

## Programowanie funkcyjne — kolokwium nr 2, 28.01.2015

**Instrukcja:** Rozwiązania zadań należy przesyłać do godziny 9:50 na adres [kolokwium.pf@gmail.com](mailto:kolokwium.pf@gmail.com) (decyduje data stempla googlowego). Każde zadanie należy przesyłać w oddzielnym pliku: Zadanie1.hs, Zadanie2.hs i Zadanie3.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.

**Zadanie 1.** Uzupełnić definicję funkcji  $f = foldl (\dots)$  tak, by otrzymać funkcję, która na wejściu otrzymuje listę liczb całkowitych i rozdziela ją na listę list składających się z kolejnych liczb; na przykład  $f [1, 2, 3, 5, 8, 9, 10] = [[1, 2, 3], [5], [8, 9, 10]]$ . W rozwiązaniu założyć, że lista na wejściu jest niepusta.

**Zadanie 2.** Napisać funkcję

$$g :: \text{Monad } m \Rightarrow (a \rightarrow b \rightarrow m\ a) \rightarrow a \rightarrow [b] \rightarrow m\ a,$$

która dla podanej funkcji dwuargumentowej  $h$ , elementu  $p$  oraz listy  $[x_1, x_2, \dots, x_n]$  działa w następujący sposób:

- aplikuje  $h$  na  $p$  oraz  $x_1$ , otrzymując  $a_2$  (opakowane przez  $m$ ),
- aplikuje  $h$  na  $a_2$  oraz  $x_2$ , otrzymując  $a_3$  (opakowane przez  $m$ ),
- ... i tak aż do końca listy; ostatecznym wynikiem jest  $h\ a_n\ x_n$ .

Jeśli wejściowa lista jest pusta, funkcja powinna zwrócić  $p$  (opakowane przez  $m$ ).

**Zadanie 3.** Napisać w Erlangu moduł, który uruchomi  $N$  procesów-„dodawaczy” ( $N \geq 3$ ). Każdy z dodawaczy ma odbierać liczbę, wypisywać ją na konsolę, dodawać do niej 1 i przekazywać ją następnemu dodawaczowi w cyklu (w nieskończonej pętli). Moduł ma udostępnić funkcję  $\text{start}(N)$ , która uruchomi dodawacze w stosownym cyklu tak, by całość wypisywała kolejne liczby naturalne. Moduł nie może uruchamiać żadnych innych procesów.