



Introduction to Embedded Systems

Edward A. Lee & Prabal Dutta

UC Berkeley

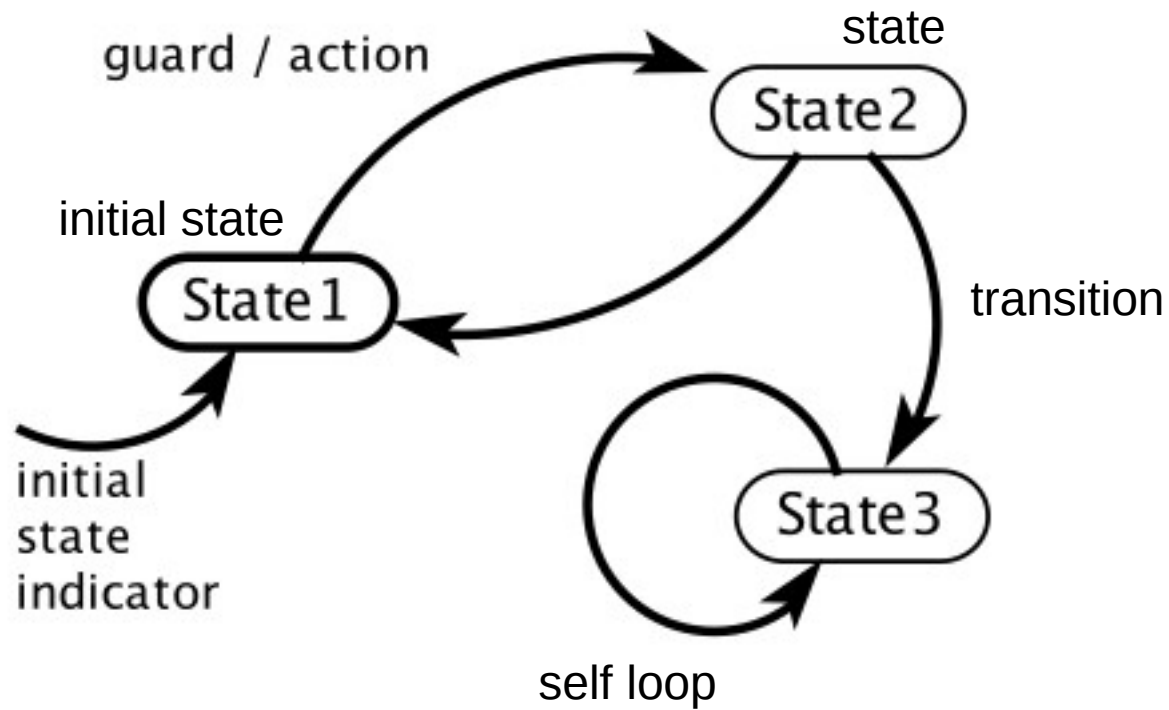
EECS 149/249A

Fall 2017

Chapter 3, 4: Extended Finite State Machines, Timed Automata, Hybrid Automata



Recall FSM Notation

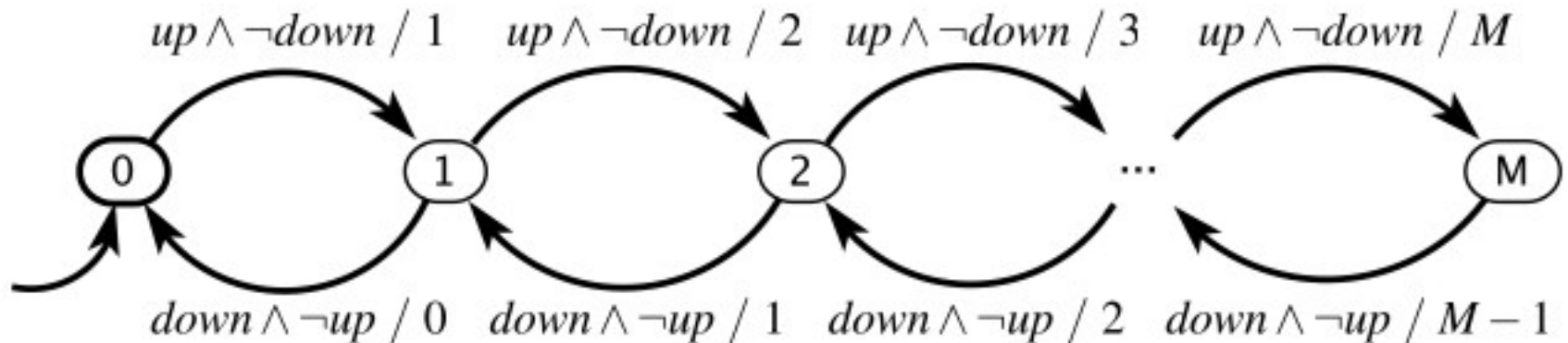


Garage Counter Example

Recall this example, which counts cars in a parking garage:

inputs: $up, down \in \{present, absent\}$

output $\in \{0, \dots, M\}$



Every state is represented separately!

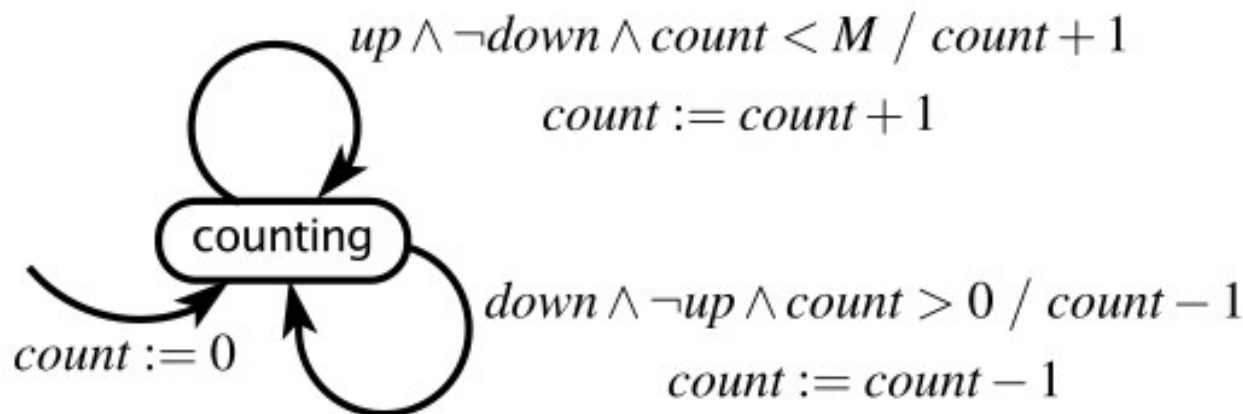
Extended State Machines

Extended state machines augment the FSM model with *variables* that may be read or written. E.g.:

variable: $count \in \{0, \dots, M\}$

inputs: $up, down \in \{present, absent\}$

output $\in \{0, \dots, M\}$



Question: What is the size of the state space?

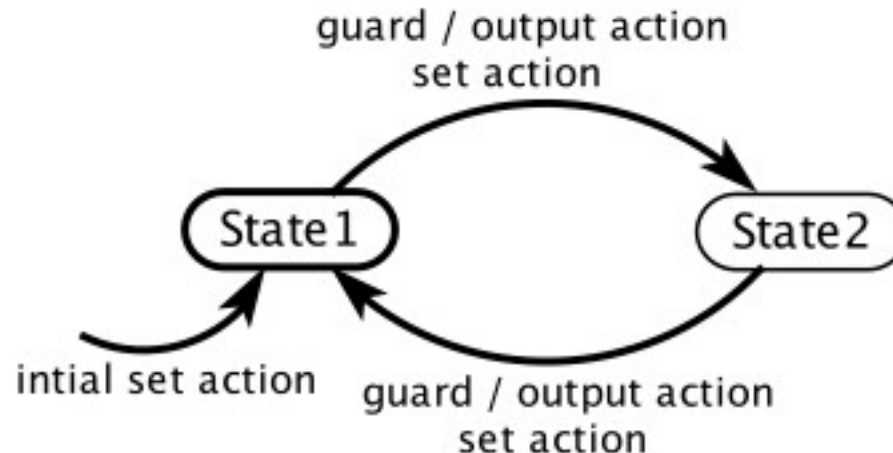
Extended State Machines

- ✦ State of an Extended State Machine = the current “bubble” **and** the value of all variables (internal data)
 - Same “bubble” but different variable value means different states
- ✦ Extended State Machines are useful for modeling time-based conditions:
 - Increment a variable every *tick* \Leftrightarrow measure passage of time
 - If the State Machine is *time-triggered* (e.g. driven by an external clock)

General Notation for Extended State Machines

We make explicit declarations of variables, inputs, and outputs to help distinguish the three.

variable declaration(s)
input declaration(s)
output declaration(s)

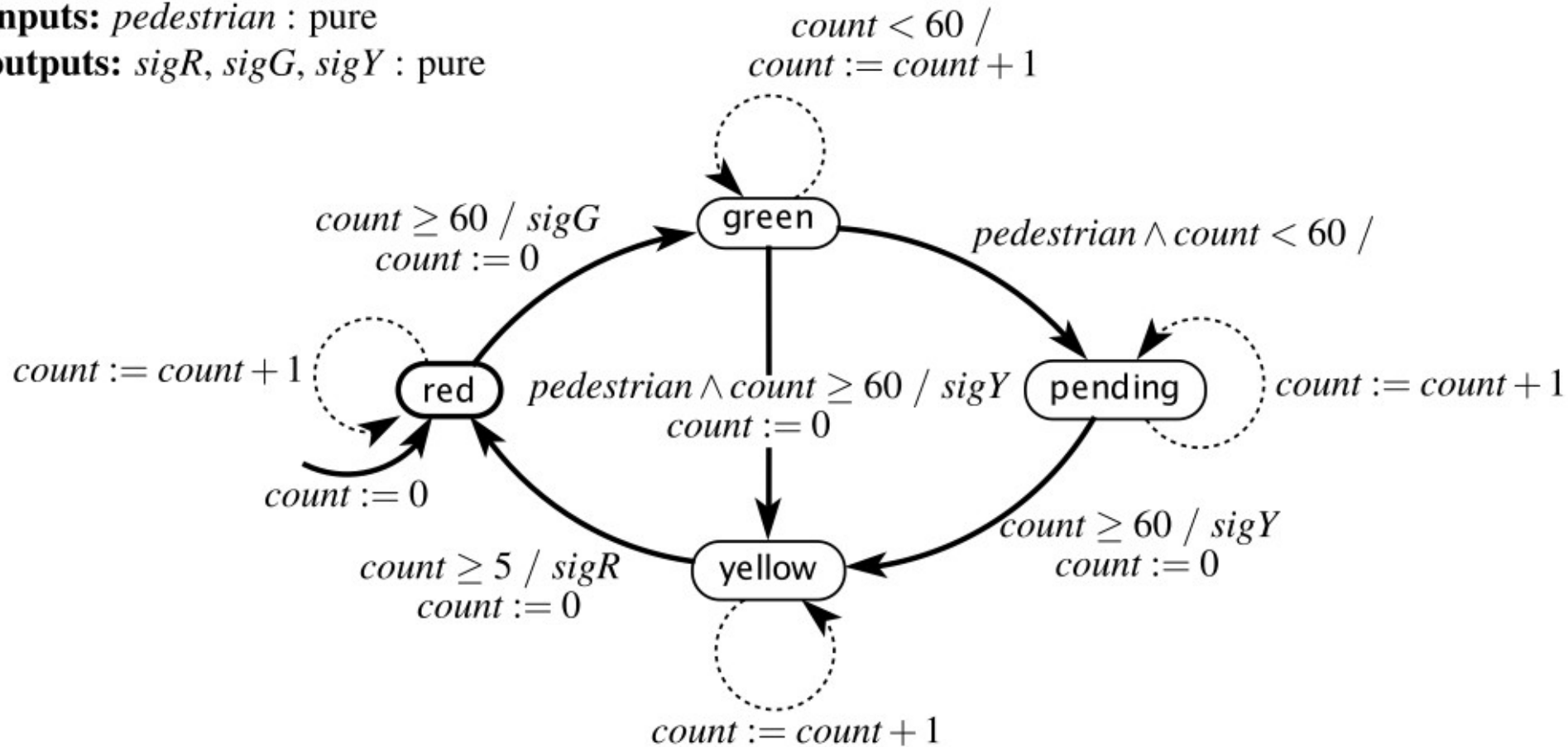


Extended state machine model of a traffic light controller at a pedestrian crossing:

variable: $count: \{0, \dots, 60\}$

inputs: $pedestrian : \text{pure}$

outputs: $sigR, sigG, sigY : \text{pure}$



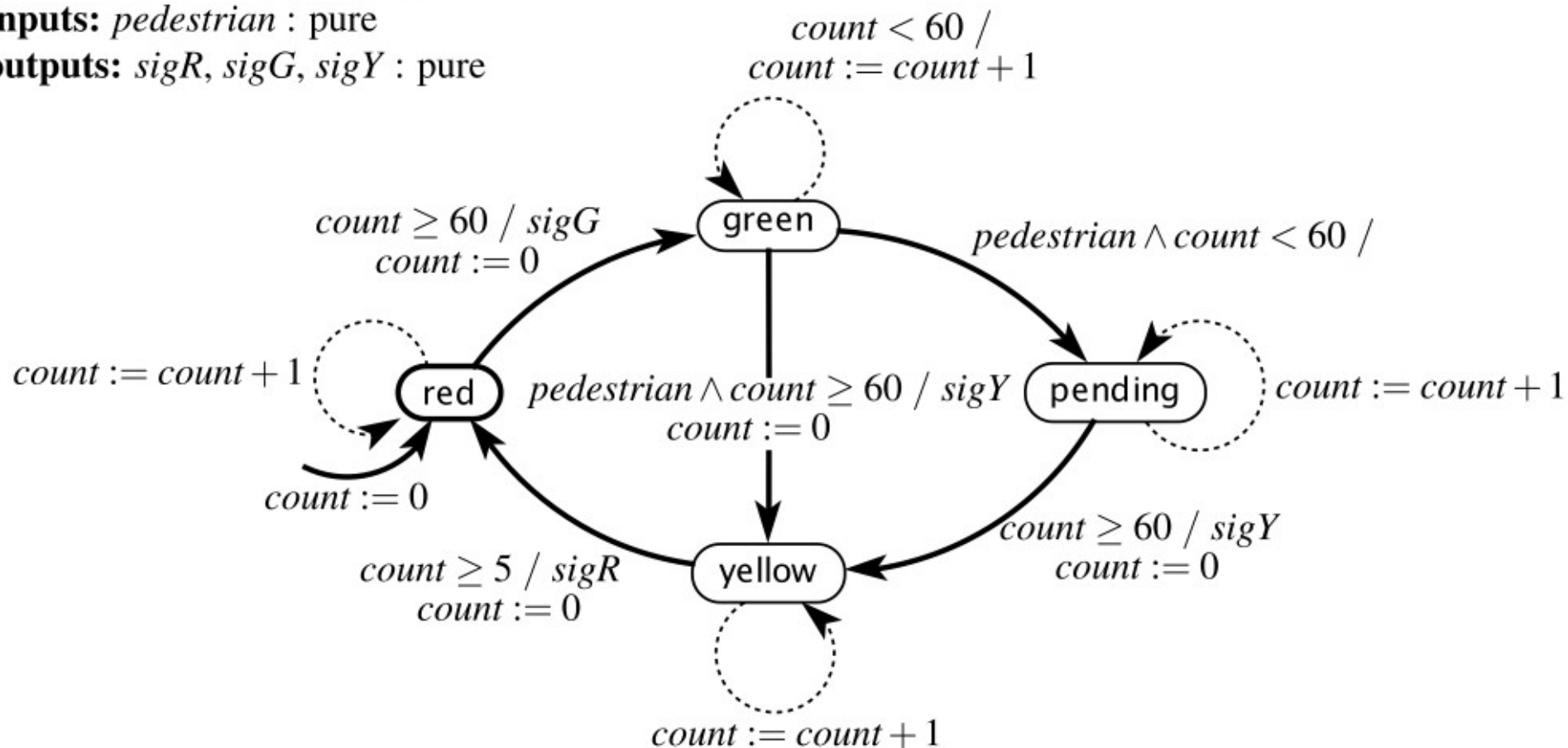
This model assumes one reaction per second
(a *time-triggered* model)

Quiz: What is the Size of the State Space for the Traffic Light Controller?

variable: $count: \{0, \dots, 60\}$

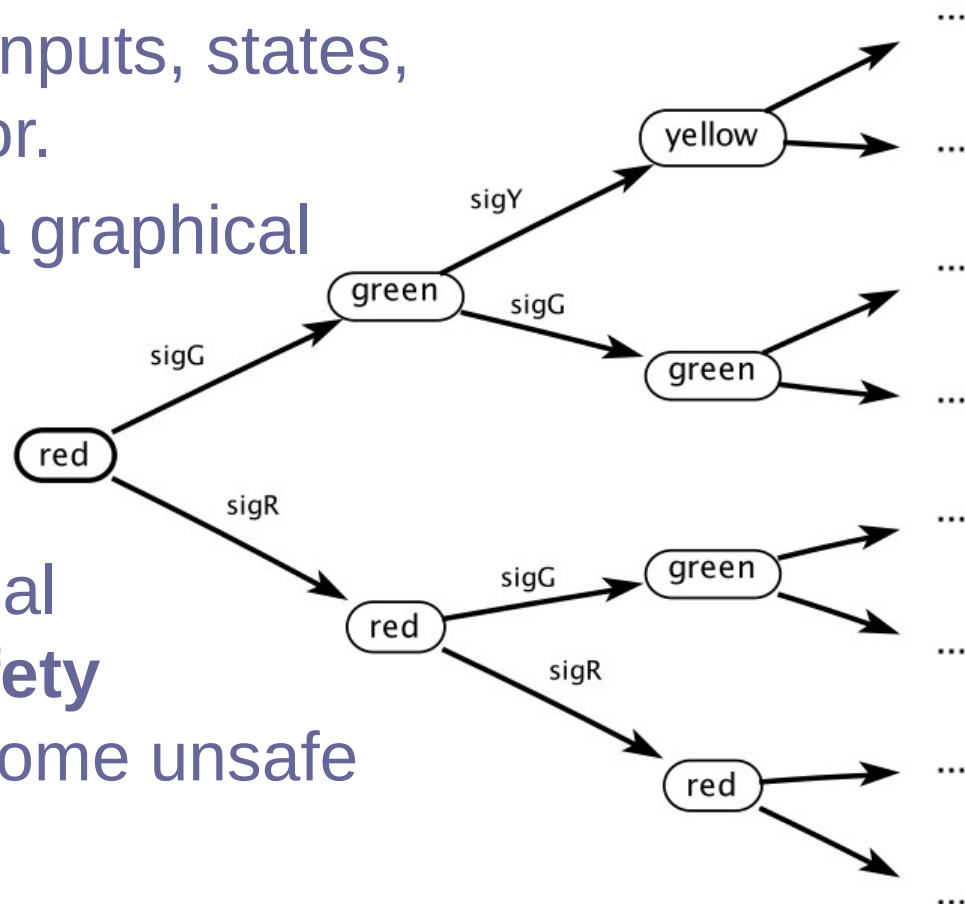
inputs: $pedestrian : \text{pure}$

outputs: $sigR, sigG, sigY : \text{pure}$



Behaviors and Traces

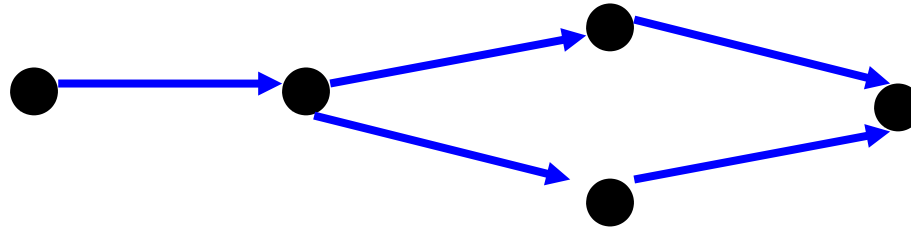
- FSM **behavior** is a sequence of (non-stuttering) steps.
- A **trace** is the record of inputs, states, and outputs in a behavior.
- A **computation tree** is a graphical representation of all possible traces.



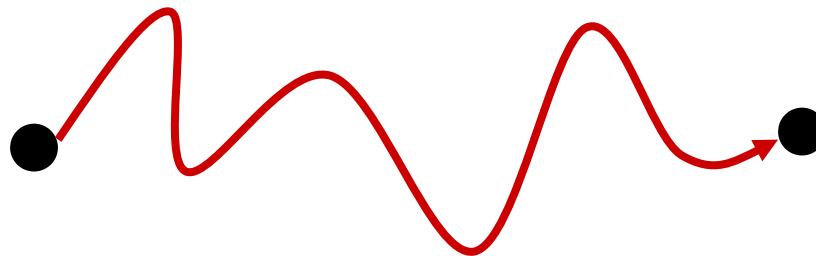
FSMs are suitable for formal analysis. For example, **safety** analysis might show that some unsafe state is not reachable.

Evolution of states for discrete/continuous/hybrid systems

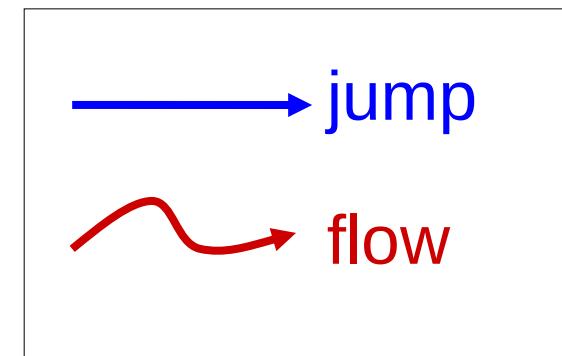
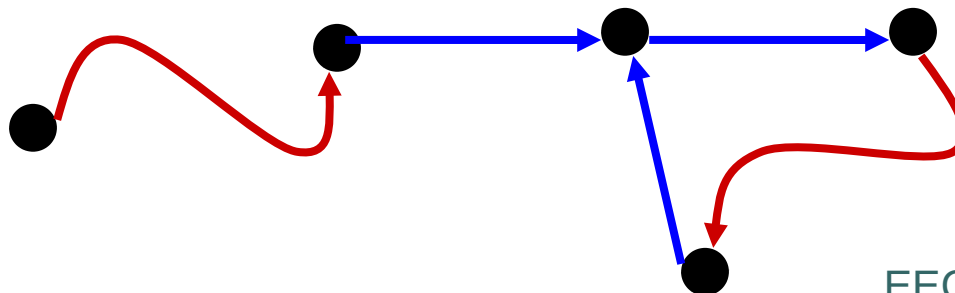
Discrete System (FSM)



Continuous System



Hybrid System



Hybrid systems

- **Hybrid systems** = systems mixing discrete and continuous behavior
- Example: a complex controller may have:
 - A set of discrete functioning modes (discrete states): e.g. Startup / Operation / Idle
 - Each discrete state is implemented as a separate continuous-time system (state refinement)
- State **refinement** = a lower-level implementation of a state

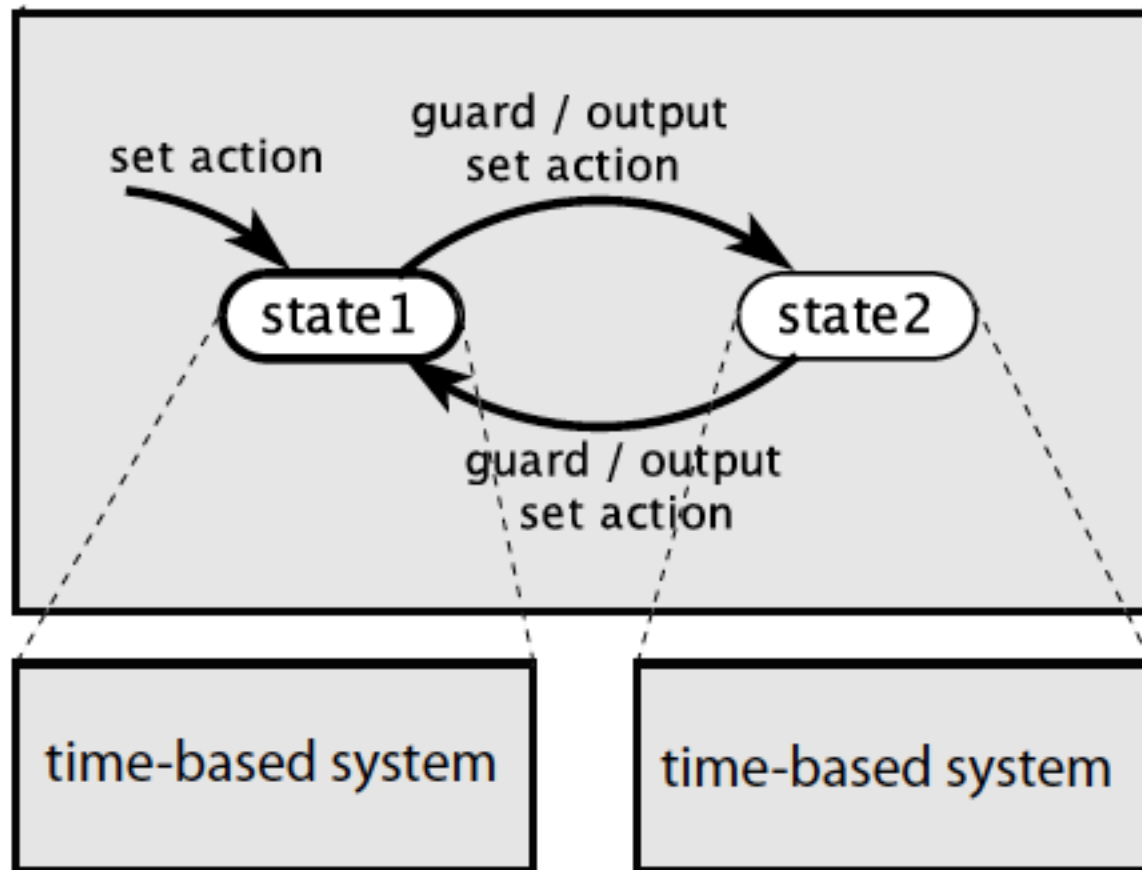
Classes/Examples of hybrid systems

- **Timed automata** = time-based hybrid system where every state refinement just measures passage of time
- **Higher-order systems** = time-based hybrid system where second order derivatives
 - e.g. like timed automata, but with more complicated behavior with respect to time
- **Tow-level control systems** = complex controllers based on two levels of operation:
 - Supervisory control: contains the higher-level discrete modes of operation (e.g.: ECU Power modes: Normal / Sleep Mode 1 / Sleep Mode 2)
 - Low-level control: contains the implementation of each operation mode, as a continuous-time model

Where do Hybrid Systems arise?

- ❑ Digital controller of physical “plant”
 - ❑ thermostat
 - ❑ intelligent cruise/powertrain control in cars
 - ❑ aircraft auto pilot
- ❑ Phased operation of natural phenomena
 - ❑ bouncing ball
 - ❑ biological cell growth
- ❑ Multi-agent systems
 - ❑ ground and air transportation systems
 - ❑ interacting robots

An alternative to FSMs that is explicit about the passage of time: ***Timed automata***, a special case of hybrid systems.

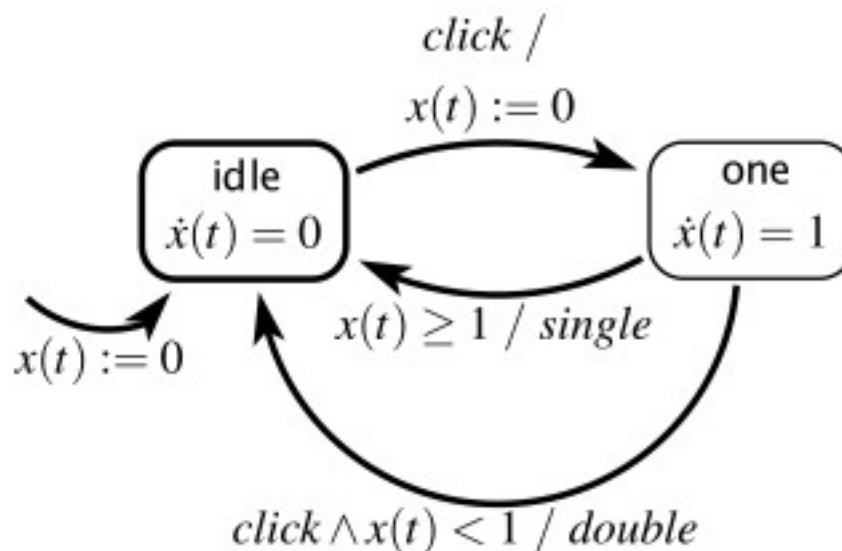


Example: Mouse Double Click Detector

continuous variable: $x(t) \in \mathbb{R}$

inputs: $click \in \{present, absent\}$

outputs: $single, double \in \{present, absent\}$



$\dot{x}(t) = 1 \iff$ “start measuring time”

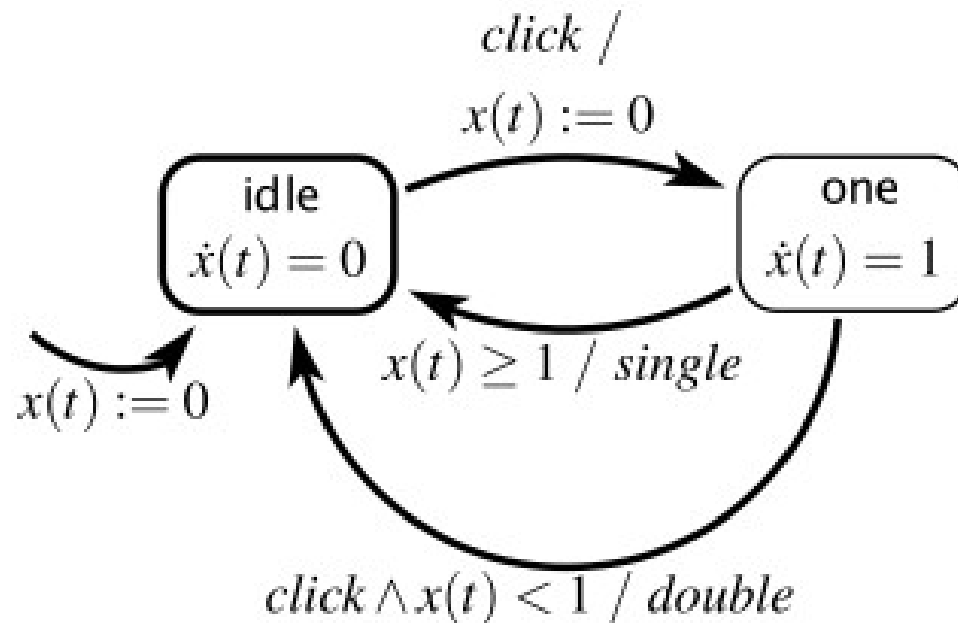
This simple form of hybrid system is called a timed automaton, where the dynamics is just passage of time.

Quiz: How many states does this automaton have?

continuous variable: $x(t) \in \mathbb{R}$

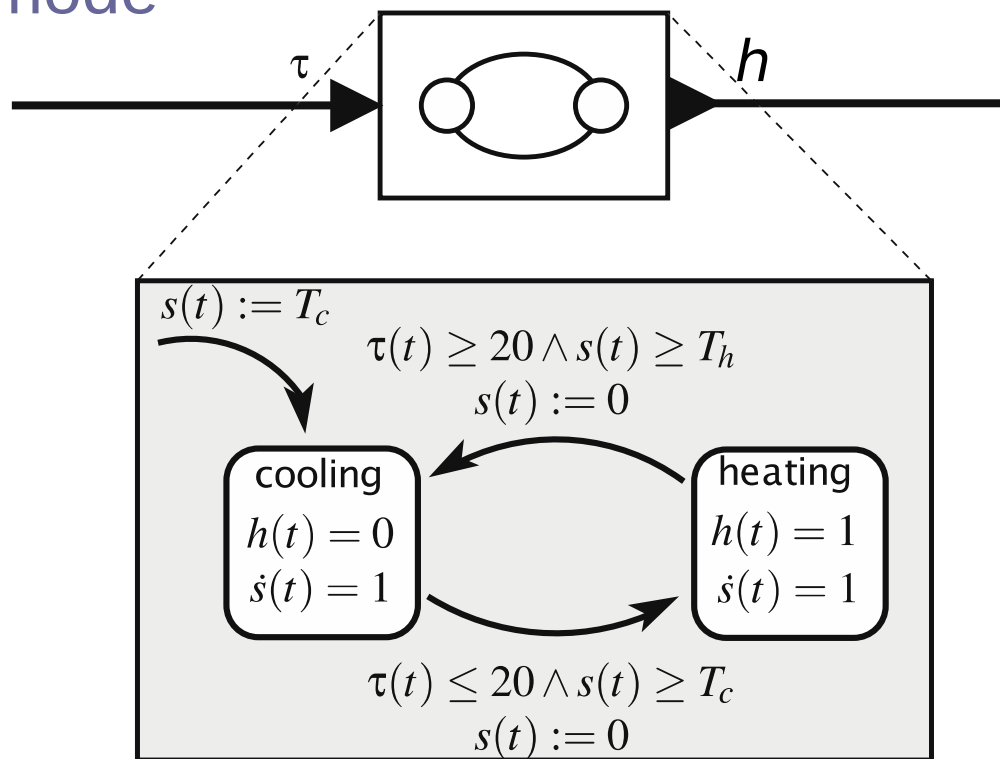
inputs: $click \in \{present, absent\}$

outputs: $single, double \in \{present, absent\}$



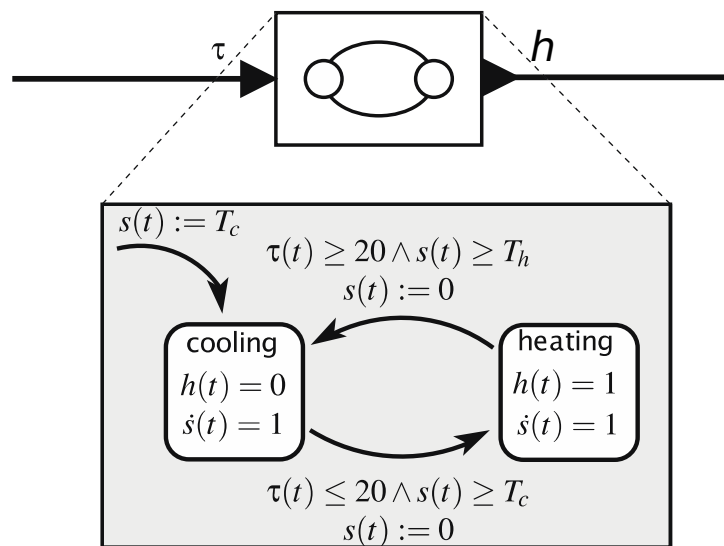
Example: Timed Automaton to Prevent Thermostat Chattering

Temperature threshold is 20 with minimum times T_c and T_h in each mode

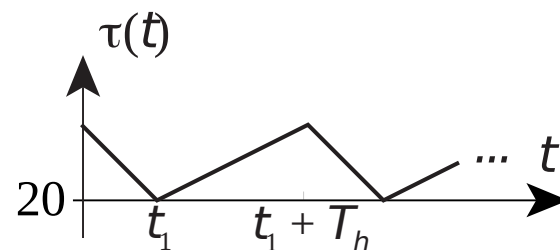


Example: Timed Automaton to Prevent Thermostat Chattering

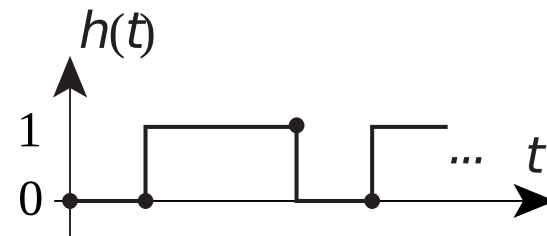
Temperature threshold is 20 with minimum times T_c and T_h in each mode



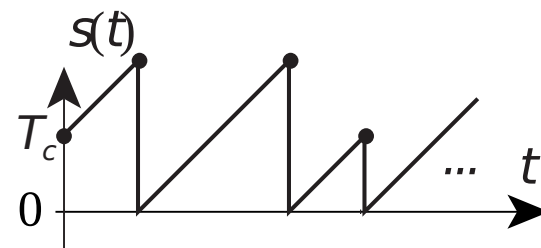
(a)



(b)



(c)

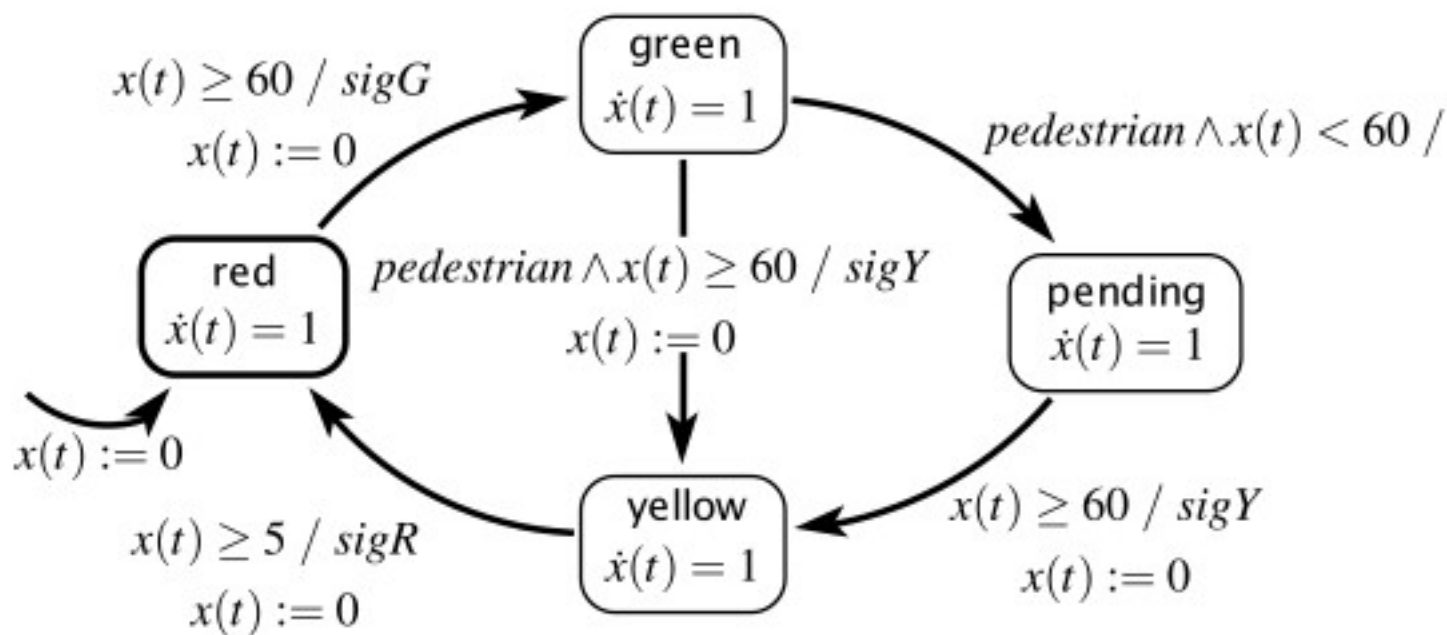


Timed automaton model of a traffic light controller

continuous variable: $x(t) : \mathbb{R}$

inputs: *pedestrian*: pure

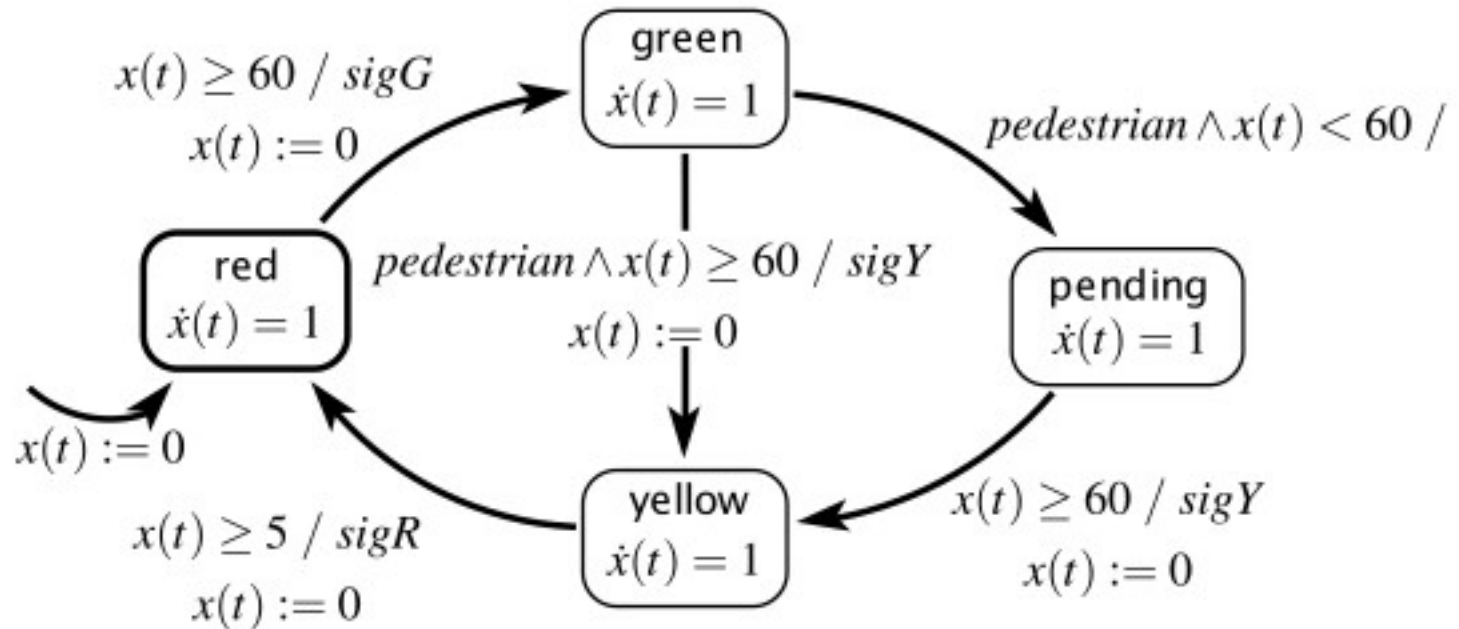
outputs: *sigR*, *sigG*, *sigY*: pure



This light remains green at least 60 seconds, and then turns yellow if a pedestrian has requested a crossing. It then remains red for 60 seconds.

When do reactions occur in a hybrid automaton?

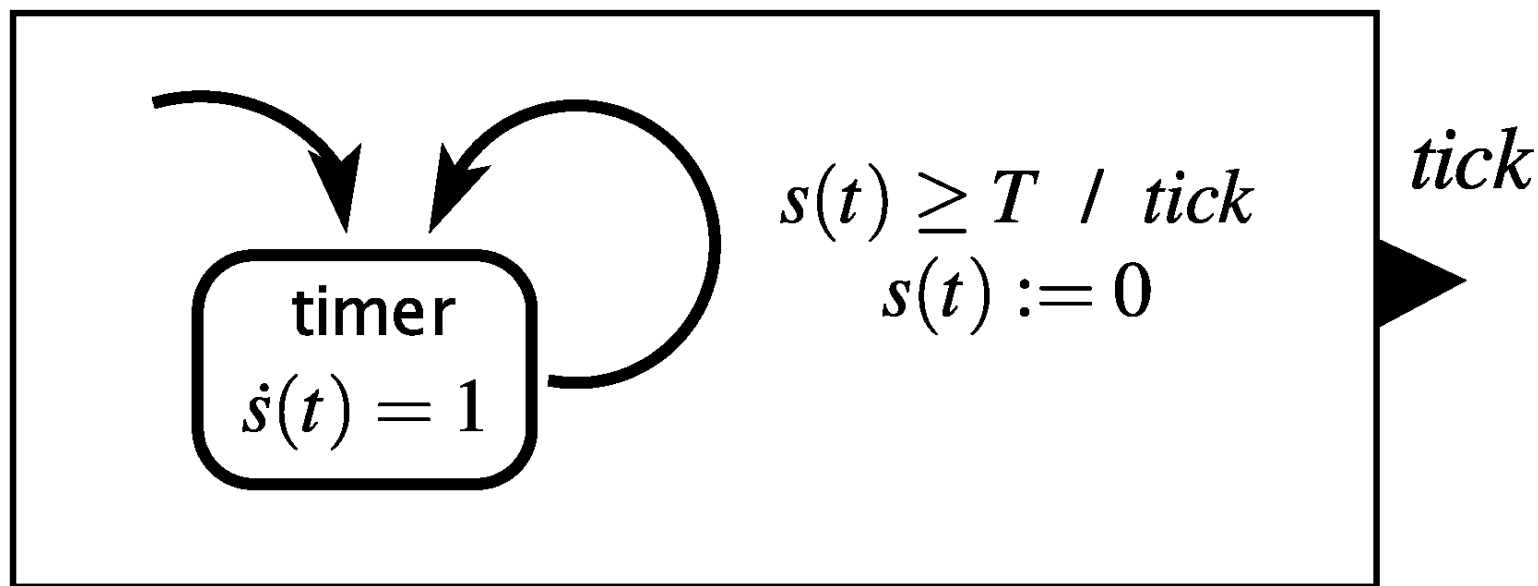
continuous variable: $x(t) : \mathbb{R}$
inputs: *pedestrian*: pure
outputs: *sigR*, *sigG*, *sigY*: pure



Reactions are occurring continually, with the continuous state variable x being continually updated.

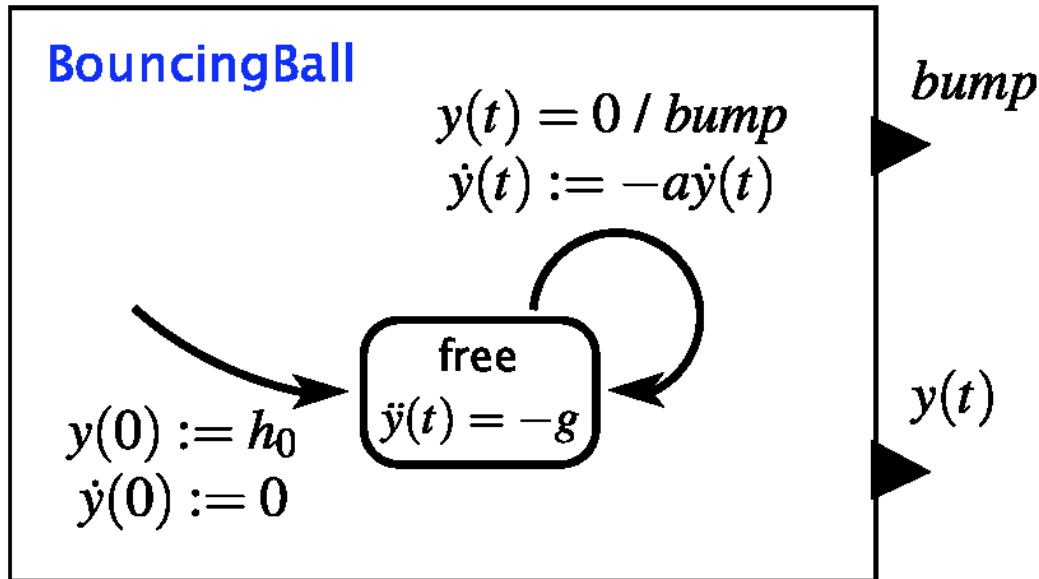
Example: “Tick” Generator (Timer)

How would you model a timer that generates a ‘tick’ each time T time units elapses?



A similar timed automaton can model a generator of a timer interrupt.

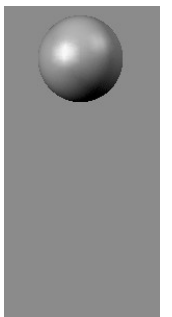
Hybrid Automaton for Bouncing Ball



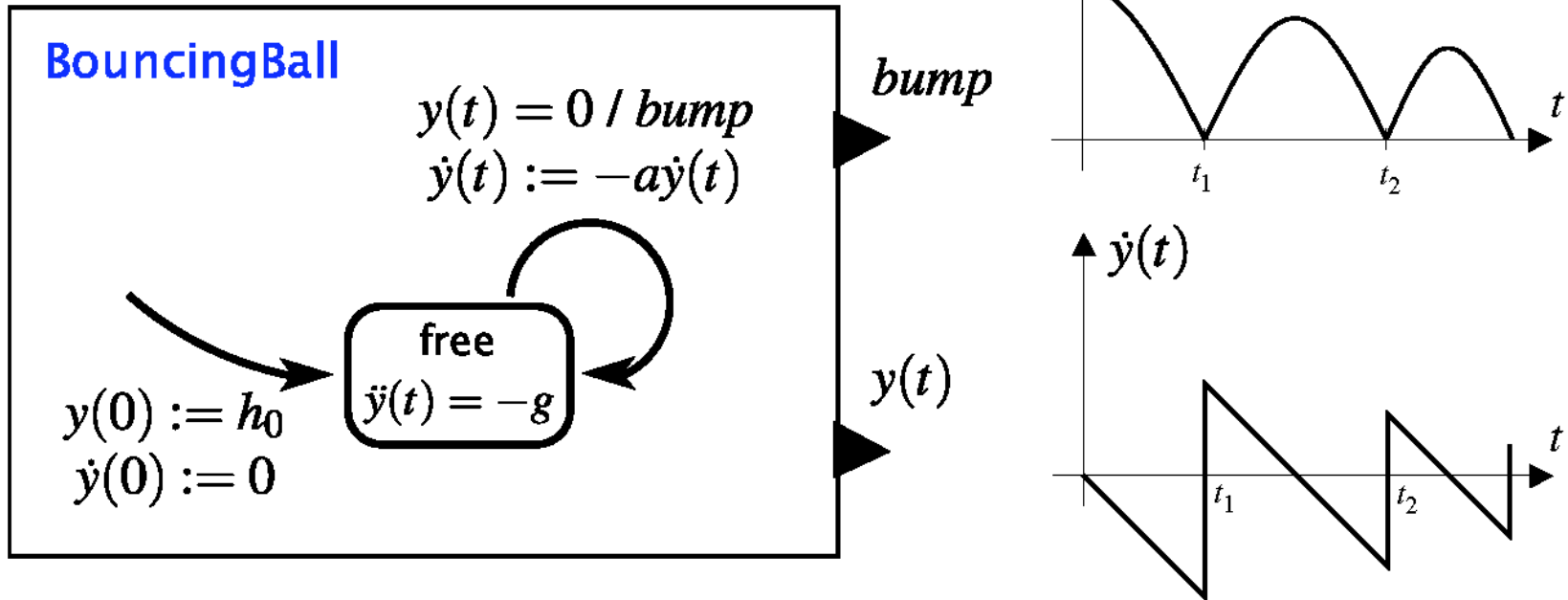
y – vertical distance from ground (position)

a – coefficient of restitution, $0 < a < 1$

If you plotted $y(t)$, what would it look like?

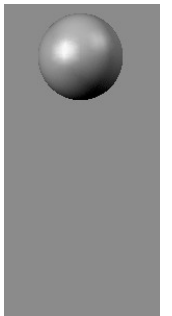


Hybrid Automaton for Bouncing Ball

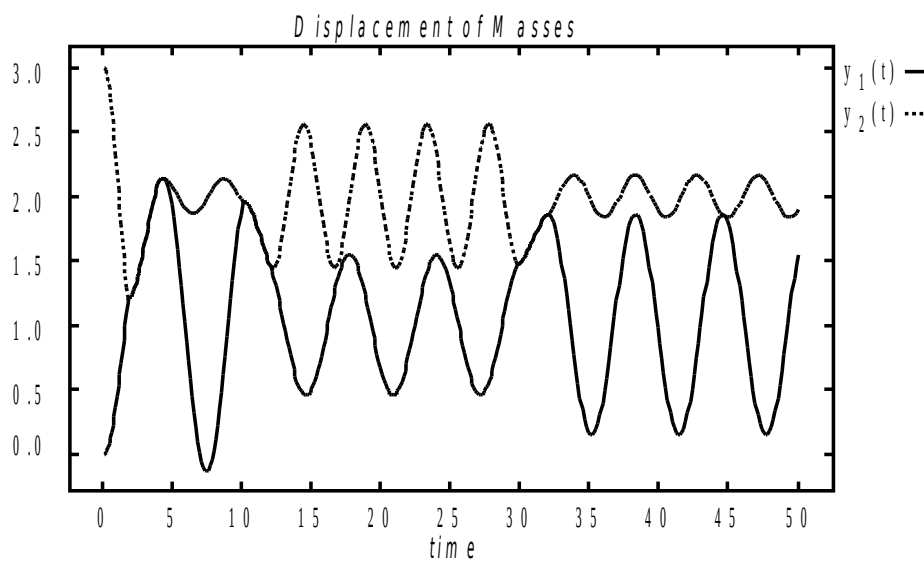
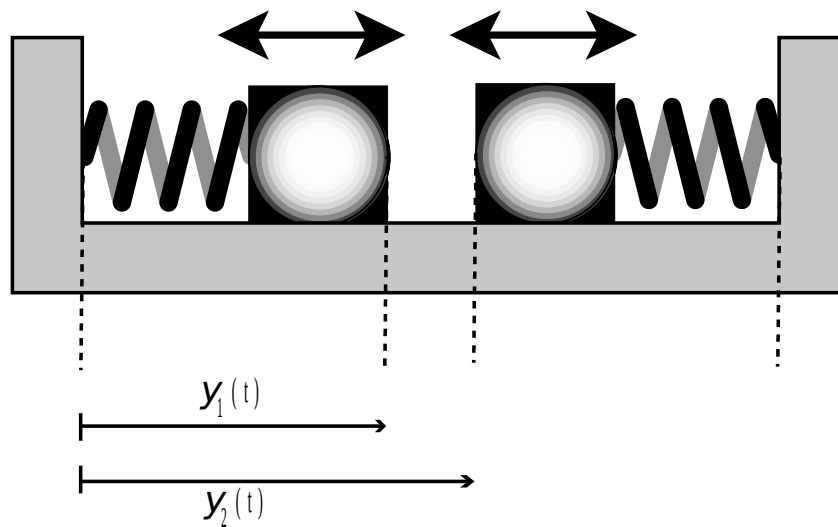


y – vertical distance from ground (position)

a – coefficient of restitution, $0 \leq a \leq 1$



Sticky Masses



Sticky Masses

