

Haskell

Функційна мова

- *вирази* (синтаксичні терми)
- *значення* (абстрактні сутності, розгл. як відповіді/результати)
- Атоми $5, 'h', \lambda x \rightarrow x^2$
- Структури $[21, 2, 3], (1, "asd")$
- Редукція $expr1 \Rightarrow expr2, 1+1 \Rightarrow 2$

Haskell

- Сильна типізація
 - Типи (вирази типу)
 - Атомарні

Integer, Char, Integer->Integer

- Структуровані [Integer], (Char, Integer)

5 :: Integer

'a' :: Char

inc :: Integer -> Integer

[1,2,3] :: [Integer]

('b',4) :: (Char,Integer)

Поліморфні типи

- $(\forall a)[a]$ – сімейство типів, що складається з типів списку з елементами із a , для довільного типу a
- a – змінна типу
- $[a]$
- **система типів Хіндлі-Мілнера**
(передбачає існування та єдиність **основного типу** - найбільш загального типу, «який містить всі екземпляри виразу»)

1. **length :: [a] -> Integer**
2. **length [] = 0**
3. **length (x:xs) = 1 + length xs**

1 - оголошення сигнатури типу

2,3 – опис, декларація

Сигнатура

ім'я :: *тип*

$(:)$:: $a \rightarrow [a] \rightarrow [a]$

$(++)$:: $[a] \rightarrow [a] \rightarrow [a]$

head ::

tail ::

take, drop :: $Int \rightarrow [a] \rightarrow [a]$

sum ::

Оператори

Оператор	Пріоритет	Асоціативність	Опис
.	9	Right	Function composition
!!	9	Left	List indexing
^, ^^, **	8	Right	Exponentiation (integer, fractional, and floating-point)
*, /	7	Left	Multiplication, Division
+, -	6	Left	Addition, Subtraction
:	5	Right	Cons (list construction)
++	5	Right	List Concatenation
`elem`, `notElem`	4	Left	List Membership
==, /=, <, <=, >=, >	4	Left	Equals, Not-equals, and other

Оператори

Оператор	Пріоритет	Асоціативність	Опис
&&	3	Right	Logical And
	2	Right	Logical Or
>>, >>=	1	Left	Monadic Bind, (piping value to next function)
=<<	1	Right	Reverse Monadic Bind
\$	0	Right	Infix Function
Application	(same as "f x", but right-associative instead of left)		

Списки

[3,1,4,1,59,265]

[Int]

['a', 'b']

[Char]

[(3,'a'), (4,'m')]

[(Int, Char)]

Конструктори

[]

:

Точкова нотація

(dot-нотація)

[1..5] [1,2,3,4,5]
[1,3..6] [1,3,5]
[2,5..12] [2,5,8,11]
[5..1] []
[5,4..1] [5,4,3,2,1]
[3..] [3,4,5,6,...]
['a'..'d'] ['a','b','c','d']
['a','c'..'g'] ['a','c','e','g']

reverse [10,2,3,4]

[4,3,2,10]

words "asdf fg jk"

["asdf","fg","jk"]

unwords

lines "this is \n a string"

["this is","a string"]

unlines

last head tail

[1,2] ++ [3,4]

[1,2,3,4]

[1..5]!!0

1

concat [[1,2],[3,4]]

[1,2,3,4]

```
>init [1..5]  
[1,2,3,4]
```

```
>last [1..5]  
5
```

```
> elem 3 [1..5]  
True
```

```
> notElem 7 [1..5]  
True
```

```
> take 3 [1..5]  
[1,2,3]
```

```
> take 3 [1..]  
[1,2,3]
```

```
> drop 3 [1..5]  
[4,5]
```

Відображення (отображение, mapping)

$\text{map} :: (a \rightarrow b) \rightarrow [a] \rightarrow [b]$

$\text{map } f [] = []$

$\text{map } f (x:xs) = f x : \text{map } f xs$

$\text{map } (^2) [1,2,3] \quad [1,4,9]$

Генератори списків (List comprehension)

`[expr | listGenerators, listPredicates]`

`[f x | x <- xs]`

`[(x,y) | x <- xs, y <- ys]`

`[x | x<-[-5..5],x>2,x<=4]` `[3,4]`

`[x*x | x <- [1..3]]` `[1,4,9]`

`[sqr x | x <- [1..3]]` `[1,4,9]`

`[(x, y) | x <- [1..3], y <- "ab"]`

`[(1,'a'),(1,'b'),(2,'a'),(2,'b'),(3,'a'),(3,'b')]`

`[(x,y) | x <- [1..4], even x, y <- "ab"]`

`[(2,'a'),(2,'b'),(4,'a'),(4,'b')]`

$\text{map} :: (a \rightarrow b) \rightarrow [a] \rightarrow [b]$

$\text{map } f \text{ xs} = [f \ x \mid x \leftarrow \text{xs}]$

`filter :: (a -> Bool) -> [a] -> [a]`

`filter p xs = [x | x <- xs, p x]`

`filter (<3) [1..5]` `[1,2]`

`filter (\x -> x^3 == x) [-5..5]` `[-1,0,1]`

Case-вирази

$$\begin{aligned} f\ x_1\ x_2\ \dots\ x_k &= \text{case } (x_1, \dots, x_k) \text{ of} \\ &\quad (p_{11}, \dots, p_{1k}) \rightarrow e_1 \\ &\quad \dots \\ &\quad (p_{n1}, \dots, p_{nk}) \rightarrow e_n \end{aligned}$$

length :: [a] -> Int

$$\begin{aligned} \text{length } xs &= \text{case } xs \text{ of } [] \rightarrow 0 \\ &\quad (x:xs) \rightarrow 1 + \text{length } xs \end{aligned}$$

УМОВНІ ВИРАЗИ

if e1 then e2 else e3

Main Lib> let x=3 ; y=2 in if x>y then x else y

3

*Main Lib> it

3

Зв'язування

let *bindings* in *expression*
expression where *bindings*

f4 *x y* = *a+b-c*

where

a=*x*

b=*y*

c=10

УМОВНІ вирази

if b then $e1$ else $e2$

case b of

True $\rightarrow e1$

False $\rightarrow e2$

УМОВНІ ВИРАЗИ

```
f1 = let x=3
```

```
    y=2
```

```
    in if x > y then x else y
```

```
*Main Lib> f1
```

```
3
```

```
*Main Lib> :t f1
```

```
f1 :: Integer
```

УМОВНІ вирази

$f2\ x\ y = \text{if } x > y \text{ then } x \text{ else } y$

*Main Lib> f2 3 5

5

*Main Lib> f2 'a' 'b'

'b'

*Main Lib> f2 ('a',3) ('b',5)

('b',5)

*Main Lib> f2 ('a',3) ('b',1)

('b',1)

УМОВНІ ВИРАЗИ

$f2\ x\ y = \text{if } x > y \text{ then } x \text{ else } y$

```
*Main Lib> f2 False True
```

```
True
```

```
*Main Lib> :t f2
```

```
f2 :: Ord t => t -> t -> t
```

```
*Main Lib>
```

УМОВНІ ВИРАЗИ

```
f3:: Int -> Int -> Int
```

```
f3 x y = if x > y then x else y
```

```
*Main Lib> f3 3 5
```

```
5
```

```
*Main Lib> f3 False True
```

```
<interactive>:14:4: error:
```

УМОВНІ вирази

$f2\ x\ y = \text{if } x > y \text{ then } x \text{ else } y$

```
*Main Lib> f2 False True
```

```
True
```

```
*Main Lib> :t f2
```

```
f2 :: Ord t => t -> t -> t
```

```
*Main Lib>
```


УМОВНІ ВИРАЗИ

```
f3:: Int -> Int -> Int
```

```
f3 x y = if x > y then x else y
```

```
*Main Lib> f3 'a' 'b'
```

```
<interactive>:15:4: error:
```