P. Thiemann, M. Neubauer Wintersemester 2004/2005

Compilerbau

http://proglang.informatik.uni-freiburg.de/teaching/compilerbau/2004/

Übungsblatt 8

Abgabe: 22.12.2004

Aufgabe 1 (Normalform):

Seien

$$\mathbf{I} \equiv \lambda x.x,$$
$$\mathbf{S} \equiv \lambda xyz.xz(yz).$$

Zeige, dass die folgenden Terme eine Normalform besitzen:

(i) $(\lambda y.yyy)((\lambda ab.a)\mathbf{I}(\mathbf{SS}))$

(ii) $(\lambda yz.zy)((\lambda x.xxx)(\lambda x.xxx))(\lambda w.\mathbf{I})$

(iii) $\mathbf{SSSSSSS}$

(iv) $\mathbf{S}(\mathbf{SS})(\mathbf{SS})(\mathbf{SS})\mathbf{SS}$

Aufgabe 2 (Fixpunkt-Theorem):

Sei $=_\beta$ die Äquivalenzrelation induziert durch die reflexive, transistive Hülle von $\rightarrow_\beta$.

Zeige mit Hilfe des Fixpunkt-Theorems, dass ein Term M existiert, so dass für alle Terme N gilt: $MN =_\beta MM$.

Aufgabe 3 (Beweisbarkeit):

Zeige, dass gilt $(\lambda y.(\lambda x.M))N =_\beta \lambda x.((\lambda y.M)N)$.

Aufgabe 4 (Church-Kodierung für Listen):

Gib eine Church-Kodierung für Listen, d.h. für die Listenkonstruktoren **cons** und **nil** sowie den Listenselektor **caseList**, im λ-Kalkül an.

Gib damit eine Term an, der zu einer Liste ihre Länge (kodiert als Church-Numeral) berechnet.

Aufgabe 5 (Implementierung des λ-Kalküls):

Implementiere in einer Programmiersprache Deiner Wahl eine Repräsentation von λ-Ausdrücken.

Implementiere außerdem eine Funktion, die zu einem λ-Ausdruck einen leftmost-outermost-β-Reduktionsschritt durchführt.