

## 5. Übung zu “Semantik von Programmiersprachen”, SS 2006

Abgabe schriftlicher Aufgaben: Di, 30.Mai 2006 (vor der Vorlesung)

Besprechung mündlicher Aufg.: (keine) Die Übung fällt aus!

---

### Mündliche Aufgaben

Dieses Blatt enthält keine mündlichen Aufgaben, weil wegen des Feiertags keine Übung stattfindet.

### Schriftliche Aufgaben

#### 5.1 repeat-Implementierung

4 Punkte

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.

#### 5.2 Übersetzung und Ausführung

5 Punkte

(a) Übersetzen Sie das Programm aus Aufgabe 1.2

$$Z := 0; \text{ while } Y \leq X \text{ do } (Z := Z + 1; X := X - Y)$$

in eine Instruktionssequenz für die abstrakte Kellermaschine AM der Vorlesung.

(b) Geben Sie die Konfigurationssequenz der Maschine für die Ausführung des Programms mit Startzustand  $\sigma_\emptyset[X \mapsto 11, Y \mapsto 4]$  an.

#### 5.3 Abstrakte Maschine mit Befehlszähler

In der abstrakten Maschine der Vorlesung soll anstelle des Codes ein Befehlszähler verwendet werden:

6 Punkte

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

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). Statt **BRANCH** und **LOOP** verwenden wir nun Anweisungen **JUMP**( $l$ ) und **JUMPFALSE**( $l$ ) mit  $l \in \mathbb{N}$ .

(a) Definieren Sie formal eine operationelle Semantik für diese Anweisungen, die folgendes leistet:

**JUMP**( $l$ ) setzt die Ausführung an der Stelle  $l$  fort.

**JUMPFALSE**( $l$ ) liest den Wert auf dem Keller. Falls dieser **false** ist, wird die Ausführung an der Stelle  $l$  fortgesetzt.

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

**Hinweis:** Bei der Übersetzung müssen Sprungziele berechnet werden, die sich aus der Länge des generierten Codes ergeben.