Superturingovský výpočetní potenciál kognitivních a evolučních systémů

Wiedermann, Jiří
2001

INSTITUTE OF COMPUTER SCIENCE

ACADEMY OF SCIENCES OF THE CZECH REPUBLIC

Superturingovský výpočetní potenciál kognitivních a evolučních systémů

Jiří Wiedermann

Technical report No. 832

únor 2001

Institute of Computer Science, Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2, 182 07 Prague 8, Czech Republic
phone: +4202 6605 3520 fax: +4202 8585789
e-mail: wieder@uivt.cas.cz

INSTITUTE OF COMPUTER SCIENCE

ACADEMY OF SCIENCES OF THE CZECH REPUBLIC

Superturingovský výpočetní potenciál kognitivních a evolučních systémů

Jiří Wiedermann[1]

Technical report No. 832
únor 2001

Abstrakt

Výpočetní systémy, jejichž chování modeluje prvky chování živých organismů a jejichž funkčnost je odvozena od našeho poznání o fungování příslušných mechanizmů těchto organismů, se nazývají kognitivní výpočetní systémy. Protože se jedná o systémy pracující na algoritmických principech, je možné je modelovat formálními výpočetními prostředky a následně studovat jejich výpočetní sílu a efektivitu. Příslušné tzv. kognitivní automaty jsou inspirovány modely běžně používanými ve výpočetní teorii složitosti, avšak liší se od nich zejména odlišným výpočetním scénářem. V nejjednodušším případě zahrnuje takový scénář nepredikovatelnou interakci modelu s pasivním či aktivním okolím (zejména s jinými kognitivními entitami), na kterou model reaguje učením, přizpůsobováním svého chování měnícím se požadavkům okolí a případně i modifikací svého okolí. Ve složitějších případech je možné uvažovat i evoluci příslušných modelů, jež vyvolává změnu jejich "architektury" a obecně vede k výkonnějším modelům. Jiným zajímavým případem jsou tzv. společenství kognitivních automatů. Příslušné výpočetní modely vedou k modelům relativně málo probádaných tzv. neuniformních interaktivních výpočetních systémů, jež v obecnosti vládnou super-turingovskou výpočetní sílou. To znamená, že tyto modely nelze simulovat pomocí běžného Turingova stroje a že v principu mohou řešit problémy běžným Turingovým strojem neřešitelné.

Keywords

Kognitivní systémy, Turingův stroj, neuniformní výpočetní složitost, evoluce

[1]Tato práce vznikla s částečnou podporou grantu GA ČR č. 201/00/1489

Superturingovský výpočetní potenciál kognitivních a evolučních systémů

Jiří Wiedermann[1]
Ústav informatiky AV ČR
Pod Vodárenskou věží 2, 182 07 Praha 8
Česká republika
jiri.wiedermann@cs.cas.cz

Abstrakt: Výpočetní systémy, jejichž chování modeluje prvky chování živých organismů a jejichž funkčnost je odvozena od našeho poznání o fungování příslušných mechanizmů těchto organismů, se nazývají kognitivní výpočetní systémy. Protože se jedná o systémy pracující na algoritmických principech, je možné je modelovat formálními výpočetními prostředky a následně studovat jejich výpočetní sílu a efektivitu. Příslušné tzv. kognitivní automaty jsou inspirovány modely běžně používanými ve výpočetní teorii složitosti, avšak liší se od nich zejména odlišným výpočetním scénářem. V nejjednodušším případě zahrnuje takový scénář nepredikovatelnou interakci modelu s pasivním či aktivním okolím (zejména s jinými kognitivními entitami), na kterou model reaguje učením, přizpůsobováním svého chování měnícím se požadavkům okolí a případně i modifikací svého okolí. Ve složitějších případech je možné uvažovat i evoluci příslušných modelů, jež vyvolává změnu jejich „architektury“ a obecně vede k výkonnějším modelům. Jiným zajímavým případem jsou tzv. společenství kognitivních automatů. Příslušné výpočetní modely vedou k modelům relativně málo probádaných tzv. neuniformních interaktivních výpočetních systémů, jež v obecnosti vládnou super-turingovskou výpočetní sílou. To znamená, že tyto modely nelze simulovat pomocí běžného Turingova stroje a že v principu mohou řešit problémy běžným Turingovým strojem neřešitelné.

1. Úvod

Pojem výpočtu je jedním z centrálních pojmů jak v kognitivní vědě, tak i v umělé inteligenci. V obou případech se výpočet používá k modelování (simulování, napodobování) kognitivních či dokonce mentálních procesů, které probíhají v živých organizmech. Cílem odpovídajícího výzkumu je návrh algoritmů či výpočetních zařízení, která by realizovala kognitivní procesy. Zejména v kognitivní filozofii existuje nepřeberné množství prací, zabývajících se vhodností či nevhodností takového přístupu pro daný účel (viz např. [5],[10],[19],[20],[26]). Jediným přesvědčivým argumentem zastánců (tzv. komputacionalistů či fundamentalistických umělých inteligentů) „dostatečnosti“ výpočtů pro daný účel by byl návrh a realizace nějakého artefaktu, který by simuloval některé netriviální aspekty kognice, nemluvě o inteligenci, pomocí výpočtů. To se zatím bohužel nikomu nepodařilo. Jediným přesvědčivým argumentem odpůrců předchozího přístupu by byla identifikace nějaké kognitivní vlastnosti, o které

[1] Tato práce vznikla s částečnou podporou grantu GA ČR č. 201/00/1489

by bylo možné dokázat, že ji nelze výpočetně realizovat. V posledním případě se často argumentuje Gödelovou větou [20], která rigorosně ukazuje limity dokazovací síly formálních systémů. Penrose [20] tvrdí, že tyto meze jsou však „překračovány“ např. matematiky, kteří občas ve svých důkazech postupují „nealgoritmicky“, když využívají svou „intuici“, „nevypočitatelnou informaci“. V tom vidí Penrose důkaz, že lidská mysl v některých případech předčí schopnosti Turingova stroje, jenž jako formální výpočetní zařízení je limitován Gödelovou větou. Dle Turingova životopisce A. Hodgese [14] si tuto skutečnost uvědomil již Turing, a proto koncem třicátých let „upgradoval“ svůj původní model [24] z r. 1936 na model, kterému v dnešní době říkáme Turingův stroj s nápovědou (oracle Turing machine) [25]. Tento model umožňoval modelovat „matematickou intuici“ tím způsobem, že při výpočtu bylo možné použít informaci, která nemusela nutně být algoritmického charakteru, tj. neexistoval Turingův stroj, který by danou informaci vypočítal. Později Turing tuto myšlenku opustil a až do konce svého života byl přesvědčen, že k modelování lidské inteligence Turingův stroj plně postačuje. Měl pravdu Turing, anebo má pravdu Penrose? Anebo, dokonce, může se stát, že pravdu nemá ani jeden z obou pánů?

Zdá se tudíž, že celý spor je možno alespoň zpočátku zúžit na otázku, jestli pro modelování kognice v principu „stačí“ Turingův stroj (Turingův názor), anebo je potřebný výkonnější výpočetní mechanismus, představovaný např. Turingovým strojem s nápovědou (subsumuje Penroseův názor). Jak vidno, k řešení sporu může přispět i informatika, jejíž cílem je studium výpočetních možností a limitů různých výpočetních zařízení. Odpověď však není jednoduchá, protože na jedné straně v každém případě jsou na první pohled komplikované živé organismy, a na druhé straně jednoduché, i když výpočetně různě silné formální modely. Odpověď tudíž bude plně závist na tom, jak „věrný“ základní výpočetní model kognitivních procesů živých organismů se nám podaří nalézt. Je zřejmé, že pokud zvolíme příliš jednoduchý výpočetní model, dopustíme se zjednodušení, která povedou k podcenění výpočetního potenciálu přirozených kognitivních systémů. Pokud zvolíme příliš silný model, dostaneme se do opačného extrému.

Algoritmické modelování kognitivních procesů živých organismů je pro informatiku relativně mladá záležitost. Kromě nepočetné skupiny iniciálních prací, zabývajících se převážně modelováním mozku, jako např. [9], [12] nebo [32], nenalezneme mnoho dalších prací zabývajících se výlučně problematikou kognitivních výpočtů (podrobnější přehled nedávného stavu viz v práci [1],[35] a [37]). Nicméně, v teoretické informatice existuje velké množství odpovídajících znalostí a zkušeností, kterých lze pro studium kognitivních systémů využít. Na jedné straně je to zejména dobře rozvinutá a bohatá teorie formálních výpočetních modelů, sahající od kombinatorických obvodů a konečných automatů přes Turingovy stroje (viz např. [15]) až po výpočetně vysoce efektivní modely, jakými jsou např. neuronové sítě [22] či kvantové počítače. Studovány byly i modely výpočetně silnější než Turingovy stroje (např. Turingovy stroje s nápovědou, neuniformní množiny obvodů [2], diskrétní a analogové neuronové sítě [1], [21], Turingovy stroje pracující s reálnými čísly [4] apod.). Na druhé straně je zde relativně bohatá výpočetní teorie učení, zabývající se modelováním a složitostními aspekty učení pomocí omezených výpočetních modelů a v rozličných situacích. Na základě uvedené znalostní báze a využitím odpovídajících znalostí z oblasti umělé inteligence, psychologie, neurovědy a kognitivní vědy si teoretická informatika v současné době troufá použitím svých vlastních paradigmat a své vlastní metodologie přispět k odhalování algoritmických principů kognice a k charakterizaci její výpočetní síly.

Cílem tohoto článku je podat argumenty, které se snaží uvést záležitosti týkající se výpočetní síly kognitivních systémů „na pravou míru" použitím nástrojů teoretické informatiky, zejména pomocí aparátu tzv. teorie neuniformní výpočetní složitosti. Přitom budeme vycházet z velmi slabého, a proto přijatelného předpokladu, že kognitivní činnost organizmů je možné a stačí simulovat pomocí konečně-stavových výpočetních zařízení, která se mohou nacházet pouze v konečném počtu konfigurací. Paradigmatickým příkladem takového zařízení je konečný automat. Výsledkem našich úvah bude pro kognitivní vědu asi poměrně překvapivé tvrzení, že za jistých předpokladů množiny takových automatů mohou skutečně jako celek nabýt superturingovské výpočetní síly, předčící sílu jakéhokoliv Turingova stroje.

Charakterizace výpočetní síly jakéhokoliv výpočetního modelu je důležitá z hlediska zařazení daného modelu do systému známých výpočetních modelů. Modely stejné výpočetní síly mohou být navzájem zaměňovány a výsledky získané pro jeden z nich mohou být „přenášeny" na jakýkoliv výpočetně ekvivalentní model. Je zajímavé, že úvahy o výpočetní síle formálních modelů lze provádět pouze na základě detailní znalosti daného modelu (tj. zhruba na základě znalosti jeho „fungování" v konečném počtu různých situací) a odpovídajícího scénáře interakce (tj. na základě znalostí, jaká data se principiálně mohou objevit na vstupu zařízení a jestli a jak tato data závisí na předchozích výstupech sytému). Není nutné znát konkrétní algoritmy, pomocí kterých daný model řeší konkrétní „praktické" problémy; dokonce není nutná ani znalost specifikace oněch problémů. Pokud je model dobře navržen, výsledné tvrzení o principiální výpočetní síle daného kognitivního výpočetního modelu vlastně udává horní odhad výpočetní síly zařízení, které modelujeme. V praxi, tj. v běžných podmínkách, tuto výpočetní sílu ani nemusíme pozorovat. To může být způsobeno např. tím, že model má tuto výpočetní sílu sice k dispozici, ale běžně ji nevyužívá (příkladem budiž „nadlidské" aritmetické schopnosti některých jedinců, kteří zpaměti dovedou násobit či odmocňovat velká čísla). Jiná možnost je, že pozorujeme pouze krátký úsek činnosti systému, případně pouze činnost části systému, ze které není zřejmá velká výpočetní síla systému jako celku či jeho síla po dobu jeho celé existence. Typickým příkladem je internet či přirozené evoluční systémy (např. evoluční vývoj živých organismů), které pozorujeme pouze lokálně a v omezeném časovém úseku. Superturingovská výpočetní síla takových systémů byla dokázána pouze v nedávné době [30]. Posledně zmíněná práce se zabývá zejména modelováním internetu a podobných umělých výpočetních systémů. Možnost aplikace podobných paradigmat na živé, přirozené (kognitivní, evoluční) systémy je v uvedených pracích naznačena, avšak nikoliv podrobněji sledována.

Zčásti inspirováni výsledky již zmíněné práce [30] se budeme v této práci podrobněji zabývat studiem výpočetní síly modelů přirozených kognitivních systémů. Ve druhé části práce osvětlíme podrobněji základní komputacionalistické stanovisko, na kterém stojí naše víra v oprávněnost uvažovaných modelů a ze které dále vyplývají naše úvahy o výpočetní síle kognitivních výpočetních modelů. Zde zavedeme i klíčový pojem kognitivního automatu jako diskrétního výpočetního zařízení, které může nabývat pouze konečného počtu různých konfigurací. Dále budeme studovat vybrané výpočetní modely kognitivních systémů. V třetí části to budou zejména méně známé neuroidní sítě, které představují velmi přesvědčivý model výpočetní kognice a fundamentální typ kognitivního automatu. Dále ve 4. části uvedeme stručně základní pojmy z teorie neuniformní výpočetní složitosti. V 5. části se budeme zabývat studiem výpočetní síly posloupností kognitivních automatů, které zachycují výpočetní sílu neuniformní evoluce. V 6. části ukážeme výpočetní sílu kognitivních robotů, tj. automatů, které algoritmickým způsobem mění své okolí a v 7. části výpočetní sílu společenstev kognitivních robotů Osmá část obsahuje diskusi a další interpretace prezentovaných výsledků o výpočetním potenciálu kognitivních systémů.

Všechny shora uvedené výsledky vycházejí ze známých (a většinou zbrusu nových, původních) výsledků teorie neuniformní výpočetní složitosti. Práce otevírá novou aplikační oblast pro tuto teorii tím způsobem, že upozorňuje na možnou interpretaci výsledků teorie výpočetní složitosti a neurovýpočtů v termínech kognitivních a evolučních systémů. Uvedené výsledky dovolují charakterizovat výpočetní sílu jak „jedince", viděného jako kognitivní automat, tak i evoluční řady jedinců, dále společenstev jedinců stejného typu, a v neposlední řadě jedince vybaveného externí pamětí. V některých případech tak dostáváme ekvivalent výpočetních zařízení vládnoucích větší výpočetní silou než Turingovy stroje - a to je zřejmě překvapivý výsledek.

Pro pochopení textu stačí základní znalosti z oblasti výpočetní složitosti, zejména problematiky konečných automatů, regulárních jazyků, Turingových strojů, dále základy teorie vyčíslitelnosti (rozhodnutelné a nerozhodnutelné problémy) a povědomí o třídách složitosti v rozsahu úvodního vysokoškolského kurzu.

2. Hledání základního modelu

Výpočetní model jakéhokoliv kognitivního systému vychází z tzv. komputacionalistického názoru, jenž věří, že výpočetní modelování kognitivních schopností je alespoň v principu možné a že tímto způsobem lze dosáhnout ne-li věrného modelování všech mentálních činností (včetně myšlení a vědomí), tak přinejmenším jejich věrné aproximace a také podání jejich vysvětlení.

Jako motivaci další diskuse uvažujme následující „myšlenkový pokus": představme si, že se rozhodneme počítat od jedné a nahlas vyslovovat příslušné číslovky. Použití jakýchkoliv dalších pomůcek, kromě naší „hlavy", je samozřejmě zakázané. Pravděpodobně nikdo nepochybuje, že dříve nebo později každý z nás dojde k tak velkým číslům, která si nebude schopen zapamatovat a počítaní bude muset ukončit, anebo bude vyslovovat nesprávná čísla. Říká takový pokus něco o našich výpočetních (nikoliv počtářských!) limitech, anebo je zcela irelevantní? Je nemožnost pokračovat libovolně dlouho v počítání daná nějakým principiálním omezením naší mysli, anebo vyplývá pouze z naší „nedostatečné motivace“ pro takovou nudnou činnost? Každý z nás asi cítí, že správný je první důvod.

Zde jsme totiž narazili na zásadní problém, týkajícího výběru vhodného výpočetního zařízení pro modelování kognitivní činnosti. Tím problémem je otázka, jestli takové zařízení má mít apriorně danou schopnost nabývat ohraničeného anebo neohraničeného počtu konfigurací. Jinými slovy, v termínech výpočetních modelů se ptáme, jestli pro daný účel „stačí“ konečný automat, anebo je potřebný Turingův stroj, případně ještě nějaký „silnější“ výpočetní model. Veškerá zkušenost napovídá, a dokonce pomocí elementárních fyzikálních úvah lze dokázat, že žádné zařízení konečných rozměrů ani žádný živý organizmus nemůže nabývat neohraničeně mnoha různých (tj. rozlišitelných) konfigurací. Takové úvahy naznačují, že odpovídající zařízení by tudíž měla být ekvivalentní konečným automatům (ale nemusí to nutně být právě konečné automaty). Bez toho, že bychom zatím takové zařízení blíže specifikovali, budeme ho nazývat kognitivním automatem. Pro toto zařízení bude charakteristické, že může nabývat pouze konečný počet konfigurací. Mimochodem, předchozí „počítací problém“ je přesně jedna z věcí, kterou žádný konečný automat či kognitivní automat nezvládne.

To poněkud nesouhlasí s osobním Turingovým přesvědčením (viz např. [26]), že „jeho“ stroj (který obecně považoval za „zařízení s diskrétními stavy“) je vhodným nástrojem pro modelování činnosti mysli. Na druhé straně, podle [3] Gödel měl námitky k Turingovým pokusům vyjádřit schopnosti lidské mysli svým idealizovaným strojem z toho důvodu, že „*Turing naprosto nebere v úvahu, že mysl, když se používá, není statická, ale se vyvíjí*“. I když lidská mysl nabývá v každém okamžiku pouze konečného počtu vnitřních stavů, „není důvodu, proč by počet stavů neměl v průběhu jejího vývoje konvergovat do nekonečna…“. Z hlediska teorie automatů se zdá, že oba pánové nebyli ve svých názorech ve velkém rozporu. Pomineme-li nepřesný termín „konvergovat do nekonečna“, rozpor byl pouze v tom, že pojem stav zařízení lze použít ve dvou významech. Zatímco Turing si představoval svůj stroj, Gödel (v dnešních termínech) argumentoval proti konečnému automatu. Obě zařízená jsou diskrétní a konečně stavová, ale Turingův stroj obecně může nabývat postupně neohraničeně mnoho paměťových konfigurací, kdežto konečný automat nikoliv. Takže pokud stav u Gödela ztotožníme s paměťovými konfiguracemi, jeho námitka padá. Zůstává zde však rozpor v názorech obou pánů s fyzikální realitou. Z tohoto pohledu se zdá, že Turingův stroj je sice postačujícím, ale ne nutným zařízením pro modelování kognitivních činností. Pro takovou aplikaci je dokonce příliš silným zařízením!

Zde je na místě si uvědomit zásadní rozdíl mezi požadavky, které klademe na výpočetní modelování, pokud chceme modelovat a studovat výpočetní sílu nějakého systému, a mezi těmi, pokud chceme modelovat chováni či činnost takového systému. V prvním případě by totiž zvolení výpočetně silnějšího modelu vedlo k nesprávným závěrům o výpočetní síle modelovaného systému. Ve druhém případě nijak zásadně nevadí, pokud modelujeme „slabší“ výpočetní zařízení pomocí „silnějšího“ - naopak, v takovém případě se může řešení problému simulace dokonce ulehčit.

Zkusme se vrátit k živým „vzorům“ kognitivních systémů a ustupme (prozatím) od ambiciózních požadavků modelovat (lidskou) mysl. Bezesporu, naše formální zařízení by měla být schopna modelovat jak kognitivní činnost jednobuněčných živočichů (např. trepky), tak i kognitivní činnost řekněme medúzy, mouchy, holuba či psa. Ve všech těchto i dalších případech by mělo jít o jeden typ zařízení. Pokud bychom zůstali u původní myšlenky konečných automatů a jim ekvivalentních systémů, tak kognitivní činnost každého z uvedených tvorů by byla simulována kognitivním automatem. Tento automat by však nebyl ve všech případech stejný, ale kromě jiného by se lišil počtem dosažitelných konfigurací, který by rostl s „náročností“ úkolu, tj. zhruba s „inteligencí“ simulovaných tvorů. Takže se vynořuje představa jakési posloupnosti konečných automatů - čím složitější tvor, tím větší, složitější automat. Bezpochyby, konečný automat simulující kognitivní činnost člověka bude ohromný: mozek člověka obsahuje řádově 10^{12} neuronů, což za předpokladu, že každý z nich může být v konečném počtu rozlišitelných stavů, vede k astronomickému odhadu počtu možných konfigurací velikosti konstanta umocněna na 10^{12}. Je vůbec vhodné takové zařízení, jakým je mozek, modelovat konečným automatem? Nebylo by přece jenom přirozenější modelovat mozek pomocí Turingova stroje? Pokud nám jde o hledání výpočetních mezí mozku, odpověď nepochybně zní, ne. I když mozek člověka je veliký, má bezesporu meze své paměťové kapacity. Lépe je to pravděpodobně vidět na menších, dříve zmíněných „jednodušších“ živočiších. Není rozumné předpokládat, že např. kognitivní činnost měkkýšů bychom ještě měli modelovat konečným automatem, ale počínaje ptáky již určitě Turingovým strojem.

To vše zatím vypadá jako dobrý argument, že pro modelování evoluce se hodí posloupnost konečných automatů rostoucí velikosti. Někdo by ale stále mohl namítat, že pro modelování vybraných „vyšších“ druhů živočichů není chybou používat Turingův stroj. Vždyť i existující počítače, které jsou v principu zařízeními nabývajícími pouze konečného počtu konfigurací, modelujeme pomocí Turingových strojů! Je zde nějaký rozdíl? Dle našeho názoru, ano. Pokud modelujeme počítače, které jsou ve skutečnosti „velikými“ konečnými automaty, pomocí Turingova stroje, tak mlčky předpokládáme, že v případě potřeby výpočtu (např. pokud požaduje více paměti než je k dispozici), můžeme tuto paměť „přidat“ (což v praxi často činíme). U živých tvorů nic takového nepřipadá v úvahu. Dalším argumentem proti použití Turingových strojů za účelem posouzení výpočetní síly reálných kognitivních systémů je fakt, který jsme zmínili již dříve: totiž, že vzhledem k jejich větší výpočetní síle, než je nutné, bychom nutně dospěli k nesprávným závěrům týkajících se výpočetní síly kognitivních systémů.

Další aspekt, který musíme zohlednit u výpočetního modelování kognitivních procesů, je zásadní rozdíl mezi standardním scénářem výpočtu konečného automatu či Turingova stroje a scénářem „výpočtů“ kognitivních systému. V prvním případě předpokládáme konečný vstup, který je k dispozici již před započetím vlastního výpočtu. Jakmile se výpočet rozběhne, již nemáme možnost přidávat další vstupy. Navíc, pokud výpočet skončí a chceme zpracovat další vstup, konečný automat i Turingův stroj začne opět pracovat ve stejné konfiguraci jako předtím. Žádným způsobem není možné přenášet informaci z minulého výpočtu do příštího. Příslušné zařízení se vlastně nemůže učit!

Kognitivní systémy zpracovávají vstupy dodávané jejich periferními zařízeními nepřetržitě. Vstupy se „objevují“ on-line, nepředvídatelně, a většinou je třeba je zpracovat v reálném čase (to je nutná podmínka pro vznik alespoň rudimentální formy vědomí - viz diskusi k otázce úlohy vědomí v 8. části). Příslušné výpočty v principu nikdy nekončí, resp. jsou omezeny pouze životností systému. Dále, vstupy proudí do kognitivních systémů zpravidla paralelně, přes množství kanálů a kognitivní systémy je zpracovávají paralelně. Nelze se vrátit ke vstupu již přečtenému. Počet vstupů také může záviset na komplexnosti (či velikosti) systému (například medúza zajisté dostává od svých senzorů méně informací než delfín). Navíc, zejména vstupy složitějších kognitivních systémů, které modifikují nějakým způsobem své okolí či s ním komunikují, mohou záviset na předchozích akcích či výstupech z těchto systémů. Tím je daná těmto systémům potenciální možnost učení se z vlastních chyb. Příslušný výpočetní scénář je tudíž scénářem nikdy nekončícího interaktivního učení.

Dále se vtírá následující otázka: jak je možné realizovat scénář nikdy nekončících interaktivních výpočtů pomocí zařízení ekvivalentních konečným automatům? Vždyť o konečných automatech se většinou uvažuje jako o rozpoznávačích jazyků (akceptorech), tj. jsou považovány za zařízení schopné v nejlepším případě klasifikovat konečné řetězce symbolů do dvou tříd: „přijaté“ a „zamítnuté“ (viz. např. [15]). My však od nich chceme nejenom schopnost zpracovat nekonečnou paralelní posloupnost informací, ale navíc i učení.

Co se týká problému paralelních vstupů, odpověď je jednoduchá: jednoduše všechny vstupy, čtené současně, zakódujeme do jednoho symbolu větší (ale stále konečné) abecedy. Problém se zpracováním nekonečného proudu informací má také řešení: automat již nebude pracovat pouze jako rozpoznávač (konečných slov) jazyka, ale současně jako překladač (transducer). Formálně to znamená, že nekonečný vstupný řetězec bude automatem transformován, přeložen, na podobný výstupní řetězec. Pokud se v nekonečném řetězci vstupů vyskytnou nějaká „zajímavá“ konečná slova, automat je může rozpoznat a reagovat na jejich výskyt vydáním příslušného výstupního symbolu či konečného řetězce symbolů (podobně jako Mealyho automat). Případná závislost vstupů na předchozích výstupech se dá jednoduše „promítnout“ do následujících vstupů (to není problém našeho automatu, ale jeho „okolí“, se kterým interaguje). Tím automat získá vlastnosti interaktivnosti. Problém s učením se zdá být složitější: vždyť činnost konečného automatu je předem popsána pomocí přechodové funkce, kterou nelze v průběhu výpočtu měnit, a tudíž (zdánlivě, jak uvidíme), se takový automat nemůže nic naučit. Nicméně, i zde je řešení možné. Pevně zadaná přechodová funkce bude realizovat předem zadaný učící se algoritmus. Tento algoritmus se lépe popisuje ve formalismu příbuzném neuronovým sítím a budeme se mu věnovat v další sekci. Ekvivalence těchto sítí s konečnými automaty dokazuje správnost předchozího tvrzení týkajícího se možnosti učení konečných automatů[2] (viz větu 1).

---

[2] Abychom alespoň naznačili možnost učení konečných automatů, ukážeme princip „zapamatování si“ určité posloupnosti dvou symbolů. Předpokládejme, že automat bude sledovat ve stavu *r* výskyt dvou po

Celkově to znamená, že běžnou učebnicovou představu o konečném automatu jako příkladu kognitvního automatu je třeba značně modifikovat. Nejedná se o hračičkářská zařízení, která pracují nad několika málo symboly a která pomocí pár stavů rozeznávají regulární jazyky typu „*a* následované *b* a pak cokoliv anebo pouze *a*“. Naopak, musíme si představit automat s obrovským počtem stavů, s obrovskou vstupní abecedou, s obrovskou přechodovou funkcí, která je zhruba tak velká jako je počet stavů a tento automat nerozeznává jednotlivá slova, která mohou být podobně jako v učebnicových příkladech libovolně dlouhá, ale rozeznává on-line způsobem v reálném čase potenciálně nekonečné posloupnosti krátkých slov, která se většinou skládají pouze z jednoho symbolu, v němž je zakódována veškerá informace „vrácená“ senzory systému.

V pracích [9],[12],[22],[35] a [36] jsou ukázána vhodnější formální zařízení (např. neuronové sítě, neuromaty, neuroidy, kogitoidy), než je konečný automat, která se hodí pro zachycení podobných učících se algoritmů a která nicméně jsou stále ekvivalentní konečným automatům. Učící se program, který řídí jejich činnost a který odpovídá přechodové funkci ekvivalentního konečného automatu, je však popsán daleko jednodušeji než v případě konečného automatu. Dále, v [32] a [36] je naznačeno, že pravděpodobně existuje pouze konečný počet pravidel, pomocí kterých se vytvářejí důmyslné datové struktury síťového typu, ve kterých je zachycena veškerá znalost automatu, získaná v procesu interakce s okolím. Tato pravidla „stačí“ k tomu, aby se automat byl schopen naučit nejen základní vzory kognitivního chování, ale rozvinout i takové náročné kognitivní činnosti, jako je myšlení či vědomí [39]. Přitom se stále využívá možnosti konečných automatů pamatovat si ve svých stavech konečně mnoho (bitů) informací. Samozřejmě, pořád je předem potřebné vědět, jaké informace se potenciálně mohou objevit na vstupu a být připraven nejen si zapamatovat jejich podstatné charakteristiky, ale i zachytit jejich všechny možné (zajímavé či pravděpodobné) kombinace. Takových možností je ale konečný – i když většinou obrovský – počet. Nezapomínejme, že v každém okamžiku může být na vstupu informace, která je v daném okamžiku rozeznávána všemi senzory systému, a těch může být několik (podobně jako smyslů) a každý z nich může „paralelně“ dodat obrovské množství bitů. To dle našeho názoru vysvětluje, proč i přirozené kognitivní systémy potřebují potenciálně obrovský konfigurační prostor.

---

sobě jdoucích symbolů – řekněme *a* následované *b* - a poté přejde do stavu $q_{r,a\to b}$ znamenajícího „ve stavu *r* bylo *a* následováno *b*“. Tento stav bude dále superponován na všechny další stavy (tj. bude v nich „zapamatován“). Dále bude v přechodové funkci přechod ze stavu *r* se zapamatovaným stavem $q_{r,a\to b}$ v situaci, kdy je na vstupu symbol *a*, do stavu $q_{r,a\to b}$.V budoucnosti, kdykoliv bude automat v právě popsaném superponovaném stavu *r* a bude číst symbol *a*, může vydat hlášení „*jako další vstup očekávám b*“. Jinými slovy to znamená, že automat se „naučil“ reagovat na výskyt symbolu a v jistém stavu „předvídáním“ symbolu *b*. Uvedený způsob je možné rozšířit i na další podobné akce - např. je možné naučit se takto reagovat pouze v případě, že symbol *b* bude vzápětí „potvrzen“ dalším symbolem, znamenajícím „odměnu“ (odpovídá formě učení pomocí odměn a trestu), apod.

Naštěstí, popisná složitost zařízení (např. neuromatů), funkčně odpovídajících konečným automatům, může být až exponenciálně nižší než je počet stavů daného automatu. To ostatně uvidíme v 5. části.

Závěrem této sekce shrňme naše poznatky. Kognitivní systémy lze modelovat pomocí vhodně naprogramovaných interaktivních zařízení, která mohou nabývat pouze konečného počtu konfigurací. Odpovídající zařízení budeme nazývat kognitivní automaty. Pokud chceme podobnými prostředky zachytit i evoluci kognitivních automatů, potřebujeme přejít k posloupnostem takových automatů. V další sekci popíšeme konkrétnější model kognitivního automatu, tzv. neuroidní síť, která dle našeho mínění nejlépe odpovídá intuitivním představám o kognitivních automatech, a to jak z hlediska jejich funkcionality, tak i architektury.

3. Neuroidní síť

Pojem neuroidu pochází od L. Valianta z r. 1988 [31]. Zhruba řečeno, jedná se o „programovatelný" neuron, který může v závislosti na svých momentálních vstupech a svém stavu měnit předepsaným způsobem své výpočetní charakteristiky. Propojením neuroidů dostaneme neuroidní síť, která díky možnosti změny svých výpočetních vlastností při vhodném naprogramování může mít schopnost učení.

Formálně vypadá neuroid jako konečný automat, který má namísto vstupní pásky k vstupních portů, pro nějaké $k \geq 1$. Výstup z neuroidu je posílán na výstupní port. Přes vstupní porty neuroid paralelně, v jednom kroku, čte svá vstupní data – k-tice tvaru $(x_1, x_2, \ldots, x_k)$ skládající se z nul a jedniček. Každému portu je přiřazena jeho váha – celé číslo z pevně zadané konečné množiny **W**. Vážená suma $\sum w_i x_i$, kde w_i je váha i-tého portu, se nazývá excitace neuroidu. Dále je neuroid popsán konečnou množinou **Q** svých stavů a konečnou množinou **T** prahových hodnot (také množina celých čísel). Chování neuroidu popisují dvě pravidla:

- pravidlo *excitační*, které říká, že neuroid bude aktivní, tj. vyšle na svůj výstupní port jedničku, právě když jeho momentální excitace dosáhne či převýší jeho momentální prahovou hodnotu;

- pravidlo *změny parametrů*, které jsou zadáno pomocí přechodové funkce. Tato funkce, na základě daných vstupů, daných vah, dané prahové hodnoty a daného stavu přiřadí vahám nové hodnoty z množiny **W**, nový práh z množiny **T** a nový stav z množiny **Q**.

Neuroidní síť dostaneme, jestliže propojíme konečný počet neuroidů tím způsobem, že výstupy z některých neuroidů budou tvořit vstupy do jiných neuroidů. Topologii výsledné sítě tudíž popisuje orientovaný graf, jehož vrcholy tvoří neuroidy a hrany propojení mezi nimi. Obecně může tento graf obsahovat cykly. Vstupní porty neuroidů, na které nejsou napojené žádné výstupní porty, tvoří množinu vstupních portů sítě a výstupní porty neuroidů, na které nejsou napojené žádné vstupní porty, množinu výstupních portů sítě. Velikost sítě je daná počtem jejích neuroidů.

Síť počítá synchronně, tzn., že všechny neuroidu čtou paralelně své vstupy. Dále se na všechny neuroidy v jednom výpočetním kroku současně uplatní obě pravidla, popisující jejich chování. Před započetím výpočtu jsou parametry všech neuroidů nastaveny na předepsané hodnoty. Nechť $\Phi : \{\{0,1\}^m\}^\omega \rightarrow \{\{0,1\}^n\}^\omega$ je zobrazení, které transformuje nekonečnou posloupnost booleovských m-tic na podobnou posloupnost n-tic. Jestliže neuroidní síť má *m* vstupních a *n* výstupních portů, pak je zřejmé, že neuroidní síť realizuje zobrazení shora uvedeného typu. Standardní neuronovou síť dostaneme jako speciální případ neuroidní sítě, u které je zadaná „identická" přechodová funkce (která nemění žádné parametry neuroidů). Z výsledků práce [38] a z výsledků z oblasti booleovské složitosti vyplývá následující Věta charakterizující výpočetní sílu základních typů kognitivních automatů:

Věta 1: Nechť $\Phi : \{\{0,1\}^m\}^\omega \rightarrow \{\{0,1\}^n\}^\omega$ je zobrazení. Potom následující tvrzení jsou ekvivalentní:

- Φ je realizováno neuroidní síti;
- Φ je realizováno neuronovou sítí;
- Φ je realizováno konečným automatem

Pokud Φ je realizováno některým se shora uvedených zařízení, budeme říkat, že Φ je *regulární překlad* (ve shodě s faktem, že konečné automaty akceptují regulární jazyky).

Jinými slovy, k dané neuroidní síti existuje ekvivalentní standardní, binární neuronová síť a ekvivalentní konečný automat. Přitom neuronová síť se oproti neuroidní síti zvětší lineárně, v závislosti na počtu všech možných konfigurací všech neuroidů sítě (každou takovou konfiguraci je potřebné reprezentovat zvláštním neuronem). Konečný automat bude mít tolik stavů, kolik je možných konfigurací neuroidní sítě. To znamená, že v obecném případě může být konečný automat až exponenciálně větší než daná ekvivalentní neuroidní síť. Nicméně, tyto výsledky znamenají, že všechny čtyři výsledné systémy mohou realizovat kognitivní automat, byť s různou efektivitou.

V pracích [32] a [36] je naznačeno, jaké základní operace musí být každý kognitivní automat schopen vykonávat, aby byly vytvořeny předpoklady pro rozvoj kognitivních schopností pomocí učení. Jak jsme zmínili v úvodu, pro charakterizaci výpočetní síly odpovídajících kognitivních automatů však není podstatné, jakým způsobem tato zařízení řeší konkrétní úkoly, a tedy i kognitivní úkoly. Proto v další kapitole přikročíme ke zkoumání výpočetní síly kognitivních automatů bez toho, že bychom se podrobněji zabývali algoritmy, jakými tyto sítě mohou kognitivní úkoly řešit.

4. Neuniformní modely

Věta 1 charakterizuje výpočetní sílu kognitivních automatů jako sílu ekvivalentní konečnému (Mealyho) automatu, který realizuje regulární translaci nad nekonečně dlouhými slovy. Tato charakteristika však nezachycuje velikost uvažovaného zařízení. Takovou charakteristiku lze získat, jestliže uvažujeme posloupnosti kognitivních automatů stejného typu, ale rostoucí velikosti:

Definice 1: Posloupnost $F_p=\{K_1,K_2,..K_i....|\ size(K_i)\leq p(i)\}$ se nazývá posloupností neuniformních kognitivních automatů polynomické velikosti $p(i)$.

Název „neuniformní" je zvolen proto, že v obecném případě neexistuje algoritmus, jenž by pro dané i generoval popis automatu K_i velikosti $size(K_i)$. Jinými slovy, neexistuje uniformní, jednotná možnost, jak konečným způsobem popsat nekonečnou posloupnost automatů. Intuitivně to znamená, že jiný způsob než vyjmenování všech členů posloupnosti neexistuje.

Definice 2: Překlad realizovaný posloupností F_p je množina dvojic $T(F_p) =$
$= \bigcup_{m\geq 0}\{(x,y)|\ x\in\{0,1\}^m$ a existuje y takové, že automat $K_m\in F_p$ se vstupem x vygeneruje výstup $y\}$

Všimněme si, že v definici předcházejícího překladu jsme omezili každý automat na zpracování vstupů jisté délky. Třídu překladů realizovanou neuniformní posloupností neuroidních či neuronových sítí polynomické velikosti označíme POLY-NN; podobnou třídu pro konečné automaty označíme POLY-FA.

V teorii výpočetní složitosti se výpočetní síla takových posloupností konečných zařízení charakterizuje pomocí Turingových strojů s rádcem (tzv. advice Turing machines), které v r. 1980 definoval Karp s Liptonem [16]. Rada je speciálním druhem orákula (nápovědy, věštby). Turingův stroj s nápovědou (orákulem) byl původně definován Turingem v r. 1939 [25] jako rozšíření jeho původního modelu z r. 1936 [24]. Tento „silnější" model umožňoval zachytit „matematickou intuici" jako nevypočitatelnou informaci, která se může také podílet na výpočtu. Z praktického hlediska nápověda umožňuje vložení externí informace do již rozběhnutého výpočtu. Tato informace může záviset na daném vstupu a je poskytována zdarma. Tímto může odpovídající stroj získat superturingovskou výpočetní sílu, to znamená, že taková zařízení mohou řešit algoritmicky nerozhodnutelné problémy (viz např. [2]). Rozdíl mezi radou a nápovědou spočívá v jejich použitelnosti: nápověda je použitelná pouze pro daný vstup (pro jiný vstup potřebujeme jinou nápovědu), kdežto rada je použitelná pro všechny vstupy dané velikosti (protože je pro všechny takové vstupy stejná). To znamená, že rada je v jistém smyslu obecnější než nápověda. Protože různých vstupů velikosti m je 2^m, je v principu možné vytvořit radu pro vstupy velikosti m „spojením" všech nápověd pro všechny možné vstupy dané velikosti. Abychom tomu zamezili, uvažují se rady menší než exponenciální velikosti.

Definice 3: *Rádce* je funkce $f\colon Z^{+}\rightarrow\{0,1\}^{*}$. Hodnota funkce f se nazývá rada. Říkáme, že rada je $S(m)$ ohraničená, jestliže pro všechna m platí, že délka zápisu funkční hodnoty $f(m)$ je ohraničená hodnotou $S(m)$.

Formálně pracuje *Turingův stroj s rádcem* definovaným funkcí f a se vstupem velikosti m podobně jako běžný Turingův stroj. Navíc však tento stroj může žádat radu příslušející vstupu velikosti m tím způsobem, že vstoupí do speciálního, tzv. *tázacího stavu*. Vzápětí se na speciální tzv. *nápovědní pásce* stroje „objeví" příslušná rada $f(m)$. Od té chvíle může stroj ve svých výpočtech používat i informace z nápovědní pásky. Pro třídu překladů realizovaných Turingovým strojem s rádcem budeme používat standardní notaci uniformní teorie složitosti (viz např. [2]).

Definice 4: Třída překladů C/F obsahuje překlady P pro které existuje překlad $P_1 \in C$ a rádce $f \in F$ tak, že pro všechna m a vstupy x délky m platí $(x,y) \in P$ právě když $\langle (x,y),f(m) \rangle \in P_1$, přičemž poslední výraz v lomených závorkách označuje konkatenaci v něm uvedených řetězců.

To znamená, že $P \in C/F$ právě když překlad P je realizován Turingovým strojem se složitostní třídy C s rádcem $f \in F$. Na místě třídy překladů C se běžně uvažuje LOGSPACE („deterministický logaritmický prostor"), PTIME („deterministický polynomický čas"), atd. Jako třída rádců se obyčejně používá l*og* , tj. třída logaritmicky omezených rad, anebo *poly*, třída polynomicky omezených rad.

5. Výpočetní síla posloupností kognitivních automatů

V teorii neuniformní výpočetní složitosti, resp. v teorii neurovýpočtů (viz např. [22]) se dokazují následující tvrzení:

Věta 2: Pro třídy neuniformních překladů platí:
- POLY-NN=PSPACE/*poly,*
- LOGSPACE-NN=LOGSPACE/*log*
- POLY-FA=LOGSPACE/*poly.*

Jak jsme naznačili ve druhé kapitole, pomocí posloupnosti kognitivních automatů můžeme modelovat evoluční posloupnosti odpovídajících kognitivních automatů (např. takové, které vyplývají z evolučních stromů, jaké se používají biologii). Z důkazů výše uvedeného tvrzení bychom viděli, že odpovídající Turingův stroj s rádcem potřebuje pro simulaci jakéhokoliv kognitivního automatu dané velikosti znát jeho popis; tento mu poskytne jeho rádce. Opačná simulace vyplývá z toho, že Turingův stroj s rádcem pro vstupy fixní velikosti používá tutéž radu a pracuje v ohraničené paměti. Tudíž je možné ho simulovat pomocí (konečného) kognitivního automatu. Jelikož pro každou délku vstupu máme jinou radu, Turingův stroj s radou „jako celek" může být simulován pomocí nekonečné posloupnosti neuniformních kognitivních automatů. Všimněme si, že v případě konečných automatů má automat ekvivalentní logaritmicky prostorově ohraničenému Turingovu stroji s polynomickou radou exponenciální počet stavů. Tyto výsledky např. znamenají, že při zachování stejného kognitivního potenciálu jsou neuronové sítě z hlediska ekonomie jejich popisu efektivnějším zařízením než konečné automaty.

Dále je vhodné si uvědomit, že uvedené výsledky znamenají, že posloupnosti kognitivních automatů vládnou superturingovskou výpočetní silou. Vyplývá to z toho, že k jejich simulaci potřebujeme stroje s rádcem, v jehož radách může být zakódována nerekurzivní (tj. nevypočitatelná) informace.

6. Od kognitivních automatů ke kognitivním Turingovým strojům

V úvodě naší práce jsme argumentovali, že Turingův stroj je příliš silným nástrojem pro věrné zachycení výpočetního kognitivního potenciálu živých organismů. Scénář našeho modelu výpočtu sice zahrnoval interakci s okolím, avšak explicitně nepředpokládal, že kognitivní automaty mohou měnit okolí. To odpovídá jednodušším živočichům (trepka, medúza), anebo situacím, kdy živočichové nějakým způsobem komunikují (např. akusticky) bez toho, že by po komunikaci někde zůstala nějaká informace, kterou by bylo možné opětovně přečíst. V reálném životě znají biologové množství živočichů, kteří nějakým způsobem značkují své okolí a tyto značky jim slouží minimálně k orientaci, ale často i k výměně informací (jako v případě lidí) [10], [11]. Dovedeme i tuto situaci zachytit v našem modelu kognitivních automatů?

Uvažujme opět náš motivační problém neomezeného počítání, ale tentokrát povolme používání tužky a papíru. Problém se tak stane směšně jednoduchým, protože teď vlastně stačí na papír zapisovat číselnou řadu, počínaje od jedničky. Právě dosažené číslo není nutné si pamatovat – vždyť je zapsané na papíře. Vše, co musíme umět, je přičíst jedničku k danému napsanému číslu. Příslušný algoritmus se učíme v první třídě.

Abychom podobné možnosti dosáhli i v našich modelech, vybavme naše automaty aparátem, který jim umožní měnit, značkovat okolí, ve kterém žijí. Doposud naše mašinky komunikovaly prostřednictvím svých portů, které jim ve formalizované podobě umožňovaly „vidět, slyšet, hmatat, cítit“ či emitovat informace, ale jim neumožňovaly žádným způsobem zanechat trvalé značky ve svém prostředí ani se v něm pohybovat. Nejjednodušší způsob, jak toho dosáhnout, je „dovybavit“ kognitivní automaty možností pohybu a možností zápisu a čtení informace z místa, na kterém se nacházejí. To znamená, že v tomto modelu mohou automaty aktivně vyhledávat informaci (kdežto předtím „pasivně“ čekaly, zda a jaká informace se k jejich portům dostane) a zaznamenávat ji do svého prostředí. Výsledné zařízení, jakýsi „automat s efektory“, se podobá robotickým kognitivním systémům.

V teorii automatů se již léta studují modely konečných automatů, které mají možnost číst vstupní data ze vstupní pásky v obou směrech a navíc položit omezený počet (nezávisející na délce vstupu) značek na tuto pásku (viz např. [34]). Tím se dokazatelně zvyšuje jejich výpočetní síla. Pokud povolíme kognitivnímu automatu klást neomezený počet značek a neomezeně se pohybovat po pásce, dostaneme ... interaktivní Turingův stroj! Odpovídající kognitivní automat budeme nazývat *aktivním kognitivním automatem*

Věta 3: Výpočetní síla aktivního kognitivního automatu je ekvivalentní výpočetní síle interktivního Turingova stroje.

Sám Turing ve své práci [23] popisuje „svůj“ stroj jako formalizovaný model matematika - počtáře, který počítá podle instrukcí konečného programu (tj. algoritmu), který si pamatuje ve své „hlavě“, a dále pomocí tužky, gumy a (čtverečkovaného) papíru. V souhlasu s Turingovým vlastním přesvědčením se v literatuře o umělé inteligenci či v úvahách kognitivních filozofů ([10],[19],[20]) se má za to, že celý Turingův stroj je modelem zmíněného počtáře. Naše předchozí úvaha ukazuje, že modelem počtáře je pouze konečné řízení Turingova stroje. Páska představuje okolí, ve kterém počtář žije, může se v něm algoritmickým způsobem pohybovat a měnit jej. Speciálně, může v něm dělat a poté vyhledávat záznamy, potřebné k provedení jeho vlastních výpočtů. Proto nemá pravdu ani A. Hodges, Turingův životopisec, když píše [14]: *„Turingův stroj je modelem činnosti lidské mysli“.* Nikoliv: modelu lidské mysli odpovídá pouze samotné konečné řízení Turingova stroje.

7. Výpočetní síla společenstev kognitivních automatů

Vraťme se k našemu motivačnímu příkladu o „počítaní z hlavy“ z počátku 2. kapitoly a ptejme se, jestli náhodou by nám při realizaci takového úkolu nemohli pomoci další lidé. Přizveme-li si pomocníky a vytvoříme „spolek přátel počítaní“, můžeme s pomocí rostoucí členské základny napočítat libovolně daleko. Teď totiž při vhodné organizaci spolku stačí, aby se jeho členové uspořádali do řady a každý člen si pamatoval pouze konečnou informaci - jednu cifru a jednoduchý prográmek, pomocí kterého instruuje v případě potřeby své sousedy, jestli mají zvýšit svou cifru o jedna. Číslo, do kterého spolek právě napočítal, je uloženo distribuovaně, cifra po cifře, v hlavách všech jeho členů. Pokud je zabezpečen dostatečný přísun nových členů, celý spolek tím získá potenciál počítat libovolně dlouho, čímž převýší schopnost konečného automatu.

Jistý problém představuje podmínka „uspořádání do řady“. Tato podmínka totiž nutí členy spolku „nehýbat se z místa“ , protože pro úspěch celé akce je prostorové uspořádání členů spolku podstatné. Pokud bychom chtěli povolit mobilitu členů (což je přirozené, protože se jedná o lidi), musel by si každý z nich minimálně pamatovat (jméno) svého předchůdce a následovníka v řadě. To je náročnější na paměť než v předchozím případě uspořádání do řady, protože pokud máme momentálně ve spolku n členů, tak jejich jména nemohou být kratší než přibližně $log\ n$ bitů. To znamená, že čím dále počítáme, tím delší jména si členové musí zapamatovat. Tudíž to již nemohou být končené automaty, protože jejich paměťová kapacita je omezená. Musí to být tedy aktivní kognitivní automaty, které jsou dle předchozí věty ekvivalentní Turingovu stroji.

Postupme teď k zobecnění předchozího příkladu uvažováním výpočetního společenství aktivních kognitivních automatů. *Společenství aktivních kognitivních automatů* (nebo krátce: společenství agentů) je v každém časovém okamžiku konečná množina aktivních kognitivních automatů stejného typu a stejné velikosti žijících ve stejném prostředí. Každý agent má své jedinečné jméno a volně se pohybuje ve svém „světě“. Každý má také svůj zvláštní vstup a výstup; všechny dohromady tvoří vstupní a výstupní porty společenství. Jelikož je počet příslušníků ve společenství proměnlivý, je proměnlivý i jeho počet vstupů a výstupů. Automaty mohou rámci jednoho společenství komunikovat mezi sebou tím způsobem, že výstupy z jednoho automatu jsou „poslány“ na vstup jiného automatu. To lze uskutečnit buď při vzájemném (nepředvídatelném, anebo záměrném) setkání, anebo “posláním zprávy“ konkrétnímu agentovi či jejím vystavením „na nástěnce“ kde si ji může každý agent přečíst, apod. Společenství agentů tak může modelovat lidskou společnost, kde výměna zpráv odpovídá „mezilidské“ komunikaci jakéhokoliv druhu: v přirozeném jazyce, prostřednictvím dopisů, internetem, mobilním telefonem apod. Kdo s kým si vymění zprávu je v obecném případě nepředvídatelné, stejně tak jako není předvídatelné, za jak dlouho po odeslání zprávy se tato dostane k adresátovi (což je pravda např. v případě dopisování či internetu). Navíc, agenti jsou smrtelní: vznikají a zanikají nepředvídatelně.

Formálně odpovídá každému agentu aktivní kognitivní automat. Agenti mohou být „v různých vývojových stadiích“ (tj. v různých konfiguracích) podle toho, jaká vstupní data zpracovali. To se projeví kromě jiného různým obsahem i různou velikostí jejich externí paměti (tj. části okolí, které agent používá jako svou externí paměť). Pro popis celého společenství potřebujeme znát v každém okamžiku soupis všech žijících agentů, jeich okamžitou konfiguraci (včetně obsahu jejich externích pamětí), dále záznam všech odeslaných zpráv, dobu odeslání, jejich adresáty a doručovací doby. Naznačíme, jak společenství agentů s (externí) pamětí logaritmické velikosti dovede simulovat (jednopáskový) Turingův stroj třídy PSPACE/*poly.*

Uvažujme nejprve jednodušší případ, kdy agenti dodržují předepsané prostorové uspořádání. Potřebujeme k tomu polynomické množství agentů uspořádaných do dvou řad a „spolupracující“ okolí, ve kterém agenti žijí. Každý agent v první řadě simuluje odpovídající políčko pásky, tj. pamatuje si symbol zapsaný na pásce. Navíc si v daném okamžiku pamatuje, jestli dané políčko v daném okamžiku zkoumá hlava simulovaného Turingova stroje. Pokud ano, tak si odpovídající agent (zvaný „velitel“) také pamatuje, v jakém stavu se Turingův stroj právě nachází. Každý agent v první řadě má také k dispozici (ve své přechodové funkci – nezapomínejme, že agent je konečně stavový automat) popis přechodové funkce Turingova stoje. Každý agent v druhé řadě na pokyn velitele přečte jeden symbol ze svého vstupu.

Na začátku simulace předpokládáme, že agenti v první řadě si pamatují vstup do Turingova stroje. Velitelem je první agent v první řadě. Společenství zřejmým způsobem simuluje kroky Turingova stoje, přičemž se velitelská funkce posouvá od souseda k sousedovi a agenti si přepisují zapamatované symboly ve shodě s přechodovou funkcí stroje, kterou má každý agent k dispozici. Tak to funguje až do okamžiku, kdy stroj vstoupí do tázacího stavu. V tomto okamžiku je potřebné, aby agenti ze druhé řady „přečetli“ na žádost momentálního velitele radu příslušnou dané velikosti vstupu. To se zařídí tak, že velitel vyšle prvnímu agentu z druhé řady žádost o přečtení vstupu. Tento danou žádost vykoná a pošle ji svému sousedovi, atd. Celá věc funguje za předpokladu, že „prostředí“ dodá v tomto okamžiku každému agentovi z druhé řady správný bit rady. Celkově to připomíná simulaci Turingova stroje celulárním automatem.

V právě naznačené simulaci si každý agent pamatuje pouze konečný, na vstupu nezávisející počet bitů, takže to v principu ani nemusí být aktivní automat (Turingův stroj). Pokud ale dovolíme mobilitu agentů, musíme je pojmenovat, čímž se ve smyslu předchozí diskuse zvýší paměťová náročnost každého z nich na logaritmickou vzhledem k jejich počtu. Také si všimněme, že agenti s logaritmicky omezenou pamětí nemohou vytvořit větší než polynomicky omezené společenství, protože na zapamatování většího počtu jmen jednoduše nemají prostor.

Opačná simulace společenství agentů logaritmické velikosti je možná za předpokladu, že Turingův stroj má k dispozici v každém okamžiku výše zmíněný popis společenství. Tento popis dodá stroji jeho rádce (jako externí informaci). Turingův stroj na základě daných informací může pro daný vstup (jenž musí, na rozdíl od společenství, číst sekvenčně), odsimulovat činnost společenství. Je zřejmé, že agentů nemůže být více než polynomické množství.

Je tedy zřejmé, že simulaci Turingova stroje s radou společenstvím agentů a i opačnou simulaci lze realizovat a její složitostní parametry popisuje následující věta.

Věta 4: Výpočetní síla společenství agentů s logaritmickou externí pamětí je ekvivalentní výpočetní síle polynomicky prostorově ohraničeného Turingova stroje s polynomickým rádcem.

Všimněme si, že i v tomto případě má výsledné společenství superturingovskou výpočetní sílu. Teď ovšem zdrojem této síly není nevypočitatelená, nealgoritmická síla evoluce. Zdrojem síly je na jedné straně polynomická kardinalita společenství (ta nám umožní simulovat třídu PSPACE), a na straně druhé nevypočitatelné charakteristiky interakce příslušníků v rámci společenství a jejich vzniku a zániku (které způsobí polynomickou neuniformitu výsledného výpočetního systému). Také si uvědomme, že naše společenství má v každém okamžiku konečnou velikost.

Je také dobré si uvědomit, že právě popsaný formalizmus se hodí i pro modelování takových systémů, jako je internet. Popravdě, obsah této kapitoly byl inspirován prací [30], kde se podobným způsobem simuloval internet.

8. Zamyšlení nad dosaženými výsledky

Získané výsledky lze interpretovat v několika rovinách. Nejprve přidáme pár poznámek k větám 1 až 4 a poté budeme zvlášť diskutovat otázky spojené s nerekurzivní výpočetní sílou některých kognitivních systémů.

Věta 1 ukazuje ekvivalenci základních modelů kognitivních automatů. V teorii složitosti jsou známé i další modely (např. kombinatorické či prahové obvody, jiné typy neuronových sítí, zejména biologicky motivované, apod.) zde nezmíněné. Nicméně, ekvivalence všech modelů naznačuje, že výpočetní kognice je robustní vlastností, kterou mohou zvládnout různě realizované výpočetní systémy ekvivalentní konečným automatům. To ovšem je „principiální" tvrzení. V praxi závisí mnoho na efektivitě těchto systémů. Např. v práci [39] je naznačen princip zrodu fungování vědomí v kognitivních systémech. V nejjednodušším případě plní vědomí roli „kontrolního" mechanismu, který prostřednictvím informací od různých senzorů ověřuje správnost provedení různých motorických akcí, ke kterým vydal systém příkazy. Pokud se akce nezdařily dle pokynů, vědomí tuto skutečnost rozezná a zabezpečí nápravné akce. Pro plnění této úlohy je nutné, aby vědomí pracovalo v reálném čase, který je definován vzhledem k „rychlosti" systému: systém musí reagovat tak rychle, aby bylo chybné provedení příkazů včas rozeznáno a aby nápravné akce ještě měly smysl. Takové požadavky v praxi diskvalifikují „pomalé" systémy a naopak podporují specializované systémy. Např. jsou známé různé typy tzv. realistických či biologických neuronů, které jsou specializované na řešení některých kognitivních úkolů, pro řešení kterých je dokazatelně potřebná standardní neuronová síť kvadratické velikosti (vzhledem k počtu vstupů biologického neuronu) [17].

Věta 2 vlastně kvantitativně dokumentuje předchozí úvahu o různé efektivitě vybraných kognitivních automatů. Z věty je vidět, ze stejně veliká neuronová síť má sílu PSPACE/*poly*, kdežto konečný automat podobné velikosti pouze LOGSPACE/*poly*.

Věta 3 ukazuje skok ve výpočetní síle kognitivních automatů, vybavených efektory a dostatečně rozvinutými kognitivními schopnostmi, které jim umožní algoritmickým vypočitatelným způsobem měnit své okolí. Tím způsobem ho principiálně mohou využívat jako externí paměť. Takto rozvinutý kognitivní automat (kognitivní robot?) má sílu interaktivního Turingova stroje, tj. větší než „jedinec" představovaný obyčejným kognitivním automatem. Jinými slovy, z původně konečně stavového zařízení, schopného dosahovat pouze konečného počtu různých konfigurací, se stává sice pořád konečně stavové zařízení, které je však nyní schopné dosahovat potenciálně nekonečně mnoha konfigurací. To je pěkný argument, kvalitativně charakterizující revoluční přínos vzniku písma pro rozvoj lidské civilizace,

Výpočetní síla interaktivních Turingových strojů byla iniciována a podrobně zkoumána v pracích [27], [28] a [29]. Zhruba řečeno, používání takových výpočetních zařízení vede k zobecnění teorie vyčíslitelnosti pro případ nekonečných výpočtů. Platí, že každý konečný počáteční úsek nekonečného výpočtu interaktivního Turingova stroje je ekvivalentní nějakému výpočtu běžného Turingova stroje, pokud mu dodáme stejné vstupy jako interaktivnímu stroji.

Věta 4 na první pohled vypadá jako kvantitativní vyjádřením přísloví „více hlav, více rozumu“. Věta však tvrdí něco navíc: přechodem od jedinců ke společenství získá výsledný systém výpočetní schopnost, která přesahuje možnosti jedince (tj. Turingova stroje). Všimněme si, že skok ve výpočetní síle od jedince ke společenství je obrovský: v našem případě má jedinec „sílu“ LOGSPACE, a víme, že LOGSPACE je vlastní podmnožinou PSPACE, nemluvě o tom, že PSPACE/*poly* posouvá výpočetní sílu společenství nad turingovsky vypočitatenou oblast. Mimochodem, posledně zmíněný fakt je v rozporu s Penroseovým přesvědčením (viz [20], část 3.10), že společenství nemá vetší výpočetní sílu než jedinec. Z našeho modelu je vidět sílu nepredikovatelných informací ovlivňujících činnost systému „zvnějšku“. Znamenají výsledky věty 4, že tyto systémy mohou řešit nerozhodnutelné problémy? Mohou, ale pouze za určitých předpokladů. Tyto předpoklady lze odvodit z naznačeného důkazu uvedené věty. Aby tyto systémy mohly simulovat Turingův stroj s rádcem, potřebují „spolupracující“ okolí, které jim poskytne stejnou informaci jakou poskytuje rádce Turingovu stroji. Takže uvedené výsledky mají nekonstruktivní charakter: víme, že jak rádce, tak i odpovídající vstupy z okolí v principu existují, ale neznáme žádný algoritmický způsob, jak je sestrojit. V praxi tyto předpoklady na existenci vhodného okolí pro řešení konkrétních nerozhodnutelných problémů, jakým je např. problém zastavení, splněny nejsou. To znamená, že bez příslušných „správných“, ale nevypočitatelných vstupů z okolí žádné společenství kognitivních automatů nám problém zastavení nevyřeší. Na druhé straně ale žádný Turingův stroj bez rádce nedovede simulovat např. existující internet – prostě proto, že internet se vyvíjí veskrze nepredikovatelným, nealgoritmickým způsobem. Dá se říct, že internet řeší nerozhodnutelné problémy, ale takové, které jaksi vznikají samy od sebe společnou souhrou nás všech, kteří ho nepredikovatelným způsobem modifikujeme (přidáváním dalších počítačů, dalšího software), vkládáme do něho nepredikovatelné informace, které se v něm šíří z hlediska času doručení k příslušným adresátům nepredikovatelným způsobem. To znamená, že my v roli operátorů a technické a fyzikální podmínky fungování počítačových sítí společně hrajeme roli „rádce“ – avšak rádce slepého, nemajícího jako celek jakékoliv smysluplné záměry. To samé platí o pokusu simulovat lidskou společnost – také se nevyvíjí algoritmicky.

Stojí za úvahu i následující námitka: nejsme i my, lidé, z hlediska komputacionalismu kognitivními roboty? Pokud bychom sami sebe v záležitostech obsluhy a využívaní internetu nahradili kognitivními roboty, nestal by se výsledný systém jiným z hlediska jeho výpočetní síly? Nuže, nestal. Stále by v systému zůstala nepredikovatelnost jeho vnitřního fungování a Věta 4 by stále platila. Teprve kdybychom nepredikovatelnost z výpočetního sytému zcela odstranili, dostali bychom se na úroveň věty 3.

Poslední zamyšlení věnujme ještě otázce, která trápila kromě jiných i Penrose [20] a Copelanda [7],[8], když hledali důvody, proč je (lidská) mysl či mozek v některých ohledech „silnější“ výpočetní zařízení než Turingův stroj. Lidé, zejména matematici, totiž často pomocí intuice dovedou vyřešit problémy „nealgoritmickým“ způsobem. Z hlediska komputacionalismu a našich modelů lze k tomu dodat dvě vysvětlení bez toho, že bychom se utíkali k nějakým „nevysvětlitelným“ (kvantovým?) schopnostem mozku[3][20] či podivným neustále se zrychlujícím Turingovým strojům [6],[8]. Za prvé, protože neznáme mechanismy myšlení, nelze tvrdit, že jsou nealgoritmické. Naopak, máme mnoho důvodů se domnívat, že skutečnost je přesně opačná. Za druhé, „intuici“ je snad možné vysvětlit tak, že mozek používá algoritmické principy plus „náhodné“ nápovědy čerpané (asi zprostředkovaně) z nepredikovatelných podnětů okolí. Příslušné podněty nemusí být úplně náhodné – např. matematik při řešení problémů cílevědomě hledá analogie a dodatečnou relevantní informaci. Na základě nových skutečností pak problém vyřeší v podstatě algoritmicky, odvozením řešení ze známých faktů a velmi často zkoumáním všech „smysluplných“ možností. Může se stát, a každý matematik to pravděpodobně zná z vlastní zkušenosti (viz. např. [13]), že k definitivnímu „rozsvícení“ (moment „heuréka!“) v hlavě dojde následkem naprosto nesouvisejících vnějších podnětů. To může být ten okamžik, kdy do systému vstoupila „správná nápověda“. Existuje přísloví, že ve vědě *„náhoda obšťastňuje pouze duchy připravené*“. To přesně vystihuje předešlou situaci: bez dostatečné znalosti problému nelze byť i správné, ale stručné nápovědi porozumět.

Jiná zkušenost říká, že k řešení problému obyčejně vede série správně položených jednodušších otázek (což většinou aposteriorně používáme při psaní a vysvětlování důkazů problémů). V rámci našich modelů vedou podobné úvahy k modelu aktivního kognitivního automatu s nápovědou anebo lépe s rádcem. Oba dva modely mohou řešit problémy běžným Turingovým strojem neřešitelné. Jak jsme zmínili ve 4. části, rada je nezávislá na konkrétním vstupu, kdežto nápověda dává odpovědi na konkrétní otázky. Zdá se, že aktivní kognitivní automat s rádcem je zajímavější model, jenž, jak naznačuje Věta 3, je ekvivalentní *interaktivnímu Turingovu stroji s rádcem*. Vždyť co jiného dělá matematik když „konzultuje“ veškerou dostupnou literaturu týkající se jeho problému? Využívá příslušných znalostí jako rádce; nenachází zde hotové odpovědi na své otázky, ale „návod“, jak se k odpovědím (možná) dopracovat anebo jak své otázky zpřesnit. Teprve poté (možná) následuje fáze „správné nápovědy“ zmíněné výše a nalezení řešení původního problému. Následně matematik svůj problém i řešení zveřejní - tj. vlastně efektivně rozšíří množinu rad, kterou může rádce poskytovat. Tento cyklus hledání a hromadění matematických (i jiných) pravd zatím nedovedeme pomocí našich nástrojů dostatečně zajímavě modelovat, ale je možné, že jakmile pochopíme přesněji algoritmické principy myšlení, dopracujeme se i k odpovídajícím modelům „aktivních kognitivních automatů s intuicí“. Zdá se, že bez pochopení algoritmické podstaty takových otázek, jakými jsou otázky spojené s porozuměním porozumění a vědomí, se podstatně dále nedostaneme.

---

[3] Kvantové operace mohou být zajímavé z hlediska zvýšení efektivity výpočtů neuronových sítí. Např. některé studie uvažují neuronové sítě typu „vítěz bere vše“ [18], které využívají operace nalezení maximální shody v množině neuronů. Tato operace je pro běžné sítě výpočetně velmi náročná, kdežto pomocí kvantových efektů by se dala realizovat v konstantním čase. To může mít ve smyslu komentáře k větě 1 v této části zásadní význam.

9. Závěr

Hledání hranic výpočetní síly kognitivních systémů je prvním krokem k pochopení algoritmické podstaty činnosti těchto systémů. Na základě předešlých úvah se zdá, že ve vývoji kognitivních systémů a v činnosti společenstev takových systémů je nepredikovatelnost, nevypočitatelnost rozhodujícím prvkem ve hře. V odpovídající literatuře se tyto vlivy přehlíželi anebo se jim nepřikládala velká váha. Je docela možné, že se tak dělo pod vlivem stále převládajícího paradigmatu Turingova stroje jako „praotce" všech výpočetních modelů, který je schopen formálně zachytit všechny výpočty. To ale není pravda - příklad internetu a v obecnější poloze Věta 4 to dokazují naprosto zřetelně. V nedávné práci van Leeuwena s Wiedermannem [30] se na základě podobných pozorováni argumentuje, že v současné době jsme svědky změny paradigmatu standardního modelu Turingova stroje na paradigma modelu interaktivního Turingova stroje s rádcem. Jak naznačil tento článek, tato změna vede i k hlubšímu pochopení výpočetního potenciálu kognitivních a evolučních systémů.

Literatura

[1] M. A. Arbib, M.A. (Editor): The Handbook of Brain Theory and Neural Networks. The MIT Press, Cambridge - Massachusetts, London, England, 1995, 1118 p.

[2] J. L. Balcázar, J. Diaz, and J. Gábarro: Structural complexity, Vol. I, 2nd edition, Springer-Verlag, Berlin, 1995.

[3] J. D. Barrow: Pí na nebesích. O počítání, myšlení a bytí. Mladá fronta, edice Kolumbus, 2000, 307 s.

[4] L. Blum, F. Cucker, M. Shub and S. Smale. Complexity of real computation. Springer-Verlag, New York, NY, 1998.

[5] D. J. Chalmers, D.J.: A Computational Foundations for the Study of Cognition. Minds and machines, Vol. 4, No. 4, 1994

[6] B. J. Copeland: Even Turing Machines Can Compute Uncomputable Functions. In: Unconventional Models of Computation, C. S. Calude et al., Eds., Springer, Singapore, 1998, pp. 150--164

[7] B. J. Copeland, D. Proudfoot: Alan Turing's Forgotten Ideas in Computer Science. Scientific American, April 1999, p.77-81

[8] B. J. Copeland: Turing's O--machines, Searle, Penrose and the brain. Analysis, an electronic journal found at http://www.shef.ac.uk/uni/academic/N-Q/phil/analysis/preprints/ preprint33.html

[9] N. G. de Bruijn: A Model for Associative Memory, a Basis for Thinking and Consciousness. In: Proc. of the 26--th International Colloquium on Automata, Languages and Programming ICALP'99. LNCS Vol. 1664, Springer--Verlag, Berlin, pp. 74--89, 1999

[10] D. C. Dennett: Consciousness Explained. Penguin Books, 1991, 511 p.

[11] D. C. Dennett: Druhy myslí. Archa, Bratislava, 1997, 178 s., taktéž Kinds of Minds: Towards an Understanding of Consciousness. New York, Basic Books, 1996, 184 p.

[12] L. G. Goldschlager: A Computational Theory of Higher Brain Function. Technical Report 233, April 1984, Basser Department of Computer Science, The University of Sydney, Australia, ISBN 0 909798 91 5

[13] J. Hadamard: The Mathematician's Mind: The Psychology of Invention in the Mathematical Field. Princeton University Press, Princeton, New Jersey, 1996, 143 p.

[14] A. Hodges: Turing – A Natural Philosopher. Phoenix, 1997, 58 s.

[15] J.E. Hopcroft, J.D. Ullman, J. D.: Introduction to Automata Theory, Languages, and Computation. Addison-Wesley Publishing Company, Reading, Mass., 1979, 417 p.

[16] R. M. Karp, R. J. Lipton: Some connections between nonuniform and uniform complexity classes, in Proc. 12th Annual ACM Symposium on the Theory of Computing (STOC'80), 1980, pp. 302-309

[17] W. Maass and C. Bishop, editors, *Pulsed Neural Networks*. MIT-Press (Cambridge), 1998

[18] W. Maass. On the computational power of winner-take-all. *Neural Computation*, 12(11):2519-2536, 2000, 26 p.

[19] J. R. Searle.: Minds, Brains, and Programs. Behavioral and Brain Sciences, No. 3, 1980, pp. 417—424

[20] R. Penrose: Shadows of the Mind (A Search for the Missing Science of Consciousness). Oxford University Press, Oxford, 1994, 457 p.

[21] H. T. Siegelmann: Computations beyond the Turing limit, Science 268 (1995) 545-548.

[22] J. Šíma, R. Neruda: Teoretické otázky neuronových sítí, Matfyzpress, 1996, 390 s.

[23] J. Šíma, J. Wiedermann,: Theory of Neuromata. Journal of the ACM, Vol. 45, No. 1, 1998, pp. 155—178

[24] A. M. Turing: On computable numbers, with an application to the Entscheidungsproblem, Proc. London Math. Soc., 42-2 (1936) 230-265; A correction, ibid., 43 (1937) 544-546.

[25] A. M. Turing: Systems of logic based on ordinals, Proc. London Math. Soc. Series 2, 45 (1939) 161-228.

[26] A. M. Turing: Computing machinery and intelligence, Mind 59 (1950) 433-460.

[27] J. van Leeuwen, J. Wiedermann: On the power of interactive computing, in: J.van Leeuwen et al (Eds), Theoretical Computer Science - Exploring New Frontiers of Theoretical Computer Science, Proc. IFIP TCS'2000 Conference, Lecture Notes in Computer Science Vol. 1872, Springer-Verlag, Berlin, 2000, pp 619-623.

[28] J. van Leeuwen, J. Wiedermann: On algorithms and interaction, in: M. Nielsen and B. Rovan (Eds), Mathematical Foundations of Computer Science 2000, 25th Int. Symposium (MFCS'2000), Lecture Notes in Computer Science Vol. 1893, Springer-Verlag, Berlin, 2000, pp. 99-112.

[29] J. van Leeuwen, J. Wiedermann: A computational Model of Interaction in Embedded Systems. Technical Report TR CS-02-2001, Department of Computer Science, Utrecht University, 2001

[30] J. van Leeuwen, J. Wiedermann: The Turing machine paradigm in contemporary computing, Techn. Report UU-CS-2000-33, Dept of Computer Science, Utrecht

University (2000), also in: B. Engquist and W. Schmidt (Eds), Mathematics Unlimited - 2001 and Beyond, Springer-Verlag, 2001, s. 1139-1155
[31] L. G. Valiant: Functionality in Neural Nets. Proc. 7th Nat. Conf. on Art. Intelligence, AAAI, Morgan Kaufmann, San Mateo, CA, 1988, pp. 629-634
[32] L. G. Valiant, L.G.: Circuits of the Mind. Oxford University Press, New York, Oxford, 1994, 237 p., ISBN 0--19--508936--X
[33] Valiant, L.G.: Cognitive Computation (Extended Abstract). In: Proc. 38th IEEE Symp. on Fond. of Comp. Sci., IEEE Press, 1995, p. 2—3
[34] K. Wagner, G. Wechsung: Computational Complexity, VEB Deutscher Verlag der Wissenschaften, Berlin 1986, 551 s.
[35] J. Wiedermann, J.: Towards Computational Models of the Brain: Getting Started. Neural Networks World, Vol 7., No.1, 1997, p. 89—120
[36] J. Wiedermann, J.: Towards Algorithmic Explanation of Mind Evolution and Functioning (Invited Talk). In: Proc. of the 23-rd International Symposium on Mathematical Foundations of Computer Science, LNCS Vol. 1450, Springer Verlag, Berlin, 1998, pp. 152--166 .
[37] J. Wiedermann: Simulated Cognition: A Gauntlet Thrown to Computer Science. ACM Computing Surveys, No. 3, electronic supplement, paper no. 16, 1999
[38] J. Wiedermann: The computational limits to the cognitive power of neuroidal tabula rasa, in: O. Watanabe and T. Yokomori (Eds.), Algorithmic Learning Theory, Proc. 10th International Conference (ALT'99), Lecture Notes in Artific. Intelligence, Vol. 1720, Springer Verlag, Berlin, 1999, pp. 63-76
[39] J. Wiedermann: Intelligence as a Large-Scale Learning Phenomenon. Technical Report ÚI AV ČR č. 792, 1999, 17 s.