Sprowadzanie formuły zdaniowej do koniunkcyjnej postaci normalnej – CNF

terminologia:

- literał - zmienna zdaniowa lub jej negacja
- klauzula -alternatywa literałów
- Mówimy, że klauzula c_1 absorbuje klauzulę c_2 wtedy i tylko wtedy, gdy c_1 jest podformułą c_2. Niech α będzie formułą w koniunkcyjnej postaci normalnej *CNF*. Przez $ABS(\alpha)$ oznaczać będziemy formułę, z której wszystkie absorbowane klauzule zostały usunięte.

Algorytm

wejście: A – formuła logiki zdań;
wyjście: formuła w CNF

1. {Eliminacja $\rightarrow$ i $\leftrightarrow$} Zastępuj w formule A

$$\begin{array}{lll} B \rightarrow C & \text{przez} & \neg B \vee C \\ B \leftrightarrow C & \text{przez} & (\neg B \vee C) \wedge (B \vee \neg C) \end{array}$$

   tak długo, aż otrzymana formuła zawiera wyłącznie spójniki $\neg, \wedge, \vee$.

2. {Przesuwanie $\neg$ "w głąb"} W formule otrzymanej w kroku 1. zastępuj

$$\begin{array}{rll} \neg(B \vee C) & \text{przez} & \neg B \wedge \neg C \\ \neg(B \wedge C) & \text{przez} & \neg B \vee \neg C \\ \neg\neg B & \text{przez} & B \end{array}$$

   tak długo, aż wszystkie wystąpienia negacji znajdą się bezpośrednio przed zmiennymi.

3. {Rozdzielenie $\wedge$ przez $\vee$} W formule otrzymanej w kroku 2. zastępuj tak długo jak to możliwe

$$\begin{array}{lll} (B \wedge C) \vee D & \text{przez} & (B \vee D) \wedge (C \vee D) \\ B \vee (C \wedge D) & \text{przez} & (B \vee C) \wedge (B \vee D) \end{array}$$

4. {Skracanie} Usuń wszystkie klauzule równoważne T (true).
5. {Absorpcja} Powtarzaj tak długo jak to możliwe:
   jeśli formuła otrzymana w kroku 4. zawiera parę klauzul c, c' takich, że c absorbuje c', to usuń c'.
6. Zwróć otrzymaną formułę jako CNF(A).∎

Dowodzenie

Terminologia cd.

- Dwie klauzule α i β mają *opozycję*, jeśli jedna z nich zawiera literał l, a druga $\neg l$.

- Przypuśćmy, że dwie klauzule α i β mają dokładnie jedną opozycję. *Rezolwentą* klauzul α i β, co oznaczamy $res(\alpha, \beta)$, jest klauzula otrzymana z dyzjunkcji $\alpha \vee \beta$ przez usunięcie literałów przeciwnych oraz zredukowanie wielokrotnych wystąpień literałów do jednego. Np. $res(\neg a \vee l, a \vee d) = l \vee d$.

Algorytm skracania

Niech α będzie formułą, a $CNF(\alpha)$ koniunkcyjną postacią normalną formuły α. *Koniunkcyjną postacią kanoniczną* α, $CCF(\alpha)$, jest formuła otrzymana z α w sposób następujący:

1. $\beta := CNF(\alpha)$.
2. Powtarzaj tak długo jak to możliwe:
   jeśli β zawiera parę klauzul δ i γ, dla których istnieje rezolwenta i żadna z klauzul β nie jest podformułą $res(\delta, \gamma)$, to $\beta := \beta \wedge res(\delta, \gamma)$.
3. $CCF(\alpha) := ABS(\beta)$.

Mówimy, że klauzula $c \not\equiv T$ jest *minimalną klauzulą implikowaną przez* formułę α, wtedy i tylko wtedy, gdy formuła $\alpha \to c$ jest tautologią i jeśli $\alpha \to c'$ jest tautologią, gdzie c' jest podformułą c, to $c' \equiv c$.

Twierdzenie 1
Niech α będzie formułą. $CCF(\alpha)$ jest koniunkcją wszystkich minimalnych klauzul implikowanych przez α. ∎

Twierdzenie 2
Literał l wynika z formuły A ($A \models l$) wtedy i tylko wtedy, gdy klauzula l występuje w CCF(A).∎

Twierdzenie 3
$A \equiv CNF(A) \equiv CCF(A)$.∎