Lecture 19: Timed automata and its reachability

Huan Zhang

huan@huan-zhang.com

Slides adapted from Prof. Sayan Mitra’s slides in Fall 2021

Timed Automata & Reachability

- We have studied hybrid automaton

automaton Bouncingball(c,h,g)

variables: x: Reals := h, v: Reals := 0

actions: bounce

transitions:

bounce

pre $x = 0 \wedge v < 0$

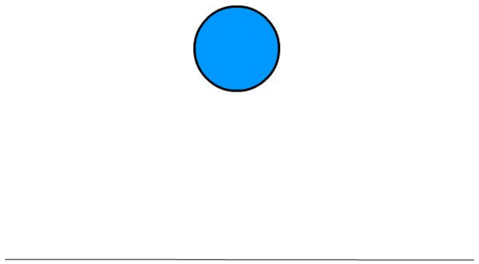
eff v := -cv

trajectories:

freefall

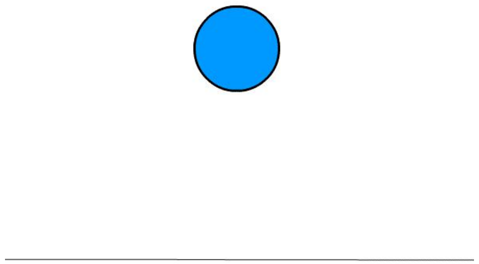
evolve d(x) = v; d(v) = -g

invariant $\boldsymbol{x \geq 0}$



Timed Automata & Reachability

- We have studied hybrid automaton
- However, verification for general hyrbid automaton is in general difficult



- Special classes of hybrid automaton:
  - (Alur-Dill’s) Timed Automata
  - Rectangular initialized hybrid automata
  - Linear hybrid automata
- Verification is feasible for these classes
  - Today and next a few lectures

Clocks and Clock Constraints

- A **clock variable** x is a continuous (analog) variable of type real such that along any trajectory τ of x, for all t ∈ $\tau.dom,\ (\tau \downarrow x)(t) = t$.
- In other words, d(x) = 1

- For a set X of clock variables, the set Φ(X) of **integral clock constraints** are expressions defined by the syntax:

  g ::= x $\leq q \mid x \geq q \mid \neg\, g \ \mid g_1 \wedge \ g_2$

  where $x \in X\ and\ q \in \ \mathbb{Z}$

- Examples: x = 10; x ∈ [2, 5] are valid clock constraints
- What do clock constraints look like?

Example: “smart” light switch

automaton Switch
variables x, y:Real := 0, loc: {on,off} := off

transitions
push
pre x ≥ 2
eff if loc = off **then** x,y := 0; loc := on
else x := 0
pop
pre y = 15 /\ loc = on
eff x := 0; loc = off

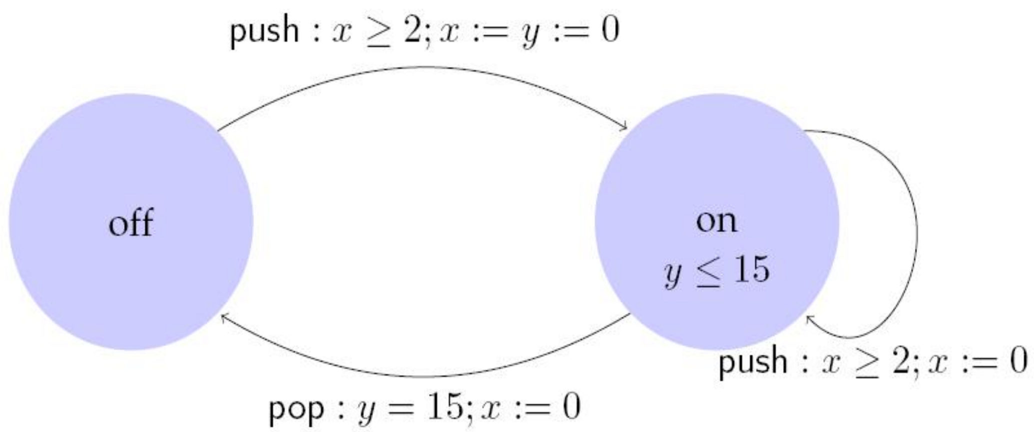
trajectories
invariant loc = off \/ y ≤ 15
evolve d(x) = 1; d(y) = 1

integral clock constraints

clock guard

clock variables

Description
Switch can be turned on whenever at least 2 time units have elapsed since the last turn off. Switches can be turned off 15 time units after the last on.



Integral Timed Automata (ITA)

- **Definition.** A **integral timed automaton** $\mathcal{A} = \langle V,\ \Theta,\ A,\ \mathcal{D},\ \mathcal{T} \rangle$ where
  - V = X ∪ $\{l\}$, where X is a set of n clocks and l is a discrete state variable of finite type L**.** The **stata space is** val(V) × L
  - A is a finite set
  - $\mathcal{D}$ is a set of transitions such that
    - The preconditions are described by clock constrainqs $\Phi(X)$
    - $\langle x, l \rangle_a \rightarrow \langle x', l' \rangle$ implies either $x' = x$ or $x' = 0$ (time is reset to 0, or no change)
  - $\mathcal{T}$ set of clock trajectories for the clock variables in X

Control State (mode) Reachability Problem

- Given an ITA $\mathcal{A}$, check if a particular (mode) control state $l^* \in L$ is **reachable** from the initial states

- cannot just enumerate all states - uncontable many states!

Control State (mode) Reachability Problem

- Given an ITA $\mathcal{A}$, check if a particular (mode) control state $l^* \in L$ is **reachable** from the initial states
  - How many states in $\mathcal{A}$?

Model Reachability of Integral Timed Automata is Decidable [Alur Dill 94]

That is, there is an algorithm that takes in $\mathcal{A}$, l^* and terminates with the correct answer.

Key idea:

- Construct a finite automaton B that is a ***time-abstract bisimilar*** to the given ITA $\mathcal{A}$
- That is, FA B behaves identically to ITA $\mathcal{A}$ w.r.t. control state reachability, but does not preserve timing information
- Check reachability of FA B

A theory of timed automata

R **Alur**, DL **Dill** - Theoretical computer science, 1994 - Elsevier

We propose timed (finite) automata to model the behavior of real-time systems over time. Our definition provides a simple, and yet powerful, way to annotate state-transition graphs with timing constraints using finitely many real-valued clocks. A timed automaton accepts …

An equivalence relation with a finite quotient

Under what conditions do two states $\mathbf{x_1}$ and $\mathbf{x_2}$ of the automaton $\mathcal{A}$ behave identically with respect to control state reachability (CSR)?

When do they satisfy the same set of clock constraints?

When would they continue to satisfy the same set of clock constraints?

Goal: infinite number of states -> finite number of states (possible because some states are **identical**)

An equivalence relation with a finite quotient

Under what conditions do two states $\mathbf{x_1}$ and $\mathbf{x_2}$ of the automaton $\mathcal{A}$ behave identically with respect to mode reachability?

$\mathbf{x_1}.loc = \mathbf{x_2}.loc$ and

$\mathbf{x_1}$ and $\mathbf{x_2}$ satisfy the same set of clock constraints:

- For all clock y: int($\mathbf{x_1}.y$) = int($\mathbf{x_2}.y$), or int($\mathbf{x_1}.y$) $\geq c_{\mathcal{A}y}$ and int($\mathbf{x_2}.y$) $\geq c_{\mathcal{A}y}$. ($c_{\mathcal{A}y}$ is the maxium clock guard of y)
- For all clock y: with $\mathbf{x_1}.y \leq c_{\mathcal{A}y}$, frac($\mathbf{x_1}.y$) $= 0$ iff frac($\mathbf{x_2}.y$) $= 0$
- For any two clocks y and z: with $\mathbf{x_1}.y \leq c_{\mathcal{A}y}$ and $\mathbf{x_1}.z \leq c_{\mathcal{A}z}$, frac($\mathbf{x_1}.y$) $\leq$ frac($\mathbf{x_1}.z$) iff frac($\mathbf{x_2}.y$) $\leq$ frac($\mathbf{x_2}.z$)

Lemma. This is an **equivalence relation** on *val(V)* (the states of $\mathcal{A}$)

The **partition** of ***val(V)*** induced by this relation is are called **clock regions**

An equivalence relation with a finite quotient

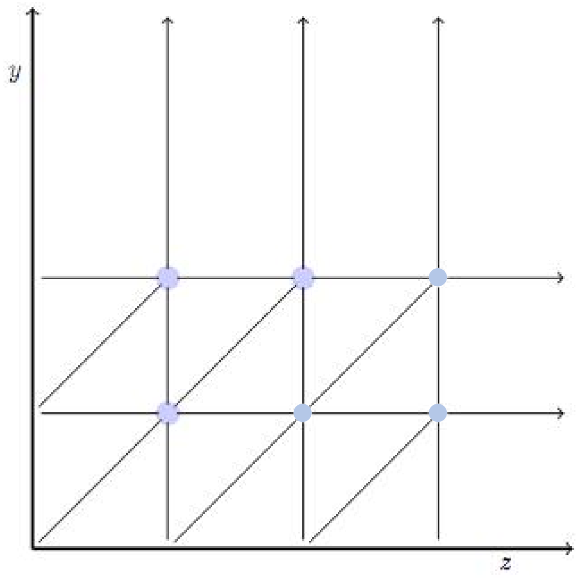
For all clock y: int($\mathbf{x_1}.y$) = int($\mathbf{x_2}.y$) or int($\mathbf{x_1}.y$) $\geq c_{\mathcal{A}y}$ and int($\mathbf{x_2}.y$) $\geq c_{\mathcal{A}y}$. ($c_{\mathcal{A}y}$ is the maxium clock guard of y)

For all clock y: with $\mathbf{x_1}.y \leq c_{\mathcal{A}y}$, frac($\mathbf{x_1}.y$) $= 0$ iff frac($\mathbf{x_2}.y$) $= 0$

For any two clocks y and z: with $\mathbf{x_1}.y \leq c_{\mathcal{A}y}$ and $\mathbf{x_1}.z \leq c_{\mathcal{A}z}$, frac($\mathbf{x_1}.y$) $\leq$ frac($\mathbf{x_1}.z$) iff frac($\mathbf{x_2}.y$) $\leq$ frac($\mathbf{x_2}.z$)

Example of
Two Clocks

X = {y,z}
$c_{\mathcal{A}y}$ = 2
$c_{\mathcal{A}z}$ = 3



Complexity

Lemma. The number of clock regions is bounded by $|L||X|!\, 2^{|X|} \prod_{z \in X} (2c_{\mathcal{A}z} + 2)$.

X is a set of clocks

L is the type of discrete states

$c_{\mathcal{A}z}$ is the clock guard

Region automaton $\mathrm{R}(\mathcal{A})$

Given an ITA $\mathcal{A} = \langle V,\ \Theta,\ \mathcal{D},\ \mathcal{T}\rangle$, we construct the corresponding **Region Automaton** $\mathrm{R}(\mathcal{A}) = \langle Q_R, \Theta_R, D_R\rangle$.

(i) $\mathrm{R}(\mathcal{A})$ visits the same set of modes (but does not have timing information) and

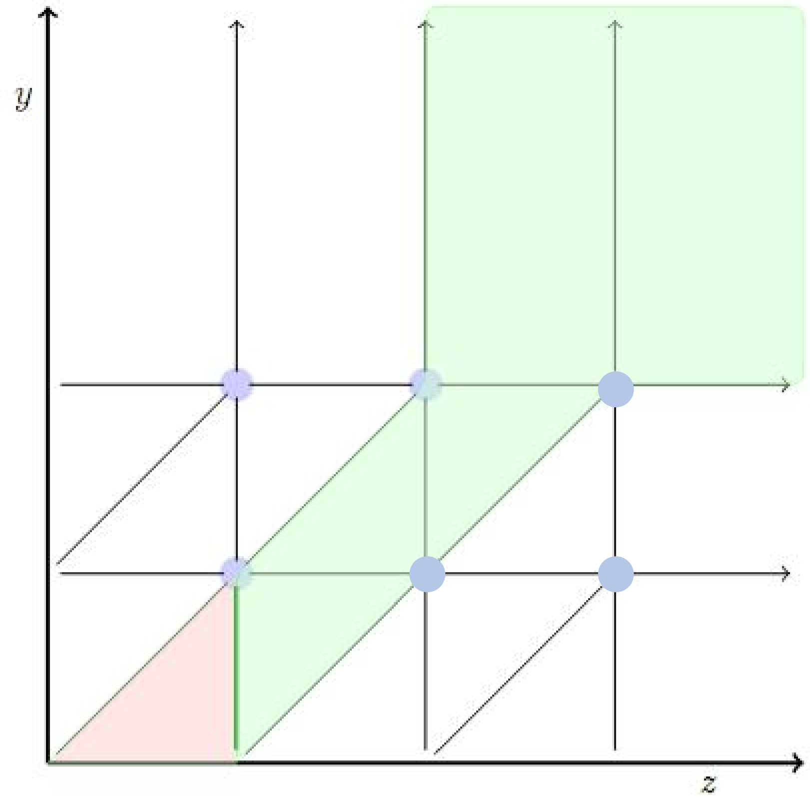
(ii) $\mathrm{R}(\mathcal{A})$ is finite state machine.

- ITA (clock constants) defines a set of clock regions, say $\mathrm{C}_{\mathcal{A}}$. The set of states $Q_R = C_{\mathcal{A}} \times L$
- $Q_0 \subseteq Q$is the set of states contain initial set Θ of $\mathcal{A}$
- D: We add the transitions between Q (regions)
  - **Time successors**: Consider two clock regions γ and γ', we say that γ' is a time successor of γ if there exits a trajectory of ITA starting from γ that ends in γ'
  - **Discrete transitions**: Same as the ITA

Theorem. A mode of ITA $\mathcal{A}$ is reachable iff it is also reachable in $\mathrm{R}(\mathcal{A})$.

(we say that $\mathrm{R}(\mathcal{A})$ is *time abstract bisimilar* to $\mathcal{A}$)

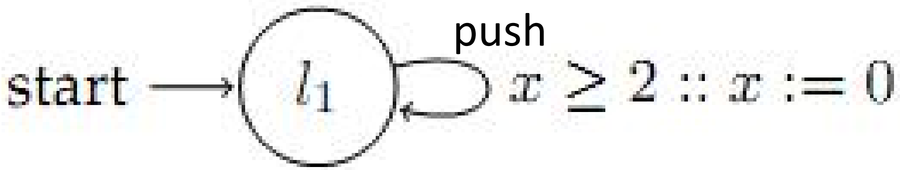
Time successors



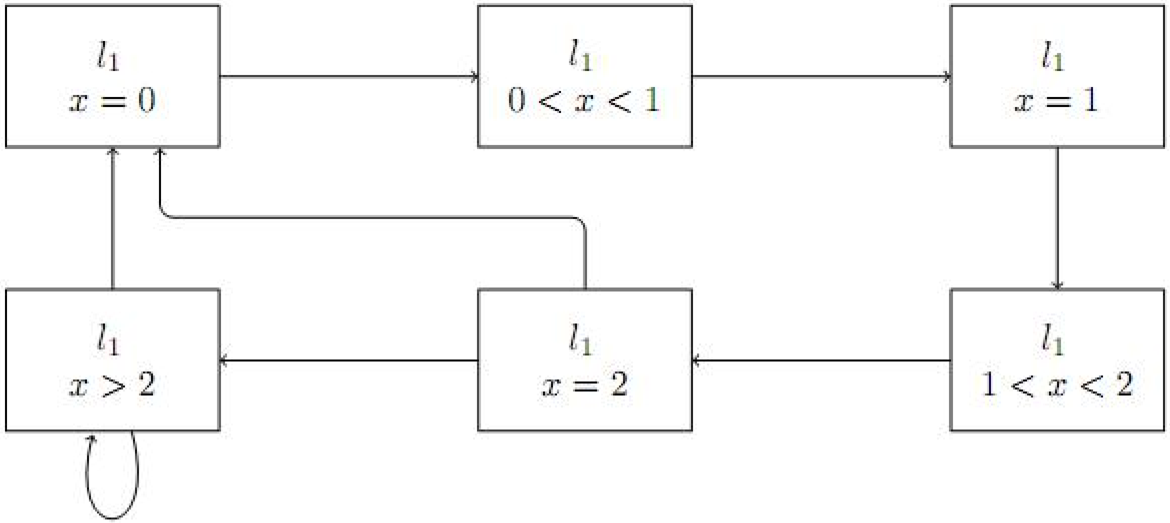
The clock regions in green are time successors of the clock region in red.

Example 1: Region Automata

ITA

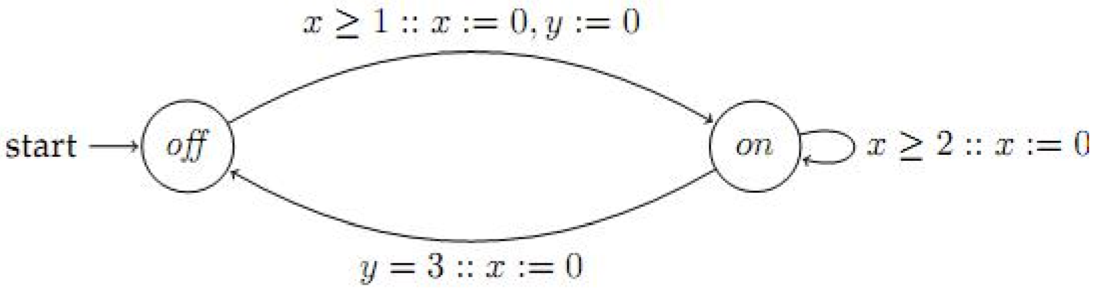


Corresponding FA

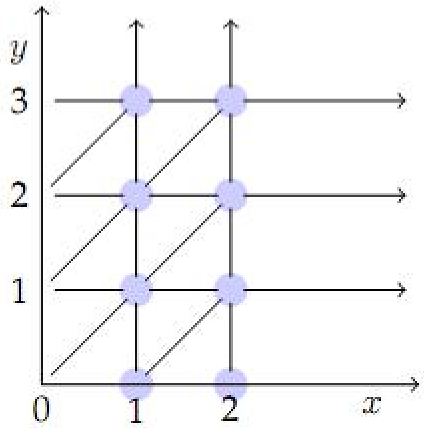


Example 2

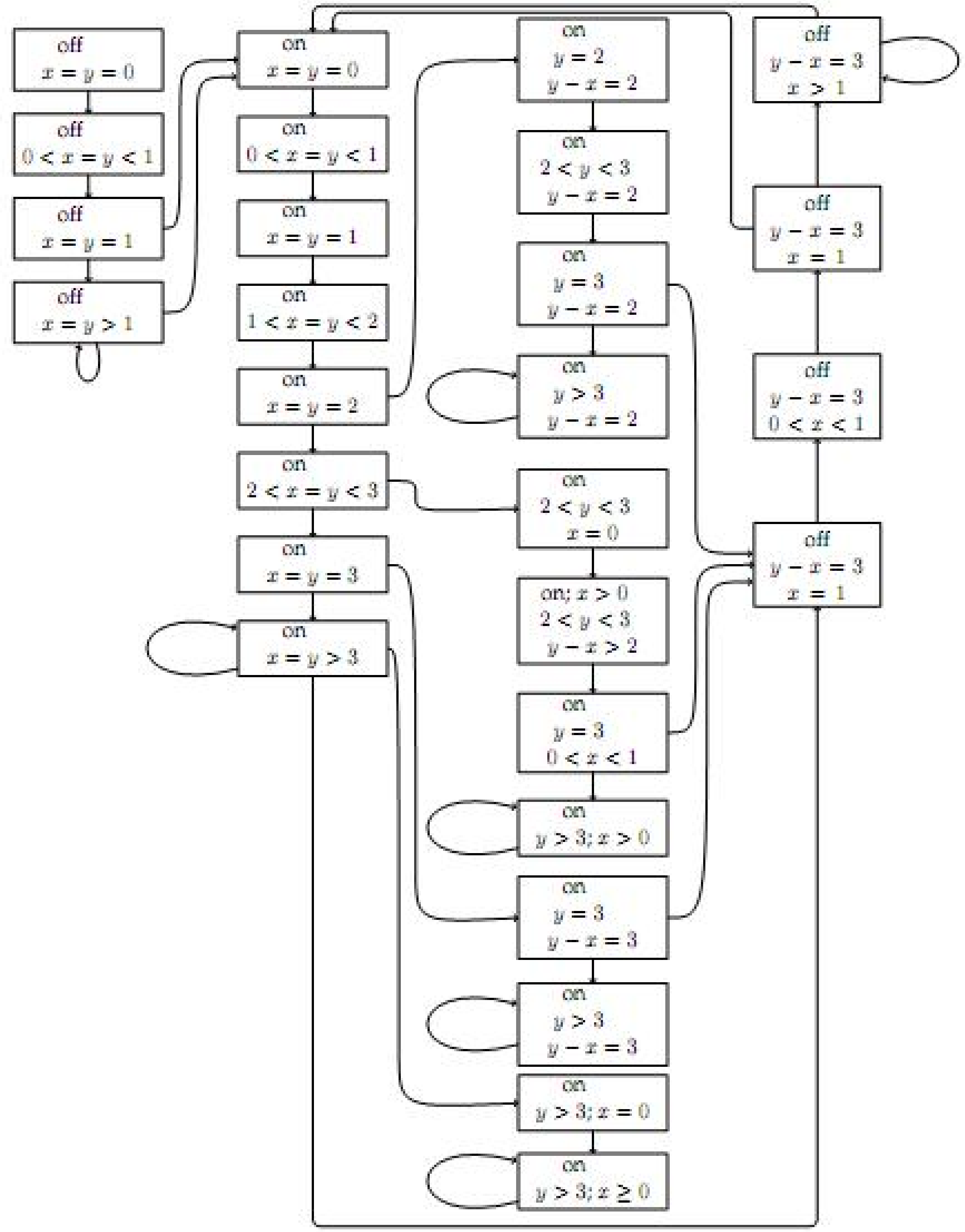
ITA



Clock Regions



Corresponding FA



$|X|!\,2^{|X|}\prod_{z\in X}(2c_{\mathcal{A}z}+2)$

Drastically increasing with the number of clocks

Special Classes of Hybrid Automata

- Finite Automata
- Integral Timed Automata ←
- Rational time automata
- Multirate automata
- Rectangular Initialized HA

- Rectangular HA

- Linear HA

- Nonlinear HA

Clocks and **Rational** Clock Constraints

- A **clock variable** x is a continuous (analog) variable of type real such that along any trajectory τ of x, for all t ∈ $\tau.dom$, $(\tau \downarrow x)(t) = t$.

- For a set X of clock variables, the set Φ(X) of ***rational*** **clock constraints** are expressions defined by the syntax:

  g ::= x $\leq q \mid x \geq q \mid \neg\, g \mid g_1 \wedge g_2$

  where $x \in X$ and $q \in \mathbb{Q}$

- Examples: x = 10.125; x ∈ [2.99, 5); true are valid rational clock constraints

- Semantics of clock constraints $[g]$

Step 1. Rational Timed Automata

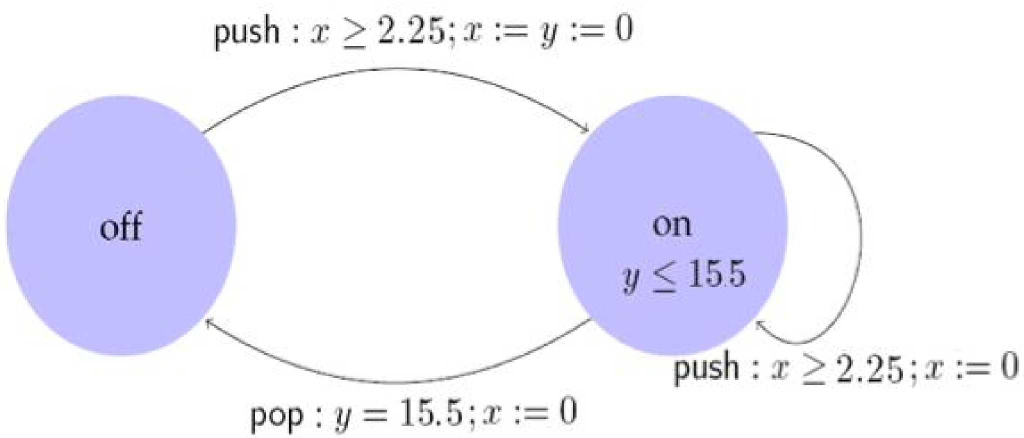
Definition. A ***rational timed automaton*** is a HA $\boldsymbol{\mathcal{A}}$ = $\langle V,\ \Theta,\ A,\ \mathcal{D},\ \mathcal{T}\rangle$ where

- V = X ∪ $\{loc\}$, where X is a set of n clocks and l is a discrete state variable of finite type L
- A is a finite set
- $\mathcal{D}$ is a set of transitions such that
  - The guards are described by **rational** clock constrings $\Phi(X)$
  - $\langle x, l\rangle - a \rightarrow \langle x', l'\rangle$ implies either $x' = x$ or $x = 0$
- $\mathcal{T}$ set of clock trajectories for the clock variables in X

Example: **Rational** Light switch

Switch can be turned on whenever at least 2.25 time units have elapsed since the last turn off or on. Switche can be turn off 15.5 time units after the last on.

automaton Switch
  internal push; pop
    variables
      internal x, y:Real := 0, loc:{on,off} := off
     transitions
       push
         pre x >=2.25
         eff if loc = on **then** y := 0 **fi;** x := 0;
        loc := off
       pop
         pre y = 15.5 ∧ loc = off
         eff x := 0; loc := off
     trajectories
       invariant loc = on ∨ loc = off
       stop when y = 15.5
       evolve d(x) = 1; d(y) = 1



Control State (Location) Reachability Problem

- Given an RTA, check if a particular mode is reachable from the initial states
- Is problem decidable?
- Yes
- Key idea:
  - Construct a ITA that has exactly same mode reachability behavior as the given RTA (timing behavior may be different)
  - Check mode reachability for ITA

Construction of ITA from RTA

- Multiply all rational constants by a factor q that make them integral
- Make d(x) = q for all the clocks
- RTA Switch reaches the same control locations as the ITA Iswitch

automaton ISwitch

internal push; pop

variables

  internal x, y:Real := 0, loc:{on,off} := off

transitions

  push

    pre x >= 9

    eff if loc = on **then** y := 0 **fi;**
    x := 0; loc := off

  pop

    pre y = 62 $\wedge$ loc = off

    eff x := 0

trajectories

  invariant loc = on $\vee$ loc = off

  stop when y = 62

  evolve d(x) = 4; d(y) = 4

Step 2. Multi-Rate Automaton

- **Definition.** A **multirate automaton** is $\boldsymbol{\mathcal{A}} = \langle V,\ Q,\ \Theta,\ A,\ \mathcal{D},\ \mathcal{T}\rangle$ where
  - V = X ∪ $\{loc\}$, where X is a set of n **continuous variables** and loc is a discrete state variable of finite type L
  - A is a finite set of actions
  - $\mathcal{D}$ is a set of transitions such that
    - The guards are described by **rational** clock constraings $\Phi(X)$
    - $\langle x, l\rangle - a \rightarrow \langle x', l'\rangle$ implies either $x' = c \text{ or } x' = x$
  - $\mathcal{T}$ set of trajectories such that

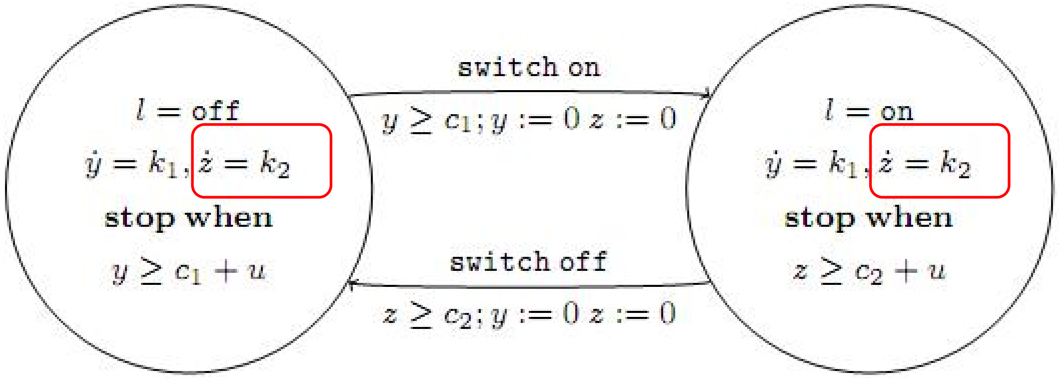
    for each variable $x \in X\ \exists k \text{ such that } \tau \in \mathcal{T},\ t \in \tau.dom$

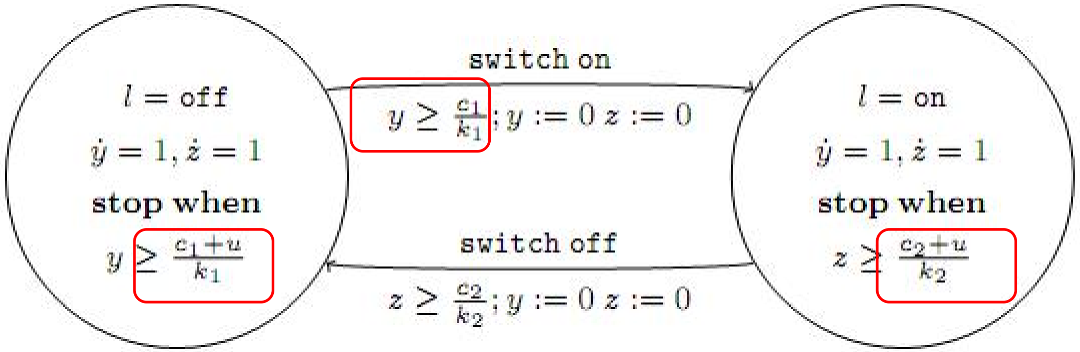
$$\tau(t).x = \tau(0).x + k\ t$$

Control State (Location) Reachability Problem

- Given an MRA, check if a particular location is reachable from the initial states
- Is problem is decidable? Yes
- Key idea:
  - Construct a RTA that is bisimilar to the given MRA

Example: Multi-rate to rational TA





Step 3. Rectangular HA

Definition. A **rectangular hybrid automaton (RHA)** is a HA $\mathcal{A} = \langle V, A, \mathcal{T}, \mathcal{D} \rangle$ where

- V = X ∪ $\{loc\}$, where X is a set of n **continuous variables** and loc is a discrete state variable of finite type L
- A is a finite set
- $\mathcal{T} = \cup_\ell \mathcal{T}_\ell$ set of trajectories for X
  - For each $\tau \in \mathcal{T}_\ell$, $x \in X$ either (i) $d(x) = k_\ell$ or (ii) $d(x) \in [k_{\ell 1}, k_{\ell 2}]$
  - Equivalently, (i) $\tau(t)\lceil x = \tau(0)\lceil x + k_\ell t$

    (ii) $\tau(0)\lceil x + k_{\ell 1} t \leq \tau(t)\lceil x \leq \tau(0)\lceil x + k_{\ell 2} t$
- $\mathcal{D}$ is a set of transitions such that
  - Guards are described by **rational** clock constrings
  - $\langle x, l \rangle \rightarrow_a \langle x', l' \rangle$ implies $x' = x$ *or* $x' \in [c_1, c_2]$

CSR Decidable for RHA?

- Given an RHA, check if a particular location is reachable from the initial states?
- Is this problem decidable? No
    - **[Henz95]** Thomas Henzinger, Peter Kopke, Anuj Puri, and Pravin Varaiya. What's Decidable About Hybrid Automata?. Journal of Computer and System Sciences, pages 373–382. ACM Press, 1995.
    - CSR for RHA reduction to Halting problem for 2 counter machines
    - Halting problem for 2CM known to be undecidable
    - Reduction in **next lecture**

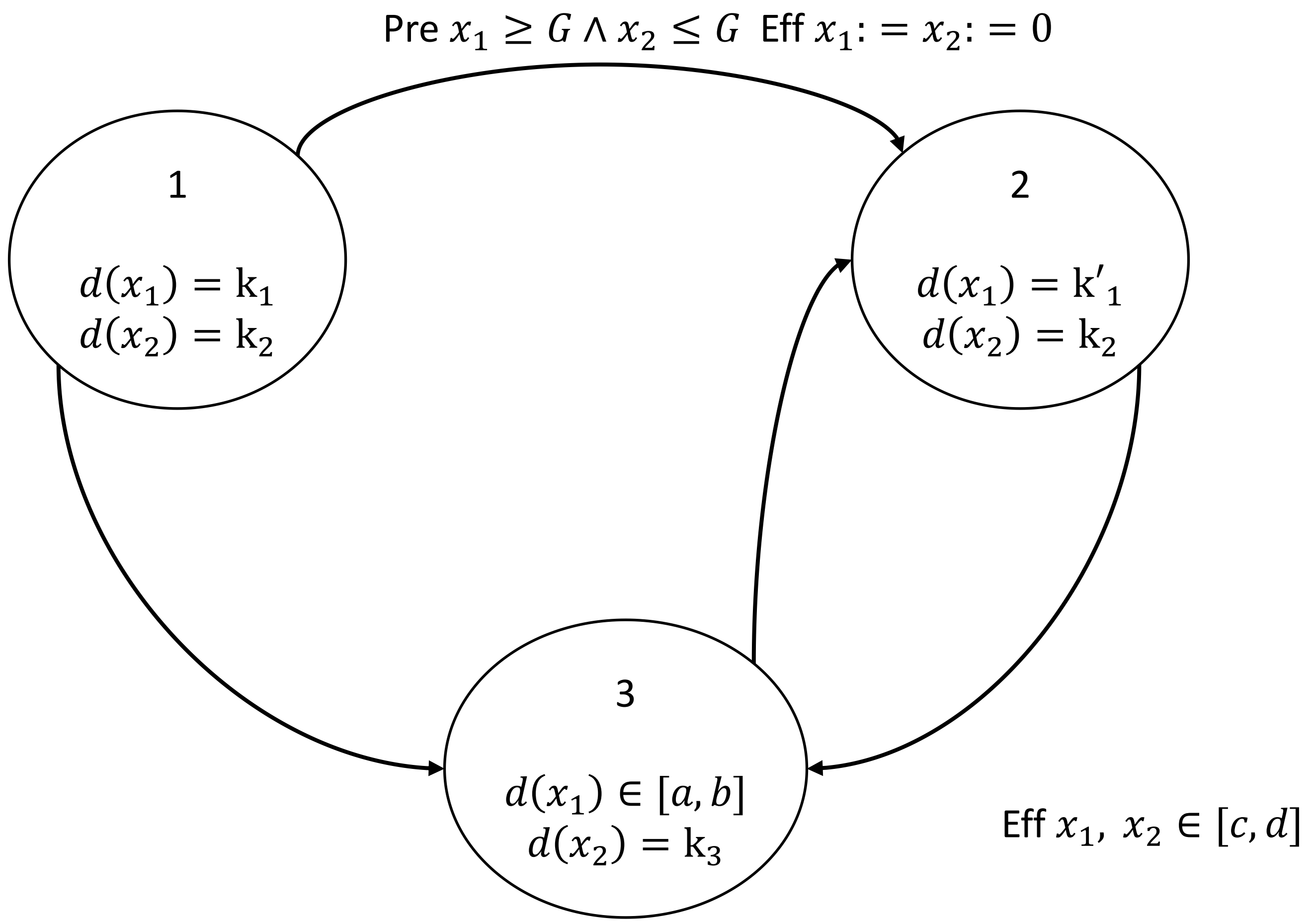
Step 4. Initialized Rectangular HA

Definition. ***An initialized rectangular hybrid automaton*** **(IRHA)** is a RHA $\mathcal{A}$ where

- V = X ∪ $\{loc\}$, where X is a set of n continuous variables and $\{loc\}$ is a discrete state variable of finite type Ł
- A is a finite set
- $\mathcal{T} = \cup_\ell \mathcal{T}_\ell$ set of trajectories for X
  - For each $\tau \in \mathcal{T}_\ell$, $x \in X$ either (i) $d(x) = k_\ell$ or (ii) $d(x) \in [k_{\ell 1}, k_{\ell 2}]$
  - Equivalently, (i) $\tau(t)\lceil x = \tau(0)\lceil x + k_\ell t$

    (ii) $\tau(0)\lceil x + k_{\ell 1} t \leq \tau(t)\lceil x \leq \tau(0)\lceil x + k_{\ell 2} t$
- $\mathcal{D}$ is a set of transitions such that
  - Guards are described by **rational** clock constrains
  - $\langle x, l\rangle \rightarrow_a \langle x', l'\rangle$ implies if dynamics changes from ℓ to ℓ' then $x' \in [c_1, c_2]$, otherwise $x' = x$

Example: Rectangular Initialized HA



Both Pre $x_1,\ x_2$ have to be reset

CSR Decidable for IRHA?

- Given an IRHA, check if a particular location is reachable from the initial states
- Is this problem decidable? Yes
- Key idea:
    - Construct a 2n-dimensional **initialized** multi-rate automaton that is bisimilar to the given IRHA
    - Construct a ITA that is bisimilar to the Singular TA

Takeaway messages

- For restricted classes of HA, e.g., ITA, IRHA, Control state reachability is decidable (Alur-Dill)
- The problem becomes undecidable for RHA (Henzinger et al.)
  - Important message to re-focus on relaxed problem
  - Bounded time, approximate reachability