Systeme hoher Sicherheit und Qualität WiSe 15/16

Christoph Lüth
Martin Ring

5. Übungsblatt

Ausgabe: 15.12.15
Abgabe: 19.01.16

Aus Kostengründen wurden Teil der Softwareentwicklung für das Projekt A^2 (das autonome Fahrzeug) nach Indien ausgelagert. Das war keine gute Idee, denn jetzt müssen Sie als Qualitätsbeauftragter irgendwie beweisen, dass die dort entwickelte Software auch das tut, was sie soll. Und was noch schlimmer ist, die entwickelte Software ist nicht frei von Flüchtigkeitsfehlern.

5.1 *Statisch Analysiert* — *8 Punkte*

Das folgende Stück Programmcode erscheint Ihnen etwas ineffizient:

```
if (x< y) {
  a := y;
  y := x;
  x := a
  };
a:= x;
b:= y;
while (x != y) {
  a := y;
  while (x > y) {
    x := x- y
    };
  b := y;
  y := x;
  x := a
  }
```

Führen Sie deshalb eine statische Programmanalyse durch (leider manuell, die Firma kann sich keine entsprechenden Werkzeuge mehr leisten, nachdem alle Gelder durch das Outsourcing verbraucht worden sind. Damit entfällt leider auch Ihr Weihnachtsgeld. Tut uns leid.):

(A) Versehen Sie die elementaren Blöcke mit Labeln, und bestimmen Sie die Mengen *init*, *final*, *blocks* und *flow*.

(B) Führen Sie eine *Reaching Definitions Analysis* und eine *Live Variables Analysis* für das Programm durch. Welche Verbesserungen legen diese Analysen an dem Programm nahe?

5.2 *Korrekt Bewiesen* — *12 Punkte*

An einer besonders kritischen Stelle findet sich in dort folgendes Stück unkommentierter Quellcode

```
x:= 0;
p:= 1;
while (x < n) {
  p:= p+ 2*x;
  x:= x+ 1
  }
```

und es stellen sich folgende Fragen:

(A) Was soll dieses Programm vermutlich berechnen?

(B) Das Programm hat leider ein oder zwei kleine (Levenshtein-Distanz insgesamt drei oder kleiner) Fehler — wo verstecken sie sich, und wie können sie behoben werden?

(C) Spezifieren Sie die Funktionalität des Programmes, und beweisen Sie die Korrektheit mit dem Floyd-Hoare-Kalkül.