Westfälische Hochschule

Fachbereich Informatik und Kommunikation
Prof. Dr. Marcel Luis

7. April 2016

Aufgabenblatt 1 zu Funktionale Programmierung

Aufgabe 1.1 (Übung)

Definieren Sie eine Funktion $uncurry$, die zu einer Funktion $f : a \to b \to c$ in Curry-Form die entsprechende Funktion mit Tupel-Argument liefert. Geben Sie den Typ von $uncurry$ an und überprüfen Sie ihn, indem Sie aus der definierenden Gleichung den Typ herleiten.

Aufgabe 1.2 (Übung)

Geben Sie für folgende Funktionen geeignete polymorphe Typen an, sofern dies möglich ist.

```
flip f x y  = f y x
strange f g = g (f g)
stranger f  = f f
```

Aufgabe 1.3 (Übung)

Es seien `f` und `g` Funktionen mit folgenden Typen:

```
f :: c -> d
g :: a -> (b -> c)
```

Die Funktion `h` sei durch folgende Regel definiert:

```
h x y = f (g x y)
```

Welche der folgenden Aussagen ist korrekt?

$$\begin{aligned} h &= f \cdot g \\ h\,x &= f \cdot (g\,x) \\ h\,x\,y &= (f \cdot g)\,x\,y \end{aligned}$$

Aufgabe 1.4 (Praktikum)

1. Definieren Sie eine Funktion $sum1$ in Curry-Form, die für eine Funktion f und zwei natürliche Zahlen a und b den Wert des folgenden Ausdrucks berechnet:

$$\sum_{i=a}^{b} f\ i$$

2. Definieren Sie nun eine Funktion $sum2$, die für eine Funktion f mit *zwei* Argumenten und natürliche Zahlen n und m den Wert des folgendes Ausdrucks berechnet:

$$\sum_{i=0}^{n} \sum_{j=0}^{m} f\ i\ j$$

   f sei hierbei in Curry-Form gegeben. Wenn Sie sich zur Definition von $sum2$ auf $sum1$ abstützen, kommen Sie ohne Rekursion aus.

Aufgabe 1.5 (Praktikum)

Die Folge der Fibonacci-Zahlen $f_0, f_1, \ldots$ ist durch die Gleichungen $f_0 = 0$, $f_1 = 1$ und $f_{n+2} = f_n + f_{n+1}$ für alle $n \geq 0$ definiert.

Definieren Sie eine Funktion fib, die zu einem Argument n den Wert f_n liefert.