

Programowanie funkcyjne — kolokwium nr 2, 28.01.2016

Instrukcja: Rozwiązania zadań należy przesłać do godziny 13:15 na adres `kolokwium.pf@gmail.com` (decyduje data stempla googlowego). Całość należy przesłać w jednym mailu, ale każde zadanie w oddzielnym pliku: `zad1.hs`, `zad2.hs` i `zad3.erl`. Plików nie należy zipować. W rozwiązaniach nie można korzystać z modułów innych niż standardowe (niedozwolone jest użycie polecenia `import`). Rozwiązania nie spełniające powyższych wymogów nie będą oceniane. Zadania 1 i 2 napisać w Haskellu, zadanie 3 — w Erlangu. Punktacja: 10 punktów za każde zadanie. Uwaga: korzystanie z internetu poza wyznaczonym czasem skutkuje automatycznym otrzymaniem 0 punktów.

Zadanie 1. Napisać bezpunktowo funkcję f , która dla podanej listy $[a_0, a_1, \dots, a_n]$ zwraca wartość ułamka $a_0 + 1/(a_1 + 1/(\dots + 1/(a_n + 1)))$; na przykład $f [3, 4, 5] = 3 + 1/(4 + 1/(5 + 1)) = 3.24$. Przyjąć, że $f [] = 1$.

Zadanie 2. Komputer ma jeden rejestr, w którym może przechowywać liczby całkowite z zakresu $-10 \dots 10$. Na tym rejestrze można wykonywać dwie operacje: `inc` oraz `dec` — zwiększenie/zmniejszenie zawartości rejestru o 1, chyba że spowoduje to przekroczenie zakresu dopuszczalnych wartości; w takim wypadku komputer wchodzi w stan błędu i dalsze operacje `inc/dec` nie odnoszą żadnych skutków.

Zdefiniować typ danych Rejestr, umożliwiający przechowywanie zawartości rejestru oraz obsługę stanu błędu. Typ można zdefiniować na dwa sposoby:

```
data R a = ...
type Rejestr = R Int
albo
type Rejestr = Maybe ...
```

W pierwszym przypadku należy również uczynić pomocniczy typ `R` instancją klasy `Monad`. Następnie zdefiniować funkcje `inc` i `dec`, które umożliwią wykonywanie opisanych wyżej operacji na rejestrze i łączenie ich w ciągi w następujący sposób:

```
(rejestr zaw. 1) >>= inc >>= inc >>= dec = (rejestr zaw. 2)
(rejestr zaw. 10) >>= inc >>= dec >>= dec = (stan błędu)
```

Zadanie 3. Napisać moduł, który uruchomi dokładnie dwa procesy. Każdy z procesów ma odbierać dwie liczby (x oraz d), wypisywać x na konsolę i przekazywać drugiemu procesowi liczby zmodyfikowane tak, by tamten wypisał $x + d$. Dodatkowo gdy $|x| > 100$, proces ma zmienić znak liczby d . Procesy mają zostać uruchomione tak, by całość wypisywała „w nieskończoność” ciąg dla wartości początkowych $x = 1$ i $d = 2$:

1, 3, 5, ..., 99, 101, 99, 98, ..., 3, 1, -1, -3, ..., -99, -101, -99, -98, ..., -3, -1, 1, ...