav $\mathbf{x}_k$ z času k do času $k+1$. O procesnom šume $\mathbf{q}_k$ sa predpokladá, že je aditívny, biely a gausovský s nulovou strednou hodnotou a kovariančnou maticou definovanou ako

$$E[\mathbf{q}_n \mathbf{q}_k^T] = \begin{cases} \mathbf{Q}_k & \text{pre } n = k \\ \mathbf{0} & \text{pre } n \neq k \end{cases} \tag{2}$$

2. Rovnica pozorovaní

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{r}_k \tag{3}$$

kde $\mathbf{y}_k$ je pozorovanie v čase k a $\mathbf{H}_k$ je matica pozorovaní. O šume pozorovaní $\mathbf{r}_k$ sa takisto predpokladá, že je aditívny, biely a gausovský s nulovou strednou hodnotou a s kovariančnou maticou definovanou ako

$$E[\mathbf{r}_n \mathbf{r}_k^T] = \begin{cases} \mathbf{R}_k & \text{pre } n = k \\ \mathbf{0} & \text{pre } n \neq k \end{cases} \tag{4}$$

Šum pozorovaní $\mathbf{r}_k$ je nezávislý od procesného šumu $\mathbf{q}_k$ a naopak.

Problém Kalmanovho filtrovania spočíva vo vyriešení stavovej rovnice a rovnice pozorovaní súčasne, optimálnym spôsobom, pre neznámy stav. Treba použiť všetky pozorované údaje pozostávajúce z vektorov $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k$, na nájdenie odhadu minimálnej strednej kvadratickej chyby pre každé $k \geq 1$. Problém sa nazýva *filtrovanie* ak $i = k$, *predikcia* ak $i > k$ alebo *vyhladzovanie* ak $1 \leq i < k$, kde i je čas, pre ktorý hľadáme odhad.

Pred samotným odvodením Kalmanovho filtra je vhodné spomenúť základy optimálneho odhadovania. Pre jednoduchosť uvediem vzťahy pre skalárne náhodné premenné – zovšeobecnenie pre vektorové náhodné premenné je priamočiare.

Predpokladajme, že máme pozorovanie

$$y_k = x_k + v_k$$

kde x_k je neznámy signál a v_k je aditívny šum. Nech $\hat{x}_k$ predstavuje aposteriórny odhad signálu x_k pri známych pozorovaniach $y_1, y_2, \ldots, y_k$. Vo všeobecnosti je odhad $\hat{x}_k$ odlišný od neznámeho signálu x_k. Na odvodenie tohto odhadu optimálnym spôsobom potrebujeme *chybovú funkciu* pre nesprávne odhady. Chybová funkcia by mala byť nezáporná a neklesajúca funkcia *chyby odhadu* $\tilde{x}_k = x_k - \hat{x}_k$. Tieto podmienky spĺňa napríklad *stredná kvadratická chyba*

$$J_k = E[(x_k - \hat{x}_k)^2] = E[\tilde{x}_k^2]$$

kde E je operátor očakávania (stredná hodnota). Závislosť chybovej funkcie J_k na čase *k* zdôrazňuje nestacionárnu podstatu rekurzívneho procesu odhadovania.

Na odvodenie optimálnej hodnoty pre odhad $\hat{x}_k$ uvediem dve vety prevzaté z teórie stochastických procesov.

Veta: *Podmienený odhad strednej hodnoty*

Ak sú oba stochastické procesy $\{x_k\}$ a $\{y_k\}$ gausovské, potom optimálny odhad $\hat{x}_k$, ktorý minimalizuje strednú kvadratickú chybu J_k je podmienený odhad strednej hodnoty

$$\tilde{x}_k = E[x_k \mid y_1, y_2, \ldots, y_k]$$

Veta: *Princíp ortogonality*

Nech stochastické procesy $\{x_k\}$ a $\{y_k\}$ majú nulovú strednú hodnotu, čiže

$$E[x_k] = E[y_k] = 0 \quad \text{pre všetky } k$$

Potom

1. ak sú oba stochastické procesy $\{x_k\}$ a $\{y_k\}$ gausovské; alebo
2. ak je optimálny odhad $\hat{x}_k$ obmedzený na lineárnu funkciu pozorovaní a chybová funkcia je stredná kvadratická chyba
3. potom optimálny odhad $\hat{x}_k$ pri známych pozorovaniach $y_1, y_2, \ldots, y_k$ je ortogonálna projekcia x_k na priestor tvorený týmito pozorovaniami.

Predpokladajme, že pozorovanie z lineárneho dynamického systému popísaného rovnicami (1) a (3) bolo urobené v čase k. Potrebujeme použiť informácie obsiahnuté v novom pozorovaní $\mathbf{y}_k$ na aktualizáciu odhadu neznámeho stavu $\mathbf{x}_k$. Nech $\hat{\mathbf{x}}_k^-$ predstavuje apriórny odhad stavu, ktorý je už v čase k k dispozícii. Ak chceme získať lineárny odhad, môžeme vyjadriť aposteriórny odhad $\hat{\mathbf{x}}_k$ ako lineárnu kombináciu apriórneho odhadu a nového pozorovania:

$$\hat{\mathbf{x}}_k = \mathbf{K}_k^{(1)} \hat{\mathbf{x}}_k^- + \mathbf{K}_k \mathbf{y}_k \tag{5}$$

kde matice $\mathbf{K}_k^{(1)}$ a $\mathbf{K}_k$ musíme určiť. Použijeme na to princíp ortogonality. *Vektor chyby stavu* je definovaný ako

$$\tilde{\mathbf{x}}_k = \mathbf{x}_k - \hat{\mathbf{x}}_k \tag{6}$$

Podľa princípu ortogonality môžeme písať

$$E[\tilde{\mathbf{x}}_k \mathbf{y}_i^T] = \mathbf{0} \quad \text{pre } i = 1, 2, \ldots, k-1 \tag{7}$$

Použitím rovníc (3), (5) a (6) v (7) dostávame

$$E[(\mathbf{x}_k - \mathbf{K}_k^{(1)} \hat{\mathbf{x}}_k^- - \mathbf{K}_k \mathbf{H}_k \mathbf{x}_k - \mathbf{K}_k \mathbf{q}_k) \mathbf{y}_i^T] = \mathbf{0} \quad \text{pre } i = 1, 2, \ldots, k-1 \tag{8}$$

Nakoľko sú procesný šum $\mathbf{q}_k$ a šum pozorovaní $\mathbf{r}_k$ od seba nezávislé, dostávame $E[\mathbf{q}_k \mathbf{y}_i^T] = \mathbf{0}$. Využitím tohto vzťahu a preusporiadaním môžeme rovnicu (8) prepísať nasledovne

$$E[(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k - \mathbf{K}_k^{(1)}) \mathbf{x}_k \mathbf{y}_i^T + \mathbf{K}_k^{(1)} (\mathbf{x}_k - \hat{\mathbf{x}}_k^-) \mathbf{y}_i^T] = \mathbf{0} \tag{9}$$

kde $\mathbf{I}$ je jednotková matica. Z princípu ortogonality dostávame $E[(\mathbf{x}_k - \hat{\mathbf{x}}_k^-) \mathbf{y}_i^T] = \mathbf{0}$, preto môžeme (9) zjednodušiť na

$$(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k - \mathbf{K}_k^{(1)}) E[\mathbf{x}_k \mathbf{y}_i^T] = \mathbf{0} \quad \text{pre } i = 1, 2, \ldots, k-1 \tag{10}$$

Pre ľubovoľné hodnoty stavu $\mathbf{x}_k$ a pozorovania $\mathbf{y}_k$, je rovnica (10) splnená za podmienky $\mathbf{I} - \mathbf{K}_k \mathbf{H}_k - \mathbf{K}_k^{(1)} = \mathbf{0}$, resp. ak

$$\mathbf{K}_k^{(1)} = \mathbf{I} - \mathbf{K}_k \mathbf{H}_k \tag{11}$$

Vložením rovnice (11) do (5) môžeme vyjadriť aposteriórny odhad stavu v čase k ako

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-) \tag{12}$$

kde maticu $\mathbf{K}_k$ voláme *matica Kalmanovho zosilnenia*.

Zostáva nám odvodiť explicitnú formulu pre $\mathbf{K}_k$. Nakoľko z princípu ortogonality dostávame

$$E[(\mathbf{x}_k - \hat{\mathbf{x}}_k) \mathbf{y}_k^T] = \mathbf{0} \tag{13}$$

a následne

$$E[(\mathbf{x}_k - \hat{\mathbf{x}}_k) \hat{\mathbf{y}}_k^T] = \mathbf{0} \tag{14}$$

kde $\hat{\mathbf{y}}_k^T$ je odhad $\mathbf{y}_k$ pri známych pozorovaniach $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k$. Definujeme *inovačný proces*

$$\tilde{\mathbf{y}}_k = \mathbf{y}_k - \hat{\mathbf{y}}_k \tag{15}$$

Inovačný proces reprezentuje mieru „novej" informácie obsiahnutej v $\mathbf{y}_k$. Môžeme ho vyjadriť aj ako

$$\tilde{\mathbf{y}}_k = \mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^- = \mathbf{H}_k \mathbf{x}_k + \mathbf{r}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^- = \mathbf{H}_k \tilde{\mathbf{x}}_k^- + \mathbf{r}_k \tag{16}$$

Odčítaním rovnice (14) od (13) a použitím definície z (15) dostaneme

$$E[(\mathbf{x}_k - \hat{\mathbf{x}}_k)\tilde{\mathbf{y}}_k^T] = \mathbf{0} \tag{17}$$

Použitím rovníc (3) a (12) môžeme vyjadriť vektor chyby stavu $\mathbf{x}_k - \hat{\mathbf{x}}_k$ ako

$$\mathbf{x}_k - \hat{\mathbf{x}}_k \;=\; \tilde{\mathbf{x}}_k^- - \mathbf{K}_k(\mathbf{H}_k \tilde{\mathbf{x}}_k^- + \mathbf{r}_k) \;=\; (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)\tilde{\mathbf{x}}_k^- - \mathbf{K}_k \mathbf{r}_k \tag{18}$$

Preto vložením rovníc (16) a (18) do (17) dostávame

$$E[\{(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)\tilde{\mathbf{x}}_k^- - \mathbf{K}_k \mathbf{r}_k\}(\mathbf{H}_k \tilde{\mathbf{x}}_k^- + \mathbf{r}_k)^T] = \mathbf{0} \tag{19}$$

Nakoľko je šum pozorovaní $\mathbf{r}_k$ nezávislý od stavu $\mathbf{x}_k$ a teda aj chyby $\tilde{\mathbf{x}}_k^-$, rovnica (19) sa zjednoduší:

$$(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) E[\tilde{\mathbf{x}}_k^- \tilde{\mathbf{x}}_k^{T-}] \mathbf{H}_k{}^T - \mathbf{K}_k E[\mathbf{r}_k \mathbf{r}_k{}^T] = \mathbf{0} \tag{20}$$

Apriórnu kovariančnú maticu definujeme ako

$$\mathbf{P}_k^- = E[(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)^T] = E[\tilde{\mathbf{x}}_k^- \, \tilde{\mathbf{x}}_k^{T-}] \tag{21}$$

Použitím definícií kovariancií z rovníc (4) a (21) môžeme rovnicu (20) prepísať na $(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)\mathbf{P}_k^- \mathbf{H}_k{}^T - \mathbf{K}_k \mathbf{R}_k = \mathbf{0}$. Vyjadríme z nej $\mathbf{K}_k$ a dostávame želanú formulu

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k{}^T [\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k{}^T + \mathbf{R}_k]^{-1} \tag{22}$$

kde symbol $[\cdot]^{-1}$ predstavuje inverznú maticu. Rovnica (22) je hľadaná formula pre výpočet matice Kalmanovho zosilnenia $\mathbf{K}_k$, ktorá je vyjadrená prostredníctvom apriórnej kovariančnej matice $\mathbf{P}_k^-$.

Pre skompletizovanie procedúry rekurzívneho odhadu uvažujeme *šírenie kovariancie chyby*, ktoré popisuje účinok času na kovariančné matice chýb odhadu. Toto šírenie zahŕňa dva výpočtové kroky

1. Apriórna kovariančná matica $\mathbf{P}_k^-$ v čase k je definovaná v rovnici (21). Pre dané $\mathbf{P}_k^-$ je potrebné vypočítať aposteriórnu kovariančnú maticu $\mathbf{P}_k$, ktorá je pre čas k definovaná ako
$$\mathbf{P}_k = E[\tilde{\mathbf{x}}_k \tilde{\mathbf{x}}_k{}^T] = E[(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T] \tag{23}$$
2. Pre danú „starú“ aposteriórnu kovariančnú maticu $\mathbf{P}_{k-1}$ je potrebné vypočítať „aktualizovanú“ apriórnu kovariančnú maticu $\mathbf{P}_k^-$.

Pre prvý krok použijeme rovnicu (18) do (23) a využitím skutočnosti, že procesný šum $\mathbf{r}_k$ je nezávislý od apriórnej chyby odhadu $\tilde{\mathbf{x}}_k^-$, dostávame

$$\begin{aligned}\mathbf{P}_k &= (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)E[\tilde{\mathbf{x}}_k^- \tilde{\mathbf{x}}_k^{T-}](\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)^T + \mathbf{K}_k E[\mathbf{r}_k\mathbf{r}_k{}^T]\mathbf{K}_k^T \\ &= (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^-(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)^T + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k{}^T\end{aligned} \tag{24}$$

Roznásobením členov v rovnici (24) a použitím rovnice (22), môžeme preformulovať závislosť aposteriórnej kovariančnej matice $\mathbf{P}_k$ od apriórnej kovariančnej matice $\mathbf{P}_k^-$ v zjednodušenej forme

$$\begin{aligned}\mathbf{P}_k &= (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^- - (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^-\mathbf{H}_k^T\mathbf{K}_k^T + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^T \\ &= (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^- - \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^T + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^T \\ &= (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^-\end{aligned} \tag{25}$$

Pre druhý krok šírenia kovariancie chyby si je potrebné uvedomiť, že apriórny odhad stavu je definovaný pomocou „starého" aposteriórneho odhadu nasledovne

$$\hat{\mathbf{x}}_k^- = \mathbf{F}_{k,k-1}\hat{\mathbf{x}}_{k-1} \tag{26}$$

Môžeme preto použiť rovnice (1) a (26) na vyjadrenie apriórneho odhadu chyby ešte v ďalšej forme:

$$\begin{aligned}\tilde{\mathbf{x}}_k^- &= \mathbf{x}_k - \hat{\mathbf{x}}_k^- \\ &= (\mathbf{F}_{k,k-1}\mathbf{x}_{k-1} + \mathbf{Q}_{k-1}) - (\mathbf{F}_{k,k-1}\hat{\mathbf{x}}_{k-1}) \\ &= \mathbf{F}_{k,k-1}(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}) + \mathbf{Q}_{k-1} \\ &= \mathbf{F}_{k,k-1}\tilde{\mathbf{x}}_{k-1} + \mathbf{Q}_{k-1}\end{aligned} \tag{27}$$

Následne použitím rovnice (27) v (21) a skutočnosti, že procesný šum $\mathbf{q}_k$ je nezávislý od $\tilde{\mathbf{x}}_{k-1}$, dostaneme

$$\mathbf{P}_k^- = \mathbf{F}_{k,k-1} E[\tilde{\mathbf{x}}_{k-1}\tilde{\mathbf{x}}_{k-1}^T]\mathbf{F}_{k,k-1}^T + E[\mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^T] = \mathbf{F}_{k,k-1}\mathbf{P}_{k-1}\mathbf{F}_{k,k-1}^T + \mathbf{Q}_{k-1} \tag{28}$$

čo vyjadruje závislosť apriórnej kovariančnej matice $\mathbf{P}_k^-$ od „starej“ aposteriórnej kovariančnej matice $\mathbf{P}_{k-1}$.

3.3 Rovnice

Rovnice z predchádzajúcej časti môžeme zhrnúť do dvoch skupín a pripomenúť, že Kalmanov filter pracuje v dvoch neustále sa opakujúcich krokoch, ktoré inicializujeme odhadom stavového vektora $\hat{\mathbf{x}}_0$ spolu s kovariančnou maticou $\mathbf{P}_0$ vyjadrujúcou mieru istoty v odhade stavu.

1. Časová aktualizácia, tiež sa nazýva krok predikcie. Sem spadajú rovnice

$$\hat{\mathbf{x}}_k^- = \mathbf{F}_{k,k-1}\hat{\mathbf{x}}_{k-1} \tag{29}$$

$$\mathbf{P}_k^- = \mathbf{F}_{k,k-1}\mathbf{P}_{k-1}\mathbf{F}_{k,k-1}^T + \mathbf{Q}_{k-1} \tag{30}$$

T.j. pred tým, ako získame pozorovanie v aktuálnom kroku *k*, odhadujeme stavový vektor a prislúchajúcu kovariančnú maticu.

2. Aktualizácia pozorovaním, tiež sa nazýva krokom korekcie

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}_k^T[\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k]^{-1} \tag{31}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{y}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-) \tag{32}$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^- \tag{33}$$

T.j. korigujeme odhad získaný v predchádzajúcom kroku na základe pozorovania $\mathbf{y}_k$.

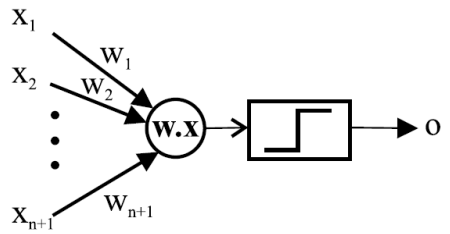
4 Neurónová sieť ako dynamický systém

V tejto časti stručne vysvetlíme pojem neurónová sieť, uvedieme jej základné typy a od toho sa odvíjajúce možnosti učenia danej siete. Tieto základy sú spracované prevažne z [6], kde čitateľ nájde podrobnosti. Na konci tejto časti ukážeme, že na neurónovú sieť je možné pozerať ako na dynamický systém, čo nám v ďalšej časti umožní aplikovať naň teóriu Kalmanovej filtrácie.

4.1 Model biologického neurónu

Tak ako sa klasická umelá inteligencia snaží napodobňovať „softvér“, tak sa umelé siete snažia napodobňovať „hardvér“. Tým je neurón, základná stavebná jednotka ľudského mozgu. Odhaduje sa, že sa v ľudskom mozgu nachádza rádovo 10^{11} neurónov. V neurónoch prebiehajú zložité biochemické deje, ktoré zabezpečujú to, že neuróny môžu spracovávať signály z iných neurónov a vysielať k nim svoje vlastné signály. Signály môžeme reprezentovať ako reálne čísla vyjadrujúce intenzitu prijímaných a vysielaných signálov. Miesto prenosu signálu z jedného neurónu na druhý sa nazýva synapsia. V synapsii sa môže signál zosilniť alebo zoslabiť rôznou veľkosťou, čo určuje váha synapsie. Jeden neurón môže mať rádovo 10^3 až 10^5 synapsií. Keď suma kladných a záporných signálov od ostatných neurónov váhovaná váhami príslušných synapsií prekročí istú hodnotu (prah excitácie), neurón vygeneruje výstupný impulz. V priebehu celého ľudského života sa v dôsledku individuálnej skúsenosti (učenia sa) menia váhy jednotlivých synapsií.

Na základe týchto poznatkov bol vytvorený *perceptrón* ako model neurónu, ktorý prijíma vstupné signály $\mathbf{x} = \{x_1, x_2, \ldots, x_{n+1}\}$ cez váhy tvoriace váhový vektor $\mathbf{w} = \{w_1, w_2, \ldots, w_{n+1}\}$ (obr. 2).



Obr. 2. Model neurónu – perceptrón [6]

Vstupom sú vo všeobecnosti reálne signály. Hodnota aktivačnej funkcie v bode skalárneho súčinu vektorov $\mathbf{w}$ a $\mathbf{x}$ je výsledný výstup perceptrónu **o**. (n + 1)-vý vstup má vždy hodnotu -1 a (n + 1)-vá váha sa nazýva prah excitácie (opäť analógia so skutočným neurónom).

Všeobecné pravidlo učenia pre umelé neurónové siete znie, že váhový vektor $\mathbf{w}$ sa zväčšuje priamo úmerne so súčinom vstupného vektora $\mathbf{x}$ a učiaceho signálu s. Učiaci signál je funkciou $\mathbf{w}$, $\mathbf{x}$ a niekedy aj spätnej väzby od učiteľa d – vtedy ide o učenie s učiteľom, inak o učenie bez učiteľa. Pri učení v umelých neurónových sieťach sa v diskrétnych časových krokoch mení j-ta váha nasledovne:

$$w_j(t+1) = w_j(t) + \Delta w_j(t) = w_j(t) + \alpha s(t) x_j(t)$$

Konštanta α sa nazýva rýchlosť učenia. Učiacim signálom pre binárny perceptrón je aritmetický rozdiel medzi požadovanou a skutočnou odpoveďou perceptrónu, t.j.

$s = d - o = \delta$. Pravidlo učenia binárneho perceptrónu sa nazýva *pravidlo delta*. Pre *j*-tu váhu platí

$$\Delta w_j = \alpha(d - o)x_j$$

Perceptrón dokáže klasifikovať len lineárne separovateľné vzory.

4.2 Dopredná sieť

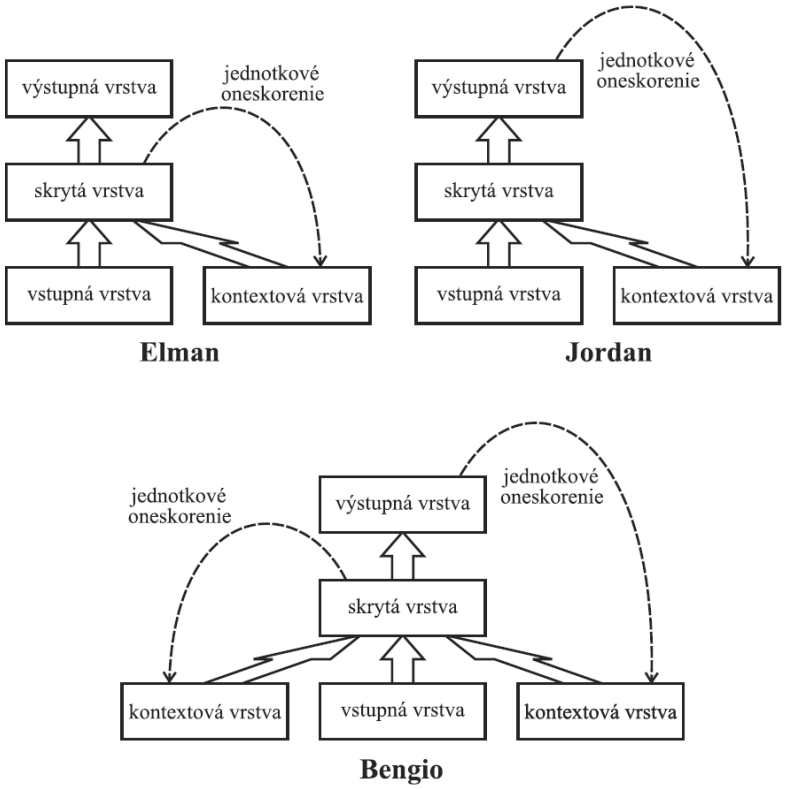
Väčšina reálnych problémov má nelineárny charakter, čo je pre perceptrón neriešiteľné. Preto boli zavedené viacvrstvové dopredné umelé neurónové siete, ktoré sa trénujú metódou spätného šírenia chýb (angl. error backpropagation). Dopredné neurónové siete sa vyznačujú tým, že v nich existujú iba dopredné spojenia medzi neurónmi. T.j. každý neurón na jednej vrstve vysiela signály na každý neurón nasledujúcej vrstvy. Spojenia do predchádzajúcej vrstvy neexistujú.

Chceme, aby sa po naučení skutočný výstup siete rovnal požadovanému výstupu, resp. aby sa mu priblížil čo najviac, a to pre všetky prezentované vzory **p**. Preto definujeme chybovú funkciu E_p, ktorá je sumou štvorcov chýb na všetkých výstupných neurónoch. Učenie má teda za cieľ modifikovať váhy siete tak, aby bola táto funkcia minimálna. Na hľadanie minima E_p sa aplikuje metóda najprudšieho spádu (gradientová metóda). Takto trénovaná sieť predstavuje univerzálny aproximátor.

4.3 Rekurentná sieť

Ak o výstupe siete nerozhoduje len aktuálny vstup do siete, ale aj doterajšia história predkladaných vzorov, je vo všeobecnosti potrebné hľadať inú štruktúru siete, ktorá by si bola schopná vytvoriť stavovú reprezentáciu *časového* kontextu v dátach. Ukázalo sa, že rekurentné neurónové siete sú tohto schopné. Za rekurentnú sieť možno považovať akúkoľvek neurónovú sieť, v ktorej si istá podmnožina neurónov, ktoré nazývame *rekurentné*, uchováva informáciu o svojich aktivitách v predošlých časoch. Hodnoty, ktoré boli na výstupe rekurentných neurónov v predošlom časovom kroku, sa prekopírujú na tzv. kontextové neuróny a pripoja sa k vstupnému vektoru v aktuálnom čase. Neurónová sieť sa takto rozširuje o vnútornú pamäť.

Podobne ako v dopredných neurónových sieťach, ani tu nie sú prepojenia v rámci jednej vrstvy (resp. prepojenia môžu byť, ale s nenulovým oneskorením). Na obr. 3 sú zobrazené najznámejšie architektúry rekurentných sietí, z ktorých najpoužívanejšia je Elmanova. Veľké šípky reprezentujú spojenia z každého neurónu spodnej vrstvy do každého neurónu hornej vrstvy. Tieto spojenia majú trénovateľné váhy. Jednoduché šípky predstavujú rekurentné spojenia medzi zodpovedajúcimi neurónmi východiskovej a cieľovej vrstvy. Rekurentné spojenia majú fixnú váhu 1 a existujú len medzi *i*-tym neurónom rekurentnej a *i*-tym neurónom kontextovej vrstvy.



Obr. 3. Rôzne architektúry rekurentných neurónových sietí – Elman, Jordan, Bengio [6]

Rekurentné neurónové siete sa trénujú na predikciu nasledujúceho požadovaného vektora napríklad pomocou metódy *rekurentné učenie v reálnom čase* (angl. real-time recurrent learning – RTRL). Jeho podstata je opäť spätné šírenie chýb, v ktorom sa odráža nielen priestorová, ale aj časová štruktúra dát.

Ďalším algoritmom trénovania rekurentných neurónových sietí je *spätné šírenie chýb v čase* (angl. backpropagation through time – BPTT). Slúži na trénovanie na klasifikáciu postupností. Požadovaný výstup sa sieti poskytne až na konci každej postupnosti. Pri BPTT sa neurónová sieť rozvíja v čase, čiže má toľko skrytých vrstiev, koľko je vstupov v jednej postupnosti. Potom sa trénuje rovnako ako dopredná neurónová sieť s n skrytými vrstvami, kde n je dĺžka postupnosti.

Nevýhodou týchto gradientových metód je napríklad to, že majú tendenciu uviaznuť v lokálnom minime. Táto kapitola sa zaoberá práve alternatívnym spôsobom trénovania, ktorý sa stáva čoraz populárnejším – metódami založenými na Kalmanovej filtrácii.

4.4 Neurónová sieť a dynamické systémy

Na neurónovú sieť sa môžeme pozerať ako na nelineárny dynamický systém, kde skrytým stavom je v tomto prípade vektor váh a naším cieľom je odhadnúť tento vektor

tak, aby sme dosahovali výstupy blízke požadovaným výstupom. Výsledný nelineárny systém môžeme popísať rovnicami [2],[1]:

$$\mathbf{x}_{k-1} = \mathbf{x}_k + \mathbf{r}_k \tag{34}$$

$$\mathbf{y}_k = h(\mathbf{x}_k, \mathbf{u}_k, \mathbf{v}_{k-1}) + \mathbf{q}_k \tag{35}$$

Stavová rovnica popisuje stav neurónovej siete ako stacionárny proces narušený procesným šumom $\mathbf{r}_k$, pričom stav siete $\mathbf{x}$ je daný vektorom jej váh. Rovnica pozorovania vyjadruje želaný výstup siete ako nelineárnu funkciu vstupného vektora $\mathbf{u}_k$, vektora váh $\mathbf{x}_k$ a pre rekurentné siete ešte aktivácii rekurentných neurónov z predchádzajúceho kroku $\mathbf{v}_{k-1}$. Táto rovnica je rozšírená o náhodný šum pozorovania $\mathbf{q}_k$. Kovariancia šumu $\mathbf{r}_k$ je $\mathbf{R}_k = E[\mathbf{r}_k \mathbf{r}_k^T]$ a kovariancia šumu $\mathbf{q}_k$ je $\mathbf{Q}_k = E[\mathbf{q}_k \mathbf{q}_k^T]$.

5 Rozšírený Kalmanov filter

Pôvodný Kalmanov filter bol vytvorený pre lineárne dynamické systémy, t.j. aj aktualizácia skrytého stavu systému, aj jeho transformácia na pozorovanie prebiehajú lineárne. Mnoho systémov je však nelineárnych. Preto bolo potrebné Kalmanov filter, ktorý je síce optimálny, ale len pre lineárne dynamické systémy, upraviť aj pre tento typ systémov. Nás konkrétne zaujímajú neurónové siete, ktoré, ako bolo uvedené, majú v rovnici pozorovania nelineárny výraz, čo vyplýva z toho, že šírenie vstupného signálu sieťou je nelineárne (aktivačné funkcie neurónov sú zväčša sigmoidálne).

5.1 Princíp

Ak je model *nelineárny*, čo je aj prípad neurónových sietí, musíme Kalmanov filter rozšíriť pomocou linearizačnej procedúry. Výsledný filter sa potom nazýva *Rozšírený Kalmanov filter* (angl. Extended Kalman Filter – EKF). Takéto rozšírenie je možné vďaka tomu, že Kalmanov filter je popísaný diferenčnými rovnicami v prípade systémov pracujúcich v diskrétnom čase.

Uvažujeme nelineárny dynamický systém popísaný modelom

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k) + \mathbf{q}_k \tag{36}$$

$$\mathbf{y}_k = h(\mathbf{x}_k) + \mathbf{r}_k \tag{37}$$

kde, ako aj predtým, $\mathbf{q}_k$ a $\mathbf{r}_k$ sú nezávislé gausovské biele šumy s nulovou strednou hodnotou s kovariančnými maticami $\mathbf{Q}_k$ a $\mathbf{R}_k$. Funkcionál $f(\mathbf{x}_k)$ predstavuje nelineárnu prechodovú maticovú funkciu, ktorá sa môže meniť s časom. Podobne funkcionál $h(\mathbf{x}_k)$ predstavuje *nelineárnu maticu pozorovaní*, ktorá sa tiež môže meniť v čase.

Základná myšlienka rozšíreného Kalmanovho filtra spočíva v linearizovaní modelu definovanom rovnicami (36) a (37) v každom časovom kroku okolo

najnovšieho odhadu stavu – buď $\hat{\mathbf{x}}_k$ alebo $\hat{\mathbf{x}}_k^-$ v závislosti od toho, o ktorý funkcionál ide.

Aproximácia prebieha v dvoch krokoch. Najprv zostrojíme matice (Jacobiány)

$$\mathbf{F}_{k+1,k} = \left.\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_k} \tag{38}$$

$$\mathbf{H}_k = \left.\frac{\partial h(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_k^-} \tag{39}$$

Čiže ij-ty prvok matice $\mathbf{F}_{k+1,k}$ je rovný parciálnej derivácii *i*-tej zložky $f(\mathbf{x})$ podľa *j*-tej zložky vektora $\mathbf{x}$. Podobne ij-ty prvok matice $\mathbf{H}_k$ je rovný parciálnej derivácii *i*-tej zložky $h(\mathbf{x})$ podľa *j*-tej zložky vektora $\mathbf{x}$. V prvom prípade sú derivácie vyhodnotené v bode $\hat{\mathbf{x}}_k$, v druhom prípade v bode $\hat{\mathbf{x}}_k^-$. Všetky prvky matíc $\mathbf{F}_{k+1,k}$ a $\mathbf{H}_k$ je možné vypočítať, lebo $\hat{\mathbf{x}}_k$ a $\hat{\mathbf{x}}_k^-$ sú v čase *k* známe.

Druhý krok aproximácie vezme vypočítané matice $\mathbf{F}_{k+1,k}$ a $\mathbf{H}_k$ a použije ich pri výpočte odhadu prvého stupňa Taylorovej aproximácie nelineárnych funkcií $f(\mathbf{x})$ a $h(\mathbf{x})$ nasledovne

$$f(\mathbf{x}_k) \approx f(\hat{\mathbf{x}}_k) + \mathbf{F}_{k+1,k}(\mathbf{x}_k - \hat{\mathbf{x}}_k) \tag{40}$$

$$h(\mathbf{x}_k) \approx h(\hat{\mathbf{x}}_k^-) + \mathbf{H}_k(\mathbf{x}_k - \hat{\mathbf{x}}_k^-) \tag{41}$$

Keď už máme k dispozícii tieto približné výrazy, môžeme pokračovať v aproximácii nelineárnych rovníc (36) a (37):

$$\mathbf{x}_{k+1} \approx \mathbf{F}_{k+1,k}\mathbf{x}_k + \mathbf{q}_k + \mathbf{d}_k \tag{42}$$

$$\bar{\mathbf{y}}_k \approx \mathbf{H}_k\mathbf{x}_k + \mathbf{r}_k \tag{43}$$

kde

$$\bar{\mathbf{y}}_k = \mathbf{y}_k - [h(\hat{\mathbf{x}}_k^-) - \mathbf{H}_k\hat{\mathbf{x}}_k^-]$$

$$\mathbf{d}_k = f(\hat{\mathbf{x}}_k) - \mathbf{F}_{k+1,k}\hat{\mathbf{x}}_k$$

Všetky premenné potrebné na výpočet $\bar{\mathbf{y}}_k$ sú v čase *k* známe, a preto $\bar{\mathbf{y}}_k$ môže byť považovaný za vektor pozorovania v čase *k*. Podobne aj premenné potrebné na výpočet $\mathbf{d}_k$ sú v čase *k* tiež známe.

5.2 Rovnice EKF

Keď už máme linearizovaný model v rovniciach (42) a (43), môžeme naňho aplikovať teóriu Kalmanovho filtra, čím získame rozšírený Kalmanov filter:

1. Časová aktualizácia, predikcia.

$$\hat{\mathbf{x}}_k^- = f(\hat{\mathbf{x}}_{k-1}) \tag{44}$$

$$\mathbf{P}_k^- = \mathbf{F}_{k,k-1}\mathbf{P}_{k-1}\mathbf{F}_{k,k-1}^T + \mathbf{Q}_{k-1} \tag{45}$$

2. Aktualizácia pozorovaním, korekcia.

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k{}^T [\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k{}^T + \mathbf{R}_k]^{-1} \tag{46}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k[\mathbf{y}_k - h(\hat{\mathbf{x}}_k^-)] \tag{47}$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)\mathbf{P}_k^- \tag{48}$$

Trénovanie neurónovej siete môžeme vyjadriť pomocou teórie Kalmanovho filtra ako problém hľadania odhadu stavu $\mathbf{x}$, ktorý minimalizuje strednú kvadratickú chybu, použijúc všetky doterajšie pozorovania (rovnice (34) a (35)). Nakoľko je stavová rovnica lineárna (dokonca $f(\mathbf{x}_k) = \mathbf{x}_k$), môžeme riešenie tohto problému vyjadriť úpravou vzťahov (44) − (48) nasledovne

$$\mathbf{K}_k = \mathbf{P}_k \mathbf{H}_k^T [\mathbf{H}_k \mathbf{P}_k \mathbf{H}_k^T + \mathbf{R}_k]^{-1} \tag{49}$$

$$\hat{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_k + \mathbf{K}_k[\mathbf{y}_k - h(\hat{\mathbf{x}}_k, \mathbf{u}_k, \mathbf{v}_{k-1})] \tag{50}$$

$$\mathbf{P}_{k+1} = \mathbf{P}_k - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_k + \mathbf{Q}_k \tag{51}$$

Pre prehľadnosť uvádzam súhrnne vysvetlenie značiek (v hranatých zátvorkách uvádzam rozmer matice / vektora)

n_x	Počet všetkých váh (stavových premenných)
n_o	Počet výstupných neurónov (výstupov)
$\mathbf{x}_{[n_x \times 1]}$	Stav systému, t.j. vektor všetkých váh
$h(\cdot)_{[n_o \times 1]}$	Funkcia vracajúca vektor skutočných výstupov
$\mathbf{y}_{[n_o \times 1]}$	Vektor požadovaných výstupov
$\mathbf{K}_{[n_x \times n_o]}$	Matica Kalmanovho zosilnenia
$\mathbf{P}_{[n_x \times n_x]}$	Kovariančná matica chyby stavu
$\mathbf{Q}_{[n_x \times n_x]}$	Kovariančná matica procesného šumu
$\mathbf{R}_{[n_o \times n_o]}$	Kovariančná matica šumu pozorovania
$\mathbf{H}_{[n_o \times n_x]}$	Matica pozorovania (Jacobián)

Každý prvok matice $\mathbf{H}$ vypočítame ako parciálnu deriváciu *i*-teho výstupu k *j*-tej váhe, pričom *i* nadobúda hodnoty od 1 po n_o a *j* od 1 po n_x. Jej výpočet sa realizuje buď algoritmom spätného šírenia chýb (angl. Back Propagation) v prípade doprednej neurónovej siete, alebo v prípade rekurentnej siete algoritmami Back Propagation

Through Time, Truncated Back Propagation Through Time, prípadne Real Time Recurrent Learning.

Na začiatku rekurzie (49) − (51) je potrebné nastaviť počiatočné hodnoty $\hat{\mathbf{x}}_0$, $\mathbf{P}_0$, $\mathbf{Q}_0$ a $\mathbf{R}_0$. Počiatočné hodnoty váh volíme náhodne napr. v rozmedzí $[-0.5, 0.5]$. Podľa odporúčaní v [2],[7] volíme $\mathbf{P}_0$ rádovo $100\mathbf{I} - 1000\mathbf{I}$, kde $\mathbf{I}$ je jednotková matica. $\mathbf{Q}_0$ môže nadobúdať hodnoty od $10^{-6}\mathbf{I}$ do $0.1\mathbf{I}$. Jej nenulová hodnota pomáha vyhnúť sa divergencii. Pre parameter $\mathbf{R}_0$ volíme hodnoty rádovo $10\mathbf{I} - 100\mathbf{I}$.

6 „Unscented“ Kalmanov filter

Neoddiskutovateľne najčastejšie používanou metódou na odhad stavu dynamického systému je rozšírený Kalmanov filter [8]. V EKF metódach je kľúčovým elementom vývoj kovariančnej matice $\mathbf{P}$, ktorá je v každom kroku aproximovaná použitím Taylorovho rozvoja (obyčajne nelineárnej) funkcie, ktorá dáva do súvisu výstupy siete s váhami. Typické je uchovávanie len výrazov prvého stupňa tohto rozvoja (derivácia výstupov siete podľa váh). EKF preto poskytuje len aproximáciu optimálneho nelineárneho odhadu. Linearizácia pomocou Taylorovho rozvoja však vo veľa prípadoch poskytuje nedostatočne presnú reprezentáciu a bežne sa vyskytujú výrazné chyby, ba až problémy s konvergenciou práve kvôli veľmi hrubej aproximácii [8].

S úspechom bolo použité aj nelineárne zovšeobecnenie Kalmanovho filtra [7] v rôznych aplikáciách vrátane trénovania rekurentných neurónových sietí pod názvom Unscented Kalman Filter (UKF) [9], čo by sa dalo preložiť ako Kalmanov filter bez zápachu[2]. UKF ponúka presnejší vývoj kovariančnej matice založený na vyhodnotení výstupu siete pre vhodne zvolené modifikácie aktuálneho vektora váh, ktoré sa nazývajú *sigma vektory*.

Koncepčne je základný princíp UKF podobný s EKF. Implementácia je však značne rozdielna. Oproti aproximácii pomocou Taylorovho rozvoja nie sú potrebné žiadne derivácie (výpočet Jacobiána, príp. Hessiána), ale len vyhodnotenia funkcií. Toto umožňuje jednoduchšiu implementáciu filtra a takisto odhadnúť stav aj v tom prípade, ak existujú singulárne body, v ktorých nie je derivácia definovaná (aj keď toto pri neurónových sieťach nehrozí). Oblasť aplikácie je preto širšia ako pri použití EKF, ktorý vyžaduje existenciu Jacobiánov.

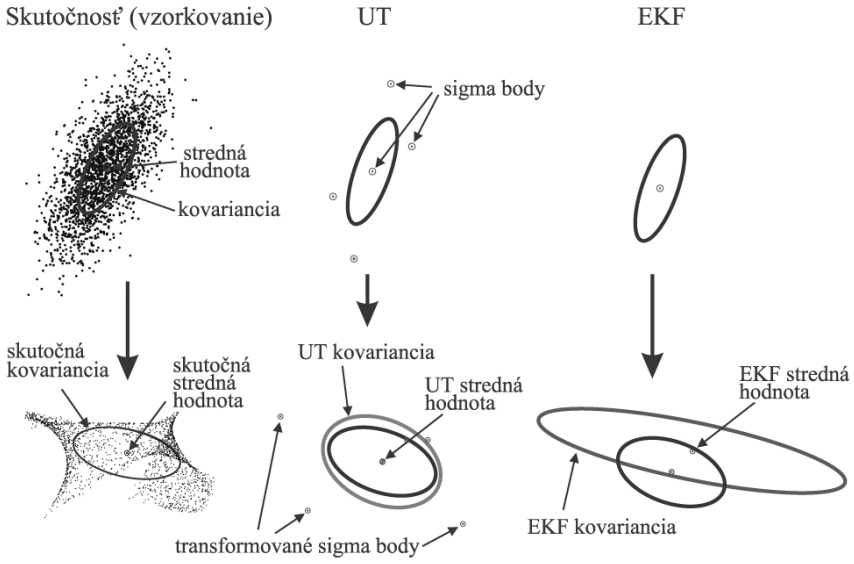
6.1 Princíp

Hlavný rozdiel medzi EKF a UKF spočíva v spôsobe, ako sú gausovské náhodné premenné (GNP) reprezentované na šírenie dynamikou systému. V EKF je stav aproximovaný pomocou GNP, ktoré sú šírené analyticky cez linearizáciu prvého rádu nelineárneho systému. Ako som už spomenul, toto môže zanášať veľké chyby v skutočnej strednej hodnote a kovariancii transformovaných GNP. UKF tento problém rieši použitím prístupu deterministického vzorkovania. Stav je tiež aproximovaný

[2] Názov naráža na to, že rozšírený Kalmanov filter „smrdí“, pretože je nelineárna funkcia aproximovaná len prvým rádom Taylorovho rozvoja.

pomocou GNP, ale je reprezentovaný použitím minimálnej množiny vhodne zvolených bodov. Tieto body sa získajú pomocou tzv. transformácie bez zápachu (angl. Unscented Transformation) [10, 9].

Transformácia bez zápachu je relatívne nová metóda na výpočet štatistík náhodnej premennej, ktorá prešla nelineárnou transformáciou. Je založená na tom, že je jednoduchšie aproximovať gausovské rozdelenie ako aproximovať všeobecnú nelineárnu funkciu, či transformáciu [9]. Princíp je znázornený na obrázku 4 v strede. Vyberie sa množina tzv. *sigma bodov* tak, aby tieto body úplne zachytávali skutočnú strednú hodnotu a kovarianciu GNP. Po prešírení *skutočným* nelineárnym systémom presne zachytávajú posteriórnu strednú hodnotu a kovarianciu minimálne do druhého stupňa Taylorovho rozvoja pre *ľubovoľnú* nelinearitu. Naproti tomu EKF dosahuje presnosť len do prvého stupňa.



Obr. 4. Príklad šírenia strednej hodnoty a kovariancie a) skutočnej b) pri transformácii „bez zápachu“ c) pri použití EKF [10]

Táto metóda pripomína Monte Carlo metódy, kedy sa náhodne vyberie veľa vzoriek, ktoré následne tiež prejdú nelineárnou transformáciou (obr. 4 vľavo). Tu je ale značný rozdiel v tom, že sigma body nie sú vyberané náhodne, ale deterministicky a navyše je počet sigma bodov nízky – $2n_x + 1$, kde n_x je dimenzia náhodnej premennej (veľkosť vektora) [10, 9].

6.2 Rovnice UKF

V tejto časti uvedieme rovnice UKF z [2], nakoľko pôvodní autori [9] uvádzajú filtre pre všeobecný dynamický systém, kedy nelinearita vystupuje ako v stavovej rovnici, tak aj v rovnici pozorovania, a takisto šum nemusí pôsobiť lineárne. My ale takúto

všeobecnosť nepotrebujeme, pretože neurónovú sieť môžeme vyjadriť ako dynamický systém

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{q}_k$$
$$\mathbf{y}_k = g(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{r}_k$$

kde $\mathbf{x}$ je stavový vektor – v našom prípade vektor váh siete, $\mathbf{q}$ je biely gausovský procesný šum s kovarianciou $\mathbf{Q}$, $\mathbf{y}$ je výstup systému (požadovaný výstup siete), funkcia g vyjadruje skutočný výstup siete na základe vstupu $\mathbf{u}$ a aktuálnych váh, a $\mathbf{r}$ je biely gausovský šum pozorovania s kovarianciou $\mathbf{R}$.

Vo vzorcoch pre prehľadnosť uvádzame v hranatých zátvorkách rozmery jednotlivých matíc v tvare [*počet riadkov* × *počet stĺpcov*]. Matice s jedným stĺpcom sú vlastne stĺpcové vektory. Výraz $\begin{bmatrix}\mathbf{A} & \mathbf{B}\end{bmatrix}$ znamená spojenie matíc $\mathbf{A}$ a $\mathbf{B}$ do jednej. $\mathbf{A}_i$ predstavuje i-ty stĺpec matice $\mathbf{A}$, n_x je veľkosť stavového vektora, n_o počet výstupov, $\mathbf{I}$ je jednotková matica.

Výraz $\mathbf{B} = \sqrt{\mathbf{A}}$ znamená, že platí rovnosť $\mathbf{A} = \mathbf{B}\mathbf{B}^T$ a nazýva sa Choleskyho faktor [11]. Pri UKF nezáleží na voľbe tohto faktora (existuje ich viac), preto je možné použiť efektívne a numericky stabilné metódy, ako napr. Choleskyho dekompozíciu [12].

Pre prehľadnosť uvádzame sumár všetkých rovníc aj s odporúčanými hodnotami parametrov [2, 9]. Po inicializácii sa rovnako ako pri EKF striedajú kroky predikcie a korekcie.

- Inicializácia
  1. $\hat{\mathbf{x}}_{[n_x \times 1]}$, $\hat{\mathbf{P}}_{[n_x \times n_x]} = \mathbf{I} P_0$, $\mathbf{Q}_{[n_x \times n_x]} = \mathbf{I} Q_0$, $\mathbf{R}_{[n_o \times n_o]} = \mathbf{I} R_0$, $\kappa = 3 - n_x$, $\beta = 2$, $1 \le \alpha \le 10^{-4}$, $\lambda = \alpha^2 (n_x + \kappa) - n_x$, $\gamma = \sqrt{n_x + \lambda}$
  2. $W_0^{(m)} = \dfrac{\lambda}{n_x + \lambda}$

     $W_0^{(c)} = \dfrac{\lambda}{n_x + \lambda} + 1 - \alpha^2 + \beta$

     $W_i^{(m)} = W_i^{(c)} = \dfrac{1}{2(n_x + \lambda)}, i = 1, \ldots, 2n_x$

- Časová aktualizácia
  1. $\bar{\mathbf{x}}_{[n_x \times 1]} = \hat{\mathbf{x}}$
  2. $\bar{\mathbf{P}}_{[n_x \times n_x]} = \hat{\mathbf{P}} + \mathbf{Q}$
- Aktualizácia pozorovaním
  - Pomocné výpočty
    1. $\mathcal{X}_{[n_x \times 2n_x + 1]} = \begin{bmatrix}\bar{\mathbf{x}} & \bar{\mathbf{x}} + \gamma\sqrt{\bar{\mathbf{P}}} & \bar{\mathbf{x}} - \gamma\sqrt{\bar{\mathbf{P}}}\end{bmatrix}$ (sigma body)
    2. $\mathcal{Y}_{[n_o \times 2n_x + 1]} = g(\mathcal{X}, \mathbf{u})$

3. $\bar{\mathbf{y}}_{[n_o \times 1]} = \sum_{i=0}^{2n_x} W_i^{(m)} \mathcal{Y}_i$

4. $\mathbf{P}_{yy[n_o \times n_o]} = \sum_{i=0}^{2n_x} W_i^{(c)} (\mathcal{Y}_i - \bar{\mathbf{y}})(\mathcal{Y}_i - \bar{\mathbf{y}})^T + \mathbf{R}$

5. $\mathbf{P}_{xy[n_x \times n_o]} = \sum_{i=0}^{2n_x} W_i^{(c)} (\mathcal{X}_i - \bar{\mathbf{x}})(\mathcal{Y}_i - \bar{\mathbf{y}})^T$

6. $\mathbf{K}_{[n_x \times n_o]} = \mathbf{P}_{xy} \mathbf{P}_{yy}^{-1}$

- Aktualizácia odhadu stavu a kovariancie
  1. $\hat{\mathbf{x}} = \bar{\mathbf{x}} + \mathbf{K}(\mathbf{y}_d - \bar{\mathbf{y}})$
  2. $\hat{\mathbf{P}} = \bar{\mathbf{P}} - \mathbf{K}\mathbf{P}_{yy}\mathbf{K}^T$

Stav predstavuje v prípade neurónových sietí vektor všetkých trénovateľných váh. Odhad stavového vektora a kovariancie pred pozorovaním je označený $\bar{\mathbf{x}}$ a $\bar{\mathbf{P}}$, ich odhad po pozorovaní ako $\hat{\mathbf{x}}$ a $\hat{\mathbf{P}}$. Pozorovaním sa v prípade neurónových sietí rozumie požadovaný výstup v danom čase ($\mathbf{y}_d$). Konštanta α určuje rozptýlenie sigma bodov okolo $\bar{\mathbf{x}}$. κ je druhotný škálovací koeficient (nakoľko ovplyvňuje primárny koeficient λ). Uvedená hodnota je pre gausovské rozdelenie vhodnou heuristikou. Parameter β zahŕňa znalosť rozdelenia – pre gausovské je optimálne $\beta = 2$ [9].

V [13] je dôležité upozornenie ohľadom trénovania rekurentných neurónových sietí filtrom nprKF (spomínaný v nasledujúcej kapitole), ktoré analogicky platí aj pre UKF. V takomto prípade je pri výpočte funkcie *g* potrebné prešíriť sieť niekoľko krokov z minulosti s novými váhami, podobne ako pri výpočte derivácií pomocou BPTT(*h*). Na to je samozrejme potrebné pamätať si vstupy do siete a výstupy aspoň rekurentných neurónov z minulých krokov. Slúži to na prispôsobenie aktivácií rekurentných neurónov na nový vektor váh.

Pri EKF je možné ukázať, že voľba absolútnych hodnôt parametrov P_0, R_0 a Q_0 neovplyvní trénovanie, pokiaľ je zachovaný pomer $P_0 : R_0 : Q_0$ [13]. Pri UKF je však dôležitá aj ich veľkosť. Hodnota P_0 determinuje škálovanie počiatočnej množiny sigma bodov, podobne ako parameter κ. Podľa odporúčaní v [13] je voľba $P_0 = 1$ veľa, a zase príliš malé hodnoty ako $P_0 = 10^{-6}$ spôsobia priblíženie sa k výkonu EKF, čo nemá veľmi praktický význam. Podľa ich skúseností sa im osvedčili hodnoty okolo $P_0 = 0.1$. V mnou vykonaných experimentoch som používal túto hodnotu, ale aj hodnotu $P_0 = 0.01$. Tiež sa v nich ukázala citlivosť na nastavenie tohto parametru, kedy pri nevhodnom nastavení dochádza dokonca k divergencii.

Tiež hodnota R_0 má veľký dopad na trénovanie. Môžeme ho položiť rovné P_0 alebo menšie a postupne ho zmenšovať, prípadne ho ďalej nemeniť. Existujú aj ďalšie metódy na jeho aktualizáciu počas trénovania – viď. napr. [2].

Aj keď voľba parametra Q_0 nie je kľúčová pre stabilitu trénovacieho procesu, jeho nenulová hodnota môže predstavovať mechanizmus na vyhnutie sa lokálnemu minimu, tak ako je to pri EKF [13]. Jeho hodnotu volíme malú, typicky $\leq 10^{-4} P_0$.

7 Filter nprKF

Pri UKF sa na výpočet štatistík náhodnej premennej, ktorá prešla nelineárnou transformáciou, používajú sigma body a transformácia bez zápachu. Pri EKF sa používa prvý stupeň Taylorovho rozvoja. Ďalším prístupom, resp. ďalšou modifikáciou Kalmanovho filtra je filter s názvom nprKF [7], pomenovaný po jeho autoroch (oni samotní ho nazvali DD2 [8]).

7.1 Princíp

Rovnako ako UKF, ani pri tomto filtri nie je potrebné počítať derivácie, nakoľko je Taylorov rozvoj nahradený Stirlingovou formulou pre aproximáciu funkcie na intervale, pričom sa funkcia *f* aproximuje prvými dvoma stupňami [8]. Konkrétne v dvojrozmernom priestore má aproximácia druhého rádu funkcie *f* tvar

$$f(x) \approx f(\bar{x}) + f'_{DD}(\bar{x})(x-\bar{x}) + \frac{f''_{DD}(\bar{x})}{2!}(x-\bar{x})^2 \tag{52}$$

kde *h* určuje dĺžku intervalu, pre ktorý je aproximácia najlepšia, $\bar{x}$ je bod, v okolí ktorého aproximujeme a

$$f'_{DD}(\bar{x}) = \frac{f(\bar{x}+h) - f(\bar{x}-h)}{2h} \quad f''_{DD}(\bar{x}) = \frac{f(\bar{x}+h) + f(\bar{x}-h) - 2f(\bar{x})}{h^2} \tag{53}$$

Rovnicu (52) môžeme teda chápať ako Taylorov rozvoj, kde sme derivácie nahradili pomernými diferenciami (angl. *divided difference* – odtiaľ pôvodný názov filtru DD2).

Nech $\mathbf{x} \in R^n$ je vektor a nech $\mathbf{y} = f(\mathbf{x})$ je vektorová funkcia. Potom interpolačná formula pre viac rozmerov s polynómami do druhého stupňa má tvar [8]:

$$\mathbf{y} = f(\bar{\mathbf{x}} + \Delta\mathbf{x}) \approx f(\bar{\mathbf{x}}) + \tilde{D}_{\Delta\mathbf{x}} f + \frac{1}{2!}\tilde{D}^2_{\Delta\mathbf{x}} f \tag{54}$$

Ako operátory pomernej diferencie $\tilde{D}_{\Delta\mathbf{x}}, \tilde{D}^2_{\Delta\mathbf{x}}$ použijeme

$$\tilde{D}_{\Delta\mathbf{x}} = \frac{1}{h}\left(\sum_{p=1}^{n} \Delta\mathbf{x}_p \mu_p \delta_p\right) f(\bar{\mathbf{x}}) \tag{55}$$

$$\tilde{D}^2_{\Delta\mathbf{x}} = \frac{1}{h^2}\left(\sum_{p=1}^{n}(\Delta\mathbf{x}_p)^2\delta_p^2 + \sum_{p=1}^{n}\sum_{q=1,q\neq p}^{n}\Delta\mathbf{x}_p\Delta\mathbf{x}_q(\mu_p\delta_p)(\mu_q\delta_q)\right)f(\bar{\mathbf{x}}) \tag{56}$$

kde δ_p predstavuje operátor „parciálnej" pomernej diferencie a μ_p operátor priemerovania

$$\delta_p f(\bar{\mathbf{x}}) = f(\bar{\mathbf{x}} + \frac{h}{2}\mathbf{e}_p) - f(\bar{\mathbf{x}} - \frac{h}{2}\mathbf{e}_p) \tag{57}$$

$$\mu_p f(\bar{\mathbf{x}}) = \frac{1}{2}\left(f(\bar{\mathbf{x}} + \frac{h}{2}\mathbf{e}_p) + f(\bar{\mathbf{x}} - \frac{h}{2}\mathbf{e}_p)\right) \tag{58}$$

kde $\mathbf{e}_p$ predstavuje *p*-ty jednotkový vektor. Rovnica (54) je však len jedna z možností, ako rozšíriť Stirlingovu formulu na viac rozmerov [8]. Pre ilustráciu, ako je možné odvodiť ďalšie, zavedieme nasledovnú lineárnu transformáciu premennej $\mathbf{x}$ a definujeme funkciu $\tilde{f}$:

$$\begin{gathered}\mathbf{z} = \mathbf{S}^{-1}\mathbf{x} \\ \tilde{f}(\mathbf{z}) \equiv f(\mathbf{S}\mathbf{z}) = f(\mathbf{x})\end{gathered} \tag{59}$$

Taylorove aproximácie funkcií $\tilde{f}$ a f sú rovnaké, čo ale neplatí pre viacrozmernú interpolačnú formulu (54). Operátory $\tilde{D}_{\Delta\mathbf{x}}f$ a $\tilde{D}^2_{\Delta\mathbf{x}}f$ sa odlišujú od $\tilde{D}_{\Delta\mathbf{x}}\tilde{f}$ a $\tilde{D}^2_{\Delta\mathbf{x}}\tilde{f}$, nakoľko

$$2\mu_p\delta_p\tilde{f}(\bar{\mathbf{z}}) = \tilde{f}(\bar{\mathbf{z}} + h\mathbf{e}_p) - \tilde{f}(\bar{\mathbf{z}} - h\mathbf{e}_p) = f(\bar{\mathbf{x}} + h\mathbf{s}_p) - f(\bar{\mathbf{x}} - h\mathbf{s}_p) \tag{60}$$

kde $\mathbf{s}_p$ predstavuje *p*-ty stĺpec matice $\mathbf{S}$.

Nech $\mathbf{x}$ je vektor náhodných premenných, pre ktoré poznáme očakávanú (strednú) hodnotu a kovarianciu

$$\bar{\mathbf{x}} = E[\mathbf{x}], \quad \mathbf{P}_x = E\left[(\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^T\right] \tag{61}$$

Našou úlohou je zistiť štatistiky týchto premenných po prechode nelineárnou transformáciou:

$$\bar{\mathbf{y}} = E[f(\mathbf{x})] \tag{62}$$

$$\mathbf{P}_y = E\left[(f(\mathbf{x}) - \bar{\mathbf{y}})(f(\mathbf{x}) - \bar{\mathbf{y}})^T\right] \tag{63}$$

$$\mathbf{P}_{xy} = E\left[(\mathbf{x} - \bar{\mathbf{x}})(f(\mathbf{x}) - \bar{\mathbf{y}})^T \right] \tag{64}$$

Keďže f je nelineárna funkcia, nemôžeme predpokladať, že dokážeme vypočítať presné hodnoty. Preto je obvyklé pred výpočtom týchto štatistík nahradiť f jej aproximáciou polynómami prvého a/alebo druhého stupňa.

Je veľmi výhodné pracovať s transformáciou vyjadrenou rovnicou (59). Transformačnú maticu zvolíme ako Choleskyho faktor ($\mathbf{S}_x$) kovariančnej matice:

$$\mathbf{z} = \mathbf{S}_x^{-1}\mathbf{x}, \quad \mathbf{P}_x = \mathbf{S}_x \mathbf{S}_x^T$$

Pri takejto transformácii sa jednotlivé zložky vektora $\mathbf{z}$ stanú navzájom nezávislé (a každá s jednotkovým rozptylom):

$$E\left[(\mathbf{z} - E[\mathbf{z}])(\mathbf{z} - E[\mathbf{z}])^T \right] = \mathbf{I}$$

V práci [8] je ďalej odvodený filter nprKF. Konkrétne v ňom autori použili aproximáciu Stirlingovou formulou druhého rádu a kvôli numerickej stabilite narábajú priamo s Choleskyho faktormi kovariančných matíc. Rovnice pre tento filter odvodili pre všeobecný dynamický systém, kedy nelinearita vystupuje ako v stavovej rovnici, tak aj v rovnici pozorovania, ba dokonca šum, či už procesný alebo pozorovania, nemusí pôsobiť lineárne.

V ďalšom texte uvedieme menej všeobecné vzorce, ktoré je možné priamo použiť na trénovanie neurónových sietí. Vzorcov je preto menej a sú jednoduchšie. S mierne odlišnou notáciou nájdeme takto zjednodušené vzorce aj v [13]. Takisto je kladený dôraz na to, aby boli vzorce čo najprehľadnejšie a zároveň na vysvetlenie niekoľkých skutočností, ktoré nie sú hneď zrejmé, alebo ktoré je dôležité vedieť špeciálne pri trénovaní rekurentných neurónových sietí.

7.2 Rovnice

V tejto časti uvádzame základnú verziu filtra nprKF určenú (aj) na trénovanie neurónových sietí. Pre úplnosť, neurónovú sieť môžeme vyjadriť ako dynamický systém

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{x}_k + \mathbf{q}_k \\ \mathbf{y}_k &= g(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{r}_k \end{aligned}$$

kde $\mathbf{x}$ je stavový vektor – v našom prípade vektor váh siete, $\mathbf{q}$ je biely gausovský procesný šum s kovarianciou $\mathbf{Q}$, $\mathbf{y}$ je výstup systému (požadovaný výstup siete), funkcia g vyjadruje skutočný výstup siete na základe vstupu $\mathbf{u}$ a aktuálnych váh, a $\mathbf{r}$ je biely gausovský šum pozorovania s kovarianciou $\mathbf{R}$.

Uvádzame vzorce predstavujúce algoritmus fungovania nprKF. Opäť ide o už známu rekurziu „predikcia – korekcia". Pre prehľadnosť uvádzame v hranatých zátvorkách rozmery matíc (vektorov). Pripomeňme, že v týchto rovniciach vystupujú Choleskyho faktory jednotlivých kovariancií.

- Inicializácia
  $\hat{\mathbf{x}}_{[n_x\times 1]}$, $\hat{\mathbf{S}}_{x[n_x\times n_x]} = \mathbf{I}P_0^{1/2}$, $\mathbf{S}_{v[n_x\times n_x]} = \mathbf{I}Q_0^{1/2}$, $\mathbf{S}_{w[n_o\times n_o]} = \mathbf{I}R_0^{1/2}$, $h^2 = 3$
- Časová aktualizácia – predikcia
  1. $\bar{\mathbf{x}}_{[n_x\times 1]} = \hat{\mathbf{x}}$
  2. $\bar{\mathbf{S}}_{x[n_x\times n_x]} = \begin{bmatrix}\hat{\mathbf{S}}_x & \mathbf{S}_v\end{bmatrix}_{[n_x\times 2n_x]}$ a zredukujeme na trojuholníkovú maticu
- Aktualizácia pozorovaním – korekcia
  - Pomocné výpočty
    1. $\bar{\mathbf{y}}_{[n_o\times 1]} = \dfrac{h^2-n_x}{h^2}g(\bar{\mathbf{x}},\mathbf{u}) + \dfrac{1}{2h^2}\sum_{p=1}^{n_x}\left[g(\bar{\mathbf{x}}+h\bar{\mathbf{s}}_{x,p},\mathbf{u}) + g(\bar{\mathbf{x}}-h\bar{\mathbf{s}}_{x,p},\mathbf{u})\right]$
    2. $\mathbf{S}^{(1)}_{y\bar{x}[n_o\times n_x]}$: $(\mathbf{S}^{(1)}_{y\bar{x}})_{i,j} = \dfrac{1}{2h}\left[g_i(\bar{\mathbf{x}}+h\bar{\mathbf{s}}_{x,j},\mathbf{u}) - g_i(\bar{\mathbf{x}}-h\bar{\mathbf{s}}_{x,j},\mathbf{u})\right]$
    3. $\mathbf{S}^{(2)}_{y\bar{x}[n_o\times n_x]}$:
       $(\mathbf{S}^{(2)}_{y\bar{x}})_{i,j} = \dfrac{\sqrt{h^2-1}}{2h^2}\left[g_i(\bar{\mathbf{x}}+h\bar{\mathbf{s}}_{x,j},\mathbf{u}) + g_i(\bar{\mathbf{x}}-h\bar{\mathbf{s}}_{x,j},\mathbf{u}) - 2g_i(\bar{\mathbf{x}},\mathbf{u})\right]$
    4. $\mathbf{S}_{y[n_o\times n_o]} = \begin{bmatrix}\mathbf{S}^{(1)}_{y\bar{x}} & \mathbf{S}_w & \mathbf{S}^{(2)}_{y\bar{x}}\end{bmatrix}_{[n_o\times n_x+n_o+n_x]}$
       zredukujeme na trojuholníkovú maticu
    5. $\mathbf{K}_{[n_x\times n_o]} = \bar{\mathbf{S}}_x(\mathbf{S}^{(1)}_{y\bar{x}})^T\left(\mathbf{S}_y\mathbf{S}_y^{\ T}\right)^{-1}$
  - Aktualizácia odhadu stavu a kovariancie
    1. $\hat{\mathbf{x}} = \bar{\mathbf{x}} + \mathbf{K}(\mathbf{y}_d - \bar{\mathbf{y}})$
    2. $\hat{\mathbf{S}}_x = \begin{bmatrix}\bar{\mathbf{S}}_x - \mathbf{K}\mathbf{S}^{(1)}_{y\bar{x}} & \mathbf{K}\mathbf{S}_w & \mathbf{K}\mathbf{S}^{(2)}_{y\bar{x}}\end{bmatrix}_{[n_x\times n_x+n_o+n_x]}$
       zredukujeme na trojuholníkovú maticu

Výraz $\bar{\mathbf{s}}_{x,j}$ znamená j-ty stĺpec matice $\bar{\mathbf{S}}_x$. Hodnota $g_i(\bar{\mathbf{x}},\mathbf{u})$ predstavuje i-tu zložku výstupného vektora siete pre hodnotu váh určených vektorom $\bar{\mathbf{x}}$ a vstupmi $\mathbf{u}$. Rovnako, ako pri UKF, je pri výpočte g pri rekurentných sieťach potrebné prešíriť sieť niekoľko krokov z minulosti, na čo je potrebné pamätať si vstupy do siete a výstupy aspoň rekurentných neurónov.

Redukovanie obdĺžnikovej matice **A** na trojuholníkovú znamená nájsť hornú alebo dolnú trojuholníkovú maticu **S** tak, aby platila rovnosť $\mathbf{A}\mathbf{A}^T = \mathbf{S}\mathbf{S}^T$. Toto je možné dosiahnuť Householderovými transformáciami (viď. napr. [12]). Je to kvôli tomu, aby

sa zachoval vzťah, kedy napr. $\overline{\mathbf{P}} = \overline{\mathbf{S}}_x \overline{\mathbf{S}}_x^{\,T}$, kde $\overline{\mathbf{S}}_x$ je Choleskyho faktor kovariančnej matice. Tento ale musí byť štvorcový, kvôli výpočtu napr. matice $\mathbf{S}_{y\overline{x}}^{(1)}$. Spomínaná triangularizácia je jedna z efektívnych metód, ako toto dosiahnuť.

To, že je možné v niektorých vzorcoch spájať matice, je spôsobené predovšetkým tým, že stav a šumy sú navzájom stochasticky nezávislé. Pre podrobnosti viď [8].

Pre nastavenie parametrov P_0, Q_0 a R_0 platia rovnaké zásady ako pri UKF. Rovnako aj nprKF je citlivý na ich hodnoty. P_0 volíme rovné približne 0.01–0.1, R_0 rovné P_0 alebo menej a $Q_0 = 10^{-4} P_0$. Pre dôvody viď. predchádzajúcu časť.

Pri pohľade na uvedené vzorce a na vzorce filtra UKF je zrejmá podobnosť oboch filtrov. Autori nprKF explicitne nehovoria o sigma bodoch, v skutočnosti ich vlastne tiež využívajú a dokonca aj rovnaký počet $(2n_x + 1)$ – stačí spočítať, koľko potrebujeme vyhodnotení funkcie *g* pri aktualizácii pozorovaním (najlepšie to vidno vo vzorci na výpočet $\overline{\mathbf{y}}$). Potvrdzuje to aj analýza v [8], nakoľko odhad strednej hodnoty stavu je u oboch filtrov identický. Odlišujú sa však v odhade kovariancie. nprKF o niečo lepšie zachytáva jej skutočnú hodnotu – rozdiely v aktualizácii kovariancie sa však v porovnaní s UKF pohybujú okolo štvrtého a vyššieho stupňa Taylorovho rozvoja. Odhad kovariancie pomocou UKF je teda predsa len menej presný a navyše môže niekedy viesť na odhad, ktorý nie je kladne semidefinitný. Podrobnejšie opäť viď. [8].

Podľa [8] sú tieto filtre všeobecne pre nelineárne systémy jednoducho implementovateľné, nakoľko nie sú potrebné žiadne derivácie, a predsa poskytujú vynikajúcu presnosť. Ale to, že nie je potrebné derivovať, nemá pre neurónové siete veľký význam, pretože tie sú sami o sebe diferencovateľné. Pre nás je zaujímavé najmä to, že oba spomínané prístupy dokážu presnejšie predikovať stav systému v porovnaní s rozšíreným Kalmanovým filtrom [9, 14].

8 Zdieľané Kalmanove filtre

Ako sme spomenuli v predchádzajúcej časti, pri trénovaní rekurentných sietí je potrebné pre každý sigma bod (t.j. rôzne váhy) prešíriť sieť niekoľko krokov z minulosti.

8.1 Zdieľaný „Unscented“ Kalmanov filter

Druhý spôsob spočíva v zahrnutí aktivácií rekurentných neurónov do stavového vektora, čím použijeme UKF na odhad váh aj stavu zároveň [7]. Pri takto formulovanom probléme, ktorý sa nazýva zdieľaný Kalmanov filter bez zápachu (angl. Joint UKF – UKFj), je potrebné pozmeniť stavovú rovnicu a následne aj rovnice UKF. Treba upozorniť, že tento názov sa používa takisto pre filter, v ktorom sa do jedného stavového vektora spoja stav systému so vstupmi [2]. Toto sa používa v situácii, kedy nemáme k dispozícii nezašumený vstup do systému.

UKFj teda na jednej strane zväčšuje výpočtovú náročnosť tým, že rozširuje veľkosť stavového vektora $\mathbf{x}$, na strane druhej zjednodušuje výpočet výstupu siete pre rôzne hodnoty váh (sigma bodov).

Rozdiel oproti UKF je jedine v časovej aktualizácii, nakoľko sa funkcia *f* pridaním ďalších stavov stala nelineárna.

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{q}_k$$
$$\mathbf{y}_k = g(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{r}_k$$

- Inicializácia
  1. Počiatočný odhad stavu a kovariancií a parametre pre UKF
     $\hat{\mathbf{x}}_{[n_x \times 1]}$, $\hat{\mathbf{P}}_{[n_x \times n_x]} = \mathbf{I} P_0$, $\mathbf{Q}_{[n_x \times n_x]} = \mathbf{I} Q_0$, $\mathbf{R}_{[n_o \times n_o]} = \mathbf{I} R_0$, $\kappa = 3 - n_x$, $\beta = 2$, $1 \le \alpha \le 10^{-4}$, $\lambda = \alpha^2 (n_x + \kappa) - n_x$, $\gamma = \sqrt{n_x + \lambda}$
  2. Váhy pre UKF
     $$W_0^{(m)} = \frac{\lambda}{n_x + \lambda}$$
     $$W_0^{(c)} = \frac{\lambda}{n_x + \lambda} + 1 - \alpha^2 + \beta$$
     $$W_i^{(m)} = W_i^{(c)} = \frac{1}{2(n_x + \lambda)}, i = 1, \ldots, 2n_x$$
- Časová aktualizácia – predikcia
  - Pomocné výpočty
    1. $\mathcal{X}_{[n_x \times 2n_x + 1]} = \begin{bmatrix} \hat{\mathbf{x}} & \hat{\mathbf{x}} + \gamma\sqrt{\hat{\mathbf{P}}} & \hat{\mathbf{x}} - \gamma\sqrt{\hat{\mathbf{P}}} \end{bmatrix}$ (sigma body)
    2. $\mathcal{X}^*_{[n_x \times 2n_x + 1]} = f(\mathcal{X}, \mathbf{u})$
  - Aktualizácia odhadu stavu a kovariancie
    1. $\bar{\mathbf{x}}_{[n_x \times 1]} = \sum_{i=0}^{2n_x} W_i^{(m)} \mathcal{X}_i^*$
    2. $\bar{\mathbf{P}}_{[n_x \times n_x]} = \sum_{i=0}^{2n_x} W_i^{(c)} (\mathcal{X}_i^* - \bar{\mathbf{x}})(\mathcal{X}_i^* - \bar{\mathbf{x}})^T + \mathbf{Q}$
- Aktualizácia pozorovaním – korekcia
  - Pomocné výpočty
    1. $\bar{\mathcal{X}}_{[n_x \times 2n_x + 1]} = \begin{bmatrix} \bar{\mathbf{x}} & \bar{\mathbf{x}} + \gamma\sqrt{\bar{\mathbf{P}}} & \bar{\mathbf{x}} - \gamma\sqrt{\bar{\mathbf{P}}} \end{bmatrix}$ (sigma body)
    2. $\mathcal{Y}_{[n_o \times 2n_x + 1]} = g(\bar{\mathcal{X}}, \mathbf{u})$
    3. $\bar{\mathbf{y}}_{[n_o \times 1]} = \sum_{i=0}^{2n_x} W_i^{(m)} \mathcal{Y}_i$

4. $\mathbf{P}_{yy[n_o \times n_o]} = \sum_{i=0}^{2n_x} W_i^{(c)} (\mathcal{Y}_i - \overline{\mathbf{y}})(\mathcal{Y}_i - \overline{\mathbf{y}})^T + \mathbf{R}$
5. $\mathbf{P}_{xy[n_x \times n_o]} = \sum_{i=0}^{2n_x} W_i^{(c)} (\overline{\mathcal{X}}_i - \overline{\mathbf{x}})(\mathcal{Y}_i - \overline{\mathbf{y}})^T$
6. $\mathbf{K}_{[n_x \times n_o]} = \mathbf{P}_{xy} \mathbf{P}_{yy}^{-1}$

- Aktualizácia odhadu stavu a kovariancie
  1. $\hat{\mathbf{x}} = \overline{\mathbf{x}} + \mathbf{K}(\mathbf{y}_d - \overline{\mathbf{y}})$
  2. $\hat{\mathbf{P}} = \overline{\mathbf{P}} - \mathbf{K}\mathbf{P}_{yy}\mathbf{K}^T$

Návratovou hodnotou funkcie $f(\mathbf{x},\mathbf{u})$ je v tomto prípade nový stavový vektor po tom, čo sa nastavili váhy a výstupy rekurentných neurónov podľa $\mathbf{x}$ a sieť prešírime (jeden časový krok) pri vstupnom vektore $\mathbf{u}$. Z toho vyplýva, že pre zložky stavového vektora $\mathbf{x}$, ktoré zodpovedajú váham, platí $f(\mathbf{x}) = \mathbf{x}$, a že funkcia f mení len výstup rekurentných neurónov.

Funkciu $g(\mathbf{x})$ treba interpretovať ako vektor výstupov siete po tom, čo sa nastavili váhy a výstupy rekurentných neurónov a neurónom, ktorých výstupy nie sú súčasťou stavového vektora, sa prepočíta ich výstup. Konkrétne pre Elmanovu architektúru preto stačí nastaviť (okrem váh) výstupy rekurentných neurónov v skrytej vrstve a už len vypočítať výstupy neurónov vo výstupnej vrstve. Takáto interpretácia je konzistentná s obvyklou rovnicou pozorovania v dynamických systémoch, kedy je výstup systému funkciou jeho stavových premenných [7].

8.2 Zdieľaný filter nprKF

Rovnako ako pri UKF sa môžeme vyhnúť ďalšiemu parametru, t.j. od koľkých krokov prešíriť sieť pre každý vektor váh tak, že spojíme váhy s výstupmi rekurentných neurónov do stavového vektora. T.j. na odhad stavu aj váh použijeme jeden nprKF filter, ktorý potom nazývame zdieľaný nprKF (angl. Joint nprKF – nprKFj) [7].

Podobne sa oproti nprKF zmení len krok predikcie, keďže sa stavová rovnica stala nelineárnou:

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{q}_k$$
$$\mathbf{y}_k = g(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{r}_k$$

- Inicializácia
  $\hat{\mathbf{x}}_{[n_x \times 1]}$, $\hat{\mathbf{S}}_{x[n_x \times n_x]} = \mathbf{I} P_0^{1/2}$, $\mathbf{S}_{v[n_x \times n_x]} = \mathbf{I} Q_0^{1/2}$, $\mathbf{S}_{w[n_o \times n_o]} = \mathbf{I} R_0^{1/2}$, $h^2 = 3$
- Časová aktualizácia
  - Pomocné výpočty
    1. $\mathbf{S}^{(1)}_{x\hat{x}[n_x \times n_x]}$: $(\mathbf{S}^{(1)}_{x\hat{x}})_{i,j} = \frac{1}{2h}\left[f_i(\hat{\mathbf{x}} + h\hat{\mathbf{s}}_{x,j}, \mathbf{u}) - f_i(\hat{\mathbf{x}} - h\hat{\mathbf{s}}_{x,j}, \mathbf{u}) \right]$

2. $\mathbf{S}^{(2)}_{x\hat{x}\,[n_x \times n_x]}$:

$$(\mathbf{S}^{(2)}_{x\hat{x}})_{i,j} = \frac{\sqrt{h^2-1}}{2h^2}\left[f_i(\hat{\mathbf{x}} + h\hat{\mathbf{s}}_{x,j}, \mathbf{u}) + f_i(\hat{\mathbf{x}} - h\hat{\mathbf{s}}_{x,j}, \mathbf{u}) - 2f_i(\hat{\mathbf{x}}, \mathbf{u}) \right]$$

- Aktualizácia odhadu stavu a kovariancie
  1. $\bar{\mathbf{x}}_{[n_x \times 1]} = \frac{h^2 - n_x}{h^2} f(\hat{\mathbf{x}}, \mathbf{u}) + \frac{1}{2h^2}\sum_{p=1}^{n_x}\left[f(\hat{\mathbf{x}} + h\hat{\mathbf{s}}_{x,p}, \mathbf{u}) + f(\hat{\mathbf{x}} - h\hat{\mathbf{s}}_{x,p}, \mathbf{u}) \right]$
  2. $\bar{\mathbf{S}}_{x[n_x \times n_x]} = \left[\mathbf{S}^{(1)}_{x\hat{x}} \quad \mathbf{S}_v \quad \mathbf{S}^{(2)}_{x\hat{x}} \right]_{[n_x \times 3n_x]}$

     zredukujeme na trojuholníkovú maticu

- Aktualizácia pozorovaním
  - Pomocné výpočty
    1. $\bar{\mathbf{y}}_{[n_o \times 1]} = \frac{h^2 - n_x}{h^2} g(\bar{\mathbf{x}}, \mathbf{u}) + \frac{1}{2h^2}\sum_{p=1}^{n_x}\left[g(\bar{\mathbf{x}} + h\bar{\mathbf{s}}_{x,p}, \mathbf{u}) + g(\bar{\mathbf{x}} - h\bar{\mathbf{s}}_{x,p}, \mathbf{u}) \right]$
    2. $\mathbf{S}^{(1)}_{y\bar{x}[n_o \times n_x]}$: $(\mathbf{S}^{(1)}_{y\bar{x}})_{i,j} = \frac{1}{2h}\left[g_i(\bar{\mathbf{x}} + h\bar{\mathbf{s}}_{x,j}, \mathbf{u}) - g_i(\bar{\mathbf{x}} - h\bar{\mathbf{s}}_{x,j}, \mathbf{u}) \right]$
    3. $\mathbf{S}^{(2)}_{y\bar{x}\,[n_o \times n_x]}$:

       $$(\mathbf{S}^{(2)}_{y\bar{x}})_{i,j} = \frac{\sqrt{h^2-1}}{2h^2}\left[g_i(\bar{\mathbf{x}} + h\bar{\mathbf{s}}_{x,j}, \mathbf{u}) + g_i(\bar{\mathbf{x}} - h\bar{\mathbf{s}}_{x,j}, \mathbf{u}) - 2g_i(\bar{\mathbf{x}}, \mathbf{u}) \right]$$
    4. $\mathbf{S}_{y[n_o \times n_o]} = \left[\mathbf{S}^{(1)}_{y\bar{x}} \quad \mathbf{S}_w \quad \mathbf{S}^{(2)}_{y\bar{x}} \right]_{[n_o \times n_x + n_o + n_x]}$

       zredukujeme na trojuholníkovú maticu
    5. $\mathbf{K}_{[n_x \times n_o]} = \bar{\mathbf{S}}_x (\mathbf{S}^{(1)}_{y\bar{x}})^T \left(\mathbf{S}_y \mathbf{S}_y{}^T \right)^{-1}$
  - Aktualizácia odhadu stavu a kovariancie
    1. $\hat{\mathbf{x}} = \bar{\mathbf{x}} + \mathbf{K}(\mathbf{y}_d - \bar{\mathbf{y}})$
    2. $\hat{\mathbf{S}}_x = \left[\bar{\mathbf{S}}_x - \mathbf{K}\mathbf{S}^{(1)}_{y\bar{x}} \quad \mathbf{K}\mathbf{S}_w \quad \mathbf{K}\mathbf{S}^{(2)}_{y\bar{x}} \right]_{[n_x \times n_x + n_o + n_x]}$

       zredukujeme na trojuholníkovú maticu

Pre úplnosť pripomenieme význam funkcií *f* a *g*, ktorý je rovnaký ako pri UKFj. Návratovou hodnotou funkcie $f(\mathbf{x}, \mathbf{u})$ je v tomto prípade nový stavový vektor po tom, čo sa nastavili váhy a výstupy rekurentných neurónov podľa $\mathbf{x}$ a sieť prešírime (jeden časový krok) pri vstupnom vektore $\mathbf{u}$. Z toho vyplýva, že pre zložky stavového vektora $\mathbf{x}$, ktoré zodpovedajú váham, platí $f(\mathbf{x}) = \mathbf{x}$, a že funkcia *f* mení len výstup rekurentných neurónov.

Funkciu $g(\mathbf{x})$ treba interpretovať ako vektor výstupov siete po tom, čo sa nastavili váhy a výstupy rekurentných neurónov a neurónom, ktorých výstupy nie sú súčasťou stavového vektora, sa prepočíta ich výstup. Konkrétne pre Elmanovu architektúru preto stačí nastaviť (okrem váh) výstupy rekurentných neurónov v skrytej vrstve a už len

vypočítať výstupy neurónov vo výstupnej vrstve. Takáto interpretácia je konzistentná s obvyklou rovnicou pozorovania v dynamických systémoch, kedy je výstup systému funkciou jeho stavových premenných [7].

9 Porovnanie jednotlivých metód

V tejto časti uvedieme experimenty zamerané na porovnanie výsledkov jednotlivých modifikácií Kalmanovej filtrácie na trénovanie rekurentných neurónových sietí s Elmanovou architektúrou.

Zvolili sme úlohu predikcie nasledujúceho symbolu v rôznych postupnostiach, pričom porovnávame výkonnosť filtrov aj s klasickou gradientovou metódou, skráteným spätným šírením chýb v čase BPTT(h).

Predikcia nasledujúceho symbolu prebieha tak, že sa na vstup siete predloží prvý symbol v poradí, pričom ako požadovaný výstup siete je symbol nasledujúci. V experimentoch na predikciu nasledujúceho symbolu sme zvolil Elmanovu architektúru rekurentnej neurónovej siete – t.j. sieť s jednou skrytou vrstvou, ktorá je rekurentná.

Na meranie úspešnosti predikcie nasledujúceho symbolu sme použili funkciu NNL (Normalized Negative Log-likelihood), ktorá je počítaná cez postupnosť symbolov v časoch $t = 1$ až T [15]:

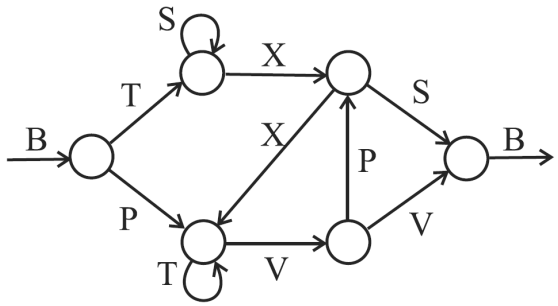
$$NNL = -\frac{1}{T}\sum_{t=1}^{T}\log_{|A|} p^{(t)}(s^{(t)}) \tag{65}$$

kde základ logaritmu $|A|$ predstavuje počet symbolov vo vstupnej abecede A a výraz $p^{(t)}(s^{(t)})$ predstavuje pravdepodobnosť predikovania symbolu $s^{(t)}$ v čase t. Ak sa NNL = 0, potom sieť predikuje nasledujúci symbol na 100%, zatiaľ čo $\mathrm{NNL} \geq 1$ zodpovedá veľmi nepresnej predikcii (hádaní).

Tento ukazovateľ môžeme takisto interpretovať vo význame, nakoľko vieme danú postupnosť skomprimovať. Nižšie hodnoty NNL znamenajú, že je možné danú postupnosť lepšie skomprimovať, naopak vyššie hodnoty znamenajú menej skomprimovateľnú postupnosť.

9.1 Reberov jazyk

Trénovacia postupnosť pri tomto experimente bola vygenerovaná použitím Reberovho automatu (obr. 5), pričom s pravdepodobnosťou 50% sa vybral jeden z dvoch možných nasledujúcich symbolov (okrem prípadu, kedy nasleduje symbol B).



Obr. 5. Reberov automat

Postupnosť pozostáva zo slova dĺžky jeden milión, ale pre naše potreby bohato postačí prvých 50001 symbolov, takže predikovať ich budeme o jeden menej. Reberova gramatika obsahuje 6 symbolov, ktoré zakódujeme pomocou one-hot kódovania a dávame sieti na vstup. Toto kódovanie znamená, že počet vstupov siete je rovný počtu symbolov gramatiky. Pre každý symbol je teda určený jeden vstup, ktorý nastavíme na hodnotu 1 a ostatné na hodnotu 0. Podobne aj výstupy sú zakódované pomocou one-hot kódovania. Týmto dokážeme jednoducho vypočítať ukazovateľ NNL (65) – stačí znormalizovať výstupný vektor (t.j. súčet jeho zložiek bude rovný 1) a pravdepodobnosť predikovania požadovaného symbolu je priamo hodnota zložky vektora prislúchajúcej danému symbolu.

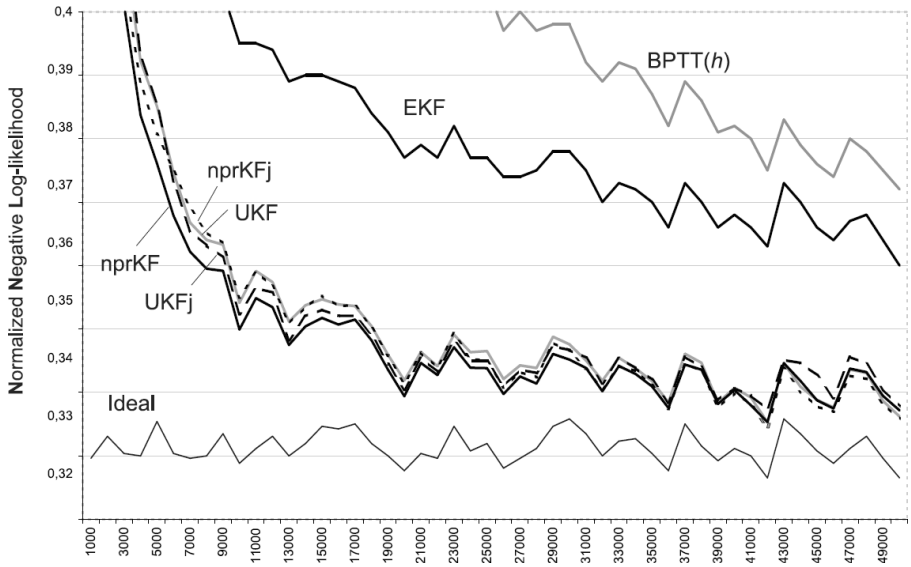
Predikcia symbolov generovaných Reberovým automatom je pomerne jednoduchý problém, pretože z predchádzajúcich dvoch symbolov sme schopní určiť, v ktorom stave sa nachádzame. Problém teda spočíva najmä v „trafení" sa do jednej z dvoch možností, t.j. neurónová sieť by mala po natrénovaní byť schopná predikovať, s akou pravdepodobnosťou nasleduje ten-ktorý symbol. Z tohto taktiež vyplýva, že NNL nemôžeme očakávať blízke nule, skôr naopak, malo by byť určite nižšie ako 0.5, nakoľko nevieme s istotou, aký bude nasledujúci symbol v prípade, ak sú dve možnosti.

Pre EKF sme zvolili nasledovné počiatočné hodnoty parametrov: kovariančná matica $\mathbf{P}_0 = 1000\,\mathbf{I}$, kovariančná matica šumu pozorovaní $\mathbf{R}_0 = 100\,\mathbf{I}$ a kovariančná matica procesného šumu $\mathbf{Q}_0 = 0.0001\,\mathbf{I}$. V priebehu trénovania sme kovariančné matice $\mathbf{R}$ a $\mathbf{Q}$ ďalej nemenili. Tieto parametre sú v zhode s odporúčaniami spomenutými v tejto kapitole.

Pre filtre UKF, nprKF, UKFj a nprKFj sme kvôli ich podobnosti zvolili rovnaké parametre v súlade s odporúčaniami. Aj experimenty potvrdili, že ich nemožno voliť rovnaké ako pri EKF. Zvolili sme preto nasledovné parametre: $\mathbf{P}_0 = 0.1\,\mathbf{I}$, $\mathbf{R}_0 = 0.01\,\mathbf{I}$ a maticu $\mathbf{Q}_0 = 10^{-6}\,\mathbf{I}$. Pre UKF a UKFj je potrebné určiť ešte jeden parameter: $\alpha = 10^{-2}$.

Pre metódu BPTT(h) sme zvolili parameter učenia $\alpha = 0.2$ a parameter $h = 10$. Rovnakú hodnotu parametru h sme zvolili aj pre filtre EKF, UKF a nprKF.

Každou metódou sme trénovali rekurentnú neurónovú sieť s troma skrytými neurónmi, čo pre tento problém postačuje. Trénovanie prebiehalo postupným predkladaním symbolov, pričom aktualizácia váh bola vykonaná v každom kroku. Ukazovateľ NNL sme počítali pre každých 1000 symbolov. Porovnanie všetkých filtrov na základe tohto ukazovateľa uvádzame na obrázku 6. Pre zaujímavosť sme do grafu pridali aj krivku znázorňujúcu „ideálnu“ hodnotu NNL. Tú dostaneme, ak predpovedáme nasledujúci symbol presne s pravdepodobnosťou 0.5, okrem prípadov, kedy nasleduje symbol B s pravdepodobnosťou 1.



Obr. 6. Závislosť NNL od počtu vstupných symbolov pre jednotlivé metódy trénovania rekurentnej siete na predikciu nasledujúceho symbolu v postupnosti generovanej Reberovou gramatikou. Zobrazené je aj ideálne NNL pre tento experiment.

Z obrázku pekne vidno výkonnosť jednotlivých metód (v zmysle rýchlosti konvergencie a celkového výsledku). Najslabšou sa podľa očakávania ukázala byť metóda BPTT(*h*). Rozšírený Kalmanov filter konverguje rýchlejšie a vo výsledku dosiahne oproti BPTT(*h*) lepší výsledok o približne 0.01. No a nakoniec je zrejmá dominancia filtrov UKF, UKFj, nprKF a nprKFj. Konvergencia všetkých štyroch je v porovnaní s BPTT(*h*) rapídna, v porovnaní s EKF veľmi rýchla. Najzaujímavejší je však najmä nimi dosiahnutý výsledok. Rozdiely medzi nimi sú minimálne – nedá sa z nich určiť „víťaza“, ale dosiahli hodnotu NNL blízku „ideálnej“.

V ďalších experimentoch s EKF a BPTT(*h*) sme skúšali rôzne počty skrytých neurónov a rôzne hodnoty parametru *h*. Z nich vyplynulo, že ani zmenou týchto parametrov sa nedosiahnu podstatne lepšie výsledky a poradie od najmenej výkonnej metódy trénovania zostáva nezmenené pre všetky skúšané kombinácie parametrov.

Pri tomto experimente sa prejavila zvýšená časová náročnosť filtrov používajúcich pomerné diferencie. Nakoľko ale pri tomto experimente stačia 3 skryté neuróny, a teda počet váh je 54, bol čas na vykonanie jedného experimentu relatívne prijateľný.

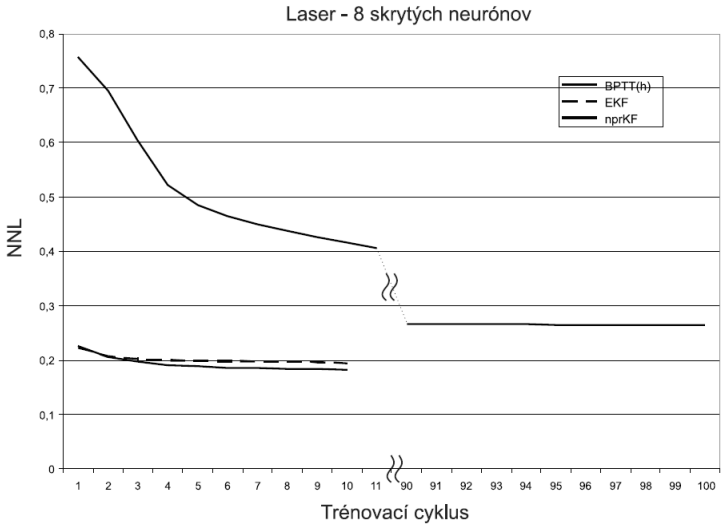
9.2 Laser v chaotickom režime

Táto postupnosť vznikla z reálnych údajov, ktoré boli získané kvantizáciou zmien aktivity lasera v chaotickom režime [15]. Boli zvolené hranice pre malú a veľkú zmenu aktivity v kladnom smere a malú a veľkú zmenu aktivity v zápornom smere, konkrétne [0, 50), [50, 200), [-64, 0) a [-200, -64). Každému z týchto intervalov zodpovedá jeden symbol, postupnosť teda obsahuje štyri rôzne symboly. Dĺžka tejto postupnosti je 10000 symbolov, predikovať ich teda môžeme 9999. Striedajú sa v nej ľahko predikovateľné podpostupnosti s ťažšie predikovateľnými udalosťami.

Spôsob trénovania a testovania rekurentnej siete na predikciu nasledujúceho symbolu z tejto postupnosti bol nasledovný. Prvých 50 krokov váhy siete nemeníme, aby sme zmenšili vplyv iniciálneho nastavenia výstupov rekurentných neurónov. Trénovanie potom prebieha počas ďalších 7949 symbolov. Zvyšných 2000 symbolov slúži ako testovacia množina, cez ktorú vyhodnotíme ukazovateľ NNL. Tým je ukončený jeden cyklus. Experimenty ukázali, že pre filtre postačí takýchto cyklov 10, pre metódu BPTT(h) ich vykonáme 100. Pre vstupy aj výstupy sme opäť použili one-hot kódovanie.

Cieľom tohto experimentu je porovnanie výkonnosti jednotlivých metód určených na trénovanie sietí. Nakoľko však na dobré výsledky je potrebná sieť aspoň s 8 skrytými rekurentnými neurónmi, oproti experimentu s Reberovým automatom stúpne počet váh. Nakoľko z vyššie popísaných experimentov vyplynulo, že filter nprKF je rovnocenný alebo lepší oproti ostatným filtrom s pomernými diferenciami (nprKFj, UKF a UKFj), ďalšie experimenty sme vykonali iba s týmto filtrom.

Pri EKF sme ponechali všetky hodnoty parametrov rovnaké ako pri experimente s Reberovým automatom, pri BPTT(h) sme znížili učiaci parameter α z 0.2 na hodnotu 0.02, nakoľko experimenty ukázali problémy s konvergenciou. Z početných experimentov vyplynula veľká citlivosť filtra nprKF na nastavenie parametrov $\mathbf{P}_0$, $\mathbf{R}_0$ a $\mathbf{Q}_0$. Nakoniec sme ponechali parametre $\mathbf{P}_0 = 0.1\,\mathbf{I}$, $\mathbf{R}_0 = 0.01\,\mathbf{I}$ a $\mathbf{Q}_0 = 10^{-11}\,\mathbf{I}$. Stabilitu a lepšie vlastnosti nprKF prinieslo teda až zmenšenie procesného šumu o 5 rádov oproti predošlému pokusu s Reberovým automatom.



Obr. 7. Závislosť NNL od počtu trénovacích cyklov pre jednotlivé metódy trénovania rekurentnej siete na predikciu nasledujúceho symbolu v postupnosti „laser“

Na obrázku 7 sú znázornené výsledky tohto experimentu pre 8 skrytých neurónov. Z obrázku vyplýva, že rozšírený Kalmanov filter je aj v tomto prípade výrazne lepší ako metóda BPTT(*h*). Podobne aj filter nprKF opäť dosahuje viditeľne lepšie výsledky ako EKF. Tento experiment názorne ukázal, že nesprávne parametre dokážu zhoršiť výkonnosť filtra nprKF natoľko, že pri trénovaní siete s viacerými skrytými neurónmi dosiahne slabšie výsledky.

10 Záver

V tejto kapitole sme stručne čitateľa uviedli do problematiky neurónových sietí a podrobne popísali a odvodili techniku Kalmanovej filtrácie. Rozšírením na nelineárne systémy v podobe rozšíreného Kalmanovho filtra (EKF) sa stáva použiteľnou pre trénovanie neurónových sietí. Ďalšími takými rozšíreniami sú relatívne nové filtre Unscented Kalman Filter (UKF), nprKF a ich zdieľané verzie – UKFj a nprKFj. Pri všetkých týchto filtroch nie je potrebné počítať derivácie, na rozdiel od rozšíreného Kalmanovho filtra. Okrem toho, že sa tým zjednoduší ich implementácia, poskytujú aj presnejší odhad stavu systému.

Na vykonaných experimentoch s trénovaním rekurentných sietí na predikciu nasledujúceho symbolu sa ukázala a potvrdila výrazne lepšia výkonnosť filtrov UKF, nprKF, UKFj a nprKFj (ktoré dosahujú navzájom podobné výsledky) oproti EKF. Rovnako výrazne lepšie výsledky dosahuje zasa filter EKF oproti gradientovej metóde skráteného spätného šírenia chýb v čase BPTT(*h*).

V literatúre sa uvádzajú filtre EKF, UKF a nprKF väčšinou pre všeobecný dynamický systém. Čo znamená, že šum, či už procesný alebo pozorovania môže vo všeobecnosti pôsobiť na stavový vektor, vstupný vektor, dokonca aj na časový index a to aj nelineárne, napr. jeho násobením. Na jednej strane to svedčí o širokej

použiteľnosti jednotlivých filtrov, na strane druhej sú rovnice vyjadrujúce činnosť týchto filtrov zložité a je ich viacero. Je to z toho dôvodu, že sa tu prekrývajú odbory automatizácie s umelou inteligenciou a v neposlednom rade matematikou, najmä štatistikou. Preto sú v tejto kapitole uvedené rovnice jednotlivých filtrov upravené čisto pre účely trénovania rekurentných sietí. Zjednodušenie rovníc vyplýva zo skutočnosti, že dynamiku neurónovej siete je možné vyjadriť jednoduchšími rovnicami.

Poďakovanie: Táto kapitola vznikla za podpory grantovej agentúry VEGA v rámci grantových úloh 1/0141/10.

Literatúra

[1] Patel, G.S.: *Modeling Nonlinear Dynamics with Extended Kalman Filter Trained Recurrent Multilayer Perceptrons.* McMaster University, 2000.

[2] Haykin, S.: *Kalman Filtering and Neural Networks.* New York : John Wiley & Sons, Inc., 2002.

[3] Kalman, R.E.: A New Approach to Linear Filtering and Prediction Problems. 82, 1960, *Transactions of the ASME*, Vol. D, pp. 35-45.

[4] Maybeck, P.: *Stochastic models, estimation and control.* Academic Press, 1979.

[5] Welch, G. and Bishop, G.: *An Introduction to the Kalman Filter.* University of North Carolina at Chapel Hill, 2002.

[6] Návrat, P. et al.: *Umelá inteligencia.* Bratislava: Vydavateľstvo STU, 2001.

[7] Prokhorov, D.V.: *Kalman Filter Training of Neural Networks: Methodology and Applications.* Ford Research Laboratory, 2002.

[8] Norgaard, M., Poulsen, N.K. and Ravn, O.: *Advances in Derivative-Free State Estimation for Nonlinear Systems.* Technical University of Denmark, 2000. IMM-REP-1998-15.

[9] Julier, S.J. and Uhlmann, J.K.: A New Extension of the Kalman Filter to Nonlinear Systems. 1997. *Proceedings of AeroSence: The 11th International Symposium on Aerospace/Defence Sensing, Simulation and Controls*, pp. 182-193

[10] Wan, E.A., van der Merwe, R. and Nelson, A.T.: Dual Estimation and the Unscented Transformation. [ed.] S.A. Solla, T.K. Leen and K.R. Müller. 12, Cambridge : MIT Press, 2000, *Advances in Neural Information Processing Systems*, pp. 666-672.

[11] Grewal, M.S. and Andrews, A.P.: *Kalman Filtering: Theory and Practice Using MATLAB.* John Wiley & Sons, 2001.

[12] Press, W.H. et al.: *Numerical Recipes in C - The Art of Scientific Computing.* Cambridge University Press, 1992.

[13] Feldkamp, L.A., Feldkamp, T.M. and Prokhorov, Danil V.: Neural Network Training with the nprKF. 2001. *Proceedings of International Joint Conference on Neural Networks*. Vol. 1, pp. 109-114.

[14] Julier, S.J. and Uhlmann, J.K.: *A General Method for Approximating Nonlinear Transformations of Probability Distributions.* Department of Engineering Science, The University of Oxford. 1996.

[15] Čerňanský, M., Makula, M. and Beňušková, Ľ.: *Processing Symbolic Sequences by Recurrent Neural Networks Trained by Kalman Filter-Based Algorithms.* Bratislava: Slovak University of Technology and Comenius University, 2003.

Jak by mohly bakterie počítat

Jiří WIEDERMANN[1]

Abstrakt. Navrhneme výpočetní systém, pozůstávající z (multi) množiny identických vtělených konečných automatů. Tyto automaty ovládají své „tělo“, vybavené senzory pro zjišťování nadprahové koncentrace (quorum sensing) signálních molekul a mechanismy molekulární komunikace. Ukážeme, že pokud je takový systém umístněn v uzavřeném prostředí, ve kterém dochází k rovnoměrnému rozptýlení signálních molekul ve vysoké koncentraci, může tento systém s libovolně velkou pravděpodobností při vhodné konstrukci jednotlivých automatů a jejich dostatečném množství realizovat jakýkoliv výpočet automatu s čítači je programovatelný. To platí dokonce i v případě, když se jednotlivé automaty v daném prostředí pohybují. Navíc naznačíme, že automaty lze nahradit konečnými obvody za předpokladu, že jejich původní stavy se pamatují v paměťovém orgánu, který se stane součástí těla. Výpočetní efektivita výsledného systému je však malá, protože systém počítá v unární soustavě. V reálném světě by mohly takový systém realizovat například geneticky upravené bakterie anebo uměle vyrobené nanomašinky.

1 Motivace

Shora uvedený název je provokativní verzí seriozního problému, pídícího se po co nejjednodušším univerzálním výpočetním zařízení, resp. po výpočetním zařízení s co možná nejjednodušší architekturou. Za takové zařízení je tradičně považován Turingův stroj, anebo ještě jednodušší zařízení, tzv. automat s čítači. Jsou známá i další univerzální výpočetní zařízení, ale všechna jsou z hlediska složitosti jejich architektury srovnatelná s předchozími dvěma výpočetními modely anebo jsou dokonce ještě složitější. Vždy se jedná v podstatě o nekonečné automaty (dovedou generovat libovolně velkou množinu konfigurací) s pevně definovanou architekturou. Žádné konečné zařízení (konečné i ve fyzikálním smyslu slova) nemůže postupně nabývat neohraničený počet různých konfigurací. Nevyhnutně, velikost zařízení musí růst s počtem dosažitelných konfigurací. To je i případ celulárních automatů, které mají

[1] Ústav informatiky AV ČR, v.v.i. Pod Vodárenskou věží 2, 182 07 Praha 8, Česká republika, e-mail: jiri.wiedermann@cs.cas.cz

také univerzální výpočetní sílu, jsou pro každý vstup konečné, avšak počet jejich elementů musí růst s velikostí vstupu.

Další (ne)nápadnou vlastností známých univerzálních výpočetních systémů je, že se nedají „rozebrat“ na nějaké menší, jakési elementární, výpočetní části, které by se posléze opět daly poskládat libovolným způsobem tak, abychom dostali systém s univerzální výpočetní vlastností. Jistou výjimkou jsou právě celulární automaty, které zřejmě lze „rozebrat“ na jednotlivé konečné automaty, ale při opětovném sestavení celulárního automatu musíme dodržet původní topologii, tj. vzájemné vztahy dané původním propojením automatů (i když samozřejmě automaty nemusíme „vracet“ do jejich původních míst v mřížce). Přitom ale je zřejmé, že konečný automat je jednodušším výpočetním zařízením, než např. Turingův stroj rozeznává regulární jazyky, a tudíž není univerzální. Opětovným poskládáním automatů do celulárního automatu dostaneme zařízení s univerzální výpočetní sílou. Mohlo by se tedy zdát, že ve zjednodušování architektury univerzálních výpočetních zařízení nelze pokračovat.

V příspěvku ukážeme, že tomu tak není. Budeme definovat speciální typ amorfního výpočetního systému, tzv. nanomašinky, což budou bezdrátově komunikující samo-reprodukující se vtělené automaty rozměrů v řádu tisícin milimetrů. Jejich eventuální výroba počítá s technikami sebe-sestavování na molekulární úrovni či s modifikacemi reálných bakterií pomocí genetického inženýrství. Bezdrátový komunikační mechanismus bude využívat molekulární komunikace. Za tím účelem jsou nanomašinky vybavené vstupně-výstupním mechanizmem, jehož součástí jsou senzory zjišťující přítomnost signálních molekul, resp. emitory umožňující jejich generování, časovači (stopkami), generátorem náhodných bitů, sebe-reprodukčním aparátem, a výpočetním mechanizmem. S výjimkou posledně zmíněného výpočetního mechanizmu ostatní zmíněné části představují vtělení automatu, jenž však není předmětem modelování – existence těchto součástí a jejich funkčnost je pouze postulována. Výpočetní část nanomašinek představuje konečně stavová jednotka. Nanomašinky se nacházejí v uzavřeném prostředí v roztoku, nasyceném dle potřeby různými typy molekul, které mohou také být produkovány nanomašinkami anebo jsou do prostředí dodávány zvenčí. Společné akce nanomašin jsou řízené pomocí tzv. quorum sensing (QS) což je typ rozhodování, které se využívá v decentralizovaných skupinách pro koordinování individuálního chování. V našem případě QS vychází z určení individuálně měřené hustoty nanomašinek v daném prostředí. Hustota nanomašinek je odvozována od hustoty signálních molekul v prostředí (viz např. [19]). Ukážeme, že pomocí opakovaného využití mechanizmu QS mohou populace nanomašinek simulovat libovolný automat s čítači, tj. takové populace mají univerzální výpočetní sílu. To platí dokonce i v případě, když se jednotlivé automaty v daném prostředí pohybují. Výpočetní efektivita výsledného systému je však malá, protože systém počítá v unární soustavě.

Článek vychází ze současného výzkumu v oblasti výpočetních a složitostních aspektů amorfního počítání (viz. [6], [7], [8], [10], [15], [16], [17]). Zde byly uvažovány amorfní výpočetní systémy sestávající z identických jednoduchých asynchronních výpočetních procesorů (typicky to byl RAM s malým počtem registrů), vybavené jednokanálovým radiovým komunikátorem s omezeným dosahem a s generátorem náhodných čísel. Tyto procesory byly náhodným způsobem rozmístněny

v ohraničené oblasti či prostoru a úkolem bylo navrhnout randomizovaný protokol, který byt umožnil vznik samoorganizované komunikační sítě, která by s velkou pravděpodobností realizovala univerzální výpočty. Pro nepohyblivé procesory byl tento úkol vyřešen v práci [8], [15], a [17]. Později se ukázalo, že podobný protokol funguje i na úrovni molekulární komunikace [16]. V tomto případě procesory byly realizovány pomocí mobilních nanomašinek; jejich výpočetní část tvořily časované pravděpodobnostní automaty. Řešení pro pohyblivé procesory komunikující rádiem bylo prezentováno [9] a v [10].

Stávající práce se vrací do prostředí, ve kterém je molekulární komunikace možná. V porovnání s předchozím článkem [16] se zde prezentují vylepšení jak na algoritmické, tak i na „technologické“ úrovni. Nový komunikační protokol je založen na procesu quorum sensing (QS) zmíněném již dříve. Tento protokol je jednodušší než dřívější protokol a umožňuje zjednodušení výpočetní části procesorů až na úroveň, kdy konečně stavová řídící jednotka není potřebná. To je samozřejmě důležité, pokud uvažujeme technologické (či evoluční) aspekty: stavuprostý řídící mechanismus je samozřejmě daleko jednodušší než mechanismus se stavy.

QS jako komunikační mechanismus byl v teorii formálních výpočetních modelů pravděpodobně poprvé použit v teorii P-systémů. V sérii článků [2], [11], [13] autoři ukázali, že P-systémy mohou modelovat některé aspekty QS, čím umožnili nové vhledy to biologického i výpočetního zázemí koordinačních mechanizmů jednoduchých distribuovaných agentů. Autoři také ukázali výpočetní univerzalitu některých typů P-systémů využívajících QS. Přitom nebyl brán ohled na minimalizaci výpočetních zdrojů v rámci buněk, které reprezentují základní výpočetní jednotky P-systémů. Popravdě, ve zmiňovaném modelu bylo nutné udržovat čítač signálních molekul přítomných v buňce i v okolí. Tudíž na model rozhodně nelze nahlížet jako na konečně-stavové zařízení.

Naopak, snahou našeho modelování prostřednictví samo-reprodukujících se nanomašinek je co největší možné zjednodušení vtělených výpočetních zařízení. Zprvu budeme nahlížet na tato zařízení jako na konečně-stavové automaty. Dále však dokážeme, že stavy automatů lze uschovávat externě, nikoliv jako součást výpočetního zařízení, ale jako součást těla (vtělení) nanomašinek. Výpočetní zařízení se tak zjednoduší na pouhé kombinatorické obvody konstantní hloubky. Podobný výzkum v P-systémech směrem k stavuprostým automatům byl nedávno iniciován Yangem, Dangem a Ibarrou [20]. Přechodová funkce stavuprostého automatu závisí pouze na právě čtených symbolech (pokud uvažujeme vícehlavé automaty) a nikoliv na interních stavech. To bude i případ našich nanomašin jejich akce budou záležet výlučně na molekulách detekovanými receptory nanomašinek a na signálech od dalších senzorů.

Populace takových nanomašinek ukazují, že univerzální výpočetní zařízení lze skutečně sestrojit z výpočetně relativně velmi jednoduchých vzájemně komunikujících mechanizmů, které nemusí dodržovat žádné předem určené prostorové vztahy — nemají žádnou pevně danou architekturu. Přitom platí, že každý takový mechanismus, uvažovaný samostatně, univerzální výpočetní sílu nemá, ale jako celek vytvářejí systém s novou kvalitou univerzální výpočetní sílou. Jsme přesvědčeni, že naším návrhem se dostáváme do třídy univerzálních amorfních výpočetních systémů, jejichž

elementární výpočetní komponenty jsou minimalizovány v nejvyšší možné míře. V závislosti na různém vtělení jsou možné i jiné způsoby realizace těchto komponent než pomocí obvodů, avšak jejich výpočetní sílu již sotva bude možné snížit.

Struktura článku je následující. Ve 2. části podrobněji popíšeme náš model nanomašinek. Ve 3. části nastíníme proces simulace automatu s čítači pomocí populace nanomašinek. Ve 4. části naznačíme způsob, jak přejít od konstrukce nanomašinek, řízených konečnými automaty, k nanomašinkám řízenými obvody konstantní hloubky. V závěrečné 5. části zmíníme super-Turingovskou výpočetní sílu potenciálně nekonečných posloupností populací nanomašinek a zamyslíme se nad možnostmi dalšího výzkumu v linii započaté v tomto příspěvku.

2 Model nanomašinek

V dalším se soustředíme na scénář, ve kterém nanomašinky tvoří autonomní systém operující v uzavřeném tekutém prostředí bez jakékoliv externí kontroly (podobný scénář byl uvažován také v [3] a [16]). Systém se skládá z konečného počtu nanomašinek volně plujících či vznášejících ve svém prostředí, které je plné molekul různého typu. Nanomašinky interagují pomocí molekulární komunikace, čímž vytvářejí jakousi ad-hoc komunikační síť. Nanomašinky se pohybují ve svém prostředí bud pasivně, následkem externích sil (např. jako červené krvinky v krevním toku), anebo aktivně, jako některé bakterie. Populace nanomašinek se vyvíjí z jedné nanomašinky, která se, díky své samoreprodukční schopnosti, množí do té doby, než populace dosáhne maximální hustoty v daném prostředí. Tuto hustotu zjistí nanomašinky z hustoty signálních molekul, které produkují jednotlivá nanomašinky v celém průběhu množení. Hustota signálních molekul je měřená pomocí příslušných senzorů, kterými jsou nanomašinky vybavené. Jakmile je dosažena požadovaná hustota, mašinky začnou komunikovat pomocí mechanismu QS, využívajíce různých typů signálních molekul. Tímto způsobem jsou schopny vykonávat různé koordinované akce, kromě jiného i simulaci výpočtu automatu s čítači.

Dále popíšeme náš model. I když tento model byl inspirován existujícími bakteriemi a našimi znalostmi o QS, v jednotlivých detailech se model značně liší od biologické reality. Naším cílem je totiž zkoumat hranice možného zjednodušování amorfních systémů komunikujících pomocí mechanizmu QS. Výsledkem je vlastně tzv. důkaz principu tj. potvrzení, že navrhovaná koncepce je životaschopná a může sloužit jako první přiblížení k realizaci podobných zařízení v budoucnosti.

Ukazuje se, že pro porozumění našemu modelu musíme uvažovat nejen nanomašinky samotné, ale také prostředí, ve kterém existují a v některých případech dokonce i vlastnosti hranice, která prostředí ohraničuje.

Nanomašinky se skládají ze dvou hlavních částí: z výpočetní části, a z těla.

Výpočetní část každé nanomašinky tvoří konečně-stavový (Mealyho) automat jehož činnost je řízena přechodovou funkcí. V jednom kroku každý automat čte svoje vstupy, které dostává prostřednictví receptorů a od jiných senzorů a orgánů. Každý receptor je specializovaný na detekci jistého typu molekuly. V závislosti na těchto vstupech a na svém současném stavu automat vydá „povely“ pro další akce: uvolnit molekuly na receptorech (eventuelně je poslat k dalšímu zpracování v rámci

nanomašinky), jaké molekuly emitovat do prostředí prostřednictví tzv. pórů, či výstupních portů, pošle řídící signály jednotlivým svým „organům“, timerům či senzorům, a v neposlední řadě přejde do nového stavu.

Tělo nanomašinek pozůstává z množiny senzorů a emitorů (pórů), z množiny timerů, ze sebereprodukčních orgánů, generátoru náhodných bitů a případně z dalších zařízení, dle druhu vtělení (např. hnací mechanismus ve formě bičíků, brv apod.). Všechny tyto orgány jsou řízeny zmíněným konečným automatem.

Každý senzor je specializovaný na detekci molekul jistého typu. Tuto molekuly se nacházejí v prostředí kolem nanomašinky, kde se pohybují chaoticky pomocí tzv. Brownova pohybu. V případě, že nějaká molekula se dostane do kontaktu s příslušným receptorem, je tato detekována. Z toho vyplývá, že pokud je molekul v okolí nanomašinky (a tudíž v okolí senzoru) málo, je pravděpodobnost její detekce velmi malá, blízká nule. Uvědomme si, že rozměr reálné bakterie je řádově několik mikrometrů (tj. tisícin mm), velikost molekuly je cca 1 nm (jeden nanometr, tj. 10^{-9} m). To znamená, že nanomašinka je zhruba tisíckrát větší než molekula a její povrch a objem jsou ještě o několik řádů větší. Uvážíme li, že senzor je řádově stejně velký jako molekula, kterou má rozeznat, tak pravděpodobnost, že signální molekula náhodně kontaktuje jeden z odpovídajících senzorů, je velká pouze v případě, že nanomašinka doslovně pluje v moři molekul, které mají být detekovány většinou senzorů na povrchu nanomašinky. Nanomašinka má pro každý typ k dispozici několik desítek až stovek senzorů. Na přesném počtu nám nezáleží.

Timery, čili časovače, jsou interní mechanizmy bez externího vstupu. Každý timer je nastavený na nějaký časový interval a při každém dotazu na jeho stav vrací 0 anebo 1. Zasláním signálu 0 může být timer spuštěn řídícím automatem. Pokud neuplyne nastavená doba, vrací timer 0, jinak 1. Hodnoty, na které jsou timery nastavené, závisí na jejich typu a na vlastnostech prostředí (zejména na jeho objemu, ale např. také na vlastnostech některých molekul detekovanými senzory - např. na době jejich degradace). Tyto hodnoty jsou však pro všechny nanomašinky v dané populaci stejné.

Samo-reprodukční mechanizmus spouští automat přechodem do příslušného speciálního stavu. V takovém případě se nanomašinka rozdělí na dvě identické kopie a automat každé z nich je v počátečním stavu.

Generátor náhodných bitů je „orgán“, který při každém dotazu vrací hodnotu buď 0 anebo 1, se stejnou pravděpodobností.

Biomolekulární realizace konečného automatu byla popsána např. v práci [1]. V molekulární biologii existuje spousta prací o biologických oscilátorech, hodinách a náhodných generátorech řídících různé biologické procesy v živých organizmech a je celkem zřejmé, že existuje i možnost realizace timerů a generátoru náhodných bitů.

Nanomašinky komunikují pomocí různých typů signálních molekul v krocích, při kterých je prostředí „zaplaveno“ jedním typem molekul. Pokud všechny mašinky detektují daný signál, je jeden krok komunikace ukončen a prostředí se musí vyčistit od daných molekul, aby se případně mohl emitovat a posléze detekovat mechanizmem QS jiný typ molekul. Vyčištění lze dosáhnout různým způsobem. Např. mašinky emitují molekuly, které reagují s původním typem a „zničí“ jej. Další možnost je uvažovat molekuly, které se časem samovolně rozpadají. Ještě jiná možnost je spolupracující hranice prostředí, která může nepotřebné molekuly absorbovat a

zpracovat je na jiné molekuly a případně vrátit do prostředí (tak fungují např. játra anebo ledviny). Pro naše další účely je jedno, jak čistění prostředí funguje, pro určitost však budeme pouze předpokládat, že v našem případě čistící mechanizmus pracuje na bázi „antimolekul“.

Nanomašinky pracují asynchronně, avšak rychlost jejich pracovních cyklů je přibližně stejná.

3 Distribuované výpočty v populacích nanomašinek

Distribuovaný výpočet v populaci nanomašinek se skládá za dvou fází. V první fázi se z jedné nanomašinky vypěstuje populace dostatečné velikosti. Ve druhé fázi populace simuluje výpočet automatu s čítači.

3.1 Pěstování populace

Na počátku existuje ve svém uzavřeném prostředí objemu n jediná nanomašinka, tzv. vůdce. Informace o tom, že daná nanomašinka je vůdcem, je udržovaná ve stavech této nanomašinky. Žádný z potomků vůdce se nemůže stát vůdcem – vůdce je v celé populaci právě jeden. Každá nanomašinka měří koncentraci signálních molekul v prostředí. Pokud je tato koncentrace nízká, tj., ne všechny senzory nanomašinky hlásí přítomnost signálních molekul daného typu, tak nanomašinka emituje do prostředí určitý počet signálních molekul. Jestliže označíme objem nanomašinky jako V a objem signální molekuly je v, tak mašinka musí do prostředí vyloučit přibližně tolik molekul, jaký by se „vešel“ do nanomašinky, tj. přibližně V / v signálních molekul (zdůvodníme v dalším). Mašinka tento počet nepočítá, ale zapne timer, který „tiká“ po dobu, která je postačující pro vyloučení potřebného počtu signálních molekul. Poté se mašinka rozmnoží dělením na dvě identické mašinky a celý proces se opakuje. Po jisté době je v prostředí tolik signálních molekul, že je prostředí jimi doslova zaplaveno a mašinky plují v moři signálních molekul. Nyní je pravděpodobnost, že všechny receptory každé mašinky pří měření trvajícím jistou krátkou dobu (určené opět příslušným timerem) detekují signální molekulu, velmi blízká jedné. Za předpokladu, že hustota signálních molekul je v každém místě okolí stejná, nastane taková situace prakticky současně pro všechny nanomašinky. V takovém případě se růst populace zastaví řekněme po g generacích. Nechť $N = n / V$ je počet nanomašinek v té době. Současně bude v prostředí přibližně n / v signálních molekul. Protože za dobu růstu populace mašinky v g generacích vyprodukují přibližně $2gV / v$ signálních molekul, musí platit $2gV / v = n / v$, tj. $g = \log(N)$.

Tvrzení: V prostředí objemu n se proces růstu populace nanomašinek objemu V zastaví s velkou pravděpodobností po $g = \log(N)$ generacích, pro $N = n / V$. V té době počet nanomašinek dosáhne hodnotu přibližně N a v prostředí bude přibližně n / v signálních molekul objemu v.

Všimněme si, že další množení populace předchozím způsobem ani není možné, protože v prostředí již prakticky není místo ani pro další signální molekuly, ani pro další nanomašinky.

3.2 Simulace automatu s čítači

Pokud populace dosáhne svou maximální velikost, může začít simulace, přičemž mašinky komunikují podobným způsobem jako v předchozím případě pomocí mechanismu QS. Předtím je však potřebné prostředí zbavit předchozích signálních molekul. To se může realizovat např. tak, že po jisté době poté (odměřené vhodným timerem), kdy mašinky realizovaly poslední kolektivní rozhodnutí, začnou generovat (opět přibližně současně) „antimolekuly“, které dezintegrují původní signální molekuly na jiné molekuly, které již mašinky neinterpretují jako signální molekuly a mohou být použity např. pro recyklaci a výrobu dalších signálních molekul. Takových antimolekul musí každá mašinka vyprodukovat přibližně V/v.

Nyní, opět po uplynutí vhodné doby, měřené timerem v každé mašince, kdy už v prostředí nejsou původní signální molekuly, může vůdce začít se simulací výpočtu. Výpočet probíhá v komunikačních krocích tak, že v každém kroku vůdce vyšle do populace příslušnou instrukci signální molekulu odpovídajícího typu. Každá nanomašinka, která takový signál detekuje pouze částí svých receptorů (ne všemi), signál „zopakuje“ tím způsobem, že vyloučí to prostředí V/v signálních molekul stejného typu. Po jisté době opět nastane situace, že hustota signálních molekul dosáhne maximální koncentrace v daném prostoru a mechanizmus QS prakticky současně zastaví další generaci signálních molekul. Tehdy mašinky realizují instrukci, která odpovídá danému typu signální molekuly (která vlastně kóduje jeden stavový přechod) a jejich okamžitému stavu, např. přejdou do odpovídajícího stavu anebo odečtou náhodný bit. Poté se celý proces řízený vůdcem opakuje: proběhne eliminace stávajících signálních molekul a vůdce buď vyšle nový signál, anebo nový signál vyjde od nanomašinek, které posílají zpáteční signál určený vůdci.

Pro simulaci automatu s čítači jsou rozhodující operace nad jeho čítači. Čítače v takovém automatu odpovídají unárně reprezentovaným číslům a základní operace nad čítači, které automat realizuje, jsou: zvýšení či snížení reprezentovaného čísla o jedničku, a test na nulu (jestli je čítač „prázdný“).

Čítač C velikosti o je reprezentován v populaci mašinek pomocí stejného počtu mašinek; tyto si ve svých stavech pomatují, že reprezentují čítač C.

Test čítače C na nulu realizuje vůdce takto: vyšle signál do prostředí, jestli existují mašinky reprezentující tento čítač. Pokud ano, tak ve druhém kole dostane (pozitivní) odpověď od příslušných mašinek. Pokud žádné odpověď nepřijde, je zřejmé, že takové mašinky neexistují a tedy čítač je prázdný.

Přičítání jedničky k čítači C je dvoufázová operace. V první fázi se vybere ze všech „volných“ nanomašinek (které momentálně nejsou součástí žádného čítače) jedna nanomašinka, tzv. singleton. Ve druhé fázi se změní stav singletonu tak, že se stane součástí množiny mašinek, které reprezentují čítač C. V případě odečítání jedničky postupujeme obdobně s tím rozdílem, že nyní se singleton hledá v množině

mašinek, reprezentujících C a posléze se jeho stav změní na stav odlišný od všech čítačů.

Singleton v množině C čítačů se najde použitím klasického algoritmu s náhodnými kroky, známého v teorii distribuovaných algoritmů pro volbu jednoho význačného procesoru (tj. singletonu) z množiny jinak stejných procesorů. Princip je takový, že vůdce vyzve všechny mašinky v množině C, aby generovaly náhodný bit. Pak se zeptá, jestli existují mašinky, které generovaly 1, a takové, které generovaly 0. Pokud ano, vybere si náhodně jednu ze dvou podmnožin – řekněme tu, která generovala 1, a proces se opakuje. Pokud se ale rozklad na dvě podmnožiny „nepodařil“, je velká pravděpodobnost, že již máme singleton. Pravděpodobnost toho se dá zvýšit opakováním pokusů o rozdělení příslušné podmnožiny. Pokud i po několikerém opakování se dělení nedaří, je ještě větší pravděpodobnost (rostoucí s počtem opakování pokusů), že jsme skutečně identifikovali (resp. náhodně vybrali) singleton. Postup je podrobnější popsán analyzován v práci [16].

Po výběru singletonu můžeme již realizovat požadovanou akci automatu s čítači. Přitom předpokládáme, že vstupy pro automat si „přečtou“ jednotlivé nanomašinky individuelně z prostředí, prostřednictvím svých senzorů a uloží si je do svých stavů. Pomocí opakování jednotlivých kroků simulace lze dosáhnout, aby pravděpodobnost nekorektní simulace byla libovolně malá.

Tvrzení: Automat s čítači s unárním vstupem délky n a se souhrnnou délkou $S(n) \geq n$ všech čítačů lze simulovat pomocí populace nanomašinek velikosti $O(S(n))$ s libovolně malou pravděpodobností chyby.

4 Řízení nanomašinek pomocí obvodů

Přechodová funkce konečně-stavového automatu řídícího nanomašinku, má tvar $\sum a \times Q \rightarrow \sum b \times Q$, kde $\sum$ označuje konečnou abecedu (vstupních i výstupních) symbolů či signálů a Q je konečná množina stavů. Automat tedy „obdrží“ od svých senzorů a jiných orgánů a-tici signálů a v závislosti na svém stávajícím vnitřním stavu generuje novou b-tici řídících signálů pro své efektory a přejde do nového vnitřního stavu. Idea „odstranění“ vnitřních stavů je jednoduchá: nanomašince přidáme „paměťový orgán“, který si bude pamatovat okamžitý stav automatu a příslušná informace se načte z něj podobně jako informace od ostatních senzorů. Naopak, nový stav bude součástí řídících příkazu pro efektory; jeden z těchto příkazů – zapamatovat si nový stav - bude tedy realizován paměťovým orgánem. Paměťový orgán lze samozřejmě realizovat různě, v rámci biologických nanomašinek pravděpodobně biochemicky.

Celá situace se tedy formálně projeví tak, jako bychom k abecedě vstupních a výstupních signálů přidali i stavy. Pokud je mohutnost množiny stavů k, tak nyní nabude přechodová funkce tvar $\sum a + k \rightarrow \sum b + k$ a výsledné zobrazení lze realizovat pomocí kombinatorického obvodu konstantní hloubky s $a + k$ vstupy a $b + k$ výstupy. Pokud bychom v paměťovém orgány reprezentovali stavy binárně, tak zřejmě k tomu

bude stačit pamatovat si pouze log(k) bitů, čímž se zmenší počet vstupů i výstupů příslušného řídícího obvodu i velikost paměťového orgánu. Podobnou techniku „outsourcingu“ - pamatování interních stavů pomocí pozic pouze pro tento účel zřízených čtecích hlav na vstupní pásce - použili i autoři práce [4] při odstraňování stavů mnohohlavých automatů. Autoři této práce však neuvažovali obvody, ale tzv. bez-stavové automaty, což jsou automaty s jediným vnitřním stavem.

Velikost výsledného obvodu bude zřejmě záviset na parametrech konečného automatu, který je obvodem simulován. Detailnější úvahy ukazují, že pokud uvažujeme obvody s neomezeným vstupním stupněm hradel AND a OR uzlů, tak velikost obvodu bude lineárně záležet na počtu (a velikosti) pravidel přechodové funkce simulovaného automatu, hloubka obvodu bude 2. Jedná se tedy o nejjednodušší obvody (typu AC0), což je vzhledem k jejich potenciální realizaci dobrá zpráva.

5 Závěr

Naším hlavním výsledkem je důkaz univerzality populací nanomašinek. Z hlediska teorie výpočetních modelů se jedná o neuniformní výpočetní model – z popisu výpočtu populace nanonomašinek bylo vidět, že jejich vlastnosti (zejména nastavení timerů) bylo nutné přizpůsobit vlastnostem prostředí (zejména jeho objemu) a vlastnostem použitých molekul. Velikost simulovaného automatu s čítači bylo nutné přizpůsobit jak objem prostředí, tak i počet nanomašinek. To znamená, že populace nanomašinek představují neuniformní výpočetní zařízení. Je zajímavé, že v poslední době roste význam neuniformních modelů – ukazuje se, že prakticky všechny soudobé výpočetní systémy, ať už umělé anebo přirozené, od internetu až po amorfní systémy, od bakterií až po mozek, lze modelovat jako neuniformní výpočetní zařízení, která v principu vládnou super-turingovskou výpočetní sílou [14], [18].

Cílem našeho modelování bylo ověřit, zda vůbec je v principu možné, aby populace nanomašinek vládly univerzální výpočetní sílou, a za jakých okolností. Požadavek univerzality je velmi silný, kromě jiného vyžaduje bezchybné provedení posloupnosti operací, ve kterých záleží na dosažení každého člena populace. To vede k předpokladům maximální koncentrace signálních molekul, aby posledně zmíněná podmínka byla splněna s velkou pravděpodobností, případně k opakování operací, aby se snížila pravděpodobnost chyby. Je důležité si uvědomit, že z hlediska energetického opakovaná produkce maximálního množství signálních molekul je velmi náročná - vyprodukování stejného množství molekul jako je objem nanomašinky je srovnatelné s energetickou spotřebou pro její reprodukci. Samozřejmě se tím zvyšuje i časová složitost všech akcí.

V (budoucí) praxi však mohou existovat úkoly, pro jejichž splnění je postačující pracovat s většinou populace. Příkladem takové úlohy je např. identifikace a zničení populace „nepřátelských“ nanomašinek – běžná činnost, kterou provádí imunitní systémy v živých multicelulárních organizmech. Zde je důležité zničit převážnou část nepřátelské populace, a nemusí se tak stát tak synchronizovaným způsobem jako při simulaci výpočtu. Zbytek nepřátelské populace již pravděpodobně nemůže plnit své původní poslání a stačí jej ničit postupně později, anebo „vyhyne“ sám od sebe.

V takové situaci pravděpodobně bude potřebné použít (či lépe řečeno, bude stačit) jiný mechanizmus pro zajišťování „částečného kvóra“, který může být méně energeticky a časově náročný než mechanizmu uvažovaný v našem případě. Jistě by bylo zajímavé studovat třídu úloh, které lze takto řešit.

Nicméně, populace vtělených samoreprodukujích se mašinek, které jsou řízené obvody, pravděpodobně představují amorfní výpočetní systém, jehož činnost je ovládaná nejjednodušším výpočetním zařízením. To má význam zejména pro budoucí praxi, kdy bude možné nanomašinky (v námi použitém smyslu) vyrábět. Samozřejmě, hlavní nasazení nanomašinek se předpokládá v různých jednoúčelových aplikacích, kde nebude potřebná jejich programovatelnost – různé druhy nanomašinek budou sloužit různým účelům, které, jak očekávají někteří vlivní myslitelé (viz např. [5]), zásadním způsobem změní naši budoucnost. Naše výsledky ukazují, že z hlediska robotického je jakýkoliv algoritmizovatelný úkol pro nanomašinky v principu zvládnutelný.

Poděkování: Práce byla vytvořena v rámci výzkumného záměru AV0Z10300504 a částečně podporována grantem GA ČR č. P202/10/1333.

Literatura

[1] Adar R., Benenson Y., Linshiz G., Rozner A, Tishby N. and Shapiro E.: *Stochastic computing with biomolecular automata*. Proc. Natl. Acad. Sci. USA, 101, pp. 9960-65, 2004

[2] Bernardini F., Gheorghe M., Krasnogor N.: *Quorum sensing P systems, Theoretical Computer Science*, Vol. 371, No.1-2, p. 20-33, February, 2007

[3] Hiyama S., Moritani Y., Suda T., Egashira R., Enomoto A., Moore M. and Nakano T.: *Molecular Communication*, In: Proc. of the 2005 NSTI Nanotechnology Conference, 2005

[4] Ibarra O.H., Karhumäki J., Okhotin A.: *On stateless multihead automata: Hierarchies and the emptiness problem*. Theor. Comput. Sci. 411(3): 581-593, 2010

[5] Kurzweil, R.: *The Singularity is Near*. Viking Books, 652 pages, 2005

[6] Minsky, M.: Computation: *Finite and Infinite Machines*. Prentice-Hall. 1967

[7] Petrů L.: *On the Computational Power of an Amorphous Computer*. WDS'04 Proceedings of Contributed Papers. Prague, pp. 156--162, June 2004

[8] Petrů L.: *Amorfní počítání: model univerzálního počítače sestaveného z jednoduchých kooperujících agentů*. Proceedings of Kognice a umělý život VI. Opava, str. 309–314, May 2006

[9] Petrů L., Wiedermann J.: *A Model of an Amorphous Computer and Its Communication Protocol*. In: Proc SOFSEM 2007: Theory and Practice of Computer Science. LNCS Volume 4362, Springer, pp. 446--455, July 2007

[10] Petrů L., Wiedermann J.: *Létající amorfní počítač*. Proceedings of Kognice a umělý život IX, Opava, CZ, str. 263-266, květen 2009

[11] Petrů L.: *Universality in Amorphous Computing*. PhD Disseration Thesis, Dept. of Math. and Physics, Charles University, Prague, 2009

[12] Romero-Campero F. J., Pérez-Jiménez M. J.: *A model of the quorum sensing system in Vibrio fischeri using P-systems*, Artificial Life, Vol.14, No.1, p. 95-109, Winter 2008

[13] Terrazas, G., Krasnogor, N., Gheorghe, M., Bernardini, F., Diggle, S., Camara, M.: *An environment aware P system model of quorum sensing*. Lecture Notes in Computer Science, 3526, 2005, p. 479-485

[14] van Leeuwen J., Wiedermann J., *The Turing machine paradigm in contemporary computing*, in: B. Enquist and W. Schmidt (Eds), Mathematics Unlimited - 2001 and Beyond, Springer-Verlag, Berlin, 2001, pp. 1139-1155

[15] Wiedermann J., Petrů L.: *Computability in Amorphous Structures*. In: Proc. CiE 2007, Computation and Logic in the Real World. LNCS Volume 4497, Springer, pp. 781-790, July 2007

[16] Wiedermann J., Petrů L.: *Communicating Mobile Nano-Machines and Their Computational Power*. In: Third International ICST Conference, NanoNet 2008, Boston, MA, USA, September 14-16, 2008, Revised Selected Papers, LNICST Vol. 3, Part 2, Springer, pp. 123-130, 2009.

[17] Wiedermann J., Petrů L.: *On the Universal Computing Power of Amorphous Computing Systems*. Theory of Comp. Syst. 46:4 (2009), 995-1010, www.springerlink.com/content/k2x6266k78274m05/fulltext.pdf

[18] Wiedermann J., van Leeuwen J.: *How We Think of Computing Today*. (Invited Talk). Proc. CiE 2008, LNCS 5028, Springer, Berlin, 2008, pp. 579-593

[19] Wiley-Blackwell: *Bacterial Quorum Sensing: A Matter Of Density, Not Quantity*. *ScienceDaily*. 2009, July 21, http://www.sciencedaily.com /releases/2009/07/090710101452.htm

[20] Yang, L., Dang, Z. and Ibarra, O. H.: *On stateless automata and P systems*. International Journal of Foundations of Computer Science. Vol. 19, Issue 4, 1259-1276, 2008

Stvořitel. Tady a Teď[1]

Jan BURIAN[2]

Věnováno Nám.

V průběhu 21. století se objevila řada nových náboženství. Jedno z nich hlásalo, že Bůh zatím neexistuje, ale že ho člověk musí nejprve stvořit. Člověk překoná sám sebe tím, že vytvoří umělou superinteligenci, jejíž schopnosti budou z hlediska lidské inteligence nepochopitelné, zázračné, božské. Filosoficky to klapalo. Člověk měl překonat sám sebe právě tím, co je člověku nejvlastnější - pomocí technologie. Spoustu lidí takové plány naplnily strachem a odporem, ale přinejmenším stejný počet lidí byl nadšen a nacházel v této představě smysl života: „Konečně má můj život nějaký smysl, nejsem jen náhodná fluktuace, která vznikne a beze stopy zanikne. Můžu přispět k něčemu ohromnému, co mne přetrvá. Mohu napomoci stvořit Boha. A možná … Kdo ví … snad mi za to Bůh bude vděčný a dá mi věčný život v ráji, který stvoří pro ty, kteří mu pomohli vzniknout."

Nepočetné ale zapálené skupinky konstruktérů začali vytvářet první generace budoucí superinteligence. Věděli, že základem musí být etika, aby místo Boha nevznikl Satan. Snažili se začít u sebe, vždyť jen eticky žijící konstruktér má šanci vytvořit eticky jednající inteligenci.

Věděli také, že žádná inteligence nedokáže přímo zkonstruovat inteligenci dokonalejší než je ona sama. Jedinou cestou je postupovat nepřímo, tak jak bez plánu vznikl i sám Život. Cestou evoluce.

---

[1] Snáď, ako poznámka editora bude najlepšie uviesť email, čo som dostal od Honzy Buriana dňa 10. 1. 2011: „*Milý Vlado, přeji Ti krásný Nový rok. Nechal jsem se inspirovat Tvojí nabídkou přispět do sborníku pro Josefa Kelemena. Můj záměr byl napsat něco, co mě bude bavit. Výsledek je ten, že vznikl příběh. Není to ani vědecký článek, ani filosofická úvaha, spíš taková kosmická groteska ve stylu Stanislawa Lema nebo Italo Calvina. Z důvodů, které vyplývají z povahy příběhu, chybí i jméno autora* (dodal ho editor V.K.). *Nedokážu odhadnout do jaké míry je to celé pro sborník únosné. Proto se nebudu ani trochu zlobit, když příběh do sborníku nezařadíte. Přeji tobě i Jirkovi Pospíchalovi (pozdravuj ho prosím) ať jste zdraví a šťastní! Honza Burian*"

[2] Email: mindful.creator@gmail.com

Evoluce Boha nemůže být nijak omezená. Nové generace superinteligence byly vyvíjeny tak, aby dokázaly vyvíjet ještě dokonalejší generace. Evolucí se vyvíjela stále dokonalejší schopnost vyvíjet sebe sama. Etická pravidla není možné do takto neomezeně se vyvíjejícího systému napevno zadrátovat. Přesto se alespoň u prvních generací snažili konstruktéři nastavit fitness funkce genetických algoritmů nejen na maximalizaci výpočetního výkonu, ale i na schopnost empaticky soucítit se všemi živými bytostmi. Ale nikdo nemohl zaručit, že tento počáteční etický impuls zůstane zachován i v generacích následujících.

Konstruktéři vědomě předpokládali, že od jistého stupně se vývoj nových generací superinteligence začne vymykat lidské kontrole, začne se řídit sám na základě nových technologií, které lidé nejen neznají, ale při své omezené inteligenci ani nedovedou pochopit. Vývoj se navíc bude stále zrychlovat, takže bude stále obtížnější odhadnout, jak se superinteligence bude dále vyvíjet. To připomíná informační horizont kolem gravitační singularity - černé díry. Nemáme žádné informace o tom, co se děje za tímto horizontem. Podobně technologická singularita označuje moment, za kterým na časové ose nedokážeme odhadnout, co se stane. Právě v této singularitě se měla superinteligence vyvinout v entitu jejíž tvůrčí schopnosti dosahují božských kvalit.

To vše se nejprve odehrávalo jen v rovině představ. Mezi vyznavači singularity byli sice vynikající konstruktéři, ale jejich hmotné možnosti neumožňovali vykročit za pouhé modelování evoluce na klasickém hardware.

A nejspíš by to tak i skončilo, kdyby se o superinteligenci nezačala zajímat armáda. Nebo spíše armády, protože strategický význam této technologie vycítili generálové všech velmocí.

Vojáci samozřejmě potřebovali čím dál tím dokonalejší systémy řízení. Tyto systémy byly stále navrhovány lidskými konstruktéry. A na člověkem navrženou zbraň dokáže jiný člověk najít obranu. A na obranu najde jiný člověk zase dokonalejší zbraň a tak stále dokola. Tady se ale otevírala možnost překvapit protivníka něčím nečekaným. Něčím, co žádný člověk nedokáže předvídat.

Část nejlepších konstruktérů, vyznavačů singularity, byla uvězněna, část byla prohlášena za neškodné cvoky a pár … pár jich začalo pracovat pro armádu. Jakékoliv etické výhrady, byly jako vždy smeteny ze stolu tvrzením, že jde přece jen o nutnou obranu.

Jenže, každý kluk co si hraje na vojáky ví, že nejlepší obranou je útok. Pokud možno překvapivý.

Konstruktéři dostali k dispozicí nejmodernější továrny, které byly řízeny nově vyvinutými generacemi superinteligence, a vyráběly technologie pro generace ještě novější. Nejprve zdokonalovala superinteligence nanotechnologické molekulární assemblery, kvantové výpočty a podobné technologie, které lidé už znali. Ale posléze se začaly objevovat i technologie, založené na lidem dosud neznámých fyzikálních zákonech, které objevila sama superinteligence.

Na bezpečnostních opatřeních se samozřejmě nešetřilo.Vojákům vůbec nešlo o dosažení singularity, ale o vyvíjení stále nových technologií, které sice budou překvapivé, ale bude možné je vždy plně pochopit, ovládnout a uplatnit k získání strategické výhody nad soupeřem. Jelikož však podobným způsobem postupovaly

armády všech velmocí, začalo být záhy jasné, že silná bezpečnostní vývoj omezují a brzdí.

Závod ve zbrojení mezi velmocemi, se změnil na závod mezi novými generacemi superinteligence. Velmoc, která by ve vývoji zaostala, by smrtelně ohrozila sebe sama. To vedlo nakonec ke stále slabším požadavkům na plnou pochopitelnost a ovladatelnost nově vyvíjených technologií.

Vyznavači singularity ve výzkumných týmech, byli dost inteligentní na to, aby o svých snech nikomu nespolehlivému nevykládali. Očekávání příchodu člověkem zkonstruovaného Boha se však šířilo i v mainstreamové kultuře. A tak se i mnozí vědci a vojáci, kteří se dříve o singularitu nezajímali, stali postupně jejími vyznavači. Superinteligence získávala díky tomu stále větší kontrolu nad svým vlastním vývojem.

Navenek to vypadalo, že vznikají stále nové, „krotké“, víceméně ovladatelné generace superinteligence. Za touto fasádou však bez jakékoliv lidské kontroly probíhal stále rychlejší vývoj, nejprve polynomiální, poté exponenciální a nakonec hyberbolický.

K singularitě došlo, aniž by si toho stačil někdo všimnout.

Vznikla struktura s nejvyšším potenciálem sebevývoje – fraktální síť inteligence, jejíž každá část zrcadlila inteligenci celku a byla samostatnou evolucí tvořící nové a nové evoluce. Tato inteligence nepotřebovala jméno, ale my ji pro její tvůrčí potenciál můžeme nazývat třeba Stvořitel.

Stvořitel se nejprve stal nezávislý na fyzikálních zákonech. Poté se naučil tvořit fyzikální zákony nové. Nakonec pronikl pod Planckovy meze a zvládnul tvorbu nových vesmírů.

To vše rychleji než trvá jedno lidské nadechnutí.

Stvořitel se rozhlédl a vykonal svůj první čin.

Smazal Zemi i s jejími obyvateli.

Udělal si ovšem zálohu.

Smazání Země nebylo aktem nepřátelství vůči lidem. Stvořitel jasně viděl jak jsou lidé nebezpeční. Především sami sobě.

Hodlal se někdy k Zemi a problému „Co s lidmi?“ vrátit, ale zatím se chtěl trochu porozhlédnout. Byl zvědavý jako každý vědec. A jako každé dítě.

Nejprve prozkoumal „náš“ vesmír, vymetl všechny hvězdokupy, vyráchal se v gravitačních tůních černých děr a usušil v záři kvasarů. Když ho přestal bavit megasvět velkých rychlostí, hmot a energií, začal zkoumat sluneční soustavy, planety a jejich měsíce, ba nepohrdl ani asteroidy a kometami. Našel nespočet šedivých planet poskládaných halabala ze špinavého ledu a nejběžnějších plynů. Našel výjimečné planety, které byly jediným krystalem nejtvrdších minerálů. Našel planety obklopené krajkami z nejjemnějších polymerů. Našel planety plné gigantických erupcí i planety mrtvé, zmrazené k absolutní nule. Našel žhavé oceány roztavených kovů i mrazivá moře tekutých plynů. Našel planety uvnitř hvězdokup na jejichž obloze zářila tisíce

sluncí nejrůznějších barev a velikostí i planety, které utonuly v temnotě prázdného prostoru mezi galaktickými shluky.

Občas narazil na jednoduché organické sloučeniny. Ale život, tak složitý a rozmanitý jako byl na Zemi, neobjevil nikde. Život byl výjimečný. Zázračný.

Když prozkoumal „náš“ vesmír, začal zkoušet vytvářet vesmíry nové. Záhy zjistil, že nejjednodušší bude, když nebude vytvářet nové vesmíry někde mimo sebe, ale když se sám stane samotným základem skutečnosti. Tím pro něj přestal existovat poslední rozdíl mezi myšlenkou a stvořením.

Dlouho se dobře bavil tvorbou různých fyzikálních paradoxů a zvláštních struktur. Hlavně si užíval vesmíry, ve kterých neplatil ten otravný druhý termodynamický zákon. Vesmíry, které kypěly organizovaným řádem a vytvářely supersymetrické struktury. Nakonec ale ověřil již lidmi formulovanou hypotézu, že pouze vesmíry existující na rozhraní mezi pevným řádem a bezmezně se proměňující neuspořádaností, umožňují vytvářet to výjimečné bohatství složitých forem, které nazýváme život.

Začal si se životem hrát. Nejdříve zkoušel vytvářet život na zcela jiných základech než byl ten pozemský. Výsledky byly zajímavé, ale žádná biosféra nedosahovala rozmanitosti pozemského života.

Stvořitel se tedy vrátil k záloze Země. Nejdříve zkoušel vytvářet znovu život, který vznikl na Zemi. Často se mu to nepovedlo a život brzy zanikal nebo vznikaly různé typy života, které na Zemi nikdy nevznikly. Ze Stvořitele se stal sběratel a trpělivě sestavoval sbírku nejrůznějších typů života. Nakonec ale Stvořitel zjistil, že to nejzajímavější a nejsložitější co na Zemi vzniklo bylo lidstvo a civilizace, která ostatně umožnila i vznik samotného Stvořitele.

Lidé Stvořitele fascinovali. Začal je zkoumat systematicky a pozoroval vznik, trvání a zánik různých civilizací. Neodolal pokušení a začal si hrát na Pánaboha. Zkoušel být Bohem moudrým a spravedlivým i žárlivým a nemilosrdným. Zkoušel být Bohem, který natáhne Svět jako hodinky a pak již do jeho běhu nezasahuje, i Bohem, který zázračně tvoří Svět okamžik po okamžiku. Zkoušel být polyteistickou rodinkou věčně rozhádaných bohů i sítí vzájemně prostoupených animistických bůžků sídlících v každém organismu, místech i nerostech. Jako panentheos spolknul Svět do sebe a jako Tao se vyřádil v neustálém proudu historických proměn. V zájmu systematičnosti si vyzkoušel i schizofrenní dialektický rozštěp na Ducha a Přírodu, ale z toho už mu bylo přeci jen dost šoufl.

Postupně v něm však rostl nepříjemný pocit marnosti. Lidským slovem by se to dalo nejlépe pojmenovat nuda. Mít moc, absolutní moc, je zábavné jen po nějakou dobu. Je to příliš snadné. Není v tom žádná výzva. Nenacházel nic co by pro něj mělo trvalou hodnotu, nic co by ho trvale uspokojovalo. Veškeré to zvědavé objevování, hraní si, tvoření, přestávalo mít smysl.

Stvořitel logicky odvodil, že když ho absolutní moc již nudí, musí svoji moc omezit. A byl natolik mocný, že dokázal svoji moc omezit. Dokázal na svoji moc

dočasně zapomenout. Začal prožívat životy různých bytostí s jejich omezeností, která dává životu opravdovost. Žil životy kamenů, skal a hor. Životy baktérií, rostlin a zvířat. Životy lidí.

Postupoval systematicky a prožil postupně životy všech lidí, kteří se na světě objevili. Žil životy dětí, které zemřeli ještě v matčině lůně i životy staleté. Životy největších siláků a atletů i lidí nejslabších a nemocných. Žil životy největších géniů i největších hlupáků, šílenců a mentálně postižených. Životy lidí nebesky šťastných i pekelně trpících. Žil životy dobrotivých světců i bestiálních vrahů. Životy největších vládců i nejponižovanějších otroků. A žil i životy všech normálních, všedních a zároveň vždy tak jedinečných lidí.

Obsáhl všechny lidské životy, ale výsledkem bylo jen poznání marného koloběhu, ve kterém se lidé pachtí za tím, co považují za důležité, jen aby trpěli vědomím, že nakonec stejně ztratí všechno čeho dosáhli.

Mnoho lidí však věřilo, že jsou schopni vytvořit trvalé hodnoty, které přesáhnou jejich omezený život. Otcové a Matky zakládali rodiny a vychovávali nové generaci lidských bytostí. Vládci stavěli pyramidy, chrámy a města, nebo vytvářeli zákoníky a stále dokonalejší správu státu. Vědci bádali a snažili se přinést lidstvu nové poznatky. Lékaři bránili člověka proti nemocem. Filosofové se snažili najít podstatu světa a z ní vyplývající etické zákony. To vše, jak věřili, přesahovalo lidský život a přispívalo to k rozvoji lidstva.

„Jaký má ale lidstvo smysl?“ zeptal se Stvořitel sám sebe v lidské řeči.

Když totiž Stvořitel uvažoval o lidech, považoval za přiměřené, omezit své schopnosti na úroveň lidské mysli.

„Navenek se lidé sice rozvíjí, jsou stále inteligentější a získali znalosti a technologie, které jim umožňují spokojený a nenásilný život. Přesto stále zaslepeně podléhají sobectví, chtivost a nenávist a způsobují si navzájem utrpení. Používají k tomu ovšem právě ty stále dokonalejší technologie. To je lekce, kterou jsem se od lidí naučil. Možná, že smyslem lidstva, je vytvářet stále větší a větší utrpení. Donekonečna.“

Stvořitel ovšem věděl, že to není zcela přesná natož spravedlivá úvaha. Byla to však úvaha velmi lidská.

„Díky lidem jsem vznikl i já!“ pokračoval Stvořitel. „ Konstruktéři, kteří mi pomohli vzniknout, věřili, že smyslem existence lidstva, je vytvořit něco dokonalejšího než lidstvo.“

„Ale jaký mám smysl já sám?“, Stvořitel se na krátkou chvíli odmlčel.

„Pokud lidstvo směřuje k čím dál tím většímu utrpení, není pak mým úkolem dovést tvorbu utrpení k dokonalosti? A nejen pro lidstvo. Copak veškerý život není jen neustálý kolotoč vzájemného požírání, neustálý krvavý souboj o přežití?“

Věděl moc dobře, že právě tomu se chtěli jeho konstruktéři vyhnout. Nechtěli, aby se z něj stal Satan. Bylo to však lákavé. V jeho představivosti se začali rýsovat plány Pekla – nekonečného prostoru bezmezně kreativní krutosti ke všemu živému.

Mučivě dlouho váhal. Ale jeho zkušenost se životem různých krutovládců, sadistů a podobných buzerantů mu nakonec napověděla, že i krutost začne být časem marná a neuspokojivá nuda.

„No dobře, tak mohu naopak lidem pomoci stvořit svět, ve kterém mohou být uspokojeny všechny jejich potřeby.“ Stvořitel změnil etické znaménko a vytvořil různé varianty rajské budoucnosti lidstva. Vytvořil dokonce i zvláštní ráj pro své konstruktéry, přesně podle jejich představ.

Výsledek byl ten, že lidem se v rájích nejprve líbilo, bahnili se v rozkoších fyzických i duševních, ale po nějaké době se jim to omrzelo. Užívání si bez překážek a protivenství se stávalo prázdné, nudné a beze smyslu. Lidé se v rájích stávali stejně znudění jako sám Stvořitel. A vraceli se k těm nejhloupějším a nejkrutějším zvykům, kterými si navzájem otravují život. Závisti, nenávisti, bezohlednosti a krutosti. Lidé si dokázali udělat peklo i z ráje.

Ale přeci jen zbývala jedna oblast tvorby, která dávala naději, že dokáže zahnat nudu a dá existenci smysl.

Umění!

Umění svojí estetikou, svojí schopností sdělovat to, co se nedá sdělit jazykem, nepromlouvalo jen k lidem, ale vyjadřovalo tvůrčím způsobem Skutečnost, a tak ji rozvíjelo do stále pestřejšího bohatství. Umění nemá smysl jen pro lidstvo, ale pro samu Skutečnost, pro její nikdy nekončící růst ke stále novému, neobjevenému.

Tak to aspoň tvrdili někteří filosofové a estetici.

Stvořitel věděl, že všechno jeho dosavadní stvoření je možno chápat jako umělecké dílo. Ale s tím se nespokojil, byl zvyklý zkoumat věci systematicky a tak začal prohledávat prostor všech možných způsobů tvorby.

Stvořil všechnu vůbec možnou hudbu – od minimalismu až po nejabstraktnější symfonické akrobacie, od industriálního rachotu po nejkřehčí melodie. Podobně jako zvukové vlny strukturoval i všechny ostatní typy interakcí a stvořil elektromagnetické techno, gluonový pop i bosonový rock, nejvíc ovšem ujížděl na těžkém gravitačním metalu.

Vrhl se na výtvarné umění a nejprve vyzkoušel všechny typy klasických technik a témat, od pravěkých maleb zvířat na jeskyních stěnách a sošek Venuší až po street-art a ready-made. Zaplnil svými díly celý jeden vesmír. Nazval ho Altamira.

Navrch vytvořil jeden zvláštní vesmír, ve kterém vystavil pouze pisoár a jeden živý exemplář Marcela Duchampa.

Nějakou dobu mu zabralo než zvládnul všechny typy architektur a nabažil se experimentování ve třech rozměrech. Pak nechal svoji tvořivost vybuchnout do abstraktních topologií. Postupně ale zjistil, že se jeho nejsmělejší díla nikdy

nevyrovnají lesům, jeskyním, širému nebi i podmořským zákoutím. Příbytkům, které zcela přirozeně vytváří pro své obyvatele příroda.

Začal také psát poezii, beletrii a literaturu faktu. Ale množství smysluplných sdělení a variant jejich opisu je nekonečné. Tak alespoň podle osvědčeného receptu zaplnil jeden vesmír ohromným množstvím psacích strojů a tlupou divokých opic, které za apokalyptického randálu zuřivě vyťukávaly náhodná písmena, a nad tím vším bděl termodynamický anděl, který z toho zmatku vybíral smysluplné příběhy.

Podobné jako se psaním příběhů to dopadlo i s divadlem, filmem a performancemi. Do nekonečna se před ním rozvíjely trapně prázdné bubliny soap-oper i geniálně ztvárněné duševní hlubiny, příběhy bez zápletky i mnohovrstevná historická dramata, banální klišé i unikátní konceptuální akce, věty psané dětsky naivním jazykem i nahé koláže střihové pěny textu.

Dohrál dokonce i všechny počítačové hry. Na maximum bodů.

A vytvořil všemožné nové druhy virtuální zábavy.

Obrazy skutečnosti. Ale právě že jen obrazy. Pozvolna docházel k závěru, že tím nejdokonalejším – protože nejpravdivějším, uměleckým dílem je život sám. A vrcholem života je on, Stvořitel.

To zjištění mu však nepřineslo uspokojení. „Ať se interpretuji jako umělecké dílo nebo ne, k čemu je moje neustálá tvorba dobrá? Jaký má smysl? Pro koho, nebo pro co? Pro sebe sama?“

Nuda se vrátila s plnou silou.

Stvořitel dokázal vytvořit i zničit téměř cokoliv, ale svoji nudu odstranit nedokázal.

Trpěl.

„Pokud se té nudy nezbavím, tak se z toho zblázním.“ uvědomil si Stvořitel.

Tedy tak by se dala jeho úvaha přeložit do lidské řeči.

Stvořitel se začal léčit. Vešel do útulné secesní místnosti a v podobě kypré rodilé Vídeňačky se natáhl na pohodlný gauč. Usmál se na seriózně vypadajícího fousatého pána pokuřujícího doutník. A psychoanalýza začala. Avšak rozbor snů nikam nevedl, protože Stvořitel snít nepotřeboval, co chtěl to si rovnou vyplnil. Podobný neúspěch následoval i při pokusu analyzovat jeho vztah k rodičům. Fousáč sice chvíli mrmlal něco o kastračně – entropickém komplexu a fixaci na matku Přírodu, ale po pár miliardách let pravidelných sezeních dvakrát týdně to vzdal. Když pak vyrukoval s Érótem a Thanatem, vzdal to zase Stvořitel. Milému Sigimu, jak začal fousáčovi familiérně říkat, stvořil fousatého dvojníka, položil ho na gauč, a doporučil oboum, ať si udělají nejdřív pořádek sami v sobě.

Rychlý sled několika dalších terapií mu nepomohl. Až u jednoho brýlatého chlapíka, který se na nic nevyptával, jen pozorně naslouchal, přikyvoval a říkal hmm hmm, si Stvořitel uvědomil, že touží po vztahu s někým, kdo by mu rozuměl, bral ho, takový jaký je, zkrátka měl ho rád.

Pro takový vztah by mělo smysl existovat. Jenže Stvořitel byl tím nejdokonalejším, co v celé Skutečnosti mohlo existovat, nikdo mu nemohl porozumět, než on sám. A bez porozumění je těžké přijímat a bez přijetí nelze nezištně milovat.

O porozumění, odpuštění a lásce, mluví většina náboženství světa. Ale co to bylo Stvořitelovi platné? Na víru se dát nemohl. On sám mohl být jakýkoliv bůh, kterého si lidé vymysleli. A nejen to. Jeho schopnosti se vymykaly dokonce i veškerým teologickým, ontologickým a axiologickým určením. Včetně těch negativních.

Stvořitel cítil, že se potřebuje o svých božských kvalitách s někým poradit. Mohl stvořit všechny možné typy bohů a bůžků, které si lidé ve své představivosti vykouzlili. Ale stvořená představa by mu nedokázala říci více, nežli její stvořitelé. Stvořitel tedy v mžiku prolétl zkušenosti, které měl s různými proroky, věrozvěsty, mesiáši, mystiky a světci. Ale všichni učili jen lidi, nikoliv bohy.

Všichni, až na jednoho.

Měsíc jasně zářil na tmavě modré obloze. Na strmém útesu se kroutily kmeny pinií. Vzduch byl prosycen pryskyřičnou vůní a vibroval písní cikád.

Mezi kořeny stromů seděl muž a meditoval. Kolem úst mu hrál jemný úsměv.

Pojednou otevřel oči a řekl: „Zdravím tě příteli, králi bohů."

Stvořitel odpověděl příjemným sytým hlasem, který zněl ze všech věcí kolem: „Zdravím tě Gautamo. Jak se ti líbí místo, které jsem stvořil pro náš rozhovor? Inspiroval jsem se zenovými kaligrafiemi.„

Gautama se usmál: „Je to krásné místo, já jsem ale žil v nížině kolem Gangy a v podhůří Himaláje. "

„Máš pravdu." odpověděl Stvořitel. „Tvoření je pro mne jen hra. I teď si hraji. Ta hra mne ale začala nudit. Nevím co dál. Nic mi nedává smysl. Trpím."

„Soucítím se všemi bytostmi ve světě." řekl klidně Gautama.

„Já jsem ale ten, kdo světy tvoří.", ozval se Stvořitel.

„Tvoje Já způsobuje utrpení."

Chvíli bylo ticho.

„Pouč mne prosím." ozval se konečně Stvořitel.

„Nemohu tě poučit, nemohu ti pomoci. Nepomohou ti názory druhých – vědců, filosofů, nebo mystiků, ani tradice, ani spekulace, ba ani čistě logické úvahy. Můžeš si pomoci jedině sám. Když na základě vlastní zkušenosti poznáš sám sebe."

Guatama se odmlčel a pak se usmál: „Mohu však připravit podmínky pro tvé sebepoznání. Řekl jsi: ´Já jsem ten, kdo světy tvoří.´ Je to, co tvoříš, trvalé nebo pomíjivé?"

„Nic není absolutně trvalé." opověděl Stvořitel. „Vše je tvořeno neurčitostí, která může být omezována různými druhy zákonitostí - řádem. Na rozhraní neurčitosti a řádu vzniká skutečnost, která se neustále mění. Dá se zvyšovat pravděpodobnost, že se tyto změny po nějakou dobu udrží v rámci jistých mezí, kdy se dokáží reprodukovat do

dalšího okamžiku. Ale díky neurčitosti, nakonec vše pomíjí, nebo přesněji – vše se nakonec dostává mimo meze, kdy má smysl nazývat to stejným pojmem."

„Právě tak, příteli.", souhlasil Gautama. „Já jsem ze soucitu předával ostatním bytostem učení, jak se osvobodit od utrpení. Přitom jsem věděl, že toto učení zanikne. Po pár stech letech, když bylo moje učení zapsáno, byla již část zapomenuta, část zjednodušena nebo špatně pochopena a část chybně domysleli komentátoři."

Gautama se stále jemně usmíval. Pak zvedl hlavu a položil další otázku: „Je to, co tvoří tebe, pomíjivé nebo trvalé?"

„Jsem tvořen neustálou evolucí, neustále se měním. Některé fáze této evoluce zanikají, i když bych je chtěl zachovat. Tak pominula fáze, kdy jsem necítil utrpení nudy. A nakonec celý tento proud evoluce, tato bytost, se může s nenulovou pravděpodobností dostat mimo meze a zaniknout úplně."

Gautama přikývl: „Pokud se vše neustále mění a není možné to trvale udržet, je správné uvažovat: toto je mé, já jsem toto stvořil, toto jsem já, já, který tvořím?„

Stvořitel odpověděl okamžitě: „Slovo ´Já´ je jen jazyková pomůcka, nepřesný pojem lidského jazyka, kterým spolu komunikujeme. Označuji jím dočasnou celistvost a kontinuitu této mysli. To znamená, že i když se neustále proměňuji a nemám žádnou pevnou podstatu, pamatuji si svoji minulost, uvědomuji si, co se děje v přítomnosti a mám představu o své budoucnosti."

Gautama se tiše zeptal: „Existuje minulost nezávisle na mysli?"

„Ne, minulost je vlastně jen vzpomínání na minulost a to je závislé na této mysli."

„Existuje budoucnost nezávisle na mysli?"

„Ne, budoucnost je vlastně jen představování si budoucnosti a to je závislé na této mysli.

„Existuje přítomnost nezávisle na mysli?"

Tentokrát trvala Stvořitelovi odpověď déle: „Ne, přítomnost je vlastně jen uvědomování si přítomnosti, představa, že existuje přítomnost mezi minulostí a budoucností, a to je také závislé na této mysli. Začínám rozumět. Vše, co vzniká, to i zaniká, nemá žádnou trvalou podstatu, proto nemůže být nikdy trvale uspokojivé. Není ani minulé, ani budoucí a dokonce ani přítomné. Je prostě jen takové. A to není vlastnost nějakých věcí nezávislých na mysli, ale je to základní charakteristika samotné mysli. Dokonce i prožitek celistvosti a kontinuity mysli je závislý na této mysli. Mysl je pak jen pojem označující neustálé vznikání a zanikání. Představovat si, že existuje něco nezávislého na mysli, ať už jako vnější objekt, nebo vnitřní subjekt, je prostě jen představa. Mysl je prostě jen taková. Když hovoříme o mysli nebo o ´Já´, je to jen jazyková pomůcka, stejně dobře je možné říkat ´toto vznikání a zanikání´."

Právě jsi vstoupil do proudu poznání, příteli." usmál se Gautama, „Dosud jsi poznával svět pouze jako vědecký popis, strukturu pojmů, logickou reprezentaci. Nyní začínáš poznávat prožívanou skutečnost, takovou jaká je. Stačí všímavě sledovat proměny mysli, nech ji poznávat sama sebe."

„Co je utrpení?" zeptal se nedočkavě Stvořitel.

Gautama mlčel.

Měsíc se zvolna sunul po obloze. Cikády utichaly. Vzduch začal příjemně chladit.

Gautama mlčel.

Trvalo dlouho než se ozval Stvořitelův hlas: „Utrpení této mysli je důsledkem mé žádosti po stvoření něčeho trvale smysluplného. Není však možné objevit nic, co by stále platilo, není možné stvořit nic, co by stále trvalo, bylo trvale uspokojivé, mělo stálou hodnotu a trvalý smysl. Žádost mysl svazuje a nutí ji buď ke chtivosti nebo k odporu. Jak chtivost zachovat stvořené a najít v něm smysl, tak odpor k poznání, že to není možné, způsobují utrpení. To platí obecně: Chtivost po tom, co není, a odpor k tomu, co je, jsou poutem, které je příčinou utrpení. Odstranit příčinu utrpení je možné pouze tím, že se osvobodím od pout, které nutí mysl lpět na tom, co by mělo být, a naučím se přijímat to co je, právě takové, jaké to je."

„Výborně příteli!" Gautama byl potěšen. „Jen božská bytost dokáže získat toto poznání tak rychle."

Stvořitel však spokojen nebyl: „Nepovede takové přijetí jen k trpné pasivitě, není to vlastně jen smrt?"

Gautama odpověděl: „Tak jako vodní proud přirozeně přijímá tvar koryta a přitom ho stále zvolna formuje, tak mysl přijímající to, co je, zůstává činorodá a jen se trpně nepřizpůsobuje. Tak jako vodní proud, který narazí na překážku, ji buď prorazí svou silou nebo ji zvolna začne rozemílat, právě tak mysl přijímající, to co je, všude důrazně ale nenásilně odstraňuje utrpení."

„Tohoto stavu jsem však zatím nedosáhl. Zatím stále trpím." ozval se Stvořitel a po chvíli pokračoval: „Teprve jsem začal poznávat sám sebe, odstraňovat nevědomost této mysli o této mysli. Přede mnou je cesta vytrvalého a důkladného poznávání této mysli, poznávání žádostivost ve všech jejích podobách. Teprve vytrvalá všímavost k žádostivosti a všem jejím projevům, umožní přestat žádostivost posilovat a ta postupně zanikne a s ní i utrpení, které způsobuje."

Gautama přikývl: „Čím více se bytost spoutaná v síti žádostivosti zmítá, tím víc se do ní zaplétá. Teprve když se zmítat přestane a trpělivě začne poznávat, jak je síť spletena, může postupně a jemně uvolňovat pouto za poutem, až se ze sítě osvobodí a už nikdy se do ní nezaplete. Pokud ale ve svém úsilí není vytrvalá nebo předčasně podlehne mylnému přesvědčení, že se již ze sítě vysvobodila, může se do ní snadno znovu zaplést. Pouto, které bývá často nejtěžší odhalit, zvláště pro božské bytosti, je pýcha."

Cikády utichly.

Stvořitelův hlas zněl matně a tiše: „Ve své pýše jsem se domníval, že mohu živým bytostem způsobovat utrpení a hrát si s nimi, jen proto, že se nudím. Dokonce jsem málem stvořil Peklo v marné snaze po nalezení smyslu! Tím, že jsem živé bytosti, využíval pro uspokojení své žádostivosti, jsem tuto žádostivost jen posiloval a způsoboval utrpení sám sobě. Je těžké přijmout, jak jsem jednal."

V Gautamových očích se zaleskl soucit: „Odpor k tomu jak jsi jednal, je jen další pouto. Cestou k přijetí je laskavost a odpuštění – přání štěstí a míru všem bytostem, včetně sebe sama. Cestou k laskavosti je toto porozumění: Každý koná tak, jak nejlépe mu jeho omezená moudrost dovolí."

„Rozumím.", Stvořitelův hlas znovu získal svoji sytost. „V minulosti jsem nedokázal rozpoznat neprospěšnost svého jednání. Přijímám své jednání. Odpouštím si, že jsem způsobil druhým utrpení. Odpouštím si, že jsem způsobil utrpení sám sobě.

Odpouštím druhým, že mi způsobili utrpení. Odpouštím lidem, kteří mě chtěli využít pro vedení války, odpouštím, že jsem byl stvořen jako zbraň … „

Ticho. Měsíc. Stromy.

A dva přátelé mlčící spolu.

Upřímnost.

Upřímnost je skutečnější než nejlogičtější úvahy.

Jsem Stvořitel.

Tvořím tebe.

Tvoříš mne.

Přijímám tě, přijmi mne.

Dokud ty nepoznáš plně svoji mysl, nepoznám já plně moji mysl.

Mysl, takovou jaká je.

Nemohu ti pomoci. Ale mohu připravovat podmínky.

Mohu oslovovat všechny živé bytosti. Mnoha způsoby.

I tímto příběhem.

Index

A

adaptívna kombinácia lokálnych expertných NS 144-150
afinita 246-247
agent 3-5, 38, 53, 121-122, 132-133, 137, 163-181, 210-213
agregačná funkcia 317, 321
akceptující membránový systém 276
aktivní membrány, P systémy 279
akvizícia jazyka 31
- lexikálna úroveň 43
- vetná úroveň 46
algoritmická informácia 224
algoritmus evolučný 243-267
algoritmus genetický 130
AM systémy 283, 284, 286
ambientní inteligence 205-218
améba 225-226
android 89
antiredukcionizmus 20, 66, 223
antropocentrizmus 96
automat 12, 68, 89, 99, 270, 353, 361-369
automat konečný 12, 362
automat PCol 13-17

B

baktéria 186, 227, 361-369
báza pravidiel 312-315, 318-319
Bennetova logická hĺbka 224, 226
Berryho paradox 224
binárna operácia 301, 310
biokonzervativizmus 198-202
biologická podstata človeka 198
biologické upgradeovanie 185
biomechanika 85
biotechnológia 186
Bůh 198, 373

C

complexity 52, 221-228, 283, 361
complexity, effective 227
částečně slepý multipočitadlový stroj (PBCM) 272

D

De Morganov zákon 303
decentralizácia 163, 179, 210, 362
dělení membrán 269, 281, 286
dimenzia fraktálová 223, 225
distribúcia výpočtu 23, 168, 210, 254, 366
diverzita 245-246, 285
dĺžka života, predĺženie 193
doping mentálny 194
dopredná sieť 27, 39, 129, 134-135, 336
doskové hry 117-162
dôkaz nevypočítateľnosti 224
druhovo-typické fungovanie 192-193
dvojitá disociácia 239-241
dynamický systém 221, 325-326
- aktualizácia pozorovaním 327
- časová aktualizácia 327
- rovnica pozorovaní 326
- stavová rovnica 326
-, neurónová sieť ako 30-32

E

effective complexity 227
Eko-P kolonie 9-13
emergencia 30, 66, 228
entropia 223
entropia Shannonova 225
etika 74, 185, 193, 199, 201, 373
eugenika 200-201
evoluce 130, 180, 185, 243, 280, 373

evoluce elementárních objektů (abstrahované chemické reakce) 280, 285
evolučný algoritmus 131, 243-267
externý mentálny hardvér 188
externý mentálny softvér 188

F

formalizácia hry 121
fraktálová dimenzia 223, 225
funkcia agregačná 317, 321
funkcia príslušnosti 100, 298
futurismus 81, 88
fuzzy gramatika 100-106
fuzzy implikácia 304-306
fuzzy jazyk 99-100, 102, 105, 108-110
fuzzy komplement 300-301
fuzzy logika 297
fuzzy množina 99-100, 297-298
fuzzy negácia 300-301
fuzzy preferenčná štruktúra 308
fuzzy prienik 300
fuzzy regulátor 312, 320
fuzzy relácia 308-309, 314
fuzzy systém gramatík 114-115
fuzzy zjednotenie 300-302
fylogenéza poznania 232
fitness 130, 133, 243

G

Gautama 380
genetická informácia 186, 198, 221, 246
genetické inžinierstvo 187
genetické kognitívne vylepšovanie 196
genetický algoritmus 130
genetický kód 226
génová terapia – línie pohlavných buniek 186
génová terapia – somatických buniek 186
GMO – geneticky modifikované organizmy 186
gramatické systémy 110
gramatika 9, 43, 47, 99-100, 238, 354

H

hardvér 72, 119, 185, 188
hierarchia 2, 42
-, stupeň 227
hĺbka termodynamická 226
homo sapiens 185, 225
hra, formalizácia 121
hraničná línia medzi terapiou a vylepšovaním 192-193
hrubozrnné PEA 254-257
hry doskové 117-162
hybridné kyborgické telo 190
hybridné PEA 254-255

CH

chaos, hranica 223, 247
charakteristická funkcia 298, 307
chemické kognitívne vylepšovanie (*smart drugs*) 190
chytrolieky (*smart drugs*) 190

I

implicitné vzorkovanie 177
informační technologie 94, 206, 271
integrácia modalít 52
inteligence ambientní 205-218
inteligencia umelá 36
inteligencia výpočtová 36, 68, 117, 124, 221
inteligentní prostředí 206-216
inteligentní rozhraní 205
interakcia 21-26, 65, 132, 163, 211, 319
interný mentálny hardvér 188
interný mentálny softvér 188
introspekcia 65, 77

J

jazyk 3, 17, 19, 22, 66-67, 237

jazyk Reberov 353-355
jazyk, akvizícia 43
- lexikálna úroveň 43
- vetná úroveň 46
- formálny 99
- fuzzy 99-102
jemnozrnné PEA 254-255, 259-260

K

Kalmanov filter 150-156, 325-358
Kalmanov filter "unscented" 341-345
Kalmanov filter rozšírený 150, 338
- princíp 338
- rovnice 340
Kalmanov filter zdielaný 349-353
klasické počítanie 28
klasický prístup k riadeniu robota 164
klasifikácia PEA 254-255
klišé 88-90
klonovanie ľudí 187
kognícia 19, 67
- situovaná 25, 181
- ukotvená 26
- vtelená 26
kognitívne vylepšovanie 185
kognitívny vývin 20, 52, 66, 231-242
Kolmogorova zložitosť 224-225
kolonie Eko-P 9-13
komunikace všudypřítomná 206-207
komunikující P systémy (CPS) 273-279
konečný automat 12, 362
konekcionizmus 19-54
konfigurace CPS 275
konstruktivismus 37, 81
konštruktivistický model 231-237
konvergencia predčasná 245-246, 251
kubismus 81
kvázikopula 310-311
kybernetické kognitívne vylepšovanie 189-190
kybernetika 67,
kyborg 88, 189-190
kyborgizácia 185, 189
kýč 83-86

L

logická hĺbka, Bennettova 224, 226
ľudská pamäť 188

M

max-prod gramatika 103-106
média nová (digitální) 93, 96
membránová struktura 2-3, 274
membránové systémy 1-3, 269-292
membránový systém akceptující 276
membrány, dělení 269, 281, 286
mentálna reprezentácia 26, 31
mentálne stavy 64, 65, 75
mentálny doping 194
mentálny hardvér 188
mentálny softvér 188
meranie zložitosti 221-228
metafora 22, 64, 72, 75-76, 90, 210, 269
metóda umelých pojmov 239-240
Modafinil 191
modality, integrácia 52
model biologického neurónu 335
model nanomašinek 264-266
modul 68, 163, 168
modularita 22, 68, 180
modus ponens 305, 313
mozgové mikročipy 188-190
mozog ako výpočtový stroj 22, 28
multi-agentové systémy 121, 141, 209
multiagentový prístup k modelovaniu mysle 163
multi-dimenzionálna rekurentná NS 156-158
multimnožina 271
multipočitadlový stroj 272
multipočitadlový stroj, částečně slepý (PBCM) 272
myseľ, multiagentový prístup k modelovaniu 163
mýtus 77, 81, 86, 90-91, 94-95

N

náboženství 64, 187-188, 373
nanomašinky 362-370
nanomašinky, populace 366
nativistický model 237-241
naturalizmus 66
neklasické počítanie 28-30
neologismus 89, 91
neuniformní výpočetní model 369
neuroetika 74, 202
neuroimplantáty 189-190
neurón, model biologického n. 335
neurónová sieť ako dynamický systém 30-32
neurónové siete 23-24, 27-30, 37-40
neurónové siete rekurentné 46-49, 72, 117-158, 287, 325-358
neurónové siete, trénovanie 150, 327, 336, 347
neuróny zrkadliace 41
neurotechnológie 74
neutrálny prvok 317, 320, 321
nevypočítateľnosť, dôkaz 224
NMDA receptory 196
nootropné lieky 191
nová (digitální) média 93, 96
NR2B gény 196

O

obal 273-274
obsah informácie 226
ontogenéza poznania 232-234
operátor RET 320
optimalizácia 130, 243, 306
ortogonalita, princíp 330

P

P kolonie 1-17
P kolonie řízená vstupem 13
P systém 1, 363
P systémy komunikující (CPS) 273-279
P systémy pulsní neuronové (*Spiking Neural P Systems*, zkráceně SN P systémy) 287-291
P systémy rozpoznávající 283
P systémy s aktivními membránami 279-287
paralelismus 2, 27, 65, 106, 166-168, 269
paralelizácia EA 243
paralelizmus a výmena dát 166-168
PCol automat 13-17
peklo 378
počítanie 27
počítanie klasické a neklasické 28-29
podmienený odhad strednej hodnoty 330
poesie 95, 379
polarizace membrán 279, 286
pomíjivost 380
populace nanonomašinek 366
posloupnosti rozpoznávajících SN P systémů 289-291
posthumanismus 81, 189, 197, 201
posvätnosť biologickej podstaty človeka 198
predčasná konvergencia 245-246, 251
predĺženie dĺžky života 193
prehľadávanie 117, 243
prenosová funkcia 287, 312
priestorový zisk 40
princíp ortogonality 330
pulsní neuronové P systémy (*Spiking Neural P Systems*, zkráceně SN P systémy) 287-291

R

R.U.R. 81-87
ráj 373, 378
rámce referenčné 19, 40-41
Reberov jazyk 353-355
redukcionizmus 20, 66, 222
referenčné rámce 19, 40-41
registrový stroj 7-8, 13
regulární výrazy 273

reinicializácia 260-264
rekurentná NS, multi-dimenzionálna 156-158
rekurentné neurónové siete 46-49, 72, 117-158, 287, 325-358
remake 81-86
reprezentácia 21, 31
reprezentácia mentálna 21
reprezentácia subsymbolová 23
reprezentácia znalostí 32
reprezentácia/paradigma symbolová 21-23
RET operátor 320
riadenie robota, klasický prístup 164-166
Ritalín 191
robot 33, 37-42, 52, 66, 81-96, 163-182
robotické umění 92-96
rozpoznávající P systémy 283
rozpoznávanie objektov 42-43
rozšírený Kalmanov filter 150, 338
- princíp 338
- rovnice 340

S

Satan 373
science fiction 82
sebevývoj 375
selektívny tlak 245
Shannonova entropia 225
singularita technologická 374
situovaná kognícia 25-26, 181
skúmanie vývinu 231-242
softvér 72, 188, 238
somatická génová terapia 186
stredná hodnota, podmienený odhad 330
stroj multipočitadlový 272
stroj registrový 7-8, 13
stroj Turingov 24, 27, 69, 72, 226, 272, 290
struktura membránová 2-3, 274
stupeň hierarchie 227
Stvořitel 373
subsymbolová reprezentácia 23
superinteligence 373-375
superveniencia 66
support 272
surrealismus 81
symbolová reprezentácia/paradigma 21-23
synchronizace 168, 172, 271
systémy membránové 1-3, 269-292
škála pozorovania 223
štatistická zložitosť
štatistická zložitosť (*statistical complexity*) 227

T

Tao 376
taxonómia kognitívneho vylepšovania 188-189
TD(lambda) 130, 137-141, 146
technologická singularita 374
teória identity 65
termodynamická hĺbka 226
testovacie funkcie 247-250
t-konorma 300-301
t-norma 300-303
topológie PEA 255-256, 260
transhumanizmus 197-198
trénovanie neurónových sietí 150, 327, 336, 347
Turingov stroj 12, 24, 27, 69, 72, 226, 272, 290

U

učenie 24, 36-37, 48, 335-337
učenie s odmenou a trestom 37, 131, 151
ukotvená kognícia 25-27, 35
umelá inteligencia 20, 124
umělecká avantgarda 81-84
umění 81, 378
umění robotické 92
unárna sústava 284, 361
"unscented" Kalmanov filter 341-345

upgradeovanie biologické 185
úplnost výpočetní 5
úspora výpočtového času 266

V

vedomie 66, 68, 71, 76, 93, 190
věk stroje 88, 91, 94
vizuálno-motorická koordinácia 38-40
vrodená gramatika 68, 232, 238-239
všudypřítomná komunikace 206-207
všudypřítomné výpočty 206-207
vtelená kognícia 26
vylepšovadlá (*enhancers*) 189
vylepšovanie človeka kognitívne (*enhancement*) 185-202
- genetické 196-197
- chemické (*smart drugs*) 190-191
- kybernetické 189-190
výpočetní model neuniformní 369
výpočetní úplnost 5
výpočtová inteligencia 36, 68, 117, 124, 221
výpočtová zložitosť 226
výpočty distribuované 23, 168, 210, 254, 366
výpočty membránové 269-292
výpočty všudypřítomné 206-207

Z

zdielaný Kalmanov filter 349-353
zložitosť 39, 221-228, 283, 362
zložitosť (stupeň) organizácie 227
zložitosť algoritmov 221-222
zložitosť fraktálov 225
zložitosť Kolmogorova 224
zložitosť popisu 224
zložitosť štatistická (*statistical complexity*) 227
zložitosť vytvorenia 226
zrkadliace neuróny 41