

## Semantik von Programmiersprachen – SS 2010

<http://pp.info.uni-karlsruhe.de/lehre/SS2010/semantik>

### Blatt 8: Typsicherheit

Besprechung: 07.06.2010

#### 1. Welche der folgenden Aussagen sind richtig, welche falsch? (H)

- (a)  $[x \mapsto \text{int}, y \mapsto \text{bool}] \vdash (x + 3 \leq 17) \parallel (\text{not } y) :: \text{bool}$
- (b) Im Kontext  $[x \mapsto \text{bool}]$  ist  $\text{true} == x$  ein typkorrekter Ausdruck.
- (c)  $\mathcal{E} \llbracket 0 * (x - y) \rrbracket [x \mapsto 5, y \mapsto \text{tt}] = 0$
- (d) Jeder Ausdruck  $e$  hat in jedem Kontext  $\Gamma$  höchstens einen Typ  $\tau$ .
- (e) Der Typ eines Wertes  $v$  ließe sich auch so definieren:

$$\text{type}(v) = \tau \quad \text{gdw.} \quad \Gamma \vdash v :: \tau$$

- (f) Es gibt keinen Kontext  $\Gamma$  mit

$$\Gamma \vdash \{ \text{var } x = x; z := x \}; \{ \text{var } y = x; \text{if } (y) \text{ then } z := 5 \text{ else skip } \} \checkmark$$

- (g) Wenn  $\Gamma \vdash c \checkmark$  und  $\langle c, \sigma \rangle \rightarrow_1 \langle c', \sigma' \rangle$  mit  $\sigma' :: \Gamma$  und  $\Gamma \vdash c' \checkmark$ , dann auch  $\sigma :: \Gamma$ .
- (h) Wenn  $\Gamma \vdash c \checkmark$  und  $\sigma :: \Gamma$ , dann gibt es  $c'$  und  $\sigma'$  mit  $\langle c, \sigma \rangle \rightarrow_1 \langle c', \sigma' \rangle$ .

#### 2. Ausgaben (H)

Erweitern Sie die Sprache  $\text{While}_T$  um Ausgaben:

- (a) Ergänzen Sie die Syntax um eine neue Anweisung  $\text{print } e$  für Ausgaben.
- (b) Passen Sie die Typ-Regeln an. Sowohl Zahlen als auch Wahrheitswerte sind als Ausgabe erlaubt.
- (c) Ändern Sie die Big-Step-Semantik, sodass die neue Auswertungsrelation  $\langle c, \sigma \rangle \Downarrow_o \sigma'$  die Anweisung  $c$  im Anfangszustand  $\sigma$  zum Endzustand  $\sigma'$  auswertet, wobei die Liste der Ausgaben  $o$  entsteht.
- (d) Passen Sie auch die Small-Step-Semantik an Ausgaben an. Unterscheidet sie sich in der Ausdrucksmächtigkeit von der Big-Step-Semantik?
- (e) Ist die erweiterte Sprache typsicher? Begründen Sie!

#### 3. Typsicherheit mit der Big-Step-Semantik (Ü)

Typsicherheit lässt sich auch für eine Big-Step-Semantik zeigen, in dieser Aufgabe für  $\text{While}_T$ :

- (a) Ändern Sie die Auswertungsrelation so, dass eine Anweisung mit einem Zustand statt zu einem Endzustand auch zu einem speziellen Fehlerwert  $\text{Err}$  auswerten kann. Fügen Sie nun für jede Regel, die einen Typcheck (implizit) durchführt, eine neue Regel für den Fehlerfall hinzu. Vergessen Sie auch nicht Propagationsregeln für den Typfehler  $\text{Err}$ .
- (b) Zeigen Sie, dass typkorrekte Programme von konformanten Zuständen aus nie Typfehler erzeugen: Wenn  $\Gamma \vdash c \checkmark$ ,  $\sigma :: \Gamma$  und  $\langle c, \sigma \rangle \Downarrow \sigma'$ , dann  $\sigma' \neq \text{Err}$  und  $\sigma' :: \Gamma$ .
- (c) Vergleichen Sie diesen Ansatz, Typsicherheit einer Sprache zu zeigen, mit dem Small-Step-Ansatz aus der Vorlesung.