

Laborator 4 – Prolog

2016-2017

Programare Logică

Laboratorul 4

TODO

- Liste în Prolog.

Material suplimentar

- Capitolul 4 și Capitolul 6 din *Learn Prolog Now!*.

Liste

- Listele în Prolog sunt un tip special de date (termeni speciali).
- Listele în Prolog sunt de forma:
 - `[]` - lista vida
 - `[Head|Tail]` - listă cu primul element `Head` și restul listei `Tail`
 - `[E1,...,En|Tail]` - listă cu primele elemente `E1,...,En`, și restul listei `Tail`

Example:

- `[a]` - listă cu un element
- `[a,b]` - listă cu două elemente
- `[a,b|Tail]` - listă cu primele două elemente `a,b` și coada `Tail`

Liste

Exercițiu. Pentru a vă obișnui cu modul de reprezentare al listelor în Prolog, testați următoarele întrebări și încercați să înțelegeți de ce primiți răspunsurile respective:

?- [a,b,c,d] = [a|[a,b,c]] .

?- [a,b,c,d] = [Head|Tail] .

?- [a,b,c,d] = [a|X] .

?- [a,X] = [a|[a,b,c]] .

?- [a,b,c,d] = [a,b|[c,d]] .

?- [a,b,c,d] = [X,Y|_] .

?- [] = [[]] .

?- [X] = [[]] .

Lucrul cu liste

Exemplul 1: `trans_a_b/2`

- un predicat care "traduce" o listă de a-uri într-o listă de b-uri
- `trans_a_b(X,Y)` trebuie să fie adevărat dacă "intrarea" `X` este o listă de a-uri și "ieșirea" `Y` este o lista de b-uri, iar cele două liste au lungimi egale.

Soluție:

% Dacă intrarea este vidă, atunci și ieșirea este vidă.

```
trans_a_b([],[]).
```

```
/* Altfel, primul a din intrare trebuie să corespundă unui b  
din ieșire. Mai departe, coada listei de ieșire trebuie să fie  
o "traducere" a coadei listei de intrare. */
```

```
trans_a_b([a|InputTail],[b|OutputTail]) :-  
    trans_a_b(InputTail,OutputTail).
```

Lucrul cu liste

Exemplul 1: `trans_a_b/2` (cont.)

?- `trans_a_b([a,a,a],L)`.

?- `trans_a_b([a,a,a],[b])`.

?- `trans_a_b(L,[b,b])`.

Lucrul cu liste

Exemplul 2: `elements_of/2`

- un predicat care verifică dacă o listă conține un anumit termen
- `element_of(X,Y)` trebuie să fie adevărat dacă X este un element al lui Y.

Soluție:

```
/* Dacă primul element al listei este termenul  
pe care îl căutăm, atunci am terminat. */
```

```
element_of(X,[X|_]).
```

```
% Altfel, verificăm dacă termenul se află în restul listei.
```

```
element_of(X,[_|Tail]) :- element_of(X,Tail).
```

- În SWI-Prolog, `element_of/2` este predefinit sub numele `member`.

Lucrul cu liste

Exemplul 2: `elements_of/2`

```
?- element_of(a,[a,b,c]).
```

```
?- element_of(X,[a,b,c]).
```


Exercițiul 1: Liste

A) Definiți un predicat `all_a/1` care primește ca argument o listă și care verifică dacă argumentul său este format doar din a-uri.

B) Definiți un predicat `list_length/2` care are primul argument o listă de numere, iar al doilea argument lungimea primului argument.

De exemplu, la întrebarea

```
?-list_length([2,7,4],Result).
```

ar trebui să obțineți `Result = 3`.

Exercițiul 2: Operații cu vectori

A) Scrieți un predicat `scalarMult/3` al cărui prim argument este un întreg, al doilea argument este o listă de întregi, iar al treilea argument este rezultatul înmulțirii cu scalari al celui de-al doilea argument cu primul.

De exemplu, la întrebarea

`?-scalarMult(3, [2,7,4], Result).`

ar trebui să obțineți `Result = [6,21,12]`.

Exercițiul 2 (cont.)

B) Scrieți un predicat `dot/3` al cărui prim argument este o listă de întregi, al doilea argument este o listă de întregi de lungimea primeia, iar al treilea argument este produsul scalar dintre primele două argumente.

De exemplu, la întrebarea

```
?-dot([2,5,6],[3,4,1],Result).
```

ar trebui să obțineți `Result = 32`.

Exercițiul 2 (cont.)

C) Scrieți un predicat `max/2` care caută elementul maxim într-o listă de întregi.

De exemplu, la întrebarea

```
?-max([4,2,6,8,1],Result).
```

ar trebui să obțineți `Result = 8`.

Exercițiul 3

Definiți un predicat `palindrome/1` care este adevărat dacă lista primită ca argument este palindrom (lista citită de la stânga la dreapta este identică cu lista citită de la dreapta la stânga).

De exemplu, la întrebarea

```
?-palindrome([r,e,d,i,v,i,d,e,r]).
```

ar trebui să obțineți `true`.



Pe săptămâna viitoare!