TECHNICAL UNIVERSITY OF DENMARK

Written examination, May 29, 2019

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

Aids allowed: All written works of reference

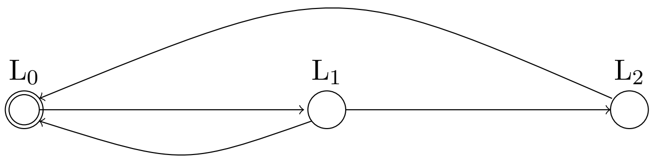
Exam duration: 2 hours

Weighting:	PROBLEM 1:	approx. 25 %	PROBLEM 3:	approx. 40 %
	PROBLEM 2:	approx. 35 %		

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. 25 %)

Consider the following timed automaton P with three location L_0, L_1 and L_2.



The timed automaton P

Question 1.1: Extend this automaton P so that

- a transition from L_0 to L_1 is possible just when the automaton has stayed in L_0 for at least 6 seconds (use a clock `x` to express this property),
- a stay in L_1 lasts at least 3 seconds and at most 7 seconds (the clock `x` should be reused to express this property),
- a transition from L_1 to L_2 is possible exactly 13 seconds after the most recent entry to location L_0 (the initial one included), and
- a stay in location L_2 lasts for at most 2 seconds.

Question 1.2:

(a) Does your extension of P satisfy the query: $\mathtt{P.L_2\ \text{-->}\ P.L_1}$? Justify your answer.

(b) Does your extension of P satisfy the query: `E<> x>12`? Justify your answer.

(c) Does your extension of P satisfy the query: $\mathtt{A[]\ ((P.L_1\ and\ x==7)\ imply\ (P.L_0\ or\ P.L_2))}$? Justify your answer.

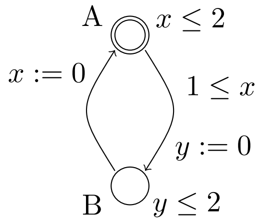
(d) Does your extension of P satisfy the query $\mathtt{A\text{<>}\ P.L_0\ and\ x==3}$? Justify your answer.

(e) Explain in your own words the meaning of a query of the form: `true -->` F.

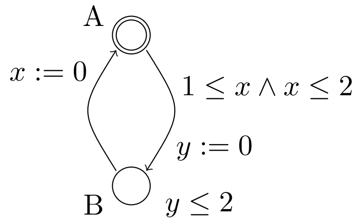
(f) Does your extension of P satisfy the query: $\mathtt{true\ \text{-->}\ P.L_0}$? Justify your answer.

PROBLEM 2 (approx. 35 %)

We shall consider the following two timed automata `P` and `Q` in this problem.



The timed automaton `P`



The timed automaton `Q`

Question 2.1:

(a) Give a TCTL formula in the form of an Uppaal query that distinguishes `P` and `Q` in the sense that it is satisfied for `P` and not satisfied for `Q`.

[Location propositions may be expressed simply as `A` or `B` leaving out the process name.]

(b) Give a TCTL formula in the form of an Uppaal query that distinguishes `P` and `Q` in the sense that it is not satisfied for `P` and satisfied for `Q`.

[Same remark as in (a) regarding location propositions.]

(c) Provide a solution to (b) without using location propositions. Justify your answer.

Question 2.2:

(a) Give two states of the transition system underlying the timed automaton `P`.

(b) Give two transitions of the transition system underlying the timed automaton `P`, where one transition should be a discrete transition and the other a progress of time.

Question 2.3:

(a) Illustrate the clock regions for `P`.

(b) How many clock regions are there? Justify your answer.

(c) What is the maximal number of states the region automaton for `P` could have? Justify your answer and present three of the states.

(d) Provide two transition of the region automaton for `P`, where one transition should illustrate a delay of time and the other a change of location (discrete step).

PROBLEM 3 (approx. 40 %)

The questions in this problem can be solved independently of each other.

A real-time system to be run on a single-processor computer has three periodic tasks a, b and c with the following parameters:

	T	C
a	8	4
b	12	3
c	25	5

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

Question 3.1:

(a) Calculate the load of each task and the total load of the system.

(b) Determine whether schedulability of the task set using fixed-priority scheduling (FPS) can be guaranteed based on the total load.

In the given system, the set of task is to be scheduled by a preemptive fixed-priority scheduler (FPS) using rate monotonic priority assignment.

Question 3.2:

Calculate the response time of each task.

Now, two protected, shared resources M_1 and M_2 are to be used by the tasks as follows: Tasks a and b both use M_1 for 1 time unit of their computation time. Tasks a and c both use M_2 for 2 time units of their computation time. Task a 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) Explain why the blocking times of tasks b and c are both 0.

(b) Determine the blocking time B_{a} of task a and illustrate using a scheduling scenario how this worst-case amount of blocking may occur.

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

(c) Assume that the principle of *priority inheritance* is applied to the system. Determine the new blocking time B_{a} of task a. Is the system schedulable under these conditions?