Programowanie funkcyjne — kolokwium nr 2, 23.01.2014

Instrukcja: Rozwiązania zadań należy przesłać do godziny 11:40 na adres `kolokwium.pf@gmail.com` (decyduje data stempla googlowego). Każde zadanie należy przesłać w oddzielnym pliku, odpowiednio zad1.hs, zad2.hs i zad3.erl. Plików nie należy zipować. Nie wysyłać zadań pustych. W rozwiązaniach nie można korzystać z modułów innych niż standardowe. Rozwiązania nie spełniające powyższych wymogów nie będą oceniane. Punktacja: po 10 punktów za zadanie, przy czym podpunkty drugiego zadania mają wagi $5 + 3 + 2$.

Zadanie 1. Napisać funkcję perm :: Ord $\alpha \Rightarrow [\alpha] \rightarrow [[\alpha]]$, generującą listę wszystkich permutacji podanej listy skończonej. Permutacje w wynikowej liście mają być uporządkowane leksykograficznie.

Zadanie 2. Napisać funkcje:

- is3 :: Int $\rightarrow$ Maybe Int, która zwraca (opakowaną w Maybe) liczbę is3$(x) = \sqrt[3]{x}$, jeśli jest to liczba całkowita, oraz Nothing w przeciwnym wypadku,
- c :: Maybe Int $\rightarrow$ Int, która zwraca wartość c$(x) = 2x + 1$, jeśli x jest poprawną wartością opakowaną w monadę, oraz 0 w przeciwnym wypadku,
- fun :: Int $\rightarrow$ Int, która zwraca wartość fun$(x) = 2\sqrt[3]{x}+1$, jeśli x jest sześcianem pewnej liczby całkowitej, oraz 0 w przeciwnym wypadku. W rozwiązaniu należy użyć funkcji z poprzednich podpunktów oraz operatora >>=. Wskazówka: do opakowania x-a w monadę użyć od razu konstruktora, a nie funkcji return.

Zadanie 3. Napisać w Erlangu moduł, który uruchomi dwa procesy. Każdy z procesów ma odbierać liczbę, wypisywać ją na konsolę, dodawać do niej 1 i przekazywać ją drugiemu procesowi (w nieskończonej pętli). Procesy mają zostać uruchomione tak, by całość wypisywała kolejne liczby naturalne.