Programowanie w logice *

Antoni Kościelski

Wykład 1.

1.1 Język pierwszego rzędu

Definiując język pierwszego rzędu posługujemy się pewnym alfabetem złożonym z symboli kilku rodzajów:

1) symboli zmiennych (zmiennych); symbole te będziemy oznaczać literami takimi, jak $x, y, z, u, v, \ldots$
2) symboli stałych (stałych), oznaczanych literami $a, b, c, d, \ldots$
3) symboli funkcyjnych, oznaczanych literami $f, g, h, \ldots$
4) symboli relacyjnych, oznaczanych literami $p, q, r, \ldots$
5) symboli **true** i **false** oznaczających stałe logiczne,
6) spójników $\neg$ (negacja), $\vee$ (alternatywa), $\wedge$ (koniunkcja), $\rightarrow$ (implikacja) i $\leftrightarrow$ (równoważność),
7) kwantyfikatorów $\exists$ (istnieje) oraz $\forall$ (dla każdego) i
8) symboli pomocniczych takich jak nawiasy (i) oraz przecinek ,.

Zakładamy, że dla każdego symbolu funkcyjnego i relacyjnego jest określona liczba naturalna zwana arnością. Zakładamy, że arność symbolu funkcyjnego jest dodatnia (czasem symbole stałych uważa się za symbole funkyjne arności 0). Symbole relacyjne arności 0 będzie nazywać zmiennymi zdaniowymi.

Zbiorem termów nazywamy najmniejszy spośród zbiorów X spełniających następujące warunki:

1) zmienne są elementami X,
2) stałe są elementami X,

*Wykłady są prowadzone w dużej mierze według pracy Introduction to Logic Programming Krzysztofa Apta (patrz Handbook of Theoretical Computer Science, vol. B)

3) jeżeli f jest symbolem funkcyjnym arności n, a $t_1, \ldots, t_n$ są elementami X, to $f(t_1, \ldots, t_n)$ jest elementem zbioru X.

Elementy zbioru termów nazywamy termami.

Formułami atomowymi (atomami) nazywamy słowa postaci $p(t_1, \ldots, t_n)$, gdzie $t_1, \ldots, t_n$ są termami, a p jest symbolem relacyjnym arności n.

Zbiorem formuł nazywamy najmniejszy spośród zbiorów X spełniających następujące warunki:

1) formuły atomowe są elementami X,

2) stałe **true** i **false** są elementami X,

3) jeżeli φ i ψ są elementami X, to także $(\neq \varphi)$, $(\varphi \vee \psi)$, $(\varphi \wedge \psi)$, $(\varphi \rightarrow \psi)$ oraz $(\varphi \leftrightarrow \psi)$ są elementami X,

4) jeżeli φ jest elementem X, a x jest zmienną, to $(\exists x\ \varphi)$ oraz $(\forall x\ \varphi)$ są także elementami X.

1.2 Tautologie

Przypuśćmy, że mamy dany alfabet języka pierwszego rzędu. Strukturą dla tego alfabetu nazywamy układ

$$\mathcal{A} = \langle A, p^{\mathcal{A}}, q^{\mathcal{A}}, \ldots, f^{\mathcal{A}}, g^{\mathcal{A}}, \ldots, a^{\mathcal{A}}, b^{\mathcal{A}}, \ldots \rangle,$$

gdzie A jest niepustym zbiorem, $p^{\mathcal{A}}, q^{\mathcal{A}}, \ldots$ są relacjami w zbiorze A, $f^{\mathcal{A}}, g^{\mathcal{A}}, \ldots$ są funkcjami określonymi i przyjmującymi wartości w zbiorze A, a $a^{\mathcal{A}}, b^{\mathcal{A}}, \ldots$ są elementami A. Zakładamy, że jeżeli p jest symbolem o arności $n > 0$, to $p^{\mathcal{A}} \subseteq A^n$, a jeżeli f jest symbolem o arności n, to $f^{\mathcal{A}} : A^n \rightarrow A$.

Wartościowaniem zmiennych (indywiduowych) w strukturze $\mathcal{A}$ nazywamy dowolną funkcję określoną w zbiorze wszystkich zmiennych i przyjmującą wartości w zbiorze A.

Wartościowaniem zmiennych zdaniowych w strukturze $\mathcal{A}$ nazywamy dowolną funkcję określoną w zbiorze wszystkich zmiennych zdaniowych (symboli relacyjnych o arności 0) i przyjmującą wartości w zbiorze $\{\textbf{true}, \textbf{false}\}$.

Niech α będzie wartościowaniem zmiennych w strukturze $\mathcal{A}$. Wartością $t[\alpha]$ termu t przy wartościowaniu α nazywamy element $\alpha(t)$, jeżeli term t jest zmienną, element $t^{\mathcal{A}}$, jeżeli term t jest stałą, oraz element $f^{\mathcal{A}}(t_1[\alpha], \ldots, t_n[\alpha])$, jeżeli $t = t(t_1, \ldots, t_n)$.

Załóżmy, że w formule φ nie występują symbole relacyjne arności 0. Napis

$$\mathcal{A} \models \varphi[\alpha]$$

oznacza, że formuła φ jest spełniona w strukturze $\mathcal{A}$ przy wartościowaniu α. Jeżeli $\varphi = p(t_1, \ldots, t_n)$, to spełnianie φ w strukturze $\mathcal{A}$ przy wartościowaniu α jest równoważne warunkowi $(t_1[\alpha], \ldots, t_n[\alpha]) \in p^{\mathcal{A}}$. Warunek $\mathcal{A} \models \textbf{true}[\alpha]$ jest prawdziwy dla każdej struktury $\mathcal{A}$ i dowolnego wartościowania α, a warunek $\mathcal{A} \models \textbf{false}[\alpha]$ jest fałszywy. Ponadto prawdziwe są następujące równoważności:

$$\mathcal{A} \models \neg\varphi[\alpha] \ \text{ wtedy i tylko wtedy, gdy } \ \mathcal{A} \not\models \varphi[\alpha],$$

$$\mathcal{A} \models (\varphi \vee \psi)[\alpha] \text{ wtedy i tylko wtedy, gdy } \mathcal{A} \models \varphi[\alpha] \text{ lub } \mathcal{A} \models \psi[\alpha],$$

$$\mathcal{A} \models (\varphi \wedge \psi)[\alpha] \text{ wtedy i tylko wtedy, gdy } \mathcal{A} \models \varphi[\alpha] \text{ oraz } \mathcal{A} \models \psi[\alpha],$$

$$\mathcal{A} \models \exists x\ \varphi[\alpha] \text{ wtedy i tylko wtedy, gdy istnieje } \alpha' \text{ takie, że } \mathcal{A} \models \varphi[\alpha'],$$

$$\mathcal{A} \models \forall x\ \varphi[\alpha] \text{ wtedy i tylko wtedy, gdy dla każdego } \alpha' \text{ zachodzi warunek } \mathcal{A} \models \varphi[\alpha']$$

przy czym w dwóch ostatnich równoważnościach rozważamy tylko wartościowania α' przyjmujące te same wartości co wartościowanie α dla wszystkich zmiennych z wyjątkiem najwyżej zmiennej x.

Jeżeli dopuścimy zmienne zdaniowe, to w definicji spełniania pojawi się jeszcze wartościowanie zmiennych zdaniowych β. Wtedy bedziemy pisać

$$\mathcal{A} \models \varphi[\alpha, \beta].$$

Pojęcie to definiujemy podobnie: wystarczy w podanych równoważnościach dopisać symbol β oraz dodać następującą równoważność

$$\mathcal{A} \models p[\alpha, \beta] \text{ wtedy i tylko wtedy, gdy } \beta(p) = \textbf{true}$$

dla dowolnej zmiennej zdaniowej p.

Formuła φ jest spełniona w strukturze $\mathcal{A}$ wtedy i tylko wtedy, gdy jest spełniona w tej strukturze przy dowolnym wartościowaniu. Napis $\mathcal{A} \models \varphi$ będzie oznaczać, że φ jest spełniona w strukturze $\mathcal{A}$. Mamy więc

$$\mathcal{A} \models \varphi \text{ wtedy i tylko wtedy, gdy dla każdego } \alpha \text{ zachodzi warunek } \mathcal{A} \models \varphi[\alpha].$$

Formuła φ jest tautologią, jeżeli jest spełniona w dowolnej strukturze. Fakt, że φ jest tautologią będzie wyrażać wzorem $\models \varphi$. Tak więc

$$\models \varphi \text{ wtedy i tylko wtedy, gdy dla każdej struktury } \mathcal{A} \text{ zachodzi warunek } \mathcal{A} \models \varphi.$$

1.3 Podstawianie

Podstawieniem nazywamy funkcję określoną na skończonym zbiorze zmiennych i przymującą jako wartości termy, która dowolnej zmiennej z dziedziny przyporządkowuje term różny od tej zmiennej. Podstawienie θ będziemy definiować pisząc wzory następującej postaci

$$\theta = \{x_1/t_1, \ldots, x_n/t_n\}.$$

Wzór ten oznacza, że podstawienie θ jest określone dla zmiennych $\{x_1, \ldots, x_n\}$ i dla zmiennej x_i przyjmuje jako wartość term t_i (zakladamy, że $t_i \neq x_i$). Ponadto zakładamy, że każda zmienna x_i w powyższym wzorze występuje przed znakiem / tylko jeden raz.

Wyrażeniem nazywamy formułę bez kwantyfikatorów, term, lub ciąg takich formuł lub termów. Podstawianie działa na wyrażeniach. Jeżeli E jest wyrażeniem i θ jest podstawieniem, to $E\theta$ jest wyrażeniem otrzymanym z E przez wykonanie podstawienia θ. Wykonanie podstawienia polega na przepisywaniu kolejnych znaków wyrażenia z tym, że jednocześnie sprawdzamy, czy przepisywany znak jest jedną ze zmiennych x_i, i jeżeli napotkamy zmienną x_i, to zamiast tej zmiennej piszemy term t_i. Dla termów operację podstawiania można zdefiniować podając następujące równości: $t\theta = t$, jeżeli term t jest stałą lub zmienną nie należącą do dziedziny θ, $t\theta = t_i$, jeżeli t jest zmienną x_i, oraz $t\theta = f(s_1\theta, \ldots, s_n\theta)$, jeżeli $t = f(s_1 \ldots, s_n)$.

Przypuśćmy, że mamy dwa podstawienia

$$\theta = \{x_1/t_1, \ldots, x_n/t_n\} \text{ oraz } \eta = \{y_1/s_1, \ldots, y_m/s_m\}.$$

Złożeniem tych podstawień nazywamy podstawienie

$$\theta\eta = (\{x_1/t_1\eta, \ldots, x_n/t_n\eta, y_1/s_1, \ldots, y_m/s_m\} \setminus \{x_1/x_1, \ldots, x_n/x_n\})$$

$$\setminus\{y_i/s_i : \ y_i = x_j \text{ dla pewnego } j\}.$$

Złożenie podstawień zostało zdefiniowane tak, aby $(E\theta)\eta = E(\theta\eta)$. Zauważmy, że $(\theta\eta)\gamma = \theta(\eta\gamma)$ dla dowolnych podstawień θ, η i γ.

Podstawienie θ nazywamy bardziej ogólnym niż η, jeżeli $\eta = \theta\gamma$ dla pewnego podstawienia γ.

1.4 Unifikatory, algorytm unifikacji

Przypuśćmy że A i B są formułami atomowymi. Podstawienie θ jest unifikatorem formuł A i B (unifikuje A i B), jeżeli $A\theta = B\theta$. Formuły, dla których istneje unifikator nazywamy unifikowalnymi. Unifikator θ formuł A i B nazywamy najbardziej ogólnym, jeżeli dla dowolnego unifikatora η formuł A i B istnieje podstawienie γ takie, że $\eta = \theta\gamma$.

Twierdzenie 1.1 *Istnieje algorytm, który dla danych dwóch formuł atomowych zwraca najogólniejszy unifikator tych formuł w przypadku, gdy te formuły są unifikowalne, a w przeciwnym razie stwierdza nieunifikowalność tych formuł.*

Dowód. Niech D będzie zbiorem par termów. Unifikację zbioru D można zbadać wykonując następujący algorytm:

$\theta :=$ **podstawienie identycznościowe;**

$X := D$;

dopóki $X \neq \emptyset$ **wykonuj:**

wybierz $(t, s) \in X$**;**

usuń (t, s) **z** X**;**

jeżeli t **i** s **zaczynają się różnymi symbolami i są to symbole funkcyjne lub stałe, to przerwij działanie algorytmu i przekaż, że** D **nie ma unifikatora;**

jeżeli t **i** s **zaczynają się identycznymi symbolami i są to symbole funkcyjne lub stałe, oraz** $t = f(t_1, \ldots, t_n)$ **i** $s = f(s_1, \ldots, s_n)$**, to dodaj do** X **wszystkie pary** (t_i, s_i) **dla** $i = 1, \ldots, n$**;**

jeżeli t **jest zmienną i występuje w** $s \neq t$**, to przerwij działanie algorytmu i przekaż, że** D **nie ma unifikatora (i analogicznie, jeżeli** s **jest zmienną);**

jeżeli t **jest zmienną i** t **nie występuje w** s**, to wykonaj podstawienia** $\theta := \theta[t/s]$ **i** $X := X[t/s]$ **(i analogicznie, jeżeli** s **jest zmienną);**

zwróć θ**.**

□