Andrzej Wiśniewski

Logika I

Materiały do wykładu dla studentów kognitywistyki

Wykład 11a. Składnia języka Klasycznego Rachunku Predykatów. Języki pierwszego rzędu.

Logika Klasyczna obejmuje dwie teorie: Klasyczny Rachunek Zdań i Klasyczny Rachunek Predykatów (dalej krótko: *KRP*). Jak zobaczymy, *KRP* jest w pewnym sensie „nadbudowany” nad Klasycznym Rachunkiem Zdań. Jednakże język *KRP* różni się istotnie od języka *KRZ*.

Zacznijmy od charakterystyki zbioru znaków (tj. „alfabetu”) języka *KRP*.

Definicja 11.1. *Następujące symbole nazywamy* znakami *języka KRP*:

$\neg \rightarrow \wedge \vee \leftrightarrow \forall \exists$	(*stałe logiczne*)
$x_1 \quad x_2 \quad x_3 \ldots$	(*zmienne indywiduowe*)
$P_1^1 \quad P_2^1 \quad P_3^1 \ldots$	(*predykaty jednoargumentowe*)
$P_1^2 \quad P_2^2 \quad P_3^2 \ldots$	(*predykaty dwuargumentowe*)
.....	
$P_1^n \quad P_2^n \quad P_3^n$	(*predykaty n-argumentowe*)
.....	
$a_1 \quad a_2 \quad a_3 \ldots$	(*nazwy indywidualne*)
$F_1^1 \quad F_2^1 \quad F_3^1 \ldots$	(*symbole funkcyjne jednoargumentowe*)
$F_1^2 \quad F_2^2 \quad F_3^2 \ldots$	(*symbole funkcyjne dwuargumentowe*)
.....	
$F_1^n \quad F_2^n \quad F_3^n \ldots$	(*symbole funkcyjne n-argumentowe*)
.....	
( ) ,	(*znaki techniczne: nawiasy i przecinek*)

Komentarze: Zmiennych indywiduowych oraz nazw indywidualnych jest przeliczalnie nieskończenie wiele. Podobnie, dla każdego $n \in \mathbf{N}$, mamy przeliczalnie nieskończenie wiele predykatów *n*-argumentowych oraz symboli funkcyjnych *n*-argumentowych.

Zamiast „predykat *n*-argumentowy” mówi się czasami „predykat *n*-członowy”.

O ile zmienne zdaniowe w *KRZ* przebiegają zdania w sensie logicznym, wartościami zmiennych indywiduowych *KRP* mogą być obiekty dowolnego rodzaju – o ile tylko w danym zastosowaniu obiekty te decydujemy się potraktować jako indywidua.

Patrząc z semantycznego punktu widzenia, odniesieniami przedmiotowymi predykatów jednoargumentowych są własności/ zbiory, predykatów *n*-członowych ($n > 1$) – relacje *n*-członowe, a *n*-argumentowych symboli funkcyjnych – funkcje *n*-argumentowe. Jednocześnie znaki języka *KRP* reprezentują wyrażenia języka naturalnego odpowiednich kategorii syntaktycznych.

Wyrażeniem języka *KRP* nazywamy dowolny skończony ciąg znaków tego języka.

Mamy dwie kategorie wyrażeń sensownych: formuły nazwowe, zwane termami, oraz formuły zdaniowe.

Definicja 11.2. (termy języka *KRP*)

(i) *Każda zmienna inywiduowa jest termem języka KRP*;

(ii) *każda stała indywidualna jest termem języka KRP*;

(iii) *jeżeli* $\tau_1, ..., \tau_n$ *są termami języka KRP, a* F_k^n jest *n-argumentowym symbolem funkcyjnym, to wyrażenie o postaci* $F_k^n(\tau_1, ..., \tau_n)$ *jest termem języka KRP*;

(iv) *nie ma żadnych innych termów języka KRP poza zmiennymi indywiduowymi, stałymi indywidualnymi oraz tymi wyrażeniami, które można utworzyć zgodnie z regułą* (iii).

Termy, w których nie występują zmienne indywiduowe określamy mianem termów zamkniętych. Są one odpowiednikami nazw jednostkowych (prostych i złożonych).

Najprostsze formuły zdaniowe języka *KRP* noszą miano formuł atomowych.

Definicja 11.3. (formuły atomowe języka *KRP*)

Formuły atomowe *języka KRP to wyrażenia mające postać*:

$$P_k^n(\tau_1, ..., \tau_n)$$

gdzie P_k^n *jest predykatem n-argumentowym, a* $\tau_1, ..., \tau_n$ *są termami języka KRP*.

Formuły atomowe są „cegiełkami”, z których – za pomocą stałych logicznych – budujemy złożone formuły zdaniowe.

Notacja: Liter *A*, *B*, *C*, *D*, ..., ewentualnie z indeksami, będziemy dalej używali jako metajęzykowych zmiennych odnoszących się do formuł zdaniowych języka *KRP*. Symbole $\tau_1, \tau_2, ..., \tau_n$ są metajęzykowymi zmiennymi reprezentującymi termy języka *KRP*. Napisy: x_i oraz a_i należą do metajęzyka; są to metajęzykowe zmienne reprezentujące odpowiednio: zmienną indywiduową oraz stałą indywidualną.

Definicja 11.4. (formuły zdaniowe języka *KRP*)

(i) *Każda formuła atomowa języka KRP jest formułą zdaniową języka KRP*;

(ii) *jeżeli A jest formułą zdaniową języka KRP, to wyrażenie mające postać* $\neg A$ *jest formułą zdaniową języka KRP*;

(iii) *jeżeli A, B są formułami zdaniowymi języka KRP, to wyrażenia mające postać*: $(A \rightarrow B)$, $(A \wedge B)$, $(A \vee B)$, $(A \leftrightarrow B)$ *są formułami zdaniowymi języka KRP*;

(iv) *jeżeli A jest formułą zdaniową języka KRP, a* x_i *jest zmienną indywiduową, to wyrażenia mające postać*: $\forall x_i A$, $\exists x_i A$ *są formułami zdaniowymi języka KRP*;

(v) *nie ma innych formuł zdaniowych języka KRP poza tymi, które można utworzyć wedle reguł* (i) – (iv).

Terminologia: Dalej zamiast „formuła zdaniowa języka *KRP*” będę mówił krótko „formuła zdaniowa”. Podobnie zamiast „zmienna indywiduowa” będę mówił krótko „zmienna”.

Formuły zdaniowe dzielimy na *zdania* i *funkcje zdaniowe*. Aby wprowadzić to rozróżnienie, musimy najpierw zdefiniować kilka pojęć pomocniczych.

Definicja 11.5. (zasięg kwantyfikatora)

Formułę zdaniową A występującą w formule zdaniowej o postaci $\forall x_i\, A$ *lub o postaci* $\exists x_i\, A$ *nazywamy zasięgiem odpowiedniego kwantyfikatora.*

Definicja 11.6 (zmienna związana na danym miejscu w formule zdaniowej)

Zmienna x_i *występująca na danym miejscu w formule zdaniowej A jest na tym miejscu związana, jeżeli występuje ona bezpośrednio po kwantyfikatorze lub też znajduje się w zasięgu jakiegoś kwantyfikatora, bezpośrednio po którym występuje zmienna* x_i.

Przykład 11.1. Za pomocą cieniowania zaznaczone zostały zmienne, które - w podanych formułach zdaniowych - są związane na miejscach ich występowania:

(1) $(\exists x_1\, P_1^2(x_1, x_2) \rightarrow \forall x_2\, P_1^2(x_1, x_2))$

(2) $(\exists x_1(P_1^2(x_1, x_2) \rightarrow \forall x_2\, P_1^2(x_1, x_2)))$

(3) $(\exists x_1\ \forall x_2\, (P_1^2(x_1, x_2) \rightarrow P_1^2(x_1, x_2)))$

Definicja 11.7. (zmienna wolna na danym miejscu w formule zdaniowej)
Jeżeli zmienna x_i ,występująca na danym miejscu w formule zdaniowej A, nie jest na tym miejscu związana, to mówimy, że jest ona na tym miejscu wolna w formule A.

Niezacieniowane zmienne z Przykładu 11.1 to zmienne wolne na miejscach ich występowania. Zauważmy, że jedna i ta sama zmienna na jednym miejscu w formule może być związana, a na innym – wolna (por. formuły (1) i (2)).

Definicja 11.8. (zmienne wolne i związane w formule zdaniowej)
Zmienna x_i , występująca w formule zdaniowej A, jest wolna w *A wtw x_i jest wolna w A na przynajmniej jednym miejscu. Zmienna x_i , występująca w formule zdaniowej A, jest* związana w *A wtw x_i jest związana na każdym miejscu w A.*

Komentarz: Zmienna wolna w formule zdaniowej to zmienna, która „gdzieś” (a więc niekoniecznie „wszędzie”) jest w niej wolna.

Definicja 11.9. (zdania i funkcje zdaniowe języka *KRP*)
Formuły zdaniowe języka KRP nie zawierające żadnych zmiennych wolnych nazywamy zdaniami *języka KRP. Formuły zdaniowe języka KRP nie będące zdaniami tego języka nazywamy* funkcjami zdaniowymi *języka KRP*.

Formuły (1) i (2) z Przykładu 11.1 są funkcjami zdaniowymi języka *KRP*, natomiast formuła (3) jest zdaniem tego języka.

Komentarz: Semantycznie rzecz biorąc, funkcje zdaniowe wyrażają otwarte warunki, natomiast zdania stwierdzają/przedstawiają stany rzeczy.

Predykat identyczności

Syntaktycznie rzecz biorąc, znak równości/ identyczności = jest predykatem dwuargumentowym. Ponieważ postać graficzna znaku jest nieistotna, możemy przyjąć, że równość/ identyczność jest reprezentowana przez jakiś predykat dwuargumentowy, np. P_1^2. Gdy chcemy budować Klasyczny Rachunek Predykatów z Identycznością (o czym dalej), wprowadzamy jednak predykat = jako stała logiczną. Składnia języka *KRP* z Identycznością (dalej krótko: $KRP_=$) różni się od składni języka *KRP* tylko tym, że mamy formuły atomowe o dwóch postaciach:

$$\tau_1 = \tau_2$$
$$P_k^n(\tau_1, ..., \tau_n)$$

gdzie $\tau_1, \tau_2, ..., \tau_n$ są termami.

Podobnie jak w przypadku języka *KRZ*, podana tu charakterystyka języka *KRP* różni się od przedstawionej na wykładzie z „Wprowadzenia do logiki” sposobem wprowadzenia nawiasów.

Podając przykłady formuł zdaniowych, wygodnie jest przyjąć pewne konwencje upraszczające zapis graficzny.

Reguły opuszczania nawiasów są takie same jak w przypadku języka *KRZ* (zob. wykład 4).

Zamiast x_1, x_2, x_3, x_4 piszemy odpowiednio: x, y, u, z.

Zamiast a_1, a_2, a_3, a_4 piszemy odpowiednio: a, b, c, d.

Litery P, Q, R będą czasami używane jako symbole predykatów; liczba argumentów tych predykatów będzie zawsze wyznaczona przez kontekst.

Liter f, g, h będziemy czasami używali w charakterze symboli funkcyjnych; podobnie jak poprzednio, liczba argumentów będzie wyznaczona przez kontekst.

W języku *KRP* kwantyfikatory wiążą zmienne indywiduowe; o ile w języku tym mamy zarówno zmienne, jak i stałe (indywidualne) odnoszące się - semantycznie rzecz biorąc – do indywiduów, to nie występują w nim zmienne odnoszące się do obiektów tych samych kategorii ontologicznych (tj. własności i relacji), co predykaty. W związku z tym kwantyfikacja w języku *KRP* „dotyczy" wyłącznie indywiduów, a nie własności czy relacji.

Język *KRP* jest *językiem pierwszego rzędu*, co więcej - najobszerniejszym takim językiem.

Definicja 11.10. *Każdy podzbiór zbioru znaków języka KRP zawierający w sobie wszystkie stałe logiczne, wszystkie zmienne indywiduowe, przynajmniej jeden predykat, oba nawiasy i ewentualnie (a więc niekoniecznie) jeszcze pewną ilość innych znaków (takich jak nazwy indywidualne, symbole funkcyjne, przecinek) nazywamy* językiem pierwszego rzędu*.*

Definicja 11.11. *Predykaty, nazwy indywidualne i symbole funkcyjne danego języka pierwszego rzędu nazywamy* stałymi pozalogicznymi *tego języka.*

Komentarz: Utożsamienie języka ze zbiorem znaków może się wydawać nieintuicyjne; bardziej intuicyjne byłoby utożsamienie języka ze zbiorem wyrażeń sensownych/ poprawnie zbudowanych. Tym niemniej wyjście proponowane przez Definicję 11.10 pozwala zachować jednoznaczność oraz upraszcza prezentacje składni poszczególnych języków. Składnia ta jest po prostu taka sama jak w przypadku języka *KRP*: pojęcia termu i formuły zdaniowej (oraz pojęcia pochodne) definiujemy analogicznie jak w przypadku języka *KRP*.

Notacja: Wprowadzone wcześniej symbole metajęzykowe będziemy używać w stosunku do języków pierwszego rzędu w taki sam sposób, jak czyniliśmy to w przypadku języka *KRP*. To samo dotyczy przyjętych konwencji upraszczania zapisu formuł zdaniowych.

Przykład 11.2. Język arytmetyki Peano

Ponieważ wszystkie zmienne indywiduowe i wszystkie stałe logiczne występują w każdym języku pierwszego rzędu, charakterystyka takiego języka sprowadza się do podania listy jego stałych pozalogicznych. [Oba nawiasy występują zawsze; przecinek jest potrzebny wówczas, gdy mamy predykaty i/lub symbole funkcyjne więcej niż jednoargumentowe.]

Oto lista stałych pozalogicznych języka arytmetyki Peano (tj. arytmetyki liczb naturalnych):

=	(*predykat dwuczłonowy:* identyczność)
0	(*nazwa indywidualna:* zero)
S	(*symbol funkcyjny jednoargumentowy:* funkcja następnika)
+ ·	(*symbole funkcyjne dwuargumentowe:* znaki/ funkcje dodawania i mnożenia)

To, że stałe pozalogiczne zapisaliśmy nie używając konwencji przyjętych w *KRP*, jest w zasadzie obojętne. W zasadzie, albowiem powyższy zapis niesie informacje, z których korzystamy na poziomie *semantyki:* w nawiasach kolorem zielonomodrym wskazaliśmy standardowe interpretacje wprowadzonych stałych pozalogicznych. Na poziomie *składni* wszystko funkcjonuje analogicznie, jak w przypadku języka *KRP*:

Termy języka arytmetyki Peano definiujemy następująco:

(i) *Każda zmienna inywiduowa jest termem języka arytmetyki Peano;*

(ii) **0** *jest termem języka arytmetyki Peano*;

(iii) *jeżeli* τ *jest termem języka arytmetyki Peano, to wyrażenie o postaci* **S**(τ) *jest termem języka arytmetyki Peano*;

(iv) *jeżeli* τ_1, τ_2 *są termami języka arytmetyki Peano*, *to wyrażenia mające postać*: (τ_1**+**τ_2), (τ_1**·**τ_2) *są termami języka arytmetyki Peano*;

(v) *nie ma żadnych innych termów języka arytmetyki Peano poza zmiennymi indywiduowymi, stałą indywidualną* **0** *oraz tymi wyrażeniami, które można utworzyć zgodnie z regułami* (iii) *i* (iv).

Formuły atomowe języka arytmetyki Peano mają postać:

$$\tau_1 = \tau_2$$

gdzie τ_1, τ_2 są termami tego języka.

Pozostałe pojęcia syntaktyczne (formuły zdaniowej, zmiennej wolnej i związanej *etc*.) definiujemy dokładnie tak samo jak w przypadku języka *KRP*.

Literatura:

Tadeusz Batóg, *Podstawy logiki*, Wydawnictwo Naukowe UAM, Poznań 1996.

Notacja przyjęta na tym wykładzie różni się nieco od stosowanej w powyższym podręczniku.