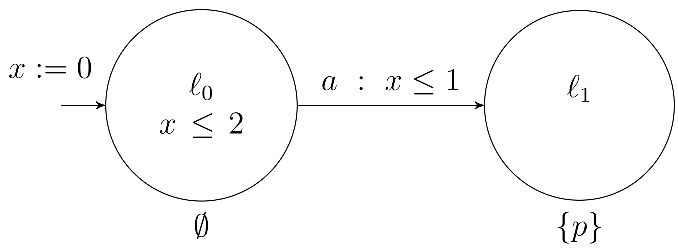
Modeling and Analysis of Hybrid Systems - SS 2015

Series 5

Exercise 1

Consider the TCTL formula $\Phi = A\mathcal{F}p$ and the following timed automaton $\mathcal{T}$:



(a) Does $\mathcal{T} \models \Phi$ hold, i.e., does $\mathcal{T}$ satisfy the TCTL formula Φ in its initial state?

(b) Please determine $RTS(\mathcal{T}, \Phi)$. It is sufficient to present the reachable fragment. Note that the TCTL formula Φ has no time bounds, therefore you do not need to introduce any auxiliary clock z.

(c) Does $\mathcal{T}$ have a path leading to a time-lock? If so, how can we recognize it on $RTS(\mathcal{T}, \Phi)$?

(d) Please apply the CTL model checking algorithm presented in the lecture to determine whether $RTS(\mathcal{T}, \Phi) \models \hat{\Phi}$, i.e., whether $RTS(\mathcal{T}, \Phi)$ satisfies $\hat{\Phi} = A\mathcal{F}p$ in its initial state. Does it hold that

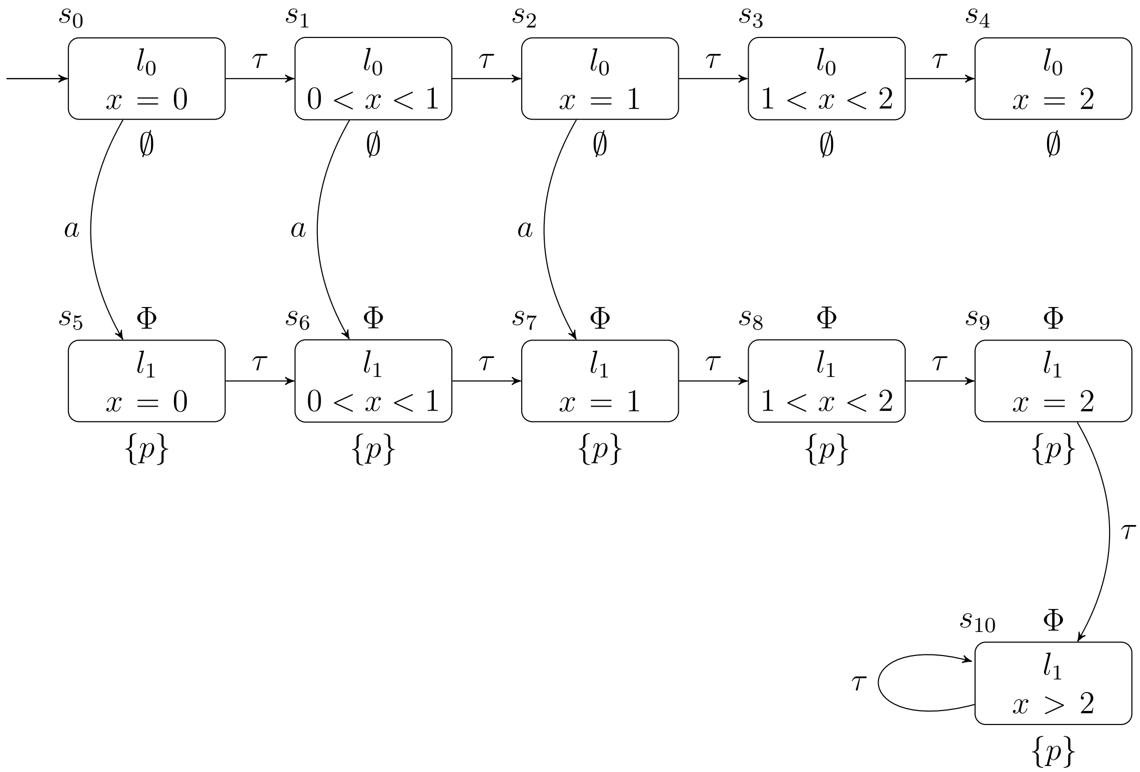
$$\mathcal{T} \models \Phi \qquad \textit{iff} \qquad RTS(\mathcal{T}, \Phi) \models \hat{\Phi} \quad ?$$

If not, why?

Solution:

(a) Yes, because all *time-divergent* paths of $\mathcal{T}$ eventually reach l_1, where p holds.

(b)

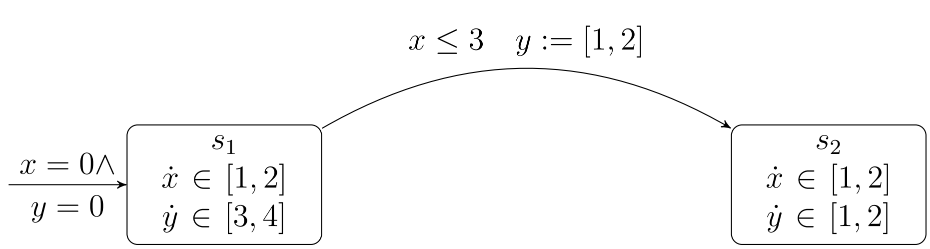


(c) Yes, $\mathcal{T}$ has time-lock paths. Clearly, $s_0 \to s_1 \to s_2 \to s_3 \to s_4$ is a finite path of $RTS(\mathcal{T}, true)$, it reflects the time-lock path $(l_0, \nu) \xrightarrow{2} (l_0, \nu')$ with $\nu(x) = 0$ and $\nu'(x) = 2$. For this Zeno-free model, we can see it on the deadlock state s_4 without any outgoing transition.

(d) They are given by the nodes labeled with Φ in $RTS(\mathcal{T}, true)$. The two model checking results do not coincide, because the timed automaton $\mathcal{T}$ is not timelock-free.

Exercise 2

Consider the following initialized rectangular automaton $\mathcal{A}$:



(a) Transform $\mathcal{A}$ into an initialized singular automaton $\mathcal{A}_1$ and specify a function f_1 mapping $\mathcal{A}$-states to $\mathcal{A}_1$-states, such that a state s is reachable in $\mathcal{A}$ iff $f_1(s)$ is reachable in $\mathcal{A}_1$.

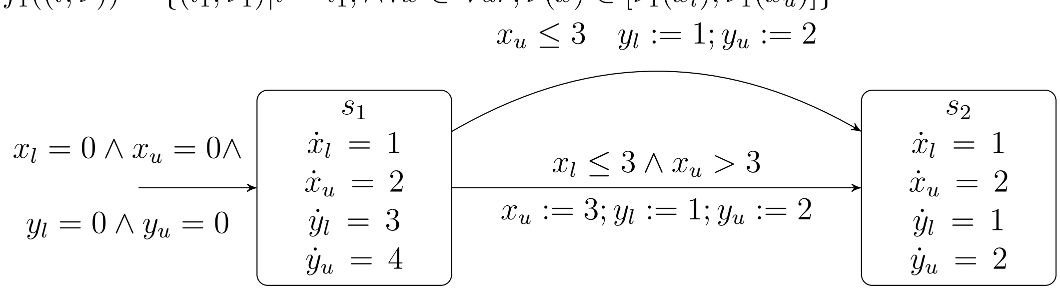
(b) Transform $\mathcal{A}_1$ into an initialized stopwatch automaton $\mathcal{A}_2$ and specify a function f_2 mapping $\mathcal{A}_1$-states to $\mathcal{A}_2$-states, such that a state s is reachable in $\mathcal{A}_1$ iff $f_2(s)$ is reachable in $\mathcal{A}_2$. You are allowed to set stopwatch values to any constants.

(c) Transform $\mathcal{A}_2$ into a timed automaton $\mathcal{A}_3$ and specify a function f_3 mapping $\mathcal{A}_2$-states to $\mathcal{A}_3$-states, such that a state s is reachable in $\mathcal{A}_2$ iff $f_3(s)$ is reachable in $\mathcal{A}_3$. You are allowed to set clock values to any constants.

(d) Transform the timed automaton $\mathcal{A}_3$ such that clocks are reset to the value 0, only.

Solution:

(a) $f_1((l,\nu)) = \{(l_1,\nu_1) | l = l_1, \wedge \forall x \in Var, \nu(x) \in [\nu_1(x_l), \nu_1(x_u)]\}$



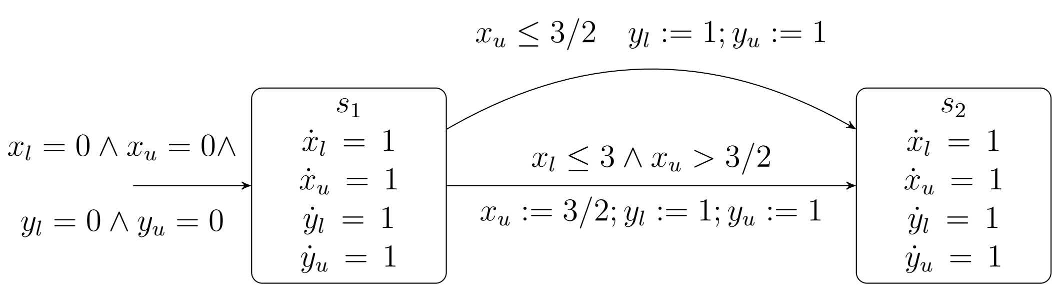
(b) $f_2((l_1,\nu_1)) = (l_2,\nu_2)$ such that $l_1 = l_2 \wedge$

(1) $\nu_2(x_l) = \nu_1(x_l)$

(2) $\nu_2(x_u) = \frac{1}{2}\nu_1(x_u)$

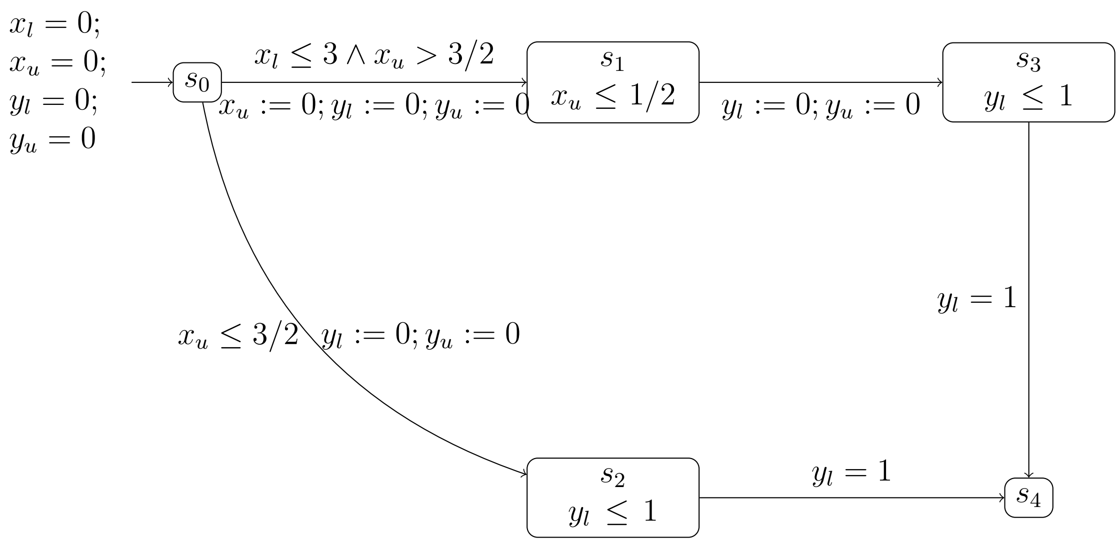
(3) $\nu_2(y_l) = \begin{cases} \frac{1}{3}\nu_1(y_l) & if\ l_2 = s_1 \\ \nu_1(y_l) & else \end{cases}$

(4) $\nu_2(y_u) = \begin{cases} \frac{1}{4}\nu_1(y_u) & if\ l_2 = s_1 \\ \frac{1}{2}\nu_1(y_l) & else \end{cases}$



(c) $f_3((l_2,\nu_2)) = (l_2,\nu_2)$
The automaton $\mathcal{A}_2$ is already a timed automaton.

(d)


$x_l = 0;$
$x_u = 0;$
$y_l = 0;$
$y_u = 0$
s_0
$x_l \leq 3 \wedge x_u > 3/2$
$x_u := 0; y_l := 0; y_u := 0$
s_1
$x_u \leq 1/2$
$y_l := 0; y_u := 0$
s_3
$y_l \leq 1$
$y_l = 1$
$x_u \leq 3/2$
$y_l := 0; y_u := 0$
s_2
$y_l \leq 1$
$y_l = 1$
s_4