EL2450 Hybrid and Embedded Control

Lecture 12: Verification of hybrid systems

- Reachability for hybrid automata
- Bisimulations of hybrid systems

Today's Goal

You should be able to

- do reachability analysis for hybrid automata
- state when transitions systems are bisimilar
- find a bisimulation quotient for a transition system

Recap: Verification

- **Prove** that a system fulfills certain property
- Based on a mathematical model and a computational tool

Recap: Safety of Transition Systems

For a transition system $T = (S, \Sigma, \to, S_0)$, let $B \subseteq S$ denote a "bad" set, i.e., a set of states that we don't want the system to enter. T is **safe** if

$$\mathrm{Reach}(S_0) \cap B = \emptyset,$$

where $\mathrm{Reach}(S_0)$ is the set of states that can be reached from S_0 by a sequence of transitions, i.e.,

$$\mathrm{Reach}(S_0) = \bigcup_{k=0,1,\dots} \mathrm{Post}^k(S_0).$$

- There is an algorithm for reach set computation
- The algorithm is guaranteed to terminate if the transition system is finite

Safety of Hybrid Automata

For a hybrid automaton $H = (Q, X, \mathrm{Init}, f, D, E, G, R)$, let $B \subseteq Q \times X$ denote a "bad" set, i.e., a set of states that we don't want the system to enter. H is **safe** if

$$\mathrm{Reach}_H(\mathrm{Init}) \cap B = \emptyset,$$

where $\mathrm{Reach}_H(\mathrm{Init}) \subseteq Q \times X$ is the set of states that can be reached by a solution of H from Init, i.e.,

$$(\bar{q}, \bar{x}) \in \mathrm{Reach}_H(\mathrm{Init})$$

if and only if there exists a solution $\chi = (\tau, q, x)$ of H such that

- $(q(0), x^0(0)) \in \mathrm{Init}$, and
- $(q(t), x^i(t)) = (\bar{q}, \bar{x})$ for some $t \in [\tau_i, \tau_i') \in \tau$.

Pre **and** Post **for Hybrid Automata**

The **predecessor operator** $\mathrm{Pre}(P), P \subseteq Q \times X$ for a hybrid automaton is

$$\mathrm{Pre}_H(P) = \{(q_p, x_p) : \exists (q, x) \in P, (q_p, x_p) \xrightarrow{t} (q, x) \text{ or } (q_p, x_p) \xrightarrow{e} (q, x)\}.$$

The **successor operator** $\mathrm{Post}(P)$ for a hybrid automaton is

$$\mathrm{Post}_H(P) = \{(q_p, x_p) : \exists (q, x) \in P, (q, x) \xrightarrow{t} (q_p, x_p) \text{ or } (q, x) \xrightarrow{e} (q_p, x_p)\}.$$

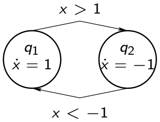
$$\mathrm{Reach}_H(\mathrm{Init}) = \bigcup_{k=0,1,\ldots} \mathrm{Post}_H^k(\mathrm{Init}).$$

Hybrid Automaton as a Transition System

A hybrid automaton is a transition system $T_H = (S, \Sigma, \rightarrow, S_0 = \text{Init})$ with interacting event-driven and time-driven evolution:

- $S = Q \times X$ and $(q, x) \in S$ denotes the state
- $\Sigma = \{g\} \cup \text{Time}$ where the generators $\{g\}$ causes the jumps and Time the continuous evolution
- $(q, x) \rightarrow (q', x')$ defines the event-driven and time-driven transitions

Example: $S = Q \times X$, with $Q = \{q_1, q_2\}$ and $X = \mathbb{R}$,
$\Sigma = \{g_1, g_2\} \cup \text{Time}$, g_1 corresponds to the event $x > 1$ and g_2 to $x < -1$,
$S_0 = \mathit{Init} = \{(q_1, x) \mid x \geq 0\}$



Reach Set for Hybrid Automata

Reach set for a hybrid automaton H can be computed in the transition system T_H

$$\mathrm{Reach}_H(\mathrm{Init}) = \mathrm{Reach}(S_0)$$

- T_H can be infinite and thus the reach set computation algorithm does not have to terminate
- **Idea:** To simplify T_H while preserving all information about its behaviors

Example

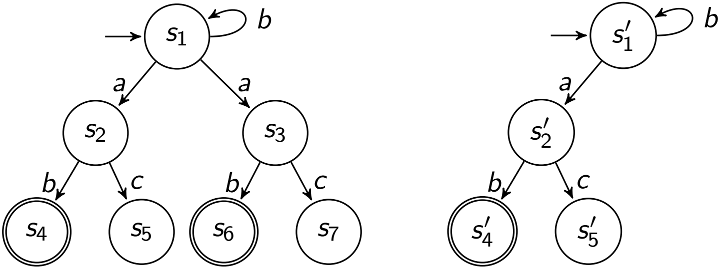


Figure: The transition systems $T_1 = (S, \Sigma, \rightarrow, S_0, S_F)$ (left) and $T_1' = (S', \Sigma, \rightarrow', S_0', S_F')$ (right).

- How formalize that T_1 (left) and T_1' (right) above have similar behaviour?

Relation

Given two sets A and B, a (binary) **relation** R from A to B is a subset of $A \times B$. We write $a\,R\,b$ if $(a, b) \in R$.

Example

$=$ is a relation from $\mathbb{N}$ to $\mathbb{N}$.

Simulation Relation

Given transition systems $T = (S, \Sigma, \rightarrow, S_0, S_F)$ and $T' = (S', \Sigma, \rightarrow', S'_0, S'_F)$. A relation $\sim \subseteq S \times S'$ is a **simulation relation** if

1. $\forall s \in S_0 \; (\exists s' \in S'_0. \; s \sim s')$
2. $s \sim s' \wedge s \in S_F \Rightarrow s' \in S'_F$
3. $\forall \sigma \in \Sigma \; (s \sim s' \wedge s \xrightarrow{\sigma} r \Rightarrow \exists r' \in S'$ such that $s' \xrightarrow{\sigma} r'$ and $r \sim r')$

We say that T' **simulates** T, i.e. that T is simulated by T', denoted by $T \sim T'$

Example: Simulation

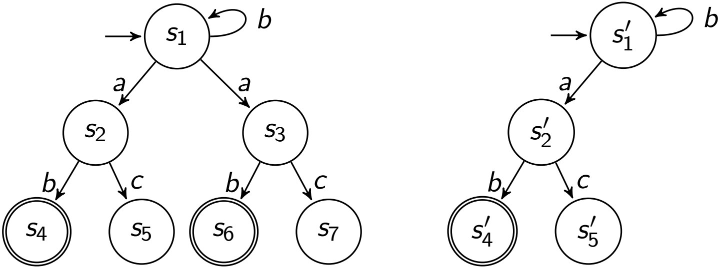


Figure: The transition systems T_1 (left) and T_1' (right).

Derive a simulation relation for T_1 and T_1'

Example: Simulation

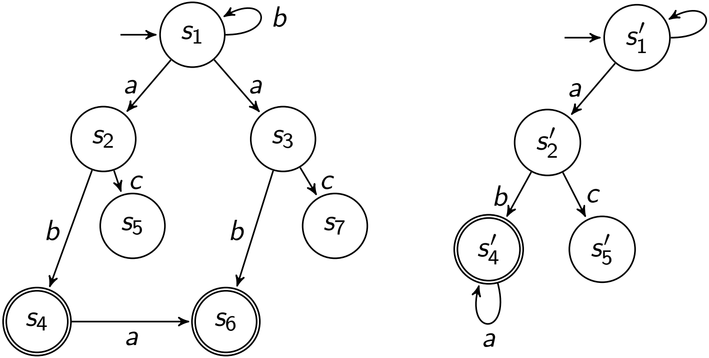


Figure: T_2 (left), T_2' (right).

Derive a simulation relation for T_2 and T_2'

Bisimulation Relation

If

- $\sim \subseteq S \times S'$ is a simulation relation from T to T' and
- $\sim' = \{(s', s) : (s, s') \in \sim\} \subseteq S' \times S$ is a simulation relation from T' to T,

then $\sim$ is a **bisimulation relation**.

- The existence of a bisimulation relation between two transition systems indicates that they are equivalent in some sense
- We say that T and T' are **bisimilar**

Examples: Bisimulation Relation

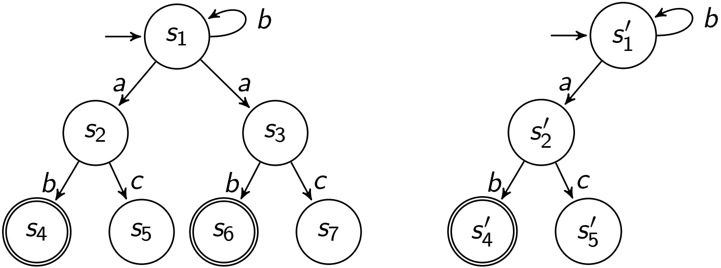


Figure: The transition systems T_1 (left) and T'_1 (right) are bisimilar.

Examples: Bisimulation Relation

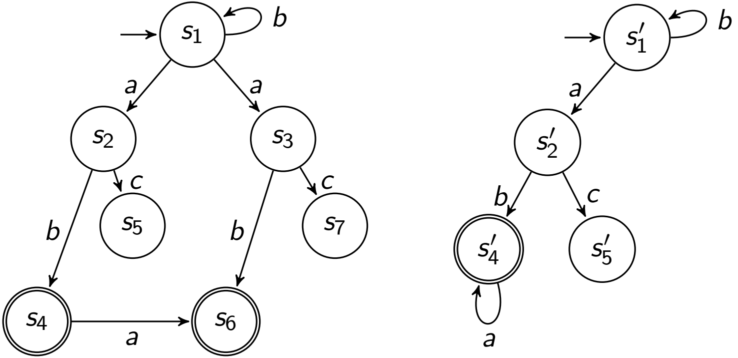


Figure: T_2 (left), T'_2 (right) are not bisimilar, because T'_2 simulates T_2, but T_2 does not simulate T'_2.

Examples: Bisimulation Relation

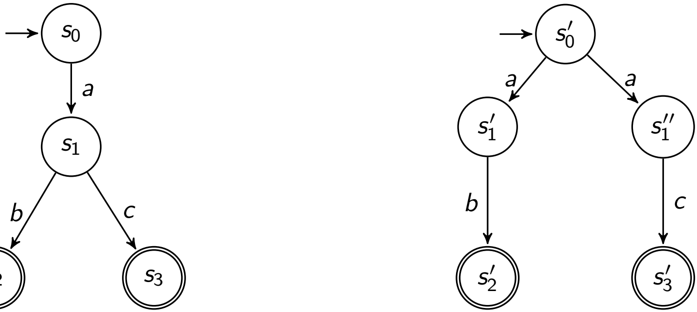


Figure: T_3 (left), T_3' (right).

Are T_3 and T_3' bisimilar?

Equivalence Relation

A relation $\equiv \subset S \times S$ is an **equivalence relation** if for all $s, s', s'' \in S$

1. $s \equiv s$ (reflexive)
2. $s \equiv s' \Rightarrow s' \equiv s$ (symmetric)
3. $s \equiv s'$ and $s' \equiv s'' \Rightarrow s \equiv s''$ (transitive)

Example

$=$ is an equivalence relation

Equivalence Class

Let $\equiv \subseteq S \times S$ be an equivalence relation. The **equivalence class** of $r \in S$ is defined as $[r] = \{s \in S \mid s \equiv r\}$.

Note

The equivalence classes constitute a partition of S, i.e., a collection of states $S/_{\equiv} = \{S_i\}_{i \in I}$ such that

$$S_i \cap S_j = \emptyset, \text{ for all } i \neq j$$

and

$$\bigcup_{i \in I} S_i = S$$

Quotient Transition System

Given a transition system $T = (S, \Sigma, \rightarrow, S_0, S_F)$ and a partition $S/_{\equiv} = \{S_i\}_{i \in I}$, the **quotient transition system** $\hat{T} = (\hat{S}, \Sigma, \hat{\rightarrow}, \hat{S}_0, \hat{S}_F)$ is defined as

1. $\hat{S} = S/_{\equiv}$
2. $\hat{s} \overset{\sigma}{\hat{\rightarrow}} \hat{s}'$ if $\exists s, s' \in S, s \in \hat{s}, s' \in \hat{s}'$, $s \xrightarrow{\sigma} s'$
3. $\hat{s} \in \hat{S}_0$ if $\exists s \in \hat{s}$, $s \in S_0$
4. $\hat{s} \in \hat{S}_F$ if $\exists s \in \hat{s}$, $s \in S_F$

Can we find a *finite* partition such that T and $\hat{T}$ are bisimilar?

Quotient Transition System

Given an equivalence relation $\equiv \subseteq S \times S$, the relation $\sim \subseteq S \times S/_{\equiv}$ such that $\sim = \{(s, [s]) | s \in S\}$ is a bisimulation relation between $T = (S, \Sigma, \rightarrow, S_0, S_F)$ and its quotient transition system $\hat{T}$ when:

1. S_F is a union of equivalence classes.
2. For each $P \subseteq S$ that is a union of equivalence classes, $\text{Pre}_\sigma(P)$ is a union of equivalence classes, for all $\sigma \in \Sigma$.

Thus, if T and $\hat{T}$ are bisimilar, all the information related to T can be derived from the evolution in $\hat{T}$

Bisimulation Quotient Algorithm

1. initialize $S/_{\equiv} = \big\{\{s \mid s \in S \setminus S_F\}, \{s \mid s \in S_F\}\big\}$
2. while $\exists P, P' \in S/_{\equiv}, \sigma \in \Sigma$, s.t. $\emptyset \neq P \cap Pre_\sigma(P') \neq P$

   $P_1 := P \cap Pre_\sigma(P')$
   $P_2 := P \setminus Pre_\sigma(P')$
   $S/_{\equiv} := (S/_{\equiv} \setminus \{P\}) \cup \{P_1, P_2\}$

   end while

If the algorithm terminates, it computes the *coarsest* quotient.
If T is infinite, the algorithm is not guaranteed to terminate.

Example: Bisimulation Relation

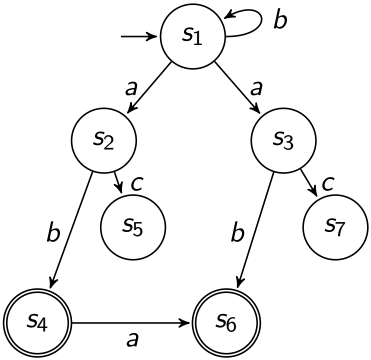


Figure: T_3

Find $\hat{T}_3$

Properties of Quotient Transition Systems

- For finite state systems an algorithm that always finds the coarsest bisimilar quotient system exists and always terminates
- Even if a transition system has infinite state space, its corresponding quotient transition system can be finite
- For time-triggered continuous-time systems (and hybrid systems) we cannot always find a finite partition

Reachability for Bisimilar Transition Systems

Given $T = (S, \Sigma, \rightarrow, S_0, S_F)$, the question whether

$$\text{Reach}(S_0) \cap S_F = \emptyset$$

in T is equivalent to the question whether

$$\text{Reach}(\hat{S}_0) \cap \hat{S}_F = \emptyset$$

in the bisimulation quotient transition system $\hat{T}$.

Safety Verification for Hybrid Automata

- A hybrid automaton $H = (Q, X, \text{Init}, f, D, E, G, R)$ and a bad set $B \subseteq Q \times X$ can be captured as a transition system $T_H = (S, \Sigma, \rightarrow, S_0, S_F = B)$, and the question whether

$$\text{Reach}_H(Init) \cap B = \emptyset$$

is equivalent to the question whether

$$\text{Reach}(S_0) \cap S_F = \emptyset$$

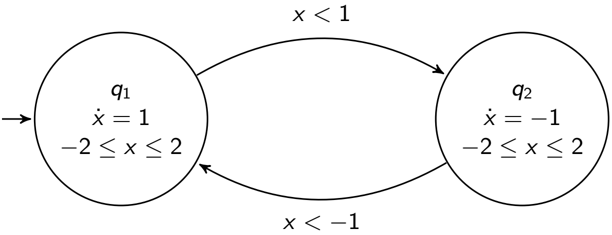
in T_H, which is equivalent to the question whether

$$\text{Reach}(\hat{S}_0) \cap \hat{S}_F = \emptyset$$

in the bisimulation quotient transition system $\hat{T}_H$.

- If $\hat{T}_H$ is finite, then the Reach Set Computation algorithm (ref. Lec. 9) terminates in a finite number of steps.

Example: Safety Verification of a Hybrid Automaton



Initial set: $Init = \{(q_1, 0)\}$
Bad set: $B = \{(q_2, x) \mid x \in [1, 2]\}$

Next Lecture

- Which classes of hybrid systems admit a finite bisimulation quotient transition system?
  - Reachability for timed automata, multi-rate automata, rectangular automata
- Over-approximations