

Počítanie ANF pre riedke booleovské funkcie

Peter Selický, Ing. Pavol Zajac¹

KAIVT

peter_selicky@hotmail.com

¹ Vedúci práce

Abstrakt

Práca sa zameriava na štúdium a experimentálne overenie efektívnosti algoritmov na výpočet ANF Booleovskej funkcie malého počtu premenných..

1. Úvod

V booleovskej logike, algebraická normálna forma (ANF) je spôsob ako štandardizovať a normalizovať logické vzorce.

V analytickej časti 2 rozoberáme základné poznatky danej problematiky. Časť 3 je venovaná metológii experimentov a v závere sumarizujeme a interpretujeme získané výsledky.

2. Základné definície

2.1 Booleovské funkcie

Funkcia $f: Z_2^n \rightarrow Z_2, n \in \mathbb{N}^+$ sa nazýva **Booleovská funkcia** n premenných.

Riedka Booleovská funkcia závisí len na málo premenných a obsahuje malý počet jednotiek, resp. núl. Totiž, ak funkcia obsahuje len malý počet núl, môžeme počítať

$$g(x) = f(x) + 1, \quad (1)$$

a teda funkcia g má malý počet jednotiek.

Vektorom hodnôt booleovskej funkcie je vektor

$[a_0, a_1, \dots, a_{2^n-1}]$, kde a_i je hodnota

$$\hat{f}(i),$$

pričom $\hat{i}$ vyjadruje binárny prepis čísla i . To znamená, že vektor hodnôt je jednoducho postupnosť všetkých možných hodnôt, ktoré nadobúda funkcia f pre postupne sa meniace x . [1]

2.2 ANF - Algebraická normálna forma

Nech $a_0, a_1, \dots, a_{1,2,\dots,n} \in \{0,1\}$, booleovskú funkciu môžeme jednoznačne zapísať v tvare:

$$\begin{aligned} f(x_1, x_2, \dots, x_n) = & a_0 + \\ & a_1 x_1 + a_2 x_2 + \dots + a_n x_n + \\ & a_{1,2} x_1 x_2 + \dots + a_{n-1,n} x_{n-1} x_n + \\ & \dots + \\ & a_{1,2,\dots,n} x_1 x_2 \dots x_n \end{aligned}$$

(2)

Uvedený zápis sa nazýva algebraická normálna forma (ANF) funkcie f . Inak povedané, ANF je polynóm viac premenných s koeficientami v Z_2 , ktorého hodnoty sú hodnoty Booleovskej funkcie. [2]

Nech $f: Z_2^n \rightarrow Z_2$ je Booleovská funkcia, a nech $f(x_1, x_2, \dots, x_n) \in Z_2[x_1, x_2, \dots, x_n]$ je polynóm predstavujúci jej algebraickú normálnu formu. Na uschovanie celej algebraickej normálnej formy potrebujeme 2^n koeficientov. Toto číslo rastie exponenciálne. Nie všetky koeficienty sú však nenulové a funkciu f vieme v niektorých prípadoch reprezentovať aj úspornejšie. Nech pre každé $x_2, x_3, \dots, x_n \in Z$ platí

$$f(0, x_2, \dots, x_n) = f(1, x_2, \dots, x_n) \quad (3)$$

Potom f nezávisí od x_1 a funkciu f môžeme zapísať pomocou polynómu

$$f' \in Z_2[x_2, x_3, \dots, x_n]$$

Podobne môžeme odstrániť aj iné premenné zo zápisu algebraickej normálnej formy. Pre riedke Booleovské funkcie je veľkosť ANF dostatočne malá pre jej uchovanie.

2.3 Algoritmy

Pracujeme s 3 algoritmami. Vstupom do každého je vektor hodnôt Booleovskej funkcie. Výstupom je algebraická normálna forma danej funkcie zapísaná v binárnej sústave. Koeficient na pozícii i zodpovedá termu, kde sa vyskytujú členy zodpovedajúce jednotkám v binárnom rozvoji i . Napr. $i = (1011)_2$ zodpovedá termu $x_1 x_2 x_4$.

2.3.1 Algoritmus 1

Majme n premenných s 2^n vstupmi. Sz je veľkosť čiastkovej tabuľky a Pos hodnota pozície v tejto čiastkovej tabuľke. Vstupný predpis vektora je g . Tento algoritmus sa nazýva Moebiusov transformačný algoritmus, ktorý je veľmi podobný Walshovmu transformačnému algoritmu.

```

for i from 0 to  $n - 1$  do
  Let  $Sz \leftarrow 2^i$ ,  $Pos \leftarrow 0$ 
  while  $Pos < 2^n$  do
    for j from 0 to  $Sz - 1$  do
       $g[Pos + Sz + j] \leftarrow g[Pos + j] \otimes g[Pos + Sz + j]$ 
    end for
    Let  $Pos \leftarrow Pos + 2 \cdot Sz$ 
  end while
end for
 $ANF(f) = g(x_1, x_2, \dots, x_n)$ 

```

[3]

2.3.2 Algoritmus 2

Vypočítame vektory pre všetky možné termy (do veľkosti danej veľkosťou vstupov) a uložíme ich do poľa vektorov V. Vstupný predpis vektora je g . Sz je veľkosť tabuľky. Jedná sa vlastne o maticové násobenie, ktorého efektívnejší zápis vyjadruje práve Algoritmus 1.

```

Let  $Sz \leftarrow 2^i$ 
for i from 0 to  $Sz - 1$  do
  for j from 0 to  $Sz - 1$  do
     $g[j] \leftarrow V[i, j] \otimes g[j]$ 
  end for
end for
 $ANF(f) = g(x_1, x_2, \dots, x_n)$ 

```

2.3.3 Algoritmus 3

Algoritmus 3 vychádza z postupného pripájania koeficientov na základe rozdielu doterajšej ANF a spracovávanej funkcie.

```

 $g(x_1, x_2, \dots, x_n) = f(0, 0, \dots, 0)$ 
for k = 1 to  $2^n - 1$  do
   $k = b_1 + b_2 2 + b_3 2^2 + \dots + b_n 2^{n-1}$ 
  if  $g(b_1, b_2, \dots, b_n) \neq f(b_1, b_2, \dots, b_n)$  then
     $g(x_1, x_2, \dots, x_n) \neq f(x_1, x_2, \dots, x_n) \otimes \prod_{i=1}^n (x_i)^{b_i}$ 
  end if
end for
 $ANF(f) = g(x_1, x_2, \dots, x_n)$ 

```

[4]

3. Metológia experimentov

3.1 Cieľ a spôsoby merania

Cieľom práce je zistenie v praxi najvýhodnejšieho algoritmu na výpočet algebraickej normálnej formy pre špecifické, tzv. riedke Booleovské funkcie.

Teoretické odhady (pozri časť 3.2) pre väčší počet jednotiek vo vektore hodnôt jednoznačne zvýhodňujú algoritmus 1, avšak pre zistenie efektívnejšieho algoritmu potrebujeme vytvoriť reálny program pre testovanie rýchlosti jednotlivých algoritmov. Kvôli merateľnosti jednotlivých algoritmov využívame desaťtisíc až milión krát opakované cykly v závislosti od počtu neznámych. V algoritme 3 kvôli zisteniu priemerného času každý cyklus ešte opakujeme 100 krát s rôznymi vstupmi.

3.2 Teoretické odhady zložitosti

Pre teoretický odhad zložitosti zadefinujeme vzorce na výpočet počtu xorov T pre prvé dva algoritmy.

Algoritmus 1:

$$T_1 = \frac{n}{2} \cdot 2^n \quad (4)$$

Algoritmus 2:

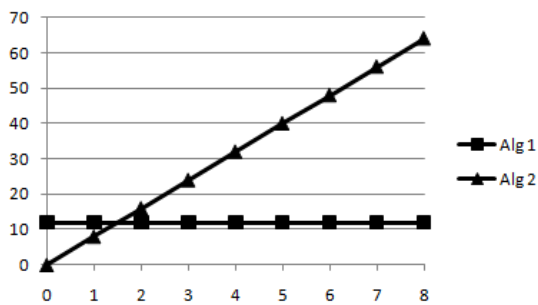
$$T_2 = k \cdot 2^n \quad (5)$$

Označme k ako počet jednotiek v danej tabuľke pravdivosti a n ako počet premenných. Pri testovaní počítame s 3 premennými pre jednotlivé variácie vstupov podľa vzorcov (4) a (5).

Tab. 1. Zložitosť podľa teoretického odhadu.

K	Algoritmus 1	Algoritmus 2
8	12	64
7	12	56
6	12	48
5	12	40
4	12	32
3	12	24
2	12	16
1	12	8
0	12	0

Na základe tabuľky zostrojíme graf pre zobrazenie priebehu zložitosti pre oba algoritmy.



Obr. 1. Zložitosť podľa teoretického odhadu.

Zložitosť Algoritmu 3 závisí navyše nielen na počte jednotiek, ale aj na ich polohe vo vektore hodnôt. Overujeme ju preto presnejšie len experimentálne a všimame si najlepší, priemerný a najhorší čas priebehu algoritmu.

4. Experimentálne výsledky

Zadefinujeme najprv premenné použité v tejto časti:
 n - počet premenných
 k - počet jednotiek v Booleovskej funkcii

$a1$ - čas priebehu Algoritmu 1

$a2$ - čas priebehu Algoritmu 2

$a3a$ - čas priebehu Algoritmu 3 v najhoršom prípade

$a3b$ - čas priebehu Algoritmu 3 v priemernom prípade

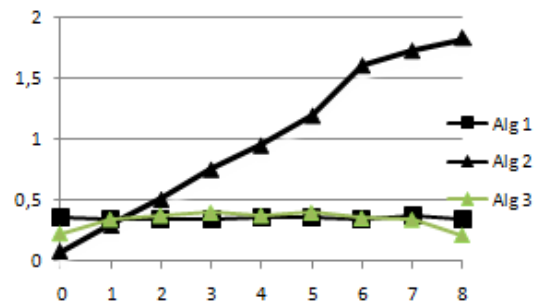
$a3c$ - čas priebehu Algoritmu 3 v najlepšom prípade

Čas priebehu algoritmu meriame v mikrosekundách. Meriame postupne pre 3,4,5,6,7 a 8 premenných, kde si všimame prvé 2 a posledných 7 riadkov daných tabuliek. Pre každú tabuľku sme zostrojili aj prislúchajúci graf.

4.1 3 premenné

Tab. 2. $n = 3$

k	a1	a2	a3a	a3b	a3c
8	1,82	0,351	0,246	0,212	0,207
7	1,717	0,373	0,572	0,341	0,227
6	1,595	0,349	0,54	0,354	0,239
5	1,2	0,359	0,527	0,4	0,279
4	0,959	0,363	0,47	0,377	0,275
3	0,754	0,35	0,557	0,4	0,288
2	0,518	0,347	0,562	0,377	0,245
1	0,306	0,342	0,582	0,343	0,235
0	0,093	0,361	0,269	0,227	0,222

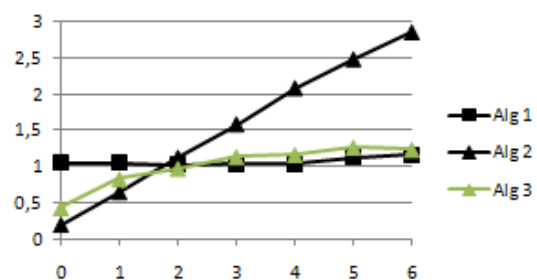


Obr. 2. Čas trvania algoritmu podľa tabuľky 2.

4.2 4 premenné

Tab. 3. $n = 4$

k	a1	a2	a3a	a3b	a3c
16	7,5	1,03	0,85	0,439	0,42
15	7,01	1,05	2,22	0,846	0,43
6	2,85	1,16	1,76	1,236	0,66
5	2,48	1,13	1,97	1,263	0,76
4	2,08	1,04	1,94	1,165	0,62
3	1,58	1,04	2,21	1,136	0,54
2	1,12	1,03	2,09	0,979	0,46
1	0,64	1,05	2,12	0,831	0,43
0	0,19	1,05	0,79	0,436	0,42

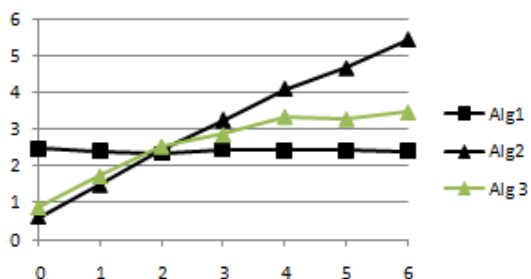


Obr. 3. Čas trvania algoritmu podľa tabuľky 3.

4.3 5 premenných

Tab. 4. $n = 5$

k	a1	a2	a3a	a3b	a3c
32	27,8	2,28	1,86	0,886	0,84
31	26,22	2,24	7,24	1,85	0,86
6	5,44	2,4	6,64	3,473	1,44
5	4,66	2,42	6,5	3,292	1,48
4	4,1	2,42	7,16	3,353	1,28
3	3,26	2,44	6,64	2,868	1,12
2	2,42	2,34	6,98	2,547	1
1	1,48	2,4	7,32	1,733	0,86
0	0,62	2,48	1,38	0,88	0,84

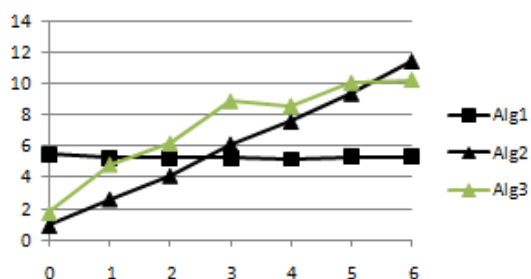


Obr. 4. Čas trvania algoritmu podľa tabuľky 4.

4.4 6 premenných

Tab. 5. n = 6

k	a1	a2	a3a	a3b	a3c
64	108,900	5,233	3,333	1,918	1,733
63	106,033	5,367	24,367	4,248	1,7
6	11,433	5,300	22,456	10,236	4,767
5	9,367	5,333	22,367	10,045	2,934
4	7,567	5,167	21,434	8,575	2,667
3	6,133	5,267	25,267	8,874	2,2
2	4,100	5,267	23,4	6,172	1,8
1	2,600	5,233	24	4,842	2
0	0,900	5,467	2,467	1,747	1,7

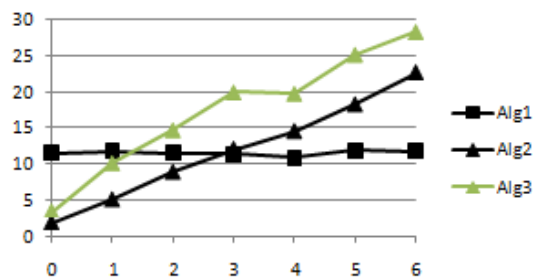


Obr. 5. Čas trvania algoritmu podľa tabuľky 5.

4.5 7 premenných

Tab. 6. n = 7

k	a1	a2	a3a	a3b	a3c
128	430,1	12,3	5,45	3,579	3,5
127	443,65	12,4	83,55	11,131	3,45
6	22,55	11,7	78,75	28,311	8,75
5	18,15	11,9	82,45	25,178	6,05
4	14,45	10,9	82,65	19,763	6,7
3	12	11,35	81,65	19,937	4,9
2	8,85	11,45	78,85	14,705	3,9
1	5,05	11,8	43,8	10,146	3,4
0	1,9	11,45	4,8	3,489	3,4

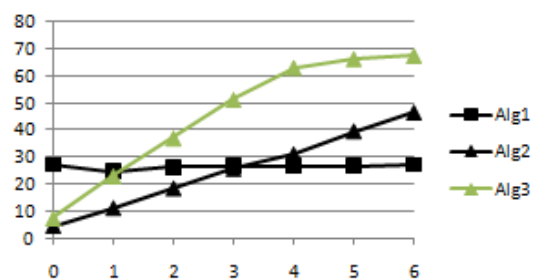


Obr. 6. Čas trvania algoritmu podľa tabuľky 6.

4.6 8 premenných

Tab. 7. n = 8

k	a1	a2	a3a	a3b	a3c
256	1727,1	24,9	10,9	7,1	6,9
255	1774,8	24,6	146,1	23,454	6,9
6	46,4	27,3	259,7	67,389	19
5	39,5	26,7	148,2	66,281	16,1
4	31	26,6	252,3	63,001	8,3
3	25,6	26,6	272,3	51,236	13,6
2	18,4	26,2	148,9	37,093	8,1
1	11,1	24,5	141,7	23,132	7
0	4,5	27,4	9,5	7,56	7,1



Obr. 7. Čas trvania algoritmu podľa tabuľky 7.

5. Interpretácia výsledkov

Algoritmus 1 nadobúda dobrý čas iba pri nízkych počtoch jednotiek. Naopak pri veľkom počte sa ukazuje ako veľmi pomalý. Náš teoretický predpoklad sa potvrdil.

Algoritmus 2 je veľmi efektívny, pretože čas na vykonanie programu je konštantný, nezávislý od počtu jednotiek. Odhad pre tento algoritmus bol správny.

Algoritmus 3 je závislý od počtu a polohy jednotiek vo vektore hodnôt. V najlepšom prípade je algoritmus najrýchlejší, ale v najhoršom prípade je značne pomalší.

V konečnom dôsledku vyslovujeme záver, že pre riedke Booleovské funkcie je v priemere najefektívnejší algoritmus 1, za ním nasleduje Algoritmus 3 a najneefektívnejší je Algoritmus 2. Pre niektoré špecifické Booleovské funkcie však je

najrýchlejší algoritmus 3, aj keď časový rozdiel je minimálny.

7. Použitá literatúra

[1] M.Polakovič, P.Kaprálík, J.Galanová. Logické systémy. Vydavateľstvo STU, 2007.

[2] Wikipedia. Algebraic normal form. Online zdroj, [citované 5.4.2010]

http://en.wikipedia.org/wiki/Algebraic_normal_form

[3] A. Joux. Algorithmic Cryptanalysis. CRC Press 2009. ISBN 978-1-4200-7002-6

[4] K. Nyberg. BOOLEAN FUNCTIONS. Online zdroj, [citované 5.4.2010]

<http://www.tcs.hut.fi/Studies/T-79.5501/2007SPR>