

Theorie der Programmierung (WS 2003/04)

Übungsblatt 12

Aufgabe 1

Welche der folgenden Ausdrücke sind (im neuen Typsystem mit den subtyping-Regeln) wohlgetypt, welche terminieren bei der Auswertung, welche bleiben stecken?

- a. **let val** $x = \text{ref } \{\}$ **in** $x := \{a = 0\}$ **end**
- b. **let val** $x = \text{ref } \{a = 0\}$ **in** $x := \{\}$ **end**
- c. **let val** $x = \text{ref } \{a = 0\}$ **in** $x := \{b = 0\}$ **end**
- d. **let val** $x = \text{ref } \{a = 0\}$ **in** $\#a (!x)$ **end**
- e. **let val** $x = \text{ref } \{a = 0\}$ **in** $x := \{\}; \#a (!x)$ **end**
- f. **let val** $x = \text{ref } \{\}$ **in** $x := \{a = 0\}; \#a (!x)$ **end**

Aufgabe 2

Zeigen Sie, dass unser Typsystem nicht mehr typsicher wäre, wenn wir die Subtyping-Regel (ARROW) durch

$$\text{(ARROW-?) } \frac{\tau_1 <: \tau_2 \quad \tau'_1 <: \tau'_2}{\tau_1 \rightarrow \tau'_1 <: \tau_2 \rightarrow \tau'_2}$$

ersetzen würden, d.h. geben Sie einen abgeschlossenen Ausdruck an, der mit dieser Regel wohlgetypt wäre und dessen Berechnung steckenbleibt.

Aufgabe 3

Zeigen Sie, dass die Typsicherheit auch verloren geht, wenn wir die Subtyping-Regel (REF) durch die Regel

$$\text{(REF-?) } \frac{\tau_1 <: \tau_2}{\tau_1 \text{ ref } <: \tau_2 \text{ ref}}$$

oder durch die Regel

$$\text{(REF-??) } \frac{\tau_2 <: \tau_1}{\tau_1 \text{ ref } <: \tau_2 \text{ ref}}$$

ersetzen.