

Paradygmaty programowania

Lista 2

Zad. 1. [2+1p] Napisać w OCaml i Scala dowolne, działające tak samo, funkcje o podanych poniżej odpowiednio typach OCaml (bez wymuszania typu). Określić działanie funkcji.

a) *int * int -> int * int*

b) *float * float -> bool*

c) *'a list * int -> 'a list*

Zad. 2. [2p] Napisać w Scala funkcję sprawdzającą czy elementy danej listy liczb całkowitych są posortowane nierosnąco.

Zad 3. [5p] Korzystając z faktu, że nieskończony iloczyn

$$(0.5)^{1/2} * (0.5+0.5*(0.5)^{1/2})^{1/2} * (0.5 + 0.5*(0.5+0.5*(0.5)^{1/2})^{1/2})^{1/2} * \dots\dots\dots$$

ma wartość $2/\pi$ napisać w OCaml funkcję (jedną!) typu *float -> float* wyznaczającą wartość liczby π zadaną dokładnością. Dokładność zostanie osiągnięta jeśli wartość bezwzględna z różnicy z dwóch sąsiednich przybliżeń liczby π będzie mniejsza od tej dokładności. W rozwiązaniu wykorzystać rekurencję ogonową.