

Übungsblatt 7  
Funktionale Programmierung, WS-2018/19  
M. Esponda , R.Rojas, O.Wiese  
Abgabe: 17.12.2018, 10 Uhr

1) (2 Punkte) Welche der folgenden Lambdafunktionen sind Kombinatoren?

$(\lambda x.xy)$   
 $(\lambda y.(\lambda x.xx)(\lambda x.xx))$   
 $(\lambda ab.abb)$

2) (4 Punkte) Reduzieren Sie folgende SKI-Ausdrücke:

- a)  $SII(SII)$
- b)  $S(K(SI))KIK$

3) (8 Punkte) Sei folgenden Datentyp gegeben

`data Expr = Addi Expr Expr | Mult Expr Expr | Sub Expr Expr | Divi Expr Expr | Numb Float  
deriving (Show,Eq)`

Schreiben Sie eine Funktion „eval“

`eval::Expr->Float`

die eine Expr numerisch auswertet und testen Sie sie.

4) (6 Punkte) In der Vorlesung haben wir folgende Haskell-Definitionen besprochen:

`data Expr = App Expr Expr | S | K | I | Var String | Lam String Expr deriving (Show,Eq)`

-----  
`transform (Lam x y) = (eliminate x y)`

`eliminate x S = App K S  
eliminate x K = App K K  
eliminate x I = App K I  
eliminate x (Var y)  
    | x==y = I  
    | otherwise = (App K (Var y))  
eliminate x (Lam y z) = eliminate x (eliminate y z)`

$\text{eliminate } x \text{ (App } y \text{ } z) = (\text{App } (\text{App } S \text{ (eliminate } x \text{ } y)) \text{ (eliminate } x \text{ } z))$

---

Schreiben Sie folgende Lambda-Funktionen in Haskell mit Hilfe von diesem Datentyp:

$(\backslash sz.z)$  (die Null)

$(\backslash wxy.x(wxy))$  (die Nachfolgerfunktion)

Testen Sie die Funktion „transform“ und zeigen Sie die Ausgabe.