Marius Furter
Institut für Mathematik
Universität Zürich

FS 2025

Logic and Foundation with Haskell

Exercise sheet 1

Exercise 1. Learn about types, type variables and typeclasses by watching this video (30 minutes).

Exercise 2. Implement the following functions. You can check https://hoogle.haskell.org to look for existing functions. You can use everything that is in `base`.

`fl :: [Double] -> [Int]` that floors (rounds down) each number in a list.
`d11 :: Int -> Bool` that takes a number and checks if it is divisible by 11.
`toLowerCase :: String -> String` that converts a string to lower case.

Exercise 3. Implement a function `map' :: (a -> b) -> [a] -> [b]` that applies a function to each element of a list.

Exercise 4. Implement a function `filter' :: Eq a => (a -> Bool) -> [a] -> [a]` that filters a list according to a predicate `p :: a -> Bool`.

Exercise 5. Implement the functions that are given by the following type signatures:

```
i :: (a,b) -> (b,a)
ii :: a -> (a -> b) -> b
iii :: (a -> b) -> (b -> c) -> a -> c
```

Exercise 6. Think of at least two functions that satisfy each of the following type signatures. After you do, look them up at https://hoogle.haskell.org.

```
f :: Eq a => a -> [a] -> Bool
g :: (a -> b) -> [a] -> [b]
h :: Ord a => [a] -> [a]
```