doc. RNDr. Ondřej Čepek, Ph.D.

Boolovské funkce a jejich aplikace

dle přednášky zapsali Tomáš Kuča a Jindřich Vodrážka
ITI 2010

1. Úvod do Boolovských funkcí

V této kapitole nadefinujeme literály, logické operátory, Boolovské formule a funkce. Popíšeme normální tvary formulí (*DNF*, *CNF*) a jejich vlastnosti. Dále zavedem implikanty, konsekventy, rezoluci (konsensus). V závěru popíšeme rezoluční metodu a dokážeme její úplnost. item

Boolovské funkce

- x, y, z jsou *boolovské funkce*
- $v(x) \in \{0, 1\}$ je *ohodnocovací funkce*
- $\neg x, \overline{x}$ je *negace* x
- proměnné a proměnné s negací jsou *literály*
- $\vee, \wedge$ ($\Rightarrow$, $\iff$) jsou binární operátory

Definice. *(Boolovská formule)*
Boolovskou formuli definujme rekurzivně:

- každý literál je *Boolovská formule* (dále jen *formule*)
- pokud jsou A, B formule, pak $\neg A, \neg B, A \vee B, A \wedge B, A \Rightarrow B, A \iff B$ jsou také formule

Definice. *(Normální formy)*
Disjunkci literálů nazýváme *klauzule*, konjunkci literálů nazýváme *term*. Formule je v *konjunktivní normální formě* (*CNF*), pokud je konjunkcí klauzulí. Formule je v disjunktivní normální formě (*DNF*), pokud je disjunkcí termů.

Definice. *(Ekvivalentní formule)*
Následující pravidla definují ekvivalentní formule, lze je využít k úpravě formulí:

1. $\neg(A \vee B) \equiv \neg A \wedge \neg B$ (de Morganovo pravidlo)
2. $\neg(A \wedge B) \equiv \neg A \vee \neg B$ (de Morganovo pravidlo)
3. $A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$ (distributivita $\vee$)
4. $A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$ (distributivita $\wedge$)

Definice. *Boolovská funkce* je funkce $f : \{0, 1\}^n \to \{0, 1\}$. Lze ji reprezentovat např. formulí nebo tabulkou.

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

Vektory, pro které $f = 1$, nazýváme *true-pointy*, ostatní, pro které $f = 0$ označujeme jako *false-pointy*.

Nebo na *DNF* (disjunkce termů odpovídajících true-pointum):

Pozorování. Ke každé formuli existuje logicky ekvivalentní formule v *CNF* i *DNF*.
Důkaz. Pokud funkce reprezentujeme tabulkou, pak můžeme sestavit normální formy:

- *DNF*: disjunkce termů "odpovídajících" true-pointům dané funkce

$$f(x_1, x_2, x_3) = (\neg x_1 \wedge x_2 \wedge \neg x_3) \vee (x_1 \wedge \neg x_2 \wedge \neg x_3) \vee (x_1 \wedge \neg x_2 \wedge x_3) \vee (x_1 \wedge x_2 \wedge x_3)$$

- *CNF*: konjunkce klauzulí "odpovídajících" false-pointům dané funkce

$$f(x_1, x_2, x_3) = (\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee \neg x_3)$$

Pozorování. Počet boolovských funkcí na n proměnných je 2^{2^n} (tabulka má 2^n řádků, na každém může být hodnota f 1 nebo 0).

Definice. *(Uspořádání boolovských funkcí)*
Definujme relaci $>$ takto: $f > g$ právě tehdy, když $\forall x$: $f(x) > g(x)$. Odbobně $\geq, \leq, <$.

V následující části se zaměříme na *DNF*. Věci, které dokážeme, platí analogicky i pro *CNF*.

Definice. Term T je *implikantem* funkce f, pokud $T = 1 \Rightarrow f = 1$ (korektně $\forall x T(x) = 1 \Rightarrow f(x) = 1$).
Term T je *primární implikant*, pokud po vypuštění libovolného literálu přestane být implikantem.

Příklad. $\underbrace{x_1\overline{x_2}}_{primární} \vee \underbrace{x_1x_3\overline{x_4}}_{neprimární} \vee \underbrace{x_2\overline{x_4}}_{primární} \vee \underbrace{x_2\overline{x_3}}_{primární}$

Druhý term není primární implikant, neboť $x_1\overline{x_4}$ je též implikant.

Tvrzení. Pokud *DNF* $\mathcal{F}$ reprezentuje funkci f a T je implikant f, pak $\mathcal{F} \vee T$ také reprezentuje f.
Důkaz.

- $\mathcal{F} \leq \mathcal{F} \vee T$ triviálně
- $\mathcal{F} \geq \mathcal{F} \vee T$ z definice implikantu

Definice. Pokud T_1, T_2 jsou termy a $T_1 \leq T_2$ (tj. T_2 je kratší, $T_2 \leq T_1$ v množinovém smyslu), říkáme, že T_2 *absorbuje* T_1.

Pozorování. Pokud $\mathcal{F} \vee T_1 \vee T_2$ reprezentuje f a $T_1 \leq T_2$, pak $\mathcal{F} \vee T_2$ reprezentuje tutéž funkci.

Definice. Termy T_1 a T_2 mají *konflikt* v proměnné x, pokud $x \in T_1$ a $\neg x \in T_2$ (nebo $\neg x \in T_1$ a $x \in T_2$) a řekneme, že T_1 a T_2 mají *konsenus*, pokud mají konflikt v právě jedné proměnné. Označíme-li termy $T_1 = A \wedge x$ resp. $T_2 = B \wedge \neg x$, pak $Cons(A, B) = A \wedge B$.

$$\left.\begin{array}{l} T_1 = A \wedge x \\ T_2 = B \wedge \bar{x} \end{array}\right\} Cons(T_1, T_2) = A \vee B$$

Tvrzení. Nechť T_1, T_2 jsou implikanty f mající konsenzus. Pak $T_3 = Cons(T_1, T_2)$ je také implikantem.
Důkaz.

- $T_1 \leq f, T_2 \leq f$
- $T_1 = A \wedge x$
- $T_2 = B \wedge \overline{x}$
- $T_3 = A \wedge B$

$T_3 = 1 \Rightarrow (A = 1) \wedge (B = 1) \Rightarrow (T_1 = 1) \vee (T_2 = 1) \Rightarrow f = 1$

Definice. *Kanonická DNF* funkce f je disjunkce všech primárních implikantů funkce f.

Algoritmus. *(Konsenzuální metoda)*
Vstup: DNF reprezentující f

1. Dokud se $\mathcal{F}$ může změnit:
2. proveď všechny absorbce, které je možné provést
3. najdi dva termy T_1, T_2, že $Cons(T_1, T_2)$ není absorbován žádným termem v $\mathcal{F}$ a přidej $Cons(T_1, T_2)$ k $\mathcal{F}$

Výstup: kanonická *DNF* reprezentující f

Věta. *(Úplnost konsenzuální metody)*

1) Konečnost: žádný term není přidán dvakrát (díky transitivitě absorbce)
2) Koreknost: nechť $\mathcal{F} = R_1 \vee R_2 \vee \ldots \vee R_m$ je výstupem algoritmu a nechť P je primární implikant f, který není obsažen v $\{R_1, \ldots, R_m\}$. Označme $\mathcal{P} = \{T | T \geq P \& \forall l : T$ má konflikt s $R_l\}$.

   TODO: ověřit definici $\mathcal{P}$

   Množina $\mathcal{P}$ je neprázdná, obsahuje alespoň samotné P.

   Nechť P^* je nejdelší (dle počtu literálů) term v $\mathcal{P}$.

   i) $\|P^*\| = \|\mathcal{F}\|$,
   pak P^* má konflikt s každým R_l. Platí $P^* = 1 \Rightarrow \forall l : R_l = 0 \Rightarrow f = 0$, ale také $P^* = 1 \Rightarrow P = 1 \Rightarrow f = 1$, což je spor.

   ii) $\|P^*\| < \|\mathcal{F}\|$,

pak $\exists x$ proměnná neobsažená v P^*
$P^*x \notin \mathcal{P} \Rightarrow \exists i$, že $P^*x \leq R_i \Rightarrow R_i$ obsahuje x a část P^* $P^*\overline{x} \notin \mathcal{P} \Rightarrow \exists j$, že $P^*\overline{x} \leq R_j \Rightarrow R_j$ obsahuje $\overline{x}$ a část P^*, což ale znamená, že $Cons(R_i, R_j)$ je obsazen v $P^*(Cons(R_i, R_j) \geq P^*)$, tj. $\not\exists R_l$ absorbující $Cons(R_i, R_j)$, což je spor s fungováním algoritmu.

Složitost konsenzualní metody.

Oba kroky, měnící $\mathcal{F}$ v algoritmu konsenzualní metody lze provést v polynomiálním čase. Cyklus se ovšem může mít exponenciální počet iteraci. Ve speciálních případech (napr. pokud je omezena délka klauzule, jako třeba pro 2-SAT, lze dosáhnout lepších výsledků).

Příklad.

$$\mathcal{F} = x_1x_2 \ldots x_n \vee \bar{x_1}y_1 \vee \bar{x_2}y_2 \vee \ldots \vee \bar{x_n}y_n$$

Kanonická formule k $\mathcal{F}$ je exponenciálně dlouhá vzhledem k délce vstupní formule.

2. Monotónní a regulární funkce

V této kapitole zavedeme pojmy monotónní a regulární funkce a s tím související pojmy.

Monotónní booleovské funkce.

Definice.
Funkce f je *pozitivní v proměnné* x_i, pokud $\forall x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$ platí:

$$f(x_1, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n) \geq f(x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n)$$

Obdobně pokud je funkce f *negativní v proměnné* x_i (pouze opačná nerovnost):

$$f(x_1, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n) \leq f(x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n)$$

Pozorování. Pokud DNF formule $\mathcal{F}$ neobsahuje žádný negovaný literál x_i, reprezentuje booleovskou funkci, která je *pozitivní v* x_i. Tvrzení platí i obráceně, pokud se omezíme pouze na *primární* DNF.

Tvrzení. Nechť f je pozitivní v x_i a nechť T je primární implikant f. Potom T neobsahuje $\bar{x_i}$.
Důkaz. Sporem, nechť $A\overline{x}_i$ je primární implikant funkce f. Přečíslujeme proměnné tak, aby A obsahovala právě jenom $x_1, \ldots, x_{i-1}$. $\forall x_{i+1}, \cdots, x_n$ platí $A\overline{x}_i$ je implikant (díky $1 = f(\underbrace{0, 1, \cdots}_{A}, \underbrace{0}_{x_i}, x_{i+1}, \ldots, x_n) \leq f(\underbrace{0, 1, \cdots}_{A}, 1, x_{i+1}, \ldots, x_n) \Rightarrow Ax_i$ je implikant $f \Rightarrow Cons(Ax_i, A\overline{x}_i) = A$ je implikant f, tedy $A\overline{x}_i$ nemohl být primární implikant.

Definice. Funkce f je *pozitivní/negativní*, pokud je pozitivní/negativní ve všech proměnných. Všechny funkce, které jsou pozitivní, nebo negativní se nazývají *monotónní*. Funkce, která je v každé proměnné pozitivní nebo negativní se nazývají *unátní*.

Pozorování. Z pozitivní *DNF* lze v polynomiálním čase vyrobit kanonickou *DNF* (stačí absorbce).

Regulární funkce.

Definice. Nechť f je booleovská funkce na n proměnných. Řekneme, že x_i *je silnější než* x_j *vzhledem k* f [1], pokud $\forall x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n$:

$$f(x_1, \ldots, x_{i-1}, \mathbf{1}, \ldots, x_{j-1}, \mathbf{0}, \ldots, x_n) \geq f(x_1, \ldots, x_{i-1}, \mathbf{0}, \ldots, x_{j-1}, \mathbf{1}, \ldots, x_n)$$

Značení. Značíme následovně: $x_i \succeq_f x_j$. Zavedeme též symboly $\succ_f$ a $\sim f$:

$$x_i \succ_f x_j \iff x_i \succeq_f x_j \wedge \neg(x_j \succeq_f x_i)$$
$$x_i \sim_f x_j \iff x_i \succeq_f x_j \wedge x_j \succeq_f x_i$$

[1] BUNO $i < j$

Věta. Relace $\succeq_f$ je předuspořádání na množině proměnných.
Důkaz. Dokážeme, že relace je reflexivní a tranzitivní.

- reflexivita: Dodefinujeme relaci $\succeq_f$ pro $i = j$.
- tranzitivita: Dokážeme, že pro libovolné i, j, k (BUNO $i < j < k$) platí:

$$x_i \succeq_f x_j \wedge x_j \succeq_f x_k \iff x_i \succeq_f x_k$$

Z definice relace $\succeq_f$ odvodíme:

1) $f(..\underbrace{1}_{i}..\underbrace{0}_{j}..\underbrace{0}_{k}..) \underbrace{\geq}_{x_i \succeq_f x_j} f(..\underbrace{0}_{i}..\underbrace{1}_{j}..\underbrace{0}_{k}..) \underbrace{\geq}_{x_j \succeq_f x_k} f(..\underbrace{0}_{i}..\underbrace{0}_{j}..\underbrace{1}_{k}..)$

2) $f(..\underbrace{1}_{i}..\underbrace{1}_{j}..\underbrace{0}_{k}..) \underbrace{\geq}_{x_j \succeq_f x_k} f(..\underbrace{1}_{i}..\underbrace{0}_{j}..\underbrace{1}_{k}..) \underbrace{\geq}_{x_i \succeq_f x_j} f(..\underbrace{0}_{i}..\underbrace{1}_{j}..\underbrace{1}_{k}..)$

Tedy dohromady $\forall x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{k-1}, x_{k+1}, \ldots, x_n$:

$$f(..\underbrace{1}_{i}..\underbrace{0}_{k}..) \geq f(..\underbrace{0}_{i}..\underbrace{1}_{k}..)$$

Q.E.D.

Definice.
Pozitivní booleovská funkce f je *regulární*, pokud uspořádání podle $\succeq_f$ je úplné uspořádání.

3. Algoritmus pro testování regularity funkcí

Cílem této kaiptoly je popsat algoritmus na rozpoznávání regulárních funkcí. V průběhu celé kapitoly uvažujeme, že na vstupu je pozitivní DNF formule.

Věta. Nechť f je pozitivní boolovská funkce v n proměnných a $\mathcal{F}$ je kanonická DNF funkce f. Zapišme $\mathcal{F}$ ve tvaru

$$\mathcal{F} = Ax_ix_j \vee Bx_i \vee Cx_j \vee D,$$

kde x_i, x_j jsou lib. proměnné a A, B, C, D jsou DNF na zbylých proměnných. Pak $x_i \succeq x_j$ právě tehdy, když $B \geq C$.
Důkaz. BÚNO $i = 1$, $j = 2$. Z definice $\succeq$ dostaneme $x_i \succeq x_j \iff f(1, 0, \ldots) \geq f(0, 1, \ldots) \iff B \vee D \geq C \vee D$. Stačí tedy dokázat $B \vee D \geq C \vee D \iff B \geq C$.
$\Leftarrow$: platí triviálně
$\Rightarrow$: stačí ukázat, že každý implikant $p \in C$ je absorbován nějakým implikantem z B.

$$p \text{ implikant } C \begin{cases} px_j \text{ je implikant } f, \text{ který je primární} \\ p \text{ je absorbován implikantem } q \text{ v } B \vee D. \end{cases}$$

q nemůže být v D, protože pak by q absorbovalo px_2, a tedy by $\mathcal{F}$ nebyla kanonická.

Jak ověřit, že $B \geq T$?

Dosadíme do B jediné ohodnocení proměnných z T, pro které $T = 1$ a po dosazení ověříme falzifikovatelnost DNF B', která vznikne. Pokud B' falzifikuje, tak $B \not\geq T$, pokud $B = 1$, tak $B \geq T$.

Nebo také méně formálně: ověřuju-li, že $B \geq C$ a B, C jsou pozitivní formule, stačí ověřit, že všechny implikanty v C jsou absorbované nějakým implikantem v B. V obecném případě je to těžké, pro pozitivní formule stačí nalézt podtermy.

Jak ověřit, že $B \sim T$? Pro každé i, j otestujeme, zda $x_i \succeq x_j$ a $x_j \succeq x_i$ v čase $\Theta(nm^2)$, kde n je počet proměnných formule a m počet termů.

Algoritmus na rozpoznávání regularity

1) $\Theta(n^3m^2)$: otestuj všechny dvojice proměnných
2) $\Theta(n^2m^2 \log n)$: pomocí třídícího algoritmu
3) $\Theta(n^2m^2)$:
   Definujeme W matici $n \times n$, kde $W[k, j] = \#$ termů obsahujících x_k délky j. Její řádky označme R_i.
   a) $x_i \sim x_j \Rightarrow R_i = R_j$

b) $x_i \succeq x_j \Rightarrow R_i > R_j$ (dle lexikografického uspořádání)

Důkaz.

a) $x_i \sim x_j \iff B = C$ (ve smyslu věty **??**) $\Rightarrow$ nechť p je implikant C, díky $\geq \exists q \in B$ absorbující p a obráceně, tj. $p = q \Rightarrow B$ a C jsou identické DNF

b) Chceme $x_i \succ x_j \Rightarrow R_i > R_j$. Platí $x_i \succ x_j \iff B > C$. Definujme

$$B(d) = \{p||p| = d \& Tx_i \text{ je term v } B\}$$
$$C(d) = \{p||p| = d \& Tx_j \text{ je term v } C\}$$

Nechť $\delta = min\{d|B(d) \neq C(d)\}$. Takové δ existuje, neboť $B > C$. Nechť $p \in C(d)$. Z $B > C$ plyne, že $\exists$ term $T \vee B$ absorbující $\prod_{k \in P} x_k$

1) $|Q| < |P|$ nemůže nastat, protože pak $T \in C$ (až do δ jsou $B(d)$ i $C(d)$ totožné, a tedy C není kanonická.
2) $|Q| = |P|$, pak $Q = P$ a $\prod_{k \in P} x_k \in B \Rightarrow C(\delta) \subseteq B(\delta)$, zároveň $C(\delta) \neq B(\delta)$ (z definice δ), tedy $R_i > R_j$

Algoritmus nejprve sestrojí matici, poté lexikograficky setřídí její řádky a nakonec pro každé dva sousední řádky ověří, že pro příslušná i, j platí $x_i \succeq x_j$.

Složitost: Inicializace matice proběhne v čase $\mathcal{O}(n^2)$. Hodnoty v matici lze spočítat v $O(nm)$ jedním průchodem formule $\mathcal{F}$. Setřídění řádků matice lze provést přihrádkovým tříděním v lineárním čase.

Poznámka. Vhodnou datovou strukturou lze porovnání $x_i \succeq x_j$ provést v $\mathcal{O}(nm)$ a celý algoritmus má pak složitost $\mathcal{O}(n^2 m)$.

4. Duální funkce

V této kapitole zavedeme pojem duálních funkcí a popíšeme jejich vlastnosti. V závěru popíšeme algoritmus, který převede regulární funkci na její duál.

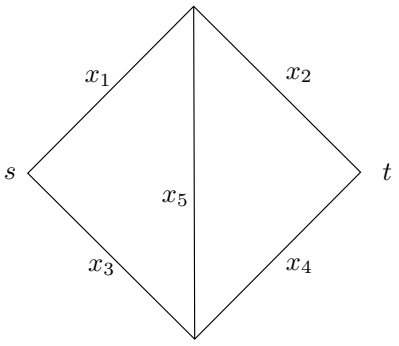
Definice. *(Duální funkce)*
Nechť $f(x_1, \ldots, x_n)$ je booleovská funkce. Pak *duální boolovská funkce* f^d je definována předpisem $f^d(x) = \neg f(\neg x)$.

Příklad.

$$\begin{aligned} f(x_1, \ldots, x_5) &= x_1x_2 \vee x_3x_4 \vee x_1x_4x_5 \vee x_2x_3x_5 \\ f^d(x_1, \ldots, x_5) &= \neg f(\neg x_1, \ldots, \neg x_5) = \\ &= \neg(\overline{x_1x_2} \vee \overline{x_3x_4} \vee \overline{x_1x_4x_5} \vee \overline{x_2x_3x_5} \\ &= (x_1 \vee x_2) \wedge (x_3 \vee x_4) \wedge (x_1 \vee x_4 \vee x_5) \wedge (x_2 \vee x_3 \vee x_5) \\ &= (x_1 \vee x_2x_4 \vee x_2x_5) \wedge (x_3 \vee x_2x_4 \vee x_4x_5) \\ &= (x_1x_3 \vee x_2x_4 \vee x_1x_4x_5 \vee x_2x_3x_5) \end{aligned}$$

Duální funkce má svou interpretaci v teorii grafů:

f ..... všechny cesty st
f^d ..... všechny řezy



Složitost dualizace je neznámá. Je známý kvazi-polynomiální algoritmus ve velikosti výstupu. DNF zápis duální funkce obecně může být exponenciálně dlouhý k délce DNF zápisu f:

$$\begin{aligned} f(x_1,\ldots,x_n) &= x_1x_2 \vee x_3x_4 \vee \ldots \vee x_{2n-1}x_{2n} \\ f^d(x_1,\ldots,x_n) &= (x_1 \vee x_2)(x_3 \vee x_4)\ldots(x_{2n-1} \vee x_{2n}) \\ &= 2^n \text{ termů délky } n \end{aligned}$$

Definice. $\vec{x}$ je *minimální true-point* funkce f (*MTP*) $\iff f(\vec{x}) = 1 \& \forall i : f(\vec{x} - e_i) = 0 \iff \vec{x}$ odpovídá primární implikantu f

$\vec{x}$ je *maximální false-point* funkce f (*MFP*) $\iff f(\vec{x}) = 0 \& \forall i : f(\vec{x} \cup e_i) = 1 \iff \vec{x}$ odpovídá primární implikantu f^d

Pozorování. Nechť $\prod_{i\in I} x_i$ je primární implikant v f^d. Pak I je minimální množina indexů taková, že

$$\left.\begin{aligned} x_i &= 1, i \in I \\ x_i &= 0, i \notin I \end{aligned}\right\} \Rightarrow f^d(x) = 1$$

$f^d(x) = 1 \iff \neg f(\neg x) = 0 \iff f(\neg x) = 0$

I je minimální množina indexů taková, že

$$\left.\begin{aligned} x_i &= 0, i \in I \\ x_i &= 1, i \notin I \end{aligned}\right\} \Rightarrow f(x) = 0$$

$\Rightarrow$ nuly v $\vec{x}$ odpovídají proměnným v primárním implikantu f^d.

V následující části ukážeme, že pro dualizaci pozitivní DNF nám stačí algoritmus, který na vstupu dostane seznam MTP a na výstupu vydá seznam *MFP*.

Věta.

Nechť $f(x_1,\ldots,x_n)$ je regulární funkce, kde $x_1 \succeq_f x_2 \succeq_f \ldots \succeq_f x_n$ a nechť $x \in \{0,1\}^{n-1}$. Pak $(\vec{x},0)$ je *MFP* $\iff$ $(\vec{x},1)$ je *MTP*

Důkaz.

Nechť $(\vec{x},0)$ je *MFP* funkce $f \Rightarrow (\vec{x},1)$ je *TP*. Zároveň $f(x_1,\ldots,x_{i-1},0,x_{i+1},\ldots,1) \leq f(\vec{x},0) = 0$, tedy $(\vec{x},1)$ je *MTP* funkce f.

Obdobně nechť $(\vec{x},1)$ je *MTP* funkce $f \Rightarrow (\vec{x},0)$ je *FP*. Zároveň $f(x_1,\ldots,x_{i-1},1,x_{i+1},\ldots,0) \leq f(\vec{x},1) = 1$, tedy $(\vec{x},0)$ je *MFP* funkce f.

Definice. Nechť $f(x_1,\ldots,x_n)$ je booleovská funkce. Definujme funkci f_i jako *restrikci* funkce f po zafixování $x_i = x_{i+1} = \ldots x_n = 1$. Dodefinujme $f_{n+1} = f$.

Věta.

Nechť f je pozitivní boolovská funkce a nechť $x \in \{0,1\}^{n-1}$. Potom $(\vec{x},1)$ je *MFP* právě tehdy, když $\vec{x}$ je *MFP* f_{n-1} (tj. f na prvních $n-1$ proměnných).

Důkaz. Zřejmý, z definice *MFP*.

Algoritmus. *(Dualizace regulárních funkcí)*

Vstup: formule f

1. Pro $i \in (n+1,\ldots,2)$:
2. vygeneruj *MFP* funkce f_i, pro které $x_{i-1} = 0$
3. zafixuj $x_{i-1} = 1$ a zjisti *MTP* funkce f_{i-1}

Výstup: formule f^d duální k f

Celkově proběhne n iterací. Složitost algoritmu je $\mathcal{O}(n^2m^2)$, nejnáročnější jsou absorbce.

Příklad.

Mějme funkci zadanou formulí

$$f(x_1,\ldots,x_5) = x_1x_2 \vee x_1x_3 \vee x_1x_4 \vee x_2x_3 \vee x_2x_4x_5$$

Lze ověřit, že funkce je regulární. Iterace algoritmu budou následující:

- $i = 6$

  jediný implikant, který obsahuje x_5 je $x_2x_4x_5$. Jeho charakteristika je 01011, tedy charakteristika *MFP* je 01010 (překlopíme $i-1$-tou složku), a té odpovídá $x_1x_3x_5$

- $i = 5$
  polož $x_5 = 1$ a absorbuj
  $f(x_1, x_2, x_3, x_4) = x_1x_2 \vee x_1x_3 \vee x_1x_4 \vee x_2x_3 \vee x_2x_4$

$$MTP\ x_1x_4 \to 10011 \to MFP\ 10001 \to x_2x_3x_4$$
$$MTP\ x_2x_4 \to 01011 \to MFP\ 01001 \to x_1x_3x_4$$

- $i = 4$
  polož $x_4 = 1$ a absorbuj
  $f(x_1, x_2, x_3) = x_1x_2$
- $i = 3$
  $f(x_1, x_2, x_3) = x_1x_2$
- $i = 2$
  $f(x_1, x_2) = x_1x_2$

$$MTP\ x_2 \to 01111 \to MFP\ 00111 \to x_1x_2$$

Tedy duální formule má tvar $(x_1 \vee x_2) \wedge (x_1 \vee x_3 \vee x_4) \wedge (x_2 \vee \vee x_3 \vee x_4) \wedge (x_1 \vee x_3 \vee x_5)$.

Jak je to se sílou proměnných?

$$f^d(1, 0, \ldots) = x_3x_4 \vee x_3x_5$$
$$f^d(0, 1, \ldots) = x_3x_4$$

Tedy $x_1 \succeq_{f^d} x_2$. Lze ověřit, že síla proměnných zůstane i v ostatních případech.

Věta. *(O regulárnosti duální funkce)*
Nechť f je regulární boolovská funkce. Pak její duální formule f^d je také regulární, navíc ke stejnému pořadí proměnných.
Důkaz.

BÚNO $i = 1$, $j = 2$. Pak $f^d(1, 0, x_3, \ldots, x_n) = \neg f(0, 1, \neg x_3, \ldots, \neg x_n) \geq \neg f(1, 0, \ldots, \neg x_n) = f^d(0, 1, \ldots,\ x_n)$

5. Prahové funkce

Definice. *(Prahová funkce)*
Boolovská funkce $f(x_1, \ldots, x_n)$ se nazývá *prahová* (lineárně separabilní), pokud $\exists v_1, \ldots, v_n, p \in R$ takové, že

$$f(x_1, \ldots, x_n) = 0 \iff \sum_{i=1}^{n} v_i x_i \leq p$$

Tedy nadrovnina definovaná pomocí $v_1, \ldots, v_n, p$ separuje 0 od 1.

Značení. Separátor a separující struktura

- *Separátorem funkce f* nazýváme hyperrovinu $\sum_{i=1}^{n} v_i x_i = p$
- *Separující struktura funkce f* je $(n+1)$-tice $(v_1, \ldots, v_n, p)$

Věta. Nechť $f(x_1, \ldots, x_n)$ je prahová booleovská funkce se strukturou $(v_1, \ldots, v_n, p)$. Mějme dány libovolné indexy $i, j : 1 \leq i < j \leq n$. Potom platí:

$$v_i \leq v_j \Rightarrow x_i \preceq_f x_j$$

Důkaz. Za jakých podmínek platí $x_i \preceq_f x_j$? Z definice $\preceq_f$ dostaneme:

$$f(x_1, \ldots, x_{i-1}, 0, \ldots, x_{j-1}, 1, \ldots, x_n) \geq f(x_1, \ldots, x_{i-1}, 1, \ldots, x_{j-1}, 0, \ldots, x_n)$$

- Levá strana je rovna 1 $\Leftrightarrow \sum_{k \neq i,j}^{n} v_k x_k + v_j > p$

· Pravá strana je rovna 1 $\Leftrightarrow \sum_{k\neq i,j}^{n} v_k x_k + v_i > p$

Pokud je tedy pravá strana rovna 1 a zároveň platí $v_i \leq v_j$, musí být také levá strana rovna 1, což přímo z definice implikuje $x_i \preceq_f x_j$.

Pozorování. Obrácená implikace neplatí (např. při $x_i \sim x_j$ se váhy nemusí rovnat).

Důsledek.

· $v_i = v_j \Rightarrow x_i \simeq_f x_j$

· každá pozitivní prahová funkce je regulární

Věta. Každá prahová funkce je unátní (tj. monotónní v každé proměnné). Přesněji řečeno: Pokud $f(x_1, \ldots, x_n)$ je prahová booleovská funkce se strukturou $(v_1, \ldots, v_n, p)$ pak platí:

1)
   · Pokud je $v_i > 0$, potom je f pozitivní v x_i.
   · Pokud je f pozitivní v x_i a závisí na x_i, tak je $v_i > 0$.
2)
   · Pokud je $v_i < 0$, tak je f negativní v x_i.
   · Pokud je f negativní v x_i a závisí na x_i, tak je $v_i < 0$.
3) Pokud $v_i = 0$, tak f nezávisí na x_i.

Důkaz. Dokážeme pouze 1. (případ 2. je obdobný a 3. plyne z definice regulárních funkcí):

· Dle definice je f pozitivní v x_i, pokud platí:

$$f(\ldots, 0, \ldots) \leq f(\ldots, 1, \ldots)$$

Levá strana je rovna 1 $\Leftrightarrow \sum_{j\neq i} v_j x_j > p$. Pravá strana je rovna 1 $\Leftrightarrow \sum_{j\neq i} v_j x_j + v_i > p$. Pokud je tedy $v_i > 0$ a levá strana je rovna 1, není možné, aby byla pravá strana menší než 1. Tvrzení tedy platí.

· Pokud je f pozitivní v x_i a závisí na x_i, musí existovat takové ohodnocení proměnných, že:

$$0 = f(\ldots, 0, \ldots) < f(\ldots, 1, \ldots) = 1$$

Levá strana nerovnosti je rovna 0 $\Leftrightarrow \sum_{j\neq i} v_j x_j \leq p$. Pravá strana je zase rovna 1 $\Leftrightarrow \sum_{j\neq i} v_j x_j + v_i > p$. Odtud nutně plyne, že $v_i > 0$.

Věta. Nechť $f(x_1, \ldots, x_n)$ je prahová booleovská funkce se strukturou $(v_1, \ldots, v_n, p)$, kde:

$$\exists k\ \forall j = 1, \ldots, k : v_j \geq 0 \wedge \forall j = k+1, \ldots, n : v_j < 0$$

Potom booleovská funkce g definovaná předpisem:

$$g(x_1, \ldots, x_n) = f(x_1, \ldots, x_k, \overline{x}_{k+1}, \ldots, \overline{x}_n)$$

je pozitivní prahová funkce se strukturou:

$$(v_1, \ldots, v_k, -v_{k+1}, \ldots, -v_n, p - \sum_{i=k+1}^{n} v_i)$$

Důkaz. Použijeme předpis definující g a definici prahové funkce:

$$\begin{aligned} g(x_1, \ldots, x_n) = 0 &\Leftrightarrow f(x_1, \ldots, x_k, \overline{x}_{k+1}, \ldots, \overline{x}_n) = 0 \\ &\Leftrightarrow \sum_{i=1}^{k} v_i x_i + \sum_{i=k+1}^{n} v_i (1 - x_i) \leq p \\ &\Leftrightarrow \sum_{i=1}^{k} v_i x_i - \sum_{i=k+1}^{n} v_i x_i + \sum_{i=k+1}^{n} v_i \leq p \\ &\Leftrightarrow \sum_{i=1}^{k} v_i x_i - \sum_{i=k+1}^{n} v_i x_i \leq p - \sum_{i=k+1}^{n} v_i \end{aligned}$$

Věta. Nechť $f(x_1, \ldots, x_n)$ je prahová booleovská funkce se strukturou $(v_1, \ldots, v_n, p)$, která je celočíselná (BÚNO - existuje-li nějaká struktura, existuje i racionální struktura, kterou stačí přenásobit společným jmenovatelem). Potom f^d je prahová booleovská funkce se strukturou $(v_1, \ldots, v_n, \sum_{i=1}^{n} v_i - p - 1)$ ∎

Důkaz.

$$\begin{aligned} f^d(x) = 0 &\Leftrightarrow \neg f(\overline{x}) = 0 \\ &\Leftrightarrow f(\overline{x}) = 1 \\ &\Leftrightarrow \sum_{i=1}^{n} v_i(1 - x_i) > p \\ &\Leftrightarrow \sum_{i=1}^{n} v_i x_i < \sum_{i=1}^{n} v_i - p \\ &\Leftrightarrow \sum_{i=1}^{n} v_i x_i \leq \sum_{i=1}^{n} v_i - p - 1 = p' \end{aligned}$$

Kde p' je nový práh pro doplňkovou prahovou funkci.

Důsledek. Duální páry jsou vždy porovnatelné:

- $p \leq p' \Rightarrow f^d \leq f$
- $p \geq p' \Rightarrow f^d \geq f$

6. Rozpoznávání prahových funkcí

V první části kapitoly se zaměříme na rozpozávání prahových funkcí. Ve zbytku se zaměříme na úvod do SATu a ukážeme algoritmus, který vyřeší 2-SAT v čase $\mathcal{O}(n)$.

Předpokládáme, že vstupní DNF je unární. Potom BÚNO platí, že vstupní DNF je pozitivní (jinak nahradím negované proměnné novými). Máme triviálně *MTP* a známe algoritmus, který z *MFP* vyrobí *MTP*.

Tvrzení.

Nechť $x^1, \ldots, x^q$ jsou *MFP* funkce f a $y^1, \ldots, y^r$ jsou *MTP* funkce f, pak f je prahová, právě když systém nerovností

$$\begin{aligned} \sum_{i=1}^{n} v_i x_i^j &\leq p \qquad 1 \leq j \leq q \\ \sum_{i=1}^{n} v_i y_i^j &\geq p + 1 \qquad 1 \leq j \leq r \\ v &\geq 0 \end{aligned}$$

má řešení $(v_1, \ldots, v_n, p)$.

Důkaz.

Důkaz vychází z toho, že každý *TP* (*FP*) je obsažen v nějakém *MTP* (*MFP*). $p+1$ můžu proto, že vždy je tam ε-ová nerovnost, po vynásobení vhodnou konstantou ji zvětším až na 1.

Algoritmus. *(Rozpoznávání prahových funkcí)*

Vstup: pozitivní funkce f

1. Otestuj regularitu f. Pokud f není regulární, není ani prahová.
2. Zkonstruuj duální f^d
3. Sestav soustavu rovnic dle tvrzení **??** a vyřeš.
4. Existuje-li řešení, je funkce prahová, jinak ne.

Výstup: rozhodnutí, zda je f prahová

Test regularity umíme v čase $\mathcal{O}(n^2m^2)$, lze v $\mathcal{O}(n^2m)$ užitím vhodných datových struktur. Stejně tak konstrukci duální funkce. Soustavu rovnic lze vyřešit v polynomiálním čase užitím lineárního programování.

Poznámka. Nalezení diskrétního, kombinatorického algoritmu, který by danou úlohu řešil, je otevřený problém. ∎

Úvod do SATu.

K čemu je užitečná úloha splnitelnosti CNF z hlediska umělé inteligence?
Mějme model sestávající se

d	dveře	otevřeno/zavřeno
s	siréna	houká/nehouká
a	majáček	bliká/nebliká
m	monitor	zobrazuje/nezobrazuje dveře d

Definujme v něm následující pravidla:

$d \to a$	pokud jsou dveře otevřené, bliká majáček
$d \to m$	pokud jsou dveře otevřené, zobrazují se na monitoru
$s \to a$	pokkud houká siréna, blíká majáček
$s \to m$	pokud houká siréna, zobrazují se dveře na monitoru
$a \wedge m \to d$	pokud bliká majáček a dveře se zobrazují na monitoru, jsou otevřené
$a \wedge m \to s$	pokud bliká majáček a dveře se zobrazují na monitoru, houká siréna

Pravidla lze přepsat na $(\neg d \vee a)$, $(\neg d \vee m))$, $(\neg s \vee a)$, $(\neg s \vee m)$, $(\neg a \vee \neg m \vee d)$, $(\neg a \vee \neg m \vee s)$. Všechna pravidla jsou splněna právě tehdy, když je odpovídající CNF splněna. Model expertního systému odpovídá splňujícímu ohodnocení CNF.

Odpovídání na dotazy:

- Otázka: Je pravda, že pokud bliká majáček, monitor zobrazuje dveře $(a \to m)$?
  Ekvivaletně: je pravda, že v každém modelu, kde $a = 1$ zároveň platí, že $m = 1$?
  Dosadíme do CNF pravidel $a = 1$, $m = 0$ a otestujeme SAT: $(\neg d) \wedge (\neg s)$ je splnitelná formule, tedy odpověď je NE.
- Otázka: Je pravda, že jsou dveře otevřené, houká siréna $(d \to s)$?
  Dosadíme do CNF pravidel $d = 1$, $s = 0$ a otestujeme SAT. $(a) \wedge (s) \wedge (\neg a \vee \neg m)$ je nesplnitelná formule, tedy odpověď je ANO.
- Otázka: Je $a \to m$ platné pravidlo?

$$\begin{aligned} \mathcal{F} = 1 \Rightarrow & (a \to m) = 1 \\ & (\neg a \vee m) = 1 \\ (\neg a \vee m) \Rightarrow & \mathcal{F} = 0 \end{aligned}$$

  TODO: zkontrolovat, zdá se mi to divné
  Dosadíme do CNF pravidel $a = 1$, $m = 0$ a otestujeme SAT. $d = s = 1$ je model, formule je tedy splnitelná a odpověď je NE.
- Otázka: Dveře jsou otevřené. Je potom pravda, že pokud bliká majáček, monitor zobrazuje dveře $(a \to m)$?
  ANO

2-SAT v řeči DNF

Nejprve ukážeme, že je korektní se omezit pouze na formule, které neobsahují termy délky 1. Ukážeme dvě varianty algoritmu, který vytvoří ekvivalentní formuli, která jednotkové termy neobsahuje.

Unit propagation.

Idea: termy délky 1 můžu přímo dosadit.

Algoritmus. *(Odstranění termů délky 1)*

Vstup: Formule $\mathcal{F}$ obsahující termy délky max. 2

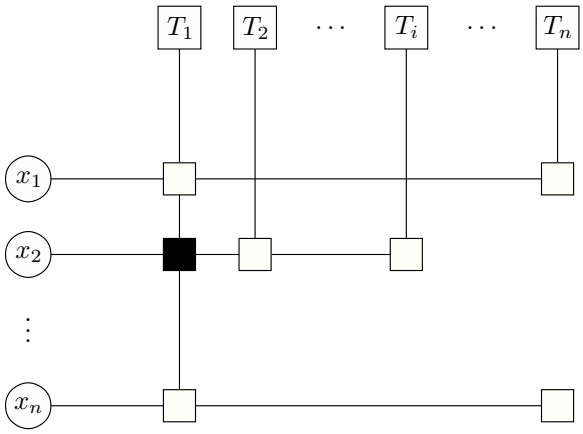
1. Dokud je v $\mathcal{F}$ jednotkový term u:

2. zafixuj $u = 0$, dosaď do $\mathcal{F}$

3. Pokud vznikl prázdný tem, $\mathcal{F}$ není falzifikovatelná, jinak mám $\mathcal{F}'$ bez jednotkových termů

Výstup: Formule $\mathcal{F}$ obsahující termy délky právě. 2

Algoritmus lze přímočaře implementovat v čase $\mathcal{O}(ln)$, kde l je počet literálů a n počet proměnných. Ukážeme, algoritmus, který propagaci provede v čase $\mathcal{O}(l)$.



Zkonstruuju strukturu, viz obr., kde $x_1, \ldots, x_n$ jsou proměnné, $T_1, \ldots, T_n$ termy . Poté procházím frontu 1-prvkových termů. Pro každou jeho proměnnou projdu odpovídající řádek: Pokud v termu i je negace, odstraním ji a zkontroluju, zda jsem nevytvořil seznam délky 1. ∎

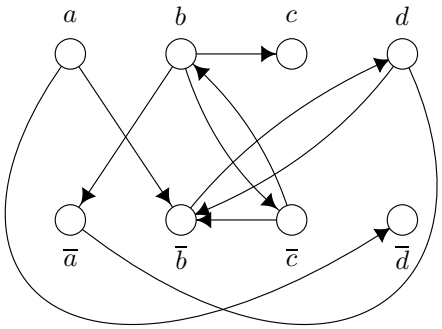
7. Falzifikovatelnost kvadratických funkcí

Nechť $\mathcal{F}$ na proměnných $x_1, \ldots, x_n$ je ryze kvadratická DNF-formule. Definujme orientovaný graf $G = (V, E)$:

$$V = \{x_1, \ldots, x_n, \overline{x_1}, \ldots, \overline{x_2}\},$$
$$E = \{(u, \overline{v}), (\overline{u}, v) | \forall \text{ term } uv\}$$

Příklad.

$$\mathcal{F} = ab \vee b\overline{c} \vee \overline{b}\overline{c} \vee db \vee ad$$



Věta.

Ohodnocení $o : V \to \{0, 1\}$ odpovídá falsifikujícímu ohodnocení $\mathcal{F}$ právě tehdy, když:

1) $\forall x \in V : o(x) \neq o(\bar{x})$
2) $\exists x, y \in V : o(x) = 1 \wedge o(y) = 0$ a v grafu existuje orientovaná cesta z x do y

Důkaz. První podmínka říká, že proměnná a její negace musí dostat každá jinou hodnotuZ konstrukce ohodnocení plyne. Druhé ohodnocení odpovídá implikaci, která není lplněna, ohodnocení tedy falzifikuje.

Poznámka. Pokud $a_1, \ldots, a_k$ tvoří silně souvislou kompomentu (SSK) grafu $G_{\mathcal{F}}$, tak také $\bar{a_1}, \ldots, \bar{a_k}$ tvoří SSK komponentu v $G_{\mathcal{F}}$ a naopak. ∎

Věta. $\mathcal{F}$ je falsifikovatelná právě když neexistuje žádný vrchol $x \in V$ takový, že x a $\bar{x}$ patří do stejné silně souvislé komponenty v grafu $G_{\mathcal{F}}$.
Důkaz.

- "$\Rightarrow$" sporem: nechť jsou x a $\bar{x}$ ve stejné SSK. Z toho vyplývá, že pro každé ohodnocení proměnných existuje cesta v grafu $G_{\mathcal{F}}$ z 1 do 0. Podle předchozí věty takové ohodnocení není falsifikující, $\mathcal{F}$ tedy není falsifikovatelná.

  TODO: Zkontrolovat. Zdá se mi, že algoritmus je psaný pro splnitelnost *CNF*, nikoliv pro falzifikovatelnost *DNF*. Sporem dokazuju, že v grafu nevznikne cesta z 0 do 1, což je ale přesně to, co pro falzifikovatelnost potřebuju.
- "$\Leftarrow$" konstruktivně: Zkonstruujeme ohodnocení o pro vrcholy splňující podmínky 1) a 2) z předchozí věty. Postupujeme následovně:
  i) Graf SSK je acyklický $\Rightarrow$ existuje topologicky poslední komponenta (žádné hrany ven). Označme ji jako C.
  ii) Všem literálům v C dáme hodnotu 1 ($\forall u \in C : o(u) = 1$), což zároveň vynutí 0 v doplňkové komponentě ($\forall \bar{u} \in \bar{C} : o(\bar{u}) = 0$), která je díky dualitě cest topologicky první (žádné hrany dovnitř).
  iii) Předchozí dva kroky iterujeme.

  TODO: Zkontrolovat i toto.

  Ověření funkčnosti postupu - sporem: předpokládejme, že nám pomocí výše uvedeného postupu vznikla cesta z 1 do 0.

  V okamžiku, kdy $C_2 \leftarrow 1$ ještě C_1 nebylo ohodnoceno. Ale díky dualitě musí existovat cesta z $\overline{C}_2$ do $\overline{C}_1$,

  což ale znamená, že $\overline{C}_2$ nebyla v okamžiku ohodnocení topologicky poslední.

Hornovské *DNF*.

Definice. *(Hornovská formule)*

$\mathcal{F}$ je *hornovská DNF* $\Leftrightarrow$ každý term z F obsahuje nejvýše 1 negativní literál. Pokud obsahuje právě 1 negativní literál jde o čistě hornovskou funkci.

Algoritmus. *(Falzifikující ohodnocení pro Horn-SAT)*
Vstup: Hornovská *DNF* $\mathcal{F}$

1. spustit na $\mathcal{F}$ algoritmus Unit propagation
2. pokud Unit propagation nalezne spor, je formule $\mathcal{F}$ falsifikovatelná - algoritmus pokračuje, jinak $\mathcal{F}$ není falsifikovatelná - algoritmus končí nezdarem
3. Falsifikující ohodnocení formule $\mathcal{F}$ je složeno z částečného ohodnocení získaného při Unit propagation v prvním kroku - zbytek jsou 0.

Výstup: falsifikující ohodnocení formule $\mathcal{F}$ (existuje-li)

Definice. *DNF* $\mathcal{F}$ je skrytě hornovská, pokud existuje podmnožina S proměnných z $\mathcal{F}$, taková, že po substituci $\forall x \in S : x \to \bar{x}, \bar{x} \to x$ vznikne z $\mathcal{F}$ hornovská *DNF* $\mathcal{F}'$.

Příklad.

$$\mathcal{F} = x_1\overline{x}_2\overline{x}_3 \vee x_1\overline{x}_4 \vee x_2x_3\overline{x}_4 \vee x_1x_3x_4 \vee x_1\overline{x}_2x_3 \vee \overline{x_1x}_2\overline{x_3}$$

Po otočení x_2, x_4 dostaneme hornovskou formuli.

8. Rozpoznávání hornovských formulí, miminalizace

Definice. *(Kvadartická DNF)*
K *DNF* $\mathcal{F}$ definujme *kvadratickou DNF* $\mathcal{F}_q$ následovně:

$$\{uv \in \mathcal{F}_q | \exists \text{ term } T \text{ v } \mathcal{F}, \text{ že } u, v \text{ jsou literály v } T\}$$

Lemma. $\mathcal{F}$ je hornovská $\iff$ $\mathcal{F}_q$ je falzifikována vektorem $(0, \ldots, 0)$
Důkaz.

- "$\Rightarrow$": $\mathcal{F}$ hornovská $\Rightarrow$ $\mathcal{F}_q$ hornovská $\Rightarrow$ $\mathcal{F}_q$ obsahuje max. 1 negativní literál v termu $\Rightarrow$ $\mathcal{F}_q$ obsahuje alespoň 1 pozitivní literál v termu (je ryze kvadratická) $\Rightarrow$ $(0, \ldots, 0)$ ji falzifikuje.
- "$\Leftarrow$": $(0, \ldots, 0)$ falzifikuje $\mathcal{F}_q \Rightarrow \nexists$ term v $\mathcal{F}_q$ obsahující dva negativní literály $\Rightarrow$ $\mathcal{F}$ je hornovská

Lemma. F je skrytě hornovská $\iff$ $\mathcal{F}_q$ je falzifikovatelná. Navíc pokud t je pravd. ohodnocení falzifikující $\mathcal{F}_q$, tak substituce

$$x \to \overline{x} \text{ pokud } t(x) = 0$$
$$x \to x \text{ pokud } t(x) = 1$$

transformuje $\mathcal{F}$ na její hornovskou verzi $\mathcal{F}'$.
Důkaz.

- "$\Rightarrow$": $\mathcal{F}$ skrytě hornovská $\Rightarrow \exists$ množina proměnných $S : \mathcal{F}'$ vzniklá z $\mathcal{F}$ komplementací proměnných v S je hornovská $\Rightarrow (0, \ldots, 0)$ falzifikuje $\mathcal{F}'_q \Rightarrow \mathcal{F}_q$ je falzifikována vektorem vzniklým z $(0, \ldots, 0)$ nahrazením nul jedničkami na pozicích odpovídajících proměnným v S.
- "$\Leftarrow$": $\mathcal{F}_q$ falzifikovatelná, má falzifikující ohodnocení $t \Rightarrow \mathcal{F}'_q$ vzniklá z $\mathcal{F}_q$ komplementací proměnnýých, pro které $t(x) = 1$ je falzifikována $(0, \ldots, 0) \Rightarrow \mathcal{F}'_q$ je hornovská $\Rightarrow$ $\mathcal{F}$ je skrytě hornovská.

Lemma nám dává přímočarý $\mathcal{O}(nl)$ algoritmus, protože $|\mathcal{F}_q| \in \mathcal{O}(nl)$, kde $|F| = l$, n je počet proměnných $\mathcal{F}$ a l počet jejich literálů.

Důsledek. Nechť $\mathcal{F}$ je kvadratická *DNF*. Pak $\mathcal{F}$ je falzifikovatelná $\Rightarrow$ $\mathcal{F}$ je skryté hornovská. Pro ryze kvadratické platí ekvivalence (pak dokonce $\mathcal{F} = \mathcal{F}_q$). Abychom dostali $\mathcal{O}(n)$ algoritmus, nahradíme *DNF* $\mathcal{F}$ za *DNF* $\mathcal{F}'$ takto:

$$\forall u_1, \ldots, u_k \text{ term v } \mathcal{F} \to u_1 y_1 \vee y_1 u_2 y_2 \vee \overline{y}_2 u_3 y_4 \vee \ldots \vee \overline{y}_{k-2} u_{k-1} y_{k-1} \vee \overline{y}_{k-1} u_k$$

tedy

$$\mathcal{F}_p = \mathcal{F}'_q = \bigvee_{c \in \mathcal{F}, |C| \geq 2} C_p$$

kde

$$C_p = u_1 y_1^C \vee \bigvee_{1 \leq i \leq k} (\overline{y}_{i-1}^C u_i \vee y_{i-1}^C y_i^C \vee u_i y_i^C) \vee \overline{y}_{k-1}^C u_k$$

Chceme ukázat: $\mathcal{F}_q$ je falzifikovatelná $\iff$ $\mathcal{F}_p$ je falzifikovatelná, přičemž $|C_p| = 3k - 2$.

Lemma. Nechť t je falzifikující ohodnocení $\mathcal{F}_p$. Pak $t_{|x_1, \ldots, x_n}$ je falzifikující ohodnocení $\mathcal{F}_q$.
Důkaz.

Nechť $u_i u_j$ je libovolný term v $\mathcal{F}_q \Rightarrow \exists$ term c v $\mathcal{F}$, že $C = u_1 \ldots u_i \ldots u_j \ldots u_k \Rightarrow u_i y_i$, $y_i \overline{y}_{i+1}$, $\overline{y}_{i+1} y_{i+2}$, $\overline{y}_{i+1} y_{i+2}$, …, $\overline{y}_{j-2} y_{j-1}$, $\overline{y}_{j-1} y_j \in \mathcal{F}_p$. V ohodnocení t jsou všechny tyto termy falizikovány $\Rightarrow$ alespoň jeden z u_i, u_j je falzifikován také.

Důsledek. f falzifikovatelná $\iff$ $\mathcal{F}_q$ falzifikovatelná.

Lemma. $\mathcal{F}_q$ falzifikovatelná $\Rightarrow$ $\mathcal{F}_p$ falzifikovatelná.
Důkaz.

Nechť t falzifikuje $\mathcal{F}_q$ a nechť $c = u_1 \dots u_k$ je term v $\mathcal{F}$. $c_q = \bigvee(u_i u_j)$, tj. $\mathcal{F}_q = 0 \Rightarrow C_q = 0 \Rightarrow$ nejvýše jeden z u_i, $1 \leq i \leq k$ má hodnotu 1. Nechť je to u_l. Dodefinujeme-li $t(y_i) = 1$, $1 \leq i \leq l \Rightarrow C_p = 0$ na dodefinovaném t.

Důsledek. *DNF* $\mathcal{F}$ je skrytě hornovská $\iff$ $\mathcal{F}_p$ je falzifikovatelná. Navíc pokud t je falzifikující ohodnocení $\mathcal{F}_p$, tak substituce z ?? transforumje $\mathcal{F}$ na hornovskou *DNF* $\mathcal{F}'$.

Algoritmus. *(Rozpoznávací algoritmus pro skrytě hornovské DNF)*

Vstup: formule $\mathcal{F}$

1. z $\mathcal{F}$ zkonstruuj $\mathcal{F}_p$ ': Pokud je $\mathcal{F}_p$ prázdná íF obsahuje jen lineární termy), pak $\mathcal{F}$ je hornovská.
2. Pomocí 2-SATu na $\mathcal{F}_p$ rozhodni falzifikovatelnost $\mathcal{F}_p$:
3. NE: $\mathcal{F}$ není skrytě hornovská
4. ANO: $\mathcal{F}$ je skrytě hornovská a falzifikující ohodnocení definuje transformaci $\mathcal{F}$ na hornovskou $\mathcal{F}'$

Výstup: rozhodnutí, zda $\mathcal{F}$ je skrytě hornovská

Minimalizace formy boolovských funckí.

DNF $\mathcal{F}$ a číslo $k \in M$

Otázka: $\exists DNF\mathcal{F}'$ taková, že $\mathcal{F}$ a $\mathcal{F}'$ reprezentují stejnou bool. funkci a $|\mathcal{F}'| \leq k$ (několik verzí podle definice $|\mathcal{F}'|$, např. počet literálů, počet termů, ...).

Obecně NP-těžké - pro $k = 1$ se tím dá ověřovat falzifikovatelnost. Ukážeme, že NP-těžké i pro hornovské *DNF*, polynomiální pro kvadratické *DNF*.

Lemma. Nechť je dána hornovská *DNF* $\mathcal{F}$ obsahující l literálů. Potom lze v čase $\mathcal{O}(l^2)$ nalézt primární a ireduntantní *DNF* $\mathcal{F}'$ reprezentující stejnou funkci jako $\mathcal{F}$.

9. Minimalizace kvadratických funkcí

V této přednášce dokážeme, jak z kvadratické DNF vyrobit ekvivalentní DNF minimální délky (ať už do počtu termůnebo literálů).

Lemma. Nechť $\mathcal{F}_1$, $\mathcal{F}_2$ jsou dvě primární hornovské DNF reprezentující tutéž funkci f a nechť $\mathcal{F}_1'$, $\mathcal{F}_2'$ jsou čistě hornovské části $\mathcal{F}_1$ resp. $\mathcal{F}_2$. Pak $\mathcal{F}_1'$ i $\mathcal{F}_2'$ reprezentují tutéž funkci f_H, nazýváme ji čistě hornovská komponenta f.

Důkaz. Uvažme, jak u hornovských formulí vypadá konsenzus.

$$Cons(\check{C}H, \check{C}H) = \check{C}H \text{ (ČH je čistě hornovská formule)}$$
$$Cons(\check{C}H, POS) = POS \text{ (POS je pozitivníformule)}$$
$$Cons(POS, POS) \dots \text{nelze}$$

Nechť $\mathcal{F}$ je kanonická DNF funkce f. Její kanonickou část lze vygenerovat z hornovských částí $\mathcal{F}_1$, $\mathcal{F}_2$. $\mathcal{F}_1'$, $\mathcal{F}_2'$ mají stejnou množinu primárních implikantů, tedy reprezentují stejnou funkci.

Proč je potřeba, aby $\mathcal{F}_1$ i $\mathcal{F}_2$ byli primární? Pokud $\mathcal{F}_1'$ primání, pak $\mathcal{F}_1' \subset \mathcal{F}$, totéž platí pro $\mathcal{F}_2'$, a tedy vše v $\mathcal{F}$ lze vygenerovat pomocí konsenzů jak z $\mathcal{F}_1'$, tak z $\mathcal{F}_2'$. Naopak, uvážíme li dvojici pozitivní formule, pozitivní formule + negativní literál absorbovaný pozitivními literály, zjistíme, že pro ni závěr lemmatu neplatí.

Lemma. Nechť $\mathcal{F}_1$, $\mathcal{F}_2$ jsou dvě primární ireduntantní hornovské DNF reprezentující tutéž funkci. Nechť $\mathcal{F}_1'$, $\mathcal{F}_2'$ jsou pozitivní části $\mathcal{F}_1$ a $\mathcal{F}_2$. Pak $\mathcal{F}_1$ a $\mathcal{F}_2$ mají stejný počet termů.

Důkaz. Nechť $G_F(V, E)$ je orientovaný graf pro DNF $\mathcal{F}$, kde

$$V = \{T |\ T \text{ je pozitivní primární implikant } \mathcal{F}\ \}$$
$$E = \{(T_1, T_2) |\ T_1 \text{ je implikantem } T_2 \cup \mathcal{F}_H\ \},$$

kde $\mathcal{F}_H$ je funkce reprezentující čistě hornovskou část $\mathcal{F}$. Tedy hrany grafu jsou tam, kde z $T_1, \mathcal{F}_H$ lze pomocí konsenzu odvodit T_2.

Vlastnosti $G_{\mathcal{F}}$:

- tranzitivně uzavřený – pokud $(T_1, T_2), (T_2, T_3)$ jsou hrany, pak T_3 je implikant $T_2 \cup \mathcal{F}$ a T_2 je implikant $T_1 \cup \mathcal{F}_H$, tj. T_3 je implikant $T_1 \cup \mathcal{F}_H \Rightarrow (T_1, T_3)$ je hrana.
- pokud $\mathcal{F}$ iredundantní a $(T_1, T_2) \in E$, nemůže nastat, aby $T_1, T_2 \in \mathcal{F}$, protože pak by z $T_1 \cup \mathcal{F}_H$ šla odvodit $\mathcal{F}_2$.

Tvrzení. Nechť T je pozitivní primární implikant hornovské funkce f a nechť $\mathcal{F}$ je primární reprezentace $\mathcal{F}$. Pak existuje pozitivní term $T' \in \mathcal{F}$ takový, že T je implikant $T' \cup \mathcal{F}_H$, kde $\mathcal{F}_H$ je čistě hornovská část $\mathcal{F}$.
Důkaz. Rozmyšlením, co jsou silně souvislé komponenty grafu.

Algoritmus. *(Minimalizace kvadratické DNF délky l)*
Vstup: kvadratická *DNF* $\mathcal{F}$

1. z $\mathcal{F}$ vyrob primární iredundantní *DNF* $\mathcal{F}_1$ (je také kvadratická)
2. rozděl $\mathcal{F}_1$ na $\mathcal{F}_1^L$ z lineárních termů a a $\mathcal{F}_1^Q$ z ryze kvadratických termů. $\mathcal{F}_1^L$ a $\mathcal{F}_1^Q$ jsou *DNF* na disjunktních množinách proměnných (pokud x je v $\mathcal{F}_1^L$, nemůže být v $\mathcal{F}_1^Q$ xy (spor s primárností) ani $\overline{x}y$ (lze provést konsenzus). Dále $\mathcal{F}_1^L$ je minimální.
3. Pokud $\mathcal{F}_1^Q$ není falzifikovatelná, pak minimální *DNF* ekvivalentní s $\mathcal{F}$ je $\mathcal{F}_1^L$ nebo 1, pokud je $\mathcal{F}_1^L$ prázdná.
4. $\mathcal{F}_1^Q$ je skrytě hornovská, lze ji tedy převést na její hornovskou verzi $\mathcal{F}_2$.
5. rozdělíme $\mathcal{F}_2$ na $\mathcal{F}_2^H$ (čistě hornovská část) a $\mathcal{F}_2^P$ (pozitivní část). Díky **??** je $\mathcal{F}_2^P$ minimální, neboť všechny primární implikanty jsou ryze kvadratické. Stačí tedy minimalizovat $\mathcal{F}_2^H$ (čistě hornovská ryze kvadratická formule bez primárních implikantů)
6. definujme $G_{F_2^H}$ takto:
$$V \dots \text{ proměmnné}$$
$$E = \{(x, y) |\ x, \overline{y} \text{ je term v } \mathcal{F}_2^H\ \}.$$
Konsenzus v $\mathcal{F}_2^H$ odpovídá tranzitivitě na $G_{F_2^H} \Rightarrow \mathcal{F}_2^H$ a $\tilde{F}_2^H$ jsou logicky ekvivalentní právě tehdy, když $G_{F_2^H}$ $G_{\tilde{F}_2^H}$ mají stejný tranzitivní uzávěr (jenž odpovídá kanonické *DNF*. Tím je problém převeden na hledání tranzitivní redukce orientovaného grafu.

Výstup: kvadratická *DNF* ekvivalentní s $\mathcal{F}$, jejíž délka je nejmenší možná

Algoritmus. *(Tranzitivní redukce orientovaného grafu.)*
Vstup: graf G

1. najdi všechny silně souvislé kompomenty grafu
2. každou nahrad jedním cyklem, což je minimální graf co do počtu hran potřebný k zajištění souvislosti. Obecně ještě musím řešit hrany mezi silně souvislými kompomentami, ale pro minimalizaci mi iredundantnost zajišťuje, že takové hrany v grafu neexistují.

Vstup: graf minimální co počtu hran, který má stejný tranzitivní uzávěr jako G

Poznámka. Najít minimální podgraf (resp. "*podDNF*") je *NP*-těžké.

10. NP-úplnost minimalizace hornovských funkcí

Cílem této kapitoly je ukázat, že minimalizace je *NP* těžké už pro čistě hornovské formule.

Značení. Každý term v hornovské klauzuli má tvar $a_1 \wedge a_2 \wedge \ldots \wedge a_k \wedge \overline{b} \equiv \neg[a_1 \wedge a_2 \wedge \ldots \wedge a_k \rightarrow b]$. Proměnným a_i říkáme *podcíle*, proměnnou b označujeme lako *hlavu* termu.

Algoritmus. *(Forward-chaining)*
Vstup: čistě hornovská *DNF* $\mathcal{F}$, podmnožina jejich proměnných M

1. $S \leftarrow M$
2. dokud v $\mathcal{F}$ existuje term, že *podcíle*$(T) \subseteq S$ a *hlava*$(T) \notin S$, tak $S \leftarrow S \cup \{$*hlava*$(T)\}$
3. vydej $S = FC_{\mathcal{F}}(M)$

Výstup: množina proměnných $FC_{\mathcal{F}}(M)$

Tvrzení. Čistě hornovský term $\bigvee_{x \in A} x \wedge \overline{y}$ je implikantem funkce reprezentující čistě hornovskou *DNF* $\mathcal{F}$ právě tehdy, když $y \in FC_{\mathcal{F}}(A)$.

Důkaz.

"⇒": Nechť $y \notin FC_{\mathcal{F}}(A)$. Pak ohodnocení $x = 1$, pokud $x \in FC_{\mathcal{F}}(A)$ a $x = 0$ jinak dává $T = 1$ a přitom toto ohodnocení falzifikuje $\mathcal{F}$, protože aby $\mathcal{F} = 1$, musel by v $\mathcal{F}$ být term T takový, že *podcíle*$(T) \subseteq FC(A)$ a $y \notin FC(A)$, což je spor, tedy T neníimplikant.

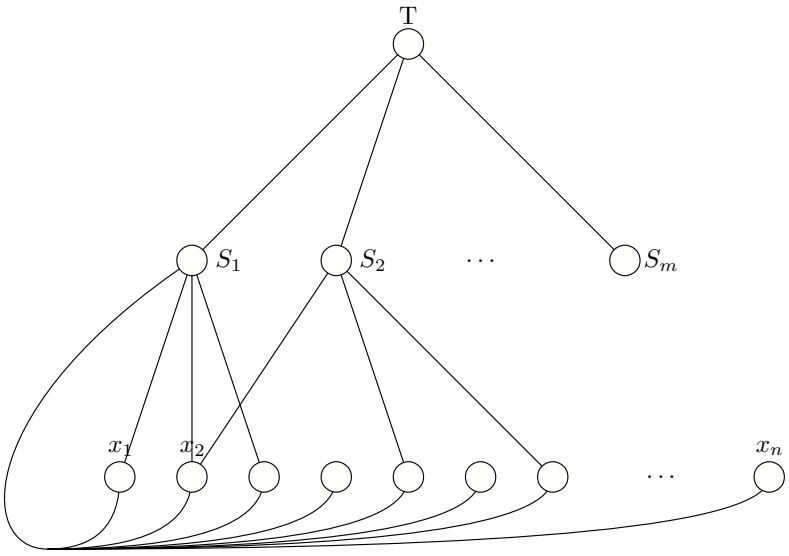
"⇐": Nechť $y \in FC(A)$. Lze ukázat indukcí, dle počtu přidání v době 2, že $\bigwedge_{x \in A} \cup \overline{y}$ je implikant $\mathcal{F}$.

Definice. *(SET-COLOR)*
Problém *SET-COLOR* patří mezi známé *NP*-úplné problémy. Nechť je dané $S = \{x_1, \ldots, x_n\}$, $S_1, \ldots, S_m \subseteq S$, takové, že $\bigcup_{i=1}^{m} S_i = S$ a $k \in N$. Otázka zní: existují indexy $i_1, \ldots, i_k$ takové, že $\bigcup_{i=1}^{k} S_{ij} = S$?

Algoritmus. *(Převod SET-COLOR na minimalizaci hornovské formule)*

Definujme proměnné: $x_1, \ldots, x_k, S_1, \ldots, S_m, T$. Formule bude obsahovat tyto termy:

a) $T\overline{S}_i$, $1 \leq i \leq m$
b) $S_i\overline{x_j}$, $1 \leq i \leq m, x_j \in S_i$
c) $x_1 \ldots x_n\overline{S}_i$, $1 \leq i \leq m$



- hrany $x_1 \ldots x_n\overline{S}_i$ jsou pro přehlednost vynechány, nakreslena je jen pro S_1
- pokud jsou dvě čistě hornovské formule ekvivalentní, mají stejnou množinu hlav
- pomocé forward-chainngu se bez použití T z S_i dostaneme jen jeho proměnných, do ostatních x_i ne
- rozmyšlením a koukáním na garf dokončíme převod

Poznámka. Miminalizace je těžká už pro kubické hornovské formule, ale důkaz je komplikovanější.

O. Obsah