Automata-theoretic analysis of hybrid systems

Madhavan Mukund

SPIC Mathematical Institute
92, G N Chetty Road
Chennai 600 017, India

Email: `madhavan@smi.ernet.in`
URL: `http://www.smi.ernet.in/~madhavan`

Tutorial at BRNS Workshop on Verification of Digital and Hybrid Systems, January 7–11, 1999, TIFR, Mumbai, India.

What is a hybrid system?

- Digital system which reads and reacts to analog environmental parameters such as time, position, temperature . . .

- *Examples:*

  - Controllers for cars, aircraft, manufacturing plants

  - Medical equipment

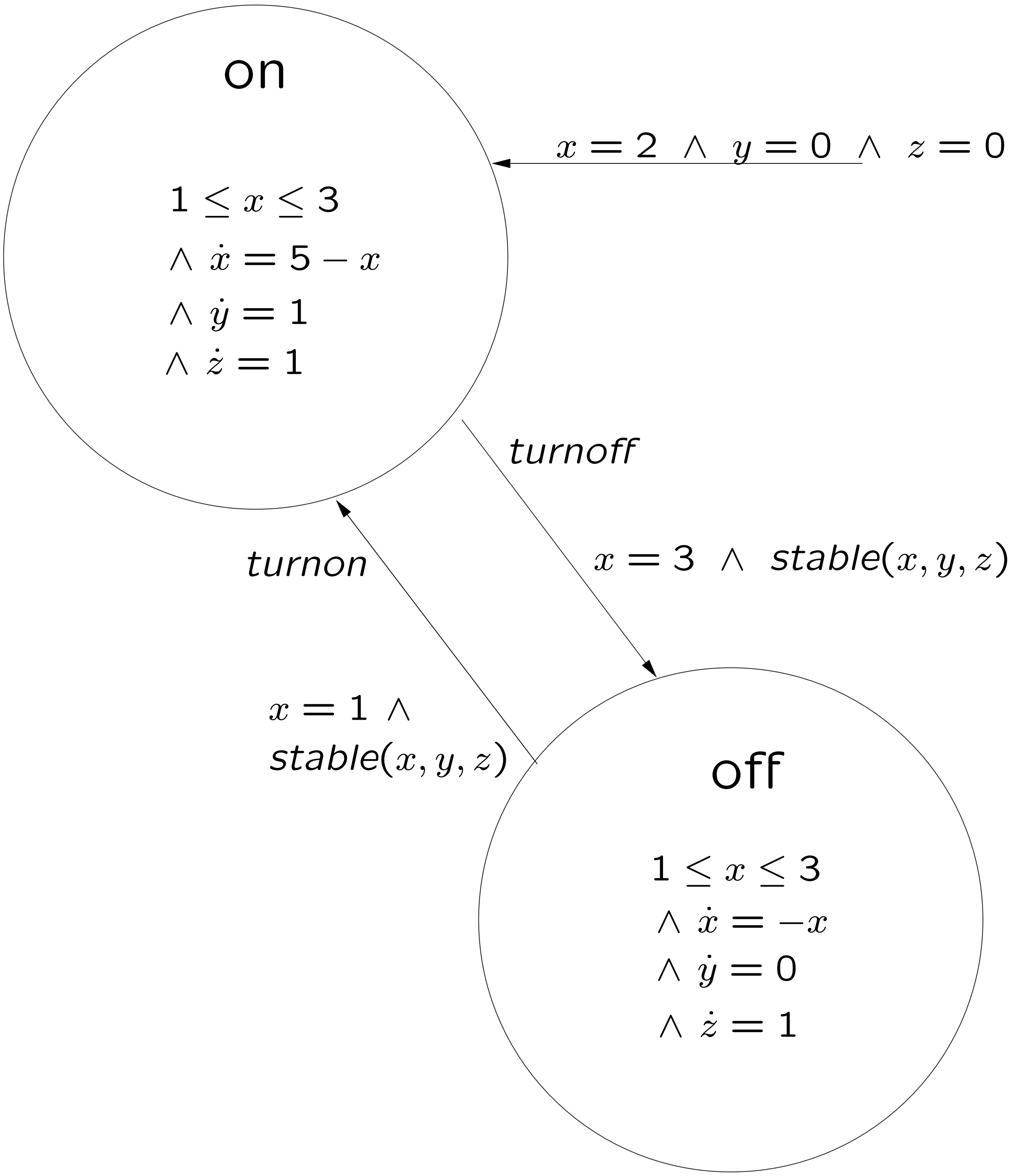
  - Robots

- Extension of finite-state automata with analog inputs— *hybrid automata* .

Example: A temperature controller (thermostat)

- Heater may be *off* or *on*.
- If heater is *off*, temperature drops exponentially — $T(t) = T_{init}\ e^{-kt}$
- If heater is *on*, temperature rises exponentially —
$$T(t) = T_{init}\ e^{-kt} + h(1 - e^{-kt})$$
- Heater switches between on and off when temperature crosses threshold values.

Typical question:
Show that heater is on for less than 50% of the first 60 units of time.



A thermostat

Hybrid automata

A hybrid automaton consists of:

- A finite set V of control modes — i.e., states, in the sense of automata theory. In the example, $V = \{\text{on}, \text{off}\}$.

- A finite set E of control switches — i.e., transitions, in the sense of automata theory. In the example, $E = \{(\text{on}, \text{off}), (\text{off}, \text{on})\}$.

  (V, E) defines a directed graph, as usual.

- A set X of variables taking values over $\mathbb{R}$. In the example, $X = \{x, y, z\}$.

  For each variable x, $\dot{x}$ denotes the first derivative of x with respect to time. This is called the *flow* of x.

Labels on control modes:

- Control modes labelled by initial condition *init*(v) and flow condition *flow*(v) — predicates over $X \cup \dot{X}$. In the example:
  - *init*(on) : $x = 2 \ \wedge \ y = 0 \ \wedge \ z = 0$
  - *flow*(on) : $1 \leq x \leq 3 \ \wedge \ \dot{x} = 5 - x \ \wedge \ \dot{y} = 1 \ \wedge \ \dot{z} = 1$

- Initial conditions marked on incoming arcs with no source state. Initial condition *false* is not marked — for instance, *init*(off).

- Flow condition *flow*(v) constrains flows in the control mode v — for instance, $\dot{x} = 5 - x$.

- Flow conditions implicitly include invariants — for instance, $1 \leq x \leq 3$.

Labels on control switches:

- Control switches (v, v') labelled by jump condition *jump*(v, v') — predicate over $X, X', \dot{X}, \dot{X}'$.

  Jump condition relates values of variables before and after the transition — x' and $\dot{x}'$ denote values of x and $\dot{x}$ after the transition.

  Example:

  $$\textit{jump}(\text{on}, \text{off}) : x = 3 \ \wedge \ \textit{stable}(x, y, z)$$

  where *stable*(x) abbreviates $x' = x$.

- Control switches also labelled by events — used for synchronization of parallel components.

  Example: (off, on) is labelled by the event *turnon*.

Special types of variables

- A clock is a variable with constant flow 1, which is either stable or reset to 0 on each control switch.

  In the thermostat automaton, z is a clock.

- A stopwatch is a variable which can have flows 0 or 1, which is either stable or reset to 0 on each control switch.

  In the thermostat automaton, y is a stopwatch which measures how much time the system spends in control mode *on*.

- Show that heater is on for less than 50% of the first 60 units of time.

  is equivalent to proving that

$$(z = 60) \text{ implies } y \leq z/2$$

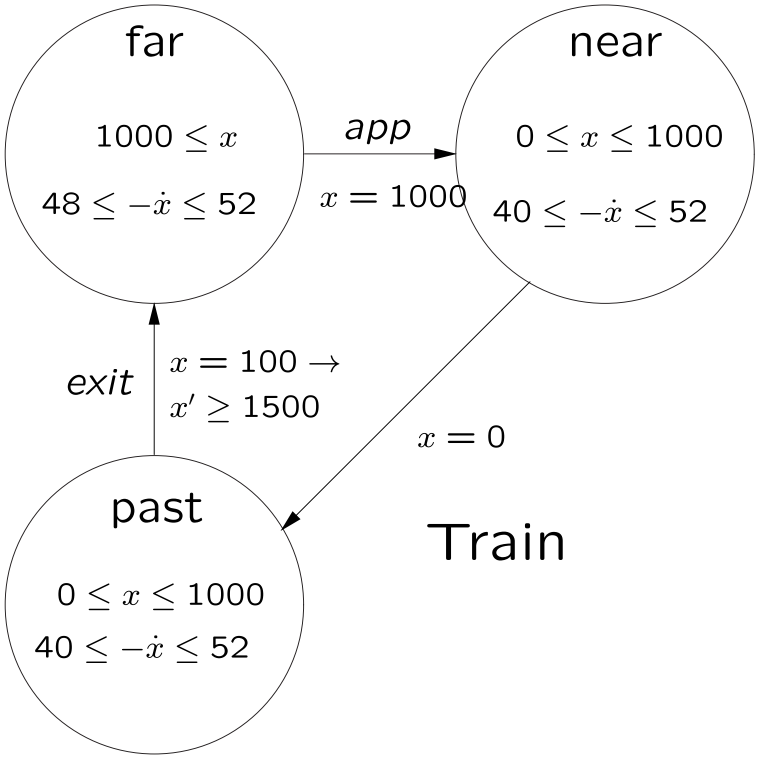
Controller for a railway level crossing

When the train is far from the gate it moves at 48 to 52 m/s. At 1000 m from the gate is a sensor. After passing the sensor, the train slows down to 40 to 52 m/s.

After sensing the train, the controller requires upto 5 secs to start lowering the gate. The gate moves at 20 deg/s.

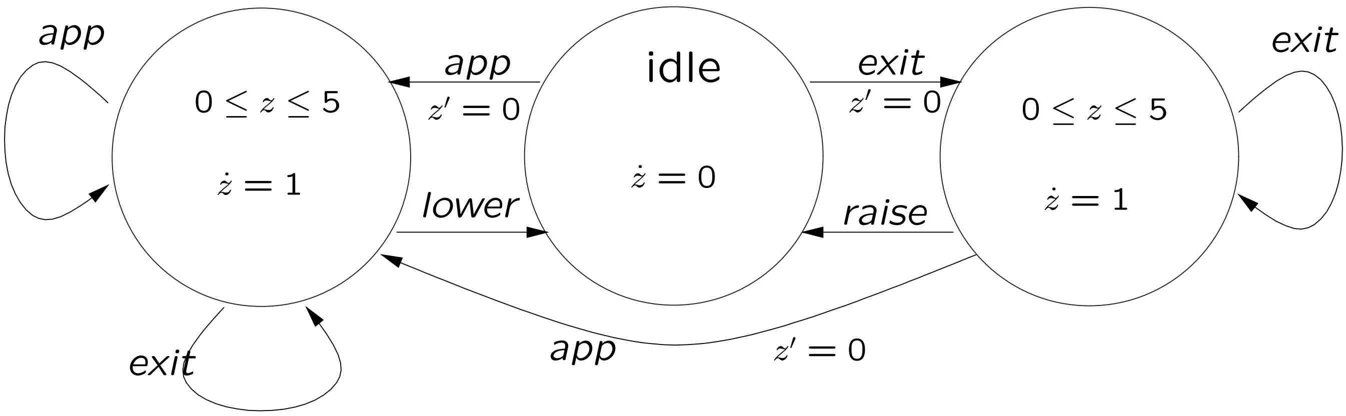
At 100 m past the gate, there is a second sensor. Once the train passes this sensor, the controller requires upto 5 secs to start raising the gate. The gate again moves at 20 deg/s.

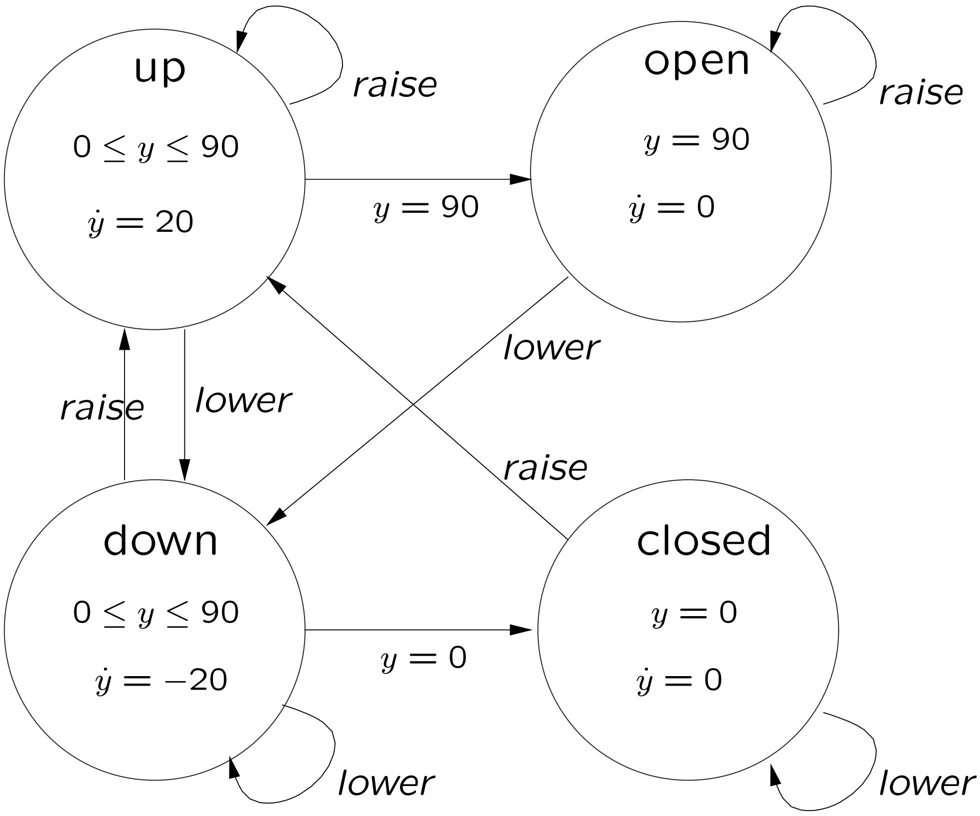
Consecutive trains are at least 1500 m apart.



Train

Controller




up
$0 \le y \le 90$
$\dot{y} = 20$
raise
open
$y = 90$
$\dot{y} = 0$
raise
$y = 90$
lower
lower
raise
raise
down
$0 \le y \le 90$
$\dot{y} = -20$
closed
$y = 0$
$\dot{y} = 0$
$y = 0$
lower
lower

Gate

Configurations

- A *configuration* is a triple $(v, \mathbf{a}, \dot{\mathbf{a}})$ where $\mathbf{a}$ is a point in $\mathbb{R}^n$ and $\dot{\mathbf{a}}$ is a vector of trajectories, also in $\mathbb{R}^n$.

- Let φ be a predicate over $X \cup \dot{X}$. The models of φ, $[\![\varphi]\!]$, is defined as:

  $$[\![\varphi]\!] = \{\langle \mathbf{a}, \dot{\mathbf{a}} \rangle \mid \varphi \text{ is true when } X \leftarrow \mathbf{a}, \dot{X} \leftarrow \dot{\mathbf{a}}\}.$$

- The configuration $(v, \mathbf{a}, \dot{\mathbf{a}})$ is *admissible* if $\langle \mathbf{a}, \dot{\mathbf{a}} \rangle$ belongs to $[\![\mathit{flow}(v)]\!]$.

- The configuration $(v, \mathbf{a}, \dot{\mathbf{a}})$ is *initial* if $\langle \mathbf{a}, \dot{\mathbf{a}} \rangle$ belongs to $[\![\mathit{init}(v)]\!]$.

Timed Transition Systems

$TTS = (Q, Q^i, \Sigma, \longrightarrow)$

- Q a set of states with initial states $Q^i \subseteq Q$.

- Set of actions Σ, includes silent action τ.

- Labelled transition relation

  $$\longrightarrow \subseteq Q \times (\Sigma \cup \mathbb{R}_{\geq 0}) \times Q.$$

  Jump transition: $q \xrightarrow{a} q'$, $a \in \Sigma$.

  If $a = \tau$, the transition is *silent*.

  Flow transition: $q \xrightarrow{\delta} q$, $\delta \in \mathbb{R}_{\geq 0}$.

Hybrid automaton A $\Longrightarrow$ *Timed transition system* $TTS_A = (Q, Q^i, \Sigma, \longrightarrow)$

Q : admissible configurations of A

Q^i : initial configurations of A

Σ : events of A

$\longrightarrow$: moves of the following form:

Jump: $(v, \mathbf{a}, \dot{\mathbf{a}}) \xrightarrow{\sigma} (v', \mathbf{a}', \dot{\mathbf{a}}')$

– σ is the event label on edge (v, v')

– $\langle \mathbf{a}, \dot{\mathbf{a}}, \mathbf{a}', \dot{\mathbf{a}}' \rangle$ belongs to $[\![\mathit{jump}(v, v')]\!]$

Flow: $(v, \mathbf{a}, \dot{\mathbf{a}}) \xrightarrow{\delta} (v, \mathbf{a}', \dot{\mathbf{a}}')$

– $\delta = 0$, $\mathbf{a} = \mathbf{a}'$ and $\dot{\mathbf{a}} = \dot{\mathbf{a}}'$,

or

– there exists $f : [0, \delta] \to \mathbb{R}^n$,
f is continuously differentiable,
$\langle f(0), \dot{f}(0) \rangle = \langle \mathbf{a}, \dot{\mathbf{a}} \rangle$,
$\langle f(\delta), \dot{f}(\delta) \rangle = \langle \mathbf{a}', \dot{\mathbf{a}}' \rangle$,
and $\langle f(t), \dot{f}(t) \rangle$ in $[\![\mathit{flow}(v)]\!]$ for all $t \in [0, \delta]$.

Reachability

- A trajectory of automaton A is a finite path $s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} \cdots \xrightarrow{a_{n-1}} s_n$ in TTS_A, where s_0 is an initial state and each move is permitted by $\longrightarrow$.

  State s is *reachable* if there is a trajectory from an initial state which ends in s.

  Question: Given an automaton A and a state s, is s reachable in A?

Non-emptiness: Infinite behaviours

- An infinite path $s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} \cdots$ in TTS_A *diverges* if the time elapsed in flow transitions tends to ∞.

  Question: Given an automaton A, does TTS_A admit at least one divergent infinite path?

Reachability and non-emptiness are decidable for very restricted classes of hybrid systems.

A timed automaton is a hybrid system where

- Every variable is a clock.

- Every jump condition is *simple* — comparison of variables to constants or the difference of two variables to a constant.

  For example, $x \leq 5 \ \wedge \ y - z \geq 3 \ \wedge \ x' = 7$.

Theorem *Reachability and non-emptiness are decidable (PSPACE-complete) for timed automata.*

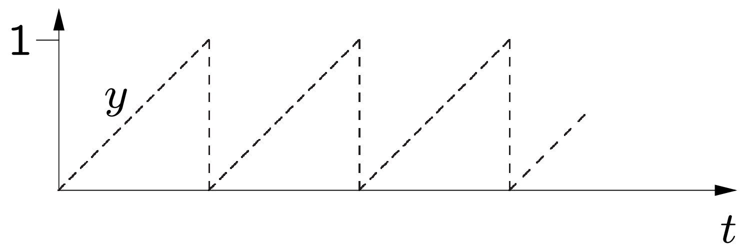
A multirate timed system extends timed automata with variables with arbitrary constant slope.

Theorem *Reachability is undecidable for 2-rate timed systems.*

Proof Reduction of halting problem for nondeterministic 2-counter machines.

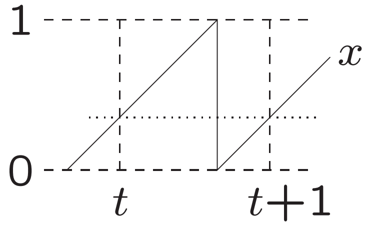
Use *accurate clocks* with slope 1 and *skewed clocks* with slope 2.

Use an accurate clock y to mark off time segments of unit length.

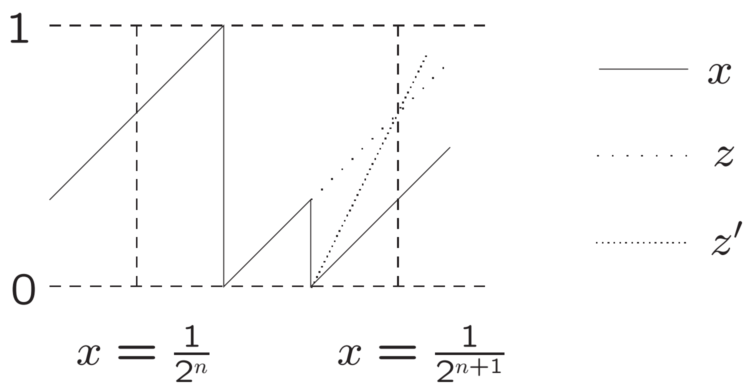


Counter value $n \Leftrightarrow$ Accurate clock value $x = \frac{1}{2^n}$

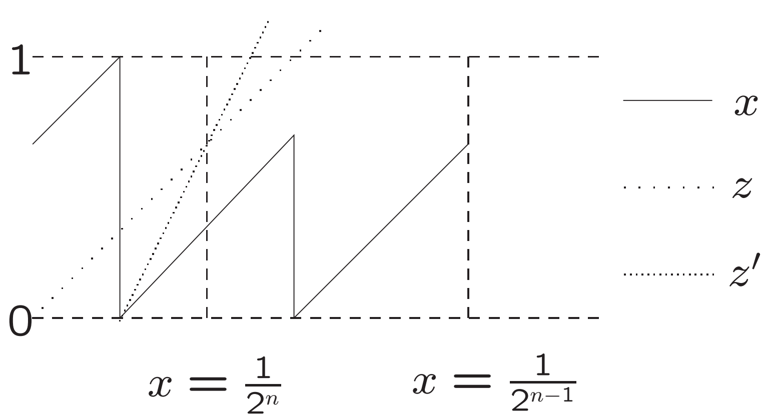
To reproduce $x(t)$ at $x(t+1)$, reset when $x = 1$.



To increment x:



To decrement x:



Rectangular automata

- $\dot{x}$ can vary within a range $[\mathit{min}, \mathit{max}]$. Can model *drifting clocks*.

- Values of variables with different flows are never compared.

- Whenever the flow constraint of a variable changes, the variable is reset.

Theorem *Reachability is decidable for rectangular automata.*

Theorem *Reachability is undecidable if either the second or the third constraint is violated.*

Linear hybrid automata

- A *linear predicate* over X built out of atomic predicates of the form $\Sigma_i a_i x_i$ *op* c, where *op* is a relational operator.

  If all the a_i's are rational, this is called a *rational linear predicate*.

- In a *linear hybrid automaton*, all initial, jump and flow conditions are written using linear predicates such that variables from X and $\dot{X}$ never appear together in an atomic predicate.

  For instance, $x + 2\dot{y} \leq 7$ or $x = -\dot{x}$ is not allowed, but $x \leq 7 \wedge 3\dot{x} + 2\dot{y} = 8$ is allowed.

Linear regions

- A *region* is a set of configurations of A.

- A region R is *linear* if there is a linear predicate φ_v for each control mode v such that $R = \bigcup_{v \in V} \{v\} \times [\![\varphi_v]\!]$.

  Example: Let A be a linear hybrid automaton and let TTS_A be its timed transition system. Then, Q, Q^i are linear regions.

- Let R be a region.

  post$(R) = \{s_2 \mid \exists s_1 \in R . s_1 \longrightarrow s_2\}$.

  pre$(R) = \{s_1 \mid \exists s_2 \in R . s_1 \longrightarrow s_2\}$.

Theorem *Let A be a linear automaton and R a linear region of A. Then, post(R) and pre(R) are also linear regions of A.*

Moreover, if all conditions used to define A and R are rational *linear predicates, then the rational linear predicates for post(R) and pre(R) can be effectively constructed from the predicate for R.*

This gives a semi-decision procedure for reachability in (rational) linear hybrid automata.

Every reachable state can be obtained from Q^i (which is a rational linear region), by taking *post*$^j(Q^i)$ for sufficiently large j.

Handling non-linearity

Replace non-linear system by *equivalent* linear system. Equivalence is defined in terms of timed bisimulation.

Stutter closure

Let $TTS = \langle Q, Q^i, \Sigma, \longrightarrow \rangle$ be a timed transition system. The *stutter closure* of $\longrightarrow$ is given as follows.

For $\sigma \in \Sigma$, $q \stackrel{\sigma}{\Longrightarrow} q'$ if there is a sequence of the form $q \stackrel{\tau}{\longrightarrow}^* q_1 \stackrel{\sigma}{\longrightarrow} q'$.

For $\delta \in \mathbb{R}_{\geq 0}$, $q \stackrel{\delta}{\Longrightarrow} q'$ if there is a sequence of the form $q \stackrel{\tau}{\longrightarrow} q_1 \stackrel{\delta_1}{\longrightarrow} r_1 \stackrel{\tau}{\longrightarrow} \cdots \stackrel{\delta_n}{\longrightarrow} q'$ such that $\Sigma_i \delta_i = \delta$n.

References:

- R. Alur, C. Courbetis, N. Halbwachs, T.A. Henzinger,
  P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, S. Yovine:
  The algorithmic analysis of hybrid systems,
  TCS 138 (1995) 3–34.

- R. Alur, T.A. Henzinger, P.-H. Ho:
  Automatic symbolic verification of embedded systems,
  IEEE Trans Software Engg 22(3) (1996) 181–201.

- T.A. Henzinger:
  The theory of hybrid automata,
  Proc 11th LICS (1996) 278–292.

- T.A. Henzinger, P.-H. Ho, H. Wong-Toi:
  Algorithmic analysis of nonlinear hybrid systems,
  IEEE Trans Automatic Control 43(4) (1998) 540–554.

- T.A. Henzinger, P.W. Kopke, A. Puri, P Variaya:
  What's decidable about hybrid automata?
  JCSS 57 (1998) 94–124.