

# Programare Logică – Laboratorul 4

## Acumulatori, Liste deschise și liste cu diferențe

Isabela Drămnesc

### 1 Întrebări din curs

- Ce înțelegeți prin domeniu inductiv? Cum definim o listă?
- Cum definim recursia în Prolog? Dați 2 exemple!
- Scrieți predicatul pentru concatenarea a două liste *concatenare*(+L1, +L2, ?Rez)
- Scrieți predicatul pentru calculul factorialului unui număr *factorial*(+N, ?F)

### 2 Concepte

- Operații pe liste
- Acumulatori
- Diferența dintre programe cu acumulatori și fără
- Liste deschise și liste cu diferențe

### 3 Acumulatori

Se descompune lista până se ajunge la condiția limită, atunci se obține soluția care este reținută în acumulator, apoi are loc procesul invers descompunerii listei, iar rezultatul este soluția care a fost reținută în acumulator.

1) Inversa listei:

a) fără acumulatori:

```
inversa ([], []).
```

```
inversa ([H|T], X):-inversa(T, T1), concatenare(T1, [H], X).
```

Explicați ce se întâmplă la interogarea (utilizați trace)

?- inversa([1,2,3],X).

b) cu acumulatori:

```
inversa2(L,R):-inversaAcc(L,[],R). /* invocarea variantei cu acumulatori */
```

```
inversaAcc([],R,R). /* conditie la limita */
```

```
inversaAcc([H|T],A,R):-
```

```
inversaAcc(T,[H|A],R). /* apel recursiv cu acumulatori */
```

Încercați un trace de `inversaAcc([1,2,3],[],R)`.

Și un trace de `inversa2([1,2,3],X)`. Comparați costurile pentru `inversa` și pentru `inversa2`.

2) Predicatul care calculează termenul  $n$  din șirul Fibonacci: 1,1,2,3,5,8,13,21,..., în care  $f(n) = f(n-1) + f(n-2)$  pentru  $n > 2$ .

(a) varianta fără acumulator:

```
fibonacci(1,1).
fibonacci(2,1).
fibonacci(N,F):-N>2,N1 is N-1,N2 is N-2,
                fibonacci(N1,F1), fibonacci(N2,F2),
                F is F1+F2.
```

(b) varianta cu acumulator:

```
/* F=f(M). F1=f(M-2), F2=f(M-1). */
fibonac(N,F):-fib(2,N,1,1,F).
fib(M,N,_,F2,F2):-M >= N.
fib(M,N,F1,F2,F):-M < N, M1 is M+1,
                  F1plusF2 is F1+F2,
                  fib(M1,N,F2,F1plusF2,F).
```

Utilizați trace pentru a vedea diferența dintre cele două programe (fără acumulator și cu acumulator).

3) Exemplu de predicat care returnează lungimea unei liste:

```
lungime([],0).
lungime([_|T],N):-lungime(T,N1),N is 1+N1.
```

Modificați predicatul astfel încât să folosească acumulatori.

4) Modificați predicatul care calculează  $n!$  astfel încât să folosească acumulatori.

5) Utilizați următoarele predicate pentru a vedea timpul de execuție în varianta cu acumulator și fără pentru `inversa` unei liste, pentru factorial, pentru lungimea unei liste, pentru fibonacci:

```
current_time(Timestamp) :- get_time(Current),
stamp_date_time(Current,Date,local),
date_time_value(time, Date, Timestamp).
```

```
time_elapsed(time(H1,M1,S1),
time(H2,M2,S2),
Seconds) :- Seconds is 3600 * (H2 - H1) + 60 * (M2 - M1) + (S2 - S1).
```

De exemplu pentru factorial:

```
?- current_time(T1), factorial(50000, X),
   current_time(T2), time_elapsed(T1,T2,T).
?- current_time(T1), factAcc(50000, X),
   current_time(T2), time_elapsed(T1,T2,T).
```

Unde `factorial`=predicatul fără acumulatori, iar `factAcc`=predicatul cu acumulatori.

## 4 Liste deschise și liste cu diferențe

a) Exemplu de listă deschisă:

6)

?- L=[1,2,3|X],X=[t,g,h].     /\* gaura din lista L se umple  
   într-un singur pas cu X \*/

L = [1, 2, 3, t, g, h],  
X=[t,g,h]

?- L2=[s,d,a|T],T=[cap|T2].     /\* gaura se poate umple si cu  
   o lista deschisa \*/

L2 = [s, d, a, cap|T2],  
T=[cap|T2]

7)

concatlistdeschise(L1,G,L2):-G=L2.

?- X=[a,b,c|G],concatlistdeschise(X,G,[d,e,f]).  
X = [a, b, c, d, e, f]

b) Liste cu diferențe se reprezintă ca diferența dintre lista deschisă și gaura sa

Exemplu:  $[1,2,3|G] - G$

8) Introduceți regula:

concatenareDeschise(LD-G,L2):-G=L2.

Încercați pentru:

?- X=[m,n,p|G]-G,concatenareDeschise(X,[1,2,3]).

Scrieți regula astfel încât să se obțină  $X = [m,n,p,1,2,3]$ .

9) Predicatul concatenare/3:

concatenare3(LD1-G1,LD2-G2,LD1-G2):-G1=LD2.

?- X=[a,f|G]-G,concatenare3(X,[p,l|G2]-G2,Raspuns).

?- X=[a,f|G]-G,concatenare3(X,[p,l|G2]-G2,Raspuns-[]).

10) Testați și pentru predicatul concatenare33.

concatenare33(LD1-G1,G1-G2,LD1-G2):-G1=LD2.

?- X=[a,f|G]-G,concatenare33(X,[p,l|G2]-G2,Raspuns).

?- X=[a,f|G]-G,concatenare33(X,[p,l|G2]-G2,Raspuns-[]).

## 5 Tema:

Terminați toate problemele care au mai rămas nerezolvate din [Tema 3](#).

Termen de realizare: laboratorul următor.