

# Theorie der Programmierung

## Wintersemester 2006/07

### Übungsblatt 4

#### Aufgabe 1

Ergänzen Sie die Ausdrücke aus Aufgabe 1 und 3 von Übungsblatt 3 so durch Typen, dass wohlgetypte Ausdrücke in  $\mathcal{L}_2^t$  entstehen. Geben Sie die Typherleitungen für *exists* (Aufgabe 3 b.) und *iter* (Aufgabe 1 e.) an.

#### Aufgabe 2

Beweisen Sie Satz 2 der Vorlesung, d.h. zeigen Sie, dass für jede Typumgebung  $\Gamma$  und jeden Ausdruck  $e$  in  $\mathcal{L}_2^t$  höchstens ein Typ  $\tau$  mit  $\Gamma \triangleright e :: \tau$  existiert.

#### Aufgabe 3

Führen Sie die (in der Vorlesung schon angedeutete) induktive Definition der Funktion  $type : TEnv \hookrightarrow Type$  zu Ende. Implementieren Sie die Funktion in einer Programmiersprache Ihrer Wahl. (Hinweis: Für eine einfache und gut lesbare Implementierung eignen sich funktionale Sprachen wie SML oder O'Caml. Versuchen Sie aber nicht, *type* als Funktionsnamen zu wählen.)

#### Aufgabe 4

Denken Sie sich Typregeln für den folgenden syntaktischen Zucker aus

- a.  $e_1 \&\& e_2$
- b.  $e_1 \parallel e_2$
- c. **let**  $id (id_1 : \tau_1) \dots (id_n : \tau_n) : \tau = e_1$  **in**  $e_2$
- d. **let rec**  $id : \tau = e_1$  **in**  $e_2$
- e. **let rec**  $id (id_1 : \tau_1) \dots (id_n : \tau_n) : \tau = e_1$  **in**  $e_2$

und zeigen Sie, dass es sich um abgeleitete Regeln handelt.