

Programowanie funkcyjne — kolokwium nr 1, 4.12.2024

Instrukcja: Każde zadanie należy przesłać na Pegaza w oddzielnym pliku: zadanie1.hs, zadanie2.hs i zadanie3.hs. Plików nie należy zipować. Rozwiązania muszą się poprawnie kompilować. W rozwiązaniach nie można korzystać z modułów innych niż standardowe; niedozwolone jest użycie polecenia `import`. Rozwiązania nie spełniające powyższych wymogów nie będą oceniane. Punktacja: 10 punktów za każde zadanie. Uwaga: korzystanie z internetu poza wyznaczonym czasem skutkuje automatycznym otrzymaniem 0 punktów. Użycie polskich znaków w sygnaturach nie jest obowiązkowe.

Zadanie 1. Napisać funkcję o sygnaturze

`wypłata :: Integer -> Integer -> Integer -> [Integer]`,

która dla parametrów a , b i n zwraca listę uszeregowanych rosnąco wszystkich liczb całkowitych dodatnich nie większych niż n , które jako kwoty da się wypłacić używając monet o nominałach a i b . Przykładowo, wywołanie `wypłata 2 5 10` powinno zwrócić listę `[2, 4, 5, 6, 7, 8, 9, 10]`, ponieważ tylko kwoty 1 i 3 nie są możliwe do wypłacenia za pomocą monet o nominałach 2 i 5.

Zadanie 2. Napisać funkcję o sygnaturze

`słowa :: Integer -> Integer -> [String]`,

która dla nieujemnych parametrów m i n zwraca listę wszystkich napisów postaci $a^i b^j$, gdzie $0 \leq i \leq m$, $0 \leq j \leq n$ oraz $i \neq j$, w dowolnej kolejności. Przykładowo, `słowa 1 2` powinno zwrócić listę `["a", "b", "bb", "abb"]` lub jej dowolną permutację.

Zadanie 3. *Gekon tkliwy* to struktura danych podobna do listy, która oprócz gromadzenia danych posiada funkcję odrzucenia ogona: jeśli przed dołączeniem nowego elementu x spełniony jest predykat odrzucenia, gekon „odrzuca ogon”, czyli usuwa dotychczasowe elementy i łączy x jako jedyny element. Dodatkowo gekon zlicza, ile ogonów zostało odrzuconych. Gekon implementuje następujące operacje: `gp` zwraca gekon pusty z „wbudowanym” predykatem odrzucenia; `de` łączy nowy element jako głowę (i ewentualnie odrzuca ogon); `oe` odczytuje głowę (o ile gekon nie jest pusty); `ue` usuwa głowę (jeśli gekon jest pusty, to pozostaje bez zmian); `loo` podaje liczbę odrzuconych ogonów; `g2l` podaje elementy jako listę, w której element ostatnio dołączony jest głową. Wszystkie operacje z wyjątkiem `g2l` działają w czasie stałym, z tym że pomijamy czas obliczania predykatu odrzucenia.

Zdefiniować typ `Gkt a`, służący do przechowywania elementów typu `a` w gekonie tkliwym, oraz następujące funkcje, realizujące opisane wyżej operacje z odpowiednią złożonością:

```
gp  :: (Gkt a -> Bool) -> Gkt a
de  :: a -> Gkt a -> Gkt a
oe  :: Gkt a -> a
ue  :: Gkt a -> Gkt a
loo :: Gkt a -> Int
g2l :: Gkt a -> [a]
```