

# LOGICZNE PODSTAWY INFORMATYKI – LPI

## Ćwiczenia 1

14.10.2012

### Zadanie 1.

Otworzyć w SWI-Prolog plik o nazwie **rodzina.pl** zawierający dane o członkach pewnej rodziny. Następnie:

- a) narysować graf zależności pomiędzy osobami,
- b) zadać 5 różnych pytań do bazy danych,
- c) zdefiniować podstawowe relacje pokrewieństwa takie, jak **ojciec**, **matka**, **rodzeństwo**, **dziadek**, **wuj**, **kuzyn**, **przodek** i dopisać do pliku rodzina.pl,  
matka(X) /\* X jest matką \*/  
rodzeństwo(X,Y) /\* X i Y są rodzeństwem \*/  
itd.
- d) zadać 5 różnych pytań dotyczących związków rodzinnych.

### Zadanie 2.

Informacje o rodzinie poszerzyć o dane dotyczące wieku. Następnie:

- a) zdefiniować relacje **starszy** oraz **młodszy**,  
starszy(X,Y) /\* X jest starszy od Y \*/  
młodszy(X,Y) /\* X jest młodszy od Y \*/
- b) zdefiniować relacje **uczeń** oraz **pracownik**,  
uczen(Z) /\* Z jest uczniem \*/ (Z jest uczniem, jeśli jest w wieku między  
6 a 19 lat)
- c) sformułować odpowiednie pytania do poszerzonej bazy danych.

### Zadanie 3.

Zdefiniować relację **większy(X, Y, Z)**, której dwoma pierwszymi elementami są dwie liczby, a trzecim elementem jest większa z nich.

### Zadanie 4.

Sprawdź, czy poniższe cele zostaną spełnione i (ewentualnie) które zmienne zostaną jak ukonkretnione:

odcinek( punkt( 1, 2), punkt( A)) = odcinek( B, punkt( 1, 2)).  
odcinek( punkt( 1, 2), punkt( A,C)) = odcinek( B, punkt( 1, 2)).  
odcinek( punkt( 1, 2), punkt( A,B)) = odcinek( B, punkt( 1, C)).  
punkt(X,Y,Z)=punkt(X1,Y1,Z1).

$\text{lot}(A, \text{londyn}) = \text{lot}(\text{londyn}, \text{paryz}).$   
 $'\text{student}' = \text{student}.$   
 $'\text{Student}' = \text{student}.$   
 $\text{Student} = '\text{Student}''.$   
 $''\text{Student}'' = '\text{Student}'$   
 $f(X, X) = f(a, b).$   
 $f(X, a(b, c)) = f(Z, a(Z, c)).$   
 $1 + 2 = 3.$   
 $1 + 2 = 1 + 2.$   
 $1 + 2 = = 2 + 1.$   
 $1 + 2 = : = 3.$   
 $1 + 2 = \backslash = 3.$   
 $X = X.$   
 $X = = X.$   
 $X = Y.$   
 $X = = Y.$

### Zadanie 5.

Sprawdzić działanie procedur działających na listach:

**is\_list** (L) - sprawdza, czy L jest listą

**append** (L1, L2, L3) – łączy listy L1 i L2 w listę L3

Sprawdzić np.  $\text{append}([b, c, [s, a], a], [a], X).$   
 $\text{append}([a], [b], [a, b]).$   
 $\text{append}(L1, L2, [b, c, [s, a]]).$

**member**(E, L) – sprawdza, czy element E należy do listy L np.  $\text{member}(5, [3, 6, 5, 7, 6])$  lub wypisuje elementy listy L np.  $\text{member}(X, [2, 3, 4, 9]).$  Sprawdzić:

$\text{member}(a, [b, c, [s, a], a]).$   
 $\text{member}(a, [b, c, [s, a]]).$   
 $\text{member}([s, a], [b, c, [s, a]]).$   
 $\text{member}(X, [a, b, c]).$   
 $\text{member}(a, X).$

**memberchk**(E, L) - równoważny predykatowi member, ale podaje tylko jedno rozwiązanie

**nextto**(X, Y, L) – predykat spełniony, gdy Y występuje bezpośrednio po X, sprawdzić np.

$\text{nextto}(X, Y, [2, 3, 4, 5]).$   
 $\text{nextto}(3, Y, [2, 3, 4, 5]).$   
 $\text{nextto}(X, 4, [2, 3, 4, 5]).$

**delete**(L1, E, L2) – z listy L1 usuwa wszystkie wystąpienia elementu E, wynik uzgadnia z listą L2

**select**((E, L, R) – z listy L wybiera element, który daje się uzgodnić z E. Lista R jest uzgadniana z listą, która powstaje z L po usunięciu wybranego elementu.

**nth0**(I,L,E) – predykat spełniony, jeśli element listy L o numerze I daje się uzgodnić z elementem E

**nth1**(I,L,E) – predykat podobny do nth0. Sprawdzić różnicę!

**last**(L,E) – ostatni element listy L

**reverse**(L1,L2) – odwraca porządek elementów listy L1 i unifikuje rezultat z listą L2

**permutation**(L1,L2) – lista L1 jest permutacją listy L2

**flatten**(L1,L2) – przekształca listę L1 w listę l2, w której każda lista składowa zostaje zastąpiona przez swoje elementy, np. `flatten([a,[b,[c,d],e,f]],X)`.

**sumlist**(L,S) – suma listy liczbowej L

**numlist**(M,N,L) – jeśli M,N są liczbami całkowitymi takimi, że  $M < N$ , to L zostanie zunifikowana z listą  $[M, M+1, \dots, N]$

**length**(L,I) – liczba elementów listy L

### Zadanie 6.

Sprawdź, czy poniższe cele zostaną spełnione i (ewentualnie) które zmienne zostaną jak uokretnione:

$[1,2,3,4]=[A|B]$ .

$[A,B]=[A|B]$ .

$[1,[A],2]=[1,0,2]$ .

$[1,2,3]=[1|2,3]$ .

$[1,2,3]=[1,2|3]$ .

$[[A],B,C]=[[a,b,c],[d,e,f],1]$ .

$[W,Z]=[1,2]$ .

$[W,Z]=[1,[2]]$ .

$[W,Z]=[1|2]$ .

$[W,Z]=[1|[2]]$ .