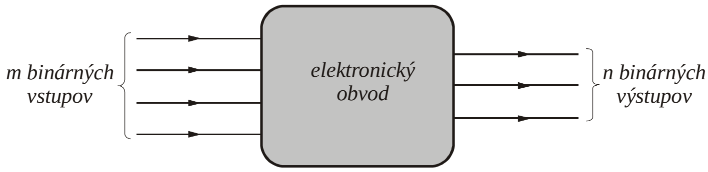
9. kapitola

Boolove funkcie a logické obvody

9.1 Boolova algebra

Elektronické obvody v počítačoch a v podobných zariadeniach sú charakterizované binárnymi vstupmi a výstupmi (rovnajúcimi sa 0 alebo 1), transformácia vstupu na výstupu sa uskutočňuje prostredníctvom elektronického obvodu, ktorý tvorí jadro tohto „transformačného“ zariadenia, pozri obr. 7.1. Elektronický obvod môže byť formálne simulovaný tzv. Boolovou funkciou, ktorá transformuje *m* binárnych vstupných premenných na *n* výstupných binárnych premenných.



Obrázok 9.1. Znázornenie elektronického obvodu, ktorý má *m* binárnych vstupov a *n* binárnych výstupov. Činnosť elektronického obvodu spočíva v transformácii binárnych vstupných hodnôt na binárne výstupné hodnoty.

Všeobecná definícia Boolovej funkcie je $f:\{0,1\}^m \to \{0,1\}^n$, táto funkcia transformuje binárny vektor dĺžky *m* na binárny vektor dĺžky *n*. Môžeme si položiť otázku, ako realizovať túto Boolovu funkciu, aby mala vopred špecifikované vlastnosti? Tento problém je realizovaný pomocou Boolovej algebry, ktorá pomocou premenných s 0-1 ohodnotením (t. j. binárnych) premenných a pomocou niekoľkých binárnych algebraických operácií a jednej unárnej algebraickej operácie je schopná dostatočne všeobecne modelovať Boolove funkcie s vopred špecifikovanými vlastnosťami. Poznamenajme, že Boolova algebra má dva známe modely, prvým je výroková logika a druhým algebra teórie množín. Pretože obe tieto zdanlivo odťažité disciplíny majú rovnakú „metateóriu“, existuje medzi zákonmi výrokovej logiky a formulami teórie množín „dualizmus“, pomocou ktorého ku každému zákonu výrokovej logiky priradíme 1-1-značne formulu teórie množín a naopak. Dobrým, ilustratívnym príkladom tohto dualizmu sú De Morganove formuly, ktoré vo výrokovej logike a v teórii množín majú tvary

$$\neg(p \wedge q) \equiv (\neg p \vee \neg q) \Leftrightarrow \overline{A \cap B} = \bar{A} \cup \bar{B}$$

$$\neg(p \vee q) \equiv (\neg p \wedge \neg q) \Leftrightarrow \overline{A \cup B} = \bar{A} \cap \bar{B}$$

V týchto formulách, vo výrokovej logike unárna logická spojka negácie má ekvivalent v teórii množín v unárnej algebraickej operácii doplnku, a podobne, vo výrokovej logike binárne spojky konjunkcie a disjunkcie majú ekvivalenty v teórii množín v binárnych algebraických operáciách prieniku resp. zjednotenia. Na záver je potrebné poznamenať, že výroková spojka ekvivalentnosti má v teórii množín ekvivalent v relácii rovnosti. Tieto priradenia vo výrokovej logike a v teórii množín môžeme zosumarizovať takto

výrokové premenné $p, q, r, \ldots \Leftrightarrow$ množiny $A, B, C, \ldots$

spojka negácie $\neg \Leftrightarrow$ operácia doplnku $\overline{}$

$$\text{spojka konjunkcie} \wedge \Leftrightarrow \text{operácia prieniku} \cap$$
$$\text{spojka disjunkcie} \vee \Leftrightarrow \text{operácia zjednotenia} \cup$$
$$\text{spojka ekvivalentnosti} \equiv \Leftrightarrow \text{relácia rovnosti} =$$

Definícia 9.1. ***Boolova algebra*** je algebraická štruktúra špecifikovaná usporiadanou 6-ticou $\left(B,+,\cdot,\overline{\ \ },\mathbf{0},\mathbf{1}\right)$, kde $B=\{a,b,...,x,y,...\}$ je neprázdna množina elementov (premenných Boolovej algebry), ktorá obsahuje dva špeciálne odlíšené elementy - konštanty $\mathbf{0},\mathbf{1}\in B$ a nad ktorou sú definované binárne operácie súčinu a súčtu

$$\cdot: B\times B\to B \tag{9.1a}$$

$$+: B\times B\to B \tag{9.1b}$$

a unárna operácia komplementu

$$\overline{\ \ }: B\to B \tag{9.1c}$$

ktoré vyhovujú týmto podmienkam

(1) komutatívnosť:

$$x\cdot y=y\cdot x, \qquad x+y=y+x \tag{9.1d}$$

(2) asociatívnosť:

$$\left(x\cdot y\right)\cdot z=x\cdot\left(y\cdot z\right), \qquad \left(x+y\right)+z=x+\left(y+z\right) \tag{9.1e}$$

(3) distributívnosť:

$$x\cdot\left(y+z\right)=\left(x\cdot y\right)+\left(x\cdot z\right), \qquad x+\left(y\cdot z\right)=\left(x+y\right)\cdot\left(x+z\right) \tag{9.1f}$$

(4) vlastnosť konštanty **0**:

$$x=x+\mathbf{0}, \qquad x\cdot\overline{x}=\mathbf{0} \tag{9.1g}$$

(5) vlastnosť konštanty **1** :

$$x=x\cdot\mathbf{1}, \qquad x+\overline{x}=\mathbf{1} \tag{9.1h}$$

V literatúre existuje mnoho alternatívnych notácií Boolovej algebry. Napríklad operácia súčinu sa alternatívne vyjadruje symbolmi $\wedge$ alebo $*$, podobne, operácia súčtu symbolmi $\vee$ a $\oplus$. Pre zjednodušenie notácie budeme vynechávať symbol súčinu, formulu $x\cdot y$ budeme zjednodušene písať ako *xy*. Z formúl (8.1g-h) vyplýva, že konštanta **1** (**0**) má úlohu neutrálneho (jednotkového) prvku pre súčin (súčet). Z bežného pohľadu na formuly (8.1g-h) by niekto mohol odvodiť záver, že výraz $\overline{x}$ je inverzná formula pre *x*. Pripomeňme si, že v kapitole 6 bol inverzný element definovaný pomocou vlastnosti $x\cdot\overline{x}=\mathbf{1}$, avšak podľa pravej formuly (A.1g) platí $x\cdot\overline{x}=\mathbf{0}$, z čoho vyplýva, že výraz $\overline{x}$ nemá vlastnosti inverzného prvku (tak vzhľadom k operácii súčtu, ako aj súčinu).

Príklad 9.1. Najjednoduchšia Boolova algebra (s veľkým významom v informatike a v logike) je založená na dvojprvkovej množine $B=\{0,1\}$. Binárne operácie súčinu, súčtu a unárna operácia komplementu sú pomocou multiplikačných tabuliek definované takto

$+$	**0**	**1**
0	**0**	**1**
1	**1**	**1**

$\cdot$	**0**	**1**
0	**0**	**0**
1	**0**	**1**

b	$\overline{b}$
0	**1**
1	**0**

Jednoducho sa môžeme presvedčiť, že pre takto špecifikované operácie sú splnené podmienky (8.1a-h), t. j. algebraická štruktúra $\left(B,+,\cdot,\overline{\ \ },\mathbf{0},\mathbf{1}\right)$ je Boolova algebra.

Príklad 9.2. Nech $A = \{a, b, c, ...\}$ je neprázdna množina, položme $B = \mathcal{P}(A)$. Operácie $\cdot$ a $+$ sú realizované pomocou množinových operácií $\cap$ resp. $\cup$, operácia komplementu je realizovaná ako množinový komplement vzhľadom k množine *A*, $\overline{x} = A - x$. Potom platí:

(a) binárne operácie sú asociatívne, komutatívne,
(b) medzi binárnymi operáciami platia distributívne zákony,
(c) prázdna množina $\emptyset$ má vlastnosti jednotkového elementu pre operáciu $\cup$
$$(\forall X \in B)(X \cup \emptyset = \emptyset \cup X = X)$$
(d) množina *A* má vlastnosti jednotkového elementu pre operáciu $\cap$
$$(\forall X \in B)(X \cap A = A \cap X = X)$$
(e) pre každé $X \in B$ existuje komplement $\overline{X} \in B$ taký, že
$$(\forall X \in B)(X \cap \overline{X} = \emptyset)$$
$$(\forall X \in B)(X \cup \overline{X} = A)$$

To znamená, že podmienky (8.1a-h) sú splnené, t. j. algebraická štruktúra $(\mathcal{P}(A), \cup, \cap, \overline{}, \emptyset, A)$ je Boolova algebra.

Príklad 9.3. Nech $B = \{p, q, r, ...\}$ je množina výrokových formúl, ktorá je uzavretá vzhľadom k binárnym operácia konjunkcie ($\wedge$), disjunkcie ($\vee$) a k unárnej operácii negácie ($\neg$). Pre túto množinu je definovaná aj relácia ekvivalentnosti ´$\equiv$´, dve formuly sú ekvivalentné vtedy a len vtedy, ak majú rovnakú pravdivostnú interpretáciu (logicky ekvivalentné). Z množiny *B* vyberieme formulu kontradikciu (napr. $p \wedge \neg p$) a označíme ju symbolom 0; podobne formula tautológia (napr. $p \vee \neg p$) je označená symbolom 1. To znamená, že symboly 0 a 1 patria do množiny *B*. Pre každú formulu *p* platia tieto vzťahy
$$p \vee \mathbf{0} = \mathbf{0} \vee p = p$$
$$p \wedge \mathbf{1} = \mathbf{1} \wedge p = p$$
Pretože logické spojky konjunkcie a disjunkcie sú komutatívne a asociatívne, pre tieto operácie platia taktiež distributívne zákony, podmienky z definície A.1 sú splnené, t. j. algebraická štruktúra $(B, \vee, \wedge, \neg, 0, 1)$ tvorí Boolovu algebru.

9.2 Vlastnosti Boolovej algebry

V úvodnej časti kapitoly A.1 bol zmienený princíp duality medzi algebrou teórie množín a výrokovou logikou. Ukážeme, že tento princíp je aplikovateľný aj pre rôzne Boolove algebry.

Jednoduchými prostriedkami je možné dokázať jednoznačnosť jednotkového (neutrálneho) elementu a taktiež aj jednoznačnosť komplementu..

Veta 9.2. Jednotkové elementy **1** a **0** existujú jednoznačne.

Podobne sa dá dokázať aj jednoznačnosť existencie inverzných elementov v Boolovej algebre.

Veta 9.3. Pre každý element $x \in B$ existuje jednoznačne element $\overline{x} \in B$ taký, že $x \cdot \overline{x} = \mathbf{0}$ a $x + \overline{x} = \mathbf{1}$ (t. j .sú splnené podmienky 7.1g-h).

Veta 9.4. Nech $\left(B,+,\cdot,\overline{},\mathbf{0},\mathbf{1}\right)$ je Boolova algebra, potom platia tieto formulu

(1) Involutívnosť komplementu

$$\left(\forall x\in B\right)\left(\overline{\overline{x}}=x\right) \tag{9.2a}$$

(2) Idempotentnosť

$$\left(\forall x\in B\right)\left(x\cdot x=x\right) \tag{9.2b}$$

$$\left(\forall x\in B\right)\left(x+x=x\right) \tag{9.2c}$$

(3) De Morganove zákony

$$\left(\forall x,y\in B\right)\left(\overline{x+y}=\overline{x}\cdot\overline{y}\right) \tag{9.2d}$$

$$\left(\forall x,y\in B\right)\left(\overline{x\cdot y}=\overline{x}+\overline{y}\right) \tag{9.2e}$$

(4) Nulitnosť

$$\left(\forall x\in B\right)\left(x+\mathbf{1}=\mathbf{1}\right) \tag{9.2f}$$

$$\left(\forall x\in B\right)\left(x\cdot\mathbf{0}=\mathbf{0}\right) \tag{9.2g}$$

(5) Absorpcia

$$\left(\forall x,y\in B\right)\left(x+\left(x\cdot y\right)=x\right) \tag{9.2h}$$

$$\left(\forall x\in B\right)\left(x\cdot\left(x+y\right)=x\right) \tag{9.2i}$$

(6) Komplementy konštánt

$$\overline{\mathbf{0}}=\mathbf{1} \tag{9.2j}$$

$$\overline{\mathbf{1}}=\mathbf{0} \tag{9.2k}$$

(7) Vlastnosti konštánt vzhľadom k binárnym operáciam

$$\mathbf{0}+\mathbf{0}=\mathbf{0},\mathbf{0}+\mathbf{1}=\mathbf{1},\mathbf{1}+\mathbf{0}=\mathbf{1},\mathbf{1}+\mathbf{1}=\mathbf{1} \tag{9.2l}$$

$$\mathbf{0}\cdot\mathbf{0}=\mathbf{0},\mathbf{0}\cdot\mathbf{1}=\mathbf{0},\mathbf{1}\cdot\mathbf{0}=\mathbf{0},\mathbf{1}\cdot\mathbf{1}=\mathbf{1} \tag{9.2m}$$

Dôkaz týchto vlastností prenecháme na cvičenie.

9.3 Boolove funkcie

V úvode k tejto kapitole bola Boolova funkcia definovaná ako funkcia nad binárnymi premennými $\{0,1\}$. Tento pomerne zjednodušený pohľad na Boolovu funkciu bude teraz rozšírený tak, aby koncepcia Boolovej funkcie bola časťou Boolovej algebry. Základný pojem pre definíciu Boolovej funkcie je pojem Boolovej premennej. Použijeme analogický prístup, aký sa používa pre definíciu reálnej premennej, je to veličina, ktorá môže nadobúdať hodnoty z množiny reálnych čísel.

Definícia 9.2. Nech $\left(B,+,\cdot,\overline{},\mathbf{0},\mathbf{1}\right)$ je Boolova algebra. Potom,

(1) ***Boolova premenná*** je taká premenná, ktorá nadobúda hodnoty z množiny *B*,

(2) ***komplement premennej*** *x*, označený $\overline{x}$, je taká premenná, ktorej hodnota sa rovná komplementu hodnoty premennej *x* (t. j. ak $x=b\in B$, potom $\overline{x}=\overline{b}\in B$,

(3) ***literál*** je Boolova premenná *x* alebo jej komplement $\overline{x}$.

V ďalšom texte budeme používať notáciu, ktorá umožní rozlíšiť literál

$$x^e = \begin{cases} x & (pre\ e = 1) \\ \overline{x} & (pre\ e = 0) \end{cases} \tag{9.3}$$

Podobne ako pre reálnu premennú, aj Boolova premenná môže byť kombinovaná do tvaru Boolových formúl použitím binárnych operácií súčinu, súčtu a komplementu.

Definícia 9.3. Nech $\left(B,+,\cdot,\overline{\ },\mathbf{0},\mathbf{1}\right)$ je Boolova algebra. Potom ***Boolova formula***, obsahujúca Boolove premenné $x_1,x_2,...,x_n$, je definovaná takto:

(1) konštanty **0** a **1** sú Boolove formuly,

(2) Boolove premenné $x_1,x_2,...,x_n$ sú Boolove formuly,

(3) ak X a Y sú Boolove formuly, potom aj výrazy $\left(X\cdot Y\right)$, $\left(X+Y\right)$, $\overline{X}$ a $\overline{Y}$ sú Boolove formuly.

V tejto definícii bod (3) môže byť rozšírený ešte o ďalšie Boolove spojky, napr. spojku ⊕, ktorá sa nazýva exkluzívna disjunkcia (XOR) a je definovaná takto: $x\oplus y = 1$ vtedy a len vtedy, ak buď $x = 1$ alebo $y = 1$, ale nie súčasne ($x \neq y$); $x\oplus y = 0$ vtedy a len vtedy, ak premenné x a y majú rovnaku hodnotu ($x = y$).

V ďalšom texte budeme používať konvenciu, že ak bude jasné o akú formulu sa jedná, tak termín ´Boolova formula´ budeme skracovať na ´formula´. Podobne, ako vo výrokovej logike môžeme si definovať rastúcu prioritu operácií takto: (1) súčet, (2) súčin a (3) komplement. Napríklad, formulu $\left(\left(x\cdot y\right)+z\right)$ môžeme pomocou tejto konvencie vyjadriť v zjednodušenom tvare bez zátvoriek $x\cdot y+z$. Konečne, podobne ako v štandardnej algebre, budeme vynechávať znak súčinu, napríklad predchádzajúci ilustračný príklad má tvar $xy+z$.

Príklad 9.5. Zjednodušte formulu $\left(\left(x+y\right)\cdot\left(\overline{x}+\overline{y}\right)\right)$.

Použitím distributívneho zákona a (A.1g-h)

$$\left(\left(x+y\right)\cdot\left(\overline{x}+\overline{y}\right)\right)=\left(x\cdot\overline{x}\right)+\left(x\cdot\overline{y}\right)+\left(y\cdot\overline{x}\right)+\left(y\cdot\overline{y}\right)=\underbrace{x\overline{x}}_{\mathbf{0}}+x\overline{y}+y\overline{x}+\underbrace{y\overline{y}}_{\mathbf{0}}=x\overline{y}+\overline{x}y$$

Definícia 9.4. Dve Boolove formule sú ***ekvivalentné*** (alebo ***rovné***) vtedy a len vtedy, ak jedna pomocou konečného počtu aplikácií axióm Booloej algebry je pretransformovaná na druhú formulu.

Podľa príkladu A.4 formule $\varphi_1=\left(x+y\right)\cdot\left(\overline{x}+\overline{y}\right)$ a $\varphi_2=x\overline{y}+\overline{x}y$ sú ekvivalentné, pretože druhú formulu získame z prvej použitím konečného počtu aplikácií axióm Boolovej algebry, potom $\varphi_1=\varphi_2$.

Konečne sa dostávame k definícii Boolovej funkcie $f\left(x_1,x_2,...,x_n\right)$ ako Boolovej formuly, ktorá obsahuje premenné $x_1,x_2,...,x_n$. Napríklad

$$f\left(x_1,x_2,x_3\right)=x_1\left(x_2+\overline{x}_3\right)$$

Definícia 9.5. Nech $\left(B,+,\cdot,\overline{\ },\mathbf{0},\mathbf{1}\right)$ je Boolova algebra.

(1) ***Boolova funkcia*** $f\left(x_1,x_2,...,x_n\right)$ premenných $x_1,x_2,...,x_n$, je zobrazenie $f:B^n\to B$,

pričom $f\left(x_1, x_2, ..., x_n\right)$ je špecifikovaná ako Boolova formula.
(2) Všetky Boolove formuly, ktoré sú navzájom ekvivalentné, definujú rovnakú funkciu.

Z tejto definície vyplýva, že ekvivaletné Boolove formuly špecifikujú rovnakú Boolovu formulu. Napríklad, máme dve funkcie

$$f : B^2 \to B \qquad f\left(x_1, x_2\right) = x_1\left(\overline{x}_1 + x_2\right)$$

$$g : B^2 \to B \qquad g\left(x_1, x_2\right) = x_1 x_2$$

Použitím distribučného zákona ľahko dokážeme, že formuly sú ekvivaletné, $x_1\left(\overline{x}_1 + x_2\right) = x_1 x_2$, potom funkcie *f* a *g* sú rovnaké.

Pretože Boolova funkcia môže byť vyjadrená mnohými rôznymi formulami, ktoré sú navzájom ekvivaletné, vzniká otázka, ako efektívne rozhodnúť, či dve Boolove formuly sú ekvivalentné, alebo či dve Boolové funkcie sú rovnaké. Ukážeme postup, ktorý nie je založený na transformácii jednej formuly na druhú, aby sme rozhodli, či funkcie sú rovnaké, ale navrhne sa „kanonická“ reprezentácia Boolovej funkcie, podľa ktorej môžeme jednoducho rozhodnúť, či dve Boolove funkcie sú rovnaké alebo nie.

Definícia 9.6. ***Súčinová klauzula*** premenných $x_1, x_2, ..., x_n$ je Boolova formula, ktorá obsahuje súčin *n* literálov (t. j. premennú alebo jej komplement) pre každú premennú.

Ako príklad súčinovej klauzuly premenných x_1, x_2, x_3 sú tieto formuly: $x_1x_2x_3$, $x_1x_2\overline{x}_3$, $x_1\overline{x}_2x_3$, $\overline{x}_1x_2x_3, \ldots$, $\overline{x}_1\overline{x}_2\overline{x}_3$. Ak použijeme formalizmus x^e, potom súčinovú klauzulu premenných $x_1, x_2, ..., x_n$, ktorá je špecifikovaná binárnym vektorom $\boldsymbol{e} = \left(e_1, e_2, ..., e_n\right)$, má tvar

$$l_{\boldsymbol{e}} = x_1^{e_1} x_2^{e_2} ... x_n^{e_n} \tag{9.4}$$

Napríklad, pre $\boldsymbol{e} = \left(11011\right)$ súčinová klauzula má tvar

$$l_{(11011)} = x_1^1 x_2^1 x_3^0 x_4^1 x_5^1 = x_1 x_2 \overline{x}_3 x_4 x_5$$

Pretože binárnych vektorov $\boldsymbol{e} = \left(e_1, e_2, ..., e_n\right)$ je 2^n, potom aj ***rôznych*** súčinových klauzúl je 2^n.

Definícia 9.7. ***Súčtová klauzula*** premenných $x_1, x_2, ..., x_n$ je Boolova formula, ktorá obsahuje súčet *n* literálov (t. j. premennú alebo jej komplement) pre každú premennú.

Podobne ako pre súčinovú klauzulu, môžeme aj súčtovú klauzulu pre premenné $x_1, x_2, ..., x_n$ špecifikovať binárnym vektorom $\boldsymbol{e} = \left(e_1, e_2, ..., e_n\right)$

$$L_{\boldsymbol{e}} = x_1^{e_1} + x_2^{e_2} + ... + x_n^{e_n} \tag{9.5}$$

Pre $\boldsymbol{e} = \left(10100\right)$ súčtová klauzula má tvar

$$L_{\boldsymbol{e}} = x_1^1 + x_2^0 + x_3^1 + x_4^0 + x_5^0 = x_1 + \overline{x}_2 + x_3 + \overline{x}_4 + \overline{x}_5$$

Pretože každá súčtová klauzula premenných $x_1, x_2, ..., x_n$ je špecifikovaná binárnym vektorom $\boldsymbol{e} = \left(e_1, e_2, ..., e_n\right)$, potom počet súčtových klauzúl je taktiež 2^n.

Týmto sa dostávame k formulácii hlavného výsledku tejto kapitoly, že každá Boolova funkcia môže byť jednoznačne vyjadrená ako sumácia súčinových klauzúl (tento tvar sa nazýva vo výrokovej logike ***disjunktívna normálna forma***, skratka DNF).

Príklad 9.6. Vyjadrite Boolovu funkciu $x_1 x_2 (x_1 + x_3)$ pomocou súčtu súčinových klauzúl (DNF)

$$\begin{aligned} x_1 x_2 (x_1 + x_3) &= x_1 x_2 x_1 + x_1 x_2 x_3 \\ &= \underbrace{x_1 x_1}_{x_1} x_2 + x_1 x_2 x_3 = x_1 x_2 + x_1 x_2 x_3 \\ &= x_1 x_2 \mathbf{1} + x_1 x_2 x_3 = x_1 x_2 (x_3 + \overline{x}_3) + x_1 x_2 x_3 \\ &= \underbrace{x_1 x_2 x_3 + x_1 x_2 x_3}_{x_1 x_2 x_3} + x_1 x_2 \overline{x}_3 \\ &= x_1 x_2 x_3 + x_1 x_2 \overline{x}_3 \end{aligned}$$

Veta 9.5. Každá Boolova funkcia $f(x_1, x_2, ..., x_n)$, ktorá sa identicky nerovná nule, môže byť špecifikovaná ako suma súčinových klauzúl

$$\begin{aligned} f(x_1, x_2, ..., x_n) &= \sum_{\boldsymbol{e}} f(e_1, e_2, ..., e_n) x_1^{e_1} x_2^{e_2} ... x_n^{e_n} \\ &= \sum_{\boldsymbol{e}} f(e_1, e_2, ..., e_n) l_{(e_1, e_2, ..., e_n)} \\ &= \sum_{\substack{\boldsymbol{e} \\ (f(\boldsymbol{e})=1)}} f(\boldsymbol{e}) l_{\boldsymbol{e}} \end{aligned} \tag{9.6}$$

V poslednej formule berieme v úvahu len také binárne hodnoty argumentov $\boldsymbol{e}$ pre ktoré sú funkčné hodnoty jednotkové, nulové funkčné hodnoty, $f(\boldsymbol{e}) = 0$, môžu byť ignorované.

Naznačíme jednoduchý konštruktívny dôkaz. Boolova funkcia $f(x_1, x_2, ..., x_n)$ je alternatívne špecifikovaná pomocou jej funkčných hodnôt $f(e_1, e_2, ..., e_n)$, pre všetky hodnoty binárneho vektora $\boldsymbol{e} = (e_1, e_2, ..., e_n) \in \{0,1\}^n$.

#	$\boldsymbol{e} = (e_1, e_2, ..., e_n)$	$f(e_1, e_2, ..., e_n)$
1	(00........00)	0
2	(00........01)	1
....		
i	$(e_1^{(i)}, e_2^{(i)}, ..., e_n^{(i)})$	1/0
....		
2^n	(11........11)	0

Súčinová klauzula $l_{(e_1, e_2, ..., e_n)}(x_1, x_2, ..., x_n) = x_1^{e_1} x_2^{e_2} ... x_n^{e_n}$, ktorá je priradená binárnemu vektoru $\boldsymbol{e} = (e_1, e_2, ..., e_n) \in \{0,1\}^n$ má zaujímavú vlastnosť, jej funkčná hodnota sa rovná 1 len pre $(x_1, x_2, ..., x_n) = (\boldsymbol{e}_1, \boldsymbol{e}_2, ..., \boldsymbol{e}_n)$, kde $\boldsymbol{e}_i \in \{0,1\}$, pre všetky iné prípady funkčná hodnota je 0

$$l_{(e_1, e_2, ..., e_n)}(x_1, x_2, ..., x_n) = \begin{cases} 1 & (pre\ (x_1, x_2, ..., x_n) = (\boldsymbol{e}_1, \boldsymbol{e}_2, ..., \boldsymbol{e}_n)) \\ 0 & (pre\ (x_1, x_2, ..., x_n) \neq (\boldsymbol{e}_1, \boldsymbol{e}_2, ..., \boldsymbol{e}_n)) \end{cases} \tag{9.7}$$

To znamená, že pre konštrukciu (9.6) sú pre nás dôležité len funkčné hodnoty 1, funkčné hodnoty 0 nie sú podstatné pre náš konštruktívny dôkaz. Zostrojíme Boolovu formulu ako sumáciu týchto klauzúl (t. j. v DNF tvare)

$$F\left(x_1,x_2,...,x_n\right)=\sum_{\boldsymbol{e}} f\left(e_1,e_2,...,e_n\right)l_{\left(e_1,e_2,...,e_n\right)} \tag{9.8}$$

Z konštrukcie tejto Boolovej funkcie, vyplýva, že jej funkčné hodnoty sú špecifikované tabuľkou funkčných hodnôt Boolovej funkcie $f\left(x_1,x_2,...,x_n\right)$. To znamená, že Boolove funkcie $f\left(x_1,x_2,...,x_n\right)$ a $F\left(x_1,x_2,...,x_n\right)$ sú ekvivalentné, t. j. majú rovnaké funkčné hodnoty pre rôzne hodnoty argumentov. Týmto sme zavŕšili jednoduchý intuitívny konštruktívny dôkaz vety 9.5.

Poznamenajme, že DNF tvar Boolovej funkcie je určený jednoznačne až na permutácie argumentov v súčtových klauzulách, alebo až na permutácie súčtových klauzúl. Táto nejednoznačnosť DNF tvaru vyplýva zo skutočnosti, že binárne operácie súčtu a súčinu sú komutatívne. Môžeme teda konštatovať, že DNF sú základné charakteristiky (niečo ako odtlačky prstov alebo zloženie DNA) Boolových funkcií. Aby sme odstránili prípadné nejednoznačnosti zapisujeme DNF v tzv. kanonickom tvare, t. j. jednotlivé argumenty sa zapisujú postupne podľa rastúceho indexu (tým sme odstránili nejednoznačnosti v dôsledku kumutatívnosti súčinu) a potom jednotlivé súčinové klauzule sú písané v poradí rastúcej číselnej hodnoty „indexu“ $\boldsymbol{e}=\left(e_1,e_2,...,e_n\right)$.

Príklad 9.7. Jednoduchý dôkaz formuly (9.6) môžeme zostrojiť pomocou Shannonovej formuly [xx] Boolovej funkcie $f(x_1)$

$$f\left(x_1\right)=\overline{x}_1 f\left(0\right)+x_1 f\left(1\right)$$

Ktorá je jednoduchým dôsledkom skutočnosti, že v Boolovej algebre funkcia $f(x_1)$ má len dve možné hodnoty **0** alebo **1**. Túto formulu môžeme aplikovať k funkcii $f\left(x_1,x_2\right)$ k premennej x_1

$$f\left(x_1,x_2\right)=\overline{x}_1 f\left(0,x_2\right)+x_1 f\left(1,x_2\right)$$

Na komponenty $f\left(0,x_2\right)$ a $f\left(1,x_2\right)$ Shanonnovu formulu

$$f\left(0,x_2\right)=\overline{x}_2 f\left(0,0\right)+x_2 f\left(0,1\right)$$

$$f\left(1,x_2\right)=\overline{x}_2 f\left(1,0\right)+x_2 f\left(1,1\right)$$

Potom pre $f\left(x_1,x_2\right)$ dostaneme

$$f\left(x_1,x_2\right)=\overline{x}_1\overline{x}_2 f\left(0,0\right)+\overline{x}_1 x_2 f\left(0,1\right)+x_1\overline{x}_2 f\left(1,0\right)+x_1 x_2 f\left(1,1\right)$$

Týmto sme dokázali, že pomocou Shanonnovej formuly pre n=1 dokážeme aj formulu pre n=2. Týmto postupom (matematickou indukciou) dokážeme, že formula platí (8.6) aj pre ľubovolné celočíselne n.

Príklad 9.8. Zostrojte Boolovu funkciu $f\left(x_1,x_2\right)=x_1+x_2$ v tvare DNF. Podľa vety 8.5 DNF tvar tejto funkcie je

$$f\left(x_1,x_2\right)=f\left(0,0\right)\overline{x}_1\overline{x}_2+f\left(0,1\right)\overline{x}_1 x_2+f\left(1,0\right)x_1\overline{x}_2+f\left(1,1\right)x_1 x_2$$

kde jednotlivé funkčné hodnoty sú uvedené v tabuľke

#	e_1	e_2	$f(e_1,e_2)$
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	1

Potom funkcia *f* má tvar

$$\begin{aligned} f(x_1,x_2) &= 0\bar{x}_1\bar{x}_2 + 1\bar{x}_1x_2 + 1x_1\bar{x}_2 + 1x_1x_2 \\ &= \bar{x}_1x_2 + x_1\bar{x}_2 + x_1x_2 \end{aligned}$$

Ľahko dokážeme, že takto definovaná funkcia $f(x_1,x_2) = \bar{x}_1x_2 + x_1\bar{x}_2 + x_1x_2$ je ekvivalentná s pôvodnou Boolovou funkciou $f(x_1,x_2) = x_1 + x_2$

$$f(x_1,x_2) = \bar{x}_1x_2 + x_1\bar{x}_2 + x_1x_2 = \bar{x}_1x_2 + x_1\bar{x}_2 + x_1x_2 + x_1x_2 = \left(\underbrace{\bar{x}_1 + x_1}_{1}\right)x_2 + x_1\left(\underbrace{\bar{x}_2 + x_2}_{1}\right) = x_1 + x_2$$

Príklad 9.9. Zostrojte Boolovu funkciu $f(x_1,x_2,x_3) = x_2x_3 + x_1x_3$ v tvare DNF. Táto Boolova funkcia je určená tabuľkou funkčných hodnôt

#	e_1	e_2	e_3	e_2e_3	e_1e_3	$e_2e_3 + e_1e_3$
1	0	0	0	0	0	0
2	0	0	1	0	0	0
3	0	1	0	0	0	0
4	0	1	1	1	0	1
5	1	0	0	0	0	0
6	1	0	1	0	1	1
7	1	1	0	0	0	0
8	1	1	1	1	1	1

Potom funkcia $f(x_1,x_2,x_3)$ (uvažujeme len jednotkové funkčné hodnoty) má DNF tvar

$$f(x_1,x_2,x_3) = \bar{x}_1x_2x_3 + x_1\bar{x}_2x_3 + x_1x_2x_3$$

Použijeme duálny princíp z vety 8.1, veta 8.5 má potom tento duálny tvar

Veta 9.6. Každá Boolova funkcia $f(x_1,x_2,...,x_n)$, ktorá sa identicky nerovná nule, môže byť špecifikovaná ako súčin sumačných klauzúl

$$\begin{aligned} f(x_1,x_2,...,x_n) &= \prod_{\boldsymbol{e}} \left(f(e_1,e_2,...,e_n) + x_1^{1-e_1} + x_2^{1-e_2} + ... + x_n^{1-e_n} \right) \\ &= \prod_{\boldsymbol{e}} \left(f(e_1,e_2,...,e_n) + L_{(1-e_1,1-e_2,...,1-e_n)} \right) \\ &= \prod_{\substack{\boldsymbol{e} \\ (f(\boldsymbol{e})=\mathbf{0})}} \left(f(\boldsymbol{e}) + L_{\bar{\boldsymbol{e}}} \right) \end{aligned} \tag{9.10}$$

kde $\bar{\boldsymbol{e}} = (\bar{e}_1,\bar{e}_2,...,\bar{e}_n) = (1-e_1,1-e_2,...,1-e_n)$.

Táto veta reprezentuje hlavný duálny výsledok tejto kapitoly, že každá Boolova funkcia môže byť jednoznačne vyjadrená ako súčin súčtových klauzúl (tento tvar sa nazýva vo výrokovej logike ***konjunktívna normálna forma***, skratka KNF).

Príklad 9.10 . Vyjadrite $f(x_1,x_2) = x_1(x_1 + x_2)$ v KNF tvare.

V prvom kroku zostrojíme tabuľku funkčných hodnôt tejto Boolovej funkcie

#	e_1	e_2	e_1+e_2	$e_1(e_1+e_2)$
1	0	0	0	0
2	0	1	1	0
3	1	0	1	1
4	1	1	1	1

Použitím (9.10) dostaneme vyjadrenú Boolovu funkciu $f(x_1,x_2)=x_1(x_1+x_2)$ v KNF

$$f(x_1,x_2)=\left(\underbrace{f(0,0)}_{0}+x_1+x_2\right)\cdot\left(\underbrace{f(0,1)}_{0}+x_1+\overline{x}_2\right)\cdot\underbrace{\left(\underbrace{f(1,0)}_{1}+\overline{x}_1+x_2\right)}_{1}\cdot\underbrace{\left(\underbrace{f(1,1)}_{1}+\overline{x}_1+\overline{x}_2\right)}_{1}$$

$$=(x_1+x_2)\cdot(x_1+\overline{x}_2)$$

Z tohto príkladu vyplýva, že pre konštrukciu KNF sú dôležité nulové funkčné hodnoty danej Boolovej funkcie. Táto vlastnosť je duálnu k vlastnosti DNF, kde sú relevantné jednotkové funkčné hodnoty Boolovej funkcie. Z tohto faktu vyplýva skutočnosť, že si zvolíme DNF tvar Boolovej funkcie vtedy, keď tabuľka obsahuje v prevažnej miere nulové funkčné hodnoty, KNF si zvolíme vtedy, keď tabuľka obsahuje v prevažnej miere jednotkové funkčné hodnoty. V prípade, že tabuľka obsahuje rovnaký počet nulových a jednotkových funkčných hodnôt z pohľadu „zložitosti“ konštrukcie je jedno, aký tvar Boolovej funkcie sme zvolili.

Príklad 9.11. Zostrojte KNF Boolovej funkcie $f(x_1,x_2,x_3)=(\overline{x}_1+x_2)\cdot(\overline{x}_1+\overline{x}_3)$. Tabuľka funkčných hodnôt má tvar

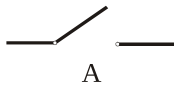
#	e_1	e_2	e_3	$\overline{e}_1$	$\overline{e}_3$	$\overline{e}_1+e_2$	$\overline{e}_1+\overline{e}_3$	$(\overline{e}_1+e_2)\cdot(\overline{e}_1+\overline{e}_3)$
1	0	0	0	1	1	1	1	1
2	0	0	1	1	0	1	1	1
3	0	1	0	1	1	1	1	1
4	0	1	1	1	0	1	1	1
5	1	0	0	0	1	0	1	0
6	1	0	1	0	0	0	0	0
7	1	1	0	0	1	1	1	1
8	1	1	1	0	0	1	0	0

KNF má potom tvar (využívame len tri riadky s nulovou výslednou funkčnou hodnotou)

$$f(x_1,x_2,x_3)=(\overline{x}_1+x_2+x_3)\cdot(\overline{x}_1+x_2+\overline{x}_3)\cdot(\overline{x}_1+\overline{x}_2+\overline{x}_3)$$

9.4 Spínacie obvody

Mnohé elektronické zariadenia, akými sú napr. počítače, telefónne ústredne, zariadenia na riadenie dopravy, obsahujú ako časť spínancie obvody. Ich teória bola vypracovaná v r. 11038 C. Shannonom v rámci jeho MSc. dizertacie [xx]. Spínač môže byť chápaný ako taký spoj v obvode, ktorý ak je uzavretý, potom ním prechádza elektrický prúd, v opačnom prípade, ak je otvorený, elektrický prúd ním neprechádza. Spínač môžeme znázorniť takto:



(9.10)

Predpokladajme , že v spínacom obvode máme spínač A. Stav tohto spínača označíme premennou x, ak $x = 1$ ($x = 0$), potom spínač A je uzavretý (otvorený).

O trochu zložitejší prípad spínacieho obvodu obsahuje dva spínače A_1 a A_2

(9.11)

Hovoríme, že v tomto prípade sú spínače zapojené *sériovo*. Nech x_1 a x_2 sú premenné popisujúce stavy spínačov A_1 resp. A_2, tieto premenné ak sa rovnajú 1 (0) , potom daný spínač je uzavretý (otvorený). Nech $f(x_1,x_2)$ je funkcia, ktorej hodnota sa rovná 1 (0) pre tie hodnoty x_1 a x_2, ktoré umožňujú (znemožňujú) tok prúdu. Táto funkcia môže byť chápaná ako binárna funkcia $f:\{0,1\}^2 \to \{0,1\}$, ktorej funkčné hodnoty sú určené tabuľkou

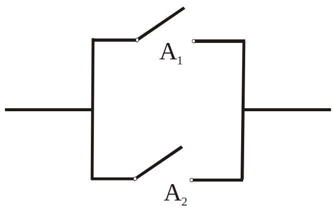
#	x_1	x_2	$f(x_1,x_2)$
1	0	0	0
2	0	1	0
3	1	0	0
4	1	1	1

Z tejto tabuľky vyplýva, že funkcia f je vyjadrená ako súčin premenných x_1 a x_2

$$f(x_1,x_2) = x_1x_2 \tag{9.12}$$

Táto funkcia je definovaná nad Boolovou algebrou $(\{0,1\},+,\cdot,\mathbf{G},0,1)$, t. j. premenné patria do množiny {0,1}.

Nový druh spínacieho obvodu, ktorý je podobný sériovému obvodu (8.11), ale v paralelnom zapojení, má tvar



(9.13)

Podobne ako v predchádzajúcom príklade (9.11), nech spínače A_1 a A_2 sú popísané premennými x_1 a x_2 , tento obvod je popísaný funkciou $g(x_1,x_2)$ nad Boolovou algebrou $(\{0,1\},+,\cdot,\mathbf{G},0,1)$, ktorá je špecifikovaná tabuľkou

#	x_1	x_2	$g(x_1,x_2)$
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	1

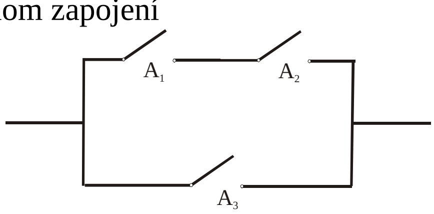
Potom funkcia $g(x_1,x_2)$ je špecifikovaná ako súčet premenných

$$g\left(x_1,x_2\right)=x_1+x_2 \tag{9.14}$$

ktorá je definovaná nad Boolovou algebrou $\left(\{0,1\},+,\cdot,\mathbf{G},0,1\right)$

Funkcie, ktoré sú podobné (9.12) a (9.14) popisujú vlastnosti spínacieho obvodu pomocou stavov spínačov, ktoré sú súčasťou daného obvodu. Takéto funkcie sa nazývajú ***spínacie funkcie***. Majme n spínačov, ktorých stavy sú špecifikované premennými x_1, x_2, ..., x_n. Spínacia funkcia $f:\{0,1\}^n \to \{0,1\}$ popisuje správanie sa spínacieho obvodu pre všetky možné 2^n stavy spínačov. Ako už bolo ukázané na predchádzajúcich ilustračných príkladoch, funkcia f môže byť reprezentovaná Boolovou formulou a teda aj Boolovou funkciou.

Nasledujúci príklad spínacieho obvodu bude zložitejší spínací obvod, ktorý obsahuje tri spoínače v sériovo-paralelnom zapojení

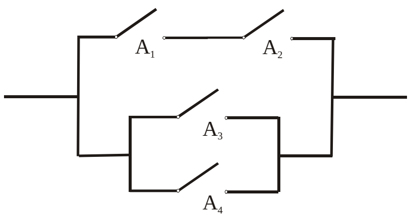


(9.15)

Nech jednotlivé spínače A_1, A_2 a A_3 sú špecifikované premennými x_1, x_2 resp. x_3. Nech funkcia $f\left(x_1,x_2\right)$ popisuje vlastnosti hornej časti obvodu, ktorý obsahuje dva sériovo zapojené spínače A_1 a A_2. Pomocou predchádzajúceho príkladu (9.11) funkcia, ktorá špecifikuje takýto obvod má tvar $f_1\left(x_1,x_2\right)=x_1x_2$. Celkový obvod potom môžeme zložiť z dvoch paralelných podštruktúr, horná je reprezentovaná funkciou $f_1\left(x_1,x_2\right)=x_1x_2$ a dolná je reprezentovaná funkciou $f_2\left(x_3\right)=x_3$. Spojením týchto dvoch funkcií pomocou (9.14) dostaneme Boolovu funkciu celého spínacieho obvodu (9.15)

$$f\left(x_1,x_2,x_3\right)=f_1\left(x_1,x_2\right)+f_2\left(x_3\right)=x_1x_2+x_3 \tag{9.16}$$

Príklad 9.12. Zostrojte spínaciu funkciu f spínacieho zariadenia

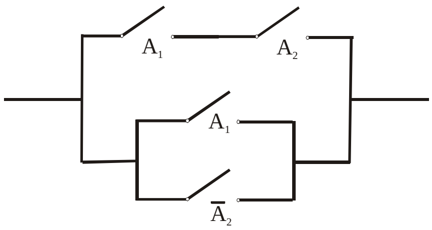


(9.17)

Nech x_1,x_2,x_3 a x_4 sú premenné označujúce stavy spínačov A_1,A_2,A_3 resp. A_4. Potom celková spínacia funkcia zariadenia má tvar $f\left(x_1,x_2,x_3,x_4\right)$. V prvom kroku určíme pomocné spínacie funkcie $f_1\left(x_1,x_2\right)$ a $f_2\left(x_3,x_4\right)$, ktoré sú priradené hornej časti obsahujúcej spínače A_1,A_2 resp. dolnej časti obsahujúcej spínače A_3,A_4. Tieto funkcie sú určené spínacími funkciami (9.12) a (9.14), $f_1\left(x_1,x_2\right)=x_1x_2$ resp. $f_2\left(x_3,x_4\right)=x_3+x_4$. Hľadaná spínacia funkcia má potom tvar

$$f\left(x_1,x_2,x_3,x_4\right)=f_1\left(x_1,x_2\right)+f_2\left(x_3,x_4\right)=x_1x_2+x_3+x_4 \tag{9.18}$$

Príklad 9.13. Zostrojte spínaciu funkciu f zariadenia, ktoré vznikne malou modifikáciou spínacieho zariadenia (9.17)



(9.110)

kde pôvodné spínače A_1 a A_3 sú teraz spolu spriahnuté, t. j. obe sú súčasne zapnuté alebo vypnuté (preto sú obe označené A_1). Ďalšie dva pôvodné spínače A_2 a A_4 sú teraz spolu taktiež spriahnuté, ale opačným spôsobom, t. j. ak je jeden spínač zapnutý, druhý je vypnutý a naopak (preto sú oba označené A_2 a $\overline{A}_2$). Spínaciu funkciu takto špecifikovaného zariadenia ľahko zostrojíme pomocou spínacej funkcie (9.18) pôvodného spínacieho zariadenia, keď položíme $x_3 = x_1$ a $x_4 = \overline{x}_2$

$$g(x_1,x_2) = f(x_1,x_2,x_1,\overline{x}_2) = x_1x_2 + x_1 + \overline{x}_2 \tag{9.20}$$

Tabuľka funkčných hodnôt tejto spínacej funkcie má tvar

x_1	x_2	$\overline{x}_2$	x_1x_2	$x_1x_2 + x_1 + \overline{x}_2$
0	0	1	0	1
0	1	0	0	0
1	0	1	0	1
1	1	0	1	1

Ak použime vetu 8.6, zostrojíme Boolovu funkciu v tvare KNF, ktorá simuluje túto tabuľku, z ktorej si vyberieme riadok s nulovou funkčnou hodnotou, potom $f(x_1,x_2) = x_1 + \overline{x}_2$. Spínacie zariadenie (8.110) má túto zjednodušenú spínaciu funkciu

$$g(x_1,x_2) = x_1 + \overline{x}_2 \tag{9.21}$$

Ľahko dokážeme, že funkcie (9.20) a (9.21) sú ekvivalentné

$$g(x_1,x_2) = x_1x_2 + x_1 + \overline{x}_2 = \left(\underbrace{x_2 + 1}_{1}\right)x_1 + \overline{x}_2 = 1x_1 + \overline{x}_2 = x_1 + \overline{x}_2$$

Príklad 9.12. Budeme riešiť veľmi praktickú úlohu, ktorá pre mnohých z nás je záhadou ako vlastne funguje. Predstavme si schodište, na začiatku a konci ktorého sú umiestnené stenové vypínače S_1 a S_2, pomocou ktorých zapneme alebo vypneme svetlo nad schodišťom. Hlavná požiadavka je taká, aby sa na jednom konci mohlo svetlo buď vypnúť, ak na druhom konci je zapnuté, alebo zapnúť , ak je na druhom konci vypnuté. Túto podmienku môžeme formulovať alternatívne tak, že ak sú oba spínače S_1 a S_2 vypnuté alebo zapnuté, potom zariadením nepreteká prúd, ale stačí, aby bolo zapnuté práve jedno, potom zariadením preteká prúd, čo môžeme vyjadriť touto tabuľkou

S_1	S_2	prúd
zapnuté	zapnuté	nie
zapnuté	vypnuté	áno
vypnuté	zapnuté	áno
vypnuté	vypnuté	nie

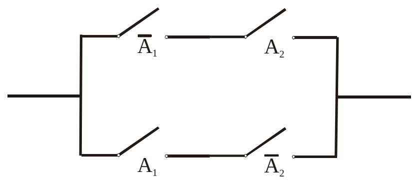
Predpokladajme, že vypínače S_1 a S_2 sú realizovaný pomocou dvoch spínačov A_1 a A_2. Ak použijeme premenné x_1 a x_2 , ktoré označujú stavy spínačov A_1 a A_2 , potom vyššie uvedená tabuľka môže byť prepísaná do tvaru

x_1	x_2	$f(x_1,x_2)$
0	0	0
0	1	1
1	0	1
1	1	0

Podľa vety 9.5, táto tabuľka špecifikuje Boolovu funkciu v DNF

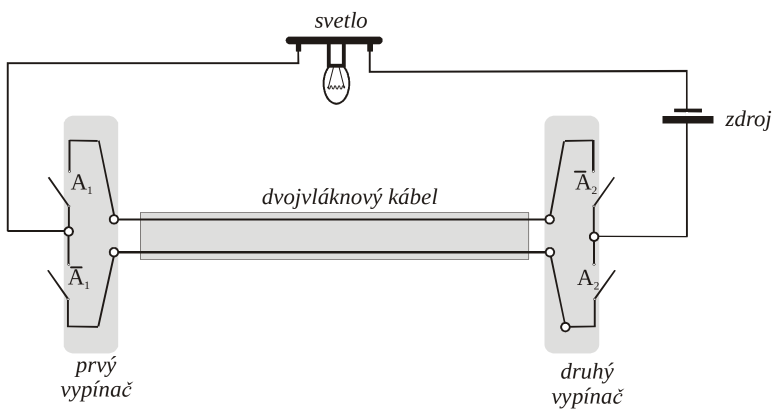
$$f\left(x_1,x_2\right)=\overline{x}_1x_2+x_1\overline{x}_2 \tag{9.22}$$

Spínací obvod so spínacou funkciou takto špecifikovanou má tvar



(9.23)

Diagram (9.26) znázorňuje realizáciu tohto spínacieho obvodu, kde vytieňované oblasti tvoria stenové vypínače, ktoré sú spojené dvojvláknovým káblom.



(9.26)

Tento jednoduchý príklad demonštruje užitočnosť teórie spínacích obvodov vybudovanej pomocou Boolových funkcií. Vyriešili sme praktický príklad, ako realizovať zapínanie a vypínanie svetla na schodišti (alebo na dlhej chodbe) tak, že každým vypínačom môžeme svetlo zapnúť alebo vypnúť, nezávisle od polohy druhého vypínača.

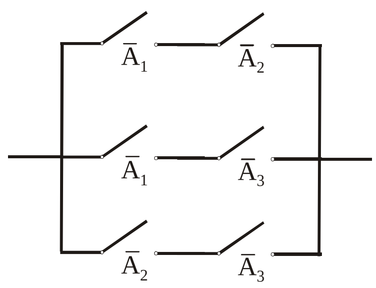
Príklad 9.13. Ústredné kúrenie v rodinnom dome je riadené troma termostatmi, ktoré sú umiestnené v každej izbe domu. Termostaty sú nastavené na 18 °C, pričom z dôvodu šetrenia energiou sa požaduje, aby systém ústredného kúrenia bol zapnutý len ak teplota aspoň v dvoch izbách je menšia ako 18 °C, v opačnom prípade je systém vypnutý. Navrhnite spínačový systém, ktorý prijíma signály z termostatov a ktorý riadi ústredné kúrenie. Pokúste sa minimalizovať navrhnutý systém, aby bol čo najjednoduchší.

x_1	x_2	x_3	$F(x_1,x_2,x_3)$
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

Boolova funkcia špecifikovaná touto tabuľkou má tvar

$$\begin{aligned} F(x_1,x_2,x_3) &= \overline{x}_1\overline{x}_2\overline{x}_3 + \overline{x}_1\overline{x}_2 x_3 + \overline{x}_1 x_2\overline{x}_3 + x_1\overline{x}_2\overline{x}_3 = \\ &= \underbrace{\overline{x}_1\overline{x}_2\overline{x}_3 + \overline{x}_1\overline{x}_2\overline{x}_3 + \overline{x}_1\overline{x}_2\overline{x}_3}_{\overline{x}_1\overline{x}_2\overline{x}_3} + \overline{x}_1\overline{x}_2 x_3 + \overline{x}_1 x_2\overline{x}_3 + x_1\overline{x}_2\overline{x}_3 = \\ &= \overline{x}_1\overline{x}_2\underbrace{(\overline{x}_3 + x_3)}_{1} + \overline{x}_1\underbrace{(x_2 + \overline{x}_2)}_{1}\overline{x}_3 + \underbrace{(\overline{x}_1 + x_1)}_{1}\overline{x}_2\overline{x}_3 \\ &= \overline{x}_1\overline{x}_2 + \overline{x}_1\overline{x}_3 + \overline{x}_2\overline{x}_3 \end{aligned}$$

Potom spínací obvod pre automatické zapínanie a vypínanie ústredného kúrenia má podobu



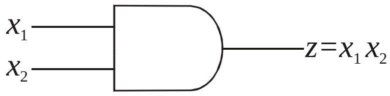
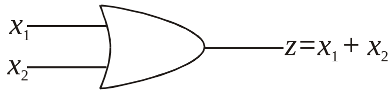
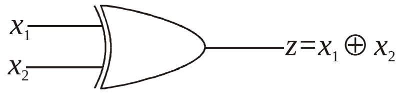
9.5 Logické obvody

Na tomto mieste je vhodné pripomenúť[1] si pioniersku prácu McCullocha a Pittsa z r. 11043, v ktorej boli formulované teoretické základy neurónových sietí s logickými neurónmi a ktorá je v súčasnosti pokladaná za jednu prvých prác, na základe ktorých vznikol nový informatický vedný odbor ***umelá inteligencia***. Autori dokázali, že ľubovoľná Boolova funkcia môže byť simulovaná pomocou obvodu (neurónovej siete) obsahujúcej elementárne obvody – neuróny, ktoré simulujú disjunktívne a konjunktívne klauzuly. V prípade logických obvodov sa jedná o abstrakciu, ktorá ignoruje vnútornú architektúru neurónov a postuluje existenciu elementárnych logických brán pre operácie disjunkcie, konjunkcie a negácie.

V tejto kapitole sa budeme zaoberať logickými obvodmi, ktoré tvoria základné funkčné jednotky v počítačoch. Logické obvody obsahujú logické brány typu disjunkcie, konjunkcie a negácie, pozri tab. A.1.

Tabuľka 9.1. Logické brány

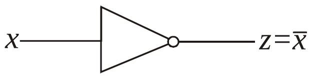
[1] Pozri náš učebný text *Matematická logika*, kapitola 4.

logická brána konjunkcie	Logická brána disjunkcie	Logická brána exkluzívnej disjunkcie
		

x_1	x_2	x_1x_2
0	0	0
0	1	0
1	0	0
1	1	1

x_1	x_2	x_1+x_2
0	0	0
0	1	1
1	0	1
1	1	1

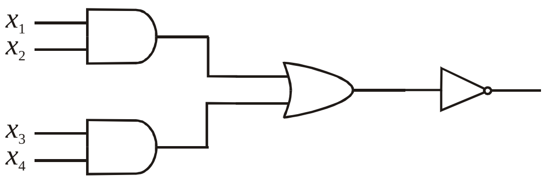
x_1	x_2	x_1+x_2
0	0	0
0	1	1
1	0	1
1	1	0

logická brána negácie


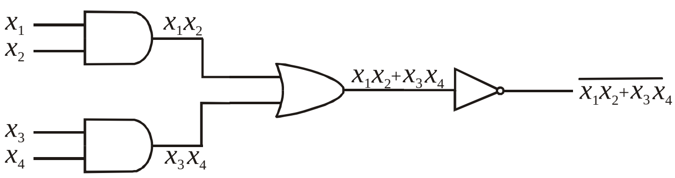
x_1	$\overline{x}_1$
0	1
1	0

Konjunktívna a disjunktívna brána má dva binárne vstupy a jeden binárny výstup, jednoduchšia brána negácie má jeden binárny vstup a jeden binárny výstup.

Príklad 9.14. Zostrojte Boolovu funkciu pre logický obvod



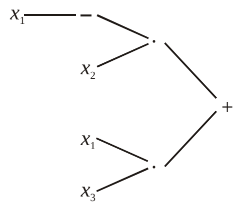
Použitím tabuľky A.1 jednotlivé spoje tohto logického obvodu ohodnotíme takto



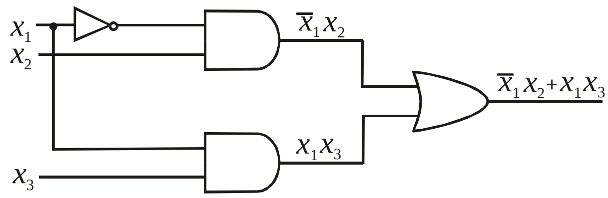
To znamená, že Boolova funkcia priradená tomuto obvodu má tvar

$$f\left(x_1,x_2,x_3,x_4\right)=\overline{x_1x_2+x_3x_4}$$

Príklad 9.15. Zostrojte logický obvod, ktorý simuluje Boolovu funkciu $f\left(x_1,x_2,x_3,x_4\right)=\overline{x}_1x_2+x_1x_3$. Pre ilustráciu zostrojíme syntaktický strom tejto funkcie



Logický obvod zostrojíme jednoducho z tohto syntaktického stromu, ktorý interpretuje Boolovu funkciu $f\left(x_1,x_2,x_3,x_4\right)=\bar{x}_1x_2+x_1x_3$, tak, že jednotlivé vrcholy znázorňujúce algebraické operácie nahradíme príslušnými logickými bránami



9.5.1 Sumátor dvoch binárnych čísel (polosumátor)

Technika logických obvodov je aplikovateľná ku konštrukcii súčtu dvoch kladných celých čísel, ktoré sú prezentované v binárnej reprezentácií. Táto konštrukciu obsahuje tri etapy. *Prvá etapa* spočíva v návrhu logického obvodu (nazývaného *polosumátor*) ktorý sčíta dve jednobitové čísla *x* a *y*

$$\begin{array}{r} x \\ y \\ \hline c\ \ s \end{array} \tag{9.27}$$

Uvedieme tabuľku všetkých prípadov tejto schémy, ktoré môžu nastať

vstup		výstup	
x	y	c	s
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

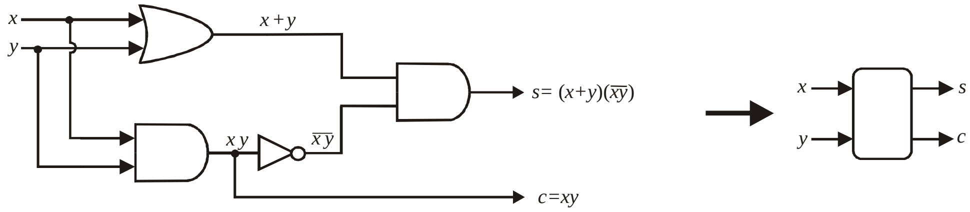
(9.28)

Ak použijeme vetu 9.5 výstupné premenné z tejto tabuľky sú určené pomocou sumácie klauzúl (v DNF tvare)

$$s=f\left(x,y\right)=\bar{x}y+x\bar{y}=\left(x+y\right)\left(\bar{x}+\bar{y}\right)=\left(x+y\right)\left(\overline{xy}\right) \tag{9.210a}$$

$$c=g\left(x,y\right)=xy \tag{9.210b}$$

kde pri konštrukcii alternatívnej pravej strany Boolovej funkcie f bol použitý distributívny zákon.



(9.30)

9.5.2 Sumátor troch binárnych čísel (dvojitý sumátor)

*K*onštrukcii logického obvodu (nazývaného *dvojitý sumátor*) pre sčítanie troch jednobitových čísel

$$\begin{array}{r} x \\ y \\ \underline{z} \\ u\,v \end{array} \qquad (9.31)$$

Všetky možné prípady tejto schémy sú uvedené v tabuľke

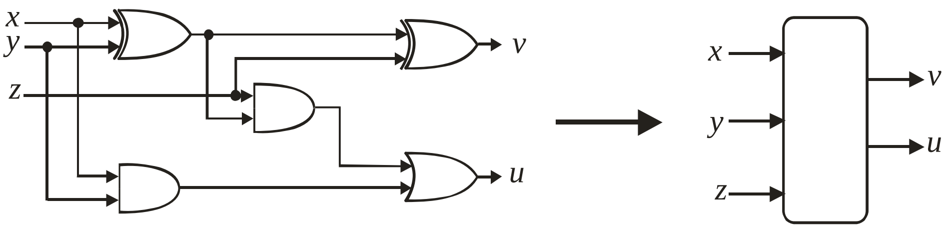
vstup			výstup	
x	y	z	u	v
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

(9.32)

Pomocou formuly (9.6) môžeme zostrojiť z tejto tabuľky DNF Boolovu funkciu pre výstupy u a v

$$\begin{aligned} u &= \overline{x}yz + x\overline{y}z + xy\overline{z} + xyz = \left(\overline{x}y + x\overline{y}\right)z + xy\left(z + \overline{z}\right) \\ &= \left(x \oplus y\right)z + xy \\ v &= \overline{x}\,\overline{y}z + \overline{x}y\overline{z} + x\overline{y}\overline{z} + xyz = \left(\overline{x}y + x\overline{y}\right)\overline{z} + \left(\overline{x}\,\overline{y} + xy\right)z \\ &= \left(x \oplus y\right) \oplus z \end{aligned} \qquad (9.33)$$

Výstupné veličiny u a v môžu byť reprezentované pomocou spoločnečj Boolovej funkcie, ktorá je reprezentované obvodom



(9.34)

9.6 Optimalizácia logických obvodov

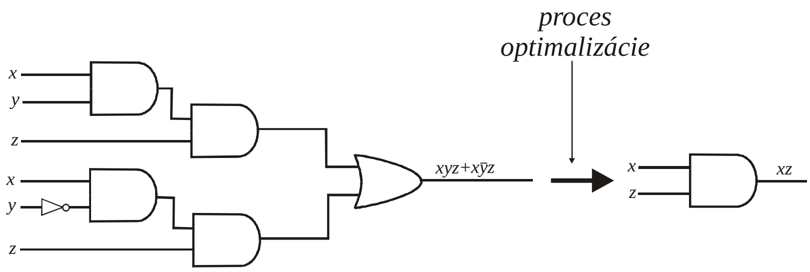
Efektívnosť logických obvodov závisí na počte a prepojení jej logických brán. Proces návrhu logického obvodu začína návrhom tabuľky špecifikujúcej Boolovu funkciu, ktorá transfomuje vstupné binárne veličiny na výstupné binárne veličiny. Boolova funkcia je zostrojená podľa vety 9.5 pomocou súčtu konjunktívnych klauzúl. Aj keď je tento prístup ku konštrukcii Boolovej funkcie všeobecný, veta 9.5 nezaručuje optimálnosť zostrojenej funkcie. Pod optimálnosťou rozumieme to, že Boolova funkcia, ktorá simuluje danú tabuľku funkčných hodnôt pre všetky kombinácie vstupných hodnôt, má minimálny počet literálov.

Uvažujme logický obvod, ktorý má výstup 1 vtedy a len vtedy, ak $x = y = z = 1$ alebo ak $x = z = 1$ a $y = 0$. Boolova funkcia zostrojená podľa vety 9.5, ktorá simuluje tento logický obvod má tvar $f(x,y,z)=xyz+z\overline{y}z$, môže byť podstatne zjednodušená takto

$$f(x,y,z)=xyz+x\overline{y}z=x(y+\overline{y})z=x\cdot 1\cdot z=xz$$

To znamená, že táto funkcia xz , ktorá obsahuje dva literály, je ekvivalentná s pôvodnou funkciou obsahujúcej 6 literálov; môžeme teda povedať, že optimálny tvar Boolovej funkcie $f(x,y,z)=xyz+z\overline{y}z$ je nová funkcia $g(x,z)=xz$, ktorá aj keď je s pôvodnou ekvivalentná, má podstatne menej literálov.

Tento jednoduchý príklad dostatočne jasne ukazuje dôležitosť hľadať optimálny tvar Boolovej funkcie, pôvodne zostrojene podľa vety 9.5, ak má táto slúžiť ako podklad pre návrh logického obvodu. Optimálny tvar Boolovej funkcie môže v špeciálnych prípadoch podstatné zjednodušiť navrhovaný logický obvodu. Ako ilustračný príklad budeme študovať logické obvody priradené Boolovej funkcii $f(x,y,z)=xyz+z\overline{y}z$ a jej optimálnej formy $g(x,z)=xz$



(9.38)

Logický obvod zostrojený pomocou optimálneho tvaru (vpravo), obsahujúceho minimálny počet literálov je podstatne jednoduchší ako logický obvod (vľavo), ktorý obsahuje šesť logických hradiel.

9.6.1 Quinova a McCluskeyho optimalizačná metóda [2,5,6]

Táto metóda patrí medzi často pouižívané prístupy k optimalizácii Boolových funkcií, jej hlavnou prednosťou pred ostatnými prístupmi je jej konceptuálna jednoduchosť a priamočiara algoritmizovateľnosť.

Quinovu a McCluskeyho metóda [2,5,6] bude ilustrovaná konkrétnym prípadom optimalizácie jednoduchej Boolovej funkcie

$$f(x,y,z)=xyz+x\overline{y}z+\overline{x}yz+\overline{x}\,\overline{y}z+\overline{x}\,\overline{y}\,\overline{z} \tag{9.310}$$

Každá súčinová klauzula (pozri definícia A.6 a formulu (9.3)) môže byť reprezentovaná bitovým reťazcom $\boldsymbol{e}=(e_1,e_2,e_3)\in\{0,1\}^3$

$$l_{\boldsymbol{e}}(x,y,z)=x^{e_1}y^{e_2}z^{e_3}\rightarrow(e_1,e_2,e_3) \tag{9.40}$$

kde $\xi^e=\xi$, ak $e=1$, $\xi^e=\overline{\xi}$, ak $e=0$, pre $\xi=x,y,z$. Pre takto definovanú binárnu reprezentáciu môžeme použiť metriku Hammingovej vzdialenosti ku kvantifikácii podobnosti medzi binárnymi vektormi. Nech $\boldsymbol{e}_i=\left(e_1^{(i)},e_2^{(i)},\dots,e_n^{(i)}\right)$ a $\boldsymbol{e}_j=\left(e_1^{(j)},e_2^{(j)},\dots,e_n^{(j)}\right)$ sú dve binárne reprezentácie klauzúl, potom

$$d_H(\boldsymbol{e}_i,\boldsymbol{e}_j)=\sum_{k=1}^{n}\left|e_k^{(i)}-e_k^{(j)}\right| \tag{9.41}$$

Táto vzdialenosť pre binárne vektory nám špecifikuje počet polôh v ktorých sa binárne vektory vzájomne odlišujú. Napríklad, ak Hammingova vzdialenosť medzi dvoma binárnymi vektormi je 2, potom tieto vektory sa navzájom odlišujú na dvoch miestach binárneho reťazca.

Každá Boolova formula v DNF forme je 1-1-značne reprezentovaná pomocou množiny bitových reťazcov, ktoré sú priradené jednotlivým klauzulám študovanej formuly. Pre ilustračný príklad (A.310) táto množina obsahuje 5 binárnych reťazcov dĺžky 3

$$U_f = \{(111),(101),(011),(001),(000)\} \tag{9.42}$$

Táto možnosť reprezentovať Boolovu funkciu v DNF forme pomocou množiny binárnych vektorov vyplýva zo skutočnosti, že operácia sumácie je komutatívna a asociatívna, čiže nezáleží v akom poradí sčítame jednotlivé klauzuly v Boolovej funkcii.

Dve klauzuly z množiny U_f môžu byť vzájomne sčítané do jednej klauzuly vtedy a len vtedy ak sa líšia ich binárne reťazce práve v jednej polohe, čiže ak ich vzájomná Hammingova vzdialenosť sa rovná 1. Napríklad, z množiny (9.42) vyberieme prvú a druhú klauzulu, ich binárne reprezentácie (111) a (101) sa líšia len hodnotou binárnej premennej v druhej polohe, $d_H(111,101)=1$. Tieto dve klauzuly sú sčítané takto

$$xyz + x\bar{y}z = x\left(\underbrace{y+\bar{y}}_{1}\right)z = xz \tag{9.43a}$$

V binárnej reprezentácii tento proces zjednodušenia formálne vyjadríme takto

$$(111)+(101) = sum\left((111),(101)\right) = (1\#1) \tag{9.43b}$$

kde bol použitý nový „*prázdny*“ symbol ´#´, ktorý reprezentuje prázdne miesto v binárnej reprezentácii novej klauzuly *xz*, pozri (A.43a). Takto zostrojené nové klauzuly obsahujúce jeden symbol ´#´ tvoria množinu

$$U_f^{(1)} = \{(1\#1),(\#11),(\#01),(0\#1),(00\#)\} \tag{9.44}$$

V ďalšej etape vytvárame z množiny $U_f^{(1)}$ novú množinu $U_f^{(2)}$, ktorá obsahuje klauzuly s dvoma prázdnymi symbolmi ´#´ a ktoré boli vytvorené operáciou súčtu klauzúl z množiny $U_f^{(1)}$

$$U_f^{(2)} = \{(\#\#1)\} \tag{9.45}$$

Proces sčítania klauzúl obsahujúcich symboly ´#´ musí byť podrobnejšie špecifikovaný:

(a) Sčítať môžeme len také dve klauzuly, ktoré obsahujú rovnaký počet symbolov ´#´, pričom tieto symboly v oboch použitých klauzulách musia byť umiestnené v rovnakých polohách v oboch binárnych reprezentáciách.

(b) Klauzuly, ktoré vyhovujú podmienke (a) môžeme sčítať len vtedy, ak ich binárne komponenty sa líšia len v jednej polohe.

Uvažujme dve klauzuly $\boldsymbol{e}_i = \left(e_1^{(i)}, e_2^{(i)}, ..., e_n^{(i)}\right)$ a $\boldsymbol{e}_j = \left(e_1^{(j)}, e_2^{(j)}, ..., e_n^{(j)}\right)$, ktoré chceme sčítať. Podľa podmienky (a) musí existovať taká množina indexov $I \subseteq \{1,2,...,n\}$, že

$$(\forall k \in I)\left(e_k^{(i)} = e_k^{(j)} = \#\right) \text{ a } (\forall k \notin I)\left(e_k^{(i)}, e_k^{(j)} \in \{0,1\}\right) \tag{9.46}$$

Ako príklad uvedieme dvojicu klauzúl $\boldsymbol{e}_i = (10\#0\#11)$ a $\boldsymbol{e}_j = (11\#0\#10)$, pre množinu indexov $I = \{3,5\}$ platia obe podmienky. V prípade, že takáto množina neexistuje, potom klauzuly $\boldsymbol{e}_i = \left(e_1^{(i)}, e_2^{(i)}, ..., e_n^{(i)}\right)$ a $\boldsymbol{e}_j = \left(e_1^{(j)}, e_2^{(j)}, ..., e_n^{(j)}\right)$ nemôžu byť použité v procese sčítania klauzúl. Zavedieme zovšeobecnenú Hammingovu vzdialenosť pre klauzuly, ktoré vyhovujú podmienke (a)

$$d_H\left(\boldsymbol{e}_i,\boldsymbol{e}_j\right)=\sum_{k\notin I}\left|e_k^{(i)}-e_k^{(j)}\right| \qquad (9.47)$$

t. j. v sumácii sú aktívne len binárne členy. Potom podmienka (b) požaduje, že Hammingova vzdialenosť medzi klauzulami musí byť rovná 1. Bez zníženia všeobecnosti môžeme postulovať, že Hammingova vzdialenosť medzi vektormi $\boldsymbol{e}_i=\left(e_1^{(i)},e_2^{(i)},...,e_n^{(i)}\right)$ a $\boldsymbol{e}_j=\left(e_1^{(j)},e_2^{(j)},...,e_n^{(j)}\right)$, ktoré nevyhovujú podmienke (a) (majú buď rôzny počet symbolov symbol ´#´, alebo ak majú rovnaký počet týchto symbolov, potom ich polohy nie sú rovnaké) je nekonečne veľká. Týmto jednoduchým predpokladom máme zabezpečené, že podmienka (a) je splnená.

Zavedieme operátor $\mathcal{A}$, ktorý špecifikuje prechod množiny $U_f^{(k)}$ na množinu $U_f^{(k+1)}$

$$U_f^{(k+1)}=\mathcal{A}\left(U_f^{(k)}\right) \qquad (9.48)$$

Musí existovať také kladné celé číslo *n*, že tento proces tvorby nových množín je ukončený, t. j. platí $U_f^{(n+1)}=\mathcal{A}\left(U_f^{(n)}\right)=\varnothing$. Rekurentný proces (9.48) je inicializovaný množinou $U_f^{(0)}=U_f$. Môžeme hovoriť o etapách procesu tvorby nových klauzúl z pôvodnej (počiatočnej) množiny klauzúl. V 1. etape vytvoríme procesom sčítania dvoch klauzúl z množiny $U_f^{(0)}=U_f$ klauzuly s jedným prázdnym symbolom ´#´, v 2. etape vytvoríme z množiny $U_f^{(1)}$ klauzuly s dvoma symbolmi #. Tento rekuretný proces je ukončený vtedy, ak operátor $\mathcal{A}$ aplikovaný na množinu $U_f^{(n)}$ produkuje prázdnu množinu, t. j. $\mathcal{A}\left(U_f^{(n)}\right)=\varnothing$.

Ako ilustračný príklad rekurentnej tvorby množín $U_f^{(k)}$ použijeme množinu (A.42), výsledky je možné uviesť vo forme tabuľky

0. etapa			1. etapa				2. etapa		
1	(111)		1	(1,2)	(1#1)		1	(1,3), (2,4)	(##1)
2	(101)		2	(1,3)	(#11)				
3	(011)		3	(2,4)	(#01)				
4	(001)		4	(3,4)	(0#1)				
5	(000)		5	(4,5)	(00#)				

V stĺpcoch pre prvú a druhú etapu sú uvedené aj dvojice indexov klauzúl z predchádzajúceho stĺpca, ktoré boli použité v sumačnom procese.

Stojíme pred úlohou, ako vybrať taký minimálny počet klauzúl zostrojených v prvej alebo v ďalších etapách, ktoré sú odvoditeľné zo všetkých pôvodných klauzúl z množiny $U_f^{(0)}=U_f$. K presnej špecifikácii tohto posledného kroku Quinovej a McCluskyeho metódy musíme zaviesť nový pojem „pokrytie“. Klauzula ***e´ pokrýva*** klauzulu ***e***, formálne $\boldsymbol{e}'\subseteq\boldsymbol{e}$, vtedy a len vtedy, ak platí pre každé *i*=1,2,...,*n* práve jedna z týchto podmienok

1. $\left(e_i'=1\right)\Rightarrow\left(e_i=1\right)$
2. $\left(e_i'=0\right)\Rightarrow\left(e_i=0\right)$

(9.410)

3. $\left(e_i'=\#\right)\Rightarrow\left(e_i\in\{0,1,\#\}\right)$

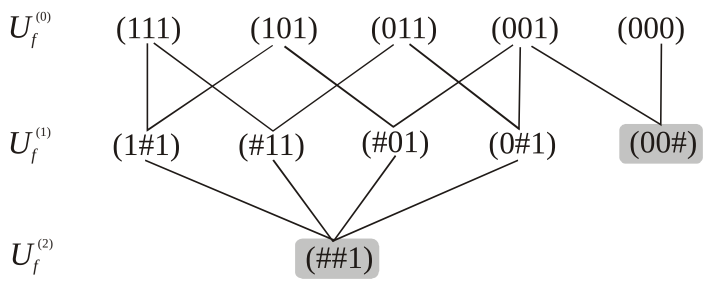
Pre ilustráciu tejto relácie uvedieme jednoduchý ilustračný príklad ktorý vyhovuje týmto podmienkam

0	1	#	1	#	0	1	0	1	***e***
0	1	#	1	#	#	1	0	#	***e′***

Ľahko sa presvedčíme, že sa jedná o ***reláciu čiastočného usporiadania*** nad množinou vektorov $\boldsymbol{e} \in \{0,1,\#\}^n$ (pozri knihu [xx], kapitolu 3.2). Potom množina klauzúl, ktorá vznikne zjednotením množín $U_f^{(0)}$, $U_f^{(1)}$, $U_f^{(3)}$,...

$$\tilde{U}_f = U_f^{(0)} \cup U_f^{(1)} \cup U_f^{(2)} \cup$$

je čiastočne usporiadaná a diagramatický znázornená Hasseho diagramom



Z tohto diagramu vyplýva, že má 5 maximálnych klauzúl (klauzuly patriace do množiny $U_f^{(0)}$) a dve minimálne klauzuly (##1) a (00#), ktoré sú na Hasseho diagrame vysvietené.

Pre každú klauzulu ***e*** obsahujúcu aspoň jeden prázdny symbol, $e \in U_f^{(1)} \cup U_f^{(2)} \cup \ldots$, zostrojíme množinu $U(\boldsymbol{e}) \subseteq U_f$, ktorá obsahuje všetky pôvodné klauzuly (neobsahujúce prázdne symboly #), ktoré sú pokryté klauzulou ***e***

$$U(\boldsymbol{e}) = \left\{ \boldsymbol{e}' ; \left(\boldsymbol{e}' \in U_f \right) \wedge \left(\boldsymbol{e} \subseteq \boldsymbol{e}' \right) \right\} \tag{9.50}$$

Pre lepšie pochopenie tejto množinovej koncepcie uvedieme príklady tejto množiny vychádzajúce z vyššie uvedeného Hassovho diagramu

$$U(1\#1) = \{(111),(101)\}$$
$$U(\#11) = \{(111),(011)\}$$
$$U(\#01) = \{(101),(001)\}$$
$$U(0\#1) = \{(011),(001)\}$$
$$\boxed{U(00\#)} = \{(001),(000)\}$$
$$\boxed{U(\#\#1)} = \{(1,1,1),(101),(011),(001)\}$$

kde posedné dve klauzuly sú označené ako minimálne

Našim cieľom je vybrať také minimálne klauzuly, ktoré pokrývajú pôvodné klauzuly z množiny $U_f^{(0)}$. Množinu týchto minimálnych klauzúl označíme V, potom v rámci tejto množiny hľadáme takú podmnožinu $\tilde{V} \subseteq V$, ktorej klauzuly plne pokrývajú množinu $U_f^{(0)}$

$$\bigcup_{e \in \tilde{V}} U(e) = U_f \tag{9.51}$$

Podmnožina je určená podmienkou minimálnosti počtu literálov, $[\tilde{V}]$, ktoré obsahuje

$$\tilde{V} = arg \min_{V' \subseteq V} [V'] \tag{9.52}$$

Riešenie tohto optimalizačného problému je pre malý počet premenných (cca do päť) obvykle zvládnuteľný ručne tak, že preberieme všetky možnosti, ktoré pokrývajú množinu U_f. Pre väčšie problémy môže byť použitá metóda spätného prehľadávania, ktorá systematicky preskúma všetky možnosti. Žiaľ, tento prístup je nepoužiteľný pre niekoľko desiatok premenných, v dôsledku exponenciálneho rastu zložitosti. V praxi sa používajú rôzne heuristické metódy, ktoré poskytujú kvalitne suboptimálne riešenie, ktoré je často rovné optimálnemu riešeniu. V ďalšej časti tejto kapitoly budeme diskutovať veľmi jednoduchú metódu uskutočnenia optimálneho pokrytia pôvodných (maximálnych klauzúl bez prázdnych symbolov #), ktorú zo súčasného pohľadu algoritmov môžeme nazvat "greedy" metóda [xx].

V tomto konkrétnom príklade sa jedná o extrémne jednoduchý problém, môžeme vybrať obe minimálne klauzuly (##1) a (00#), ktoré pokrývajú pôvodné klauzuly. Potom môžeme písať Boolovu funkciu (7.310) v ekvivalentnom tvare

$$f\left(x,y,z\right)=z+\overline{x}\,\overline{y}$$

čo reprezentuje podstatné zjednodušenie (optimalizáciu) pôvodnej Boolovej funkcie (7.310), ktorej počet literálov z 15 klesol na 3.

Príklad 9.15. Quinova a McCluskeyho metóda pôvodne nevyužívala algoritmus spätného prehľadávanie, bola založená na heuristickom prístupe naformulovanom v podobe tabuliek a ktorý v súčasnosti nazývame „greedy" metóda . Uvažujme Boolovu funkciu

$$f\left(w,x,y,z\right)=wxy\overline{z}+w\overline{x}yz+\overline{w}xyz+w\overline{x}y\overline{z}+\overline{w}x\overline{y}z+\overline{w}\overline{x}yz+\overline{w}\,\overline{x}\,\overline{y}z$$

V nasledujúcej tabuľke je znázornený postup vytvárania všetkých možných sumácií medzi klauzulami (v binárnej reprezentácii) k tejto Boolovej funkcie

0. etapa			1. etapa				2. etapa		
1	(1110)		1	(1,4)	(1#10)		2	(4,7)	(0##1)
2	(1011)		2	(2,4)	(101#)		3	(5,6)	(0##1)
3	(0111)		3	(2,6)	(#011)				
4	(1010)		4	(3,5)	(01#1)				
5	(0101)		5	(3,6)	(0#11)				
6	(0011)		6	(5,7)	(0#01)				
7	(0001)		7	(6,7)	(00#1)				

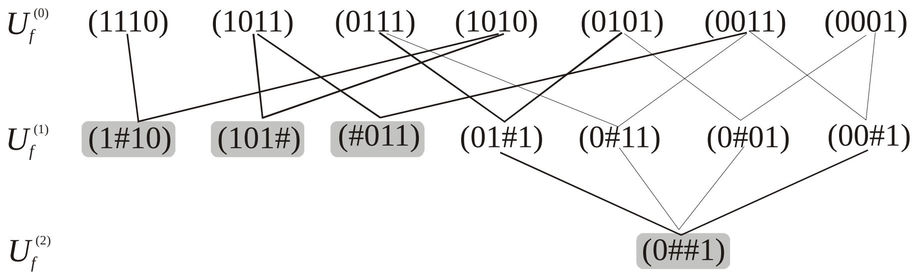
V 0. etape máme zoznam všetkých klauzúl z Boolovej funkcie, množina $U_f^{(0)}$ má tvar

$$U_f^{(0)}=\left\{(1110),(1011),(0111),(1010),(0101),(0011),(0001)\right\}$$

V 1. etape vytvoríme sumáciami nové klauzuly, ktoré obsahujú jeden prázdny znak # (tieto klauzuly sú uvedené v šiestom stĺpci vyššie uvedenej tabuľky

$$U_f^{(1)}=\mathcal{A}\left(U_f^{(0)}\right)=\left\{(1\#10),(101\#),(\#011),(01\#1),\left(0\#11\right),(0\#01),(00\#1)\right\}$$

V 2. etape vytvoríme množinu kroku množina $U_f^{(2)}=\mathcal{A}\left(U_f^{(1)}\right)=\left\{\left(0\#\#1\right)\right\}$ obsahuje už len jednu klauzulu (0##1), ktorá vznikla dvoma alternatívnymi spôsobmi, súčtom klauzúl 5 a 6 resp. 4 a 7 zo 6. stĺpca. Hasseho diagram má tvar



Teraz stojíme pred problémom ako vybrať taký minimálny počet klauzúl, ktoré nám budú pokrývať celú pôvodnú množinu klauzúl. V prvom kroku vyberieme takú minimálnu klauzulu, ktorá pokrýva najväčší počet maximálnych klauzúl. V prípade, že existuje niekoľko rovnocenných minimálnych klauzúl, ktoré pokrývajú rovnaký počet maximálnych klauzúl, tak vyberieme takú minimálnu klauzulu, ktorá má minimálny počet literálov. V tomto prvom kroku vyberieme minimálnu klauzulu (0##1), ktorá pokrýva štyri maximálne klauzuly. V ďalšom kroku máme dve alternatívne možnosti, a to výber minimálnej klauzuly (1#10) alebo výber minimálnej klauzuly (101#). V oboch prípadoch tieto mimimálne klauzuly pokrývajú dve maximálne klauzuly, pretože majú rovnaký počet literálov, tak sú rovnocenné. Na záver máme dve rovnocenné možnosti výberu minimálných klauzúl (101#) resp. (#011), vybrali sme prvú možnosť (101#), tým sme dokázali, že sedem maximálnych klauzúl môže byť pokrytých pomocou troch minimálnych klauzúl (0##1), (1#10) a (101#), ktoré majú dohromady osem literálov. Ekvivalentná (minimálna) Boolova funkcia, ktorá je určená týmito klauzulami má tvar

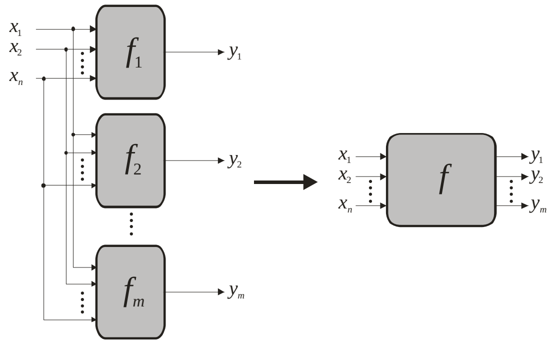
$$f_1\left(w,x,y,z\right)=\overline{w}z+wy\overline{z}+w\overline{x}y$$

Ak by sme vybrali druhú alternatívnu možnosť (#011), potom alternatívny tvar minimálnej Boolovej funkcie je

$$f_2\left(w,x,y,z\right)=\overline{w}z+wy\overline{z}+\overline{x}yz$$

9.6.1 Kombinatoriálne Boolove funkcie

V tejto podkapitole budeme študovať prípad Boolovej funkcie, ktorá má dva alebo viac výstupov (občas sa tieto Boolove funkcie nazývajú kombinatoriálne). Použitím štandardného prístupu k syntéze Boolových funkcií tento problém môžeme jednoducho vyriešiť tak, že pre každú výstupnú hodnotu funkcie zostrojíme štandardným postupom špecifikovaným v predchádzajúcej časti tohto dodatku Boolovu funkciu pre danú premennú (pozri obr. 9.53), kde pôvodná Boolova funkcia f je vyjadrená pomocou nezávislých (dekaplovaných) funkcií f_1 , f_2 , ..., f_m , ktoré medzi sebou neinteragujú.



(9.53)

Nasledujúca tabuľka špecifikuje 2-argumentovú Boolovu funkciu: $\left(y_A,y_B\right)=f\left(x_1,x_2,x_3,x_4\right)$

#	x_1	x_2	x_3	x_4	y_A	y_B
1	0	0	0	0	0	0
2	0	0	0	1	0	1
3	0	0	1	0	1	1
4	0	0	1	1	0	0
5	0	1	0	0	0	0
6	0	1	0	1	1	1
7	0	1	1	0	0	0
8	0	1	1	1	0	0
9	1	0	0	0	0	0
10	1	0	0	1	0	0
11	1	0	1	0	1	1
12	1	0	1	1	0	0
13	1	1	0	0	0	0
14	1	1	0	1	0	0
15	1	1	1	0	1	1
16	1	1	1	1	1	0

Boolove funkcie určené touto tabuľkou majú tvar

$$f_A\left(x_1,x_2,x_3,x_4\right)= \boxed{\bar{x}_1\bar{x}_2x_3\bar{x}_4}+\boxed{\bar{x}_1x_2\bar{x}_3x_4}+\boxed{x_1\bar{x}_2x_3\bar{x}_4}+\boxed{x_1x_2x_3\bar{x}_4}+x_1x_2x_3x_4$$

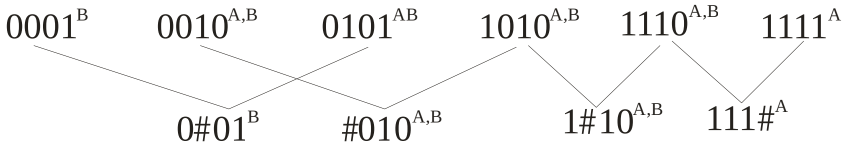
$$f_B\left(x_1,x_2,x_3,x_4\right)=\bar{x}_1\bar{x}_2\bar{x}_3x_4+\boxed{\bar{x}_1\bar{x}_2x_3\bar{x}_4}+\boxed{\bar{x}_1x_2\bar{x}_3x_4}+\boxed{x_1\bar{x}_2x_3\bar{x}_4}+\boxed{x_1x_2x_3\bar{x}_4}$$

kde sú vyznačené spoločné konjunktívne klauzuly. Každá klauzula je špecifikovaná horným indexom *A* alebo *B*,

#	x_1	x_2	x_3	x_4	*subgroup indices*
2	0	0	0	1	*B*
3	0	0	1	0	*A,B*
6	0	1	0	1	*A,B*
11	1	0	1	0	*A,B*
15	1	1	1	0	*A,B*
16	1	1	1	1	*A*

#	x_1	x_2	x_3	x_4	*pair of subgroup indices*
1(2-6)	0	#	0	1	*B*
2(3-11)	#	0	1	0	*A,B*
3(11-15)	1	#	1	0	*A,B*
4(15-16)	1	1	1	#	*A*

Pomocou tejto tabuľky ľahko môžeme zostrojiť spoločný Hasseho diagram pre obidva podsystémy *A* a *B*

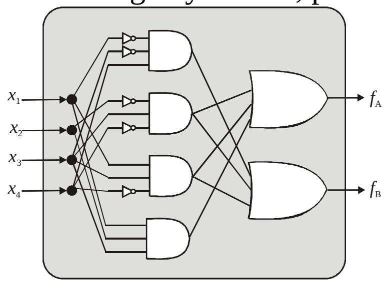


Potom optimalizované Boolove funcie pre podsystémy majú tvar, ktorý obsahuje spoloočné „pobloky“ pre tieto podsystémy

$$f_A = \#010^A + 1\#10^A + 111\#^A = \overline{x}_2 x_3 \overline{x}_4 + x_1 x_3 \overline{x}_4 + x_1 x_2 x_3$$

$$f_B = 0\#01^B + \#010^B + 1\#10^B = \overline{x}_2 \overline{x}_3 x_4 + \overline{x}_2 x_3 \overline{x}_4 + x_1 x_3 \overline{x}_4$$

Tieto dve Boolove funkcie majú spoločné dva bloky $\#010^{A,B}$ a $1\#10^{A,B}$, čo môže podstatne zjednodušiť ich reprezentáciu pomocou logických brán, pozri nasledujúci obrázok (9.54)



(9.54)

Cvičenia

Cvičenie 9.1. Aká je hodnota Boolovej premennej, ktorá je určená podmienkou
(a) $x \cdot \mathbf{1} = \mathbf{0}$,
(b) $x + x = \mathbf{0}$,
(c) $x \cdot \mathbf{1} = x$,
c(d) $x + \overline{x} = \mathbf{1}$.
(e) $x \cdot \overline{x} = \mathbf{0}$

Cvičenie 9.2. Zostrojte tabuľku funkčných hodnôt Boolovej funkcie
(a) $f(x,y,z) = \overline{x}y$,
(b) $f(x,y,z) = x + yz$,
(c) $f(x,y,z) = x\overline{y} + \overline{xyz}$,

Cvičenie 9.3. Znázornite Boolove funkcie $f(x,y,z)$ z cvičenia A.2 na 3-rozmernej kocke tak, že hodnoty 1 (0) budú reprezentované na kocke čiernym (bielym) bodom.
Cvičenie 9.4. Pre ktoré hodnoty x a y platí $xy = x + y$.

Cvičenie 9.5. Zostrojte tabuľku všetkých možných binárnych Boolových funkcií a identifikujte v nej známe Boolove binárne operácie súčinu a súčtu. Vyjadrite ostatné binárne operácie pomocou súčtu, súčinu a komplementu.

Cvičenie 9.6. Riešte nasledujúce rovnice s exkluzívnou disjunkciou
(a) $x \oplus \mathbf{0}$,
(b) $x \oplus \mathbf{1}$,
(c) $x \oplus x$,
(d) $x \oplus \overline{x}$.

Cvičenie 9.7. Dokážte, že platia rovnosti
(a) $x \oplus y = (x+y)\left(\overline{xy}\right)$,
(b) $x \oplus y = \overline{x}y + x\overline{y}$.

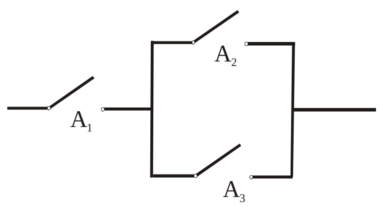
Cvičenie 9.8. Zostrojte duálne výrazy k týmto Boolovym funkciam
(a) $x+y$,
(b) $\overline{x}\,\overline{y}$,
(c) $xyz+\overline{x}\,\overline{y}\,\overline{z}$.

Cvičenie 9.9. Zostrojte Boolovu funkciu $f(x,y,z)$ vo forme sumy produktov klauzúl k premenným *x*, *y* a *z*, ktorá má hodnotu **1**vtedy a len vtedy, ak
(a) $x=y=\mathbf{0}, z=\mathbf{1}$,
(b) $x=\mathbf{0}, y=\mathbf{1}, z=\mathbf{0}$,
(c) $y=z=\mathbf{1}$.

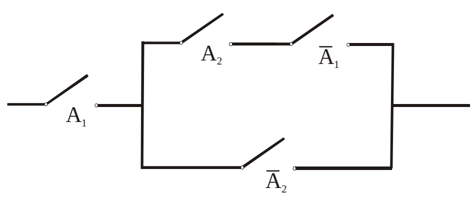
Cvičenie 9.9. Zostrojte Boolovu funkciu $f(x,y,z)$ vo forme sumy produktov klauzúl k premenným *x*, *y* a *z*, ktorá je ekvivalentná s funkciou
(a) $F(x,y,z)=x+y+\overline{z}$,
(b) $F(x,y,z)=x\overline{z}$.

Cvičenie 9.11. Zostrojte spínacie funkcie pre spínacie obvody

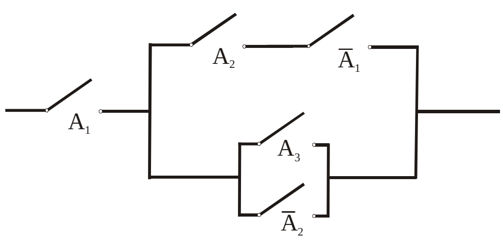
(a)



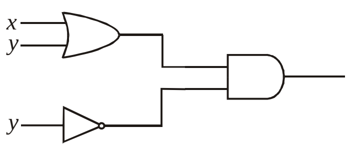
(b)



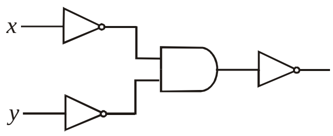
(c)



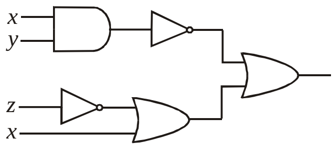
Cvičenie 9.12. Zostrojte tabuľku výstupov logických obvodov
(a)



(b)

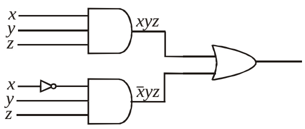


(c)

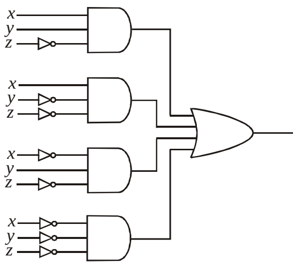


Cvičenie 9.13. Zostrojte logické obvody, ktoré simulujú Boolove funkcie
(a) $\overline{x}+y$,
(b) $\left(\overline{x+y}\right)x$,
(c) $xyz+\overline{x}\,\overline{y}\,\overline{x}$,
(d) $\overline{\left(\overline{x}+z\right)\left(y+\overline{z}\right)}$.

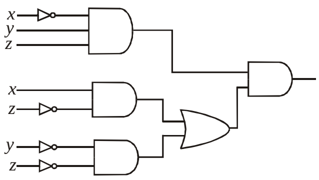
Cvičenie 9.14. Zjednodušte logické obvody
(a)



(b)



(c)



Cvičenie 9.15. Pomocou Quinovej a McCluskeyho metódy nájdite optimálne výrazy k Boolovým funkciám
(a) $wxyz+wx\overline{y}z+wx\overline{y}\,\overline{z}+w\overline{x}y\overline{z}+w\overline{x}\,\overline{y}z$,
(b) $wxy\overline{z}+wx\overline{y}z+w\overline{x}yz+\overline{w}x\overline{y}z+\overline{w}\,\overline{x}y\overline{z}+\overline{w}\,\overline{x}\,\overline{y}z$,
(c) $wxyz+wxy\overline{z}+wx\overline{y}z+w\overline{x}\,\overline{y}z+w\overline{x}\,\overline{y}\,\overline{z}+\overline{w}x\overline{y}z+\overline{w}\,\overline{x}y\overline{z}+\overline{w}\,\overline{x}\,\overline{y}z$.

Cvičenie 9.16. Boolova funkcia f je zadaná pomocou tabuľky funkčných hodnôt. Zostrojte binárne rozhodovacie diagramy a vykonajte ich redukciu, pomocou týchto diagramov zostrojte explicitný tvar Boolovej funkcie.

(a) (b)

#	*x*	*y*	*z*	*f*
1	0	0	0	1
2	0	0	1	0
3	0	1	0	1
4	0	1	1	0
5	1	0	0	1
6	1	0	1	1
7	1	1	0	0
8	1	1	1	1

#	*x*	*y*	*z*	*f*
1	0	0	0	0
2	0	0	1	0
3	0	1	0	1
4	0	1	1	1
5	1	0	0	0
6	1	0	1	0
7	1	1	0	1
8	1	1	1	1

Literatúra

[1] Astola, J. T., Stankovic, R. S.:*Fundamentals of Switching Theory and Logic Design*. Springer, Berlin, 2006.

[2] Frištacký, N., Kolesár, M., Kolenička, J., Hlavatý, J.: *Logické systémy*. Alfa, Bratislava, 110100.

[3] Chen, W. K.: *Logic Desing*. CRC Press, Boca Raton, 2003.

[4] Kvasnička V., Pospíchal, J.: *Algebra a diskrétna matematika*. Vydavateľstvo STU, Bratislava, 2008.

[5] Kvasnička V., Pospíchal, J.: *Matematická logika*. Vydavateľstvo STU, Bratislava, 2006.

[6] Shannon, C. E.: A Symbolic Analysis of Relay and Switching Circuit. *AIEE Transaction*s, **57** (11038), 713- 723.