

Pokroky matematiky, fyziky a astronomie

Grigorij S Cejtin

Od logicizmu k proceduralizmu (autobiografické zamyslenie)

Pokroky matematiky, fyziky a astronomie, Vol. 29 (1984), No. 3, 155--164

Persistent URL: <http://dml.cz/dmlcz/138619>

Terms of use:

© Jednota českých matematiků a fyziků, 1984

Institute of Mathematics of the Academy of Sciences of the Czech Republic provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.



This paper has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ: The Czech Digital Mathematics Library* <http://project.dml.cz>

diskuse

OD LOGICIZMU K PROCEDURALIZMU
(AUTOBIOGRAFICKÉ ZAMYSLENIE)

Grigorij S. Cejtin, Leningrad

V diskusii o vplyve počítačov a nimi vyvolaných ideí na formovanie našich nových predstáv o tradičných oblastiach ľudského záujmu pokračujeme, po témach bližších pedagógom (S. A. Papert, PMFA XXVI/1981 a A. P. Jeršov, PMFA XXVII/1982,) článkom doktora fyzikálno-matematických vied Grigorija Samuiloviča Cejtina, pracovníka katedry aplikovanej lingvistiky Leningradskej štátnej univerzity. Článok je písaný pre matematikov a očakávame, že práve v nich vzbudí aj najzreteľnejšiu odozvu.

Je samozrejmé, že na základe osobného príbehu konverzie, ktorý nám autor vyrozpráva, nemožno robiť zovšeobecňujúce uzávery. Sme si toho vedomí rovnako ako autor. Jeho spoveď však môže provokovať introspekciu čitateľa, nútiť ho k nesúhlasu, súhlasu, k polemike, ... k zamysleniu nad východiskami, cieľmi a podstatou matematikovej práce v kontexte nových podmienok, ktorých spolupracom je aj počítač.

Jozef Kelemen

Rozpoviem vám, ako sa zmenilo moje pôvodné presvedčenie, že dobré poznatky musia byť reprezentované nevyhnutne

G. S. TSEYTIM: *From Logicism to Proceduralism (An Autobiographical Account)*. In: A. P. ERSHOV and D. E. KNUTH (eds.): *Algorithms in Modern Mathematics and Computer Science*, LNCS vol. 122, Springer-Verlag, Berlin—Heidelberg—New York, 1981, pp. 390—396.

© Springer-Verlag, Berlin—Heidelberg—New York, 1981

množinou logických výrazov v rámci vhodného matematického modelu, v môj terajší názor, podľa ktorého sú poznatky v zásade algoritmické.

Musím vysvetliť, prečo som sa rozhodol namiesto systematického výkladu proceduralistických názorov zahĺbiť sa radšej do niektorých detailov mojej minulosti. Prvý z dôvodov je, že v súčasnosti jednoducho nie som schopný — až na pár všeobecných pojmov — podať taký výklad, ktorý by ukázal, ako proceduralistický pohľad funguje. Musel som teda nájsť isté referenčné body, s ktorými porovnávam svoje názory, a najjednoduchšie sa mi zdalo podrobiť kritike vlastné omyly. (Všetky odkazy na moje práce v tomto texte majú ilustrovať moje chybné závery, a nie súčasné stanovisko). Druhým dôvodom je, že moje terajšie presvedčenie je založené na osobnej skúsenosti, ktorá nemusí presvedčiť každého. Vynasnažím sa však ukázať, ako presvedčila mňa.

Začínal som (na počiatku 50. rokov) ako čistý matematik s podvedomým presvedčením, že matematické entity sú niečo, čo skutočne existuje a dá sa cestou úvah skúmať, že každá zmysluplná otázka o nich má „objektívnu“ odpoveď, ktorá sa nakoniec dá nájsť, že napriek vetám o neúplnosti sa dajú chýbajúce informácie získať z „reality“ nejakým — zatiaľ nejasným — spôsobom. (Vedel som pochopiteľne, že matematické pojmy sú abstrakcie odvodené z fyzikálneho sveta, neovplyvnilo to však moje zmýšľanie.) Vedel som, ako písať algoritmy — pôvodne v tvare Markovových normálnych algoritmov — ale považoval som ich za ďalší typ matematického objektu, ktorého vlastnosti by bolo možné dokazovať štandardnými matematickými úvahami (v štýle [1]), aj keby boli bezprostredne zrejmé. Uvedomoval som si Markovovu kritiku klasickej teórie

množín a zúčastnil som sa jeho programu vývoja konštruktívnej matematiky. Jeho filozofiu som však neakceptoval a svoju prácu v oblasti konštruktivismu považujem skôr za istý druh cvičenia v sebaobmedzovaní.

Tento platonický postoj našiel v ďalšom svoje vyjadrenie v mojom prístupe k automatickému prekladu jazykov. Touto oblasťou som sa zaoberal radšej, než hociktorou konvenčnou aplikačnou oblasťou matematiky. Konečný výsledok aplikovaného výskumu tu obyčajne býva výpočtová procedúra (v mojom prípade algoritmus syntaktickej analýzy). Empirické procedúry však zvyčajne nebývajú považované za vhodnú formu reprezentácie poznatkov. Existuje názor, podľa ktorého je potrebné vytvoriť „matematický model“, t.j. ideálny konštrukt (zostavený z matematických entít), približne vyjadrujúci vlastnosti reálneho objektu. Všetky praktické procedúry by mali byť odvodené matematicky na jeho základe. Empirický prístup k oblasti konštrukcie syntaktických analyzátorov sa ukázal byť veľmi skoro nevhodným. Prinajmenšom, ako by som mohol dodať dnes, s našimi programovacími metódami z roku 1960, kedy sme mali veľmi málo skúseností s modularnosťou a prísne technologické ohraničenia. Naša výskumná skupina vyvinula systém na logický opis syntaxe (závislostné gramatiky – dependence grammars) a publikovala prácu [2] podporiac nealgoritmickú reprezentáciu lingvistických poznatkov.

Výhody nealgoritmických poznatkov sú očividné. Jednoduchá rovnica, akou je Ohmov zákon $U = I \cdot R$, nahrádza niekoľko procedúr, totiž $U := I * R$, $I := U/R$ a $R := U/I$. Navyše, v kombinácii s ďalšími rovnicami tvorí opis zložitých obvodov, vďaka čomu môžu byť pri riešení

celého radu ďalších problémov použité dobre známe matematické metódy.

Takto som sa počiatkom šesťdesiatych rokov ocitol v rozličných oblastiach mojej práce zoči-voči algoritmickému reprezentácii poznatkov. V oblasti teórie algoritmov a konštruktívnej analýzy som vyvinul metódu nahradenia priamej konštrukcie algoritmov (používaných pri existenčných dôkazoch) istými manipuláciami so spočítateľnými množinami [3, 4], čo sa ukázala byť kratšia a bližšia metóda vžitému matematickému štýlu práce. V oblasti programovania som dospel k poznaniu, že ani jazyk vysokej úrovne (ktorú nazývame „automatické programovanie“) neuľahčuje programátorovi rutinnú prácu spojenú s „algoritmizáciou“. Reálny systém automatického programovania, domnieval som sa, by mal byť schopný zostrojiť program – alebo prinajmenšom jeho triviálnu časť – z nejakého iného tvaru špecifikácie úlohy, *pravdepodobne* z logickej špecifikácie. To ma priviedlo k oblasti nazývanej dnes verifikácia programov [5, 6], ktorú som považoval za predstupeň automatickej algoritmizácie.

O logickom prístupe som uvažoval aj pri natoľko konkrétnej práci, ako je programovanie komerčného informačného systému. Navrhol som [7] množinu axiém v teórii prvého rádu a všeobecný návod pre nové axiomy, nazývaný princíp úplnosti. Podľa tohto princípu je ľubovoľná zmena v systéme v rámci daného časového intervalu dôsledkom jednej spomedzi explicitne vymenovaných udalostí.

Môj logistický prístup k aplikáciám stál na nevyslovenom predpoklade, ktorý zasluhuje zvláštny rozbor. Bol to predpoklad, že v predikátovom počte je reprezentovateľné univerzálne ľubovoľné správne uvažovanie, t.j., že ľubovoľné zmysluplné úvahy sa dajú vhodnou voľbou no-

tácie preložiť do predikátového počtu. Niekto môže namietaf, že ide o extrémne zjednodušenie a že pri uvažovaní dokáže vždy rozoznať také zložky, ako je heuristika, plauzibilné úvahy, nezreteľné tvrdenia, intuícia, vhlad (a tým skôr jasnozrivosť). Javy tohto druhu sa často považujú za dôkaz ohraničenosti štandardného logického prístupu a bývajú používané ako východisko k pokusom prekonať tieto hranice. Zdôrazňujem však, že môj predpoklad sa nevzťahoval na myslenie v takejto šírke, skôr iba na to, čo bežne označujeme ako správne, deterministické alebo logické myslenie. (To bola príčina mojej viery, že uvažovanie obsiahnuté v programoch je založené na logickom odvodzovaní, ktorá ma priviedla ku koncepcii tvorby programov na základe logických špecifikácií.)

Dnes mám na ilustráciu tohto klamu poruke veľmi jednoduchý príklad. Nech má Peter dve ceruzky. Koľko ich bude mať po tom, čo mu Anna dá ďalšie tri ceruzky? Odpoveď je päť a je to veľmi exaktný a logický uzáver. Teraz dodám, že Peter jednu ceruzku stratil. Tým sa odpoveď mení. Dá sa záver získaný odvodením v nejakom systéme predikátového počtu urobiť neplatným pridaním nejakého zvláštneho predpokladu? Alebo by mala moja predchádzajúca odpoveď znieť: „5 za predpokladu, že sa s Petrom a/alebo jeho ceruzkami nič ďalšieho nestalo“?

Isteže, je to dosť surový argument a pred 15 rokmi by som ho neprijal. V rámci logického prístupu sa takýto problém dá zdolať pomocou princípu úplnosti alebo použitím pojmu „minimálneho dôsledku“ („minimal entailment“), ktorý zaviedol J. McCarthy*). Existuje

*) Definícia napr. v: MCCARTHY, J.: *Circumscription — A Form of Non-Monotonic Reasoning*. *Artificial Intelligence* 13 (1980), (pozn. prekl.)

však jemnejšia limitácia predikátového počtu vyplývajúca zo spôsobov používania mien (alebo premenných).

Meno považujeme za samostatný objekt viazaný trvale alebo dočasne (ak to je premenná) na nejakú entitu a jedinou funkciou mena je reprezentovať túto entitu. Nie je dovolené brať do úvahy aj vonkajší výskyt mena alebo jeho zloženie; termy a formuly môžeme považovať za zložené mená, ale ich štruktúra udáva skôr operácie na označených objektoch než na menách. Väzba konkrétneho mena s konkrétnou entitou je čiste náhodná. Ak sa stane, že dve mená definované v rôznych oblastiach sa zdajú byť rovnaké, je to vec, ktorej by sa mal dobre vychovaný logik vyhnúť alebo ju aspoň ignorovať.

Väčšina dnešných programovacích jazykov (okrem LISPU) zaujíma rovnaký postoj. Správa o Algole 68*) citovala Shakespearea**); „Záleží na mene? Čo ružou zveme, aj inak zvané voňalo by krásne.“ Algolovské pravidlo nájdania identifikátora, ktorý nie je deklarovaný v procedúre, kde sa používa, napodobuje pravidlo substitúcie predikátového počtu: nahradením premennej y funkciou $f(x)$ vo formuli $\exists xP(x, y)$ dostávame $\exists xP(x, f(x))$. Iba na metajazykovej úrovni sa dá s menom narábať ako s autonómnym objektom, čo vysvetľuje čiastočne aj striktný rozdiel medzi metajazykom a jazykom (v programovaní medzi časom kompilácie a časom výpočtu). Trochu neskôr uvidíme, prečo je toto stanovisko príliš obmedzujúce.

*) WIJNGAARDEN VAN, A., MAILLOUX, B. J., PECK, J. E., KOSTER, C. H. A.: *Report on the Algorithmic Language ALGOL 68*. MR 101, *Mathematish Centrum*, Amsterdam, 1969; (pozn. prekl.)

***) *Rómeo a Júlia, II.2. (Capulettiho záhrada)*, preklad Z. JESENSKEJ (LITA, Bratislava, 1969), (pozn. prekl.)

Zmena mojich názorov bola postupná a podmienili ju rôzne príčiny: Postupne vychádzala najavo vnútorná slabosť pojmu „matematický svet“. Pokrok v oblastiach automatického dokazovania viet, automatického programovania a reprezentácie sémantiky prirodzených jazykov stavajúcich na striktnom logickom prístupe bol nevyrazný v porovnaní s úspešnejšími empirickými výsledkami. No a moje vlastné skúsenosti s vývojom problémovo orientovaných jazykov. Neexistovali nijaké bezprostredné dôvody deklaratívnej reprezentácie poznatkov, ktorej som sa pridržiaval preto, lebo to bola jediná forma reprezentácie, plynúca z logistických názorov. Teraz nazerám na deklaratívny spôsob ako na práve jeden typ reprezentácie, ktorý nemôže fungovať izolovane, bez podpory procedurálnych poznatkov. Nijaká rovnica, nech by bola hocako všeobecná a elegantná, sa nedá použiť bez algoritmu na jej riešenie. (Sledujúc paralelu s Virchowovým*) dobre známym princípom týkajúcim sa živých buniek, možno povedať, že algoritmy možno získať iba z algoritmov.) Prechod k proceduralizmu sa teda uskutočnil v troch krokoch: odstránením ohraničení daných logicizmom, rozšírením metód proceduralnej reprezentácie poznatkov a pochopením univerzálnej úlohy procedurálnych poznatkov.

Moja dôvera vo svet matematiky bola vážne narušená aj ťažkosťami v základoch matematiky, t.j. skutočnosťou, že množstvo otázok vyzerajúcich zmysluplne (lebo boli syntakticky správne) nemôže byť zmysluplne zodpovedaných. Vysvetlenia,

*) R. VIRCHOW (1821—1902), nemecký patológ a antropológ. Autor tu naráža zrejme na jeho teóriu bunky (*Cellular pathologie*, 1858) a jej funkcií v organizme a pri jeho ochoreniach (pozn. prekl.).

že „množina všetkých ordinálov neexistuje, zatiaľčo množina všetkých reálnych čísel áno“, sa dajú ľahostajne akceptovať, no dôkaz nezávislosti hypotézy kontinua podaný P. Cohenom ukazuje, že veci sú oveľa horšie. V matematike pozorujeme neustávajúcu eskaláciu neúplnosti: N. I. Lobačevskij, J. Bólyai — absolútna geometria postráda informácie potrebné k rozhodnutiu, ktorá z dvoch geometrií je pravdivá; J. Gödel — množina teorém dokázateľných v ľubovolnej teórii je rekurzívne spočítateľná a nemôže teda obsahovať všetky skutočnosti o prvkoch rekurzívne nespočítateľnej množiny; P. Cohen — žiadna spočítateľná množina tvrdení nemôže aproximovať popis nejakej nespočítateľnej množiny entít. Zdalo sa mi, že matematické otázky prestávajú mať väčší zmysel než otázky o postavách nejakého románu. Čo je potom predmetom matematiky? Konštruktívna matematika pre mňa nepredstavovala východisko: zavedenie jemnejších rozdielov medzi tvrdeniami zvýšilo počet očividne zmysluplných otázok, poskytovalo však menej prostriedkov na ich zodpovedávanie. Raz som prednášal o základoch matematiky skupine fyzikov a podarilo sa mi vysvetliť im moje zmätenie nad predmetom matematiky. Reagovali so sympatiami. Povedali mi totiž, že matematika sa dostáva na úroveň fyziky.

Aplikácia matematickej logiky na opis bežného uvažovania bola ďalšou oblasťou, kde sa moje očakávania nesplnili. Moju prácu v oblasti prirodzených jazykov spravádzala viera, že jazyk sa dá považovať za zložitý kalkul so syntaxou definovanou pomocou generatívnej gramatiky a so sémantikou udávajúcou význam každého generovaného objektu pomocou významov jeho zložiek. Pochybnosti začínajú otázkou, ktoré objekty sa dajú použiť na

reprezentáciu významov. Význam sa dá koniec-koncov reprezentovať v pojmochovania. Keďže však bola takáto vyhládka príliš neurčitá, vychádzal som radšej z myšlienky použiť nejaký jazyk podobný jazyku logiky, v ktorom má chovanie (dokazovanie teorém) svoju formálnu definíciu. Začal som teda budovať reprezentácie viet, ako aj ich zložiek v predikátovej logike (pozri [8]).

Tento plán zlyhával už od začiatku, trvalo mi však viac než rok, kým som si to uvedomil. Obmedzil som sa predovšetkým na matematické texty, a to vzhľadom na príklady typu „On vie, že...“. V nich môže nahradenie ekvivalentných výrokov na mieste bodiek vyústiť do neekvivalentného tvrdenia (to, čo „on“ vie, je skôr text než výrok a intenzionálna teória R. Carnapa*) založená na modálnosti ma nikdy nepresvedčila o opaku). Navyše, aj ten najnudnejší matematický text, o ktorom som uvažoval, obsahoval iba zlomok tvrdení preložiteľných perfektne do predikátovej logiky. Značná časť každého textu obsahovala, či už explicitne alebo implicitne, informácie o štruktúre dôkazu, atď.; teda veci, ktoré možno považovať za istý typ procedurálnej informácie.

Prirodzenou logickou reprezentáciou podstatných mien sa javili byť individuové premenné (s definičnými obormi závisiacimi na konkrétnych podstatných menách). Pokračoval som v tomto smere ďalej a na reprezentáciu významu zložitých nominálnych skupín som zaviedol zvláštny druh podriadených premenných, ktorých obory záviseli na okamžitých hodnotách iných premenných [9]. Vyzeralo to nemotorne.

*) Teória je prezentovaná napr. v knihe CARNAP, R.: *Meaning and necessity*. The University of Chicago Press, Chicago, 1955 (pozn. prekl.)

Po niekoľkých rokoch bolo objavené jednoduchšie a všeobecnejšie riešenie, v ktorom sa na reprezentáciu zložiek, nevytvárajúcich úplné vety, použili zvláštne bezvýznamové symboly, ktoré boli podobné kvantifikátorom. A v tom istom čase sa podarilo skupine lingvistov [10], ktorých nezaujímal predikátový počet, nájsť formálny opis istého počtu sémantických ekvivalencií, o ktorom som sa domnieval, že bol odvodený z nejakej mne dovtedy neznámej logickej reprezentácie. Vyskytlo sa viac príkladov úspešnej sémantickej manipulácie s fragmentami prirodzeného jazyka, napr. v jazyku COBOL, alebo pri komunikácii s bázami údajov. Logici ich však s opovrhnutím odmietali [11], lebo predstavovali iba špeciálne prípady a nenaznačovali cestu ku všeobecnému riešeniu. (Dnes sa domnievam, že pre prirodzený jazyk nie je možné žiadne „všeobecné“ riešenie, pretože ide skôr o súbor veľkého počtu relatívne nezávislých systémov zdieľajúcich spoločný mechanizmus nízkej úrovne než o jediný premyslený celok.)

Pokiaľ ide o logickú verifikáciu programov, ukázalo sa, že napísať špecifikácie v predikátovej logike nie je o nič ľahšie než jednoducho napísať samotné programy. Kedysi som si myslel, že nám vtedy chýbal vhodnejší špecifikačný jazyk, obsahujúci viac „syntactickej omáčky“ [6], pokračovať som však v tomto smere už nedokázal. Na druhej strane však E. Tyugu a ďalší, všetko nematematici, objavili veľmi účinný prístup [12] k automatickému programovaniu, založený na kombinácii istých procedurálnych „výpočtových modelov“. Keď som sa však s ich myšlienkou oboznámil, bol som sklamaný. Vytvárali nové programy zo starých.

V automatickom dokazovaní viet sa neobjavila žiadna efektívna všeobecná

metóda a dôraz sa presunul na počítačmi asistované dokazovanie, aby sa tak umožnilo ľudskému používateľovi suplovať isté mystické funkcie („intuíciu“), ktoré automatické systémy nemajú. (Dnes verím, že táto chýbajúca funkcia je skôr procedurálnosť než jasnozrivosť.) Existovalo niekoľko úspešnejších prác z oblasti dokazovania viet pre ohraničené druhy problémov (napr. formálna manipulácia) alebo z oblastí systémov založených bezprostredne na reprezentácii matematických tvrdení v prirodzenom jazyku (s ohraničenou množinou inferenčných pravidiel a s priamou indikáciou ich použitia; také pravidlá sa dajú nájsť v texte napísanom v prirodzenom jazyku, ale v logickej reprezentácii sa strácajú).

V oblasti programovania som sa obvykle zaoberal navrhovaním a implementáciou rôznych problémovo orientovaných jazykov. Mal som teda príležitosť vytvoriť programové konštrukty, ktoré stáli bližšie ku spôsobu rozmyšľania (a hovorenia) v istej aplikačnej oblasti než ku konvenčným programovým konštruktom. Niektoré časti mojej práce, napr. rozpracovaný systém porovnávania so vzorkou, boli paralelou istých trendov vo vývoji jazykov umelej inteligencie, o ktorých som sa dozvedel oveľa neskôr. Táto práca mi ukázala hlbokú súvislosť medzi prirodzenými a programovacími jazykmi a priviedla ma k použitiu programových konštrukcií spolu s logickými ako reprezentácií významov konštrukcií v prirodzenom jazyku. Veľmi skoro som dospel k myšlienke reprezentovať význam vety v pojmoch boolovskej procedúry s vedľajšími efektmi. Zdalo sa mi totiž, že tento mechanizmus by mohol vysvetliť, ako sa používajú zámená. Pokus ale zlyhal. Práve v tejto oblasti som (okolo roku 1973) narazil na príklad, ktorý rozhodným spô-

sobom prispel k tomu, že som opustil logiku.

Mal som vyvinúť simulačný jazyk pre triedu ekologických systémov zahŕňajúcich niekoľko populácií rýb (každú s niekoľkými vekovými skupinami) spolu s ich rastom, množením, výživou, dravosťou atď. Každá veková skupina každej populácie mala niekoľko numerických charakteristík, ktorým bolo treba v počítači priradiť pamäťové miesta. To boli jediné objekty majúce zmysel z hľadiska počítača, nie však z hľadiska používateľa. Používateľ dokonca ani nemusel vedieť o niektorých hodnotách, ktoré boli bezprostredne použité pri simulácii. Pre neho mal zmysel pojem populácia rýb a niektoré explicitne či implicitne určené „zákonitosti“, ako „počet potomkov je rovný populácii násobenej plodnosťou“ alebo „plodnosť je takou a takou funkciou váhy istej ryby“ a pod. Dôležité je, že zákonitosť je špecifikáciou vzťahov medzi niektorými charakteristikami populácie bez odkazov na jej ďalšie charakteristiky. Jediný konvenčný spôsob reprezentácie zákonitostí spočíval v zavedení akejsi univerzálnej štruktúry pre všetky populácie, ktorá mala obsahovať úplnú množinu charakteristík, a to bez ohľadu na skutočnosť, že iba malý počet spomedzi týchto charakteristík bol použiteľný pre konkrétnu populáciu a vekovú skupinu. K takémuto riešeniu som sa však staval odmietavo. Chcel som totiž zachovať systém otvorený pre nové zákonitosti a nové charakteristiky. Po asi ročnom tápaní som dospel k riešeniu, ktoré bolo pre mňa úplne neobvyklé.

Reprezentáciou populácie bolo jednoducho meno (postupnosť znakov), ktoré sa dalo kombinovať s *menom* charakteristiky a s číslom vekovej skupiny, čím bolo vytvorené označenie pamäťového miesta pre zodpovedajúcu hodnotu. Zákonitosť

bola explicitne aplikovaná na konkrétne meno populácie, ale pre túto zákonitosť boli charakteristické mená zahrnutých charakteristík. Nové pamäťové miesto sa v pracovnej pamäti vytvorilo vždy, keď bolo syntetizované dosadenie istej kvantity. Skutočnosť, že aplikácia rôznych zákonitostí sa vzťahovala na tú istú kvantitu, bola zrejma z totožnosti dosadení, a nie naopak.

Skutočnosť, že na dosiahnutie zmysluplného výsledku som musel pracovať s menami ako s bezvýznamovými postupnosťami znakov namiesto pridržania sa ich významu a že výsledky „o chovaní“ som odvodil priamo z jazyka, bez akejkoľvek sprostredkujúcej „sémantickej“ reprezentácie, ma naplnila údivom. Vyzeralo to, ako keby sa na ceste od textu ku správaniu lingvistický znak odrazu stratil, a to po odstránení jeho neporušenosti počas istých syntaktických transformácií, ale v poslednej chvíli sa odrazu opäť ukázal, teraz už však ako fyzický objekt. Mohol by som uviesť viac príkladov tohto druhu (pri nachádzaní súčtu dvoch dekadických čísel musíme napríklad narábať s ich číslicami). Porovnal som túto skutočnosť aj s analýzou významu prídavných mien v prirodzenom jazyku (*regulárna* pyramída nemusí byť nutne *regulárny* mnohosten; *dobrý* matematik nemusí byť *dobrym* prednášateľom). V tomto prípade význam prídavného mena závisí skôr od *slova* než od *objektu*, ktorý spresňuje. Aj niekoľko príkladov z obdobia mojich predchádzajúcich lingvistických výskumov sa ukázalo teraz v novom svetle. To všetko nasvedčovalo tomu, že som dosiahol bod, v ktorom obvyklá matematická abstrakcia stráca na svojej účinnosti, stav, v ktorom je nevhodné hovoriť naďalej o akejsi abstraktnej entite aj v prípade, keď sa jedná o skutočný objekt a kedy je potrebné

brať v úvahu mentálnu symbolickú reprezentáciu objektov.

Objekty (presnejšie ich mentálne reprezentácie) som začal chápať ako množiny menných atribútov, ktorých hodnoty možno vyberať alebo meniť špecifikovaním ich mien, podobne ako množiny údajov a katalogizované procedúry v systéme OS/360 firmy IBM. Takýto druh reprezentácie umožňuje i definovanie rôznych očakávaní a spôsobov, ako sa s nimi vypořadať. Toto všetko je pochopiteľne v protiklade so štýlom logiky: prečo by nemohli byť atribútmi pravouholníka diagonála a plocha namiesto základne a výšky? A ak sa rozhodneme pre prvú možnosť, ako vykonať zámenu?

Myslím si, že úlohou (podstatného) mena v jazyku nie je špecifikovať pevný objekt alebo triedu objektov, ale skôr fungovať ako akýsi selektor, vyberajúci v rôznych kontextoch alebo situáciách isté objekty s (regulárne špecifikovanými) funkciami. Takto špecifikované mená môžu potom definovať ďalšie štruktúry mien a atribútov. Toto pokladám za jednu alternatívu Carnapovej teórie extenzie a intenzie.

V spomenutom simulačnom systéme pre ryby neboli zákonitosti procedúrami. Boli to statické schémy (podobné programovým konštrukciám typu makro) a používali sa na vytvorenie zoznamu „usporiadateľných prvkov“; boli potom usporiadané a spracované v pevnej postupnosti. Neskôr som však začal budovať procedurálne systémy, v ktorých výber podľa mena mohol byť realizovaný procedúrami, a navyiac, vybratá hodnota mohla byť ďalšou procedúrou. Môžem povedať, že voľnosť a flexibilita takéhoto prístupu je veľmi cenná. Môj pocit iste zdieľajú všetci používatelia jazyka LISP. Pre niektoré praktické účely sa dajú procedúry

považovať za priame reprezentácie významov.

O sile procedurálnej reprezentácie poznatkov som sa viac naučil z prác z oblasti umelej inteligencie, konkrétne z prác C. Hewitta, T. Winograda a samozrejme z Minského teórie rámcov*). A to bolo zavŕšením môjho prechodu k proceduralizmu (okolo roku 1976).

Zostáva ešte zodpovedať otázku, prečo neboli algoritmy ako vhodná forma poznatkov akceptované už skôr. Odpoveď je, že matematický pojem algoritmu je príliš hrubý pre účely reprezentácie poznatkov. Častejšie poznáme neformálnu metódu než matematický algoritmus. Čo je základný rozdiel medzi nimi a o čo je treba rozšíriť naše metódy programovania, aby sme mohli postihnúť tento rozdiel a prekonať ho? Obvykle je navrhovaný nedeterminizmus, čo je veľmi jednoduché rozšírenie pojmu algoritmus.

Iný obľúbený návrh je, aby sa neformálna metóda mohla vzťahovať na podciele, pre ktoré však nedefinuje spôsoby, ako ich dosiahnuť. Matematický pojem algoritmu sa však rovnako vzťahuje aj na

*) Autor tu má na mysli pravdepodobne nasledujúce práce:

HEWITT, C.: *Description and the theoretical analysis (using schemata) of PLANNER: a language for proving theorems and manipulating models in a robot*. MIT Artificial Intelligence Laboratory. Memo No. 251, MIT, Cambridge (Mass.), 1972

WINOGRAD, T.: *Understanding natural language*. Academic Press, New York, 1972

WINOGRAD, T.: *Frame representation and the declarative-procedural controversy*. In: *Representation and understanding* (BOBROW, D. G. and COLLINS, A., eds.) Academic Press, New York, 1975

MINSKY, M.: *A framework for representing knowledge*. In: *The psychology of computer vision* (WINSTON, P. H., Ed.) McGraw-Hill, New York, 1975 (Pozn. prekl.)

nedefinované podciele: pokús sa pridať písmeno A k slovu, ak sa v tvom pere minul atrament, a pod. Keby vzťah k podcieľom bol hlavným rozdielom medzi formálnymi a neformálnymi metódami, ľahko by sme mohli použiť pojem relatívnej rekurzívnej a definovať nejakú neformálnu metódu ako schému redukcie na podciele.

Navrhujem inú interpretáciu tohoto rozdielu. Algoritmus v matematickom zmysle je úplne samostatný, a akonáhle boli špecifikované údaje, nepotrebuje už žiadnu ďalšiu informáciu. Na rozdiel od tohto skutočná procedúra (a v istom rozšírenom chápaní aj moderný počítačový program) môže získavať informácie z okolia spôsobom, ktorý nevyžaduje vopred špecifikáciu. Ak poviem, že procedúra definuje nejaký podcieľ, znamená to, že sa pokúša vybrať metódu na dosiahnutie tohoto podcieľa z akéhosi nešpecifikovaného zdroja, t.j. zo svojej vlastnej pamäti alebo z okolia, v ktorom je vyvolaná. Tento výber môže mať formu volania podľa mena (naschvál som tu nehovoril o volaní pomocou porovnávania so vzorom). Táto organizácia procedúr sa dá opísať inými slovami ako modularita: iba jeden modul musí byť v danom okamihu definovaný alebo zmenený, zvyšok je súčasťou okolia.

Zdá sa, že existuje aj tretí rozdiel medzi neformálnymi metódami a algoritmami. Spočíva v použití nejakého typu rozpoznávania obrazcov (pravdepodobne rámcová identifikácia, ale nie klasické porovnávanie obrazcov) za účelom identifikácie situácie a následného výberu akcie. Je pravdepodobné, že obvyklé deklaratívne poznatky vstupujú do všeobecnej schémy práve tu.

Vo svetle procedurálnej organizácie poznatkov nestrácajú deklaratívne poznatky svoj význam. Treba si však pamätať, že

stoja na vrchole zložitého procedurálneho systému a že je skôr šťastnou náhodou, ak ich môžeme získať v takejto všeobecnej a účinnej podobe. Občas sa aj divím, ako sme vôbec mohli na takúto podobu prísť.

Literatúra

- [1] A. A. MARKOV: *Teorija algorifmov*. Trudy Matem. inst. im. V. A. Steklova 42, 1954.
- [2] B. M. LEJKINA, T. N. NIKITINA, M. I. OTKUPČIKOVA, S. JA. FITALOV, G. S. CEJTIN: *Sistema avtomatičeskogo perevoda, razrabatyvaemaja v gruppe matematičeskoj lingvistiki VC LGU*. Naučno-techničeskaja informacija No. 1, 1966, 40—50.
- [3] G. S. CEJTIN: *Odin sposob izloženiya teorii algorifmov i perečisljmych množstv*. Trudy Matem. inst. im. V. A. Steklova 72, 1964, 69—98.
- [4] G. S. CEJTIN: *O verchnich granicach perečisljmych množstv konstruktivnyh veččestvennyh čisel*. Trudy Matem. inst. im. V. A. Steklova 113, 1970, 102—172.
- [5] G. S. CEJTIN: *O logičeskom podchode k avtomatizaciji programirovanija*. Tezis dokladov Vsesajuznoj konferenciji po problemam teoretičeskoj kibernetiki 9—13 ijunija, Novosibirsk, 1969, 5—6.
- [6] G. S. TSEYTIM: *Some features of a language for a proof-checking programming system*. In: Proc. International Symposium on Theoretical Programming, LNCS vol. 5, Springer-Verlag, Berlin, 1974, 394—407.
- [7] G. S. CEJTIN: *Logiko-matematičeskij podchod k postrojeniju ekonomiko-informacionoj sistemy*. Metody vyčislenij, vyp. 6, Izdatelstvo LGU, 1970, 107—127.
- [8] G. S. CEJTIN: *Jazyk matematičeskoj logiki kak sredstvo issledovanija semantiki jestestvennogo jazyka*. Problemy prikladnoj lingvistiki (Tezisi meždunarodnoj konferenciji časť 2) MGPIIJa, Moskva, 1969, 326—335.
- [9] G. S. CEJTIN: *O promežutočnom etape pri perevode s jestestvennogo jazyka na jazyk isčislenija predikatov*. Tezis dokladov konferenciji po obrabotke informaciji, mašinomu perevodu i avtomatičeskomu čteniju teksta, VINITI, Moskva, 1961, 107—111.

- [10] A. K. ŽOLKOVSKIJ, I. A. MELČUK: *O semantičeskom sinteze*. Problemy kibernetiki 19, 1967, 177—238.
- [11] G. S. TSEJTIM: *Features of natural languages in programming languages*. Proc. Fourth International Congress for Logic, Methodology and Philosophy of Sciences (SUPPES, P., HENKIN, L., MOISIL, G. C., JOJA, A., eds.) North-Holland, Amsterdam, 1973, 215—222.
- [12] K. A. TINN, E. CH. TYUGU, M. I. UNT: *Sistema modulnogo programirovanija dlja CVM Minsk-22. VKP-2*, Trudy Vsesojuznoj konferenciji po programirovaniju (zasedanije G), Novosiborsk, 1970, 23—39.

*Preložili: Peter Mikulecký
a Jozef Kelemen*

Doslov redakce

Článek je zajímavým pohledem do „matematického zákulisí“. Ve formě autobiografické úvahy uvádí různé postoje k možnostem vyjádření matematických objektů a poznatků a práce s nimi. Autor aktivně a úspěšně působil v různých oblastech matematiky a computer science — počínaje logikou a konstruktivní matematikou, přes matematickou lingvistiku až po verifikaci programů a návrhy problémově orientovaných programovacích jazyků. Zde všude je otázka reprezentace znalostí a jejich použití velmi podstatná. Na příkladu svého vlastního matematického vývoje autor předvádí přechod od deklarativní reprezentace poznatků (v predikátové logice nebo jiném formálním systému) k jejich procedurálnímu chápání (tedy k víře, že vhodnou formou reprezentace poznatků jsou algoritmy). Možná, že argumentace pro procedurální reprezentaci nevyznívá vždy zcela přesvědčivě, možná, že oblasti, v nichž se vede, jsou pro široký okruh čtenářů poně-

kud speciální. Na druhé straně se domnívám, že podstata článku je především v položených otázkách — o čem matematik přemýšlí, jak své objekty rozpoznává a popisuje a jak s nimi pracuje. Tyto otázky jsou jistě důležité pro každého matema-

tika bez ohledu na jeho profesionální zaměření. Vylíčení způsobu, jakým se s nimi postupně vyrovnával autor článku, může pak čtenáře inspirovat k vlastnímu zamyšlení a k ujasnění a formulaci jeho postoje.

Zdeněk Renc

O obsahu a významu matematiky panuje řada falešných představ. Především musím zdůraznit, že matematika není pouze nástrojem kvantitativního vyhodnocování. Poskytuje stejně jako jiné vědy (a troufám si říci, že ještě mnohem víc než jiné vědy) prostředky také pro kvalitativní analýzu. Posledních sto až sto padesát let proběhlo dokonce ve znamení bouřlivého rozvoje kvalitativních metod. Asymptotické teorie, teorie diferenciálních rovnic, kombinatorická topologie, funkcionální analýza, teorie množin a mnohé další disciplíny, to všechno jsou především teorie kvalitativní.

V matematice se dále vždy uplatňuje i neformální, intuitivní prvek. Představa, že matematika je něco navýsost dokonalého, neomylného, se ukazuje jako zastaralá. Dnes, kdy se v matematice projevilo mnoho logických rozporů (např. v teorii množin) a zejména kdy se jasně ukázalo, že nelze vybudovat exaktní základy aritmetiky, začíná být zřejmé, že se matematika jako věda liší od ostatních přírodních věd vlastně daleko méně, než se předpokládalo.

Matematika přesto zůstává matematikou. Za tisíciletí své existence vytvořila podivuhodnou kulturu myšlení a abstraktní jazyk, umožňující jednotně popsat i velmi rozdílné procesy. Proto se také říká, že úroveň rozvoje dané vědecké disciplíny charakterizuje zejména její schopnost užívat matematických metod, tj. schopnost napojit se na systém univerzálních výzkumných postupů, které matematika jako celek představuje.

Ale ať je matematika, tato matka věd, jakkoli dokonalá, bezprostředně plní jen jedinou funkci: zpracovává informace tak, aby se podle nich dalo rozhodovat.

Nynější rozvoj výpočetní techniky přinesl nové možnosti pro uplatnění matematiky. Metody zpracování informací se značně zefektivnily a možnosti aplikovat matematické postupy se tím zásadně změnily a rozšířily. Díky tomu dospěla řada vědních oborů bez obtíží do „věku zralosti“ (měřeno rozsahem využití matematiky). Platí to především o ekonomii. Zvláště důležité však je, že se tím otevírají nové možnosti také pro mezioborové výzkumy.

A tak se mění nejen role a postavení matematiky v životě lidské společnosti, ale i představy o obsahu matematiky jako vědního oboru.

Problémy matematického modelování — sestavení systému vzájemně propojených modelů — jsou ústředním bodem celé té bezpříkladně rozsáhlé výzkumné činnosti, která se začíná rozvíjet ve všech vyspělých zemích. Matematik přitom není pouze pasivním účastníkem. Tam, kde je to třeba, začíná hrát roli architekta či konstruktéra, protože všechny konkrétní výzkumy v oboru biologie, ekonomiky, demografie, geochemie atd. musí do sebe zapadat jako kvádry v budově, která se staví. Mezi dosud započatými výzkumy ještě zdaleka není dostatečná návaznost, a do značné míry jsou dokonce výsledkem jen náhodného zájmu jednotlivých specialistů nebo jejich intuice. Proto se mnohé výzkumy překrývají, zatímco v celé řadě důležitých směrů vědeckého výzkumu se nepracuje vůbec. A přitom je rok od roku zřejmější, jak naléhavě bychom potřebovali vytvořit ucelený systém modelů, které by nám umožnily přesněji odhadovat vývojové perspektivy lidské společnosti a koncipovat mezinárodní programy.

N. N. Moisejev