Prof. Dr. Ir. Dr. h. c. Joost-Pieter Katoen

Jip Spel, Tim Quatmann

Modeling and Verification of Probabilistic Systems

— Exercise Sheet 4 —

Notes:

- The exercise sheets need to be solved in groups of 2 students.
- Hand in the solution until 14.11.2018 before the exercise class.
- We do not accept solutions via L2p or email.
- Write your names and matriculation numbers on the front page and staple all pages.

Exercise 1 (Weak Until Theorem): (10 Points)

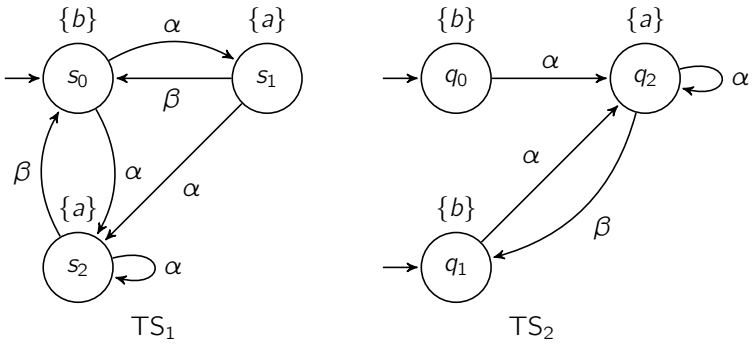
Prove the following theorem: For any finite DTMC it holds that

$$\mathbb{P}_{=1}(\Diamond a) \quad \equiv \quad \forall((\exists \Diamond a)\, \mathrm{W}\, a)$$

where W is the weak until operator defined by $\phi\ \mathrm{W}\,\psi = (\phi\ \mathbf{U}\ \psi) \vee \Box\phi$.

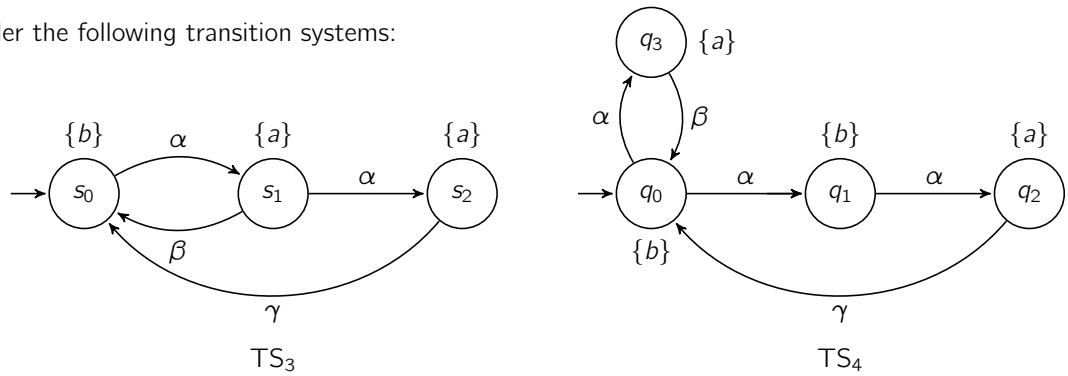
Exercise 2 (Bisimulation): (10 + 10 + 10 = 30 Points)

a) Consider the following transition systems:



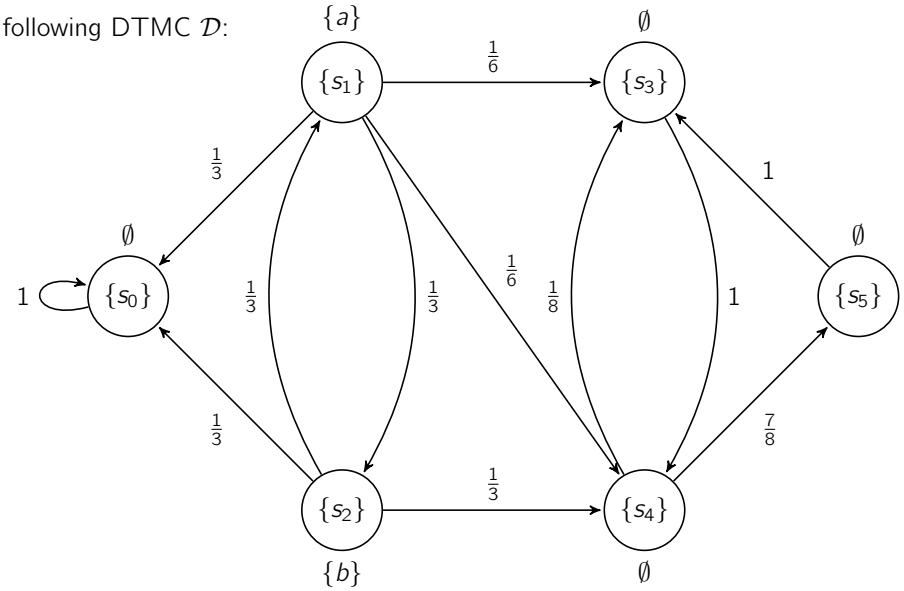
Depict if TS_1 and TS_2 are strongly bisimilar. Justify your answer by either providing a bisimulation relation or a counterexample.

b) Consider the following transition systems:



Depict if TS_3 and TS_4 are strongly bisimilar. Justify your answer by either providing a bisimulation relation or a counterexample.

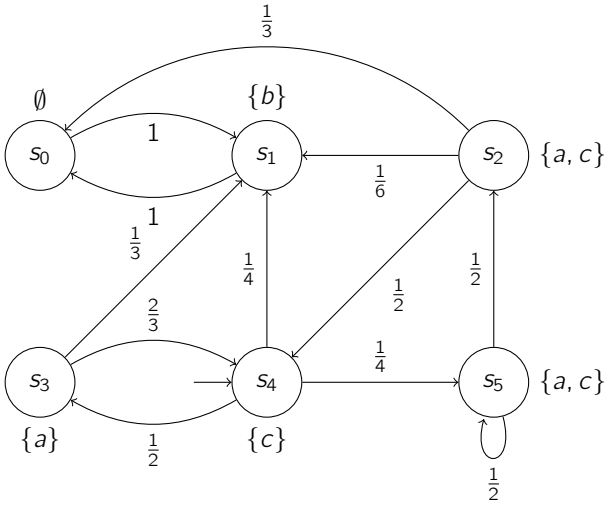
c) Consider the following DTMC $\mathcal{D}$:



Give $\mathcal{D}/\sim_p$, i.e. the quotient of $\mathcal{D}$ under probabilistic bisimulation

Exercise 3 (PCTL Modelchecking): (20 Points)

Consider the DTMC D'.



Determine the set $Sat(\Phi)$ where $\Phi = \mathbb{P}_{<\frac{3}{4}}(\bigcirc \mathbb{P}_{\geq \frac{1}{3}}(a\ \mathbf{U}^{\leq 3}\ (b \vee \neg c)))$ using the algorithm from the lecture. Please give intermediate results and explain your reasoning. A full computation, however, is not needed.

Exercise 4 (Fairness): (15 Points)

We call an infinite DTMC $\mathcal{D} = (S, \mathbf{P}, \iota_{init}, AP, L)$ *finitely* branching if $\text{Post}(s) = \{t \in S \mid \mathbf{P}(s,t) > 0\}$ is finite for all states $s \in S$. For a path $\pi \in \text{Paths}(\mathcal{D})$ let $\inf(\pi) = \{s \mid \pi \text{ visits } s \text{ infinitely often}\}$.
We consider a naive extension of fair CTL for infinite, finitely branching DTMCs:

A path $\pi \in \text{Paths}(\mathcal{D})$ is called *fair* iff $\text{Post}(s) \subseteq \inf(\pi)$ for all $s \in \inf(\pi)$.

Show that under this fairness condition, the claim

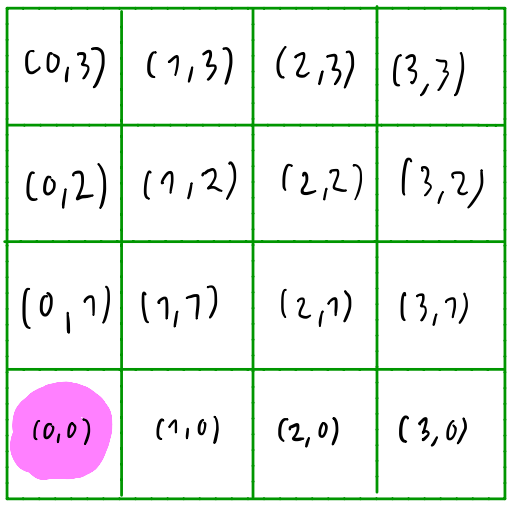
$$s \models \mathbb{P}_{=1}(\Diamond a) \iff s \models_{fair} \forall \Diamond a$$

does not hold for infinite but finitely branching DTMCs.

Exercise 5 (Cleaning Robot): (15 + 10 = 25 Points)

You have been hired by a company that produces cleaning robots. The flagship of your new company is the `ranDormCleaner` which is a cleaning robot that behaves as follows:

- The robot always cleans a square-shaped room with 4x4 tiles (as sketched below).
- Initially, the robot is placed on the tile at the bottom left corner of the room.
- The next action of the robot is determined randomly: With probability 0.25 it moves to the right, left, top, or bottom tile, respectively. If there is a wall in the corresponding direction, the position of the robot does not change.
- Every such step of the robot drains the battery by 1%, including the steps where the robot does not change its position because it hits a wall.
- A tile is cleaned as soon as the robot stayed on it for one step.



a) Your first task at the company is to determine the probability that starting with a fully charged battery (100%), the robot cleans every tile in the room. To this end, write a PRISM program (cf. Exercise Sheet 3) that models the robot and invoke Storm to compute the desired step-bounded reachability probability.

Hints:

- Make sure that the probabilities can be changed easily (this is necessary for exercise b)
- Use integer variables x and y over the domain $\{0, 1, 2, 3\}$ to model the current position of the robot.
- One way to model the fact that the tile at position (x, y) has been cleaned, is to introduce a third variable c over the domain $\{0, 1, \ldots, 2^{4 \cdot 4} - 1\}$. This variable is interpreted as a vector of bits where the bit at position $x \cdot 4 + y$ is 1 iff the tile at position (x, y) is clean. One can set the bit for the tile at position (x, y) to 1 by setting

$$c' := \begin{cases} c + 2^{x \cdot 4 + y} & \text{if } \lfloor c / (2^{x \cdot 4 + y} \rfloor \bmod 2 < 1, \\ c & \text{otherwise.} \end{cases}$$

  In the PRISM syntax, this corresponds to:

```
(c'= mod(floor(c/pow(2,(x*N+y))), 2) < 1 ? c + pow(2,(x*N+y)) : c)
```

- Assuming that the states where all tiles are cleaned are specified with a label "`goal`" and that the PRISM program is stored in the file `robot.prism`, the invocation of Storm is as follows:

```
./storm --prism robot.prism --prop 'P=? [ F<=100 "goal"]'
```

  You might also append the option `-explchecks` to better debug your PRISM program.

- Do some sanity checks, e.g.,
  - it is not possible to clean the room with only 15 steps, so the query `'P=? [ F<=15 "goal"]'` should yield probability 0
  - Similarly, the query `'P=? [ F<=16 "goal"]'` should yield a positive (but small) probability

b) The customers aren't happy with the `ranDormCleaner`. A Software Update should iron out the problems. The company asks you to tweak the probabilities with which the robot chooses the next tile it moves to.

Find a probability distribution between the four actions (left, right, up, down) that increases the probability from part a). Write down the best distribution you were able find together with the resulting probability.

The distribution should not depend on the current position of the robot, the tiles cleaned so far, or the amount of drained battery. You can, however, assume that the robot always starts at the bottom left.