

## Paradygmaty programowania

### Lista 4

**Zad. 1. [4p]** Napisać w Scala funkcję, wykorzystującą metodę połowienia przedziału, do wyznaczenia zadaną dokładnością pierwiastka  $n$ -tego stopnia ( $n > 0$ ) z liczby rzeczywistej  $a$ . Dokładność zostanie osiągnięta jeśli wartość bezwzględna z różnicy z dwóch sąsiednich przybliżeń pierwiastka będzie mniejsza od tej dokładności. W rozwiązaniu wykorzystać rekurencję ogonową.

Wskazówka 1. Metoda połowienia odcinka służy do znajdowania miejsc zerowych funkcji. W pierwszym kroku znajdowane są dwa punkty  $a$  i  $b$ , dla których wartości funkcji mają różne znaki. Następnie znajdowany jest środek przedziału  $c$  między wartościami  $a$  i  $b$  i wyznaczana jest wartość funkcji w punkcie  $c$ . Jeśli wartości funkcji w  $a$  i  $c$  mają różne znaki w następnym kroku brane są pod uwagę punkty  $a$  i  $c$  (znajdowany jest środek między  $a$  i  $c$ ), jeśli nie w następnym kroku brane są pod uwagę punkty  $c$  i  $b$  (znajdowany jest środek między  $c$  i  $b$ ). Liczba kroków zależy od założonej dokładności.

Wskazówka 2.  $a^{1/n} = x$  stąd  $a = x^n$  stąd  $f(x) = a - x^n = 0$ .

**Zad. 2. [3p]** Wykorzystując mechanizm dopasowania wzorca napisać w OCaml funkcję typu *float list* -> *float list*, (jedną!) stosującą rekurencję ogonową, która przekształca wejściową listę do postaci, w której każdy element listy zwielokrotniony o kolejne swoje potęgi do  $n$ -tej potęgi włącznie, gdzie  $n$  jest pozycją elementu w liście wejściowej.

**Zad. 3. [3p]** Wykorzystując mechanizm dopasowania wzorca napisać w Scala funkcję (jedną!) stosującą rekurencję ogonową usuwającą z obu końców zadanej listy liczb całkowitych po liczbie elementów równej zadanej wartości.