

Vytváranie konjunktívnej normálnej formy

(Aplikácia logiky v inteligentných systémoch)

M. Mach

Katedra kybernetiky a umelej inteligencie, FEI, TUKE

september 2020

Tvorba CNF

- Ľubovoľná veta vo výrokovom počte môže byť transformovaná na CNF
 - výsledná CNF môže byť jednoduchšia ale aj zložitejšia ako pôvodná veta
 - v počte klauzúl
 - v počte symbolov
 - existuje viacero spôsobov transformácie
- Vygenerovaná CNF nemusí byť minimálna, môže byť redukovateľná
 - vypustením symbolu
 - vypustením klauzuly

Naivná transformácia

- Založená na postupnosti transformačných krokov
- Každý transformačný krok založený na jednom alebo viacerých transformačných pravidlách
- Transformačné kroky
 - 1 eliminácia zakázaných operátorov
 - 2 vnorenie negácií
 - 3 úprava na CNF
- Eliminačné kroky sú typicky dva
 - 1 eliminácia ekvivalencií
 - 2 eliminácia implikácií
- Výsledkom môže byť exponenciálne veľký nárast veľkosti výrazu

Pravidlá pre naivnú transformáciu

- Eliminácia ekvivalencií $P \leftrightarrow Q \equiv (P \rightarrow Q) \wedge (Q \rightarrow P)$
- Eliminácia implikácií $P \rightarrow Q \equiv \neg P \vee Q$
- Vnorenie negácie

$$\neg(P \wedge Q) \equiv \neg P \vee \neg Q$$

$$\neg(P \vee Q) \equiv \neg P \wedge \neg Q$$

$$\neg(\neg P) \equiv P$$

- Úpravy na CNF (využívajú sa vlastnosti operátorov pre distributívnosť, komutatívnosť a asociatívnosť)

$$P \wedge (Q \vee R) \equiv (P \wedge Q) \vee (P \wedge R)$$

$$P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$$

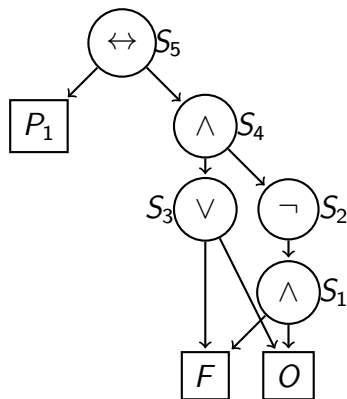
Tseitinova transformácia

- Založená na
 - rozklade logického výrazu na podvýrazy
 - pridávaní nových symbolov reprezentujúcich podvýrazy
- Pozostáva zo štyroch krokov
 - 1 parsovanie výrazu
 - 2 doplnenie pomocných symbolov
 - 3 vytvorenie ekvivalencií
 - 4 úprava klauzúl
- Garantuje menší nárast zložitosti ako exponenciálny (lineárny)

Tseitinova transformácia – kroky 1 až 3

$$P_1 \leftrightarrow (F \vee O) \wedge \neg(F \wedge O)$$

- Parsovací strom



- Pomocné symboly

S_1, S_2, S_3, S_4, S_5

- Vytvorené ekvivalencie

$$S_1 \leftrightarrow (F \wedge O)$$

$$S_2 \leftrightarrow \neg S_1$$

$$S_3 \leftrightarrow (F \vee O)$$

$$S_4 \leftrightarrow (S_3 \wedge S_2)$$

$$S_5 \leftrightarrow (P_1 \leftrightarrow S_4)$$

Tseitinova transformácia – prevod do CNF

- Prevod ekvivalencií do CNF

Typ ekvivalencie	CNF ($z \rightarrow$)	CNF ($z \leftarrow$)
$S \leftrightarrow \neg A$	$(\neg S \vee \neg A)$	$(A \vee S)$
$S \leftrightarrow (A \vee B)$	$(\neg S \vee A \vee B)$	$(\neg A \vee S) \wedge (\neg B \vee S)$
$S \leftrightarrow (A \wedge B)$	$(\neg S \vee A) \wedge (\neg S \vee B)$	$(\neg A \vee \neg B \vee S)$
$S \leftrightarrow (A \rightarrow B)$	$(\neg S \vee \neg A \vee B)$	$(A \vee S) \wedge (\neg B \vee S)$
$S \leftrightarrow (A \leftrightarrow B)$	$(\neg S \vee \neg A \vee B) \wedge$ $(\neg S \vee A \vee \neg B)$	$(A \vee B \vee S) \wedge$ $(\neg A \vee \neg B \vee S)$
$S \leftrightarrow (A \oplus B)$	$(\neg S \vee A \vee B) \wedge$ $(\neg S \vee \neg B \vee \neg A)$	$(\neg A \vee B \vee S) \wedge$ $(A \vee \neg B \vee S)$

Tseitinova transformácia – kroky 3 až 4

- Výraz na konci kroku 3 ...

$$\begin{aligned} S_5 \wedge & (S_5 \leftrightarrow (P_1 \leftrightarrow S_4)) \wedge (S_4 \leftrightarrow (S_3 \wedge S_2)) \wedge \\ & (S_3 \leftrightarrow (F \vee O)) \wedge (S_2 \leftrightarrow \neg S_1) \wedge \\ & (S_1 \leftrightarrow (F \wedge O)) \end{aligned}$$

- ... a po transformácii do tvaru CNF

$$\begin{aligned} S_5 \wedge & (\neg S_5 \vee \neg P_1 \vee S_4) \wedge (\neg S_5 \vee \neg S_4 \vee P_1) \wedge (P_1 \vee S_4 \vee S_5) \wedge \\ & (\neg P_1 \vee \neg S_4 \vee S_5) \\ \wedge & (\neg S_4 \vee S_3) \wedge (\neg S_4 \vee S_2) \wedge (\neg S_3 \vee \neg S_2 \vee S_4) \\ \wedge & (\neg S_3 \vee F \vee O) \wedge (\neg F \vee S_3) \wedge (\neg O \vee S_3) \\ \wedge & (\neg S_2 \vee \neg S_1) \wedge (S_1 \vee S_2) \\ \wedge & (\neg S_1 \vee F) \wedge (\neg S_1 \vee O) \wedge (\neg F \vee \neg O \vee S_1) \end{aligned}$$

Naivná vs Tseitinova transformácia

Naivná transformácia

- typicky jednoduchší výsledný výraz
- najhorší prípad – exponenciálna závislosť na počte klauzúl orig. výrazu

Tseitinova transformácia

- typicky zložitejší výsledný výraz
- najhorší prípad – lineárna závislosť na počte klauzúl orig. výrazu
- umožňuje logické obvody z prepojených logických hradieľ

Tseitinova transformácia a ekvivalencia

- Výsledok Tseitinovej transformácie nie je ekvivalentný s pôvodnou vetou
 - pred transformáciou tautológia $A \vee \neg A$
 - po transformácii už nie je tautológia
$$S \wedge (S \leftrightarrow (A \vee \neg A)) \equiv (\neg S \vee A \vee \neg A) \wedge (\neg A \vee S) \wedge (\neg \neg A \vee S)$$
- Transformácia zachováva splniteľnosť
- Tseitinova transformácia nedokáže vytvoriť validnú vetu
- Ekvisplniteľnosť – obe vety sú splniteľné alebo nesplniteľné, teda platí
 - prvá je splniteľná ak je splniteľná druhá
 - druhá je splniteľná ak je splniteľná prvá

Zjednodušovanie CNF

- Ak veta F v tvare CNF nie je minimálna, tak môže byť redukovaná na jednoduchšiu vetu F^r
- Redukcia môže byť
 - vypustením symbolu
 - vypustením klauzuly
- Ak F a F^r majú rovnakú množinu symbolov, tak musia byť *ekvivalentné*
- Ak množina symbolov F^r je podmnožina množiny symbolov F , tak musia byť *ekvisplniteľné*
 - jedna je splniteľná iba vtedy, ak je splniteľná aj druhá

Eliminácia premenných propagáciou

- Symbol je eliminovaný iba z nejakej klauzuly, v iných klauzulách ostáva
- Založené na rezolvencii tvaru

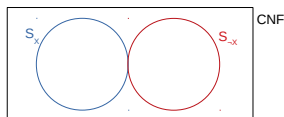
$$\frac{P \quad \neg P \vee R}{R}$$

pretože P musí byť TRUE (je klauzulou v CNF), tak $\neg P$ je nespĺnené a môže byť z disjunktie odstránené

- P je vo všeobecnosti disjunktia
 - Ak P je iba jeden literál, hovoríme o *jednotkovej* propagácii

Eliminácia premenných distribúciou klauzúl

- Symbol je eliminovaný z celej vety
- Založené na použití rezolvenčného odvodzovacieho pravidla (vypúšťajú sa komplementárne literály)
- Ak S_X je množina klauzúl obsahujúcich literál X a $S_{\neg X}$ množina klauzúl obsahujúcich literál $\neg X$



$$S_x \otimes S_{\neg x} = \{C_1 \otimes C_2 \mid C_1 \in S_x, C_2 \in S_{\neg x}, C_1 \otimes C_2 \neq \top\}$$

- Výsledná CNF bude: $(CNF \setminus (S_x \cup S_{\neg x})) \cup (S_x \otimes S_{\neg x})$
- Náhrada iba ak sa počet klauzúl nezvýši

Skrytá eliminácia tautológií

- Eliminácia nadbytočných klauzúl
- Založené na $P \wedge (P \vee Q) \equiv P$
 - klauzula P zahŕňa klauzulu $P \vee Q$
 - zahrnutá klauzula $P \vee Q$ je nadbytočná
- Algoritmus identifikácie zahrnutých klauzúl
 - predpokladá sa, že P je disjunkciou dvoch literálov
 - v kontrolovanej klauzule sa vyberie jeden literál (napr. A)
 - hľadá sa klauzula s dvomi literálmi, obsahujúca vybraný a nejaký iný literál (napr. $A \vee B$)
 - do kontrolovanej klauzuly sa vloží negácia druhého literálu ($\neg B$) a zisťuje sa, či vznikla tautológia
 - ak áno, tak kontrolovaná klauzula obsahuje nielen A ale aj B a je zahrnutá (tautológia bola kvôli B a $\neg B$)

Eliminácia klauzúl pridávaním premenných

- Je to inverzný postup k eliminácii premenných distribúciou klauzúl
- Vo vete, ktorá neobsahuje literál symbolu X , sa hľadá taká množina klauzúl S , pre ktorú je možné
 - vytvoriť množinu klauzúl S_x (v ktorých bude symbol X v tvare priameho literálu)
 - vytvoriť množinu klauzúl $S_{\neg x}$ (v ktorých bude symbol X v tvare negovaného literálu)

tak, aby platilo $S = S_x \otimes S_{\neg x}$

- Výsledná CNF bude: $(CNF \setminus (S_x \otimes S_{\neg x})) \cup (S_x \cup S_{\neg x})$
- Náhrada iba ak sa počet klauzúl zníži