

Laborator 6

Logică matematică și computațională

Inversarea listelor

Predicatul următor, `listN/2`, va fi util pentru generarea unei liste de lungime dată:

```
listN([],0).
```

```
listN([a|T], N) :- N > 0, M is N - 1, listN(T,M).
```

Introducem și metapredicatul `listing/1`, care afișează toate clauzele corespunzătoare unui predicat. Interogați:

```
?- listing(listN).
```

Reamintim, acum, din soluțiile Laboratorului 3, definirea predicatului `rev/2` de inversare a listelor:

```
rev([],[]).
```

```
rev([H|T],L) :- rev(T,N), append(N,[H],L).
```

Soluția dată nu este prea eficientă, având o complexitate pătratică.

Soluția următoare o îmbunătățește pe cea precedentă, adăugând un predicat auxiliar, care are un parametru în plus, care joacă rol de acumulator. Complexitatea devine liniară (testați pentru liste de lungime 1000-10000):

```
reva(L,R) :- revah(L, [],R).
```

```
revah([], R, R).
```

```
revah([H|T], S, N) :- revah(T, [H|S],N).
```

Contemplați adevărul următoarei afirmații: pentru orice A, B, C, avem că `revah(A,B,C)` dacă și numai dacă, notând cu M inversa listei A, avem că `append(M,B,C)`.

În continuare, ținând cont de această afirmație, vom rescrie soluția de mai sus, permițând generalizarea ei la alte probleme.

Difference lists

Reamintim că afirmația era: pentru orice A, B, C , avem că $\text{revah}(A, B, C)$ dacă și numai dacă, notând cu M inversa listei A , avem că $\text{append}(M, B, C)$. Altfel spus, inversa lui A este „ C fără B ”.

Vom reprezenta expresia „ C fără B ” sub forma unei perechi (C, B) și o vom numi *difference list* sau **difflist**.

Definiția anterioară devine:

$\text{revd}(L, R) :- \text{revdh}(L, (R, []))$.

$\text{revdh}([], (R, R))$.

$\text{revdh}([H|T], (N, S)) :- \text{revdh}(T, (N, [H|S]))$.

Exercițiul 1

Definiți un predicat `flatten/2` care aplatizează structura unei liste.

Exemplu:

```
?- flatten([1,2,[3,a],[[7],2],5],L).
```

```
L = [1, 2, 3, a, 7, 2, 5]
```

Dați o soluție care folosește `append/3` și una care folosește `difflist-uri`.

Indiciu: Folosiți metapredicatul `is_list/1`.

Exercițiul 2

Reamintim, tot din soluțiile Laboratorului 3, definirea predicatului quicksort/2:

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

Rescrieți această definiție folosind difflist-uri (fără a mai folosi append/3).