Modeling and Analysis of Hybrid Systems - SS 2013

First Exam

Monday, July 29, 2013

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.
- Please use a separate sheet for each task. If you need more sheets, indicate this by a hand signal.
- 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:	6	11	7	9	9	8	50
Reached score:							

Good luck!

Task 1. Hybrid System Modeling

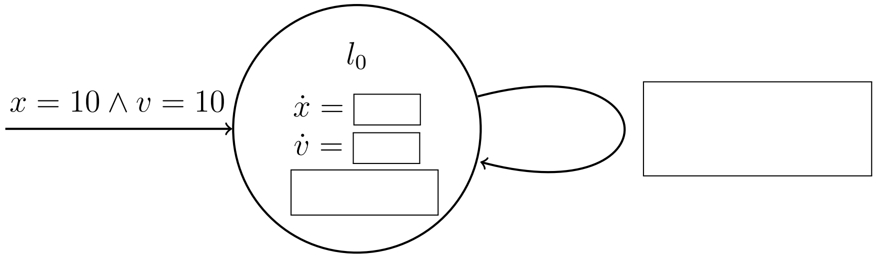
($3 + 2 + 1$ points)

Assume a bouncing ball with vertical movement. Let

- x $[m]$ denote the ball's height (distance from the ground),
- $v = \frac{dx}{dt}$ $[\frac{m}{s}]$ its velocity and
- $g = \frac{dv}{dt} = -9.8$ $[\frac{m}{s^2}]$ the acceleration due to gravity.

First, the ball raises with decreasing velocity until it starts to fall. When it hits the ground, it bounces and starts to raise again. We model the bouncing as a discrete event, inversing the sign of the velocity and reducing its absolute value by 50%.

(1) Please define the missing components of the following hybrid automaton to model the bouncing ball:

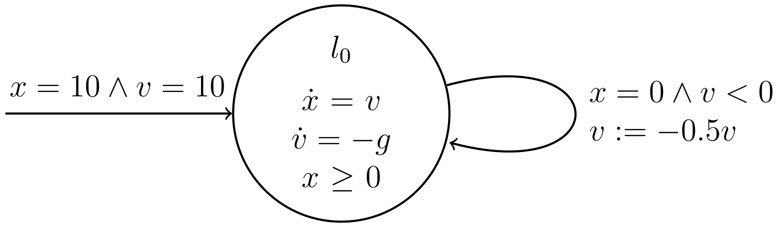


(2) Is the above automaton Zeno-free? Explain your answer!

(3) Is the above automaton a linear hybrid automaton? Justify your answer!

Solution:

(1) The missing components are as follows:



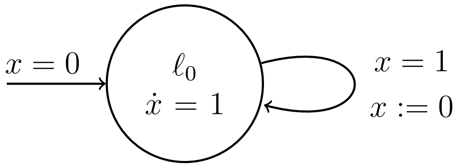
(2) No. Since the ball loses the half of its kinetic energy upon bouncing, the time between two successive bounces converges to 0 when time proceeds. Thus all paths of this automaton are time-convergent. Those paths that contain infinitely many discrete steps, for example the path having only time steps of maximal durations, are Zeno paths.

(3) Our model is not a linear hybrid automaton, because its behaviour is not linear: the derivative of x is not constant.

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

(1) Please define the operational semantics of timed automata by formalizing the rules for time evolution and discrete transitions.

(2) Assume the following timed automaton $\mathcal{T}$:



We want to check whether $\mathcal{T}$ satisfies the TCTL formula $EGEF^{\leq 1}(x = 1)$.

(i) How many abstract states are generated by the state space abstraction? Explain!

(ii) Which of the abstract states have a self-loop in the corresponding region transition system? Why?

Solution:

(1)

$$\frac{\begin{array}{c}(l, a, (g, \mathcal{R}), l') \in Edge \\ \nu \models g \quad \nu' = reset\ \mathcal{R}\ in\ \nu \quad \nu' \models Inv(l')\end{array}}{(l, \nu) \overset{a}{\to} (l', \nu')} \quad \texttt{Rule}_{\texttt{Discrete}}$$

$$\frac{t > 0 \quad \nu' = \nu + t \quad \nu' \models Inv(l)}{(l, \nu) \overset{t}{\to} (l, \nu')} \quad \texttt{Rule}_{\texttt{Time}}$$

(2) For a timed automaton $\mathcal{T} = (Loc, \mathcal{C}, Lab, Edge, Inv, Init)$ we define $ExecTime : (Lab \cup \mathbb{R}^{\geq 0}) \to \mathbb{R}^{\geq 0}$ with

- $ExecTime(a) = 0$ for $a \in Lab$ and
- $ExecTime(d) = d$ for $d \in \mathbb{R}^{\geq 0}$.

Furthermore, for a path $\rho = s_0 \overset{\alpha_0}{\to} s_1 \overset{\alpha_1}{\to} s_2 \overset{\alpha_2}{\to} \ldots$ of $\mathcal{T}$ we define

$$ExecTime(\rho) = \sum_{i=0}^{\infty} ExecTime(\alpha_i).$$

A path of $\mathcal{T}$ is time-divergent iff $ExecTime(\rho) = \infty$, and time-convergent otherwise.

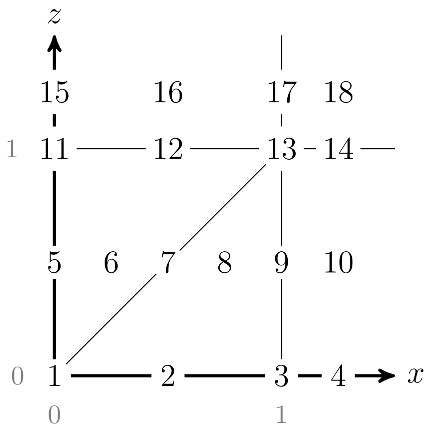
Time-convergent paths cannot be excluded by proper modeling, since for each time step of duration $t > 0$ there is an infinite time-convergent path consisting of time-steps of durations $\frac{t}{2}$, $\frac{t}{4}$, $\frac{t}{8}$, ….

(3) (i) By the transformation of the TCTL formula to a CTL formula a new clock z is introduced with $c_z = 1$ the maximal constant to which z is compared to in the CTL formula ($\mathcal{T}$ is only extended with the new clock z but z is not compared to any value in the extension). The largest constant to which x is compared to in the CTL formula or in the automaton is also $c_x = 1$.

Note that the automaton has a single location. Therefore, the abstraction defines two states (l_0, ν) and (l_0, ν') to be equivalent if

- either $\nu(x) > c_x \wedge \nu'(x) > c_x$ or
$$\lfloor \nu(x) \rfloor = \lfloor \nu'(x) \rfloor \ \wedge \ (\mathit{frac}(\nu(x)) = 0 \quad \mathit{iff} \quad \mathit{frac}(\nu'(x)) = 0)$$
- either $\nu(z) > c_z \wedge \nu'(z) > c_z$ or
$$\lfloor \nu(z) \rfloor = \lfloor \nu'(z) \rfloor \ \wedge \ (\mathit{frac}(\nu(z)) = 0 \quad \mathit{iff} \quad \mathit{frac}(\nu'(z)) = 0)$$
- if $\nu(x), \nu'(x) \leq c_x$ and $\nu(z), \nu'(z) \leq c_z$ then
$$\begin{aligned}
\mathit{frac}(\nu(x)) < \mathit{frac}(\nu(z)) &\quad \mathit{iff} \quad \mathit{frac}(\nu'(x)) < \mathit{frac}(\nu'(z)) \ , \\
\mathit{frac}(\nu(x)) > \mathit{frac}(\nu(z)) &\quad \mathit{iff} \quad \mathit{frac}(\nu'(x)) > \mathit{frac}(\nu'(z)) \ , \\
\mathit{frac}(\nu(x)) = \mathit{frac}(\nu(z)) &\quad \mathit{iff} \quad \mathit{frac}(\nu'(x)) = \mathit{frac}(\nu'(z)).
\end{aligned}$$

Therefore, the state space will define the following 18 abstract states:



$1 \ : x = 0 \wedge z = 0$
$2 \ : 0 < x < 1 \wedge z = 0$
$3 \ : x = 1 \wedge z = 0$
$4 \ : x > 1 \wedge z = 0$
$5 \ : x = 0 \wedge 0 < z < 1$
$6 \ : 0 < x < 1 \wedge 0 < z < 1 \wedge x < z$
$7 \ : 0 < x < 1 \wedge 0 < z < 1 \wedge x = z$
$8 \ : 0 < x < 1 \wedge 0 < z < 1 \wedge x > z$
$9 \ : x = 1 \wedge 0 < z < 1$
$10 : x > 1 \wedge 0 < z < 1$
$11 : x = 0 \wedge z = 1$
$12 : 0 < x < 1 \wedge z = 1$
$13 : x = 1 \wedge z = 1$
$14 : x > 1 \wedge z = 1$
$15 : x = 0 \wedge z > 1$
$16 : 0 < x < 1 \wedge z > 1$
$17 : x = 1 \wedge z > 1$
$18 : x > 1 \wedge z > 1$

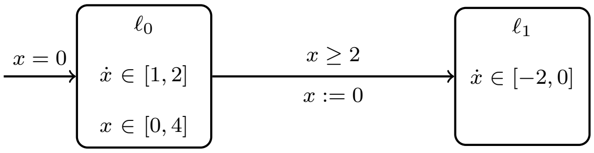
(ii) Only state 18 has a self-loop for two reasons: Firstly, the discrete transition changes the abstract state from one satisfying $x = 1$ to another one with $x = 0$, therefore there are no self-loops representing discrete steps (if there would be one than the system would be Zeno). Secondly, in order to avoid abstract paths that represent *only* time-convergent paths, no time-step-representing self-loops are added to the states 1-17. However, there is a self-loop on state 18 to represent infinite stay in the upper-unbounded region.

Task 3. Rectangular Automata (2 + 1 + 4 points)

(1) Please explain the differences between *rectangular automata* and *timed automata.*

(2) When is a rectangular automaton *initialized*?

(3) Please transform the following initialized rectangular automaton into an *initialized singular automaton*. You may skip irrelevant parts of the result like unreachable locations, invariant components that are satisfied by all states reachable in the given location, etc.

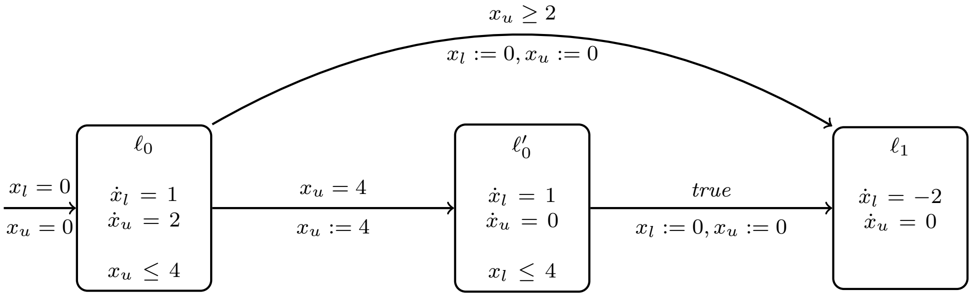


Solution:

(1) In a rectangular automaton, the derivative of a variable can be defined by an interval, however, all variable derivatives in a timed automaton should be 1. For a discrete transition, a rectangular automaton may reset a variable nondeterministically to a value from an interval, however, a timed automaton can only reset a variable to value 0.

(2) We call a rectangular automaton *initialized*, if for each discrete transition e and each variable x the following holds: if the derivative of x in the source location of e differs from the derivative of x in the target location of e, then x is reset by e.

(3)



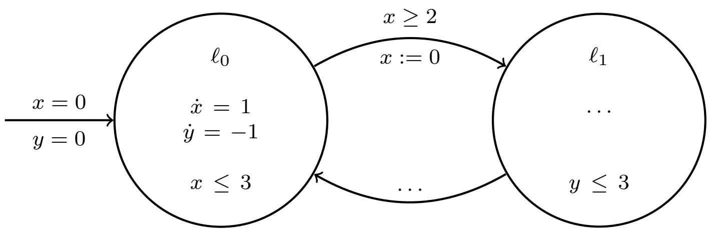
Task 4. Linear Hybrid Automata

($3 + 3 + 3$ points)

(1) Is the *bounded* reachability problem *decidable* for linear hybrid automata (with linear behavior)? Prove your answer!

(2) Which *state set representation* did we use for linear hybrid automata? How can the operations for *union*, *intersection*, *membership* and *test for emptiness* be computed for that representation?

(3) Assume the following linear hybrid automaton $\mathcal{A}$:



Let I be the representation of the initial state set $\{(l_0, \nu) \in \Sigma \mid \nu(x) = \nu(y) = 0\}$. Compute the *forward time closure* $\mathcal{T}^+_{l_0}(I)$ (or $\langle I \rangle^{\nearrow}_{l_0}$ in the notation of the lecture notes). Don't forget to reduce the result using quantifier elimination.

Solution:

(1) Yes, the bounded reachability problem is decidable on linear hybrid automata, because paths of bounded length can be encoded in linear real arithmetic, which is a decidable logic.

(2) Assume that the linear hybrid automaton $\mathcal{A}$ has N locations $\ell_1, \dots, \ell_N$. We may represent the a state set of $\mathcal{A}$ by N tuples $\langle \ell_1, \varphi_1 \rangle, \dots, \langle \ell_N, \varphi_N \rangle$ such that $\varphi_1, \dots, \varphi_N$ are linear real arithmetic formulas. For two state set representations $S_1 = \{\langle \ell_1, \varphi_1 \rangle, \dots, \langle \ell_N, \varphi_N \rangle\}$ and $S_2 = \{\langle \ell_1, \psi_1 \rangle, \dots, \langle \ell_N, \psi_N \rangle\}$, and a state $s = \langle \ell_i, \nu \rangle$ of $\mathcal{A}$, the operations can be computed as follows:

- $S_1 \cup S_2 = \{\langle \ell_1, \varphi_1 \vee \psi_1 \rangle, \dots, \langle \ell_N, \varphi_N \vee \psi_N \rangle\}$,
- $S_1 \cap S_2 = \{\langle \ell_1, \varphi_1 \wedge \psi_1 \rangle, \dots, \langle \ell_N, \varphi_N \wedge \psi_N \rangle\}$,
- $s \in S_1$ if and only if $\nu \models \varphi_i$, and
- $S_1 = \emptyset$ if and only if all φ_1 are unsatisfiable.

(3) We represent the initial set as $I = \langle \ell_0, x = 0 \wedge y = 0 \rangle$. Therefore

$$\begin{aligned} \mathcal{T}^+_{\ell_0}(I) &= \langle \ell_0, \exists x'. \exists y'. \exists t. (t \geq 0 \wedge x' = 0 \wedge y' = 0 \wedge x = x' + t \wedge y = y' - t \wedge x \leq 3) \rangle \\ &= \langle \ell_0, x + y = 0 \wedge x \geq 0 \wedge x \leq 3 \rangle \ . \end{aligned}$$

Task 5. Reachability Analysis (5 + 4 points)

(1) Please complete the following table with the information whether for the given subclasses of hybrid automata the reachability and bounded reachability problems are decidable or not!

Automata subclass	Is the reachability problem decidable?	Is the bounded reachability problem decidable?
Timed automata		
Initialized rectangular automata		
Rectangular automata		
Linear hybrid automata		
General hybrid automata		

(2) Please specify in pseudo-code the general (i.e., representation-independent) algorithm for *forward reachability* computation (i.e., to compute the set of states reachable from a given initial state set). Use I to represent the set of initial states, $\mathit{Reach}(R)$ to represent the set of states reachable in one step from R, and the notations for standard set operations.

Solution:

(1)

Automata subclass	Is the reachability problem decidable?	Is the bounded reachability problem decidable?
Timed automata	Yes	Yes
Initialized rectangular automata	Yes	Yes
Rectangular automata	No	Yes
Linear hybrid automata	No	Yes
General hybrid automata	No	No

(2)

```
Input: the initial state set $I$.
Algorithm:
    $R^{\mathsf{new}} := I;$
    $R := \emptyset;$
    while $(R^{\mathsf{new}} \neq \emptyset)\{$
        $R \quad := R \cup R^{\mathsf{new}};$
        $R^{\mathsf{new}} := \mathsf{Reach}(R^{\mathsf{new}}) \backslash R;$
    $\}$
Output: the reachable state set $R$
```

Task 6. Convex Polytopes **($1+2+2+3$ points)**

(1) What is the difference between *polyhedra* and *polytopes*?

(2) Please describe the *two representations* that we discussed in the lecture for *polytopes.*

(3) How can we compute the *convex hull of the union* of two polytopes in those representations?

(4) Using polytopes to represent state sets, in the approximation of a flow pipe segment we used *bloating.* What is it and what do we need it for?

Solution:

(1) Polyhedra can be unbounded. Polytopes are bounded polyhedra.

(2) Polytopes can be represented in two ways.

- $\mathcal{V}$-polytopes - A polytope P is represented by the convex hull of finitely many points.
- $\mathcal{H}$-polytopes - A polytope P is represented by an intersection of finitely many half-spaces.

(3) If both of the polytopes are $\mathcal{V}$-polytopes, say $P_1 : \{v_1, \ldots, v_n\}$ and $P_2 : \{u_1, \ldots, u_m\}$, then the convex hull of their union can be represented by the $\mathcal{V}$-polytope $conv(P_1 \cup P_2) : \{v_1, \ldots, v_n, u_1, \ldots, u_m\}$. If at least one of the polytopes is not a $\mathcal{V}$-polytope, we may converse it into a $\mathcal{V}$-polytope and use the previous method to compute their convex hull.

(4) To compute a polytope over-approximation of a flow pipe segment, say from time t_1 to t_2, we first compute the reachable sets R_1 and R_2 at time t_1 and t_2 respectively, and compute a convex hull of $R_1 \cup R_2$ which is a polytope P. However, the convex hull P does not include some non-linear trajectories from R_1 to R_2, therefore we need to bloat P to P^+ such that all trajectories are included in P^+.