An Introduction to Hybrid Systems Modeling

Ashutosh Trivedi

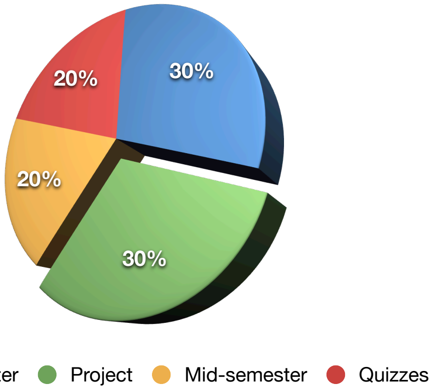
Department of Computer Science and Engineering,
IIT Bombay

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

Course overview

1. Formal Modeling of CPS
   - Discrete Dynamical Systems (Extended Finite State Machines)
   - Continuous Dynamical Systems (Ordinary Differential Equations)
   - Hybrid Dynamical Systems
     - Timed automata,
     - Hybrid automata,
     - PCDs, Multi-mode systems, and other decidable subclasses
2. Tools for modeling CPS
   - UPPAAL
   - HyTech
   - Stateflow/Simulink
3. Verification and Synthesis
   - Classical temporal logics LTL and CTL
   - Real-time extensions of these logics, in particular MTL
   - Model-Checking for timed and hybrid automata
   - Automatic Synthesis for CPS (satisfiability, controller-environment games, code-generations, etc.)

Grading



Dynamical Systems

Dynamical System: A system whose state evolves with time governed by a fixed set of rules or dynamics.

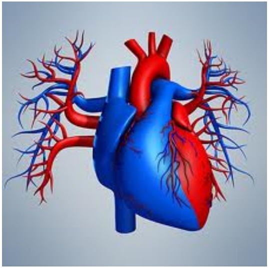
- state: valuation to variables (discrete or continuous) of the system
- time: discrete or continuous
- dynamics: discrete, continuous, or hybrid

Dynamical Systems

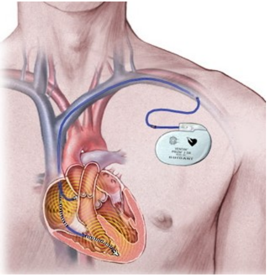
Dynamical System: A system whose state evolves with time governed by a fixed set of rules or dynamics.

- state: valuation to variables (discrete or continuous) of the system
- time: discrete or continuous
- dynamics: discrete, continuous, or hybrid

Discrete System



Continuous System



Hybrid Systems.

Dynamical Systems

Discrete Dynamical Systems

Most General Model for Dynamical Systems

Definition (State Transition Systems)

A state transition system is a tuple $\mathcal{T} = (S, S_0, \Sigma, \Delta)$ where:

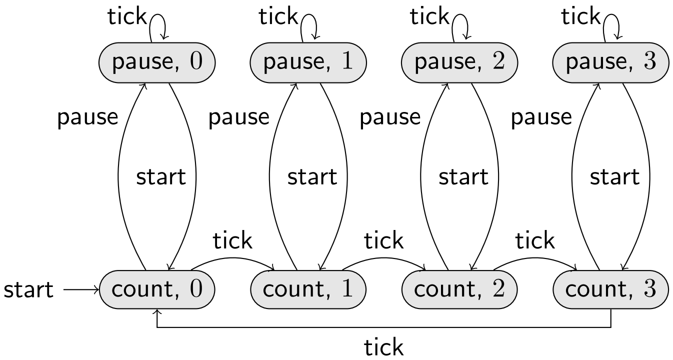
- S is a (potentially infinite) set of states;
- $S_0 \subseteq S$ is the set of initial states;
- Σ is a (potentially infinite) set of actions; and
- $\Delta \subseteq S \times \Sigma \times S$ is the transition relation;

Most General Model for Dynamical Systems

Definition (State Transition Systems)

A state transition system is a tuple $\mathcal{T} = (S, S_0, \Sigma, \Delta)$ where:

- S is a (potentially infinite) set of states;
- $S_0 \subseteq S$ is the set of initial states;
- Σ is a (potentially infinite) set of actions; and
- $\Delta \subseteq S \times \Sigma \times S$ is the transition relation;



State Transition Systems

Definition (State Transition Systems)

A state transition system is a tuple $\mathcal{T} = (S, S_0, \Sigma, \Delta)$ where:

- S is a (potentially infinite) set of states;
- $S_0 \subseteq S$ is the set of initial states;
- Σ is a (potentially infinite) set of actions; and
- $\Delta \subseteq S \times \Sigma \times S$ is the transition relation;

- Finite and countable state transition systems
- A finite run is a sequence

$$\langle s_0, a_1, s_1, s_2, s_2, \ldots, s_n \rangle$$

  such that $s_0 \in S_0$ and for all $0 \leq i < n$ we have that $(s_i, a_{i+1}, s_{i+1}) \in \Delta$.
- Reachability and Safe-Schedulability problems

State Transition Systems

Definition (State Transition Systems)

A state transition system is a tuple $\mathcal{T} = (S, S_0, \Sigma, \Delta)$ where:

- S is a (potentially infinite) set of states;
- $S_0 \subseteq S$ is the set of initial states;
- Σ is a (potentially infinite) set of actions; and
- $\Delta \subseteq S \times \Sigma \times S$ is the transition relation;

- Finite and countable state transition systems
- A finite run is a sequence

$$\langle s_0, a_1, s_1, s_2, s_2, \ldots, s_n \rangle$$

  such that $s_0 \in S_0$ and for all $0 \leq i < n$ we have that $(s_i, a_{i+1}, s_{i+1}) \in \Delta$.
- Reachability and Safe-Schedulability problems

We need efficient computer-readable representations of infinite systems!

Extended Finite State Machines

- Let X be the set of variables (real-valued) of the system
- let $|X| = N$.
- A valuation ν of X is a function $\nu : X \to \mathbb{R}$.
- We consider a valuation as a point in $\mathbb{R}^N$ equipped with Euclidean Norm.

Extended Finite State Machines

- Let X be the set of variables (real-valued) of the system
- let $|X| = N$.
- A valuation ν of X is a function $\nu : X \to \mathbb{R}$.
- We consider a valuation as a point in $\mathbb{R}^N$ equipped with Euclidean Norm.
- A predicate is defined simply as a subset of $\mathbb{R}^N$ represented (non-linear) algebraic equations involving X.
  - Non-linear predicates, e.g. $x + 9.8\sin(z) = 0$

Extended Finite State Machines

- Let X be the set of variables (real-valued) of the system
- let $|X| = N$.
- A valuation ν of X is a function $\nu : X \to \mathbb{R}$.
- We consider a valuation as a point in $\mathbb{R}^N$ equipped with Euclidean Norm.
- A predicate is defined simply as a subset of $\mathbb{R}^N$ represented (non-linear) algebraic equations involving X.
  - Non-linear predicates, e.g. $x + 9.8\sin(z) = 0$
  - Polyhedral predicates:

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n \sim k$$

  where $a_i \in \mathbb{R}$, $x_i \in X$, and $\sim = \{<, \leq, =, \geq, >\}$.

Extended Finite State Machines

- Let X be the set of variables (real-valued) of the system
- let $|X| = N$.
- A valuation ν of X is a function $\nu : X \to \mathbb{R}$.
- We consider a valuation as a point in $\mathbb{R}^N$ equipped with Euclidean Norm.
- A predicate is defined simply as a subset of $\mathbb{R}^N$ represented (non-linear) algebraic equations involving X.
  - Non-linear predicates, e.g. $x + 9.8\sin(z) = 0$
  - Polyhedral predicates:

    $$a_1x_1 + a_2x_2 + \cdots + a_nx_n \sim k$$

    where $a_i \in \mathbb{R}$, $x_i \in X$, and $\sim = \{<, \leq, =, \geq, >\}$.
  - Octagonal predicates

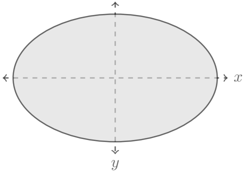
    $$x_i - x_j \sim k \text{ or } x_i \sim k$$

    where $x_i, x_j \in X$, and $\sim = \{<, \leq, =, \geq, >\}$.
  - Rectangular predicates

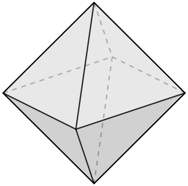
    $$x_i \sim k$$

    where $x_i \in X$, and $\sim = \{<, \leq, =, \geq, >\}$.
  - Singular Predicates $x_i = c$.

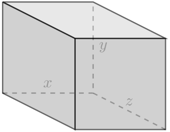
Poly-, Rect-, and Octa- Predicates



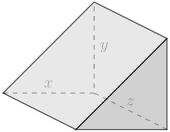
(a)



(b)

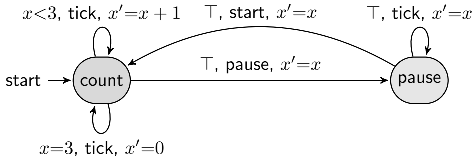


(b)



(c)

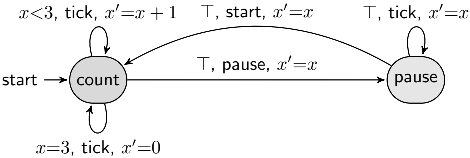
Extended Finite State Machines



Extended Finite State Machines (EFSMs):

- Finite state-transition systems coupled with a finite set of variables
- The valuation remains unchanged while system stays in a mode (state)
- The valuation changes during a transition when it jumps to the valuation governed by a predicate over $X \cup X'$ specified in the transition relation.
- Transitions are guarded by predicates over X
- Mode invariants
- Initial state and valuation

Extended Finite State Machines

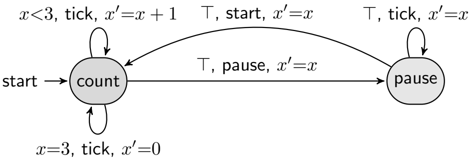


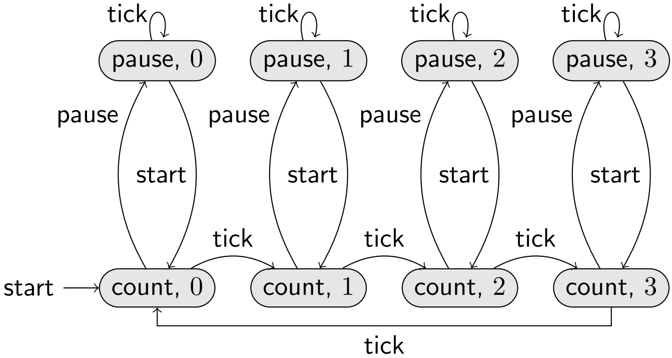
Definition (EFSM: Syntax)

An extended finite state machine is a tuple $\mathcal{M} = (M, M_0, \Sigma, X, \Delta, I, V_0)$ such that:

- 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, and
- $V_0 \in \mathrm{pred}(X)$ is the set of initial valuations.

EFSM: Semantics





EFSM: Semantics

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

- $S^{\mathcal{M}} \subseteq (M \times \mathbb{R}^{|X|})$ is the set of configurations of $\mathcal{M}$ such that for all $(m, \nu) \in S^{\mathcal{M}}$ we have that $\nu \in I(m)$;
- $S_0^{\mathcal{M}} \subseteq S^{\mathcal{M}}$ such that $(m, \nu) \in S^{\mathcal{M}}$ if $m \in M_0$ and $\nu \in V_0$;
- $\Sigma^{\mathcal{M}} = \Sigma$ is the set of labels;
- $\Delta^{\mathcal{M}} \subseteq S^{\mathcal{M}} \times \Sigma^{\mathcal{M}} \times S^{\mathcal{M}}$ is the set of transitions such that $((m, \nu), a, (m', \nu')) \in \Delta^{\mathcal{M}}$ if there exists a transition $\delta = (m, g, a, j, m') \in \Delta$ such that
  - the current valuation ν satisfies the guard of δ, i.e. $\nu \in g$;
  - the pair of current and next valuations (ν, ν') satisfies the jump constraint of δ, i.e. $(\nu, \nu') \in j$; and
  - the next valuation satisfies the invariant of the target mode of δ, i.e. $\nu' \in I(m')$.