

Paradygmaty programowania

Lista 8

Zad. 1. [2p] Dla drzewa o elementach lista liczb, które w OCaml posiadałoby typ `type 'a bt = Empty | Node of 'a * 'a bt * 'a bt` zdefiniować w Scala funkcję przekształcającą dane drzewo list do drzewa sum elementów tych list.

Zad. 2. [4p] Założyć, że graf otwarty modelowany jest za pomocą drzewa z węzłem zawierającym wartość dowolnego typu oraz listę węzłów powiązanych. Napisać w Scala funkcję przekształcającą tak zamodelowany graf na drzewo binarne, które w OCaml posiadałoby typ `type 'a bt = Empty | Node of 'a * 'a bt * 'a bt`. Zaproponować rozwiązanie problemu nadmiaru węzłów w liście.

Zad. 3. [1+2+1p] Napisać w OCaml trzy funkcje:

- sprawdzającą czy dwa drzewa są sobie równe (typ `'a -> 'a -> bool`),
- sprawdzającą czy dane drzewo jest poddrzewem innego danego drzewa (typ `'a bt -> 'a bt -> bool`),
- obliczającą ile razy dane drzewo jest poddrzewem innego danego drzewa (typ `'a bt -> 'a bt -> int`).