Laborator 3 – Prolog

2017-2018

Programare Logică

Laboratorul 3

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 se scriu între paranteze drepte, cu elementele despărțite prin virgulă.
- `[]` este lista vidă.

Exemplu

- `[elephant, horse, donkey, dog]`
- `[elephant, [], X, parent(X, tom), [a, b, c], f(22)]`

Head & Tail

- Primul element al unei liste se numește *head*, iar restul listei *tail*.
- Evident, o listă vidă nu are un prim element.
- În Prolog există o notație utilă pentru liste cu separatorul `|`, evidențiind primul element și restul listei.

Exemplu

```
?- [1, 2, 3, 4, 5] = [Head | Tail].
Head = 1
Tail = [2, 3, 4, 5]
```

Cu această notație putem să returnăm ușor, de exemplu, al doilea element dintr-o listă.

```
?- [quod, licet, jovi, non, licet, bovi] = [_, X | _].
X = licet
```

Lucrul cu liste

Exemplu (element_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.

```
/* Dacă primult 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).

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

Lucrul cu liste

Exemplu (`concat_lists/3`)

- un predicat care este poate fi folosit pentru a concatena două liste
- al treilea argument este concatenarea listelor date ca prime două argumente

```
concat_lists([], List, List).
concat_lists([Elem | List1], List2, [Elem | List3]) :-
      concat_lists(List1, List2, List3).


?- concat_lists([1, 2, 3], [d, e, f, g], X).
?- concat_lists(X, Y, [a, b, c, d]).
```

Lucrul cu liste

În Prolog există niște predicate predefinite pentru lucrul cu liste. De exemplu:

- `length/2`: al doilea argument întoarce lungimea listei date ca prin argument
- `member/2`: este adevărat dacă primul argument se află în lista dată ca al doilea argument
- `append/3`: identic cu predicatul anterior `concat_lists/3`
- `last/2`: este adevărat dacă al doilea argument este identic cu ultimul element al listei date ca prim argument
- `reverse/2`: lista din al doilea argument este lista data ca prim elemenent în oglindă.

Practică

Exercițiul 1

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.

```
?- all_a([a,a,a,a]).
?- all_a([a,a,A,a]).
```

B) Scrieti un predicat `trans_a_b/2` 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 listă de b-uri, iar cele două liste au lungimi egale.

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

Practică

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 intrebarea

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

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

Practică

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`.

Practică

Exercițiul 2 (cont.)

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

De exemplu, la întrebarea

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

ar trebui să obțineți `Result = 8.`

Practică

Exercițiul 3

Definiți un predicat `palindome/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`.

Nu folosiți predicatul predefinit `reverse`, ci propria implementare a acestui predicat.

Practică

Exercițiul 4

Definiți un predicat `remove_duplicates/2` care șterge toate duplicatele din lista dată ca prim argument și întoarce rezultatul în al doilea argument.

De exemplu, la întrebarea

```
?- remove_duplicates([a, b, a, c, d, d], List).
```

ar trebui să obțineți `List = [b, a, c, d]`.

Pe săptămâna viitoare!