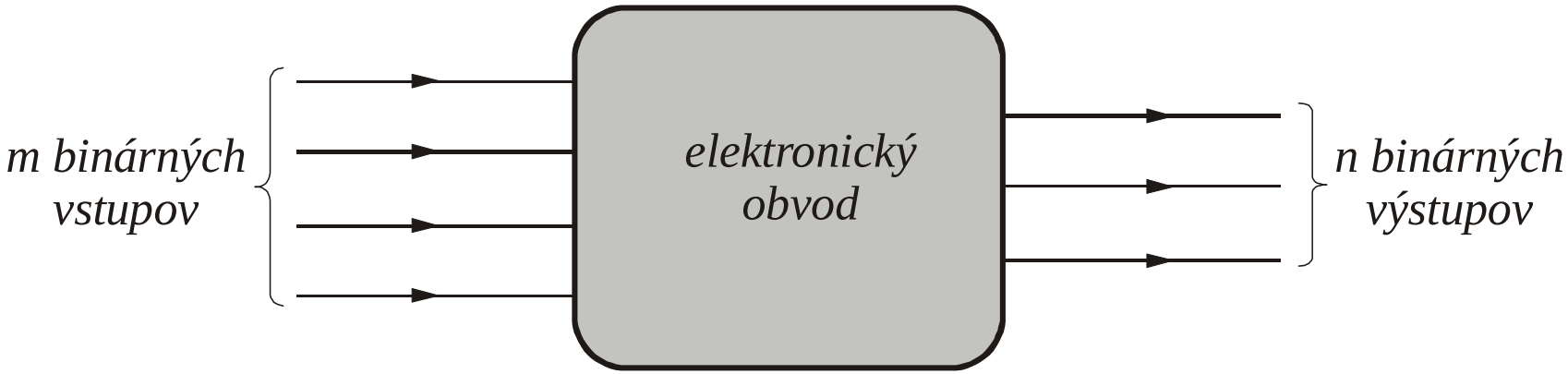
9. kapitola

Boolove funkcie a logické obvody

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. Elektronický obvod môže byť formálne simulovaný tzv. Boolovou funkciou, ktorá transformuje *m* binárnych vstupných premenných na *n* výstupných binárnych premenných.



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.

Boolova algebra má dva známe modely, prvým je výroková logika a druhým algebra teórie množín. Dobrým, ilustratívnym príkladom tohto dualizmu sú De Morganove formuly, ktoré vo výrokovej logike a v teórii množín majú tvary

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

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

Definícia. ***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

$$\cdot : B\times B\to B, \qquad + : B\times B\to B$$

a unárna operácia komplementu

$$\overline{\ } : B\to B$$

ktoré vyhovujú týmto podmienkam

(1) komutatívnosť:

$$x\cdot y=y\cdot x, \qquad x+y=y+x$$

(2) asociatívnosť:

$$\left(x\cdot y\right)\cdot z=x\cdot\left(y\cdot z\right), \qquad \left(x+y\right)+z=x+\left(y+z\right)$$

(3) distributívnosť:

$$x\cdot\left(y+z\right)=\left(x\cdot y\right)+\left(x\cdot z\right), \qquad x+\left(y\cdot z\right)=\left(x+y\right)\cdot\left(x+z\right)$$

(4) vlastnosť konštanty **0**:

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

(5) vlastnosť konštanty **1** :

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

Príklad

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

$$(\forall X\in B)(X\cap\overline{X}=\emptyset)$$

$$(\forall X\in B)(X\cup\overline{X}=A)$$

Algebraická štruktúra $(\mathcal{P}(A),\cup,\cap,\overline{\ },\emptyset,A)$ je Boolova algebra.

Príklad

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 ´≡´, 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, t. j. algebraická štruktúra $(B, \vee, \wedge, \neg, 0, 1)$ tvorí Boolovu algebru.

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

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

Veta. Pre každý element $x \in B$ existuje jednoznačne element $\overline{x} \in B$ taký, že $x \cdot \overline{x} = \mathbf{0}$ a $x + \overline{x} = \mathbf{1}$.

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

Involutívnosť komplementu

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

Idempotentnosť

$$\left(\forall x\in B\right)\left(x\cdot x=x\right),\ \left(\forall x\in B\right)\left(x+x=x\right)$$

De Morganove zákony

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

Nulitnosť

$$\left(\forall x\in B\right)\left(x+\mathbf{1}=\mathbf{1}\right),\ \left(\forall x\in B\right)\left(x\cdot\mathbf{0}=\mathbf{0}\right)$$

Absorpcia

$$\left(\forall x,y\in B\right)\left(x+\left(x\cdot y\right)=x\right),\ \left(\forall x\in B\right)\left(x\cdot\left(x+y\right)=x\right)$$

Komplementy konštánt

$$\overline{\mathbf{0}}=\mathbf{1},\ \ \overline{\mathbf{1}}=\mathbf{0}$$

Vlastnosti konštánt vzhľadom k binárnym operáciam

$$\mathbf{0}+\mathbf{0}=\mathbf{0},\mathbf{0}+\mathbf{1}=\mathbf{1},\mathbf{1}+\mathbf{0}=\mathbf{1},\mathbf{1}+\mathbf{1}=\mathbf{1}$$

$$\mathbf{0}\cdot\mathbf{0}=\mathbf{0},\mathbf{0}\cdot\mathbf{1}=\mathbf{0},\mathbf{1}\cdot\mathbf{0}=\mathbf{0},\mathbf{1}\cdot\mathbf{1}=\mathbf{1}$$

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.

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

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

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

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

V ďalšom texte budeme používať notáciu, ktorá umožní rozlíšiť literál

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

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

Definícia. 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 $\oplus$, ktorá sa nazýva exkluzívna disjunkcia (XOR).

Definícia. 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.

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

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

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

(1) ***Boolova funkcia*** $f(x_1,x_2,...,x_n)$ premenných $x_1,x_2,...,x_n$, je zobrazenie $f:B^n\to B$, pričom $f(x_1,x_2,...,x_n)$ je špecifikovaná ako Boolova formula.

(2) Všetky Boolove formuly, ktoré sú navzájom ekvivalentné, definujú jednu funkciu.

Pretože Boolova funkcia môže byť vyjadrená rôznymi formulami, ktoré sú navzájom ekvivaletné, vzniká otázka, ako efektívne rozhodnúť, či dve Boolove formuly sú ekvivalentné. 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. ***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.

Ak použijeme formalizmus x^e, potom súčinovú klauzulu premenných $x_1, x_2, ..., x_n$, ktorá je špecifikovaná binárnym vektorom $\boldsymbol{e} = (e_1, e_2, ..., e_n)$, má tvar

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

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

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

Definícia. ***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 $\boldsymbol{e} = (e_1, e_2, ..., e_n)$

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

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

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

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

Vyjadrite Boolovu funkciu $x_1 x_2 \left(x_1 + x_3 \right)$ pomocou súčtu súčinových klauzúl (DNF)

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

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

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

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

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

To znamená, že sú 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)

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

Z konštrukcie tejto Boolovej funkcie, vyplýva, že jej funkčné hodnoty sú špecifikované tabuľkou funkčných hodnôt Boolovej funkcie $f(x_1,x_2,...,x_n)$. To znamená, že Boolove funkcie $f(x_1,x_2,...,x_n)$ a $F(x_1,x_2,...,x_n)$ sú ekvivalentné, t. j. majú rovnaké funkčné hodnoty pre rôzne hodnoty argumentov.

Príklad

Jednoduchý dôkaz formuly (9.6) môžeme zostrojiť pomocou Shannonovej formuly 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 $f\left(1,x_2\right)$ použijeme opakovane 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

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

Príklad

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

$$f(x_1,x_2)=f(0,0)\overline{x}_1\overline{x}_2+f(0,1)\overline{x}_1x_2+f(1,0)x_1\overline{x}_2+f(1,1)x_1x_2$$

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

$$f(x_1,x_2)=0\overline{x}_1\overline{x}_2+1\overline{x}_1x_2+1x_1\overline{x}_2+1x_1x_2$$

$$=\overline{x}_1x_2+x_1\overline{x}_2+x_1x_2$$

Príklad

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

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

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

$$f(x_1,x_2,...,x_n)=\prod_{\boldsymbol{e}}\left(f(e_1,e_2,...,e_n)+x_1^{1-e_1}+x_2^{1-e_2}+...+x_n^{1-e_n}\right)$$

$$=\prod_{\boldsymbol{e}}\left(f(e_1,e_2,...,e_n)+L_{(1-e_1,1-e_2,...,1-e_n)}\right)$$

$$=\prod_{\boldsymbol{e}\,(f(\boldsymbol{e})=\boldsymbol{0})}\left(f(\boldsymbol{e})+L_{\overline{\boldsymbol{e}}}\right)$$

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

Príklad

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

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

$$=(x_1+x_2)\cdot(x_1+\overline{x}_2)=x_1\cancel{x_1}+x_1\overline{x}_2+x_2x_1+\cancel{x_2\overline{x}_2}=x_1(x_1+\overline{x}_2+x_2)=x_1$$

Príklad

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

#	e_1	e_2	e_3	$\overline{e}_1$	$\overline{e}_3$	$\overline{e}_1+e_2$	$\overline{e}_1+\overline{e}_3$	$\left(\overline{e}_1+e_2\right)\cdot\left(\overline{e}_1+\overline{e}_3\right)$
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)

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

C. Shannon (1916 – 2001)

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. 1938 C. Shannonom v rámci jeho MSc. dizertacie. 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:

A

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

A_1 A_2

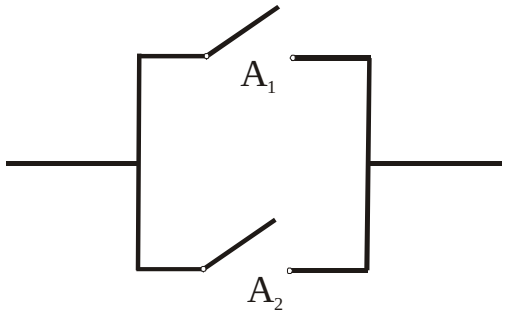
Hovoríme, že v tomto prípade sú spínače zapojené *sériovo*. Nech x_1 a x_2 sú premenné popisujúce stavy spínačov A_1 resp. A_2, tieto premenné ak sa rovnajú 1 (0) , potom daný spínač je uzavretý (otvorený). Nech $f(x_1,x_2)$ je funkcia, ktorej hodnota sa rovná 1 (0) pre tie hodnoty x_1 a x_2, ktoré umožňujú (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	$f\left(x_1,x_2\right)$
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

$$f\left(x_1,x_2\right)=x_1x_2$$

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



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 $(\{0,1\},+,\cdot,\overline{\ },0,1)$, ktorá je špecifikovaná tabuľkou

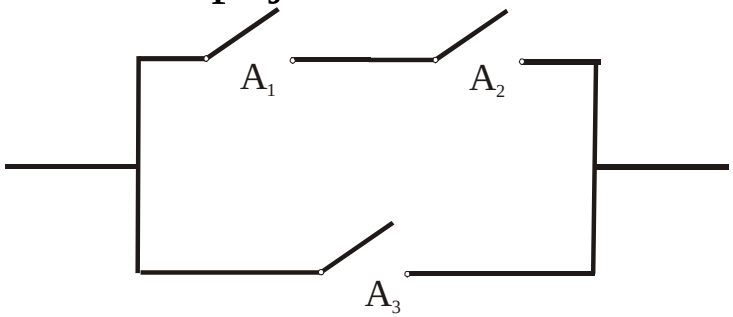
#	x_1	x_2	$g(x_1,x_2)$
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	1

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

$$g(x_1,x_2)=x_1+x_2 \qquad (9.14)$$

Funkcie 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

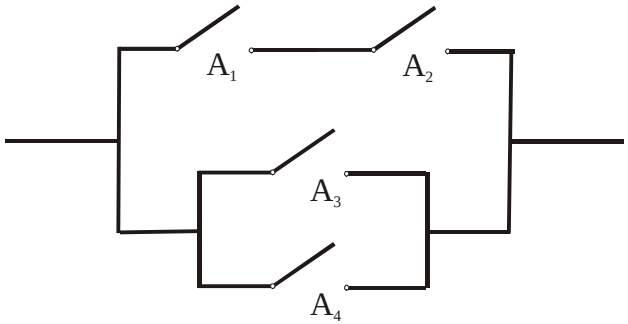


Nech jednotlivé spínače A_1, A_2 a A_3 sú špecifikované premennými x_1, x_2 resp. x_3. Celkový obvod potom môžeme zložiť z dvoch paralelných podštruktúr, horná je reprezentovaná funkciou $f_1(x_1, x_2) = x_1 x_2$ a dolná je reprezentovaná funkciou $f_2(x_3) = x_3$. Spojením týchto dvoch funkcií pomocou (9.14) dostaneme Boolovu funkciu celého spínacieho obvodu (9.15)

$$f(x_1, x_2, x_3) = f_1(x_1, x_2) + f_2(x_3) = x_1 x_2 + x_3$$

Príklad

Zostrojte spínaciu funkciu *f* spínacieho zariadenia



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 $f(x_1, x_2, x_3, x_4)$. V prvom kroku určíme pomocné spínacie funkcie $f_1(x_1, x_2)$ a $f_2(x_3, x_4)$, 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, $f_1(x_1, x_2) = x_1 x_2$ resp. $f_2(x_3, x_4) = x_3 + x_4$. Hľadaná spínacia funkcia má potom tvar

$$f(x_1, x_2, x_3, x_4) = f_1(x_1, x_2) + f_2(x_3, x_4) = x_1 x_2 + x_3 + x_4$$

Príklad

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

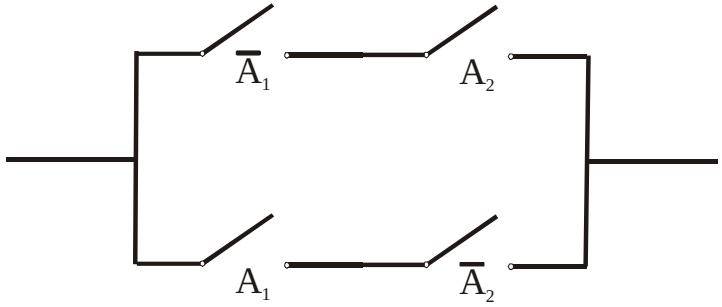
Predpokladajme, že vypínače S_1 a S_2 sú realizovaný pomocou dvoch spínačov A_1 a A_2. Ak použijeme premenné x_1 a x_2 , ktoré označujú stavy spínačov A_1 a A_2 , potom vyššie uvedená tabuľka môže byť prepísaná do tvaru

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

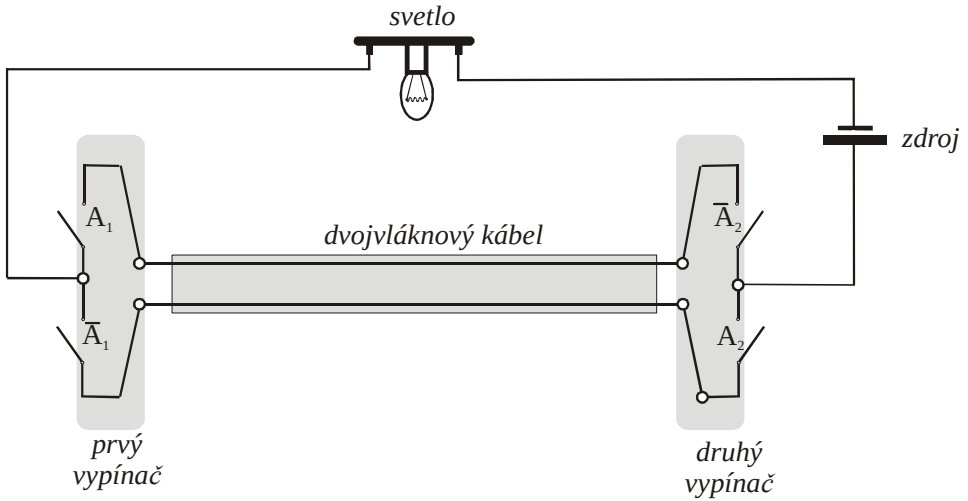
Táto tabuľka špecifikuje Boolovu funkciu v DNF

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

Spínací obvod so spínacou funkciou takto špecifikovanou má tvar



Tanto diagram znázorňuje realizáciu tohto spínacieho obvodu, kde vytieňované oblasti tvoria stenové vypínače, ktoré sú spojené dvojvláknovým káblom.



(9.26)

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

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

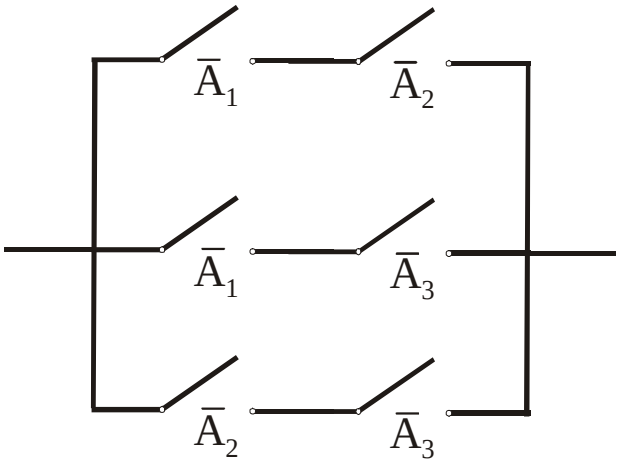
$$F(x_1,x_2,x_3) = \overline{x}_1\overline{x}_2\overline{x}_3 + \overline{x}_1\overline{x}_2x_3 + \overline{x}_1x_2\overline{x}_3 + x_1\overline{x}_2\overline{x}_3 =$$

$$= \underbrace{\overline{x}_1\overline{x}_2\overline{x}_3 + \overline{x}_1\overline{x}_2\overline{x}_3 + \overline{x}_1\overline{x}_2\overline{x}_3}_{\overline{x}_1\overline{x}_2\overline{x}_3} + \overline{x}_1\overline{x}_2x_3 + \overline{x}_1x_2\overline{x}_3 + x_1\overline{x}_2\overline{x}_3 =$$

$$= \overline{x}_1\overline{x}_2\underbrace{\left(\overline{x}_3 + x_3\right)}_{1} + \overline{x}_1\underbrace{\left(x_2 + \overline{x}_2\right)}_{1}\overline{x}_3 + \underbrace{\left(\overline{x}_1 + x_1\right)}_{1}\overline{x}_2\overline{x}_3$$

$$= \overline{x}_1\overline{x}_2 + \overline{x}_1\overline{x}_3 + \overline{x}_2\overline{x}_3$$

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



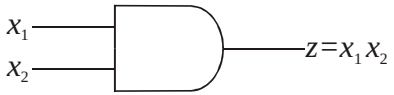
Logické obvody

Na tomto mieste je vhodné pripomenúť si pioniersku prácu McCullocha a Pittsa z r. 1943, 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

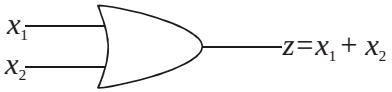
Logické brány

logická brána konjunkcie



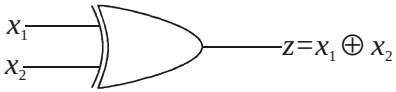
x_1	x_2	x_1x_2
0	0	0
0	1	0
1	0	0
1	1	1

Logická brána disjunkcie



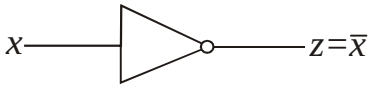
x_1	x_2	x_1+x_2
0	0	0
0	1	1
1	0	1
1	1	1

Logická brána exkluzívnej disjunkcie



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

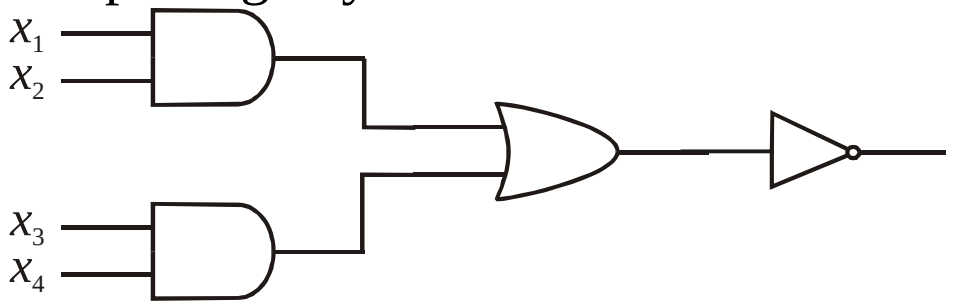
logická brána negácie



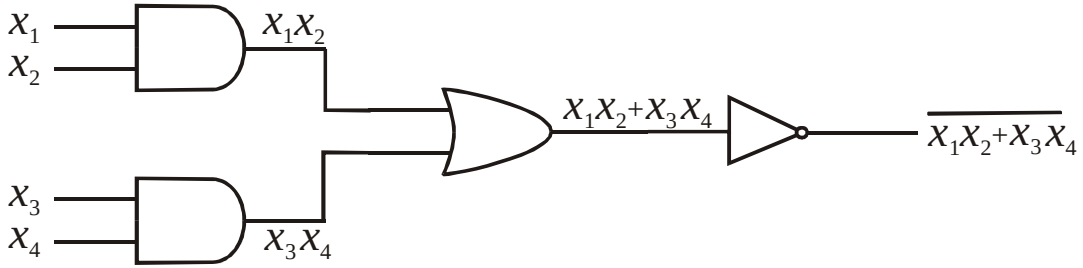
x	$\overline{x}$
0	1
1	0

Príklad

Zostrojte Boolovu funkciu pre logický obvod



Použitím tabuľky logických brán jednotlivé spoje tohto logického obvodu ohodnotíme takto



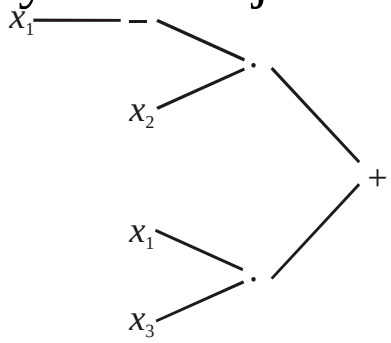
To znamená, že Boolova funkcia priradená tomuto obvodu má tvar

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

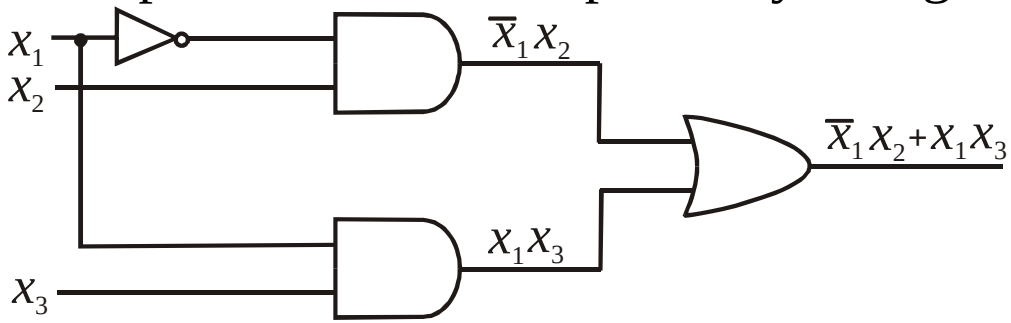
Príklad

Zostrojte logický obvod, ktorý simuluje Boolovu funkciu $f(x_1,x_2,x_3,x_4)=\overline{x}_1x_2+x_1x_3$.

Pre ilustráciu zostrojíme syntaktický strom tejto funkcie



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



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

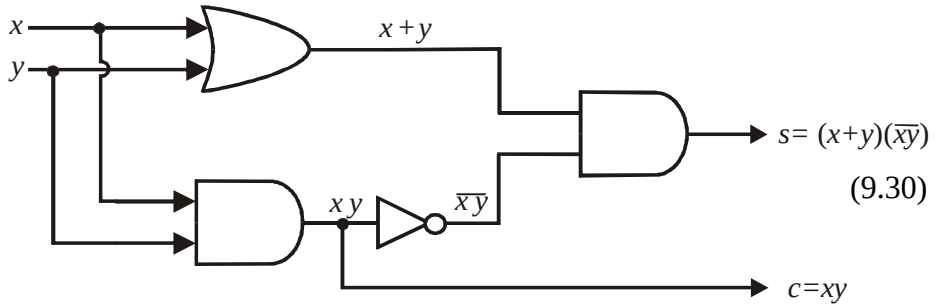
$$\begin{array}{r} x \\ y \\ \hline c\ \ s \end{array}$$

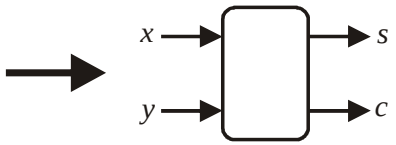
Uvedieme tabuľku všetkých prípadov tejto schémy, ktoré môžu nastať

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

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

$$c = g(x,y) = xy$$





Sumátor troch binárnych čísel (dvojitý sumátor)

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

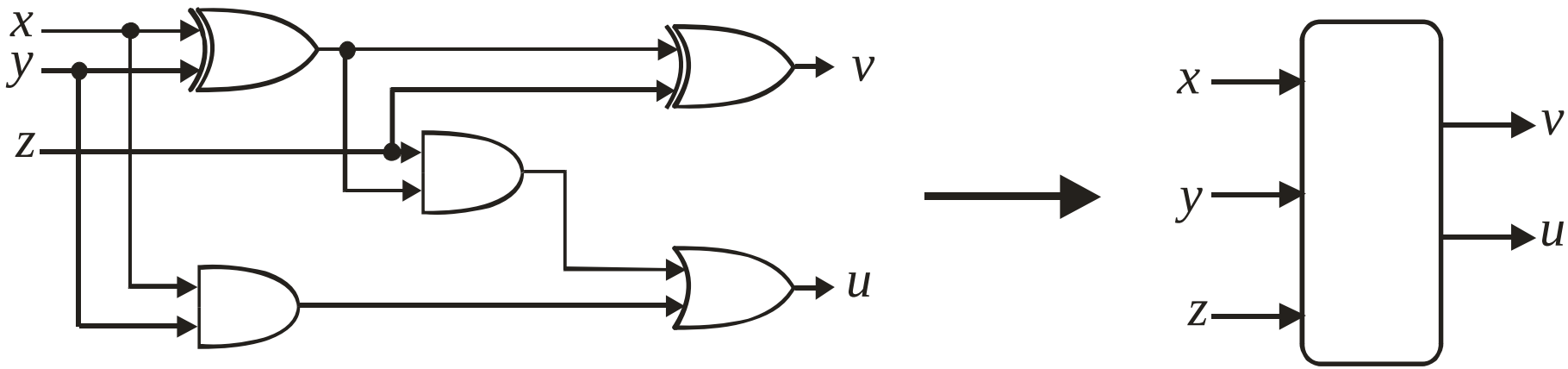
$$\begin{array}{r} x \\ y \\ z \\ \hline u\ v \end{array}$$

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

$$u = \bar{x}yz + x\bar{y}z + xy\bar{z} + xyz = \left(\bar{x}y + x\bar{y}\right)z + xy\left(z + \bar{z}\right)$$
$$= \left(x \oplus y\right)z + xy$$
$$v = \bar{x}\bar{y}z + \bar{x}y\bar{z} + x\bar{y}\bar{z} + xyz = \left(\bar{x}y + x\bar{y}\right)\bar{z} + \left(\bar{x}\bar{y} + xy\right)z$$
$$= \left(x \oplus y\right) \oplus z$$

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



Optimalizácia logických obvodov

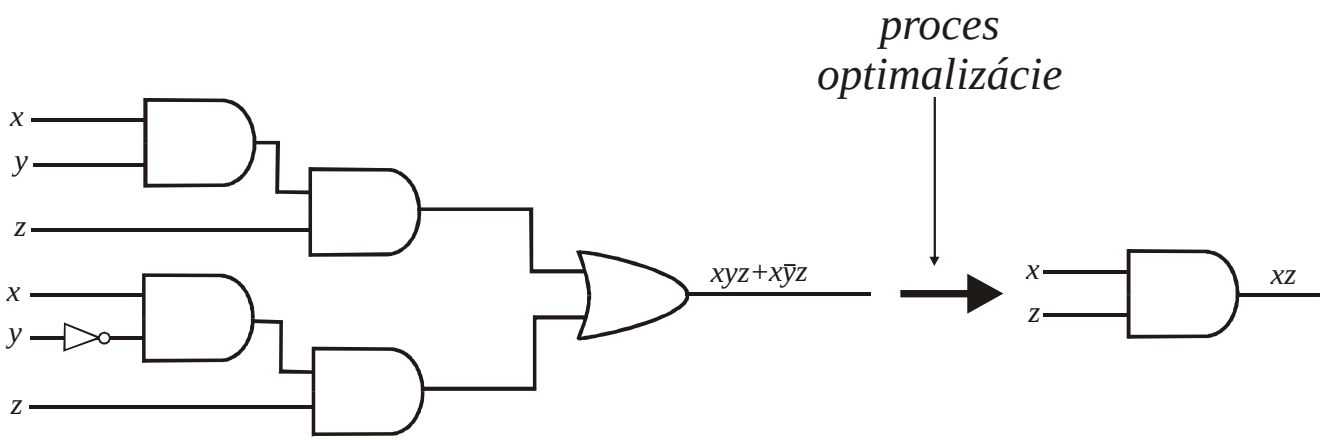
Efektívnosť logických obvodov závisí na počte a prepojení jej logických brán. Proces návrhu logického obvodu začína návrhom tabuľky špecifikujúcej Boolovu funkciu, ktorá transfomuje vstupné binárne veličiny na výstupné binárne veličiny. Boolova funkcia je zostrojená pomocou súčtu konjunktívnych klauzúl. Táto konštrukcia nezaručuje optimálnosť zostrojenej funkcie (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, ktorá simuluje tento logický obvod má tvar $f(x,y,z) = xyz + z\bar{y}z$, môže byť podstatne zjednodušená takto

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

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 $f(x,y,z) = xyz + z\bar{y}z$ je nová funkcia $g(x,z) = xz$, ktorá aj keď je s pôvodnou ekvivalentná, má podstatne menej literálov.

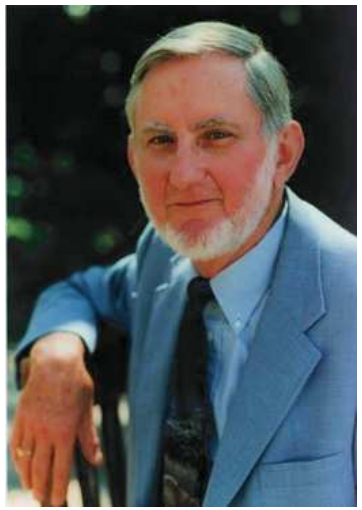
Tento jednoduchý príklad dostatočne jasne ukazuje dôležitosť hľadať optimálny tvar Boolovej funkcie, ktorý môže v špeciálnych prípadoch podstatné zjednodušiť navrhovaný logický obvodu. Ako ilustračný príklad budeme študovať logické obvody priradené Boolovej funkcii $f(x,y,z) = xyz + z\bar{y}z$ a jej optimálnej formy $g(x,z) = xz$



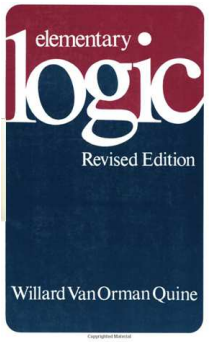
Logický obvod zostrojený pomocou optimálneho tvaru (vpravo), obsahujúceho minimálny počet literálov je podstatne jednoduchší ako logický obvod (vľavo), ktorý obsahuje šesť logických hradiel.

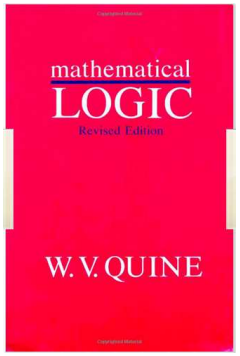
Optimalizácia Boolových funkcií pomocou Quinovej a McCluskeyovej metódy

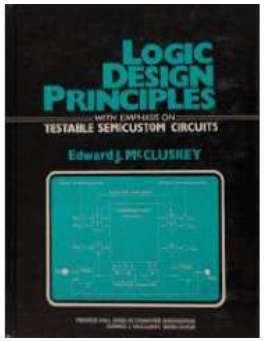
Willard Van Orman Quine (* 1908 – † 2000)



Edward J. McCluskey (*1929)







Ilustračný príklad

Quinovu a McCluskeyho metóda bude ilustrovaná konkrétnym prípadom optimalizácie jednoduchej Boolovej funkcie:

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

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

Každá klauzula môže byť reprezentovaná bitovým reťazcom $\boldsymbol{e} = (e_1, e_2, e_3) \in \{0,1\}^3$

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

kde $\xi^e = \xi$, ak $e = 1$, $\xi^e = \overline{\xi}$, ak $e = 0$, pre $\xi = x, y, z$.

$$f(x,y,z) = (000) + (001) + (011) + (101) + (111)$$

Pre takto definovanú binárnu reprezentáciu môžeme použiť metriku ***Hammingovej vzdialenosti*** ku kvantifikácii podobnosti medzi binárnymi vektormi. Nech $\boldsymbol{e}_i = \left(e_1^{(i)}, e_2^{(i)}, ..., e_n^{(i)}\right)$ a $\boldsymbol{e}_j = \left(e_1^{(j)}, e_2^{(j)}, ..., e_n^{(j)}\right)$ sú dve binárne reprezentácie klauzúl, potom

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

Táto vzdialenosť pre binárne vektory nám špecifikuje počet polôh v ktorých sa binárne vektory vzájomne odlišujú.

Pre ilustračný príklad Boolova funkcia je reprezentovaná pomocou množiny 5 binárnych reťazcov dĺžky 3

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

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.

Proces sčítania klauzúl

Z množiny U_f 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

$$xyx + x\bar{y}z = x\underbrace{\left(y+\bar{y}\right)}_{1}z = xz$$

V binárnej reprezentácii tento proces zjednodušenia formálne vyjadríme takto

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

„*prázdny*“ symbol ´#´ reprezentuje prázdne miesto v binárnej reprezentácii novej klauzuly *xz*.

Nové klauzuly obsahujúce jeden symbol ´#´ tvoria množinu

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

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 $U_f^{(1)}$

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

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 (1) môžeme sčítať len vtedy, ak ich binárne komponenty sa líšia len v jednej polohe.

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

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

Rekurentná formula je inicializovaná množinou $U_f^{(0)} = U_f$. Musí existovať také kladné celé číslo *n*, že tento proces tvorby nových množín je ukončený, t. j. platí $U_f^{(n+1)} = \mathcal{A}\left(U_f^{(n)}\right) = \varnothing$.

- 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 rekurentný proces je ukončený vtedy, ak operátor $\mathcal{A}$ aplikovaný na množinu $U_f^{(n)}$ produkuje prázdnu množinu, t. j. $\mathcal{A}\left(U_f^{(n)}\right) = \varnothing$.

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

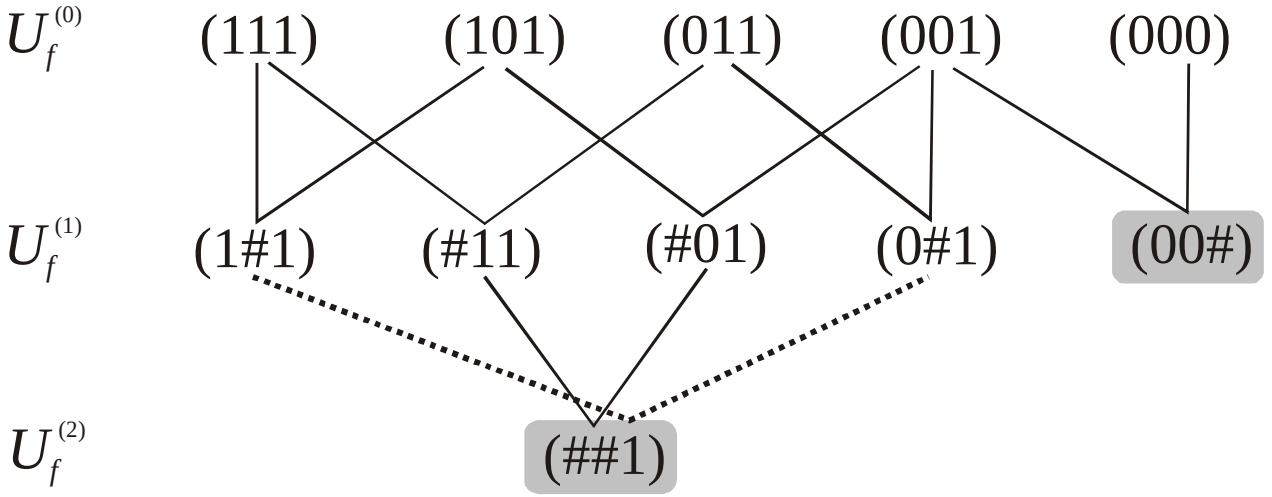
V stĺpcoch pre prvú a druhú etapu sú uvedené aj dvojice indexov klauzúl z predchádzajúceho stĺpca, ktoré boli použité v sumačnom procese.

Úloha: ako vybrať taký minimálny počet klauzúl zostrojených v prvej alebo v ďalších etapách, ktoré sú odvoditeľné zo všetkých pôvodných klauzúl z množiny $U_f^{(0)} = U_f$.

Množina klauzúl, ktorá vznikne zjednotením množín $U_f^{(0)}$, $U_f^{(1)}$, $U_f^{(3)}$,...

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

je čiastočne usporiadaná a je 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.

Úloha: Vybrať také minimálne klauzuly, ktoré pokrývajú pôvodné klauzuly z množiny $U_f^{(0)}$. Pre každú klauzulu $\boldsymbol{e} \in U_f^{(1)} \cup U_f^{(2)} \cup ...$, ktorá obsahuje aspoň jeden prázdny symbol, zostrojíme množinu

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

ktorá obsahuje všetky pôvodné klauzuly (neobsahujúce prázdne symboly #), ktoré sú pokryté klauzulou $\boldsymbol{e}$

Nech množina minimálnych klauzúl je označená V, potom hľadáme takú jej podmnožinu $\widetilde{V} \subseteq V$, ktorej klauzuly plne pokrývajú množinu $U_f^{(0)}$

$$\bigcup_{e \in \widetilde{V}} U(e) = U_f$$

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

$$\widetilde{V} = arg \min_{V' \subseteq V} [V']$$

- 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. Potom nastupujú ***evolučné metódy***, ktoré poskytujú v reálnom čase kvalitné suboptimálne riešenie, ktoré je často rovné optimálnemu riešeniu.

V tomto konkrétnom prípade sa jedná o jednoduchý problém, musíme vybrať obe minimálne klauzuly (##1) a (00#), ktoré pokrývajú pôvodné klauzuly.

Potom môžeme písať Boolovu funkciu

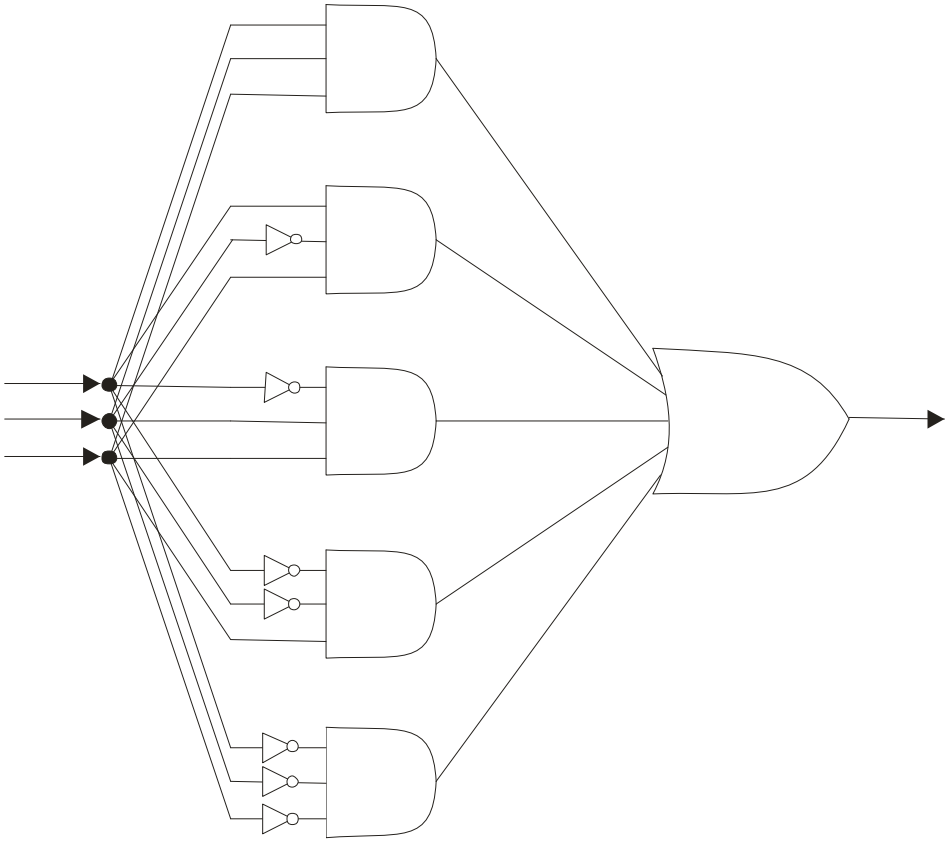
$$f\left(x,y,z\right)=xyz+x\overline{y}z+\overline{x}yz+\overline{x}\,\overline{y}z+\overline{x}\,\overline{y}\,\overline{z}$$

v ekvivalentnom tvare

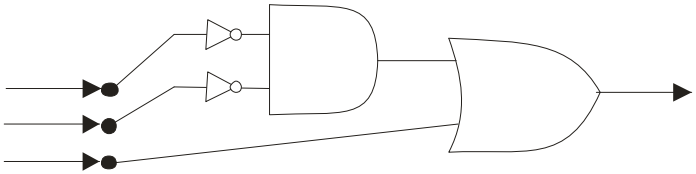
$$f\left(x,y,z\right)=z+\overline{x}\,\overline{y}$$

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

$$f(x,y,z) = xyz + x\bar{y}z + \bar{x}yz + \bar{x}\bar{y}z + \bar{x}\bar{y}\bar{z}$$



$$f(x,y,z) = z + \bar{x}\,\bar{y}$$



Príklad

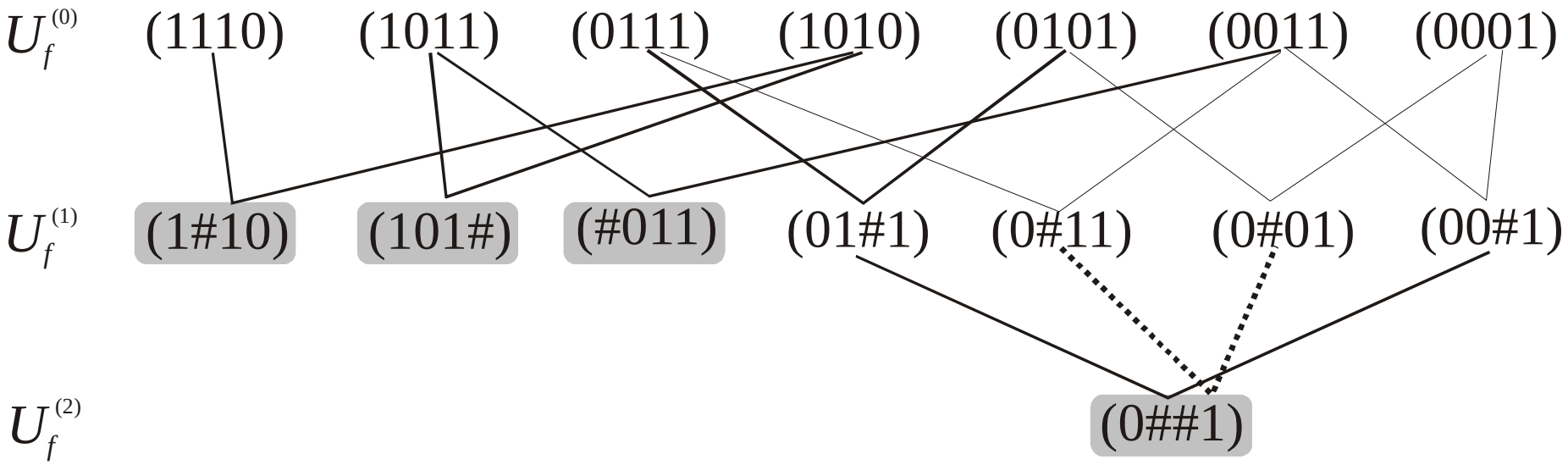
Uvažujme Boolovu funkciu

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

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

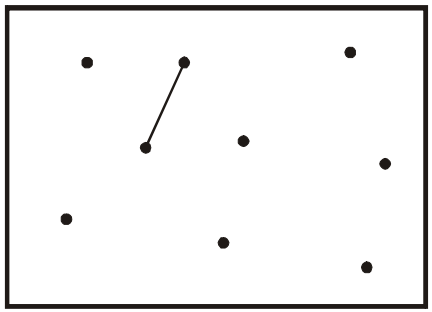
0. etapa			1. etapa				2. etapa		
1	(1110)		1	(1,4)	(1#10)		1	(4,7),(5,6)	(0##1)
2	(1011)		2	(2,4)	(101#)				
3	(0111)		3	(2,6)	(#011)				
4	(1010)		4	(3,5)	(01#1)				
5	(0101)		5	(3,6)	(0#11)				
6	(0011)		6	(5,7)	(0#01)				
7	(0001)		7	(6,7)	(00#1)				

Hasseho diagram

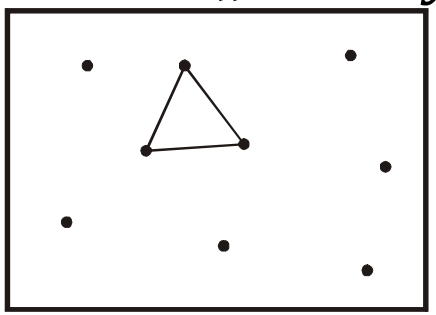


Teraz stojíme pred problémom ako vybrať taký minimálny počet klauzúl, ktoré nám budú pokrývať celú pôvodnú množinu klauzúl.

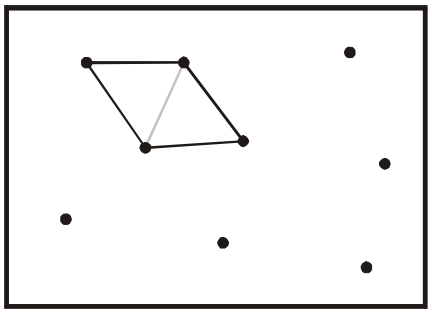
„Greedy“ metóda



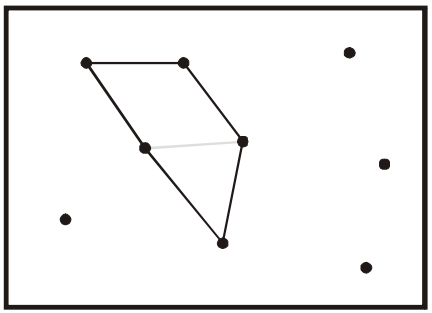
1. *krok*



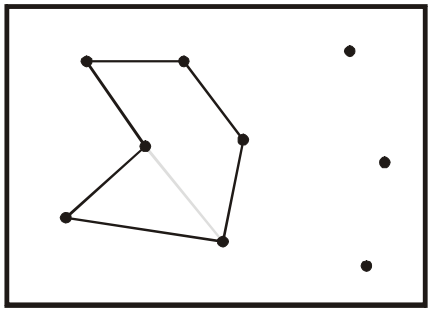
2. *krok*



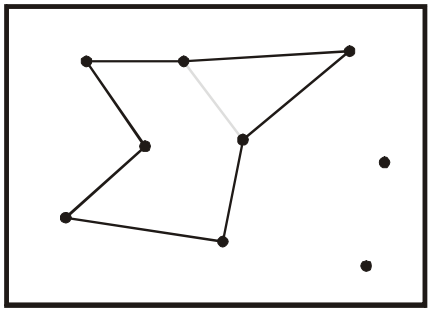
3. *krok*



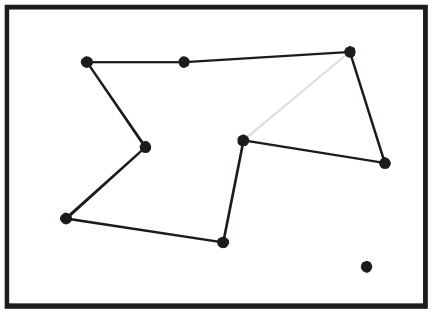
4. *krok*



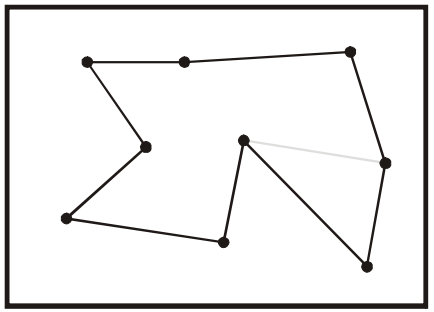
5. *krok*



6. *krok*



7. *krok*



8. *krok* (*konečné riešenie*)

Použitie jednoduchej “greedy” heuristiky pre konštrukciu cesty obchodného cestujúceho. Algoritmus je inicializovaný trojuholníkom, ktorý je zostrojený z najkratšej hrany doplnenej o najbližší vrchol. V ďalšom kroku nájdeme voľný vrchol x a hranu (y,z) tak, aby veličina $d(x,y)+d(x,z)-d(y,z)$ bola minimálna. Aktuálny cyklus rozšírime o hrany (x,y) a (x,z), pričom pôvodnú hranu (y,z) odstránime. Tento proces opakujeme tak dlho, až každý vrchol je zapojený do vytvoreného cyklu. Naše simulačné výpočty naznačujú, že takto vytvorená uzavretá cesta obchodného cestujúceho poskytuje kvalitné suboptimálne riešenie (ktoré v mnohých prípadoch je totožné s optimálnym riešením).

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. Táto možnosť je znázornená na nasledujúcej tabuľke:

minimálne klauzuly	maximálne (pôvodne) klauzuly						
	(1110)	(1011)	(0111)	(1010)	(0101)	(0011)	(0001)
(1#10)	1			1			
(101#)		1		1			
(#011)		1				1	
(0##1)			1		1	1	1

V druhom 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. Predpokladajme, že vyberieme prvú možnosť, potom predchádzajúca tabuľka má tvar:

minimálne klauzuly	maximálne (pôvodne) klauzuly						
	(1110)	(1011)		(1010)			
(1#10)	1			1			
(101#)		1		1			
(#011)		1					
(0##1)							

Ak z tejto tabuľky odstránime vybranú klauzulu, potom sa tabuľka ďalej zjednoduší do tvaru:

minimálne klauzuly	maximálne (pôvodne) klauzuly						
		(1011)					
(1#10)							
(101#)		1					
(#011)		1					
(0##1)							

Na záver ***v treťom kroku*** máme dve rovnocenné možnosti výberu minimálnych 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

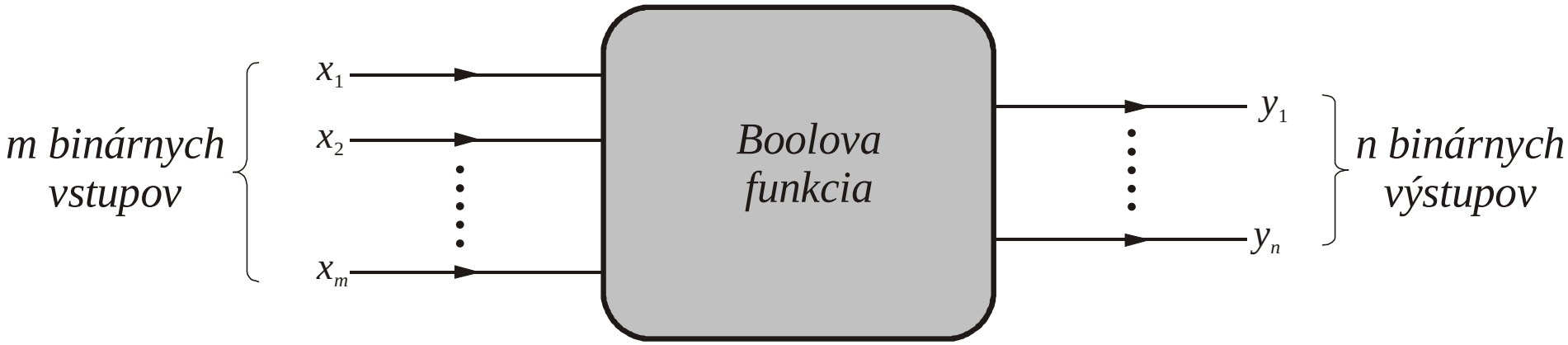
$$f_1(w,x,y,z) = \overline{w}z + wy\overline{z} + w\overline{x}y$$

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

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

Optimalizácia Boolovej funkcie s *viacerými* funkčnými hodnotami

$$f : \{0,1\}^m \rightarrow \{0,1\}^n, \text{ kde } n \geq 2$$



Poznámka: Optimalizačná metóda Q-M sa ľahko zovšeobecní aj pre tento prípad. Jednotlivé členy Boolovej funkcie budú označené horným indexom výstupu, ktorý špecifikuje každú klauzulu.

Tabuľka špecifikujúca Boolovu funkciu

#	x_1	x_2	x_3	x_4	f_A	f_B
1	0	0	0	0	0	0
2	0	0	0	1	1	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

#	x_1	x_2	x_3	x_4	f_A	f_B
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

Poznámka: Budeme vyberať do procesu „súčtu“ len také dva riadky, ktoré

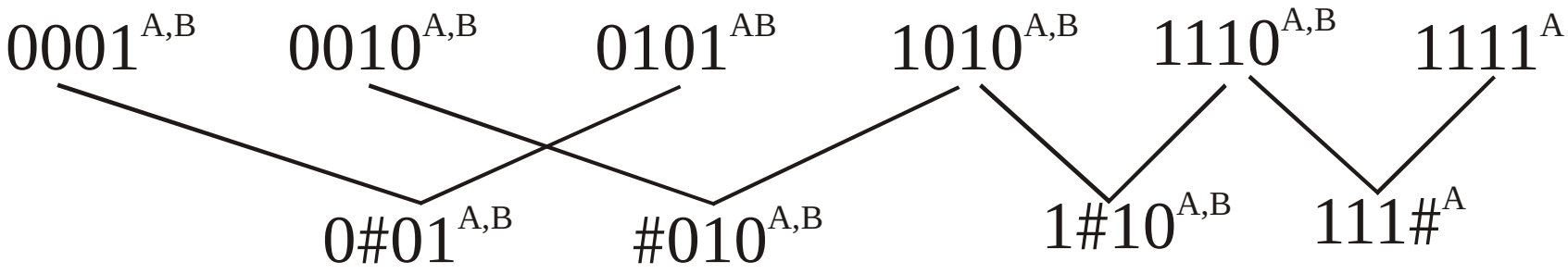
(1) majú jednotkovú Hammingovu vzdialenosť,

(2) prienik indexov funkčných hodnôt je neprázdna množina, pričom výsledný súčet je označený týmto prienikom.

Vybrané riadky s jednotkovou funkčnou hodnotou

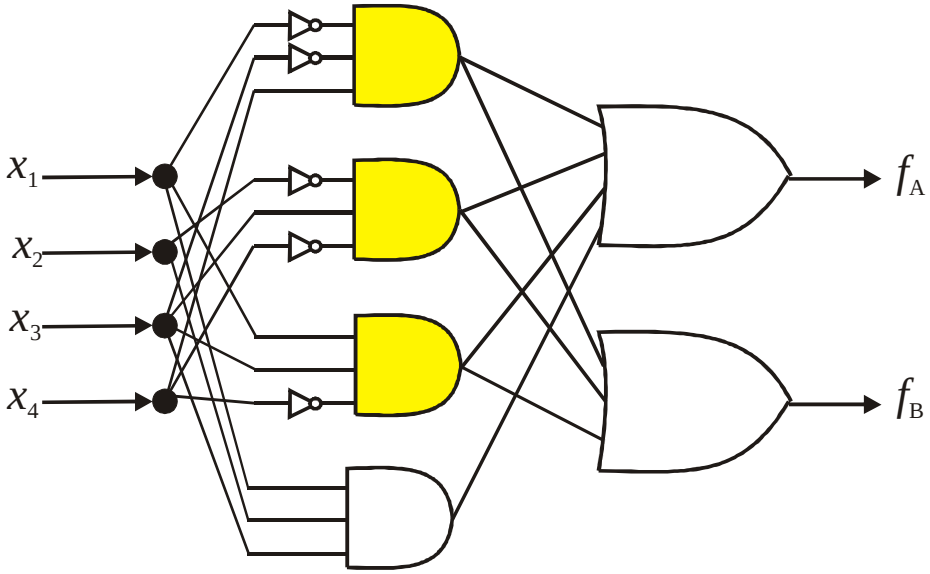
#	x_1	x_2	x_3	x_4	f_X
2	0	0	0	1	*A,B*
3	0	0	1	0	*A,B*
6	0	1	0	1	*A,B*
11	1	0	1	0	*A,B*
15	1	1	1	0	*A,B*
16	1	1	1	1	*A*

#	r_i - r_j	x_1	x_2	x_3	x_4	f_X
1	2 - 6	0	#	0	1	*A,B*
2	3 - 11	#	0	1	0	*A,B*
3	11 - 15	1	#	1	0	*A,B*
4	15 - 16	1	1	1	#	*A*

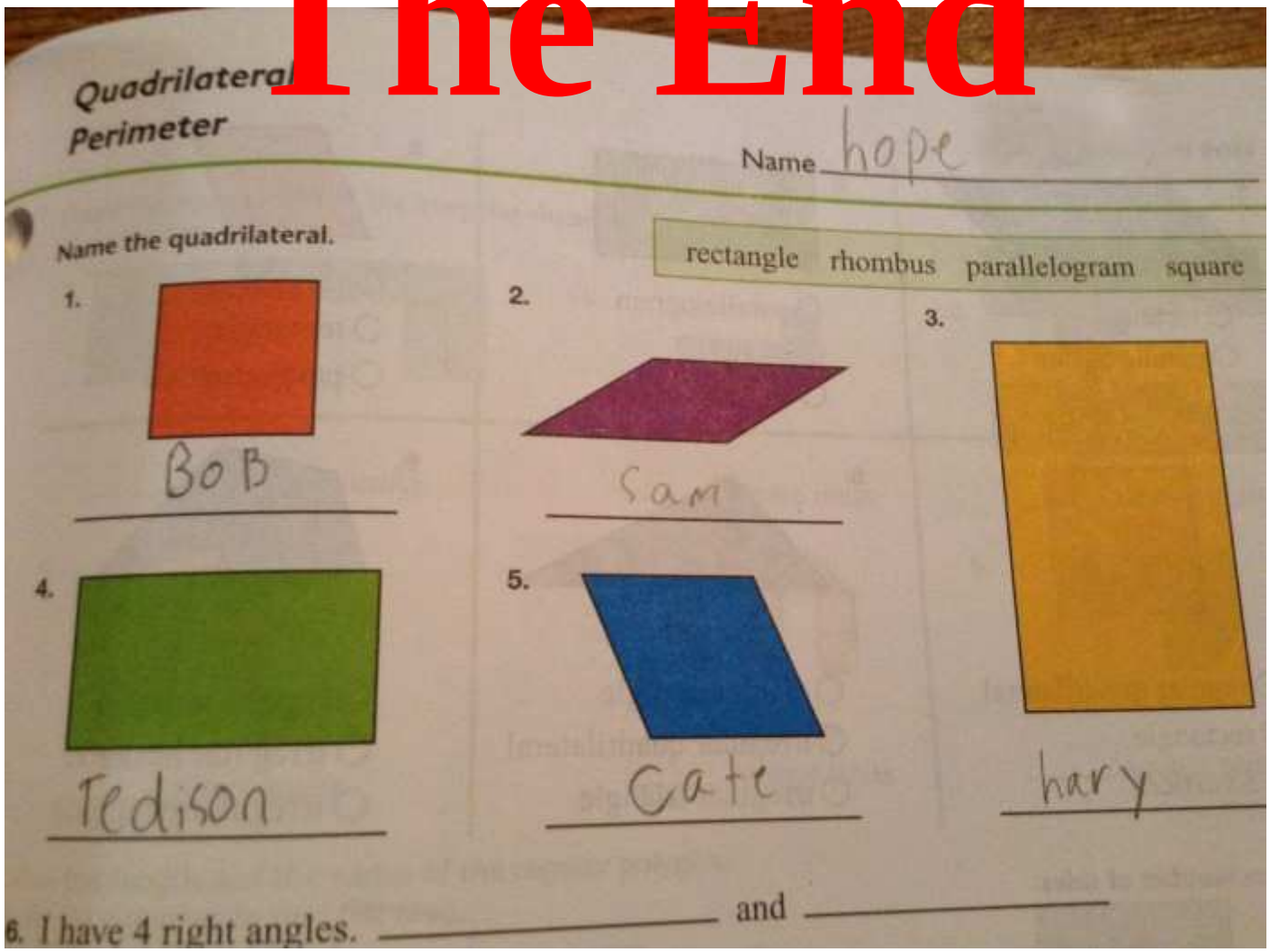


$$f_A = 0\#01 + \#010 + 1\#10 + 111\#$$

$$f_B = 0\#01 + \#010 + 1\#10$$



The End



Quadrilateral
Perimeter

Name hope

Name the quadrilateral.

rectangle rhombus parallelogram square

1. BOB

2. Sam

3. hary

4. Tedison

5. Cate

6. I have 4 right angles. _____ and _____