

ALP I: Funktionale Programmierung

Bearbeiter: A. Rudolph

Tutor: Stephanie Hoffmann

Tutorium 06

Abgabe: bis Montag, den 27. Januar 2020, 10:10 Uhr

1. **Aufgabe** (4 Punkte) Aus der Aufgabe wird nicht wirklich ersichtlich welche Ausgaben die Funktion wann machen soll, aber ich nehme mal an es ist so wie es hintereinander da steht. Ist also Liste $A < \text{Liste } B$, erhalten wir 1. Sind beide gleich, bekommen wir 0 und ist $B < A$ erhalten wir -1. Außerdem gehe ich davon aus, dass die Lambda Ausdrücke für die Zahlen 0, 1 und -1 bekannt und trivial sind. Zusätzlich wurden auch $\{\text{NIL}\}$ und $\{\text{TAIL}\}$ in der Vorlesung gegeben und deren Funktionsweise ist klar, also wird hier nicht ausgeführt, wie sie aussehen, sondern sie werden einfach angenommen.

$$\leq \equiv (Y(\lambda rab. \{\text{NIL}\} \ a \ ((\{\text{NIL}\} \ b) \ 0 \ 1) \ ((\{\text{NIL}\} \ b) \ (\{\text{NIL}\} \ a) \ 0 \ -1) \ (r \ (\{\text{TAIL}\} \ a) \ (\{\text{TAIL}\} \ b))))$$

2. **Aufgabe** (8 Punkte) Hier kommt nicht so ein langer Text. Wir nehmen an, dass NIL (Testet ob Liste leer ist), HEAD (Erstes Element einer Liste), TAIL (Rest einer Liste), E (Gleichheitsoperator und T, F (Wahrheitswerte) bekannt sind.

$$(a) \ \{\text{ELEM}\} \equiv (Y(\lambda rxa. \{\text{NIL}\} \ a \ F \ ((E \ x \ (\{\text{HEAD}\} \ a)) \ T \ (r \ x \ (\{\text{TAIL}\} \ a)))))$$

$$(b) \ \{\text{REMOVE}\} \equiv (Y(\lambda rab. \{\text{NIL}\} \ a \ (T, -, -) \ ((E \ x \ \{\text{HEAD}\} \ a) \ (\{\text{TAIL}\} \ a) \ (F, (\{\text{HEAD}\} \ a), (r \ x \ \{\text{TAIL}\} \ a))))$$

3. **Aufgabe** (7 Punkte) Falls diese Funktionen noch getestet werden sollen, sind sie in der beiliegenden Haskell Datei zu finden.

$$(a) \ (2^{**}) \equiv \backslash x \rightarrow x * x * 1.0$$

Erklärung: $x * x$ simuliert die Zweierpotenz und das $* 1.0$ wandelt den Integer in ein Double um, da (2^{**}) immer einen Double ausgibt.

$$(f.g.h) \equiv \backslash x \rightarrow f(g(h(x)))$$

Erklärung: $(f.g.h)$ ist einfach nur die Funktion f angewendet auf das Ergebnis von g angewendet auf das Ergebnis von $h(x)$

$$(b) \ \text{reverse' } ys = \text{foldl } (\backslash xs \ x \rightarrow x:xs) \ [] \ ys$$

Erklärung: Die Eingabeliste wird gefaltet und die anonyme Faltungsfunktion Nimmt die Liste und das letzte Element und setzt das an den Anfang der Restliste

4. **Aufgabe** (5 Punkte) Warum auch immer das mit der fix Funktion gemacht werden sollte, ich bekomme es nicht hin. Also (in der Hoffnung auf ein paar Punkte) hier eine anonyme Funktion, die die Collatz Folge genau so berechnet, wie gefordert, aber ohne fix:

```
collatz = \x -> if (x == 1) then [1] else (if (mod x 2 == 0) then x:(collatz (div x 2))  
else x:(collatz (x * 3 + 1)))
```