

Tutorium Funktionale Programmierung 2019

Part 8 - Higher-Order-Functions,
Partial Application, “(.) and (\$)” and λ Abstraction

VO - Part 4

Benedikt Dornauer, 23.11.2019



- Which topics should be discussed once more?
 - Questions.
 - Wishes for the tutorialum 2019-12-3.
- ▶ Arsnova
- ▶ Session:
- 50 09 60 50**

Partial Application

(e.g. 1) *drop3* = *drop 3*

(e.g. 2) *multi2* = *(*2)*

- ▶ Produce another function with smaller arity

Higher-Order-Functions

- ▶ “functions are just like any other value”
- ▶ takes other functions as input
- ▶ OR return a function as result
- ▶ used for abstraction

Ref. : https://en.wikibooks.org/wiki/Haskell/Higher-order_functions

8.1. Exercise: Higher-Order-Functions

You want receive the absolute value of a given function at a specific point. Something like *myAbs*: $(f, x) \rightarrow |f(x)|$. A function *f* has the following type

f :: Double -> Double

Create a **high-order-function** for the *myAbs* and test it with $g\ x = x^2 - 4$ and $x = 0$ as well as $x = 3$.

8.2. Exercise: Function Composition and Function Application

Use the minimal amount of parentheses for

`(null (tail (tail [True, False]))) == (not (not False))`

- ▶ Use `(.)`
- ▶ Use `($)`

λ -Abstraction

e.g. `average = \ x y z -> (abs(x)+abs(y)+abs(z))/3`

- ▶ anonymous function - “have no name”
- ▶ Written `\ ... ->`



8.3. exercise: λ -Abstraction, Higher-Order-Functions and Partial Application

A customer of a bank can have several passbooks. Internally the bank infrastructure all passbooks of a customer are digitally stored **as a list**. Each element of the list expresses a specific value of a passbook e.g. 70.1000€. Also use the given type synonyms.

`[70.1000, 120004.412, 13 000]`
└───┬───┬───┘
First Second Third
passbook passbook passbook

```
type Rate = Double
type Value = Double
type Passbook = [Value]
```




8.3. exercise: λ -Abstraction, Higher-Order-Functions and Partial Application CONTINUED

1. Express a function *myMap* that takes a function $f :: a \rightarrow b$ and a list of type `[a]`. The function f is applied to each element of the list. (**Higher-order-function**)
2. Create a function *newRate* that takes as input a list of passbooks, an annual interest rate and a returns a list with the calculated value. You should use the function *myMap* in combination with **λ abstraction**.
3. Define a function with a fixed annual rate of 0.03 called *annualRate003*. Use **partial application**.

Questions? Need help? Feedback? etc.

▶ benedikt.dornauer@student.uibk.ac.at