

Teil Grundlagen der Softwarezuverlässigkeit

Aufgabe 1

4+3 Punkte

- (a) Betrachten Sie das folgende Programm:

```
if (x>1 and y<2) then
  y:= x+y;
  if y>5 then x:=0;
  else x:=1;
else y:=0
```

Entwerfen Sie zwei möglichst kleine Mengen von Testfällen mit jeweils 100% Überdeckung bezüglich der folgenden zwei Kriterien:

- i) statement coverage
- ii) multiple condition coverage

- (b) Sei P ein Program mit der folgenden Struktur, wobei Z_{i_j} eine Zuweisung und b_{i_j} ein atomares Prädikat bezeichnet (d.h., b_{i_j} ist keine boolesche Kombination kleinerer Prädikate).

```
if (b_0_1 and b_0_2 and b_0_3) then Z_0_1 else Z_0_2;
if (b_1_1 and b_1_2 and b_1_3) then Z_1_1 else Z_1_2;
...
if (b_8_1 and b_8_2 and b_8_3) then Z_8_1 else Z_8_2;
if (b_9_1 and b_9_2 and b_9_3) then Z_9_1 else Z_9_2;
```

Wie viele Testfälle werden *mindestens* und *höchstens* gebraucht, um 100% Überdeckung bezüglich der in (a) genannten Kriterien zu erreichen (unter der Annahme, dass 100% Überdeckung möglich sind)? Geben Sie jeweils eine kurze Begründung für Ihre Antwort.

Aufgabe 2

4+2 Punkte

- (a) Betrachten Sie die folgenden beiden CTL-Formeln: $\varphi_1 \equiv \mathbf{AG EF} \varphi$ und $\varphi_2 \equiv \mathbf{AG EF AG} \varphi$, wobei φ eine beliebige Unterformel sei.

Gilt für beliebige Kripke-Strukturen \mathcal{K} und beliebige Zustände s

- i) $\mathcal{K}, s \models \varphi_1 \implies \mathcal{K}, s \models \varphi_2$?
- ii) $\mathcal{K}, s \models \varphi_2 \implies \mathcal{K}, s \models \varphi_1$?

Begründen Sie Ihre Antworten.

- (b) Betrachten Sie folgende Formel der Aussagenlogik: $((x \leftrightarrow z) \vee w) \wedge y$.

Geben Sie einen BDD für diese Formel an; wählen Sie dabei eine Variablenordnung, mit der der BDD möglichst wenige Knoten enthält.

Aufgabe 3

4+3 Punkte

(a) Betrachten Sie das folgende Programm C :

```
11: if x>10 then
12:   x := x*x;
    else
13:   x := 2*x;
    end if;
14: skip;
```

Es sei $A = \{\emptyset, \mathbf{g}, \mathbf{u}, \mathbf{gu}\}$ ein abstrakter Wertebereich mit der Konkretisierung

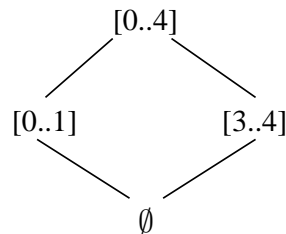
$$\gamma(\emptyset) = \emptyset \quad \gamma(\mathbf{g}) = \{x \mid x \text{ ist gerade}\}$$

$$\gamma(\mathbf{u}) = \{x \mid x \text{ ist ungerade}\} \quad \gamma(\mathbf{gu}) = \gamma(\mathbf{g}) \cup \gamma(\mathbf{u}).$$

Geben Sie die zugehörigen abstrakten Versionen der in C vorkommenden Zuweisungen und Bedingungen an, und approximieren Sie (mit dem Verfahren aus der Vorlesung) die abstrakte Sammelsemantik (*collecting semantics*) unter der Annahme, dass x anfangs den Wert 1 hat.

(b) Betrachten Sie den unten gezeigten Verband, dessen Elemente den Wert einer Programmvariablen x abstrahieren sollen, wobei jede Menge S des Verbands folgende Konkretisierung hat:

$$\gamma(S) = \{x \mid (x \bmod 5) \in S\}$$



Geben Sie ein Programm mit der folgenden Eigenschaft an: Die Konstruktion und Analyse des zugehörigen abstrakten Zustandsraums liefert die Information, dass x am Programmende unabhängig vom Anfangswert keinesfalls den Wert 2 haben kann; die Approximation der abstrakten Sammelsemantik liefert diese Information jedoch nicht. Liefern Sie auch eine kurze Begründung für Ihre Lösung.

Hinweis: Die Aufgabe kann mit einem sehr kurzen Programm gelöst werden.