Institut für Informatik
der Universität München
Dr. Andreas Abel

WS 2008/09

27. Dezember 2008

Übungen zur Vorlesung
Lambda-Kalkül

Blatt 5

Aufgabe P-16 (Trennung I): Geben Sie für die folgenden Termpaare $t_1 \# t_2$ einen Kontext C an, so dass $C[t_1] = \mathsf{T}$ und $C[t_2] = \mathsf{F}$.

a) $\mathsf{I} \# \mathsf{T}$

b) $\lambda x.\, x\, \lambda y.\, x\, y \# \lambda x.\, x\, \lambda y.\, y\, x$

Aufgabe P-17 (Trennung II): Geben Sie für die folgenden Termpaare $t_1 \# t_2$ einen Kontext C an, so dass $C[t_1]$ eine Normalform hat und $C[t_2]$ keine.

a) $\mathsf{I} \# \mathsf{T}$

b) $\lambda x.\, x\, \lambda y.\, y\, \Omega \# \lambda x.\, x\, \Omega$

Aufgabe P-18 (SN für Paare): Betrachten Sie die Erweiterung des λ-Kalküls um Paare (Blatt 4, P-14), mit dem zusätzlichen Axiomenschema
$\mathsf{Pair}\,(\mathsf{fst}\, t)\,(\mathsf{snd}\, t) \longrightarrow_\eta t$.

Ein *Evaluationskontext* ist ein Term mit einem Loch $[\,]$ in schwacher Kopfposition, gegeben durch die Grammatik $E ::= [\,] \mid E\, s \mid \mathsf{fst}\, E \mid \mathsf{snd}\, E$. Die Einsetzung von r in das Loch von E bezeichnen wir mit $E[r]$. Beispiele für Evaluationskontexte sind $[\,]\,\vec{s}$ und $(\mathsf{fst}\,([\,]\, s))\, t$; keine Evaluationskontexte sind z.B. $\lambda x.\,[\,]$ und $r\,[\,]$.

Für den erweiterten Kalkül definieren wir SN wie folgt:

$$\frac{\vec{s} \in \mathsf{SN}}{x\,\vec{s} \in \mathsf{SN}} \qquad \frac{t \in \mathsf{SN}}{\lambda x t \in \mathsf{SN}} \qquad \frac{r, s \in \mathsf{SN}}{\mathsf{Pair}\, r\, s \in \mathsf{SN}} \qquad \frac{E[t[s/x]] \in \mathsf{SN} \qquad s \in \mathsf{SN}}{E[(\lambda x t)\, s] \in \mathsf{SN}}$$

$$\frac{E[r] \in \mathsf{SN} \qquad s \in \mathsf{SN}}{E[\mathsf{fst}\,(\mathsf{Pair}\, r\, s)] \in \mathsf{SN}} \qquad \frac{E[s] \in \mathsf{SN} \qquad r \in \mathsf{SN}}{E[\mathsf{snd}\,(\mathsf{Pair}\, r\, s)] \in \mathsf{SN}}$$

$$\frac{}{\mathsf{fst} \in \mathsf{SN}} \qquad \frac{}{\mathsf{snd} \in \mathsf{SN}} \qquad \frac{}{\mathsf{Pair} \in \mathsf{SN}} \qquad \frac{r \in \mathsf{SN}}{\mathsf{Pair}\, r \in \mathsf{SN}}$$

a) Finden Sie einen stark $\beta\eta$-normalisierenden Term t mit $t \notin \mathsf{SN}$. [Hinweis: Dies ist notwendigerweise ein "unsinniger" Term.]

b) Zeigen Sie: Jeder Term in SN ist stark $\beta\eta$-normalisierend. Orientieren Sie sich am Beweis der Vorlesung. Die dort behandelten Fälle müssen Sie nicht nochmal aufschreiben, es reicht ein Verweis auf die Vorlesung.

Die ersten beiden Aufgaben noch vor Weihnachten lösen!

Aufgabe H-16 (W-Bäume I): Als W-Baum bezeichnet man einen λ-Term, gesehen als Baum, in dem der rechte Teilbaum jedes Applikationsknotens *einfach* ist, d.h. selbst keine Applikation ist.

Sei t ein Term in β-Normalform. Beweisen oder widerlegen Sie: Ist t ein W-Baum, so auch

$$t' \equiv (\lambda x f r.\, x\, f)\, (\lambda z.\, (\lambda y x o.\, x\, y)\, (\lambda e s.\, z\, e\, s\, t)\, (\lambda y h. y)).$$

Aufgabe H-17 (W-Bäume II): Sei t' wie in H-23. Berechnen Sie die β-Normalform von t'.

Aufgabe H-18 (Trennung): Geben Sie für die folgenden Termpaare $t_1 \# t_2$ einen Kontext C an, so dass $C[t_1] = \mathsf{T}$ und $C[t_2] = \mathsf{F}$.

a) $\lambda x\, y\, z.\, (x\, y)\, (y\, z)\, (z\, x) \# \lambda x\, y\, z.\, (x\, z)\, (z\, y)\, (y\, x)$

b) $\lambda x.\, x\, x\, \lambda y.\, x\, y \# \lambda x.\, x\, (\lambda y.\, y\, x)\, x$

Aufgabe H-19 (Vollständige η-Normalisierung, 4 Punkte): Die Funktion $(\cdot)^\eta$ ist rekursiv definiert wie folgt:

$$\begin{array}{lcl} x^\eta & = & x \\ (r\, s)^\eta & = & r^\eta\, s^\eta \\ (\lambda x t)^\eta & = & \begin{cases} t' & \text{falls } t^\eta = t'\, x \text{ und } x \notin \mathsf{FV}(t') \\ \lambda x.\, t^\eta & \text{sonst} \end{cases} \end{array}$$

Machen Sie sich an einigen Beispielen klar, dass $(\cdot)^\eta$ alle η-Redexe entfernt. Beweisen Sie für alle t: $t \longrightarrow_\eta^* t^\eta \not\longrightarrow_\eta$.

Aufgabe H-20 (Normalisierer): Implementieren Sie einen Auswerter für λ-Terme, der die

a) β-Normalform

b) $\beta\eta$-Normalform

berechnen, in JAVA oder einer funktionalen Programmiersprache.

Abgabe am 12.01.2007 in der Vorlesung.