WS 2013/2014 21.01.2014

Übungen zur Vorlesung
Bäume, Ordnungen und Anwendungen
Blatt 12

Juniorprof. Dr. Roland Meyer Abgabe bis 28.01.2014 um 14h

Aufgabe 12.1 (Control-State-Reachability)
Eine *Safety-Eigenschaft* oder *Invariante* eines Programms ist eine Menge **Safe** $\subseteq$ **Prog** $\times$ **State** von Konfigurationen. Man sagt, die Eigenschaft ist *erfüllt*, falls alle erreichbaren Konfigurationen in **Safe** liegen, anderenfalls ist sie *verletzt*. In der Vorlesung wurden Beispiele für Safety-Eigenschaften gegeben, z.B.: „Es gibt keine Divisionen durch 0“ oder: „Die Variable x ist immer positiv“.
Erklären Sie, wie man die Überprüfung von Safety-Eigenschaften eines Programms auf einen Erreichbarkeitscheck für einen Kontrollzustand c_{bad} reduzieren kann. Machen Sie Ihre Vorgehensweise an den beiden folgenden Beispielen deutlich:

a) „Es gibt keine Divisionen durch 0.“

b) „Zwei parallele Programme greifen niemals gleichzeitig schreibend auf eine gemeinsame Datei zu.“

Hinweis: Diese Eigenschaft heißt „Wechselseitiger Ausschluss“. Gehen Sie davon aus, dass in zwei parallelen Programmen jeweils ein Befehl **write(file)** vorkommt. Wie überprüfen Sie, dass diese Befehle nicht gleichzeitig ausgeführt werden?

Aufgabe 12.2 (Abstraktionsverfeinerung)
Betrachten Sie das folgende Programm, das für $x, y \in \mathbb{N}$ das Produkt $z = x \cdot y$ berechnet.

$[z := 0]^1$
while $[x > 0]^2$ **do**
  $[z := z + y]^3$
  $[x := x - 1]^4$
if $[z \equiv 0 \mod y]^5$ **then**
  $[\text{skip}]^6$
else
  $[\text{skip}]^7$

Zeigen Sie mittels Abstraktionsverfeinerung, dass Block 7 nicht erreichbar ist.
Hinweis: Der Test $[z \equiv 0 \mod y]^5$ ist für $y = 0$ nicht definiert. Nehmen Sie daher bei der Erfüllbarkeitsanalyse der Formeln grundsätzlich $y > 0$ an.

Aufgabe 12.3 (Abstraktionsverfeinerung)
Betrachten Sie das folgende Programm:

$[x := a]^1$
$[y := b]^2$
while $[x \neq 0]^3$ **do**
    $[x := x - 1]^4$
    $[y := y - 1]^5$
if $[a = b \wedge y \neq 0]^6$ **then**
    $[\text{skip}]^7$
else
    $[\text{skip}]^8$

Wir wollen zeigen, dass man mittels der Abstraktionsverfeinerung aus der Vorlesung nicht beweisen kann, dass Block 7 nicht erreichbar ist. Genauer wollen wir zeigen, dass die Verfeinerung nicht terminiert.

a) Wir betrachten zunächst abstrakte Läufe des Programms. Geben Sie Prädikate $p_i^{(k)}$ für $k \geq 1$ und $i \in \{3, 4, 5\}$ an, die jeweils gelten, wenn Block i zum k-ten Mal betreten wird. *Achtung:* Der Fall $k = 1$ ist ein Sonderfall.

b) Zeigen Sie dann, dass im n-ten Verfeinerungsschritt ein (spurious) Gegenbeispiel mit n Schleifendurchläufen erzeugt wird, aber keines mit weniger als n Durchläufen.

c) Schließen Sie, dass die Verfeinerung nicht terminiert.

Abgabe bis 28.01.2014 um 14h im Kasten neben Raum 34-401.4