

VLASTNOSTI PŘEKLADAČE ROZHODOVACÍCH TABULEK PROTAB - 25

1. Úvod

Základní koncepce překladače PROTAB-25 byla převzata z překladače PROTAB-21, který se osvědčil v praxi. Během uživatelského provozu překladače PROTAB-21 jsme postupně shromažďovali připomínky a náměty uživatelů na vylepšení funkce překladače, a tyto změny spolu se změnami plánovanými nezávisle na konzultacích s uživateli jsou realizovány v překladači PROTAB-25 pro operační systém DCS-3/JS.

Překladač PROTAB-25 pracuje v principu stejně jako PROTAB-21, a to jako předprocesor kompilátoru vyššího, tzv. hostitelského programovacího jazyka. Čte program zapsaný v hostitelském jazyce, který může obsahovat RT. Každou nalezenou RT opiše jako poznámku a bezprostředně za její opis vygeneruje překlad RT do hostitelského jazyka. Tisk tohoto vygenerovaného úseku programu může být v protokolu o překladu na přání potlačen (prostřednictvím monitorového příkazu OPTION).

Pro konverzi byla pro své známé výhody opět použita metoda postupného větvení; poskytuje srozumitelný a současně nejrychlejší přeložený program, a je vhodná pro menší RT, které budou podle předpokladu tvořit dalekou většinu zpracovávaných RT.

V dalším postupně popíšeme jednotlivá rozšíření možností překladače PROTAB-25.

2. Rozšířený zápis

Stejně jako PROTAB-21, i PROTAB-25 dovoluje zpracovávat binární RT, tj. RT, v nichž stavy podmínek a volby čin-

ností jsou formulovány ve dvouhodnotové logice: podmínka může být splněna nebo nesplněna, činnost se má nebo nemá provést. Kromě toho lze však překládat také rozšířené a smíšené RT, v nichž stavy všech resp. některých podmínek a/nebo volby všech resp. některých činností jsou formulovány ve vícehodnotových logikách: zjišťuje se, jakým způsobem je podmínka splněna, a zadává se, jak se má činnost provést.

Podmínka s rozšířeným zápisem stavů je vlastně zkrácené vyjádření několika binárních podmínek, které nazýváme dílčí podmínky. Dílčí podmínky mají část formulace společnou a část rozdílnou. Společná část může být i prázdná, rozdílná část musí být pro každou dílčí podmínku jeden souvislý řetěz znaků. Rozšířený zápis podmínky vypadá tak, že společná část formulací dílčích podmínek se zapíše na místo formulace podmínky v kvadrantu podmínek (1. kvadrant), přičemž na místě rozdílných částí se zapíše znak otazník. Rozdílné části formulací dílčích podmínek se zapíše do kvadrantu stavů jako stavy podmínky.

Nevýznamný stav resp. prázdná volba se u rozšířeného zápisu zapisuje znakem mezera nebo pomlčka.

Jednotlivé stavy rozšířeného zápisu se oddělují oddělovačem: pro jazyky COBOL a FORTRAN je oddělovačem znak svíslá čárka, pro jazyk PL/I je oddělovačem znak vykřičník. Před prvním a za posledním rozšířeným stavem musí být rovněž zapsán oddělovač.

Jako stav podmínky s rozšířeným zápisem lze kromě části formulace dílčí podmínky, mezer a pomlčky zapsat ještě klíčové slovo ELSE ve významu "není splněna žádná z dílčích podmínek". Aby byl rozšířený zápis podmínky úplný, tj. aby formálně postihoval všechny možnosti, musí obsahovat klíčové slovo ELSE a/nebo nevýznamný stav.

Dílčí podmínky se v přeloženém programu testují odleva doprava (opakování rozšířeného stavu nemá vliv). Uživatel to musí mít na paměti v případě, že se dílčí podmínky vzájemně nevylučují; v tom případě může naopak známého pořadí testování podmínek využít.

Příklad:

MESICNI-PRIJEM < ? | 3000 | 6000 | 3000 | ELSE |

Dílčí podmínky se budou testovat v tomto pořadí:

MESICNI-PRIJEM < 3000, MESICNI-PRIJEM < 6000, ELSE

Pozn.: Uvedené pravidlo o pořadí testování dílčích podmínek zleva doprava platí pouze pro explicitně uvedené dílčí podmínky; podmínka ELSE se testuje vždy jako poslední bez ohledu na to, ve kterém sloupci je klíčové slovo zapsáno.

Činnost s rozšířeným zápisem voleb je zkrácené vyjádření několika dílčích činností. Pro rozšířený zápis činností platí stejná pravidla jako pro rozšířený zápis podmínky, jen s tím rozdílem, že jako rozšířenou volbu nelze zapsat klíčové slovo ELSE.

U RT, která je zapsána celá binárně, se za pravidlo RT považuje každý sloupec, který obsahuje významný stav nebo neprázdnou volbu.

Rozšířený zápis stavu podmínky nebo volby činností platí pro všechny sloupce, které zabírá, tj. sloupce mezi oddělovači rozšířeného zápisu. Tentýž rozšířený zápis lze tedy použít pro několik pravidel RT.

Jestliže RT obsahuje aspoň jeden binární zápis (podmínky nebo činnosti), pak se za pravidla RT považují právě ty sloupce, které obsahují nějaký nemezerový binární zápis stavu podmínky nebo volby činnosti.

Jestliže RT obsahuje pouze rozšířené zápisy, považují se za pravidla ty sloupce, které neobsahují oddělovač. Mezi identickými těsně sousedícími sloupci se nehleší nejednoznačnost - skupina takových sloupců se považuje za jedno pravidlo.

Je-li v nějakém sloupci současně nemezerový binární zápis a oddělovač, nepovažuje se takový sloupec za pravidlo, ale hlásí se jako chybný.

Pravidlo JINAK se zapisuje u binárního i rozšířeného zápisu jako pravidlo se samými nevýznamnými stavy a aspoň jednou neprázdnou volbou; lze je zapsat kdekoli mezi ostatními pravidly.

Příklady:

PŘÍJEM ?	< 1000	< 2000	< 3000	ELSE
ZENATY	1 0	1 0	1 0	1 0
pravidla	↑ ↑	↑ ↑	↑ ↑	↑ ↑

A ?	< B	= C	ELSE	
C ?	= E	ELSE	< B	
pravidla	↑	↑	↑	↑

3. Hostitelské jazyky

Zatímco překladač PROTAB-21 pracuje pouze v těsné návaznosti na kompilátor COBOL, překladač PROTAB-25 překládá RT do tří hostitelských jazyků: COBOL, FORTRAN a PL/I. Návaznost na kompilátor není tak těsná: PROTAB-25 kompilátor sám nevyvolává, ale spoluprací obou programů řídí procedura příkazů pro systémové programy MONITOR a POWER.

Pro jednotlivé hostitelské jazyky se liší jak vstup, tak také výstup překladače PROTAB-25. Formát zápisu RT je pro všechny tři hostitelské jazyky téměř stejný. Zejména jsou stejné druhy štítků a členění jednotlivých štítků na pole. Rozdíly jsou pouze ve štítku záhlaví, který musel být přizpůsoben jednotlivým jazykům. Štítek záhlaví slouží k rozpoznání RT ve zdrojovém programu a k předání některých informací o RT překladači. Kanadská norma, podle které byl formát štítků navržen, počítá s použitím RT ve všech těchto jazycích, a proto nebylo třeba upravovat štítek záhlaví pro rozpoznání RT v textu. U jednotlivých jazyků se však liší informace předávané překladači ve štítku záhlaví:

a) jméno RT: v jazyce COBOL a PL/I má shodný formát, a to řetěz nejvýše 30 znaků, skládající se z číselic, písmen a pomlček, který začíná písmenem a nekončí pomlčkou. V jazyce FORTRAN však má jméno RT tvar nejvýše pětimístného čísla.

b) příznak rozskoku: týká se pouze jazyka COBOL; indikuje, že se má podmínková část RT přeložit do více paragrafů. Podrobnosti jsou popsány v následujícím § 4 tohoto referátu.

Výstup překladače (přeložená RT) se samozřejmě liší v důsledku odlišné syntaxe hostitelských jazyků. Liší se ale nejenom formálně, ale i funkčně.

a) COBOL: přeložená RT má tvar jednoho nebo několika paragrafů; vždy se však chová jako jeden paragraf se jménem RT, na který se lze odvolat příkazem GO TO nebo PERFORM, nebo který lze provést postupným prováděním programu. Počet paragrafů, které tvoří úsek programu vygenerovaný z jedné RT, není explicitně omezen. Konkrétní tvar překladu závisí na tom, zda RT obsahuje úvodní pravidlo, zda obsahuje podmínky, zda obsahuje podmíněné činnosti a zda se její podmínková část překládá do více paragrafů. Celkem je možných deset různých tvarů vygenerovaného úseku programu z jedné RT. Jednotlivé tvary zde vzhledem k informativnímu zaměření referátu nebudeme popisovat.

b) FORTRAN: přeložená RT má tvar posloupnosti příkazů, z nichž první má číslo příkazu shodné se jménem RT. Na RT se tak lze v programu odvolat příkazem GO TO, nebo ji lze provést při sekvenčním provádění příkazů programu. Jestliže by uživatel chtěl RT volat příkazem CALL, musel by RT uzavřít do deklarace příslušné vnější procedury.

c) PL/I: přeložená RT má tvar skupiny DO, tj. posloupnosti příkazů uzavřené do příkazových závorek DO; ... END; U skupiny DO je návěští shodné se jménem RT. Na přeloženou RT se tedy lze odvolat příkazem GO TO, či ji lze provést při sekvenčním provádění příkazů programu. Jestliže by uživatel chtěl volat RT příkazem CALL, musel by RT uzavřít do deklarace příslušné vnější nebo vnitřní procedury.

4. Překlad velkých RT do jazyka COBOL

Při překladu RT do jazyka COBOL se celá podmínková část RT zpravidla přeloží do jediné podmínkové věty, která se skládá z několika do sebe vložených příkazů IF. Do této podmínkové věty se opíše formulace podmínek, formulace ne-odmínaných činností a odkazy na podmíněné činnosti. U složitějších a větších RT vychází tato podmínková věta velmi dlouhá. Kompilátor COBOL však vyžaduje, aby překlad jednoho paragrafu nepřesahoval ve strojovém kódu 4096 znaků. Zkušenosti s používáním překladače PROTAB-21 ukázaly, že toto omezení COBOLu je dale nezávažnější ze všech omezení na pře-

ložitelné RT. Proto byl pro překladáč PROTAB-25 vytvořen nový alternativní způsob překladu, kdy se místo do jediné věty překládá do několika kratších vět umístěných po jedné do zvláštních paragrafů. Na tyto paragrafy se odkazuje příkazem PERFORM podle stavu prvních dvou podmínek RT. V paragrafu vlastní podmínkové části se testují první dvě podmínky, a podle jejich stavů se provede jeden z dalších paragrafů, obsahujících podmínkové věty s testy zbývajících podmínek a s formulacemi činností (či odkazy na činnosti). Celkem se podmínková část RT přeloží do $1 + a_1 a_2$ paragrafů, kde a_i je arita i -té podmínky (tj. počet hodnot logiky i -té podmínky).

5. Optimalizace

Překladáč PROTAB-21 překlad RT neoptimalizuje, mj. proto, aby uživatel měl zaručeno, že se podmínky budou testovat v pořadí, které uživatel určil v zápisu RT. Pořadí testování podmínek může totiž být v některých případech důležité.

U překladáče PROTAB-25 se překlad RT optimalizuje nej-jednodušší metodou - přerovnáním podmínek před překladem. Na případy, kdy na pořadí testování podmínek záleží, je pamatováno tak, že uživatel má možnost optimalizaci zakázat (indikací ve šíťtku záhlaví); podmínky se pak testují v pořadí zápisu v RT.

Pořadí testování podmínek při optimalizovaném překladu RT je dáno srovnáním podmínek před překladem vzestupně podle hodnot součinu počtu nevýznamných stavů a arity jednotlivých podmínek: každý nevýznamný stav totiž obsahuje tolik jednoduchých stavů, jaká je arita podmínky.

Účinnost tohoto optimalizačního kritéria ukážeme na následujícím příkladu:

P1	1	1	1	1	1	0	0	0	0
P2	1	2	2	0	0	1	2	0	-
P3	-	1	0	1	0	1	1	1	0

Tato RT je úplná a jednoznačná. Její překlad metodou postupného větvení při zachování pořadí testů podmínek ze zápisu RT (tj. bez optimalizace) bude obsahovat celkem 10 testů podmínek; součet počtů testů nutných pro rozlišení

jednotlivých pravidel je 39. Schematické znázornění překladu do jazyka COBOL bychom mohli zapsat např. takto:

```
IF P1
IF P2 = 1 pravidlo 1 ELSE
IF P2 = 2
IF P3 pravidlo 2 ELSE pravidlo 3 ELSE
IF P3 pravidlo 4 ELSE pravidlo 5 ELSE
IF P2 = 1
IF P3 pravidlo 6 ELSE pravidlo 9 ELSE
IF P2 = 2
IF P3 pravidlo 7 ELSE pravidlo 9 ELSE
IF P3 pravidlo 8 ELSE pravidlo 9.
```

Součin počtu nevýznamných stavů a arity je pro podmínku P1 roven 0, pro podmínku P2 je roven 3 a pro podmínku P3 je roven 2. Proto se při optimalizovaném překladu nejprve přehodí druhá a třetí podmínka. Výsledný překlad bude obsahovat celkem 9 testů podmínek a součet počtů testů nutných pro rozlišení jednotlivých pravidel bude 35.

```
IF P1
IF P3
IF P2 = 1 pravidlo 1 ELSE
IF P2 = 2 pravidlo 2 ELSE pravidlo 4 ELSE
IF P2 = 1 pravidlo 1 ELSE
IF P2 = 2 pravidlo 3 ELSE pravidlo 5 ELSE
IF P3
IF P2 = 1 pravidlo 6 ELSE
IF P2 = 2 pravidlo 7 ELSE pravidlo 8 ELSE pravidlo 9.
```

6. Závěr

Podali jsme stručný přehled některých hlavních zlepšení vlastností překladače RT PROTAB-25 proti překladači PROTAB-21.

Překladač PROTAB-25 je nyní rozpracován do stadia programové realizace. V r. 1981 má projít mezinárodními zkouškami a stát se součástí operačního systému DOS-3/JS.

Věříme, že bude platným pomocníkem při snaze o racionalizaci programování ve všech oblastech vhodných pro nasazení rozhodovacích tabulek.