Written examination, May 31, 2021

Course: Modelling and Analysis of Real-Time Systems Course no. 02224

Aids allowed: All

Exam duration: 2 hours

Weighting:	PROBLEM 1:	approx. 20 %	PROBLEM 3:	approx. 40 %
	PROBLEM 2:	approx. 40 %		

You are encouraged to briefly justify your answers. If you happen to be in doubt about the precise meaning of a question, you should write down how you choose to understand it.

PROBLEM 1 (approx. 20 %)

For each of the below Uppaal queries `q1`, …, `q4` you should provide two timed automata P_i^{t} and P_i^{f}, $1 \leq i \leq 4$, so that

- `q`i is not satisfied by P_i^{f} and
- `q`i is satisfied by P_i^{t}.

Aim at simple automata constructions.

In the following queries, Q is a placeholder for the automata P_i^{t} and P_i^{f}, for $1 \leq i \leq 4$.

`q1: A<>` $Q.\mathrm{L}_1$ `and` $x > 2$.

`q2: E[]` $(Q.\mathrm{L}_1$ `and` $x \leq 2)$ `or` $Q.\mathrm{L}_2$.

`q3:` $Q.\mathrm{L}_1$ `-->` $(Q.\mathrm{L}_2$ `and` $x = 2)$.

`q4: A[]` $Q.\mathrm{L}_1$ `imply` $Q.\mathrm{L}_2$.

PROBLEM 2 (approx. 40 %)

Question 2.1:

(a) Construct an untimed automaton A that has

- 4 locations, named $L_0, \ldots, L_3$, where L_0 is the start location and
- 5 transitions,

so that every location is reachable from the start location.

b) Turn A into a timed automaton T_A by adding

- two clocks, named x and y, each of which must be reset once,
- two invariants, and
- three guards,

so that every location is reachable also when timing constraints are taken into account. Furthermore, some runs should allow time to progress towards infinity.

Consider now the timed transition system tts_{T_A} underlying the timed automaton T_A.

Question 2.2:

(a) Give an informal explanation of the notion *state* of tts_{T_A}.

(b) Give three example states of tts_{T_A}.

(c) Give an informal explanation of the notion *transition* of tts_{T_A}.

(d) Give two example transitions of tts_{T_A}, where one should be a discrete transition and the other should be a progress of time.

(e) A timed transition system is in general an *infinite state* system. Give a brief explanation why this is the case. Illustrate why tts_{T_A} is an infinite state system.

Consider now the region automaton for the timed automaton T_A.

Question 2.3:

(a) Give an informal explanation of the notion *clock region.*

(b) Give examples of the different kinds of clock regions, on the basis of T_A.

(c) Give an informal explanation of the notion *transition in a region automaton.*

(d) Give two examples of transitions in the region automaton for T_A, where one should reflect a discrete transition and the other a progress of time.

(e) The region automaton for T_A is a finite automaton. Give an explanation why this is the case.

PROBLEM 3 (approx. 40 %)

For a real-time system to be run on a single-processor computer four periodic tasks a, b, c, and d are considered. The system is supposed to run a selection of three out of the four tasks. The tasks have the following parameters:

	T	C
a	5	2
b	16	4
c	36	6
d	40	10

The deadline of each task is equal to its period. Initially, the tasks are assumed to be independent.

Question 3.1:

(a) Calculate the load of each task. [In percentages with one decimal.]

(b) Based on these loads, determine for which of selections of three tasks schedulability can be guaranteed for preemptive fixed-priority scheduling (FPS) and explain your results.

For the questions below, the first three tasks, a, b, and c, are selected for execution on the system. Furthermore, these three tasks are to be scheduled by a preemptive fixed-priority scheduler (FPS) using rate monotonic priority assignment.

Question 3.2:

Calculate the response time for each of the three tasks. Show all intermediate response time estimates.

Now, two protected, shared resources M_1 and M_2 are to be used by the tasks as follows: Tasks a and c both use M_1 for 1 time unit of their computation time. Tasks b and c both use M_2 for 3 time units of their computation time. Task c does not use M_1 and M_2 at the same time.

Recall that the *blocking time* B_t of a task t is the maximum time the task t may experience being blocked waiting for *lower priority tasks* to release resources (not including any preemption by tasks with higher priority than task t).

Question 3.3:

(a) State the blocking time B_t for each task t and justify your results.

[Notice that blocking of task a may be subject to *priority inversion*.]

(b) Explain why the task set using M_1 and M_2 is not schedulable.

(c) Explain in your own words the principle of *priority inheritance*.

(d) Assume that priority inheritance is applied to the system. Determine the new blocking time B_t for each task t and calculate the response times. Is the system schedulable under these conditions?