Možný, Ivo

K problematice ekonomiky tvorby a užívání programů pro hromadné zpracování dat na samočinných počítačích

Sborník prací Filozofické fakulty brněnské univerzity. G, Řada sociálněvědná. 1973, vol. 22, iss. G17, pp. [123]-129

SBORNÍK PRACÍ FILOSOFICKÉ FAKULTY
BRNĚNSKÉ UNIVERSITY
STUDIA MINORA FACULTATIS PHILOSOPHICAE
UNIVERSITATIS BRUNENSIS
G 17, 1973

SDĚLENÍ, ZPRÁVY A RECENZE

K PROBLEMATICE EKONOMIKY TVORBY A UŽÍVÁNÍ PROGRAMŮ PRO HROMADNÉ ZPRACOVÁNÍ DAT NA SAMOČINNÝCH POČÍTAČÍCH

Ekonomický aspekt programů pro hromadné zpracování dat není předmětem *explicitního odborného zájmu* ani u zadavatelů, ani u programátorů. Přesto jak zadavatele, tak programátory tato stránka věci nutně zajímá, protože právě tu narážejí na limity, které jako nevyslovený předpoklad často rozhodujícím způsobem ovlivňují jejich práci: ekonomika určitého programu je významným kritériem, podle kterého posuzujeme jeho hodnotu. Zlepšování každého otevřeného systému programů, který je pravidelně užíván a podle zkušeností s jeho užíváním propracováván, ubírá se obvykle dvěma směry: na jedné straně je to úsilí o rozšíření jeho možností, na druhé straně úsilí o zekonomičtění jeho provozu.

Ekonomiku každého programového systému posuzujeme ze tří aspektů:

1. *ekonomika zadávání výpočtů* — pracnost formulace požadovaných úkonů, náklady na převod z jazyka uživatele do jazyka stroje;

2. *ekonomika provádění výpočtů* — rychlost zpracování dat, spotřeba strojového času;

3. *ekonomika čtení výsledků* — pracnost interpretace získaných výsledků, náklady na jejich převod z jazyka počítače do jazyka uživatele.

Kdybychom nebrali v úvahu praktické aspekty, čistě abstraktně jde jen o dvě dimenze: v systému člověk—počítač stojí na jedné straně nároky na práci stroje a na druhé straně nároky na práci člověka, který se strojem pracuje. Protože však v praxi ani zdaleka nejsou vždycky programátor a uživatel jeden a tentýž člověk a protože právě aspekty, které rozhodují o ekonomice práce uživatele a práce programátora nebývají shodné, musíme uvažovat o variabilitě tří, nikoli dvou dimenzí.

Tyto tři dimenze ovšem spolu vzájemně souvisejí. Za předpokladu, že eliminujeme všechny zbytečné kroky a získáme technicky ideální program, ocitáme se v situaci, kdy nemůžeme dále ubírat v jedné dimenzi, aniž bychom přidávali v druhé. Zmíněné tři dimenze vytvářejí víceméně uzavřený systém, kde zisk v jedné dimenzi zvyšuje náklady ve zbývajících dimenzích. Rozvaha o vhodném postupu se pak stává úvahou typu *cost—benefit*: musím posoudit, zda to, co získávám, mi vynahradí to, co jsem za to musel obětovat. Obecně tu platí jen jedno pravidlo: čím universálnější program, tím náročnější na počítačový čas. Toto pravidlo nás však nemůže vést ani k tomu, abychom dávali přednost speciálním programům, ani k tomu, abychom je odvrhli. Ekonomiku právě nelze zredukovat na otázku rychlosti zpracování dat v počítači — je určena celým složitým komplexem konkrétních podmínek, jež se mění případ od případu.

Aby mé poznámky o rozhodováních, ke kterým dochází při tvorbě programu a budování programových systémů pro zpracování hromadných dat byly maximálně konkrétní, pokusím se je demonstrovat na určitém systému programů, a to na systému pro

zpracování sociologických dat v LPS Brno na těch programech, jejichž jádrem je tvorba kontingenčních tabulek.

První rozhodnutí, které tu muselo být učiněno, je už samo vybudování takového systému. Určité konkrétní požadavky na třídění je možno splnit také tak, že žádný programový systém není budován a požadovaná kontingenční tabulka je zadána počítači přímo v jeho základním jazyce. Je to postup velmi neekonomický v nákladech na *čas programátora* — ale velmi ekonomický na *výpočetní čas* a velmi ekonomický na *čtení výsledků*, protože tabulka může potom mít úplný popis. Vyplatí se jen při extrémně velkém souboru dat a při velkém okruhu uživatelů výsledků. Pracuje se tak např. při zpracování výsledků sčítání lidu.

Pokud se tedy rozhodneme neprogramovat každou tabulku zvlášť, začínáme budovat systém programů více či méně universálních a přitom usilujeme o dosažení co nejlepších parametrů ve všech třech aspektech ekonomiky jeho provozu: ekonomiky zadávání programů, ekonomiky provádění výpočtů a ekonomiky čtení výsledků.

Ekonomika zadávání programů

Pochopitelně, že usilujeme o co nejjednodušší způsob zadávání, takový, aby s vynaložením co nejmenší námahy, prostřednictvím co nejmenšího počtu grafických symbolů (čím více symbolů, tím spíše bude některý z nich napsán či vyděrován chybně), bylo možno zadat co největší počet tabulek. Musíme se však smířit s tím, že platí: *čím jednodušší zadání, tím menší variabilita zadání je možná.*

Největší možnosti variability poskytuje v demonstrovaném programovém systému program MF2B. Umožňuje do každého řádku a každého sloupce zadávané tabulky vytřídit dokumenty, charakterizované jakoukoli kombinací všech existujících variant všech existujících znaků v logických vztazích konjunkce, disjunkce, negace, ekvivalence a nerovnosti. Zadání čtyřpolní tabulky pak vypadá například takto:

$$5=1,\ 7=1+2/\ 9\mathrm{M}1.9\mathrm{V}1,\ 12=3.14\mathrm{N}0/\ 31\mathrm{V}0/ \qquad (1)$$

Toto zadání čte počítač jako pokyn k vytřídění tabulky, kde jsou v prvním *sloupci* všechny dokumenty, které nabývají ve znaku „5" hodnotu 1, ve druhém sloupci všechny dokumenty, které nabývají ve znaku „7" hodnotu 1 nebo hodnotu 2. V prvním *řádku* pak jsou všechny dokumenty, které mají ve znaku „9" hodnotu větší než 1 a přitom menší než 6, ve druhém řádku všechny dokumenty, které ve znaku „12" mají hodnotu 3 a přitom nemají ve znaku „14" hodnotu 0. Takulka se třídí pro všechny dokumenty, které mají ve znaku „31" hodnotu větší než 0.

Je zjevné, že tento způsob zadávání je sice ideálně variabilní, ale pro vytřídění pouhých čtyř polí jedné tabulky jsme museli použít 30 grafických symbolů, na jedno pole tedy potřebujeme 7,50 symbolu. I pro to nejjednodušší zadání, když chceme například tabulku

$$5=1,2,3,4,5,6\ /\ 9=1,2,3,4\ /\ 0=0\ / \qquad (2)$$

tedy třídění znaku „5" v rozsahu hodnot od 1 do 6 se znakem „9" v rozsahu hodnot od 1 do 4 pro celý soubor, tedy všechny, kteří ve znaku „0" mají hodnotu 0, potřebujeme pro vytřídění tabulky o 24 polích použít k zápisu zadání 28 grafických symbolů, tedy 1,17 symbolu na pole.

Program je neekonomický také z hlediska ekonomiky výpočtů — vytváří každou tabulku zvláštním průchodem magnetické pásky s uloženým souborem dokumentů — v demonstrovaném systému vytváří tabulku podle zadání (2) ze souboru 1000 dokumentů o 100 znacích asi za 3 minuty strojového času. Chceme-li vytřídit dvě tabulky, první podle zadání (2) a druhou s týmiž sloupci, jen se změnou v řádcích, např.

$$\begin{aligned} &5=1,2,3,4,5,6\ /\ 9=1,2,3,4\ /\ 0=0\ / \\ &5=1,2,3,4,5,6\ /\ 10=1,2,3,4,5\ /\ 0=0\ / \end{aligned} \qquad (3)$$

pak nejen že u druhé tabulky opisujeme znovu zadání sloupců, ačkoli je stejné, ale počítač při vytváření druhé tabulky prochází celý soubor dokumentů znovu, a vytří-

dění dvou tabulek trvá tedy asi 6 minut; každá nová tabulka nás stojí znovu tentýž čas na celý průchod. Jak je vidět, platíme za ideální variabilitu dosti draze.

Většinou však tak velkou variabilitu při zadávání ani zdaleka nepotřebujeme. Zejména pokud máme možnost transformovat si znaky slučováním variant, jejich přesunováním a vytvářením nových znaků s variantami danými kombinacemi za použití logických operací, upravíme si znaky předem tak, že nám potom postačuje zadávat třídění tabulek zápisem, který jen udává, který znak třídit s kterým a jaké rozsahy variant u tříděných znaků brát v úvahu. V demonstrovaném systému takto pracuje program MB1. Zadání tabulky (2) tímto, co do variability třídění omezeným, ale zato rychlejším programem, pak vypadá následovně:

$$5=1:6\ /\ 9=1:4\ //\qquad (4)$$

tedy tatáž tabulka o 24 polích, ale k jejímu zadání potřebujeme nikoli 28, ale už jen 13 grafických znaků, tedy na jedno pole potřebujeme pouze 0,54 grafického znaku.

Chceme-li vytřídit dvě tabulky s týmž znakem ve sloupcích a rozdílným znakem v řádcích jako v příkladě (3), neopisujeme už opakující se zadání sloupců, ale píšeme jen

$$5=1:6\ /\ 9=1:4,\ 10=1:5\ //\qquad (5)$$

Základní ekonomičnost zadání programu MB1 se však projeví při zadávání velkého počtu tabulek. Nezadáváme tu totiž každou tabulku zvlášť, ale *maticovým zápisem*:

$$5=1:6,\ 6=1:4,\ 7=1:2,\ 8=0:3,\ 9=1:4,\ 10=1:4\ /$$
$$9=1:4,\ 10=1:5,\ 11=0:5,\ 12=1:2\ //\qquad (6)$$

Na tento zápis potřebujeme sice 61 grafických znaků, ale zadáváme jím vytvoření 24 tabulek celkem o 376 polích, tedy na jedno pole potřebujeme už jen 0,16 grafického znaku.

Třídí totiž všechny znaky z matice

$$|\ a_{i,j}\ |\qquad (7)$$

kde i = *znak 5,6,7,8,9,10*

j = znak 9,10,11,12

a ekonomika zadání roste se čtvercem počtu vzájemně tříděných znaků.

Ještě výrazněji se zvyšuje ekonomika zadávání u třídění třetího stupně. Chceme-li tedy třídění výše zadaných tabulek nikoli pro celý soubor, ale např. pro podsoubory podle deseti variant znaku „17", zápis zadání se rozšíří jen nepatrně:

$$5=1:6,\ 6=1:4,\ 7=1:2,\ 8=0:3,\ 9=1:4,\ 10=1:4\ /$$
$$9=1:4,\ 10=1:5,\ 11=0:5,\ 12=1:2\qquad (8)$$
$$17=1:10\ /$$

ale počítač na tento pokyn vytřídí ne už 24, ale 240 tabulek o celkem 3760 polích. Na zadání jednoho pole potřebujeme pak už jen 0,017 grafického znaku.

Zlepšuje se však nejenom ekonomika zadávání, ale i ekonomika výpočtu: Program MB1 při maticovém zadávání třídí současně celou matici kontingenčních tabulek, nikoli jednu po druhé. Spotřeba strojového času na vytřídění jedné tabulky se u stejného souboru nepohybuje okolo tří minut, jako u programu MF2B, ale okolo pouhých 10 vteřin. Maticové zadání umožňuje také výpočet a tisk matic koeficientů, což značně přispívá k ekonomice čtení výsledků, jak se o tom zmíníme dále.

Obětovali jsme tedy něco na variabilitě, ale získali nesmírně mnoho na ekonomice zadávání, které je v průměru řádově stokrát rychlejší, a na ekonomice výpočtu, který je rovněž v průměru téměř stokrát rychlejší. Nedosáhli jsme však pořád ani zdaleka mezí zestručnění. Ekonomičnost zadávání je možno ještě zvýšit, a už jen s malou obětí variability. Nástrojem dalšího zestručnění jsou dva triky: jednak oddělené zadání rozsahu používaných variant zpracovávaných znaků a jednak tzv. *se-*

kvenční zápis matic. V demonstrovaném systému realizoval tyto myšlenky program MB2.

Matice z příkladu (6) se zde zadává:

$$5—10\ /\ 9—12\ //\qquad (9)$$

a dostáváme se ke krajně jednoduchému zadání, kde na zadání jednoho pole u zvoleného příkladu je zapotřebí použít už jen 0,023 znaku. U třetího stupně pak vypadá zadání matice (7) takto:

$$5—10\ /\ 9—12\ /\ 17=1:10\ /\qquad (10)$$

a zadáváme jedno pole matice 0,004 znaku, tedy řádově tisíckrát hospodárněji, než u programu MF2B. Tím, že rozsah variant se zadává zvlášť, ztrácíme opět něco na variabilitě. Tam ale, kde zpracováváme určitý výzkum větší sérií výpočtů (tj. postupujeme tak, že po analýze první sestavy zadáváme — už poučenější — sestavu druhou, po její analýze opět poučenější zadáváme sestavu třetí, pak čtvrtou atd.) nám zůstává zápis rozsahu variant zpracovávaných znaků zachován na děrné pásce, resp. v paměti počítače; zadávali jsme ho pro každý znak jen jednou, pro první sestavu. V dalších sestavách, jakmile se v zadání příslušný znak vyskytne, počítač si ho sám vyhledá, bez ohledu na to, kolikrát ho ještě potřebuje.

Ekonomika výpočtů

Řekli jsme, že ekonomika zadávání a ekonomika výpočtů jsou na sobě závislé, úspora v jedné dimenzi se projevuje jako náklad v druhé. Neplatí to však vždy a neplatí to lineárně. Sekvenční zadávání se nijak významně neprojeví zvýšením nákladů na ekonomiku výpočtu. Maticové zadání už ano: na jedné straně šetří čas počítače tím, že umožňuje třídění celé matice tabulek při jednom průchodu pásky se souborem dokumentů, na druhé straně však počítač zbytečně třídí tabulky v diagonále matice a zrcadlové tabulky matice tam, kde se znaky v řádcích a sloupcích matice opakují. V matici o 10 znacích, kde řádky i sloupce tvoří tytéž znaky, by bylo smysluplných jen

$$\frac{10 \cdot 10 - 10}{2} = 45 \text{ tabulek ze } 100 \text{ vytříděných.} \qquad (11)$$

Třídit více než polovinu tabulek zbytečně je už podstatná ztráta počítačového času. Proto se vyplatí zavést do sady program, který nejprve vyloučí opakující se (zrcadlové) a diagonální tabulky. Zvlášť významné je to u třídění vyšších stupňů, kde počet tabulek velmi rychle roste, ale zároveň velmi rychle roste počet tabulek, které můžeme vyloučit, protože se opakují nebo nepřinášejí novou informaci, anebo nedávají smysl, protože jsou neúplné. Tak např. matice pouhých tří znaků o 10 variantách počítaná do pátého stupně třídění dává v pětidimenzionální matici

$$1—3\ /\ 1—3\ /\ 1—3\ /\ 1—3\ /\ 1—3\ /\qquad (12)$$

celkem 243099 tabulek, ale z toho je 243066 tabulek zrcadlových, diagonálních nebo kusých, které lze vyloučit, aby nezatěžovaly výpočet; smysluplných je jen 33 tabulek.

Rozhodujícím krokem ke zvýšení ekonomičnosti výpočtů je však oddělení třídících a tiskových programů. Samočinný počítač, který pracuje s daty uloženými ve vnější paměti na magnetické pásce, potřebuje relativně mnoho času na třídící proces — probrání souboru dokumentů je vázáno na mechanický pohyb mg pásky nebo disku vnější paměti a je tedy relativně pomalé. Transformace vytříděných dat z tabulky absolutních čísel do tabulek relativních čísel a výpočty koeficientů, průměrů, rozptylů a dalších statistických charakteristik probíhají už ve vnitřní paměti a jsou proti tomu velmi rychlé — počítač zde může využít plnou kapacitu svých 200 tisíc operací za vteřinu. Tisk je pak už zase mechanická záležitost, je tedy opět relativně pomalý. Vyplatí se proto nejprve třídícím programem vytvořit tabulky absolutních četností a uložit je ve vnější paměti na magnetické pásce, a teprve pak tiskovým programem

v dalším kroku z tabulek absolutních četností vypočítávat transformované tabulky a koeficienty. Počítač pak už neprobírá dokumenty, ale hodnoty polí vytříděných tabulek, kterých bývá řádově tisíckrát méně. Výpočet procent a statistických charakteristik je pak tak rychlý, že se zhruba rovná době tisku tabulky a tak zcela nebo téměř zcela zmizí v tiskovém čase.

Oddělení třídících programů od programů výpočetních a tiskových nám umožňuje:

1. Vypočítat a vytisknout jen matici koeficientů a teprve po její analýze jen ty tabulky z ní, které se jeví jako relevantní. Tímto způsobem šetříme u velkých matic tisk velkého počtu bezvýznamných tabulek, aniž bychom o jejich nevýznamnosti museli rozhodovat arbitrálně předem.

2. Ať už tiskneme všechny tabulky z matice anebo jen jejich výběr, máme možnost volit ještě u každé tabulky, které statistické charakteristiky a které transformace tabulek abs. četností do rel. četností budeme potřebovat a jen ty vypočítávat a tisknout.

Zejména možnost volby jen některých charakteristik podstatně zkracuje dobu tisku, ovlivňuje ekonomiku čtení výpočtů (viz dále) a umožňuje zahrnout do programu poměrně rozsáhlou sadu statistických ukazatelů. Při každém zadání pak vyžadujeme jen určitou, pro ten případ vhodnou jejich kombinaci. Nakolik to zvyšuje variabilitu programů si ukážeme nejlépe na programu HPCH, který v tiskovém programu umožňuje volbu z těchto charakteristik:

1. tabulka absolutních četností
2. tabulka relativních četností
3. tabulka relativních četností sloupců
4. tabulka relativních četností řádků
5. tabulka korelací alternativ
6. tabulka znaménkového testu
7. $chí^2$
8. hladina významnosti $chí^2$
9. Pearsonův koeficient pro pořadí
10. Čuprovův koeficient kontingence
11. Pearsonův koeficient kontingence
12. hladina významnosti pořadového koeficientu
13. průměry řádků a sloupců tabulky
14. mediány řádků a sloupců tabulky
15. tabulka součtů hodnot znaku, udaného jako „charakteristika"
16. tabulka relativních četností součtů charakteristiky
17. tabulka řádkových relativních četností součtů charakteristiky
18. tabulka sloupcových relativních četností součtů charakteristiky
19. tabulka absolutních průměrů charakteristiky
20. tabulka relativních četností průměrů charakteristiky
21. tabulka řádkových relativních četností průměrů charakteristiky
22. tabulka sloupcových relativních četností průměrů charakteristiky
23. tabulka absolutních rozptylů charakteristiky
24. tabulka relativních četností rozptylů charakteristiky
25. tabulka sloupcových relativních četností rozptylů charakteristiky
26. tabulka řádkových relativních četností rozptylů charakteristiky
27. tabulka absolutních směrodatných odchylek charakteristiky
28. tabulka relativních četností směrodatných odchylek charakteristiky
29. tabulka sloupcových relativních četností směrodatných odchylek charakteristiky
30. tabulka řádkových relativních četností směrodatných odchylek charakteristiky

Úplnou sestavu se všemi třiceti možnostmi neužíváme téměř nikdy; obvykle volíme tři–čtyři transformace tabulky a dva–tři statistické ukazatele. Možnost volby činí program velmi variabilním a zároveň krajně ekonomickým.

Ekonomika čtení výsledků

Ekonomika čtení výsledků je namnoze dána už ekonomikou zadávání a ekonomikou výpočtů: tím, že tabulky zadáváme uspořádané do matice, jsme nuceni vnášet do dat určitou strukturu, která nám potom zpětně pomáhá při jejich interpretaci; tím,

že volíme výpočet a tisk jenom relativních částí tabulky, šetříme nejenom strojový čas počítače, ale i čas svůj při interpretaci.

Kromě toho však o ekonomice čtení výsledků rozhoduje to, nakolik se nám podařilo při vytváření programů nalézt kompromis mezi dvěma navzájem se vylučujícími kritérii:

1. nárokem na co nejotevřenější popis tabulek
2. nárokem na co nejsevřenější presentaci výsledků.

Rozsáhlý a zevrubný slovní popis tabulek samozřejmě znemožňuje jejich presentaci na malé ploše. Šetří čas interpreta, protože ho nenutí překládat si názvy proměnných a názvy jejich variant z číselných symbolů do slovních označení.

Provádíme-li ale hlubší a zevrubnější analýzy vztahů omezeného počtu proměnných a studujeme-li velký počet tabulek vytvářených z téže omezené sady znaků, vžije se nám jejich číselný kód do paměti a slovní označení přestáváme postrádat. Číselné označení proměnných a jejich variant umožňuje pak sevřenější presentaci výsledků. To oceníme zejména, když před sebou máme tisíc i více tabulek; tehdy někdy přichází okamžik, kdy interpret pocítí, že by na čtení tabulek potřeboval stroj, protože člověk začíná být zahlcen množstvím informací, ztrácí přehled v příliš husté a rozsáhlé síti vztahů a začíná dávat přednost třeba omezené, ale přehledné informaci. Tu přicházejí ke cti přehledné matice koeficientů, tisk tabulek koeficientů z třídění vyšších stupňů ve formě větvících se síťových uspořádání a další zpřehledňující postupy.

V jistém smyslu počítač za člověka tabulky skutečně „číst" může. Příslušný algoritmus bývá založen na tom, že počítač testuje nulovou hypotézu na všech korelacích v matici všech znaků se všemi znaky. Tiskne však jenom ty tabulky, kde zavedený koeficient korelace anebo kontingence ukazuje hodnotu, významnou na předem zvolené hladině. Hladina významnosti může být stanovena fixně (např. 95 %), anebo může být pro každý výpočet volena a při hledání relevantních vztahů snižována nebo zvyšována. Je to onen známý postup, který bývá přirovnáván k výlovu rybníka: v tomto případě jako bychom zmenšovali postupně oka sítě, když se při prvním zátahu nechytlo dosti velkých ryb a my se rozhodli zátah opakovat s tím, že se spokojíme i s menšími.

Jemnější variantou tohoto postupu jsou pak programy, které nevybírají automaticky tabulky, ve kterých se ukázal statisticky významný vztah mezi dvěma znaky, ale pracují o úroveň níž: vyhledávají například na základě znaménkového testu jednotlivá pole, ve kterých se objevil statisticky významný vztah mezi variantami dvou znaků a vybírají za interpreta a tisknou jen informaci o těch kombinacích variant znaků, mezi kterými je takový vztah. Tento postup má smysl především při práci s nominálními znaky. Máme-li mezi sledovanými znaky například znak „povolání" a znak „vlastnictví předmětů", počítač sám vyhledá a vytiskne, že lékaři mezi ostatními povoláními častěji vlastní auto. Projde tak všechna povolání a všechny sledované předměty a vytiskne seznam povolání, která častěji vlastní auto, a seznam předmětů, jež častěji vlastní např. lékaři. Pokud program pracuje i s negativními korelacemi, zjistí a vytiskne pro nás i seznamy negativních vztahů — např. že zemědělští dělníci častěji nevlastní rekreační chalupu apod.

Proti těmto postupům, jež algoritmizují už určitou část analýzy výsledků a automatizují výběr relevantních vztahů, existují některé meritorní námitky: počítač samozřejmě neodliší triviální vztahy (tak např. vytřídí jako relevantní a vytiskne tabulku o vztahu mezi věkem a rodinným stavem) a bráníme-li se redundanci zvýšením nároku na hladinu významnosti, pomine některé vztahy, které mohou meritorně mít značný význam. Jako všude, i tu platí, že kritický rozum nelze žádným mechanismem nahradit a i počítač může být dobrý sluha, ale zlý pán. V rukou kvalifikovaného výzkumníka, který je si vědom jejich limitů a dokáže správně rozhodnout o jejich vhodném nasazení, jsou však tyto programy velmi cenným nástrojem.

Stranou jsme vědomě ponechali programy, které složitějšími statistickými operacemi přinášejí novou informaci o struktuře vztahů mezi znaky v množině dokumentů, jako je faktorová analýza, trsová analýza, rozpoznávání obrazců, path-analýza apod. I tyto programy pochopitelně zvyšují také ekonomiku „čtení" výsledků, ale výraz „čtení" tu musíme brát velmi abstraktně: umožňují totiž „čtení" struktur, které jsou jinak nezřetelné, a to už je kvalitativně jiná záležitost.

Samočinné počítače používáme při zpracování dat proto, aby nám ušetřily práci, čas a náklady. Příprava programů pro analýzy výsledků sociologických výzkumů a sociálních ukazatelů a dat je záležitost pracná, nákladná a náročná na čas. Tak například na systému programů SPSS, který je užíván v řadě států a nedávno byl zakoupen i pro některé počítače v Československu, pracovalo deset odborníků a jeho dovedení do konečné podoby trvalo téměř deset let. Výsledkem je velmi variabilní sada programů s širokým spektrem možností, pracující opravdu efektivně. Nevyplatí se nám ale vždycky zakoupit tak rozsáhlý programový systém, je velice drahý. I při tvorbě specializovaných systémů s užším rejstříkem možností však musíme bedlivě sledovat otázku jejich ekonomiky, a to nejenom ekonomiky tvorby programů, ale i jejich výsledné parametry z hlediska efektivnosti při zadávání, zpracování a vyhodnocení dat. Všechny tyto aspekty jsou u každého, i relativně jednoduchého systému ve vzájemné závislosti a složitě se ovlivňují. Konečné cílevědomé vyvážení má pak rozhodující vliv na produktivitu systému a mnohonásobně se vyplatí. Bylo by chybou tuto skutečnost přehlížet. Trvalý zřetel na ekonomiku není nedůstojný vědy. Volba přiměřené ekonomické strategie se stává neodlučnou součástí každého výzkumného projektu, programování pro samočinné počítače nevyjímaje, pochopitelně.

Ivo Možný

PROGRAM SOC, JEHO MOŽNOSTI, MEZE A DOSAVADNÍ ZKUŠENOSTI

Již více než čtyři roky mají zaměstnanci a studenti oddělení sociologie filozofické fakulty UJEP možnost využívat vlastní program na statistické zpracování empirických údajů — program SOC. Celý program vznikl ve Vědeckometodickém středisku pro výpočetní techniku při katedře aplikované matematiky na přírodovědecké fakultě UJEP na tamním samočinném počítači MSP-2A.[1]

V době, kdy jsme vypracovali program SOC nebylo technické vybavení počítače MSP—2A příliš vhodné pro užití při zpracování materiálů z empirických sociologických výzkumů, které klade vysoké požadavky zejména na vstupní a výstupní zařízení samočinného počítače a na kapacitu jeho paměti. A právě tyto prvky patří na počítači MSP—2A mezi slabé. Proto jsme na návrh M. Gregora omezili formu vstupních dat na *data výhradně předem vytříděná.*

Druhou odlišností programu SOC oproti ostatním programům používaným ke zpracování dat ze sociologických výzkumů na samočinných počítačích je jeho obsah. Program SOC se zaměřuje na stále ještě nejčastější případ znaků, které se v současných sociologických výzkumech vyskytují — tj. na *znaky nominální a ordinální.*

Konečně třetí koncepční zvláštností programu SOC měla být *jednoduchost vstupních instrukcí a přehlednost i srozumitelnost výsledků.* Program SOC by měl umožnit i sociologům neškoleným v otázkách využívání samočinných počítačů počítat některé statistické charakteristiky.

Program SOC byl vyhotoven v programovacím jazyku AUTOKÓD AU—MSP—2A, přesněji v jeho variantě AU 10.

Základní charakteristiky programu SOC:

I. Forma vstupních dat

Program SOC není, jak jsme již uvedli, v žádném případě schopen provádět třídění dat. Pro jeho vstup je třeba mít data již předem vytříděná — u malého počtu zkoumaných případů ručně; u větších na mechanických třídičkách, tabelátorech apod., či používat již vytříděných tabulek na jiných samočinných počítačích atd.

---

[1] Počítač MSP-2A je československým počítačem druhé generace a koncepčně zapadá do stavu našich projektů samočinných počítačů v letech 1963—1965. Dosahuje rychlosti okolo 7000 operací za vteřinu. V době vzniku programu SOC v letech 1969—1970 byla jeho jedinou pamětí ferritová paměť s kapacitou 10 000 slov. Nejrychlejším vstupem je snímač děrné pásky typu FS 1500 s rych-