Hybrid Systems/ Time Automata

01266212
CYBER PHYSICAL SYSTEM DESIGN

SEMESTER 1-2022

Original contents from
Edward A. Lee and Prabal Dutta, UC Berkeley, EECS 149/249A

Topics

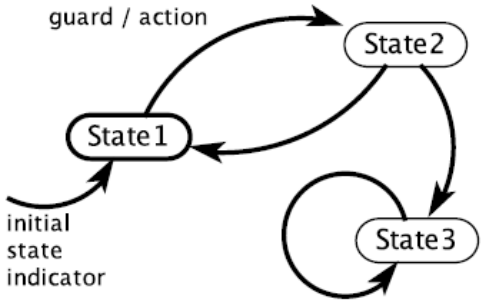
- Recap: Dynamic System and State Machin
- Time-based State Machine or Hybride
- Time Automata

Recap – Discrete Dynamic (FSM)

- **Discrete models** with finite state spaces are called **finite-state machines (FSMs).**
- A state machine is a model of a system with *discrete dynamics* that at each reaction maps *valuations* of the inputs to *valuations* of the outputs, where the map may depend on its current state.

$$s(Current\ State, Input) = Output$$

- For a FSM system with a small number of states , each state is represented by a bubble, so for this diagram, the set of states is given by



Mathematical Notation FSM

A finite-state machine is a five-tuple (States; Inputs; Outputs; update; initialState)

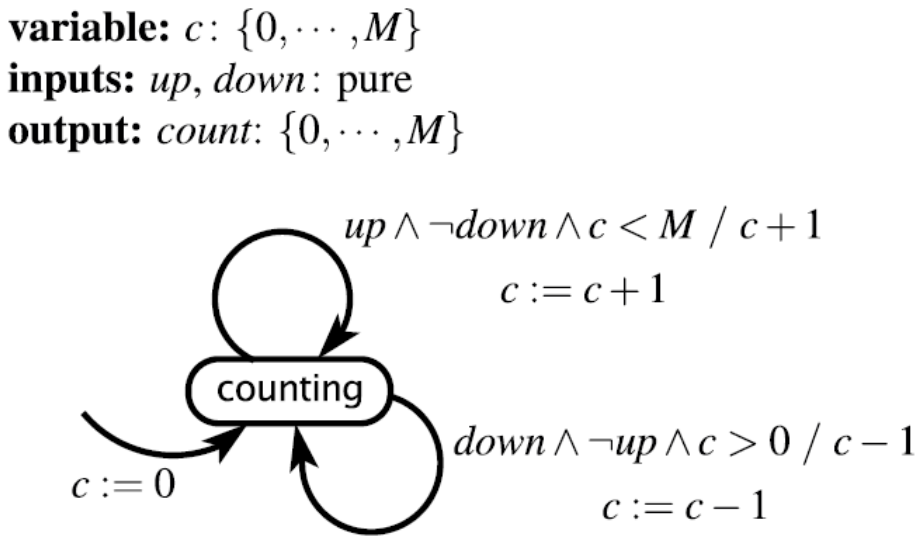
- *States* is a finite set of states;
- *Inputs* is a set of input valuations;
- *Outputs* is a set of output valuations;
- $update : States \times Inputs \rightarrow States \times Outputs$ is an **update function**, mapping a state and an input valuation to a *next* state and an output valuation;
- *initialState* is the initial state.

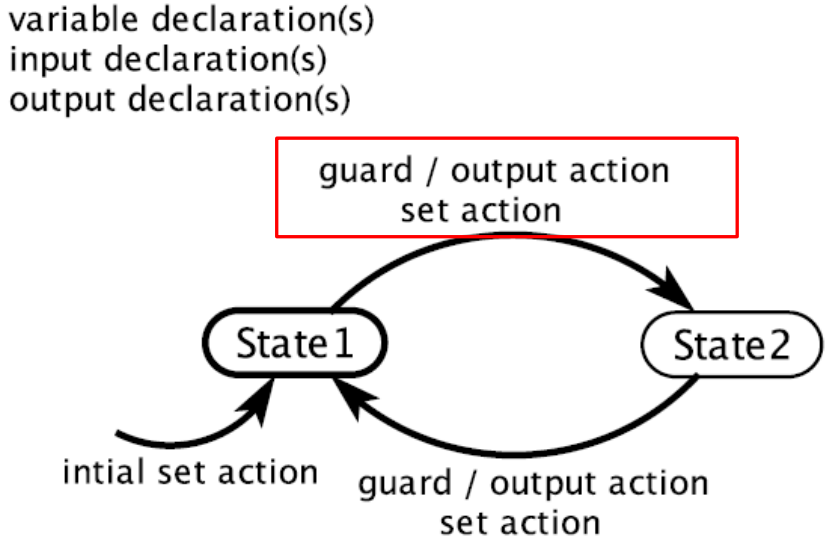
The FSM reacts in a sequence of reactions. At each reaction, the FSM has a current state, and the reaction may transition to a next state, which will be the current state of the next reaction.

Extended State Machines

An extended state machine solves this problem by augmenting the FSM model with variables that may be read and written as part of taking a transition between states.

In this example, a variable *c*, is declared explicitly at the upper left to make it clear that *c* is a variable and neither an input nor an output.

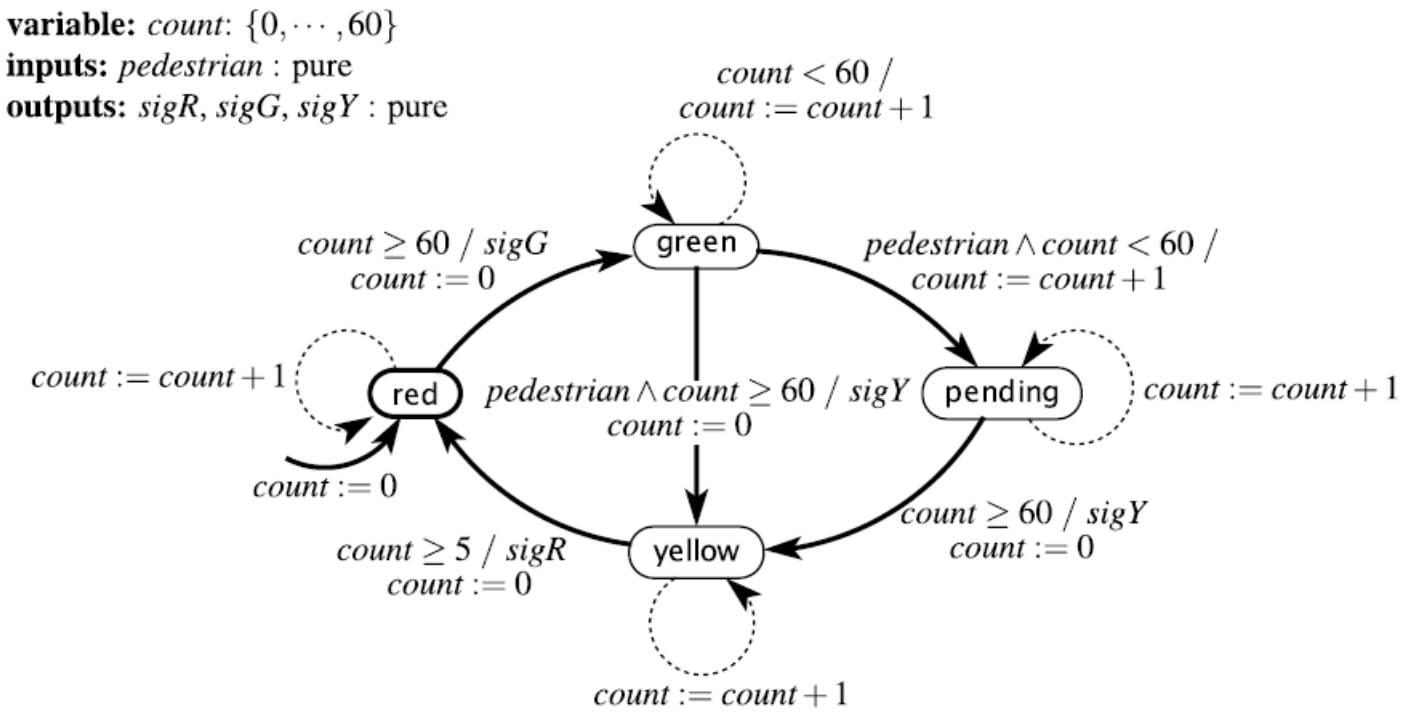




SOURCE: LEE, BERKELEY

Extended State Machines Example

This is a time triggered machine that assumes it will react once per second.



Number of States

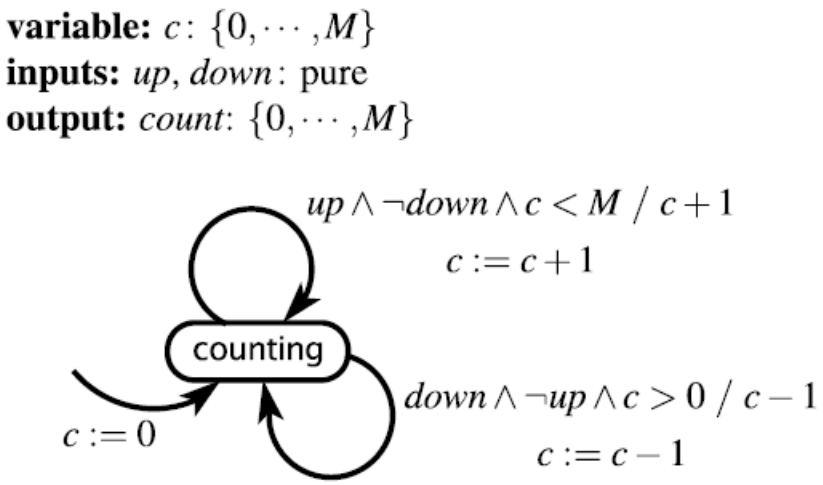
If there are *n* discrete states (bubbles) and *m* variables each of which can have one of *p* possible values, then the size of the state space of the state machine is

$$|States| = np^m$$

E.g. The garage counter has $n = 1$, $m = 1$, and

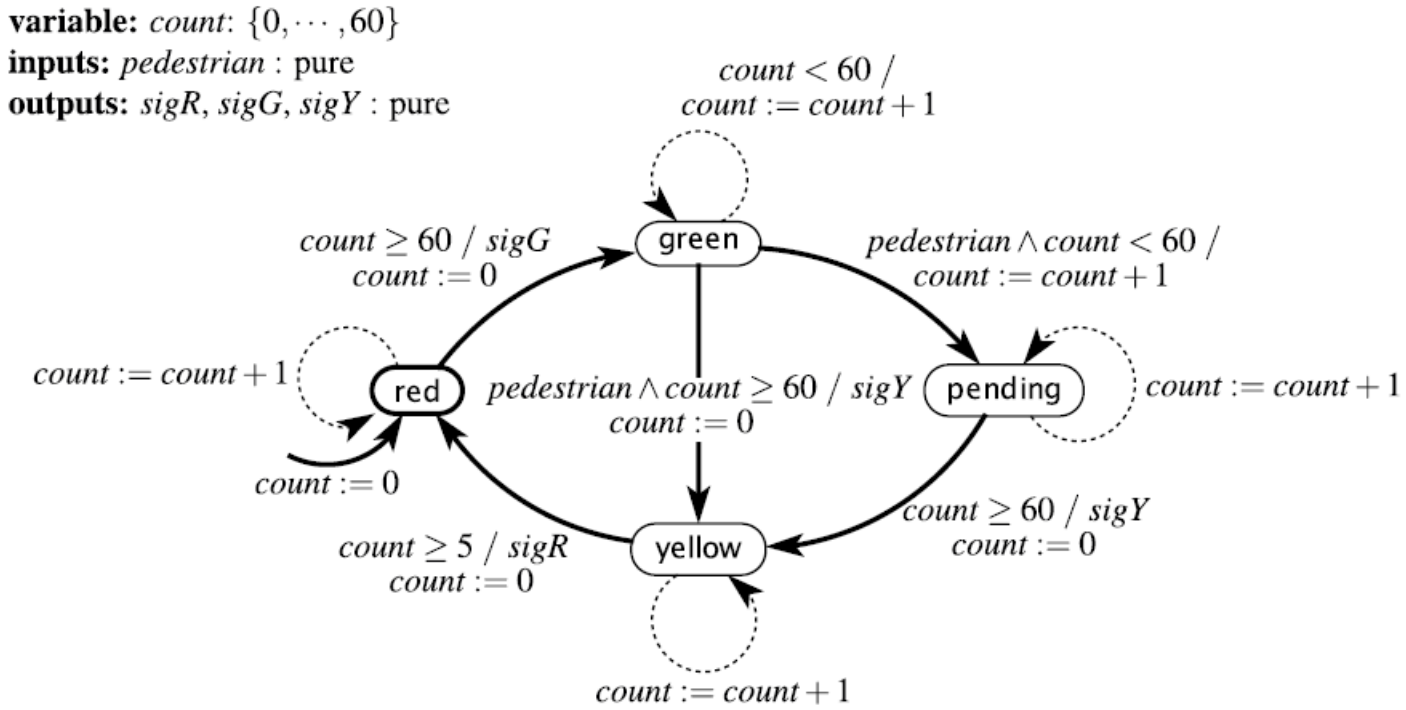
$p = M + 1$, so the total number of states is

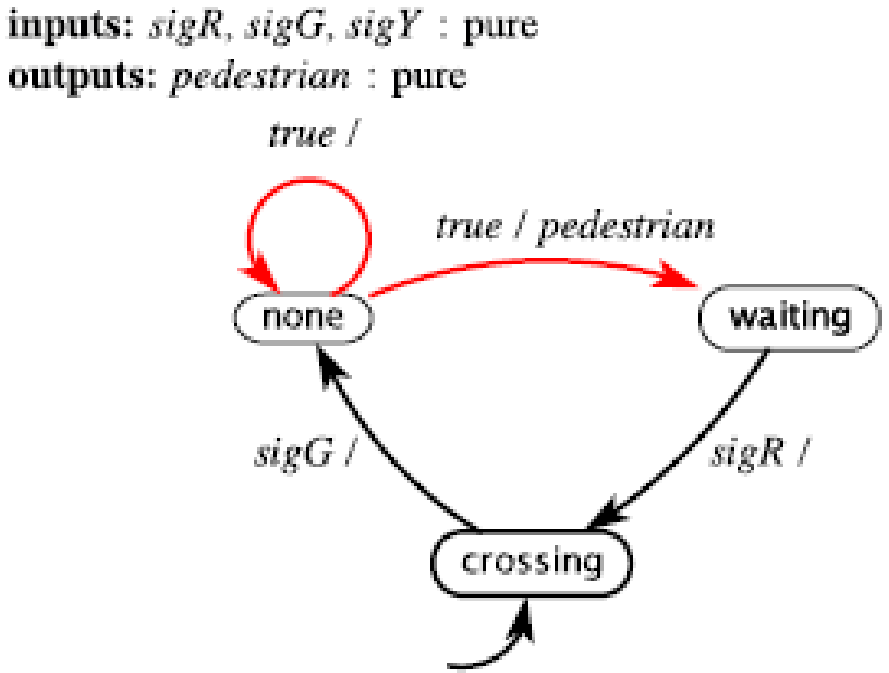
M + 1.



Number of States

What is the number of state?





When *sigG* is received, the FSM transitions to *none*.
Both transitions from this state have guard true, indicating that they are always enabled. Since both are enabled, this machine is nondeterministic

Nondeterministic FSM

- If for any state of a state machine, there are two distinct transitions with guards that can evaluate to true in the same reaction,
- Then the state machine is nondeterministic.

Behaviors

A behavior may be more conveniently represented as a sequence of valuations called an **observable trace**.

Let x_i represent the valuation of the input ports and y_i the valuation of the output ports at reaction i.

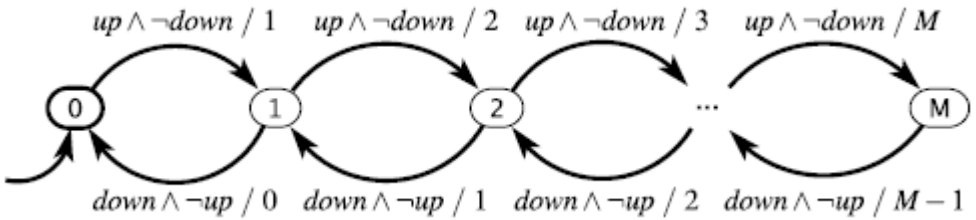
$$((x_0, y_0), (x_1, y_1), (x_2, y_2), \cdots) .$$

An **execution trace** includes the state trajectory

$$((x_0, s_0, y_0), (x_1, s_1, y_1), (x_2, s_2, y_2), \cdots) ,$$

$$s_0 \xrightarrow{x_0/y_0} s_1 \xrightarrow{x_1/y_1} s_2 \xrightarrow{x_2/y_2} \cdots$$

Example

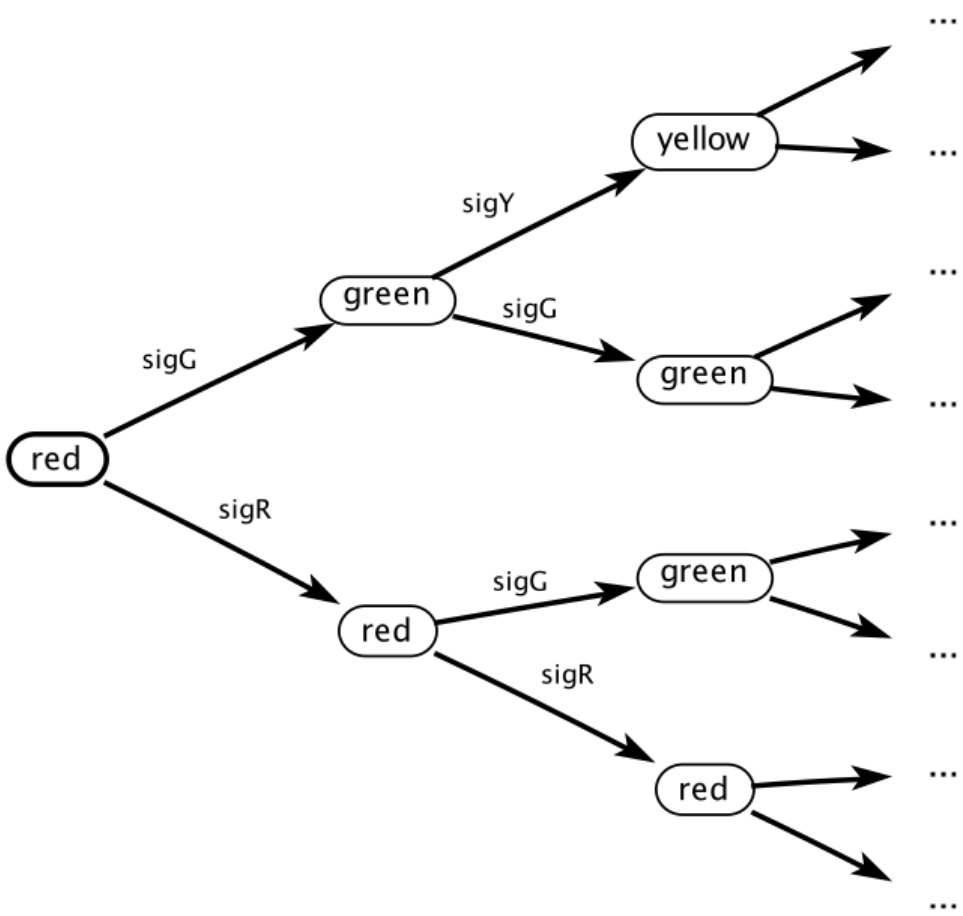


$$0 \xrightarrow{up \wedge down\ /} 0 \xrightarrow{\ /\ } 0 \xrightarrow{up\ /\ 1} 1 \xrightarrow{down\ /\ 0} 0 \xrightarrow{up\ /\ 1} \cdots$$

Computation Tree and Traces

- 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.

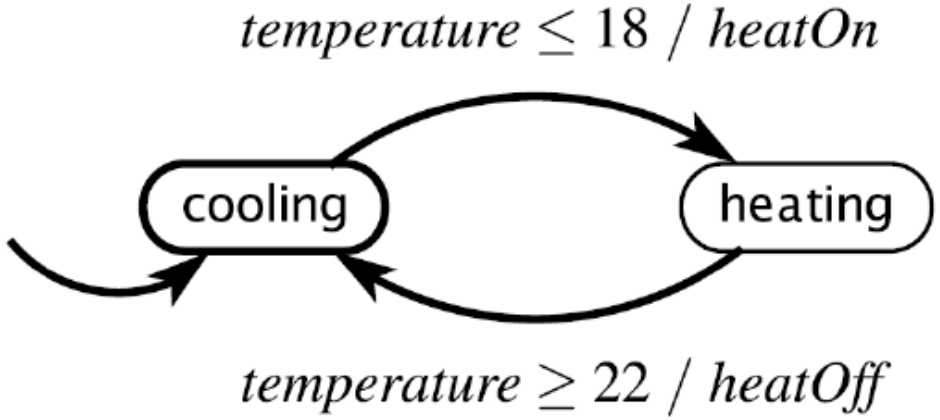


Hysteresis: A model of a thermostat

$temperature : \mathbb{R} \rightarrow \{absent\} \cup \mathbb{R}$

$heatOn, heatOff : \mathbb{R} \rightarrow \{absent, present\}$

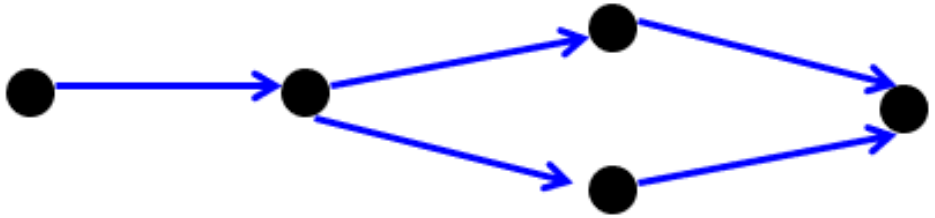
input: *temperature* : $\mathbb{R}$
outputs: *heatOn*, *heatOff* : pure



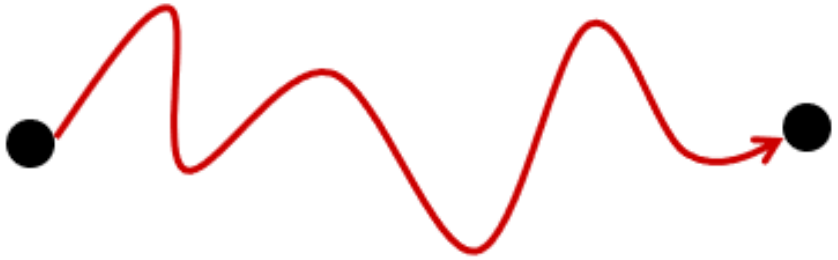
Hysteresis avoids **chattering**, where the heater would turn on and off rapidly when the temperature is close to the setpoint temperature.

Hybrid Systems/ Time Automata

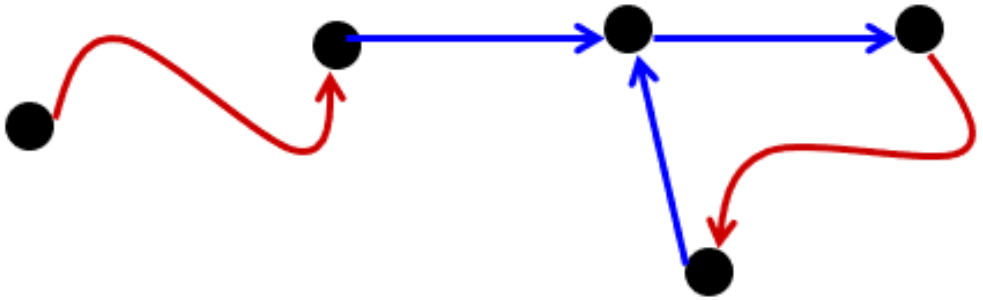
Discrete System (FSM)

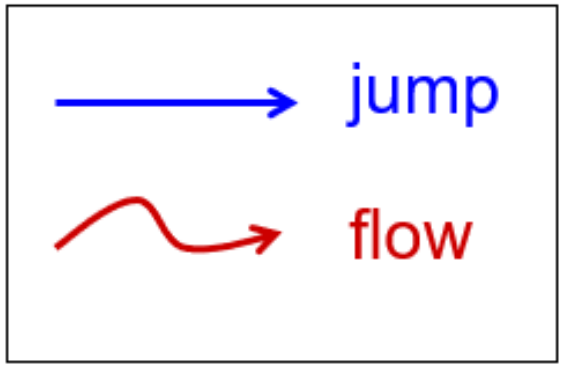


Continuous System



Hybrid System

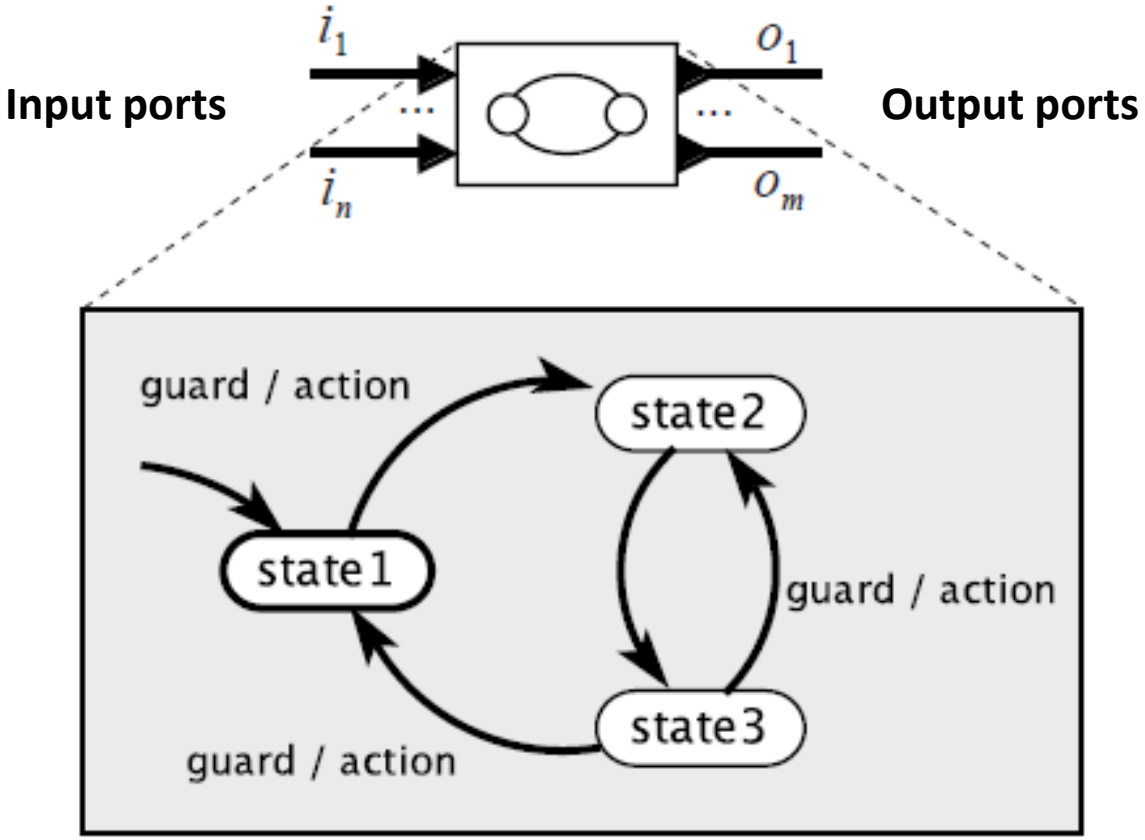




Where do Hybrid Systems arise?

- 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

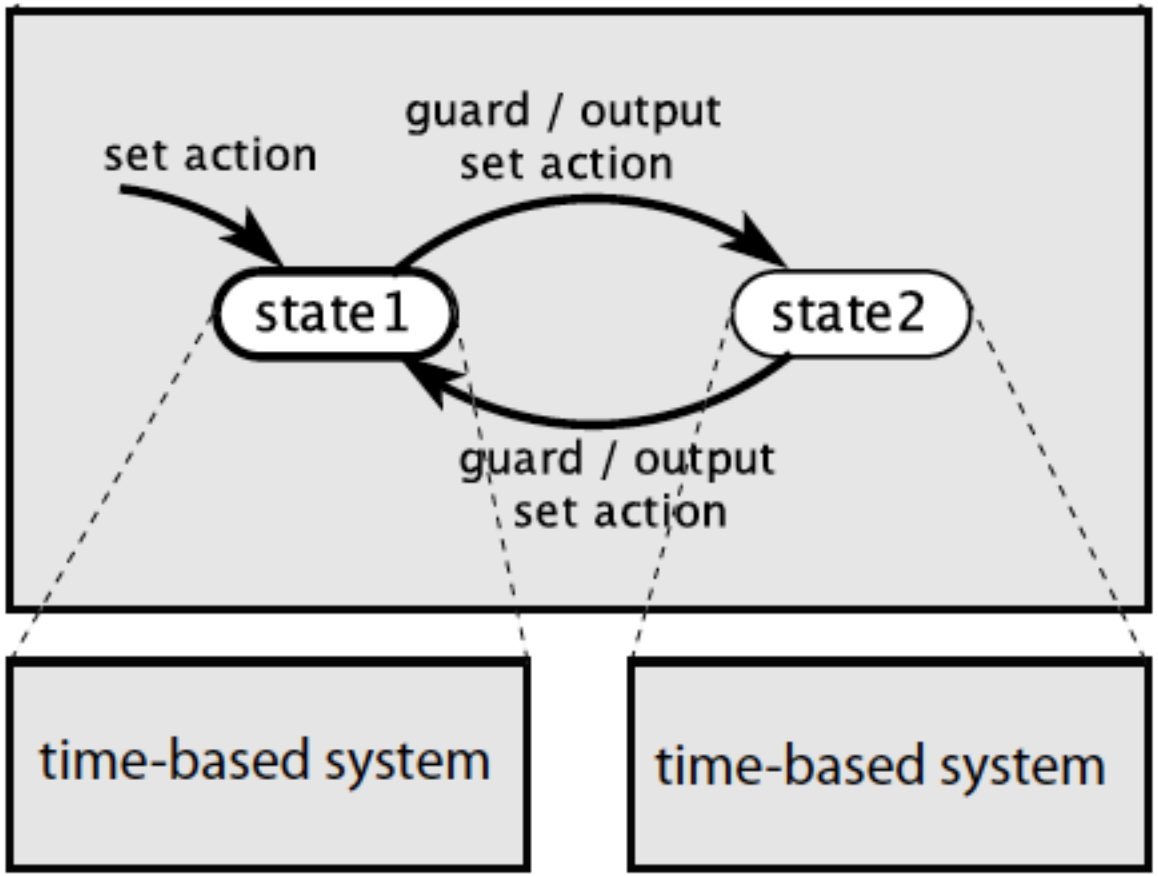
Actor Model for State Machine



Both types of **port**s have a value *present* or *absent* (for a pure signal) or a member of some set of *values* (for valued signal).
The **guards** on the transitions define subsets of possible values on input ports, and the **actions** assign values to output ports.

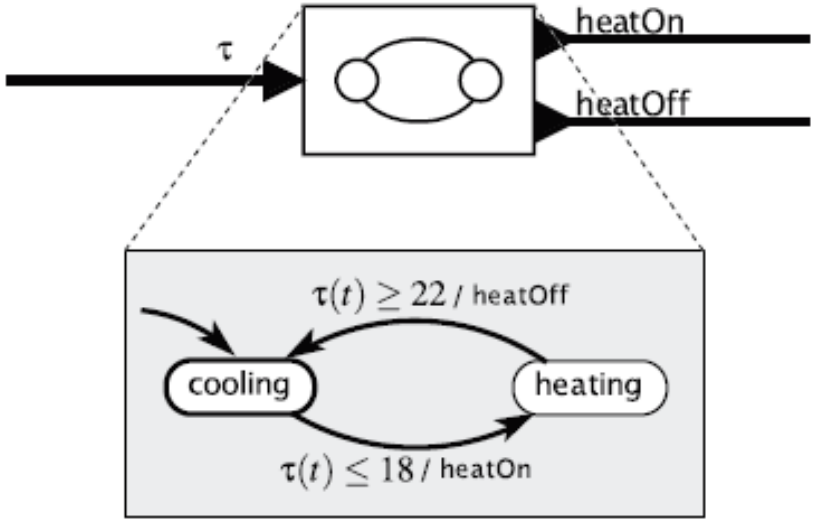
Assumed that state machines operate in a sequence of discrete reactions and that inputs and outputs are absent between reactions.

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



Continuous Inputs

- State machine model allows **inputs** and **outputs** to be **continuous-time** signals.
- State transitions occur on **the same timeline** used for the time-based portion of the system.
- A transition is occurred when a **guard** on an outgoing transition from the current state becomes enabled.
- During the time between reactions, a state machine is understood to not transition between modes, but the inputs and outputs are no longer required to be absent during that time.



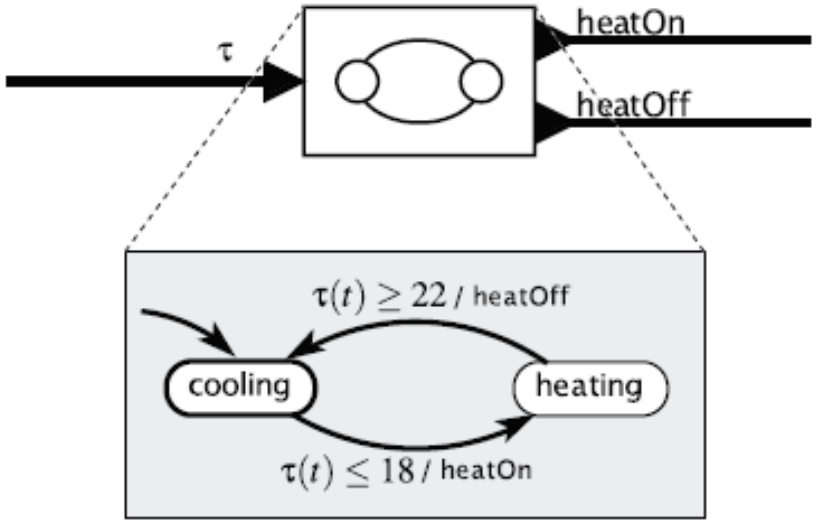
Thermostat with a Continuous-time Input Signal

A state machine with states

$$\Sigma = \{\text{heating},\ \text{cooling}\}$$

and input is a continuous-time temperature signal at time t:

$$\tau(t): \mathbb{R} \rightarrow \mathbb{R}$$



The initial state is **cooling**, and the transition out of this state is enabled at the earliest time t after the start time when $\tau(t) \leq 18$.

The outputs are present only at the times the transitions are taken.

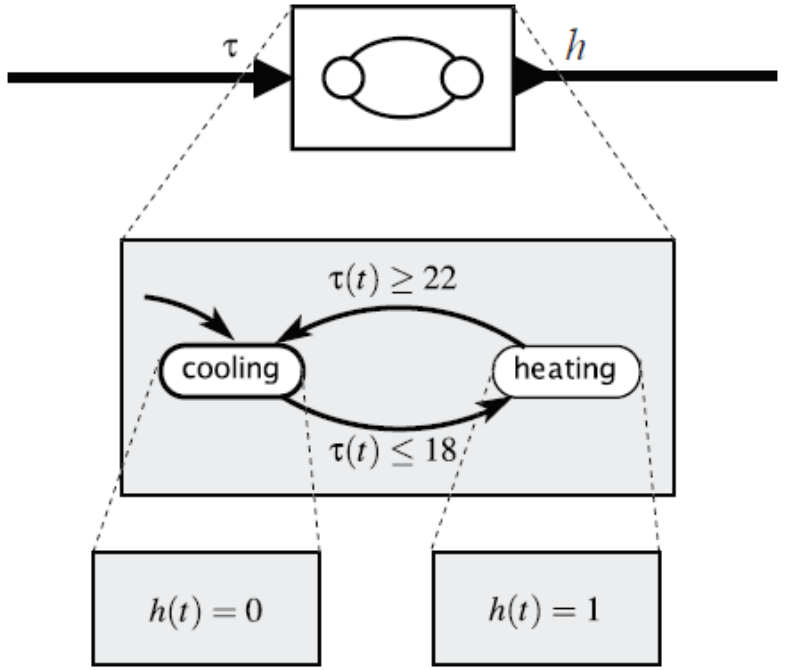
State Refinements: Continuous-time Outputs

We wish to produce a control signal whose value is 1 when the **heat** is *on* and 0 when the **heat** is *off*.

Such a control signal could directly drive a heater.

Each state has a refinement that gives the value of the **output** *h* while the state machine is in that **state**.

The **state refinement** defines dynamic behavior of the outputs and (possibly) additional **continuous state variables.**



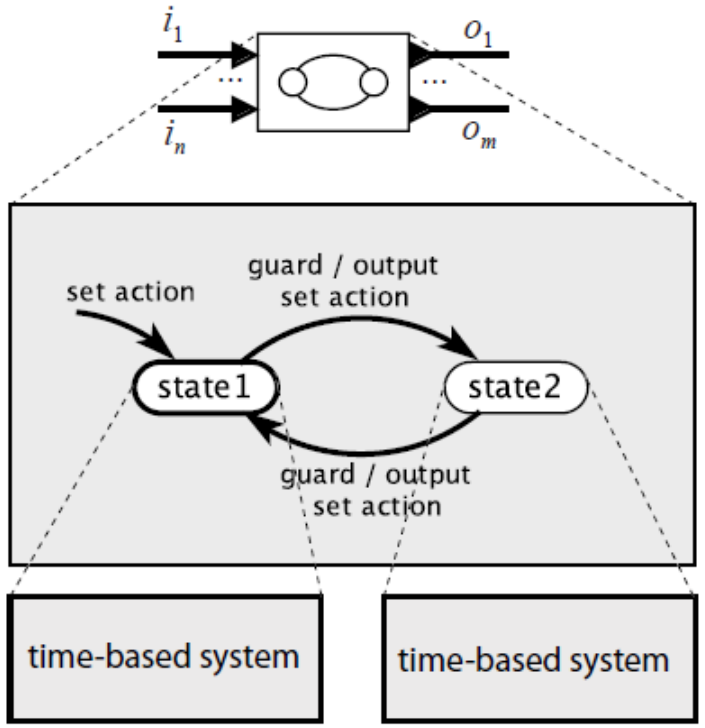
A thermostat with continuous-time output

Hybrid Systems

Each state is associated with **a state refinement** labeled in the figure as a **"time-based system."**

The **set actions** set the values of such additional **state variables** when a transition is taken.

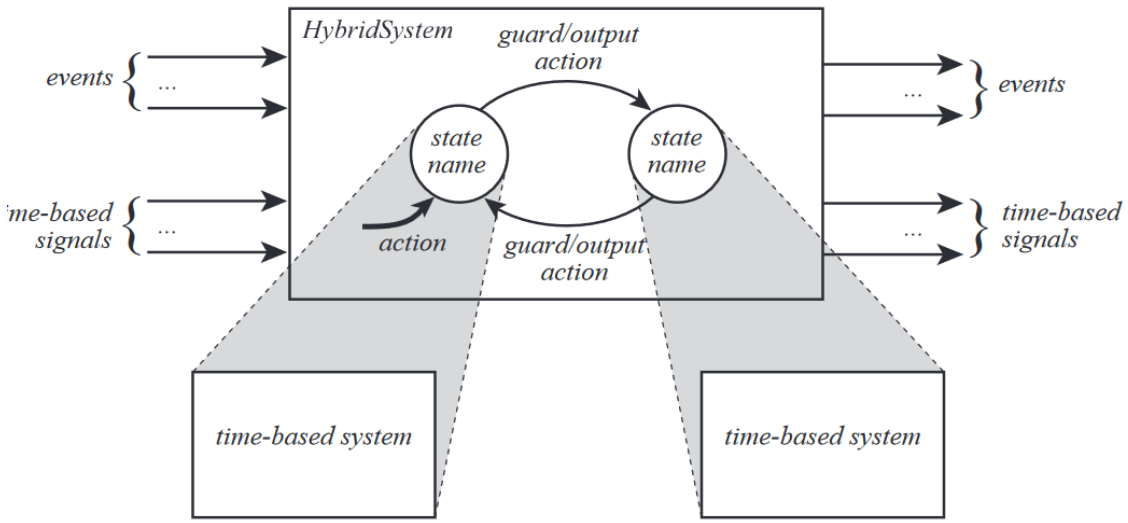
The states of the FSM may be referred to as **modes** rather than states.



Hybrid Systems (State Refinements)

A hybrid system is sometimes called a **modal model** because it has a finite number of **modes**, one for each state of the FSM.

When it is in a **mode**, it has dynamics specified by **the state refinement** $S_1(t)$.



Timed Automata/Finite State Machine

- Most cyber-physical systems require measuring the passage of time and performing actions at specific times.
- A device that measures the passage of time, a **clock**, that has a particularly simple dynamics: its state progresses linearly in time.
- **Timed automata**, a formalism introduced by Alur and Dill (1994), enable the construction of more complicated systems from such simple clocks.
- **Timed automata** are modal models where the time-based refinements have very simple dynamics; all they do is measure the passage of time.

Clock in Timed Automata

A clock is modeled by a first-order differential equation,

$$\forall\, t \in T_m, \quad \dot{s}(t) = a,$$

where s: $\mathbb{R} \rightarrow \mathbb{R}$ is a continuous-time signal, $s(t)$ is the value of the clock at time t, and $T_m \in \mathbb{R}$ is the subset of time during which the hybrid system is in mode m.

The rate of the clock, a, is a constant while the system is in this mode.

If $a = 1$ and $T_m = 1$, $s(t = T_m)$ = ?

Example: Timed Automaton to Prevent Thermostat Chattering

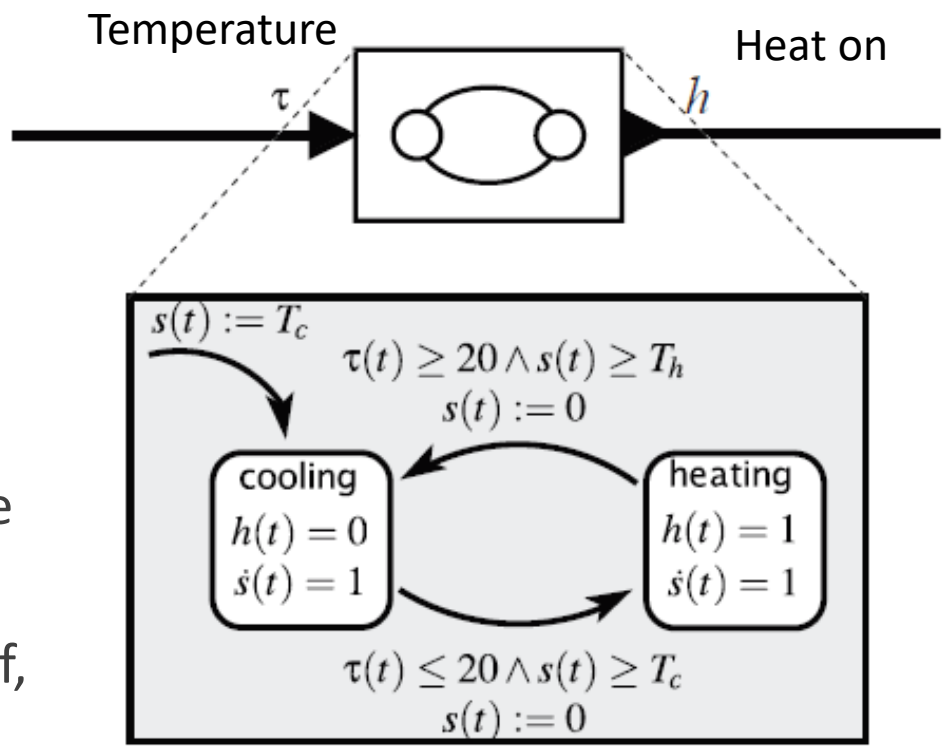
Each state refinement has a clock, which is a continuous-time signal ***s*** with dynamics given by $\dot{s}(t) = 1$.

The initial state cooling has a set action on the dangling transition indicating the initial state, $s(t) := T_c$

The other two transitions each have set actions that reset the clock *s* to zero. $s(t) := 0$

The guard $(s(t) \geq T_h)$ ensures that the heater will always be on for at least time T_h (E.g. T_h =5 s)

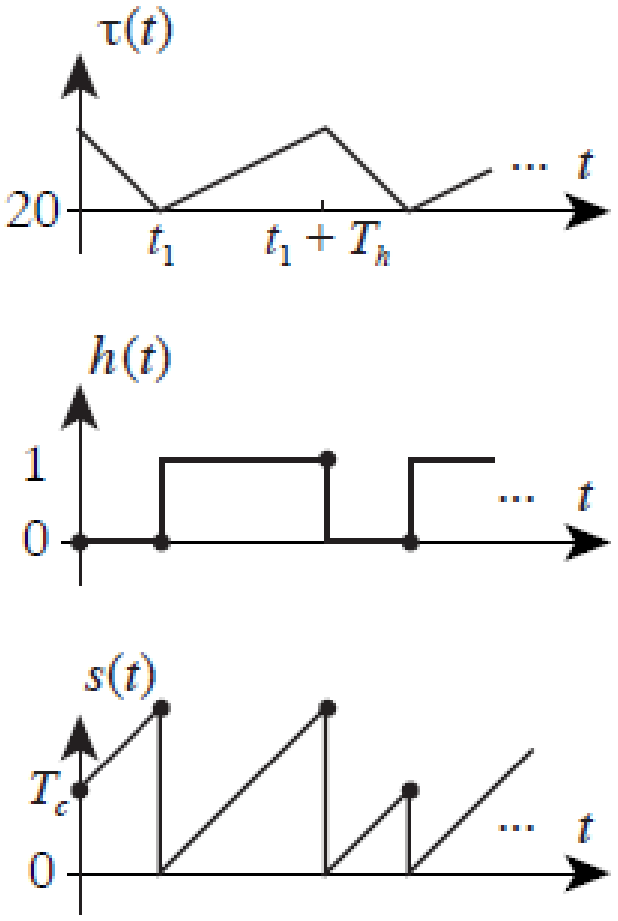
The guard $(s(t) \geq T_c)$ specifies that once the heater goes off, it will remain off for at least time T_c. (E.g. T_c =5 s)

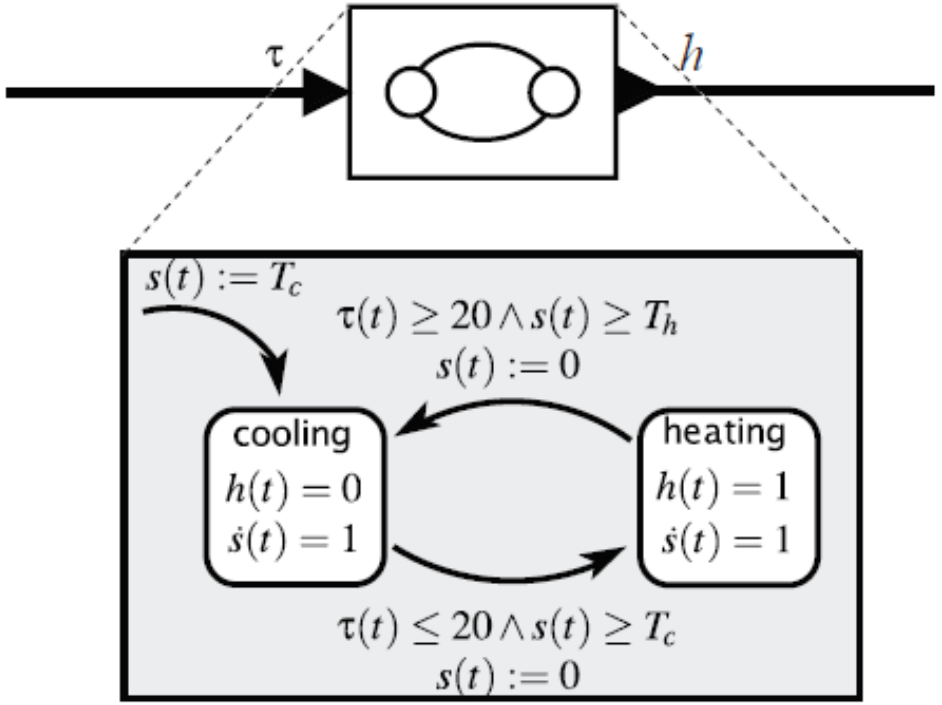


Temperature threshold is 20 with minimum times T_c and T_h in each mode

Reading Assignment Example 4.3 at page 84-85

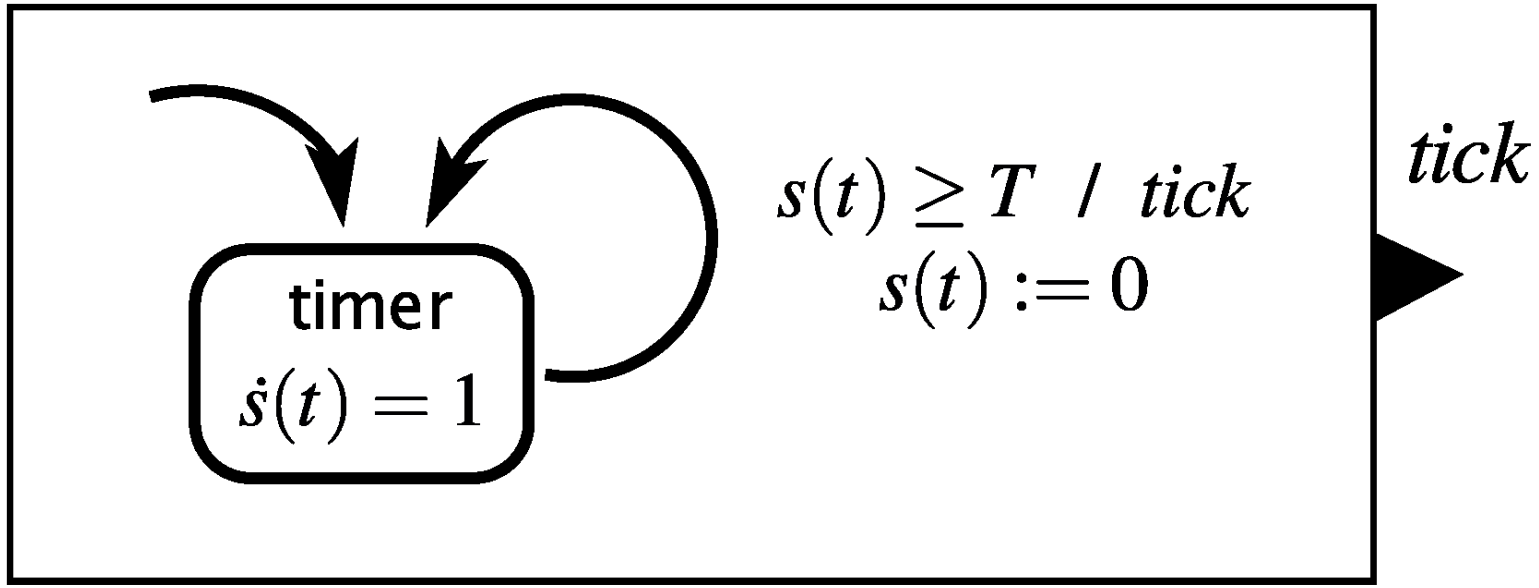
Possible Execution of the Timed Automaton





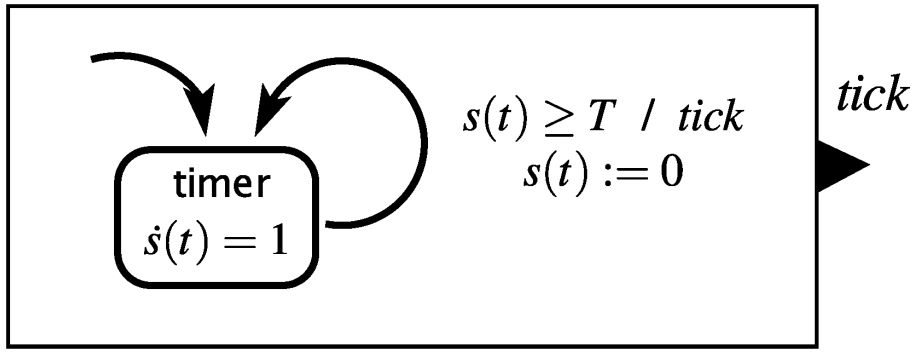
Example: “Tick” Generator (Timer)

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



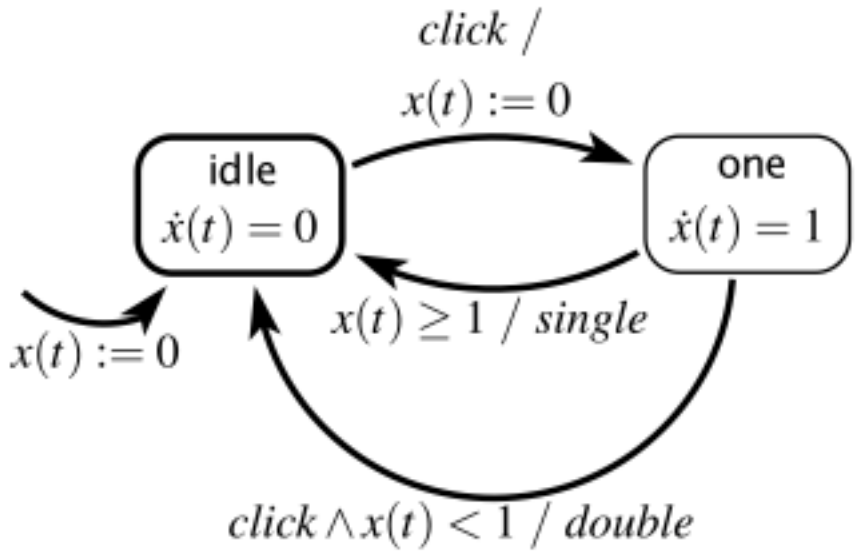
A similar timed automaton can model a generator of a timer interrupt.

Time



Example: Mouse Double Click Detector

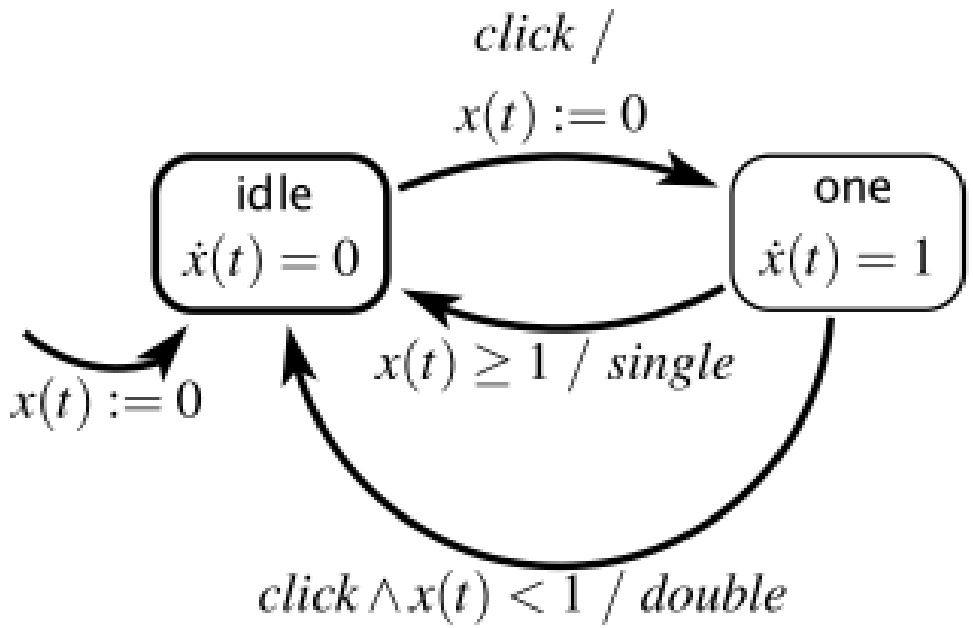
continuous variable: $x(t) \in \mathbb{R}$
inputs: $click \in \{present, absent\}$
outputs: $single, double \in \{present, absent\}$



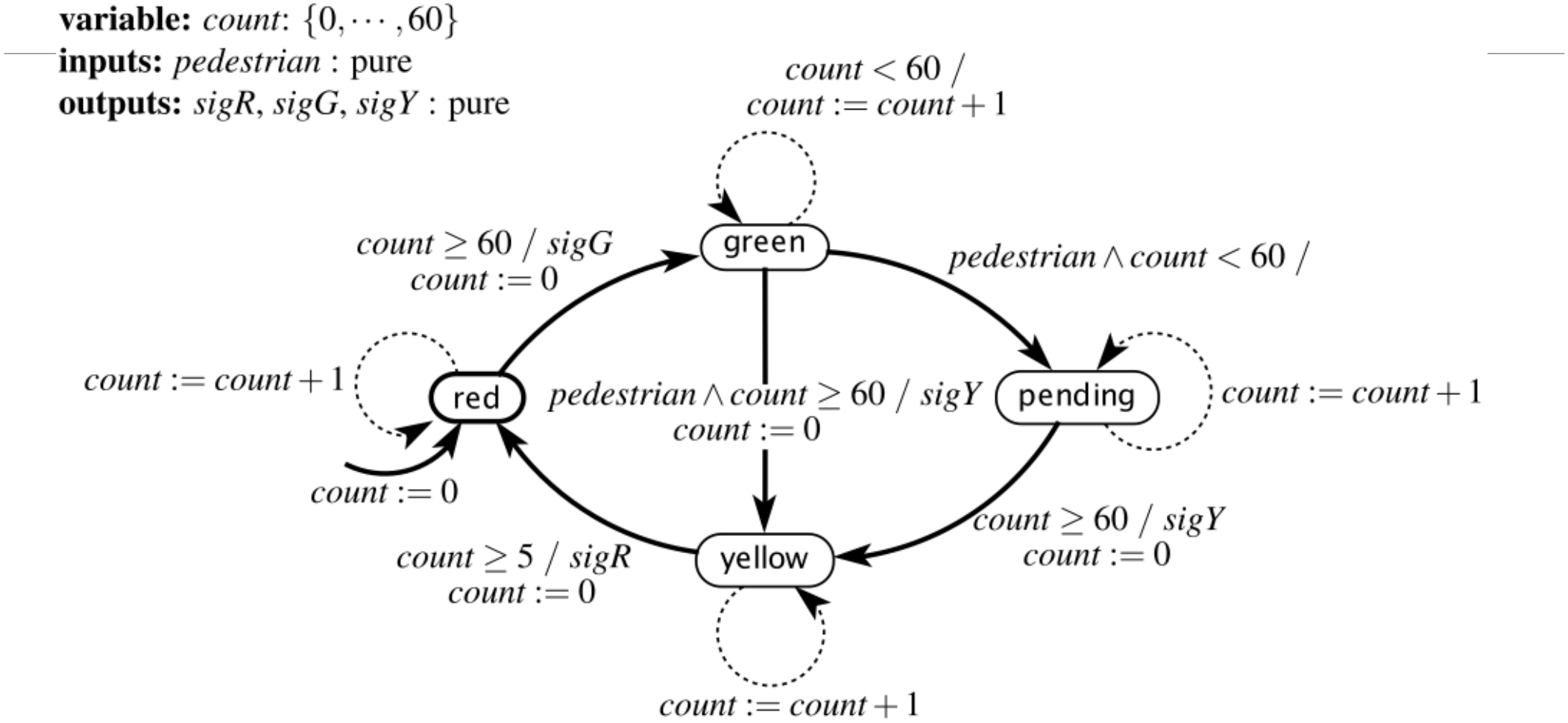
This simple form of hybrid system is called a timed automaton, where the dynamics is just passage of time.

Quiz: How many states does this automaton have?

continuous variable: $x(t) \in \mathbb{R}$
inputs: $click \in \{present, absent\}$
outputs: $single, double \in \{present, absent\}$



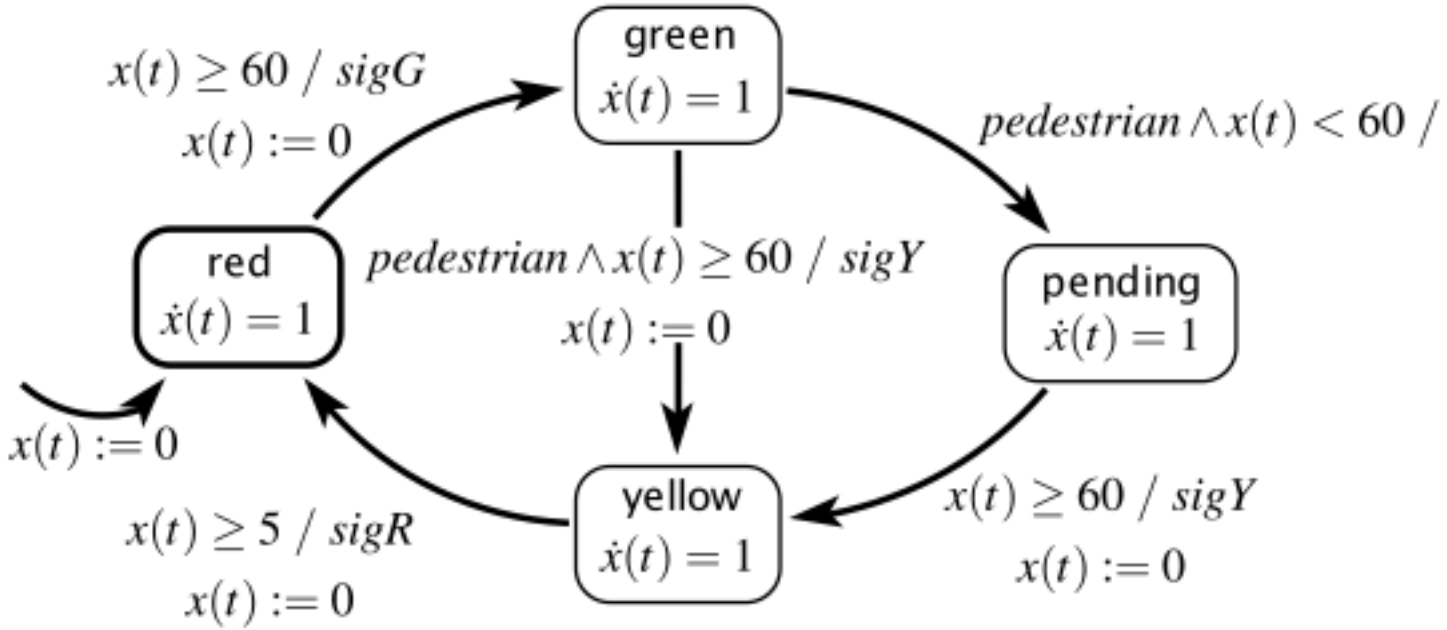
Extended state machine model of a traffic light controller at a pedestrian crossing:



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

Timed automaton model of a traffic light controller

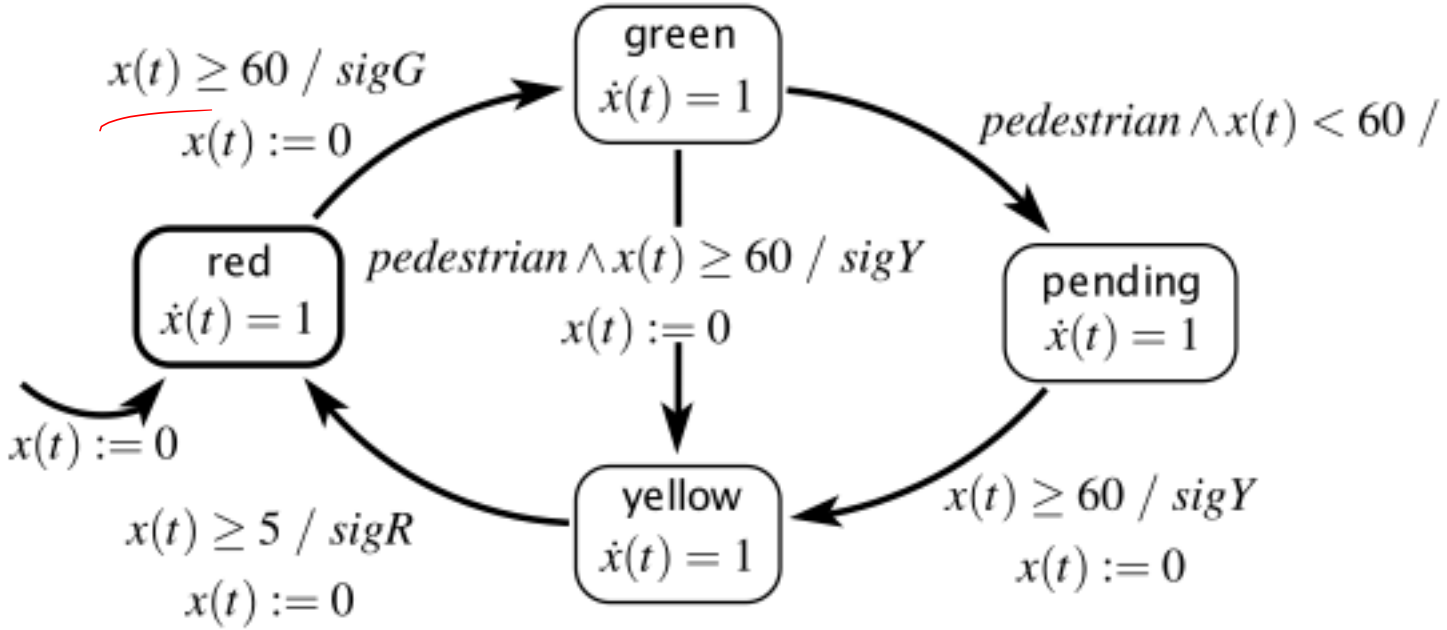
continuous variable: $x(t) : \mathbb{R}$
inputs: *pedestrian*: pure
outputs: *sigR*, *sigG*, *sigY*: pure



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



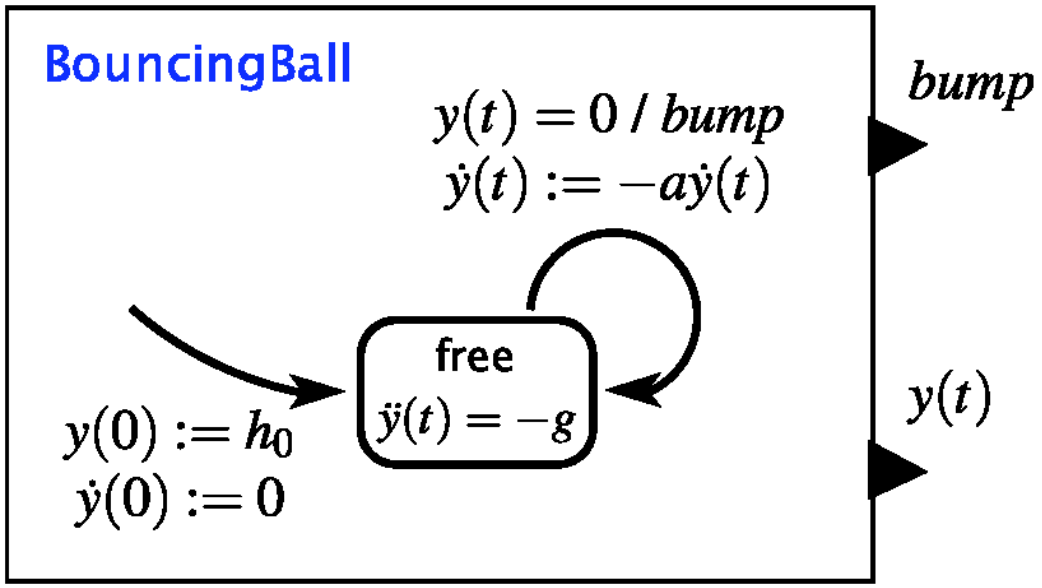
Reactions are occurring continually; with the continuous state variable *x* being continually updated.

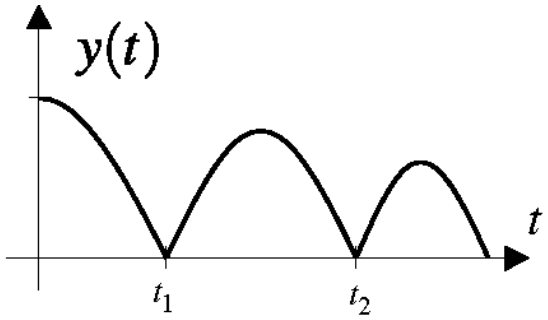
Higher-Order Dynamics

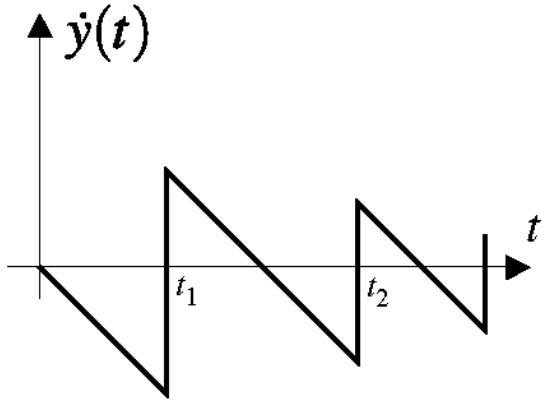
Hybrid systems are much more interesting when the behavior of the refinements is more complex.

The actions on one or more state transitions define the discrete event behavior that combines with the time-based behavior.

Hybrid Automaton for Bouncing Ball

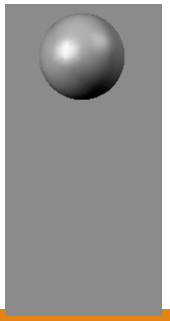






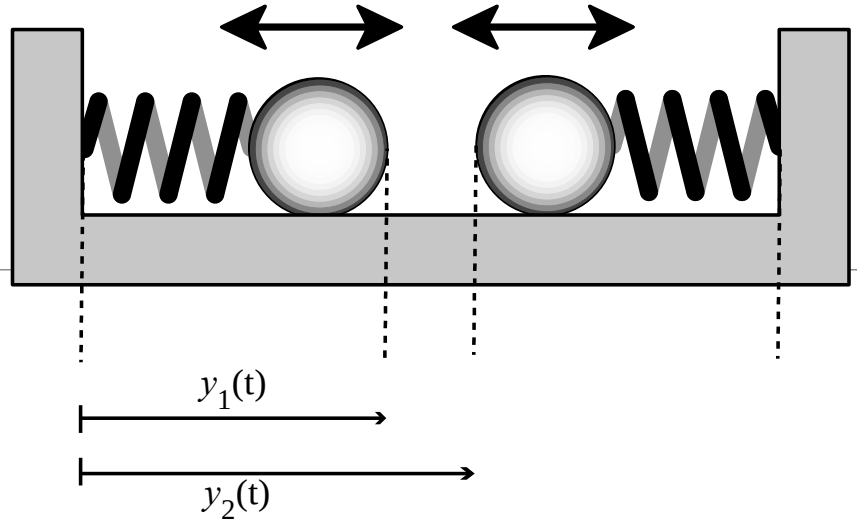
y – vertical distance from ground (position)

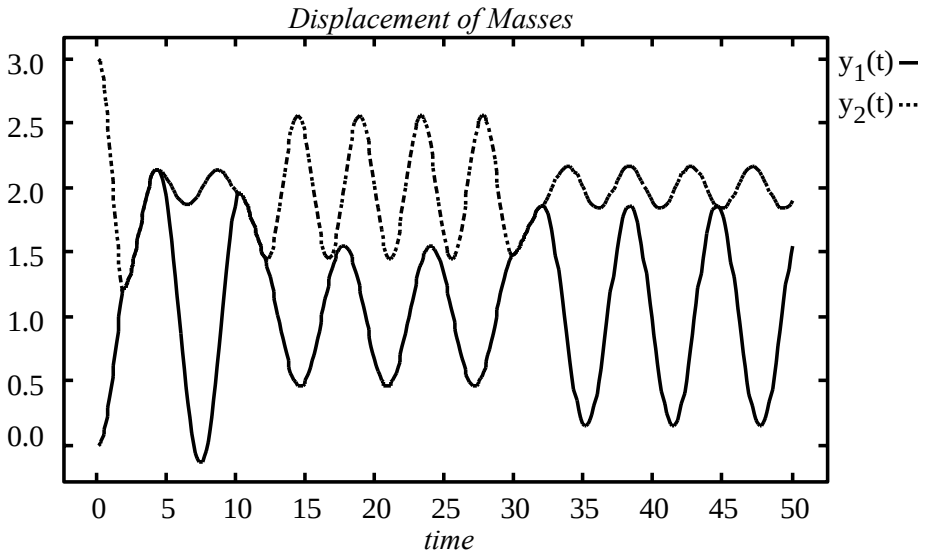
a – coefficient of restitution, $0 \cdot a \cdot 1$



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

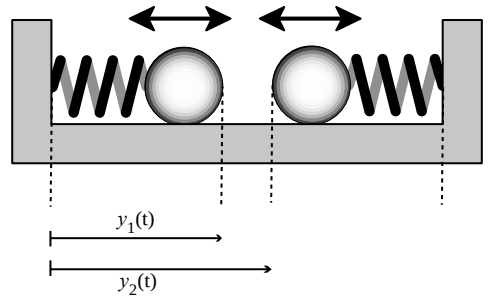
Sticky Masses



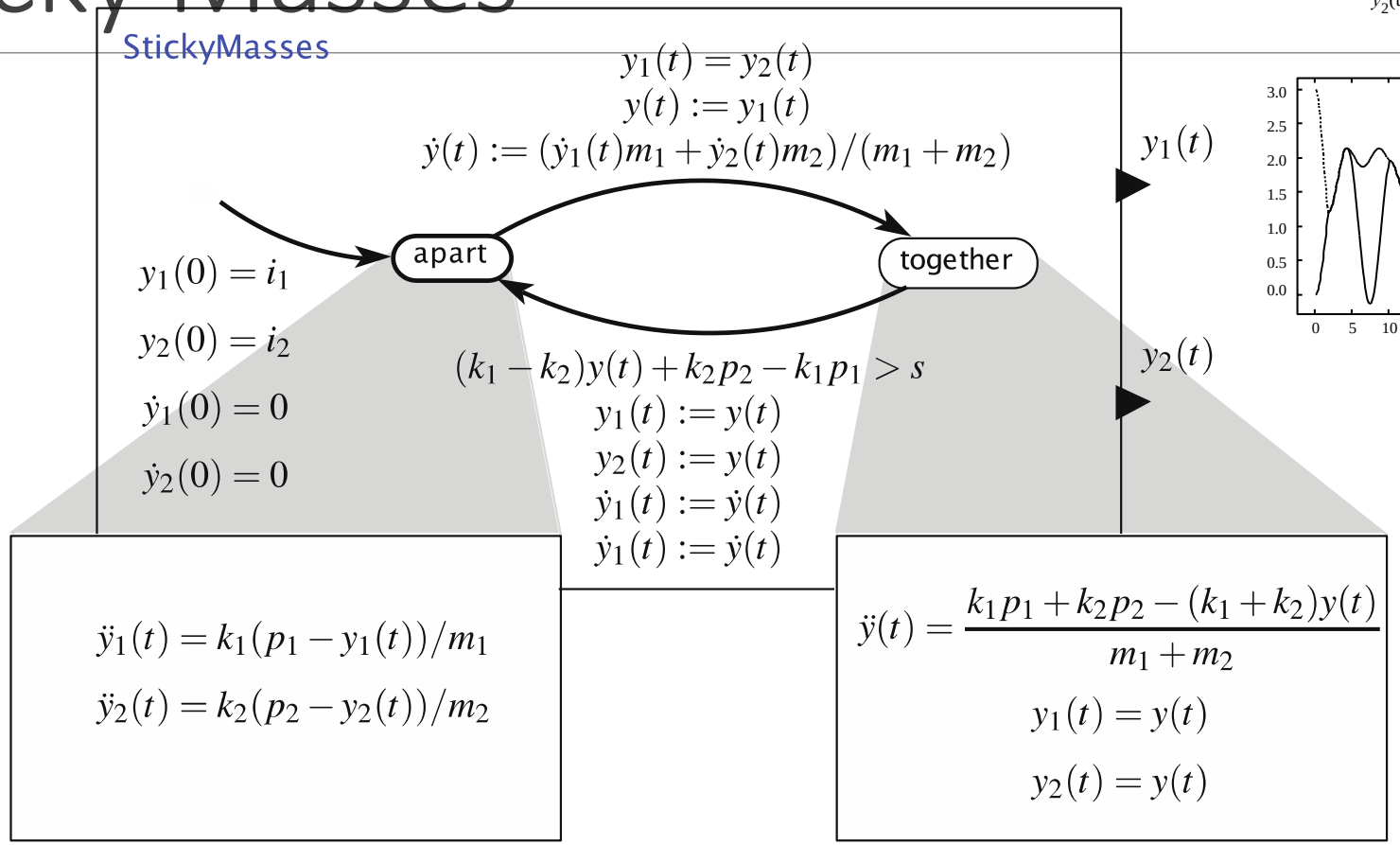


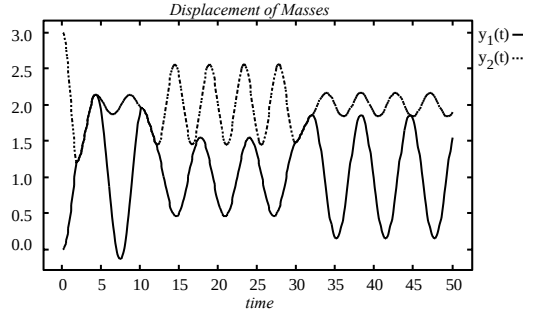
Reading Assignment Example 4.6 at page 88-91

Sticky Masses



StickyMasses





Reading Assignment

Section 4.2.1, 4.2.2 and 4.2.3 from the main Textbook

Reference

- Lee, Edward & Seshia, Sanjit. (2011). Introduction to Embedded Systems - A Cyber-Physical Systems Approach.
- Lecture Note Slides from EECS 149/249A: Introduction to Embedded Systems (UC Berkeley) by Prof. Prabal Dutta and Sanjit A. Seshia