LOGIKA MATEMATYCZNA

WYKŁAD 10: METODA REZOLUCJI W KRZ (20XII2007)

II. 10. Dowody rezolucyjne w KRZ

Pokażemy teraz działanie pewnej metody dowodowej, mającej istotne zastosowania m.in. w automatycznym dowodzeniu twierdzeń.

10.1. Przypomnienia i kilka definicji

Przypomnijmy, że:

Literałami nazywamy zmienne zdaniowe oraz negacje zmiennych zdaniowych.
Literałem *komplementarnym* do literału ℓ nazywamy literał $\overline{\ell}$, zdefiniowany następująco:

- jeśli ℓ jest literałem pozytywnym ℓ', to $\overline{\ell}$ jest literałem negatywnym $\neg\ell'$
- jeśli ℓ jest literałem negatywnym $\neg\ell'$, to $\overline{\ell}$ jest literałem pozytywnym ℓ'.

- *Koniunkcją elementarną* nazwiemy dowolną koniunkcję literałów.
- *Alternatywą elementarną* nazwiemy dowolną alternatywę literałów.
- *Alternatywną postacią normalną* (*apn*) nazwiemy dowolną alternatywę koniunkcji elementarnych.
- *Koniunkcyjną postacią normalną* (*kpn*) nazwiemy dowolną koniunkcję alternatyw elementarnych.
- Apn (odpowiednio: kpn) α nazywamy *istotną* i oznaczamy *iapn* (odpowiednio: *ikpn*), jeśli każda zmienna zdaniowa formuły α występuje w każdej elementarnej koniunkcji (odpowiednio: alternatywie) dokładnie raz, zaprzeczona bądź niezaprzeczona.
- Każdą apn (odpowiednio: kpn, iapn, ikpn) semantycznie równoważną danej formule α nazywamy *apn* (odpowiednio: *kpn, iapn, ikpn*) *formuły* α.

Pamiętamy, że każda formuła języka KRZ jest:

- semantycznie równoważna z pewną formułą w koniunkcyjnej postaci normalnej;
- semantycznie równoważna z pewną formułą w alternatywnej postaci normalnej;

- inferencyjnie równoważna z pewną formułą w koniunkcyjnej postaci normalnej;
- inferencyjnie równoważna z pewną formułą w alternatywnej postaci normalnej.

Pokazanie, że zachodzą równoważności, o których mowa powyżej, ma charakter ***algorytmiczny.***

Zachodzą implikacje:

- Jeśli α jest tautologią KRZ oraz $\alpha \sim \beta$ (α semantycznie równoważna z β), to także β jest tautologią KRZ.
- Jeśli α jest tezą KRZ oraz $\alpha \approx \beta$ (α inferencyjnie równoważna z β), to także β jest tezą KRZ.

Jeśli α jest kpn, to jest postaci: $\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n$, gdzie każda formuła α_i jest alternatywą elementarną postaci:

$$\ell_1^i \vee \ell_2^i \vee \ldots \vee \ell_{m_i}^i,$$

gdzie z kolei każda formuła ℓ_j^i jest literałem.

Koniunkcja $\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n$ jest tautologią KRZ wtedy i tylko wtedy, gdy wszystkie formuły α_i są tautologiami.

Formuła α_i (czyli formuła $\ell_1^i \vee \ell_2^i \vee \ldots \vee \ell_{m_i}^i$) jest tautologią KRZ wtedy i tylko wtedy, gdy wśród ℓ_1^i, ℓ_2^i, …, $\ell_{m_i}^i$ występuje co najmniej jedna para literałów komplementarnych.

Klauzulą nazwiemy dowolny skończony zbiór literałów.

Klauzule odpowiadają alternatywom elementarnym. Tak więc, jeśli $\ell_1 \vee \ell_2 \vee \ldots \vee \ell_n$ jest alternatywą elementarną, to odpowiadająca jej klauzula jest zbiorem $\{\ell_1, \ell_2, \ldots, \ell_n\}$. Umawiamy się, że literały, które (ewentualnie) występują więcej niż raz w danej alternatywie elementarnej zapisujemy tylko raz w odpowiadającej jej klauzuli. Ponieważ $(\alpha \vee \alpha) \equiv \alpha$ jest tezą KRZ, umowa ta niczego nie „psuje".

Zbiory klauzul są więc rodzinami zbiorów literałów. Każdej formule w kpn odpowiada pewien zbiór klauzul. Jeśli α jest kpn, to jest postaci: $\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n$, gdzie każda formuła α_i jest alternatywą elementarną postaci:

$$\ell_1^i \vee \ell_2^i \vee \ldots \vee \ell_{m_i}^i,$$

gdzie z kolei każda formuła ℓ_j^i jest literałem. Formule α odpowiada wtedy zbiór klauzul:

$$\{\{\ell_1^1, \ell_2^1, \ldots, \ell_{m_1}^1\}, \{\ell_1^2, \ell_2^2, \ldots, \ell_{m_2}^2\}, \ldots, \{\ell_1^n, \ell_2^n, \ldots, \ell_{m_n}^n\}\}.$$

Umawiamy się, że alternatywy elementarne, które (ewentualnie) występują więcej niż raz w danej koniunkcyjnej postaci normalnej zapisujemy tylko raz w odpowiadającej jej rodzinie zbiorów. Również ta umowa jest poprawna.

Dla przykładu, formule w koniunkcyjnej postaci normalnej:

$$(p_1 \vee p_2 \vee \neg p_3) \wedge (p_3 \vee p_4) \wedge \neg p_1 \wedge (\neg p_2 \vee \neg p_4)$$

odpowiada następujący zbiór klauzul:

$$\{\{p_1, p_2, \neg p_3\}, \{p_3, p_4\}, \{\neg p_1\}, \{\neg p_2, \neg p_4\}\}.$$

Mówienie „formule α w kpn odpowiada zbiór klauzul S" oraz „zbiór klauzul S reprezentuje formułę α w kpn" jest nieco rozwlekłe. Można wprowadzić jakiś symbol relacyjny, powiedzmy $\rightleftharpoons$, pozwalający na skrótowe zapisywanie takich wypowiedzi:

- $\alpha \rightleftharpoons S$ czytamy: „formule α w kpn odpowiada zbiór klauzul S" ***lub, równoznacznie***
- $\alpha \rightleftharpoons S$ czytamy: „zbiór klauzul S reprezentuje formułę α w kpn".

Symbol $\rightleftharpoons$ należy oczywiście do metajęzyka.

Klauzulę pustą (nie zawierającą żadnych elementów) oznaczamy przez $\Box$.

Klauzule zawierające najwyżej jeden literał pozytywny nazywamy ***klauzulami Hornowskimi***.

Są zatem trzy możliwości dla klauzuli Hornowskiej C:

- (a) C zawiera tylko literały negatywne;
- (b) C zawiera dokładnie jeden literał pozytywny i żadnych negatywnych;
- (c) C zawiera dokładnie jeden literał pozytywny oraz literały negatywne.

Będziemy posługiwać się ***stałymi zdaniowymi***:

- *falsum* $\bot$, reprezentującą dowolną kontrtautologię;
- *verum* $\top$, reprezentującą dowolną tautologię.

Zauważmy, że klauzula Hornowska:

- (a) postaci $\{\neg p_1, \neg p_2, \ldots, \neg p_n\}$ jest inferencyjnie równoważna formule: $(p_1 \wedge p_2 \wedge \ldots \wedge p_n) \rightarrow \bot$
- (b) postaci $\{p_1\}$ jest inferencyjnie równoważna formule: $\top \rightarrow p_1$
- (c) postaci $\{\neg p_1, \neg p_2, \ldots, \neg p_n, p_{n+1}\}$ jest inferencyjnie równoważna formule: $(p_1 \wedge p_2 \wedge \ldots \wedge p_n) \rightarrow p_{n+1}$.

Fakt powyższy można wykorzystać do podania ***szybkiego*** algorytmu dla sprawdzania, czy klauzula Hornowska odpowiada spełnialnej formule języka KRZ.

Przypomnijmy, że formuła α jest:

- *tautologią*, gdy ma wartość 1 przy *każdym* wzz;
- *kontrtautologią*, gdy ma wartość 0 przy *każdym* wzz;
- *spełnialna*, gdy ma wartość 1 przy *co najmniej jednym* wzz;
- *odrzucalna*, gdy ma wartość 0 przy *co najmniej jednym* wzz.

Tak więc, formuła α:

- jest tautologią, gdy nie jest odrzucalna;
- jest kontrtautologią, gdy nie jest spełnialna.

Ponadto, mamy oczywiście:

- α jest tautologią wtedy i tylko wtedy, gdy $\neg\alpha$ nie jest spełnialna.
- α jest kontrtautologią wtedy i tylko wtedy, gdy $\neg\alpha$ nie jest odrzucalna.

Pamiętamy, że algorytm ustalania, czy dana formuła języka KRZ jest tautologią ma złożoność wykładniczą: aby sprawdzić, czy formuła o n zmiennych zdaniowych jest tautologią KRZ trzeba sprawdzić, jaka jest jej wartość dla 2^n wzz.

Na mocy Twierdzenia o Pełności KRZ, jeśli formuła α ***nie jest*** spełnialna, to możemy to wykazać na drodze dedukcyjnej,

- pokazując, że: $\emptyset \vdash_{krz} \neg\alpha$ ***lub***
- pokazując, że: $\vdash_{jas} \neg\alpha$.

Nie możemy jednak, ani używając konsekwencji $\vdash_{krz}$, ani używając konsekwencji $\vdash_{jas}$ pokazać, że jakaś formuła ***jest*** spełnialna.

Podobnie, jeśli α wynika logicznie z X (czyli jeśli zachodzi $X \models_{KRZ} \alpha$), to możemy to wykazać,

- pokazując, że: $X \vdash_{krz} \alpha$ ***lub***
- pokazując, że: $X \vdash_{jas} \alpha$.

Jeśli jednak $X \not\models \alpha$, (czyli gdy przy ***co najmniej jednym*** wartościowaniu h, $h[X] \subseteq \{1\}$ oraz $h(\alpha) = 0$), to nie mamy możliwości przedstawienia dowodu (w terminach konsekwencji $\vdash_{krz}$ lub $\vdash_{jas}$), że ***istnieje*** wartościowanie h takie, że $h[X] \subseteq \{1\}$ oraz $h(\alpha) = 0$.

Reguła rezolucji, którą omówimy za chwilę, dostarcza możliwości wykazywania środkami czysto syntaktycznymi, że dana formuła nie jest spełnialna.

10.2. Reguła rezolucji

DEFINICJA 10.2.1.

Niech C_1 i C_2 będą klauzulami i niech literał ℓ występuje w C_1, a literał $\overline{\ell}$ występuje w C_2. Wtedy każdą klauzulę postaci:

$$(C_1 - \{\ell\}) \cup (C_2 - \{\overline{\ell}\})$$

nazywamy ***rezolwentą*** klauzul C_1 i C_2. Zamiast *rezolwenta* używa się też terminu: ***rezolwent***. Logice jest oczywiście obojętny rodzaj gramatyczny. Jeśli C_1 i C_2 są powyższej postaci, to mówimy też, że C_1 i C_2 ***kolidują*** ze względu na literały ℓ oraz $\overline{\ell}$.

PRZYKŁAD 10.2.1.

Niech:

- $C_1 = \{p_1, \neg p_2, p_3\}$
- $C_2 = \{p_2, \neg p_3, p_4\}$.

Widać, że C_1 i C_2 kolidują ze względu na następujące pary literałów komplementarnych:

- (a) $(\neg p_2, p_2)$,
- (b) $(p_3, \neg p_3)$.

Wtedy rezolwentami C_1 i C_2 są klauzule:

- (a) $\{p_1, p_3, \neg p_3, p_4\}$
- (b) $\{p_1, p_2, \neg p_2, p_4\}$.

DEFINICJA 10.2.2.

(i) ***Dowodem rezolucyjnym*** klauzuli C ze zbioru klauzul S nazywamy każdy skończony ciąg klauzul $C_1, \ldots, C_n$ taki, że:

- C jest identyczna z C_n
- każda klauzula C_i ($1 \leqslant i \leqslant n$) jest albo elementem zbioru S albo rezolwentą pewnych klauzul C_j oraz C_k dla $j, k < i$.

(ii) Jeśli istnieje dowód rezolucyjny C z S, to mówimy, że C jest ***rezolucyjnie dowodliwa*** (lub: ***rezolucyjnie wyprowadzalna***) z S i oznaczamy ten fakt przez $S \vdash_{res} C$.

(iii) Każdy dowód rezolucyjny klauzuli pustej $\square$ ze zbioru S nazywamy ***rezolucyjną refutacją*** S. Jeżeli istnieje rezolucyjna refutacja S, to mówimy, że S jest ***rezolucyjnie odrzucalny*** i oznaczamy ten fakt przez $S \vdash_{res} \square$.

(iv) Dla dowolnego zbioru klauzul S niech $res(S)$ będzie zbiorem wszystkich rezolwent wszystkich par elementów S. Zdefiniujmy:

- $res_0(S) = S$
- $res_n = res_{n-1}(S) \cup res(res_{n-1}(S))$ dla $n > 0$
- $\mathcal{R}(S) = \bigcup\{res_n(S) : n \in \mathcal{N}\}$.

Zbiór $\mathcal{R}(S)$ nazywamy ***domknięciem rezolucyjnym*** zbioru S.

(v) ***Rezolucyjnym drzewem dowodowym*** klauzuli C ze zbioru klauzul S nazywamy każde drzewo binarne T o następujących własnościach:

- korzeniem T jest C
- liśćmi T są pewne elementy zbioru S

- bezpośrednimi następnikami wierzchołka D niebędącego liściem są klauzule D_1 oraz D_2, których rezolwentą jest D.

Uwaga. Rozważamy drzewa, których wierzchołki są znakowane ***zbiorami*** literałów.

Uwaga. Dla dowolnego skończonego zbioru klauzul S istnieje liczba naturalna m taka, że $res_{m+1} = res_m$, czyli wszystkie wyrazy ciągu $res_n(S)$ są od pewnego miejsca identyczne. Jest to prosta konsekwencja faktu, że każdy skończony zbiór klauzul S zawiera tylko skończenie wiele literałów.

Uwaga. Nietrudno sprawdzić (korzystając z indukcji po długości dowodu rezolucyjnego), że zachodzi następująca równoważnośc:

- Istnieje rezolucyjne drzewo dowodowe dla C z S wtedy i tylko wtedy, gdy C jest rezolucyjnie dowodliwa z S, czyli gdy $S \vdash_{res} C$.

Uwaga. Często mówi się o dowodach rezolucyjnych *formuł* ze zbiorów *formuł*. Rozumiemy przez to, że wszystkie brane pod uwagę formuły:

- (1) zostały przekształcone do równoważnych im inferencyjnie kpn;
- (2) zostały zastąpione (przy uwzględnieniu (1)) odpowiadającymi im zbiorami klauzul.

Wtedy oczywiście należy powiedzieć, co rozumiemy przez dowód rezolucyjny ***zbioru klauzul*** ze ***zbiorów zbiorów*** klauzul. Jeśli piszemy skrótowo $S \vdash_{res} \alpha$, gdzie S jest zbiorem *formuł*, a α jest *formułą* to rozumiemy przez to, że:

- α została zastąpiona przez swoją kpn, a ta z kolei przez odpowiedni zbiór klauzul,
- każda formuła $\beta \in S$ została zastąpiona przez swoją kpn, a ta z kolei przez odpowiedni zbiór klauzul,
- $S \vdash_{res} \alpha$ oznacza, że ***każda*** klauzula występująca w zbiorze klauzul odpowiadającym kpn formuły α ma dowód rezolucyjny ze zbioru klauzul odpowiadającemu ***koniunkcji*** pewnych formuł z S.

Uwaga. Możemy rozważać ***dowolne*** zbiory klauzul jako poprzedniki relacji $\vdash_{res}$. Z Twierdzenia o Zwartości (zobacz Dodatek 4) oraz z Twierdzeń o Trafności i Pełności metody rezolucyjnej (które udowodnimy za chwilę) wynika, że jeśli $S \vdash_{res} \alpha$, to istnieje ***skończony*** zbiór $S' \subseteq S$ taki, że $S' \vdash_{res} \alpha$.

Koniec uwag.

PRZYKŁAD 10.2.2.

Niech $S = \{p_1 \to p_2, p_2 \to p_3, p_1, \neg p_3\}$ i niech $\bigwedge S$ będzie koniunkcją wszystkich formuł ze zbioru S.

Pokażemy, że $\bigwedge S \vdash_{res} \square$. Formuła $\bigwedge S$ ma następującą kpn:

$$(\neg p_1 \vee p_2) \wedge (\neg p_2 \vee p_3) \wedge p_1 \wedge \neg p_3.$$

Odpowiada jej zatem zbiór klauzul:

$$\{\{\neg p_1, p_2\}, \{\neg p_2, p_3\}, \{p_1\}, \{\neg p_3\}\}.$$

A oto zapowiadany dowód rezolucyjny:

1. $\{\neg p_1, p_2\}$ przesłanka
2. $\{\neg p_2, p_3\}$ przesłanka
3. $\{p_1\}$ przesłanka
4. $\{\neg p_3\}$ przesłanka
5. $\{\neg p_1, p_3\}$ rezolwenta (1) i (2)
6. $\{p_3\}$ rezolwenta (3) i (5)
7. $\square$ rezolwenta (4) i (6).

Zwykle takie dowody rezolucyjne zapisuje się w poniższej postaci:

1. $\neg p_1 \vee p_2$ przesłanka
2. $\neg p_2 \vee p_3$ przesłanka
3. p_1 przesłanka
4. $\neg p_3$ przesłanka
5. $\neg p_1 \vee p_3$ rezolwenta (1) i (2)
6. p_3 rezolwenta (3) i (5)
7. $\square$ rezolwenta (4) i (6).

Informatycy stosują inne jeszcze skróty notacyjne, czym nie będziemy się tutaj przejmować.

Zauważmy, że $\{p_1 \to p_2, p_2 \to p_3, p_1\} \models_{KRZ} p_3$, co oznacza, że zbiór

$$\{p_1 \to p_2, p_2 \to p_3, p_1, \neg p_3\}$$

nie jest spełnialny (nie istnieje wartościowanie, przy którym wszystkie elementy tego zbioru mają wartość 1). Pokażemy za chwilę, że zbiór klauzul S jest rezolucyjnie odrzucalny dokładnie wtedy, gdy nie jest spełnialna formuła, której kpn odpowiada (skończonemu podzbiorowi) S.

PRZYKŁAD 10.2.3.

Pokażemy, że zbiór formuł

$$S = \{p_1 \to (\neg p_2 \vee (p_3 \wedge p_4)), p_1, p_2, \neg p_4\}$$

jest rezolucyjnie odrzucalny. Tworzymy koniunkcję $\bigwedge S$ wszystkich formuł z S:

$$(p_1 \to (\neg p_2 \vee (p_3 \wedge p_4))) \wedge p_1 \wedge p_2 \wedge \neg p_4,$$

a po przekształceniu tej formuły do kpn tworzymy odpowiadający jej zbiór klauzul:

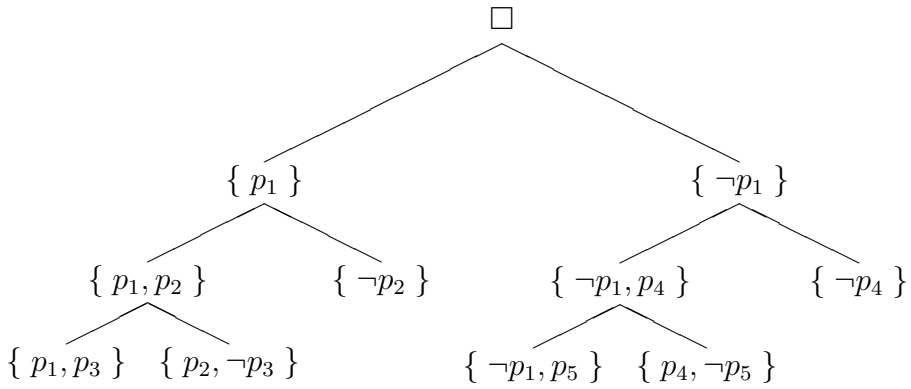
$$\{\{\neg p_1, \neg p_2, p_3\}, \{\neg p_1, \neg p_2, p_4\}, \{p_1\}, \{p_2\}, \{\neg p_4\}\}.$$

Dowód rezolucyjny zapiszemy korzystając z uproszczenia notacji zastosowanego w poprzednim przykładzie:

1. $\neg p_1 \vee \neg p_2 \vee p_3$ przesłanka
2. $\neg p_1 \vee \neg p_2 \vee p_4$ przesłanka
3. p_1 przesłanka
4. p_2 przesłanka
5. $\neg p_4$ przesłanka
6. $\neg p_1 \vee \neg p_2$ rezolwenta 2 i 5
7. $\neg p_1$ rezolwenta 4 i 6
8. $\square$ rezolwenta 3 i 7.

PRZYKŁAD 10.2.4.

Niech $S = \{\{p_1, p_3\}, \{p_2, \neg p_3\}, \{\neg p_2\}, \{\neg p_1, p_5\}, \{\neg p_4\}, \{p_4, \neg p_5\}\}$ będzie zbiorem klauzul. Poniższe drzewo jest rezolucyjnym drzewem dowodowym klauzuli $\square$ ze zbioru S:



Skoro $S \vdash_{res} \square$, to zbiór S jest rezolucyjnie odrzucalny. Zauważmy, że:

$$\{p_1 \to p_3, p_5 \to p_4, p_3 \to p_2, \neg p_2, \neg p_4\} \models_{KRZ} \neg(p_1 \wedge p_3).$$

PRZYKŁAD 10.2.5.

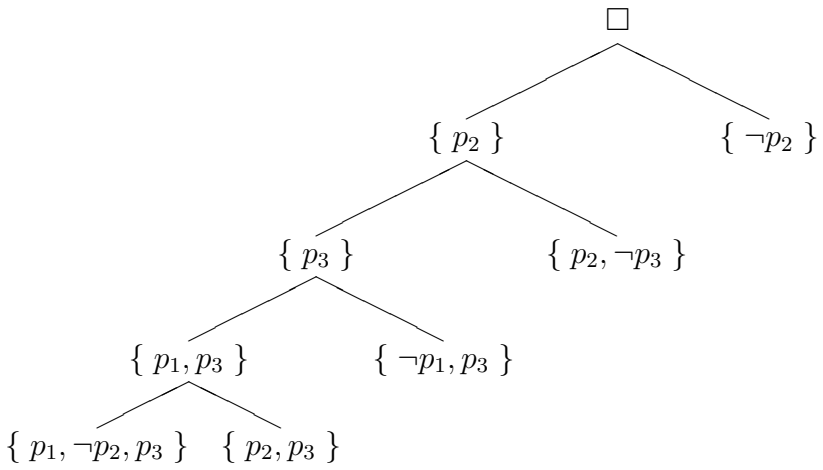
Pokażemy, że ze zbioru:

$$\{\{p_1, \neg p_2, p_3\}, \{p_2, p_3\}, \{\neg p_1, p_3\}, \{p_2, \neg p_3\}, \{\neg p_2\}\}$$

wyprowadzić można klauzulę pustą $\square$.

1. $\{p_1, \neg p_2, p_3\}$ przesłanka
2. $\{p_2, p_3\}$ przesłanka
3. $\{\neg p_1, p_3\}$ przesłanka
4. $\{p_2, \neg p_3\}$ przesłanka
5. $\{\neg p_2\}$ przesłanka
6. $\{p_1, p_3\}$ rezolwenta 1 i 2
7. $\{p_3\}$ rezolwenta 6 i 3
8. $\{p_2\}$ rezolwenta 7 i 4
9. $\square$ rezolwenta 8 i 5.

Powyższe wyprowadzenie reprezentowane jest przez następujące rezolucyjne drzewo dowodowe:



Powyższe przykłady pokazują, że stosowanie reguły rezolucji jest banalnie proste. Mogą więc skłaniać do (pochopnej!) konkluzji, że reguła rezolucji może zastąpić wszelkie skomplikowane techniki dowodowe (metodę aksjomatyczną, dedukcję naturalną, itd.). Rzecz ma się następująco. Owszem, reguła rezolucji nie jest skomplikowana i — jak pokażemy za chwilę — jest trafna i pełna. Jednak owa prostota ma też swoją cenę: zbiory klauzul odpowiadają formułom w koniunkcyjnych postaciach normalnych, i choć istnieje algorytm znajdowania dla każdej formuły równoważnej jej inferencyjnie formuły w kpn, to postępowanie wedle jego zaleceń jest dla Człowieka wielce czasochłonne. Inaczej rzecz się ma z maszynami liczącymi, które stosunkowo szybko znajdują kpn, a potem przeprowadzają dowody rezolucyjne.

Tak więc, nie ma ucieczki: choć bezmyślną pracę można powierzyć Maszynom, to praca twórcza (np. znajdowanie dowodów) stale należy do Człowieka.

10.3. Trafność metody rezolucji

Trzeba pokazać, że metoda rezolucji jest ***trafna***, tj. pokazać, że jeśli klauzula pusta należy do rezolucyjnego domknięcia zbioru S, to S nie jest spełnialny.

Można to uczynić na dwa sposoby:

- (A) czysto syntaktyczny (odwołujący się do relacji konsekwencji $\vdash_{krz}$ lub $\vdash_{jas}$), a potem skorzystać z Twierdzenia o Pełności KRZ;
- (B) semantyczny, tj. odwołując się bezpośrednio do relacji $\models_{KRZ}$ wynikania logicznego w KRZ.

Ponieważ czujemy, że Audytorium tego oczekuje, pokażemy oba sposoby.

Sposób (A)

Wykorzystamy relację konsekwencji założeniowej $\vdash_{jas}$.

W Dodatku 4 udowodniono, że relacja konsekwencji $\vdash_{jas}$ ma podobne własności, jak relacja $\vdash_{krz}$, tj. pokazano, że:

Wniosek 8.1.

Dla dowolnych zbiorów formuł X, Y, Z oraz dowolnej formuły α zachodzą następujące warunki:

- (1) $\vdash_{jas}$ jest zwrotna: $X \vdash_{jas} X$
- (2) $\vdash_{jas}$ jest przechodnia: jeśli $X \vdash_{jas} Y$ oraz $Y \vdash_{jas} Z$, to $X \vdash_{jas} Z$
- (3) $\vdash_{jas}$ jest monotoniczna względem pierwszego argumentu:
  jeśli $X \vdash_{jas} Y$ oraz $X \subseteq Z$, to $Z \vdash_{jas} Y$
- (4) $\vdash_{jas}$ jest antymonotoniczna względem drugiego argumentu:
  jeśli $X \vdash_{jas} Y$ oraz $Z \subseteq Y$, to $X \vdash_{jas} Z$.

Tu potrzebne będą nam warunki zwrotności, przechodniości oraz monotoniczności relacji $\vdash_{jas}$.

Teraz możemy pokazać, że tworzenie dowodów rezolucyjnych można reprezentować w systemie dedukcji naturalnej (w systemie założeniowym) KRZ:

TWIERDZENIE 10.3.1.

Jeśli R jest rezolwentą klauzul C_1 i C_2, oraz $\vdash_{jas} C_1$ i $\vdash_{jas} C_2$, to $\vdash_{jas} R$. W konsekwencji, $\{C_1, C_2\} \vdash_{jas} R$.

DOWÓD.

Uwaga. Zapis $\{C_1, C_2\} \vdash_{jas} R$ rozumiemy w ten sposób, że relacja $\vdash_{jas}$ zachodzi pomiędzy kpn reprezentowaną przez zbiór $\{C_1, C_2\}$, a alternatywą elementarną reprezentowaną przez klauzulę R.

Po tym wyjaśnieniu, możemy już przystąpić do dowodu. Jeśli R jest rezolwentą C_1 i C_2, to istnieje zmienna zdaniowa p taka, że p jest literałem w jednej z klauzul C_1 i C_2, a $\neg p$ literałem w pozostałej z klauzul C_1 i C_2. Niech np. p będzie literałem występującym w C_1, a $\neg p$ niech występuje w C_2. Zatem klauzula C_1 reprezentuje alternatywę elementarną $\alpha \vee p$, a klauzula C_2 reprezentuje alternatywę elementarną $\alpha \vee \neg p$, dla pewnych alternatyw elementarnych α i β. Ponieważ R jest rezolwentą C_1 i C_2 (względem pary kolidujących literałów p i $\neg p$), to R reprezentuje alternatywę elementarną $\alpha \vee \beta$. Pokażemy, że: jeśli $\vdash_{jas} p \vee \alpha$ i $\vdash_{jas} \neg p \vee \beta$, to $\vdash_{jas} \alpha \vee \beta$.

1.	$\vdash_{jas} p \vee \alpha$	założenie	
2.	$\vdash_{jas} \neg p \vee \beta$	założenie	
3.	$\{\neg p\} \vdash_{jas} p \vee \alpha$	1, monotoniczność $\vdash_{jas}$	
4.	$\{\neg p\} \vdash_{jas} \neg p$	zwrotność $\vdash_{jas}$	
5.	$\{\neg p\} \vdash_{jas} \alpha$	OA: 4,3	
6.	$\{\neg p\} \vdash_{jas} \alpha \vee \beta$	DA: 5	
7.	$\{\neg\neg p\} \vdash_{jas} \neg p \vee \beta$	2, monotoniczność $\vdash_{jas}$	
8.	$\{\neg\neg p\} \vdash_{jas} \neg\neg p$	zwrotność $\vdash_{jas}$	
9.	$\{\neg\neg p\} \vdash_{jas} \beta$	OA: 7,8	
10.	$\{\neg\neg p\} \vdash_{jas} \alpha \vee \beta$	DA: 9	
11.	$\vdash_{jas} \neg p \rightarrow (\alpha \vee \beta)$	6, twierdzenie o dedukcji	
12.	$\vdash_{jas} \neg\neg p \rightarrow (\alpha \vee \beta)$	10, twierdzenie o dedukcji	
13.	$\vdash_{jas} \alpha \vee \beta$	11,12, reguła wtórna: $\frac{\gamma\rightarrow\delta, \neg\gamma\rightarrow\delta}{\delta}$	Q.E.D.

TWIERDZENIE 10.3.2. (***Trafność rezolucji w KRZ***)

Niech S będzie zbiorem klauzul (reprezentującym pewną formułę w kpn). Jeśli $\square \in \mathcal{R}(S)$, to formuła reprezentowana przez S nie jest spełnialna.

DOWÓD.

Niech $\square \in \mathcal{R}(S)$. Wtedy $\square \in res_n(S)$ dla pewnego n (zob. uwagę po definicji 10.2.2.). Ponieważ $\square \notin res_0(S)$ (bo $\square$ nie jest klauzulą występującą w zbiorze S), więc istnieje liczba $m > 0$ taka, że:

- (a) $\square \notin res_m(S)$
- (b) $\square \in res_{m+1}(S)$.

Na mocy (b), $\square$ jest rezolwentą dwóch klauzul z $res_m(S)$. Jednak $\square$ może być rezolwentą jedynie pary literałów komplementarnych, czyli literałów postaci p i $\neg p$, gdzie p jest zmienną zdaniową. Nadto, zarówno p, jak i $\neg p$ są elementami $res_m(S)$. Na mocy:

- tego, że $m > 0$,
- definicji zbiorów $res_i(S)$,
- przechodniości relacji $\vdash_{jas}$,
- twierdzenia 10.3.1.,

zarówno p, jak i $\neg p$ są konsekwencjami (w sensie relacji $\vdash_{jas}$) formuły w kpn języka KRZ, której reprezentacją jest S. Stąd $p \wedge \neg p$ jest konsekwencją tej formuły. Ponieważ $p \wedge \neg p$ nie jest spełnialna, więc (na mocy Twierdzenia o Pełności KRZ) i owa formuła, reprezentowana przez S, nie jest spełnialna.

Q.E.D.

Pokazaliśmy zatem (korzystając z własności syntaktycznej relacji $\vdash_{jas}$ oraz z Twierdzenia o Pełności KRZ), że rezolucyjna odrzucalność zbioru klauzul reprezentującego formułę języka KRZ w koniunkcyjnej postaci normalnej implikuje niespełnialność tej formuły.

Sposób (B)

Twierdzenia o trafności metody rezolucyjnej możemy też dowieść „na drodze semantycznej", odwołując się bezpośrednio do wartościowań.

Zauważmy najpierw, że (na mocy stosownych definicji) zachodzi następująca równoważność:

- Klauzula C jest rezolucyjnie wyprowadzalna ze zbioru klauzul S wtedy i tylko wtedy, gdy $C \in \mathcal{R}(S)$.
- W szczególności, istnieje rezolucyjna refutacja S wtedy i tylko wtedy, gdy $\square \in \mathcal{R}(S)$.

Zamiast twierdzenia 10.3.1. posłużymy się teraz jego semantycznym odpowiednikiem:

TWIERDZENIE 10.3.3.

Jeśli $S = \{C_1, C_2\}$ jest spełnialny oraz C jest rezolwentą C_1 i C_2, to C jest spełnialna w KRZ. Co więcej, każde wartościowanie zmiennych zdaniowych, które spełnia S, spełnia też C.

DOWÓD.

Uwaga. Mówiąc o spełnialności zbioru klauzul S mamy na myśli, że jeśli $\alpha \rightleftharpoons S$, to formuła α jest spełnialna. Podobnie dla (spełnialności) pojedynczych klauzul.

Jeśli C jest rezolwentą C_1 i C_2, to istnieje literał ℓ oraz klauzule D_1, D_2, takie, że :

- $C_1 = D_1 \cup \{\ell\}$
- $C_2 = D_2 \cup \{\overline{\ell}\}$
- $C = D_1 \cup D_2$.

Załóżmy, że istnieje wartościowanie h takie, że $h(C_1) = 1$ oraz $h(C_2) = 1$. Z definicji wartościowania, wykluczone są przypadki:

- $h(\ell) = 1$ i $h(\overline{\ell}) = 1$
- $h(\ell) = 0$ i $h(\overline{\ell}) = 0$.

Pozostają zatem możliwości:

- (a) $h(\ell) = 1$ i $h(\overline{\ell}) = 0$
- (b) $h(\ell) = 0$ i $h(\overline{\ell}) = 1$.

Załóżmy, że zachodzi przypadek (a). Wtedy, ponieważ $h(C_2) = 1$ i $h(\overline{\ell}) = 0$, więc musi być $h(D_2) = 1$. Wtedy oczywiście także $h(C) = h(D_1 \cup D_2) = 1$.
Załóżmy, że zachodzi przypadek (b). Wtedy, ponieważ $h(C_1) = 1$ i $h(\ell) = 0$, więc musi być $h(D_1) = 1$. Wtedy oczywiście także $h(C) = h(D_1 \cup D_2) = 1$.

Q.E.D.

Uwaga. Zauważmy, że w istocie mamy silniejszą wersję twierdzenia 10.3.3.: ***każde*** wartościowanie, które spełnia (tj. przypisuje wartość 1) zbiór $S = \{C_1, C_2\}$, spełnia także ***każdą*** rezolwentę klauzul C_1 i C_2.

TWIERDZENIE 10.3.4. (***Trafność rezolucji w KRZ***).

Jeżeli istnieje rezolucyjna refutacja S, to S nie jest spełnialny w KRZ.

DOWÓD.

Niech $C_1, C_2, \ldots, C_n$ będzie rezolucyjną refutacją S. Wtedy oczywiście C_n jest identyczna z klauzulą $\Box$.

Z twierdzenia 10.3.3. wynika natychmiast, przez indukcję po długości rezolucyjnej refutacji zbioru S, że każde wartościowanie, które spełnia S, spełnia też każdą klauzulę C_i dla $1 \leqslant i \leqslant n$.

Ponieważ żadne wartościowanie nie spełnia klauzuli pustej $\Box$, więc nie istnieje wartościowanie spełniające S.

Q.E.D.

10.4. Pełność metody rezolucji

Trzeba pokazać, że metoda rezolucji jest ***pełna***, tj. udowodnić, że jeśli zbiór S jest niespełnialny, to można z niego rezolucyjnie wyprowadzić klauzulę pustą $\square$.

Podobnie jak w przypadku Twierdzenia o Trafności Rezolucji, można tego dokonać na dwa sposoby.

Sposób (A)

TWIERDZENIE 10.4.1. (***Pełność rezolucji w KRZ***).

Jeżeli S jest niespełnialny, to $\square \in \mathcal{R}(S)$.

DOWÓD.

Niech $S = \{C_1, \ldots, C_k\}$. Możemy oczywiście założyć, że żadna C_i nie jest tautologią. W przeciwnym przypadku możemy usunąć wszystkie tautologie z S i wyprowadzić $\square$ z klauzul pozostałych w S.

Przeprowadzimy dowód przez indukcję po n: liczbie zmiennych zdaniowych występujących w S.

POCZĄTKOWY KROK INDUKCJI. Niech $n = 1$. Niech p będzie jedyną zmienną zdaniową występującą w S. Są wtedy trzy możliwości:

- każda C_i jest postaci $\{p\}$
- każda C_i jest postaci $\{\neg p\}$
- każda C_i jest postaci $\{p, \neg p\}$.

Trzecią z tych możliwości wykluczyliśmy. Tak więc, jedynymi klauzulami w S są $\{p\}$ oraz $\{\neg p\}$. Są trzy możliwości:

- $S = \{\{p\}\}$
- $S = \{\{\neg p\}\}$
- $S = \{\{p\}, \{\neg p\}\}$.

W pierwszych dwóch S byłby spełnialny (a nie jest, z założenia). Tak więc, $S = \{\{p\}, \{\neg p\}\}$. Oczywiście $\square \in \mathcal{R}(S)$.

NASTĘPNIKOWY KROK INDUKCJI. Załóżmy teraz, że jedynymi zmiennymi zdaniowymi występującymi w S są $p_1, p_2, \ldots, p_n, p_{n+1}$. Załóżmy też, że $\square \in \mathcal{R}(T)$ dla każdego niespełnialnego zbioru T, w którym występują jedynie zmienne zdaniowe $p_1, p_2, \ldots, p_n$.

Zdefiniujmy następujące formuły:

- S^0 jest koniunkcją tych wszystkich C_i z S, które nie zawierają literału $\neg p_{n+1}$
- S^1 jest koniunkcją tych wszystkich C_i z S, które nie zawierają literału p_{n+1}.

S^0 i S^1 są formułami w kpn. Zauważmy, że $S = S^0 \cup S^1$, gdy S^0 i S^1 traktujemy jako ***zbiory*** klauzul. Gdyby było inaczej, to istniałaby klauzula C_i z S, która nie byłaby elementem $S^0 \cup S^1$. Wtedy C_i zawierałaby zarówno p_{n+1}, jaki $\neg p_{n+1}$, a więc (jako alternatywa elementarna) byłaby tautologią, co już wcześniej wykluczyliśmy. Dowodzi to inkluzji $S^0 \cup S^1 \subseteq S$. Ponieważ oczywiście $S \subseteq S^0 \cup S^1$, więc zachodzi równość $S = S^0 \cup S^1$.

Niech teraz:

- $S_0 = \{C_i - \{p_{n+1}\} : C_i \in S^0\}$
- $S_1 = \{C_i - \{\neg p_{n+1}\} : C_i \in S^1\}$.

Zauważmy, że:

- jeśli zastąpimy p_{n+1} w S przez $\bot$, to otrzymana formuła jest (semantycznie) równoważna z S_0
- jeśli zastąpimy p_{n+1} w S przez $\top$, to otrzymana formuła jest (semantycznie) równoważna z S_1.

Wynika stąd, że S jest (semantycznie) równoważny z $S_0 \vee S_1$. Ponieważ S jest niespełnialny, więc S_0 i S_1 są niespełnialne. W klauzulach z S_0 i z S_1 występują jedynie zmienne zdaniowe $p_1, p_2, \ldots, p_n$. Z założenia indukcyjnego, zachodzi zarówno $\Box \in \mathcal{R}(S_0)$, jak i $\Box \in \mathcal{R}(S_1)$.

S_0 utworzono z S^0 poprzez wyrzucenie literału p_{n+1} z każdej klauzuli w S^0. Ponieważ możemy wyprowadzić $\Box$ z S_0, więc z S^0 możemy wyprowadzić $\Box$ lub $\{p_{n+1}\}$.

Podobnie, S_1 utworzono z S^0 poprzez wyrzucenie literału $\neg p_{n+1}$ z każdej klauzuli w S^1. Ponieważ możemy wyprowadzić $\Box$ z S_1, więc z S^1 możemy wyprowadzić $\Box$ lub $\{\neg p_{n+1}\}$.

Jeśli możemy wyprowadzić $\{p_{n+1}\}$ z S_0, a $\{\neg p_{n+1}\}$ z S_1, to możemy wyprowadzić $\Box$ z $S^0 \cup S^1$. Ponieważ $S = S^0 \cup S^1$, więc ostatecznie $\Box \in \mathcal{R}(S)$.

Q.E.D.

TWIERDZENIE 10.4.2.

Niech α i β będą formułami języka KRZ i niech γ będzie koniunkcyjną postacią normalną formuły $\alpha \wedge \neg\beta$. Wtedy następujące warunki są równoważne:

- (1) $\alpha \models_{KRZ} \beta$
- (2) $\{\alpha\} \vdash_{krz} \beta$
- (3) $\{\alpha\} \vdash_{jas} \beta$
- (4) $\Box \in \mathcal{R}(\{\gamma\})$.

DOWÓD.

Równoważność (2) i (3) pokazano w Dodatku 4.

Implikacja (2)⇒(1) to Twierdzenie o Trafności w KRZ (udowodnione w Dodatku 3).

Implikacja (1)⇒(4) jest konsekwencją udowodnionego przed chwilą twierdzenia o pełności metody rezolucyjnej.

Pokażemy, że (4) implikuje (3).

Na mocy równości $\vdash_{krz} = \vdash_{jas}$ oraz twierdzenia 5.5. (zob. Dodatek 2), mamy: $\{\alpha \wedge \neg\beta\} \vdash_{jas} \gamma$. Reguła DK dołączania koniunkcji daje: $\{\alpha, \neg\beta\} \vdash_{jas} \alpha \wedge \neg\beta$. Z przechodniości $\vdash_{jas}$ (wniosek 8.1.(2)) mamy: $\{\alpha, \neg\beta\} \vdash_{jas} \gamma$.

Ponieważ $\Box \in \mathcal{R}(\{\gamma\})$, więc dla pewnej zmiennej zdaniowej p mamy: $p, \neg p \in \mathcal{R}(\{\gamma\})$. Stąd, na mocy twierdzenia 10.3.1.:

- $\{\gamma\} \vdash_{jas} p$ ***oraz***
- $\{\gamma\} \vdash_{jas} \neg p$.

Z przechodniości relacji $\vdash_{jas}$ otrzymujemy zatem:

- $\{\alpha, \neg\beta\} \vdash_{jas} p$ ***oraz***
- $\{\alpha, \neg\beta\} \vdash_{jas} \neg p$.

To oznacza, że:

- $\{\alpha\} \vdash_{jas} \neg\beta \rightarrow p$ ***oraz***
- $\{\alpha\} \vdash_{jas} \neg\beta\neg p$.

Na mocy tezy $(\alpha \rightarrow \beta) \rightarrow ((\alpha \rightarrow \neg\beta) \rightarrow \neg\alpha)$ otrzymujemy stąd: $\{\alpha\} \vdash_{jas} \neg\neg\beta$. Na mocy prawa opuszczania negacji mamy ostatecznie $\{\alpha\} \vdash_{jas} \beta$.

Q.E.D.

Sposób (B)

TWIERDZENIE 10.4.3.

Dla dowolnego zbioru klauzul T oraz dowolnego literału ℓ: jeśli T jest niespełnialny, to niespełnialny jest także zbiór $T(\ell) = \{C \in \mathcal{R}(T) \ : \ \ell, \overline{\ell} \notin C\}$.

DOWÓD.

Uwaga. Zapis $\ell \notin C$ oznacza, że literał ℓ nie występuje w klauzuli C.

Niech T będzie niespełnialny. W terminologii używanej w wykładach 3–4 powiedzielibyśmy, że T jest semantycznie sprzeczny.

Przypuśćmy, dla dowodu nie wprost, że h jest wartościowaniem takim, że h spełnia (tj. przyjmuje wartość 1) wszystkie elementy zbioru $T(\ell)$. Określmy wartościowania h_1 oraz h_2 tak, aby:

- $h_1(\ell) = 1$ oraz $h_2(\overline{\ell}) = 1$
- h_1 i h_2 przyjmowały takie same wartości jak h dla wszystkich pozostałych (tj. różnych od ℓ i $\overline{\ell}$) literałach występujących w T.

Ponieważ T jest niespełnialny, więc istnieją klauzule C_1 oraz C_2 w T takie, że:

- $h_1(C_1) = 0$

- $h_2(C_2) = 0$.

Ponieważ $h_1(\ell) = 1$, a $h_1(C_1) = 0$, więc ℓ nie występuje w C_1. Gdyby także $\overline{\ell}$ nie występował w C_1, to, na mocy definicji, mielibyśmy $C_1 \in T(\ell)$. Ponieważ przeczyłoby to założeniu, że h spełnia wszystkie elementy zbioru $T(\ell)$, więc $\overline{\ell}$ występuje w C_1. Analogicznie rozumując, pokazujemy, że ℓ występuje w C_2.

Pokazaliśmy, że klauzule C_1 i C_2 zawierają parę literałów komplementarnych. Możemy zatem otrzymać ich rezolwentę $D \in \mathcal{R}(T)$, która nie zawiera literału ℓ (oraz, oczywiście, nie zawiera też literału $\overline{\ell}$).

Z definicji wartościowania h mamy $h(D) = 1$. Z kolei, z definicji wartościowań h_1 i h_2 oraz z faktu, że $h_1(\ell) = 1$ i $h_2(\overline{\ell}) = 1$ wynika, że niemożliwe jest, aby $h(C_1) = 0$ ***oraz*** $h(C_2) = 0$. Zachodzi zatem alternatywa:

- $h(C_1) = 1$ ***lub***
- $h(C_2) = 1$

Otrzymujemy sprzeczność z założeniem, że T jest niespełnialny.

Q.E.D.

TWIERDZENIE 10.4.4. (***Pełność rezolucji w KRZ***).

Jeżeli S jest niespełnialny w KRZ, to istnieje rezolucyjna refutacja S.

DOWÓD.

Na mocy Twierdzenia o Zwartości (zobacz Dodatek 4, twierdzenie 8.5.), jeśli S jest niespełnialny, to istnieje skończony zbiór $S' \subseteq S$, który jest niespełnialny. Ponieważ każda rezolucyjna refutacja z S' jest też refutacją z S, możemy założyć, że S jest skończony. Skoro istnieje tylko skończenie wiele klauzul w S i każda klauzula jest zbiorem skończonym, więc istnieje tylko skończona liczba literałów, występujących w klauzulach z S. Niech będą to literały $\ell_1, \ell_2, \ldots, \ell_n$.

Niech S będzie niespełnialny. Pokażemy, że istnieje rezolucyjna refutacja S.

Niech $S_n = S(\ell) = \{C \in \mathcal{R}(S) : \ell, \overline{\ell} \notin C\}$. Z definicji, S_n jest zbiorem tych wszystkich konsekwencji rezolucyjnych S, które nie zawierają literałów ℓ_n oraz $\overline{\ell_n}$. Ponieważ S jest niespełnialny, więc na mocy twierdzenia 10.4.3., S_n również jest niespełnialny.

Niech z kolei $S_{n-1} = S(\ell_{n-1})$. Wtedy S_{n-1} jest zbiorem tych wszystkich konsekwencji rezolucyjnych S_n (a więc także S), które nie zawierają literałów ℓ_n, ℓ_{n-1}, $\overline{\ell_n}$ i $\overline{\ell_{n-1}}$.

Powtarzamy tę procedurę aż do otrzymania zbioru S_0, który jest niespełnialny i nie zawiera żadnych literałów. Jedynym takim zbiorem jest $\{\Box\}$. Pokazaliśmy zatem, że $\Box$ jest rezolucyjną konsekwencją S.

Q.E.D.

10.5. Dalsze przykłady

Skoro metoda rezolucji jest trafna i pełna, to można jej używać np. dla ustalania, czy:

- formuła języka KRZ jest tautologią KRZ

- formuła języka KRZ jest spełnialna
- formuła języka KRZ nie jest spełnialna
- formuła α wynika logicznie ze zbioru formuł X
- zbiór formuł X jest spełnialny
- zbiór formuł X nie jest spełnialny, itd.

PRZYKŁAD 10.5.1.

Rozważmy zbiór klauzul:

$$S = \{\{p_1, p_2, \neg p_3\}, \{p_3\}, \{p_1, \neg p_2, p_3\}, \{\neg p_3\}\}.$$

Zauważmy, że w zależności od kolejności doboru klauzul, do których stosujemy regułę rezolucji, możemy otrzymać różne wyniki końcowe:

(a)

1.	$\{p_1, p_2, \neg p_3\}$	przesłanka
2.	$\{p_3\}$	przesłanka
3.	$\{p_1, \neg p_2, p_3\}$	przesłanka
4.	$\{\neg p_3\}$	przesłanka
5.	$\square$	rezolwenta 2 i 4.

(b)

1.	$\{p_1, p_2, \neg p_3\}$	przesłanka
2.	$\{p_3\}$	przesłanka
3.	$\{p_1, \neg p_2, p_3\}$	przesłanka
4.	$\{\neg p_3\}$	przesłanka
5.	$\{p_1, p_2\}$	rezolwenta 1 i 2.
6.	$\{p_1, \neg p_2\}$	rezolwenta 3 i 4
7.	$\{p_1\}$	rezolwenta 5 i 6.

Tak więc, zbiór S ***nie jest*** spełnialny, ponieważ istnieje ***co najmniej jedno*** wyprowadzenie $\square$ ze zbioru S.

PRZYKŁAD 10.5.2.

Pokażemy, że

$$((\alpha \to \beta) \wedge (\beta \to \gamma) \wedge (\gamma \to \alpha) \wedge (\alpha \vee \beta \vee \gamma)) \to (\alpha \wedge \beta \wedge \gamma)$$

jest tautologią KRZ.

Jest tak dokładnie wtedy, gdy zbiór

$$\{\alpha \to \beta, \beta \to \gamma, \gamma \to \alpha, \alpha \vee \beta \vee \gamma, \neg(\alpha \wedge \beta \wedge \gamma)\}$$

jest semantycznie sprzeczny (nie jest spełnialny). To z kolei jest równoważne temu, że zbiór

$$\{\neg\alpha \vee \beta, \neg\beta \vee \gamma, \neg\gamma \vee \alpha, \alpha \vee \beta \vee \gamma, \neg\alpha \vee \neg\beta \vee \neg\gamma\}$$

nie jest spełnialny. Każda z formuł tego zbioru jest podstawieniem jakiejś alternatywy elementarnej: otrzymujemy je, gdy dokonamy np. podstawień $p_1/\alpha,\ p_2/\beta,\ p_3/\gamma$. W takich przypadkach usprawiedliwione jest pisanie dowodów rezolucyjnych z użyciem *metazmiennych* reprezentujących dowolne formuły języka KRZ i traktowanie pojedynczych metazmiennych jak literałów.

Na mocy pełności metody rezolucji wystarczy pokazać, że ze zbioru

$$\{\neg\alpha \vee \beta, \neg\beta \vee \gamma, \neg\gamma \vee \alpha, \alpha \vee \beta \vee \gamma, \neg\alpha \vee \neg\beta \vee \neg\gamma\}$$

można wyprowadzić klauzulę $\square$:

1.	$\neg\alpha \vee \beta$	przesłanka
2.	$\neg\beta \vee \gamma$	przesłanka
3.	$\neg\gamma \vee \alpha$	przesłanka
4.	$\alpha \vee \beta \vee \gamma$	przesłanka
5.	$\neg\alpha \vee \neg\beta \vee \neg\gamma$	przesłanka
6.	$\alpha \vee \beta$	rezolwenta 4 i 3
7.	β	rezolwenta 6 i 1
8.	γ	rezolwenta 7 i 2
9.	α	rezolwenta 8 i 3
10.	$\neg\beta \vee \neg\gamma$	rezolwenta 9 i 5
11.	$\neg\gamma$	rezolwenta 7 i 10
12.	$\square$	rezolwenta 8 i 11.

Stosujemy tu jeszcze jedno świństewko notacyjne, pisząc poszczególne alternatywy elementarne, a nie odpowiadające im klauzule. Inne z tego typu świństewek to milczące korzystanie z praw pochłaniania dla alternatywy: w wierszu 6 piszemy $\alpha \vee \beta$ zamiast $\alpha \vee \beta\alpha \vee \alpha$, a w wierszu 7 piszemy β zamiast $\beta \vee \beta$.

Takie (i dalsze jeszcze świństewka notacyjne) często spotykamy w niektórych podręcznikach. Czujemy się więc trochę usprawiedliwieni, także z nich korzystając. Jak pisał St.I. Witkiewicz:

> *Ciężko jest żyć w plugawej naszej atmosferze,*
> *Czasami, ach, wprost nawet kogoś z boku litość bierze —*
> *Pociecha w tym, że gorzej być plugawcem, ach, samemu,*
> *Bo nic już nie pomoże, ach, takiemu.*

PRZYKŁAD 10.5.3.

Pokażemy, że formuła:

$$(\bigstar) \quad \neg((\alpha \to \beta) \to ((\alpha \vee \gamma) \to (\beta \vee \gamma)))$$

nie jest spełnialna. Oznacza to, że formuła:

$$(\bigstar\bigstar) \quad (\alpha \to \beta) \to ((\alpha \vee \gamma) \to (\beta \vee \gamma))$$

jest tautologią KRZ.

W tym celu wystarczy pokazać, że ze zbioru klauzul otrzymanego z kpn formuły $(\bigstar)$ można wyprowadzić $\square$. Koniunkcyjną postacią normalną formuły $(\bigstar)$ jest:

$$(\neg\alpha \vee \beta) \wedge (\alpha \vee \gamma) \wedge (\neg\beta) \wedge (\neg\gamma).$$

Przeprowadzamy dowód rezolucyjny:

1. $\neg\alpha \vee \beta$ przesłanka
2. $\alpha \vee \gamma$ przesłanka
3. $\neg\beta$ przesłanka
4. $\neg\gamma$ przesłanka
5. α rezolwenta 2 i 4
6. β rezolwenta 1 i 5
9. $\square$ rezolwenta 3 i 6.

PRZYKŁAD 10.5.4.

Pokażemy, że formuła γ wynika logicznie ze zbioru formuł:

$$S = \{\alpha, (\alpha \wedge \beta) \rightarrow \gamma, \tau \rightarrow \beta, \tau\}.$$

W tym celu wystarczy pokazać, że zbiór

$$\{\alpha, (\alpha \wedge \beta) \rightarrow \gamma, \tau \rightarrow \beta, \tau, \neg\gamma\}$$

nie jest spełnialny.

Każda formuła ze zbioru S jest równoważna alternatywie elementarnej:

1. α
2. $\neg\alpha \vee \neg\beta \vee \gamma$
3. $\neg\tau \vee \beta$
4. τ.

Pokazujemy, że z powyższych klauzul można wyprowadzić $\square$:

1. α przesłanka
2. $\neg\alpha \vee \neg\beta \vee \gamma$ przesłanka
3. $\neg\tau \vee \beta$ przesłanka
4. τ przesłanka
5. $\neg\gamma$ przesłanka
6. $\neg\alpha \vee \neg\beta$ rezolwenta 2 i 5
7. $\neg\beta$ rezolwenta 6 i 1
8. $\neg\tau$ rezolwenta 3 i 7
9. $\square$ rezolwenta 4 i 8.

PRZYKŁAD 10.5.5.

Pokażemy, że formuła β wynika logicznie z następującego zbioru formuł:

$$S = \{\alpha \rightarrow \beta, (\gamma \wedge \delta) \rightarrow \alpha, (\tau \wedge \gamma) \rightarrow \delta, (\theta \wedge \alpha) \rightarrow \gamma, (\theta \wedge \tau) \rightarrow \gamma, \theta, \tau\}.$$

Każda formuła ze zbioru S jest równoważna alternatywie elementarnej:

1. $\neg\alpha \vee \beta$
2. $\neg\gamma \vee \neg\delta \vee \alpha$
3. $\neg\tau \vee \neg\gamma \vee \delta$
4. $\neg\theta \vee \neg\alpha \vee \gamma$
5. $\neg\theta \vee \neg\tau \vee \gamma$
6. θ
7. τ.

Budujemy rezolucyjne wyprowadzenie β z powyższych klauzul:

1.	$\neg\alpha \vee \beta$	przesłanka
2.	$\neg\gamma \vee \neg\delta \vee \alpha$	przesłanka
3.	$\neg\tau \vee \neg\gamma \vee \delta$	przesłanka
4.	$\neg\theta \vee \neg\alpha \vee \gamma$	przesłanka
5.	$\neg\theta \vee \neg\tau \vee \gamma$	przesłanka
6.	θ	przesłanka
7.	τ	przesłanka
8.	$\neg\tau \vee \gamma$	rezolwenta 5 i 6
9.	γ	rezolwenta 7 i 8
10.	$\neg\delta \vee \alpha$	rezolwenta 2 i 9
11.	$\neg\gamma \vee \delta$	rezolwenta 3 i 7
12.	$\neg\gamma \vee \alpha$	rezolwenta 2 i 11
13.	α	rezolwenta 9 i 12
14.	β	rezolwenta 1 i 13.

Ponieważ uzyskaliśmy rezolucyjne wyprowadzenie β z S, więc na mocy twierdzenia o pełności metody rezolucyjnej otrzymujemy, że $S \models_{KRZ} \beta$.

Dla porównania, przytoczmy jeszcze dowód założeniowy, że $S \vdash_{jas} \beta$:

1.	$\alpha \to \beta$	założenie
2.	$(\gamma \wedge \delta) \to \alpha$	założenie
3.	$(\tau \wedge \gamma) \to \delta$	założenie
4.	$(\theta \wedge \alpha) \to \gamma$	założenie
5.	$(\theta \wedge \tau) \to \gamma$	założenie
6.	θ	założenie
7.	τ	założenie
8.	$\theta \wedge \tau$	DK: 6,7
9.	γ	RO: 5,8
10.	$\tau \wedge \gamma$	DK: 7,9
11.	δ	RO: 3,10
12.	$\gamma \wedge \delta$	DK: 9,11
13.	α	RO: 2,12
14.	β	RO: 1,13.

Zauważmy, że dowód ten jest porównywalny — pod względem skomplikowania — z podanym wyżej dowodem rezolucyjnym.

Powyższe przykłady mogą osobie nieufnej nasunąć pytanie, po co właściwie zajmować się metodą rezolucji, skoro mamy inne, dobre metody dowodzenia tez. Podkreślamy, że metoda rezolucji znajduje zastosowanie przede wszystkim w automatycznym dowodzeniu twierdzeń. Przekształcenie nawet bardzo skomplikowanych formuł na równoważne im inferencyjnie kpn nie jest problemem dla szybkich maszyn liczących. Drugi krok w metodzie rezolucyjnej dowodzenia twierdzeń, czyli stosowanie samej reguły rezolucji, jest oczywiście także bardzo prostym zadaniem dla maszyn liczących. Warto zatem wyobrazić sobie np. zbiór liczący tysiące skomplikowanych przesłanek i odetchnąć z ulgą, że możemy w takiej sytuacji powierzyć robotę dedukcyjną Maszynom.

10.6. Konsekwencja rezolucyjna

Jest jasne, jak zdefiniować operację C_{res} konsekwencji wyznaczoną przez metodę rezolucji:

$$C_{res}(X) = \{\alpha \in F_{KRZ}X \vdash_{res} \alpha\}.$$

Tak zdefiniowana operacja konsekwencji ma własności (C1)–(C4) podane na wykładach 5–7.

* * *

Na wykładach w styczniu 2008 roku poznamy jeszcze jedną relację konsekwencji w KRZ, a mianowicie konsekwencję wyznaczoną przez metodę ***tablic analitycznych***.

10.7. Uwagi końcowe

Uwaga. Jest wiele różnych, bardziej subtelnych od powyższego — całkowicie ogólnego — rodzajów rezolucji. Problematyka ta jest intensywnie badana, przede wszystkim w związku z zastosowaniami metody rezolucji w automatycznym dowodzeniu twierdzeń.

Dlaczego ważne są klauzule Hornowskie? Klauzulę Hornowską, w której występuje dokładnie jeden literał pozytywny i dowolna (skończona) liczba literałów negatywnych nazywamy *klauzulą programową*. Ów literał pozytywny nazywamy *nagłówkiem* klauzuli, a pozostałe literały negatywne *treścią klauzuli*. Zbiór klauzul o tym samym nagłowku jest *procedurą*, a zbiór procedur jest *programem*. Klauzulę Hornowską o jednym literale pozytywnym i bez literałów negatywnych nazywamy *faktem*, a klauzulę bez literału pozytywnego nazywamy *klauzulą negatywną*. Przez *regułę obliczeniową* rozumiemy regułę wybierania literału z klauzuli.

Język programowania PROLOG wykorzystuje właśnie takie konstrukcje syntaktyczne. Przypomnijmy, że jest to język *deklaratywny*. Wykorzystujemy w nim tzw. SLD-rezolucję (*Selection-rule driven Linear resolution for Definite clauses*), określoną dla programów w wyżej rozumianym sensie, a więc rodzin zbiorów klauzul Hornowskich.

Uwaga. Przypomnijmy następujące porównanie reguły rezolucji z regułą modus ponens w KRZ:

- REGUŁA REZOLUCJI: z formuł $\alpha \vee \gamma$ oraz $\neg\alpha \vee \beta$ wywnioskuj $\gamma \vee \beta$
- REGUŁA MODUS PONENS: z formuł $\alpha \rightarrow \beta$ oraz α wywnioskuj β (lub, w postaci równoważnej: z formuł $\neg\alpha \vee \beta$ oraz α wywnioskuj β).

Reguła *rezolucji* jest zatem szczególnym przypadkiem ogólniejszej reguły, tzw. *reguły cięcia*. W ramach niniejszego kursu nie przewiduje się omówienia tej problematyki.

Uwaga. Zainteresowanych zachęcamy do zajrzenia do rozdziału 3 książki Fitting 1990 (zwłaszcza do podrozdziału 3.3.), gdzie znajdujemy opis metody rezolucji w połączeniu z pewną (prostą i wielce naturalną!) techniką przekształcania formuł do równoważnych im inferencyjnie kpn oraz apn.

Uwaga historyczna. Church i Turing udowodnili podstawowe twierdzenia implikujące nierozstrzygalność klasycznego rachunku logicznego (logiki pierwszego rzędu). Z kolei, z wyników uzyskanych przez Herbranda i Skolema wynika, że rachunek ten jest półrozstrzygalny: jeśli jakaś formuła jest tautologią logiki pierwszego rzędu, to można tego dowieść (w skończonej liczbie kroków). Fakt ten, łącznie z powstaniem i rozwojem elektronicznych maszyn liczących inspirował do poszukiwania systemów automatycznego dowodzenia twierdzeń.

Algorytm zaproponowany przez Davisa i Putnama nawiązywał do twierdzenia Herbranda (redukującego, w ściśle określonym sensie, problem ustalania czy dana formuła jest tautologią logiki pierwszego rzędu do problemu tautologiczności pewnych formuł KRZ). Mówi się, że reguła rezolucji została po raz pierwszy wykorzystana przez J.A. Robinsona (1963–1964). Wiadomo też jednak, że już w XIX wieku używał jej konsekwentnie Lewis Carroll, w swoim algebraicznym ujęciu sylogistyki.

Więcej na ten temat np. w rozdziale siódmym monografii Marciszewski, Murawski 1995.

Uwaga. Należy być świadomym różnic pomiędzy:

- ***Wynikaniem logicznym*** a ***uzasadnianiem*** oraz ***uznawaniem*** zdań. Pierwsze z tych pojęć ma, w dzisiejszym rozumieniu, charakter obiektywny; drugie i trzecie mogą odwoływać się do różnych czynników, także natury pragmatycznej. Uzasadnianie może mieć postać precyzyjnego dowodu, ale może też odwoływać się do zabawnych reguł LOGIKI UZNANIOWEJ.
- ***Dowodzeniem*** a procedurami czysto ***algorytmicznymi***. Pierwsza z tych aktywności ma charakter ***twórczy***, drugie są działaniami wedle określonego przepisu.

Student, który zna jedynie KRZ i ma dopiero przed sobą wykład na temat KRP (Klasycznego Rachunku Predykatów) może odnieść (złudne i pochopne!) wrażenie, że dowodzenie ma w dowolnym systemie logicznym charakter czysto algorytmiczny i że w takim systemie zawsze istnieje efektywna metoda rozstrzygania, czy dana formuła jest tautologia tego systemu.

Zwróćmy uwagę, że jest zasadnicza różnica pomiędzy:

- ***sprawdzeniem***, że dany ciąg formuł jest dowodem jakiejś formuły α, ***a***
- ***znalezieniem*** dowodu formuły α.

Więcej na ten temat powiemy w semestrze letnim.

Wykorzystywana literatura

Bachmair, L., Ganzinger, H. 2001. Resolution theorem proving. W: *Handbook of Automated Reasoning.*, 19–99.

Bartley, W.W., III. 1977. *Lewis Carroll's Symbolic Logic.* Clarkson N. Potter, New York.

Fitting, M. 1990. *First-Order Logic and Automated Theorem Proving.* Springer Verlag, New York Berlin Heidelberg London Paris Tokyo Hong Kong.

Handbook of Automated Reasoning. 2001. A. Robinson, A. Voronkov (eds.), Elsevier, Amsterdam London New York Oxford Paris Shannon Tokyo, The MIT Press, Cambridge, Massachusetts.

Handbook of Tableau Methods. 1999. Edited by: D'Agostino, M., Gabbay, D.M., Hähnle, R., Posegga, J., Kluwer Academic Publishers, Dordrecht Boston London.

Hedman, S. 2004. *A first course in logic.* Oxford University Press.

Marciszewski, W., Murawski, R. 1995. *Mechanization of Reasoning in a Historical Perspective.* Rodopi, Amsterdam – Atlanta.

Nerode, A., Shore, R.A. 1997. *Logic for applications.* Springer.

* * *

JERZY POGONOWSKI
Zakład Logiki Stosowanej UAM
www.logic.amu.edu.pl