Timed to Hybrid Automata

Sayan Mitra

Verifying cyberphysical systems

mitras@illinois.edu

Review. Forward simulation relation

Consider a pair of automata $\mathcal{A}_1 = \langle Q_1, \Theta_1, A_1, D_1 \rangle$ and $\mathcal{A}_2 = \langle Q_2, \Theta_2, A_2, D_2 \rangle$.
Recall *trace* of an execution preserves the visible part of an execution

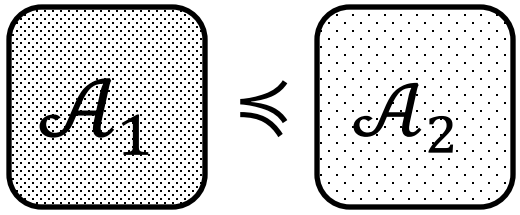
Definition. A **relation** $R \subseteq Q_1 \times Q_2$ is a forward simulation relation from $\mathcal{A}_1$ to $\mathcal{A}_2$ if

1. For every $q_1 \in \Theta_1$ there exists a $q_2 \in \Theta_2$ such that $q_1 R q_2$
2. For every transition $q_1 \rightarrow_1^{a_1} q_1'$ and $q_1 R q_2$ there exists q_2', a_2 such that
   - $q_2 \rightarrow_2^{a_2} q_2'$
   - $q_1' \; R \; q_2'$
   - $Trace(q_1, a_1, q_1') = Trace(q_2, a_2, q_2')$

Theorem. If there exists a forward simulation from $\mathcal{A}_1$ to $\mathcal{A}_2$ then $Traces_1 \subseteq Traces_2$.

Abstraction recap

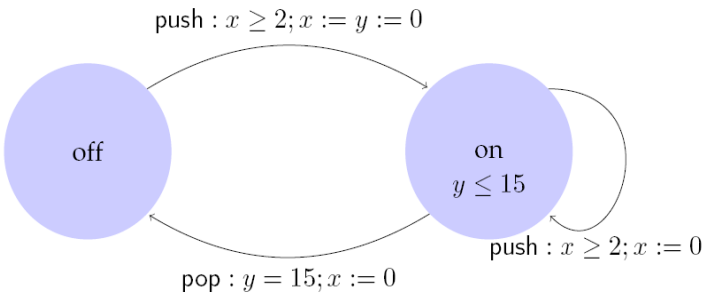
- Defined what it means for $\mathcal{A}_2$ to be abstraction of$\mathcal{A}_1$
- $Traces_{\mathcal{A}_1} \subseteq Traces_{\mathcal{A}_2}$
- $\mathcal{A}_1 \preccurlyeq_T \mathcal{A}_2$
- If $\mathcal{A}_1 \preccurlyeq_T \mathcal{A}_2$ and $\mathcal{A}_2 \preccurlyeq_T \mathcal{A}_1$ then $\mathcal{A}_1 \preccurlyeq_T \mathcal{A}_3$
- Transitive, $\preccurlyeq_T$ defines a preordering on compatible automata
- We saw methods for proving $\mathcal{A}_1 \preccurlyeq_T \mathcal{A}_2$
  - *Forward simulation and backward simulation*
- $\preccurlyeq_T$ defines a preorder (reflexive and transitive)



Review: Integral Timed Automata

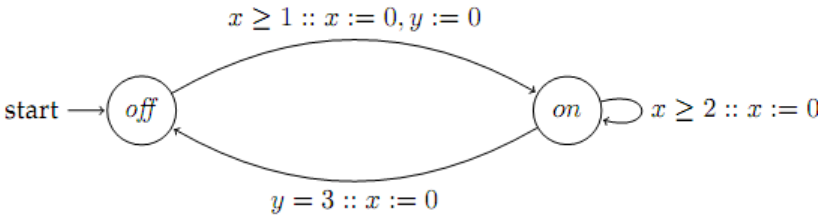
Definition. An integral timed automaton is a HIOA $\mathcal{A} = \langle V, \Theta, A, \mathcal{D}, \mathcal{T} \rangle$ where

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

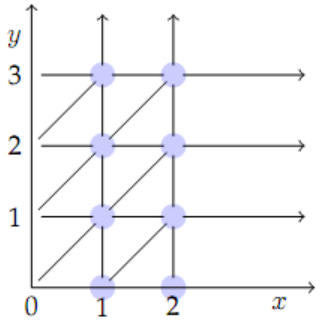


Control State Reachability of ITA 2

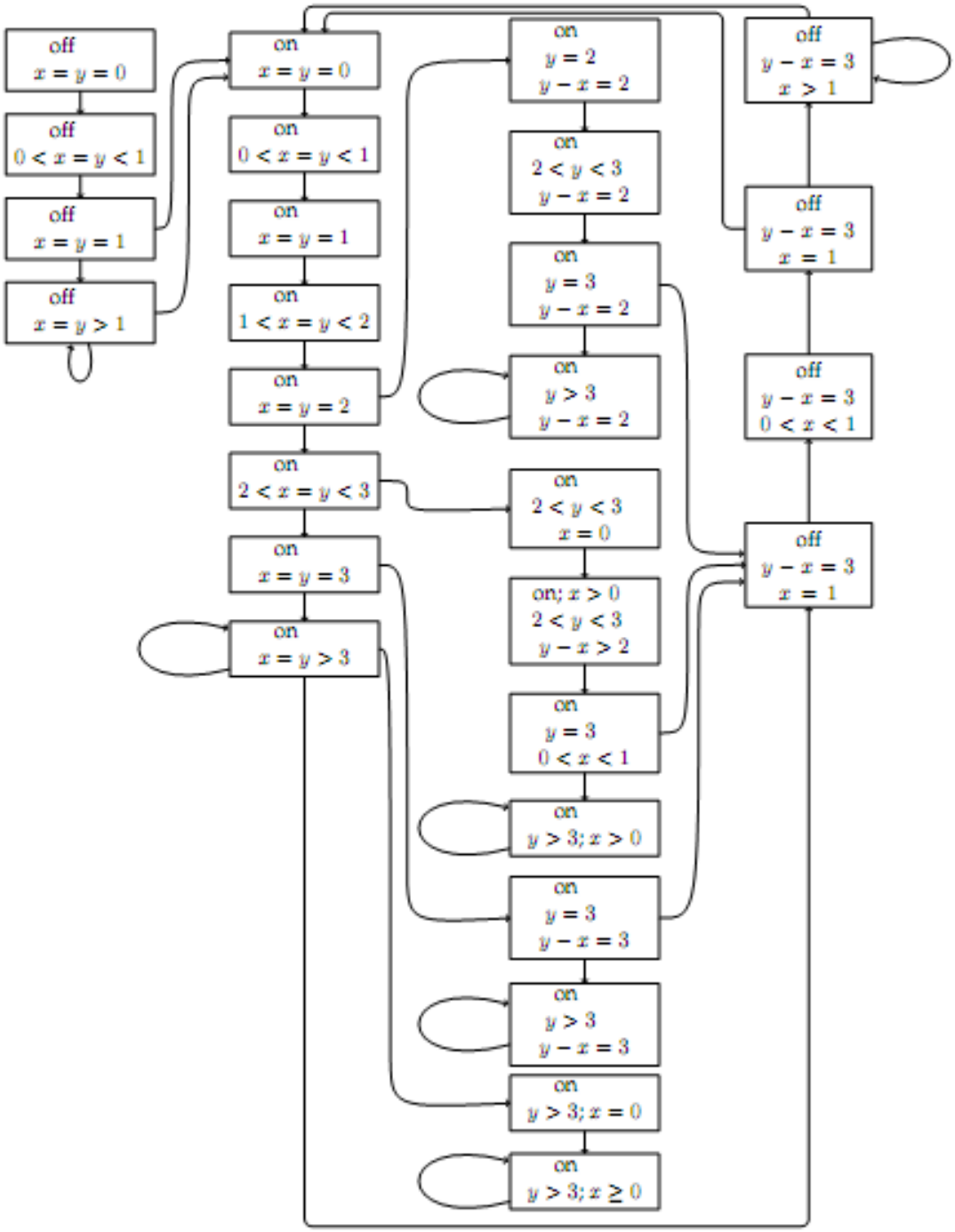
ITA



Clock Regions



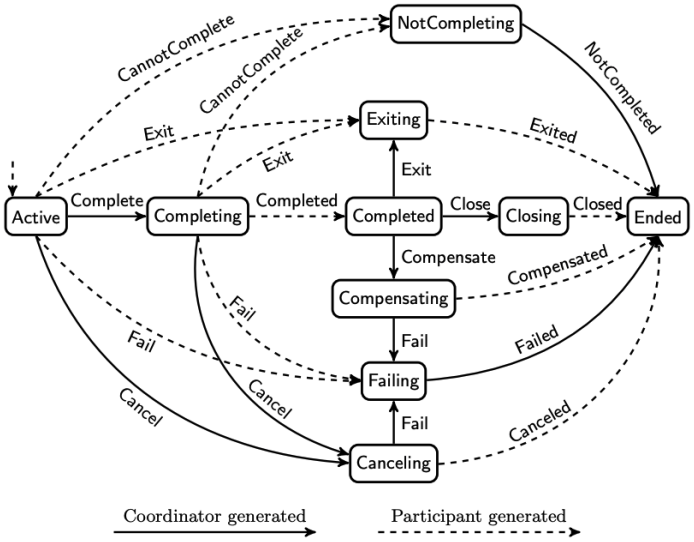
Abstraction Integer Timed Automaton as Region Automaton

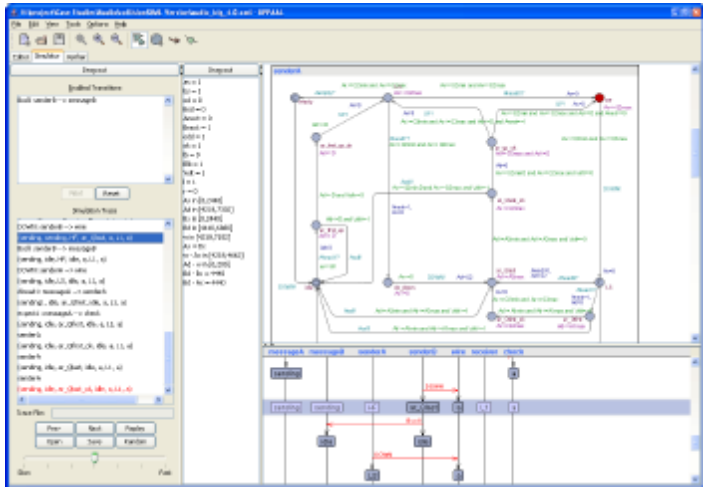


$$|X|!\, 2^{|X|} \prod_{z \in X} (2c_{\mathcal{A}z} + 2)$$

Drastically increasing with the number of clocks

Timed Automaton application in Web Services (WS)





WS-Coordination describes a framework for coordinating transactional web services

Network protocol described in state tables

600+ lines of C-like code in the protocol model

Modeled and Verified using the UPPAAL tool

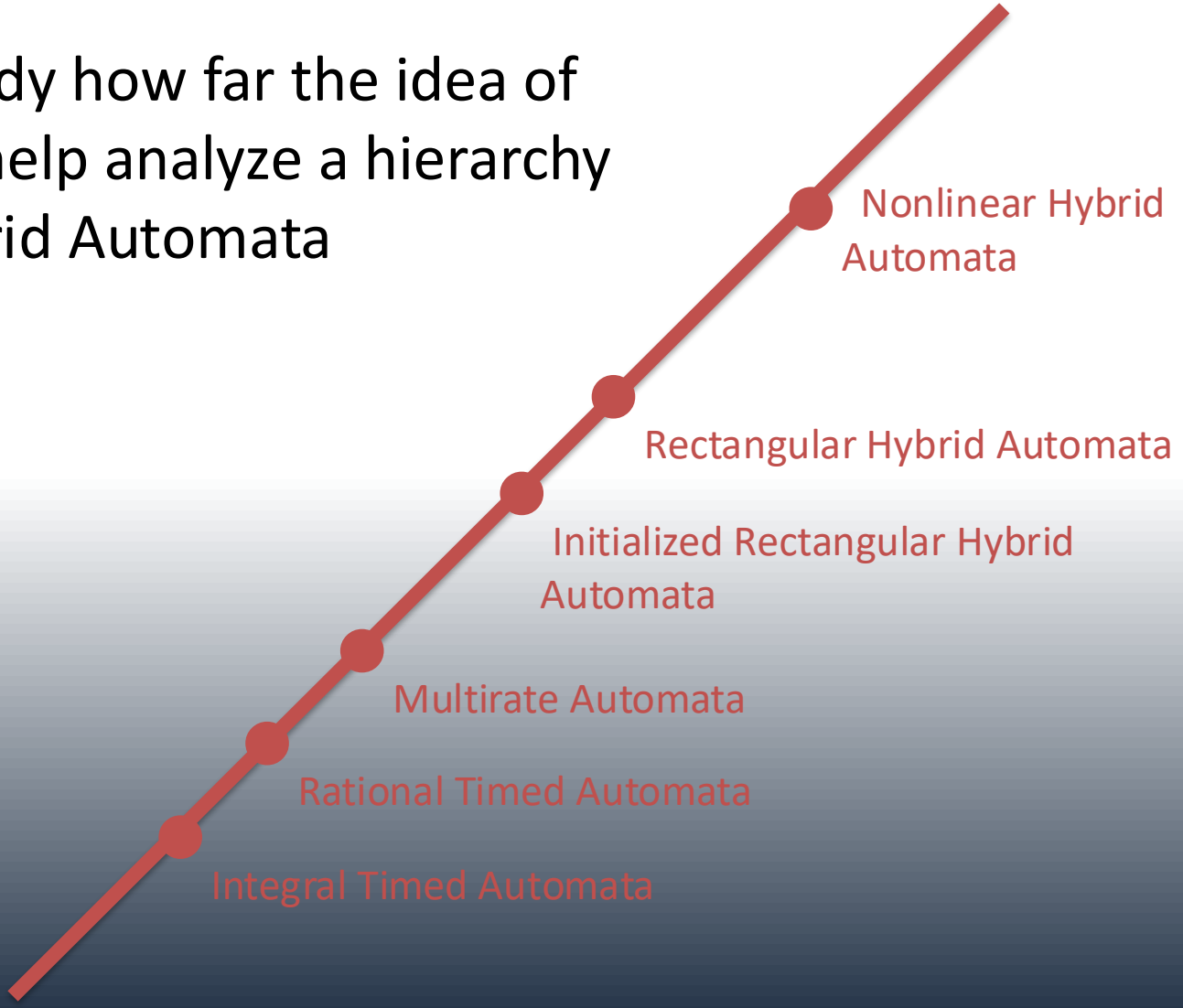
Analysis considers different channel models

The main safety property: protocol **does not enter invalid state**

Property violated in all but the FIFO channel model

Modelling and Verification of Web Services Business Activity Protocol Anders P. Ravn, Jiri Srba, and Saleem Vighio, RV 2010

Later we will study how far the idea of abstractions can help analyze a hierarchy of Hybrid Automata



Lecture Slides by Sayan Mitra
mitras@illinois.edu

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 ≤ q | $x \geq q$ | ¬ g | $g_1 \wedge g_2$

  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 $[g]$

Rational Timed Automata

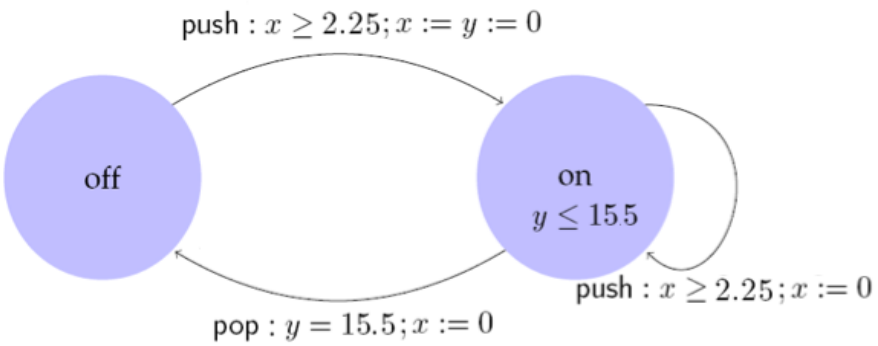
Definition. A ***rational timed automaton*** is a HA $\mathcal{A} = \langle V, \Theta, A, \mathcal{D}, \mathcal{T} \rangle$ where

- V = X ∪ $\{loc\}$, where X is a set of n clocks and l is a discrete state variable of finite type Ł
- A is a finite set
- $\mathcal{D}$ 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 $x' = x$ 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. Switches off automatically 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 x := 0 fi;
            y := 0; loc := on
    pop
      pre y = 15.5 ∧ loc = on
      eff x := 0; loc = off
  trajectories
    invariant loc = on ⇒ y ≤15.5
    evolve d(x) = 1; d(y) = 1
```



Control State (Location) Reachability Problem

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

- Simulation relation R is given by
- (**u,s**) $\in R$ iff **u.**x = 4 **s.**x and **u.**y = 4 **s.**y

```
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 x := 0 fi; y := 0; loc := on
  pop
    pre y = 62
    eff x := 0; loc = off
trajectories
    invariant loc = on ⇒ y ≤ 62
    evolve d(x) = 4; d(y) = 4
```

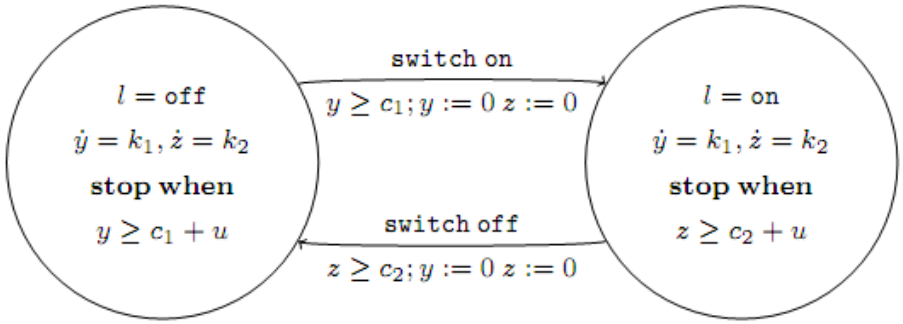
Step 2. Multi-Rate Automaton

Definition. A multirate automaton is $\mathcal{A} = \langle V, Q, \Theta, A, \mathcal{D}, \mathcal{T} \rangle$ where

- V = X ∪ $\{loc\}$, where X is a set of n continuous variables and loc is a discrete state variable of finite type Ł
- 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 $x' = c \; or \; x' = x$
- $\mathcal{T}$ set of trajectories such that

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

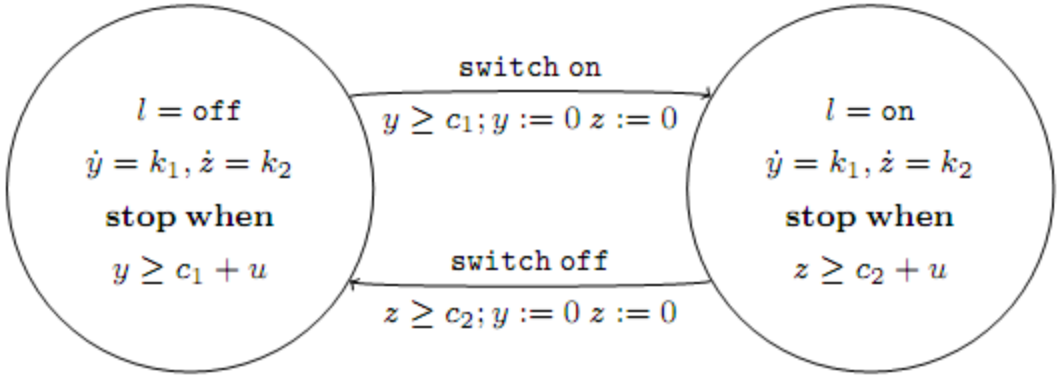
$$\tau(t).x = \tau(0).x + k\,t$$

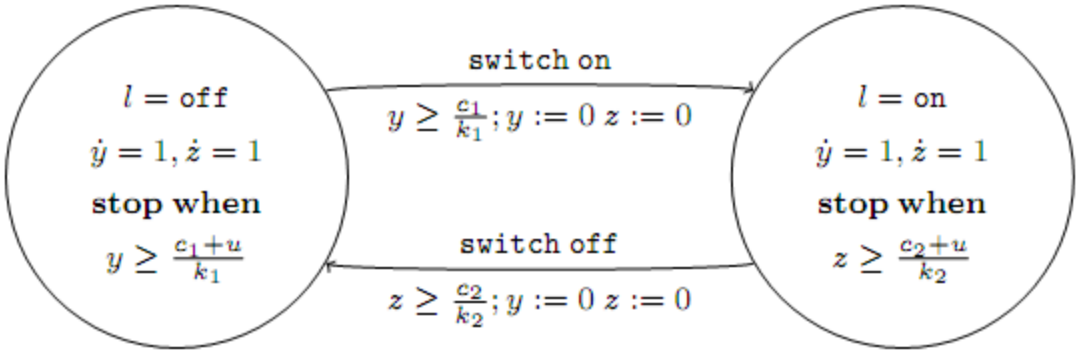


Control State (Location) Reachability Problem

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

Example: Multi-rate to rational TA



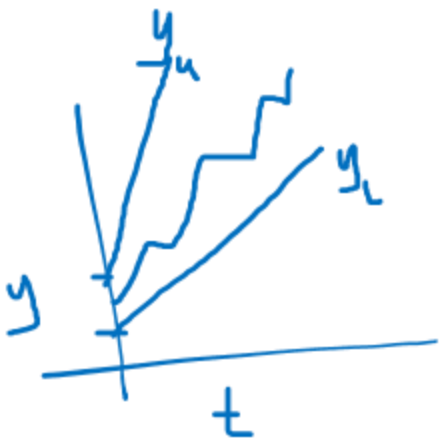


Step 3. Rectangular HA

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 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 constraings
  - $\langle x, l \rangle \rightarrow_a \langle x', l' \rangle$ 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

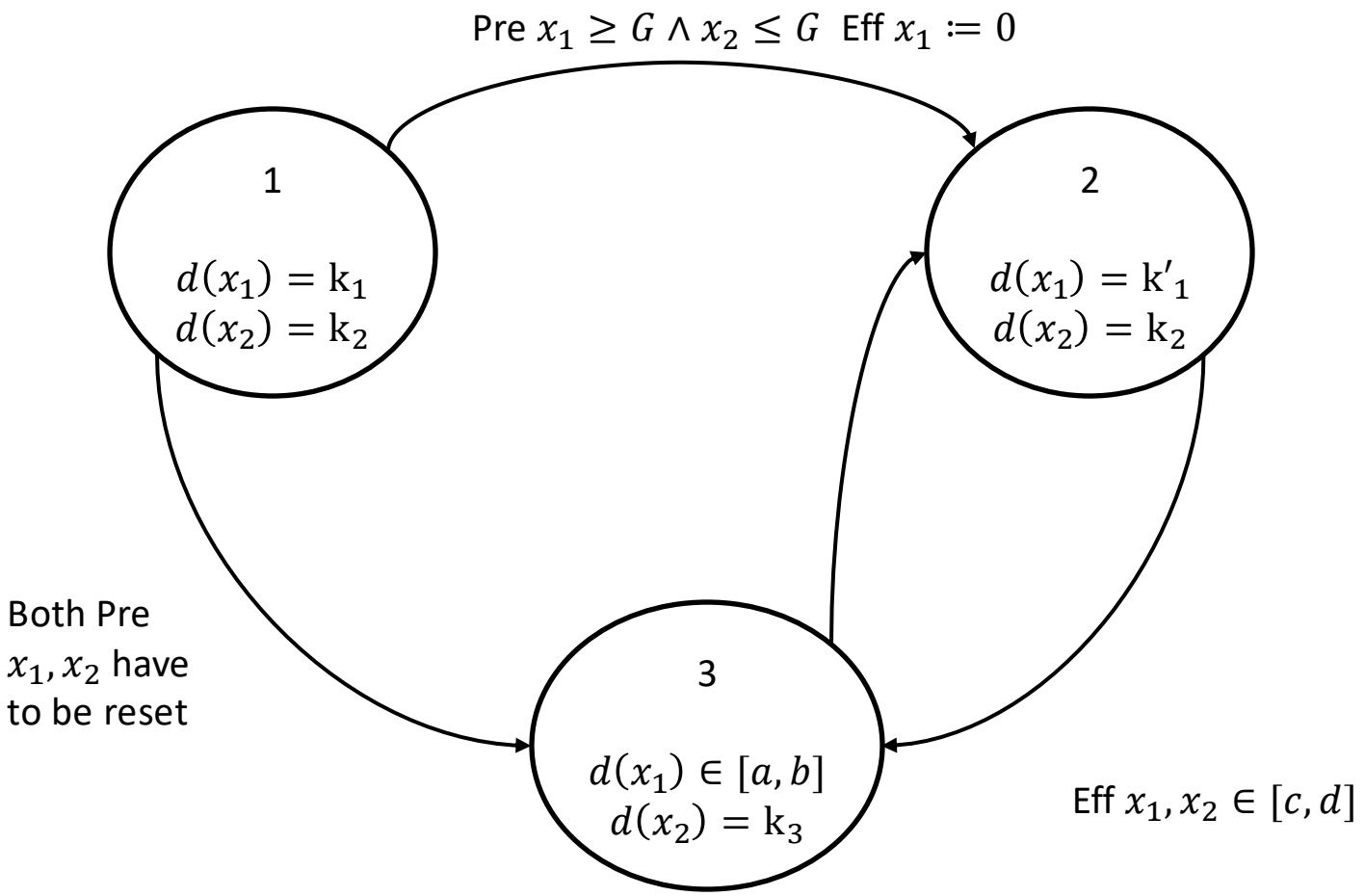
Step 4. Initialized Rectangular HA

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 constraings
  - $\langle x, l\rangle \rightarrow_a \langle x', l'\rangle$ implies if dynamics changes from ℓ to ℓ' then $x' \in [c_1, c_2]$, otherwise $x' = x$

Example: Rectangular Initialized HA



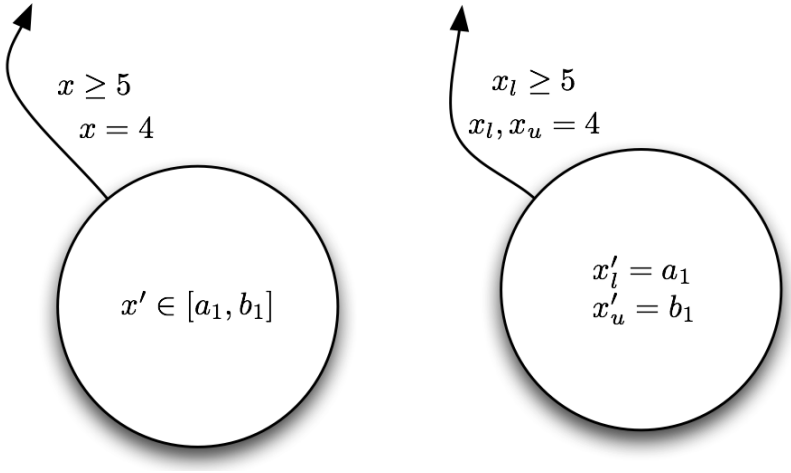
CSR Decidable for IRHA?

- 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 m**ulti-rate automaton that is bisimilar to the given IRHA
  - Construct an ITA that is bisimilar to the Singular TA

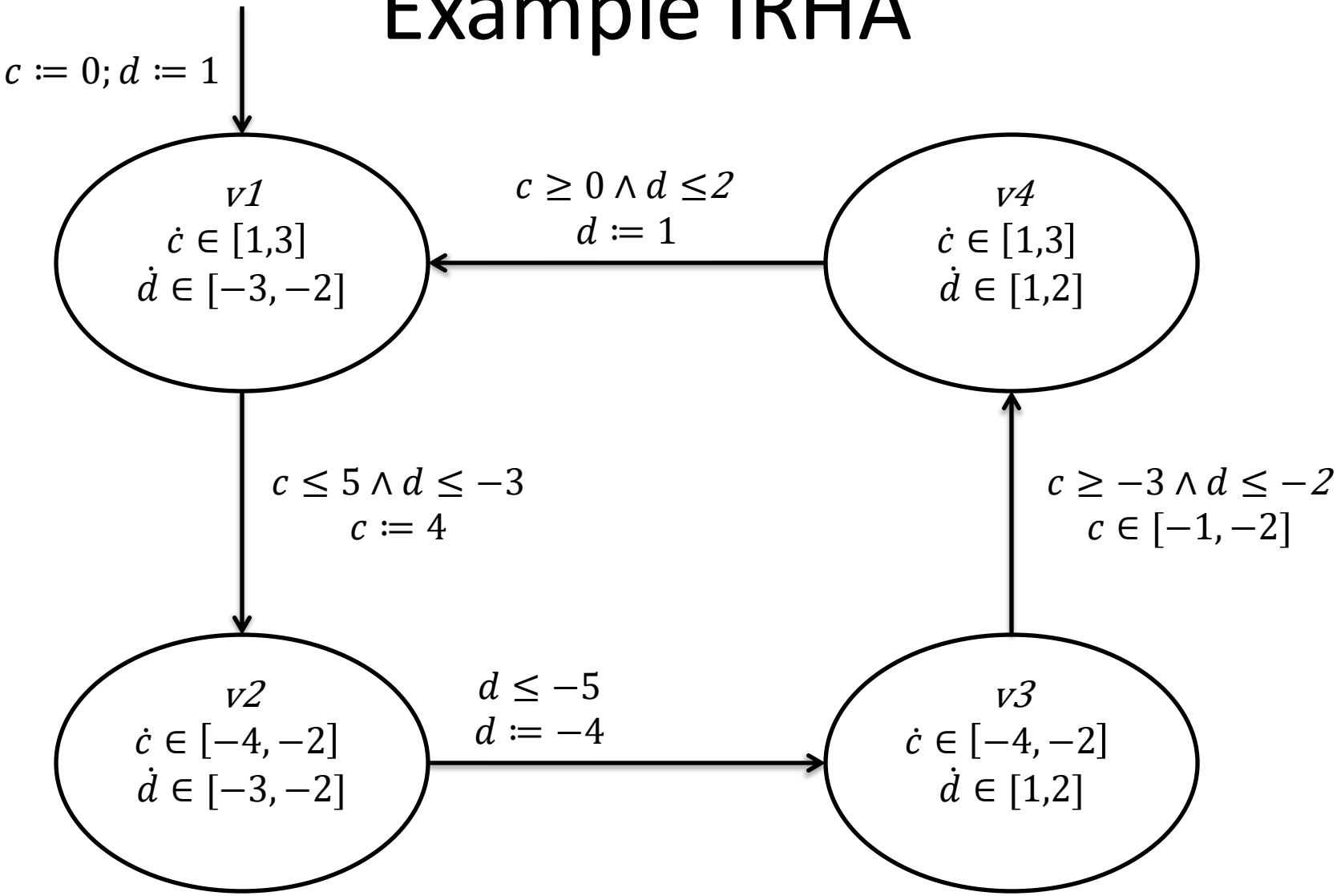
From IRHA to Singular HA conversion

For every variable create two variables---tracking the upper and lower bounds

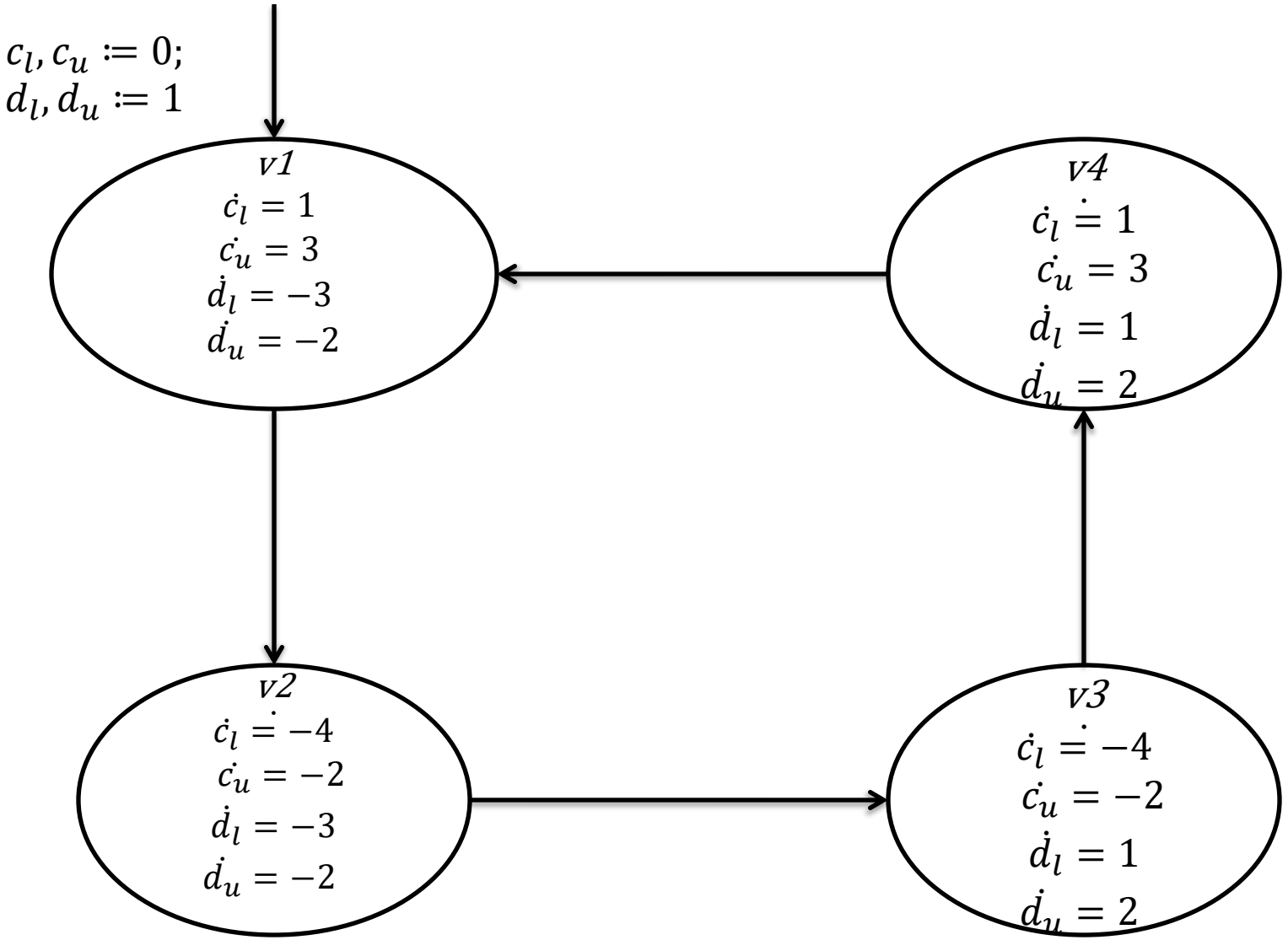
IRHA	MRA
x	x_ℓ ; x_u
Evolve: $d(x) \in [a_1, b_1]$	Evolve: $d(x_\ell) = a_1; d(x_u) = b_1$
Eff: $x' \in [a_1, b_1]$	Eff: $x_\ell = a_1; x_u = b_1$
$x' = c$	$x_\ell = x_u = c$
Guard: $x \geq 5$	$x_l \geq 5$
	$x_l < 5 \wedge x_u \geq 5$ Eff $x_l = 5$



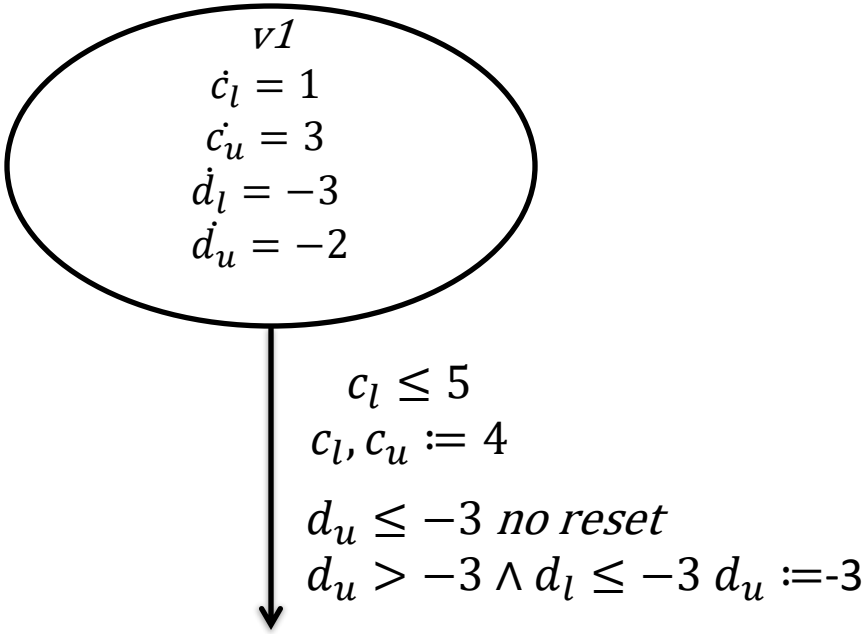
Example IRHA

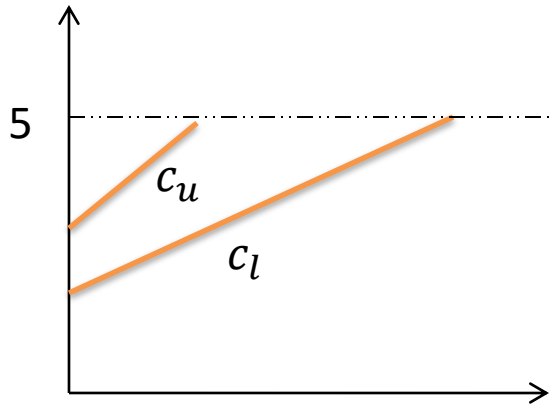


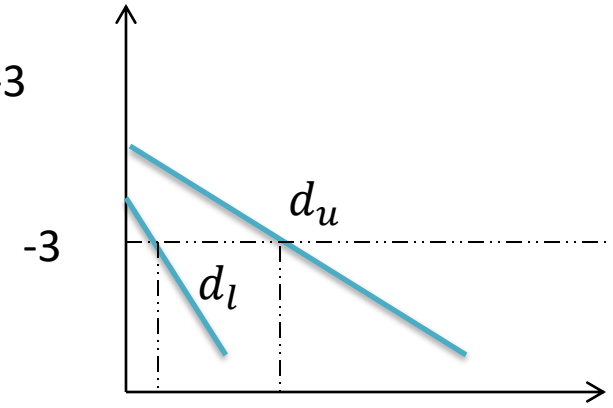
Initialized Singular HA



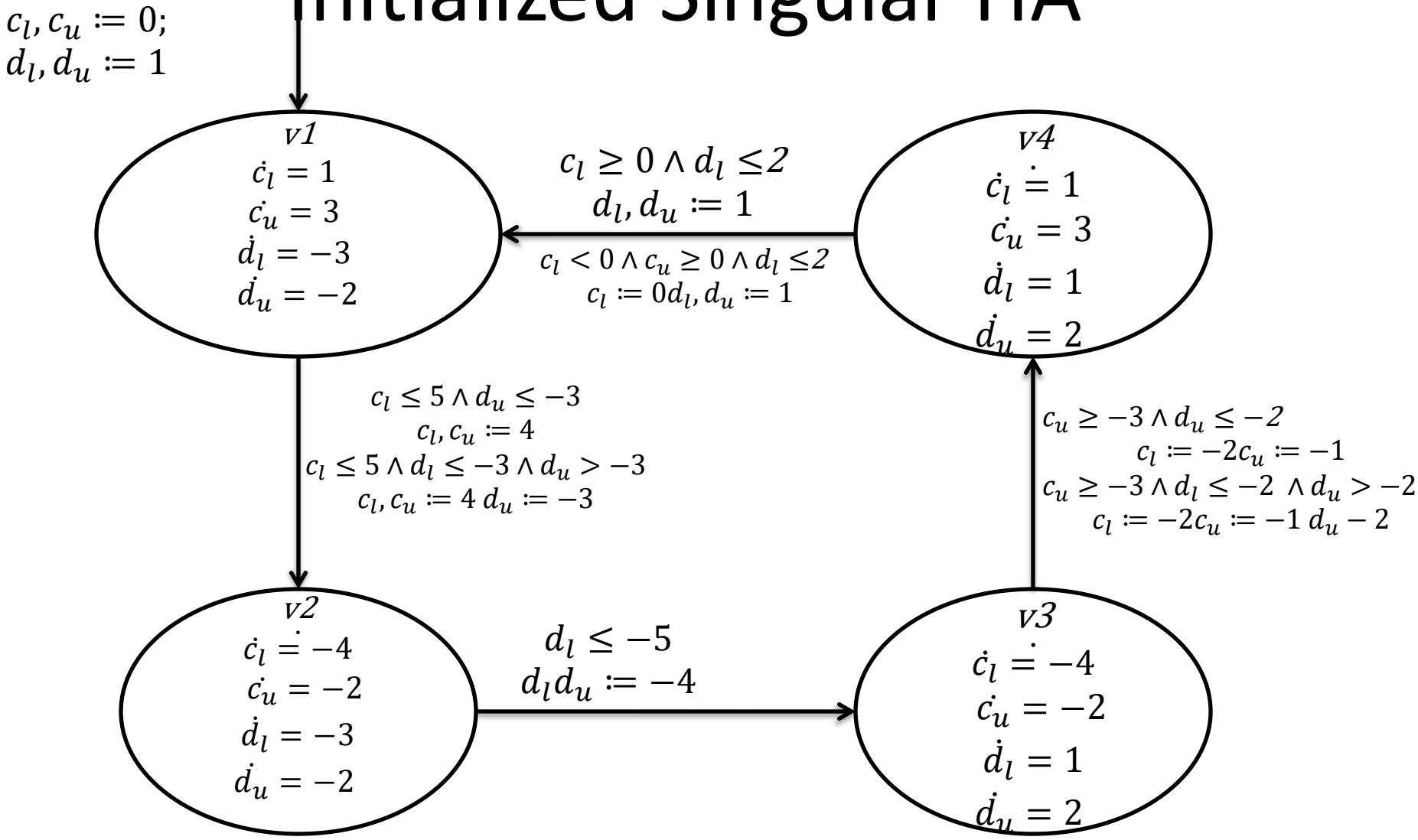
Transitions







Initialized Singular HA



Lecture Slides by Sayan Mitra
mitras@illinois.edu

Can this be further generalized ?

- For initialized Rectangular HA, control state reachability is decidable
  - Can we drop the initialization restriction?
    - No, problem becomes undecidable (next time)
  - Can we drop the rectangular restriction?
    - No, problem becomes undecidable

Practical reachability

Algorithm: BasicReach
Input: $\mathbf{A} = \langle V, \Theta, A, \mathbf{D}, \mathbf{T} \rangle$, $d > 0$
  $Rt, Reach : val(V)$
  $Rt := \Theta$;
  $Reach := \emptyset$;
  While $(Rt \not\subseteq \text{Reach})$
    $Reach := Reach \cup Rt$;
    $Rt := Rt \cup Post_{\mathbf{D}}(Rt)$;
    $Rt := Post_{\mathbf{T}(d)}(Rt)$;
Output: $Reach$

Algorithm: Post_{D}
\\ computes post of all transitions
Input: $A, \mathbf{D}, S_{in}$
  $S_{out} = \emptyset$
  For each $a \in A$
    For each $\langle g_1, g_2 \rangle \in S_{in}$
      If $[[g_1, g_2]] \cap [[g_{ga1}, g_{ga2}]] \neq \emptyset$
        $S_{out} := S_{out} \cup \langle g_{ra1}, g_{gra2} \rangle$
Output: S_{out}

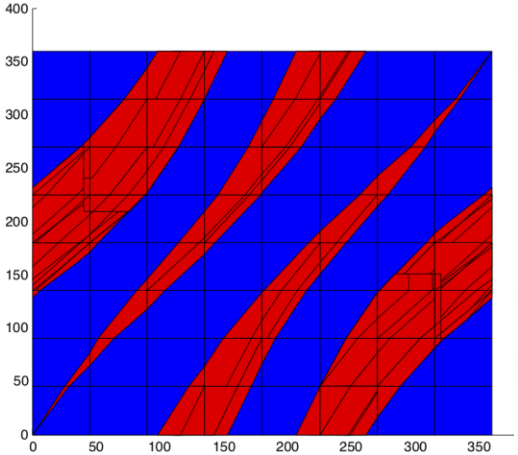
Algorithm: $\text{Post}_{\mathbf{T}(d)}$
\\ computes post of all trajectories
Input: $A, \mathbf{T}, S_{in}, d$
  $S_{out} = \emptyset$
  For each $\ell \in L$
    For each $\langle g_1, g_2 \rangle \in S_{in}$
      $P := \cup_{t \leq d} [[g_1, g_2]] \oplus [[t g_{\ell 1}, t g_{\ell 2}]]$
      $S_{out} := S_{out} \cup Approx(P)$
Output: S_{out}

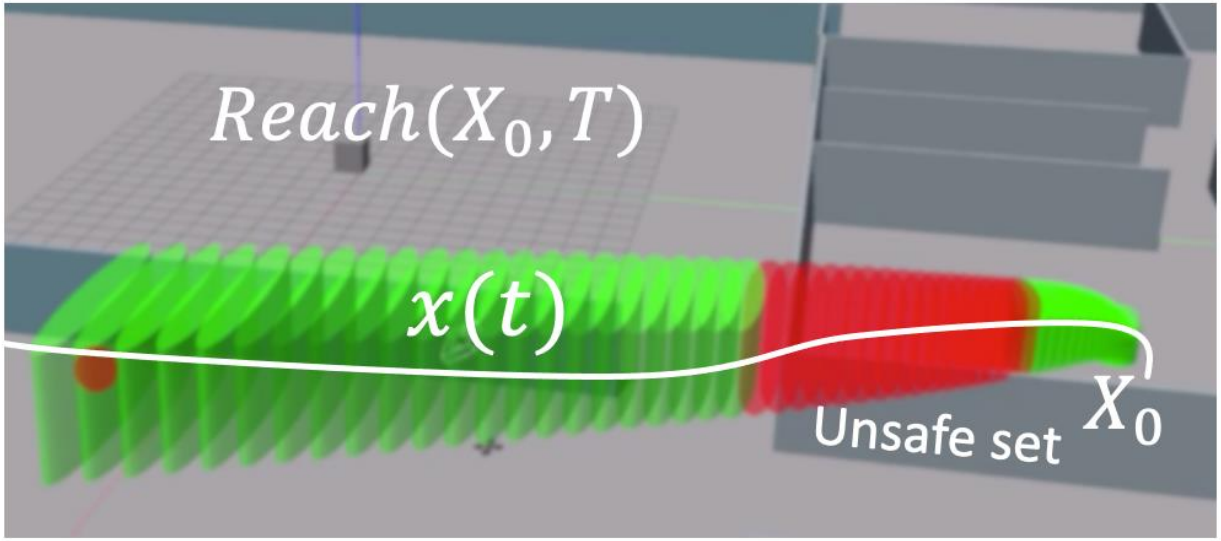
Tools:
SpaceEX
CORA
C2E2
Flow*
DryVR

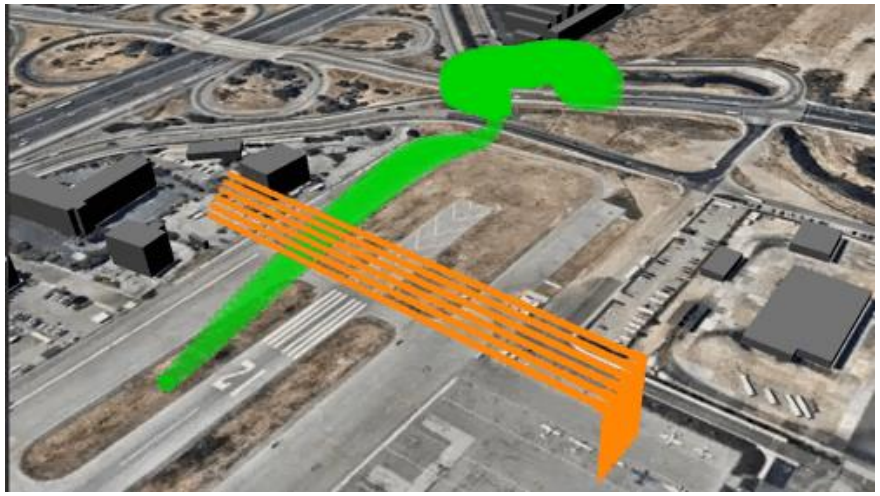
Data structures critical for reachability

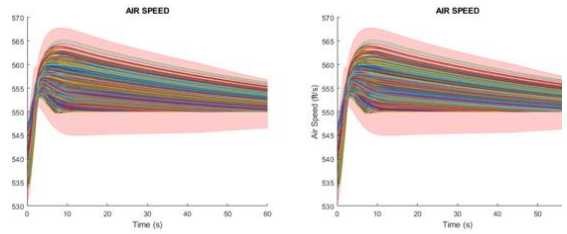
- Hyperrectangles
  - $[[g_1; g_2]] = \{x \in R^n \mid \ ||x - g_1||_\infty \leq ||g_2 - g_1||_\infty\} = \Pi_i [g_{1i}, g_{2i}]$
- Polyhedra
- Zonotopes [Girard 2005]
- Ellipsoids [Kurzhanskiy 2001]
- Support functions [Guernic et al. 2009]
- Generalized star set [Duggirala and Viswanathan 2018]

Reachability in practice

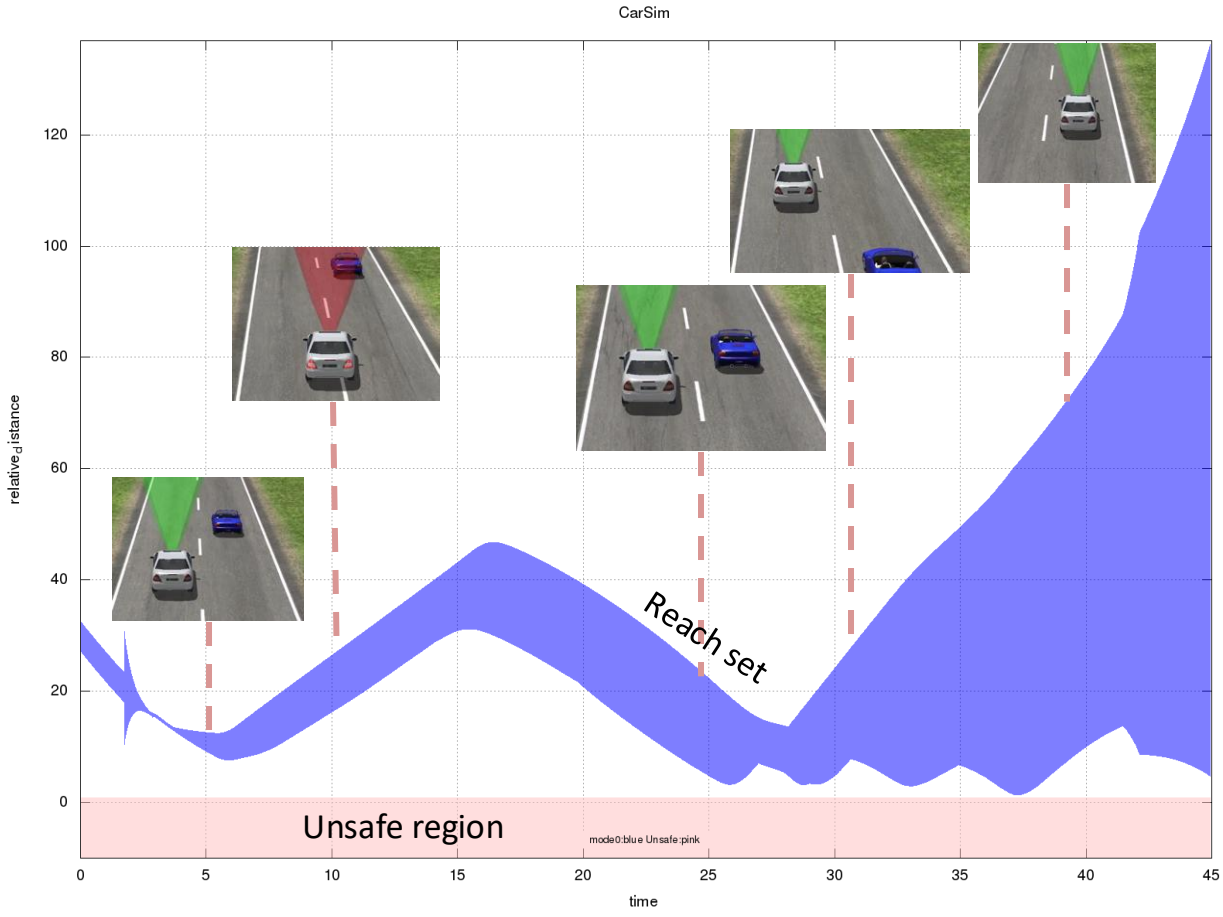


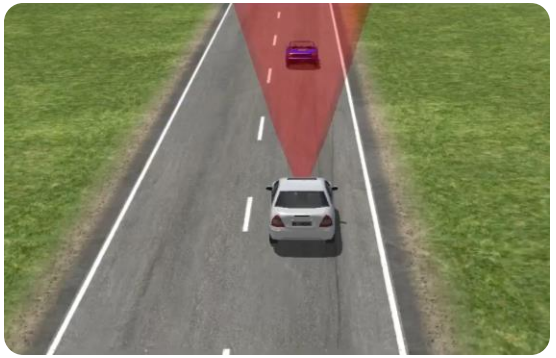






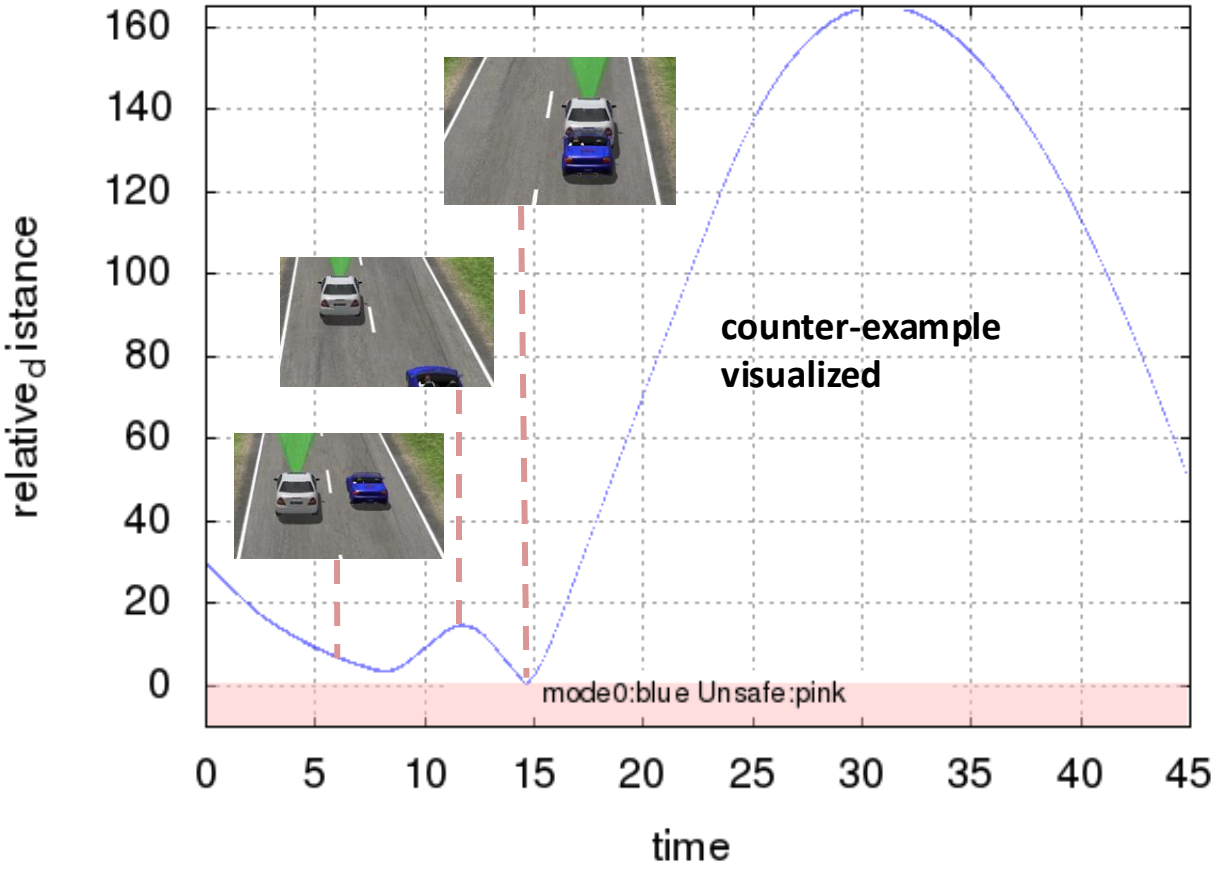
C2E2 generated safety certificate for a given user model

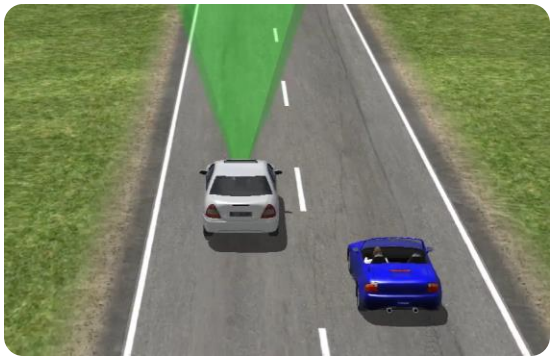




Verify no collision with **uncertainties**: speeds in [70, 85] mph and acceleration range of NPC

For a different user model C2E2 finds a corner case

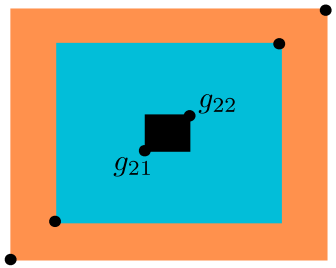




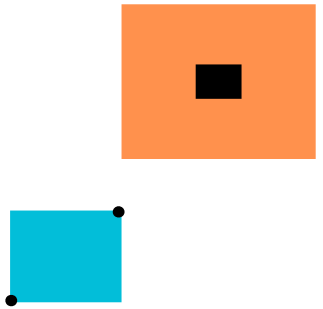
Verify no collision with **uncertainties** like speeds in [70, 85] mph and **bigger** acceleration range of NPC

Data structures: rectangles and ellipsoids

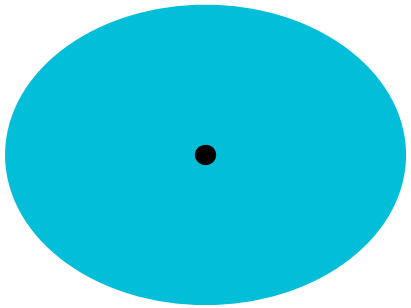
$[[g_{11}, g_{12}]]$



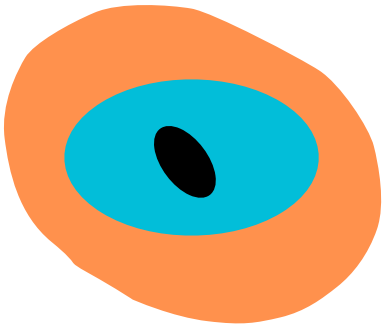
$[[g_{11}, g_{12}]] \oplus [[g_{21}, g_{22}]]$
$= [[g_{11} + g_{12}, g_{21} + g_{22}]]$



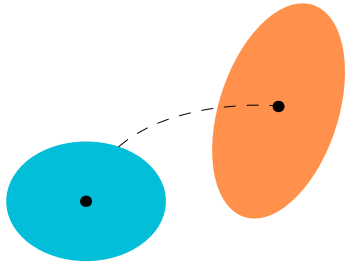
$[[g_{11}, g_{12}]] \oplus [[t.g_1, t.g_2]]$



$[[c_1, Q]]$

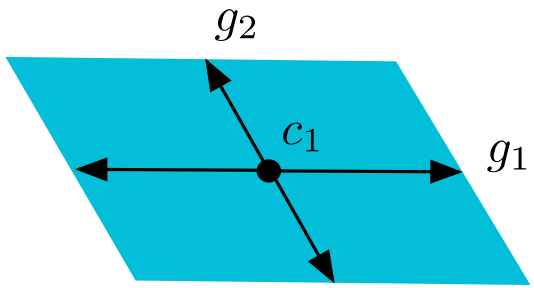


$[[c_1, Q_1]] \oplus [[c_2, Q_2]] \neq [[c_3, Q_3]]$

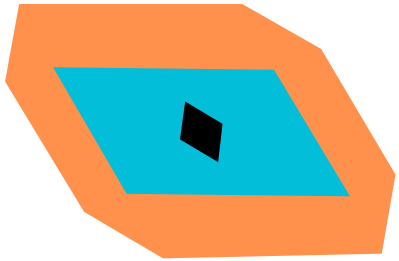


$[[Ac_1, AQA^T]]$

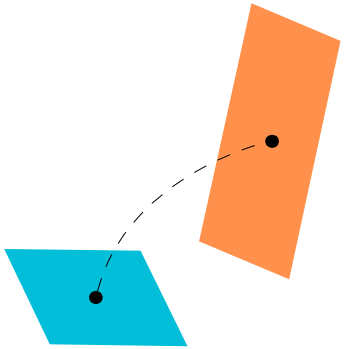
Zonotopes and polytopes



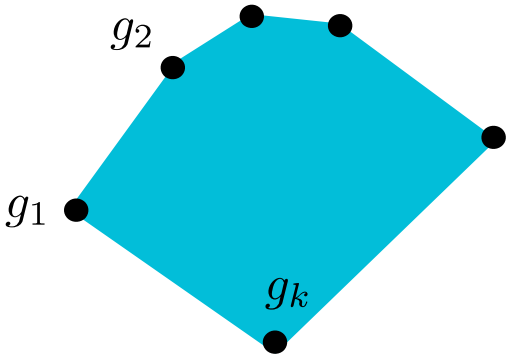
$[[c_1, \langle g_1, g_2 \rangle]]$



$[[c_1, \langle g_1, g_2 \rangle]] \oplus [[c_2, \langle g'_1, g'_2 \rangle]]$
$= [[c_1 + c_2, \langle g_1, g'_1, g_2, g'_2 \rangle]]$

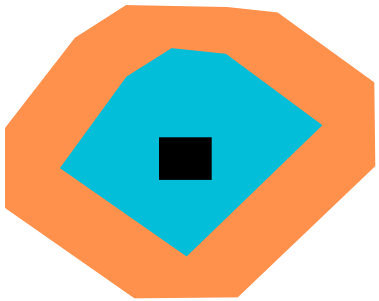


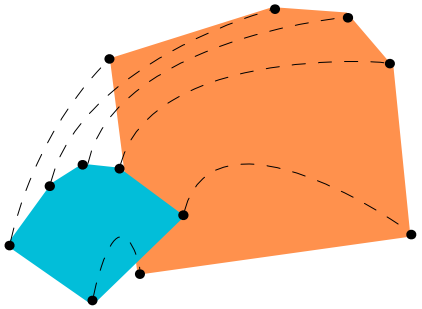
$[[Ac_1, \langle Ag_1, Ag_2 \rangle]]$



$[[A, b]]$

$[[g_1, \ldots, g_k]]$





$[[\xi(g_1, t), \ldots, \xi(g_k, t)]]$

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
- Many tools and successful applications using iterative ***Post*** computations
- Choice of data-structure critical for practical performance