

Materiale suplimentare de Prolog

Logică matematică și computațională

Arbori binari

Reamintim că, în Prolog, listele erau de două feluri:

- lista vidă `[]`;
- listă nevidă `[H|T]` formată din capul `H` (care putea fi orice) și coada `T` (care era o listă).

Similar, arborii binari vor fi de două feluri:

- arborele vid;
- arbore nevid format dintr-o rădăcină și doi subarbori.

Arborii binari nu sunt o funcționalitate inherentă a Prolog-ului, de aceea pentru ei se vor folosi simboluri de funcție *ad hoc*, lucru permis de limbaj.

Arbori binari – definire și parcurgere

Pentru arborele vid, vom folosi atomul `vid`. Pentru arborele care are rădăcina `R` și cei doi subarbori `T` și `U`, vom folosi termenul `arb(R,T,U)`.

Interogați:

```
X = arb(3,arb(2,vid,vid),vid).
```

Definirea predicatului `srd/2` folosit la parcurgerea unui arbore binar în inordine:

```
srd(vid, []).
```

```
srd(arb(R,T,U),L) :- srd(T,L1), srd(U,L2),
```

```
append(L1,[R|L2],L).
```

Testați acest predicat!

Exercițiul 1: arbori binari de căutare

Definiți următoarele predicate:

- `bt_ins/3` care inserează numărul natural din primul argument în arborele binar de căutare din al doilea argument;
- `bt_list/2` care transformă lista din primul argument într-un arbore binar de căutare;
- `bt_sort/2` care sortează lista din primul argument trecând prin arborele binar de căutare asociat ei.