
Grundlagen von Programmiersprachen

<http://www.informatik.uni-freiburg.de/proglang/teaching/ws2003-2004/gvps>

Übungsblatt 8

Abgabe: 15.12.2003

Aufgabe 1 (Einfach getypter Lambda-Kalkül – Normalisierung).

Zeige, dass es Lambda-Ausdrücke gibt, die eine Normalform besitzen, für die jedoch im einfach getypten Lambda-Kalkül kein Typ herleitbar ist.

- (i) Gib im Lambda-Kalkül eine Funktion `and` an, die als ersten Parameter eine natürliche Zahl n erwartet und als weiter Parameter gerade n zusätzliche boolesche Parameter akzeptiert, um als Ergebnis die boolesche Konjunktion der n booleschen Parameter zurückzuliefern. (Zahlen und boolesche Wert sollen jeweils in der Church-Codierung übergeben werden.)
- (ii) Zeige, dass $(\text{and } [n] [b_1] \dots [b_n])$ für alle $n \in \mathbf{N}, b_i \in \mathbf{B}$ eine Normalform besitzt.
- (iii) Zeige, dass für `and` im einfach getypten Lambda-Kalkül keine Typherleitung existiert.

Aufgabe 2 (Typkorrektheit — Big-Step-Semantik).

In der Vorlesung wurde die Typkorrektheit des einfach getypten Lambda-Kalküls bezüglich der Reduktions- und der denotationellen Semantik gezeigt. Beweise das entsprechende Resultat bezüglich der Big-Step-Semantik.

- (i) Erweitere zuerst die Big-Semantik um einen Fehlerzustand und passende Regeln, die den Fehlerzustand einführen und weiterpropagieren.
- (ii) Gib eine Typregel für *Closures* an. Spezifiziere ein neues Typurteil für korrekt getypte Umgebungen, $\Gamma \vdash \rho$.
- (iii) Zeige, das gilt:
Ist $\rho \vdash e \hookrightarrow y$ herleitbar und existiert eine Typherleitung $\Gamma \vdash e : \tau$ mit $\Gamma \vdash \rho$, dann gilt auch $\Gamma \vdash y : \tau$.