Institut für Computersprachen
Technische Universität Wien
Prof. Dr. J. Knoop

A-1040 Wien
Argentinierstr. 8
Tel.: 01/58801-18510

"Analyse und Verifikation (185.276, VU 2.0, ECTS 3.0)" WS 2007/2008

Übungsblatt 1

08.10.2007

Aufgabe 1 : (8+2 Punkte)

Unter den *freien Variablen* eines Booleschen Ausdrucks b verstehen wir die Menge der in ihm vorkommenden Variablen. Diese Menge lässt sich induktiv wie folgt definieren:

$$\begin{array}{rcl} FV(true) & = & \emptyset \\ FV(false) & = & \emptyset \\ FV(a_1 = a_2) & = & FV(a_1) \cup FV(a_2) \\ FV(a_1 \leq a_2) & = & FV(a_1) \cup FV(a_2) \\ FV(\neg b_1) & = & FV(b_1) \\ FV(b_1 \wedge b_2) & = & FV(b_1) \cup FV(b_2) \\ FV(b_1 \vee b_2) & = & FV(b_1) \cup FV(b_2) \end{array}$$

1. Beweisen Sie induktiv: Sind σ und σ' zwei Zustände mit $\sigma(x) = \sigma'(x)$ für alle $x \in FV(b)$, dann gilt:
$$[\![b]\!]_B(\sigma) = [\![b]\!]_B(\sigma')$$
2. Was bedeutet die vorstehende Aussage anschaulich?

Aufgabe 2 : (8+2 Punkte)

Gegeben sei das folgende WHILE-Programm π:

```
z:=0; while y<=x do z:=z+1; x:=x-y od
```

1. Geben Sie die Ableitungsfolge an, die sich ergibt, wenn das Programm π auf einen Zustand σ mit $\sigma(x) = 31$ und $\sigma(y) = 9$ angesetzt wird.
2. Geben Sie einen Zustand τ an, für den π angesetzt auf τ divergiert.

Aufgabe 3 : (5 Punkte)

In dieser Aufgabe erweitern wir die Sprache WHILE um das Konstrukt

`repeat` π `until b taeper`

Geben Sie eine SOS-Regel [rep_*sos*] an, die diesem Konstrukt die "gewohnte" Semantik gibt, ohne bei dieser Angabe die Existenz des while-Konstrukts in WHILE auszunutzen.

Abgabe: Montag, den 29.10.2007, vor der Vorlesung (HS 14, Karlsplatz 13).