Functional Programming
Using Haskell

Bruno Giao

December 8, 2022

1 Exercises

Let us consider the following functions:

- length (size),
- (++) (merge),
- reverse (reverse),
- nub (removal of repeats),
- words (words of a phrase),
- unwords (inverse of previous),
- sort (sort),
- (==) (equality),
- lines (lines of a text),
- unlines (inverse of previous),
- take (get prefix),
- drop (get sufix),
- head
- tail
- last
- init
- map
- zip
- fst
- snd

- succ
- pred

1. Run the following instructions and describe what they do.

    values=[1,2,3,2,10,40,30]
    head values
    tail values
    last values
    init values

2. Define the following functions in haskell.

    (a) length
    (b) (++)
    (c) reverse
    (d) head
    (e) tail
    (f) take
    (g) drop

    NOTE: To make some functions easier consider the following notation:

    function input| condition = output
                  | otherwise = output

    To represent programming with multiple conditions or:

    function input = if condition then output else output

    To represent programming with one condition.

3. Run the following commands:

    (a) succ 1
    (b) pred 'B'
    (c) succ 'A'
    (d) map (succ [1,2,3,4,5])
    (e) map (pred "A,B,C,D,E")

    What do these functions do?

4. Define the following functions in Haskell:

    (a) succ
    (b) pred
    (c) map

5. Run the following commands:

   (a) zip [1,2,3] ”ABC”

   (b) map fst (zip [1,2,3] ”ABC”)

   (c) map snd (zip [1,2,3] ”ABC”)

   What do they do.

6. Define the functions introduced in the previous question.

2 Summary

Using and interpreting Pre-Defined functions in GHCi.
Definition of pre-defined functions.