

My Report

Me

Autumn 2016

Abstract

We give a toy example of a report in *literate programming* style. The main advantage of this is that source code and documentation can be written and presented next to each other. We use the listings package to typeset Haskell source code nicely.

Contents

1	How to use this?	2
2	The most basic library	2
3	Wrapping it up in an executable	3
4	Conclusion	3
	Bibliography	3

1 How to use this?

To generate the PDF, open the report.tex in your favorite LaTeX editor and hit compile, or manually do this:

```
pdflatex report
bibtex report
pdflatex report
pdflatex report
```

You should have stack installed (see <http://haskellstack.org/>) and open a terminal in the same folder.

To compile everything: `stack build`.

To open ghci and play with your code: `stack ghci`

To run the executable generated from Section 3: `stack exec myprogram`

2 The most basic library

This section describes a module which we will import later on.

```
module Basics where

thenumbers :: [Integer]
thenumbers = [1..]

somenumbers :: [Integer]
somenumbers = take 10 thenumbers
```

We can interrupt the code anywhere we want.

```
funnyfunction :: Integer -> Integer
funnyfunction 0 = 42
```

Even in between cases, like here. It's always good to cite something [Knu11].

```
funnyfunction n | even n      = funnyfunction (n-1)
                | otherwise = n*100
```

That's it, for now.

3 Wrapping it up in an executable

We will now use the library from Section 2 in a program.

```
module Main where

import Basics

main :: IO ()
main = do
  putStrLn "Hello!"
  print somenumbers
  print (map funnyfunction somenumbers)
  putStrLn "GoodBye"
```

The output of the program is:

```
Hello!
[1,2,3,4,5,6,7,8,9,10]
[100,100,300,300,500,500,700,700,900,900]
GoodBye
```

Note that the above `showCode` block is only shown, but it gets ignored by the Haskell compiler.

4 Conclusion

Finally, we can see that [LW13] is a nice paper.

References

- [Knu11] Donald E. Knuth. *The Art of Computer Programming. Combinatorial Algorithms, Part 1*, volume 4A. Addison-Wesley Professional, 2011.
- [LW13] Fenrong Liu and Yanjing Wang. Reasoning about agent types and the hardest logic puzzle ever. *Minds and Machines*, 23(1):123–161, 2013.