Examination
“Quantitative Evaluation of Embedded Systems”

3TU, the Netherlands, January 20th 2014, 9:00 - 12:00 am
Dr.ir. Pieter Cuijpers, Dr. Anne Remke, Dr. Marielle Stoelinga

This exam consists of 4 assignments.

You have 3 hours to work on this exam.

Switch off your mobile phone; no phones on your desk. Write readable with a black or blue pen, not with a pencil. You may use a standard calculator. Books are not allowed. You are allowed to use a two-sided A4 paper with hand-written notes.

Success!

You have to return this examination form!

Your name:
Your student number:
Your university:

<table>
<thead>
<tr>
<th>assignment</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>maximum</td>
<td>15</td>
<td>15</td>
<td>10</td>
<td>10</td>
<td>50</td>
</tr>
<tr>
<td>score</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1. Timed Automata (15 points)

(a) A fire alarm system in a tunnel works as follows. Two sensors, $A$ and $B$, measure the number of smoke particles in the tunnel. If for a period of 2 seconds or more, both sensors indicate that the number of particles exceeds a threshold $T$, then an alarm goes off.

(i) Model the system as a timed automaton. Use the following action labels. (2 points)

- $\text{high}_A$ denotes the moment that the number of particles measured by sensor $A$ is growing and first (since the number of particles started growing) exceeds the threshold $T$.
- $\text{low}_A$ denotes the moment that the number of particles measured by sensor $A$ is decreasing and first (since the number of particles started decreasing) falls beneath the threshold $T$.
- Similarly, we use $\text{high}_B$ and $\text{low}_B$.

(ii) Same question, but now there are 100 sensors, and 30 need to be above the threshold for the alarm to go off. Use variables and parameterized actions in your timed automaton model. (2 points)

(b) Answer the following questions about the paper Verification of Printer Datapaths using Timed Automata by Georgeta Igna and Frits Vaandrager.

(i) Why is it relevant to analyze real-time properties of data paths in printers? (1 point)

(ii) What is, in your opinion, the most important conclusion of the paper? (1 point)

(c) Consider the TAs $P$ and $Q$ in Figure 1; note that $Q$ is the same as $P$ except that location $l_2$ has been deleted. Recall that TA stands for timed automaton; TLTS for timed labeled transition system; RA for region automaton.

(i) Is location $l_2$ reachable in $P$? If so, determine a path in $P$ that reaches $l_2$, otherwise explain why $l_2$ is not reachable. (1 point)

(ii) Same question for $l_3$. (1 point)

(iii) Determine the TLTS $T$ of $Q$. (2 points)

(iv) Determine the RA of $Q$. (2 points)

(d) Are the following statements true or false? Write down the answer; no explanation is required. For each correct answer, you get 1 point; for each incorrect one -1; and if no answer is given, you get 0 points.

(i) Assume that location $G$ is reachable in TA $R$. Now we replace each edge $l \xrightarrow{g \land inv(l)} a r l'$ in $R$ by $l' \xrightarrow{g \land inv(l') \land r} l'$. Here, $l$ and $l'$ are locations, $g$ is a clock guard, $a$ an action and $r$ a reset set. We claim that $G$ is still reachable after the transformation.

(ii) Same question, but now we replace each edge $l \xrightarrow{g \land inv(l') \land r} l'$ by $l' \xrightarrow{g \land inv(l') \land r} l'$. (2 points)

(iii) Assume that location $G$ is reachable in TA $R$. Then $l$ is also reachable in the parallel composition $R || R'$. (2 points)
\[ \begin{align*}
P: &\quad l_0 \\
&\quad x < 2 \quad a \quad x := 0 \\
&\quad l_2 \xrightarrow{x-y \leq 3} l_1 : x < 1 \quad 5 \leq x-y \rightarrow l_3 \\
\end{align*} \]

Figure 1: TA P with initial location \( l_0 \)

\[ \begin{align*}
Q: &\quad l_0 \\
&\quad x < 2 \quad a \quad x := 0 \\
&\quad l_1 : x < 1 \quad 5 \leq x-y \rightarrow l_3 \\
\end{align*} \]

Figure 2: TA Q with initial location \( l_0 \)
For the graph above, *in which numbers below tokens represent the number of initial tokens on an edge*, perform the following calculations.

(a) (3 points) Determine the (max,+) algebra matrix equations for this graph;

(b) (2 points) Determine the maximum achievable throughput and maximum cycle mean of this graph;

(c) (4 points) Determine a latency-optimal periodic schedule, for a period $\mu = 7$ (or determine a non-optimal schedule for 3 points);

(d) (1 point) Give a bound on the worst-case latency from $u$ to $y$ for a source with period $\mu = 7$;

(e) (1 point) Give a bound on the worst-case latency from $u$ to $y$ for a source with period $\mu = 7$ if we delay the start-time of the source by 30 ms;

(f) (2 points) Consider the arc from actor $A$ to actor $B$. We want to limit the buffer-size of this arc to at most $b$ tokens. Extend the graph to model this, and calculate the minimum buffer-size necessary to keep all throughput and latency estimates as they are.

(g) (2 points) Each actor in the graph represents a calculation in which the number of processors working on that calculation is unlimited. How would you model a situation in which each calculation is given a fixed number of dedicated processors, and how many processors are needed for each of the different calculations in order to achieve the same latency and throughput bounds as the unrestricted graph? (Provide a table, stating the number of processors for each actor.)
3. PCTL model checking (10 points)

Consider the following discrete-time Markov chain:

(a) Determine the set of states for which the PCTL-formula $P_{\geq p}(bUc)$ holds for $p = \frac{10}{13}$;

(b) Determine the set of states for which the PCTL-formula $P_{< p}(bU\leq^2 c)$ holds for $p = \frac{1}{2}$.
4. CSL model checking

Consider the CTMC of Figure 12 with $AP = \{a, b, c\}$. The middle-leftmost state is the initial state. Inside each state it is denoted which atomic properties are valid in that state.

Check whether the following properties are valid in the initial state:

a) $S_{>0.05}(b)$

b) $P_{>0.5}(X^{\leq 1} a)$

c) $P_{>0.2}(\neg a \cup^{\leq 2}(a \lor b \lor c))$

(N.B: this can be calculated exactly. Use $\sum_{k=0}^{\infty} \frac{x^k}{k!} = e^x$)
SOLUTIONS
1. Timed Automata - With solutions (15 points)

(a) A fire alarm system in a tunnel works as follows. Two sensors, A and B, measure the number of smoke particles in the tunnel. If for a period of 2 seconds or more, both sensors indicate that the number of particles exceeds a threshold $T$, then an alarm goes off.

(i) Model the system as a timed automaton. Use the following action labels. (2 points)

- **high**$_A$ denotes the moment that the number of particles measured by sensor A is growing and first (since the number of particles started growing) exceeds the threshold $T$.
- **low**$_A$ denotes the moment that the number of particles measured by sensor A is decreasing and first (since the number of particles started decreasing) falls beneath the threshold $T$.
- Similarly, we use **high**$_B$ and **low**$_B$.

Answer: See Figure 4 for the automata. Additional definitions are:

- **four channels** named **low**$_A$, **high**$_A$, **low**$_B$, **high**$_B$
- **a clock** $c$ for the Controller automaton

(ii) Same question, but now there are 100 sensors, and 30 need to be above the threshold for the alarm to go off. Use variables and parameterized actions in your timed automaton model. (2 points) Answer: See Figure 5. Additional definitions are:

- **two arrays** of channels (parameterized actions), each of **length 100**: **low**[100], **high**[100]
- **a clock** $c$ and a counter **counter** for the Controller automaton
- **id_t** defines the range of the sensor identifiers: it is equivalent to **int**[0, 99]

**counter** is increased each time a sensor reaches the “high” state, and decreased each time a “low” state is reached. Please note that the clock must be reset only when we first reach 30 “high” sensors: from then, we start counting two seconds and eventually switch to the ALARM location unless the number of “high” sensors has decreased in the meantime.
(b) Answer the following questions about the paper *Verification of Printer Datapaths using Timed Automata* by Georgeta Igna and Frits Vaandrager.

(i) Why is it relevant to analyze real-time properties of data paths in printers? (1 point)

*Answer: The major constraint in system performance of multicore computers is the bandwidth allocation to all tasks. Therefore, it is necessary to analyze real time properties of data paths in printers to verify, if the bandwidth allocation is sufficient to meet WCET requirements.*

(ii) What is, in your opinion, the most important conclusion of the paper? (1 point)

*Answer: This paper presented the first ever method to do a performance analysis for the systems having a dynamic memory bus and uncertain arrival times.*

(c) Consider the TAs $P$ and $Q$ in Figure 1; note that $Q$ is the same as $P$ except that location $l_2$ has been deleted. Recall that TA stands for timed automaton; TLTS for timed labeled transition system; RA for region automaton.

(i) Is location $l_2$ reachable in $P$? If so, determine a path in $P$ that reaches $l_2$, otherwise explain why $l_2$ is not reachable. (1 point)

*Answer: Location $l_2$ is reachable in $P$. An example path is $(l_0, \langle x = 0, y = 0 \rangle) \xrightarrow{a} (l_1, \langle x = 0, y = 0 \rangle) \xrightarrow{\tau} (l_2, \langle x = 0, y = 0 \rangle).$*

(ii) Same question for $l_3$. (1 point)

*Answer: Location $l_3$ cannot be reached in $P$. When location $l_1$ is entered, we have that $x \leq y$. That inequality is not changed by the self-transition in $l_1$, so the clock difference $x - y$ is always less than or equal to zero. This means that the guard $5 \leq x - y$ can never be true, and the transition to $l_3$ can never be taken.*
(iii) Determine the TLTS $T$ of $Q$. (2 points)

**Answer:**

$S = \{l_0, l_1, l_2, l_3\} \times \{ (z = j, y = k) \mid j, k \in \}$

$S_0 = (l_0, (z = 0, y = 0))$

$A = \{a\}$

$\rightarrow = \{ (l_0, u) \xrightarrow{a} (l_0, v) \mid u, v : C \rightarrow \land d \in \land v = u + d \}$

$\cup \{ (l_0, u) \xrightarrow{g} (l_1, v) \mid u, v : C \rightarrow \land u = x < 2 \land v = u[x := 0] \}$

$\cup \{ (l_1, u) \xrightarrow{d} (l_1, v) \mid u, v : C \rightarrow \land d \in \land u = x < 1 \land v = u + d \}$

$\cup \{ (l_1, u) \xrightarrow{r} (l_2, v) \mid u, v : C \rightarrow \land u = x < 1 \land v = u[x := 0] \}$

$\cup \{ (l_2, u) \xrightarrow{d} (l_2, v) \mid u, v : C \rightarrow \land d \in \land v = u + d \}$

(iv) Determine the RA of $Q$. (2 points)

**Answer:** See Figure 6.

![Figure 6: Graphical representation of the Region Automaton for TA $Q$. The initial state is marked with a thick border.](image)

(d) Are the following statements true or false? Write down the answer; no explanation is required.

For each correct answer, you get 1 point; for each incorrect one -1; and if no answer is given, you get 0 points.

(i) Assume that location $G$ is reachable in TA $R$. Now we replace each edge $l \xrightarrow{g a r} l'$ in $R$ by $l \xrightarrow{g \land Inv(l)} a r \xrightarrow{r} l'$. Here, $l$ and $l'$ are locations, $g$ is a clock guard, $a$ an action and $r$ a reset set. We claim that $G$ is still reachable after the transformation.

**Answer:** True

(ii) Same question, but now we replace each edge $l \xrightarrow{g a r} l'$ by $l \xrightarrow{g \land Inv(l')} a r \xrightarrow{r} l'$.

**Answer:** True

(iii) Assume that location $G$ is reachable in TA $R$. Then $l$ is also reachable in the parallel composition $R || R'$.

**Answer:** True
\[ P : \]
\[
\begin{align*}
  &l_0 \\
  &\quad \downarrow \quad x < 2 \quad a \quad x := 0 \\
  &\quad \downarrow \quad x := 0 \\
  &l_2 \quad x - y \leq 3 \quad l_1 : x < 1 \quad \frac{5 \leq x - y}{x := 0} \quad l_3
\end{align*}
\]

Figure 7: TA \( P \) with initial location \( l_0 \)

\[ Q : \]
\[
\begin{align*}
  &l_0 \\
  &\quad \downarrow \quad x < 2 \quad a \quad x := 0 \\
  &\quad \downarrow \quad x := 0 \\
  &l_1 : x < 1 \quad \frac{5 \leq x - y}{x := 0} \quad l_3
\end{align*}
\]

Figure 8: TA \( Q \) with initial location \( l_0 \)
2. Dataflow - With solutions (15 points)

For the graph above, in which numbers below tokens represent the number of initial tokens on an edge, perform the following calculations.

(a) (3 points) Determine the \((\text{max},+)\) algebra matrix equations for this graph;
   Answer:

   \[
   \mathbf{x}(n+1) = \begin{pmatrix}
   -\infty & 0 & -\infty & -\infty & -\infty & -\infty \\
   -\infty & -\infty & 0 & -\infty & -\infty & -\infty \\
   -\infty & -\infty & -\infty & -\infty & 0 & -\infty \\
   18 & -\infty & -\infty & 11 & -\infty & -\infty \\
   18 & -\infty & -\infty & 11 & -\infty & -\infty \\
   \end{pmatrix}
   \mathbf{x}(n) \max
   \begin{pmatrix}
   -\infty \\
   -\infty \\
   11 \\
   11 \\
   11 \\
   \end{pmatrix}
   u(n)
   \]

   \[
   y(n) = \begin{pmatrix}
   20 \\
   -\infty \\
   -\infty \\
   13 \\
   -\infty \\
   10 \\
   \end{pmatrix}
   \mathbf{x}(n) \max
   \begin{pmatrix}
   13 \\
   \end{pmatrix}
   u(n)
   \]

   Note that you may have permutations of the matrices if you numbered the tokens differently, but be aware that it is not transposed!

(b) (2 points) Determine the maximum achievable throughput and maximum cycle mean of this graph;
   Answer: There are three simple cycles in the graph: ABFEA, ACFEA, and CFC. The cycle mean of ABFEA is 18 ms over 3 tokens, so \(\frac{18}{3}\), the cycle mean of ACFEA is \(\frac{16}{5}\), and the cycle mean of CFC is \(\frac{11}{2}\). The maximum cycle mean is therefore \(\frac{18}{3} = 6\), with ABFEA as critical cycle. The maximum achievable throughput is \(\frac{1}{\text{MCM}} = \frac{1}{6}\) tokens per ms.

(c) (4 points) Determine a latency-optimal periodic schedule, for a period \(\mu = 7\) (or determine a non-optimal schedule for 3 points);
   Answer: the first algorithm explained in the lectures will give you the schedule depicted in figure 9, while the latency optimization will increase the value of the center actor as in figure 10.

(d) (1 point) Give a bound on the worst-case latency from \(u\) to \(y\) for a source with period \(\mu = 7\);
   Answer: The output actor starts at 11 in the periodic schedule, hence produces at 21. There are no tokens on the path \(u-C-D-y\) from input to output. So the worst-case latency of this graph is bounded by 21.

(e) (1 point) Give a bound on the worst-case latency from \(u\) to \(y\) for a source with period \(\mu = 7\) if we delay the start-time of the source by 30 ms; Answer: If the input is delayed by 7 miliseconds
(in the latency optimal solution) or by 5 milliseconds (if the suboptimal solution was chosen), the output actor can still start at 11 in the periodic schedule, hence produces at 21. The difference between production time and start time now becomes 21-7=14 or 21-5=16 ms, respectively. If the input is delayed any further (e.g. 30 ms), the remaining delay will simply shift the start-times of the entire periodic schedule and not influence the latency bound any further. Therefore, for a delayed start-time of 30 ms the best bound we can give on the latency is 14 ms.

(f) (2 points) Consider the arc from actor A to actor B. We want to limit the buffer-size of this arc to at most \( b \) tokens. Extend the graph to model this, and calculate the minimum buffer-size necessary to keep all throughput and latency estimates as they are.

Answer: the graph needs to be extended by an arc from B to A carrying \( b \) tokens. In order to avoid changing the throughput and latency estimates, \( b \) needs to be chosen in such a way that the MCM of the new graph equals that of the old graph. The only new cycle introduced by the new arc is ABA, which has a cycle mean of \( \frac{9}{b} \). If we chose \( b \geq 2 \) then this cycle mean is below the maximum cycle mean of 6, so the estimates do not change.

(g) (2 points) Each actor in the graph represents a calculation in which the number of processors working on that calculation is unlimited. How would you model a situation in which each calculation is given a fixed number of dedicated processors, and how many processors are needed for each of the different calculations in order to achieve the same latency and throughput bounds as the unrestricted graph? (Provide a table, stating the number of processors for each actor.)
Figure 11: Adapted graph modeling a buffer of size $b$ between A and B

Answer: In order to model that every actor executes on a dedicated number of processors, each actor should be given a self-cycle carrying that number of initial tokens. The next step is to determine, for each actor, how many initial tokens are needed to ensure that the MCM does not change. Since self-cycles do not introduce new cycles except for the self-cycle, we only need to check the execution time of each actor. This gives us the following table:

<table>
<thead>
<tr>
<th>actor</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>minimal number of processors</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>
3. PCTL model checking (10 points)

Consider the following discrete-time Markov chain:

(a) Determine the set of states for which the PCTL-formula $P_{\geq p}(bUc)$ holds for $p = \frac{10}{13}$;

Answer:

$I - \hat{P} = \begin{pmatrix}
1 & -\frac{1}{4} & 0 & -\frac{3}{4} & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
-\frac{1}{4} & 0 & -\frac{1}{4} & 1 & -\frac{1}{4} & 0 & -\frac{1}{4} & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & -\frac{1}{4} & 0 & -\frac{1}{4} & -\frac{1}{4} & 1 & -\frac{1}{4} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}$

$i_\psi^T = (0, 0, 1, 0, 1, 0, 0, 0)$

Compute $I - \hat{P} \cdot x = i_\psi$.

I: $x_0 - \frac{1}{4}x_1 - \frac{3}{4}x_3 = 0$
II: $x_1 = 0$
III: $x_2 = 1$
IV: $-\frac{1}{4}x_0 - \frac{1}{4}x_2 + x_3 - \frac{1}{4}x_4 - \frac{1}{4}x_6 = 0$
V: $x_4 = 1$
VI: $x_5 = 0$
VII: $-\frac{1}{4}x_2 - \frac{1}{4}x_4 - \frac{1}{4}x_5 + x_6 - \frac{1}{4}x_7 = 0$
VIII: $x_7 = 0$
and solve:

Insert III, V, VI in VII: \( x_6 = \frac{1}{2} \)
Insert I in IV: \( -\frac{3}{16} x_3 - \frac{1}{4} x_6 = \frac{1}{2} \)
Combine to: \( x_3 = \frac{10}{13} \)
Insert in I: \( x_0 = \frac{15}{26} \)

It follows that \( Sat = \{ s_2, s_3, s_4 \} \).

(b) Determine the set of states for which the PCTL-formula \( \mathcal{P}_{<p}(bUc) \) holds for \( p = \frac{1}{2} \).

Answer:

\[
P' = \begin{pmatrix}
0 & \frac{1}{3} & 0 & \frac{3}{4} & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{2} & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & \frac{1}{4} & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & \frac{1}{4} & 0 & \frac{1}{4} & \frac{1}{4} & 0 & \frac{1}{4} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\end{pmatrix}
\]

Then compute: \( (P' \cdot \overline{x}_G) \cdot \overline{x}_G \) with \( \overline{x}_G = (0, 0, 1, 0, 1, 0, 0, 0) \).

\( (P' \cdot \overline{x}_G) = (0, 0, 1, \frac{1}{2}, 1, 0, \frac{1}{2}, 0)^T \)

\( (P' \cdot \overline{x}_G) = (\frac{3}{8}, 0, 1, \frac{5}{8}, 1, 0, \frac{1}{2}, 0)^T \)

It follows that \( Sat = \{ s_0, s_1, s_5, s_7 \} \).
4. CSL model checking (10 points)

Consider the CTMC of Figure 12 with $AP = \{a, b, c\}$. The middle-leftmost state is the initial state. Inside each state it is denoted which atomic properties are valid in that state.

![Figure 12: A Continuous-Time Markov Chain](image)

Check whether the following properties are valid in the initial state:

a) $S_{>0.05}(b)$

Answer: The CTMC has only one bottom strongly connected component with label $b$, which is in the upper right corner. The probability to be in the state with label $b$, given that one is in that BSCC $B_1$ is $\frac{1}{2}$.

Then one has to compute the probability to reach that BSCC from the starting state. Hence we formulate the following equations:

\[
\begin{align*}
\text{Prob}\{s_1 \to B_1\} &= \frac{1}{3} \cdot \text{Prob}\{s_3 \to B_1\} \\
\text{Prob}\{s_3 \to B_1\} &= \frac{1}{2} \cdot \text{Prob}\{s_4 \to B_1\} + \frac{1}{2} \cdot \text{Prob}\{s_2 \to B_1\} \\
\text{Prob}\{s_4 \to B_1\} &= \frac{1}{2} \cdot \text{Prob}\{s_5 \to B_1\} \\
\text{Prob}\{s_2 \to B_1\} &= \frac{1}{2} \cdot \text{Prob}\{s_3 \to B_1\} + \frac{1}{2} \text{Prob}\{s_1 \to B_1\} \\
\text{Prob}\{s_5 \to B_1\} &= \frac{1}{2} + \frac{1}{2} \cdot \text{Prob}\{s_4 \to B_1\}
\end{align*}
\]

And solve this system of linear equations:

\[
\begin{align*}
\text{Prob}\{s_5 \to B_1\} &= \frac{2}{3} \\
\text{Prob}\{s_4 \to B_1\} &= \frac{3}{4} \\
\text{Prob}\{s_3 \to B_1\} &= \frac{5}{6} + \frac{1}{2} \cdot \text{Prob}\{s_2 \to B_1\} \\
\text{Prob}\{s_2 \to B_1\} &= \frac{1}{12} + \frac{1}{4} \cdot \text{Prob}\{s_2 \to B_1\} + \frac{1}{6} \text{Prob}\{s_3 \to B_1\}
\end{align*}
\]

Solving the last two equations, leads to:

\[
\begin{align*}
\text{Prob}\{s_2 \to B_1\} &= \frac{1}{6} \\
\text{Prob}\{s_3 \to B_1\} &= \frac{1}{4}
\end{align*}
\]

And we yield:

\[
\begin{align*}
\text{Prob}\{s_1 \to B_1\} &= \frac{1}{3} \cdot \frac{1}{4} = \frac{1}{12}
\end{align*}
\]

Hence, the probability to be in a $b$ state in steady-state is given by: $\frac{1}{12} \cdot \frac{1}{2}$, which is smaller than 0.5 and hence, the property does not hold for the starting state.
b) $P_{>0.5}(X^{\leq 1} a)$

\[ \text{Answer:} \]

Use the following equation to compute the probability that the next state is an $a$ state.

\[ \text{Prob}(s_1, X^{(0,1)} a) = (1 - e^{-E(s)}) \cdot \sum_{s'|=a} \frac{R(s,s')}{E(a)} = (1 - e^{-3}) \cdot \frac{2}{3} = 0.633 \]

The computed probability is larger than 0.5, hence the property holds for the starting state.

c) $P_{>0.2} \left( -a U^{\leq 2} \neg(a \lor b \lor c) \right)$

\[ \text{(N.B: this can be calculated exactly. Use } \sum_{k=0}^{\infty} \frac{x^k}{k!} = e^x) \]

\[ \text{Answer:} \]

First, make all states with label $a$ and those with $\neg(a \lor b \lor c)$ absorbing. Then, we only need to consider the part of the CTMC that can be reached from the starting state (6 states). Choosing the uniformization constant $q = 3$, the following embedded DTMC has to be considered:

\[
U = \begin{pmatrix}
0 & \frac{1}{3} & 0 & 0 & \frac{1}{3} & \frac{1}{3} \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{2}{3} & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}
\]

The starting vector is given by $\hat{\pi}(0) = (1, 0, 0, 0, 0, 0)$, and for time 2, we have to compute:

\[ \pi(2) = \sum_{i=0}^{\infty} e^{-6i} \cdot \hat{\pi}(0) \cdot U^i \]

First, we compute the transient probabilities in the embedded DTMC:

- $\hat{\pi}(1) = (0, \frac{1}{3}, 0, 0, \frac{1}{3}, \frac{1}{3})$
- $\hat{\pi}(2) = (0, 0, \frac{1}{3}, 0, \frac{1}{3}, \frac{1}{3})$
- $\hat{\pi}(3) = (0, 0, 0, \frac{2}{3}, 1, \frac{1}{3})$
- $\hat{\pi}(4) = (0, 0, 0, \frac{2}{3}, 1, \frac{1}{3})$
- $\hat{\pi}(5) = (0, 0, 0, \frac{2}{3}, 1, \frac{1}{3})$

After three steps, $\hat{\pi}(i)$ does not change anymore. Using $\sum_{k=0}^{\infty} \frac{x^k}{k!} = e^x$, we obtain for the goal state $s_4$:

\[ \pi_4(2) = e^{-6}(e^6 - \frac{(qt)^0}{0!} - \frac{(qt)^1}{1!} - \frac{(qt)^2}{2!}) \cdot \frac{2}{5} = 0.208, \]

which is larger than 0.2, hence the property holds for the starting state.