Modeling and Analysis of Hybrid Systems - SS 2016

First Exam

Tuesday, August 9, 2016

Forename and surname:	**Matriculation number:**
Sign here:	

- Do not open the exam until we give the start signal.
- Please place your student identity card on your desk for identification purposes.
- The duration of the exam is 120 minutes.
- Use a blue or black (permanent) pen only.
- Please write your name and matriculation number on each page of this exam.
- Please write clear and legible answers.
- If you need more sheets, indicate this by a hand signal. Please use a separate sheet for each task.
- Please clearly cross out parts you do *not* wish to be evaluated.
- If you have problems understanding a task, indicate this by a hand signal.
- You are not allowed to use auxiliary material except for a pen. In particular, switch off your electronic devices! Cheating disqualifies from the exam.

Task:	1.)	2.)	3.)	4.)	5.)	6.)	7.)	**Total**
Maximum score:	16	27	14	16	19	9	19	120
Reached score:								

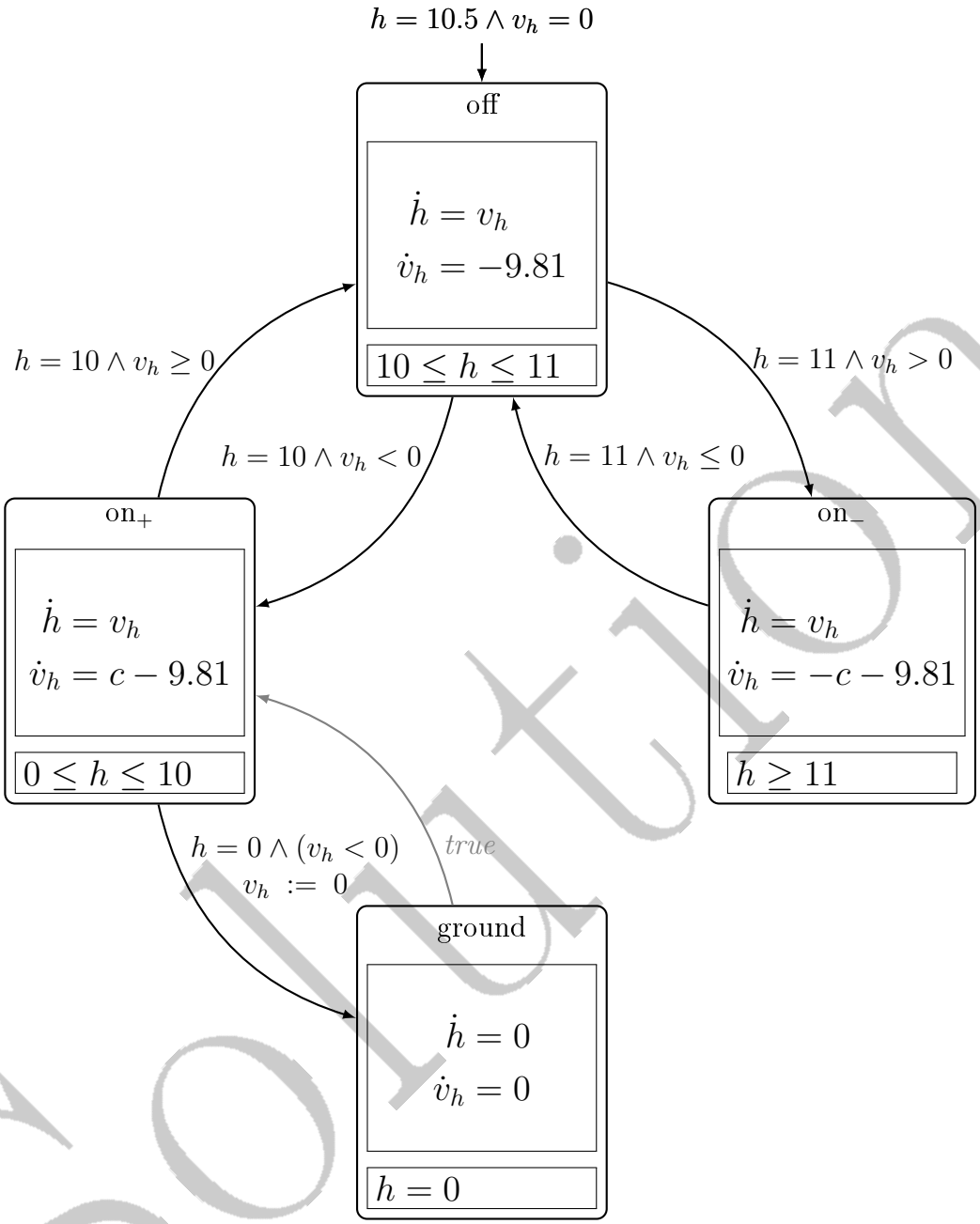
Good luck!

Task 1. Hybrid systems modeling (13 + 3 points)

a) We want to create a model for a digital controller for the stability regulation of a helicopter subject to the following constraints:

- The simplified helicopter monitors its current height h $[m]$ and its vertical velocity v_h $[m/s]$.
- The motor of the helicopter has 3 states, either it is $\mathbf{on}_+$, when the helicopter has a height $h \leq 10$ (motor pulls upwards), $\mathbf{on}_-$, when the helicopter has a height $h \geq 11$(the motor pulls downwards) or **off**, if its height is $10 \leq h \leq 11$. The motor *cannot switch directly* between $\mathbf{on}_+$ and $\mathbf{on}_-$ but has to be **off** before changing the acceleration.
- The upwards, respectively downwards *accelerating* force of the motor is constant (denoted by c respectively $-c$ $[m/s^2]$).
- Independently of the motor state, earth's gravity accelerates the helicopter with a constant factor of -9.81 $[m/s^2]$. (Note that when the helicopter is on the ground, the helicopter's total vertical acceleration is zero due to *Newton's third law.*)
- Initially, the helicopter is at height 10.5 $[m]$ and the motor is **off** with a vertical velocity $v_h = 0$ $[m/s]$.

Please complete the hybrid automaton below to derive a formal model for the helicopter's movement.


$h = 10.5 \wedge v_h = 0$
off
$\dot{h} = v_h$
$\dot{v}_h = -9.81$
$10 \leq h \leq 11$
$h = 10 \wedge v_h \geq 0$
$h = 11 \wedge v_h > 0$
$h = 10 \wedge v_h < 0$
$h = 11 \wedge v_h \leq 0$
on_+
$\dot{h} = v_h$
$\dot{v}_h = c - 9.81$
$0 \leq h \leq 10$
on_-
$\dot{h} = v_h$
$\dot{v}_h = -c - 9.81$
$h \geq 11$
$h = 0 \wedge (v_h < 0)$
$v_h := 0$
true
ground
$\dot{h} = 0$
$\dot{v}_h = 0$
$h = 0$

b) Please formalize the sufficient condition to ensure non-Zeno behavior for a timed automaton $\mathcal{T}$ as presented in the lecture and argue why it holds.

Let $\mathcal{T}$ be a timed automaton with clocks $\mathcal{C}$ such that for every control cycle

$$l_0 \overset{a_1:g_1,r_1}{\hookrightarrow} l_1 \overset{a_2:g_2,r_2}{\hookrightarrow} l_2 \ldots \overset{a_n:g_n,r_n}{\hookrightarrow} l_n = l_0$$

in $\mathcal{T}$ there exists a clock $x \in \mathcal{C}$ such that

- $x \in r_i$ for some $0 < i \leq n$, and
- for all evaluations $\nu \in V$ there exist some $0 < j \leq n$ and $d \in \mathbb{N}^{>0}$ with

$$\nu(x) < d \quad \textit{implies} \quad (\nu \not\models \mathit{Inv}(l_j) \ \textit{or}\ \nu \not\models g_j).$$

Then $\mathcal{T}$ is non-Zeno.

This sufficient condition states, that there is at least one state in every control cycle which enforces that at least d time units pass in this cycle such that there is no way the system can loop without letting time pass.

Task 2. TCTL

(2 + 7 + 15 + 3 points)

a) What does it mean that a hybrid automaton contains a *timelock*?

For a reachable state $\sigma \in \Sigma$ let $paths_{div}(\sigma)$ be the set of time-divergent paths starting in σ.

A state $\sigma \in \Sigma$ contains a *timelock* iff $paths_{div}(\sigma) = \emptyset$.

b) Consider the following TCTL formulas. For each formula φ_i, please construct $\hat{\varphi}_i$ by **first** eliminating syntactic sugar and **second** eliminating timing parameters.

$\varphi_1 = AG^{\leq 4}a$:

Add new clock c_1 which is never reset and use it as follows:

$$\begin{aligned}\hat{\varphi}_1 =& \neg EF^{\leq 4}\neg a \\ \Leftrightarrow& \neg E(trueU^{\leq 4}\neg a) \\ \Leftrightarrow& \neg E(trueU(c_1 \leq 4 \wedge \neg a))\end{aligned}$$

$\varphi_2 = AGEF^{\leq 2}a$:

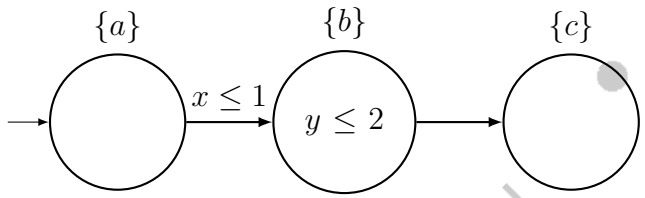
Add new clock c_2 which is never reset and use it as follows

$$\begin{aligned}\hat{\varphi}_2 =& AGE(trueU^{\leq 2}a) \\ \Leftarrow& \neg EF\neg E(trueU^{\leq 2}a) \\ \Leftarrow& \neg E(trueU\neg E(trueU^{\leq 2}a)) \\ \Leftarrow& \neg E(trueU\neg E(trueU(c_2 \leq 2 \wedge a)))\end{aligned}$$

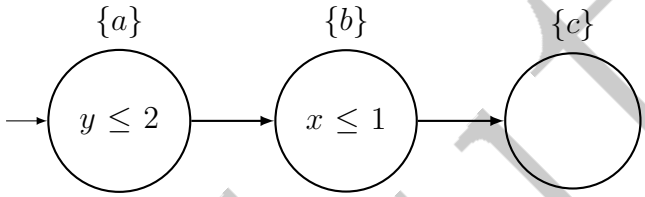
c) Consider the timed automata given below and the following TCTL formulas:

1) $\varphi_1 = EGa$
2) $\varphi_2 = EGa \wedge EFEGb$
3) $\varphi_3 = AF^{\leq 4}c \wedge AF^{\leq 1}b$
4) $\varphi_4 = EGa \wedge EFEGb \wedge EFEGc$
5) $\varphi_5 = AF^{\leq 1}c$
6) $\varphi_6 = AF^{[1,2]}c$

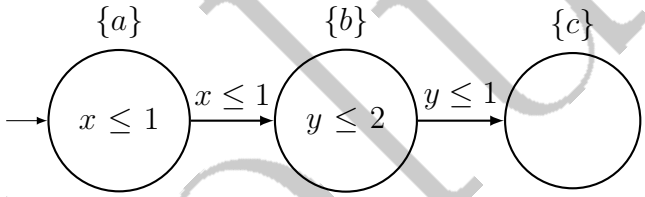
For each automaton, please fill out the table on the right side. Add a symbol +, where the respective formula holds and a symbol - where the formula does not hold.



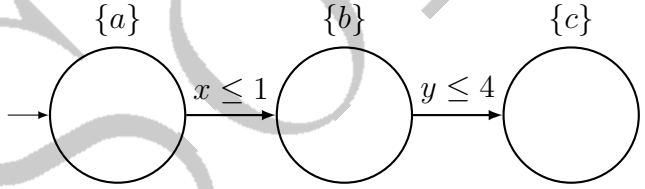
φ_1	φ_2	φ_3	φ_4	φ_5	φ_6
+	-	-	-	-	-



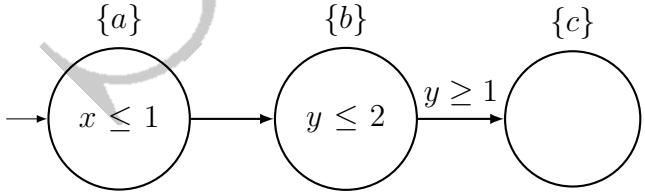
φ_1	φ_2	φ_3	φ_4	φ_5	φ_6
-	-	+	-	+	+



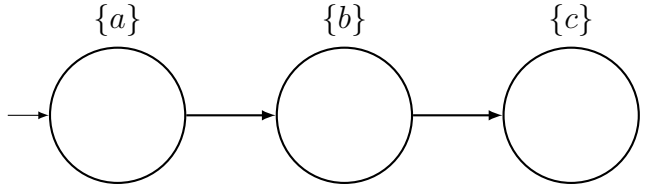
φ_1	φ_2	φ_3	φ_4	φ_5	φ_6
-	-	+	-	+	+



φ_1	φ_2	φ_3	φ_4	φ_5	φ_6
+	+	-	+	-	-



φ_1	φ_2	φ_3	φ_4	φ_5	φ_6
-	-	+	-	-	+



φ_1	φ_2	φ_3	φ_4	φ_5	φ_6
+	+	-	+	-	-

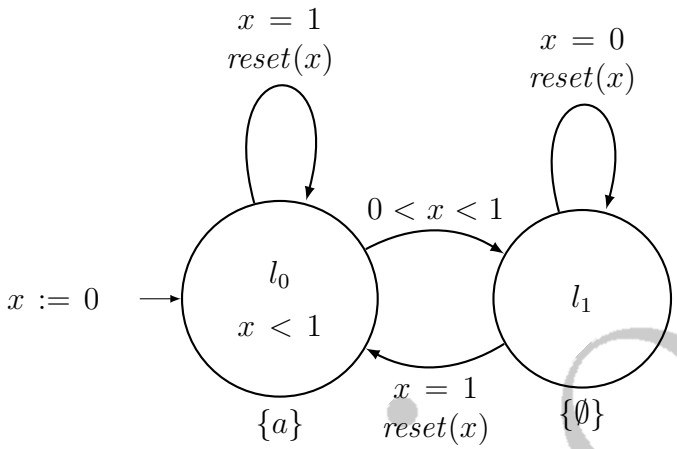
d) Please state, which of the previously (c.f. Task 2c) presented timed automata contain a *timelock* path. Justify your answer!

The second and the third automata contain a timelock:

- second automaton: When the control stays longer than 1 time unit in the first location, the control can never leave this location and the path will eventually invalidate the invariant.
- third automaton: When the control stays in the second location until the global time is larger than 1 time unit, it can never leave this location and will eventually invalidate the invariant.

Task 3. Timed automata model checking (7 + 4 + 3 points)

a) Consider the following timed automaton $\mathcal{T}$ and the TCTL formula $\varphi = EF^{\leq 2}\neg a$:

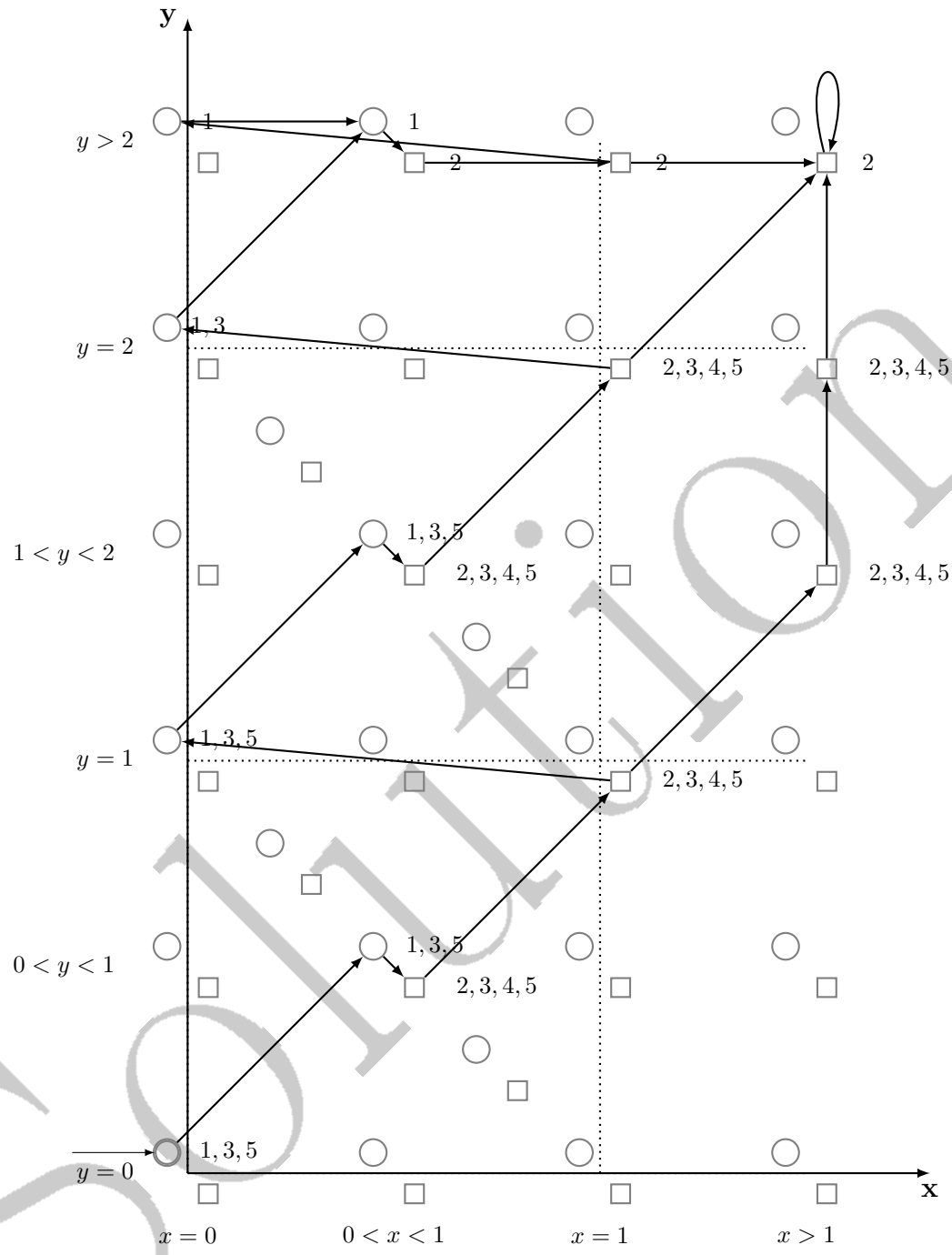


After eliminating timing parameters and syntactic sugar from φ, we obtain:

$$\begin{aligned}\hat{\varphi} &= EF(y \leq 2 \wedge \neg a) \\ &\Leftrightarrow E(true\ \mathcal{U}(y \leq 2 \wedge \neg a))\end{aligned}$$

i) Construct the *region transition system* (*RTS*) $\mathcal{R}$, such that $\mathcal{T} \models_{TCTL} \varphi$ iff $\mathcal{R} \models_{CTL} \hat{\varphi}$ with $\hat{\varphi} = E(true\ \mathcal{U}(y \leq 2 \wedge \neg a))$. As $\mathcal{R}$ will become big, use the prepared grid below to sketch the RTS (by adding the RTS 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 determines, which clock region the state represents.
- Please draw only the reachable fragment of $\mathcal{R}$.



ii) Apply *CTL model checking* to determine whether or not $\mathcal{R} \models_{CTL} \hat{\varphi}$. Please use the provided subformula naming and label *all reachable* RTS states on the previous page **either with** ψ_i **or** $\neg\psi_i$. Does $\hat{\varphi}$ hold in $\mathcal{R}$, i.e., does $\mathcal{R} \models_{CTL} \hat{\varphi}$ hold?

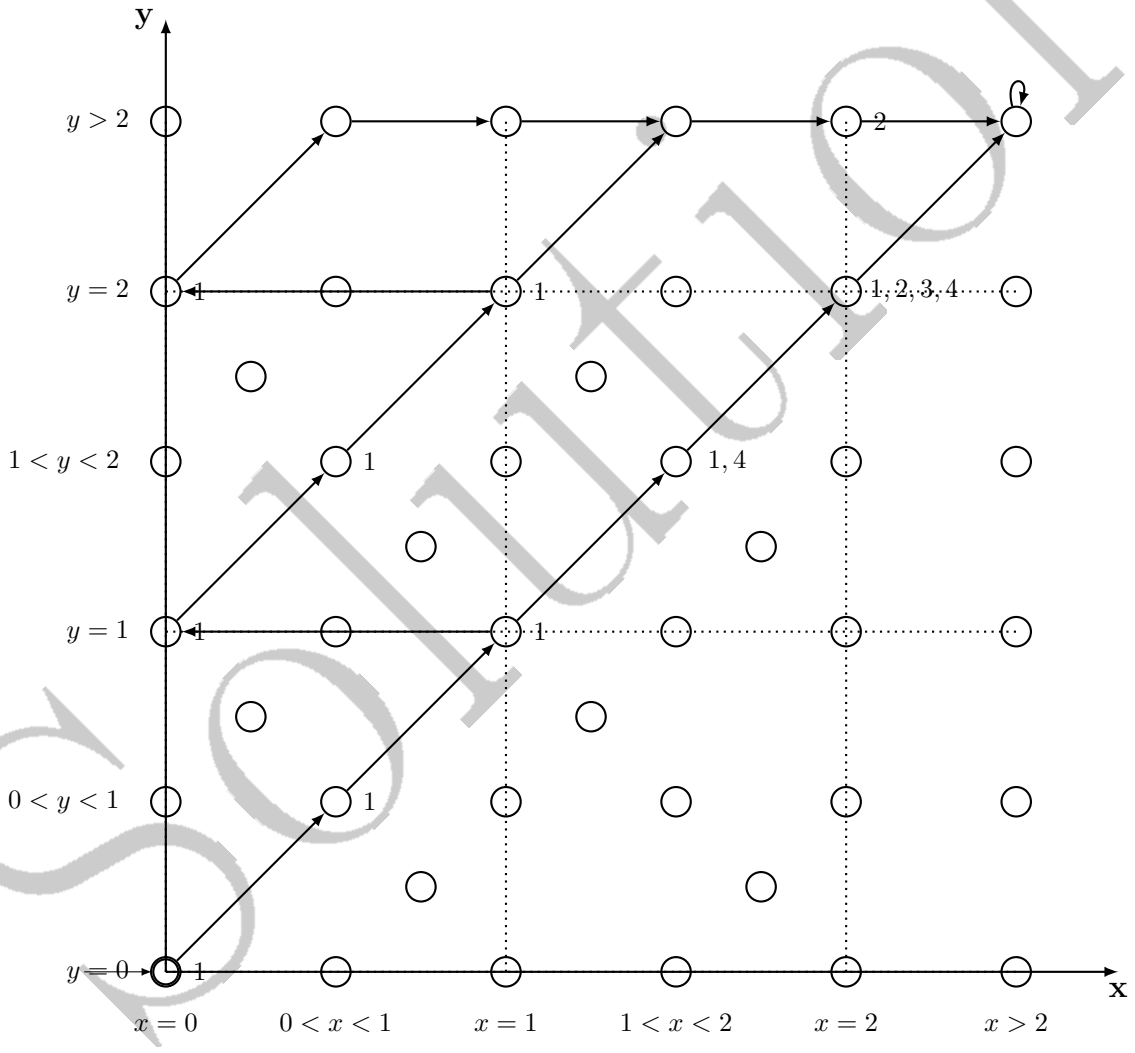
$$\hat{\varphi} = E(\mathit{true}\ \mathcal{U}\ \underbrace{\underbrace{(y \leq 2)}_{\psi_3} \wedge \underbrace{\neg\ \underbrace{a}_{\psi_1}}_{\psi_2}}_{\psi_4})}_{\psi_5}$$

As the initial state with $y = 0$ is labeled with the subformula representing the whole formula, $\mathcal{R} \models_{CTL} \hat{\varphi}$ holds.

b) Please perform *CTL model checking* on the RTS $\mathcal{R}$ given below and determine whether or not $\mathcal{R} \models_{CTL} \hat{\varphi}$ holds, where

$$\hat{\varphi} = A(\mathit{true}\ U\ (\underbrace{\underbrace{y \leq 2}_{\psi_1} \wedge \underbrace{x = 2}_{\psi_2}}_{\psi_3}))}_{\psi_4}.$$

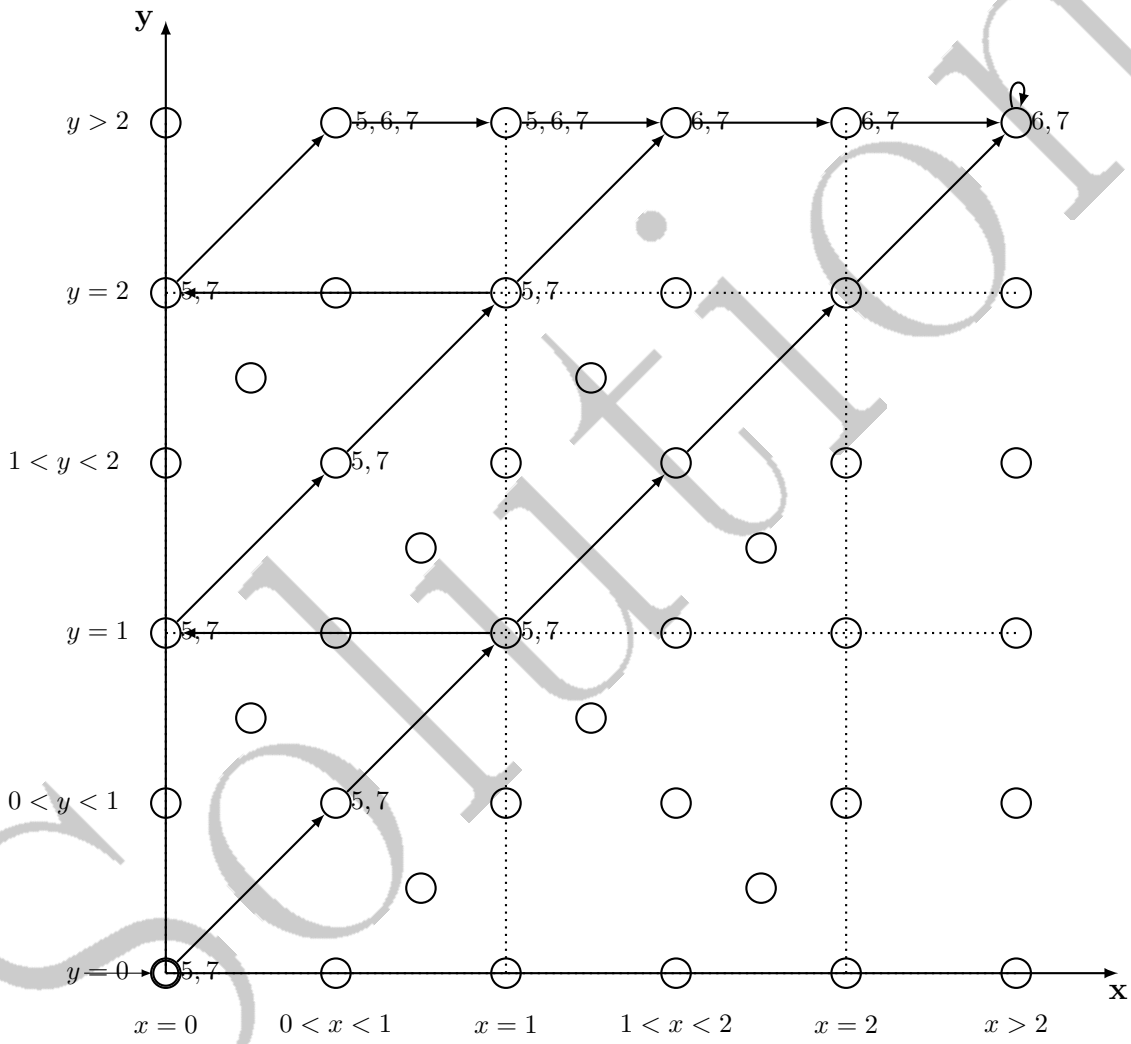
Please use the provided subformula naming and label **all reachable** RTS states **either with** ψ_i **or** $\neg\psi_i$ for every subformula. Does $\hat{\varphi}$ hold in $\mathcal{R}$, i.e., does $\mathcal{R} \models_{CTL} \hat{\varphi}$ hold?



c) Please perform *CTL model checking* on the RTS $\mathcal{R}$ given below and determine whether or not $\mathcal{R} \models_{CTL} \hat{\varphi}$ using

$$\hat{\varphi} = \underbrace{E(\underbrace{x \leq 1}_{\psi_5}\ U(\underbrace{y > 2}_{\psi_6}))}_{\psi_7}.$$

Please use the provided subformula naming and label **all reachable** RTS states **either with** ψ_i **or** $\neg\psi_i$ for every subformula. Does $\hat{\varphi}$ hold in $\mathcal{R}$, i.e., does $\mathcal{R} \models_{CTL} \hat{\varphi}$ hold?



Task 4. Rectangular automata

(2 + 2 + 5 + 7 points)

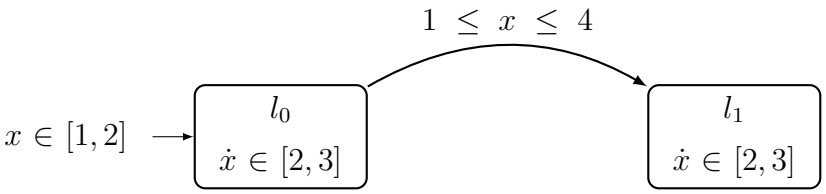
a) When is a hybrid automaton called *initialized*?

A hybrid automaton is called initialized, when the following holds:
Whenever the slope of any variable x changes when taking a discrete transition, the valuation $\nu(x)$ of x is explicitly set in the reset function of the respective transition.

b) Is there a timed automaton, which is not initialized? If yes, give an example, if not, argue why!

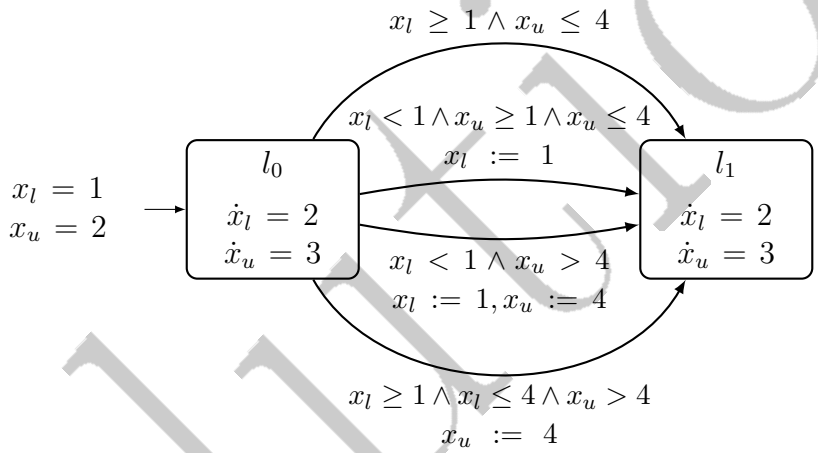
Every timed automaton is automatically intialized, as the slope of every variable is 1 and is never changed - thus no reset is required on discrete transition.

c) Consider the following initialized rectangular automaton $\mathcal{R}$:

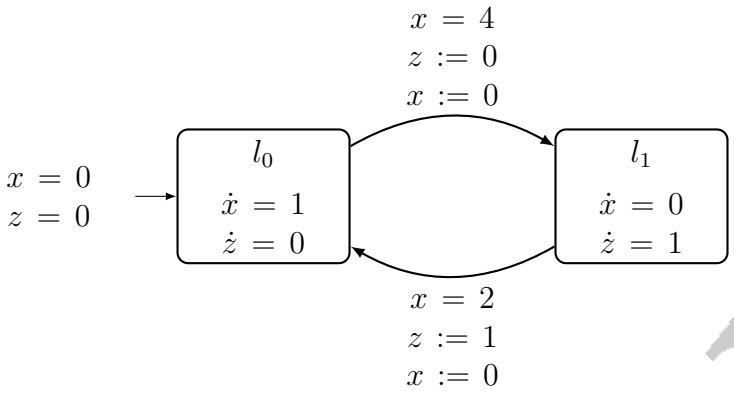


Please reduce $\mathcal{R}$ to an *initialized singular* automaton $\mathcal{R}'$ using the transformation presented in the lecture.

Solution:

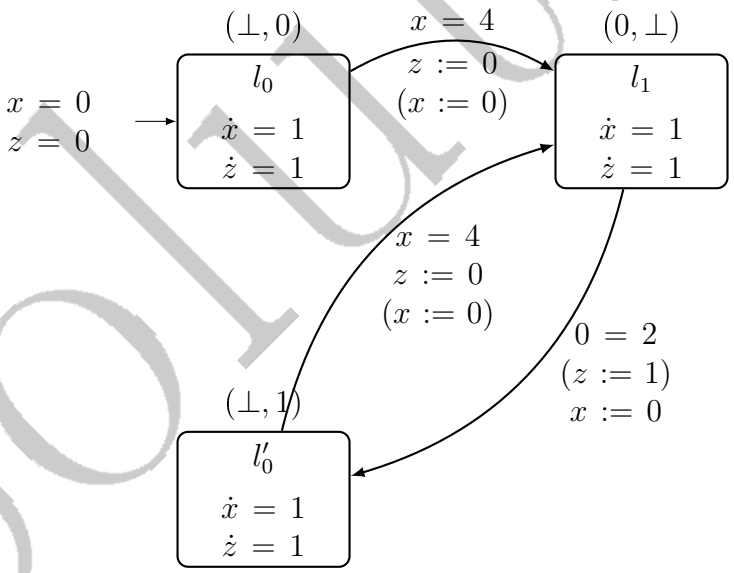


d) Consider the following initialized stopwatch automaton $\mathcal{S}$:



Please transform $\mathcal{S}$ to a *timed automaton* $\mathcal{T}$ which allows clock resets to arbitrary constrants $c \geq 0$. Please create the *full* automaton, not only the reachable fragment.

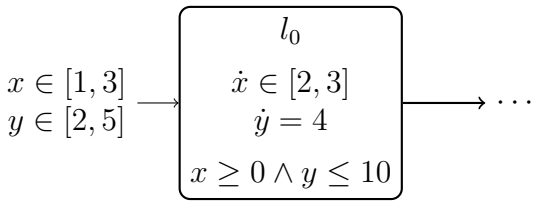
Solution: We transform $\mathcal{S}$ to a timed automaton by adjusting clocks and annotating nodes, which requires to create a new node l_0':



Task 5. Linear hybrid automata

(11 + 8 points)

Consider the following excerpt of a linear hybrid automaton:



a) Please compute the set $T^+_{l_0}(1 \leq x \leq 3 \wedge 2 \leq y \leq 5)$ reachable from $x \in [1,3] \wedge y \in [2,5]$ in location l_0 by letting *time elapse*, using forward analysis as presented in the lecture. Reduce your result whenever possible and eliminate all quantifiers in the order y^{pre}, x^{pre}, t. Please use *Gaussian elimination* when possible and *Fourier-Motzkin variable elimination* otherwise, and eliminate all fractions from your final result!

$$
\begin{aligned}
&T^+_{l_0}(1 \leq x \leq 3 \wedge 2 \leq y \leq 5)\\
&= \exists x^{pre}.\underline{\exists y^{pre}}.\exists t.\ t \geq 0 \wedge x^{pre} \geq 1 \wedge x^{pre} \leq 3 \wedge y^{pre} \geq 2 \wedge y^{pre} \leq 5 \wedge\\
&\quad x \geq x^{pre} + 2t \wedge x \leq x^{pre} + 3t \wedge\\
&\quad \underline{y = y^{pre} + 4t} \wedge\\
&\quad x \geq 0 \wedge y \leq 10
\end{aligned}
$$

Eliminate x^{pre}:

$\leq$	
1	3
$x - 3t$	$x - 2t$

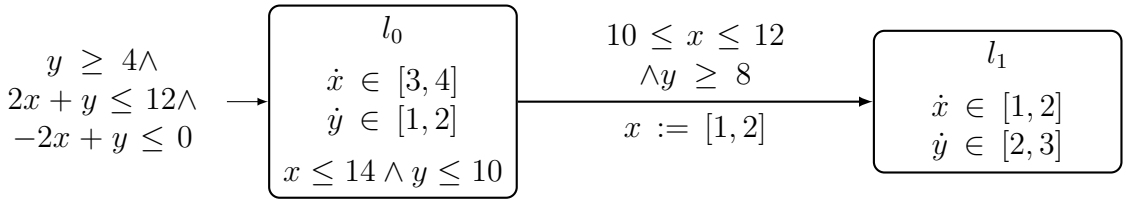
$$
\begin{aligned}
&= \exists x^{pre}.\exists t.\ t \geq 0 \wedge \underline{x^{pre} \geq 1} \wedge \overline{x^{pre} \leq 3} \wedge y - 4t \geq 2 \wedge y - 4t \leq 5 \wedge\\
&\quad \overline{x \geq x^{pre} + 2t} \wedge \underline{x \leq x^{pre} + 3t} \wedge\\
&\quad x \geq 0 \wedge y \leq 10
\end{aligned}
$$

Eliminate t:

$\leq$	
0	$\frac{y}{4} - \frac{1}{2}$
$\frac{y}{4} - \frac{5}{4}$	$\frac{x}{2} - \frac{1}{2}$
$\frac{x}{3} - 1$	

$$
\begin{aligned}
&= \exists t.\ \underline{t \geq 0} \wedge \overline{y - 4t \geq 2} \wedge \underline{y - 4t \leq 5} \wedge\\
&\quad \underline{x - 3t \leq 3} \wedge \overline{1 \leq x - 2t} \wedge\\
&\quad x \geq 0 \wedge y \leq 10\\
&= y \geq 2 \wedge x \geq 1 \wedge\\
&\quad \frac{y}{4} - \frac{5}{4} \leq \frac{x}{2} - \frac{1}{2} \wedge \frac{x}{3} - 1 \leq \frac{y}{4} - \frac{1}{2} \wedge \frac{x}{3} - 1 \leq \frac{x}{2} - \frac{1}{2} \wedge\\
&\quad x \geq 0 \wedge y \leq 10\\
&= y \geq 2 \wedge x \geq 1 \wedge\\
&\quad y - 2x \leq 3 \wedge 4x - 3y \leq 6 \wedge x \geq -3 \wedge\\
&\quad x \geq 0 \wedge y \leq 10\\
&= x \geq 1 \wedge 2 \leq y \leq 10 \wedge y - 2x \leq 3 \wedge 4x - 3y \leq 6
\end{aligned}
$$

b) Consider the following linear hybrid automaton:



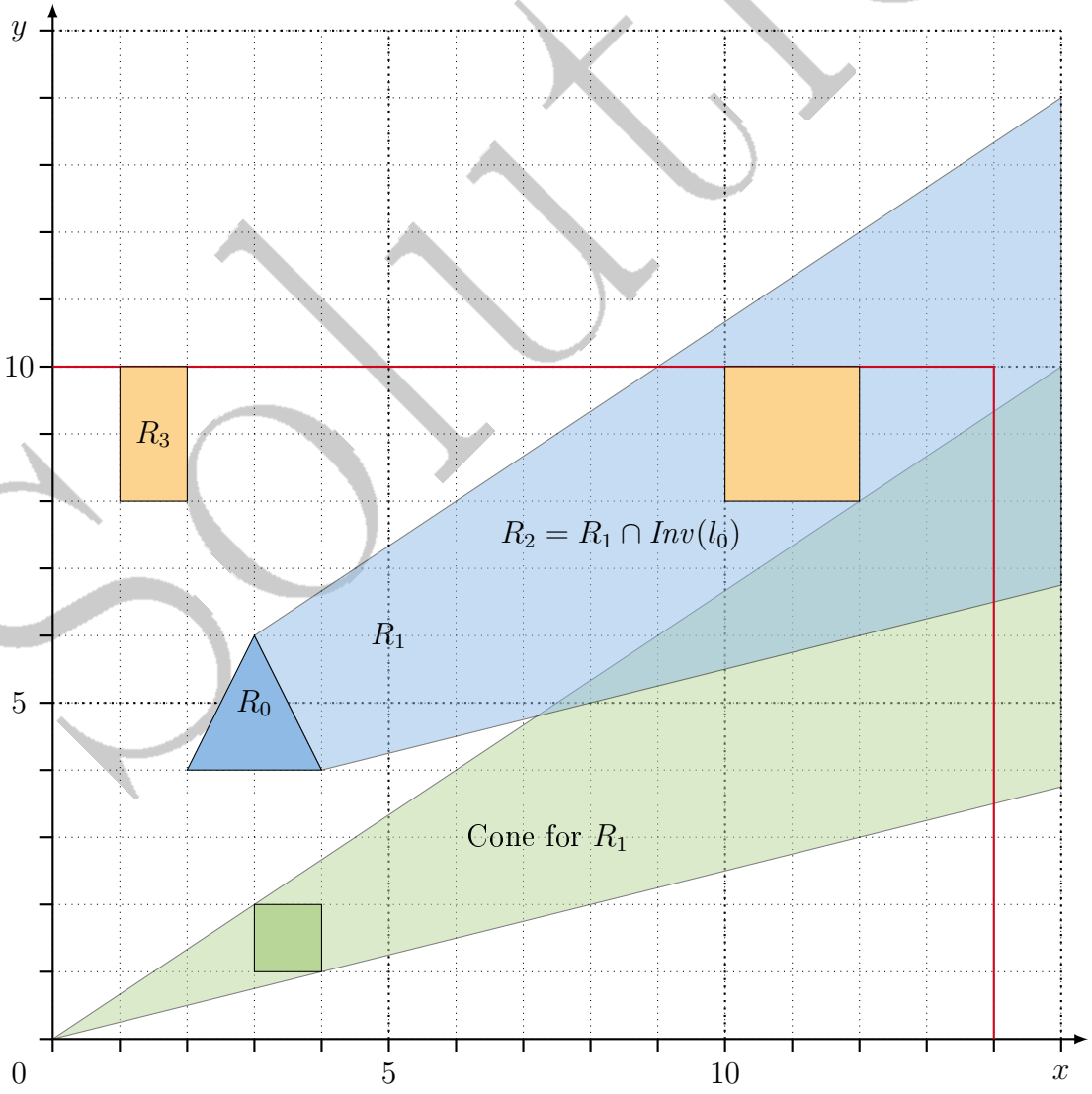
Please use the prepared canvas to sketch the following:

i) Please sketch the initial set R_0 in location l_0.

ii) Extend your sketch by adding the set of reachable states by depicting the states R_1 reachable from R_0 via time ellapse in l_0, without considering invariants. Construct the cone first, then add it to the initial set.

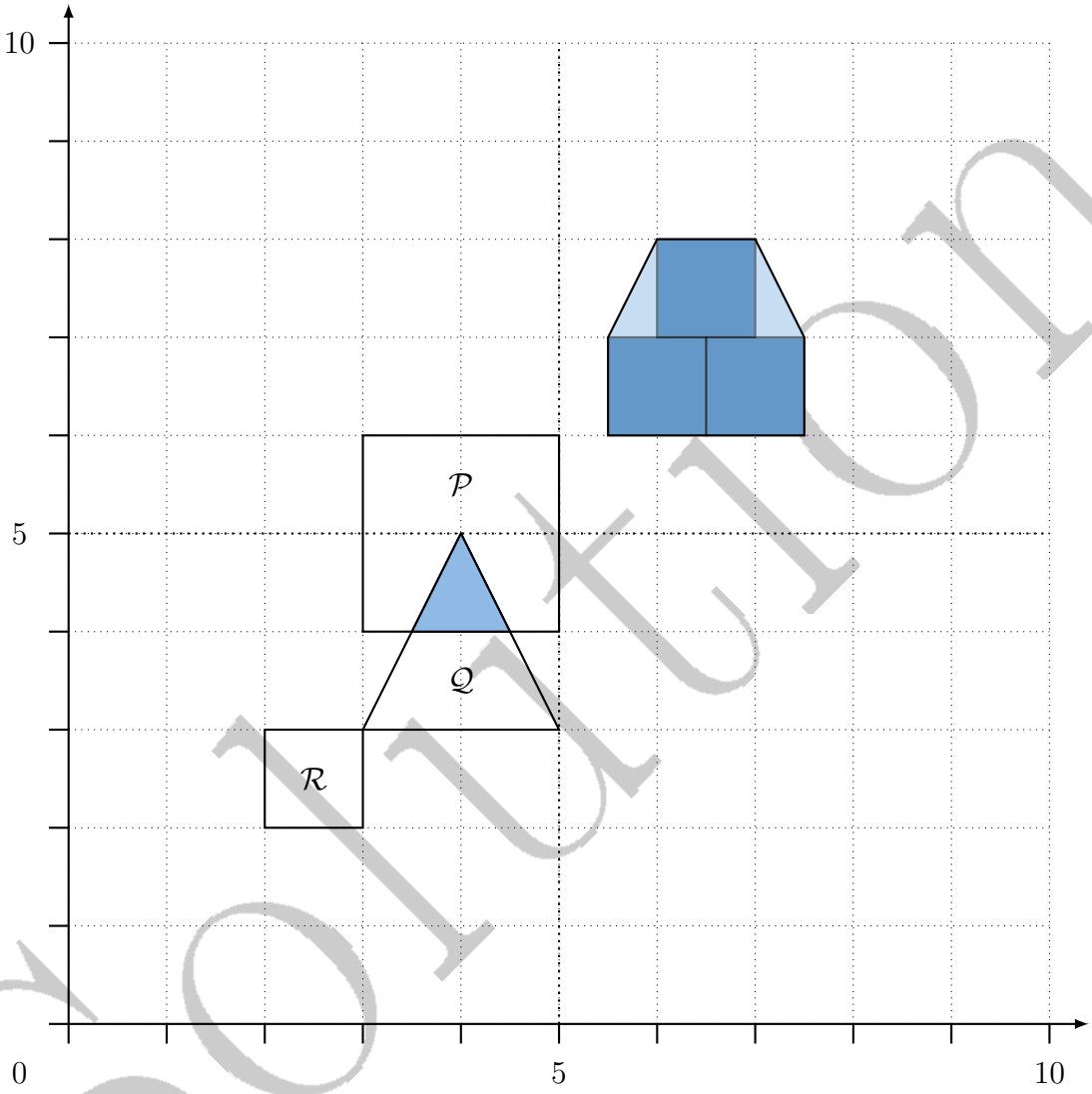
iii) Sketch the time successors R_2 of R_0 in l_0 when considering invariants.

iv) State whether (l_0, R_2) is empty and, if it is not empty, sketch the state set R_3 reachable from it in l_1 via the jump from (l_0, R_2).



Task 6. Polyhedra and Boxes (3 + 6 points)

a) Given three convex polyhedra $\mathcal{P}, \mathcal{Q}, \mathcal{R}$, please sketch below the result of the nested operation $\mathcal{R} \oplus (\mathcal{P} \cap \mathcal{Q})$.



b) Assume two boxes $A, B \in \mathbb{R}^d$ represented by the d-dimensional interval vectors

$$I_A = [\underline{a_0}, \overline{a_0}] \times \ldots \times [\underline{a_d}, \overline{a_d}], \quad I_B = [\underline{b_0}, \overline{b_0}] \times \ldots \times [\underline{b_d}, \overline{b_d}].$$

Please give a formal definition for the following operations. In case boxes are not closed under this operation, make sure the result is the smallest box over-approximating the resulting set.

i) Union of A, B: $C = A \cup B$

$$I_C = [\min(\underline{a_0}, \underline{b_0}), \max(\overline{a_0}, \overline{b_0})] \times \ldots \times [\min(\underline{a_d}, \underline{b_d}), \max(\overline{a_d}, \overline{b_d})]$$

ii) Minkowski-sum of A, B: $C = A \oplus B$

$$I_C = [\underline{a_0} + \underline{b_0}, \overline{a_0} + \overline{b_0}] \times \ldots \times [\underline{a_d} + \underline{b_d}, \overline{a_d} + \overline{b_d}]$$

Task 7. General hybrid automaton reachability analysis

(14 + 2 + 2 + 1 points)

a) Assume a location of a hybrid automaton with the following matrix A representing the flow and $\mathcal{I}$ being an initial state set:

$$A = \begin{pmatrix} 0 & 2 \\ 0 & 0 \end{pmatrix}, \quad \mathcal{I} = convHull\left\{\begin{pmatrix}1\\1\end{pmatrix}, \begin{pmatrix}1\\2\end{pmatrix}, \begin{pmatrix}2\\1\end{pmatrix}, \begin{pmatrix}2\\2\end{pmatrix}\right\}$$

Please compute the matrix exponential exactly for time step length $\delta = 1$. Note that you do not need to extend the matrix to cope with constants. Assume that the box $B_1 = [-1, 1]^2$ can be used for bloating to account for non-linear behavior. Please compute the over-approximation Ω_0 of the first flowpipe segment using $\mathcal{V}$-polytopes as a state set representation starting from the initial set $\mathcal{I}$. Please sketch your result in the prepared canvas and give a *reduced* list of vertices representing the computed first segment.

The first set can be obtained by computing $convHull\left\{\mathcal{I}, e^{\delta A}\mathcal{I} \oplus \mathcal{B}_\alpha\right\}$ where: $e^{\delta A} = \sum_{k=0}^{\infty} \frac{(\delta A)^k}{k!} = \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix} + \begin{pmatrix}0 & 2\\0 & 0\end{pmatrix} = \begin{pmatrix}1 & 2\\0 & 1\end{pmatrix}$ such that

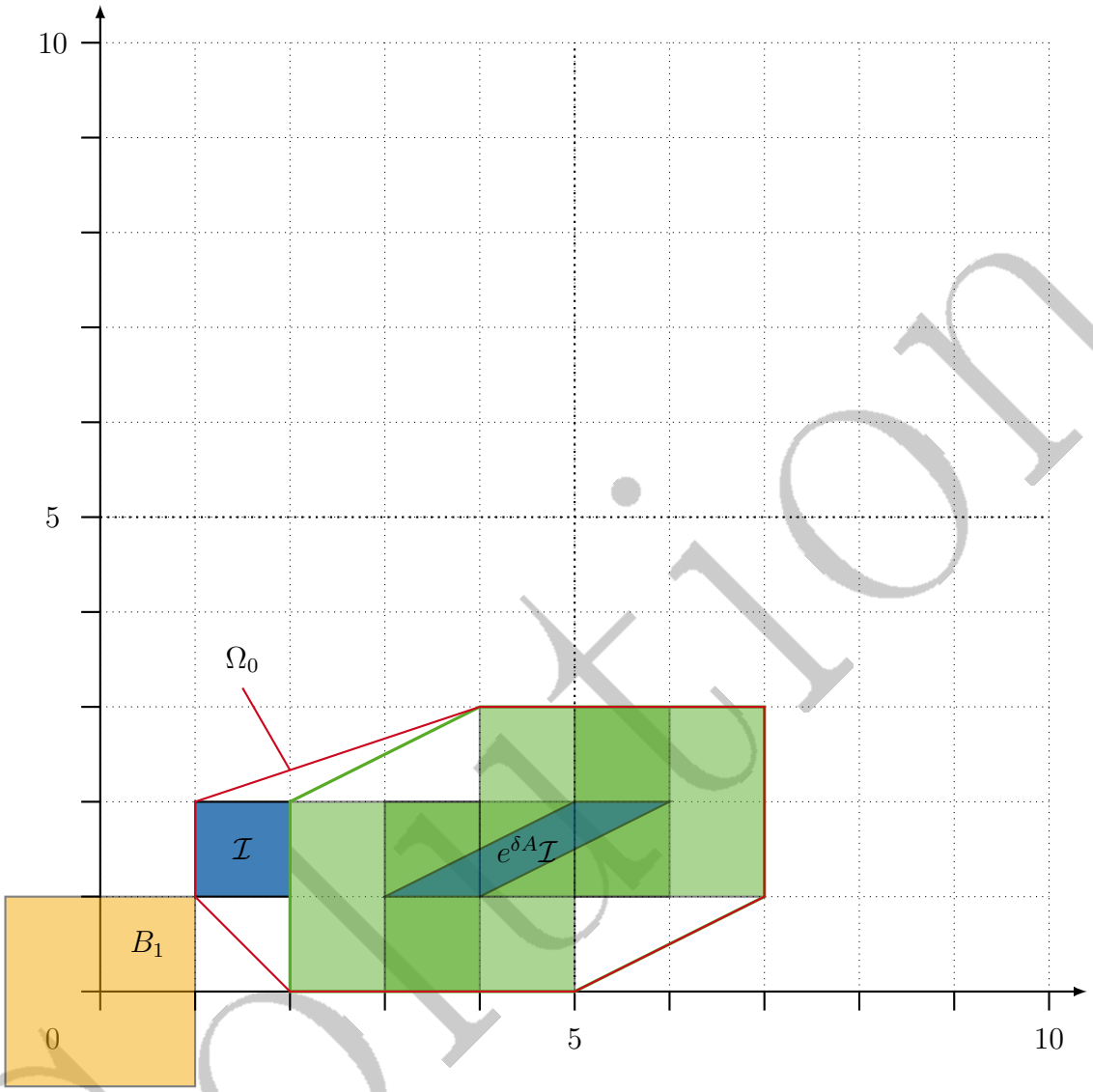
$e^{\delta A}\mathcal{I} = convHull\left\{\begin{pmatrix}3\\1\end{pmatrix}, \begin{pmatrix}5\\2\end{pmatrix}, \begin{pmatrix}4\\1\end{pmatrix}, \begin{pmatrix}6\\2\end{pmatrix}\right\}$. We use $[-1, 1]^2$ for bloating:

$e^{A}\mathcal{I} \oplus \mathcal{B}_1 = convHull\left\{\underline{\begin{pmatrix}2\\0\end{pmatrix}}, \begin{pmatrix}3\\0\end{pmatrix}, \begin{pmatrix}4\\0\end{pmatrix}, \underline{\begin{pmatrix}5\\0\end{pmatrix}}, \begin{pmatrix}4\\1\end{pmatrix}, \begin{pmatrix}5\\1\end{pmatrix}, \begin{pmatrix}6\\1\end{pmatrix}, \underline{\begin{pmatrix}7\\1\end{pmatrix}}, \underline{\begin{pmatrix}2\\2\end{pmatrix}}, \begin{pmatrix}3\\2\end{pmatrix}, \begin{pmatrix}4\\2\end{pmatrix}, \begin{pmatrix}5\\2\end{pmatrix}, \underline{\begin{pmatrix}4\\3\end{pmatrix}}, \begin{pmatrix}5\\3\end{pmatrix}, \begin{pmatrix}6\\3\end{pmatrix}, \underline{\begin{pmatrix}7\\3\end{pmatrix}}\right\}$. The whole first segment now is the convex hull of the initial set $\mathcal{I}$ and the previously computed bloated set such that

$\Omega_0 = convHull\left\{\underline{\begin{pmatrix}1\\1\end{pmatrix}}, \underline{\begin{pmatrix}1\\2\end{pmatrix}}, \begin{pmatrix}2\\1\end{pmatrix}, \begin{pmatrix}2\\2\end{pmatrix}, \underline{\begin{pmatrix}2\\0\end{pmatrix}}, \begin{pmatrix}3\\0\end{pmatrix}, \begin{pmatrix}4\\0\end{pmatrix}, \begin{pmatrix}5\\0\end{pmatrix}, \begin{pmatrix}4\\1\end{pmatrix}, \begin{pmatrix}5\\1\end{pmatrix}, \begin{pmatrix}6\\1\end{pmatrix}, \underline{\begin{pmatrix}7\\1\end{pmatrix}}, \underline{\begin{pmatrix}2\\2\end{pmatrix}}, \begin{pmatrix}3\\2\end{pmatrix}, \begin{pmatrix}4\\2\end{pmatrix}, \begin{pmatrix}5\\2\end{pmatrix}, \underline{\begin{pmatrix}4\\3\end{pmatrix}}, \begin{pmatrix}5\\3\end{pmatrix}, \begin{pmatrix}6\\3\end{pmatrix}, \underline{\begin{pmatrix}7\\3\end{pmatrix}}\right\}$

$\Omega_0 = convHull\left\{\begin{pmatrix}1\\1\end{pmatrix}, \begin{pmatrix}2\\0\end{pmatrix}, \begin{pmatrix}5\\0\end{pmatrix}, \begin{pmatrix}7\\1\end{pmatrix}, \begin{pmatrix}7\\3\end{pmatrix}, \begin{pmatrix}4\\3\end{pmatrix}, \begin{pmatrix}1\\2\end{pmatrix}\right\}$. The resulting set and all intermediate sets can be visualized as shown in the canvas below.

Visualization:



b) Please specify, why the *choice of state set representation* is crucial in the reachability analysis for hybrid systems.

The choice of an appropriate state set representation is always a trade-off between precision and computational effort. A more precise representation reduces the over-approximation error but usually introduces more complex computations. A less precise representation reduces the computational effort but introduces a larger over-approximation error.

c) For which computation steps in the reachability analysis for hybrid automata is the operation *intersection* needed and why?

Intersection is used whenever we want to test, if a guard is satisfied. When the intersection is nonempty, the corresponding transition is enabled and we can take a jump to the respective target location. Another step where intersection is needed is when verifying against the invariant or to check if the bad states are reachable.

d) If we use over-approximative computations, which information can be derived from reachability analysis results for hybrid automata?

- If the set of reachable states intersects with the set of bad states we cannot derive any information.
- If the set of reachable states does not intersect with the set of bad states we can derive that the system is safe.