Real-time and Probabilistic Systems Verification
Assignment 2

Luca Tesei

Academic Year 2016/17

Instructions

Reply to all questions justifying your answers as clearly as possible. Send an electronic (also handwritten and scanned, but readable) version to

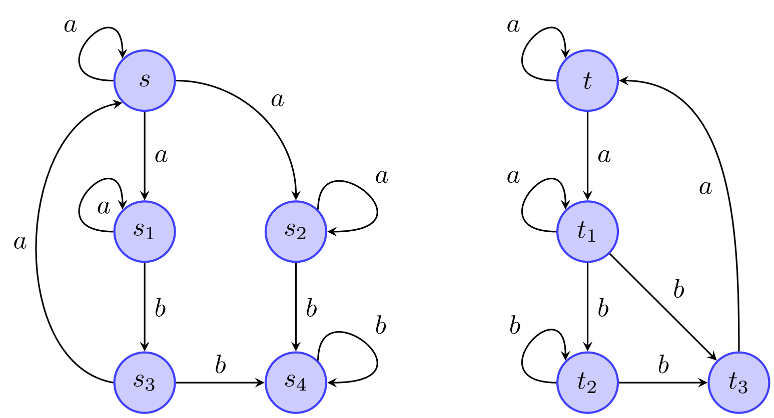
`luca <dot> tesei <at> unicam <dot> it`

by

10th February 2017 23.59

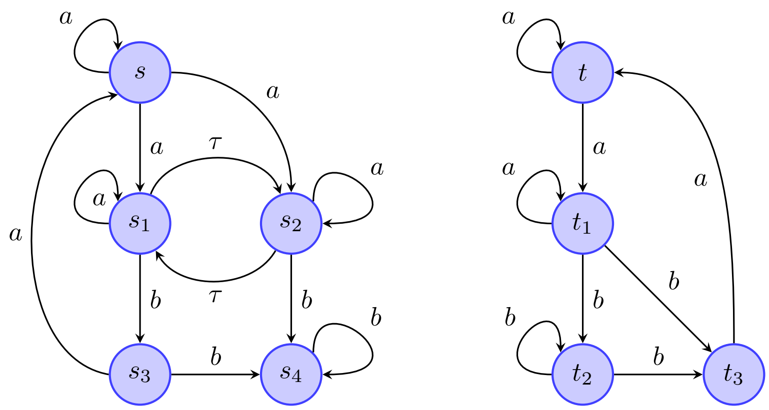
Exercise 1 - 30%

Consider the following labelled transition system.



1. Determine whether or not $s \sim t$. Justify your answer formally.
2. Calculate $\sim$, the largest bisimulation relation in the given labelled transition system, as the maximum fix-point of the appropriate function $R\colon 2^{Proc \times Proc} \to 2^{Proc \times Proc}$.
3. Find all the states that satisfy the following HML formulas:
   (a) $\langle a\rangle\langle a\rangle[b][b]\langle a\rangle tt$
   (b) $\langle a\rangle\langle a\rangle\langle b\rangle[b]\langle a\rangle tt$
   (c) $[a]\langle a\rangle[b]\langle b\rangle tt$
   (d) $[a][a]\langle b\rangle[b] ff$

Exercise 2 - 10%



1. Determine whether or not $s \approx t$. Justify your answer formally.

Exercise 3 - 10%

Express the following properties using HML with recursion:

1. Whenever it is possible to make an a it is possible to also make a b.
2. It is possible to reach a state in which a can be done and b can not be done.
3. In every evolution of the system, sooner o later, action a can not be done.
4. There is a way along the system in which b is always disabled.
5. Whenever a is done, then eventually b will be done.

Exercise 4 - 20%

Consider the following definitions of Timed CCS processes:

$$P_1 = \epsilon(2).\epsilon(1).\bar{b}.P_1 + \epsilon(2).a.P_1 + \epsilon(1).c.P_1$$
$$P_2 = \epsilon(2).b.P_2 + \epsilon(3).\bar{a}.P_2 + \epsilon(4).d.P_2$$
$$P_3 = (P_1 \mid P_2)\backslash\{a, b\}$$

1. Determine whether or not there is a timed trace of the form $(5, c)\cdots$
2. Determine if there exists at least a timed trace in which d can be observed at some timestamp.

Justify your answers formally, giving at least two full derivations.

Exercise 5 - 30%

Consider the following scenario.

A closed parking place has NP parking places and NE entrances. There are then NC cars (where NC > NP) allowed to use the parking place. Whenever a car is *away*, it can decide at any moment to try entering the parking place. After this decision, it has 1 minute to select one of the entrances and send to it an approach signal. Then, the entrance has 1 minute to reply: it will allow the car in if there are still free parking places or it will send the car away if the are no free parking places.

If the car gets inside, then it has 2 minutes to park (thus decreasing the number of free places) or to realise that there are actually no free parking places (it checks the same variable checked by the entrance one minute before...). If the latter case occurs, then the car has further 2 minutes to go away, otherwise the car is *parked* and can stay in the parking place as much as it wants before going away and freeing its parking place.

1. Model the scenario in UPPAAL. Instantiate the constant numbers NP, NE and NC with values that permit to execute the following checks in reasonable time.
2. Check that your model is deadlock free
3. Check that the number of parked cars is always between 0 and NP
4. Check that it is *not* possible to guarantee that if a car tries to park then it will always find a place
5. Check that it is possible that a car is allowed to enter the parking place but then it actually finds that there are no free places
6. Discuss if and how the previous problem can be avoided