Programmierung

Bitte beachten Sie die Übungsverlegungen im Zuge des Dies academicus am 22. Mai.

Aufgabe 1 (AGS 12.4.29 $\star$)

(a) Geben Sie einen Kombinator A an, so dass $A\ t\ s\ u \Rightarrow^* s$ für alle Lambdaterme t, s und u.

(b) Geben Sie einen Kombinator B an, so dass $B\ t\ s \Rightarrow^* s\ t$ für alle Lambdaterme t und s.

(c) Geben Sie einen Kombinator C an, so dass $C\ C \Rightarrow_\beta C\ C$.

(d) Geben Sie einen Kombinator D an, so dass $D \Rightarrow_\beta D$.

(e) Geben Sie einen Kombinator E an, so dass $E\ E\ t \Rightarrow^* E\ t\ E$ für jeden Lambdaterm t.

Aufgabe 2 (AGS 12.4.36)

Gegeben sei der λ-Term $\langle pow \rangle = \Big(\lambda nfz.\ n\ (\lambda gx.\ g\,(g\,x))\ f\ z\Big)$.

(a) Berechnen Sie die Normalform von $\langle pow \rangle\ \langle 2 \rangle$. Sie dürfen Zwischenschritte weglassen.

(b) Welche mathematische Funktion berechnet $\langle pow \rangle$?

(c) *Zusatzaufgabe:* Verallgemeinern Sie $\langle pow \rangle$.

Aufgabe 3 (AGS 12.4.32)

(a) Berechnen Sie die Normalform des λ-Terms $(\lambda fx.\ f\,f\,x)\ (\lambda y.\ x)\ z$, indem Sie ihn *schrittweise* reduzieren. Geben Sie dabei vor jedem Schritt für die relevanten Teilausdrücke die Mengen der gebunden bzw. frei vorkommenden Variablen an.

(b) Gegeben sei der λ-Term

$$\langle F \rangle = \Bigg(\lambda fxyz.\langle ite \rangle\ (\langle iszero \rangle\ (\langle sub \rangle\ x\,y))\ (\langle add \rangle\ y\,z)$$
$$\Big(\langle succ \rangle\ \ (f\,(\langle pred \rangle\ x)\,(\langle succ \rangle\ y)\,(\langle mult \rangle\ \langle 2 \rangle\ z))\Big)\Bigg)\ .$$

Berechnen Sie schrittweise die Normalform des Terms $\langle Y \rangle\langle F \rangle\langle 6 \rangle\langle 5 \rangle\langle 3 \rangle$. Schreiben Sie für jeden Aufruf von $\langle F \rangle$ jeweils zwei Zeilen: eine in der Sie die Werte der Parameter des Aufrufs protokollieren, und eine in der Sie ihre Auswertung skizzieren. Führen Sie im Rechenprozess zweckmäßige Abkürzungen der λ-Terme ein.

(c) Gegeben sei die folgende Haskell-Funktion:

```
g :: Int -> Int -> Int
g 0 y = 2 * (y + 1)
g x 0 = 2 * (x + 1)
g x y = 4 + g (x - 1) (y - 1)
```

Geben Sie einen λ-Term $\langle G \rangle$ an, so dass `g` $= \langle Y \rangle\langle G \rangle$ gilt. Sie dürfen dabei die in der Vorlesung vorgestellten Terme nutzen.

https://www.orchid.inf.tu-dresden.de *thomas.ruprecht@tu-dresden.de*

Zusatzaufgabe 1 (AGS 12.4.5)

(a) Berechnen Sie schrittweise die Normalform des λ-Terms $(\lambda zx.z\ x\ (\lambda y.y\ x))\ (\lambda y.z\ x)\ (\lambda z.z)$.

(b) Eine Funktion $f : \mathbb{N}^+ \times \mathbb{N}^+ \to \mathbb{N}^+$ sei wie folgt definiert:

$$
\begin{aligned}
f(x,y) &= 2 \cdot y && \text{für} \quad x = 1\\
f(x,y) &= y + (x+1) \cdot f(x-1, x+y) && \text{für alle } x \geq 2
\end{aligned}
$$

Geben Sie einen λ-Term $\langle F\rangle$ an, sodass $f = \langle Y\rangle\langle F\rangle$ gilt.

(c) Gegeben sei der Kombinator $\langle G\rangle = \Big(\lambda gxy.\ \langle \mathit{ite}\rangle\ (\langle \mathit{iszero}\rangle\ y)\ (\langle \mathit{succ}\rangle\ x)\ (g\,x\,(\langle \mathit{pred}\rangle\ y))\Big)$.
Berechnen Sie $\langle Y\rangle\langle G\rangle\langle 3\rangle\langle 0\rangle$. Dokumentieren Sie die Berechnung wie üblich.

Zusatzaufgabe 2 (AGS 12.3.28)

```
data IntTree = Node Int [IntTree]

yield :: IntTree -> [Int]
yield (Node i []) = [i]
yield (Node i ts) = concat (map yield ts)

yieldProd :: IntTree -> Int
yieldProd (Node i []) = i
yieldProd (Node i ts) = product (map yieldProd ts)
```

Zeigen Sie unter Verwendung der obigen Definitionen durch strukturelle Induktion die Gültigkeit der Gleichung

$$\texttt{product (yield t)} = \texttt{yieldProd t} \tag{A}$$

für jeden Baum `t :: IntTree`. Sie dürfen dabei nutzen, dass für alle Typen `a`, `b`, positive Integer $\texttt{k} > 0$, Integer `i :: Int`, Funktionen `f :: a -> b`, Werte `a1, ..., ak :: a` und Listen `l1, ..., lk :: [Int]` folgende Gleichungen gelten:

$$\texttt{product [i]} = \texttt{i} \tag{L1}$$

$$\texttt{map f [a1, ..., ak]} = \texttt{[f a1, ..., f ak]} \tag{L2}$$

$$\texttt{product (concat [l1, ..., lk])} = \texttt{product [product l1, ..., product lk]} \tag{L3}$$