Abstractions

Sayan Mitra

Verifying cyberphysical systems

mitras@illinois.edu

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

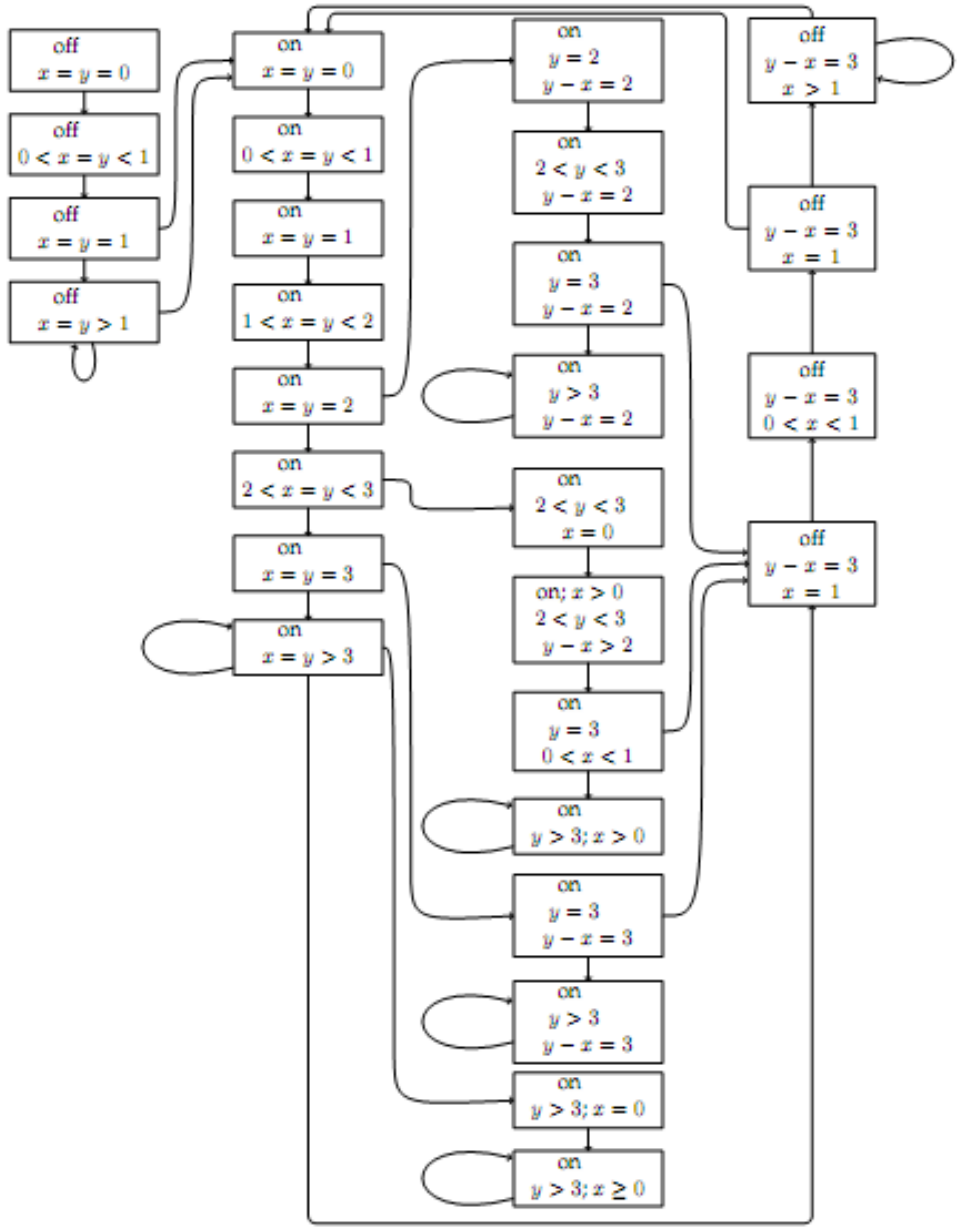
Given an ITA $\mathcal{A} = \langle V, \Theta, \mathcal{D}, \mathcal{T} \rangle$, we construct the corresponding **Region Automaton** $\mathrm{R}(\mathcal{A}) = \langle Q_R, \Theta_R, D_R \rangle$ such that (i) $\mathrm{R}(\mathcal{A})$ visits the same set of locations (but does not have timing information) 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 location 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 $\mathcal{A}$)

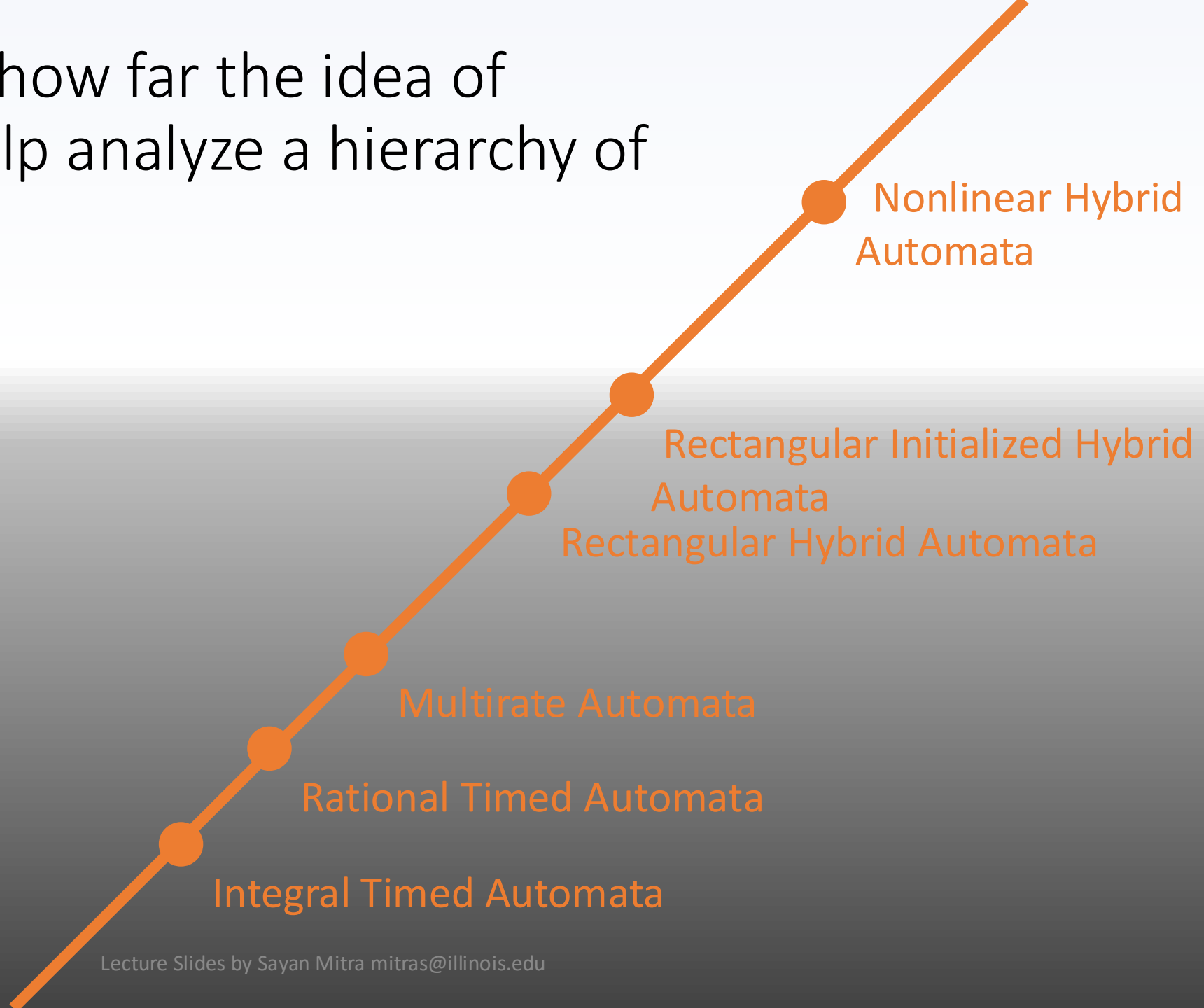
Corresponding FA



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

Drastically increasing with the number of clocks

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



Outline

- Abstractions
- Simulation relations
- Composition and substitutivity

Abstractions and Simulations

Consider models that have the same external interface (input/output variables and actions)

We would like to *approximate* one (hybrid) automaton H_1 with another one H_2

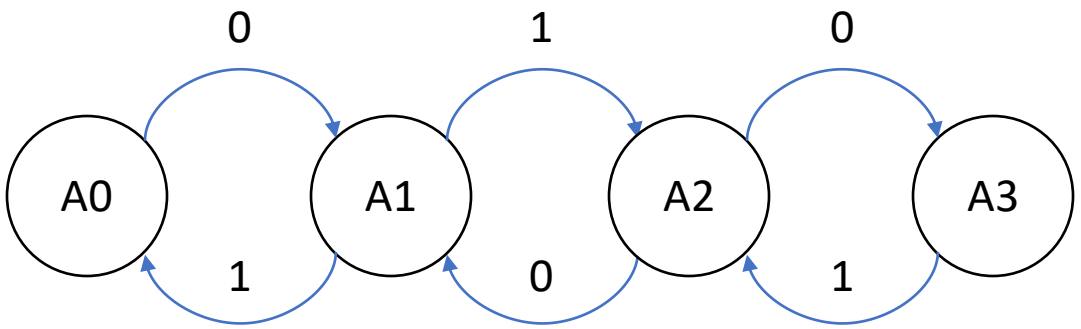
- We can over-approximate the reachable states of H_1 with those of H_2
- This would ensure that invariants of H_2 *carry over* to H_1
- We would like to go beyond invariants, and want to have more general requirements (e.g., CTL) carry over

H_2 should be *simpler* (smaller description, fewer states, transitions, linear dynamics, etc.) and preserve some properties of H_1 (and not others)

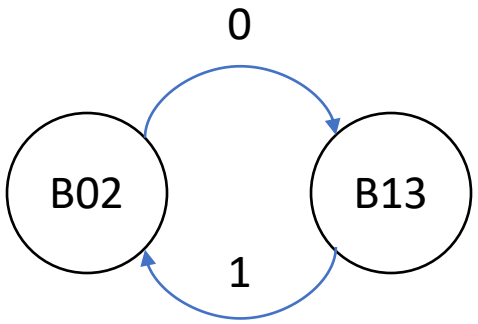
Verifying some requirements of H_2 can then carry over requirements to H_1

Finite state examples

A

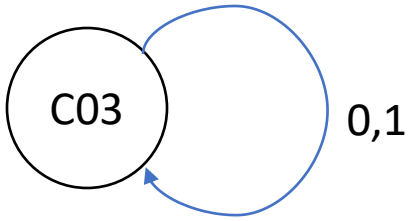


B



$Traces_B$= 01*

C



$Traces_C$= {0,1}*

An execution of A is

$\alpha = A0,0, A1,1, A2,0, A3, 1, A2$

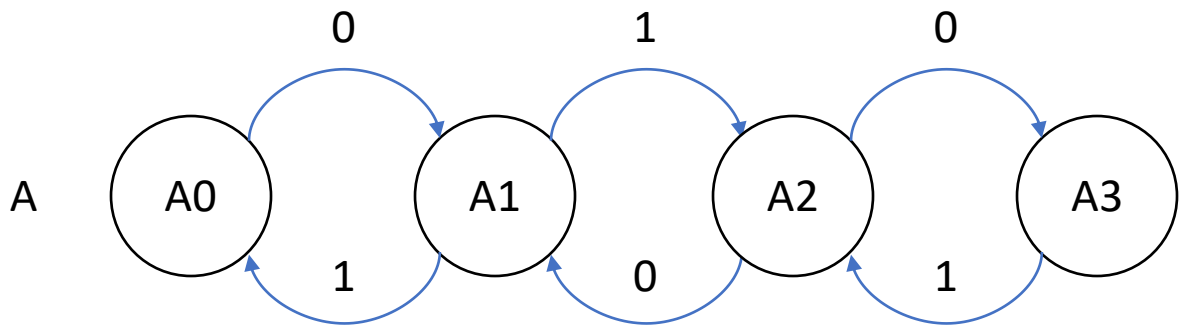
Trace is the **visible** or **observable** part of an execution

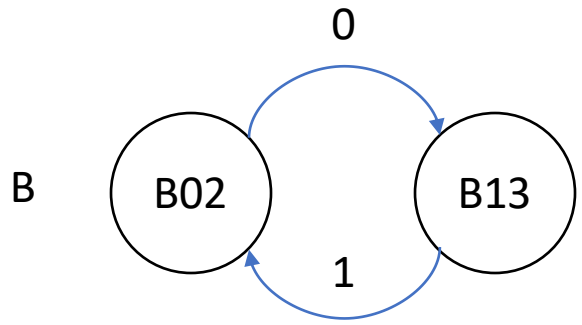
$trace(\alpha) = 0,1,0,1$

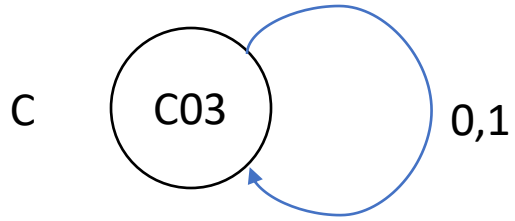
$Traces_A$: Set of all traces of A

$Traces_A = (01)^*$

Finite state examples



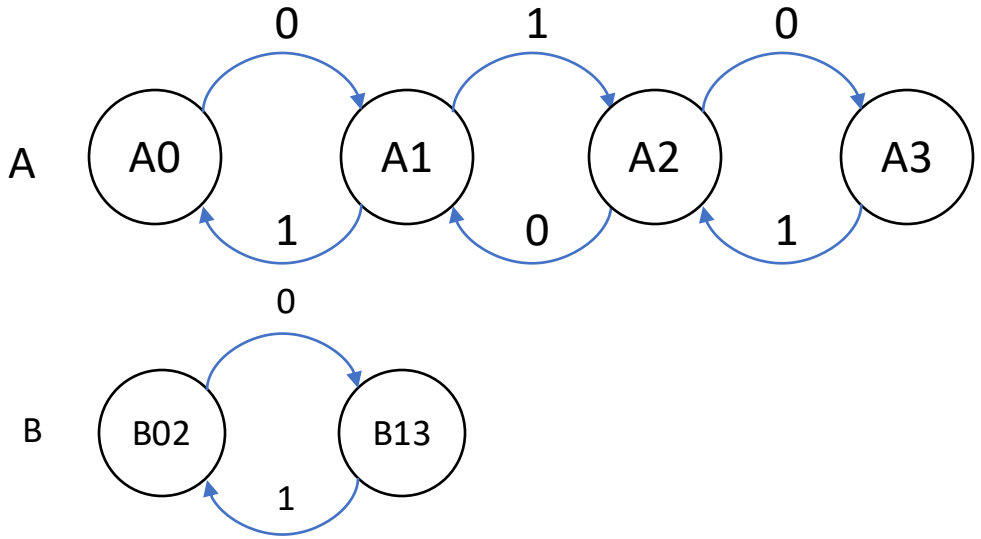




B **simulates** A and vice versa.
A and B are **bisimilar**.

C simulates both A and B.
C is an abstraction of both A and B.

How to prove B simulates A?



Show there exists a **simulation relation** from states of A to states of B. Say, $R = ((A0, B02), (A2, B02), (A1, B13), (A3, B13))$

Show that for every transition $Ai \rightarrow_A Ai'$ and $(Ai, Bj) \in R$ there exists Bj' such that

1. $Bj \rightarrow_B Bj'$
2. $(Ai', Bj') \in R$
3. $Trace(Bj \rightarrow_B Bj') = Trace(Ai \rightarrow_A Ai')$

Forward simulation relation

Consider a pair of automat $\mathcal{A}_1 = \langle Q_1, \Theta_1, A_1, D_1 \rangle$ 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 $Traces_1 \subseteq Traces_2$.

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

Proof.

Consider any trace β of $\mathcal{A}_1$. There must exist a corresponding execution α such that $trace(\alpha) = \beta$

We inductively construct a corresponding execution α' of $\mathcal{A}_2$ such that $trace(\alpha') = \beta$

$\exists q_0'$ such that $\alpha.fstate\ R\ q_0'$ and $q_0' \in \Theta_2$ [by start condition]

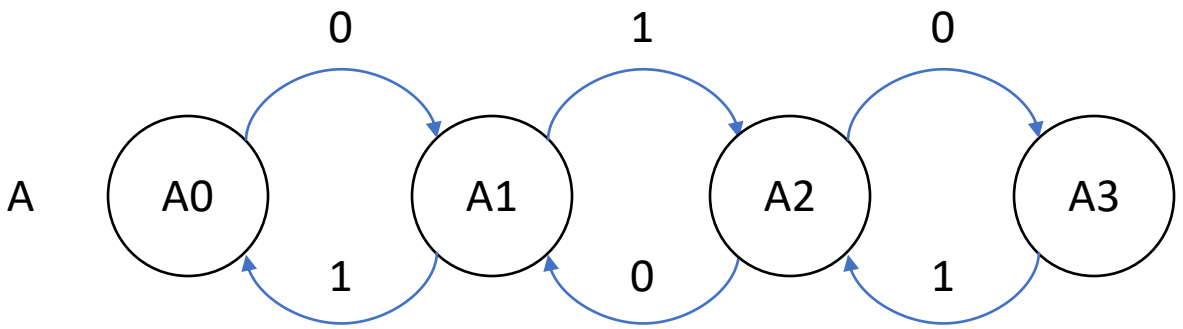
Suppose we have already constructed prefix α of length i ending in q_i and q_i' such that $q_i\ R\ q_i'$ and the traces corresponding to the prefixes are equal.

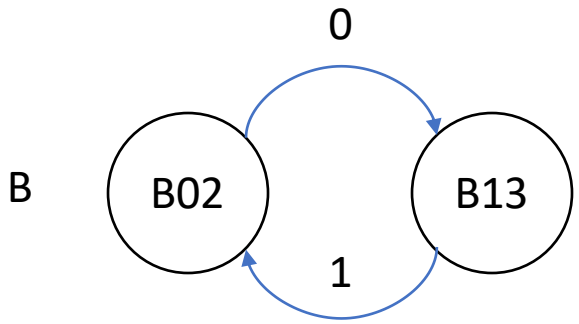
For any $q_i \rightarrow^{a_1}_{\mathcal{A}_1} q_{i+1}$ such that $q_i\ R\ q_i'$ there exists

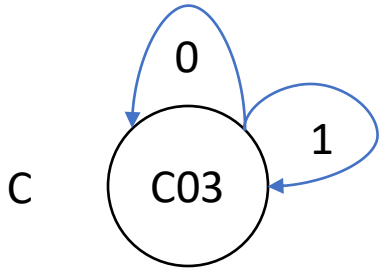
$q_i' \rightarrow^{a_2}_{\mathcal{A}_2} q_{i+1}'$ such that $q_{i+1} \mathrm{R}\, q'_{i+1}$ and

$Trace(q_i, a_1, q_{i+1}) = Trace(q_i', a_2, q_{i+1}')$ [by step condition]

Finite state examples







Check that A also simulates B and that C simulates both A and B.

Therefore, $\text{Traces}_A = \text{Traces}_B \subseteq Traces_C$?

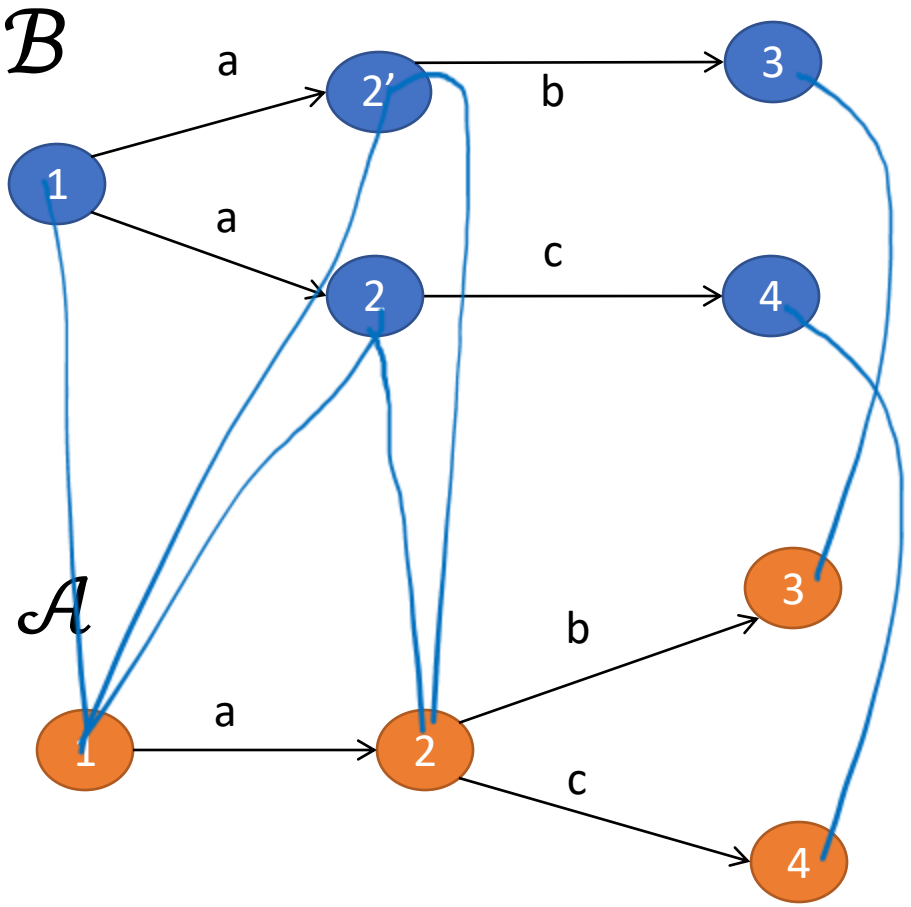
Does A simulate C?

A Simulation Example

- $\mathcal{A}$ is an implementation of $\mathcal{B}$
- Is there a forward simulation from $\mathcal{A}$ to $\mathcal{B}$?

- Consider the forward simulation relation

- $\mathcal{A} : 2 \rightarrow_c 4$ cannot be simulated by $\mathcal{B}$ from 2' although (2,2') are related.



Simulations for hybrid systems

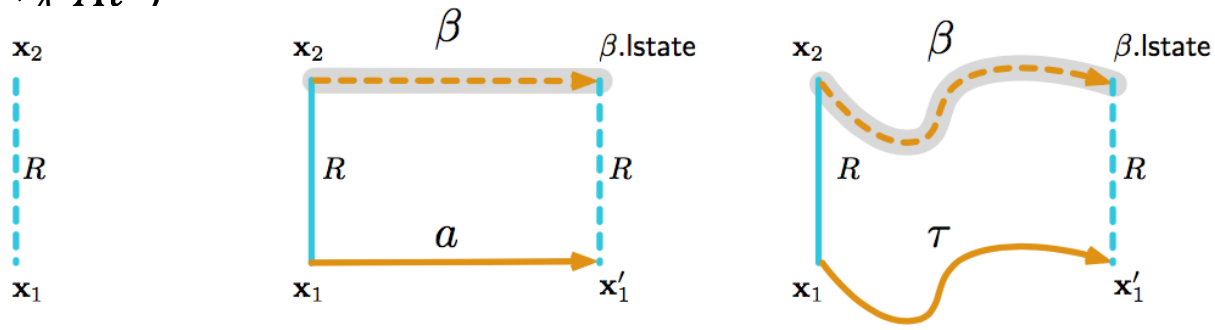
Forward simulation relation from $\mathcal{A}_1$ to $\mathcal{A}_2$ is a relation R $\subseteq val(X_1) \times val(X_2)$ such that

1. For every $x_1 \in \Theta_1$ there exists $x_2 \in \Theta_2$ such that x_1 R x_2
2. For every $x_1 \rightarrow_{a_1} x_1' \in \mathcal{D}$ and x_2 such that x_1 R x_2, there exists x_2' such that
   - $x_2 \rightarrow_{a_1} x_2'$ and
   - x_1' R x_2'
3. For every $\tau_1 \in \mathcal{T}_1$ and x_2 such that $\tau_1.fstate$ R x_2, there exists $\tau_2 \in \mathcal{T}_2$ that
   - $x_2 = \tau_2.fstate$ and
   - x_1' R $\tau_2.lstate$
   - $\tau_2.\mathrm{dom} = \tau_1.dom$

Theorem. If there exists a forward simulation relation from hybrid automaton $\mathcal{A}_1$ to $\mathcal{A}_2$ then for every execution of $\mathcal{A}_1$ there exists a corresponding execution of $\mathcal{A}_2$.

Simulation relations for hybrid automata

- Recall condition 3 in definition of simulation relation: $Trace(Bj \rightarrow_B Bj') = Trace(Ai \rightarrow_A Ai')$



- Hybrid automata have transitions and trajectories
- Different types of simulation depending on different notions for "Trace"
  - Match for all variable values, action names, and time duration of trajectories (abstraction)
  - Match variables but not time (time abstract simulation)
  - Match a subset (external) of variables and actions (trace inclusion)
  - Match single action/trajectory of A with a sequence of actions and trajectories of B

Timer simulates Ball (w.r.t. timing of bounce actions)

Automaton Ball(c,v_0,g)
  variables:
    x: Reals := 0
    v: Reals := v_0
  actions: bounce
    transitions:
      bounce
        pre *x = 0 /\ v < 0*
        eff v := -cv
    trajectories:
        evolve d(x) = v; d(v) = -g
        invariant $\boldsymbol{x \geq 0}$

Automaton Timer(c, v_0, g)
  variables: analog
  timer: Reals := $2v_0/g$,
  n:Naturals=0;
  actions: bounce
    transitions:
      bounce
        pre *timer = 0*
        eff n:=n+1; timer := $\frac{2v_0}{gc^n}$
    trajectories:
        evolve d(timer) = -1
        invariant timer ≥ 0

Some nice properties of Forward Simulation

Let $\mathcal{A}, \mathcal{B}, \text{and } \mathcal{C}$ be comparable TAs. If R_1 is a forward simulation from $\mathcal{A}$ to $\mathcal{B}$ and R_2 is a forward simulation from $\mathcal{B}$ to $\mathcal{C}$, then $R_1 \circ R_2$ is a forward simulation from $\mathcal{A}$ to $\mathcal{C}$

$\mathcal{A}$ implements $\mathcal{C}$

The **implementation relation** is a preorder of the set of all (comparable) hybrid automata

(A preorder is a reflexive and transitive relation)

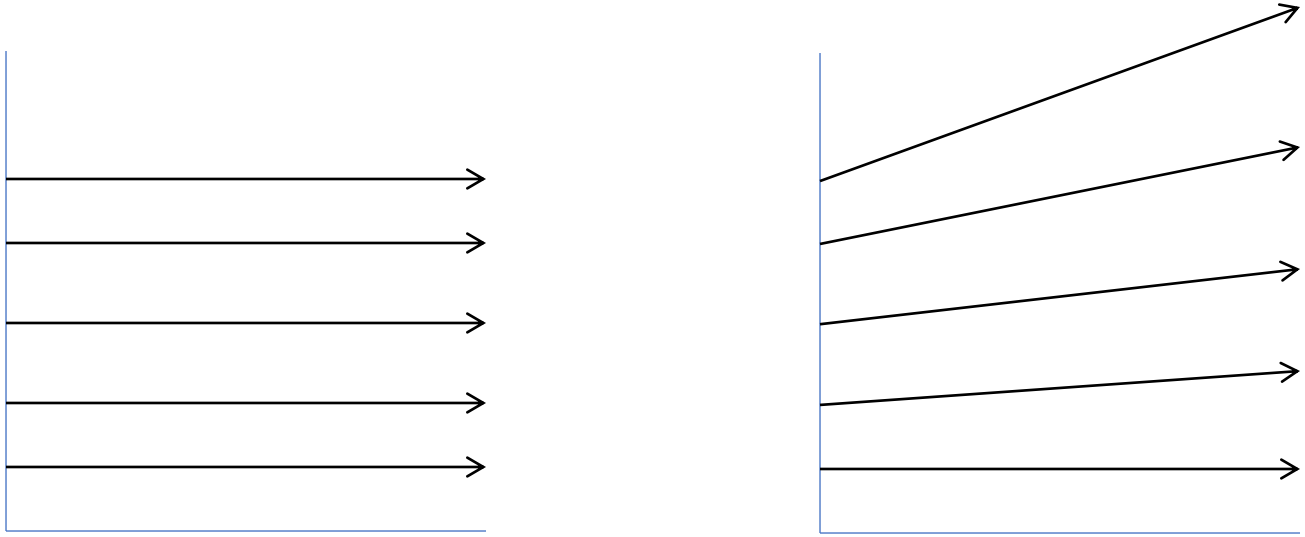
If R is a forward simulation from $\mathcal{A}$ to $\mathcal{B}$ and R^{-1} is a forward simulation from $\mathcal{B}$ to $\mathcal{A}$ then R is called a **bisimulation** and $\mathcal{B}$ are $\mathcal{A}$ **bisimilar**

Bisimilarity is an **equivalence relation**

(reflexive, transitive, and symmetric)

Remark on Simulations and Stability

Stability not preserved by ordinary simulations and bisimulations [Prabhakar, et. al 15]



Stability Preserving Simulations and Bisimulations for Hybrid Systems, Prabhakar, Dullerud, Viswanathan IEEE Trans. Automatic Control 2015

Backward Simulations

Backward simulation relation from $\mathcal{A}_1$ to $\mathcal{A}_2$ is a relation $R \subseteq Q_1 \times Q_2$ such that

1. If $x_1 \in \Theta_1$ and $x_1 \, R \, x_2$ then $x_2 \in \Theta_2$ such that
2. If $x'_1 \, R \, x'_2$ and $x_1 \text{—}a\text{→} x_1'$ then
   - $x_2 \text{–}\boldsymbol{\beta}\text{→} x_2'$ and
   - $x_1 \, R \, x_2$
   - $\text{Trace}(\boldsymbol{\beta}) = a_1$
3. For every $\boldsymbol{\tau} \in \mathcal{T}$ and $x_2 \in Q_2$ such that $x_1' \, R \, x_2'$, there exists x_2 such that
   - $x_2 \text{–}\boldsymbol{\beta}\text{→} x_2'$ and
   - $x_1 \, R \, x_2$
   - $\text{Trace}(\boldsymbol{\beta}) = \boldsymbol{\tau}$

Theorem. If there exists a backward simulation relation from $\mathcal{A}_1$ to $\mathcal{A}_2$ then $\text{ClosedTraces}_1 \subseteq \text{ClosedTraces}_2$