Prof. Dr. A. Podelski, Dr. B. Westphal
D. Gauß

Sommersemester 2013

Real-Time Systems

http://swt.informatik.uni-freiburg.de/teaching/SS2013/rtsys

Exercise Sheet 5

Early submission: Tuesday, 2013-07-02, 10:00   Regular submission: Wednesday, 2013-07-03, 10:00

Exercise 1: Regions (3/20 Points)

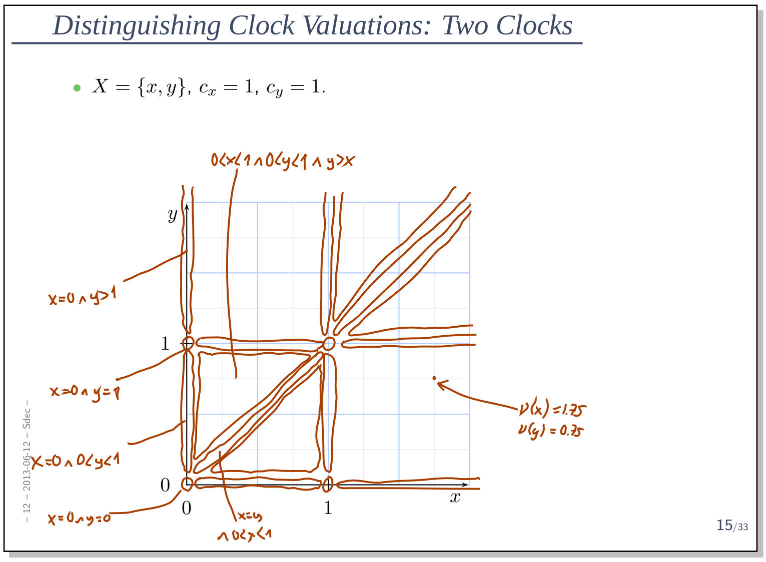


Figure 1: Partial Indication of Regions.

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 $\{x \mapsto p, y \mapsto q\}$. The equivalence classes shown in Figure 1 are actually not correct.

- What is wrong? Why is it wrong? (The correct equivalence classes are in the book [2]). (3)
- Outline which equivalence classes we get if we have $c_x = 2$. (2)

As usual, convince the tutors of the correctness of your proposal.

Exercise 2: Region Construction [2] (5/20 Points)

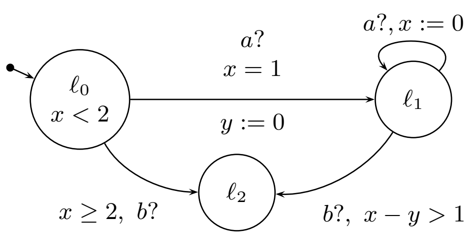


Figure 2: Timed Automaton for Exercise 2.

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.
Hint: you need not present all configurations of $\mathcal{R}(\mathcal{A})$ if you explain why the ones, that you do present, are sufficient for the exercise.

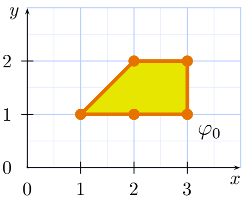


Figure 3: Zone φ_0 for Exercise 3.

Exercise 3: Zone Construction [2] (3/20 Points)

Compute

$$\text{Post}_e(\ell_0, z)$$

for the zone φ_0 given by Figure 3 and for both edges originating at ℓ_0; give the intermediate steps up to φ_5.

What can you conclude about the reachability of ℓ_1 and ℓ_2?

You may represent zones graphically or symbolically.

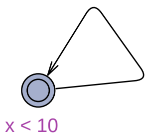
Exercise 4: Deadlock (5+5/20 Points)

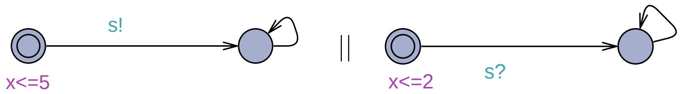
(i) Please give (possibly from (correctly cited) literature) an exact formal definition of deadlock in Uppaal [1], i.e. please explain (formally) *using the notions and definitions from the lecture* when exactly a network of timed automata satisfies

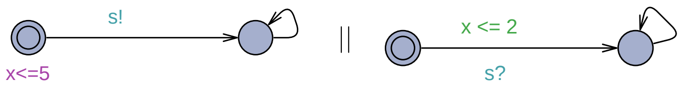
`E <>deadlock`.

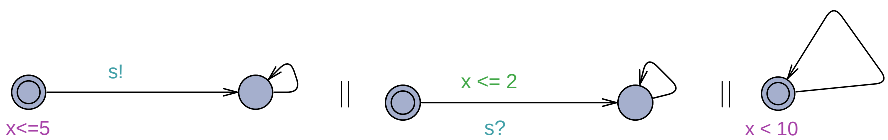
Consider the following examples:

a) 

b) 

c) 

d) 

e) 

Do they have a deadlock according to your definition?
And according to Uppaal (i.e., what does Uppaal's deadlock check yield)? (3/5)

(ii) How does deadlock relate to timelock? (1/5)

(iii) What is checking for deadlocks good for? (1/5)

(iv) Can Uppaal check for timelock? What would checking for timelock be good for? (5 Bonus)

Exercise 5: 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. (1/4)

(ii) Use the model checker to verify that your modified user from Sheet 4, Exercise 2, part (iii) cannot reach the Bright location as requested. (1/4)

(iii) Check whether the original user is able to keep the lamp at location Bright for more than 5 time units. (1/4)

(iv) Check whether the original user is able to switch the lamp to Bright twice. (1/4)

Explain your approach.

References

[1] Gerd Behrmann, Alexandre David, and Kim G. Larsen. A tutorial on uppaal 2004-11-17. Technical report, Aalborg University, Denmark, November 2004.

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