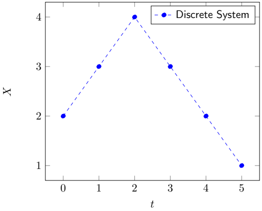
Formal Modeling Using Hybrid Automata

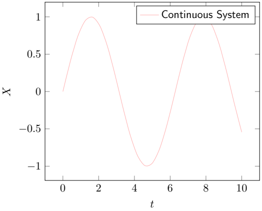
Ashutosh Trivedi

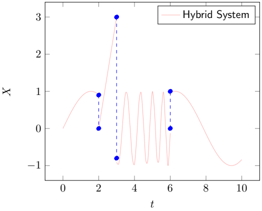
Department of Computer Science and Engineering,
IIT Bombay

CS620: New Trends in IT: Modeling and Verification of Cyber-Physical Systems
(14 August 2013)

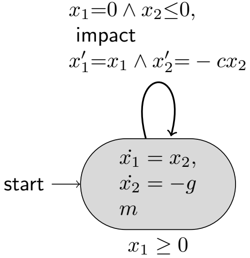
Discrete, Continuous, and Hybrid Systems







Hybrid Automata



- Consider a bouncing ball system dropped from height ℓ and velocity 0.
- variables of interest : height of the ball x_1 and velocity of the ball x_2
- flow function: a system of first-order ODEs

$$\dot{x_1} = x_2 \text{ and } \dot{x_2} = -g$$

- Jump in the dynamics at impact!
- $x_1' = x_1$ and $x_2' = -cx_2$ where c is Restituition coefficient.

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

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

- A configuration (m, ν) and a timed action (t, a)
- A transition $((m, \nu), (t, a), (m', \nu')$
  - solve flow ODE of mode m with ν as the starting state $\nu \oplus_{F(m)} t$.
  - invariant, guard, and jump conditions.
- A run or execution is a sequence of transitions

$$(m_0, \nu_0), (t_1, a_1), (m_1, \nu_1), (t_2, a_2) \ldots$$

Hybrid Automata: Semantics

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

- $S^{\mathcal{H}} \subseteq (M \times \mathbb{R}^{|X|})$ is the set of configurations of $\mathcal{H}$ such that for all $(m, \nu) \in S^{\mathcal{H}}$ we have that $\nu \in [\![I(m)]\!]$;
- $S_0^{\mathcal{H}} \subseteq S^{\mathcal{H}}$ 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
  - $(\nu \oplus_{F(m)} t) \in [\![g]\!]$;
  - $(\nu \oplus_{F(m)} \tau) \in [\![I(m)]\!]$ for all $\tau \in [0, t]$;
  - $\nu' \in (\nu \oplus_{F(m)} t)[j]$; and
  - $\nu' \in [\![I(m')]\!]$.

Hybrid Automata: Modeling Exercise

Some examples:

1. Leaking water-tank system
2. Leaking water-tank system with delayed switch
3. Gas-burner system
4. Light-bulb system with three modes- dim, bright, and off.
5. Job-shop scheduling problem
6. Rail-road crossing example

Job-Shop Scheduling Problem

A job-shop scheduling problem is given as a tuple:

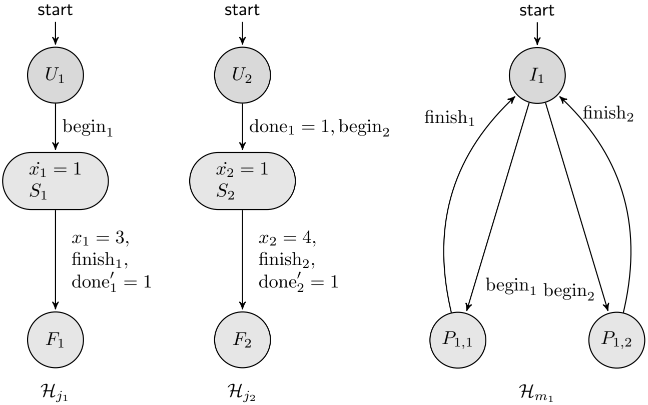
- A finite set $\mathbb{J} = \{j_1, \ldots, j_n\}$ of jobs
- A finite set $\mathbb{M} = \{m_1, \ldots, m_k\}$ of machines
- Strict precedence requirements between the jobs given as a partial order $\prec$ over the set of jobs in $\mathbb{J}$.
- A mapping $\zeta : \mathbb{J} \to 2^{\mathbb{M}}$ specifies the set of machines where a job can be executed,
- a function $\delta : \mathbb{J} \to \mathbb{R}_{\geq 0}$ specifying the time duration of a job.

We impose the following restrictions:

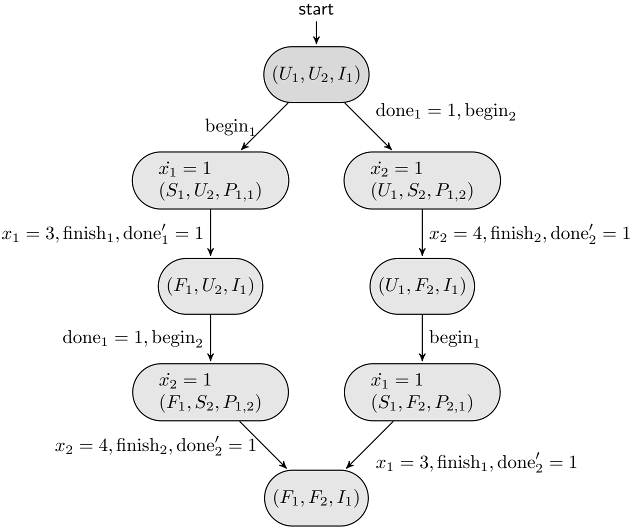
1. a job j can be executed iff all jobs in its precedence, $j{\downarrow} = \{j' \mid j' \prec j\}$, have terminated;
2. each machine $m \in \mathbb{M}$ can process atmost one job at a time; and
3. a job, once started, cannot be preempted.

Let's model the system using a network of hybrid automata!

Job-Shop Scheduling Problem



Job-Shop Scheduling Problem



$\mathcal{H}_{j_1} \otimes \mathcal{H}_{j_2} \otimes \mathcal{H}_{m_1}$

Composition of a Network of Hybrid Automata

Let $\mathcal{C} = \{\mathcal{H}^1, \mathcal{H}^2, \ldots, \mathcal{H}^n\}$ be a network of hybrid automata where for each $1 \leq i \leq n$ let $\mathcal{H}^i$ be $(M^i, M_0^i, \Sigma^i, X^i, \Delta^i, I^i, F^i, V_0^i)$.

The product automata $\mathcal{H}_1 \otimes \mathcal{H}_2 \otimes \cdots \mathcal{H}_n$ of $\mathcal{C}$ is defined as a hybrid automaton $H = (M, M_0, \Sigma, X, \Delta, I, F, V_0)$ where

- $M = M^1 \times M^2 \times \cdots M^n$,
- $M_0 = M_0^1 \times M_0^2 \times \cdots M_0^n$,
- $\Sigma = \Sigma^1 \cup \Sigma^2 \cup \ldots \Sigma^n$,
- $X = X^1 \cup X^2 \cup \ldots X^n$,
- $\Delta \subseteq (M \times \text{pred}(X) \times \Sigma \times \text{pred}(X \cup X') \times M)$ is defined such that $((m_1, \ldots, m_n), g, a, j, (m'_1, \ldots, m'_n)) \in \Delta$ if and only if:
  - For all i such that $a \notin \Sigma^i$ we have $m_i = m'_i$
  - For all i such that $a \in \Sigma^i$ we have corresponding transitions (m_i, g_i, a, j_i, m'_i)
  - The guard g and jump function j are defined as conjunct of all such g_i and j_i resp.
- I is such that $I(m_1, \ldots, m_n) = \wedge_{i=1}^n I^i(m_i)$;
- F is such that $F(m_1, \ldots, m_n) = \wedge_{i=1}^n F^i(m_i)$; and
- V_0 is such that $V_0 = \wedge_{i=1}^n V_0^i$.

Hybrid Automata: Modeling Exercise

Some examples:

1. Leaking water-tank system
2. Leaking water-tank system with delayed switch
3. Gas-burner system
4. Light-bulb system with three modes- dim, bright, and off.
5. Job-shop scheduling problem
6. Rail-road crossing problem