ECE/CS 584: Hybrid Automaton Modeling Framework Invariance, Abstractions, Simulation

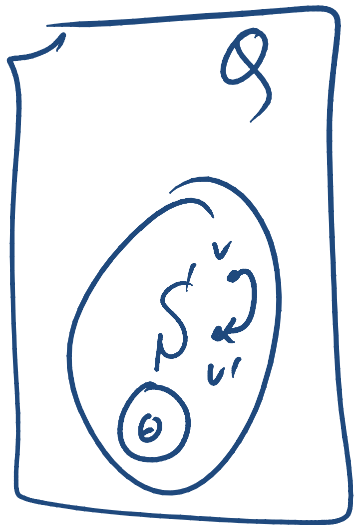
Lecture 04

Sayan Mitra

Plan for Today

- Invariants (continued)
- Abstraction
- Simulation relations

Inductive Invariants

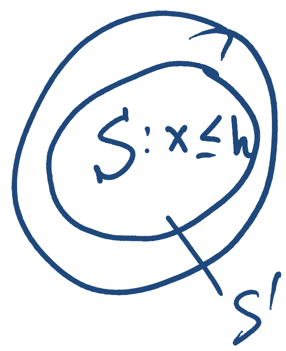


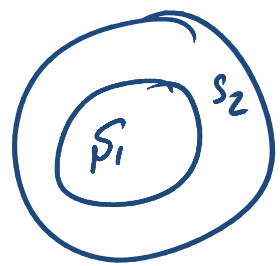
- Given a hybrid automaton $\mathcal{A}$= $(X, Q, \Theta, E, H, \mathcal{D}, \mathcal{T})$
- An S ⊆ Q is an **invariant** if $\boldsymbol{Reach_{\mathcal{A}}}$ ⊆ S
- An invariant S is **inductive** if for any v ∈ S
  - If v—a→ v' then v' ∈ S
  - If v—τ→ v' then v' ∈ S
- Theorem: For any set of states S if
  1. for any v ∈ Θ start state, v ∈ S
  2. If v ∈ S and v—a→ v' then v' ∈ S
  3. If v ∈ S and v—τ→ v' then v' ∈ S

  Then Reach$_{\mathcal{A}}$ ⊆ S
- Proof rule for establishing an inductive invariant S
- Checking an inductive invariant is relatively simple
- Finding useful invariants is in general more involved

Invariants and Inductive Invariants

- All invariants inductive? NO
  - Examples: $x \leq h$ (not inductive)
  - $x \leq h \wedge v^2 = 2g(h-x)$





S_1 is stronger than S_2

Pre and Post Computations

- For a given set of states $Q' \subseteq Q$, and action $a \in A$
  - $Post\_trans(Q', a) = \{ v' \mid \exists\, v \in Q', v \text{—}a\rightarrow v' \}$
  - $Post\_trans(Q', A') = \{ v' \mid \exists\, v \in Q', a \in A,\ v \text{—}a\rightarrow v' \}$
  - $Post\_taj(Q') = \{ v' \mid \exists\, v \in Q', \tau \in T, v \text{—}\tau\rightarrow v' \}$
  - $Post(Q') = Post\_trans(Q', A) \cup Post\_taj(Q')$

- **Theorem:** S is an inductive invariant iff it is a fixpoint of Post() and it contains Θ.

  - $Pre\_trans(Q', A') = \{ v \mid \exists\, v' \in Q', a \in A,\ v \text{—}a\rightarrow v' \}$
  - $Pre\_taj(Q') = \{ v \mid \exists\, v's \in Q', \tau \in T, v \text{—}\tau\rightarrow v' \}$
  - $Pre(Q') = Pre\_trans(Q', A) \cup Pre\_taj(Q')$

Abstractions

- Invariants overapproximate the set of reachable states
- E.g. “height is always less than h”
- Abstractions overapproximate executions
- E.g. “there is a bounce every c^n seconds”

Pablo Picasso, *Portrait of Gertrude Stein*, 1906, MOMA, New York. When someone commented that Stein didn't look like her portrait, Picasso replied, "She will". *From Wikipedia.*

Abstraction and Implementation ($\leq$)

- $\mathcal{A}_1$ and $\mathcal{A}_2$ are **comparable** if they have the same external interface, i.e., $E_1 = E_2$
- For two comparable automata, $\mathcal{A}_1$ **implements** $\mathcal{A}_2$ if $Traces_1 \subseteq Traces_2$
- $\mathcal{A}_2$ is an **abstraction** of $\mathcal{A}_1$ if $Execs_1 \subseteq Execs_2$
- $\mathcal{A}_1$ is a **refinement** of $\mathcal{A}_2$

- Examples ?

Abstract Bounce

Concrete

Automaton Bouncingball(c, v_0, g)

variables: analog x: Reals := 0, v: Reals := v_0

actions: external bounce

transitions:

bounce

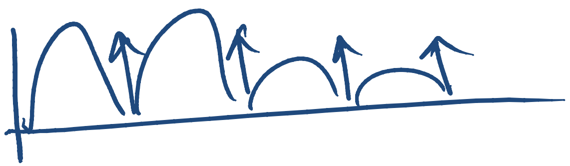
pre $x = 0 \wedge v < 0$

eff v := -cv

trajectories:

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

invariant $x \geq 0$



Abstract

Automaton BounceAbs(c,h,g)

variables: analog timer: Reals := v_0

n:Naturals=0;

actions: external bounce

transitions:

bounce

pre *timer = 0*

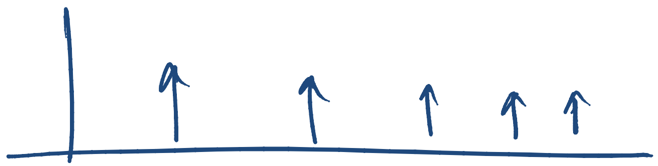
~~2 timer c~~ ^{h}

eff n:=n+1; timer := $\cancel{2vc^n}$ $\frac{2v_0^2c^n}{g}$

trajectories:

evolve d(timer) = -1

invariant timer $\geq$ 0



Simulations

- **Forward simulation** relation from $\mathcal{A}_1$ **to** $\mathcal{A}_2$ is a relation R $\subseteq Q_1 \times Q_2$ such that
  1. For every $\mathbf{x_1} \in \Theta_1$ there exists $\mathbf{x_2} \in \Theta_2$ such that $\mathbf{x_1}$ R $\mathbf{x_2}$
  2. For every $\mathbf{x_1} \xrightarrow{\mathbf{a_1}} \mathbf{x_1}' \in \mathcal{D}$ and $\mathbf{x_2} \in Q_2$ such that $\mathbf{x_1}$ R $\mathbf{x_2}$, there exists $\mathbf{x_2}'$ such that
     - $\mathbf{x_2} \xrightarrow{\boldsymbol{\beta}} \mathbf{x_2}'$ and
     - $\mathbf{x_1}'$ R $\mathbf{x_2}'$
     - **Trace($\boldsymbol{\beta}$)** = $\mathbf{a_1}$ trace(a_1)
  3. For every $\boldsymbol{\tau} \in \mathcal{T}$ and $\mathbf{x_2} \in Q_2$ such that $\mathbf{x_1}$ R $\mathbf{x_2}$, there exists $\mathbf{x_2}'$ such that
     - $\mathbf{x_2} \xrightarrow{\boldsymbol{\beta}} \mathbf{x_2}'$ and
     - $\mathbf{x_1}'$ R $\mathbf{x_2}'$
     - **Trace($\boldsymbol{\beta}$)** = $\boldsymbol{\tau}$

- **Theorem.** If there exists a forward simulation relation from $\mathcal{A}_1$ to $\mathcal{A}_2$ then Traces$_1 \subseteq$ Traces$_2$

Forward Simulation for Abstraction

- **Forward simulation** relation from $\mathcal{A}_1$ **to** $\mathcal{A}_2$ is a relation R $\subseteq Q_1 \times Q_2$ such that
  1. For every $\mathbf{x_1} \in \Theta_1$ there exists $\mathbf{x_2} \in \Theta_2$ such that $\mathbf{x_1}$ R $\mathbf{x_2}$
  2. For every $\mathbf{x_1} \xrightarrow{\mathbf{a_1}} \mathbf{x_1}' \in \mathcal{D}$ and $\mathbf{x_2} \in Q_2$ such that $\mathbf{x_1}$ R $\mathbf{x_2}$, there exists $\mathbf{x_2}'$ such that
     - $\mathbf{x_2} \xrightarrow{\boldsymbol{a_1}} \mathbf{x_2}'$ and
     - $\mathbf{x_1}'$ R $\mathbf{x_2}'$
  3. For every $\boldsymbol{\tau} \in \mathcal{T}$ and $\mathbf{x_2} \in Q_2$ such that $\mathbf{x_1}$ R $\mathbf{x_2}$, there exists $\mathbf{x_2}'$ such that
     - $\mathbf{x_2} \xrightarrow{\boldsymbol{\tau}} \mathbf{x_2}'$ and
     - $\mathbf{x_1}'$ R $\mathbf{x_2}'$

- **Theorem.** If there exists a forward simulation relation from $\mathcal{A}_1$ to $\mathcal{A}_2$ then $\text{Execs}_1 \subseteq \text{Execs}_2$

Characteristics of Hybrid Automata

- Guards, Transition relations, Invariants, DAEs written in some language
- These objects define the Transitions and Trajectories
- Transitions and trajectories define executions and traces
- Decidability of verification problem will depend on the choice of the language
- Nondeterministic
  - Transition choice
  - Transition relation
  - Branching trajectories
- External interface
  - External actions
  - Further partitioned into I/O actions
  - External variables available in the hybrid I/O automaton model

- Special cases $\dot{x} \in [a,b]$ $\dot{x} = a$ Not $\dot{x} = ax$ $\dot{x} = f(x,y)$
  - Deterministic HA $a \le x \le b \wedge c \le y \le d$
  - Rectangular HA $x := c$
  - (Alur-Dill) Timed Automata $x \in [a,b]$

    $\dot{x} = c$
  - X = Finitely many variables with finite types → Finite State Machine with Labeled transitions
  - X = n real valued variables {x1, …, xn} and A = {} D = {} → Dynamical System