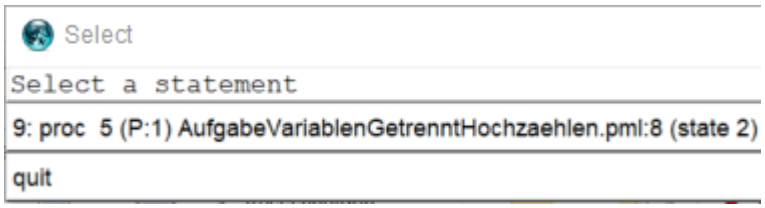
Prof. Dr. Stephan Kleuker
Fachhochschule Osnabrück
Fakultät Ing.-Wissenschaften und Informatik
- Software-Entwicklung -

Software Quality Management

Sommersemester 2023
9. Aufgabenblatt

(Besprechung 5.6)

Aufgabe 12
Zu entwickeln ist eine Spezifikation, die drei ganzzahlige globale Variablen x, y und z hat, die mit dem Wert Null starten. Weiterhin gibt es drei Prozesse, wobei jeder eine dieser globalen Variablen schrittweise bis einschließlich 10 hochzählt. Die Prozesse sollen sich so synchronisieren, dass jeweils immer nur ein Prozess fortschreiten kann, genauer erst P1 um eins hochzählt, dann P2 um eins hochzählt, dann P3 um eins hochzählt und dann wieder von vorne.
Bei einer interaktiven Simulation sollte damit die Spezifikation durchlaufen, ohne dass Sie neben der Abbruchmöglichkeit eine Alternative erhalten, irgendeine Auswahl zu treffen.



Übertragen Sie Ihre Idee auf eine Spezifikation mit N Variablen und N Prozessen, die sich genauso verhält. Simulieren Sie das Verhalten.

Aufgabe 13

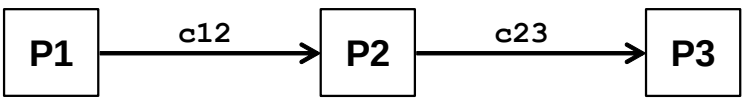
a) Gegeben sei der folgende PROMELA-Prozess mit einer globalen Variablen x.

```
byte x=0;
active proctype P(){
  atomic{
    x=x+1;
    x=x+1
  }
}
```

Schreiben Sie einen zweiten Prozess, der immer terminiert, wenn P die vorgestellte Form hat und der möglicherweise nicht terminiert, wenn man den atomaren Bereich auflöst, also atomic weglässt.

b) In den Unterlagen steht eine Behauptung, wie der jeweilige Wert von _pid berechnet wird. Überlegen Sie sich eine Spezifikation mit der die Behauptung zumindest validiert oder wiederlegt werden kann.

Aufgabe 14
In dieser Aufgabe wird schrittweise eine komplexe Spezifikation entwickelt. Es ist ratsam, die Ergebnisse für jede Teilaufgabe getrennt zu speichern. Nach einer erfolgreichen Verifikation sollte man mal einen Fehler einbauen, damit man lernt, das Resultat zu interpretieren.



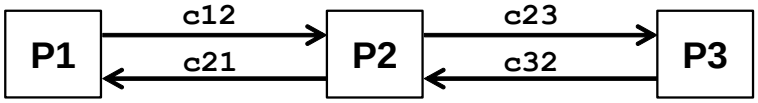
Zu spezifizieren sind 3 Prozesse, wobei P1 die Zahlen von 1 aufsteigend bis maximal 10 an P2 überträgt und P2 alle erhaltenen Werte an P3 überträgt. P1 kann dabei nichtdeterministisch ab der Zahl 5 entscheiden, ob die Übertragung beendet wird. P2 und P3 summieren jeweils in einzelnen globalen Variablen die erhaltenen Werte unmittelbar nach Erhalt auf. P1 summiert in einer eigenen globalen Variablen die gesendeten Werte vor dem Senden.

Prof. Dr. Stephan Kleuker
Fachhochschule Osnabrück
Fakultät Ing.-Wissenschaften und Informatik
- Software-Entwicklung -

a) Übertragen Sie die informelle Spezifikation in eine PROMELA-Spezifikation mit synchroner Kommunikation. Dabei sind zunächst keine Besonderheiten, wie Endzustände, zu berücksichtigen, allerdings sollen Sie atomic bzw. d_step nur beim Aufsummieren nutzen. Überprüfen Sie mit einem Simulationslauf das korrekte Verhalten (warum ist hier eine Verifikation mit Hilfe der Simulation möglich?).

b) Für die drei Summenwerte kann man sich eine globale Invariante, also eine Invariante, die in jedem Zustand gilt, ausdenken. Formulieren Sie eine solche Invariante und spezifizieren Sie einen zusätzlichen Prozess, der die Invariante prüft. Führen Sie eine Verifikation mit iSpin aus. Nutzen Sie dazu ein assert in einem eigenen Prozess und die Standardverifikationseinstellungen.

c) Nachdem P1 terminiert, werden sich P2 und P3 typischerweise nicht im Endzustand befinden, was Ihnen die Verifikation in b) auch mitteilen wird. Ergänzen Sie das Kommunikationsprotokoll so, dass P1 am Ende eine Terminierungsnachricht an P2 schickt, die von P2 an P3 weitergeleitet wird. Überprüfen Sie mit iSpin, dass alle Prozesse terminieren (Sie benötigen keine end-Markierungen).



d) Ergänzen Sie das Kommunikationsprotokoll so, dass P2 und P3 dem jeweiligen Sender den Empfang einer Zahl durch eine Bestätigungsnachricht (z. B. ack) zusammen mit dem empfangenen Wert und die Terminierungsnachricht (z. B. mit termack) bestätigen. Führen Sie die Verifikation aus b) und c) durch.

e) Erweitern Sie das Kommunikationsprotokoll so, dass P2 und P3 nichtdeterministisch entscheiden, dass eine empfangene Nachricht nicht in Ordnung ist. In diesem Fall soll dem jeweiligen Sender eine Nachricht mit der Aufforderung zur Wiederholung (repeat) geschickt werden. Beim erneuten Versenden soll der Wert nicht summiert werden. Beachten Sie, dass auch die Terminierungsnachricht eventuell wiederholt werden muss. Führen Sie die Verifikation aus b) und c) durch.

f) Am Ende der Prozessausführung sollten alle Summenwerte der Prozesse übereinstimmen. Spezifizieren Sie einen zusätzlichen Prozess, der von P1, P2 und P3 am Ende der Prozessausführung den Wert der jeweiligen Summe mitgeteilt bekommt (da die Summen global sind, würde es auch ausreichen, den neuen Prozess über die Terminierung zu informieren) und der mit einer Zusicherung prüft, dass alle Summen gleich sind.

g) Gehen Sie jetzt von einer synchronen zu einer asynchronen Spezifikation über, dabei soll jeder Kanal die Puffergröße 2 haben. Überlegen Sie sich, welche Probleme mit Ihrem für e) entwickelten Protokoll entstehen können. Führen Sie die Verifikation aus b) und c) durch.