Vladimir Vladimirovich Uspenskij
Algoritmus

Pokroky matematiky, fyziky a astronomie, Vol. 8 (1963), No. 4, 197--210

Persistent URL: http://dml.cz/dmlcz/138279

ALGORITMUS

(Filosofskaja enciklopedija)

V. USPENSKIJ

Algoritmus je jedním ze základních pojmů logiky a matematiky. Je to přesný předpis, jímž je jednoznačně určen početní proces vedoucí od počátečních dat, jež mohou být variabilní, k hledanému výsledku.

Slova, výpočty, početní, resp. vyčíslitelný, s nimiž se v souvislosti s tím setkáváme, není na místě chápat v úzkém smyslu číselných výpočtů. Tak již ve školském výkladu algebry se mluví o počítání s písmeny, a i když zde písmena ještě zastupují čísla, objevují se již v aritmetických výpočtech symboly neoznačující žádné veličiny: závorky, znak rovnosti, znaky aritmetických operací. Můžeme jít v tomto směru dále a zkoumat výpočty s libovolnými symboly a jejich kombinacemi; právě v tomto širokém smyslu chápeme termíny výpočty, vyčíslení při popisu pojmu algoritmus. Tak můžeme mluvit o algoritmu překladu z jednoho jazyka do druhého, o algoritmu práce vlakového dispečera (zpracovávajícího informaci o pohybech vlaků v příkazy) a o jiných případech algoritmického popisu procesů řízení, jež zkoumá kybernetika.

VÝZNAM POJMU ALGORITMUS

Slovo algoritmus vzniká v 9. století (je odvozeno z výrazu algoritmi, který je zase latinskou transkripcí — provedenou zřejmě ve dvanáctém století — arabského jména chorezmijského matematika AL-CHOREZMI). V dnešní době se setkáváme s nejjednoduššími algoritmy již na základní škole: jsou to algoritmy aritmetických operací. (Ve středověké Evropě se algoritmem nazývala právě soudobá školská aritmetika, tj. desítková poziční soustava zápisu čísel a umění počítání v této soustavě, neboť Al-Chorezmiho pojednání bylo jedním z prvních, ne-li vůbec první, jehož zásluhou se Evropa seznámila s poziční soustavou.) Zdůrazněme, že na základní škole se dbá především algoritmické stránky sčítání. Rovněž říkáme-li, že člověk dovede sčítat, nemáme zpravidla na mysli, že pro dvě libovolná čísla dokáže dříve či později nalézt jejich součet, nýbrž že ovládá nějaký jednotný postup sčítání, tj. algoritmus sčítání, jejž dovede aplikovat na libovolné dva konkrétní zápisy čísel. (Příkladem takového algoritmu je známý algoritmus sčítání čísel po sloupcích.)

S algoritmy se ve vědě setkáváme na každém kroku: umět řešit úlohu obecně znamená vždy v podstatě ovládat nějaký algoritmus. Pojem úlohy v obecném tvaru se zpřesňuje pomocí pojmu *hromadného problému*. Termínem problém můžeme rozumět úlohu nalézt objekt mající jisté vlastnosti; tento objekt nazýváme řešením problému. (Speciálně: řešením problému nalézt kladnou či zápornou odpověď na nějakou otázku je odpověď ano nebo ne na položenou otázku.) Problém je neřešitelný, nemá-li řešení, tj. neexistuje-li objekt, jenž by měl požadované vlastnosti. Je proto zřejmé, že neřeši-

telnost problému nás neopravňuje k agnostickým závěrům; naopak, zjištění neřešitelnosti konkrétního problému je důležitý akt poznání. Hromadný problém vzniká shrnutím celého souboru jednotlivých, jedinečných problémů a záleží v požadavku nalézt obecnou metodu (tj. algoritmus) jejich řešení. Pod nerozhodnutelností hromadného problému rozumíme pak neřešitelnost úlohy najít odpovídající algoritmus. Hromadné problémy jsou mimořádně charakteristické a závažné pro logiku a matematiku. Dokonce i řešení jedinečných problémů je často cenné právě proto, že zároveň umožňuje nalézt obecnou metodu řešení celé třídy problémů; na druhé straně sama formulace hromadného problému obsahuje v sobě akt přeměny určité třídy problémů v jedinečný problém — problém nalezení algoritmu pro řešení všech problémů z této třídy; zde se projevuje souvislost takových kategorií dialektiky jako jedinečné, zvláštní a obecné. Úloha, jakou hrají hromadné problémy, určuje význam algoritmů.

Zjištění, že určitý hromadný problém je nerozhodnutelný (tj. že neexistuje jednotný algoritmus, jenž by umožnil nalézt řešení všech jedinečných problémů dané série), je velice závažným aktem poznání, jenž ukazuje, že k řešení konkrétních jedinečných problémů je principiálně zapotřebí metod specifických pro každý takový problém. Existence nerozhodnutelných hromadných problémů je takto konkrétním ztělesněním nevyčerpatelnosti procesu poznání.

Konkrétní jevy, které vedly k vytvoření pojmu algoritmus, zaujímaly odedávna důležité místo ve vědě. Mnoho úloh vznikajících v matematice a logice záleželo v hledání různých konstruktivních metod. Hledání takovýchto metod, jež nabylo na intenzitě zvláště v souvislosti s vytvořením vhodné matematické a logické symboliky, jakož i uvědomění si toho, že v řadě případů tyto metody principiálně neexistují — to vše bylo mocným činitelem ve vývoji vědeckého poznání. Uvědomění si nemožnosti řešit libovolnou úlohu přímým výpočtem vedlo v 19. století k vytvoření koncepce teorie množin. Teprve po období bouřlivého rozvoje této koncepce, v jejímž rámci vůbec nevzniká otázka konstruktivních metod v jejich soudobém pojetí) se v posledních desetiletích ukázalo, že je možno se znovu vrátit k otázkám konstruktivnosti, ale již na nové úrovni, obohacené pojmem algoritmu, který zatím vykrystalizoval. (To je opět další ilustrace Leninovy teze o spirálovitém charakteru vývoje poznání.) A i když pojem algoritmus není abstrakcí, jež by šla tak daleko jako např. pojem množina, nelze pokládat za náhodné, že přesný význam pojmu historicky prvého byl explicitně formulován později než u druhého.

PŘÍKLADY ALGORITMŮ

Podobně jako pojmy množina, přiřazení, přirozené číslo, relace atp. je i pojem algoritmus prvotním logicko-matematickým pojmem (jednou z kategorií logiky a matematiky). Nelze jej formálně definovat jednoduššími pojmy, nýbrž abstrahujeme jej (tak jako i jiné matematické kategorie) bezprostředně ze zkušenosti. Pojem algoritmus si můžeme osvojit pouze na příkladech.

Příklad 1. Možnými výchozími daty jsou konečné neprázdné kombinace sestavené z čárek (I), tj. objekty I, II, III atd. Algoritmus je dán těmito pravidly (která je třeba provádět počínaje pravidlem 1):

1. Podtrhni čárku, která je první zleva, a přejdi k provedení pravidla 2.

2. Nadepiš pruh nad čárku, která je první zprava, a přejdi k provedení pravidla 3.

3. Pohleď na podtrženou čárku, a není-li nahoře opatřena pruhem, přejdi k provedení pravidla 4.

4. Pohleď na čárku, která následuje bezprostředně za podtrženou čárkou; není-li nahoře opatřena pruhem, přejdi k provedení pravidla 5; je-li nahoře opatřena pruhem, přejdi k provedení pravidla 7.

5. Přenes spodní pruh z podtržené čárky na čárku bezprostředně následující a přejdi k provedení pravidla 6.

6. Přenes horní pruh z čárky jím opatřené na čárku bezprostředně předcházející a přejdi k provedení pravidla 3.

7. Smaž čárku opatřenou horním pruhem a všechny čárky, které za ní následují, a přejdi k provedení pravidla 8.

8. Smaž spodní pruh u podtržené čárky; to, co zbylo, je výsledek.

Aplikujeme-li tento algoritmus na kombinace I I I I, zvolenou za výchozí (vstupní) údaj, dostaneme postupně: podle pravidla 1 : I̲ I I I, podle pravidla 2 : I̲ I I I̅, podle pravidel 3, 4, 5 : I I̲ I I̅, podle pravidel 6 , 3, 4 : I I̲ I̅ I̅, podle pravidla 7 : I I̲, podle pravidla 8 : I I (výsledek). Pokusme se však aplikovat algoritmus na kombinaci I I I! Dostaneme: podle pravidla 1 : I̲ I I, podle pravidla 2 : I̲ I I̅, podle pravidel 3, 4, 5 : I I̲ I̅, podle pravidla 6 : I I̲ I; dále máme přejít k provedení pravidla 3, ale pravidlo 3 lze provést jen za podmínky, že podtržená čárka není opatřena horním pruhem. Pro situaci, jež se takto vytvořila, neobsahuje tedy algoritmus pokyny, jak postupovat dále; došlo k tzv. bezvýslednému zastavení (k zastavení, s nímž není spojeno obdržení výsledku). Snadno si povšimneme, že obecně náš algoritmus dá výsledek tehdy, aplikujeme-li jej na libovolnou kombinaci sudého počtu čárek, a že v tom případě se výsledná kombinace skládá z poloviního počtu čárek; algoritmus nedá výsledek, aplikujeme-li jej na libovolnou kombinaci lichého počtu čárek.

Příklad 2. V logice a matematice nazýváme každý konečný soubor znaků *abecedou*, znaky, z nichž se skládá, *písmeny* abecedy a konečnou (včetně prázdné) posloupnost za sebou napsaných písmen nějaké abecedy nazýváme *slovem* v této abecedě. Tak např. arabské číslice tvoří abecedu a každý desítkový zápis celého čísla je slovem v této abecedě.

Zkoumejme abecedu (*a*, *b*) ze dvou písmen: *a* a *b*. Příklady slov v této abecedě: *b*, *ab*, *bba*, *aababb* atd. Smluvme se, že přechod od jednoho slova v této abecedě ke druhému slovu v téže abecedě budeme nazývat přípustným, jestliže ho lze provést použitím jednoho z těchto dvou pravidel:

1. Má-li slovo tvar *aP*, kde *P* je libovolné slovo, přejdeme ke slovu *Pb*.

2. Má-li slovo tvar *baP*, kde *P* je libovolné slovo, přejdeme ke slovu *Paba*.

Dále stanovme tento předpis: „Vyjdi z nějakého slova (zvoleného za počáteční

data) a prováděj přípustné přechody do té doby, dokud nevyjde slovo tvaru *aaP*; když vyjde slovo tohoto tvaru, odstraň v něm prvá dvě písmena a to, co zůstane, je výsledek." Protože pokaždé můžeme realizovat nejvýše jedno pravidlo přechodu, je uvedený předpis algoritmem, jehož možnými počátečními daty jsou slova v abecedě (*a*, *b*). Zvolme za počáteční data slovo *babaa*. Podle pravidla 2 dostaneme *baaaba*. Užijeme znovu pravidla *2* a dostaneme *aabaaba*. Podle našeho předpisu se musíme zastavit; výsledkem (aplikace algoritmu na slovo *babaa*) je *baaba*. Zvolme za počáteční data slovo *baaba*. Podle pravidla 2 dostaneme *abaaba*. Podle pravidla 1 dostaneme *baabab*. Dále dostaneme postupně *abababa*, *bababa*, atd. Lze dokázat, že tento proces nemůže být nikdy zakončen (tj. nikdy nevzniká slovo, jež by začínalo dvěma písmeny *a*, a pro každé ze získaných slov bude možno uskutečnit přípustný přechod). Algoritmus tedy nedává výsledek, aplikujeme-li jej na slovo *baaba*. Zvolme za počáteční data slovo *abaab*. Postupně dostaneme *baabb*, *abbaba*, *bbabab*. Dále nelze užít ani jednoho z pravidel 1, 2 a zároveň jsme nedostali výsledek. Proto aplikace algoritmu na slovo *abaab* rovněž nedává výsledek.

ZÁKLADNÍ RYSY ALGORITMU

Podle A. A. Markova jsou pro algoritmus charakteristické tyto základní rysy: a) *určitost* (jednoznačnost) algoritmického předpisu, která záleží v jeho přesnosti, neponechávající místa pro libovůli, a v jeho obecné srozumitelnosti (důsledkem této jednoznačnosti předpisu je determinovanost algoritmu: každé stadium procesu určuje jednoznačně následující stadium); b) *hromadnost*, záležející v tom, že každý algoritmus může vycházet z výchozích dat, jež jsou v jistých mezích variabilní; c) *rezultativnost* (účelnost), záležející v tom, že je zaměřen na získání hledaného výsledku. Determinovanost algoritmu zabezpečuje to, že jedna osoba jej může sdělovat druhé tak, že tato druhá osoba dovede provádět algoritmus bez účasti prvé; táž vlastnost determinovanosti umožňuje svěřit provedení algoritmu stroji. Hromadnost algoritmu předpokládá, že existuje nějaký soubor (pro různé algoritmy obecně různý) možných výchozích dat. Jakým způsobem je tento algoritmus stanoven, je již jiná otázka. Obecně lze předpokládat, že soubor možných výchozích dat odpovídající nějakému algoritmu není stanoven odděleně od algoritmu, nýbrž přímo tvoří přirozeným způsobem součást definice tohoto algoritmu (tak např. pro algoritmus sčítání po sloupcích se příslušný soubor skládá ze všech dvojic zápisů čísel v desítkové soustavě). Když volíme nějaký konkrétní objekt za počáteční data algoritmu, mluvíme o aplikaci algoritmu na tento objekt. Jestliže algoritmus při aplikaci na nějaký objekt dá výsledek, říkáme, že je aplikovatelný na tento objekt. Rezultativnost algoritmu nikterak neznamená, že algoritmus musí být aplikovatelný na libovolný objekt z příslušného souboru počátečních dat (viz příkl. 1 a 2). Zde je vhodné připomenout, že lze sestrojit takový algoritmus, že k němu neexistuje žádný jiný algoritmus, který by u libovolných výchozích dat prvého algoritmu dovedl rozpoznat, zda je na něj prvý algoritmus aplikovatelný, či nikoliv.

ZÁKLADNÍ ABSTRAKCE TEORIE ALGORITMŮ

Ve vědecké praxi se vytvořila řada abstrakcí specifických pro matematiku a logiku. Takovými abstrakcemi jsou především abstrakce *aktuálního nekonečna*, abstrakce *ztotožnění*, abstrakce *potenciální uskutečnitelnosti*. Sovětský vědec A. A. MARKOV ukázal, že dvou naposled zmíněných abstrakcí je zapotřebí při studiu algoritmů. Algoritmický proces je rozdělen na jednotlivé kroky, z nichž každý je podle předpokladu natolik elementární, že možnost jeho faktického uskutečnění nevyvolává pochyb. Zároveň počet těchto elementárních kroků, jakého je zapotřebí pro získání výsledku, může být natolik veliký, že dosažení výsledku můžeme pokládat za prakticky neuskutečnitelné. Avšak představa praktické uskutečnitelnosti či neuskutečnitelnosti určitého počtu kroků je relativní. Mění se s rozvojem početních prostředků (v zásadě se může měnit i představa elementárnosti jednotlivého kroku). Proto v teorii algoritmů abstrahujeme od „praktické uskutečnitelnosti" a pokládáme za uskutečnitelný libovolný konečný počet kroků. Tím připouštíme při studiu algoritmů abstrakci potenciální uskutečnitelnosti: tato abstrakce záleží v tom, že abstrahujeme od reálných hranic našich možností. Rozvoj rychlých elektronkových počítačů rychle posunuje tyto hranice stále dál a dál. To, co bylo jen potenciálně uskutečnitelné včera, stává se prakticky uskutečnitelným dnes. To sbližuje teorii algoritmů s praxí práce na počítacích strojích a umožňuje, že tyto dvě disciplíny se navzájem obohacují. Než je možno stroji předat řešení nějakého souboru úloh, je nutno předběžně sestavit algoritmus řešení. Sestavení takovéhoto algoritmu má zpravidla pricipiální význam (tak např. v problému strojového překladu je hlavní úlohou právě sestavení algoritmu překladu).

Abstrakce potenciální uskutečnitelnosti je nezbytná nejen při zkoumání algoritmických procesů, nýbrž i při zkoumání samých objektů, jež vstupují do těchto procesů (včetně výchozích dat a výsledků). Tak např. abychom mohli mluvit o libovolném přirozeném čísle (přesněji o zápisu tohoto čísla např. v desítkové soustavě), musíme jako předmět našich úvah připustit i zápisy tak velkých čísel, že by se tyto zápisy nevešly na zeměkouli; tak i zde tím, že abstrahujeme od fyzické uskutečnitelnosti takového zápisu, využíváme abstrakce potenciální uskutečnitelnosti tehdy, když chceme uvažovat jakkoliv dlouhá slova v dané abecedě.

Objekty, jejichž sestrojení a zkoumání je možné v rámci abstrakce potenciální uskutečnitelnosti (v protikladu k abstrakci aktuálního nekonečna), nazýváme *konstruktivními objekty*. Jsou jimi přirozená čísla reprezentovaná svými zápisy v nějaké soustavě značení, slova v dané abecedě atd. a také dvojice, trojice a vůbec konečné posloupnosti sestavené ze zápisů čísel, slov v abecedě atd., racionální čísla (jež lze vyjádřit jako dvojice přirozených čísel) aj. Konstruktivními objekty jsou také výrazy tzv. kalkulů anebo formálních systémů, a proto na ně můžeme aplikovat aparát teorie algoritmů. Každý algoritmus (chápaný jako předpis) můžeme pokládat (když tento předpis zapíšeme ve formě kombinace nějakých symbolů) za konstruktivní objekt. Naproti tomu objekty, jež nemůžeme zkoumat bez použití abstrakce aktuálního ne-

konečna, nepatří mezi konstruktivní objekty. Tak např. konstruktivními objekty nejsou reálná čísla (ve smyslu CANTOROVĚ, DEDEKINDOVĚ nebo WEIERSTRASSOVĚ), geometrické body (poněvadž rozbor takové abstrakce, jako je ‚bod', vede k představě bodu jako aktuálně nekonečného systému malých těles) atd. Konstruktivní objekty tvoří přirozeným způsobem soubory: takovým souborem je např. soubor všech slov v dané abecedě a vůbec libovolný soubor nějakého typu z počtu výše vyjmenovaných typů konstruktivních objektů. Každý takový soubor konstruktivních objektů je stanoven předpisem, podle něhož se konstruují objekty, jež do něho patří.

Druhou základní abstrakcí, jíž používáme při zkoumání konstruktivních objektů a algoritmů, je abstrakce ztotožnění. V některých případech mluvíme o dvou objektech jako o stejných. Podmínky stejnosti stanovíme pokaždé podle dané situace. Tak např. když člověk provádí výpočty na papíře, nezáleží na písmu, jakým jsou psány číslice, a nápisy *1647* a 1647 pokládáme za stejné; lze si však představit situace, kdy rozdíl stojatého a kurzívního písma je podstatný (jako např. při chápání smyslu předchozí věty toho článku). Pak už budeme takové dva zápisy pokládat za nestejné, avšak zápisy 1647 a 1647 přesto — v obvyklých případech — za stejné (i když jde o fyzicky různé objekty). Obvykle předpokládáme, že konstruktivní objekty se skládají z určitých dostatečně jednoduchých elementárních částí (podobně jako slova se skládají z písmen), a dva konstruktivní objekty pokládáme za stejné, skládají-li se ze stejných elementárních částí stejně uspořádaných. Bez pojmu stejnosti, na jehož základě pokládáme za stejné např. číslice napsané křídou na tabuli a číslice napsané inkoustem v sešitu, je nemožná výuka. Abstrakce ztotožnění umožuje mluvit o stejných předmětech jako o jednom a témž objektu. Vede k vytvoření pojmu abstraktního objektu: dva stejné konkrétní objekty pokládáme totiž za reprezentanty jednoho a téhož abstraktního objektu. Každý algoritmus aplikovaný na stejné objekty vede také ke stejným objektům. Proto lze mít za to, že každým algoritmem je stanoven nějaký proces transformování abstraktních konstruktivních objektů. Tato vlastnost algoritmů (spolu s determinovaností) podmiňuje jejich opakovatelnost neboli reprodukovatelnost: když byl algoritmus vypracován jako algoritmus na abstraktních konstruktivních objektech, může být opětovně reprodukován pro libovolné konkrétní konstruktivní objekty přípustné pro daný algoritmus.

Z toho, co bylo uvedeno, jasně vyplývá, že výchozí data právě tak jako konečné výsledky vznikající při realizaci některého algoritmu jsou vždy konstruktivní objekty (každý stav algoritmického procesu je konstruktivní objekt!). Nemožnost třebas i potenciálně uskutečnitelných procesů na nekonstruktivních objektech je spjata i se skutečností, že neexistuje způsob, jak tyto objekty rozpoznávat jako stejné či nestejné (srovn. známé tvrzení kybernetiky o přednostech diskrétních forem uchovávání informace před spojitými).

Pokud jde o metody přípustné při zkoumání algoritmů, existují různá hlediska. Podle jednoho z nich, jež zastávají představitelé konstruktivismu v matematice a v logice, musí se rozvíjení teorie algoritmů dít v rámci abstrakce ztotožnění a abstrakce potenciální uskutečnitelnosti, neboť k vytvoření pojmu algoritmu tyto dvě abstrakce

stačí. Druhé hledisko připouští při zkoumání algoritmů libovolné metody, jež jsou vůbec přípustné v logice a v matematice, včetně metod, jež vyžadují abstrakci aktuálního nekonečna. Tak si lze např. představit případ, kdy k důkazu toho, že určitý algoritmus dá při aplikaci na určitý objekt výsledek, bude zapotřebí využít zákona vyloučeného třetího, spjatého úzce s abstrakcí aktuálního nekonečna.

ZÁKLADNÍ POJMY TEORIE ALGORITMŮ

K základním pojmům vznikajícím na základě pojmu algoritmu patří pojem *vyčíslitelné funkce*, *rozhodnutelné množiny**) a *rekurzívně spočetné množiny*. Funkci nazýváme *vyčíslitelnou*, jestliže existuje algoritmus, který tuto funkci vyčísluje v tomto smyslu: a) algoritmus je aplikovatelný na každý objekt patřící do definičního oboru fukce a jako výsledek dá tu hodnotu funkce, jaké tato funkce nabývá, zvolíme-li za její argument tento objekt; b) algoritmus není aplikovatelný na žádný objekt, který nepatří do definičního oboru funkce. Množinu nacházející se v určitém souboru konstruktivních objektů (tj. množinu skládající se z nějakých objektů tohoto souboru) nazýváme *rozhodnutelnou* (vzhledem k tomuto nadřazenému souboru), když existuje algoritmus rozhodující o prvcích této množiny (vzhledem ke zmíněnému souboru) v tomto smyslu: algoritmus je aplikovatelný na libovolný objekt z nadřazeného souboru a jako výsledek dává odpověď na otázku, zda tento objekt patří do zkoumané množiny, či ne. Konečně neprázdnou množinu nazýváme *rekurzívně spočetnou*, když existuje algoritmus dovolující získat (vyčerpat) všechny prvky této množiny v tomto smyslu: a) výsledek aplikace algoritmu na libovolné přirozené číslo existuje a patří do zkoumané množiny; b) každý prvek zkoumané množiny můžeme dostat jako výsledek aplikace algoritmu na nějaké přirozené číslo. Na základě definice zařazujeme také prázdnou množinu obvykle do třídy rekurzívně spočetných množin. Pro jednu a touž vyčíslitelnou funkci (resp. rozhodnutelnou množinu, rekurzívně spočetnou množinu) může platit, že je vyčíslována (resp. že je o jejích prvcích rozhodováno, že její prvky jsou vyčerpány) pomocí různých algoritmů. Z definice plyne, že argumenty a hodnoty vyčíslitelné funkce, prvky rozhodnutelné nebo rekurzívně spočetné množiny jsou vždy konstruktivní objekty. Nahradíme-li konstruktivní objekty (určitého zafixovaného souboru) jejich čísly v libovolném algoritmickém očíslování (tj. v takovém očíslování, pro něž existuje algoritmus, jímž z objektu dostaneme jeho číslo a obráceně), můžeme se, jako se to často dělá v teorii algoritmů, omezit na zkoumání takových vyčíslitelných funkcí, jejichž argumenty a hodnoty jsou přirozená čísla, a jen takových rozhodnutelných a rekurzívně spočetných množin, jejichž prvky jsou rovněž přirozená čísla.

Lze dokázat, že každá rozhodnutelná množina je rekurzívně spočetná. Zároveň se podařilo sestrojit rekurzívně spočetnou, ale nerozhodnutelnou množinu. Tento první konkrétní příklad (uveřejnil jej americký vědec A. CHURCH r. 1936 v článku „O jed-

*) nebo (v češtině): *rekurzívní množiny* (pozn. překl.).

nom nerozhodnutelném problému elementární teorie čísel") neexistence algoritmu (totiž algoritmu, jenž by dovedl rozhodovat o příslušnosti prvků do sestrojené množiny) se stal zdrojem nebo vzorem pro téměř všechny další příklady tohoto druhu. Ukázalo se, že množina je rozhodnutelná tehdy a jen tehdy, když je rekurzívně spočetná ona sama i její doplněk (vzhledem k nadřazenému souboru objektů). Existují tedy takové doplňky rekurzívně spočetných množin, jež samy nejsou rekurzívně spočetné.

SOUVISLOST TEORIE ALGORITMŮ S LOGIKOU

Pojmy rozhodnutelné a rekurzívně spočetné množiny jsou úzce spjaty s klasifikací definic (omezujeme se zde jen na takové definice, z nichž každá definuje objekty určitého typu nebo, což je totéž, určitou třídu objektů). Jak známo, existují dvě základní schémata definic: ‚rodem a druhovým rozdílem' a ‚induktivní'.

Při definici rodem a druhovým rozdílem se určí jistý nadřazený soubor (rod), a ukazuje se vlastnost (druhový rozdíl), která mezi objekty zmíněného souboru odlišuje třídu definovaných předmětů. Předpokládáme-li, že tato definice je konstruktivní, tj. že objekty jsou konstruktivní a že přítomnost či nepřítomnost druhového rozdílu u prvku rodu je algoritmicky rozpoznatelná, ukazuje se, že definovaná množina je rozhodnutelná (a každou rozhodnutelnou množinu lze definovat takovýmto způsobem). Takto lze rozhodnutelné množiny ztotožnit s množinami definovanými konstruktivně rodem a druhovým rozdílem.

Induktivní definice se skládá ze dvou částí: z báze, jež obsahuje určitý výčet objektů, o nichž se prohlásí, že patří do definované třídy, a z induktivní části, v níž se říká, že jestliže objekty takového a takového druhu patří do definované třídy, pak i objekty takového a takového druhu spojené s objekty prvými určitou relací patří také do definované třídy. (Jsou možné i složitější případy tzv. zkřížených (simultánních) definic, kdy se současně vzájemně definuje několik tříd objektů.) Předpokládáme-li, že definice je konstruktivní, tj. že objekty jsou konstruktivní, že výčet výchozích objektů obsažený v bázi je konečný a že v induktivní části obsažená pravidla přechodu od již definovaných objektů k novým objektům jsou algoritmická (v tom smyslu, že přítomnost nebo nepřítomnost relace, o kterou jde v induktivní části, je rozpoznatelná pomocí nějakého algoritmu), dospíváme k pojmu množiny konstruktivně definované indukcí čili (což je synonymum) *efektivně vytvářené* (generované) množiny (neboť tato definice zadává efektivní vytvářející (generující) proces, při jehož rozvíjení vznikají nebo se vytvářejí na jednotlivých etapách definované objekty). Příkladem konstruktivní definice indukcí je definice přípustných šachových pozic (tj. pozic, k nimž může během hry dojít na šachovnici). Báze obsahuje jedinou výchozí pozici. Induktivní část obsahuje pravidla tahů figurami. Množina přípustných pozic je tedy efektivně vytvořitelná. Jiným příkladem efektivně vytvořené množiny je množina všech dokazatelných formulí nějakého formálního systému či kalkulu: báze definice dokazatelných formulí obsahuje axiómy, induktivní část pravidla vyvozování (axiómy se prohlašují za dokazatelné ex definitione a dále se říká, že jsou-li nějaké

libovolné formule dokazatelné, jsou dokazatelné i formule, jež z nich dostaneme uplatněním pravidel vyvozování). Vytvářejícím procesem je zde proces dokazování všech dokazatelných formulí. Konečně proces vyvracení všech vyvratitelných formulí kalkulu je také příkladem efektivně vytvářejícího procesu.

Pojem efektivního vytvářejícího procesu je velmi úzce spjat s pojmem algoritmu. Podali jsme (přibližnou) definici efektivního vytvářejícího procesu, která se opírala o pojem algoritmu. Na druhé straně lze na základě pojmu vytvářejícího procesu definovat ne-li sám pojem algoritmu, tedy v každém případě pojem vyčíslitelné funkce. Nechť totiž je určitý vytvářející proces schopen vytvářet objekty, jež mají tvar dvojic (x, y), a nechť každé dvojice se shodnými prvými členy mají shodné i druhé členy. Pak tento proces definuje funkci $y = f(X)$ takto: funkce je pro objekt x_0 definována tehdy a jen tehdy, když x_0 je prvý člen některé vytvořené dvojice: hodnota x funkce pro argument x_0 se v takovém případě rovná druhému členu této dvojice. Funkce definovaná ve zmíněném smyslu efektivním vytvářejícím procesem je zřejmě vyčíslitelná (abychom našli $f(X)$, musíme proces rozvíjet dotud, dokud nenajdeme dvojici, jejímž prvým členem je x_0). A naopak každou vyčíslitelnou funkci lze definovat pomocí efektivního vytvářejícího procesu. Algoritmické procesy a vytvářející procesy jsou si z logického hlediska blízké. Základy každého z nich tvoří pouze konstruktivní pojmy. Rozdíl mezi nimi záleží v tom, že algoritmický proces se rozvíjí na základě požadavku, kdežto vytvářející proces na základě svolení jednat určitým způsobem. Zde se projevuje rozdíl mezi nutným a možným (v algoritmickém procesu je každá etapa jednoznačně, tj. s nutností, určena předchozí etapou, kdežto při rozvíjení vytvářejícího procesu vzniká po každé etapě jen množina možností pro následující etapu).

Při vhodných zpřesněních pojmu efektivního vytvářejícího procesu se ukazuje, že každá efektivně vytvořená množina je rekurzívně spočetná a naopak. Tato okolnost spolu s výše uvedenými vzájemnými vztahy mezi rekurzívně spočetnou a rozhodnutelnou množinou dovoluje provést tento závěr: Každá třída objektů, která připouští konstruktivní definici rodem a druhovým rozdílem, připouští i konstruktivní definci indukcí, ale ne naopak: existuje třída objektů definovaná konstruktivně indukcí, ale nepřipouštějící konstruktivní definici rodem a druhovým rozdílem; doplněk této třídy objektů (vzhledem k nadřazenému souboru konstruktivních objektů) nepřipouští efektivní induktivní definici. Každý konstruktivní vytvářející proces lze vyjádřit v podobě procesu získávání dokazatelných formulí jistého kalkulu. Proto můžeme za příklad třídy mající právě popsané vlastnosti zvolit třídu všech dokazatelných formulí nějakého kalkulu. Nadto se ukázalo, že tato okolnost zůstává v platnosti pro libovolný dostatečně obsažný kalkul (např. pro predikátový kalkul nebo pro kalkuly, na jejichž základě se formalizuje aritmetika), neboť je-li kalkul dostatečně obsažný, lze v něm vyjádřit libovolný efektivní vytvářející proces. Třída všech dokazatelných formulí takového kalkulu (která je ovšem rekurzívně spočetná) není rozhodnutelná, takže neexistuje algoritmus, který by rozpoznával dokazatelnost formulí kalkulu; v tom smyslu říkáme, že kalkul je nerozhodnutelný. Protože třída všech dokazatelných formulí kalkulu není rozhodnutelná, není k ní doplňková třída všech nedokaza-

telných formulí rekurzívně spočetná, a nemůžeme ji tedy dostat žádným vytvářejícím procesem; speciálně nelze sestrojit takový kalkul, v němž by se vyvracely všechny nedokazatelné formule původního kalkulu a jen ony. Tím spíše všechny tyto nedokazatelné formule nemohou být vyvráceny prostředky samotného původního kalkulu, takže v původním kalkulu existují tzv. nerozhodnutelné, tj. nedokazatelné a zároveň nevyvratitelné formule. V těchto úvahách se můžeme omezit jen na takové formule, jež při obsahové interpretaci kalkulu vyjadřují smysluplné, tj. buď pravdivé, nebo nepravdivé soudy, a můžeme nalézt tedy i mezi takovými formulemi ty, jež jsou nerozhodnutelné. Z toho plyne, že lze ukázat formuli, která vyjadřuje pravdivý soud, ale není dokazatelná v kalkulu; v tom smyslu říkáme, že systém je neúplný. Zdůrazněme, že uvedené úvahy mají obecný charakter, takže vlastnost neúplnosti má každý dostatečně obsažný kalkul.

Pojem nerozhodnutelnosti kalkulu se opírá o pojem algoritmu, a není proto divu, že fakt nerozhodnutelnosti zjišťujeme na základě výzkumů v oblasti teorie algoritmů. Velice podstatnou (na na první pohled možná neočekávanou) je ta okolnost, že k takové obecně logické skutečnosti, jako je neúplnost kalkulů (skutečnosti, jež vyjadřuje zásadní nemožnost úplně formalizovat proces logického vyvozování a kterou poprvé exaktně dokázal rakouský vědec K. GÖDEL již r. 1931 ještě před zpřesněním pojmu algoritmu), lze dojít, jak jsme právě viděli, pomocí prostředků teorie algoritmů. Již tato okolnost sama o sobě ukazuje na obrovské možnosti aplikací teorie algoritmů na otázky logiky. Tyto aplikace se neomezují na uvedený příklad. Již r. 1932 navrhl sovětský vědec A. N. KOLMOGOROV interpretaci konstruktivní logiky, kterou vytvořili intuicionisté, na základě obsahových prostředků nezávislých na předpokladech intuicionismu; Kolmogorov navrhl, aby každá věta konstruktivní logiky byla interpretována jako problém. Pojem problému vyžadoval však konkretizaci, která mohla být podána pouze na podkladě již vypracované teorie algoritnů. Americký vědec S. C. KLEENE (r. 1945) a sovětský vědec J. T. MEDVĚDĚV (r. 1955) předložili každý po jedné třídě problémů vhodných k interpretaci konstruktivní logiky. R. 1956 vytyčil sovětský vědec N. A. ŠANIN novou koncepci, podle níž není třeba, aby každý výrok konstruktivní logiky byl interpretován jako problém.

K tomuto okruhu idejí se přimykají otázky konstruktivizace neboli nacházení konstruktivních analogií klasických matematických pojmů a vět; tyto otázky lze také vyřešit pouze na základě teorie algoritmů. Konstruktivizace základních pojmů matematické analýzy vedla k tzv. konstruktivní matematické analýze, která se nyní zpracovává. Existují náznaky způsobů konstruktivizace i jiných matematických teorií. Jednou ze základních metod, jichž používáme při konstruktivizaci, je přechod od zkoumaných předmětů k jejich názvům, které jsou vždy konstruktivními objekty.

PROBLÉMY ROZHODNUTÍ

Dílčím příkladem hromadných problémů jsou problémy rozhodnutí. Problém rozhodnutí nějaké množiny je problém sestrojení algoritmu, který by tuto množinu

rozhodl. Odpovídající série jedinečných problémů se zde skládá z problémů odpovědět na otázku příslušnosti k množině, přičemž tato otázka se klade pro každý objekt z nadřazeného souboru konstruktivních objektů. A naopak: každý hromadný problém odpovídající sérii jedinečných problémů odpovědi na jistou otázku může být chápán jako problém rozhodnutí určité množiny, a to množiny těch jedinečných problémů, na něž odpověď zní ano. Z toho je zřejmá důležitost problémů rozhodnutí. Právě tyto problémy byly zkoumány z hlediska jejich rozhodnutelnosti. Mezi problémy rozhodnutí mají zvláštní postavení problémy kladené pro třídy dokazatelných formulí určitého kalkulu. Problém rozhodnutí třídy všech dokazatelných formulí určitého kalkulu nazýváme také problémem rozhodnutí kalkulu sama. (V ruských textech se problém rozhodnutí nazývá obvykle problémem rozhodnutelnosti, je však lépe nazývat problémem rozhodnutelnosti problém odpovědět, zda daný problém rozhodnutí má řešení.)

NEROZHODNUTELNÉ HROMADNÉ PROBLÉMY

Problém rozhodnutí pro nějaký kalkul je vždy problémem rozhodnutí rekurzívně spočetné množiny. Vůbec o všech problémech rozhodnutí, jež vznikaly v matematice přirozeným způsobem, se ukázalo, že to jsou problémy rozhodnutí rekurzívně spočetných množin. Takovým je i výše připomenutý první příklad nerozhodnutelného problému rozhodnutí (a zároveň první příklad nerozhodnutelného hromadného problému vůbec), který uveřejnil CHURCH r. 1936. Takovým je i tzv. problém totožnosti pro asociativní systémy, jehož nerozhodnutelnost dokázali r. 1947 nezávisle na sobě A. A. MARKOV a americký vědec E. L. POST; tento výsledek je zajímavý jako první důkaz nerozhodnutelnosti hromadného problému vzniklého (již r. 1914) mimo logiku a teorii algoritmů. Takovým byl i proslulý problém totožnosti pro grupy, formulovaný již r. 1912, jehož nerozhodnutelnost dokázal r. 1952 sovětský vědec P. S. NOVIKOV (Leninova cena 1957). Každý z problémů totožnosti záleží ve vyhledání algoritmu, který by zjistil ekvivalenci nebo neekvivalenci libovolných dvou slov v dané abecedě, (na tom, jak je definována ekvivalence, závisí, jde-li o asociativní systém nebo o grupu). Proto lze problém totožnosti chápat jako problém rozhodnutí množiny všech dvojic vzájemně ekvivalentních slov (vzhledem k souboru všech možných dvojic slov). Protože však lze stanovit vytvářející proces, který dovoluje získat dvojice vzájemně ekvivalentních slov, je množina všech takových dvojic rekurzívně spočetná.

REDUKOVATELNOST

Počínaje CHURCHOVÝM problémem byly až do r. 1944 všechny důkazy nerozhodnutelnosti hromadných problémů prováděny (nebo mohly být prováděny) touto jedinou metodou: Prokázaně nerozhodnutelný problém zkoumaný Churchem se redukoval na zkoumaný hromadný problém, takže kdyby zkoumaný hromadný problém byl

rozhodnutelný, muselo by totéž platit i o Churchově problému (v tom smyslu lze říci, že důkaz nerozhodnutelnosti zkoumaného problému se redukoval na důkaz nerozhodnutelnosti Churchova problému). Vznikla otázka, zda o každém nerozhodnutelném problému rozhodnutí platí, že jeho nerozhodnutelnost lze zjistit tímto způsobem. Tuto otázku, jež dostala název problému *redukovatelnosti*, formuloval POST r. 1942; zároveň Post uvedl několik příkladů nerozhodnutelných problémů rozhodnutí, jejichž nerozhodnutelnost zjistil metodou odlišnou od výše popsané (tyto příklady neřešily ještě problém redukovatelnosti, neboť zůstávala otevřenou otázka, zda i pro ně nelze najít takové důkazy nerozhodnutelnosti, jež by se redukovaly na důkaz nerozhodnutelnosti Churchova problému; nakonec byly pro některé ze zmíněných příkladů takové důkazy skutečně nalezeny). Problém redukovatelnosti byl předmětem soustředěných výzkumů o teorii algoritmů až do r. 1956, kdy jej nezávisle na sobě vyřešili sovětský vědec A. A. MUČNIK a americký vědec R. M. FRIDBERG. Byl sestrojen příklad nerozhodnutelného problému rozhodnutí (pro rekurzívně spočetné množiny), jehož nerozhodnutelnost nelze dokázat redukcí Churchova problému na tento problém. Mučnik ukázal ještě více: že totiž nejen Churchův problém, ale ani žádný jiný problém nemůže hrát roli ‚standardního nerozhodnutelného problému' v tom smyslu, že by důkaz nerozhodnutelnosti kteréhokoliv nerozhodnutelného problému rozhodnutí pro rekurzívně spočetnou množinu mohl být redukován na důkaz nerozhodnutelnosti tohoto standardního problému.

ZPŘESNĚNÍ POJMU ALGORITMU A PŘIDRUŽENÝCH POJMŮ

Dojít k závěrům o neexistenci algoritmu nelze bez dalšího zpřesnění a formalizace pojmu algoritmu. Zpřesnit tento pojem můžeme jen tak, že popíšeme určitou konkrétní třídu algoritmů, jež si bude činit nárok na to, že libovolný algoritmus může být nahrazen jemu ekvivalentním algoritmem z této třídy. Podobná zpřesnění se počala objevovat od r. 1936, kdy E. L. POST a anglický vědec A. M. TURING navrhli nezávisle na sobě shodné definice abstraktních počítacích strojů určených k provádění algoritmických procesů. (Je pozoruhodné, že tyto definice, jež se objevily před vytvořením matematických strojů, přejaly mnoho jejich podstatných rysů.) R. 1950 popsal A. A. MARKOV speciální třídu algoritmů (kterou nazval *normálními algoritmy*), které v průběhu svého rozvíjení postupně transformují slova; a na základě tohoto zpřesnění vypracoval poprvé systematickou teorii algoritmů. R. 1953 zavedl A. N. KOLMOGOROV do teorie algoritmů topologické komplexy jakožto prostředek pro zachycení etap rozvíjení algoritmu. Existují i jiná zpřesnění pojmu algoritmu.

Všechna tato zpřesnění vycházejí z těchto obecných představ (nebo na ně mohou být snadno redukována): Algoritmický proces se rozpadá na jednotlivé dostatečně elementární kroky. Každý krok záleží v tom, že jeden stav procesu je vystřídán druhým (výchozí údaj je pak počátečním stavem). Přechod od nějakého stavu k bezprostředně následujícímu se děje na základě tzv. pravidel bezprostřední transformace, o nichž předpokládáme, že jsou dostatečně elementární. Některé stavy uznáváme za

koncové (na základě dostatečně elementárních pravidel ukončení) a z nich odvozujeme konečný výsledek (také na základě dostatečně elementárních pravidel). Při aplikaci algoritmu na nějaký objekt jsou možné tři způsoby, jakými probíhá algoritmický proces: 1. Každý stav je vystřídáván dalším a proces se nikdy nezastavuje; 2. při určitém kroku vzniká stav, na který nelze aplikovat ani pravidla bezprostřední transformace ani pravidla ukončení, a dochází k bezvýslednému zastavení; 3. při určitém kroku vzniká stav, který uznáváme za koncový, a dochází k výslednému zastavení, které je spojeno se získáním konečného výsledku. Algoritmus lze tedy aplikovat jen na ty objekty, pro něž se algoritmický proces rozvíjí třetím způsobem.

Každé ze známých zpřesnění pojmu algoritmu záleží v podstatě v tom, že se fixuje určitý konkrétní druh pravidel bezprostřední transformace, pravidel ukončení a pravidel stanovení konečného výsledku. Přitom se nezbytně fixuje také soubor (soubory) konstruktivních objektů, na jehož (nebo na jejichž) objekty má smysl aplikovat uvedená pravidla. Pro každé zpřesnění se formuluje základní hypotéza záležející v tom, že k libovolnému algoritmu můžeme ukázat ekvivalentní algoritmus z třídy algoritmů popisovaných daným zpřesněním. (Ekvivalence znamená, zhruba řečeno, že oba algoritmy vedou k jedněm a týmž výsledkům; sám pojem ekvivalence vyžaduje však další zpřesnění, neboť původní algoritmus může být aplikovatelný na takové objekty, které vůbec nejsou možnými výchozími daty pro algoritmy popisované zkoumaným zpřesněním.) Tato základní hypotéza nemůže být předmětem matematického důkazu, protože součástí její formulace je pojem libovolného algoritmu. Má charakter přírodovědecké hypotézy připomínající např. fyzikální zákony. Pro každé z dosud navržených zpřesnění je příslušná hypotéza v dobrém souhlasu s praxí. Ve prospěch těchto hypotéz mluví i to, že všechna navržená zpřesnění se ukázala v určitém přirozeném smyslu vzájemně ekvivalentními (toto poslední tvrzení už podléhá důkazu a skutečně může být dokázáno).

Za první varianty zpřesnění pojmu efektivního vytvářejícího procesu je nutno pokládat logistické systémy s formálně popsanými pravidly vyvozování, jež se objevily již v 19. stol. (poprvé v pracích německého logika G. FREGE). Konečně při formalizaci stále širších a širších oblastí matematiky (hlavně v pracích anglických logiků A. N. WHITEHEADA a B. RUSSELLA) se objevily kalkuly dostatečně silné k tomu, aby bylo možno na základě v nich zadávaných vytvářejících procesů definovat (v dříve objasněném smyslu) libovolnou vyčíslitelnou funkci jako funkci definovanou pravidly vyvozování v některém kalkulu. POSTOVY a TURINGOVY práce umožnily definovat vyčíslitelnou funkci na podkladě jimi navržených zpřesnění pojmu algoritmu. (Definice vyčíslitelné funkce jsou ovšem možné i na podkladě jiných zpřesnění.) Zároveň KLEENE navrhl takovou definici vyčíslitelné funkce, která nezávisí na žádném zpřesnění pojmu algoritmu nebo vytvářejícího procesu. (Všechny tyto definice vyčíslitelné funkce se ukázaly vzájemně ekvivalentními.)

O pojem algoritmu se neopírající definice vyčíslitelné funkce je zajímavá jak z hlediska logického (neboť se ukazuje, že pojem vyčíslitelné funkce má svůj obsah nezávislý na pojmu algoritmu), tak i z hlediska matematického (neboť v celé řadě úloh

není zapotřebí konstruovat explicitně algoritmus, nýbrž stačí konstatovat vyčíslitelnost příslušné funkce). Tak základní pojmy, jako pojmy rekurzívně spočetné a rozhodnutelné množiny, mohou být definovány na základě pojmu vyčíslitelné funkce (aniž saháme k pojmu algoritmu). Rozumí se ovšem, že konec konců cena vyčíslitelných funkcí záleží právě v tom, že pro každou z nich lze ukázat algoritmus, který ji vyčísluje.

V současné době se vyčíslitelné funkce s přirozenými čísly jako argumenty a hodnotami ztotožňují s tzv. *částečně rekurzívními* funkcemi, jež lze exaktně matematicky definovat.

Literatura

ADJAN S. I.: Problema algoritma. Nauka i žizň 1957, No, 8, str. 13.

KOLMOGOROV A. N. i USPENSKIJ V. A.: K opreděleniju algorifma. Uspechi matěmatičeskich nauk *13*, 3 (1958).

MARKOV A. A.: Těorija algorifmov. Trudy Matěmatičeskogo instituta AN SSSR *38*, 176 (1951).

MARKOV A. A.: Těorija algorifmov, tamtéž *42* (1954).

PETER R.: *Rekursivnyje funkcii* (překl. z němčiny). Moskva 1954 (§ 20, 21).

TRACHTĚNBROT B. A.: Algorifmy i mašinnoje rešenije zadač. Moskva 1957, 2. vyd. 1960. (Český překlad NČSAV, 1963.)

ŠANIN N. A.: O některych logičeskich problemach arifmetiki. Trudy Mat. instituta AN SSSR *43* (1955) (úvod, § 1, § 4, § 5—6).

Přeložil *Pavel Materna*

RADIAČNÍ PORUCHY V PEVNÝCH LÁTKÁCH

MILOŠ MATYÁŠ, Praha

Již v roce 1942 vyslovil WIGNER doměnku, že záření vznikající v reaktorech nebo v urychlovačích částic může porušit krystalickou mřížku použitých materiálů. Odůvodňoval to tím, že toto záření má vysokou energii, ať již jde o částice nebo o elektromagnetické rozruchy. Dnes označujeme mřížkové poruchy, které vznikají v krystalu účinkem energeticky bohatého záření, jako radiační poruchy. Pod pojmem energeticky bohaté záření rozumíme takové záření, jehož jednotlivé částice nebo kvanta mají energii minimálně 10^4 eV = 1,6 . 10^{-19} Ws = 1,6 . 10^{-12} erg. Přivedeme-li tuto energii tělesu makroskopických rozměrů, např. 1cm^3 vody, nestane se nic; změna jeho teploty bude prakticky nezměřitelná. Avšak zcela jiná situace nastane při elementárním procesu, kdy např. neutron dopadající s touto energií na krystal, přijde do styku s jednotlivými atomy. V tomto případě lze očekávat, že mohou nastat změny v mřížce krystalu. Tento předpoklad se ukáže ještě pravděpodobnějším, srovnáme-li energii dopadající částice s vazebnou energií krystalu, která přepočtena na 1 atom činí několik eV.