

Zbyněk Štěpánek

Dopravní stavby Olomouc

ZKUŠENOSTI S POUŽÍVÁNÍM PARAMETRICKÝCH PROGRAMŮ

V OBLASTI ZPRACOVÁNÍ DAT

1. Úvod

Jednou z nejstarších metod úspory programátorské práce, uplatňované již při navrhování vyšších programovacích jazyků, je metoda procedur a podprogramů. Symbolické parametry se nahrazují parametry aktuálními a několik podobných algoritmů se realizuje jediným, několikrát použitým algoritmem. Většinu úloh z oblasti zpracování hromadných dat můžeme rozdělit do tříd podle jisté algoritmické podobnosti. Tak jednu třídu vytvoří konverze datových souborů, do jiné třídy zařadíme modifikace souborů apod. Zde se nabízí myšlenka rozšíření principu procedur a parametry na samostatné parametrické programy. Každý takový program by řešil úlohy jedné třídy. V minulosti byly skutečně parametrické programy vytvořeny. Dnes můžeme pomocí parametrů zadávat konverze dat, změnová řízení, výstupy do tiskových souborů a další vysoce frekventované úlohy z oblasti zpracování dat.

Obsahem tohoto příspěvku nebude popis nějakého nového systému parametrických programů, ale rozbor některých zkušeností získaných dlouhodobým používáním známých parametrických programů. Jedná se o program OPPT /obecný parametrický program tisku a výběru/ a program OPFM /obecný parametrický program modifikací/ ze známé řady programů OPP, dodávaných z výpočetního střediska n.p. Geodetický ústav Praha. Zkušenosti byly získány na počítači EC 1033 s operačním systémem DOS.

2. Struktura parametrických programů OPPT a OPFM

Programy OPPT a OPFM se v současné době využívají asi ve 40 výpočetních střediscích. Nebudeme se proto zabývat jejich podrobným popisem, ale soustředíme se pouze na ty funkce programů, které nás budou v dalších odstavcích zajímat.

Označme symboly A-F tyto programové sekce :

- A - čtení věty hlavního souboru
- B - vyhledání odpovídající věty v doplňkovém souboru
- C - zařazení věty do skupiny podle zadaných podmínek
- D - výpočty a operace s údaji věty.
- E - zápis věty do výstupního souboru
- F - zpracování věty pro tiskový výstup

Program OPPT potom realizuje tyto algoritmy: ABCDEF, ACDEF, ABCDF, ACDF, ABCDE. Např. ACDEF znamená, že program čte větu vst. souboru, zařadí ji do skupiny, jejíchž podmínkám vyhověla, realizuje výpočty a modifikace s údaji věty, zapíše výslednou větu do vstupního souboru a některé údaje převede do tiskového souboru. Podrobněji si všimněme činnosti C. Věta splňuje zadanou podmínku, patří-li určený údaj z věty do daného intervalu hodnot. Interval může být spojitý nebo může jít o seznam diskrétních hodnot. Máme-li např. program OPPT vybrat ze vstupního souboru právě ty věty, pro které platí $ZAVOD = 4$ nebo $ZAVOD = 5$ a současně $UCET = 311$, vytvoříme 2 podmínky: $ZAVOD \in \{4; 5\}$ a $UCET = 311$, které musí být splněny současně. Podle splnění nebo nesplnění těchto podmínek se věty rozdělí do dvou skupin, program OPFM rozděluje věty do tolika skupin, kolik je parametry zadáno.

3. Systematický popis parametrů

Uživatel se seznamuje s parametrickým programem zpravidla tak, že si přečte návod k použití programu. Ten se obvykle po stručně načrtnuté funkci programu zabývá především syntaxí parametrů. Méně pozorností se už věnuje sémantické stránce parametrů a také tomu, že ne každé syntakticky správné zadání parametrů vede k úspěšnému průběhu programu. Pokusme se nyní o systematický a úplný popis parametrů.

Buďte s_1, \dots, s_n symbolické parametry programu, tedy místa, na které dosazujeme konkrétní hodnoty p_1, \dots, p_n aktuálních parametrů. Pro každé s_i je dána množina F_i hodnot aktuálních parametrů. Ze všech kombinací aktuálních parametrů je dále vybrána podmnožina správných zadání parametrů, pro která má program smysl /je zaručen jeho správný průběh/. Ke každému symbolickému parametru může být určena jeho implicitní hodnota nebo může jít o tzv. parametr s prázdnou hodnotou, což značí, že za symbolický parametr nemusí být dosazena žádná konkrétní hodnota.

Příklad zadání některých parametrů programu GFPT :

s_i	F_i	impl.hodn.	prázdná h.	sémantika
LEN	/1;99/	-	ne	délka údaje v bytech
ADR	/0;999/	-	ne	adr. údaje ve větě
F	/C,P,R,Z/	-	ne	tvar údaje
N	/0;5/	0	ano	počet des.míst
S	/S/	-	ano	znak sumarizace
LI	/1;99/	-	ne	řádek tisku
COL	/1;160/	-	ne	počet.sloupec tisku
V	/V/	-	ano	údaj z doplňkového souboru
A	/1;160/	120	ano	délka tisk.řádku
RS	/18;30 000/-	-	ne	délka věty v bytech
BS	/18;30 000/-	-	ne	délka bloku v bytech

Z příkladu vyplývá, že délka údaje musí být vždy zadána hodnotou z intervalu /1;99/, počet desetinných míst nemusí být zadán a je v tom případě roven nule. Podobně znak S nemusí být uveden a sumarizace údaje se neprovádí. V ustálené formě zápisu pro syntaktická pravidla je např. tisk údaje zadán takto : ADR, LEN, .. F, LI, COL, [N] [S] [V].

Stanovením pravidel pro zápis parametrů/posloupnost parametrů, oddělovací znaménka/ je dokončen jejich syntaktický popis a vysvětlena sémantika. Je ale zřejmé, že ne každé zadání parametrů, které vyhoví výše popsaným požadavkům, bude mít smysl. Musíme stanovit další omezující podmínky pro parametry tak, abychom dostali množinu správných zadání.

Tak například musí platit vztahy :

$$ADR + LEN \leq 999$$

údaj nesmí přesáhnout hranici věty

$$COL + LEN + 1 \leq A$$

tisk údaje v rámci řádku /včetně 1 místa na des.čárku/

$$LEN \geq N$$

údaj neobsahuje více des.míst, než je jeho délka

$$ADR < RS$$

adresa údaje je v rámci věty

$$BS = n \cdot RS$$

/n - celé číslo/ délka bloku je celistvým násobkem délky věty.

Parametrický program obvykle provádí podrobnou analýzu syntaktické správnosti parametrů. Testuje jejich správnou posloupnost, oddělovací symboly/čárka, středník/ a prověřuje, zda konkrétní hodnoty p_i patří do množin P_i . Další podmínky pro parametry, které vyplývají z jejich vzájemných vazeb, nebývají při analýze důsledně prověřeny. Ani návod pro uživatele je všechny nepopisuje a ponechává se na zkušenostech a důvtipu uživatele programu, jak se s nimi vypořádá. Praxe však ukazuje, že je efektivnější zahrnout kontrolu vazeb mezi parametry do syntaktické analýzy a snížit tak pravděpodobnost nenormálního konce programu v důsledku chybně zadaných parametrů.

V souvislosti s parametrickými programy vyatupuje programátor pouze jako jeho tvůrce, případně udržovatel, dále pracují s těmito programy převážně ne-programátoři, operátoři, prostě uživatelé parametrických programů. Pro ně má úplný popis parametrů zásadní význam. Jsou vodítkem při seznamování s funkcí programu, při provádění zásahů do hotového zadání parametrických programů. Celková koncepce parametrů má nesmírný psychologický význam. Ze dvou parametrických programů, které řešily tutéž třídu úloh, jsme vybrali program s jednodušším, přehlednějším a logičtějším zadáváním parametrů. Mnohé programy s parametry se nedočkaly většího rozšíření především pro těžkopádnou práci s parametry. Příklad srozumitelných a snadných parametrů dávají programy OPPT a OPPM. Podle zkušeností je uživatel těchto programů schopen po vytvoření 10 až 15 správných zadání pracovat dále z paměti,

bez používání příručky. Tato okolnost je důležitá také proto, že parametrické programy jsou určeny mj. pro řešení náhlých požadavků provozu výpočetního střediska.

4. Požadavky na logiku parametrických programů

Vešle požadavku základní logické správnosti musí parametrický program splňovat jistou logickou úplnost. V předchozím odstavci jsme definovali množinu správných zadání. Parametrický program je v rámci určité své činnosti logicky plný, jestliže realizuje každé správné zadání a logický součet zadání je pro tuto činnost základní množina jevů /jistý jev/. Příklad vezměme opět z programů OPPT a OPFM. Zkoumejme logickou úplnost OPPT pro činnost C -zařazení věty do skupiny podle splnění daných podmínek. Jsou-li a_1, \dots, a_n elementární podmínky, umožňuje program OPPT sestavit výslednou podmínku pouze jako konjunkci elementárních podmínek. Příklad alternativy mezi podmínkami nelze programem OPPT obsloužit. Tak např. výběr položek materiálů, pro něž $PPC > 100$ nebo $VC > 80$, není pomocí OPPT řešitelný. Výběrová činnost uvedeného programu tedy není logicky úplná. Rozšířením výběru o negaci, tedy výběr položek, které nespĺňují dané podmínky, by OPPT dokázal řešit všechny úlohy výběru. Podle tautologií $A' \wedge B = (A \vee B)'$ a $A \Rightarrow B \equiv A' \vee B$ /symbol ' značí negaci/, by program realizoval i alternativní a implikativní vztahy mezi podmínkami. Vezměme nyní výběr vět, jak jej provádí program OPFM. Věta je zařazena do určité skupiny, jestliže splňuje současně všechny elementární podmínky dané skupině příslušné. Mezi skupinami podmínek je vztah alternativy. Výše uvedený výběr položek materiálů se zajistí programem OPFM definováním podmínky $PPC > 100$ pro skupinu 1, podmínky $VC > 80$ pro skupinu 2 a podmínky $PPC \leq 100, VC \leq 80$ pro skupinu 3. Ke skupinám 1 a 2 zadáme parametry dalšího zpracování, ke skupině 3 určíme parametr vyřazení věty ze zpracování. Program OPFM umožňuje provést výběr vět z daného souboru podle libovolně zadaných podmínek, jeho výběrová činnost je logicky úplná. Podobně můžeme prověřit ostatní činnosti parametrických programů.

5. Stanovení třídy úloh řešitelných parametrickými programy

Všechny předchozí úvahy směřovaly k určení třídy úloh, které můžeme řešit pomocí určitého parametrického programu. V první řadě musí úloha splňovat specifikace dané programem s parametry. Jsou to předpoklady jako setříděnost souborů, organizace souborů, typy zařízení a další. Komplikovanější je otázka podobnosti mezi algoritmem úlohy a algoritmem parametrického programu. V odstavci 2 jsme popsali algoritmy programů OPPT a OPFM. Říkejme jim pro jednoduchost parametrické algoritmy. Máme-li úlohu vytvořit tiskový soubor z údajů souboru stavových položek materiálů a údajů z ceníku materiálů jako doplňkového souboru, je algoritmus úlohy v našem značení ABCDF. Shodnost tohoto algoritmu s parametrickým algoritmem určuje, že příklad lze řešit pomocí OPPT. Máme nyní toto zadání: Srovnáním položek stavů materiálů a odpovídajícími položkami ve starém a novém ceníku vypočítat hodnotu přecenění a přeceňované položky vytisknout. Zde máme úlohu se dvěma doplňkovými soubory a algoritmem ABCDF. Programy OPPT a OPFM neumožňují použití dvou doplňkových souborů. Rozložíme úlohu na dva kroky: ABCDE - programem OPFM porovnáme materiálové položky se starým ceníkem a uchováme potřebné údaje; ABCDF - programem OPPT porovnáme vzniklý soubor s novým ceníkem, vypočteme hodnotu přecenění a provedeme tisk. Po tomto rozkladu je tato úloha parametricky řešitelná. Pravidlo pro stanovení řešitelnosti problému použitím parametrických programů: Úloha je řešitelná programy OPPT a OPFM, jestliže její algoritmus dostaneme z parametrických algoritmů zjetzením s případným opakováním jednotlivých symbolů. Algoritmus s n -násobným opakováním symbolu nahradíme n parametrickými algoritmy s daným symbolem. Jako příklad poslouží zadání provést rozdělení vět vstupního souboru do tří skupin podle tří podmínek a jejich zápis do tří různých souborů. Algoritmus úlohy je ACDEEE. Protože v OPPT i OPFM je možný jen jeden vstupní soubor, provedeme tři kroky ACDE ACDE ACDE, vždy se stejným vstupním souborem a jednou podmínkou. Podobně se řeší i složitější úlohy.

Příklad zadání, které není parametrickými programy řešitelné: 2 každé věty vstupního souboru vytvořit dvě věty a zapsat je do výstupního souboru. Schema je ACDEDE, což nevyhovuje pravidlu pro řešitelnost pomocí OPPT a OPTM, protože se zde opakuje skupina dvou symbolů.

Naše úvahy se týkaly pouze dvou parametrických programů. Pokud bychom mohli uvažovat propojení více programů a získali bychom další rozšíření třídy úloh řešitelných parametrickými programy. S růstem počtu kroků při rozkladu algoritmu úlohy vznikají větší nároky na strojový čas, zpracování je delší než při řešení jediným cobolským programem. Porovnejme nároky parametrických programů s programováním v Cobolu:

Cobol	parametrický program
analýza, vývojový diagram	rozklad na kroky
zápis programu	zápis parametrů
děrování programu	děrování parametrických štítků/obvykle v počtu 5 - 40/
kompilace, sestavení	
případné opravy	případné opravy parametrických štítků
ladění	
spuštění	spuštění

Největší výhodou parametrických programů - odstranění fáze kompilace a ladění / v 90% případů jsme pro úspěšné spuštění parametrických programů nepotřebovali více jak dva pokusy/ - převažuje nad skutečností, že u složitějších úloh s rozkladem na více kroků je spotřeba strojového času při spuštění větší než u jediné cobolského programu. Zvláště při řešení nestandardních požadavků je použití parametrických programů efektivní, je-li poměr strojového času při řešení s parametry ke strojovému času při použití Cobolu do poměru 1,7 : 1. Rozdělení úlohy na kroky řešené parametrickými programy má další výhody. Je to lepší přehlednost,

lepší možnosti restartu, snadnější hledání chyb při nenormálním konci programu.

6. Postup při nenormálním konci parametrického programu

V příručkách s popisem parametrického programu pravidelně chybí jakýkoliv návod k postupu při nenormálním konci programu. Oddělení osoby tvůrce programu a osoby uživatele programu nás nutí k vypracování záseď pro opravu chybných zadání a chybných průběhu parametrických programů. Situace uživatele parametrického programu má na rozdíl od situace tvůrce např. cobolského programu některé zvláštnosti:

- Uživatel většinou nemá k dispozici zdrojový text programu.
- Má-li zdrojový text, těžko se v něm pro jeho rozsáhlou a komplikovanost orientuje.
- Spoléhá na správnost algoritmu programu a hledá chyby především v zadání parametrů a v datech.
- Nezná chování programu při zadání parametrů, které vyhovuje syntaktickým pravidlům, ale není správné.

Podíl hlavních příčin nenormálního konce parametrických programů je podle dlouhodobého pozorování tento:

syntaktické chyby v parametrech	65%
sémantické chyby v parametrech	15%
chyby dat v souborech	10%
chybný algoritmus param. programu	1%
jiné příčiny/technické apod./	9%

Syntaktické chyby v parametrech zachytí syntaktická analýza v programu s příslušným výpisem zpráv o chybách. Sémantické chyby parametrů jsou způsobeny buď nesplněním některé z podmínek pro správné zadání parametrů nebo nesouhlasem parametrů se skutečným stavem dat. Stejná situace vzniká při správném zadání, ale chybné struktuře dat. Překvapující může být zjištění, že dochází k chybám v algoritmu parametrického programu. Značný počet možností zadání parametrů způsobuje, že důkladné prověření činnosti programu je možné až během rutinního používání programu. Po odstranění reklamovaných chyb je spolehlivost algoritmu

parametrického programu téměř 100%.

7. Závěr

Cílem příspěvku bylo shrnout některé poznatky získané při dlouhodobém používání parametrických programů. Nešlo o úplný přehled otázek spojených s uvedenými programy, ale spíše o volně řazené problémy, jak vyplynuly z praxe. Zaměřili jsme se na dva parametrické programy - OPPT a OPFM, které prokázaly ze všech využívaných parametrických programů největší efekt. Jejich největší přínos spatřujeme v tom, že jsou pro nás jedinými prostředky pro rychlý zásah v případě nečekaných chyb při rutinním zpracování hromadných dat. Dřívější programování opravných programů, případně znovuspouštění chodů po opravách dat, je dnes nahrazeno několika minutami práce na parametrech programu, kterým se opraví chybná data nebo vytiskne několik chybějících stran sestavy.

Snad je toho přesvědčování již dost. Zřejmě stále platí, co napsal Pascal: "Zpravidla nás přesvědčují ty důvody, které sami objevíme než ty, na které přišli jiní."

Literatura:

- /1/ Obecný parametrický program pro využití a zhodnocení informací pořízených a udržovaných systémy hromadného zpracování dat na počítači EC 1030, ÚVT Ostrava, 1976
- /2/ Obecný parametrický program modifikací, Geodetický ústav Praha
- /3/ Vl. Mach: Programy OPPT, OPFM, OPFS, OPGT, rukopis