

III.8. Rezolucja

Pokażemy teraz działanie pewnej metody dowodowej, związanej z metodą drzew semantycznych i mającej istotne zastosowania m.in. w automatycznym dowodzeniu twierdzeń.

III.8.1. Definicje

Jeśli P jest predykatem, to stosujemy skrótowe zapisy $P(\vec{x})$ oraz $P(\vec{t})$ dla formuł atomowych utworzonych z predykatu P oraz stosownej liczby jego argumentów — zmiennych (w pierwszym przypadku) lub termów (w drugim przypadku).

Klauzulą nazwiemy dowolny skończony zbiór literalów.

Literałem **komplementarnym** do literalu ℓ nazywamy literal $\bar{\ell}$, zdefiniowany następująco:

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

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

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

Klauzulą programową nazywamy każdą klauzulę z dokładnie jednym literalą pozytywnym.

Jeśli klauzula programowa zawiera jakieś literały negatywne, to nazywamy ją **regułą**; w przeciwnym przypadku nazywamy ją **faktem**.

Klauzulą celową nazywamy klauzulę bez literalów pozytywnych.

Programem nazywamy zbiór klauzul programowych (reguł lub faktów). Programy odpowiadają programom rozważanym w PROLOGu.

Klauzule reprezentują formuły w skolemowej postaci normalnej. Tak więc, np. klauzula $\{\neg P(x), Q(x)\}$ reprezentuje formułę $\forall x\forall y (\neg P(x) \vee Q(x))$ lub, co na jedno wychodzi, formułę $\forall x\forall y (P(x) \rightarrow Q(x))$.

Niech C_1 i C_2 będą dwiema klauzulami, które nie mają żadnych wspólnych zmiennych i są postaci $D_1 \cup \{P(\vec{t}_1), \dots, P(\vec{t}_n)\}$ oraz $D_2 \cup \{\neg P(\vec{s}_1), \dots, \neg P(\vec{s}_m)\}$, odpowiednio. Jeśli σ jest najbardziej ogólnym unifikatorem dla $\{P(\vec{t}_1), \dots, P(\vec{t}_n), P(\vec{s}_1), \dots, P(\vec{s}_m)\}$, to $D_1\sigma \cup D_2\sigma$ jest **rezolwentą** dla C_1 i C_2 . Czasem mówi się wtedy także, że $D_1\sigma \cup D_2\sigma$ jest **dzieckiem** swoich **rodziców** C_1 oraz C_2 .

Dowodem rezolucyjnym klauzuli C ze zbioru formuł S nazywamy każdy skończony ciąg klauzul C_1, \dots, C_n taki, że:

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

Jeśli istnieje dowód rezolucyjny C z S , to mówimy, że C jest **rezolucyjnie dowodliwa** z S i oznaczamy ten fakt przez $S \vdash_R C$.

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 $S \vdash_R \square$.

Rezolucyjnym drzewem dowodowym klauzuli C ze zbioru S nazywamy każde drzewo binarne T o następujących własnościach:

- korzeniem T jest C

- liśćmi T są pewne elementy zbioru S
- pozostałe (oprócz korzenia i liści) wierzchołki T są klauzulami
- bezpośrednimi następnikami wierzchołka D niebędącego liściem są klauzule D_1 oraz D_2 , których rezolwentą jest D .

Niech $res(S)$ będzie zbiorem zawierający wszystkie elementy S oraz rezolwenty wszystkich par elementów S . Dla $n \geq 1$, niech $res_{n+1} = res(res_n(S))$. Wreszcie, niech $\mathcal{R}(S)$ będzie sumą wszystkich zbiorów $res(S)$. Zbiór $\mathcal{R}(S)$ nazywamy **domknięciem rezolucyjnym** zbioru S .

Uwaga! W definicji dowodu rezolucyjnego oraz rezolucyjnego drzewa dowodowego dopuszczamy (jako przesłanki dowodu lub liście drzewa, odpowiednio) **formuły otrzymane z elementów zbioru S przez zastosowanie podstawień przemianowujących zmienne**.

Uwaga. Rozważamy teraz drzewa, których wierzchołki są znakowane **zbioremami** formuł.

III.8.2. Przykłady dowodów rezolucyjnych

PRZYKŁAD III.8.2.1. (Nerode, Shore 1997: 146–147).

Rezolwentą klauzul:

- $C_1 = \{Q(x), \neg R(y), P(x, y), P(f(z), f(z))\}$
- $C_2 = \{\neg N(u), \neg R(w), \neg P(f(a), f(a)), \neg P(f(w), f(w))\}$

jest klauzula:

$$C_3 = \{Q(f(a)), \neg R(f(a)), \neg N(u), \neg R(a)\}.$$

Aby się o tym przekonać, należy:

- zauważyć, że $C_1 = \{Q(x), \neg R(y)\} \cup \{P(x, y), P(f(z), f(z))\}$
- zauważyć, że $C_2 = \{\neg N(u), \neg R(w)\} \cup \{\neg P(f(a), f(a)), \neg P(f(w), f(w))\}$
- zastosować najbardziej ogólny unifikator $\sigma = \{x \mapsto f(a), y \mapsto f(a), z \mapsto a, w \mapsto a\}$ dla uzgodnienia zbioru literalów $\{P(x, y), P(f(z), f(z)), P(f(a), f(a)), P(f(w), f(w))\}$
- $\{Q(x), \neg R(y)\}\sigma = \{Q(f(a)), \neg R(f(a))\}$
- $\{\neg N(u), \neg R(w)\}\sigma = \{\neg N(u), \neg R(a)\}$.

PRZYKŁAD III.8.2.2. (Nerode, Shore 1997: 147).

Pokażemy, że warunki przechodniości i symetrii, tj. warunki:

$$\forall x \forall y \forall z ((P(x, y) \wedge P(y, z)) \rightarrow P(x, z))$$

$$\forall x \forall y (P(x, y) \rightarrow P(y, x))$$

implikują następujący warunek (euklidesowości):

$$\forall x \forall y \forall z ((P(x, y) \wedge P(z, y)) \rightarrow P(x, z)).$$

Powyższe warunki mają następujące reprezentacje w postaci klauzul (po rozdzieleniu zmiennych):

- $C_1 = \{\neg P(x, y), \neg P(y, z), P(x, z)\}$

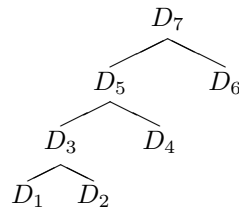
- $C_2 = \{\neg P(u, v), P(v, u)\}$
- $C_3 = \{\neg P(x, y), \neg P(z, y), P(x, z)\}$.

Chcemy zatem uzyskać dowód rezolucyjny C_3 z C_1 oraz C_2 . Będzie on się składał z trzech kroków. W każdym z nich z pary klauzul otrzymamy rezolwentę tej pary. W każdym kroku podkreślamy ten literal, względem którego dokonujemy rezolucji (tj. ten, który eliminujemy w wyniku danego kroku).

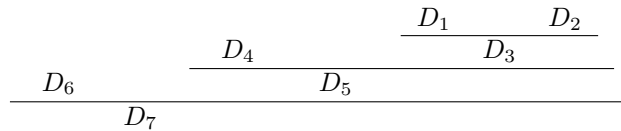
Tak więc, dowód rezolucyjny C_3 z C_1 oraz C_2 jest następującym ciągiem klauzul D_1, \dots, D_7 :

- $D_1 = C_1 = \{\neg P(x, y), \neg P(y, z), \underline{P(x, z)}\}$
- $D_2 = C_2\{u \mapsto x, v \mapsto z\} = \{\underline{\neg P(u, v)}, P(v, u)\}\{u \mapsto x, v \mapsto z\} = \{\underline{\neg P(x, z)}, P(z, x)\}$
- $D_3 = \{\neg P(x, y), \neg P(y, z), \underline{P(z, x)}\}$ rezolwenta D_1 oraz D_2
- $D_4 = C_2\{u \mapsto z, v \mapsto x\} = \{\underline{\neg P(u, v)}, P(v, u)\}\{u \mapsto z, v \mapsto x\} = \{\underline{\neg P(z, x)}, P(x, z)\}$
- $D_5 = \{\neg P(x, y), \underline{\neg P(y, z)}, P(x, z)\}$ rezolwenta D_3 i D_4
- $D_6 = C_2\{u \mapsto z, v \mapsto y\} = \{\underline{\neg P(u, v)}, P(v, u)\}\{u \mapsto z, v \mapsto y\} = \{\underline{\neg P(z, y)}, P(y, z)\}$
- $D_7 = \{\neg P(x, y), \neg P(z, y), P(x, z)\} = C_3$ rezolwenta D_5 i D_6 .

Rezolucyjne drzewo dowodowe wygląda w tym przypadku następująco:



Zwykle rezolucyjne drzewa dowodowe przedstawia się „korzeniem w dół, liśćmi do góry”. W takiej notacji rozważane rezolucyjne drzewo dowodowe wygląda następująco:



PRZYKŁAD III.8.2.3. (Hedman 2004: 124-125).

Niech:

- $C_1 = \{Q(x, y), P(f(x), y)\}$
- $C_2 = \{R(x, c), \neg P(f(c), x), \neg P(f(y), h(z))\}$.

Chcemy znaleźć rezolwentę C_1 oraz C_2 . Najpierw dokonamy przemianowania zmiennych (ponieważ pewne zmienne występują zarówno w C_1 , jak i w C_2). Mamy: $C_1\{x \mapsto u, y \mapsto v\} = \{Q(u, v), P(f(u), v)\}$. Widać, że $C_1\{x \mapsto u, y \mapsto v\}$ oraz C_2 nie mają wspólnych zmiennych.

Zauważmy, że predykat P występuje zarówno w literalach pozytywnych, jak i negatywnych rozpatrywanych klauzul. Dla zbioru:

$$\{P(f(u), v), \neg P(f(c), x), \neg P(f(y), h(z))\}$$

będziemy zatem szukać najbardziej ogólnego unifikatora. Po zastosowaniu (ćwiczenie!) algorytmu unifikacji widzimy, że takim mgu jest:

$$\sigma = \{u \mapsto c, v \mapsto h(z), x \mapsto h(z)\}.$$

Tak więc, rezolwentą klauzul C_1 oraz C_2 jest klauzula:

$$R = (C_1\sigma - \{P(f(u), v)\}) \cup (C_2\sigma - \{\neg P(f(c), x), \neg P(f(y), h(z))\}) = \\ = \{Q(u, v), R(x, c)\}\sigma = \{Q(c, h(z)), R(h(z), c)\}.$$

Sprawdzimy, że R jest logiczną konsekwencją C_1 oraz C_2 . Zgodnie z przyjętymi umowami notacyjnymi, klauzule reprezentują formuły w skolemowej postaci normalnej. Tak więc:

- C_1 reprezentuje formułę (1): $\forall x\forall y (Q(x, y) \vee P(f(x), y))$
- C_2 reprezentuje formułę (2): $\forall x\forall y\forall z (R(x, c) \vee \neg P(f(c), x) \vee \neg P(f(y), h(z)))$
- R reprezentuje formułę (3): $\forall z (Q(c, h(z)) \vee R(h(z), c))$.

Mamy pokazać, że dla dowolnej interpretacji \mathfrak{M} , jeśli formuły reprezentowane przez C_1 oraz C_2 są prawdziwe w \mathfrak{M} , to formuła reprezentowana przez R jest prawdziwa w \mathfrak{M} .

Przypuśćmy, że (1) i (2) są prawdziwe w interpretacji \mathfrak{M} . Ponieważ są to zdania uniwersalne, więc w \mathfrak{M} prawdziwe są też wszystkie podstawienia dowolnych termów (bez zmiennych) za zmienne x oraz y w (1) i (2). W szczególności:

- (3) $\forall z (Q(c, h(z)) \vee P(f(c), h(z)))$, czyli formuła reprezentowana przez klauzulę $C_1\{x \mapsto u, y \mapsto v\}\sigma$
- (4) $\forall z (R(h(z), c) \vee \neg P(f(c), h(z)))$, czyli formuła reprezentowana przez klauzulę $C_2\sigma$

są obie prawdziwe w \mathfrak{M} . Łatwo zauważyć, że formuły (3) i (4) są, odpowiednio, równoważne z (5) oraz (6):

- (5) $\forall z (\neg Q(c, h(z)) \rightarrow P(f(c), h(z)))$
- (6) $\forall z (P(f(c), h(z)) \rightarrow R(h(z), c))$

Z (5) i (6), na mocy praw KRP, otrzymujemy: (7) $\forall z (\neg Q(c, h(z)) \rightarrow R(h(z), c))$. Z kolei, formuła (7) jest równoważna z (8): $\forall z (Q(c, h(z)) \vee R(h(z), c))$.

Pokazaliśmy, że jeśli (1) i (2) są prawdziwe w dowolnej interpretacji \mathfrak{M} , to także (8) jest prawdziwa w interpretacji \mathfrak{M} .

Formuła (8) jest reprezentowana przez klauzulę R , a więc zakończyliśmy dowód.

III.8.3. Pożyczka z rozdziału IV: modele Herbranda

W rozdziale IV pokazujemy szczegółowe dowody twierdzeń metalogicznych dotyczących KRP. Jednym z najważniejszych takich twierdzeń jest twierdzenie Herbranda. Sformułujemy tutaj jego wersję uproszczoną, wystarczającą do dalszych rozważań tego podrozdziału.

Przypomnijmy (z podrozdziału III.0.) niektóre potrzebne pojęcia (uniwersa Herbranda, modele Herbranda).

Jeśli S jest dowolnym zbiorem formuł języka KRP (ustalonej sygnatury), to przez **uniwersum Herbranda** dla S rozumiemy zbiór H_S określony indukcyjnie następująco:

- (i) jeśli stała indywidualowa a_k występuje w jakiejś formule ze zbioru S , to $a_k \in H_S$
- (ii) jeśli t_1, \dots, t_{n_j} są dowolnymi termami należącymi do H_S , to $f_j^{n_j}(t_1, \dots, t_{n_j})$ także należy do H_S , dla dowolnego symbolu funkcyjnego $f_j^{n_j}$.

Jeśli w formułach z S nie występuje żadna stała indywidualowa, to warunek (i) definicji zbioru H_S zastępujemy warunkiem: $a_k \in H_S$ dla dowolnie wybranej stałej indywidualowej a_k .

Jeśli w formułach z S występuje co najmniej jeden symbol funkcyjny, to H_S jest zbiorem nieskończonym.

Uniwersum Herbranda dla danego zbioru formuł S jest zatem zbiorem wszystkich termów bez zmiennych utworzonych (z użyciem symboli funkcyjnych) ze stałych indywidualowych występujących w formułach zbioru S .

Interpretacją Herbranda dla zbioru formuł S nazywamy interpretację $\langle H_S, \Delta_S \rangle$ spełniającą następujące warunki:

- $\Delta_S(a_k) = a_k$ dla dowolnej stałej indywidualowej a_k należącej do H_S ;
- $\Delta_S(f_j^{n_j}(t_1, \dots, t_{n_j})) = f_j^{n_j}(t_1, \dots, t_{n_j})$ dla dowolnych termów t_1, \dots, t_{n_j} należących do H_S .

Modelem Herbranda dla zbioru formuł S nazywamy każdą interpretację Herbranda dla S , w której prawdziwe są wszystkie formuły z S .

Zauważmy, że uniwersa Herbranda tworzone są z wyrażeń języka KRP. **Alfabetem Herbranda** dla zbioru formuł S nazywamy zbiór wszystkich stałych pozalogicznych występujących w formułach z S (jeśli w S nie występuje żadna stała indywidualowa, to dodajemy dowolną ustaloną stałą indywidualową). Niech V_S oznacza alfabet Herbranda dla S .

W dowodzie twierdzenia Herbranda wykorzystywane jest twierdzenie o pełności metody drzew semantycznych w KRP. Twierdzenie to, wraz z dowodem, podajemy w rozdziale IV.

Ważną konsekwencją twierdzenia Herbranda jest możliwość wykazania niespełnialności zbioru formuł języka KRP w KRZ.

TWIERDZENIE 8.3.1. Niech $\Gamma = \{A_1, A_2, \dots, A_n \dots\}$ będzie zbiorem formuł w skolemowych postaciach normalnych nie zawierających wystąpień symbolu identyczności. Wtedy: Γ jest spełnialny wtedy i tylko wtedy, gdy Γ ma model Herbranda.

DOWÓD. Jeśli Γ ma model Herbranda, to Γ jest oczywiście spełnialny. Pozostaje udowodnić implikację w drugą stronę.

Przypuśćmy, że Γ jest spełnialny. Niech \mathfrak{N} będzie dowolną strukturą sygnatury V_S taką, że $\mathfrak{N} \models \Gamma$. Niech \mathfrak{M}' będzie interpretacją Herbranda. Zbudujemy interpretację \mathfrak{M} sygnatury V_S taką, że $\mathfrak{M} \models \Gamma$.

Uniwersum dla \mathfrak{M} jest uniwersum Herbranda H_Γ . Trzeba podać interpretację w \mathfrak{M} symboli funkcyjnych oraz predykatów. Mówiąc intuicyjnie:

- \mathfrak{M} interpretuje symbole funkcyjne tak, jak robi to \mathfrak{M}' (co nie wymaga uściśleń, ponieważ mowa tu o interpretacjach Herbranda: \mathfrak{M} jest interpretacją Herbranda sygnatury V_Γ , tak samo jak \mathfrak{M}');
- \mathfrak{M} interpretuje predykaty tak, jak robi to \mathfrak{N} (co wymaga uściślenia, bo uniwersa struktur \mathfrak{M} oraz \mathfrak{N} mogą być różne).

Dla dowolnego n -argumentowego predykatu R w V_Γ oraz termów t_1, \dots, t_n należących do H_Γ musimy określić, która z poniższych (nawzajem się wykluczających oraz dopełniających) możliwości zachodzi:

- $R(t_1, \dots, t_n)$
- $\neg R(t_1, \dots, t_n)$.

Ponieważ każdy z powyższych termów t_i jest termem bez zmiennych, a \mathfrak{N} jest strukturą sygnatury V_Γ , więc zachodzi dokładnie jedno z dwojga:

- $\mathfrak{N} \models R(t_1, \dots, t_n)$
- $\mathfrak{N} \models \neg R(t_1, \dots, t_n)$.

Definiujemy interpretację predykatu R w \mathfrak{M} w ten sposób, aby zachodziła równoważność:

$$\mathfrak{M} \models R(t_1, \dots, t_n) \text{ wtedy i tylko wtedy, gdy } \mathfrak{N} \models R(t_1, \dots, t_n).$$

Trzeba teraz pokazać, że $\mathfrak{M} \models \Gamma$. Dowód przeprowadzimy w trzech krokach:

- (1) pokażemy, że dla dowolnego zdania A języka sygnatury V_Γ , które nie zawiera ani kwantyfikatorów, ani znaku identyczności zachodzi: $\mathfrak{M} \models A$ wtedy i tylko wtedy, gdy $\mathfrak{N} \models A$.
- (2) pokażemy, że z (1) wynika, że dla dowolnego zdania A języka sygnatury V_Γ w skolemowej postaci normalnej, które nie zawiera znaku identyczności zachodzi: jeśli $\mathfrak{N} \models A$, to $\mathfrak{M} \models A$.
- (3) pokażemy, że z (2) wynika $\mathfrak{M} \models \Gamma$.

Dowód (1).

Niech A będzie formułą bez kwantyfikatorów. Pokażemy, że $\mathfrak{M} \models A$ wtedy i tylko wtedy, gdy $\mathfrak{N} \models A$ przez indukcję po złożoności A .

Jeśli A jest formułą atomową, to — ponieważ nie zawiera wystąpień predykatu identyczności — musi być postaci $R(t_1, \dots, t_n)$ dla pewnego predykatu n -argumentowego R z alfabetu V_Γ oraz termów t_1, \dots, t_n . Ponieważ A jest zdaniem, więc żaden z termów t_i nie może zawierać zmiennych. Oznacza to, że wszystkie termy t_i są elementami H_Γ . Z definicji \mathfrak{M} otrzymujemy wtedy, że $\mathfrak{M} \models A$ wtedy i tylko wtedy, gdy $\mathfrak{N} \models A$.

Przypuśćmy, że:

- $\mathfrak{M} \models A_1$ wtedy i tylko wtedy, gdy $\mathfrak{N} \models A_1$
- $\mathfrak{M} \models A_2$ wtedy i tylko wtedy, gdy $\mathfrak{N} \models A_2$.

Wtedy oczywiście także:

- $\mathfrak{M} \models \neg A_1$ wtedy i tylko wtedy, gdy $\mathfrak{N} \models \neg A_1$
- $\mathfrak{M} \models A_1 \wedge A_2$ wtedy i tylko wtedy, gdy $\mathfrak{N} \models A_1 \wedge A_2$.

(podobnie dla innych spójników zdaniowych). To kończy dowód (1).

Dowód (2).

Dowód przeprowadzimy przez indukcję względem liczby kwantyfikatorów w A . Jeśli A nie zawiera żadnych kwantyfikatorów, to na mocy (1) mamy: $\mathfrak{M} \models A$ wtedy i tylko wtedy, gdy $\mathfrak{N} \models A$.

Przypuśćmy, że A jest postaci $\forall x_1 \dots \forall x_n B$, gdzie B nie zawiera ani kwantyfikatorów, ani predykatu identyczności. Założenie indukcyjne głosi, że (2) zachodzi dla wszystkich formuł, które mają mniej niż n kwantyfikatorów. Niech $C(x_1)$ będzie formułą otrzymaną z $\forall x_1 \dots \forall x_n B$ poprzez opuszczenie pierwszego kwantyfikatora. Niech t będzie termem bez zmiennych z języka o sygnaturze V_Γ (oznacza to, że t jest elementem H_Γ). Mamy:

- jeśli $\mathfrak{N} \models C(x_1)$, to $\mathfrak{N} \models C(t/x_1)$ (z definicji relacji \models),
- jeśli $\mathfrak{N} \models C(t/x_1)$, to $\mathfrak{M} \models C(t/x_1)$ (z założenia indukcyjnego).

Tak więc, jeśli $\mathfrak{N} \models C(x_1)$, to $\mathfrak{M} \models C(t/x_1)$. Term t był dowolnie wybranym elementem zbioru H_Γ . Mamy zatem: jeśli $\mathfrak{N} \models A$, to $\mathfrak{M} \models C(t/x_1)$ dla wszystkich $t \in H_\Gamma$. Ponieważ H_Γ jest uniwersum struktury \mathfrak{M} , więc z definicji relacji \models otrzymujemy: $\mathfrak{M} \models \forall x_1 C(x_1)$. Ponieważ A jest identyczne z $\forall x_1 C(x_1)$, dowód (2) został zakończony.

Dowód (3).

Dla każdego $A_i \in \Gamma$ mamy:

- $\mathfrak{N} \models A_i$
- A_i jest w skolemowej postaci normalnej

- A_i nie zawiera predykatu identyeczności.

Spełnione są zatem wszystkie założenia (2). Tak więc, $\mathfrak{M} \models \Gamma$. To kończy dowód (3), a zarazem całego twierdzenia 8.3.1.

Z powyższego twierdzenia wynika w szczególności, że:

- (†) Jeśli A jest formułą w skolemowej postaci normalnej bez predykatu identyeczności, to: A jest spełnialna wtedy i tylko wtedy, gdy A ma model Herbranda.

Dlaczego w twierdzeniu 8.3.1. istotne było założenie, że Γ nie zawiera wystąpień znaku identyeczności? Poniższy przykład stanowi odpowiedź na to pytanie.

PRZYKŁAD III.8.3.1.1. (Hedman 2004: 115).

Rozważmy zdanie A postaci $\forall x ((f(x) \neq x) \wedge (f(f(x)) = x))$. Słownikiem Herbranda dla A jest zbiór $\{c, f\}$, gdzie c jest dowolną stałą indywiduową. Uniwersum Herbranda dla A jest zbiorem nieskończonym: $H_A = \{c, f(c), f(f(c)), f(f(f(c))), \dots\}$. W każdej interpretacji Herbranda elementy c oraz $f(f(c))$ są oczywiście różne. Ponieważ konsekwencją A jest zdanie $\forall x f(f(x)) = x$, więc w szczególności $c = f(f(c))$ także jest konsekwencją A . Zdanie A nie może zatem posiadać żadnego modelu Herbranda. Łatwo jednak zobaczyć, że zbiór $\{A\}$ jest spełnialny: modelem dla A jest np. zbiór wszystkich liczb całkowitych (bez zera), w którym symbol funkcyjny f interpretujemy tak, aby $f(x) = -x$.

Jak radzić sobie w przypadku, gdy rozważany zbiór formuł zawiera wystąpienia znaku identyeczności? Oto stosowna procedura.

Przypuścmy, że A jest formułą w skolemowej postaci normalnej i że w A występuje predykat identyeczności. Wtedy (†) nie zachodzi dla A . Zdefiniujemy formułę A^* taką, że:

- (†) zachodzi dla A^*
- (‡) A jest spełnialna wtedy i tylko wtedy, gdy A^* jest spełnialna.

Niech E będzie dwuargumentowym predykatem nie występującym w V_A . Poszukiwana formuła A^* jest koniunkcją formuł A_1, A_2, A_3 oraz A_4 , zdefiniowanych następująco:

- A_1 jest formułą powstającą z A poprzez zastąpienie każdej równości termów $t_1 = t_2$ występującej w A przez formułę $E(t_1, t_2)$.
- A_2 jest koniunkcją warunków stwierdzających, że E denotuje relację równoważności (tj. zwrotną, przechodnią i symetryczną).
- Dla każdego n -argumentowego predykatu R z V_A niech A_R będzie formułą:

$$\forall x_1 \dots \forall x_n \forall y_1 \dots \forall y_n \left(\bigwedge_{i=1}^n (E(x_i, y_i) \wedge R(x_1, \dots, x_n)) \rightarrow R(y_1, \dots, y_n) \right).$$

Niech A_3 będzie koniunkcją wszystkich formuł A_R .

- Dla każdego n -argumentowego symbolu funkcyjnego f z V_A niech A_f będzie formułą:

$$\forall x_1 \dots \forall x_n \forall y_1 \dots \forall y_n \left(\bigwedge_{i=1}^n E(x_i, y_i) \rightarrow E(f(x_1, \dots, x_n), f(y_1, \dots, y_n)) \right).$$

Niech A_4 będzie koniunkcją wszystkich formuł A_f .

Niech A^* będzie skolemową postacią normalną koniunkcji formuł A_1, A_2, A_3 oraz A_4 . Z konstrukcji A^* widać, że nie zawiera ona predykatu identyeczności. Tak więc, (†) zachodzi dla A^* . Niech wykazanie,

że zachodzi także (\dagger) będzie ćwiczeniem dla czytelniczek. Wskazówka: należy oczywiście rozważyć model ilorazowy.

Rozważmy jeszcze raz formułę A postaci $\forall x ((f(x) \neq x) \wedge (f(f(x)) = x))$ z przykładu 8.3.1.1. i zastosujemy do niej powyższą opisaną procedurę. Ponieważ formuła A^* spełnia warunek (\dagger), więc A^* posiada model Herbranda o uniwersum $H_A = \{c, f(c), f(f(c)), f(f(f(c))), \dots\}$. Możemy interpretować E w tym modelu jako relację równoważności o dwóch klasach: w jednej są terms zawierające parzystą liczbę wystąpień symbolu f , a w drugiej pozostałe terms. Wynika stąd również, że formuła A ma model dwuelementowy o uniwersum np. postaci $\{a, b\}$, w którym symbol funkcyjny f interpretujemy tak, aby: $f(a) = b$ oraz $f(b) = a$.

Teraz możemy opisać procedurę pozwalającą ustalać, czy dowolna formuła języka KRP jest niespełnialna.

Niech A będzie dowolną formułą języka KRP w skolemowej postaci normalnej, nie zawierającą predykatu identyczności. Tak więc, A jest postaci $\forall x_1 \dots \forall x_n B(x_1, \dots, x_n)$, gdzie B nie zawiera ani kwantyfikatorów, ani predykatu identyczności. Przypominamy, że H_A jest uniwersum Herbranda dla A . Zdefiniujemy zbiór:

$$E(A) = \{B(t_1, \dots, t_n) : t_1, \dots, t_n \in H_A\}.$$

Zbiór $E(A)$ otrzymujemy zatem przez podstawienia wszelkich możliwych termów z H_A za zmienne w B , na wszelkie możliwe sposoby. Niech $\{A_1, A_2, \dots\}$ będzie wyliczeniem wszystkich elementów zbioru $E(A)$. Pokażemy, że A jest spełnialna wtedy i tylko wtedy, gdy $E(A)$ jest spełnialny.

LEMAT 8.3.2. Niech A będzie dowolną formułą języka KRP w skolemowej postaci normalnej, nie zawierającą predykatu identyczności. Wtedy: A jest spełnialna wtedy i tylko wtedy, gdy $E(A)$ jest spełnialny.

Dowód.

Dowód implikacji z lewa na prawo jest dość prosty. Jeśli \mathfrak{M} jest modelem dla A , to:

$$\mathfrak{M} \models \forall x_1 \dots \forall x_n B(x_1, \dots, x_n).$$

W szczególności, $\mathfrak{M} \models B(t_1, \dots, t_n)$ dla wszystkich $t_1, \dots, t_n \in H_A$. Oznacza to, że $\mathfrak{M} \models A_i$ dla wszystkich i , a więc $\mathfrak{M} \models E(A)$.

Dla dowodu implikacji w drugą stronę założymy, że $E(A)$ jest spełnialny. Wtedy, na mocy twierdzenia 8.3.1., $E(A)$ ma model Herbranda \mathfrak{M} . Alfabet Herbranda dla $E(A)$ jest taki sam jak alfabet Herbranda dla A . Tak więc, uniwersum \mathfrak{M} jest równe H_A . Dla wszystkich $t_1, \dots, t_n \in H_A$, mamy $\mathfrak{M} \models B(t_1, \dots, t_n)$, ponieważ $B(t_1, \dots, t_n) \in E(A)$. Z definicji relacji \models otrzymujemy, że $\mathfrak{M} \models \forall x_1 \dots \forall x_n B(x_1, \dots, x_n)$. Oznacza to, że $\mathfrak{M} \models A$, czyli że A jest spełnialna.

Z powyższego wynika, że A **nie** jest spełnialna wtedy i tylko wtedy, gdy $E(A)$ **nie** jest spełnialny. Ponieważ $E(A)$ zawiera jedynie zdania bez kwantyfikatorów, więc możemy uważać $E(A)$ za zbiór formuł języka KRZ (po stosownych podstawieniach zmiennych zdaniowych za formuły atomowe). Ponieważ A jest w skolemowej postaci normalnej, więc każda formuła w $E(A)$ jest w koniunkcyjnej postaci normalnej. W rozdziale II udowodniono, że zbiór S formuł języka KRZ jest niespełnialny wtedy i tylko wtedy, gdy $\square \in \mathcal{R}(S)$ (zob. też Lemat 8.4.2. poniżej). Wiemy zatem, że $E(A)$ jest niespełnialny wtedy i tylko wtedy, gdy $\square \in \mathcal{R}(E(A))$. Z TWIERDZENIA O ZWARTOŚCI dla KRZ (również udowodnionego w rozdziale II) wynika, że $E(A)$ jest niespełnialny wtedy i tylko wtedy, gdy pewien skończony podzbiór $\{A_1, \dots, A_m\}$ zbioru $E(A)$ jest niespełnialny. Tak więc, jeśli A jest niespełnialna, to $\square \in \mathcal{R}(\{A_1, \dots, A_m\})$ dla pewnego m . Zauważmy, że $\square \in \mathcal{R}(\{A_1, \dots, A_m\})$ jest zbiorem **skończonym**.

Procedura powyższa dostarcza metody ustalania, że A jest niespełnialna. Sprawdzamy, czy dla pewnego m zachodzi $\square \in \mathcal{R}(\{A_1, \dots, A_m\})$. Jeśli odpowiedź jest twierdząca, to A jest niespełnialna. W przeciwnym przypadku sprawdzamy, czy $\square \in \mathcal{R}(\{A_1, \dots, A_m, A_{m+1}\})$, itd. Jeśli A **jest** niespełnialna, to ta procedura poda tę odpowiedź po skończonej liczbie kroków. Jeśli natomiast A **jest** spełnialna, to omawiana procedura nie zakończy się.

Zauważmy, że nie ma żadnego ograniczenia (z góry) liczby kroków, w której powyższa procedura ewentualnie się zakończy.

III.8.4. Trafność i pełność rezolucyjna

Pokażemy teraz, że metoda rezolucji ma „porządne” własności metalogiczne, tj. że jest (w ściśle określonym sensie) trafna oraz pełna.

Przypuśćmy, że literały pozytywne C_1, \dots, C_n nie zawierają zmiennych wolnych. Niech S będzie dowolnym zbiorem klauzul. Chcielibyśmy, aby rezolucyjny dowód klazuli pustej \square ze zbioru $S \cup \{\neg C_1, \dots, \neg C_n\}$ implikował, że wszystkie C_i ($1 \leq i \leq n$) wynikają logicznie ze zbioru S . Będzie to konsekwencją podanego niżej twierdzenia o pełności metody rezolucji.

Jeśli P jest programem, a $G = \{\neg A_1, \dots, \neg A_n\}$ klauzulą celową, to powiemy, że podstawienie θ (za zmienne z G) jest **podstawieniem wyznaczającym poprawną odpowiedź**, gdy $(A_1 \wedge \dots \wedge A_n)\theta$ wynika logicznie z (uniwersalnego domknięcia) P . W rozdziale IV pokażemy, że jeśli program $P \cup \{G\}$ nie jest spełnialny, to istnieje podstawienie wyznaczające poprawną odpowiedź o wartościach w zbiorze termów bez zmiennych. Będzie to konsekwencją twierdzenia Herbranda.

W rozdziale II omówiono metodę rezolucji dla KRZ. Pojęcia: dowodu rezolucyjnego, rezolucyjnej refutacji, rezolucyjnego drzewa dowodowego, domknięcia rezolucyjnego są w KRZ analogiczne, jak podane wyżej dla KRP (z oczywistymi modyfikacjami). Przypomnimy niżej (bez dowodów) pewne twierdzenia dotyczące rezolucji w KRZ. Będą one potrzebne w dowodach trafności i pełności rezolucji w KRP.

- LEMAT 8.4.1. Istnieje rezolucyjne drzewo dowodowe dla C z S wtedy i tylko wtedy, gdy C jest rezolucyjnie dowodliwa z S w KRZ.
- LEMAT 8.4.2. C jest rezolucyjnie dowodliwa z S w KRZ wtedy i tylko wtedy, gdy $C \in \mathcal{R}(S)$. W szczególności, istnieje rezolucyjna refutacja S wtedy i tylko wtedy, gdy $\square \in \mathcal{R}(S)$.
- LEMAT 8.4.3. Jeśli $S = \{C_1, C_2\}$ jest spełnialny w KRZ 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 .
- LEMAT 8.4.4. Dla dowolnego zbioru formuł T języka KRZ oraz dowolnego literału ℓ : jeśli T jest niespełnialny, to niespełnialny jest także zbiór $\{C \in \mathcal{R}(T) : \ell, \bar{\ell} \notin C\}$.
- TWIERDZENIE 8.4.5. (**Trafność rezolucji w KRZ**). Jeżeli istnieje rezolucyjna refutacja S , to S nie jest spełnialny w KRZ.
- TWIERDZENIE 8.4.6. (**Pełność rezolucji w KRZ**). Jeżeli S jest niespełnialny w KRZ, to istnieje rezolucyjna refutacja S .

Dowody tych twierdzeń podano w rozdziale II. Przypomnijmy jeszcze następujące porównanie reguły rezolucji z regułą modus ponens w KRZ:

- REGUŁA REZOLUCJI: z formuł $A \vee C$ oraz $\neg A \vee B$ wywnioskuj $C \vee B$
- REGUŁA MODUS PONENS: z formuł $A \rightarrow B$ oraz A wywnioskuj B (lub, w postaci równoważnej: z formuł $\neg A \vee B$ oraz A wywnioskuj B).

Najpierw udowodnimy trafność metody rezolucji, tj. pokażemy, że jeśli klauzula pusta należy do rezolucyjnego domknięcia zbioru S , to S nie jest spełnialny.

TWIERDZENIE 8.4.7. (**Trafność metody rezolucji w KRP**). Jeśli $\square \in \mathcal{R}(S)$, to S nie jest spełnialny.

DOWÓD. Niech $\square \in \mathcal{R}(S)$. Przypuśćmy, dla dowodu nie wprost, że S jest spełnialny, czyli że istnieje interpretacja \mathfrak{M} taka, że $\mathfrak{M} \models S$.

Dla udowodnienia twierdzenia wystarczy pokazać, że:

- (★) jeśli $\mathfrak{M} \models C_1$ i $\mathfrak{M} \models C_2$, to $\mathfrak{M} \models C$, gdzie C jest rezolwentą C_1 oraz C_2 .

Istotnie, jeśli zachodzi (\star), to można pokazać (przez indukcję po złożoności formuł), że $\mathfrak{M} \models C$ dla wszystkich $C \in \mathcal{R}(S)$. Ponieważ $\square \in \mathcal{R}(S)$, więc byłoby wtedy $\mathfrak{M} \models \square$, a to jest sprzeczność.

Dowód (\star) jest uogólnieniem procedury, którą wykonywaliśmy w przykładzie III.8.2.3.

Jeśli C jest rezolwentą C_1 i C_2 , to C ma postać $D_1\sigma \cup D_2\sigma$ (gdzie D_1, D_2 oraz σ są oznaczeniami takimi, jakie występują w definicji rezolwenty). Przypominamy, że klauzula odpowiada formule generalnie skwantyfikowanej.

Dla każdego podstawienia τ , którego wartościami są termy bez zmiennych mamy:

- $\mathfrak{M} \models D_1\sigma\tau$ albo
- $\mathfrak{M} \models D_2\sigma\tau$.

W konsekwencji, także $C\tau$, które jest sumą $D_1\sigma\tau$ oraz $D_2\sigma\tau$, jest prawdziwe w \mathfrak{M} .

Tak więc, jeśli ze zbioru S klauzul da się rezolucyjnie wyprowadzić klauzulę pustą \square (reprezentującą sprzeczność), to zbiór S jest niespełnialny, nie ma modelu.

Pokażemy teraz dowód implikacji odwrotnej, tj. tego, że jeśli zbiór S jest niespełnialny, to można z niego rezolucyjnie wyprowadzić klauzulę pustą \square .

Niech A będzie dowolną formułą języka KRP w skolemowej postaci normalnej, nie zawierającą predykatu identyczności. W Lemacie 8.3.2. pokazano, że wtedy: A jest spełnialna wtedy i tylko wtedy, gdy $E(A)$ jest spełnialny. Ponieważ elementami zbioru $E(A)$ są zdania (!) otrzymane z A przez opuszczenie kwantyfikatorów oraz zastąpienie wszystkich zmiennych termami z H_A , więc elementy $E(A)$ można traktować jak formuły języka KRZ.

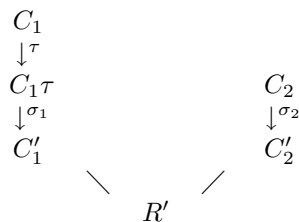
Przypuśćmy, że D_1 oraz D_2 są elementami $E(A)$ i że R' jest rezolwentą D_1 oraz D_2 *w sensie rezolucji w KRZ*. Istnieją wtedy klauzule C_1 oraz C_2 otrzymane z A (tj. klauzule odpowiadające dwóm członom koniunkcyjnej postaci A) takie, że $D_1 = C_1\sigma_1$ oraz $D_2 = C_2\sigma_2$ dla pewnych σ_1 oraz σ_2 . Pokażemy w następnym lemacie, że istnieje wtedy rezolwenta *w sensie rezolucji w KRZ* R dla C_1 oraz C_2 oraz podstawienie σ takie, że $R\sigma = R'$. Mówiąc nie całkiem precyzyjnie, lemat ten stwierdza, że cokolwiek może zostać wyprowadzone rezolucyjnie *w sensie KRZ* z $E(A)$, może też zostać wyprowadzone rezolucyjnie *w sensie KRP* z A .

LEMAT 8.4.8. (**Lemat o podnoszeniu**). Niech A będzie dowolną formułą języka KRP w skolemowej postaci normalnej. Jeśli $R' \in res(E(A))$, to istnieje $R \in res(A)$ taka, że $R\sigma' = R'$.

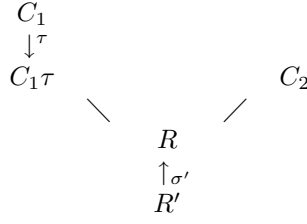
DOWÓD.

Może, przed właściwym dowodem, pożyteczna będzie pewna ilustracja (wyjaśniająca jednocześnie nazwę lematu).

Niech A będzie dowolną formułą języka KRP w skolemowej postaci normalnej. Niech C_1 i C_2 będą dwiema klauzulami z A . Niech τ będzie podstawieniem takim, że $C_1\tau$ oraz C_2 nie mają żadnych wspólnych zmiennych. Niech C'_1 i C'_2 będą takimi elementami $E(A)$, że $C_1\tau\sigma_1 = C'_1$ oraz $C_2\sigma_2 = C'_2$ dla pewnych podstawień σ_1 oraz σ_2 . Wreszcie, niech R' będzie rezolwentą (w sensie KRZ) dla C'_1 i C'_2 . Sytuację tę przedstawia diagram (kreski ukośne odpowiadają rezolucji):



Teza lematu o podnoszeniu mówi wtedy, że istnieje rezolwenta R dla $C_1\tau$ oraz C_2 (w sensie KRP) taka, że $R\sigma' = R'$, co symbolizuje poniższy diagram:



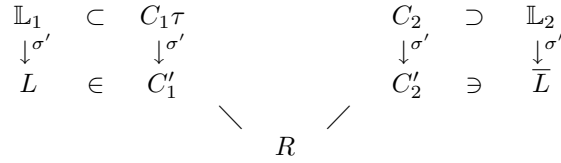
Pierwszy z powyższych diagramów dotyczy rezolucji w KRZ, drugi rezolucji w KRP.

Przejdźmy do dowodu. Przypuśćmy, że zachodzą założenia lematu (zob. pierwszy diagram). Wtedy, na mocy definicji rezolucji w KRZ, musi istnieć literał $L \in C_1'$ taki, że $\overline{LC_2}'$ oraz $R' = (C_1' - \{L\}) \cup (C_2' - \overline{L})$.

Niech $\sigma' = \sigma_1\sigma_2$. Ponieważ $C_1\tau$ oraz C_2 nie mają żadnych wspólnych zmiennych, więc $C_1\tau\sigma' = C_1\tau\sigma_1 = C_1'$ oraz $C_2\sigma' = C_2\sigma_2 = C_2'$.

Niech $\mathbb{L}_1 = \{L_1, \dots, L_n\}$ będzie zbiorem tych wszystkich literałów L_i z $C_1\tau$, dla których $L_i\sigma' = L$. Podobnie, niech $\mathbb{L}_2 = \{L'_1, \dots, L'_m\}$ będzie zbiorem tych wszystkich literałów L_i z C_2 , dla których $L'_i\sigma' = \overline{L}$.

Zilustrujmy dokonane konstrukcje diagramem:



Niech $\mathbb{L} = \mathbb{L}_1 \cup \mathbb{L}_2$. Zbiór \mathbb{L} jest uzgadnialny, ponieważ $\mathbb{L}\sigma' = \{\overline{L}\}$. Niech σ będzie najbardziej ogólnym unifikatorem dla \mathbb{L} . Z definicji rezolucji w KRP znajdujemy rezolwentę R dla C_1 oraz C_2 :

$$R = ((C_1\tau - \mathbb{L}_1) \cup (C_2 - \mathbb{L}_2))\sigma.$$

Trzeba jeszcze pokazać, że R' może zostać otrzymana z R przez podstawienie. Pokażemy, że $R\sigma' = R'$. Zwróćmy uwagę, że ponieważ σ' jest unifikatorem dla \mathbb{L} , a σ jest najbardziej ogólnym unifikatorem dla \mathbb{L} , więc $\sigma\sigma' = \sigma'$. Mamy następujący ciąg równości:

$$\begin{aligned}
R\sigma' &= ((C_1\tau - \mathbb{L}_1) \cup (C_2 - \mathbb{L}_2))\sigma\sigma' = \\
&= ((C_1\tau - \mathbb{L}_1) \cup (C_2 - \mathbb{L}_2))\sigma' = \\
&= (C_1\tau\sigma' - \mathbb{L}_1\sigma') \cup (C_2\sigma' - \mathbb{L}_2\sigma') = \\
&= (C_1' - \{L\}) \cup (C_2' - \{\overline{L}\}) = \\
&= R'
\end{aligned}$$

Tym samym dowód lematu o podnoszeniu został zakończony. Konsekwencją tego lematu jest lemat następujący.

LEMAT 8.4.9. Niech A będzie dowolnym zdaniem języka KRP w skolemowej postaci normalnej. Jeśli $C' \in \mathcal{R}(E(A))$, to istnieje $C \in \mathcal{R}(A)$ oraz podstawienie σ' takie, że $C\sigma' = C'$.

DOWÓD.

Jeśli $C' \in \mathcal{R}(E(A))$, to $C' \in res_n(E(A))$ dla pewnego n . Dowód przeprowadzimy przez indukcję po n .

Gdy $n = 0$, to $C\sigma' \in E(A)$. Wtedy, z definicji $E(A)$, $C\sigma'$ otrzymujemy przez podstawienie za zmienne termów bez zmiennych w pewnej klauzuli C z A .

W kroku indukcyjnym skorzystamy z lematu o podnoszeniu. Przypuśćmy, że dla pewnego m każda klauzula w $res_m(E(A))$ otrzymana jest z pewnej klauzuli z $\mathcal{R}(A)$ poprzez podstawienie. Niech $\tilde{A} \subset \mathcal{R}(A)$ będzie taka, że każda klauzula z $res_m(E(A))$ otrzymana jest z pewnej klauzuli z \tilde{A} . Wtedy $res_m(E(A)) \subset E(\tilde{A})$. Jeśli $C' \in res_{m+1}(E(A))$, to $C' \in res(E(\tilde{A}))$. Na mocy lematu o podnoszeniu, istnieje $C \in res(\tilde{A})$ taka, że $C\sigma' = C'$ dla pewnego podstawienia σ' . Ponieważ $\tilde{A} \subset \mathcal{R}(A)$, więc $C \in \mathcal{R}(A)$. To kończy dowód.

Powyższe lematy są potrzebne do udowodnienia najważniejszego wyniku w tym podrozdziale, a mianowicie twierdzenia o pełności rezolucji.

TWIERDZENIE 8.4.10. (*Twierdzenie o pełności rezolucji w KRP*). Niech A będzie dowolnym zdaniem języka KRP w skolemowej postaci normalnej. Jeśli A jest niespełnialna, to $\square \in \mathcal{R}(A)$.

DOWÓD.

Na mocy lematu 8.4.9., jeśli $\square \in \mathcal{R}(E(A))$, to istnieje $C \in \mathcal{R}(A)$ taka, że $C\sigma' = \square$ dla pewnego podstawienia σ' . To jednak jest możliwe jedynie wtedy, gdy $C = \square$. Tak więc, jeśli $\square \in \mathcal{R}(E(A))$, to $\square \in \mathcal{R}(A)$. Wnioskujemy stąd, że jeśli A jest niespełnialna, to $\square \in \mathcal{R}(A)$.

Uwaga. Czasem przez twierdzenie o pełności rezolucji w KRP rozumie się łącznie twierdzenia 8.4.7. oraz 8.4.10.

* * *

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. O pewnych zastosowaniach tej metody piszemy w rozdziale V.

III.8.5. Uwagi końcowe do rozdziału III

*Brauchen wir eine Lebensregel, so sei es diese:
„Bewahre den Geist der Jugend!”
Denn er ist der Sinn des Lebens.*
Moritz Schlick: Vom Sinn des Lebens.
Symposion 1, 1927: 354.

Rozdział III skryptu ma charakter propedeutyczny. Podrozdziały III.0.–III.1. wprowadzały podstawowe pojęcia związane z metodą drzew semantycznych w KRP. Część tego materiału zostanie przeniesiona do rozdziału I, gdzie piszemy m.in. o: językach formalnych, grafach i drzewach, dowodach indukcyjnych, itd. Podrozdziały III.2.–III.5. miały charakter zabawowy: omówione w nich przykłady mogą posłużyć do urozmaicenia kursu logiki. Podrozdziały III.6.–III.8. miały charakter kompilacyjny: sprawozdano w nich pewne ustalenia dotyczące prostych zastosowań KRP (III.6.) oraz pojęć unifikacji i rezolucji, ważnych dla materiału przedstawionego w rozdziale IV. Problematyka podrozdziału III.6. jest w bardziej rozwiniętej postaci przedstawiona w podrozdziale IV.9. Zadania podane w podrozdziale III.9. zostaną włączone do rozdziału VI, grupującego wszystkie zadania skryptu.

Metodę drzew semantycznych dla KRZ przedstawiono w rozdziale II. Związki metody drzew semantycznych z innymi metodami dowodowymi w Klasycznym Rachunku Logicznym omawiamy w rozdziale IV (przede wszystkim w IV.6.). Zastosowania metody drzew semantycznych omawiane są w rozdziale V.

* * *

Skrypt nie jest pomyślany jako standardowy podręcznik logiki. Ma być jedynie prezentacją określonej metody dowodowej — metody drzew semantycznych. Wydaje się, że tekst można zalecić jako uzupełnienie do przeprowadzenia *dwóch* kursów:

- KURS ZABAWOWY. Rozdziały: I, II, III (oraz część zadań z rozdziału VI). Materiał ten może stanowić lekturę uzupełniającą w kursie LOGIKI dla np.: filolożek, psycholożek, socjolożek, filozofek, lingwistek, itp.
- KURS STANDARDOWY. Rozdziały: I, IV, V (oraz część zadań z rozdziału VI). Ten materiał może być wykorzystany jako lektura uzupełniająca w kursie LOGIKI MATEMATYCZNEJ dla np.: studentów matematyki oraz informatyki.

Całość skryptu jest tak skomponowana, aby materiał dla każdego z obu wymienionych wyżej kursów mógł być poznawany niezależnie od drugiego.

* * *

Jak wiadomo, LOGIKA zajmuje się pojęciami: DOWODU oraz WYNIKANIA LOGICZNEGO. Związki między tymi pojęciami ustalają twierdzenia metalogiczne: TWIERDZENIE O TRAFNOŚCI oraz TWIERDZENIE O PEŁNOŚCI. Pewne TWIERDZENIA LIMITACYJNE określają samoograniczenia stosowalności metod KRP. Tak więc, KLASYCZNY RACHUNEK LOGICZNY jest dyscypliną o dobrze rozwiniętej metodologii. Należy jednak pamiętać, że LOGIKA nie jest dyscypliną zamkniętą. Nowe inspiracje dla niej znajdujemy na kilku obszarach. Wymieńmy trzy z nich:

- PRAKTYKA BADAWCZA MATEMATYKI. Pojęcie DOWODU rozważane w logice jest tylko idealizacją (normatywną względem *Przeszłości?*) praktyki matematycznej. Przy tym, to owa praktyka matematyków jest fundamentalna dla tworzenia pojęć logicznych (a nie na odwrót). Tak więc, rozwój logiki może być inspirowany przez badania matematyczne. W ten sposób powstały np. logiki wyższych rzędów, logiki infinitarne, logiki z uogólnionymi kwantyfikatorami, logiki intuicjonistyczne, itd.

- **TEORETYCZNE PODSTAWY INFORMATYKI.** Z oczywistych powodów badania informatyczne muszą być wspomagane badaniami logicznymi. Źródła informatyki teoretycznej poszukiwać należy przecież w rozważaniach logicznych. Rozwój informatyki inspiruje z kolei nowe badania logiczne. W ten sposób powstały np. logiki algorytmiczne, nowe interpretacje dla logik modalnych, itd.
- **PRAGMATYKA LOGICZNA.** Pojęcie **NIEZAWODNEJ REGUŁY WNISKOWANIA** zostało wyabstrahowane (w cywilizacji Zachodu) z rozumowań przeprowadzanych w językach naturalnych. W tej postaci, w jakiej stosowane jest ono obecnie (odwołującej się do czysto FORMALNYCH, składniowych, własności komunikatów oraz do znaczenia ustalonego zestawu STAŁYCH LOGICZNYCH) jest ono adekwatne do opisu tworzenia i przekształcania WIEDZY w aparaturze pojęciowej poszczególnych nauk. Jest problemem otwartym, czy *obecnie* znane systemy logiczne potrafią trafnie reprezentować wszelkie rozumowania przeprowadzane w języku naturalnym, którym chcielibyśmy nadać walor — jakoś pragmatycznie rozumianej — prawomocności. Stąd kolejna inspiracja dla badań logicznych.

* * *

Jednym ze skromnych celów niniejszego skryptu jest to, aby uświadomić ewentualnym czytelnikom różnice między:

- **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.

Jeśli lektura skryptu nie przeszkodzi w osiągnięciu tych celów, to pozwolimy sobie uznać, że praca nad nim miała SENS.

Odnosiniki bibliograficzne do podrozdziału III.8.

- Bachmair, L., Ganzinger, H. 2001. Resolution theorem proving. W: *Handbook of Automated Reasoning.*, 19–99.
- Ben-Ari, M. 2005. *Logika matematyczna w informatyce*. Wydawnictwa Naukowo Techniczne.
- 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