Zur Erinnerung: λ-Kalkül

- Motivation \o/: minimale universelle Programmiersprache
- Lambda-Abstraktion immer geklammert: $(\lambda x.F)$
- Applikation von Termen ist linksassoziativ: EFG ist implizit geklammert als $(EF)G$
- Kurzschreibweise: $(\lambda xy.F)$ für $(\lambda x.(\lambda y.F))$
- Vor Ausführung einer β-Reduktion $(\lambda x.F)G$ müssen die freien Variablen in G und die gebundenen in F bestimmt werden.
- Sind die Mengen nicht disjunkt (d.h. $FV \cap GV \neq \emptyset$), müssen bei der α-Konversion Variablen umbenannt werden.
- β-Reduktion: In $(\lambda x.F)G$ werden alle Vorkommen der Variablennamen x in F durch G ersetzt und $\lambda x.$ wird entfernt.
- Die Applikation wird so lange wiederholt, bis keine Reduktion mehr möglich ist.
- Nicht für alle Terme existiert eine β-Normalform.
  (vgl. Blatt 5, Übung 3 (b) (4))

Übung 2 (a)

$$
\begin{aligned}
& (\lambda f\ \underbrace{x.ffx}_{GV=\{x\}}\ )\ \underbrace{(\lambda y.x)}_{FV=\{x\}}\ z \\
\Rightarrow_\alpha\ & (\lambda f\,\underbrace{x_1.ffx_1}_{GV=\{x_1\}})\ \underbrace{(\lambda y.x)}_{FV=\{x\}}\ z \\
\Rightarrow_\beta\ & (\lambda x_1.(\lambda y.\ \underbrace{x}_{GV=\emptyset}\ )\ \underbrace{(\lambda y.x)}_{FV=\{x\}}\ x_1)z \\
\Rightarrow_\beta\ & (\lambda x_1.\ \underbrace{xx_1}_{GV=\emptyset}\ )\ \ \underbrace{z}_{FV=\{z\}} \\
\Rightarrow_\beta\ & xz
\end{aligned}
$$

Übung 2 (b)

$$\begin{aligned}
& \langle Y\rangle\langle F\rangle = (\lambda z.((\lambda u.z(uu))(\lambda u.z(uu))))\langle F\rangle \\
\Rightarrow_\beta \; & ((\lambda u.\langle F\rangle(uu))(\lambda u.\langle F\rangle(uu))) = \langle Y_{\boldsymbol{F}}\rangle \\
\Rightarrow_\beta \; & \langle F\rangle((\lambda u.\langle F\rangle(uu))(\lambda u.\langle F\rangle(uu))) = \langle F\rangle\langle Y_{\boldsymbol{F}}\rangle
\end{aligned}$$

Übung 2 (b)

$$
\begin{aligned}
& \langle Y\rangle\langle F\rangle\langle 6\rangle\langle 5\rangle\langle 3\rangle \\
\Rightarrow^* \;& \langle F\rangle\langle Y_F\rangle\langle 6\rangle\langle 5\rangle\langle 3\rangle \\
\Rightarrow^* \;& \langle \mathsf{ite}\rangle \; \underbrace{(\langle \mathsf{iszero}\rangle \; \underbrace{(\langle \mathsf{sub}\rangle\langle 6\rangle\langle 5\rangle)}_{\Rightarrow^*\langle 1\rangle})}_{\Rightarrow^*\langle \mathsf{false}\rangle}(\,\ldots\,) \\
& (\langle \mathsf{succ}\rangle(\langle Y_F\rangle \; \underbrace{(\langle \mathsf{pred}\rangle\langle 6\rangle)}_{\Rightarrow^*\langle 5\rangle} \; \underbrace{(\langle \mathsf{succ}\rangle\langle 5\rangle)}_{\Rightarrow^*\langle 6\rangle} \; \underbrace{(\langle \mathsf{mult}\rangle\langle 2\rangle\langle 3\rangle)}_{\Rightarrow^*\langle 6\rangle})) \\
\Rightarrow^* \;& \langle \mathsf{succ}\rangle(\langle Y_F\rangle\langle 5\rangle\langle 6\rangle\langle 6\rangle) \\
\Rightarrow^* \;& \langle \mathsf{succ}\rangle(\langle F\rangle\langle Y_F\rangle\langle 5\rangle\langle 6\rangle\langle 6\rangle) \\
\Rightarrow^* \;& \langle \mathsf{succ}\rangle(\langle \mathsf{ite}\rangle \; \underbrace{(\langle \mathsf{iszero}\rangle \; \underbrace{(\langle \mathsf{sub}\rangle\langle 5\rangle\langle 6\rangle)}_{\Rightarrow^*\langle 0\rangle})}_{\Rightarrow^*\langle \mathsf{true}\rangle} \; \underbrace{(\langle \mathsf{add}\rangle\langle 6\rangle\langle 6\rangle)}_{\Rightarrow^*\langle 12\rangle}(\,\ldots\,)) \\
\Rightarrow^* \;& \langle \mathsf{succ}\rangle\langle 12\rangle \\
\Rightarrow^* \;& \langle 13\rangle
\end{aligned}
$$

Übung 2 (c)

$$\begin{aligned}\langle G\rangle = \Bigg(\lambda g\,x\,y.\langle\mathsf{ite}\rangle\Big(\langle\mathsf{iszero}\rangle\,x\Big)\\
&\Big(\langle\mathsf{mult}\rangle\,\langle 2\rangle\,(\langle\mathsf{succ}\rangle\,y)\Big)\\
&\Big(\langle\mathsf{ite}\rangle\Big(\langle\mathsf{iszero}\rangle\,y\Big)\\
&\quad\Big(\langle\mathsf{mult}\rangle\,\langle 2\rangle\,(\langle\mathsf{succ}\rangle\,x)\Big)\\
&\quad\Big(\langle\mathsf{add}\rangle\,\langle 4\rangle\,(g\,(\langle\mathsf{pred}\rangle\,x)\,(\langle\mathsf{pred}\rangle\,y))\Big)\Big)\Bigg)\end{aligned}$$