Quantitative Verification
Chapter 8: Hybrid Automata

Jan Křetínský

Technical University of Munich

Winter 2016/17
Discrete, Continuous, and Hybrid Systems

- **Discrete System**
- **Continuous System**
- **Hybrid System**
Hybrid Automata
Consider a<bouncing ball system</b> dropped from height \( \ell \) and velocity 0. <br>
variables of interest: height of the ball \( x_1 \) and velocity of the ball \( x_2 \) <br>
flow function: a system of first-order ODEs <br>
\[
\begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= -g
\end{align*}
\]

Jump in the dynamics at impact! <br>\[
x_1' = x_1 \quad \text{and} \quad x_2' = -cx_2 \quad \text{where} \quad c \text{ is Restitution coefficient.}
\]
Modeling general hybrid systems: Hybrid automata

Let’s take again the thermostat as an example.

\[
\begin{align*}
\text{on} & \quad \dot{x} = -x + 50 \\
& \quad x \leq 23 \\
\text{off} & \quad \dot{x} = -x \\
& \quad x \geq 17 
\end{align*}
\]

\[x := 20\]
Hybrid Automata: Syntax

Definition (HA: Syntax)

A hybrid automaton is a tuple $\mathcal{H} = (M, M_0, \Sigma, X, \Delta, I, F, V_0)$ where:

- $M$ is a finite set of control modes including a distinguished initial set of control modes $M_0 \subseteq M$,
- $\Sigma$ is a finite set of actions,
- $X$ is a finite set of real-valued variable,
- $\Delta \subseteq M \times \text{pred}(X) \times \Sigma \times \text{pred}(X \cup X') \times M$ is the transition relation,
- $I : M \rightarrow \text{pred}(X)$ is the mode-invariant function,
- $F : M \rightarrow \text{pred}(X \cup \dot{X})$ is the mode-dependent flow function, and
- $V_0 \in \text{pred}(X)$ is the set of initial valuations.
Definition (HA: Syntax)

A hybrid automaton is a tuple $\mathcal{H} = (M, M_0, \Sigma, X, \Delta, I, F, V_0)$ where:

- $M$ is a finite set of control modes including a distinguished initial set of control modes $M_0 \subseteq M$,
- $\Sigma$ is a finite set of actions,
- $X$ is a finite set of real-valued variable,
- $\Delta \subseteq M \times \text{pred}(X) \times \Sigma \times \text{pred}(X \cup X') \times M$ is the transition relation,
- $I : M \rightarrow \text{pred}(X)$ is the mode-invariant function,
- $F : M \rightarrow \text{pred}(X \cup \dot{X})$ is the mode-dependent flow function, and
- $V_0 \in \text{pred}(X)$ is the set of initial valuations.

- A configuration $(m, \nu)$ and a timed action $(t, a)$
- A transition $((m, \nu), (t, a), (m', \nu'))$
  - solve flow ODE of mode $m$ with $\nu$ as the starting state $\nu \oplus F(m)t$.
  - invariant, guard, and jump conditions.
- A run or execution is a sequence of transitions
  
  $$(m_0, \nu_0), (t_1, a_1), (m_1, \nu_1), (t_2, a_2) \ldots$$
**Definition (HA: Semantics)**

The semantics of a HA $\mathcal{H} = (M, M_0, \Sigma, X, \Delta, I, F, V_0)$ is given as a state transition graph $T^\mathcal{H} = (S^\mathcal{H}, S_0^\mathcal{H}, \Sigma^\mathcal{H}, \Delta^\mathcal{H})$ where

- $S^\mathcal{H} \subseteq (M \times \mathbb{R}^{\left|X\right|})$ is the set of configurations of $\mathcal{H}$ such that for all $(m, \nu) \in S^\mathcal{H}$ we have that $\nu \in \llbracket I(m) \rrbracket$;
- $S_0^\mathcal{H} \subseteq S^\mathcal{H}$ s.t. $(m, \nu) \in S_0^\mathcal{H}$ if $m \in M_0$ and $\nu \in V_0$;
- $\Sigma^\mathcal{H} = \mathbb{R}_{\geq 0} \times \Sigma$ is the set of labels;
- $\Delta^\mathcal{H} \subseteq S^\mathcal{H} \times \Sigma^\mathcal{H} \times S^\mathcal{H}$ is the set of transitions such that $((m, \nu), (t, a), (m', \nu')) \in \Delta^\mathcal{H}$ if there exists a transition $\delta = (m, g, a, j, m') \in \Delta$ such that
  - $(\nu \oplus F(m) t) \in \llbracket g \rrbracket$;
  - $(\nu \oplus F(m) \tau) \in \llbracket I(m) \rrbracket$ for all $\tau \in [0, t]$;
  - $\nu' \in (\nu \oplus F(m) t)[j]$; and
  - $\nu' \in \llbracket I(m') \rrbracket$. 
Hybrid systems

Continuous systems with a **phased** operation:
- bouncing ball
- walking robots
- biological cell growth and division

Continuous systems **controlled by discrete** logic:
- thermostat
- chemical plants with valves, pumps
- control modes for complex systems, e.g. intelligent cruise control in automobiles, aircraft autopilot modes

**Coordinating** processes:
- air and ground transportation systems, e.g. swarms of micro-air vehicles
A **timed automaton** is a hybrid system where

- every variable is a clock,
- every jump condition is simple: comparison of variables to constants or the difference of two variables to a constant.

Reachability is decidable (PSPACE-complete) for TA.

(region construction)
A **timed automaton** is a hybrid system where
- every variable is a clock,
- every jump condition is simple: comparison of variables to constants or the difference of two variables to a constant.

Reachability is decidable (PSPACE-complete) for TA.
(region construction)

A **multirate timed system** extends TA with variables with arbitrary constant slope.
Reachability is undecidable for 2-rate timed systems.
(counter value \( n \leftrightarrow \text{accurate clock value } 1/2^n \))
A rectangular HA

- $x' \in [\text{min, max}]$
- Values of variables with different flows are never compared.
- Whenever the flow constraint of a variable changes, the variable is reset.

Reachability is decidable for rectangular HA.
Reachability is undecidable if either the second or the third constraint is violated.
A rectangular HA

- \( x' \in [\text{min, max}] \)
- Values of variables with different flows are never compared.
- Whenever the flow constraint of a variable changes, the variable is reset.

Reachability is decidable for rectangular HA.
Reachability is undecidable if either the second or the third constraint is violated.

A linear HA

- all initial, jump and flow conditions are written using linear predicates such that variables from \( X \) and \( X' \) never appear together in an atomic predicate, e.g., \( x + 2y' \leq 7, x = x' \) not allowed, \( x \leq 7 \land 3x' + 2y' = 8 \) is ok.

Bounded reachability is decidable for linear HA.
HA – Reachability II

A rectangular HA

- \( x' \in [\text{min}, \text{max}] \)
- Values of variables with different flows are never compared.
- Whenever the flow constraint of a variable changes, the variable is reset.

Reachability is decidable for rectangular HA.

Reachability is undecidable if either the second or the third constraint is violated.

A linear HA

- all initial, jump and flow conditions are written using linear predicates such that variables from \( X \) and \( X' \) never appear together in an atomic predicate, e.g., \( x + 2y' \leq 7, x = x' \) not allowed, \( x \leq 7 \land 3x' + 2y' = 8 \) is ok.

Bounded reachability is decidable for linear HA.

We want to approximate reachable sets for general HA.
Set-Based Reachability

Extending numerical simulation from numbers to sets

- account for nondeterminism
- exhaustive
- infinite time horizon

Downsides:

- only approximate for complex dynamics
- generally not scalable in # of variables
- trade-off between runtime and accuracy
Reachability Algorithm

One-step successors by time elapse from set of states $S$,

$$\text{Post}_C(S) = \{(\ell, \xi(\delta)) \mid \exists (\ell, x) \in S : (\ell, x) \xrightarrow{\delta, \xi} (\ell, \xi(\delta))\}.$$ 

One-step successors by jump from set of states $S$,

$$\text{Post}_D(S) = \{(\ell', x') \mid \exists (\ell', x') \in S, \exists \alpha \in \text{Lab} \cup \{\tau\} : (\ell, x) \xrightarrow{\alpha} (\ell', x')\}.$$
Reachability Algorithm

Compute sequence

\[ R_0 = \text{Post}_C(\text{Init}), \]
\[ R_{i+1} = R_i \cup \text{Post}_C(\text{Post}_D(R_i)). \]

If \( R_{i+1} = R_i \), then \( R_i = \) reachable states.

- may not terminate if states unbounded (counter)
- problem undecidable in general\(^6\)

Reachability of Affine Continuous Dynamics

\[ x(t) = e^{A\delta}x(0) + \int_0^t e^{A(\delta-\tau)}u(\tau)d\tau \]

- solution at discrete time steps
- cover flowpipe with convex sets \( \Omega_i \): approximation model
Representing of Convex Sets

- Approximation with Supporting Halfspaces
  - given template directions = outer polyhedral approximation

- **axis** ($\pm x_i$) → bounding box
  - $2n$ facets

- **octagonal** ($\pm x_i \pm x_j$) → bounding polytope
  - $2n^2$ facets

- **all directions** → exact set
Ball on String: Reachable States

(clip from SpaceEx output)
Example: Controlled Helicopter

- 28-dim model of a Westland Lynx helicopter
  - 8-dim model of flight dynamics
  - 20-dim continuous $H_\infty$ controller for disturbance rejection
  - stiff, highly coupled dynamics
Tools
SpaceEx Model Editor

Components = Hybrid Automata

- real-values variables
- ODE, linear DAE
SpaceEx Model Editor

Networks of Hybrid Automata
–templates
–hierarchy
SpaceEx Reachability Algorithms

**PHAVer**
- constant dynamics (LHA)
- formally sound and exact

**Support Function Algo**
- many continuous variables
- low discrete complexity

**Simulation**
- nonlinear dynamics
- based on CVODE

spaceex.imag.fr
SpaceEx Web Interface

Browser-based GUI
- 2D/3D output
- runs remotely
Conclusions

• Hybrid systems are easy to model with hybrid automata but difficult to analyze.

• Numerical simulation scales, but is not exhaustive and critical behavior may be missed.

• Set-based reachability covers all runs, sufficient for safety and bounded liveness.
  • computational cost,
  • scalable for piecewise affine dynamics

• Remaining challenges: trade-off between approximation accuracy and computational cost, scalable extension to nonlinear dynamics