

Real-Time Systems

<http://swt.informatik.uni-freiburg.de/teaching/SS2012/rtsys>

Exercise Sheet 4

Early submission: Monday, 2012-07-16, 12:00 Regular submission: Tuesday, 2012-07-17, 12:00

Exercise 1: Regions

(3/20 Points)

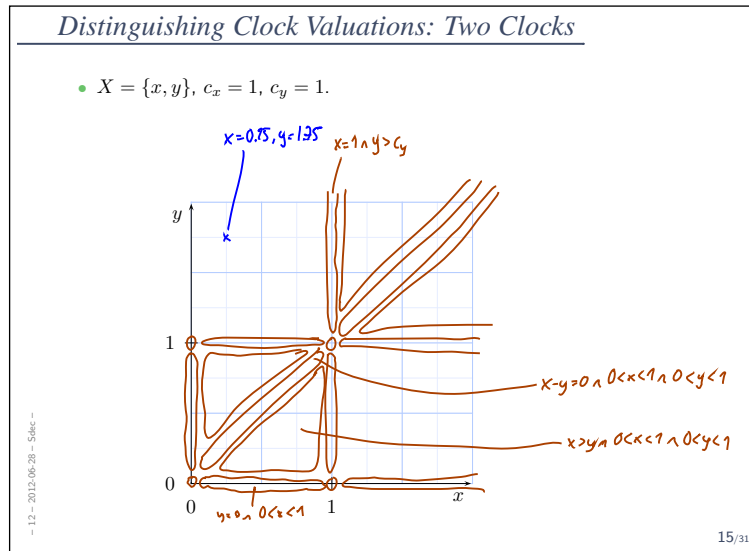


Figure 1: Partial Indication of Regions.

Recall that we started to indicate the equivalence classes on clock valuations of  $X = \{x, y\}$  in the graph shown in Figure 1. A point  $(p, q)$  in the graph represents the unique clock valuation  $\{x \mapsto p, y \mapsto q\}$ .

Please complete the drawing by indicating all equivalence classes according to the definition of equivalence from the lecture. As usual, convince the tutors of the correctness of your proposal.

Exercise 2: Region Construction [1]

(5/20 Points)

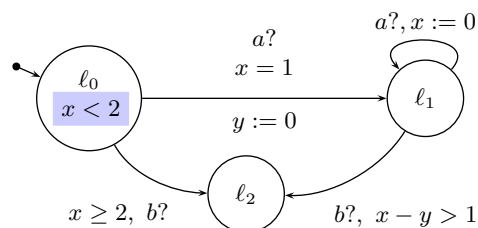


Figure 2: Timed Automaton for Exercise 2.

Consider the timed automaton  $\mathcal{A}$  in Figure 2. In the tutorial, we had the impression that location  $l_2$  is not reachable. Prove this statement by constructing the region automaton.

Note: this is not exactly the same automaton from the previous exercise sheet!

### Exercise 3: Deadlock

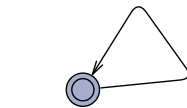
(8/20 Points)

- (i) Please give (possibly from (correctly cited) literature) an exact formal definition of deadlock in Uppaal, i.e. please explain (formally) when exactly does

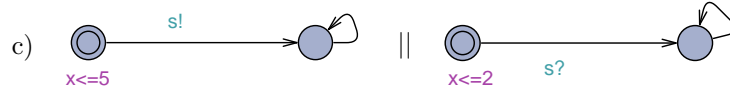
$$\mathcal{A} \models E \langle \rangle \text{deadlock}$$

hold using the definitions from the lecture.

Consider the following examples:



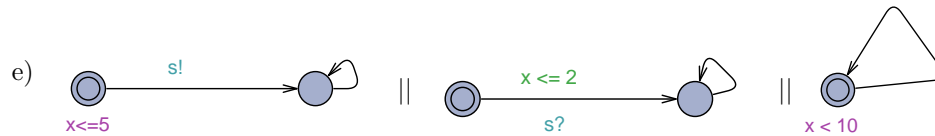
b)  $x < 10$



c)  $x \leq 5$



d)  $x \leq 5$



e)  $x \leq 5$

$x < 10$

Do they have a deadlock according to your definition (wherever it comes from)?

And according to Uppaal (i.e., what does Uppaal's deadlock check yield)? (4/8)

- (ii) How does deadlock relate to livelock? (2/8)
- (iii) What is checking for deadlocks good for? (2/8)

### Exercise 4: Model-Checking with Uppaal

(4/20 Points)

Consider the Off/Light/Bright model from Exercise Sheet 4.

- (i) Use the model checker to verify whether the original user can reach the Bright location. (0.5/4)
- (ii) Use the model checker to verify whether your modified user from Sheet 4, Exercise 2, part (iii) cannot reach the Bright location as requested. (0.5/4)
- (iii) Check whether the original user is able to keep the lamp at location Bright for more than 5 time units. (1.5/4)
- (iv) Check whether the original user is able to switch the lamp to Bright twice. (1.5/4)

Explain your approach.

### References

- [1] Ernst-Rüdiger Olderog and Henning Dierks. *Real-Time Systems - Formal Specification and Automatic Verification*. Cambridge University Press, 2008.