Prof. Dr. R. Loogen, M. Dieterle
Fachbereich Mathematik und Informatik
Hans-Meerwein-Straße
D-35032 Marburg

20. April 2010

Philipps Universität Marburg

Übungen zu „Semantik von Programmiersprachen“, Sommer 2010

Nr. 2, Abgabe der Aufgaben: 27. April 2010 vor der Vorlesung

Aufgaben

2.1 Operationelle Semantik

3 Punkte

Gegeben seien die nebenstehende Anweisung und der Startzustand $\sigma = \sigma_\emptyset[X \mapsto 1, Y \mapsto 2]$. Geben Sie den resultierenden Endzustand und einen Herleitungsbaum dazu an.

```
while (X > 0) do
    while (Y > 1) do
       Z := X + Y;
       Y := Y - 2;
    X := X - 1
```

2.2 Erweiterung der `While`-Sprache

6 Punkte

(a) Erweitern Sie die Sprache WHILE um ein Konstrukt **repeat** c **until** b (Anweisung c wird wiederholt, bis b gilt) analog zum gleichnamigen Konstrukt in Pascal. Geben Sie entsprechende semantische Regeln an, so dass gilt:

$$(\ \textbf{repeat}\ c\ \textbf{until}\ b) \sim c;\ \textbf{if}\ b\ \textbf{then skip else}\ (\ \textbf{repeat}\ c\ \textbf{until}\ b);$$

Benutzen Sie dabei *nicht* das **while** -Konstrukt.

(b) Zeigen Sie, dass die Äquivalenz aus (a) auch wirklich gilt.

2.3 Äquivalenz von Schleifenkonstruktionen

3 Punkte

Beweisen oder widerlegen Sie für $b_1, b_2 \in \textbf{BExp}$ und $c \in \textbf{Cmd}$ die folgenden Äquivalenzaussagen:

(a) **while** $(b_1$ **or** $b_2)$ **do** $c \quad \sim \quad$ **while** $(b_1$ **or** $b_2)$ **do** c; **while** b_1 **do** c

(b) **while** $(b_1$ **and** $b_2)$ **do** $c \quad \sim \quad$ **while** $(b_1$ **and** $b_2)$ **do** c; **while** b_1 **do** c