

Programowanie funkcyjne — kolokwium nr 1, 10.12.2014

Instrukcja: Rozwiązania zadań należy przesłać do godziny 9:50 na adres `kolokwium.pf@gmail.com` (decyduje data stempla googlowego). Każde zadanie należy przesłać w oddzielnym pliku `Zadanie n .hs` ($n = 1, 2, 3$). W rozwiązaniach nie można korzystać z modułów innych niż `Prelude`, tzn. niedozwolone jest użycie polecenia `import`. Punktacja: po 10 punktów za zadanie.

Zadanie 1. Napisać bezpunktowo funkcję `numocc`, która zlicza wystąpienia wskazanego elementu w podanych listach, tzn. $\text{numocc } x [\ell_1, \ell_2, \dots, \ell_n] = [a_1, a_2, \dots, a_n]$, gdzie a_i to liczba wystąpień x w liście ℓ_i . Na przykład

$\text{numocc } 1 [[1, 2], [2, 3, 2, 1, 1], [3]] = [1, 2, 0]$.

Podać najogólniejszą możliwą sygnaturę.

Zadanie 2. Obliczenia z użyciem operacji dodawania, mnożenia i negacji na elementach typu a (będącego instancją klasy `Num`) będziemy reprezentować przez drzewo obliczeń typu `CT a`, zdefiniowane jako

$\text{data CT } a = \text{Empty} \mid \text{Leaf } a \mid \text{Join (CT } a) \text{ Op (CT } a)$,

przy czym `Empty` to drzewo puste, `Leaf` — liść drzewa zawierający wartość typu a , zaś `Join` — drzewo reprezentujące wykonanie operacji `Op` na obliczeniach z lewego i prawego poddrzewa. Typ `Op` jest określony jako

$\text{data Op} = \text{Add} \mid \text{Mul} \mid \text{Neg}$.

(a) Napisać funkcję $\text{wf} :: \text{CT } a \rightarrow \text{Bool}$, która dla podanego drzewa sprawdza, czy jest ono poprawne, w tym sensie, że nie jest puste i nie zawiera poddrzew postaci $(\text{Join Empty Add } r)$ lub $(\text{Join } \ell \text{ Add Empty})$ i analogicznie dla `Mul`, oraz poddrzew postaci $(\text{Join } \ell \text{ Neg } r)$, gdzie oba drzewa ℓ, r są niepuste.

(b) Napisać funkcję $\text{eval} :: \text{Num } a \Rightarrow \text{CT } a \rightarrow a$, która dla podanego drzewa wyliczy wartość reprezentowanego w nim obliczenia. Na przykład dla

$T = (\text{Join } (\text{Join } (\text{Leaf } 3) \text{ Add } (\text{Leaf } 2)) \text{ Mul } (\text{Join } (\text{Leaf } 2) \text{ Neg Empty}))$

wartość $\text{eval } T = -10$. Jeśli drzewo nie jest poprawne, funkcja powinna sygnalizować błąd.

Zadanie 3. Napisać bezpunktowo funkcję `h`, która zwraca elementy podanej listy znajdujące się na pozycjach o numerach parzystych, w takiej kolejności, w jakiej występują w oryginalnej liście. Na przykład

$h [0, 1, 2, 3, 4] = [0, 2, 4]$,
 $h \text{ "AlaMaKota"} = \text{ "Aaaoa"}$.