
Haskell — Definition of higher-order functions (1)

P90677_en

This problem explores the definition of high-order functions on lists. Implement the following functions that work as the original Haskell functions without using the original function eachself (i.e., you cannot use *foldl* to implement *myFoldl* but you can use it to implement *myAll*). Additionally, you can only use recursion to implement *myFoldl*, *myFoldr*, *myIterate*, *myUntil* and *myZip*.

1. $myFoldl :: (a \rightarrow b \rightarrow a) \rightarrow a \rightarrow [b] \rightarrow a$
2. $myFoldr :: (a \rightarrow b \rightarrow b) \rightarrow b \rightarrow [a] \rightarrow b$
3. $myIterate :: (a \rightarrow a) \rightarrow a \rightarrow [a]$
4. $myUntil :: (a \rightarrow \mathbf{Bool}) \rightarrow (a \rightarrow a) \rightarrow a \rightarrow a$
5. $myMap :: (a \rightarrow b) \rightarrow [a] \rightarrow [b]$
6. $myFilter :: (a \rightarrow \mathbf{Bool}) \rightarrow [a] \rightarrow [a]$
7. $myAll :: (a \rightarrow \mathbf{Bool}) \rightarrow [a] \rightarrow \mathbf{Bool}$
8. $myAny :: (a \rightarrow \mathbf{Bool}) \rightarrow [a] \rightarrow \mathbf{Bool}$
9. $myZip :: [a] \rightarrow [b] \rightarrow [(a, b)]$
10. $myZipWith :: (a \rightarrow b \rightarrow c) \rightarrow [a] \rightarrow [b] \rightarrow [c]$

Scoring

Each function scores 10 points.

Sample input

```
myFoldl (+) 1 [1..5]
myFoldr (+) 1 [1..5]
take 10 $ myIterate (*2) 1
myUntil (>100) (*2) 1
myMap ("la "++) ["joana", "mireia"]
myFilter odd [1..10]
myAll odd [1,3,5,3,1]
myAny odd [2,4,6,8,10]
myZip [1..4] [1..3]
myZipWith (+) [1..4] [1..3]
```

Sample output

```
16
16
[1,2,4,8,16,32,64,128,256,512]
128
["la joana","la mireia"]
[1,3,5,7,9]
```

True

False

[(1, 1), (2, 2), (3, 3)]

[2, 4, 6]

Problem information

Author : Albert Rubio / Jordi Petit

Translator : Jordi Petit

Generation : 2018-03-07 15:08:27

© *Jutge.org*, 2006–2018.

<https://jutge.org>