

# Übungsblatt 2

Abgabe: 28.11.2007

Wir betrachten die While-Sprache, die durch folgende Grammatik  $G_{While}$  definiert wird:

$$\begin{aligned}
 exp & ::= c \mid v \mid (exp_1 + exp_2) \mid (exp_1 - exp_2) \\
 boolexp & ::= \mathbf{true} \mid \mathbf{false} \mid (exp_1 > exp_2) \mid (exp_1 == exp_2) \mid (exp_1 != exp_2) \\
 stmt & ::= v := exp \mid stmt_1; stmt_2 \mid \\
 & \quad \mathbf{if} (boolexp) \mathbf{then} \{ stmt \} \mid \\
 & \quad \mathbf{while} (boolexp) \mathbf{do} \{ stmt \} \mid \\
 & \quad \mathbf{do} \{ stmt \} \mathbf{while} (boolexp) \mid \\
 & \quad \mathbf{for} (stmt_1, boolexp, stmt_2) \{ stmt_3 \} \mid \\
 & \quad \perp \\
 prog & ::= stmt
 \end{aligned}$$

Hierbei stellt jedes  $v \in X$  eine Variable dar. Es werden als Variablenwerte lediglich Ganzzahlen betrachtet, so dass zur Auswertung von Variablen im Rahmen der Semantikdefinition dieser Sprache eine partielle Valuationsfunktion  $\sigma \in X \not\rightarrow \mathbb{Z}$  verwendet werden kann. Entsprechend sind alle Konstanten  $c \in \mathbb{Z}$ . Das Terminal  $\perp$  ist der leere String.

Das Nichtterminal  $prog$  bildet den Einstieg in die Programmgrammatik.

## Aufgabe 1: Semantik der Sprache

(40%)

Als *Locations* für unser Programm betrachten wir noch nicht ausgeführten Programmtext. Diese sind also Elemente der Sprache, die durch  $G_{While}$  erzeugt werden.

$$Loc =_{def} Lang(G_{While})$$

Um Variablenvaluationen zu beschreiben verwenden wir wie erwähnt partielle Funktionen  $\sigma \in X \not\rightarrow \mathbb{Z}$ . Entsprechend ist die Menge aller Valuationsfunktionen:

$$\Sigma =_{def} X \not\rightarrow \mathbb{Z}$$

Der Einfachheit halber sei jedes  $\sigma \in \Sigma$  bereits auf natürliche Weise zur Auswertung von Ausdrücken erweitert, so dass etwa

$$\begin{aligned}
 \sigma(x + y) &= \sigma(x) + \sigma(y) \\
 \sigma(a > 0) &= \sigma(a) > 0
 \end{aligned}$$

und generell

$$\sigma(expr(x_1, \dots, x_n)) = expr(\sigma(x_1), \dots, \sigma(x_n))$$

Unser Zustandsraum stellt sich nun dar als:

$$S =_{def} Loc \times \Sigma$$

Für ein gegebenes Programm  $P_1$  in unserer Sprache mit Eingabevariablen  $E_1 \subseteq X$  stellt sich die Menge der Startzustände nun dar als:

$$S_0 =_{def} \{(P, \sigma) \in S \mid P = P_1 \wedge E_1 \subseteq \mathbf{dom} \sigma\}$$

Um die Semantik unserer Sprache zu definieren, verwenden wir ein Transitionssystem  $TS$ :

$$TS =_{def} (S, S_0, \longrightarrow)$$

Hierbei stellt  $\longrightarrow \subseteq S \times S$  die Zustandsübergangsrelation für unser Transitionssystem dar. Sie kann nun definiert werden, indem folgende operationelle Plotkin-Regeln angegeben werden:

1. Zuweisung

$$\frac{}{\langle x := exp; P, \sigma \rangle \longrightarrow \langle P, \sigma[x \mapsto \sigma(exp)] \rangle}$$

2. **if** mit **true**

$$\frac{\sigma(bexp) = \mathbf{true}}{\langle \mathbf{if} (bexp) \mathbf{then} \{P_1\}; P_2, \sigma \rangle \longrightarrow \langle P_1; P_2, \sigma \rangle}$$

3. **if** mit **false**

...

4. **while** mit **true**

...

5. **while** mit **false**

$$\frac{\sigma(bexp) = \mathbf{false}}{\langle \mathbf{while} (bexp) \mathbf{do} \{P_1\}; P_2, \sigma \rangle \longrightarrow \langle P_2, \sigma \rangle}$$

6. **do – while**

...

7. **for**

...

Vervollständigt diese Regeln.

## Aufgabe 2: Ausführung eines Programmes (30%)

Wir betrachten das folgende Programm zur Berechnung des größten gemeinsamen Teilers zweier Zahlen  $a$  und  $b$ :

```
while(a != b){
  if(b > a){
    a = a + b;
    b = a - b;
    a = a - b
  };
  a = a - b
};
```

Eine *Berechnung* eines Programmes ist eine Folge von Zuständen  $\langle s_0, \dots, s_n \rangle$  mit  $s_i \in S$  und  $i \in 0, \dots, n$ , so wie sie anhand eines gegebenen Startzustandes  $s_0$  durch die in der Semantik definierte Zustandsübergangsrelation entsteht. Eine Berechnung terminiert, wenn als Location der leere String entstanden ist, also kein Programmtext mehr auszuführen ist.

Um hierbei Schreibarbeit zu sparen werden nun Namen für Teilprogramme vergeben. Es sei  $IB \in Loc$  der innerhalb der `if`-Bedingung gegebene Programmtext,  $WB \in Loc$  sei der innerhalb der `while`-Schleife gegebene Programmtext, also:

IB :

```
a = a + b;
b = a - b;
a = a - b
```

WB :

```
if(b > a) { IB };
a = a - b
```

Diese müssen im Rahmen der Berechnung natürlich zu gegebenem Zeitpunkt wieder expandiert werden.

Sei als Startzustand gegeben:

$$s_0 = \langle \text{while}(a! = b)\{WB\}, [a \mapsto 16, b \mapsto 24] \rangle$$

Gebt die vollständige Berechnung anhand dieses Startzustandes an, so wie sie durch Anwendung der Plotkin-Regeln für  $\longrightarrow$  entsteht. Annotiert jeden Schritt mit der Nummer der Regel, die Ihr verwendet habt.

### Aufgabe 3: Transitionssystem über den Potenzmengenverband (30%)

Gebt zu unserem bisherigen Transitionssystem  $TS$  das Vergleichstransitionssystem  $TS_{\mathbf{P}}$  über den Potenzmengenverband an. Hierzu muss der entsprechende Zustandsraum  $S_{\mathbf{P}}$ , die Menge der möglichen Startzustände  $S_{0_{\mathbf{P}}}$ , sowie die Zustandsübergangsrelation  $\longrightarrow_{\mathbf{P}}$  angegeben werden.