

Jiří Boleslav

TOS Kufin

STANDARDIZACE METODY RT

1. ÚVOD

Je bohužel skutečností, že metoda RT není přes své, podle názoru mého i mnoha jiných svých zastánců, nepopíratelné přednosti stále ještě dostatečně využívána v programátorské a analytické praxi. Myslím, že jedním z důvodů tohoto pro nás tak zarmocujícího faktu (i když rozhodně ne důvodem nejdůležitějším), je nedostatečná standardizace v této oblasti.

Ve světě a nakonec i u nás existuje značné množství překladačů RT, lišících se jak co do formy zápisu, tak co do způsobu jejího překladu. Chceme-li přenést RT z jednoho typu počítače na jiný, narazíme obvykle na značné potíže. Obtíže rovněž vznikají při přechodu programátora, zvyklého používat určitý překladač RT, na jiný typ počítače, vybavený buďto jiným (v lepším případě) nebo dokonce žádným překladačem RT. O potížích při přenesení překladače RT do jiného softwarově - hardwarového prostředí už raději pomlčme vůbec. Přitom vlastní princip metody je stále týž.

Je tedy nasnadě, že zavedení určitých standardů by mohlo celý problém značně zjednodušit.

2. OBLASTI STANDARDIZACE

Všimněme si nyní oblastí, jichž by se měla výše zmíněná standardizace dotýkat. Jsou to především oblasti

- terminologie
- formy zápisu RT
- tvorby překladačů RT.

Přitom terminologie je zde kladena na první místo ne snad kvůli své závažnosti, ale především z metodických důvodů.

3. TERMINOLOGIE

V domácí literatuře existuje bohužel značná terminologická nejednoznačnost. Pokusme se proto opřít o literaturu anglickou a uvést české ekvivalenty některých základních pojmů z dané oblasti, jichž se tyto problémy týkají především:

condition stub	- formulace podmínek
condition entry	- volba podmínek
action stub	- formulace činností
action entry	- volba činností
limited entry tables	- tabulky s jednoduchou volbou
extended tables	- tabulky s rozšířenou volbou
mixed tables	- " " "

Poznámka: Použití téhož českého termínu pro oba poslední anglické je záměrné. Domnívám se totiž, že rozlišovat mezi nimi je zbytečné.

4. NORMA CSA

Za první krok ke standardizaci metody RT lze považovat normu, vydanou Kanadským standardizačním úřadem (CSA). Tato norma se zabývá především formou zápisu RT.

Vychází z ní např. překladáč PROTAB pro EC 1021, vytvořený ve VÚMŠU Praha. Nedostatkem této normy je, že pojednává pouze o tabulkách s jednoduchou volbou. Rozhodovací tabulka je chápána jako struktura, sestávající ze tří částí: (viz obr. 1)

INICIALIZACE	
FORMULACE PODMÍNEK	VOLBA PODMÍNEK
FORMULACE ČINNOSTÍ	VOLBA ČINNOSTÍ

- obr. 1 -

- inicializační sekce
- sekce podmínek
- sekce činností.

Sekce podmínek se dále dělí na formulaci podmínek (tzv. 1. kvadrant) a volbu podmínek (3. kvadrant), sekce činností na formulaci činností (2. kvadrant) a volbu činností (4. kvadrant).

Inicializační sekce obsahuje příkazy, které se provádějí nepodmíněně. Jsou vykonávány postupně, před první podmínkou. Sekce podmínek i sekce činností mají obsah obvyklý u tabulek s jednoduchou volbou, tj. ve 3. kvadrantě lze uvádět pouze N, Y, nebo mezeru, ve 4. kvadrantě X, číslice 0 - 9 nebo mezeru.

V každé rozhodovací tabulce lze použít devět typů štítků:

- štítek řízení chodu
- štítek záhlaví tabulky
- inicializační štítek
- podmínkový štítek
- činnostní štítek
- pokračovací štítek
- poznákový štítek

- štítek konce tabulky
- štítek konce souboru

Všechny tyto štítky mají následující společná pole:

Sloupec 1 - 6	POŘADOVÉ ČÍSLO	Toto pole je volitelné a může být prázdné. Je-li použito, může obsahovat pouze číslice a pomáhá uživateli k ověření, že štítky jsou správně uspořádány.
7	PRÁZDNÝ	Toto pole musí být ponecháno prázdné.
8	TYP	Toto pole určuje typ štítku a musí být vždy vyplněno.
73 - 80	IDENTIFIKACE	Tyto sloupce lze použít k identifikaci štítků, tvořících tabulku. Mohou být použity libovolné znaky včetně mezery.

Pro zajištění maximální flexibility rozsahu formulace a volby tabulky je hranice mezi nimi proměnná. První sloupec (na štítku) části volby se udává ve štítku záhlaví tabulky v tzv. "sloupci volby". V tomto standardu se odkazuje na poslední sloupec formulace jako na XX a na první sloupec volby jako na YY. Zřejmě musí být $XX < YY$. Věnujme si nyní blíže jednotlivých typů štítků.

4.1 Štítek řízení chodu

Je volitelný, lze ho použít pouze jako prvního štítku RT.

Jeho struktura je následující:

Sloupec 8 TYP Musí obsahovat 'R'

Sloupec 9 - 72 NEDEFINOVÁN Tyto sloupce jsou k dispozici pro libovolné uživatelské informace o chodu.

4.2 Štítek záhlaví tabulky

Tento štítek je povinný pro všechny tabulky a musí být vždy použit před všemi dalšími štítky (s výjimkou předchozího). Jeho struktura je

Sloupec 8	TYP	Mustí být 'H'
9 - 16	ČÍSLO TABULKY	Toto pole je volitelné a může být prázdné nebo obsahovat čísla, pokud je číslování tabulky nutné
17	PRÁZDNÝ	
18 - 47	JMÉNO TABULKY	1 - 30 znaků
48	PRÁZDNÝ	
49 - 50	POČÁTEK VOLBY	Obsahuje dvomístné číslo, udávající počátek 3. a 4. kvadrantu
51 - 60	REZERVOVÁNO	Rezervováno pro další využití
61 - 72	NEDEFINOVÁN	Tyto sloupce jsou k dispozici pro libovolné další informace

4.3 Inicializační štítek

Je volitelný a používá se v případě, že tabulka obsahuje inicializační sekci. Lze ji použít libovolný počet. Mají následující tvar:

Sloupec 8	TYP	Mustí obsahovat 'I'
9 - 72	INICIALIZAČNÍ PŘÍKAZY	Obsahuje libovolné inicializační příkazy

4.4 Podmínkový štítek

Je volitelný a používá se pouze v případě, že tabulka obsahuje sekci podmínek. Počet podmínkových štítků v tabulce není omezen. Každý štítek představuje jeden řádek sekce podmínek. Obsahuje tyto informace:

Sloupec	B	TYP	Musí být 'C'
	9 - XX	FORMULACE PODMÍNKY	Obsahuje příkaz formulace podmínky
	YY - 72	VOLBA	Obsahuje volby podmínky, sepsané ve sloupcích 9 - XX. Může obsahovat pouze znaky N, Y, - a mezera

4.5 Činnostní štítek

Tento štítek je volitelný a užívá se pouze v případě existence sekce činností. Počet těchto štítků není omezen. Každý z nich obsahuje následující informace:

Sloupec	B	TYP	Musí obsahovat 'A'
	9 - XX	FORMULACE ČINNOSTÍ	Tyto sloupce obsahují příkaz formulace činností
	YY - 72	VOLBA ČINNOSTÍ	Obsahuje volby činností, vztahující se k formulaci činností ve sloupcích 9 - XX. Může obsahovat pouze znaky abecedy, čísla 1 - 9 nebo mezery.

4.6 Pokračovací štítek

Tento štítek je rovněž volitelný a používá se pouze v případě, že se příkaz, vyskytující se v části formulace, nevyskytne na 1 řádek. Jeho struktura je

Sloupec	B	TYP	Musí být 'S'
---------	---	-----	--------------

9 - IX POKRAČOVÁNÍ
FORMULACE

Obsahuje pokračování for-
mulace činnosti nebo pod-
mínky

YY - 72 PRAZDNE

4.7 Poznámkový štítek

Je volitelný a slouží ke vkládání libovolného kome-
ntáře za účelem zvýšení přehlednosti textu, k dokumentač-
ním a jiným účelům. Tento štítek lze použít na libovolném
místě mezi štítkem řízení obsahu a štítkem konce souboru.
Jeho struktura je:

Sloupec 8	TYP	Musí obsahovat '*'
9 - 72	KOMENTÁŘ	Libovolné kombinace znaků

4.8 Štítek konce tabulky

Tento štítek je povinný, označuje konec tabulky.

Vypadá takto:

Sloupek 8	TYP	Musí obsahovat 'E'
IX - 72	FREKVENCE PRAVIDEL	Toto pole je volitelné a může být prázdné. Je-li vyplněno, lze ho využít k zadání relativních frek- vencí jednotlivých pravi- del. Lze použít libovol- né a číselic 0 - 9, které je nutno psát do téhož sloupce, jako příslušná pravidla

4.9 Štítek konce souboru

Je volitelný; je-li použit, musí být uveden jako
poslední. Má tuto strukturu:

Sloupek 8	TYP	Musí být 'F'
-----------	-----	--------------

5. PŘEDNOSTI A NEDOSTATKY NORMY CSA

Snad za největší klad dané normy je třeba považovat fakt, že vůbec existuje. Její existence představuje první krok ke standardizaci metody. K dalším z jejích předností patří to, že není nikterak závislá na hostujícím jazyku. Lze jí použít jak pro libovolný přirozený jazyk, tak pro vyšší programovací jazyky. Za další výhodu normy lze považovat její ucelenou koncepci a formu.

Jednou z hlavních nevýhod této normy je, jak již bylo řečeno dříve, že se týká pouze tabulek s jednoduchou volbou. Další její nevýhodou podle mého názoru je neprávě nejšťastnější způsob zadávání frekvencí pravidel až na konci tabulky, v závislosti na poloze pravidel. Tato nevýhoda nemusí být příliš zřejmá z hlediska uživatele, z hlediska tvorby překladače RT však ano. Kromě toho by snad bylo možné věnovat větší pozornost způsobu volby překladu RT. Za další nevýhodu normy lze považovat její orientaci na černé štítky. V současnosti je však tato nevýhoda vzhledem k hardware počítačů celkem nepodstatná. Kromě toho lze považovat za diskutabilní opodstatněnost existence štítku konce souboru.

Přes výše zmíněné nedostatky představuje tato norma vhodnou pomůcku pro tvorbu RT, do značné míry nezávislou na softwarově-hardwarem prostředí.

Příklad

Uvedme nyní pro ilustraci příklad rozhodovací tabulky, psané podle výše uvedené normy, v níž hostujícím jazykem je Cobol.

H	COBOL EXAMPLE			50
*				
I	MOVE ZERO TO QUOTIENT.			
*				
C	IF DIVIDEND POSITIVE	Y	N	
C	IF DIVIDEND NEGATIV		Y	N
*	ADD DIVISOR TO DIVIDEND		X	
S	GIVING REMAINDER.			
A	SUBTRACT 1 FROM QUOTIENT		X	
A	SUBTRACT DIVISOR FROM	X		
S	DIVIDEND.			
A	ADD 1 TO QUOTIENT.	X		
A	MOVE ZERO TO REMAINDER.			X
A	GO AGAIN.	X		
S	MOVE ANSWER TO LINE - A .			
S	MOVE QUOTIENT TO LINE - B.			
S	MOVE REMAINDER TO LINE - C.			
S	MOVE REMAINDER TO LINE - D.			
S	WRITE PRINTFILE FROM LINE.			
A	GO TO NEXTABLE.	X	X	
*				
E				

Poznámka:

Příkaz GO AGAIN způsobí, že se předá řízení za inicializační část, tj. na test první podmínky.

6. TABULKY S ROZŠÍŘENOU VOĽBOU

Chceme-li aplikovat předchozí normu na tabulky s rozšířenou volbou, narazíme především na problém vhodného oddělení jednotlivých pravidel. Existují v podstatě tři možnosti, jak toho lze docílit:

- 1) pomocí vhodného oddělovače
- 2) pomocí zvláštního štítku, kde by byly zadány počáteční sloupce jednotlivých pravidel
- 3) zadáním počtu sloupců, potřebných pro jedno pravidlo (např. ve štítku záhlaví tabulky).

Vzhledem k černoštitkové orientaci této metody a z důvodů flexibility se snad jeví jako nejvhodnější druhý způsob. Tento jakýsi "ustavovací" štítek by mohl přitom udávat buď relativní frekvence jednotlivých pravidel (štítek F - frequency) nebo pořadí, v němž mají být jednotlivá pravidla zpracovávána (štítek O - order). První symboly jednotlivých frekvencí či pořadí by přitom určovaly počáteční sloupce jednotlivých pravidel. Tento štítek by byl volitelný; v případě, že by nebyl uveden, předpokládá by se typ O, pořadí provádění pravidel zleva doprava. Struktura těchto štítků by byla následující:

6.1 Štítek frekvence pravidel

Sloupec 8	TYP	Obsahuje 'P'
9 - XX	PRÁZDNÝ	
YY - 72	FREKVENCE PRAVIDEL	Obsahuje relativní frekvence jednotlivých pravidel v procentech. Lze použít pouze numerické údaje. Jednotlivé frekvence je možno oddělit sezerami.

6.2 Štítek pořadí pravidel

Sloupec 8	TYP	Musí být '0'
9 - XI	PRÁZDNÝ	
YY - 72	POŘADÍ PRAVIDEL	Obsahuje pořadí, v němž mají být jednotlivá pravidla prováděna. Lze použít číslíce i písmena.

Oba předchozí štítky by měly tedy význam jednak pro vertikální nastavení jednotlivých pravidel, jednak pro optimalizaci překladu. Je sňejné, že štítek P by mohl úspěšně plnit svou roli pouze za předpokladu, že překladač RP bude za tímto účelem schopen poskytovat určitou statistiku. Tato otázka by si jistě zasloužila větší pozornost, měl by jí však být věnován jiný referát tohoto semináře.

Další problém, plynoucí z možnosti rozšířené volby, se týká štítků podmínek a činností. Především je nutno specifikovat speciální znak v části formulace, který má být nahrazen. Vhodným znakem pro tento účel by mohl být znak '?'. Větším problémem je nejednoznačnost, k níž mohlo dojít při současném použití rozšířené volby činností a variabilního pořadí činností. Zdá se, že nejpřirozenějším řešením je svedení štítku dosazení hodnot. Jeho struktura by mohla být následující:

6.3 Štítek dosazení hodnot

Sloupec 8	TYP	Obsahuje 'V' (value)
9 - XI	PRÁZDNÝ	
YY - 72	HODNOTY	Obsahuje hodnoty, které mají být dosazeny do činností nebo podmínek na místo znaku '?'

7. VOLBA METODY PŘEKladu

Další otázka, již by měla být věnována určitá pozornost, je volba metody překladu RT. Domnívám se, že každý solidní překladač RT by měl být uživateli poskytovat možnost volit mezi více metodami překladu. Tato volba metody by mohla být buď explicitní, tj. uživatel sám by zadal určitou metodu, které se má použít, nebo implicitní, kdy by volbu metody provedl překladač na základě jistých kritériálních údajů uživatele. Uvažme nyní, o jaké údaje by se mělo asi jednat. Každé metodě překladu RT bychom mohli (uvnitř překladače) přiřadit nějaký kritériální vektor

$$\bar{k}_i = (k_{i1}, \dots, k_{in}) \quad , \quad 1 \leq i \leq n,$$

kde n je počet kritérií, m počet metod překladu.

Jako kritéria lze zvolit např. požadavky na

- dobu překladu RT
- rozsah generovaného kódu
- efektivitu - " -
- přehlednost - " -

apod.

Uživatel by potom provedl volbu metody překladu zadáním vektoru vah:

$$\bar{v} = (v_1, \dots, v_n),$$

kde v_j vyjadřuje míru závažnosti určitého kritéria. Překladač potom zvolí metodu překladu jako

$$\max_i \bar{v} \cdot \bar{k}_i = \max_i \sum_{j=1}^n v_j k_{ij}$$

Pro výše popsaný postup by bylo možno využít štítku řízení ohodu, jehož obsah by pak byl následující

Sloupec 8	TYP	'R'
9	PRÁZDNÝ	
10	EXPLICITNÍ VOLBA METODY PŘEKladu	Obsahuje explicitní volbu metody překladu. Je-li prázdny nebo obsahuje-li

nalu, použije se impli-
cítní volby

11 PRÁZDNÝ

12 - 72 IMPLICITNÍ VOLBA Obsahuje vektor vah, od-
METODY PŘEKLADU dělených mezeraní

V případě, že tento štítek není uveden, nebo že nebyla me-
toda zadána ani prvním ani druhým způsobem, nebo v přípa-
dě, že by se ukázalo několik metod stejně účinných, bylo
by dále možno přihlídnout k vlastním složení tabulky,
nebo by překladač automaticky mohl preferovat některou
z metod.

Příklad tabulky s rozšířenou volbou

R	2	50		
H	EXT-TABLE			
*	TABULKA S ROZŠIŘENOU VOLBOU V JAZYCE			
*	COBOL			
F		20	35	40
C	IF RTYP1	Y	Y	N
C	IF RTYP2	N	N	Y
C	IF E1-ZAV = ?	55	45	-
C	IF R - CITAC ? LNCT	<	=	>
C	IF R-1 ? R-2	-<	<	-<
C	IF X1 ? '7'	=	-	-
C	IF X2 ? '7'	->	>	>
C	IF ?	CH-1	-	CH-1
C	IF ?	-	- CH-2	CH-2
C	IF ?	CH-3	- CH-3	-
C	IF ?	- CH-4	- CH-4	- CH-4
C	IF C-BLOKU 1000	Y	N	-
*				
A	PERFORM X THRU Y.			2
A	PERFORM ?	1	1	- 1
V		A-100	B-100	C-100
A	MOVE A TO B.			1 1
A	GO TO F-CHYBA.			1
*	- ZNAMENA NEGACI			
B				

8. PORTABILNÍ PŘEKLADAČ RT

V této části svého příspěvku bych se chtěl především obrátit k těm účastníkům semináře, kteří se ocitli v podobném postavení vůči metodě RT jako naše výpočtové středisko, t.j. v pozici uživatelů, kteří tuto metodu ovládají a rádi by jí používali, avšak v programovém vybavení jejich počítače překladač RT chybí. Rovněž naděje, že by se nám podařilo nějaký přenést z jiného počítače, je mizivá. Budeme tedy bohužel touto realitou pravděpodobně dále čekat nějaký vytvořit. Aby před počasný problém nemuseli být v budoucnosti postaveni všichni naši počítačův uživatelé, bylo by vhodné vytvořit překladač, snadno přenositelný na různé typy počítačů. Aby mohlo být pojetím portability snadno vyhoveno, měl by tento překladač splňovat několik podmínek:

1, Měl by být psán metodou top - down, tak, aby jednotlivé moduly bylo možno v případě potřeby snadno upravit, případně nahradit jinými.

2, Měl by být napsán pokud možno v maximálním rozsahu ve vyšším programovacím jazyce.

3, Měl by být v maximální možné míře nezávislý na operačním systému a hardware počítače.

Z dostupných překladačů této podmínkám nejlépe vyhovuje překladač RT z VŠB Kocice, který je psán v jazyce PL 1 a hostujícího jazyka je rovněž PL 1. Bylo by vhodné vytvořit podobné překladače i pro další jazyk, např. pro Cobol.

Jinou otázkou jistě je efektivita takového překladače. Nejmenší důležitý bude asi oo do doby, potřebné na překlad RT. Tento faktor by však nemusel být rozhodujícím. Další ukazatele, jako rozsah, efektivita a přehlednost generovaného kódu by se použitím vyššího programovacího jazyka nemusely narkantně zhoršit. Domnívám se však,

že otázka efektivity překladače není ve většině případů rozhodující, alespoň ne v počáteční fázi zavádění metody. Když se potom časem ukáže naléhavá potřeba jejího zvýšení, je možno kritické části překladače přepsat do jiného jazyka, třeba do Assembleru.

9. ZÁVĚR

Překladač, tvořený na základě předchozích principů, by měl i řadu dalších výhod. Byl by značně nezávislý na hostujícím jazyce (vyjma závěrečné části generování výstupního kódu a otázky knihoven); mohl by být proto snadno zařazen do standardního software většiny počítačů, čímž by byl problém standardizace metody jednou provždy vyřešen.

Vytvoření takového překladače by se mělo stát cílem všech zastánců metody RT, neboť tudy vede cesta k podstatnému rozšíření počtu uživatelů metody.

Dopínám se totiž stále, že existence inteligentního překladače RT, zařazeného vhodným způsobem do software počítače, nejlépe přímo do kompilátoru hostujícího jazyka, hraje stěžejní roli v otázce využívání metody. Chtěl bych proto vyzvat všechny zde přítomné, kteří projeví zájem, ke spolupráci v této oblasti.

Literatura

- (1) V. Chvalovský: Rozhodovací tabulky, SFTL, 1974
- (2) E. Humby: Programy na základe rozhodovacích tabuliek, Alfa, 1976
- (3) J. Kešner: Rozhodovací tabulky a jejich využití v budování ASŘ, Výběr, 1976/3-5
- (4) Decision Tables, CSA Standard Z 243.1 - 1970
- (5) P. Jiříček: Rozhodovací tabulky, OKR, Standard č. 12
- (6) S. Pollack, H. T. Hicks Jr., W. J. Harrison: Tablice decyzyjne, PWN, Warszawa 1975