

ДЕКЛАРАТИВНОЕ ПРОГРАММИРОВАНИЕ

Свертки

Что общего?

$\text{sum}' [] = 0$

$\text{sum}' (x:xs) = x + \text{sum}' xs$

$\text{product}' [] = 1$

$\text{product}' (x:xs) = x * \text{product}' xs$

$\text{concat}' [] = []$

$\text{concat}' (x:xs) = x ++ \text{concat}' xs$

Свернем их!

$\text{foldr}' \text{ op acc []} = \text{acc}$

$\text{foldr}' \text{ op acc (x:xs)} =$
 $\quad \text{x `op` (foldr' op acc xs)}$

$\text{sum''} = \text{foldr}' (+) 0$

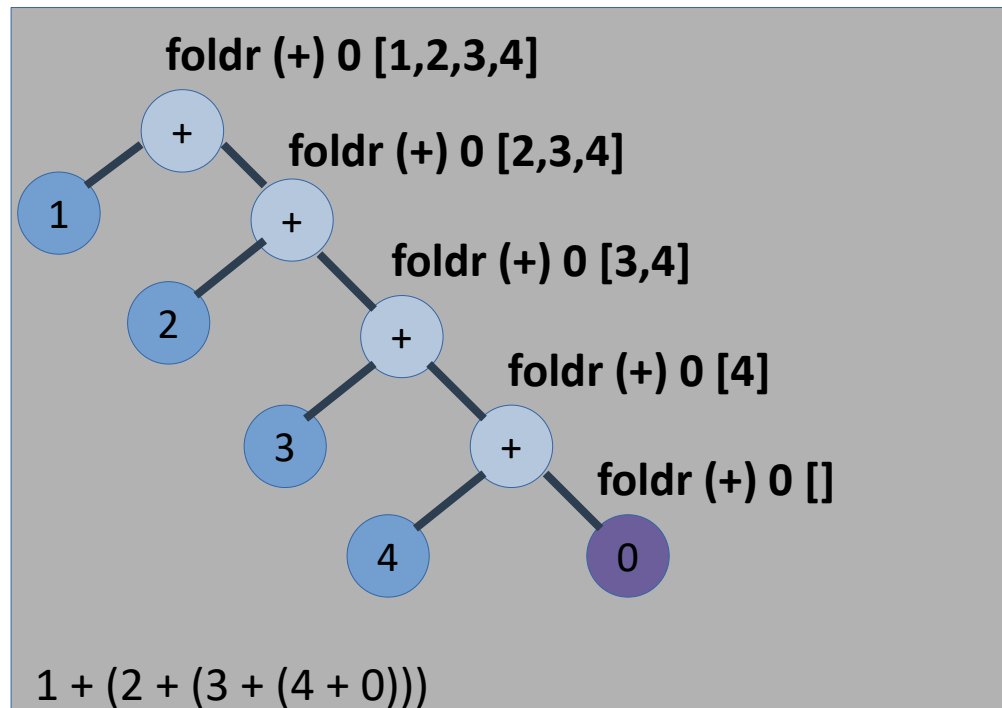
$\text{product''} = \text{foldr}' (*) 1$

$\text{concat''} = \text{foldr}' (++) []$

Свернем их!

`foldr' op acc [] = acc`

`foldr' op acc (x:xs) = x `op` (foldr' op acc xs)`



Правая свертка

`foldr f z [3,4,5,6]`

`f 3 (f 4 (f 5 (f 6 z)))`

`foldr (+) 0 [3,4,5,6]`

`3 + (4 + (5 + (6 + 0)))`

Упражнение

Используя функцию `foldr`, напишите реализацию функции `length'`, вычисляющей количество элементов в списке.

Упражнение

Посчитайте сумму квадратов элементов списка с использованием свертки

`sumSquares :: [Integer] -> Integer`

`sumSquares xs = foldr ? ?`

Упражнение

Посчитайте сумму квадратов положительных элементов списка с использованием свертки.

```
sumPosSquares :: [Integer] -> Integer
```

```
sumPosSquares xs = foldr ? ?
```


Правая vs. левая

`foldl' :: (b -> a -> b)`

`-> b -> [a] -> b`

`foldl' op acc [] = acc`

`foldl' op acc (x:xs) =`

`foldl' op (acc `op` x) xs`

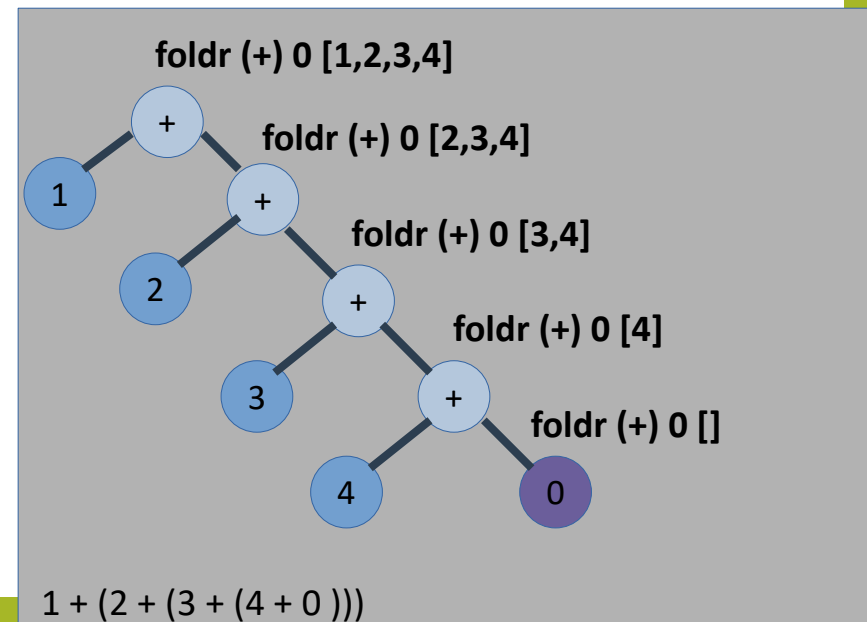
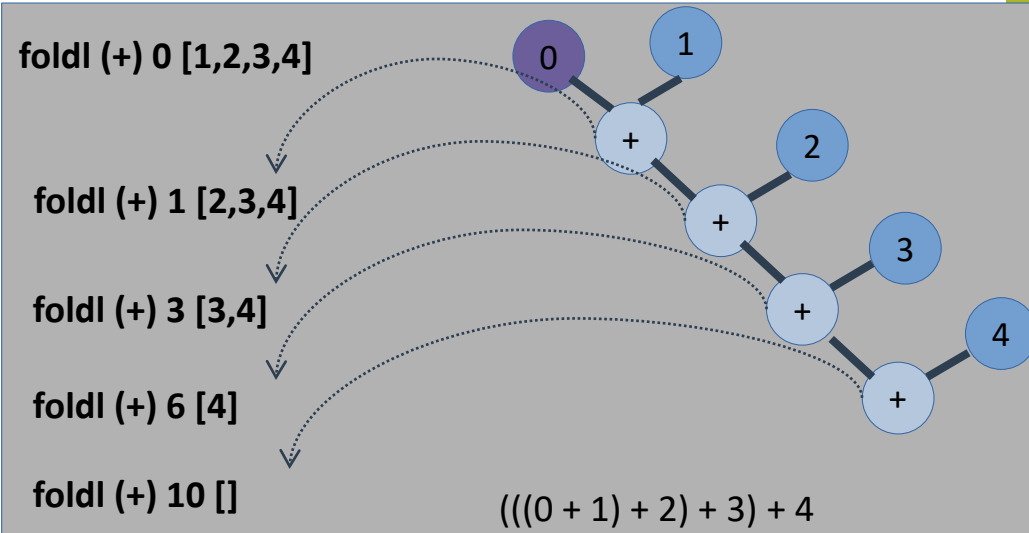
`foldr' :: (a -> b -> b)`

`-> b -> [a] -> b`

`foldr' op acc [] = acc`

`foldr' op acc (x:xs) =`

`x `op` foldr' op acc xs`



Правая vs. левая

foldr f z [3,4,5,6]

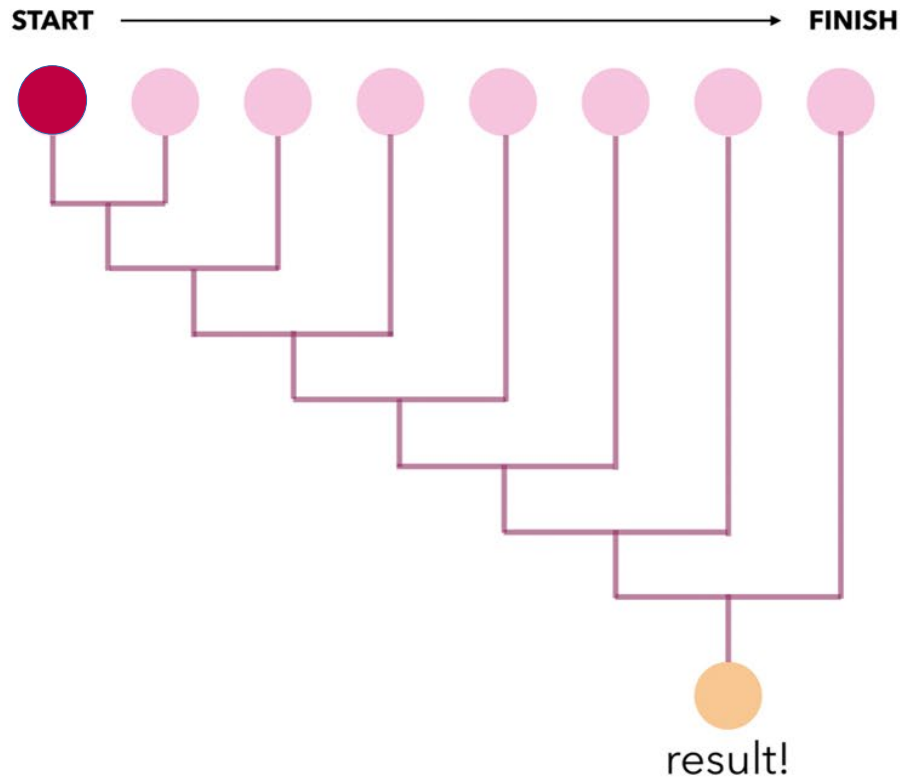
f 3 (f 4 (f 5 (f 6 z)))

foldl g z [3,4,5,6]

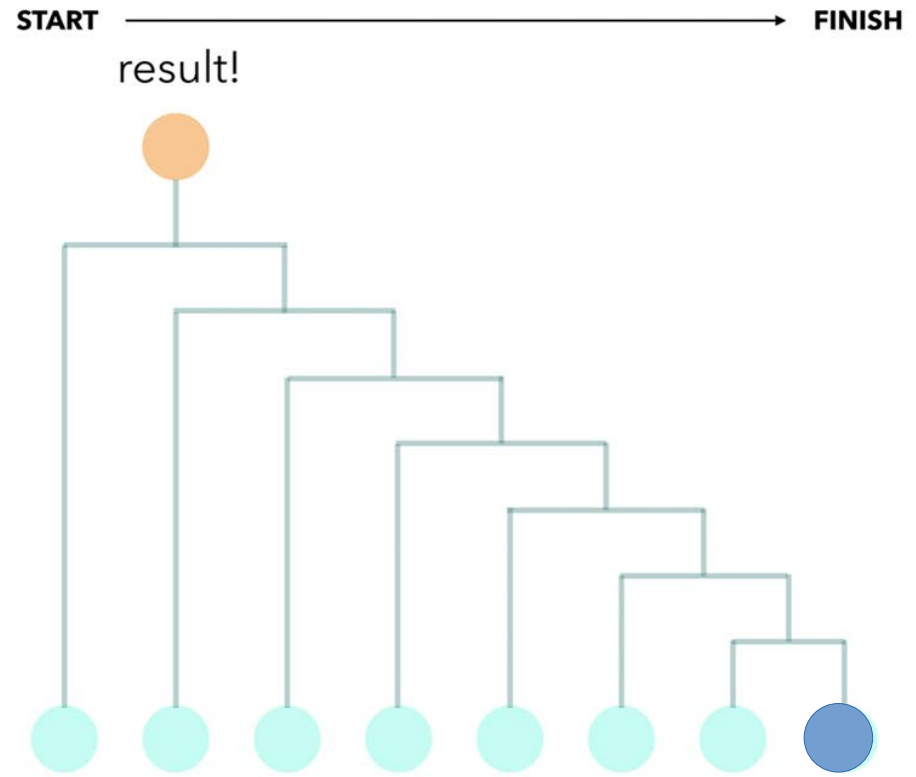
g (g (g (g z 3) 4) 5) 6

Правая vs. левая

foldl



foldr



Упражнение

При каком значении переменной x следующие два выражения примут одно и то же значение (отличное от неопределенного)?

- 1) `foldr (-) x [2,1,5]`
- 2) `foldl (-) x [2,1,5]`

Упражнения

- Напишите функцию `myMaximum`, возвращающую максимум среди всех элементов списка, используя однократный вызов функции свертки
- Реализуйте функцию `meanList`, которая находит среднее значение элементов списка, используя однократный вызов функции свертки

Упражнения

- Посчитайте сумму и произведение элементов списка с помощью одной свертки
- Реализуйте свою версию функции `reverse` используя свертку
- Реализуйте свою версию функции `last` с помощью `foldl1`

Упражнение*

Используя однократный вызов свертки, реализуйте функцию `evenOnly`, которая выбрасывает из списка элементы, стоящие на нечетных местах, оставляя только элементы на четных позициях (у головы списка – нечетная позиция).

```
GHCi> evenOnly [1..10]
```

```
[2,4,6,8,10]
```

```
GHCi> evenOnly ['a'..'z']
```

```
"bdfhjlnprtvxz"
```