Written examination, May 19, 2017

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

Aids allowed: All written works of reference

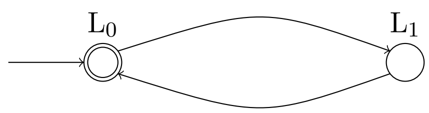
Exam duration: 2 hours

Weighting: PROBLEM 1: approx. 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 %)

In this problem we consider the following timed automaton Q with two locations: L_0 and L_1, and one clock *time* that should used to measure the elapsed time (since system start).



The timed automaton Q

Question 1.1: Extend this timed automaton so that:

- A stay in location L_0 lasts for at least one time unit and at most 5 time units.
- A stay in location L_1 lasts for exactly one time unit.

Question 1.2: Formulate Uppaal queries for the below properties. Notice that you may have to extend your automaton further to express some of these queries.

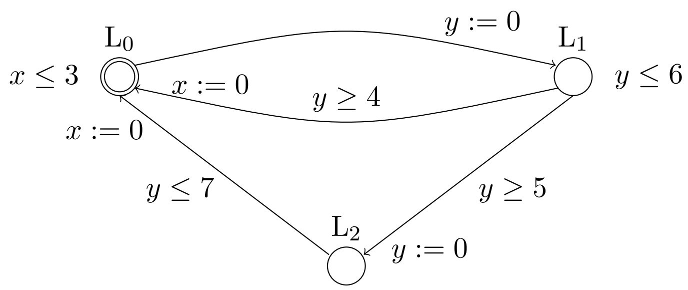
(a) It is possible that the automaton never stays for more than 2 time units in L_0.

(a) It is possible that L_1 is entered 3 times within the first 5 time units.

(c) It is inevitable that at most 6 time units elapse from one entry to L_1 to the next.

PROBLEM 2 (approx. 40 %)

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



The timed automaton P

Question 2.1: Give a brief description of the timing constraints relating to each of the three locations L_0, L_1 and L_2.

Question 2.2:

(a) Does P satisfy the query: `A<> P.L`$_1$?

(b) Does P satisfy the query: `A<> P.L`$_2$?

(c) Does P satisfy the query: `E<> P.L`$_2$ `and` $x - y \leq 5$?

(d) Does P satisfy the query: `P.L`$_2$ `--> (P.L`$_1$ `and` $y > 0$)?

Answers to the above questions must be accompanied with brief justifications.

Question 2.3: There is a run in P having infinitely many visits to L_0 and L_1 but no visit to L_2. How could Uppaal be used to support a formal argument that such a run exists?

Question 2.4: The semantics of a timed automaton is given in terms of a transition system. Give a brief informal account of the states and transitions of a transition system underlying a timed automaton. Supply your answer with examples based on P.

Question 2.5: A region automaton (or region transition system) is a finite state automaton constructed on the basis of a timed automaton. Give a brief informal account on the of the states and transitions of a region automaton. Supply your answer with examples based on P.

PROBLEM 3 (approx. 40 %)

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

	T	C
a	4	1
b	10	5
c	32	4

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.

(b) Based on the total load, which of the two scheduling principles *fixed-priority scheduling* (FPS) and *earliest deadline first* (EDF) can be concluded to be feasible for scheduling the given set of tasks?

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 for each of the three tasks.

Now, a shared protected resource M is to be used by all tree tasks. Task a is going to use M for all of its 1 unit of computation time, task b is going to use M for 2 of its its 5 computation time units and task c is going to use M for 1 of its 4 computation time units. For task b, the usage takes place at the end of its computation.

Recall that the *blocking time* B_t of a task t is the maximum time the task t may experience being suspended while *lower priority tasks* execute.

Question 3.3:

(a) State the blocking times of tasks b and c.

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

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

(c) Explain why the task set using M is not schedulable.

(d) Assume that *priority inheritance* is applied to the scheduling.

State the new blocking time B_t for each task t, and determine whether the set of tasks has now become schedulable.