

## Semantik von Programmiersprachen – SS 2012

<http://pp.info.uni-karlsruhe.de/lehre/SS2012/semantik>

### Blatt 3: Small-Step-Semantik

Besprechung: 8.05.2012

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

- (a)  $b_1 \ \&\& \ b_2$  verhält sich semantisch wie `if (b1) then true else b2`.
- (b)  $\langle \text{if } (b) \text{ then } c_1 \text{ else } c_2, \sigma \rangle$  und  $\langle \text{if } (\text{not } b) \text{ then } c_2 \text{ else } c_1, \sigma \rangle$  haben die gleichen Ableitungsfolgen.
- (c) Wenn  $\langle c_1; c_2, \sigma \rangle \xrightarrow{*}_1 \langle c'_1; c_2, \sigma' \rangle$ , dann auch  $\langle c_1, \sigma \rangle \xrightarrow{*}_1 \langle c'_1, \sigma' \rangle$ .
- (d) Wenn  $\langle c, \sigma \rangle \xrightarrow{n}_1 \langle c, \sigma \rangle$ , dann  $n = 0$ .
- (e)  $\langle x := 1; \text{while } (x \leq 2) \text{ do } x := x * 2, \sigma \rangle \xrightarrow{12}_1 \langle \text{skip}, \sigma[x \mapsto 4] \rangle$ .
- (f) Wenn  $\langle c, \sigma \rangle \xrightarrow{*}_1 \langle \text{skip}, \sigma' \rangle$ , dann enthält  $c$  syntaktisch keine Schleife der Form `while (true) do c'`.
- (g) Wenn  $(\gamma_i)_i$  und  $(\delta_j)_j$  Ableitungsfolgen für  $\langle c, \sigma \rangle$  sind, dann  $(\gamma_i)_i = (\delta_j)_j$ .
- (h) Wenn  $\langle c, \sigma \rangle \xrightarrow{*}_1 \langle c', \sigma' \rangle$ , dann enthält  $c'$  höchstens dreimal so viele AST-Knoten wie  $c$ .

#### 2. Small-Step simuliert Big-Step (H)

Beweisen Sie durch Induktion über die Regeln der Big-Step-Semantik, dass jede Ausführung in der Big-Step-Semantik eine äquivalente Ausführung in der Small-Step-Semantik besitzt, d.h.: Aus  $\langle c, \sigma \rangle \Downarrow \sigma'$  folgt  $\langle c, \sigma \rangle \xrightarrow{*}_1 \langle \text{skip}, \sigma' \rangle$ .

#### 3. Semantische Äquivalenz bei Small-Step (Ü)

Zwei Programme  $c$  und  $c'$  sind *äquivalent* bezüglich der Small-Step-Semantik, falls für alle Zustände  $\sigma, \sigma'$  gilt:

- (i)  $\langle c, \sigma \rangle \xrightarrow{*}_1 \langle \text{skip}, \sigma' \rangle$  genau dann, wenn auch  $\langle c', \sigma \rangle \xrightarrow{*}_1 \langle \text{skip}, \sigma' \rangle$  und
- (ii)  $\langle c, \sigma \rangle \xrightarrow{\infty}_1$  genau dann, wenn auch  $\langle c', \sigma \rangle \xrightarrow{\infty}_1$ .

Vergleichen Sie diesen Äquivalenzbegriff mit dem semantischen Äquivalenzbegriff für die Big-Step-Semantik. Sie können dazu davon ausgehen, dass für alle  $c, \sigma$  und  $\sigma'$  gilt:

$$\langle c, \sigma \rangle \Downarrow \sigma' \quad \text{gdw.} \quad \langle c, \sigma \rangle \xrightarrow{*}_1 \langle \text{skip}, \sigma' \rangle.$$

Beweisen Sie, dass die folgenden Programme im Small-Step-Sinn äquivalent sind:

- (a) `while (b) do c` und `if (b) then c; while (b) do c else skip`
- (b) `c; while (true) do c'` und `while (true) do c''`
- (c) Wenn  $c$  und  $c'$  äquivalent sind, dann auch `while (b) do c` und `while (b) do c'`.

Überlegen Sie sich einen sinnvollen Äquivalenzbegriff, für den (c) nicht gilt.