

Lista 4

Na wykładzie został zdefiniowany następujący typ dla drzew binarnych:

type 'a bt = Empty | Node of 'a * 'a bt * 'a bt;;

let t = Node(1,Node(2,Empty,Node(3,Empty,Empty)),Empty);;

1. Napisz funkcję sumBT: int bt -> int, obliczającą sumę liczb całkowitych, przechowywanych w węzłach danego drzewa.

np. sumBT t = 6;;

2. Napisz funkcjonal foldBT : ('a -> 'b * 'b -> 'b) -> 'b -> 'a bt -> 'b, uogólniający funkcję sumowania wartości z węzłów drzewa binarnego tak, jak funkcjonal fold_right uogólnia funkcję sumowania elementów listy (patrz wykład 3).

3. Wykorzystaj foldBT do obliczenia

a) sumy liczb całkowitych sumBTfold : int bt -> int

np. sumBTfold t = 6;;

b) listy wartości pamiętanych w węzłach drzewa (w obejściu infiksowym)

inorderBTfold : 'a bt -> 'a list

np. inorderBTfold t = [2; 3; 1];;

4. Wykorzystując foldBT zdefiniuj funkcjonal mapBT: ('a -> 'b) -> 'a bt -> 'b bt, aplikujący daną funkcję do wartości we wszystkich węzłach drzewa.

np. mapBT (fun v -> 2*v) t = Node (2, Node (4, Empty, Node (6, Empty, Empty)), Empty);;

Zdefiniowany jest następujący typ dla drzew binarnych:

type 'a tree = L of 'a | N of 'a tree * 'a tree;;

Napisz funkcje:

5. store: 'a tree -> 'a option list, przedstawiającą drzewo w postaci listy elementów typu 'a option w porządku postfiksowym; wartość None reprezentuje wierzchołek wewnętrzny, Some x reprezentuje liść o etykiecie x.

6. load: 'a option list -> 'a tree odtwarzającą drzewo na podstawie listy wygenerowanej przez funkcję store. Jeśli lista nie reprezentuje żadnego drzewa, to należy zgłosić wyjątek Load.