Karlsruher Institut für Technologie

Lehrstuhl für Programmierparadigmen

Denis Lohner denis.lohner@kit.edu

Sebastian Ullrich sebastian.ullrich@kit.edu

Semantik von Programmiersprachen – SS 2017

http://pp.ipd.kit.edu/lehre/SS2017/semantik

Blatt 4: Big-Step- und Small-Step-Semantik Besprechung: 22.05.2017

1. Semantik mit Ausführungszeiten (H)

Die Big-Step-Semantik für While berücksichtigt nicht, wie viele Schritte die Ausführung eines Programms benötigt. Dies sollen Sie in dieser Aufgabe modellieren:

(a) Definieren Sie, auf Papier oder in Prolog, eine Auswertungsrelation $\langle c,\ \sigma\rangle \Downarrow_t \sigma'$ mit der Bedeutung, dass die Ausführung von c im Anfangszustand σ im Endzustand σ' endet und dafür t Schritte benötigt. Finden Sie dafür eine geeignete Definition, was ein Schritt sein soll.

(b) Beschreiben Sie formal, in welcher Beziehung $\langle c,\ \sigma\rangle \Downarrow \sigma'$ und $\langle c,\ \sigma\rangle \Downarrow_t \sigma'$ stehen. Wie würde man diese Beziehung beweisen?

(c) Überprüfen Sie, welche der folgenden Eigenschaften der Big-Step-Semantik $\langle c,\ \sigma\rangle \Downarrow \sigma'$ sich auf $\langle c,\ \sigma\rangle \Downarrow_t \sigma'$ übertragen lassen. Formulieren Sie die Aussagen entsprechend. Wie müssten die Beweise angepasst werden?

- Schleifenabwicklungslemma (Lem. 9)
- Determinismus (Thm. 10)
- Äquivalenz zwischen Big-Step- und Small-Step-Semantik (Kor. 23)

2. Semantische Äquivalenz bei Small-Step (Ü)

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

(i) $\langle c,\ \sigma\rangle \overset{*}{\to}_1 \langle \texttt{skip},\ \sigma'\rangle$ genau dann, wenn auch $\langle c',\ \sigma\rangle \overset{*}{\to}_1 \langle \texttt{skip},\ \sigma'\rangle$ und

(ii) $\langle c,\ \sigma\rangle \overset{\infty}{\to}_1$ genau dann, wenn auch $\langle c',\ \sigma\rangle \overset{\infty}{\to}_1$.

Vergleichen Sie diesen Äquivalenzbegriff mit dem semantischen Äquivalenzbegriff für die Big-Step-Semantik.

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) `while (true) do` c' und c`; 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.