

Ing. Branislav LACKO

TOS n.p. Kufín

POUŽÍVÁNÍ ROZHODOVACÍCH TABULEK

Úvod

Rozhodovací tabulky nejsou u nás zatím příliš rozšířeny. Je to zapříčiněno zejména tím, že poměrně málo počítačů II.generace, které jsou v československých výpočtových střediscích nejvíce rozšířeny, umožňuje přímé zpracování rozhodovacích tabulek prostřednictvím kompilátorů. U počítačů III.generace se již stalo pravidlem, že jsou v jejich programovém vybavení implementovány procesory a kompilátory pro přímé zpracování rozhodovacích tabulek, čímž se vytvářejí základní předpoklady pro rozšíření této progresivní metody zápisu složitých logických rozhodovacích sítí.

Tento příspěvek chce přehledně shrnout základní poznatky o rozhodovacích tabulkách a vyzdvihnout jejich přednosti.

Protože dosud ještě nebyla sjednocena a stanovena závazná terminologie /připravuje se ve VÚMS/, používám výrazů, které vznikly překladem anglických termínů.

Forma zápisu rozhodovací tabulky

Základní forma zápisu rozhodovací tabulky vychází z dvojrozměrného tabulkového uspořádání vztahů, které algoritimizují rozhodovací proces, t.j. zachycuje pravidla pro provede-

ní určitých akcí, závislých na splnění předepsaných podmínek.

Rozhodovací tabulka se skládá ze čtyř kvadrantů viz obr.1.

I.kvadrant obsahuje seznam podmínek, na základě kterých se provádí rozhodování.

II.kvadrant zahrnuje seznam pravidel, vytvořených z podmínek prvního kvadrantu.

III.kvadrant předepisuje seznam akcí, které jsou výslednou činností rozhodovacího procesu.

IV.kvadrant je řešen jako řídicí část rozhodovací tabulky, určující, které akce se mají provést, jsou-li splněny podmínky určitého pravidla.

Všechny čtyři kvadranty jsou rozděleny na řádky.

Druhý a čtvrtý kvadrant je navíc rozdělen do sloupců /viz obr.2/. Každá tabulka je označena svým názvem.

Binární rozhodovací tabulka

Binární rozhodovací tabulka /viz obr.3/ předpokládá, že všechny podmínky /nezávisle proměnné rozhodovacího procesu/ jsou formulovány ve výrokovém tvaru. Do II.kvadrantu se vždy pro příslušné pravidlo zapíše pravdivostní hodnoty, které vyjadřují, zda podmínka má nebo nemá být splněna. Vyžadujeme-li splnění podmínky zapisujeme znak "y", vyžadujeme-li nesplnění podmínky zapisujeme znak "n". Provedení předepsané akce označujeme znakem "x". Vynechání akce zaznamenáme pomlčkou. Při vyhodnocování binární tabulky můžeme postupovat dvojím způsobem:

a/Metoda testování podle pravidel.

Postupujeme tak, že nejprve vyhodnotíme všechny podmínky. Postupně pak testujeme pravidlo za pravidlem, až se najde právě jedno pravidlo, které souhlasí s vyhodnocenými podmínkami. U nalezeného pravidla akce, označené hvězdičkou, v pořadí shora dolů, se provedou.

b/Metoda testování podle podmínek.

Při této metodě vyhodnotíme první podmínku. Je-li její hodnota např. "ne", nebudeme již vyhodnocovat všechna pra-

vidla, která mají předepsanou pravdivostní hodnotu "ano", protože jejich akce již nemohou být prováděny. Pro zbývající pravidla vyhodnotíme další podmínku. Je-li její hodnota např. "ano", pak z dalšího zpracování vypadávají pravidla, která mají pro tuto podmínku předepsanou hodnotu "ne" atd.

Jak bylo uvedeno, stavu podmínky "ano" odpovídá zápis "y" ve sloupci pravidla, stavu "ne" znak "n". Aby se provedly akce určitého pravidla, všechny podmínky^{prv} musí být ve stavu, který předepisuje pravdivostní hodnota podmínky v pravidle.

Počet pravidel binární tabulky je dán vztahem $n=2^i$, kde "n" je počet pravidel a "i" počet podmínek.

Rozšířená rozhodovací tabulka

Postupným rozšiřováním možností v zápisu rozhodovacích tabulek, byla ortodoxní binární forma tabulky rozšířena rozličnými formami:

a/ Podmínka závislá na pravidlu není vyjádřena výrokem, ale může se měnit v každém pravidlu. Např. na obr.4 jsou to podmínky 1. a 2.

b/ Akce závislá na pravidlu se liší pravidlo od pravidla ve svém provedení. Např. zvolené proměnné se přiřadí v každém pravidlu jiná hodnota, viz 1. akce na obr.4.

c/ Pravidlo JINAK, umožňuje zkrátit zápis tabulky pro všechny ostatní případy, které již nemusí být jednotlivě vyjmenovávány.

d/ Ignorování podmínky znamená, že při vyhodnocování pravidla nebereme tuto podmínku v úvahu. Ignorování podmínky označujeme psalčkou.

e/ Řetězení tabulek provedeme tak, že připustíme akce, které mohou zařadit provedení jiné rozhodovací tabulky, a které mohou předávat řízení výpočtu na další rozhodovací tabulku. Bývá obvykle povolen případ rekurzivního volání tabulky /přímé i nepřímé/ za předpokladu, že provádění akcí ovlivnilo stav podmínek /jinak by došlo k zacyklení výpočtu/. Řetězením tabulek se vytváří z izolovaných tabulek systém rozhodovacích tabulek viz obr. 5.

f/ Předepsané pořadí akcí dává možnost stanovit jinou posloupnost provedení akcí, než je přirozené pořadí shora dolů podle zápisu ve III.kvadrantu/viz obr.6/.

g/ Disjunktivní pravidlo je splněno již tehdy, když jedna z podmínek má skutečnou hodnotu shodnou s předepsanou; normálně se všechna pravidla chápou jako konjunktivní /všechny podmínky v pravidlu musí mít shodné hodnoty s předepsanými pravdivostními hodnotami/.

Rozšířené rozhodovací tabulka dovoluje zkrácený zápis ve srovnání s binární tabulkou, do které bychom museli zepsat formou výroků všechna pravidla. Rozšířené rozhodovací tabulky jsou při praktickém používání výhodnější než binární rozhodovací tabulky.

Používání rozhodovacích tabulek v programech

Rozhodovací tabulka dovoluje, aby zadavatel, analytik i programátor použili stejného prostředku pro formulaci zadání, postupu řešení i zápisu programu. Ine ještě převažuje praxe, kdy uživatel formuluje zadání úlohy volným popisným textem, analytik nakreslí vývojový diagram algoritmu a programátor napíše program v určitém programovacím jazyku. Praxe dokazuje, že celý vyjmenovaný proces se odebejde bez chyb, jejichž příčinou je nesprávně provedená transformace z jednoho druhu zápisu do druhého, zvláště v případě složitého popisu logických rozhodovacích sítí.

Zadavatel a analytik mohou použít formy rozhodovací tabulky okamžitě, jakmile získají znalost sestavování rozhodovacích tabulek. Jestliže má metoda rozhodovacích tabulek použít i programátor, musí mít možnost zapsat rozhodovací tabulku přímo ve svém programu.

Začlenění rozhodovacích tabulek do programového vybavení počítače lze provést dvěma způsoby:

a/vytvořením samostatného programovacího jazyka rozhodovacích tabulek;

b/obohacením běžného programovacího jazyka / ALGOL, FORTRAN, COBOL/ o část jazyka, která umožní zapsat rozhodovací tabulku jako součást programu.

Je-li zvolen způsob a/, je nutno navrhnout zvláštní

příkazy vstupu a výstupu, zvláštní příkazovací příkazy a pod. Zdá se, že toto řešení je příliš těžkopádné a pro většinu druhů programů i značně neefektivní. Způsob b/ nemá uvedené nevýhody způsobu a/, a navíc je zde výhoda, že nemusíme programátory učit novému programovacímu jazyku. Pouze provedeme instruktáž o rozšířených možnostech základního programovacího jazyka.

Realizovat řešení b/ znamená:

a/ provést úpravu kompilátoru, aby překládal i zápisy rozhodovacích tabulek,

nebo

b/ sestavit speciální předkompilátor.

Předkompilátor je program, který přepracuje v rámci samostatného výpočtového chodu kódovaný zápis rozhodovací tabulky do příkazů základního programovacího jazyka. Ostatní části programu /t.j. příkazy základního programovacího jazyka/ pouze okopíruje do výsledného textu. Výstupem z předkompilace je řetězec znaků, které tvoří text programu výhradně v příkazech základního programovacího jazyka a můžeme ho zkompileovat standardním kompilátorem příslušného jazyka. Poznamenejme, že předkompilátor může překládat rozhodovací tabulku i do nižšího programovacího jazyka než je základní programovací jazyk /např. jazyk symbolických adres/, jestliže použitý kompilátor základního programovacího jazyka dovolu-
je výskyt takovýchto částí.

První řešení je vysoce efektivní /krátká doba překladu, lepší využití paměti/, ale vyžaduje úpravu kompilátoru, což mohou prakticky provést jen jeho tvůrci u dodavatele programového vybavení náročným zásahem. Druhý způsob vyžaduje větší spotřebu strojového času, ale jeho výhoda spočívá v možnosti, vytvořit předkompilátor ve výpočtovém středisku jako speciální systémový program. Vytvoření předkompilátoru bývá někdy ulehčeno existencí různých makrogenerátorů v programovém vybavení počítačů III.generace. Na předkompilátor je nutno klást požadavek, aby pracoval rychle a splňoval podmínky úplné kompatibility na použitý operační systém /např. aby předkompilované texty byly ukládány ihned do knihovny spd./.

Pro automatický překlad rozhodovacích tabulek byly vypracovány a aplikovány dvě hlavní metody překladu, které se mohou použít obecně bez ohledu na programovací jazyk, do kterého jsou výsledné příkazy překládány.

Metoda postupného rozkladu, která zaručuje dosažení každého pravidla tabulky v minimálním čase. Její nevýhodou je, že se při vzrůstajícím počtu podmínek a pravidel zvyšuje počet opakovaně testovaných podmínek.

Metoda binárních matic, která vychází z techniky binárních vztahů, sestavených pro každou tabulku, zaručuje generování minimálního kódu. Metoda binárních matic je zvláště vhodná pro rozšířené rozhodovací tabulky.

Od svého vzniku byly obě metody zdokonalovány, aby byl překlad i generovaný kód co nejvíce optimalizován. Rozsah příspěvku nedovoluje podrobný popis obou metod.

Při začlenění rozhodovacích tabulek do programovacího jazyka je nutno vyřešit dva závažné problémy:

a/ Způsob zápisu rozhodovací tabulky do programu.

Programovací jazyky jsou lineární posloupností alfanumerických a speciálních znaků. Tomuto požadavku je nutno přizpůsobit i dvojrozměrný zápis rozhodovací tabulky. Jazyky, které pro zápis vstupního zdrojového textu používají děrných štítků, vycházejí z t.zv. pevné formy zápisu rozhodovací tabulky. Jednotlivé řádky se děrují do samostatných štítků. Sloupce na štítcích jsou rozděleny tak, aby vyhovovaly svislému dělení rozhodovací tabulky ve II. a III. kvadrantu. Použije-li se na vstupu do kompilátoru děrná páska, syntaxe jazyka musí dovolit pomocí typografických znaků /tabulátor, mezery/ uspořádat text do formy tabulky, viz obr. 7, který ukazuje tabulku z obr. 4, zapsanou v jazyku BETAB/viz níže/.

b/ Připojení zápisu rozhodovací tabulky do syntaxe a sémantiky jazyka. Objekty, které se vyskytnou ve všech čtyřech kvadrantech, musí být logickou podmnožinou syntaktických objektů zvoleného programovacího jazyka. Obecně lze stanovit tyto zásady:

- jako podmínky může být použit libovolný výrokový výraz /relace, logický výraz/

- jako akce může být použit libovolný výkonný příkaz programovacího jazyka /přifazovací příkazy, příkazy cyklu, příkazy vstupu a výstupu/,

- rozhodovací tabulka může tvořit výkonný příkaz programovacího jazyka.

Uvažujeme-li otevřené tabulky, pak musíme připustit, aby se určitá část podmínky nebo akce dala zapsat i do sloupců II. a IV. kvadrantu. Tím získáme možnost vytvářet podmínky a akce závislé na pravidlech.

Někdy působí potíže narůstající délka složitých podmínek a akcí. Zápis velmi složitých výrazů se nevejde do vyhrazených částí I. a III. kvadrantu. Zde je možno s výhodou využít skutečnosti, že téměř každý programovací jazyk dovoluje vytvářet funkční procedury a podprogramy. Složitě výrazy deklarujeme podle potřeby jako funkční procedury typu boolean, integer např. v jazyku ALGOL, FUNCTION, SUBROUTINE v jazyku FORTRAN apd. Na příslušných místech rozhodovací tabulky uvádíme pak jen identifikátory.

V programovém vybavení počítačů DATASAB byl rozšířen jazyk ALGOL 60 o možnost přímého zápisu rozhodovací tabulky v textu programu. Byla provedena potřebná úprava kompilátoru, která dovoluje rozšíření jazyka ALGOL 60 podle následující syntaktické definice:

```
< složený příkaz bez návěští > ::=  
    begin < konec složeného příkazu > |  
    begin < deklarace rozhodovací tabulky >
```

```
< deklarace rozhodovací tabulky > ::=  
    decisiontable < zápis tabulky > end
```

Jazyk, pomocí kterého se provede zápis rozhodovací tabulky, se nazývá BETAB ze švédského BEsluts TABeller. Následující příklad ukazuje zápis rozhodovací tabulky v jazyku BETAB /srovnej s obr. 4/, kdy rozhodovací tabulka tvoří operační část funkční procedury.

```

real procedure KOEFICIENT(a,b,c);
integer a,b,c;
begin
decisiontable KOEF;
if c = :      , 1, 1, 2, 2, 3, 3;
if a : b      , >, <, >, <, =, ≠;
if EXTRA     , Y, N, Y, N, Y, N;
( KOEFICIENT := , a-b, b-a, a, b, a*b, a/b, 0);
( VYPISCH     , -, -, -, -, -, -, #);
( goto OPAKUJ , #, #, #, #, #, #, -);
end;

```

Obr. 7

Poznámky:

1. Zápisová část akcí musí vždy obsahovat o jeden sloupec víc, který se chápe jako seznam akcí pro pravidlo JINAK.
2. Identifikátor EXTRA je globální booleovská proměnná.
3. Identifikátor VYPISCH označuje globální proceduru.
4. Návěští OPAKUJ označuje některý příkaz hlavního bloku programu.

Bezchybnost rozhodovací tabulky

Má-li rozhodovací tabulka splnit správně svoji funkci, musí být bezchybně sestavena. Při automatickém překladu bývá správnost rozhodovací tabulky průběžně kontrolována a případné chyby jsou programátorovi hlášeny výpisem na rychlotiskárně.

Jednoznačnost rozhodovací tabulky nepřipouští, aby nastal případ, kdy je současně splněno více než jedno pravidlo. Pokud by příslušné seznamy prováděných akcí byly totožné, jedná se o nadbytečnost pravidel. Pokud by pravidla vedle k rozdílným činnostem, dochází k rozpornosti v rozhodovací tabulce.

Při sestavování pravidel musí programátor vyloučit případy iracionálních pravidel t.j. takových, která nemohou nikdy nastat. Např. pravidlo, které spojuje dvě podmínky, z nichž jedna vyžaduje, aby určitá proměnná měla hodnotu rovnou 5 a druhá, aby tatáž proměnná měla hodnotu 10. Iracionální pravidla se vyskytují v případech, kdy existují určité vztahy mezi podmínkami t.zv. závislé podmínky /např. dvě podmínky používají v relacích stejné proměnné apd./.

Správně sestavená rozhodovací tabulka musí řešit všechny kombinace podmínek, které mohou nastat. Tato vlastnost se nazývá úplnost rozhodovací tabulky a je jednou ze základních požadavků na rozhodovací tabulku.

K chybám, které mohou nastat při sestavování rozhodovací tabulky, přistupují případy, kdy programátor poruší syntaktická pravidla pro zápis rozhodovací tabulky nebo překročí různá omezení, předepsaná pro používaný překladač /limitovaný počet pravidel, řádků apd./.

Praktické zkušenosti

Používání rozhodovacích tabulek v podniku TOS Kuřim, kde je v provozu počítač DATASAB D21, umožnilo získat řadu zkušeností s touto pokrokovou metodou programování.

Přesvědčili jsme se, že na začátku všeobecného používání rozhodovacích tabulek musí být připraven plán účinného vyškolení všech analytiků, programátorů a pracovníků z odborných útvarů. Školení je nejlépe zaměřit na takovou formu rozhodovacích tabulek, kterou můžeme s minimálními úpravami zapsat do programů a tuto formu vyžadovat.

Všichni, kteří chtějí používat v podniku rozhodovací tabulky, musí mít možnost opřít se o přehledně napsanou příručku, kde najdou specifikaci, popis a příklady s návody pro použití. Technická knihovna v podniku, by měla zajistit dostupnou knižní literaturu z této oblasti do svého knižního fondu, aby mohla posloužit těm, kteří chtějí důkladněji prostudovat metodu rozhodovacích tabulek.

V praxi se ukázalo, že nejtěžší je přimět pracovníky výpočtového střediska a odborných útvarů k sestavení prvních rozhodovacích tabulek. Konkrétním používáním zjistí pracovníci sami výhody zaváděné metody a začnou tabulky hojně používat.

Řada pracovníků potvrdila, že při sestavování rozhodovacích tabulek byla nucena se zamyslet nad variantami kombinací podmínek, na které by se jistě zapomnělo při klasických metodách zápisu algoritmu např. pomocí běžných vývojových diagramů. K tomu napomáhá u rozhodovacích tabulek přehledný způsob zápisu

podmínek a jejich vzájemných kombinací! Ve složitých vývojových diagramech, které by řešily ekvivalentní problémy, je nutno často používat mnoho propojovacích spojek. To činí vývojové diagramy nepřehlednými, takže se zapomene na určitou možnou vzájemnou kombinaci podmínek a k ní náležející seznam akcí.

V programátorské praxi se musíme vypořádávat s požadavky na změny v programech, které jsou nutné z nejrůznějších příčin. Přepřeprogramování rozhodovací tabulky se provede podstatně snadněji než překreslení vývojového diagramu a oprava klasických příkazů v programu. Opravy se v rozhodovací tabulce omezují na přidání nebo vynechání pravidel, sloupcové úpravy, přidání nebo vynechání podmínek případně akcí, řádkové úpravy, nebo na změny ve spojovací části IV. kvadrantu. Vynecháme-li pravidlo, podmínku nebo akci, poskytne rozhodovací tabulka přehled o případných následcích provedené změny. Podobné zásahy bez použití rozhodovacích tabulek jsou doprovázeny častými chybami.

Rozhodovací tabulky přehledněji dokumentují použité algoritmy. Snadněji se zapisují v projektové dokumentaci na psacím stroji, než se kreslí vývojové diagramy.

Dostane-li rozhodovací tabulku jiný pracovník než autor, rychleji se v ní orientuje než v cizím vývojovém diagramu. Má to praktický význam při předávání dokumentace a programů.

V některých výpočtových střediscích se udržuje nesprávný názor, že uplatnění rozhodovacích tabulek se omezuje jen na oblast hromadného zpracování dat. Zapomíná se, že rozhodovací tabulky se mohou uplatnit ve všech případech, kdy je nutno zachytit složitými podmínkami rozvětvený algoritmus. V podnikovém výpočetním středisku TOS Kuřim byly rozhodovací tabulky začleněny do řady programů z oblasti technických výpočtů strojírenské a elektrotechnické problematiky. V literatuře je popsán příklad použití rozhodovacích tabulek při vývoji programového vybavení.

Organizujeme-li využívání rozhodovacích tabulek v rámci početnější skupiny programátorů a analytiků, mohou se vykytovat buď izolované rozhodovací tabulky - vytvářené případ od případu pro řešení specifických problémů - nebo typové rozhodova-

cí tabulky, řešící vybrané, často se vyskytující algoritmy, např. v TOS Kufim jsou to rozhodovací tabulky pro programovou realizaci algoritmu několikaúrovňového součtování v tiskových programech, které prakticky realizují myšlenku standardizace programátorské práce.

Závěr

Metoda zobrazení složitých logických sítí pomocí rozhodovacích tabulek má řadu nesporných výhod, které ji zařazují mezi perspektivní metodické a programovací prostředky počítačů III. generace. V ČSSR se je metoda rozhodovacích tabulek intenzivně rozvíjena pracovištěm VÚMS s cílem, vyřešit přímé zpracování rozhodovacích tabulek na počítačích JSEP. Pro rozvoj rozhodovacích tabulek mají velký význam školení pořádaná Českým komitétem pro vědecké řízení. Vydání souhrnné monografické publikace SNTL "Rozhodovací tabulky", jejíž autorem je ing. V. Chvalovský z ČKD Praha, jistě přispělo k další popularizaci této metody.

Rozsah příspěvku neumožňuje detailně rozebrat všechny problémy rozhodovacích tabulek. Proto v závěru článku je uveden přehled literatury, kde může zájemce vyhledat další podrobnosti. Zaměřil jsem se na československá vydavatelství a časopisy, protože tato literatura je snadno dostupná. V citované literatuře najde čtenář bohaté odkazy na literaturu zahraniční.

Přehled doporučené literatury

Publikace knižní:

Chvalovský V.: Rozhodovací tabulky, SNTL, Praha 1974

Kešner J.: Rozhodovací tabulky, ČKVŘ, Praha 1972

Kešner J.: Úvod do rozhodovacích tabulek, INORGA, Praha 70

Hýbl E.: Logické rozhodovací tabulky, SUPRO, Praha 1970

Humby E.: Programy na základe rozhodovacích tabuliek, ALFA,

Bratislava /připravuje se k vydání/

Gürtler Z.: Rozhodovací tabulky, Příloha k informátoru SUPRO

čís. 6, Praha, 1969

- Vlček J.: Systémové zpracování dat na počítači, SNTL, Praha 1973, str. 74-87
- Nečas J.: Volba a zavádění samočinného počítače, SNTL, Praha 1968, str. 154-157
- Chalandor A.: Praktická systémová analýza, ALFA, Bratislava 1974, str. 117-120
- Habr J. - Vepřek J.: Systémová analýza a syntéza, SNTL, Praha 72, str. 156-161

Články v časopisech:

- Chvalovský V.: Aplikácia rozhodovacích tabuliek, Informačné systémy č.2, 1973, str. 129-139
- Chvalovský V.: Použití programovacích tabulek při vývoji software, INORGA č.1., 1971, str. 9-15
- Chvalovský V.: Automatické vytváření programů z rozhodovacích tabulek, INORGA č.3, 1973, str. 86-91
- Kešner J.: Blokova schemata nebo rozhodovací tabulky?, INORGA č. 1. 1971, str. 4-8
- Kobyda F.: Rozhodovacie tabulky - metoda zobrazenia logických rozhodnutí, Informačné systémy č.1, 1972, str.77-86
- Kratochvíl E.: Co jsou rozhodovací tabulky, NETTO č.6, 1972 str. 14-17
- Kratochvíl E. - Sedlář M.: Rozhodovací tabulky, Aktuality výpočetní techniky VJMS č.4, 1973, str. 73-88
- Metoda rozhodovacích tabulek, Podniková organizace č.7, str. 17-20
- Lacko B.: Rozhodovací tabulky, Mechanizace a automatizace administrativy č. 9, 1970, str. 256-258
- Lacko B.: Používání rozhodovacích tabulek, Informačné systémy č.4 1973, str. 345 - 352
- Sedlář M.: Speciální aplikace rozhodovacích tabulek, Automatizace č.7, 1972, str. 178-179
- Suchomel E.: Rozhodovací tabulky jako racionalizační pomůcky, Podniková organizace č.5, 1972, str. 34-35
- Vlček J.: Rozhodovací tabulky, Ekonomicko-matematický obzor č. 4, 1967, str. 379-384
- Kratochvíl E.: Automatická konverze rozhodovacích tabulek, Mechanizace a aut. administrativy č.11, 1974, 430-433

RT	KOEF	1	2	3	4	5	6	JINAK
1	$c = :$	1	1	2	2	3	3	
2	$a : b$	>	<	>	<	=	≠	
3	VÝJIMKA	Y	N	Y	N	Y	N	
1	KOEFICIENT :=	$a-b$	$b-a$	a	b	$a \times b$	a/b	0
2	VÝPIS CH	-	-	-	-	-	-	*
3	OPAKU VÝP.	*	*	*	*	*	*	-

Obr. 4

Příklad 6		1	2	3	4
1	$S = 1$	Y	N	N	Y
2	$B > 0$	N	Y	N	Y
1	PROCES A	1	1	3	-
2	PROCES B	-	2	2	2
3	PROCES C	2	3	1	1

Obr. 6

I. kvadrant Sexnam podmínek	II. kvadrant Sexnam pravidel
III. kvadrant Sexnam akcí	IV. kvadrant Řidičů část

Obr. 1

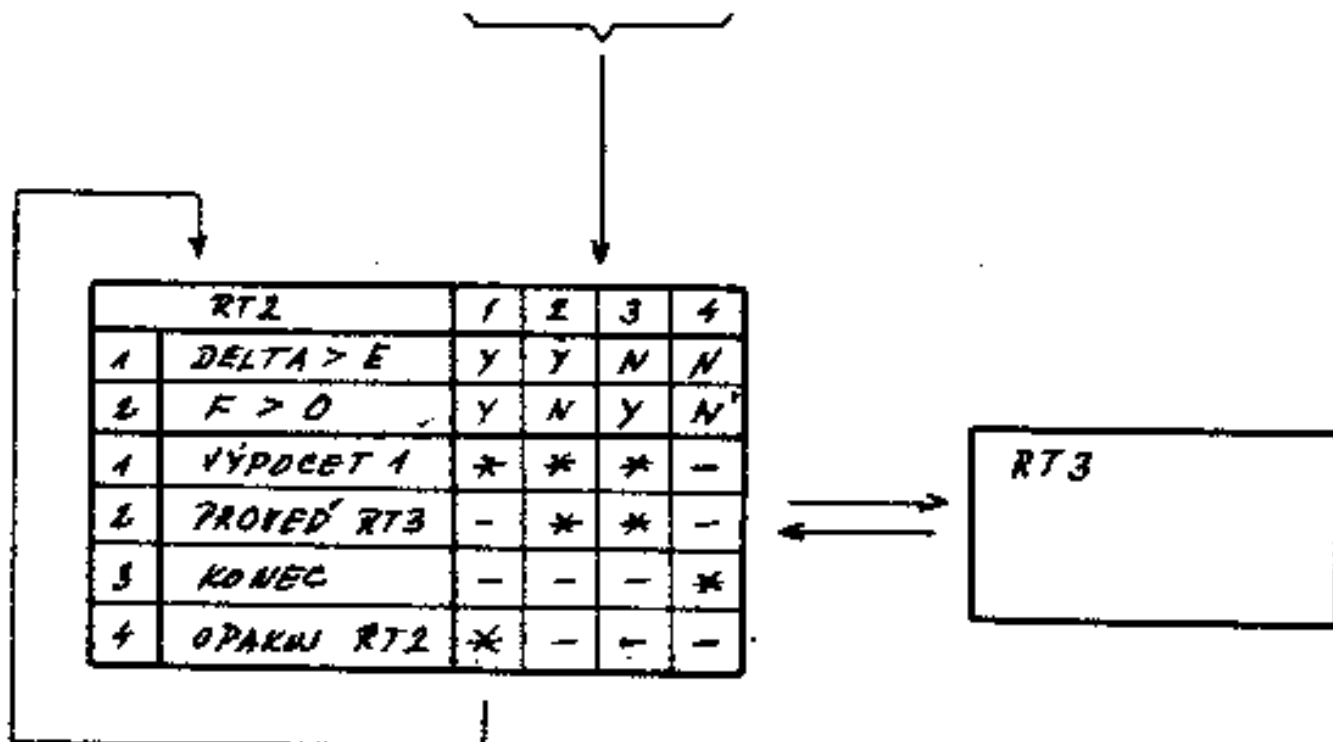
Název RT		Pravidla				
		1	2	3	...	n
Podmínky	1					
	2					
	...					
	i					
Akce	1					
	2					
	...					
	j					
	m					

Obr. 2

Příklad binární RT		1	2	3	4
1	Kvalifikační zvýhodnění	N	Y	N	Y
2	Provozní zvýhodnění	N	N	Y	Y
1	Příplatek I.	-	*	-	*
2	Příplatek II.	-	-	*	*
3	Bez příplatku	*	-	-	-

Obr. 3

RT ÚVOD		1	2	3	4
1	A > 0	Y	N	Y	N
2	B > 0	Y	Y	N	N
1	OPEN A	*	-	*	-
2	OPEN B	*	*	-	-
3	Pokračuj RT2	*	*	*	*



obr. 5