

PODSTAWY PROGRAMOWANIA DEKLARATYWNEGO

PROLOG

Ćwiczenia 10

Zadanie 1.

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

```
[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]].
```

Zadanie 2.

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.

```
append([b,c,[s,a],a],[a],X).
append([a],[b],[a,b]).
append(L1,L2,[b,c,[s,a]]).
```

member(E,L) – sprawdza, czy element E należy do listy L
Sprawdzić np.

```
member(a,[b,c,[s,a],a]).
member(a,[b,c,[s,a]]).
member([s,a],[b,c,[s,a]]).
member(X,[a,b,c]).
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.

```
nextto(X,Y,[a,c,d,r]).
nextto(w,Y,[q,w,e,r]).
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 3. (będzie sprawdzone na kolokwium)

Korzystając z online manual (SWI Prolog help) wyjaśnić znaczenie predykatów **sort/2**, **msort/2**.

Zadanie 4. (będzie sprawdzone na kolokwium)

Czy można użyć **select/3** do dodawania elementu do listy? Jeśli tak, to w jaki sposób?

Zadanie 5. (1p.)

Zdefiniować predykat **max/2** znajdujący największą wartość w liście liczbowej.

?-max([1,4,2,7,3,0],B).

B=7.

Zadanie 6. (1p.)

Zdefiniować predykat **znajdz/3**, którego działanie polega na znalezieniu elementu listy o podanym numerze.

?-znajdz([1,9,7,2,5,4,0,5],4,M).

M=2.

Zadanie 7. (1p.)

Zdefiniować predykat **początek/2** sprawdzający, czy podana lista stanowi początek innej listy.

```
?-początek([1,3], [1,3,2,4,3]).  
true.  
?-początek([3,1], [1,3,2,4,3]).  
false.
```

Zadanie 8. (1p.)

Zdefiniować predykat **ostatni/2** znajdujący ostatni element listy.

```
?-ostatni([2,3,2,4,3,2],O).  
O=2.  
?-ostatni([1,2,1,4,3],6).  
false.
```

Zadanie 9. (1p.)

Zdefiniować predykat **nty/3** znajdujący pozycję danego elementu w liście.

```
?-nty(p,[a,w,q,e,p,w,r],X).  
X=5.
```

Zadanie 10. (1p.)

Zdefiniować predykat **rosnacy/1** który sprawdza, czy kolejne elementy listy L tworzą ciąg rosnący.

```
?-rosnacy([3,6,7,12,29]).  
true.  
?-rosnacy([3,2,7,12,2]).  
false.
```

Uwaga:

Wykonane zadania 5-10 należy przekazać do **13.05.2015, 23:59** przez OLAT „Ćwiczenia 10”,

Rozwiązania wszystkich zadań mają być zapisane w jednym pliku .pl z numerem zadania w komentarzu. Proszę, by w nazwie pliku było nazwisko Studenta.