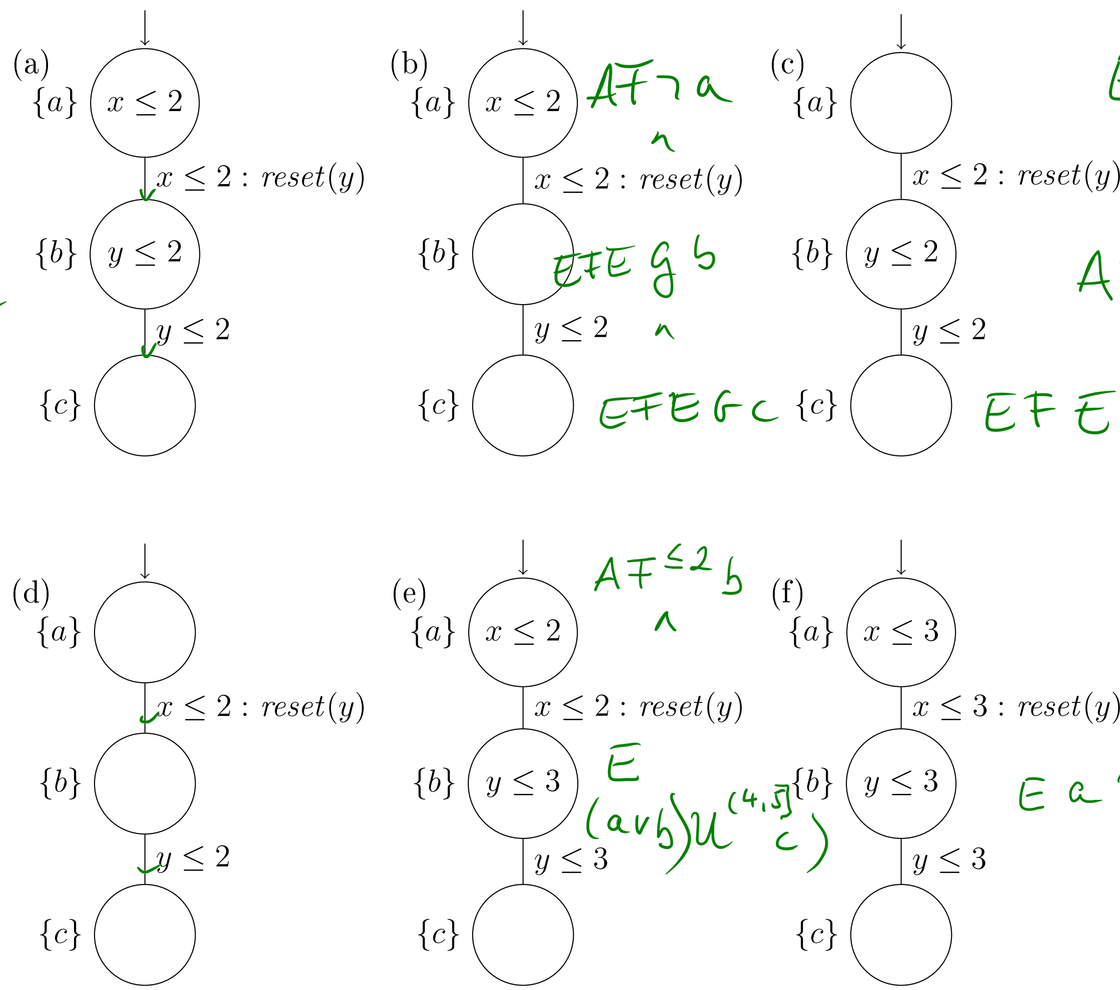
Modeling and Analysis of Hybrid Systems

Series 2/3

Exercise 1

Consider the following six timed automata:



(a) $AF^{\leq 4} c$

(b) $AF \neg a \wedge EFEG\, b \wedge EFEG\, c$

(c) $EG\, a \wedge AF \neg b \wedge EFEG\, c$

(d) $EG\, a \wedge EFEG\, b \wedge EFEG\, c$

(e) $AF^{\leq 2} b \wedge E((a \vee b)\, U^{(4,5]}\, c)$

(f) $E\, a\, U^{(2,3]}\, b$

Give for each automaton a TCTL formula that distinguishes it from all other ones. It is only allowed to use the atomic propositions a, b and c and clock constraints.

Solution:

(a) $A\mathcal{F}^{\leq 4}c$

(b) $A\mathcal{F}E\mathcal{G}b$

(c) $(E\mathcal{G}a) \wedge (\neg E\mathcal{F}E\mathcal{G}b)$

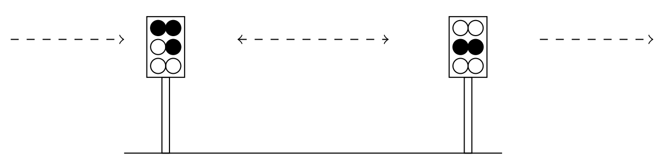
(d) $(E\mathcal{G}a) \wedge (E\mathcal{F}E\mathcal{G}b)$

(e) $(A\mathcal{F}^{\leq 5}c) \wedge (E\mathcal{G}^{(4,5)}\neg c)$

(f) $(A\mathcal{F}^{\leq 6}c) \wedge (E\mathcal{G}^{(5,6)}\neg c)$

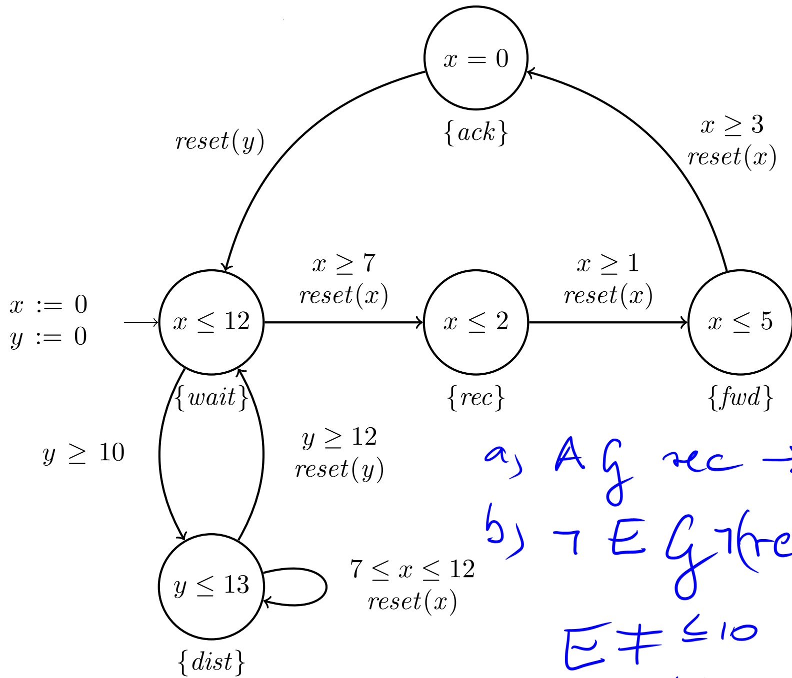
Exercise 2

The "clacks" are a visual telegraph tower system operated by the "Grant Trunk Company" of Ankh-Morpork (cf. Terry Pratchett: "Going postal"). It consists of a network of semaphore towers located about 20 miles from each other spread all over Discworld. Each tower has 6 semaphores which can show either a black panel or a white panel. Each tower is operated by a "clacks operator", whose task it is to watch his predecessing tower and in case there is a message it has to forward the message to the successor tower and after that send back an acknowledgement to the predecessor.



- For each tower, the time till the first incoming message and between two incoming messages from the predecessor is between 7 and 12 minutes.
- As it is very boring to sit and wait for a message, after 10 minutes of concentrated waiting the operator can get distracted, and then he or she is distracted for at least 2 and at most 3 minutes. When the operator is distracted, incoming messages will be lost. When the operator is not distracted, incoming messages will be successfully received.
- The operator needs between 1 and 2 minutes to forward a successfully received message.
- After forwarding, the operator needs another 3 to 5 minutes to send back an acknowledgement to the predecessor.

A timed automaton modelling one clacks-tower is given below, the set of atomic propositions is AP= $\{wait, rec, fwd, ack, dist\}$:



a) $A\mathcal{G}\ rec \rightarrow A\ \neg(wait)\,\mathcal{U}^{\leq 2}\ ack$

b) $\neg E\mathcal{G}\neg(rec) \equiv A\mathcal{F} rec$

$E\mathcal{F}^{\leq 10}\ dist$

c) $E\mathcal{F}^{\leq 10}(dist \wedge x = 0)$

Please give suitable TCTL-formulas, which formalize the following statements:

a) Each successfully received message is acknowledged within 2 minutes. (To assure that the acknowledgment is for the given received message, state that the waiting state is avoided between reception and acknowledgement.)

b) It cannot happen that all messages get lost.

c) It is possible that a message gets lost within the first 10 minutes.

Which of the above formulas holds for the modelled system? Please give reasons for your answer.

Solution:

a) $A\mathcal{G}(rec \rightarrow (A(\neg wait)\ \mathcal{U}^{\leq 2}\ ack))$

b) $A\mathcal{F} rec$

c) $E\mathcal{F}^{\leq 10}(dist \wedge x = 0)$

The first formula is not satisfied, as there is a path, where it takes 7 minutes from reception till acknowledgement.

The second formula does not hold, because it can happen periodically that the operator gets distracted after 10 minutes, a messages arrives (and gets lost) 1 minute later, and the operator goes back to the waiting state 1 further minute later.

Formula c) holds, because a message can get lost at time point 10, directly (without time delay) after the operator got distracted at time point 10.

Exercise 3

Please give a timed automaton for the following system. You can use as many clocks as you want, but you are restricted to use 4 locations, which are distinguished by the atomic propositions AP= $\{ferry_{left}, ferry_{right}, process\_cargo, travel\}$.

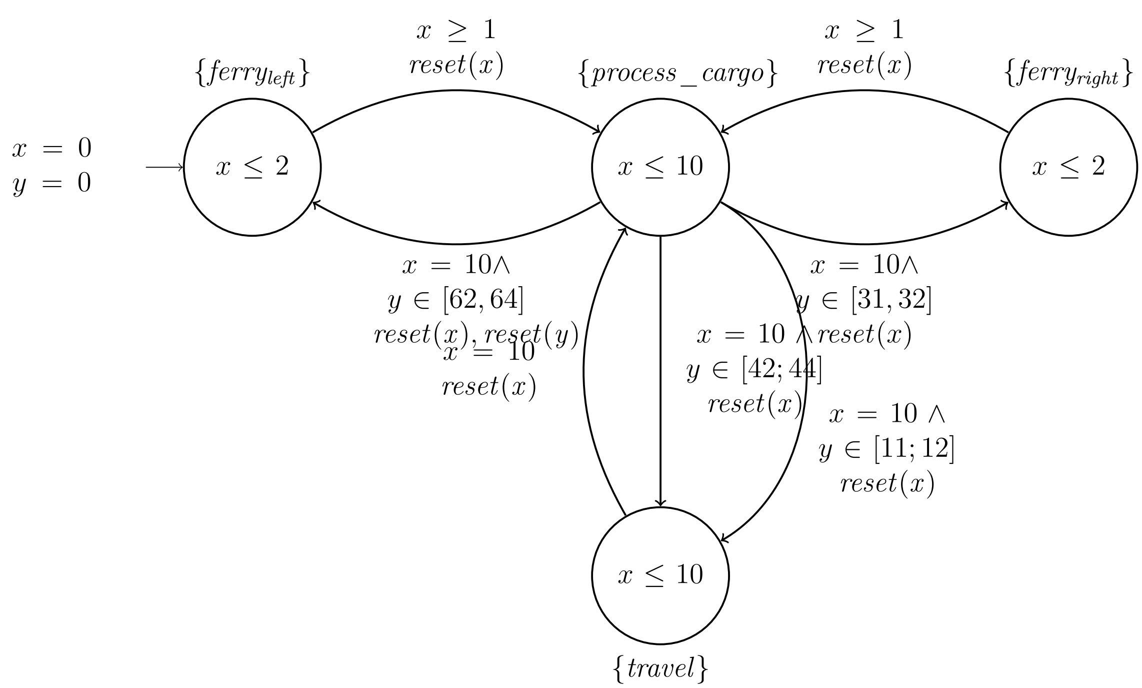
A river can be crossed by taking a ferry which has the following properties:

- Initially the ferry is on the left side of the river ($ferry_{left}$).
- Initially and after each unloading, the ferry waits 1-2 minutes for a new customer ($ferry_{left}/ferry_{right}$).
- Once a customer arrives, the ferry is loaded ($process\_cargo$), it crosses the river ($travel$), and it is unloaded ($process\_cargo$).
- Loading, crossing and unloading take exactly 10 minutes each.

Hint: You can encode certain properties by a clever usage of different clocks, resets and guards.

Solution:

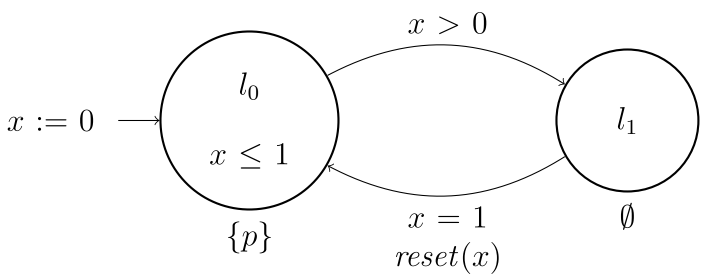
We require 2 clocks in total, one monitoring the time passed inside the locations (x) and one (y), which allows us to encode which way the ferry crosses the river.



Exercise 4

Consider the following timed automaton $\mathcal{T}$:

$$\begin{array}{ccc} \mathcal{T} & \models_{TCTL} & \varphi \\ \downarrow & \updownarrow & \downarrow \\ RTS & \models_{CTL} & \hat{\varphi} \end{array}$$

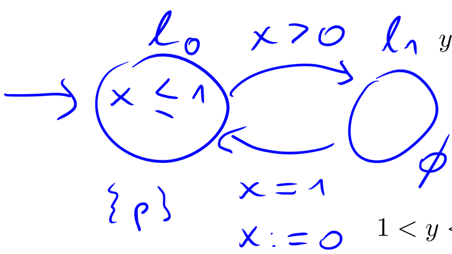


Please perform the TCTL model checking algorithm as presented in the lecture on $\mathcal{T}$ and verify $\mathcal{T} \models \varphi$, where $\varphi = AF^{\leq 2}p$. $\rightsquigarrow AF(p \wedge z \leq 2) \equiv A\,true\ U\ (p \wedge z \leq 2)$

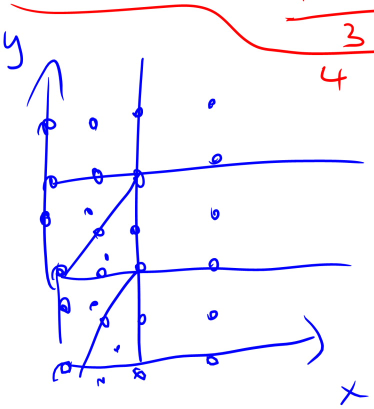
a) Construct $\hat{\varphi}$ by eliminating timing parameters from φ. Use the name y for the auxiliary clock.

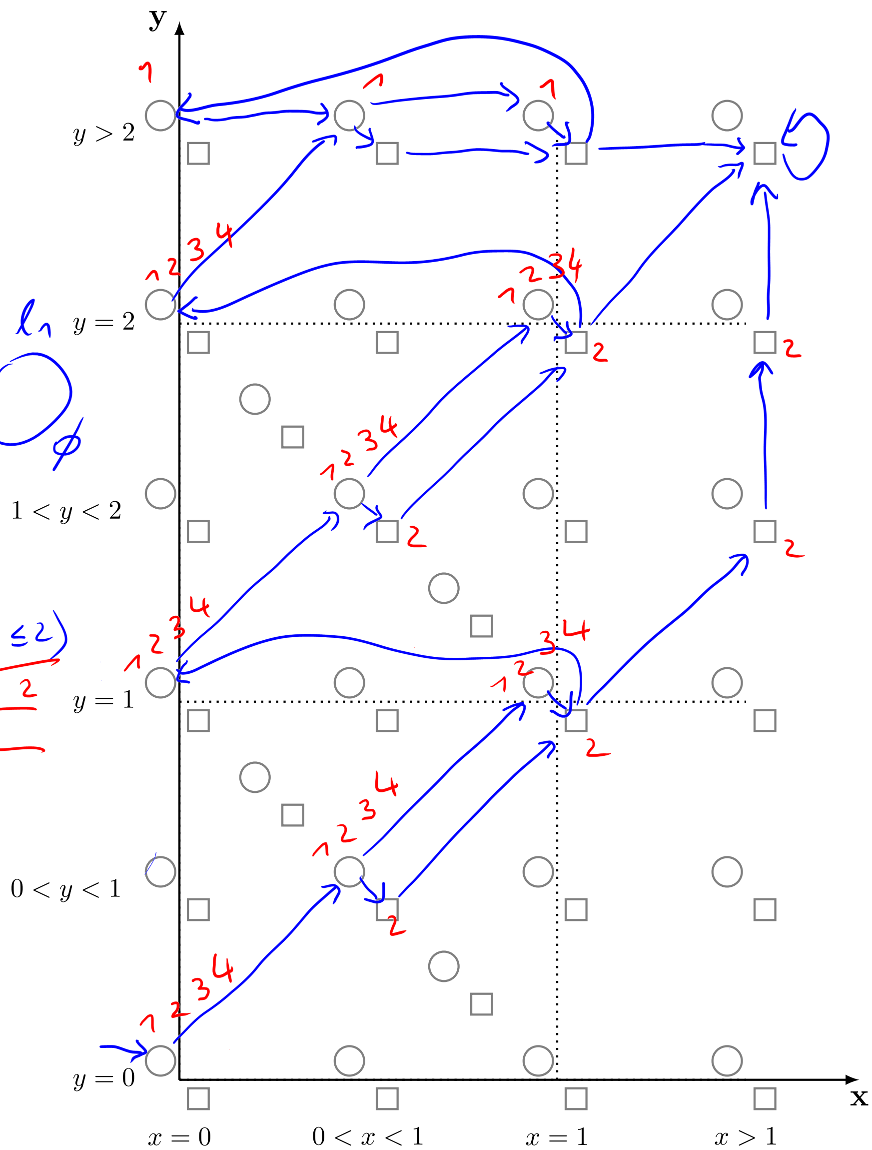
b) Construct a *RTS* $\mathcal{R}$, such that $\mathcal{T} \models_{TCTL} \varphi$ iff $\mathcal{R} \models_{CTL} \hat{\varphi}$. As $\mathcal{R}$ will become big, use the prepared grid below to sketch the *RTS* (by adding the required transitions) as follows:

- ◯ represents a state, where the location is l_0.
- ☐ represents a state, where the location is l_1.
- The position of a state in the grid remarks, which clock region the state represents.
- Please draw only the reachable fragment of $\mathcal{R}$.



$A\ \mathrm{true}\ \mathcal{U}\ (p \wedge y \le 2)$

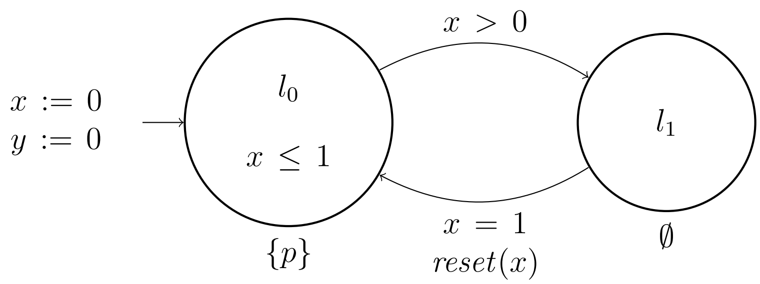




c) Apply CTL model checking to verify $\mathcal{R} \models_{CTL} \hat{\varphi}$. You can color states in your previously created *RTS* to indicate that a certain subformula holds in the respective state.

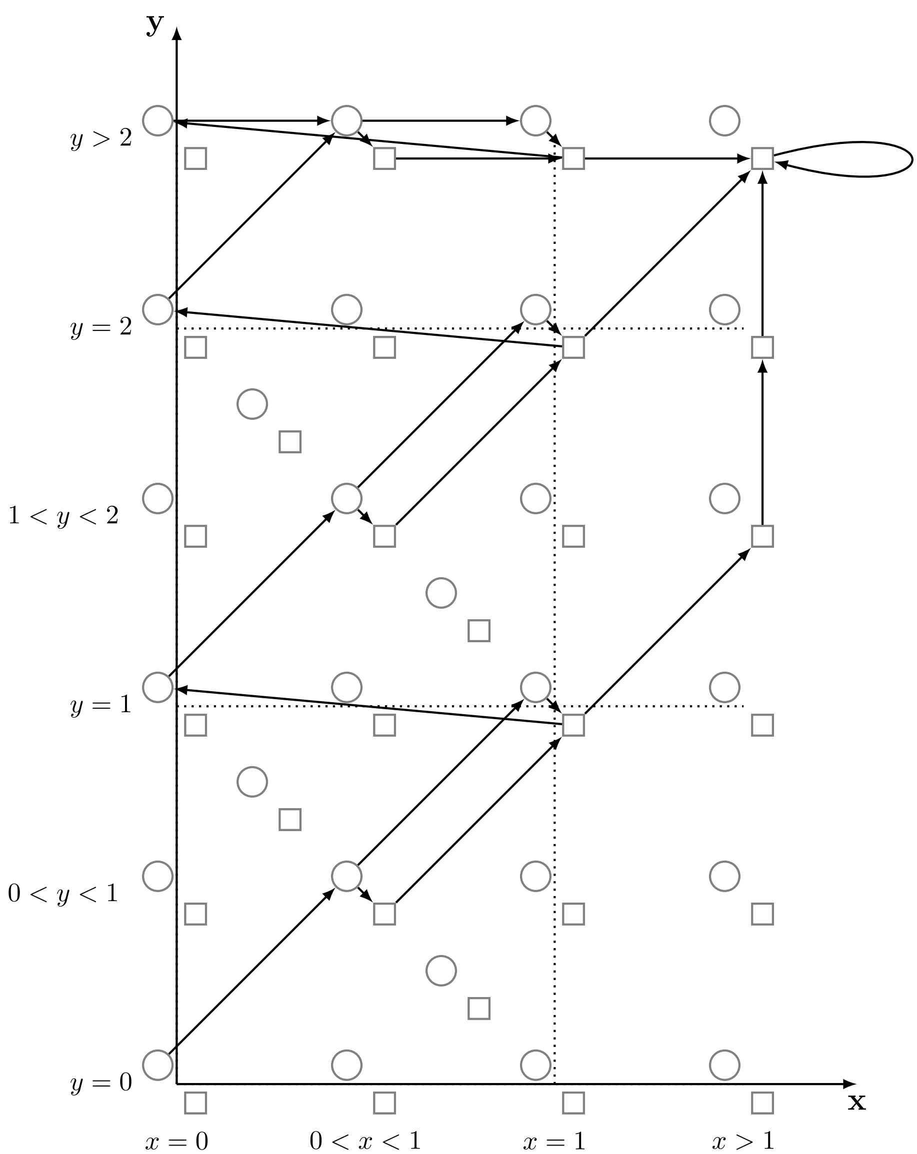
Solution:

a) We add an additional clock y to $\mathcal{T}$, such that $\mathcal{T}'$:

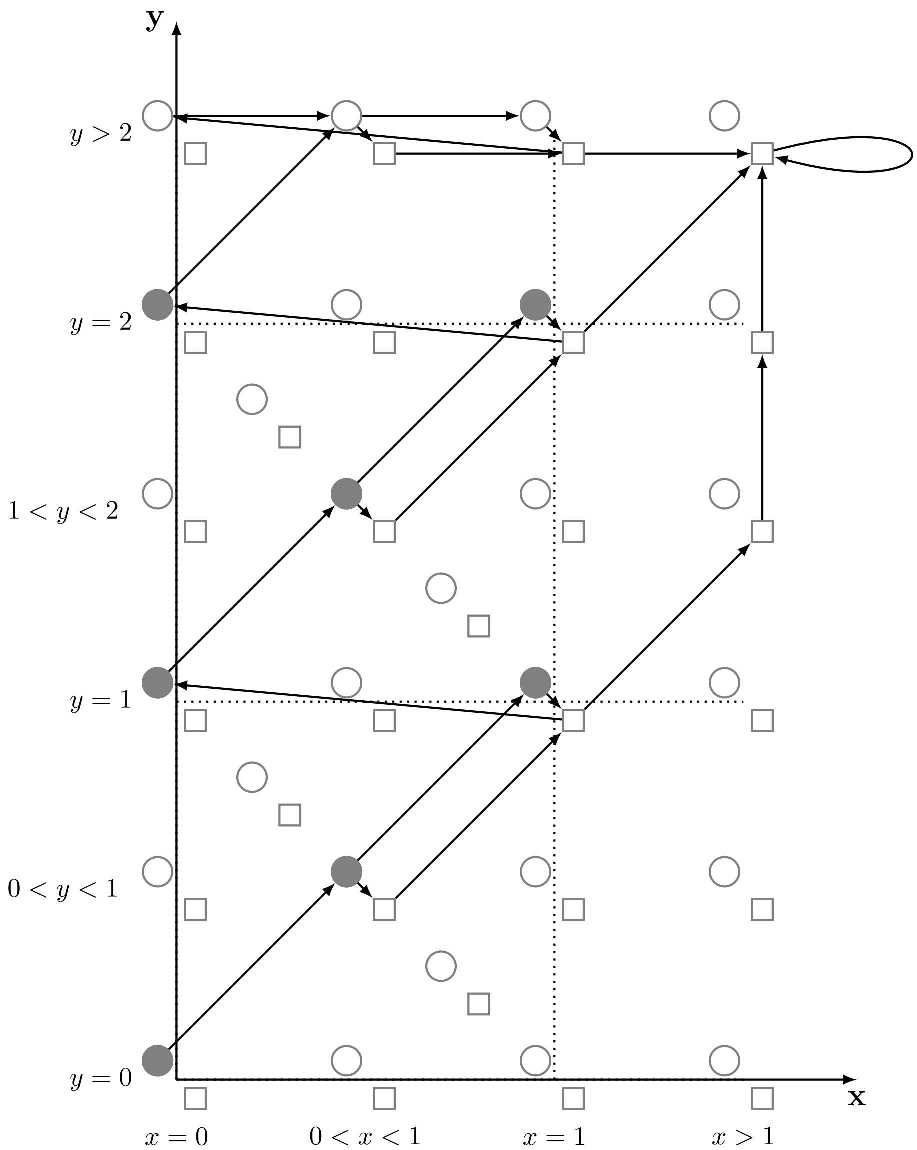


Removing syntactic sugar from φ yields $\varphi = A(true\ U^{\le 2}\ p)$ and finally removing time parameters yields $\hat{\varphi} = A(true\ U\ ((y \le 2) \wedge p)))$.

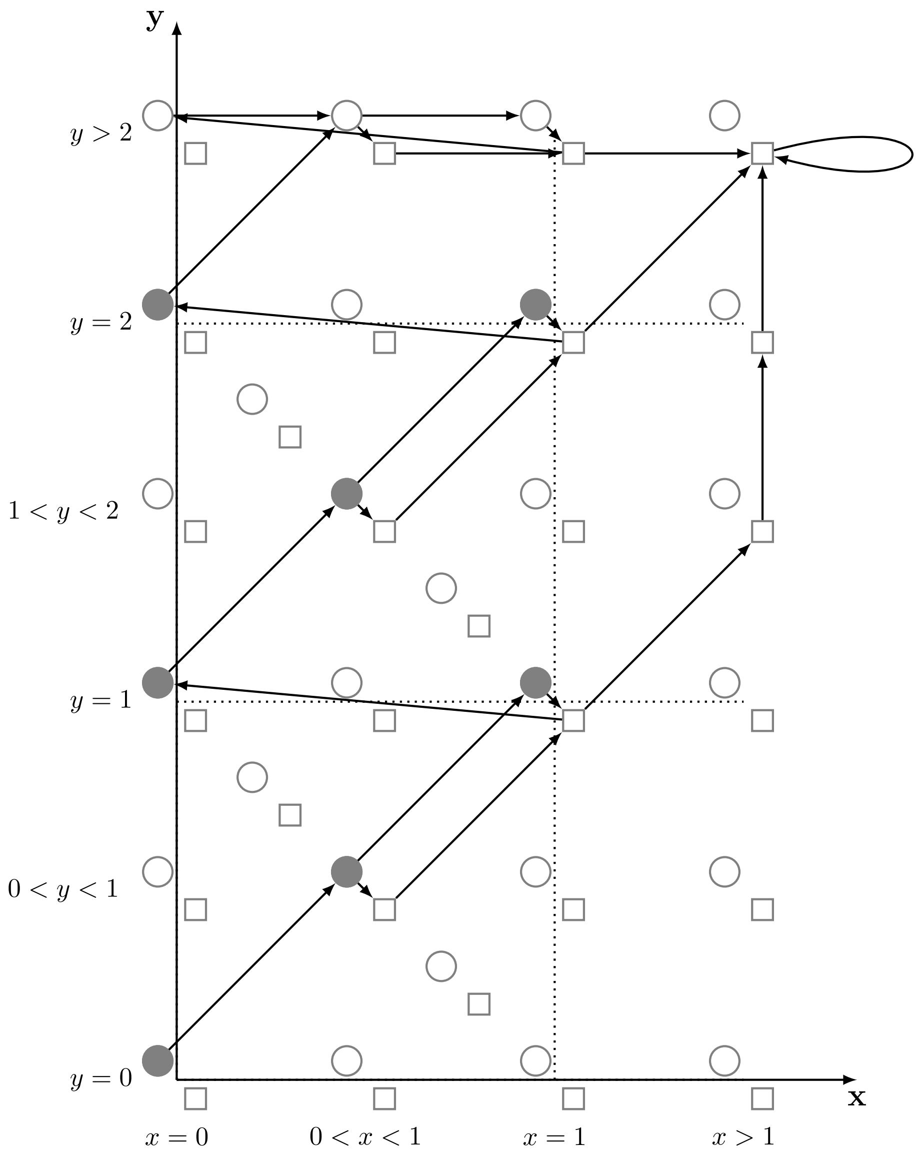
b) The *RTS* $\mathcal{R}$ is specified as follows:



c) Model checking $\mathcal{R} \models_{CTL} \hat{\varphi}$
Step 1: $\psi_1 = (y \le 2) \wedge p$



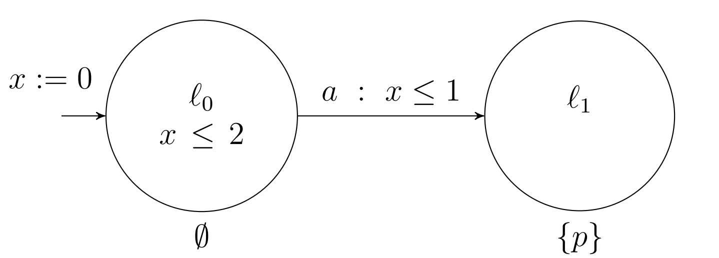
Model checking $\mathcal{R} \models_{CTL} \hat{\varphi}$
Step 2: $\psi_2 = A(\mathit{true}\ U\ \psi_1)$



As for all initial states $\sigma = (l, \nu) \in \mathcal{R}$ with $\nu(y) = 0$ it holds that $\sigma \models \hat{\varphi}$, we conclude $\mathcal{R} \models_{CTL} \hat{\varphi}$, and thus $\mathcal{T} \models_{TCTL} \varphi$.

Exercise 5

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

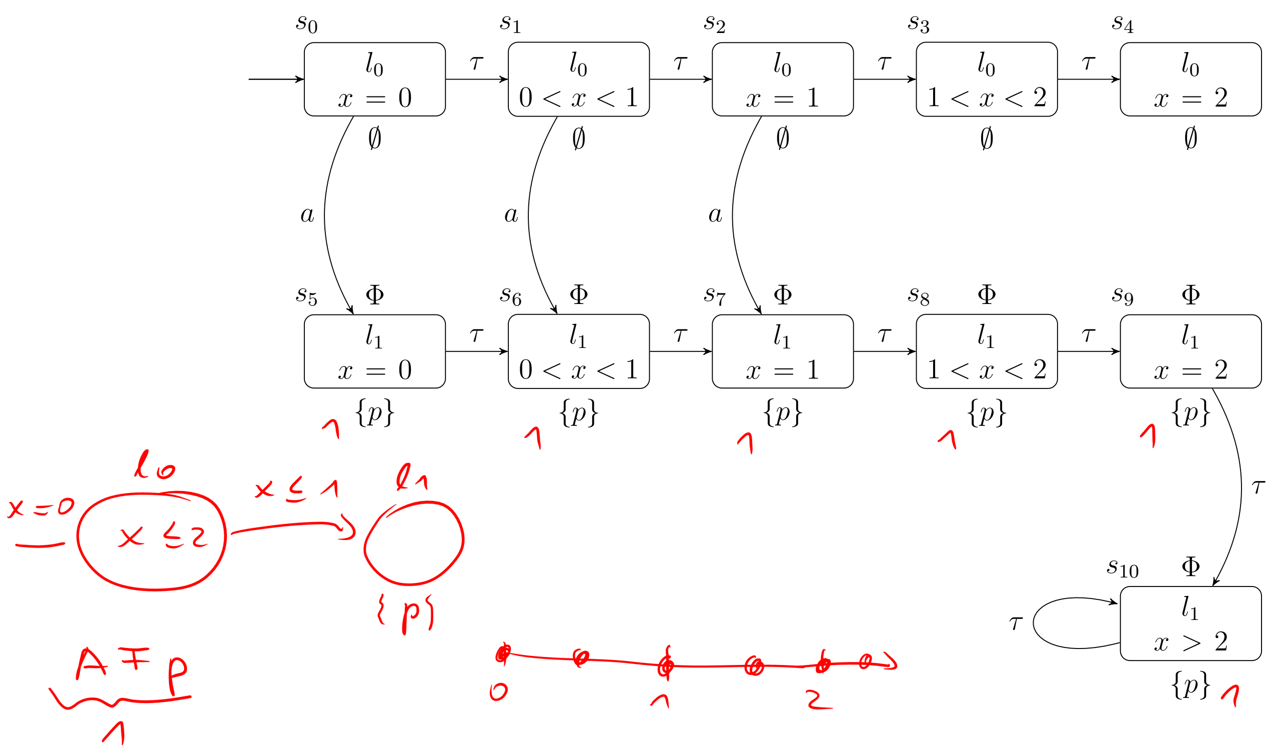
$$\mathcal{T} \models \Phi \qquad \textit{iff} \qquad RTS(\mathcal{T}, \Phi) \models \hat{\Phi} \quad ?$$

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 $RTS(\mathcal{T}, \mathit{true})$, 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.

1. TA $\quad \dot{x} = 1 \quad x \sim c$
2. RA $\quad \dot{x} \in [a,b] \quad x \in [a,b]$ $\quad (\ ) \quad \in \mathbb{Z}$
3. LHA I $\quad \dot{x} = c \quad$ linear formulas "$x \geq 2y + z$"
4. LHA II $\quad \dot{x} = Ax + Bu \quad$ linear formulas

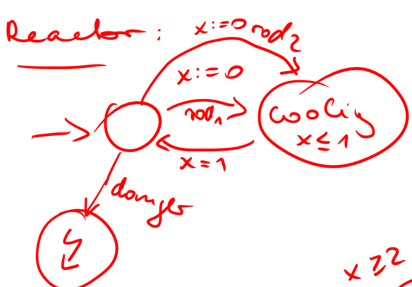
Not

5. HA $\quad \dot{x} = f(x)$

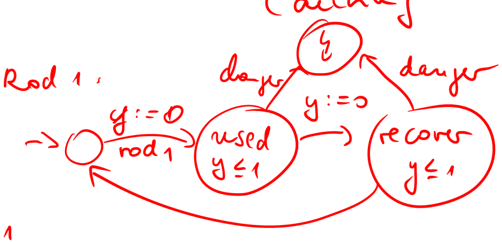
1. Abstraction: Decidability proof via decision procedure
2. Decidability proof by transformation
3. Logical encoding
4. Geometrical encoding

- Reactor can get hot, 2 rods for cooling available
- Cooling time: 1 time unit, recover time of rod after usage: 1 time unit (during this time cannot be used)

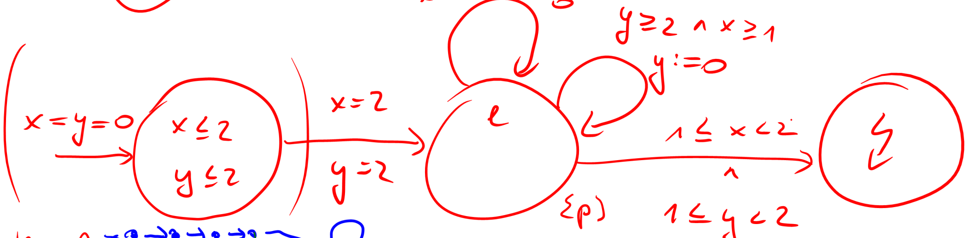
Reactor:



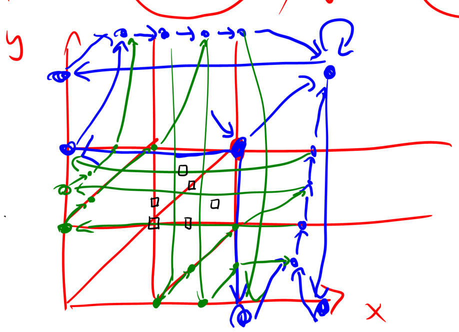
Rod 1:

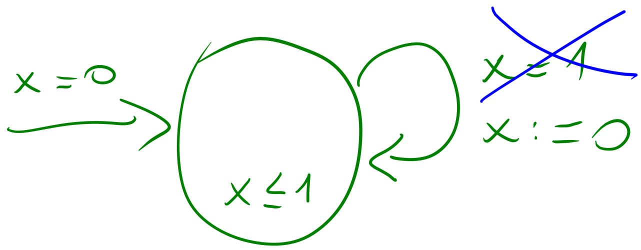


Rod 2: ...



$EF(p \wedge 1 \leq x < 2 \wedge 1 \leq y < 2)$





$AG\,AF\; x = 1$

$AF\; x = 1$