

Modellierung und Analyse eingebetteter und verteilter Systeme

Übungsblatt 12

Ausgabe: 14. Januar, Abgabe: 21. Januar

Hinweis: Die Besprechung findet am 25.01 statt.

Aufgabe 12.1 (4 Punkte) CTL Äquivalenzen

Seien Φ und Ψ zwei beliebige CTL-Formeln. Welche der folgenden Äquivalenzen sind korrekt?

- a) $AX AF \Phi \equiv AF AX \Phi$
- b) $EX EF \Phi \equiv EF EX \Phi$
- c) $AX AG \Phi \equiv AG AX \Phi$
- d) $EX EG \Phi \equiv EG EX \Phi$
- e) $EF EG \Phi \equiv EG EF \Phi$

Aufgabe 12.2 (4 Punkte) CTL System

Stellen Sie sich ein Aufzugssystem für $n > 0$ Etagen vor (nummeriert mit 0 bis $n - 1$). Auf jeder Etage ist eine Aufzugstür mit zugehörigen Rufknopf. Neben den Knöpfen leuchtet jeweils eine Lampe wenn der Aufzug gerufen wird. In der Aufzugskabine sind n Knöpfe für das Fahrziel und entsprechend n Signallichter die anzeigen, auf welche Etage der Aufzug fahren soll. Zur Vereinfachung gilt $n = 4$.

Geben Sie eine Menge von atomaren Aussagen (möglichst wenige) an, die gebraucht werden, um folgende Sachverhalte des Aufzugssystems als CTL-Formeln zu beschreiben und geben Sie die jeweiligen CTL-Formeln an:

1. Die Türen sind *sicher*, d.h. eine Etagentür ist niemals geöffnet wenn die Kabine nicht da ist.
2. Die Anzeigelichter zeigen die momentanen Anfragen für die Kabine korrekt an. Das heißt, jedes mal wenn ein Knopf gedrückt wird gibt es eine Anfrage, die bis zur Erfüllung gespeichert werden muss (falls sie jemals erfüllt wird).
3. Der Aufzug bedient nur die gewünschten Etagen und bewegt sich nicht wenn keine Anfrage existiert.

4. Alle Anfragen werden schließlich erfüllt.

Aufgabe 12.3 (4 Punkte) PROMELA / Spin Model Checker

Diese Aufgabe wird mit dem Programm SPIN (www.spinroot.com) gelöst. Nutzen Sie die Rechner-Pools der Universität, eine ssh-Verbindung ist dafür ausreichend. Loggen Sie sich mit ihrem Pool-Account auf den Rechner `marvin.cs.uni-dortmund.de` ein und führen Sie folgende Kommandos aus:

```
mkdir spin && cd spin
wget http://spinroot.com/spin/Bin/spin516_sunos
chmod u+x spin516_sunos && ln -s spin516_sunos spin
export PATH=$HOME/spin:$PATH
```

Testen sie die korrekte Installation durch Ausgabe der Kommandozeilenoptionen: `spin -help`.

Für die Übung sollen zwei Modelle aus dem Buch von G. Holzmann: *The SPIN Model Checker* ausprobiert werden. Laden Sie bitte dazu das Archiv mit den Beispielen:

```
mkdir examples && cd examples
wget http://spinroot.com/spin/Doc/Book_extras/examples.tar.gz
gunzip * && tar -xf * && cd App_C
```

Hier liegen PROMELA-Modelle zweier Petri-Netze. Sie können sie mit `more petrinet1` die erste Datei betrachten. Wie wird hier das PROMELA-Sprachelement `guard` → `effect` zur Beschreibung der Petri-Netze genutzt?

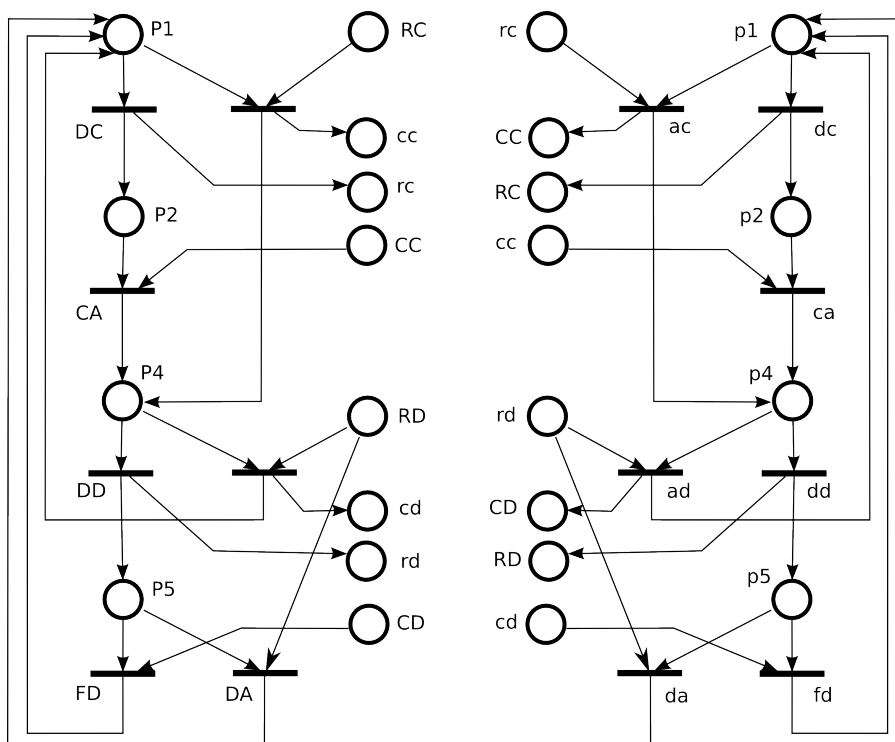
Das erste Modell beinhaltet eine Deadlock-Situation. Nutzen Sie SPIN um mittels Erreichbarkeitsanalyse den Deadlockzustand zu finden:

```
spin -v petrinet1
```

Warum erkennt SPIN hier die Deadlock-Situation? Welche Suchmethode wendet SPIN hier an? Das zweite Petrinetz ist aus dem Papier [1] entnommen. Darin wurde es als frei von Deadlocks bewiesen. Analysieren Sie das Netz mit

```
spin -v petrinet2
```

Welcher Fehler tritt hier auf?



Literatur

- [1] G. Berthelot, and R. Terrat, 'Petri Net Theory for the correctness of protocols,' IEEE Trans. on Comm., Vol COM-30, No. 12, Dec. 1982, pp. 2497-2505