Modellierung und Programmierung 2

Hinweise Abgabe: https://moodle2.uni-leipzig.de/course/view.php?id=3907

Übungsserie 5

Formale Semantik und Verifikation

Aufgabe 17 *(Denotationale Semantik)* Definieren Sie für die folgenden Anweisungen die semantische Funktion F der denotationalen Semantik in der Form:

$$F[\langle Anweisung \rangle](z) = \ldots$$

Die Menge der Zustände Z_p soll dabei als Menge von Abbildungen der Variablennamen V auf den Bereich der ganzen Zahlen $\mathbb{Z}$ aufgefasst werden. Verwenden Sie die aus der Vorlesung und Übung bekannten Schreibweisen für Fallunterscheidungen und Variablenbelegungen $z\,\langle v \leftarrow w \rangle$, welche für die Variablenbelegung z die Variable v auf den Wert w zuweist und alle anderen Variablen unverändert läßt.

(a) ϵ `(* die leere Anweisung *)`

(b) `x := x + 1`

(c) `if x = 0 then` <A_1> `ELSE` <A_2> `END`

Aufgabe 18 *(Operationale Semantik)* Gegeben sei das folgende MINI-Programm P:

```
1  read x, y;
2  y := 0;
3  while x ≠ 0 do
4        y := y + 1;
5        x := x - 1;
6        if x = 0 then
7           ϵ
8        else
9           x := x - 1
10       end;
11 end;
12 write y.
```

(a) Geben Sie in der Schreibweise der operationalen Semantik an, wie P die Eingabe $x = 3$ und $y = 42$ verarbeitet. Um Schreibarbeit zu sparen, vereinbaren wir die folgenden Abkürzungen:
$i \equiv$ `if y = 0 then` ϵ `else x := x - 1 end;`
$w \equiv$ `while x` $\neq$ `0 do y := y + 1; x := x - 1; i end;`

(b) Welche Funktion berechnet P?

(c) Erweitern Sie die operationale Semantik von MINI um die `repeat`-Anweisung. Damit entsteht die Sprache MINI+ mit folgender Syntax-Ergänzung:
Anweisung $\rightarrow \ldots \mid$„repeat“ Anweisung „until“ Bezeichner „$= 0$“

Aufgabe 19 *(Axiomatische Semantik)* Im folgenden Pseudocode-Programmfragment sind Vor- und Nachbedingungen gegeben, welche das Programm spezifizieren.

```
1 (*   P_SPEC ≡ {−128 ≤ y ∧ y ≤ 128}  *)
2 x := -y;
3 IF x > 0 THEN y := x - 47 ELSE y := x + 11 END;
4 (*   Q_SPEC ≡ {−89 ≤ y ∧ y ≤ 53}  *)
```

Genügt dieses Programmfragment der Spezifikation? Begründen Sie Ihre Aussage *formal* mit Hilfe des *wp-Kalküls*!

Aufgabe 20 *(Axiomatische Semantik)* Im folgenden Pseudocode-Programmfragment sind Vor- und Nachbedingungen gegeben, welche das Programm spezifizieren.

```
1 (*   P_SPEC ≡ {−9 ≤ y ∧ y ≤ 9}  *)
2 x := y
3 if x < 0 then y = −1 ∗ x ∗ x else y := x*x end;
4 (*   Q_SPEC ≡ {−100 ≤ y ∧ y ≤ 100}  *)
```

Genügt dieses Programmfragment der Spezifikation? Begründen Sie Ihre Aussage *formal* mit Hilfe des *wp-Kalküls*!

Zusatzaufgabe 5 *(Schleifeninvariante)* Gegeben sei folgendes Programm:

```
1 read n;
2 a := 1;
3 i := 0;
4 while i < n do
5   a := a * (i+1);
6   i := i + 1;
7 end;
8 write a;
```

(a) Welche Funktion berechnet dieses Programm? Formulieren Sie die entsprechende Nachbedingung Q_{SPEC}.

(b) Geben Sie die Schleifeninvariante I zu der Schleife im Programm an (ohne Beweis). Berechnen Sie anschließend die schwächste Vorbedingung der Schleife (nicht des gesamten Programms) mit Hilfe des wp-Kalküls.
Hinweis: Gehen Sie analog zu den Seminarfolien vor und bestimmen Sie in einem ersten Schritt die einzelnen Bestandteile der schwächsten Vorbedingung. Notieren und vereinfachen Sie in einem letzten Schritt die gefundene Vorbedingung.

(c) Bestimmen Sie mit Hilfe der vorherigen Teilaufgabe nun die schwächste Vorbedinung des gesamten Programms. Was können Sie über `n` aussagen?