SEMINAR 3 – Liste eterogene in Prolog

[H|T]

|  |  |  |
| --- | --- | --- |
| H \ T | 3 | [3] |
| 2 | [2|3] | [2,3] |
| [2] | [[2]|3] | [[2],3] |

Is\_list(H) => true, daca H este lista; false, altfel

Number(H) => true, daca H este numar; false, altfel

Atom(H) => true, daca H este simbol; false, altfel

1. Se da o lista eterogena formata din numere si liste de numere.

Se cere sa se determine numarul sublistelor avand aspect de munte.

Se presupune ca o lista numerica are aspect de munte daca este formata dintr-o secventa nevida strict crescatoare urmata de o secventa nevida strict descrescatoare.

Ex:

[1, 3, 2, [4, 7, 9], 6, [1,2,3,0], [9,7,6,5], 4, 6, [7,6,5], [1,9,8,7,4,3]]

=> 2

% urca = 1, daca suntem pe partea crescatoare

% urca =- 1, daca suntem pe partea descrescatoare

Munte(l1l2...ln,urca) = { true, daca n=1 si urca =-1

= Munte(l2..ln, urca), daca l1<l2 si urca = 1

= Munte(l2...ln,urca), daca l1>l2 si urca =-1

= Munte(l2...ln,-1), daca l1>l2 si urca = 1

= false, altfel

Principal(l1l2...ln) = Munte(l1l2..ln,1), n>2 si l1 < l2

Fals, altfel

% Munte(L:lista liniara de numere, Urca:Intreg)

% Model de flux (I,i) - determinist

Munte([\_],-1).

Munte([H1|[H2|T]], 1):-

H1 < H2,!,

Munte([H2|T], 1).

Munte([H1|[H2|T]], -1):-

H1 > H2,

Munte([H2|T], -1).

Munte([H1|[H2|T]], 1):-

H1>H2,

Munte([H2|T],-1).

% Principal(L: lista de numere)

% Model de flux (I), determinist

Principal([H1|[H2|[H3|T]]]) :-

H1 < H2,

Munte([H1|[H2|[H3 | T]]],1).

Numar\_munti(l1,…,ln)={ 0, daca l este vida

{ numar\_munti(l2,…,ln) +1 daca l1 este lista si principal(l1)=true

{numar\_munti(l2,...ln), atfel

% NrMunti(L: lista eterogena, NrAp: Intreg)

% Modele de flux: (I, o) - determinist

NrMunti([], 0).

NrMunti([H|T], NrAp) :-

Is\_list(H),

Principal(H),!,

NrMunti(T, NrAp1),

NrAp is NrAp1 + 1.

NrMunti([\_|T], NrAp) :-

NrMunti(T, NrAp).

1. Se da o lista eterogena formata din numere si liste de numere.

Se cere sa se elimine din toate sublistele numerele palindroame.

Ex: [121. 56. 3. [11. 7, 78, 5, 32], 6, 77, [555]]

=> [121. 56. 3. [78, 32], 6, 77, []]

Oglindit(nr) = { 0, daca nr=0

{ oglindit([nr/10]) \* 10 + nr%10, altfel

% Oglindit(Nr: numar intreg, Aux: intreg)

% model de flux (I,o)

Oglindit(0,AUX):- AUX is 0.

Oglindit(NR,AUX) :-

X is NR/10,

oglindit(X,AUX2).

AUX is AUX2\*10+mod(NR,10).