Andrzej Wiśniewski

Logika I

Materiały do wykładu dla studentów kognitywistyki

Wykład 9. Koniunkcyjne postacie normalne i rezolucja w KRZ

Definicja 9.1. *Formuła A jest inferencyjnie równoważna formule B wtw formuła postaci $A \leftrightarrow B$ jest tezą KRZ.*

Notacja: To, że formuła A jest inferencyjnie równoważna formule B, zapisujemy $A \approx B$.

Przykład 9.1. Formuła p jest inferencyjnie równoważna formule $\neg\neg p$ (ponieważ $p \leftrightarrow \neg\neg p$ jest tezą *KRZ*).

Przykład 9.2. Formuła $p \to (q \to r)$ jest inferencyjnie równoważna formule $p \wedge q \to r$ (jako że tezami są: $(p \to (q \to r)) \to (p \wedge q \to r)$, $(p \wedge q \to r) \to (p \to (q \to r))$ oraz Ax. 14).

Twierdzenie 9.1. Relacja $\approx$ jest zwrotna, symetryczna i przechodnia w zbiorze formuł, tj. jest ona relacją równoważnościową w zbiorze formuł.

Twierdzenie 9.2. (i) *Jeżeli $A \approx B$ oraz A jest tezą KRZ, to B jest tezą KRZ*; (ii) *Jeżeli $A \approx B$ oraz A jest tautologią KRZ, to B jest tautologią KRZ.*

Definicja 9.2. *Literałami nazywamy wyrażenia o postaci p_i oraz $\neg p_i$, gdzie p_i jest zmienną zdaniową.*

Przykład 9.3. Literałami są m.in.: p, $\neg p$, q, $\neg q$, r, $\neg r$ (ponieważ $p = p_1$, $q = p_2$, $r = p_3$), p_{1276}, $\neg p_{9864321}$, etc.

Definicja 9.3. (*Alternatywy elementarne*)

- (i) *Każdy literał jest alternatywą elementarną*;
- (ii) *jeżeli A jest alternatywą elementarną i L jest literałem, to wyrażenie postaci* $(A \vee L)$ *jest alternatywą elementarną*;
- (iii) *nie ma żadnych innych alternatyw elementarnych poza tymi, które są wymienione w punkcie* (i) *i tymi, które można zbudować wedle reguły* (ii).

Przykład 9.4. Następujące formuły są alternatywami elementarnymi: p, $\neg p$, $(p \vee \neg p)$, $((p \vee q) \vee r)$, $((p \vee q) \vee \neg r)$, $(((p \vee q) \vee \neg r) \vee p)$.

Komentarz: „Składnikami” alternatyw elementarnych są wyłącznie literały, tj. zmienne zdaniowe i/lub negacje zmiennych zdaniowych.

Twierdzenie 9.3. *Jeżeli A i B są dowolnymi alternatywami elementarnymi, to formuła o postaci* ($A \vee B$) *jest inferencyjnie równoważna pewnej alternatywie elementarnej.*

Notacja: W alternatywach elementarnych opuszczamy nawiasy (poza najbardziej zewnętrznymi). Przykładowo, zamiast $(((p \vee q) \vee \neg r) \vee p)$ piszemy $(p \vee q \vee \neg r \vee p)$.

Twierdzenie 9.4. *Alternatywa elementarna jest tautologią KRZ wtw co najmniej jedna zmienna zdaniowa występuje w niej zarówno ze znakiem negacji, jak i bez tego znaku.*

Definicja 9.4. (*Koniunkcyjna postać normalna*)

(i) *Każda alternatywa elementarna jest formułą o koniunkcyjnej postaci normalnej*;

(ii) *jeżeli A jest dowolną formułą o koniunkcyjnej postaci normalnej, zaś B jest dowolną alternatywą elementarną, to wyrażenie o postaci* ($A \wedge B$) *jest formułą o koniunkcyjnej postaci normalnej*;

(iii) *nie ma żadnych innych formuł o koniunkcyjnej postaci normalnej prócz tych, które są alternatywami elementarnymi lub też dają się zbudować wedle reguły* (ii).

Notacja: W formułach o koniunkcyjnej postaci normalnej opuszczamy nawiasy (poza tymi, które „otaczają” alternatywy elementarne).

Przykład 9.5. Poniższe wyrażenia są formułami o koniunkcyjnej postaci normalnej (zastosowaliśmy opuszczanie nawiasów):

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

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

Twierdzenie 9.5. *Formuła o koniunkcyjnej postaci normalnej jest tautologią KRZ wtw każda z alternatyw elementarnych składających się na tę formułę jest tautologią.*

Komentarz: Przypominam, że zgodnie z twierdzeniem 9.4 dla tautologiczności alternatywy elementarnej potrzeba i wystarcza, aby jakaś zmienna występowała w niej ze znakiem negacji i bez tego znaku.

Twierdzenie 9.6. *Każda formuła jest inferencyjnie równoważna pewnej formule o koniunkcyjnej postaci normalnej.*

Twierdzenie 9.7. *Każda tautologia KRZ jest inferencyjnie równoważna pewnej formule o koniunkcyjnej postaci normalnej, takiej, że w każdej wchodzącej w jej skład alternatywie elementarnej co najmniej jedna zmienna zdaniowa występuje zarówno ze znakiem negacji, jak i bez tego znaku.*

Metoda sprowadzania formuł do koniunkcyjnej postaci normalnej

Dowolną formułę można „sprowadzić” do koniunkcyjnej postaci normalnej (tj. można znaleźć formułę, która jest jej inferencyjnie równoważna i ma koniunkcyjną postać normalną), postępując następująco:

(1) Usuwamy z formuły spójniki równoważności i implikacji zgodnie z następującymi prawami:

$$(p \leftrightarrow q) \leftrightarrow (p \rightarrow q) \wedge (q \rightarrow p),$$
$$(p \rightarrow q) \leftrightarrow \neg p \vee q.$$

Uwaga: W miarę potrzeb korzystamy z odpowiednich podstawień powyższych praw. Sytuacja jest analogiczna, gdy korzystamy z innych praw (zob. niżej).

(2) Negacje znajdujące się przed nawiasami obejmującymi formuły o postaci koniunkcji i alternatyw wprowadzamy do nawiasów w oparciu o prawa De Morgana:

$$\neg(p \wedge q) \leftrightarrow \neg p \vee \neg q,$$
$$\neg(p \vee q) \leftrightarrow \neg p \wedge \neg q.$$

Uwaga: Najlepiej jest rozpoczynać od negacji „najbardziej zewnętrznych.”

Metoda sprowadzania formuł do koniunkcyjnej postaci normalnej

(3) Podwójne i wielokrotne negacje likwidujemy w oparciu o prawo podwójnego przeczenia:

$$\neg\neg p \leftrightarrow p.$$

(4) Stosujemy prawa lewostronnej i prawostronnej rozdzielności alternatywy względem koniunkcji:

$$p \vee (q \wedge r) \leftrightarrow (p \vee q) \wedge (p \vee r),$$
$$(p \wedge q) \vee r \leftrightarrow (p \vee r) \wedge (q \vee r).$$

Przykład 9.6.

$$p \rightarrow (q \rightarrow r)$$

$$\neg p \vee (\neg q \vee r)$$

$$\neg p \vee \neg q \vee r$$

Metoda sprowadzania formuł do koniunkcyjnej postaci normalnej

Przykład 9.7.

$$p \wedge q \rightarrow r$$
$$\neg(p \wedge q) \vee r$$
$$(\neg p \vee \neg q) \vee r$$
$$\neg p \vee \neg q \vee r$$

Przykład 9.8.

$$(p \rightarrow q) \wedge p \rightarrow q$$
$$\neg((\neg p \vee q) \wedge p) \vee q$$
$$(\neg(\neg p \vee q) \vee \neg p) \vee q$$
$$((\neg\neg p \wedge \neg q) \vee \neg p) \vee q$$
$$((p \wedge \neg q) \vee \neg p) \vee q$$
$$((p \vee \neg p) \wedge (\neg q \vee \neg p)) \vee q$$
$$((p \vee \neg p) \vee q) \wedge ((\neg q \vee \neg p) \vee q)$$
$$(p \vee \neg p \vee q) \wedge (\neg q \vee \neg p \vee q)$$

Metoda sprowadzania formuł do koniunkcyjnej postaci normalnej

Przykład 9.9.

$$\neg((p \to q) \wedge p \to q)$$
$$\neg(\neg((\neg p \vee q) \wedge p) \vee q)$$
$$\neg\neg((\neg p \vee q) \wedge p) \wedge \neg q$$
$$((\neg p \vee q) \wedge p) \wedge \neg q$$
$$(\neg p \vee q) \wedge p \wedge \neg q$$

Przykład 9.10.

$$\neg(p \wedge q \to p)$$
$$\neg(\neg(p \wedge q) \vee p)$$
$$\neg\neg(p \wedge q) \wedge \neg p$$
$$(p \wedge q) \wedge \neg p$$
$$p \wedge q \wedge \neg p$$

Pojęcie wartościowania zostało wprowadzone na wykładzie 4, podobnie jak pojęcie wynikania logicznego na gruncie *KRZ*. Dalej zakładam, że pojęcia te są znane.

Definicja 9.4. *Formuła A jest spełnialna wtw istnieje co najmniej jedno wartościowanie* **v** *takie, że* **v**(A) = **1**.

Można łatwo udowodnić:

Twierdzenie 9.8. *Formuła A jest tautologią KRZ wtw formuła o postaci* $\neg A$ *nie jest spełnialna.*

Twierdzenie 9.9. *Formuła B wynika logicznie na gruncie KRZ ze zbioru formuł* $\{A_1, A_2, \ldots, A_n\}$ wtw formuła $\neg(A_1 \wedge A_2 \wedge \ldots \wedge A_n \to B)$ *nie jest spełnialna.*

Tak więc zagadnienia tautologiczności i wynikania logicznego (na gruncie *KRZ*) sprowadzają się do zagadnienia spełnialności odpowiednich formuł.

Metoda rezolucji dostarcza nam sposobu rozwiązywania zagadnienia spełnialności. Jednocześnie oparte na niej algorytmy stanowią podstawę wielu aplikacji z zakresu Sztucznej Inteligencji.

Potrzebujemy trochę teorii.

Definicja 9.5. *Klauzula jest to skończony zbiór literałów.*

Szczególnym przypadkiem klauzuli jest *klauzula pusta*, tj. taka, w której skład nie wchodzą żadne literały. Klauzulę pustą oznaczamy przez $\square$.

Niepusta klauzula jest, mówiąc ogólnie, *odpowiednikiem alternatywy elementarnej*.

Skończone zbiory klauzul nazywamy *klauzulowymi postaciami formuł*. Są one *odpowiednikami formuł o koniunkcyjnej postaci normalnej*.

Definicja 9.6. *Niech A będzie formułą o koniunkcyjnej postaci normalnej*:

$$(L_1^{\,1} \vee \ldots \vee L_1^{\,n_1}) \wedge \ldots \wedge (L_k^{\,1} \vee \ldots \vee L_k^{\,n_k}).$$

Postacią klauzulową formuły A nazywamy następujący zbiór:

$$\{\{L_1^{\,1}, \ldots, L_1^{\,n_1}\}, \ldots, \{L_k^{\,1}, \ldots, L_k^{\,n_k}\}\}.$$

Komentarz: Gdy mamy formułę o koniunkcyjnej postaci normalnej, jej postać klauzulową otrzymujemy poprzez „zapisanie" zbioru, którego elementami są zbiory literałów alternatyw elementarnych wchodzących w skład tej formuły.

Uwaga: Literały w alternatywie elementarnej mogą się powtarzać; ponieważ klauzula jest zbiorem, zapisując ją eliminujemy powtórzenia.

Może się też zdarzyć, że w formule o koniunkcyjnej postaci normalnej wystąpią alternatywy elementarne, którym odpowiada ta sama klauzula. Wówczas zapisujemy ją tylko raz.

Przykład 9.11. Sprowadzenie formuły $\neg(p \vee \neg p)$ do koniunkcyjnej postaci normalnej daje formułę $\neg p \wedge p$. Jej postacią klauzulową będzie:

$$\{\{p\}, \{\neg p\}\}.$$

Przykład 9.12. Formuła $p \vee p$ jest już w koniunkcyjnej postaci normalnej. Jej postacią klauzulową jest:

$$\{\{p\}\}$$

Przykład 9.13. Postać klauzulowa formuły:

$$(p \vee \neg p \vee \neg p \vee q) \wedge (\neg q \vee \neg q \vee \neg p \vee q)$$

jest następującym zbiorem zbiorów literałów:

$$\{\{p, \neg p, q\}, \{\neg q, \neg p, q\}\}.$$

Uwaga: Powyższą postać klauzulową informatyk zapisze skrótowo tak oto:

$$\{p\neg pq\ ,\ \neg q\neg pq\}.$$

Definicja 9.7. *Dla dowolnej zmiennej zdaniowej p_i, literały p_i oraz $\neg p_i$ nazywamy komplementarnymi.*

Jeżeli literał L ma postać p_i, to jego dopełnieniem jest literał $\neg p_i$. Jeżeli literał ma postać $\neg p_i$, to jego dopełnieniem jest literał p_i. Dopełnienie literału L oznaczamy symbolem L.*

Definicja 9.8. *Niech C_1, C_2 będą klauzulami takimi, że $L \in C_1$ i $L^* \in C_2$. Klauzule C_1, C_2 nazywamy klauzulami kolidującymi i mówimy, że kolidują one ze względu na komplementarne literały L, L^*. Resolwentem klauzul C_1 i C_2 nazywamy klauzulę C postaci:*

$$(C_1 \setminus \{L\}) \cup (C_2 \setminus \{L^*\}).$$

Przykład 9.14. Klauzule $\{p, q, r\}$ oraz $\{r, \neg p, s\}$ kolidują ze względu na literały: p, $\neg p$. Ich resolwentem jest klauzula $\{q, r, s\}$.

Przykład 9.15. Klauzule $\{\neg p, q\}$ i $\{p, \neg q\}$ kolidują ze względu na literały: $\neg p$, p. Ich resolwentem jest klauzula: $\{q, \neg q\}$. Rozważane klauzule kolidują również ze względu na literały: q, $\neg q$. W związku z tym ich resolwentem jest także klauzula $\{\neg p, p\}$.

Przykład 9.16. Klauzule $\{p\}$ oraz $\{\neg p\}$ kolidują ze względu na literały: p, $\neg p$. Ich resolwentem jest klauzula pusta $\Box$.

Reguła rezolucji: *Z klauzul kolidujących ze względu na komplementarne literały wolno wyprowadzić resolwent tych klauzul.*

Rozwiązując zagadnienie spełnialności formuły metodą rezolucji, staramy się wyprowadzić – za pomocą reguły rezolucji – klauzulę pustą $\Box$ z wyjściowego zbioru klauzul; zbiór ten jest klauzulową postacią analizowanej formuły. Gdy uda się wyprowadzić klauzulę pustą, wnosimy stąd, że analizowana formuła nie jest spełnialna.

W procesie wyprowadzania reguła rezolucji jest stosowana tylko do takich par klauzul kolidujących, do których nie była ona stosowana wcześniej. Tak więc do danej pary klauzul kolidujących reguła rezolucji jest stosowana co najwyżej jeden raz.

Kolejne wprowadzane resolwenty mogą pełnić funkcję przesłanek, z uwagi na które stosujemy następnie regułę rezolucji.

Przykład 9.17. Chcemy ustalić, czy formuła $\neg(p \vee \neg p)$ jest spełnialna.

Krok 1: Sprowadzamy formułę $\neg(p \vee \neg p)$ do koniunkcyjnej postaci normalnej; otrzymujemy:

$$\neg p \wedge p.$$

Krok 2: Przedstawiamy otrzymaną formułę w postaci klauzulowej; dostajemy:

$$\{\{p\}, \{\neg p\}\}.$$

Krok 3: Na mocy reguły rezolucji przechodzimy od klauzul kolidujących: $\{p\}$, $\{\neg p\}$ do ich resolwentu*;* w tym przypadku będzie nim klauzula pusta $\Box$. Dostajemy następującą formułę w postaci klauzulowej:

$$\{\Box\}.$$

Ponieważ pojawiła się klauzula pusta $\Box$, wnosimy, że analizowana formuła nie jest spełnialna.

Komentarz: Ściśle rzecz biorąc, sama metoda rezolucji jest stosowana tylko w kroku 3.

Uwaga: Informatyk przedstawi zbiór klauzul $\{\{p\}, \{\neg p\}\}$ następująco:

$$\{(1)p,\ (2)\neg p\ \}$$

gdzie numeracja służy do identyfikacji klauzul. Krok 3 zostanie teraz opisany tak oto:

$$3.\quad \Box \qquad 1, 2$$

gdzie numery po prawej stronie wskazują klauzule służące do utworzenia danego resolwentu.

Notacja: W dalszych przykładach będziemy zapisywać formuły w postaci klauzulowej „tak, jako to uczyni informatyk”: będziemy pomijać nawiasy klamrowe dla klauzul oraz przecinki w klauzulach, a także w miarę potrzeby będziemy numerować klauzule.

Przykład 9.18. Chcemy ustalić, czy formuła $\neg(p \wedge q \to p)$ jest spełnialna.

Przekształcamy analizowaną formułę do koniunkcyjnej postaci normalnej; otrzymujemy (por. Przykład 9.10):

$$p \wedge q \wedge \neg p.$$

Następnie przedstawiamy otrzymaną formułę w postaci klauzulowej; dostajemy:

$$\{(1)p,\ (2)q,\ (3)\neg p\}.$$

Klauzule p i $\neg p$ kolidują z uwagi na literały: p, $\neg p$. Na mocy reguły rezolucji wprowadzamy ich resolwent, którym jest klauzula pusta $\square$:

$$4. \qquad \square \qquad 1, 3$$

Ponieważ udało się nam się wyprowadzić klauzulę pustą, wnosimy stąd, że formuła $\neg((p \wedge q \to p)$ nie jest spełnialna.

Przykład 9.19. Chcemy ustalić, czy formuła $\neg((p \to q) \wedge p \to q)$ jest spełnialna. Sprowadzamy analizowaną formułę do koniunkcyjnej postaci normalnej; otrzymujemy (por. Przykład 9.9):

$$(\neg p \vee q) \wedge p \wedge \neg q.$$

Przedstawiamy otrzymaną formułę w postaci klauzulowej; dostajemy:

$$\{(1)\neg pq,\ (2)p,\ (3)\neg q\}$$

Klauzule $\neg pq$ i p kolidują z uwagi na literały: p, $\neg p$. Na mocy reguły rezolucji wprowadzamy ich resolwent, którym jest klauzula q:

$$4. \quad q \qquad\qquad 1, 2$$

Klauzule q i $\neg q$ kolidują z uwagi na komplementarne literały q i $\neg q$. Ich resolwentem jest klauzula pusta $\square$, którą wprowadzamy stosując regułę rezolucji:

$$5. \quad \square \qquad\qquad 4, 3$$

Wnosimy stąd, że formuła $\neg((p \to q) \wedge p \to q)$ nie jest spełnialna.

Przykład 9.19., raz jeszcze.

Chcemy ustalić, czy formuła $\neg((p \to q) \wedge p \to q)$ jest spełnialna.

KPN formuły $\neg((p \to q) \wedge p \to q)$: $(\neg p \vee q) \wedge p \wedge \neg q$

$\{(1)\neg pq,\ (2)p,\ (3)\neg q\}$

4. q 1, 2

5. $\square$ 4, 3

Formuła $\neg((p \to q) \wedge p \to q)$ nie jest spełnialna.

Rozważmy na koniec przykład formuły, której analiza metodą rezolucji nie prowadzi do uzyskania klauzuli pustej.

Przykład 9.20. Chcemy ustalić, czy formuła:

$$(p \vee \neg p \vee \neg p \vee q) \wedge (\neg q \vee \neg q \vee \neg p \vee q)$$

jest spełnialna.

Ponieważ powyższa formuła jest już w koniunkcyjnej postaci normalnej, przedstawiamy ją w postaci klauzulowej; dostajemy:

$$\{p\neg pq,\ \neg q\neg pq\}.$$

Występujące klauzule kolidują z uwagi literały: p, $\neg p$ oraz q, $\neg q$. Gdy zastosujemy regułę rezolucji z uwagi na p, $\neg p$, dostaniemy następującą formułę w postaci klauzulowej:

$$\{\neg pq\neg q\}$$

nie zawierającą pary klauzul kolidujących. Podobnie gdy zastosujemy regułę rezolucji ze względu na q, $\neg q$, dostaniemy:

$$\{p\neg pq\}$$

W obu przypadkach nie mamy możliwości dalszego zastosowania reguły rezolucji tak, aby otrzymać klauzulę pustą. Wnosimy stąd, że analizowana formuła wyjściowa jest spełnialna.

Zastanówmy się teraz, dlaczego metoda rezolucji działa. Potrzebujemy kilku definicji.

Definicja 9.9. *Wartościowanie* ***v*** *spełnia klauzulę C wtw istnieje literał L* $\in$ *C taki, że* ***v***(*L*) = **1**.

Klauzula C jest spełnialna wtw co najmniej jedno wartościowanie spełnia klauzulę C.

Wniosek 9.1. *Klauzula pusta* $\square$ *nie jest spełnialna*.

Powód jest prosty: gdyby klauzula pusta $\square$ była spełnialna, to istniałby literał *L* taki, że *L* $\in \square$ oraz dla pewnego wartościowania ***v***, ***v***(*L*) = **1**. A tymczasem dla każdego literału *L* mamy *L* $\notin \square$.

Definicja 9.10. *Wartościowanie **v** spełnia zbiór klauzul **K** wtw wartościowanie **v** spełnia każdą klauzulę należącą do zbioru klauzul **K**.*

*Zbiór klauzul **K** jest spełnialny wtw co najmniej jedno wartościowanie spełnia zbiór klauzul **K**.*

Wniosek 9.2. *Jeżeli klauzula pusta $\Box$ jest elementem zbioru klauzul **K**, to zbiór klauzul **K** nie jest spełnialny.*

Znów powód jest prosty: na mocy wniosku 9.1 żadne wartościowanie nie spełnia klauzuli pustej $\Box$, podczas gdy dla spełnialności zbioru klauzul niezbędne jest, aby jakieś wartościowanie (jednocześnie) spełniało wszystkie klauzule z tego zbioru.

Można łatwo udowodnić:

Twierdzenie 9.10. *Niech A będzie formułą i niech B będzie formułą o koniunkcyjnej postaci normalnej taką, że $A \approx B$. Niech $\boldsymbol{K}_B$ będzie postacią klauzulową formuły B. Następujące warunki są równoważne*:

(i) *formuła A jest spełnialna*;

(ii) *formuła B jest spełnialna*;

(iii) *$\boldsymbol{K}_B$ jest spełnialny*.

Teraz udowodnimy:

Twierdzenie 9.11. *Jeżeli wartościowanie* ***v*** *spełnia zbiór klauzul* ***K*** *oraz klauzula* C *jest resolwentem klauzul* C_1, $C_2 \in \mathbf{K}$, *to wartościowanie* ***v*** *spełnia zbiór klauzul*:

$$\mathbf{K} \cup \{C\}.$$

Dowód: Skoro C jest resolwentem klauzul C_1, C_2, to istnieje literał L taki, że $L \in C_1$ oraz $L^* \in C_2$. Mamy dwie możliwości:

(i) $\mathbf{v}(L) = \mathbf{0}$. Ponieważ ***v*** spełnia C_1, zatem istnieje literał $L' \in C_1$ (gdzie $L' \neq L$) taki, że $\mathbf{v}(L') = \mathbf{1}$. Jednocześnie $L' \in C$ na mocy definicji 9.8. Tak więc ***v*** spełnia C.

(ii) $\mathbf{v}(L) = \mathbf{1}$. Wówczas $\mathbf{v}(L^*) = 0$. Ponieważ ***v*** spełnia C_2, zatem istnieje literał $L'' \in C_2$ (gdzie $L'' \neq L^*$) taki, że $\mathbf{v}(L'') = \mathbf{1}$. Jednocześnie $L'' \in C$. Zatem ***v*** spełnia C.

Mamy zatem:

Wniosek 9.3: *Jeżeli zbiór klauzul **K** jest spełnialny oraz klauzula C jest resolwentem klauzul C_1, $C_2 \in$ **K**, to zbiór klauzul **K** $\cup \{C\}$ jest spełnialny*.

oraz na mocy wniosku 9.2 i definicji 9.10:

Wniosek 9.4: *Jeżeli klauzula pusta □ jest resolwentem klauzul C_1, C_2 należących do zbioru klauzul **K**, to zbiór klauzul **K** nie jest spełnialny*.

Możemy teraz wyjaśnić, dlaczego metoda rezolucji działa.

Przypuśćmy, że mamy skończony i niepusty zbiór klauzul $\boldsymbol{K}$, zawierający klauzule kolidujące. Rozważmy teraz skończony ciąg **s** = $<\boldsymbol{K}_1, ..., \boldsymbol{K}_n>$ zbiorów klauzul, zdefiniowany następująco:

(i) $\boldsymbol{K}_1 = \boldsymbol{K}$;

(ii) $\boldsymbol{K}_{i+1} = \boldsymbol{K}_i \cup \{C_{i+1}\}$, gdzie C_{i+1} jest resolwentem pewnych klauzul ze zbioru $\boldsymbol{K}_i$ $(1 \leq i < n)$

taki, że $C_n = \Box$.

Ciąg **s** reprezentuje „udane” przekształcenie wyjściowego zbioru klauzul $\boldsymbol{K}$ za pomocą metody rezolucji: $\boldsymbol{K}_n = \boldsymbol{K}_{n-1} \cup \{\Box\}$.

Zachodzi:

($) dla dowolnego $i \leq n$: zbiór klauzul $\boldsymbol{K}_i$ nie jest spełnialny.

Dlaczego? Na mocy wniosku 9.4 zbiór klauzul $\boldsymbol{K}_n$ nie jest spełnialny. Jednocześnie na mocy wniosku 9.3 jeżeli zbiór klauzul $\boldsymbol{K}_n$ nie jest spełnialny, to zbiór klauzul $\boldsymbol{K}_{n-1}$ nie jest spełnialny, a gdy zbiór $\boldsymbol{K}_{n-1}$ nie jest spełnialny, to zbiór $\boldsymbol{K}_{n-2}$ nie jest spełnialny etc. [ściśle rzecz biorąc, zależności ($) dowodzimy indukcyjnie].

Wnosimy stąd, ze zbiór klauzul $\boldsymbol{K}_1$ nie jest spełnialny. Jednakże zbiór $\boldsymbol{K}_1$ jest naszym wyjściowym zbiorem klauzul $\boldsymbol{K}$.

I właśnie dlatego metoda rezolucji działa: wyprowadzenie klauzuli pustej jest równoznaczne z wykazaniem, że analizowany zbiór klauzul nie jest spełnialny, a zatem (na mocy Twierdzenia 9.10) również analizowana formuła nie jest spełnialna.

Innymi słowy, właśnie naszkicowaliśmy dowód następującego twierdzenia:

Twierdzenie 9.12. *Jeżeli na podstawie metody rezolucji ze zbioru klauzul zostanie wyprowadzona klauzula pusta, to ten zbiór klauzul nie jest spełnialny*.

Prawdziwe jest również:

Twierdzenie 9.13. *Jeżeli zbiór klauzul nie jest spełnialny, to za pomocą metody rezolucji można wyprowadzić z tego zbioru klauzulę pustą.*

Dodajmy na koniec, że metoda rezolucji jest metodą rozstrzygania, czy zbiór klauzul jest spełnialny, czy też nie. Można ją przedstawić w postaci **algorytmu**. O tym jednak powiemy później.

Literatura:

Dowody twierdzeń 9.1. - 9.7. można znaleźć (m.in.) w podręczniku Tadeusza Batoga: *Podstawy logiki*, Wydawnictwo Naukowe UAM, Poznań 1996.

Metoda rezolucji dla *KRZ* została omówiona m.in. w podręczniku: Mordechai Ben-Ari, *Logika matematyczna w informatyce*, Wydawnictwa Naukowo-Techniczne, Warszawa 2005. Nasza prezentacja różni się kilkoma szczegółami od zawartej w tym podręczniku.

Dowód Twierdzenia 9.13 można znaleźć w podręczniku Ben-Ariego.