

Übungen zur „Deskriptiven Programmierung“ Blatt 10

Aufgabe 28. Die folgenden Gleichungen seien gegeben:

$$\begin{aligned} \textit{once } f &= f \\ \textit{twice } f &= f \cdot f \\ (f \cdot g) a &= f (g a). \end{aligned}$$

Bestimme die Redexe der unten aufgeführten Ausdrücke und reduziere sie mittels LI- bzw. LO-Reduktion auf ihre Normalform.

$$\begin{aligned} &\textit{once twice once twice} \\ &\textit{once twice (once twice)} \\ &\textit{once (twice (once twice))} \end{aligned}$$

Beachte, daß die Ausdrücke und damit ihre Normalformen einen funktionalen Typ besitzen.

Aufgabe 29. Im λ -Kalkül können die Booleschen Werte *true* und *false* zum Beispiel durch die folgenden Terme codiert werden.

$$\begin{aligned} \textit{true} &= \lambda x . \lambda y . x \\ \textit{false} &= \lambda x . \lambda y . y \end{aligned}$$

Programmiere die Funktionen *not*, *and* und *or* und die Fallunterscheidung *if*.

Aufgabe 30. Auf A. Church geht die folgende Darstellung der natürlichen Zahlen zurück: die Zahl n wird durch eine zwei-parametrische Funktionen repräsentiert, die ihr erstes Argument n -mal auf ihr zweites Argument anwendet. Die Zahl 3 wird somit durch $\lambda f . \lambda a . f (f (f a))$ codiert. Implementiere Addition, Multiplikation und Exponentiation. Wie läßt sich die Subtraktion realisieren?

Aufgabe 31. Ein Fixpunktkombinator Θ hat die Eigenschaft, daß für alle λ -Terme f gilt

$$f(\Theta f) =_{\beta} \Theta f.$$

Zeige, daß F mit

$$\begin{aligned} F &= G^{26} \\ G &= \lambda abcdefghijklmnopqrstuvwxyzr.(dasisteinfixpunktkombinator) \end{aligned}$$

ein Fixpunktkombinator ist.