Lehrstuhl für Software and Computational Systems
Prof. Dr. Gidon Ernst, Marian Lingsch-Rosenfeld

Übungen zur Vorlesung
Formale Spezifikation und Verifikation

Wintersemester 2024/25

Übungsblatt 04

Bekanntgabe am 22.11.2023

Für diese Aufgabe werden Aufgaben 1.1 und 1.2 in den Tutorien vorgerechnet.

Aufgabe 2 ist zur Bearbeitung in Präsenz in den Tutorien vorgesehen.

1 Invarianten

1.1 Addieren durch Inkrementieren

Gegeben ist folgende Java-Methode/C-Funktion:

```
1  void add(int x0, int y0) {
2    assume(y0 >= 0);
3    int x = x0, y = y0;
4
5    while (y > 0) {
6      x = x + 1;
7      y = y - 1;
8    }
9
10   assert(x == x0 + y0);
11 }
```

a) Zeichnen Sie den Kontrollflussautomat für `add`.

b) Prüfen Sie die Korrektheit des Programms mit Hilfe einer symbolischen Erreichbarkeitsanalyse (informell genügt). Verwenden Sie, wie in der Vorlesung gezeigt, eine Invariante:

$$I(x_0, y_0, x, y) \iff (y \geq 0) \wedge (x + y = x + 0 + y_0)$$

Dazu stellen wir zunächst die Bedingungen für den Schleifeneintritt sowie einer beliebigen Iteration auf. Führen Sie die Beweise auf Papier aus:

$$\begin{aligned} y_0 \geq 0 &\implies I(x_0, y_0, x_0, y_0) \\ x > 0 \wedge I(x_0, y_0, x, y) &\implies I(x_0, y_0, x + 1, y - 1) \end{aligned}$$

Prüfen Sie anschließend, ob die `assert`-ion am Ende der Funktion immer gilt.

$$x \leq 0 \wedge I(x_0, y_0, x, y) \implies x = x_0 + y_0$$

1.2 Substraktion durch Dekrementieren

Gegeben ist folgende Java-Methode/C-Funktion:

```
1 int sub(int a, int b) {
2   int y = a;
3   while (b>0) {
4     y = y - 1;
5     b = b - 1;
6   }
7   int z = y;
8   return z;
9 }
```

Analog zu Aufgabe 1.1

a) Bestimmen Sie eine geeignete Invariante

b) Führen Sie die Beweise durch

c) Welche Laufzeitkomplexität besitzt diese Art der Addition und Subtraktion?

d) Recherche (z.B. auf Google Scholar): Finden Sie eine konkrete Technik heraus, mit der Invarianten dieser Art automatisch bestimmt werden können.

2 Präsebzaufgabe: Berechnung einzelner Hoare-Triples

Begründen Sie für jedes Hoare-Tripel an, ob dieses wahr oder falsch ist. Die Rechenoperationen agieren auf der Menge der ganzen Zahlen ohne Ganzzahlüberläufe.

1. $\{ \mathit{false} \}$ `j = i + 1;` $\{ i > 7 \}$

2. $\{ x < y \}$ `x = x + y;` $\{ x \geq y \}$

3. $\{ j = 0 \}$ `j = i * i;` $\{ j \geq i \}$

4. $\{ i < n \}$ `while`$(i < n)\{i = i + 1;\}$ $\{ i = n \}$

5. $\{ a = 2 \cdot b \}$ `if` $(a < b)$ $\{m = -b;\}$ `else` $\{m = a;\}$ $\{ m \geq 0 \}$

6. $\{ a = d \}$ `b = a + c;` `a = b - c;` $\{ a = d \}$