ECE/CS 584: Hybrid Automaton Modeling Framework Simulations and Composition

Lecture 05

Sayan Mitra

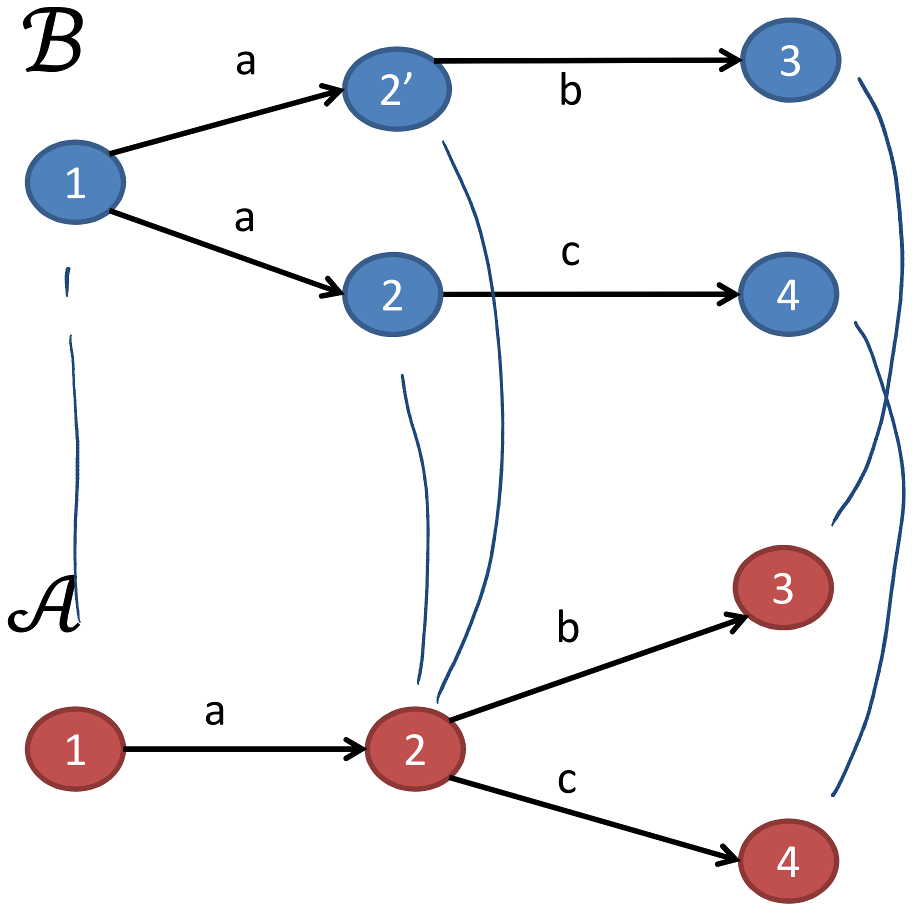
Plan for Today

- Abstraction and Implementation relations (continued)
- Composition
- Substitutivity
- Looking ahead
  - Tools: PVS, SpaceEx, Z3, UPPAAL
  - Decidable classes
  - Invariant generation
  - CEGAR
  - …

Some nice properties of Forward Simulation

- Let $\mathcal{A}, \mathcal{B},$ 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 $\mathcal{C}$ $\quad \{(a,b) \in R_1 \circ R_2 \mid \exists c \; (a,c) \in R_1 \; (c,b) \in R_2\}$
- The **implementation relation** is a preorder of the set of all (comparable) hybrid automata $\quad <, =, \leq$
  - (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)

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

{(R1,B1), (R2,B2), (R2,B2'), . . }

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

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 $\mathbf{x_1} \in \Theta_1$ and $\mathbf{x_1}$ R $\mathbf{x_2}$ then $\mathbf{x_2} \in \Theta_2$ such that
  2. If $\mathbf{x'_1}$ R $\mathbf{x'_2}$ and $x_2 \text{—}a\rightarrow x_2'$ then
     - $\mathbf{x_2} -\boldsymbol{\beta}\rightarrow \mathbf{x_2'}$ and
     - $\mathbf{x_1}$ R $\mathbf{x_2}$
     - **Trace($\boldsymbol{\beta}$)** = $\mathbf{a_1}$
  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
     - $\mathbf{x_2} -\boldsymbol{\beta}\rightarrow \mathbf{x_2'}$ and
     - $\mathbf{x_1}$ R $\mathbf{x_2}$
     - **Trace($\boldsymbol{\beta}$)** = $\boldsymbol{\tau}$

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

Composition of Hybrid Automata

- The parallel **composition** operation on automata enable us to construct larger and more complex models from simpler automata modules
- $\mathcal{A}_1$ to $\mathcal{A}_2$ are **compatible** if $\mathrm{X}_1 \cap \mathrm{X}_2 = \mathrm{H}_1 \cap \mathrm{A}_2 = \mathrm{H}_2 \cap \mathrm{A}_1 = \emptyset$
- Variable names are disjoint; Action names of one are disjoint with the internal action names of the other

Composition

- For compatible $\mathcal{A}_1$ and $\mathcal{A}_2$ their composition $\mathcal{A}_1 \,||\, \mathcal{A}_2$ is the structure $\boldsymbol{\mathcal{A}}$= $(X, Q, \Theta, E, H, \mathcal{D}, \mathcal{T})$
- $X = X_1 \cup X_2$ (disjoint union)
- $Q \subseteq val(X)$
- $\Theta = \{\, \boldsymbol{x} \in Q \,|\, \forall\, i \in \{1,2\}\colon\ \boldsymbol{x}.Xi \in \Theta_i\}$
- $H = H_1 \cup H_2$ *(disjoint union)*
- $E = E_1 \cup E_2$ *and* A= E ∪ H
- $(\boldsymbol{x}, a, \boldsymbol{x}') \in \mathcal{D}$ *iff*
  - $a \in H_1$ *and* $(\boldsymbol{x}.X_1, a, \boldsymbol{x}'.X_1) \in \mathcal{D}_1$ *and* $\boldsymbol{x}.X_2 = \boldsymbol{x}.X_2$
  - $a \in H_2$ *and* $(\boldsymbol{x}.X_2, a, \boldsymbol{x}'.X_2) \in \mathcal{D}_2$ *and* $\boldsymbol{x}.X_1 = \boldsymbol{x}.X_1$
  - *Else,* $(\boldsymbol{x}.X_1, a, \boldsymbol{x}'.X_1) \in \mathcal{D}_1$ *and* $(\boldsymbol{x}.X_2, a, \boldsymbol{x}'.X_2) \in \mathcal{D}_2$ $\leftarrow a \in E_1 \cup E_2$
- $\mathcal{T}$*:* set of **trajectories** for X
  - $\tau \in \mathcal{T}$ iff $\forall\, i \in \{1,2\},\ \tau.Xi \in \mathcal{T}_i$

Theorem. $\boldsymbol{\mathcal{A}}$ is also a hybrid automaton.

Example: Send || TimedChannel

Automaton Channel(b,M)
variables: queue: **Queue**[M,Reals] := {}
clock1: Reals := 0
actions: external send(m:M), receive(m:M)
transitions:
send(m)
pre true
eff queue := append(<m, clock1+b>, queue)
receive(m)
pre head(queue)[1] = m
eff queue := queue.tail
trajectories:
evolve d(clock1) = 1
stop when ∃ m, d, <m,d> ∈ queue
/\ clock1=d

Automaton PeriodicSend(u, M)
variables: analog clock: Reals := 0
states: True
actions: external send(m:M)
transitions:
send(m)
pre clock = u
eff clock := 0
trajectories:
evolve d(clock) = 1
stop when clock=u

Composed Automaton

Automaton SC(b,u)
  variables: queue: **Queue**[M,Reals] := {}
          clock_s, clock_c: Reals := 0
  actions: external send(m:M), receive(m:M)
  transitions:
    send(m)
    pre clock_s = u
    eff queue := append(<m, clock_c+b>, queue); clock_s := 0
    receive(m)
    pre head(queue)[1] = m
    eff queue := queue.tail
  trajectories:
    evolve d(clock_c) = 1; d(clock_s) = 1
    stop when
      (∃ m, d, <m,d> ∈ queue /\ clock_c=d)
      \/ (clock_s=u)

Some properties about composed automata

- Let $\mathcal{A} = \mathcal{A}_1 \,||\, \mathcal{A}_2$ and let α be an execution fragment of $\mathcal{A}$.
  - Then $\alpha_i = \alpha \lceil (A_i, X_i)$ is an execution fragment of $\mathcal{A}_i$
  - α is time-bounded iff both α_1 and α_2 are time-bounded
  - α is admissible iff both α_1 and α_2 are admissible
  - α is closed iff both α_1 and α_2 are closed
  - α is non-Zeno iff both α_1 and α_2 are non-Zeno
  - α is an execution iff both α_1 and α_2 are executions
- $\text{Traces}_{\mathcal{A}} = \{ \boldsymbol{\beta} \mid \boldsymbol{\beta} \lceil E_i \in \text{Traces}\ \mathcal{A}_i \}$ — (1)
- See examples in the TIOA monograph

Substitutivity

- **Theorem.** Suppose $\mathcal{A}_1$, $\mathcal{A}_2$ and $\mathcal{B}$ have the same external interface and $\mathcal{A}_1$, $\mathcal{A}_2$ are compatible with $\mathcal{B}$. If $\mathcal{A}_1$ implemens $\mathcal{A}_2$ then $\mathcal{A}_1 || \mathcal{B}$ implements $\mathcal{A}_2 || \mathcal{B}$

$Traces_{A_1||B} \subseteq Traces_{A_2||B}$

$A_1||B \leq A_2||B$

$\uparrow$

- Proof sketch.
- Define the simulation relation:

$R \subseteq Q_{A_1||B} \times Q_{A_2||B}$

$x \in Q_{A_1||B}$

$y \in Q_{A_2||B}$

(x,y) iff (1) $(x.X_1 , y.X_1) \in \underline{R_{A_1A_2}}$

(2) $x.X_B = y.X_B$

$y.X_1 \in \Theta_{A_2}$

$X_{A_1||B} = X_1 \cup X_B$

$$\begin{tikzcd} y \arrow[r, dashed, "\beta"] & y' \\ x \arrow[u, no head] \arrow[r, "a"'] & x' \arrow[u, dashed, no head] \end{tikzcd}$$

<u>Case1</u> if $a \in H_1$

$\beta \triangleq a$

$(x'.X_1 , y'.X_1) \in R_{A_1A_2}$

$x'.X_B = y'.y_B$ (by Comp)

Substutivity

- Theorem. Suppose $\mathcal{A}_1$ $\mathcal{A}_2$ $\mathcal{B}_1$ and $\mathcal{B}_2$ are HAs and $\mathcal{A}_1$ $\mathcal{A}_2$ have the same external actions and $\mathcal{B}_1$ $\mathcal{B}_2$ have the same external actions and $\mathcal{A}_1$ $\mathcal{A}_2$ is compatible with each of $\mathcal{B}_1$ and $\mathcal{B}_2$
- If $\mathcal{A}_1$ $\mathrm{implements}$ $\mathcal{B}_1$ and $\mathcal{A}_2$ $\mathrm{implements}$ $\mathcal{B}_2$ then $\mathcal{A}_1$ || $\mathcal{B}_1$ implements $\mathcal{A}_2||\mathcal{B}_2$.

- Proof. $\mathcal{A}_1$ || $\mathcal{B}_1$ implements $\mathcal{A}_2||\mathcal{B}_1$

  $\mathcal{A}_2||\mathcal{B}_1$ implements $\mathcal{A}_2||\mathcal{B}_2$

  By transitivity of implementation relation

  $\mathcal{A}_1$ || $\mathcal{B}_1$ implements $\mathcal{A}_2||\mathcal{B}_2$

- Theorem. $\mathcal{A}_1 \mid\mid \mathcal{B}_2$ implements $\mathcal{A}_2 \mid\mid \mathcal{B}_2$ and $\mathcal{B}_1$ implements $\mathcal{B}_2$ then $\mathcal{A}_1 \mid\mid \mathcal{B}_1$ implements $\mathcal{A}_2 \mid\mid \mathcal{B}_2$.

$\beta \in trace_{A_1 \| B_1}$

By (1) $\beta \lceil A_1 \in trace_{A_1}$ & $\beta \lceil B_1 \in trace_{B_1}$

B_1 implements $B_2 \Rightarrow \beta \lceil B_1 \in trace_{B_2}$

B_1, B_2 same interface $\beta \lceil B_1 = \beta \lceil B_2 \in trace_{B_2}$

$\beta \lceil B_2 \in trace_{B_2}$ & $\beta \lceil A_1 \in trace_{A_1} \Rightarrow$ (1) $\beta \in trace_{A_1 \| B_2}$

Summary

- Implementation Relation
  - Forward and Backward simulations
- Composition
- Substitutivity