ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Computer Engineering and
Networks Laboratory

HS 2010 Dr. K. Lampka

Discrete Event Systems

Exercise Sheet 14

1 Computation Tree Logic Model Checking

Remember the following Kripke structure from the last exercise

$$\mathcal{K} := \left\{ \begin{array}{l} \text{States} := \{1,2,3,4\} \\ \mathbb{S}_0 := \{s_0\} = \{1\} \\ \rightarrow := \{(1,3),\ (3,2),\ (2,1),\ (2,4),\ (4,2)\} \\ AP := \{\text{green}, \text{yellow}, \text{red}, \text{black}\} \\ L := \{1 \mapsto \text{red},\ 2 \mapsto \text{yellow},\ 3 \mapsto \text{green},\ 4 \mapsto \text{black}\} \end{array} \right\}$$

along with the following – syntactically correct – CTL formulas:

$$\Omega_1 = \exists\Box\text{green} \qquad \Omega_4 = \forall(\text{black} \bigcup \text{black})$$

$$\Omega_2 = \forall\Box\text{yellow} \qquad \Omega_5 = \forall(\neg\text{yellow} \bigcup (\exists \bigcirc \text{black}))$$

$$\Omega_3 = \forall\Diamond\text{black} \qquad \Omega_6 = \exists(\text{black} \bigcup \text{black})$$

a) Transform the syntactically correct CTL formulas into existential normal form (ENF).

b) Construct the syntax trees for the ENFs of the syntactically correct CTL formulas.

c) Annotate the nodes in the syntax trees with the satisfiability sets $\text{Sat}(\Omega)$ w.r.t. to $\mathcal{K}$.

d) Which of these formulas are satisfied by $\mathcal{K}$, i.e. for which formulas Ω do we have $\mathcal{K} \models \Omega$? Justify your answers.

e) Give counter-examples for the unsatisfiable formulas starting with the universal quantifier.

2 Timed Automata

Remark: You can download the JAVA-based timed model checker Uppaal from: `http://www.uppaal.com`, where you also find some tutorials. This high-level modeling tool makes use of Timed Automata and allows you to solve the following exercises, but its use is not mandatory; you may also solve the exercise, at least the most important parts, with pen and paper only.

2.1 Modeling of task activation patterns with Timed Automata

Scheduling analysis of real-time tasks employs so called PJD traffic models. A PJD-model represents periodic task activations, denoted as task releases, with a jitter. It is defined by the following three paramters: P stands for the period of a task's activation, J refers to the jitter which is a bounded delay by which the task's ideal periodic activation might be delayed. Parameter D refers to the minimal distance of any two consecutive task activations. In addition

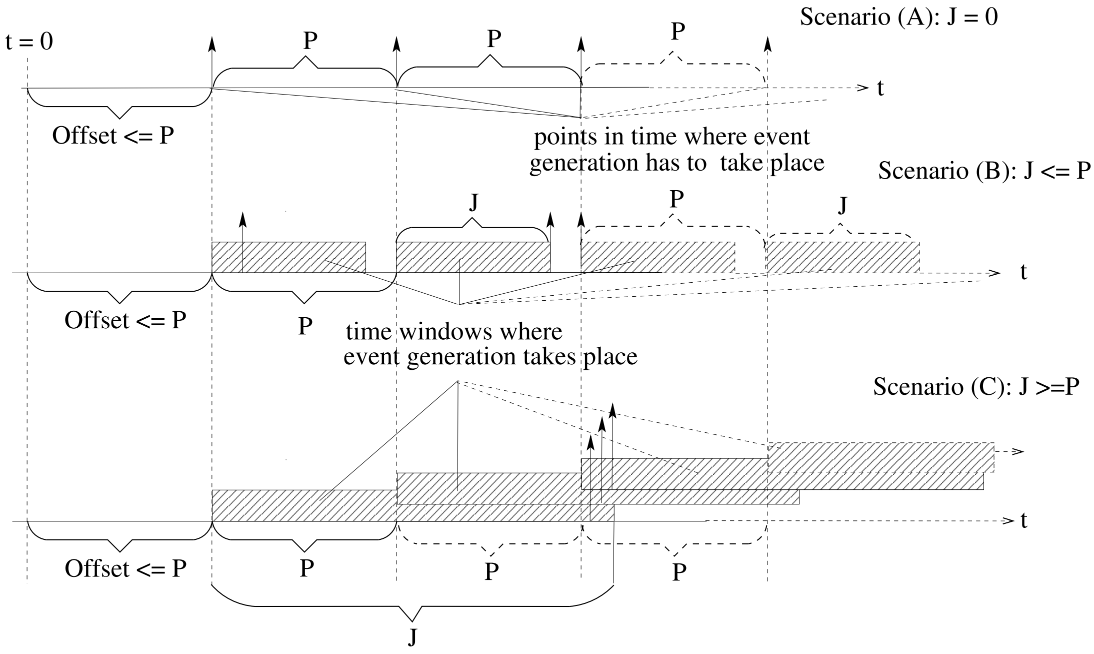

Figure 1: PJD-based activation patterns

the task's activation may be equipped with an initial offset. An example of a PJD activation scheme over the time-line is depicted in Fig. 1, the arrows indicate a task activation. Please note, with scenario (A) the n'th release has to happen exactly at time $\mathit{offset} + n \cdot P$. With scenario (B) and (C) the release has to take place once per hatched area, the exact placement is, however, not known. Such situations are referred to as being non-deterministic, i.e., the choice where to activate the task is non-deterministically chosen by the model checker.
The scenario depicted in Fig. 1 (Scenario (A)), is modeled by the TA of Fig. 2.

Please extend the TA of Fig. 2, such that it implements scenario (B) of Fig. 1 where $J < P, D = 0$ holds.

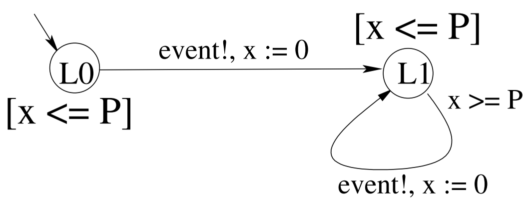

Figure 2: TA modeling the PJD task activation scheme for $J = D = 0$

Supplement: Jitter larger than the period (difficult!)

Extend your solution such that the specified TA models a PJD-pattern where $J > P$ and $D = 0$ holds. Note, you need to track the number of elapsed periods (= number of releases), and the number of pending releases (activations not emitted so far).

2.2 Scheduling with Timed Automata: Modeling

Four tasks are supposed to be executed on a single processing device. These tasks are specified by the following parameters:

Task	BC	WC	P	
P_1	2	2	5	BC = best case execution time,
P_2	2	3	17	WC = worst case execution time,
P_3	3	3	23	P = periodic activation
P_4	2	3	100	

The tasks are activated according to a PJD traffic model with the above given period; for simplicity we assume that $J = D = 0$ holds for now.

a) Please model this scenario as a non-preemptive system. The goal of the modeling and analysis is now to check, whether the tasks can be executed on a single resource without a deadline miss or not. The obtained sequence of task activations which avoids deadline miss, if it exists, is commonly denoted as schedule. Hint: Each task activation should be modeled by a separate TA. The task-activation of task P_i should be triggered by a TA which implements the respective PJD model. Furthermore you also need to model the resource which coordinates and realizes the execution of the tasks. Resource access is granted arbitrarily.

b) The deadline of a task's n'th invocation, i.e., the time the processing of a task has to be finished, equals the task's period; the n'th invocation of a task has to be processed at time $\mathit{offset} + (n+1)P$.

(i) Which property allows one to assert this behaviour?

(ii) Please specify an obsever TA which flags the violation of this property.

Hint: A deadline is is missed if there is more than one task activation queued in the system for being processed.

2.3 Scheduling with Timed Automata: Model-checking

For the following exercise you have to use Uppaal.

a) Is the system from above free of deadline misses (schedulable)?

b) Extend the system from above such that task P_i has priority over task P_{i+1}. Is this system still free of deadline misses (schedulable)? In fact you can answer the question without modeling this scenario, as the choice which task to serve next is non-deterministically taken, i.e., it is left to the model checker. Why is an additional analysis of this scenario not necessary?

c) Assume that the priorities defined above are present and that the deadlines are dropped, resp. set to ∞. The task releases may show now the following jitters: $J_1 := 10, J_2 := 14, J_3 := 20, J_4 := 97$.

(i) What is the maximum number of tasks activations (= number of issued releases) of task P_1 waiting in the system for being processed?

(ii) What is the maximal delay between a release of task 1 and the termination of the associated execution. Hint: for solving this question you need to specify a respective observer TA.

d) Given that $J_1 = 20$. Is it possible that there are more than 5 pending releases for task T_2, T_3 and T_4 in the system?