

Programowanie funkcyjne — kolokwium nr 2, 25.01.2017

Instrukcja: Rozwiązania zadań należy przesłać do godziny 9:30 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`. Zadania 1 i 2 napisać w Haskellu, zadanie 3 — w Erlangu. 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.

Zadanie 1. Napisać funkcję `zipM :: Monad m => [m a] -> [m b] -> m [(a,b)]`, która działa jak „monadowy zip”. Na wejściu funkcja otrzymuje dwie listy wartości opakowanych w monadę. Następnie łączy te wartości w pary i całość opakuje w monadę. W przypadku list różnych długości należy przyjąć, że wynik ma długość krótszej z list. Na przykład:

```
zipM [Just 5, Just 6, Just 7] [Just 4, Just 3] = Just [(5,4), (6,3)]
```

Można przyjąć, że wypakowanie jest zawsze możliwe (np. na listach nie ma `Nothing`).

Zadanie 2. Napisać funkcję `fmap`, która dla podanej listy funkcji i elementu x zwraca listę zawierającą wartości kolejnych funkcji z listy wejściowej na argumentie x . Na przykład:

```
fmap [(+1), (*2), abs] (-2) = [-1, -4, 2]
fmap [reverse, filter (=='a'), tail] "AlaMaKota" =
    ["atoKaMaLa", "aaa", "laMaKota"]
```

Napisać funkcję `fmap` bezpunktowo i określić jej najogólniejszą możliwą sygnaturę.

Zadanie 3. Napisać moduł, który uruchomi N procesów ($N \geq 3$). Każdy z procesów ma odbierać listę liczb i wypisywać ją na konsolę, a następnie dołączać na początku listy liczbę o 1 większą niż dotychczasowa głowa i przekazywać ją następnemu procesowi w cyklu (w nieskończonej pętli). Moduł ma udostępniać funkcję `start(N)`, która uruchomi procesy w stosownym cyklu tak, by całość wypisywała coraz dłuższe listy:

```
[1]
[2,1]
[3,2,1]
...
```