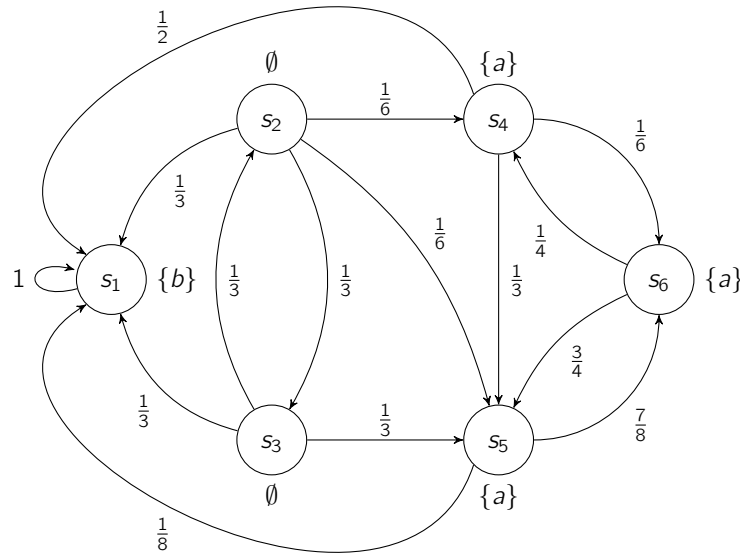


Exercise 1 (Bisimulation Minimization):**(7 points)**

- a) Develop an algorithm in pseudocode that computes the coarsest bisimulation for a given finite DTMC $D = (S, \mathbf{P}, s_{init}, AP, L)$ in polynomial time (in the number of states and transitions). Briefly explain the ideas behind your algorithm and why it guarantees the complexity bound.
- b) Apply your algorithm from a) to the following DTMC.



- c) Suppose you are given another DTMC $D' = (S, \mathbf{P}, s_{init}, \{a, b\}, L)$ and a property $\varphi = aUb$ ($a, b \in AP$) and are only interested in the probabilities $Pr(s \models \varphi)$ for all $s \in S$. Therefore, you want a quotient model that certainly preserves the probabilities for φ but not necessarily all PCTL path formulae. Indicate how your algorithm from a) can be improved to potentially (i.e. for some models) obtain a smaller quotient model while preserving the satisfaction probabilities for φ .
- d) Suppose you are interested in preserving only properties in $PCTL \setminus \{\bigcirc, U^{\leq k}\}$, i.e. all PCTL properties that do not involve the $\bigcirc$ and $U^{\leq k}$ operators. Modify the definition of bisimulation from the lecture in a way that preserves this fragment of PCTL but potentially (i.e. for some models) allows for coarser quotients.
Hint: Try to find a notion of bisimulation that in particular treats the states $\{s_4, s_5, s_6\}$ of D as bisimilar.

Solution: _____

- a) The algorithm could look as follows.

```

 $\Pi \leftarrow \{S\}$ 
for  $a \in AP$ 

```

```

 $\Pi' \leftarrow \emptyset$ 
for  $B \in \Pi$ 
     $\Pi' \leftarrow \Pi' \cup \{\{s \in B \mid a \in L(s)\}, \{s \in B \mid a \notin L(s)\}\}$ 
 $\Pi \leftarrow \Pi'$ 

do {
     $\Pi \leftarrow \Pi'$ 
     $\Pi' \leftarrow \{\{s \mid \text{sign}(s) = \text{sign}(t)\} \mid t \in S\}$ 
} while ( $\Pi' \neq \Pi$ )
return  $\Pi$ 

```

where

$$\text{sign}(s) = \{(p, B) \mid B \in \Pi \wedge p = \sum_{s' \in B} P(s, s')\}$$

The polynomial complexity bound is guaranteed, since

- the initial partition can be computed in polynomial time,
- the signatures of all states can be computed in polynomial time,
- the grouping of states into classes with the same signature can be done in polynomial time,
- in each iteration at least one state is split off from another block, so there can only be polynomially many iterations.

b) First, the algorithm splits the initial partition $\Pi = \{S\}$ according to the labeling with atomic proposition. This yields

$$\Pi = \underbrace{\{\{s_1\}\}}_{B_1}, \underbrace{\{\{s_2, s_3\}\}}_{B_2}, \underbrace{\{\{s_4, s_5, s_6\}\}}_{B_3}$$

Next, the algorithm tries to split B_2 , but $\text{sign}(s_2) = \text{sign}(s_3)$, so the block remains unsplit. However, block B_3 is unstable and is split into three separate blocks, which yields

$$\Pi = \underbrace{\{\{s_1\}\}}_{B_1}, \underbrace{\{\{s_2, s_3\}\}}_{B_2}, \underbrace{\{\{s_4\}\}}_{B_3}, \underbrace{\{\{s_5\}\}}_{B_4}, \underbrace{\{\{s_6\}\}}_{B_5}$$

Now, block B_2 became unstable and needs to be split. This yields the final partitioning:

$$\Pi = \underbrace{\{\{s_1\}\}}_{B_1}, \underbrace{\{\{s_2\}\}}_{B_2}, \underbrace{\{\{s_3\}\}}_{B_3}, \underbrace{\{\{s_4\}\}}_{B_4}, \underbrace{\{\{s_5\}\}}_{B_5}, \underbrace{\{\{s_6\}\}}_{B_6}$$

c) The algorithm can be improved as follows:

- only the atomic propositions a, b need to be taken into account for the initial partition,
- the initial partition can be chosen as $\Pi = \{S_{=0}, S_{=1}, S_{\neq}\}$ (wrt. φ).
- the blocks $S_{=0}$ and $S_{=1}$ of the initial partition can be made absorbing.

Applying these optimizations yields a partitioning that is potentially coarser than before, but preserves the property φ . For instance, applied to the DTMC of **b)**, we get the following partitioning

$$\Pi = \underbrace{\{\{s_1, s_4, s_5, s_6\}\}}_{S_{=1}}, \underbrace{\{\{s_2, s_3\}\}}_{S_{=0}}$$

d) The notion of weak probabilistic bisimulation is what we are looking for. Intuitively, this is the probabilistic counterpart of stutter-bisimulation for transition systems. The definition goes as follows.

Definition 1. For a DTMC $D = (S, \mathbf{P}, s_{\text{init}}, \{a, b\}, L)$, $R \subseteq S \times S$ is a weak probabilistic bisimulation if for any $(s, t) \in R$

a) $L(s) = L(t)$

b) $P(s, [s]_R) < 1$ and $P(t, [t]_R) < 1$, then

$$\frac{P(s, C)}{1 - P(s, [s]_R)} = \frac{P(t, C)}{1 - P(t, [t]_R)} \text{ for all } C \in S/R, C \neq [s]_R = [t]_R$$

c) s can reach a state not in $[s]_R$ if and only if t can reach a state not in $[t]_R$.

This bisimulation preserves step-agnostic properties and therefore the fragment $\text{PCTL}_{\{\emptyset, u^{\leq k}\}}$.

Exercise 2 (Markov Decision Processes):

(3 points)

Reconsider the Truel between Mr. Black, Mr. Gray and Mr. White.

- The truel proceeds in rounds, Mr. Black shoots first, then Mr. Gray, then Mr. White.
- If it's Mr. Black's turn, he can decide to shoot in the air or at one of the enemies that are still alive.
- If aiming at an enemy, Mr. Black will hit with a probability of $\frac{1}{3}$.
- Mr. Gray will aim at Mr. White if he's still alive and otherwise at Mr. Black.
- Mr. White will aim at Mr. Gray if he's still alive and otherwise at Mr. Black.
- Mr. Gray has a probability of $\frac{2}{3}$ of hitting an enemy and Mr. White will kill the enemy with probability 1.

- Model the Truel as an MDP in the PRISM language.
- Use PRISM to determine the minimal and maximal probability for the event "Mr. Black wins".
- Use PRISM to obtain the schedulers (also called adversaries) that lead to the "worst-case" and "best-case" probability, respectively. Does your result match the discussion in the last tutorial?
Hint: Look at the command-line options of PRISM. In particular, options to generate adversaries and options to export the model.
- Why is it not a smart idea to also model the (nondeterministic) choices of Mr. Gray and Mr. White (in the sense that they themselves can pick the enemy they are aiming at or to shoot into the air) when considering extremal (min, max) probabilities?

Solution: _____

- The truel can be modeled like this.

```
mdp

formula someoneAlive = b_alive | g_alive | w_alive;

module truel
  b_alive : bool init true;
  g_alive : bool init true;
  w_alive : bool init true;

  turn : [0 .. 2] init 0;
```

```

[shoot_gray] turn=0 & b_alive & g_alive -> 1/3 : (turn'=turn+1)&(g_alive'=false) + 2/3 : (turn'=turn+1)&(g_alive'=false);
[shoot_white] turn=0 & b_alive & w_alive -> 1/3 : (turn'=turn+1)&(w_alive'=false) + 2/3 : (turn'=turn+1)&(w_alive'=false);
[shoot_air] turn=0 & b_alive -> 1 : (turn'=turn+1);

[] turn=0 & !b_alive & someoneAlive -> 1 : (turn'=turn+1);
[] turn=0 & !someoneAlive -> 1 : true;

[] turn=1 & g_alive & w_alive -> 2/3 : (turn'=turn+1)&(w_alive'=false) + 1/3 : (turn'=turn+1);
[] turn=1 & g_alive & !w_alive & b_alive -> 2/3 : (turn'=turn+1)&(b_alive'=false) + 1/3 : (turn'=turn+1);
[] turn=1 & !g_alive -> 1 : (turn'=turn+1);

[] turn=2 & w_alive & g_alive -> 1 : (turn'=0)&(g_alive'=false);
[] turn=2 & w_alive & !g_alive & b_alive -> 1 : (turn'=0)&(b_alive'=false);
[] turn=2 & !w_alive -> 1 : (turn'=0);
endmodule

label "bWins" = turn=0 & b_alive & !g_alive & !w_alive;

```

b) Calling PRISM like this

```
./bin/prism truel.nm -pctl "Pmin=? [F \"bWins\"]"
```

or

```
./bin/prism truel.nm -pctl "Pmax=? [F \"bWins\"]"
```

yields a probability of 0 and ≈ 0.396825 , respectively.

c) Executing

```
./bin/prism truel.nm -pctl "Pmin=? [F \"bWins\"]"
-exporttransdotstates truel.dot -exportadv adv.out
```

or

```
./bin/prism truel.nm -pctl "Pmax=? [F \"bWins\"]"
-exporttransdotstates truel.dot -exportadv adv.out
```

produces a graph (in the dot format) that shows the structure of the system (in particular including state ids) and an adversary file that indicates which of the choices needs to be taken in each state. Obviously, the probability for Mr. Black to win the truel is 0 if he continuously shoots in the air. He maximizes his survival odds when first shooting into the air and then aiming at the only other remaining contestant.

d) Modelling the choices of Mr. Gray and Mr. White as nondeterministic choices produces extremal probabilities of limited use. When maximizing the survival probability of Mr. Black, Mr. Gray and Mr. White would pick the option that is best for Mr. Black, for example shooting into the air, which, clearly, does not model the behaviour of the contestants in the Truel.