

Laborator 3

Logică matematică și computațională

Cuprins

- Alte exerciții cu liste
- Sortări

Alte exerciții cu liste

Exercițiul 1

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`.

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

Exercițiul 2

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]`.

Exercițiul 3

Definiți un predicat `atimes/3` care să fie adevărat exact atunci când elementul din primul argument apare în lista din al doilea argument de numărul de ori precizat în al treilea argument.

Interogați:

```
?- atimes(3,[3,1,2,1],X).  
?- atimes(1,[3,1,2,1],X).  
?- atimes(N,[3,1,2,1],2).  
?- atimes(N,[3,1,2,1],1).  
?- atimes(N,[3,1,2,1],0).  
?- atimes(N,[3,1,2,1],X).
```

Sortări

Sortarea prin inserție (*insertion sort*)

Predicatul `insertsort/2` sortează lista de pe primul argument folosind algoritmul *insertion sort*.

```
insertsort([], []).  
insertsort([H|T], L) :- insertsort(T, L1), insert(H, L1, L).
```

Exercițiul 4: scrieți regulile care definesc comportamentul predicatului ajutor `insert/3`.

Predicatul `quicksort/2` sortează lista de pe primul argument folosind algoritmul *quicksort*.

```
quicksort([], []).  
quicksort([H|T],L) :-  
    split(H,T,A,B), quicksort(A,M), quicksort(B,N),  
    append(M,[H|N],L).
```

Exercițiul 5: scrieți regulile care definesc comportamentul predicatului ajutător `split/4`.