

PODSTAWY PROGRAMOWANIA DEKLARATYWNEGO

HASKELL

Ćwiczenia 3

Zadanie 1.

Napisać definicję 3-argumentowej funkcji wypisującej elementy rosnącego ciągu arytmetycznego o danym pierwszym i ostatnim elemencie oraz o danej różnicy.

- a) podać definicję rekurencyjną
- b) napisać definicję funkcji bez rekurencji

Np. ciąg 1 20 3 = [1,4,7,10,13,16,19]
ciąg 4 20 4 = [4,8,12,16,20]

Zadanie 2.

Napisać definicję 3-argumentowej funkcji, której wartością jest podlista zawierająca elementy od n do m ($m > n$).

- a) podać definicję rekurencyjną
- b) napisać definicję funkcji bez rekurencji

Np. wycinek 1 3 [5,6,7,8,9,10] = [5,6,7]
wycinek 5 10 [2,4 ..] = [10,12,14,16,18,20]
wycinek 5 10 ['a' ..] = "efghij"

Zadanie 3.

Napisać funkcję numerującą kolejne elementy listy począwszy od dowolnej liczby. Argumentami funkcji są: liczba i lista dowolnego typu, a wartością lista par, gdzie pierwszy element krotki jest numerem, a drugi elementem listy o tym numerze.

Np. numerowanie 1 [5,6,7,8,9] = [(1,5),(2,6),(3,7),(4,8),(5,9)]
numerowanie (-2) [10,12 .. 20] = [(-2,10),(-1,12),(0,14),(1,16),(2,18),(3,20)]
numerowanie 5 ['a' .. 'd'] = [(5,'a'),(6,'b'),(7,'c'),(8,'d')]

Zadania 4,5,6 - należy przypomnieć sobie definicje działań na zbiorach.

W Haskellu zbiory reprezentujemy za pomocą list. Pamiętajmy, że kolejność elementów w zbiorze nie jest istotna.

Zadanie 4.

Zakładamy, że listy $l1$ i $l2$ reprezentują zbiory. Zdefiniować funkcję, której wartością jest część wspólna zbiorów $l1$ i $l2$.

Np. iloczyn_z [1,2,3,4] [5,3,8,7,1] = [1,3]

Zadanie 5.

Zakładamy, że listy *l1* i *l2* reprezentują zbiory. Zdefiniować funkcję, której wartością jest suma zbiorów *l1* i *l2*.

Np. suma_z [1,2,3,4] [2,3,8,7,1] = [1,2,3,4,7,8]

Zadanie 6.

Zakładamy, że listy *l1* i *l2* reprezentują zbiory. Zdefiniować funkcję, której wartością jest różnica zbiorów *l1* i *l2*.

Np. roznica_z [1,2,3,4] [2,3,8,7] = [1,4]

Zadanie 7.

Zdefiniować polimorficzną funkcję, której wartością jest lista wszystkich podlist danej listy.

Np. powerlist [1,2,3] = [[],[1],[2],[3],[1,2],[1,3],[2,3],[1,2,3]]

Zadanie 8.

Zapisać kolejność obliczeń wartości każdej z funkcji (przykład na wykładzie):

- a) foldr (/) 2 [6,12,24,8]
- b) foldr (&&) True [1>2,3>2,5==5]
- c) foldr max 18 [3,6,12,4,55,11]
- d) foldr (\x y -> (x+y)/2) 54 [24,4,10,6]
- e) foldl (\x y -> (x+y)/2) 54 [2,4,10,6]
- f) foldl (/) 64 [4,2,4]
- g) foldl (\x y -> 2*x + y) 8 [1,2,3]

Zadanie 9.

Za pomocą **foldl** zdefiniować funkcję **nalezy** sprawdzającą, czy dany element należy do listy.

Zadanie 10.

Zdefiniować funkcję **map** (funkcji nadać nową inną nazwę) za pomocą foldl (za pomocą foldr – patrz wykład).

Zadanie 11.

Zdefiniować funkcję a) last, b) head, c) maksimum (największy element listy) za pomocą foldr1 lub foldl1.