

Übung 8: Model Checking & Timed Automata

Remark: You can download the JAVA-based Uppaal modelling tool from: <http://www.uppaal.com/>, where you also find some tutorials. This high-level modelling tool makes use of Timed Automata and allows you to solve some of the following exercises, but its use is not mandatory.

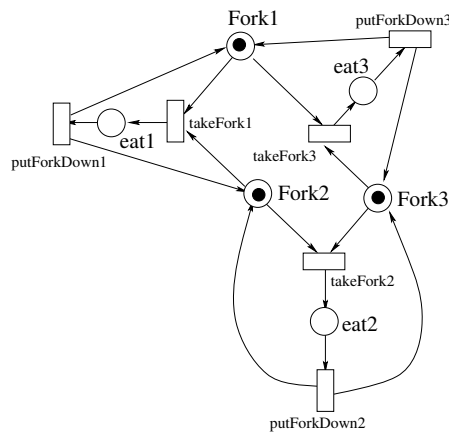


Abbildung 1: Petri-Netz, das das “Dining Philosopher”-Protokoll modelliert

Aufgabe 1: Safety Properties and Invarianten

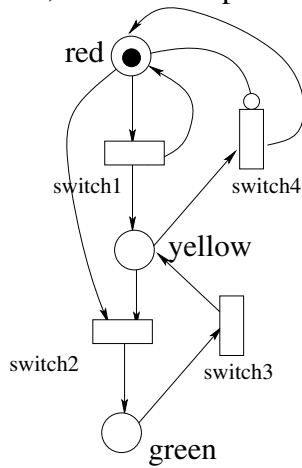
Sei das in Abbildung 1 gezeigte Petri-Netz gegeben. Es modelliert das in der Vorlesung erwähnte “Dining Philosopher” System.

1. Wie können sie die Verklemmungsfreiheit (deadlock- freeness) des Modells überprüfen.
2. Geben Sie ein entsprechende aussagenlogischen Term an, der die Verklemmungsfreiheit beschreibt. (Hinweis: Ein Philosoph wartet, wenn er seine Gabel nicht aufnehmen kann.)
3. Bitte führen Sie die Überprüfung ihrer Invarianten manuell am obigen Modell durch.

Aufgabe 2: LTL-Properties

1. Sei das in Abbildung 2.A gezeigte Petri-Netz gegeben. Es modelliert eine Ampel, vor der Grünphase leuchten die gelbe und rote Lampe gemeinsam. Es sei nun folgende LTL-Formeln zu überprüfen: $\square \diamond green$ und $\diamond \square green$. Bitte führen Sie die entsprechenden Schritte zum MC manuell durch.

(A) Petri-Netz, das eine Ampel modelliert



(B) LTS eines Senders

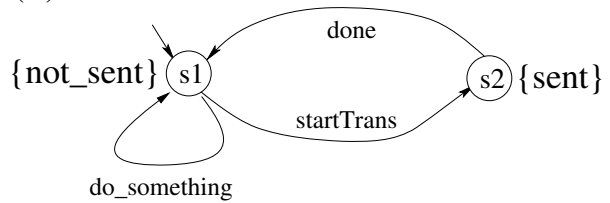


Abbildung 2: Petri-Netz und LTS zu Aufgabe 2

2. Sei das in Abbildung 2.B gezeigte “labelled transition system” (LTS) gegeben. Es gilt nun folgende LTL-Formel zu überprüfen: $\Phi := \square(not_sent \Rightarrow \diamond sent)$. Bitte führen Sie die entsprechenden Schritte zum MC manuell durch.

Aufgabe 3: Taskaktivierungsmuster und Timed Automata

In der Schedulinganalyse verwendet man sog. *PJD*-Modelle. Dabei steht das *P* für Periode der Aktivierung, das *J* für eine (positive) maximale Verzögerung der Aktivierung und das *D* für den minimalen Abstand zweier Aktivierungen. Modellieren Sie nun solche Taskaktivierungsschemata mit Hilfe von Uppaal (oder auf dem Papier), Abbildung 3 diene ihnen dabei als Hilfe, wobei dort gilt $J = 0$ und $D = 0$. D.h. erweitern Sie diesen Automaten sukzessive für die Fälle $J < P, D = 0$ und $J \leq P, D = 0$.

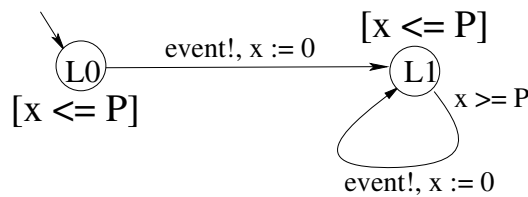


Abbildung 3: TA für ein PJD-Modell mit $J = D = 0$

Aufgabe 4: Scheduling und Timed Automata

4 Prozesse möchten eine gemeinsame Ressource benutzen, es gibt folgende Parameter:

Process	BC	WC	P
P_1	2	2	5
P_2	2	3	17
P_3	3	3	23
P_4	2	3	100

(BC = best case execution time,
WC = worst case execution time,
P = periodic activation)

Im obigen Szenario ist anzunehmen, dass die deadline, d.h. der Zeitpunkt wann der Task mit

seiner Aufgabe auf der Ressource fertig ist, gleich der Periode der Aktivierung ist.

Modellieren Sie das obige System mit Uppaal als nicht präemptives System. Ziel der Modellierung ist es herauszufinden, ob alle Deadlines eingehalten werden können, bzw. das System "schedulable" ist.

Zusatzaufgabe :

Präemptives Scheduling Modellieren Sie ein System mit 2 Tasks, die jeweils ihren kritischen Abschnitt betreten wollen. Hierbei hat allerdings Task 2 die höhere unterbrechende Priorität. D.h. will er seinen Abschnitt betreten, so verdrängt er nicht nur Task 1, sondern dieser muss sogar die Abarbeitung unterbrechen. Was ist könnte hier ein Problem sein und wie können sie es dennoch ohne Stoppuhr lösen?