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

Jip Spel, Tim Quatmann

Modeling and Verification of Probabilistic Systems
— Exercise Sheet 3 —

Notes:

- The exercise sheets need to be solved in groups of 2 students.
- Hand in the solution until 07.11.2018 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 (DBA and DRA): (12 + 3 + 15 = 30 Points)

Let $AP = \{a, b, c\}$, consider the following properties:

- $P_1 = \Box b \wedge \Box\Diamond(a \vee c)$.
- $P_2 = (\Diamond\Box(a \wedge \neg b)) \vee (\Diamond\Box(\neg a \wedge b))$.

a) Give a DBA for P_1.

b) Give a DRA for P_1, make clear what the acceptance condition is.

c) Give a DRA for P_2.

Exercise 2 (PCTL Equivalences): (10 + 10 + 10 = 30 Points)

Prove or disprove the following PCTL equivalences.

a) $\mathbb{P}_{=1}(\bigcirc \mathbb{P}_{=1}(\Box a)) \equiv \mathbb{P}_{=1}(\Box \mathbb{P}_{=1}(\bigcirc a))$

b) $\mathbb{P}_{>0.5}(\bigcirc \mathbb{P}_{>0.5}(\Diamond a)) \equiv \mathbb{P}_{>0.5}(\Diamond \mathbb{P}_{>0.5}(\bigcirc a))$

c) $\mathbb{P}_{=1}(\bigcirc \mathbb{P}_{=1}(\Diamond a)) \equiv \mathbb{P}_{=1}(\Diamond \mathbb{P}_{=1}(\bigcirc a))$

Exercise 3 (step-bounded properties): (8 + 12 = 20 Points)

For a DTMC $\mathcal{D}$ with set of states S, two sets $G, H \subseteq S$, and $k \in \mathbb{N}$ recall the following definitions:

$$\Diamond^{\leq k} G = \{s_0 s_1 \cdots \in \text{Paths}(\mathcal{D}) \mid \exists i \leq k\colon\ s_i \in G\}$$
$$\Box^{\leq k} H = \{s_0 s_1 \cdots \in \text{Paths}(\mathcal{D}) \mid \forall i \leq k\colon\ s_i \in H\}$$

a) Characterize the sets $\Diamond^{\leq k} G$ and $\Box^{\leq k} H$ using only countable unions/intersections as well as complements of cylindersets. You do not have to prove that your expressions are correct.

b) Formally prove that for all $G \subseteq S$, and $S_? := \{s \in S \mid \Pr(s \models \Diamond G) > 0\} \setminus G$:

$$\Pr(\Diamond G) - \Pr(\Diamond^{\leq k} G) \leq \Pr(\Box^{\leq k} S_?)$$

Exercise 4 (Storm): (12 + 8 = 20 Points)

For this exercise you need to install the model checker Storm. There are several possibilities:

- Download a Virtual Machine (VM) with a pre-installed version of Storm at

  `https://rwth-aachen.sciebo.de/s/GBeq4uYJCnxBpAw/download`

  You can open the downloaded .ova file with, for example, VirtualBox (`https://virtualbox.org`). The username and password are both `storm`. Open a terminal in the VM and type in `storm`. It should show something like:
  `Storm 1.2.1 (dev)`

- Homebrew (only MacOS)
- Docker
- Build from source (only MacOS and Linux, requires some expertise)

See `http://www.stormchecker.org/documentation/installation/installation.html` for the latter three options.

Hints:

- Please contact us as soon as possible, if there is any trouble with installing Storm.
- There will be more exercises in the future, that require Storm as well.

The `PRISM language` is a common way to describe DTMCs in a succinct way. Consider the following example of a program in the `PRISM` language

```
1  dtmc
2
3  const int N;
4  module coin
5      n : [0..N] init 0;
6      c : [0..N] init 0;
7      odd: bool init false;
8
9      [] n<N & odd -> 0.5 : (n'=n+1) & (c'=c+1) & (odd'=!odd) + 0.5 : (n'=n+1) & (odd'=!odd);
10     [] n<N & !odd -> 1 : (n'=n+1) & (odd'=!odd);
11     [] n=N -> 1 : (n'=N);
12 endmodule
13
14 label "goal" = c>floor(N/4);
```

- Line 1 describes the type of the model (for now we only consider `dtmc` here)
- Line 3 declares a constant `N` which can be set to a specific integer value later on.
- Lines 5 to 7 declare variables `n`, `c`, and `odd` with their domains and initial values: `n` and `c` both have the domain $\{0, 1, \dots, N\}$ while `odd` has the domain $\{\texttt{true}, \texttt{false}\}$. The set of states S of the described DTMC is then the cross-product of all domains, i.e., $S = \{0, 1, \dots, N\} \times \{0, 1, \dots, N\} \times \{\texttt{true}, \texttt{false}\}$. In addition, these line specify the (only) initial state $s_{init} = (0, 0, \texttt{false}) \in S$. A state $s = (i, j, b) \in S$ expresses the fact that variable `n` has value i, variable `c` has value j, and variable `odd` has value b.
- Lines 9 to 11 describe the probability matrix of the DTMC using so-called *guarded commands*. For example, the command

  ```
  [] n<N & odd -> 0.5 :  (n'=n+1) & (c'=c+1) & (odd'=!odd) + 0.5 :  (n'=n+1) & (odd'=!odd);
  ```

means the following: The command is *enabled* at a state $s = (i, j, b) \in S$ iff the guard 'n <N & odd' is satisfied in s (that is, $i < N$ and $b =$ `true` holds). If the command is enabled, there are two outgoing transitions that both have probability 0.5. One transition leads to the state $(i+1, j+1, \neg b) \in S$ (i.e, the values for variables `n` and `c` are both incremented and the value for `odd` is negated). The other transition leads to the state $(i+1, j, \neg b)$.

- Line 14 specifies the atomic propositions $AP = \{\texttt{goal}\}$ and the labeling L with

$$L((i,j,b)) = \begin{cases} \{goal\} & \text{if } j > \lfloor (N/4) \rfloor \\ \emptyset & \text{otherwise.} \end{cases}$$

More information on the PRISM language can be found here: `http://www.prismmodelchecker.org/manual/ThePRISMLanguage/`

a) Draw the DTMC described by the PRISM program above, assuming that the constant N is set to 3 (i.e. every occurrence of N is replaced by 3). It suffices to depict the fragment of the DTMC that is reachable from the initial state $s = (0, 0, \texttt{false})$

Hints:

- The resulting DTMC has 6 states and 7 transitions.

b) Use Storm to compute the reachability probabilities $\Pr(\Diamond \texttt{goal})$ for

i) N=3

ii) N=30

iii) N=300

iv) N=3000

Write down the probabilities on your solution sheet. Storm can be executed on this model as follows:

- store the contents of the listing above into a file called `model.prism`
- invoke storm using the following command line:

```
storm --prism model.prism --prop 'P=?[F "goal"]' --constants N=3
```