

Übungen zu „Semantik von Programmiersprachen“, WS 2004/05

Nr. 3, Besprechung mündlicher Aufgaben: 15. November 2004 in der Übung,

Abgabe der Hausaufgaben: 17. November 2004 vor der Vorlesung

Mündliche Aufgaben

3.1 Beweisen Sie den folgenden Satz (2.6 b) der Vorlesung:

Es seien $is_1, is_2 \in \text{Instr}^*$, $d, d' \in (N \cup T)^*$ und $\sigma, \sigma' \in \Sigma$, sowie

$$(is_1 : is_2, d, \sigma) \triangleright^k (\varepsilon, d', \sigma')$$

Dann gilt:

$$\begin{aligned} \exists k_1, k_2 \in \mathbb{N}, d'' \in (N \cup T)^*, \sigma'' \in \Sigma : & \quad (is_1, d, \sigma) \triangleright^{k_1} (\varepsilon, d'', \sigma'') \\ & \wedge (is_2, d'', \sigma'') \triangleright^{k_2} (\varepsilon, d', \sigma') \\ & \wedge k_1 + k_2 = k \end{aligned}$$

Hinweis: Eine ähnliche Aussage wurde in der Vorlesung bewiesen.

3.2 Gemäß der Übersetzungsfunktion *trans* aus der Vorlesung wird **skip** zu NOOP übersetzt. Ein alternativer Ansatz könnte auch $trans(\mathbf{skip}) = \varepsilon$ sein.

Diskutieren Sie die Konsequenzen dieser geänderten Übersetzung für den Korrektheitsbeweis der Implementierung.

3.3 Wir verändern die abstrakte Maschine der Vorlesung, indem wir anstelle des Codes einen Befehlszähler einführen:

$$\mathbf{AM}' = \mathbb{N} \times (N \cup T)^* \times \Sigma$$

Der Befehlszähler wird in der Regel bei Ausführung einer Anweisung um 1 inkrementiert. Die auszuführende Anweisungsfolge stehe in einem indizierten Feld *is* zur Verfügung (Schreibweise: $is[n]$ für die n-te Anweisung).

(a) Statt BRANCH und LOOP verwenden wir nun Anweisungen LABEL(*l*), JUMP(*l*) und JUMPFALSE(*l*) mit $l \in \mathbb{N}$. Definieren Sie formal eine operationelle Semantik für diese Anweisungen, die folgendes leistet:

LABEL(*l*) erhöht nur den Befehlszähler.

JUMP(*l*) setzt die Ausführung an der Stelle mit LABEL(*l*) fort (falls existent).

JUMPFALSE(*l*) setzt die Ausführung an der Stelle mit LABEL(*l*) fort (falls existent), falls auf der Datenkellerspitze der Wert **false** steht. Ansonsten wird der Befehlszähler um 1 inkrementiert.

(b) Skizzieren Sie die Veränderungen für die Übersetzung der **if**- und **while**-Anweisungen.

Schriftliche Aufgaben

- 3.4 Beweisen Sie die **Vollständigkeit der Einzelschrittsemantik** bezüglich der Gesamtschrittsemantik:

5 Punkte

$$\forall c \in \mathbf{Cmd} \forall \sigma, \sigma' \in \Sigma : (c, \sigma) \rightarrow \sigma' \curvearrowright (c, \sigma) \Rightarrow^* \sigma'$$

Sie können das entsprechende Resultat für arithmetische und Boolesche Ausdrücke voraussetzen.

- 3.5 Gegeben sei die Anweisung

4 Punkte

$$X := -1; \mathbf{while} X \leq 0 \mathbf{do} (X := X - 1 \mathbf{or} X := (-1) * X)$$

welche die nichtdeterministische Erweiterung **or** von IMP benutzt.

- (a) Bestimmen Sie die Menge aller aus einem Anfangszustand $\sigma \in \Sigma$ resultierenden Ergebniszustände in der Gesamtschrittsemantik.
- (b) Geben Sie alle möglichen Berechnungen der Einzelschrittsemantik an.

Kann man Gesamtschrittsemantik und Einzelschrittsemantik für dieses Programm als äquivalent betrachten?

- 3.6 Setzen Sie die Semantik der **repeat**-Anweisung aus Aufgabe 1.3 in der abstrakten Maschine um, welche in der Vorlesung definiert wurde. Geben Sie eine erweiterte Übersetzungsfunktion an und begründen Sie die Korrektheit Ihrer Übersetzung.

3 Punkte