3 Minimalizace logických funkcí

3.1 Základní pojmy

Cílem minimalizace je nalézt co nejjednodušší vyjádření zadané logické funkce. Dalším cílem je realizace této funkce pomocí různých typů integrovaných obvodů; podle typu použitých obvodů volíme i způsob vyjádření funkcí (uvidíme dále). Dále bychom měli přesně vymezit pojem „co nejjednodušší".

Kriterium minimality
Vyjádření logické funkce je minimální, jestliže obsahuje :
- minimální počet termů;
- minimální počet nezávisle proměnných v každém termu;
- minimální počet negovaných proměnných.

V tomto smyslu se dále užívají pojmy:
minimální normální disjunktivní forma (dále jen MNDF) - logický součet minimálního počtu minimálních P-termů (tedy termů obsahujících co nejmenší počet proměnných v logickém součinu) a
minimální normální konjunktivní forma (MNKF) - logický součin minimálního počtu minimálních S-termů.
Dále se někdy používá pojmů:
implikant logické funkce f. Je to výraz ve tvaru součinového (P) termu, pro který platí, že implikuje funkci f; tzn. jestliže nabývá hodnoty logické 1, funkce f nabývá také hodnoty logické 1.
přímý implikant funkce f je potom takový implikant funkce f, který po vypuštění libovolné proměnné přestává být implikantem funkce f (přímý implikant tedy obsahuje minimální počet nezávisle proměnných).

Existuje několik možností, jak minimalizovat logickou funkci. Je možné vycházet z algebraického vyjádření logické funkce a aplikováním základních a odvozených zákonů Booleovy algebry postupně daný výraz upravit až na minimální v požadovaném tvaru. Další metody jsou založeny na hledání tzv. sousedních stavů:
sousední stavy jsou takové dvě kombinace hodnot nezávisle proměnných, které se liší hodnotou pouze jedné proměnné. Např. pro funkci $f(a, b, c)$ dva P-termy $a.b.\bar{c}$, $a.b.c$ popisují dva sousední stavy (1 1 0 a 1 1 1); podobně S-termy $a + \bar{b} + c$, $a + \bar{b} + \bar{c}$ popisují dva sousední stavy (0 1 0 a 0 1 1).

V dalším textu jsou popsány způsoby minimalizace logických funkcí pomocí zákonů Booleovy algebry, pomocí mapy a metodou Quine-McCluskey.

3.2 Minimalizace pomocí zákonů Booleovy algebry

Tento přístup vychází z používání základních a odvozených zákonů Booleovy algebry 1 až 12 (odst. 1.1). Dále uvedeme tzv. zobecněné zákony absorbce, negace a rozkladu logické funkce. Nechť $a, b, c, ..., z \in B, F$ je logická funkce.

zobecněné zákony absorpce (logické proměnné) :
$a + F(a, \bar{a}, b, c, ..., z) = a + F(0, 1, b, c, ..., z)$
$\bar{a} + F(a, \bar{a}, b, c, ..., z) = \bar{a} + F(1, 0, b, c, ..., z)$
$a.F(a, \bar{a}, b, c, ..., z) = a.F(1, 0, b, c, ..., z)$
$\bar{a}.F(a, \bar{a}, b, c, ..., z) = \bar{a}.F(0, 1, b, c, ..., z)$

zobecněný zákon negace (logické funkce) :
$\overline{F}(a, b, ..., z, 0, 1, +, .) = F(\bar{a}, \bar{b}, ..., \bar{z}, 1, 0, ., +)$

rozklad logické funkce (*Shannonův expanzní teorém*)
$F(a, b, ..., z) = a.F(1, b, ..., z) + \bar{a}.F(0, b, ..., z)$
$F(a, b, ..., z) = [a + F(0, b, ..., z)].[\bar{a} + F(1, b, ..., z)]$

a b	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0 0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0 1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1 0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1 1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

f_0 - nulová funkce, 0
f_1 - logický součin (AND), $a.b$
f_2 - nonimplikace (inhibice), $\overline{a \Rightarrow b}$ (negace f_{13})
f_3 - kopie vstupní proměnné a
f_4 - nonimplikace zpětná, zpětná inhibice, $\overline{b \Rightarrow a}$ (negace f_{11})
f_5 - kopie vstupní proměnné b
f_6 - nonekvivalence, $\overline{a \Leftrightarrow b}$ nebo $a \oplus b$ (pro dvě proměnné totožná s funkcí XOR)
f_7 - logický součet (OR), $a + b$
f_8 - negovaný logický součet (NOR), $\overline{a + b}$
f_9 - ekvivalence, $a \Leftrightarrow b$ nebo $\overline{a \oplus b}$
f_{10} - negace b, $\bar{b}$
f_{11} - zpětná implikace, $b \Rightarrow a$
f_{12} - negace a, $\bar{a}$
f_{13} - implikace, $a \Rightarrow b$
f_{14} - negovaný logický součin (NAND), $\overline{a.b}$
f_{15} - jedničková funkce, 1

Tab. 3.1 Logické funkce pro $n = 2$

Nyní něco statistiky. Pro n nezávisle proměnných existuje 2^n stavů (různých kombinací hodnot nezávisle proměnných, též řádků v pravdivostní tabulce, též políček v mapě, též různých mintermů i maxtermů) a 2^{2^n} různých logických funkcí. Tedy pro $n = 2$ je to 16 funkcí (viz Tab. 3.1, kde každá z těchto funkcí má svůj vlastní název a označení), pro $n = 3$ už je to 256 funkcí, atd. Přitom jsme se v minulé kapitole přesvědčili, že každou funkci je možné vyjádřit v ÚNDF nebo v ÚNKF, tedy pouze pomocí logického součinu, součtu a negace. Tyto tři funkce (v tomto případě jsou to základní operace) tvoří tzv. *úplný soubor logických funkcí* (takových

skupin funkcí je více). Soubor minimálního počtu takovýchto funkcí se potom nazývá *minimální úplný soubor logických funkcí* .
Je možné dokázat, [3], že existuje jediná funkce, která tvoří minimální úplný soubor logických funkcí. Je to funkce NAND nebo NOR. Tato vlastnost je zvláště výhodná při realizaci logických funkcí pomocí integrovaných obvodů (hradel).

3.2.1 Cvičení

1. Nalezněte minimální normální disjunktivní formu funkce f zadanou Booleovým výrazem
$f = \bar{a}d + \bar{b}cd + a\bar{b}(c + d) + \bar{b}\bar{c}\bar{d}$

Řešení:

Nejprve podle distributivního zákona upravíme závorku:
$f = \bar{a}d + \bar{b}cd + a\bar{b}c + a\bar{b}d + \bar{b}\bar{c}\bar{d}$
dále podle zákona absorbce negace upravíme 1. a 4. člen :
$\bar{a}d + a\bar{b}d = d(\bar{a} + a\bar{b}) = d(\bar{a} + \bar{b})$, tedy
$f = \bar{a}d + \bar{b}cd + a\bar{b}c + \bar{b}d + \bar{b}\bar{c}\bar{d}$
dále opět podle zákona absorpce negace upravíme 4. a 5. člen
$\bar{b}d + \bar{b}\bar{c}\bar{d} = \bar{b}(d + \bar{c}\bar{d}) = \bar{b}(d + \bar{c})$ tedy
$f = \bar{a}d + \bar{b}cd + a\bar{b}c + \bar{b}d + \bar{b}\bar{c}$
dále opět podle zákona absorpce negace 3. a 5. člen:
$f = \bar{a}d + a\bar{b} + \bar{b}cd + \bar{b}d + \bar{b}\bar{c}$
podle zákona absorpce zjednodušíme 3. a 4. člen na $\bar{b}d$:
$f = \bar{a}d + a\bar{b} + \bar{b}d + \bar{b}\bar{c}$
nyní podle zákona consensus vypustíme term $\bar{b}d$ a dostaneme výsledný tvar v MNDF :
$\bar{a}d + a\bar{b} + \bar{b}\bar{c}$

Poznámka. Z de Morganových zákonů (viz odst. 1.1) a ze zobecněného zákona negace (odst. 3.2) vyplývá, že operace negace má vlastnost závorky (co se týče priority).

2. Nalezněte MNDF následujících logických funkcí. Použijte zákony Booleovy algebry.
a) $a(\bar{b} + \bar{c}) + a\bar{b}d + abcd$
b) $\bar{c}\bar{b}a + \bar{c}b\bar{a} + \bar{c}ba + cba$
c) $(a + b + c)[a(b + d) + \bar{b}c(ad + \bar{a}b + \bar{a}\bar{b}d + e(b + cd)]$
d) $(a + b + c)(b + c + d)$
e) $(a + b)[d(d + e) + \bar{d}(a + c) + (g + h)\bar{h}a]$
f) $a + \bar{a}\bar{b} + \bar{b}(c + de)$
g) $\bar{a}d + \bar{b}cd + a\bar{b}(c + d) + \bar{b}c\bar{d}$
e) $(a + b + \bar{c})(a + \bar{a}b + \bar{a}d)$

f) $\overline{a + \overline{b(c + a\bar{d}) + \overline{a\bar{b}c + ab\bar{c}}}}[a + b\bar{c} + ab(a + \bar{a}c)]\overline{(\bar{a} + b)}$
g) $\bar{c}\bar{d} + b[\bar{a} + \bar{d}(b + abd)(\bar{a} + c)] + a$
h) $(\overline{a + b} + c + d)(a + b)(c + d) + \overline{(c + d)}$

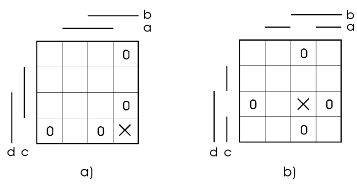
3. Nalezněte MNKF logických funkcí z příkladů 2a) až h). (*Pomůcka:* Jestliže máte k dispozici

funkci v MNDF, můžete použít následující postup: Vytvořte negaci MNDF - získáte negovanou funkci v MNKF, tu upravte pomocí distributivního zákona na MNDF této negované funkce a po opětovné negaci získáte původní funkci v MNKF.)

Poznámka: Ve všech logických funkcích v příkladech 1 a 2 jsme si odpustili zápis znaménka logického součinu (.), podobně jako je to zvykem v algebře reálných čísel. Protože v Booleově algebře mají obě binární operace stejnou prioritu, zachovali jsme se poněkud nekorektně. Tímto způsobem zápisu logických funkcí jsme si také ztížili převody do konjunktivních forem.

3.3 Minimalizace pomocí map

Minimalizace logické funkce je založena na hledání maximálních skupin sousedních stavů (viz výše). Karnaughova mapa je vytvořena tak, aby sousední stavy byly topologicky vedle sebe. Sousední stavy jsou ovšem i oba krajní sloupce a oba krajní řádky, a to po čtyřech políčkách. Všechny sousední stavy stavy stavu 10 pro funkci 4 proměnných jsou vyznačeny na obr. 3.1. (Situace je znázorněna v mapě Karnaughově i Svobodově, $\times$ označuje stav 10, 0 stavy sousední.)



Obr. 3.1 Sousední stavy stavu $\bar{a}b\bar{c}d$ $(s = 10)$

3.3.1 Cvičení

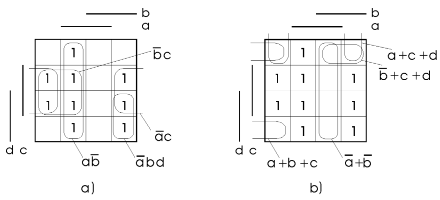
1. Nalezněte MNDF a MNKF logické funkce zadané výčtem jedničkových stavů
$f(d, c, b, a) = \sum(1, 4, 5, 6, 9, 10, 12, 13, 14)$

Řešení

Zadanou funkci zapíšeme do Karnaughovy mapy (viz obr. 3.2). Pro hledání MNDF si nejprve budeme všímat rozmístění logických jedniček. Snažíme se najít co největší skupiny sousedních jedniček. Tyto skupiny musí být vždy dvojice, čtveřice, osmice, atd.; tedy skupiny o velikosti mocniny 2 (čím větší skupina sousedních stavů, tím menší počet nezávisle proměnných potřebujeme k jejímu popisu). V našem případě jsme nalezli tři čtveřice a jednu dvojici sousedních stavů. Nyní musíme vybrat ty skupiny sousedů, které odpovídají kriteriu minimality. Tedy čtveřice pokrývající stavy 4, 5, 12, 13 (term $\bar{b}c$) je nadbytečná, pro vyjádření funkce v MNDF stačí popsat dvě čtveřice a jednu dvojici sousedních stavů. Při popisu skupin sousedních stavů postupujeme tak, že k vyjádření použijeme ty proměnné, které pro celou skupinu nemění svoji hodnotu. Řešení našeho příkladu je podle obr. 3.2a):

$$\begin{aligned} f(d,c,b,a) &= a\bar{b} + \bar{a}c + \bar{a}bd \text{ (MNDF)} \\ &= (\bar{a}+\bar{b})(a+b+c)(a+c+d) \text{ (MNKF) podle obr. 3.2b)} \end{aligned}$$

V případě hledání MNKF jsme měli dvě možnosti výběru dvojice pro pokrytí stavu 2, $(a+c+d)$ nebo $(\bar{b}+c+d)$, vybrali jsme podle 3. bodu kritéria minimality tu, která obsahovala menší počet negací.



Obr. 3.2 Skupiny sousedních stavů a)jedničkových b)nulových

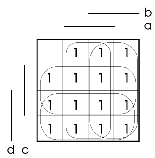
2. Nalezněte MNDF a MNKF logických funkcí f:
a)$f(d,c,b,a) = \sum(0,1,4,6,7,8,9,15)$
b)$f(d,c,b,a) = \sum(1,3,4,5,7,9,11,15)$
c)$f(d,c,b,a) = \sum(1,2,4,5,7,8,10,11,13,14)$
d)$f(d,c,b,a) = \sum(0,1,4,5,8,9,12,13)$
e)$f(e,d,c,b,a) = \sum(0,1,5,8,9,13,15,16,21,22,23,24,30,31)$

3. Nalezněte MNDF a MNKF logických funkcí zadaných booleovým výrazem:
a) $f = a + b + \bar{a}\bar{b}c + d$

Řešení

Nejprve musíme funkci zapsat do mapy. Postupujeme jako při hledání průniků. Funkce je zadaná v disjunktivní formě, tedy budeme zapisovat jedničky. Term a znamená, že zapíšeme do mapy osm jedniček všude tam, kde je proměnná a jedničková (prostřední dva sloupce - viz obr. 3.3). Term b popisuje též osm jedniček, ale čtyři z nich již v mapě jsou, tedy zapíšeme pouze čtyři do sloupce vpravo. Term $\bar{a}\bar{b}c$ popisuje dvojici jedniček v levém sloupci uprostřed, tedy zapíšeme dvě jedničky, a konečně d popisuje osmici v dolních dvou řádcích, z nichž už zapíšeme jen jednu jedničku. Dále minimalizujeme stejně jako v předchozím příkladu, tedy hledaná řešení v našem případě stejná pro obě formy jsou:

$$f = a + b + c + d$$



Obr. 3.3 Hledání MNKF a MNKF funkce f z příkladu 3

b) $f = \bar{a}d + \bar{b}cd + a\bar{b}(c+d) + \bar{b}\bar{c}\bar{d}$
c) $a(\bar{b}+\bar{c}) + a\bar{b}d + abcd$

4. Nalezněte MNDF a MNKF logických funkcí zadaných výčtem jedničkových stavů a tzv. neurčených stavů.

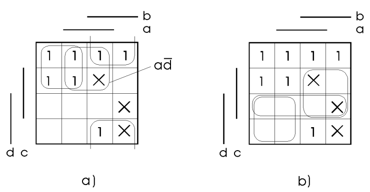
Neurčený stav (angl. don't care) je takový stav logické funkce, pro který daná funkce není definovaná, tedy ta kombinace hodnot nezávisle proměnných, která z nějakých důvodů nemůže nastat. Označujeme $\times$ a s výhodou používáme při minimalizaci, protože neurčené stavy můžeme zahrnout zároveň do skupin jedniček i nul, můžeme zvětšit skupiny sousedních stavů, a tím zmenšit počet proměnných v termech.

a) $f(d,c,b,a) = \sum_{(1)}(0,1,2,3,4,5,11) + \sum_{\times}(7,10,14)$

Řešení

Řešení je znázorněno na obr. 3.4a) a b). Neurčené stavy můžeme zahrnout do skupin sousedních jedniček, nul nebo také nikam. Tedy skupina popsaná termem $a\bar{d}$ je nadbytečná:

$$f(d,c,b,a) = \bar{b}\bar{d} + b\bar{c} = (b+\bar{d})(\bar{b}+\bar{c})$$



Obr. 3.4. Hledání MNDF a MNKF v příkladu 4a)

b) $f(d,c,b,a) = \sum(1,3,4,12,13) + \sum_{\times}(9,10,11)$
c) $f(d,c,b,a) = \sum(0,1,6,7,10) + \sum_{\times}(8,9)$
d) $f(d,c,b,a) = \sum(2,3,6,13) + \sum_{\times}(0,4,12)$
e) $f(e,d,c,b,a) = \sum(0,1,8,9,14,16,17,18,19,24,25,26,30) + \sum_{\times}(10,15,22,31)$

3.4 Minimalizace metodou Quine-McCluskey

Metoda Quine-McCluskey je stejně jako ostatní minimalizační metody založena na hledání maximálních skupin sousedních stavů a minimálního pokrytí těchto stavů. Byla vyvinuta pro použití na číslicových počítačích a hlavně pro minimalizaci logických funkcí většího počtu nezávisle proměnných. Minimalizace v mapě pro 6 a více proměnných je velmi nepřehledná a vlastně již nepoužitelná, zatímco metoda Quine-McCluskey je omezena hlavně rozsahem použitelné paměti počítače. Přesto pro názornost ukážeme způsob použití metody Quine-McCluskey na funkci pouze čtyř proměnných.

3.4.1 Cvičení

1. Najděte MNDF funkce $f(d,c,b,a) = \sum(1,4,7,8,9,10,11,12,14,15)$

Řešení:

Vyjdeme z Tab. 3.2. Nejprve vyjádříme všechny jedničkové stavy binárně (první sloupec tabulky). Dále seřadíme stavy podle počtu jedniček a poznamenáme si, které stavy z prvního sloupce jsme dále použili (hvězdičky ve druhém sloupci), abychom na žádný nezapomněli. Skupiny stavů s jednou, dvěma, třemi a čtyřmi jedničkami jsou ve třetím sloupci tabulky vždy oddělené prázdným řádkem. Je jasné, že sousední stavy můžeme nalézt jen v sousedních skupinách (představme si, o jaké ušetření práce jde např. pro funkci o 15ti proměnných!). Nyní budeme hledat všechny sousední dvojice (viz Tab. 3.2, sloupec 5. a 6.) a opět označíme hvězdičkou, které stavy se nám podařilo sloučit do dvojic. Dvojici sousedních stavů pro funkci čtyřech proměnných popíšeme pomocí pouze třech proměnných, proto jsou hodnoty 0 a 1 v 5. sloupci tabulky jen na třech pozicích a na čtvrté je pomlčka. Jsou to tedy všechny dvojice sousedních jedniček funkce f. Dvojice jsou opět rozděleny do skupin podle počtu jedniček a skupiny jsou v tabulce odděleny prázdným řádkem (dtto i pro čtveřice). Dále hledáme všechny čtveřice - slučujeme stavy ze sloupce 5 a vytváříme sloupec 7 a 8. Zároveň označujeme hvězdičkou ve sloupci 6 ty dvojice, které byly sloučeny (pokryty nějakou čtveřicí). Může se stát, že čtveřici vytvoříme dvěma způsoby, protože se jedná o jednu a tutéž čtveřici stavů, použijeme dále pouze jednu (čtveřice stavů 8, 9, 10, 11 a 8, 10, 9, 11). Dále bychom hledali všechny skupiny 8, 16, atd. stavů; ty už v našem příkladu nejsou. Nyní si označíme velkými písmeny všechny skupiny stavů nebo i jednotlivé stavy u kterých se nevyskytuje hvězdička (tzn. nejsou pokryty žádnou větší skupinou stavů). V Tab. 3.2 jsou to A až F, které označují všechny přímé implikanty funkce f (jedná se o hledání MNDF). Z nich musíme vybrat minimální pokrytí jedničkových stavů.

Pro nalezení MNDF používáme tzv. *tabulku pokrytí*. Ve sloupcích jsou všechny jedničkové stavy (tedy stavy, které musíme pokrýt) a v řádcích jsou všechny přímé implikanty (tedy všechny skupiny sousedních stavů z Tab. 3.2). Cílem je nalezení takové množiny přímých implikantů,

1	2	3	4	5	6	7	8
d c b a	s pozn.	d c b a	s pozn.	d c b a	s,s pozn.	d c b a	s,s,s,s pozn.
0 0 0 1	1 *	0 0 0 1	1 *	- 0 0 1	1,9 F	1 0 - -	8,9,10,11 C
0 1 0 0	4 *	0 1 0 0	4 *	- 1 0 0	4,12 E	1 0 - -	8,10,9,11
0 1 1 1	7 *	1 0 0 0	8 *	1 0 0 -	8,9 *	1 - - 0	8,10,12,14 B
1 0 0 0	8 *			1 0 - 0	8,10 *	1 - - 0	8,12,10,14
1 0 0 1	9 *	1 0 0 1	9 *	1 - 0 0	8,12 *		
1 0 1 0	10 *	1 0 1 0	10 *			1 - 1 -	10,11,14,15 A
1 0 1 1	11 *	1 1 0 0	12 *	1 0 - 1	9,11 *	1 - 1 -	10,14,11,15
1 1 0 0	12 *			1 0 1 -	10,11 *		
1 1 1 0	14 *	0 1 1 1	7 *	1 - 1 0	10,14 *		
1 1 1 1	15 *	1 0 1 1	11 *	1 1 - 0	12,14 *		
		1 1 1 0	14 *				
				- 1 1 1	7,15 D		
		1 1 1 1	15 *	1 - 1 1	11,15 *		
				1 1 1 -	14,15 *		

Tab. 3.2 Tabulka všech skupin sousedních stavů (s je stavový index)

která jednak pokrývá všechny jedničkové stavy a jednak obsahuje minimální počet prvků (implikantů). V tabulce pokrytí (viz Tab. 3.3) označíme hvězdičkou, ze kterých stavů je daný imlikant vytvořen (které pokrývá). Tuto informaci nalezneme ve sloupcích 8 a 6 Tab. 3.2.

S tabulkou pokrytí (zde Tab. 3.3) pracujeme následujícím způsobem:

1. nalezneme sloupce, které obsahují pouze jednu hvězdičku. (V Tab. 3.3. jsou označené šipkou v řádku poznámky.) Tzn. že tyto stavy jsou pokryty pouze jedním z implikantů, který se tudíž v MNDF funkce musí vyskytovat. Tento implikant se nazývá *nesporný přímý implikant*. Množina nesporných přímých implikantů tvoří jádro řešení. V našem příkladu jsou to termy příslušné řádkům, které jsou označené D, E, F.

2. prohlédneme řádek s nesporným přímým implikantem a jestliže narazíme v tomto řádku na hvězdičku, potom sloupec, ve kterém je tato hvězdička, vyškrtneme (tento stav už je pokrytý). V našem příkladu jsou to sloupce 9, 12 a 15 (v Tab. 3.3 jsou označené $\times$ v řádku poznámky).

3. pokračujeme podle bodů 1 a 2 se zmenšenou tabulkou - bez řádků s nespornými přímými implikanty a bez vyškrtnutých sloupců. Zde jsme zmenšenou tabulku překreslili pro názornost ještě jednou, viz Tab. 3.4. Z Tab. 3.3 dále pracujeme pouze s řádky A, B, C a sloupci 8, 10, 11, 14.

4. jestliže najdeme takový sloupec, který obsahuje hvězdičky všude tam, kde je obsahuje některý z ostatních sloupců a ještě někde navíc (tedy dalo by se říci, že některý sloupec je jeho podmnožinou), můžeme ho vyškrtnout (tzv. *dominance sloupců*). V našem příkladu je to sloupec 10. Je to logické, protože hvězdičky ve všech třech řádcích znamenají, že ať vybereme kterýkoli řádek (z A, B, C), vždy pokryjeme stav 10; tedy na výběru nezáleží.

5. jestliže najdeme řádek, který je podmnožinou některého z ostatních řádků, můžeme ho

		1	4	7	8	9	10	11	12	14	15
A	bd						*	*		*	*
B	$\bar{a}d$				*		*		*	*	
C	$\bar{c}d$				*	*	*	*			
D	abc			*							*
E	$\bar{a}\bar{b}c$		*						*		
F	$a\bar{b}\bar{c}$	*				*					
poznámky		↑	↑	↑		×			×		×

Tab. 3.3 Tabulka pokrytí

		8	10	11	14
A	bd		*	*	*
B	$\bar{a}d$	*	*		*
C	$\bar{c}d$	*	*	*	

Tab. 3.4 Zmenšená tabulka pokrytí získaná z Tab. 3.3

vyškrtnout (tzv. *dominance řádků*). V našem případě by to znamenalo, že tytéž stavy jsou pokryty větší skupinou sousedních stavů. Tento případ se v našem příkladu nevyskytuje.

6. nakonec vybíráme termy podle kritéria minimality. Jedna z možných cílových situací (vysvětlení viz dále) je znázorněna v Tab. 3.4 - zůstaly tři sloupce a tři řádky, každý po dvou hvězdičkách (sloupec 10 vypadl podle bodu 4). Pro pokrytí všech sloupců (stavů) je nutné vybrat dva ze tří termů. V našem příkladu jsou dvě možnosti (další dvě obsahují více negací), tedy výsledné řešení je:

$$f(d,c,b,a) = a\bar{b}\bar{c} + \bar{a}\bar{b}c + abc + bd + \bar{a}d$$

nebo

$$f(d,c,b,a) = a\bar{b}\bar{c} + \bar{a}\bar{b}c + abc + bd + \bar{c}d$$

Tabulku pokrytí můžeme použít i při minimalizaci pomocí map, kde do řádků tabulky zahrneme všechny skupiny sousedních stavů získané z mapy. Problém pokrytí je jednou ze základních úloh, se kterou se setkáváme nejen v teorii automatů nebo v logickém návrhu, ale též v diagnostice nebo při řešení dopravních a ekonomických problémů. Proto je metodika jeho řešení celkem podrobně propracována a též naprogramována.

Metody řešení minimálního pokrytí jsou založeny na prohledávání všech možností, takže jsou velmi zdlouhavé a nákladné. Pro přehledný zápis všech možných pokrytí, z nichž pak můžeme vybrat minimální, byla zavedena tzv. Patrickova funkce, [13]. Jedná se o algebraický zápis podmínek pokrytí jednotlivých sloupců ve tvaru normální konjunktivní formy funkce. Převedením do disjunktivní formy pak získáme výčet všech pokrytí. Patrickovu funkci sestavujeme tak, že pro každý sloupec (j-tý) tabulky vytvoříme disjunkci všech těch řádků (i-tých), které obsahují v daném sloupci hvězdičku ($f_{i,j} = 1$), protože pro pokrytí stačí jeden řádek (operace OR). Vzniklé disjunkce, příslušející jednotlivým sloupcům pak logicky znásobíme, protože všechny sloupce musí být pokryty současně. V našem případě získáme tímto způsobem logický výraz složený z názvů řádků (A až F), podle kterého určíme, které termy, které můžeme vybrat pro minimální pokrytí jedničkových stavů funkce f.

Pro náš příklad by Patrickova funkce podle Tab. 3.4 byla vyjádřena pomocí symbolů v prvním sloupci následujícím způsobem :

$$(\mathrm{B} + \mathrm{C}).(\mathrm{A} + \mathrm{B} + \mathrm{C}).(\mathrm{A} + \mathrm{C}).(\mathrm{A} + \mathrm{B})$$

Převedením do disjunktivní formy (minimalizací pomocí zákonů Booleovy algebry) získáme vyjádření :

$$\mathrm{AB} + \mathrm{AC} + \mathrm{BC}$$

Tento výraz nám říká, že musíme vybrat libovolné dva řádky tabulky pokrytí, tedy stejný výsledek, ke kterému jsme dospěli ovšem intuitivním způsobem v bodě 6.

Patrickovu funkci by bylo možné použít hned zpočátku řešení a hledání v tabulce pokrytí (Tab. 3.3), ale bylo by to příliš zdlouhavé (složité algebraické úpravy). Existuje ještě jedna možnost řešení výběru z tabulky pokrytí, a to použít nějakou přibližnou metodu. Heuristická metoda hledání tzv. subminimálního pokrytí, [16], je založena na vyškrtávání řádků, které považujeme za nejméně pravděpodobné z hlediska zařazení do minimálního řešení. Každý sloupec ohodnotíme tzv. výškou pokrytí v_j (počet hvězdiček ve sloupci) a pro každý řádek spočítáme hodnotící funkci q_i podle vztahu :
(m je počet sloupců a $f_{i,j}$ má hodnotu 1, jestliže je v i-tém řádku a j-tém sloupci hvězdička)

$$g_i = \sum_{j=1}^{m} \frac{f_{i,j}}{v_j}$$

Potom řádky s nejnižším ohodnocením vyškrtáváme. Pro podrobnější a matematicky přesnou informaci viz [16].

2. Najděte MNDF pomocí metody Quine-McCluskey:
a) $f(d, c, b, a) = \sum(0, 1, 2, 5, 7, 8, 9, 10, 13, 15)$
b) $f(d, c, b, a) = \sum(0, 2, 3, 5, 7, 10, 12, 13, 14, 15)$
c) $f = ab\bar{c} + \bar{a}cd + bd + ac\bar{d} + \bar{a}\bar{b}\bar{c}\bar{d} + \bar{b}c\bar{d}$
(*Pomůcka:* upravte pomocí zákonů Booleovy algebry na ÚNDF)

3. Co všechno musíte změnit, jestliže použijete metodu Quine-McCluskey pro hledání MNKF?

4. Metodou Quine-McCluskey vyřešte příklady z předcházející kap. 4.3.
(*Pomůcka:* jestliže jsou zadané i neurčené stavy, postupujte tak, jako by to byly jedničkové stavy (ev. nulové pro hledání MNKF), ale nezahrnujte je do sloupců tabulky pokrytí).