

Programowanie funkcyjne

Kolokwium
14 stycznia 2014

1. (2+2)

- a) Proszę zdefiniować funkcję `delete_all x l`, która eliminuje wszystkie elementy `x` w liście `l`.
- b) Proszę zdefiniować funkcję `flatten l`, która transformuje listę list do zwykłej listy.
Przykład:

```
> flatten [[1,2,3], [8,9], [4,6]]  
[1,2,3,8,9,4,6]
```

2. (2+2+2+1)

Proszę napisać funkcję `filter p l`, która zwraca listę elementów z listy `l` spełniających predykat `p`.

- a) używając rekurencji,
- b) używając set comprehension,
- c) używając odpowiedniego kombinatora (oprócz `filter` oczywiście :-).
- d) Proszę podać typ funkcji z a) - c).

3. (2+3+2+3+4)

Niech będzie dany następujący typ drzew:

```
data Tree a = Nil | Node a (Tree a) (Tree a)
```

- a) Proszę zdefiniować funkcję `tree_size t`, która obliczy gęstość drzewa `t`.
- b) Proszę zdefiniować funkcję `tree_max`, która zwraca największy element w drzewie `t`.
- c) Proszę zdefiniować funkcję `map f t`, która zastosuje funkcję `f` do wszystkich elementów w drzewie `t`.
- d) Dla drzew z listami jako elementy proszę napisać funkcję, która do każdego węzła dołącza długość jego listy.
- e) Proszę podać typ funkcji z a) - d).

4. (3)

Proszę obliczyć najbardziej ogólny typ następującego wyrażenia.

$$\lambda x \rightarrow f x y$$