B. A. Trachtěnbrot

Algoritmy a počítací stroje

Pokroky matematiky, fyziky a astronomie, Vol. 2 (1957), No. 3, 285--291

Persistent URL: `http://dml.cz/dmlcz/137224`

ALGORITMY A POČÍTACÍ STROJE*)

§ 1.

V posledních letech vzbudily značnou pozornost široké veřejnosti zprávy o postavení velkých, vysoce výkonných samočinných počítacích strojů, které byly zkonstruovány na příklad v USA, v Anglii, v SSSR a v poslední době též u nás v ČSR. Tyto samočinné počítače slouží v prvé řadě k řešení složitých matematických problémů, v nichž je třeba provádět nezvykle velký počet matematických operací (na příklad řešení velkých systémů algebraických nebo diferenciálních rovnic). Samočinné počítače mají na druhé straně velký význam ve výrobě, kde se podílejí rozhodujícím způsobem na řešení problémů regulace a automatisace, a tak přispívají k neustálému zdokonalování výrobních technologických procesů a k všeobecnému rozvoji techniky. Pomocí samočinných počítačů lze na příklad provést úplnou automatisaci některých výrobních procesů, jako je obrábění výrobně složitých strojních součástí. Je dále jasné, jaký význam mají tyto samočinné počítače ve vojenství, kde jich lze na př. užít při samočinném řízení letu raketové střely nebo při řízení palby protileteckých baterií a pod. Neméně významné je nakonec užití těchto strojů k řešení úloh povahy logické, jako je na příklad samočinné překládání literárních textů z jednoho jazyka do druhého.

Samočinné počítače tedy mají velký vědecký a hospodářský význam. Jednak nesmírně usnadňují nebo dokonce úplně nahrazují práci mnoha kvalifikovaných pracovníků, na druhé straně dávají nové perspektivy mnoha vědním oborům, kde se mohou díky obrovské rychlosti a samočinnému řízení práce počítacích strojů řešit problémy dříve prakticky neřešitelné.

Moderní samočinný počítací stroj můžeme definovat jako stroj, který provádí samočinně velké množství aritmetických operací, samočinně manipuluje s čísly, nad nimiž má operovat, a samočinně upravuje svůj pracovní postup.

O samočinných počítačích existuje pochopitelně jednak celá rozsáhlá odborná literatura (u nás na př. Sborníky o strojích na zpracování informací, vydávané Ústavem matematických strojů ČSAV), jednak informativní publikace nebo práce uveřejňované v různých časopisech, na příklad též v tomto časopise, v jehož 3. čísle roč. 1956 byl uveřejněn referát o článku F. V. Majorova „Elektronické počítací stroje“.

V tomto článku bude osvětlen jeden speciální aspekt samočinných počítačů, a to jejich souvislost s algoritmy. Tato souvislost se tu objevuje proto, že samočinné počítače pracují při řešení určité úlohy podle určitého předpisu, zvaného pracovní program stroje. V článku bude uvedená souvislost objasněna zejména na příkladu samočinného počítacího stroje, navrženého již v r. 1937 matematikem A. M. Turingem. Tento stroj byl zvolen proto, že jeho princip je poměrně nejjednodušší, a dále proto, že všechny moderní elektronické samočinné počítače mají zásadně stejnou strukturu jako tento stroj.

Povězme si něco o algoritmu. Pod tímto pojmem rozumíme všeobecně souhrn formálních pravidel (předpisů), které umožňují úplně mechanicky řešit všechny úlohy určitého druhu.

Jinak lze též říci, že algoritmus je přesný předpis pro početní proces, který vede od výchozích údajů úlohy k hledanému výsledku.

Je zřejmé, že zde nejde o přesné matematické definice algoritmu, nýbrž spíše o výklad

---

*) Referát z článku B. A. Trachtěnbrot, *Algoritmy i mašinnoje rešenije zadač,* Mat. v škole, 1956, č. 4 a 5.

tohoto pojmu. Matematicky přesně byl tento pojem definován dosti složitým způsobem teprve v třicátých letech tohoto století několika autory současně (S. C. Kleene, A. Church, E. L. Post a další). Jejich definice algoritmu se sice navzájem formálně liší, jsou však ve své podstatě ekvivalentní.

Typickým příkladem algoritmu je známý *algoritmus Euklidův, kterým se hledá největší společný dělitel dvou libovolných daných přirozených čísel.* Podle „definice" uvedené na druhém místě hraje zde roli výchozích údajů libovolná dvojice přirozených čísel. Roli předpisu zde hraje známý předpis (resp. řada pokynů) pro vytvoření klesající posloupnosti čísel, kde první je větší z obou daných čísel, druhé je menší z nich; třetí dostaneme jako zbytek po dělení prvého čísla druhým; čtvrté je zbytek po dělení druhého čísla třetím, a tak dále, až dostaneme dělení beze zbytku. Dělitel v tomto posledním dělení je pak největším společným dělitelem daných dvou čísel čili hledaným výsledkem úlohy.

Popišme si nyní podrobněji zmíněný předpis k řešení Euklidova algoritmu, a to v té úpravě, kdy jsme převedli dělení na opakované odčítání.

Pokyn 1. Zapamatuj si dvě čísla a, b v tomto pořadí. Přejdi k dalšímu pokynu.

Pokyn 2. Srovnej tato čísla podle velikosti ($a = b$, $a < b$ nebo $a > b$?). Přejdi k dalšímu pokynu.

Pokyn 3. Jsou-li obě čísla stejná, pak každé z nich je hledaným výsledkem. Zastav výpočet. Nejsou-li stejná, přejdi k dalšímu pokynu.

Pokyn 4. Je-li prvé číslo menší než druhé, zaměň jejich pořadí, a zapamatuj si je. Přejdi k dalšímu pokynu.

Pokyn 5. Odečti od prvého čísla číslo druhé a zapamatuj si tato dvě čísla: odečítané číslo a rozdíl v tomto pořadí. Přejdi k pokynu 2.

Po absolvování všech pěti pokynů je tedy nutno vrátit se k pokynům 2., 3., 4., 5. a pak opět k pokynům 2., 3., atd. a opakovat stále tento cyklus až do té doby, kdy bude splněna podmínka z pokynu 3. Proces se pak zastaví.

Již tento jednoduchý příklad ilustruje obě dvě základní vlastnosti, které má každý algoritmus mít:

1. Determinovanost algoritmu. Algoritmický předpis musí být úplně přesný tak, aby jej bylo možno sdělit i jiné osobě, která není předem seznámena s podstatou věci, která však musí podle tohoto předpisu umět úlohu zcela samostatně řešit. Proces řešení úlohy je tedy algoritmem naprosto přesně determinován.

2. Hromadnost algoritmu. Algoritmickým předpisem se řeší nejen jedna zvláštní úloha, nýbrž celá třída úloh určitého typu.

Ukazuje se, že algoritmicky můžeme řešit i složité matematické úlohy, vyžadující na příklad těchto výkonů:

1. Sestavení matematických tabulek.
2. Numerického derivování až do libovolně vysokého řádu.
3. Interpolace.
4. Numerické integrace.
5. Harmonické analysy.
6. Řešení soustav rovnic (algebraických i diferenciálních).
7. Sčítání a násobení řad.
8. Operací s maticemi a výpočtu determinantů.
9. Postupných aproximací.

Jak známe z numerické analysy, tento aparát je totiž možno nahradit v podstatě čtyřmi

STAV PŘED SPUŠTĚNÍM STROJE:

01	2	12	13	14	NEPODMÍNĚNÉ PŘÍKAZY K ARITMET. OPERACÍM
02	2	12	13	08	
03	0	00	00	00	PODMÍNĚNÉ PŘÍKAZY
04	1	12	03	14	
05	1	01	09	01	NEPODMÍNĚNÉ PŘÍKAZY K PŘEADRESOVÁNÍ
06	1	02	10	02	
07	1	04	11	04	
08	0	00	00	01	KONTROLNÍ ČÍSLO
09	0	01	01	01	KONSTANTY
10	0	01	01	00	
11	0	01	00	01	
12	0	00	00	36	
13	0	00	00	16	
14					
15					

a)

STAV PO PRVNÍM CYKLU:

2	13	14	15	01
2	13	14	15	02
0	00	00	00	03
1	13	03	14	04
1	01	09	01	05
1	02	10	02	06
1	04	11	04	07
0	00	00	20	08
0	01	01	01	09
0	01	01	00	10
0	01	00	01	11
0	00	00	36	12
0	00	00	16	13
0	00	00	20	14
				15

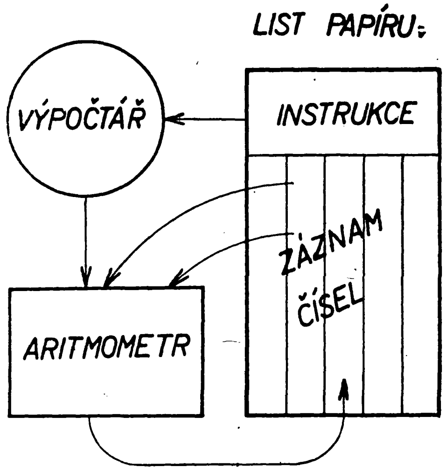
b)

Obr. 1

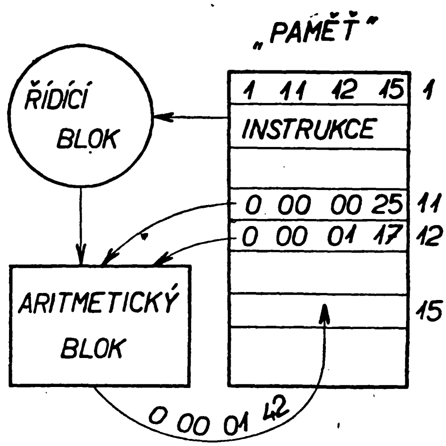
základními početními operacemi, které je možno nakonec převést na sčítání a odčítání, a které jsou proto vhodné pro algoritmické zpracování.

§ 2.

Připomeňme si nyní stručně strukturu běžného elektronického samočinného počitače. Samočinný počítací stroj musí mít takové orgány, aby mohl vykonávat ty úkoly, které



a)
Obr. 2a



b)
Obr. 2b

při řešení dané úlohy bez použití samočinného počitače musí provádět člověk-výpočtář. Kromě orgánů, které umožňují vstup daných informací do stroje a dále výstup výsledků ze stroje, a kromě dalších orgánů, které mají spíše technický význam, obsahuje každý samočinný počitač tyto tři orgány základního významu:

1. Paměť. 2. Aritmetický blok. 3. Řídící blok.

Do paměti jsou ukládány v určitém kodu všechny počáteční údaje dané úlohy, a dále všechny potřebné instrukce. Současně se zde přechodně uchovávají i dílčí výsledky vznikající během práce stroje.

PROVÉST OPERACI	S ČÍSLY V BUŇKÁCH	A VÝSLEDEK PŘENÉST DO BUŇKY
α	β γ	δ

Obr. 3

V aritmetickém bloku se provádějí všechny potřebné aritmetické operace vedoucí k hledanému výsledku, a to podle instrukcí vložených do stroje na začátku procesu i podle instrukcí vytvořených ve stroji během procesu.

Úkolem řídícího bloku je zajišťovat správné řízení (resp. řazení) jednotlivých operací.

Paměť sestává z buněk, které jsou očíslovány přirozenými čísly 1, 2, 3, . . . , tak zvanými adresami buněk. V každé buňce může být uložen jen jeden zakódovaný údaj. Na *obr. 1a)* je schematicky naznačena paměť stroje, složená z 15 buněk. Některé z nich jsou obsazeny čísly. Čísla v prvních 11 buňkách jsou čísly zakodované instrukce, vložené do stroje před jeho spuštěním. V buňkách 12 a 13 jsou uložena výchozí data řešené úlohy. Během práce stroje změní některé obsazené buňky svůj obsah, jiné neobsazené buňky mohou sloužit během výpočtu k uložení nových údajů vznikajících při chodu stroje.

Na *obr. 1b)* je schematicky zobrazen stav paměti po několika prvních taktech práce stroje. V závěrečném taktu práce stroje se vyprázdní všechny buňky paměti kromě té, do které je uložen hledaný výsledek.

V aritmetickém bloku se provádí vlastní výpočet, a to postupnou transformací vstupních signálů zobrazujících počáteční údaje. Tyto signály se dopravují do aritmetického bloku z paměti, kde jsou uloženy. Signál, který je výsledkem jisté operace a který zobrazuje určitý dílčí výsledek, se přenese zpět do určité buňky paměti, kde se přechodně ukládá. Schematicky je tento postup zobrazen na *obr. 2b)*, kde je současně ilustrován postup při sčítání čísel z buněk 11 a 12, jehož výsledek se má přenést do buňky 15.

Celou operaci řídí řídící blok, který vysílá podle instrukce příkazy k spolupráci těch uzlů stroje, které mají provést žádanou operaci.

Popišme si nyní podrobněji postup, kterým se provádí určitá operace. Je třeba si nejprve říci, že instrukce vkládaná do paměti stroje sestává z příkazů a z pomocných čísel. Při tom se nejvíce užívá t. zv. tříadresových příkazů, které mají obvykle tvar podle *obr. 3.* Příkaz se koduje ve formě určité konečné posloupnosti číslic α, β, γ, δ, kde první číslice představuje pořadové číslo operace a další tři číslice značí adresy určitých buněk paměti. Tak na příklad v *obr. 2b)* je v buňce 1 zapsána posloupnost číslic 1 11 12 15, která značí tento příkaz: Proveď operaci č. 1 (t. j. sčítání) s čísly uloženými v buňkách 11 a 12, a výsledek přenes do buňky 15. Na začátku každého pracovního taktu stroje je z jisté buňky paměti vyslán do řídícího bloku určitý tříadresový příkaz. Ihned po přenesení tohoto příkazu do řídícího bloku uskuteční se úplně automaticky příslušná spojení a provede se tak žádaná dílčí operace. Potom je vyslán do řídícího bloku další příkaz z paměti, stroj provede další operaci, což pokračuje až do okamžiku, kdy stroj již rozřešil celý daný úkol, takže se pak zastaví.

Podívejme se nyní na to, jak stroj skutečně pracuje při řešení určité úlohy, jak spolupracují jednotlivé jeho orgány při provádění určité instrukce, v jakém pořadí postupují jednotlivé příkazy instrukce do řídícího bloku a jak se reguluje řazení těchto příkazů.

K tomu, abychom se blíže seznámili s prací stroje při řešení určitého úkolu, popíšeme si zde podrobně postup při hledání největšího společného dělitele dvou daných přirozených čísel, to jest postup při provádění algoritmu Euklidova. Připomeňme si zde především znovu, že dělení nahrazujeme postupným odčítáním, Mějme tedy za úkol nalézti pro dvě daná přirozená čísla a, b jejich největšího společného dělitele (a, b). Předpokládejme $a > b$. Protože $(a, b) = (b, a - b)$, vidíme, že po prvním odečtení je úloha převedena na tutéž úlohu, kterou však máme provést s dvojicí a_1, b_1, při čemž za první číslo a_1 vezmeme větší z obou čísel b, $a - b$, a druhé je menší z nich. Zřejmě platí tyto nerovnosti: $a_1 < a$, $b_1 < b$. Po opětném odečtení dostaneme čísla a_2, b_2, a tak pokračujeme dále až do okamžiku, kdy dostaneme dvojici stejných čísel $a_i = b_i$, jejichž rozdíl je tedy nulový. Tím okamžikem se proces zastavuje a hledaný výsledek je $(a, b) = a_i$.

Chceme-li nyní realisovat právě popsaný postup v samočinném stroji, musíme celý tento postup zapsat ve tvaru instrukce pro stroj „srozumitelné". Tato instrukce sestává z jedenácti číslicových příkazů, které jsou uloženy v buňkách paměti s adresami 01 až 11 (viz *obr. 1*). Buňky 01 až 07 obsahují sedm příkazů zapsaných v tříadresovém kodu, ve kterém má sčítání znak 1, a odčítání znak 2 (je to vždy první cifra příkazu). Příkaz 0 · 00 · 00 · 00 při tom značí příkaz k zastavení stroje. V buňkách 08 až 11 jsou uloženy pomocné číslice. Při tom obsah buněk 09 až 11 se během práce stroje nemění, protože to jsou konstanty potřebné pro úspěšný průběh řešení úlohy.

Příkazy jsou nepodmíněné a podmíněné. Nepodmíněný příkaz vyslaný z paměti do řídícího bloku způsobí vždy okamžité spuštění stroje, takže příkaz je splněn bezpodmínečně. Přijde-li však z paměti do řídícího bloku podmíněný příkaz, začne zařízení pracovat jen tehdy, je-li splněna určitá vedlejší podmínka, na příklad podmínka, aby číslo umístěné v určité zvláštní (kontrolní buňce) mělo potřebné znaménko (plus nebo minus). V našem případě je příkaz z buňky 03 podmíněný, protože působí jen tehdy, jestliže v kontrolní buňce je číslo 0. Příkaz z buňky 04 působí jen tehdy, je-li v kontrolní buňce číslo záporné.

Dále rozlišujeme příkazy podle jejich všeobecného určení. V buňkách 01, 02 a 04 jsou uloženy příkazy k provedení aritmetických operací s čísly uloženými v jiných buňkách (tyto příkazy zajišťují dílčí výsledky, které vycházejí během chodu stroje). V buňkách 05 až 07 jsou uloženy příkazy, které mění příkazy z buněk 01, 02 a 04 tak, že ponechávají nezměněnu vždy jejich první cifru (t. j. pořadové číslo aritmetické operace), avšak mění další cifry, takže tím provedou přeadresování původního příkazu. Tak na příklad po provedení příkazu z buňky 05 se v buňce 01 objeví příkaz 2, 13, 14, 15, který nyní předpisuje odečtení čísel z buněk 13 a 14 a přenos výsledku do buňky 15.

Příkazy přicházejí do řídícího bloku v cyklickém pořadí. To znamená, že nejprve přijde příkaz z buňky 01, potom z buňky 02, a tak dále, až z buňky 08, načež přichází opět příkazy z buňky 01, 02, atd., čili celý cyklus se opakuje až do té doby, kdy přijde příkaz k zastavení stroje. Zapíšeme-li do buněk 12 a 13 daná dvě přirozená čísla a, b, objeví se po prvním cyklu práce stroje v buňce 14 číslo $a - b$ v případě $a > b$, nebo číslo a, je-li $a < b$. V druhém cyklu se podrobí obdobné operaci již čísla z buněk 13 a 14, a výsledek se zapíše do buňky 15, a tak dále. Tak dostáváme ve všech buňkách počínaje buňkou 13 řadu čísel $a_1, a_2, \ldots, a_\nu$, kde každé číslo a_ν je rovno buďto rozdílu $a_{\nu-1} - a_{\nu-2}$ (pro $a_{\nu-1} > a_{\nu-2}$) nebo číslu $a_{\nu-2}$ (pro $a_{\nu-1} < a_{\nu-2}$). Proces vytváření dalších čísel se zastaví tím okamžikem, kdy se objeví číslo $a_k = 0$. Hledané řešení je pak rovno číslu a_{k-1} $(= a_{k-2})$.

Na *obr. 1b)* je znázorněna paměť stroje po proběhnutí prvého cyklu, při čemž na počátku práce byla do počitače vložena čísla $a = 36$, $b = 16$.

Poznamenejme, že počáteční instrukci pro práci stroje bychom snadno mohli doplnit požadavkem, aby výsledek byl zaregistrován v jisté předem určené buňce.

Z uvedeného výkladu již plynou tyto dva obecné principy, podle nichž pracuje samočinný počitač:

1. Počitač provádí jednotlivé příkazy instrukce v tom pořadí, v jakém byly zapsány do paměti. Stroj je však schopen samočinně měnit průběh početního procesu podle dílčích výsledků, které se objevují během práce stroje. To je umožněno právě zavedením podmíněných příkazů.

2. Počítač je schopen provádět libovolně dlouhé řetězce výpočtů, a to vzhledem k tomu, že může sám přetvářet a mnohokrát opakovat tutéž instrukci nebo některé její části.

§ 4.

Z předcházejícího výkladu již vyplývá hluboká souvislost mezi algoritmy a samočinnými počítacími stroji. Tato souvislost záleží v tom, že každý proces, který se postupně realisuje v samočinném počítači, může být popsán určitým algoritmem. Na druhé straně však můžeme aspoň theoreticky každý algoritmus realisovat v určitém vhodném počítacím stroji. Jediné praktické omezení možnosti realisace jakéhokoli algoritmu v samočinném počítači je dáno tím, že skutečný stroj má ohraničenou paměť, to znamená, že si nemůže pamatovat ani pracovat s libovolně velkými čísly. Další překážkou může být i neobyčejně velký počet operací, jejichž provedení by trvalo neúnosně dlouho.

Dříve se vystačilo s nepřesnou definicí pojmu „algoritmus", protože se tohoto termínu užívalo jen v souvislosti s konstrukcí určitých konkretních algoritmů, kterými se řešily jen ojedinělé úlohy. Později se však stále naléhavěji projevovala snaha matematiků vytvářet takové algoritmy, které by řešily stále širší třídy úloh. Tak vznikala potřeba matematicky přesné definice algoritmu, která byla, jak už bylo řečeno, skutečně podána v třicátých letech tohoto století současně v pracích několika matematiků.

V otázce souvislosti algoritmu s počítacím strojem hraje jistou úlohu tak zvaný **rozhodovací problém**. Jak známo, v moderní matematice se skoro všechny matematické theorie budují na axiomatickém základě. Axiomatická methoda v matematice záleží v tom, že všechny věty určité matematické theorie lze odvodit z několika základních tvrzení (axiomů), která se v dané theorii přijímají bez důkazu. V souvislosti s tím umožňuje matematická logika vytvoření zvláštní „řeči" logických formulí, v níž je možno vyjádřit libovolné tvrzení dané matematické theorie ve tvaru jisté úplně určené formule. Vzniká tedy nyní tato hromadná úloha: Nechť $A_1, A_2, \ldots, A_k$ jsou určité formule, které zobrazují nějaké axiomy, a budiž L libovolná formule zobrazující nějaké tvrzení. Jde o to, rozhodnout, zda tvrzení L lze či nelze formálně logicky odvodit z axiomů $A_1, A_2, \ldots,$ $\ldots, A_k$. Ukazuje se, že kdyby bylo možno vytvořit algoritmus k řešení libovolné speciální úlohy tohoto druhu, bylo by pak též možno vytvořit obecnou methodu k automatickému řešení nejrůznějších úloh ze všech oblastí matematiky. V souvislosti s tím by bylo možno vytvořit i universální počítací stroj. Po mnoha bezvýsledných pokusech se však ukázalo, že takovýto universální algoritmus (a tedy ani universální počítací stroj) k řešení **všech možných** matematických úloh sestrojit nelze.

S hlediska strojové matematiky má zvláštní význam ta definice algoritmu, která je založena na ztotožnění procesu aplikace algoritmu a procesu, který se odehrává v samočinném počítači při řešení úlohy. V tomto případě je nutno zachytit celý proces práce stroje pomocí určitého standardního schematu, pokud možno jednoduchého, avšak natolik přesného, aby mohl být předmětem matematického vyšetřování. Jako první provedl tento úkol matematik A. M. **Turing**, který navrhl nejobecnější a současně nejjednodušší koncepci počítacího stroje. Uveďme si v dalším výkladu principy stroje Turingova.

(Dokončení v příštím čísle)