Laborator 8 - Maude

2016-2017

Programare Logică

Laboratorul 8

TODO

- Arbori binari în Maude.

Arbori binari

Considerăm că un arbore poate fi

- vid sau
- un număr întreg cu un subarbore stâng și unul drept.

```
fmod TREE is
  protecting INT .
  sort Tree .
  op empty : -> Tree .
  op _ _ _ : Tree Int Tree -> Tree .
endfm
```

Operații pe arbori

Următoarea operație întoarce imaginea în oglindă a unui arbore

- recursiv înlocuiește fiecare subarbore stâng cu imaginea în oglindă a subarborelui drept și vice-versa.

```
fmod MIRROR is
  protecting TREE .
  op mirror : Tree -> Tree .
  vars L R : Tree . var I : Int .
  eq mirror(empty) = empty .
  eq mirror(L I R) = mirror(R) I mirror(L) .
endfm
```

```
red mirror((empty 3 (empty 1 empty)) 5 ((empty 6 empty) 2 empty)) .
***> should be (empty 2 (empty 6 empty)) 5 ((empty 1 empty) 3 empty)
```

Practică

Exercițiul 1

Plecând de la specificația `TREE`, definiți următoarele operații:

1. Căutarea în arbori binari
   ```
   op search :  Int Tree -> Bool
   ```
2. Adâncimea unui arbore
   ```
   op depth :  Tree -> Int
   ```
3. Traversarea în
   1. inordine `op SRD : Tree -> List`
   2. preordine `op RSD : Tree -> List`
   3. postordine `op SDR : Tree -> List`

Practică

Exercițiul 2

Scrieți o specificație care folosește arbori binari de căutare pentru a sorta liste de întregi.

- Definiți operația

  ```
  bt-insert :  NList Tree -> Tree
  ```

  care inserează fiecare întreg din listă în locul său în arborele de căutare.

- Definiți operația

  ```
  btsort :  NList -> NList
  ```

  care sortează lista dată ca argument și folosește operația `SRD`.

Baftă la parțial!