

# Wstęp do logiki

## Klasyczny Rachunek Predykatów II

Język KRP należy do pewnej klasy języków, zwanych językami pierwszego rzędu.

Język  $J$  nazywamy **językiem pierwszego rzędu**, jeśli spełnia on następujące dwa warunki:

- Słownik jego zawiera: (a) nieskończenie wiele zmiennych indywidualnych (nazwowych); (b) przynajmniej jeden symbol relacyjny, tj. predykat; (c) skończoną liczbę spójników; (d) kwantyfikatory wiążące wyłącznie zmienne indywidualne; (e) znaki techniczne tj. nawiasy. (Dodatkowo mogą wystąpić stałe indywidualne, będące nazwami jednostkowymi określonych przedmiotów oraz symbole funkcyjne.)
- Wyrażeniami sensownymi są takie ciągi znaków ze słownika języka  $J$ , które są schematami poprawnie zbudowanych zdań lub nazw (jakiegoś języka etnicznego).

**Dygresja.** Wprowadzając zmienne reprezentujące bardziej złożone przedmioty – takie, jak relacje i funkcje – i zezwalając na kwantyfikację tych zmiennych, dostajemy języki coraz wyższych rzędów. ■

**Dygresja.** Wielu logików uznaje tezę, zwaną *Tezą Hilberta*, że logika pierwszego rzędu jest podstawowa w tym sensie, iż:

- Wszystkie logiczne założenia matematyki dadzą się sformułować w języku pierwszego rzędu.
- Nieformalne pojęcie dowodu praktyki matematycznej jest równoważne formalnemu pojęciu dowodu w logice pierwszego rzędu. Uzasadnieniem tego jest twierdzenie Gödla o pełności logiki pierwszego rzędu (w myśl niego – mówiąc swobodnie – pojęcie dowodliwości jest równoważne pojęciu wynikania logicznego). ■

DEF.1 (Słownik). W skład słownika języka KRP wchodzi następujące grupy symboli:

zmienne indywiduowe (nazwowe):  $x_1, x_2, \dots$

stałe indywiduowe (nazwowe):  $a_1, a_2, \dots$

symbole funkcyjne:  $f_1^1, f_2^1, \dots, f_1^2, f_2^2, \dots, f_k^n, \dots$

symbole relacyjne (predykaty):  $P_1^1, P_2^1, \dots, P_1^2, P_2^2, \dots, P_k^n, \dots$

(górny indeks określa liczbę argumentów, a dolny służy do odróżnienia poszczególnych symboli funkcyjnych i relacyjnych)

spójniki:  $\sim, \wedge, \vee, \rightarrow, \equiv$

kwantyfikatory:  $\forall$  (generalny),  $\exists$  (egzystencjalny)

kwantyfikator generalny czytamy: dla każdego, dla dowolnego, wszystkie, a

kwantyfikator egzystencjalny czytamy: dla pewnego; istnieje takie, że; niektóre.

znaki techniczne (nawiasy):  $), ($

**Terminologia:** Spójniki i kwantyfikatory tworzą zbiór tzw. **stałych logicznych**. Stałe indywidualowe, symbole funkcyjne i symbole relacyjne tworzą zbiór tzw. **stałych pozallogicznych**.

DEF.2 (Wyrażenie). Każdy skończony ciąg symboli ze słownika języka KRP nazywamy **wyrażeniem** tego języka.

Mamy tu dwie kategorie **wyrażeń sensownych**:

- **formuły nazwowe**, zwane **termami**;
- **formuły zdaniowe**;

Pojęcia te definiujemy w następujący sposób:

DEF.3 (Term).

- (1) Wszystkie zmienne i stałe indywidualowe są termami języka KRP; nazywamy je **termami prostymi**.
- (2) Jeżeli  $f_k^n$  jest  $n$ -argumentowym symbolem funkcyjnym, zaś  $t_1, \dots, t_n$  są dowolnymi termami, to wyrażenie postaci  $f_k^n(t_1, \dots, t_n)$  jest również termem (tzw. **termem złożonym**).
- (3) Nie ma innych termów poza wymienionymi w punkcie (1) i takimi, które powstają dzięki zastosowaniu reguły (2).

Termy bez zmiennych to **termy domknięte**, czyli **nazwy**.

DEF.4 (Formuła zdaniowa atomowa). **Formułą atomową** języka KRP nazywamy dowolne wyrażenie postaci  $P_k^n(t_1, \dots, t_n)$ , gdzie  $P_k^n$  jest  $n$ -argumentowym predykatem, zaś  $t_1, \dots, t_n$  są dowolnymi termami.

Formuły atomowe są „cegiełkami”, z których za pomocą stałych logicznych budujemy bardziej złożone formuły zdaniowe. Dalej, zamiast mówić „formuła zdaniowa” będę mówił po prostu „formuła”.

DEF.5 (Formuła zdaniowa).

(1) Wszystkie formuły atomowe są formułami języka KRP.

(2) Jeżeli  $A, B$  są dowolnymi formułami języka KRP, to wyrażenia postaci:

$$\sim(A), (A) \wedge (B), (A) \vee (B), (A) \rightarrow (B), (A) \equiv (B), \forall x_i(A) \text{ i } \exists x_i(A)$$

są również formułami języka KRP.

(3) Nie ma innych formuł języka KRP poza formułami atomowymi i takimi, które powstają dzięki zastosowaniu reguły (2).

Jeżeli kwantyfikator znajduje się w jakiejś formule, to zawsze bezpośrednio za nim występuje zmienna indywidualowa. Mówimy wówczas, że kwantyfikator wiąże tę zmienną.

DEF. 6. Jeżeli formuła ma postać  $\forall x_i(A)$  lub  $\exists x_i(A)$ , to mówimy, że odpowiedni kwantyfikator wiąże zmienną  $x_i$ .

Umowy notacyjne:

- Dużych liter z początku alfabetu  $A, B, C, D$  (ewentualnie z indeksami) używam jako **zmiennych metajęzykowych** oznaczających dowolne formuły języka KRP;
- Dużych liter z końca alfabetu  $X, Y, Z$  (ewentualnie z indeksami) używam jako **zmiennych metajęzykowych** oznaczających dowolne zbiory formuł języka KRP;
- Liter  $t, s$  (ewentualnie z indeksami) używam jako **zmiennych metajęzykowych** oznaczających dowolne termy języka KRP. Podobnie,  $x_i$  oraz  $a_i$  są **zmiennymi metajęzykowymi**, których wartościami są, odpowiednio, zmienne indywidualowe i stałe indywidualowe.



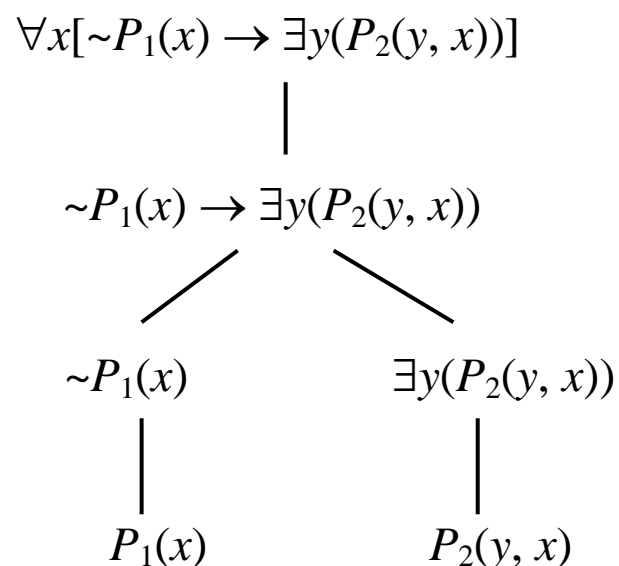
- Dla uproszczenia notacji pisać będziemy:

$x$  zamiast  $x_1$ ,  $y$  zamiast  $x_2$ ,  $z$  zamiast  $x_3$  itd.;  $a$  zamiast  $a_1$ ,  $b$  zamiast  $a_2$ ,  $c$  zamiast  $a_3$  itd.

Pomijać będziemy też górny indeks określający liczbę argumentów danego symbolu funkcyjnego lub relacyjnego, jeżeli występuje on wraz ze swymi argumentami.

- Dla większej przejrzystości używać będziemy – oprócz nawiasów okrągłych – także nawisów kwadratowych i klamrowych.
- Podobnie jak w przypadku rachunku zdań przyjmujemy, że  
spójnik  $\sim$  wiąże najsilniej;  
spójniki  $\wedge$  i  $\vee$  wiążą silniej niż spójniki  $\rightarrow$  i  $\equiv$ .
- Aby jeszcze bardziej uprościć symbolikę, predykatów na ogół nie pozostawia się w wersji oryginalnej, lecz zastępuje symbolami  $P, Q, R, S, \dots$ . Liczbę argumentów wyznacza kontekst. Jako symboli funkcyjnych będziemy używać też małych liter  $f, g, h$ . ■

**Przykład:** Zgodnie z DEF.5 każda formuła jest formułą atomową bądź powstaje z formuł przez poprzedzenie negacją, połączenie spójnikami koniunkcji, alternatywy, implikacji lub równoważności, bądź poprzedzenie kwantyfikatorem wiążącym dowolną zmienną. Formułą języka KRP jest m.in.  $\forall x[\sim P_1(x) \rightarrow \exists y(P_2(y, x))]$ . Sposób jej budowy ilustruje diagram:



**Formułami nie są:**  $P_1^3(x, y)$ ,  $P_1^2(x, P_1^1(y))$ ,  $\sim x$ ,  $P_1^2(x, y) \rightarrow \exists z$ . ■

DEF.7 (Zasięg kwantyfikatora) Formułę  $A$  w formule  $\forall x_i(A)$  lub  $\exists x_i(A)$  nazywamy **zasięgiem** odpowiedniego kwantyfikatora.

**Notacja:** Czasem, kiedy wiadomo, jaki jest zasięg kwantyfikatora, nawiasów się nie używa; zamiast więc  $\forall x(\exists y(P_1(x, y)))$  piszemy:  $\forall x\exists yP_1(x, y)$ .

**Przykłady:**

$$\forall x [\underbrace{\sim P_1(x) \rightarrow \exists y P_2(y, x)}_{\text{zasięg } \exists}],$$

zasięg  $\forall$

$$\forall x \underbrace{\exists y [P_1(x) \rightarrow P_2(x, y)]}_{\text{zasięg } \exists}$$

zasięg  $\forall$



DEF.8 (Zmienna związana). Zmienna  $x_i$  występująca na danym miejscu w formule  $A$  jest w tym miejscu **związana** wtw albo występuje ona w tej formule bezpośrednio po którymś kwantyfikatorze, albo znajduje się w zasięgu jakiegoś kwantyfikatora wiążącego tę zmienną. Zmienna  $x_i$  występująca w formule  $A$  jest w niej związana wtw  $x_i$  jest związana na każdym miejscu w formule  $A$ .

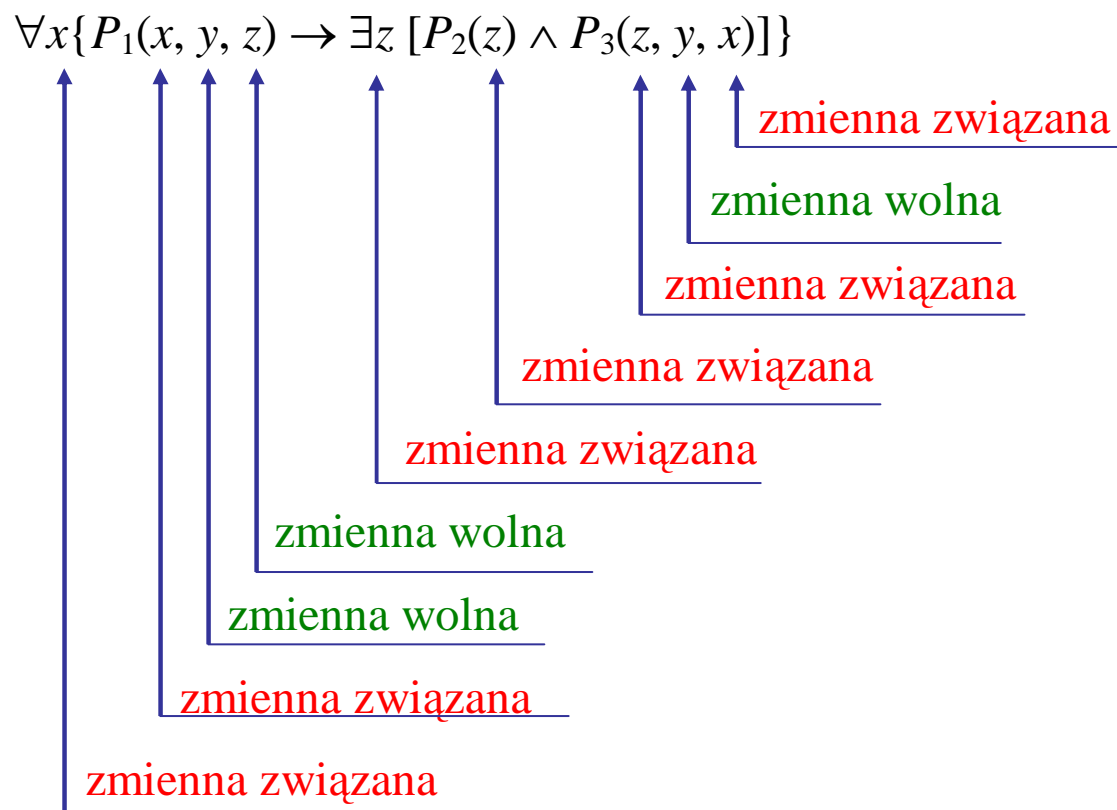
DEF. 9 (Zmienna wolna). Jeżeli zmienna  $x_i$  występująca na danym miejscu w formule  $A$ , nie jest na tym miejscu związana, to mówimy, że jest ona na tym miejscu **wolna** w  $A$ . Mówimy, że zmienna  $x_i$  jest wolna w formule  $A$  wtw jest ona wolna na przynajmniej jednym miejscu w formule  $A$ .

Przykłady: **z.w.** – zmienna wolna, **z.z.** – zmienna związana.

$P_1(x),$	$\forall x[P_1(x) \rightarrow P_2(x)],$	$P_1(x) \rightarrow \forall xP_2(x),$	$\forall x[P_1(x) \rightarrow \forall xP_2(x)],$
<b>z.w.</b>	<b>z.z.</b>	<b>z.z.</b>	<b>z.z.</b>

$P_1(x) \rightarrow \forall xP_2(x),$	$\forall x[P_1(x) \rightarrow \forall xP_2(x)],$
<b>z.w.</b>	<b>z.z.</b>



Widzimy, że ta sama zmienna może być w formule w jednym miejscu związana, w innym zaś – wolna. ■

DEF.10 (Zdanie, funkcja zdaniowa). Formuły bez zmiennych wolnych nazywamy **zdaniami**. Pozostałe formuły nazywamy **funkcjami zdaniowymi**.

Tak więc, W języku KRP mamy następujące rodzaje formuł:

