K-Map

Wllader

18. 10. 2023

1 Úvod

Karnaughova mapa je jedna z možností zápisu výsledku ohodnoecení logického výroku. Jiným takovým způsobem je například Pravdivostní tabulka.

V tomto dokumentu si definujeme, co je Karnaughova mapa, jak ji sestrojit, a jak s její pomocí dojít k co možná nejkratší konjunktivní (CNF), respektive disjunktní normální formě (DNF) původního výroku. K-Mapy k tomu využívají lidskou schopnost rychle a efektivně rozpoznávat šablony.

2 Terminologie

Na úvod jen několik formalit, abychom si rozuměli:

1. Výrok, nebo funkce se skládá z výrokových proměnných (prvovýroků)
2. Jazyk, nad kterým může být funkce definovaná, je množina všech výrokových proměnných: $\mathbb{P} = \{p, q, r, s\}$
3. Každá výroková proměnná může nabývat hodnoty 0, nebo 1. Při ohodnocení výroku (nebo dosazení hodnot do funkce) proměnným hodnoty přiřazujeme. Můžeme říct, že $v(p) \in \{0, 1\}$, ale pokud to bude z kontextu jasné, budeme zkracovat místo $v(p) = 1$ na $p = 1$.

3 Definice

Karnaughova mapa, zkráceně K-Map, pro f, funkci $k \in \mathbb{N}$ výrokových proměnných, je matice $K_f \in \{0, 1\}^{n \times m}$, kde platí:

1. $n \cdot m = 2^k$
2. Každý prvek matice odpovídá výsledku f pro jedno z možných ohodnocení
3. Každé dva sousední prvky se liší právě v ohodnocení jedné výrokové proměnné

4 Konstrukce

Konstrukce K je jednoduchá. Přímo plyne z pravdivostní tabulky a jde z ní dostat prakticky mechanicky, pokud je pravdivostní tabulka sestrojená následujícím způsobem:

1. Řádky PT jsou číslovány od 0 do $2^k - 1$
2. Proměnné jsou v opačném pořadí, než v $\mathbb{P}$
3. Ohodnocení každého řádku dává binární zápis čísla řádku.

Obecně může mít pravdivostní tabulka libovolnou posloupnost proměnných a jejich ohodnocení napříč řádky, ale tomuto formátu budeme říkat Výchozí tvar pravdivostní tabulky.

Pro účely demonstrace uvažme dvě funkce na $\mathbb{P} = \{p, q, r, s\}$

- $f(p, q, r, s)$: Neostrá majorita, tj.:

$$f = 1 \Leftrightarrow \left| \{p \in \mathbb{P} : v(p) = 1\} \right| \geq \frac{|\mathbb{P}|}{2} = 2$$

- $g(p, q, r, s)$: Neostrá minorita, tj.:

$$g = 1 \Leftrightarrow \left| \{p \in \mathbb{P} : v(p) = 1\} \right| \leq \frac{|\mathbb{P}|}{2} = 2$$

Zde je jejich pravdivostní tabulka ve Výchozím tvaru:

R	s	r	q	p	f	g
0	0	0	0	0	0	1
1	0	0	0	1	0	1
2	0	0	1	0	0	1
3	0	0	1	1	1	1
4	0	1	0	0	0	1
5	0	1	0	1	1	1
6	0	1	1	0	1	1
7	0	1	1	1	1	0
8	1	0	0	0	0	1
9	1	0	0	1	1	1
10	1	0	1	0	1	1
11	1	0	1	1	1	0
12	1	1	0	0	1	1
13	1	1	0	1	1	0
14	1	1	1	0	1	0
15	1	1	1	1	1	0

Konstrukce K je obdobná. Řádky a sloupce jsou indexované možnými hodnotami proměnných a každý prvek je na pozici, která při složení bitů ze sloupců a řádek v odpovídajícím pořadí dá binární číslo řádku pravdivostní tabulky. Zavádět pravidla pro konstrukci K formálně a numericky lze, ale není to potřeba. Mnemotechnika lze jednoduše vykoukat z obrázku:

$\frac{qp}{sr}$	00	01	11	10
00	0	1	3	2
01	4	6	7	6
11	12	13	15	14
10	8	9	11	10

Zde je každý prvek K číslo řádku v pravdivostní tabulce, kterému odpovídá ohodnocení proměnných, kterým je indexovaný.

K je ale s PV redundantní, není tedy třeba ani tvořit PV a na pořadí, v jakém zvolíme proměnné vyplnit do K pak nezáleží. Důležitá jsou pravidla, která K musí splňovat, aby šlo o K-Mapu.

Nyní sestrojme K-Mapu pro funkci f, např. takto:

$\frac{qp}{sr}$	00	01	11	10
00	0	0	1	0
01	0	1	1	1
11	1	1	1	1
10	0	1	1	1

Pro méně psaní při návrhu tabulky do sešitu a pro některé lepší přehled při vyčítání hodnot proměnných se většinou používá grafické znázornění ohodnocení:

f		p	q	
	0	0	1	0
r	0	1	1	1
s	1	1	1	1
	0	1	1	1

Zde jednotlivé proměnné platí ($v(p) = 1$), právě když jsou ve sloupci, nebo řádku, který je nadtržený odpovídající čarou. Samotný prvek matice pak odpovídá funkční hodnotě f pro toto ohodnocení.

Nyní jak K použít k sestavení f_{CNF} a f_{DNF}.

4.1 Sestrojení DNF

V K nyní budeme hledat podmatice Q, které splňují:

1. $Q \in \{0,1\}^{n' \times m'}$
2. $n' \cdot m' = 2^{k'}, k' \in \mathbb{N}$
3. $Q_{i,j} = 1,\ \forall i \leq n', j \leq m'$

Taková podmatice Q definuje jednu zredukovanou konjunkci takto:

- Pokud je pro všechny prvky Q hodnota $v(p) = 1 : p \in Q$, pak je ve výsledné konjunkci p
- Pokud je pro všechny prvky Q hodnota $v(p) = 0 : p \in Q$, pak je ve výsledné konjunkci $\neg p$
- Pokud se hodnota proměnné pro libovolné dva prvky Q liší, pak se ve výsledné konjunkci p ani $\neg p$ nenachází

Z toho vyplývá, že abychom dosáhli co největší redukce oproti elemntárním konjunkcím, které lze vyčíst z Pravdivostní tabulky, chceme hledat co největší podmatice Q.

Podmatic většinou musíme najít několik, aby byla výsledná forma kompletní. Kompletnost výsledného výroku poznáme tak, že jsme každý prvek $p \in K$, jehož hodnota je 1 použili alespoň v jedné podmatici Q_i.

Nyní

$$f_{DNF} = \bigvee_{i=1}^{|\mathcal{Q}|} Q_i$$

kde $\mathcal{Q}$ je množina všech nalezených $Q_i \subseteq K$

4.2 Sestrojení CNF

Podobně jako u hledání DNF budeme nyní hledat podmatice $\hat{Q}$, které ale tentokrát splňují:

1. $\hat{Q} \in \{0,1\}^{n' \times m'}$
2. $n' \cdot m' = 2^{k'}, k' \in \mathbb{N}$
3. $\hat{Q}_{i,j} = 0,\ \forall i \leq n', j \leq m'$

A analogicky taková podmatice $\hat{Q}$ definuje jednu zredukovanou disjunkci:

- Pokud je pro všechny prvky $\hat{Q}$ hodnota $v(p) = 0 : p \in \hat{Q}$, pak je ve výsledné disjunkci p
- Pokud je pro všechny prvky $\hat{Q}$ hodnota $v(p) = 1 : p \in \hat{Q}$, pak je ve výsledné disjunkci $\neg p$
- Pokud se hodnota proměnné pro libovolné dva prvky $\hat{Q}$ liší, pak se ve výsledné konjunkci p ani $\neg p$ nenachází

A analogicky pak dostáváme, že

$$f_{CNF} = \bigwedge_{i=1}^{|\hat{\mathcal{Q}}|} \hat{Q}_i$$

kde $\hat{\mathcal{Q}}$ je množina všech nalezených $Q_i \subseteq K$

5 Příklady

Pojďme pomocí konstrukce, kterou jsme si ukázali, najít normální formy f a g.

5.1 f_{DNF}

Pro zopakování, tady je K_f:

f		p	q	
	0	0	1	0
r	0	1	1	1
s	1	1	1	1
	0	1	1	1

Instinkt pro hledání největší jedničkové podmatice by velel označit devět jedniček v pravém dolním rohu, ale pozor, $9 \neq 2^k$ - musíme se spokojit se čtyřmi maticemi 2×2.

Pojmenujme je Q_1 - Q_4

Chybí nám ještě jedničky vlevo a nahoře. Mohli bychom je označit samostatně, protože $1 = 2^0$, ale my chceme co největší podmatici. Tady můžeme najít celý řádek 1×4, respektive sloupec 4×1, a pojmenovat je Q_5 a Q_6.

f		p	q	
	0	0	1	0
r	0	1	1	1
s	1	1	1	1
	0	1	1	1

Všechny jedničky máme nyní označené, tudíž můžeme z $\mathcal{Q}$ vyjádřit jejich konjunkce:

- Pro Q_1 je $p = 1$ v obou sloupcích a $r = 1$ v obou řádcích, proto obě tyto proměnné budou v konjunkci bez negace. Naopak q a s mezi sloupci, respektive řádky Q_1 hodnotu mění, proto v konjunkci nebudou vůbec.

$\rightarrow$ $Q_1 = p \wedge r$

- Pro Q_2 až Q_4 je postup analogický:

$\rightarrow$ $Q_2 = q \wedge r$

$\rightarrow$ $Q_3 = p \wedge s$

$\rightarrow$ $Q_4 = q \wedge s$

- Pro Q_5 po celém řádku platí $r, s = 1$, ale p, q svou hodnotu mění, proto:

$\rightarrow$ $Q_5 = r \wedge s$

- A analogicky pro sloupec Q_6:

$\rightarrow$ $Q_6 = p \wedge q$

Nyní už stačí jen jednotlivé konjunkce spojit do kompletní f_{DNF}:

$$(p \wedge r) \vee (q \wedge r) \vee (p \wedge s) \vee (q \wedge s) \vee (r \wedge s) \vee (p \wedge q)$$

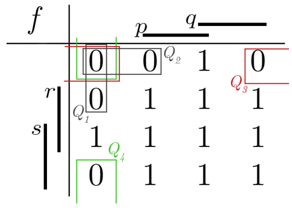
5.2 f_{CNF}

Analogicky, ale nyní hledáme nuly. To bude u funkce f trochu záludnější.

Triviální dvě dvojice v levém horním rohu budiž $\hat{Q}_1$ a $\hat{Q}_2$

Zbylé nuly bychom mohli označit samostatně, ale tomu se chceme vyhnout. Bude se nám hodit trik, který jsem i pečlivému čtenáři až do teď tajil, a sice, že Karnaughovy mapy jsou horizontálně i vertikálně cyklické - tedy tvoří něco, čemu by matematik řekl torus a barbarští fyzici zase donut.

V každém případě můžeme cyklicky levou horní nulu spojit do dvojic s pravou horní a levou dolní, abychom vytvořili $\hat{Q}_3$ a $\hat{Q}_4$.



Nyní ke stavění disjunkcí - tady pozor na rozdíl od konjunkcí, neznegované proměnné jsou tam, kde je jejich hodnota v K_f nulová:

- Pro Q_1 je hodnota $p, q, s = 0$ a r přeskakuje, tudíž:

→ $Q_1 = p \vee q \vee s$

- Podobně Q_2:

→ $Q_2 = q \vee r \vee s$

- K cyklicky zvoleným Q_i se budeme chovat stejně

→ $Q_3 = p \vee r \vee s$

→ $Q_4 = p \vee q \vee r$

Dohromady dostáváme:

$$f_{CNF} = (p \vee q \vee s) \wedge (q \vee r \vee s) \wedge (p \vee r \vee s) \wedge (p \vee q \vee r)$$

5.3 g_{DNF}

Doteď to byla trochu nuda, nepodařilo se nám dostat ani jeden případ negované proměnné. Zkusme si tedy převést ještě druhou funkci, g, do jejích normálních forem.

Nejdřív sestavme K_g:

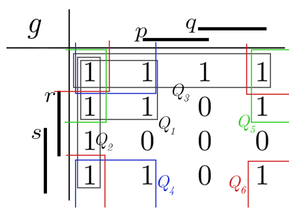
$\frac{qp}{sr}$	00	01	11	10
00	1	1	1	1
01	1	1	0	1
11	1	0	0	0
10	1	1	0	1

Nyní najděme největší jedničkové podmatice. Jednoduché úlovky jsou čtveřice vlevo nahoře, první sloupec a první řádek: Q_1 - Q_3

Další by intuitivně mohla být dvojice dole, vpravo nahoře, a cyklická dvojice s libovolnou jedničkou nahoře, nebo dole, ale to není nejlepší řešení (ale fungovalo by, jen by výsledek byl méně redukovaný)

Trochu těžší na vypátrání jsou 2×2 matice cyklicky v prvních dvou sloupcích, Q_4, a v prvních dvou řádcích, Q_5.

Stále nám ale schází jednička vpravo dole. Cyklicky horizontálně i vertikálně ji můžeme spojit do 2×2 matice se všemi ostatními jedničkami v rozích, aby vznikla poslední Q_6.



To byl o trochu tajuplnější příklad. Teď vyčíst konjunkce:

- Hned pro Q_1 platí, že $q, s = 0$, což u konjunkcí znamená, že jsou přítomny jejich negace. p, r oscilují, takže je vynecháme:

$\rightarrow$ $Q_1 = \neg q \wedge \neg s$

- Analogicky:

$\rightarrow$ $Q_2 = \neg p \wedge \neg q$

$\rightarrow$ $Q_3 = \neg r \wedge \neg s$

- U cyklických podmatic je to trochu hůř vidět, ale dostaneme:

$\rightarrow$ $Q_4 = \neg q \wedge \neg r$

$\rightarrow$ $Q_5 = \neg p \wedge \neg s$

$\rightarrow$ $Q_6 = \neg p \wedge \neg r$

Výslednou funkcí je pak disjunkce všech našich redukovaných konjunkcí.

5.4 g_{CNF}

Proti poslednímu příkladu je výpočet CNF pro g už jen formalitou:

Všechny matice budou dvojice a všechny se budou dělit o prostřední nulu, začněme pojmenovávat zprava a po směru hodinových ručiček $\hat{Q}_1$ - $\hat{Q}_4$.

g		p	p, q	q
	1	1	1	1
r	1	1	0	1
r, s	1	0	0	0
s	1	1	0	1

Vyčtěme z nich disjunkce:

$\rightarrow$ $\hat{Q}_1 = \neg q \vee \neg r \vee \neg s$

$\rightarrow$ $\hat{Q}_2 = \neg p \vee \neg q \vee \neg s$

$\rightarrow$ $\hat{Q}_3 = \neg p \vee \neg r \vee \neg s$

$\rightarrow$ $\hat{Q}_4 = \neg p \vee \neg q \vee \neg r$

a g_{CNF} bude jejich konjunkce.

6 "Don't care" případy

Občas se při řešení úloh v praxi naskytne případ, ve kterém nejsou dobře definované případy pro všechna ohodnocení. Typicky pokud k některé konstelaci ohodnocení proměnných jednoduše nemůže dojít, nemá smysl pro takový případ definovat výstup.

Takovému případu říkáme "Don't care case" a při vyčítání redukovaných disjunkcí, nebo konjunkcí, jej můžeme použít jako obojí, jedničku, i nulu. To je přirozeně velmi výhodné, protože v obou případech nám přibudou prvky, ze kterých potenciálně sestavit větší podmatice.

Jako příklad rychle uveďme alternaci pro funkci g, g', která bude stále fungovat jako Minorita, ale v případě remízy počtu nulových a nenulových proměnných nás výstup funkce nezajímá:

$\frac{qp}{sr}$	00	01	11	10
00	1	1	–	1
01	1	–	0	–
11	–	0	0	0
10	1	–	0	–

Samotný výpočet g'_{DNF} a g'_{CNF} necháme čtenáři jako cvičení.

7 Libovolný počet proměnných

K-Mapy jsme si představili na funkcích čtyřech logických proměnných, což je asi druhý, nebo třetí nejvyšší počet, s jakým je praktické pracovat.

Pro rozšíření na libovolný počet stačí rozvržení mapy s každou další proměnnou zrcadlit libovolným směrem. V případě, že by proměnné byly tři, mapa bude poloviční, než v našich případech:

$\frac{qp}{r}$	00	01	11	10
0	0	1	3	2
1	4	6	7	6

Pro dvě proměnné:

$\frac{qp}{r}$	00	01
0	0	1
1	2	3

Pro pět proměnných:

$\frac{rqp}{us}$	000	001	011	010	110	111	101	100
00	0	1	3	2	6	7	5	4
01	8	9	11	10	14	15	13	12
10	24	25	27	26	30	31	29	28
11	16	17	19	18	22	23	21	20

K-Mapy je možné dělat libovolně velké, ale pro praktické využití pravděpodobně nebudete používat nic většího, než 8×8 (tedy K-Mapu pro 6 proměnných).

Vzhledem k tomu, že princip K-Map těží z lidské schopnosti rychle rozeznávat vzorce, pro větší jazyky se ani nevyplácí jejich implementaci v SW, jelikož existují mnohem přístupnější metody jak podobné problémy řešit algoritmicky.

Pro rychlé zjednodušení malých problémů na papíře jsou ale poměrně silným nástrojem, ať už jde o slaboproudé logické obvody typu mikrovlnné trouby, senzory rozbitého skla a podobně, nebo třeba wiring v Terrarii, nebo redstone v Minecraftu.

8 Závěr

V tomto dokumentu jsme si představili Karnaughovy mapy, ukázali, jak je možné je zkonstruovat pro libovolné (ale preferovaně malé) velikosti jazyka, a jak pomocí nich efektivně najít redukované normální formy výroků a funkcí, pro které známe všechna ohodnocení. Zmínili jsme, co jsou a jak je možné řešit "Don't care cases" a vysvětlili si užitečnost pro specifické obory malých problémů.