Paradygmaty programowania - ćwiczenia
Lista 2

1. Jaka będzie głębokość stosu w Scali, a jaka w OCamlu dla wywołania evenR(3) (funkcja zdefiniowana na wykładzie)?

Poniższe funkcje należy napisać w obu językach: OCaml i Scala (wykorzystując mechanizm dopasowania do wzorca).

2. Liczby Fibonacciego są zdefiniowane następująco:

$f(0) = 0$
$f(1) = 1$
$f(n+2) = f(n) + f(n+1)$

Napisz dwie funkcje, które dla danego n znajdują n-tą liczbę Fibonacciego:
   a) opartą bezpośrednio na powyższej definicji,
   b) wykorzystującą rekursję ogonową.

Porównaj (bez mierzenia) ich szybkość wykonania, obliczając np. 42-gą liczbę Fibonacciego.

3. Dla zadanej liczby rzeczywistej a oraz dokładności ε można znaleźć pierwiastek trzeciego stopnia z a wyliczając kolejne przybliżenia x_i tego pierwiastka (metoda Newtona-Raphsona):

$x_0 = a/3$ dla $a > 1$
$x_0 = a$ dla $a \leq 1$
$x_{i+1} = x_i + (a/x_i^2 - x_i)/3$

Dokładność jest osiągnięta, jeśli $|x_i^3 - a| \leq \varepsilon * |a|$.
Napisz efektywną (wykorzystującą rekursję ogonową) funkcję *root3*, która dla zadanej liczby a znajduje pierwiastek trzeciego stopnia z dokładnością 10^{-15}.
Uwaga. Pamiętaj, że OCaml nie wykonuje automatycznie żadnych koercji typów.

4. Związ zmienną x z wartością 0 konstruując wzorce, do których mają się dopasować następujące wyrażenia:
a) [-2; -1; 0; 1; 2] b) [(1,2); (0,1)]
Np. dla wyrażenia (true,"hello",0) wymaganym wzorcem jest (_ ,_ ,x).

5. Zdefiniuj funkcję initSegment : 'a list * 'a list -> bool sprawdzającą w czasie liniowym, czy pierwsza lista stanowi początkowy segment drugiej listy. Każda lista jest swoim początkowym segmentem, lista pusta jest początkowym segmentem każdej listy.

6. a) Zdefiniuj funkcję replaceNth : 'a list * int* 'a -> 'a list, zastępującą n-ty element listy podaną wartością (pierwszy element ma numer 0), np.

replaceNth (['o'; 'l'; 'a'] ,1, 's') => ['o'; 's'; 'a']

Nie wykorzystuj żadnej funkcji bibliotecznej!
   b) Jaka jest złożoność obliczeniowa tej funkcji? Zilustruj rysunkiem (patrz wykład str.37-40) reprezentację obu list. Liniowa