

Currying

доц. Атанас Семерджиев

1

Currying

Функция на n-аргумента може да се сведе до поредица от n-функции на един аргумент.

```
(define f  
  (lambda (x y z)  
    (+ x y z)))
```

```
(define g  
  (lambda (x)  
    (lambda (y)  
      (lambda (z)  
        (+ x y z))))))
```

2

2

```
(define g  
  (lambda (x)  
    (lambda (y)  
      (lambda (z)  
        (+ x y z))))))
```

(g 10)

(lambda (y)
 (lambda (z) (+ 10 y z)))

((g 10) 20)

(lambda (z) (+ 10 20 z))

((((g 10) 20) 30)

60

3

3

Currying в Haskell

```
f :: Int -> Int -> Int  
f x y = x+y
```



```
f :: Int -> (Int -> Int)  
f x y = x+y
```

4

4

Partially Applied Functions

В Хаскел може да се извърши така нареченото частично прилагане на функция:

```
mySum x y = x + y
```

```
add1 = mySum 1
```

```
add2 = mySum 2
```

```
add3 = mySum 3
```

```
var = add1 10
```

```
(define (mySum x y) (+ x y))
```

```
(define add1 (lambda (y) (mySum 1 y)))
```

```
(define add2 (lambda (y) (mySum 2 y)))
```

```
(define add3 (lambda (y) (mySum 3 y)))
```

```
(define var (add1 10))
```

5

5

Partially Applied Functions

```
mySum x y = x + y
```

```
map (mySum 1) [1,2,3,4,5]
```

```
(define (mySum x y) (+ x y))
```

```
(map (lambda (y) (mySum 1 y))  
     '(1 2 3 4 5))
```

6

6

Частично прилагане на инфиксни операции

`div10 x = x / 10``div10 = (/ 10)``div10by x = 10 / x``div10by = (10 /)``between1and100 x = x `elem` [1..100]``between1and100 = (`elem` [1..100])`

7

7

Частично прилагане на инфиксни операции

```
> filter (`elem` [10..20]) [1,5,10,100,20,15]  
[10,20,15]
```

```
> map (+1) [1,2,3,4,5]  
[2,3,4,5,6]
```

8

8

Частично прилагане на инфиксни операции

```
prepend10 = (10:)  
> prepend10 [20,30]      -- [10,20,30]  
  
append123 = (: [1,2,3])  
> append123 0            -- [0,1,2,3]  
  
subtract10 = (-10)  
> subtract10 100         -- Грешка!
```

9

9

Относно „минус“

```
((-) 1)      -- редът има значение!  
             -- Резултат:  
             --   функция на един аргумент x,  
             --   която пресмята (1 - x)  
  
negate 9     -- Резултат: числото -9  
  
subtract 5   -- Резултат:  
             --   функция на един аргумент x,  
             --   която пресмята (x - 5)
```

10

10

