Modeling and Analysis of Hybrid Systems - SS 2015

Second Exam

Wednesday, September 16, 2015

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.

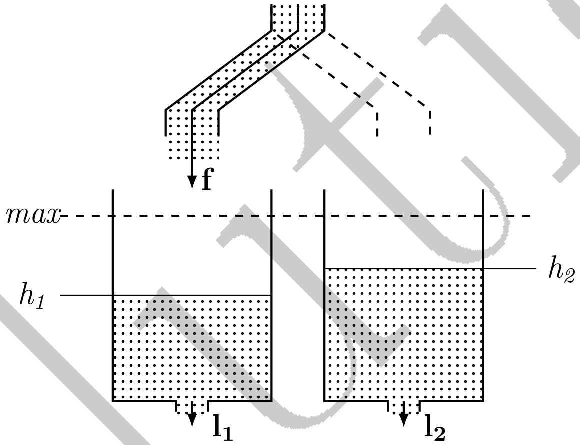
Task:	1.)	2.)	3.)	4.)	5.)	6.)	**Total**
Maximum score:	13	17	14	19	10	7	80
Reached score:							

Good luck!

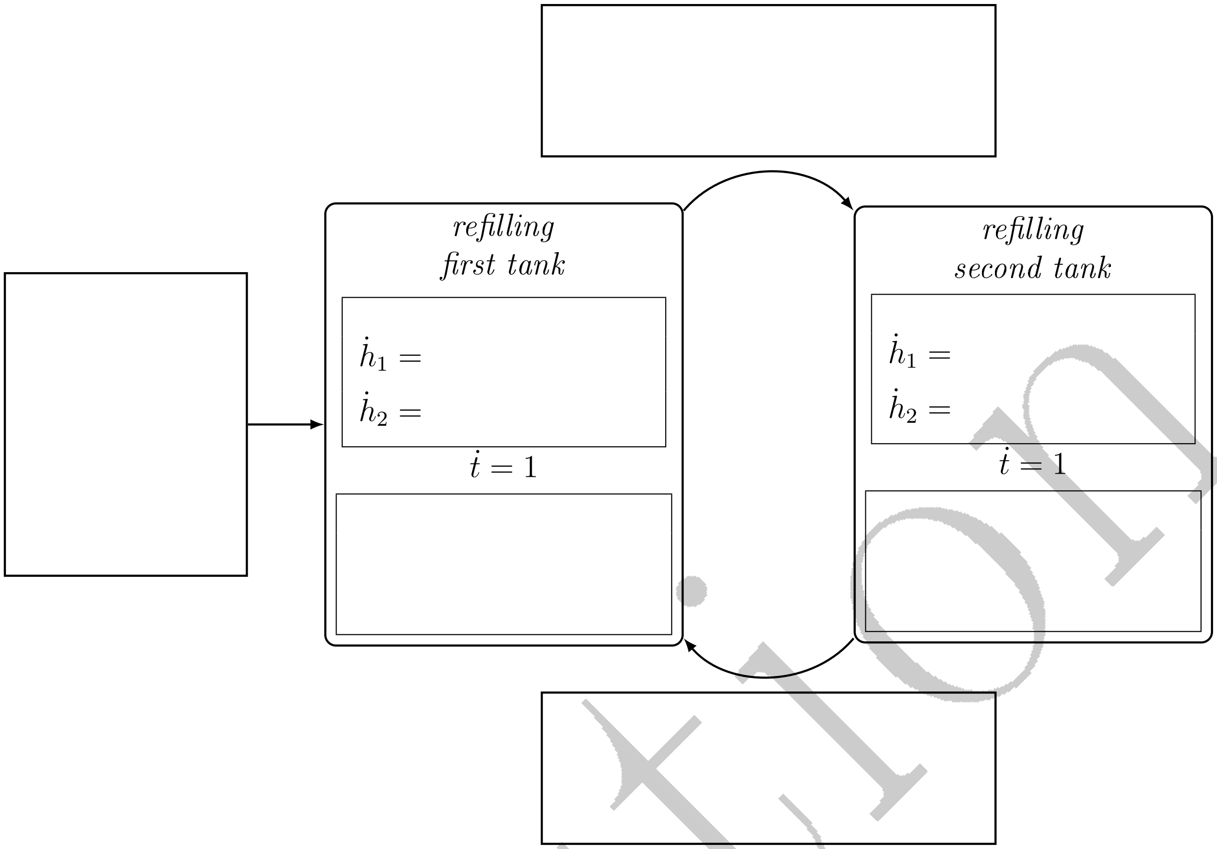
Task 1. Hybrid systems modeling (5 + 3 + 5 points)

a) Consider a system of *two leaking water tanks* specified as follows:

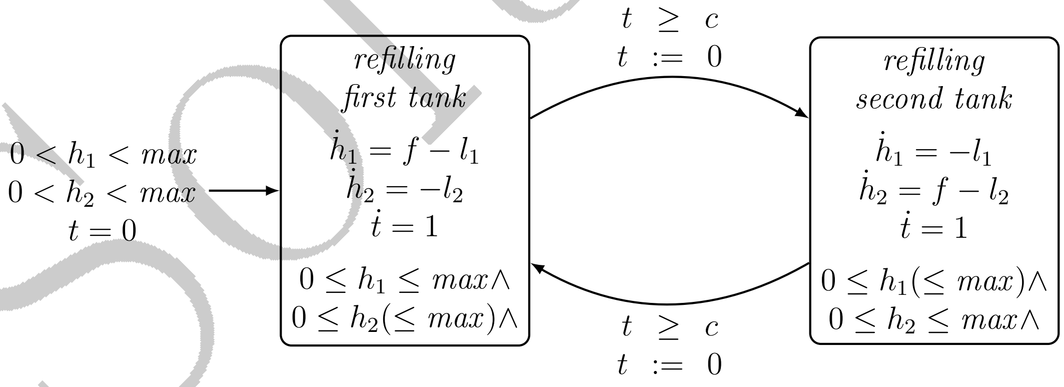
- There are two identically shaped water tanks with water levels of h_1 respectively h_2 height units.
- Both tanks are leaking with a *constant outflow*, leading to a water level decrease of l_1 respectively l_2 ($l_1, l_2 > 0$) height units per time unit.
- A water hose refills *exactly one tank at a time* with a *constant inflow* of f ($f > 0$) height units per time unit for *at least* $c \geq 0$ time units.
- The water levels never go above a *maximal water level* of max ($max > 0$) height units and it holds for each $h_i \geq 0$.
- Initially, the water levels in both tanks are larger than zero and less than max, and the hose is refilling the first tank.



Please *fill in the missing parts* in the following incomplete hybrid automaton according to the above specification.


refilling
first tank
$\dot{h}_1 =$
$\dot{h}_2 =$
$\dot{t} = 1$
refilling
second tank
$\dot{h}_1 =$
$\dot{h}_2 =$
$\dot{t} = 1$

Solution:


$0 < h_1 < max$
$0 < h_2 < max$
$t = 0$
refilling
first tank
$\dot{h}_1 = f - l_1$
$\dot{h}_2 = -l_2$
$\dot{t} = 1$
$0 \leq h_1 \leq max \wedge$
$0 \leq h_2 (\leq max) \wedge$
$t \geq c$
$t := 0$
refilling
second tank
$\dot{h}_1 = -l_1$
$\dot{h}_2 = f - l_2$
$\dot{t} = 1$
$0 \leq h_1 (\leq max) \wedge$
$0 \leq h_2 \leq max \wedge$
$t \geq c$
$t := 0$

b) Please consider the following configurations for the constants f and c in the previous automaton and fill in the last two columns in the table below using ✗ and ✓signs to indicate whether the automaton has any *Zeno* paths and whether the automaton has any reachable *time lock* state.

Solution:

Definition Zeno paths: Paths along which infinitely many discrete steps are performed in a finite amount of time are called Zeno paths. Note that all Zeno paths are time-convergent.

Definition time lock: There could be states in the LSTS of a timed automaton from which all paths are time-convergent, such that there is no possibility that time progresses forever. Such states do not allow time divergence, and are therefore called time locks.

f, l$_\mathbf{i}$	**c**	**Zeno**	**time lock**
$f < l_1 + l_2$	$c = 0$	✓	✓
$f > l_1 + l_2$	$c > 0$	✗	✓
$f < l_1 + l_2$	$0 < c < min\{\frac{l_1}{f}, \frac{l_2}{f}\}$	✗	✓
$f = l_1 + l_2$	$c = 0$	✓	✗
$f = l_1 + l_2$, $l_1 = l_2$	$c > 0$	✗	✓
$f = 2$, $l_1 = l_2 = 1$	$c = max/2$	✗	✓

Zeno: Whenever $c = 0$ there is Zeno behavior.
time lock: Whenever both reach water level 0 or max.

row 1: all paths in this automaton are time lock paths as the outflow is larger than the inflow in the whole system → the invariants will be invalidated eventually.

row 2: at some point the invariants will be invalidated ($h_i \leq max$) and as $c > 0$ you cannot switch locations.

row 3: when initially $0 < h_2 < c \cdot l_2$ holds, the invariant for the second tank will be invalidated.

row 4: no time lock as both values can always be kept between the limits by taking the discrete transition.

row 5: see row 3.

row 6: see row 3.

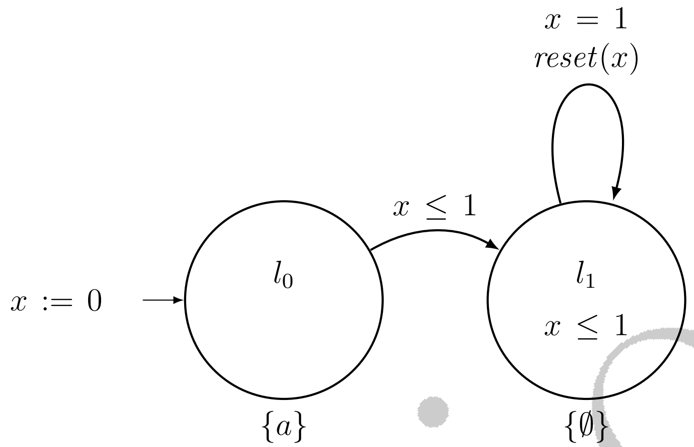
c) Please give the formal *operational semantics of a time step* for general hybrid automata.

Solution:

$$\frac{\begin{array}{c} t \geq 0 \quad f \in Act(l) \quad f(0) = \nu \quad f(t) = \nu' \\ \forall t' \in [0,t].\ f(t') \in Inv(l) \end{array}}{(l,\nu) \xrightarrow{t} (l,\nu')} \; \texttt{Rule}_{\texttt{time}}$$

Task 2. Timed automata **($2+2+5+4+4$ points)**

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



a) Does the timed automaton above have any *reachable timelock states*? Does it have any *Zeno paths*? Please justify your answers!

Solution: The automaton exhibits neither time lock paths nor Zeno behavior, as there is no possibility that a path ends in a location, where only time convergent paths can be taken and there is no transition, which can be taken infinitely often in a finite amount of time, as the first transition can only be taken once and the loop transition can only be taken after a delay of 1 time unit.

b) Please eliminate in φ syntactic sugar (G, F) and *construct* $\hat{\varphi}$ by eliminating timing parameters. Use the name y for the auxiliary clock.

Solution:

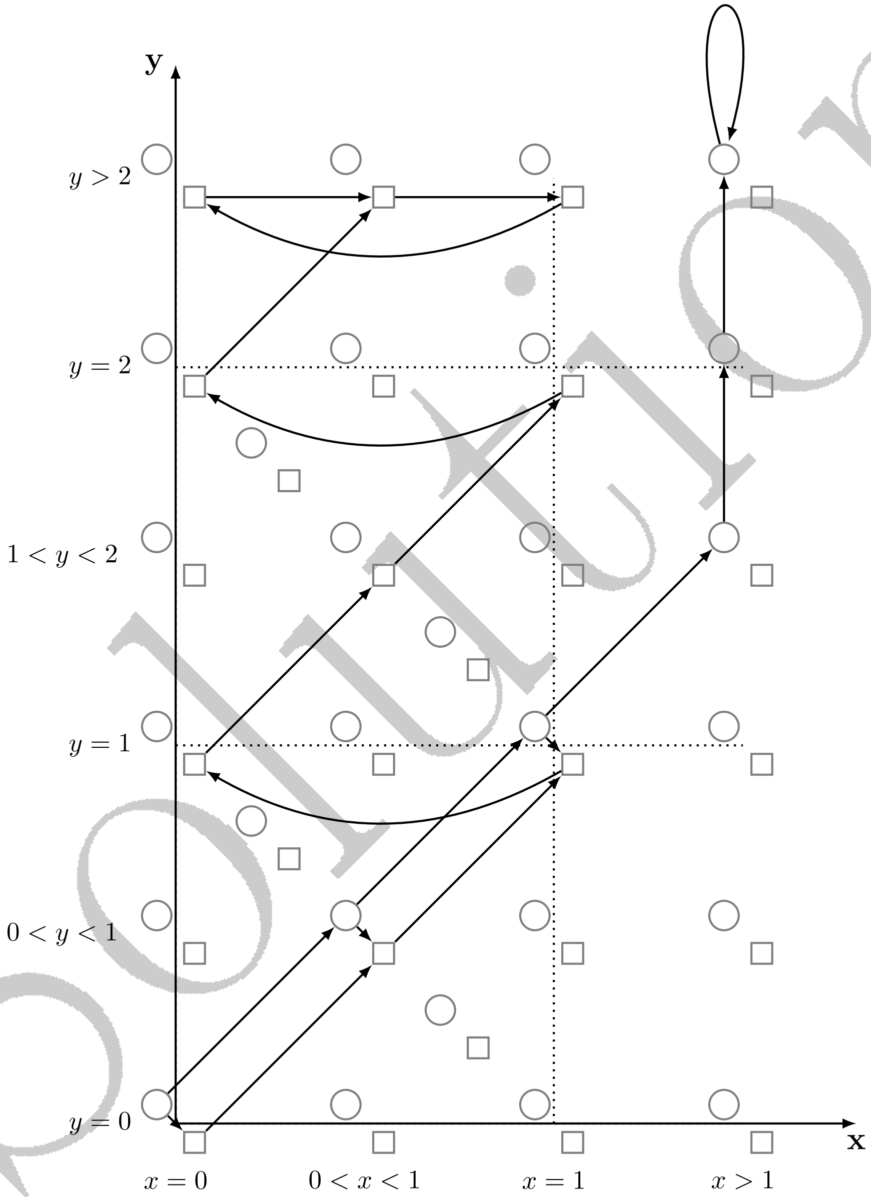
$$\begin{aligned}\varphi &= E\ G\ ^{\leq 2}a\\ &\Leftrightarrow \neg A\ F\ ^{\leq 2}\neg a\\ &\Leftrightarrow \neg A\ true\ U\ ^{\leq 2}\ \neg a\\ \hat{\varphi} &= \neg A\ true\ U\ (y \leq 2 \wedge \neg a)\end{aligned}$$

c) Construct the *region transition system* (*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 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 $\mathcal{R}$.

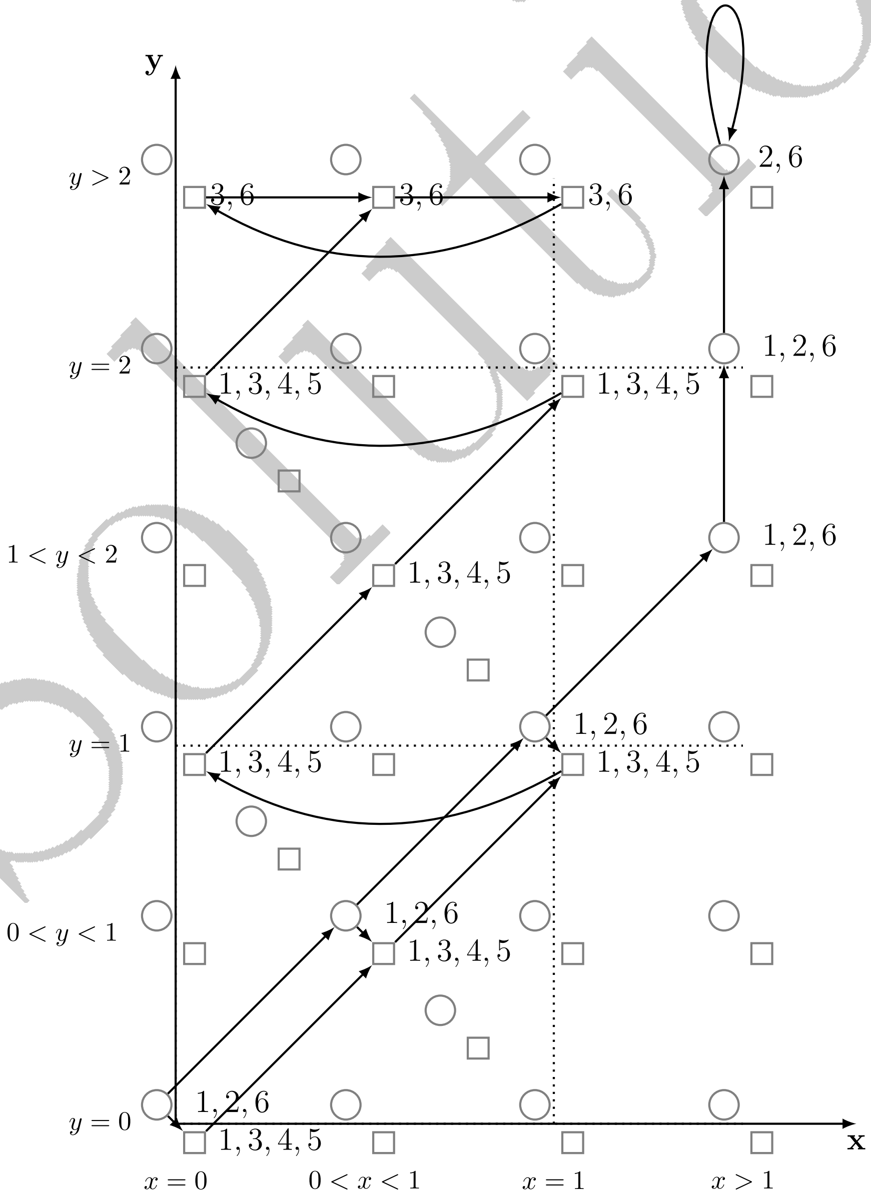
Solution:



d) Apply *CTL model checking* to determine whether or not $\mathcal{R} \models_{CTL} \hat{\varphi}$. Please give names for the subformulas of $\hat{\varphi}$ and label the reachable RTS states with them on the previous page. Does $\hat{\varphi}$ hold in $\mathcal{R}$, i.e., does $\mathcal{R} \models_{CTL} \hat{\varphi}$ hold?

Solution:

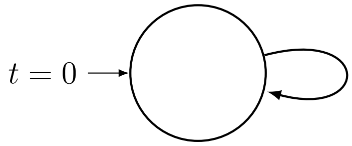
$$\underbrace{\neg\, \underbrace{Atrue\; U\; (\underbrace{\underbrace{y \leq 2}_{\psi_1} \wedge \underbrace{\neg \underbrace{a}_{\psi_2}}_{\psi_3}}_{\psi_4})}_{\psi_5}}_{\psi_6}$$



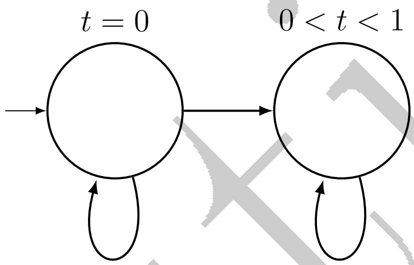
In the initial state where $y = 0$, the formula ψ_4 holds such that $\hat{\varphi}$ holds.

e) Prove by giving a counterexample that the TCTL model checking algorithm from the lecture is not correct in general if the timed automaton has Zeno paths.

Solution: We can give a timed automaton $\mathcal{T}$ as a counterexample:



Considered the formula $\varphi = AF(t > 0)$, the RTS of the automaton is



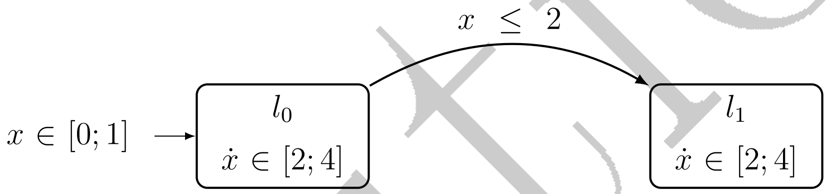
The original automaton satisfies the formula while the RTS abstraction does not as TCTL only considers time divergent paths.

Task 3. Rectangular automata (3 + 5 + 6 points)

a) In the transformation of initialized rectangular automata to timed automata, which transformation steps are not applicable if the rectangular automaton is *not initialized*? Please justify your answer!

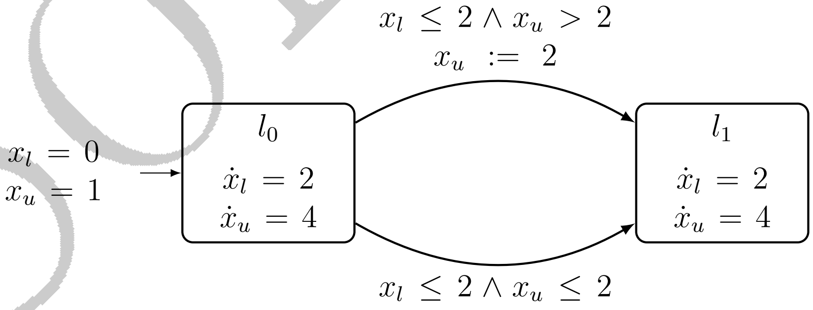
Solution: In the transformation from stopwatch to timed automata we could get uncountably infinite number of copies to remember the value of the stopwatch when it was stopped the last time. In the transformation from singular to stopwatch automata the transitions between locations defining different rates for the clock the transition should rescale the clock value. This is possible, if the value is reset on the transition, however if there is no reset then we must use linear expressions in the reset function which is not in the syntax of timed automata.

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

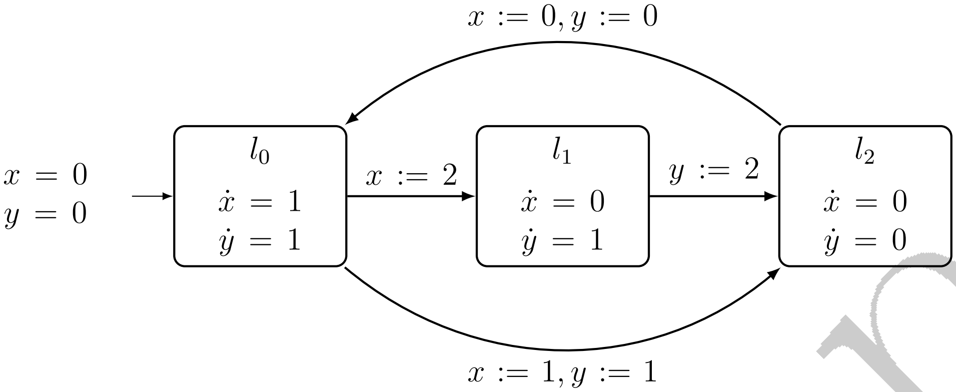


Please reduce $\mathcal{R}$ to an *initialized singular* automaton $\mathcal{R}'$ by using the method presented in the lecture.

Solution:

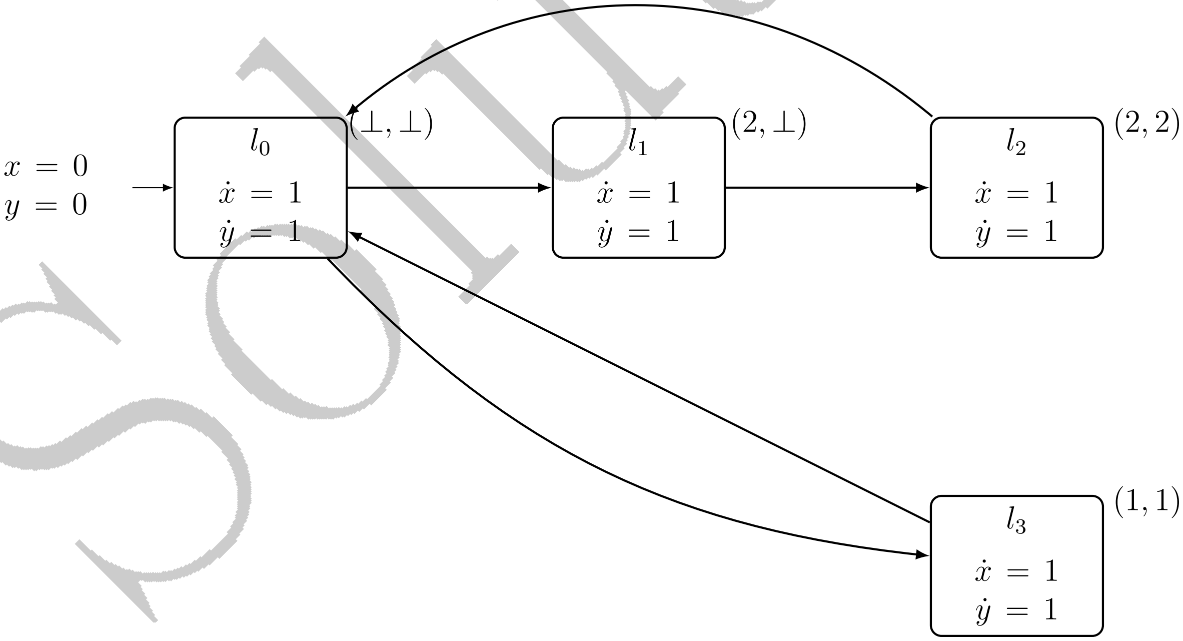


c) Consider the following initialized stopwatch automaton $\mathcal{S}$:



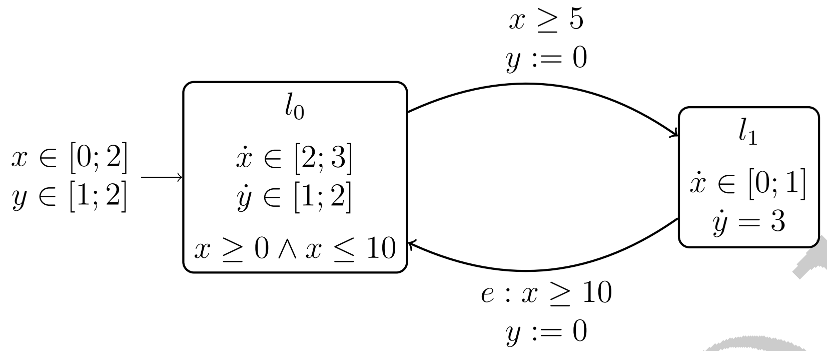
Please transform $\mathcal{S}$ to a *timed* automaton $\mathcal{S}'$.

Solution: We transform $\mathcal{S}$ to timed automaton by adding the information of the current value to locations. Note that in this case this is not necessary but as a general procedure is prefered we need this to cope with possible guards:



Task 4. Linear hybrid automata I (11 + 3 + 2 + 3 points)

Consider the following linear hybrid automaton:



a) Please compute the set $T^+_{l_0}(x \in [0;2] \wedge y \in [1;2])$ reachable from $x \in [0;2] \wedge y \in [1;2]$ 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 (eliminate x^{pre} first, then y^{pre} and then t)!

Solution:

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

b) Please compute the set $D^+_e(x \geq 10 \wedge y \leq 30)$ reachable from $x \geq 10 \wedge y \leq 30$ in location l_1 by *taking the transition* e from l_1 to l_0, using forward analysis as presented in the lecture.

Reduce your result whenever possible and eliminate all quantifiers!

Solution:

$$\begin{aligned} D_e^+(x \geq 10 \wedge y \leq 30) =& \exists x^{pre}.\exists y^{pre}. \\ & x^{pre} \geq 10 \wedge y^{pre} \leq 30 \wedge \\ & x \geq 10 \wedge y = 0 \wedge 0 \leq x \leq 10 \\ =& x \geq 10 \wedge y = 0 \wedge x \geq 0 \wedge x \leq 10 \\ =& x = 10 \wedge y = 0 \end{aligned}$$

c) When does the forward analysis for linear hybrid automata I terminate? Which conclusions can be drawn from the result?

Solution:

Either the algorithm has reached a fix-point and the intersection of the current reachable set with the set of bad states is empty - in this case the system is proven to be safe. Otherwise the algorithm stops whenever the bad states are reachable. However, in this case it is not possible to deduce unsafety of the system as the computation is overapproximating.

d) Can we apply the forward reachability algorithm for linear hybrid automata I also to models with non-convex invariants? Explain your answer!

Solution: No, as the assumptions for time steps $Inv(\nu)$ and $Inv(\nu')$ are no longer sufficient. Recall the definition of a time step:

$$\frac{\begin{array}{c} t \geq 0 \quad f \in Act(l) \quad f(0) = \nu \quad f(t) = \nu' \\ \forall t' \in [0,t].\ f(t') \in Inv(l) \end{array}}{(l,\nu) \xrightarrow{t} (l,\nu')} \ \texttt{Rule}_{\texttt{time}}$$

In this case it can happen that during a time step the invariant gets invalidated.

Task 5. Polyhedra **(3 + 4 + 1 + 2 points)**

a) Which extension to *V-polytopes* is required to be able to represent unbounded sets? Please specify this representation type (V, U) and define the polytope $P = (V, U)$ that is encoded by such a representation!

Solution: To represent unbounded sets we have to extend V-polytopes by cones such that a V-polytope $\mathcal{P} = conv(\mathcal{V}) \oplus coneU$, where $\mathcal{V}$ is the set of vertices of the original

V-polytope and (*cone*) is a polyhedral cone of vectors in U which is added to the convex hull of the vertices. The cone itself represents an unbounded set such that the Minkowski sum of the cone and the convex hull of the vertices allows to represent unbounded sets.

b) Please sketch the idea of checking membership of a point x in a given V-polytope $\mathcal{V}$! What is the worst-case complexity of this operation?

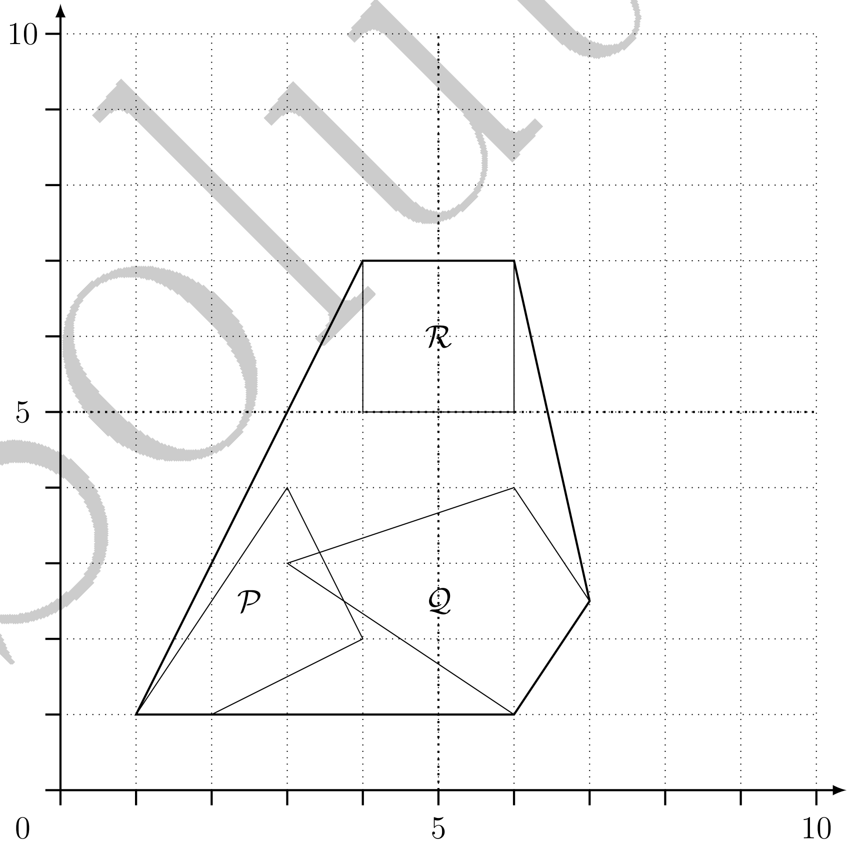
Solution: To check the membership of x for $\mathcal{V}$ we need to be able to represent x as a linear combination of the vertices v_i of $\mathcal{V}$ where the sum of coefficients equals 1:

$$x = \sum_{i}^{n} \lambda_i \cdot v_i, \sum_{i}^{n} \lambda_i = 1$$

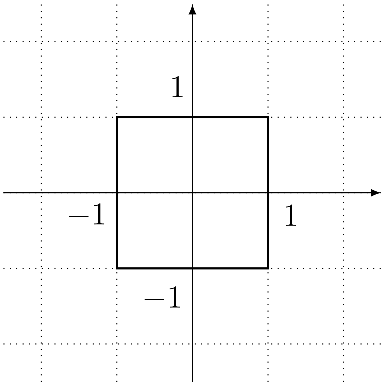
The operation is polynomial in the number of vertices.

c) Given three convex polyhedra $\mathcal{P}, \mathcal{Q}, \mathcal{R}$, please sketch below the result of the operation $conv(conv(\mathcal{P} \bigcup \mathcal{Q}) \bigcup \mathcal{R})$.

Solution:



d) Give the V-representation as a set of vertices V and the H-representation as a pair (C, z) the following set in $\mathbb{R}^2$:



Solution:

V-representation: $V = \{(-1,-1), (-1,1), (1,-1), (1,1)\}$

H-representation: $\left(\begin{pmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}\right)$

Task 6. General hybrid automaton reachability analysis

(3 + 2 + 2 points)

a) What influences has the *time step size* in hybrid automata reachability analysis as presented in the lecture on the resulting flowpipe and its segments?

Solution: The time step decides about the size and number of flowpipe segments. A larger time step results in lesser, larger segments, whereas a smaller time step increases the precision but also the computational effort, as there are more segments in the resulting flowpipe.

b) At which point during reachability analysis can nondeterminism occur and why?

Solution: Whenever a guard is satisfied, the reachability analysis algorithm has to follow both, the enabled transition as well as the dynamics inside the location. Same holds if more than one guard is enabled at the same time.

c) If you should illustrate the evolution of the reachability analysis as a tree, what would be the nodes and what would be the edges? What are the reasons for a node to have several children?

Solution: Each node represents a time interval, the edges are either time steps or discrete steps in case a transition is enabled. This is also the reason for the branching: Either another time transition is taken, if applicable or one or more discrete transitions are enabled.