Hybrid Automata

CS60030 FORMAL SYSTEMS

PALLAB DASGUPTA,
FNAE, FASc,
A K Singh Distinguished Professor in AI,
Dept of Computer Science & Engineering
Indian Institute of Technology Kharagpur
Email: pallab@cse.iitkgp.ac.in
Web: http://cse.iitkgp.ac.in/~pallab

INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

FMSAFE
FORMAL METHODS FOR SAFETY CRITICAL SYSTEMS

This presentation acknowledges the contributions of various authors and teachers, including Prof. Rajeev Alur, Prof. Tom Henzinger, and Prof. Goran Frehse

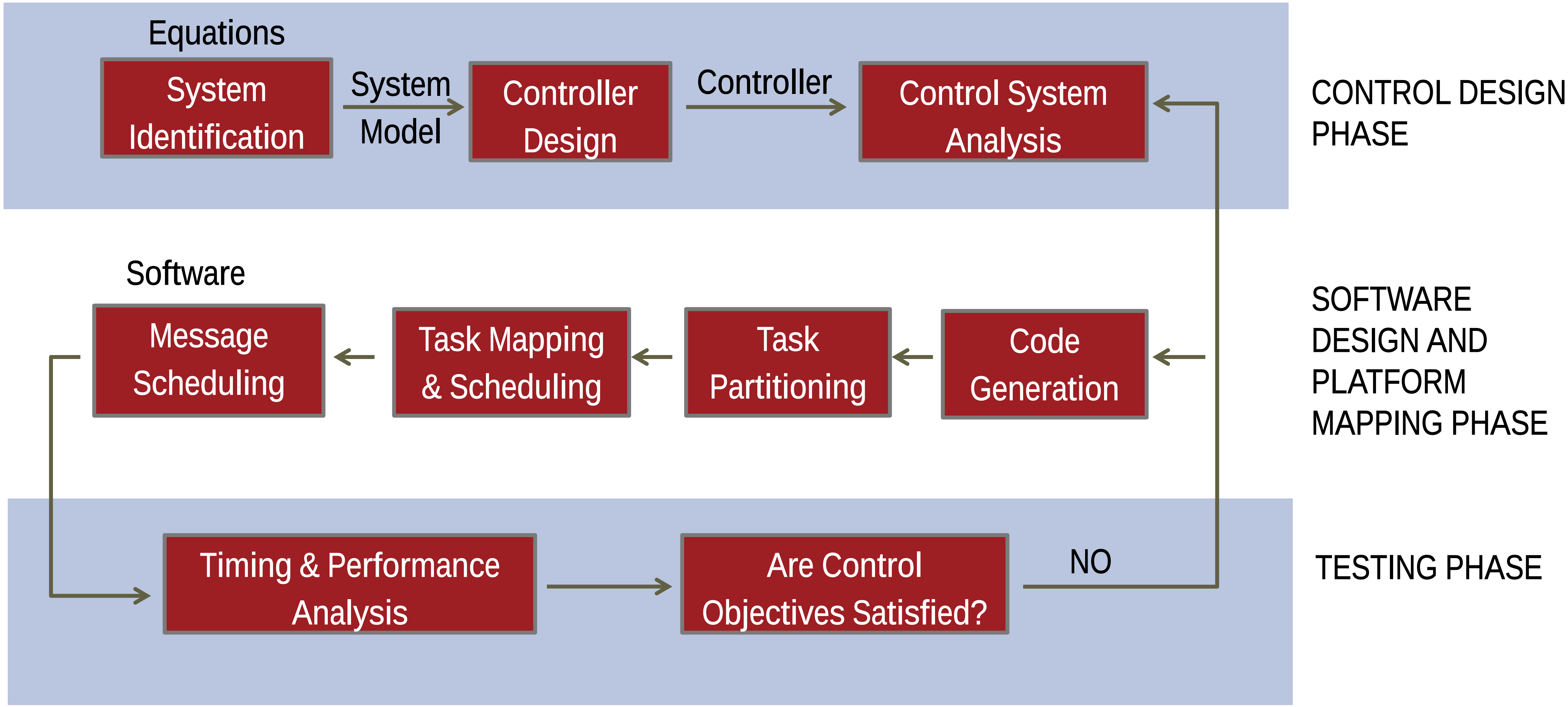
Why are formal methods significant in control domain?

- Today, safety critical control systems:
  - Are designed using CAD tools (with hidden optimization algorithms)
  - Are component based – often from multiple vendors
  - Use electronic components ubiquitously
  - Are often controlled / monitored in real time using embedded software (*cyber-physical systems*)
- Examples:
  - Aircraft stability
  - Electronic braking in automobiles
  - Smart Electrical Grids
  - Atomic reactors
- How to prove that such systems are designed correctly?
- A big challenge, but highly recommended in international safety standards

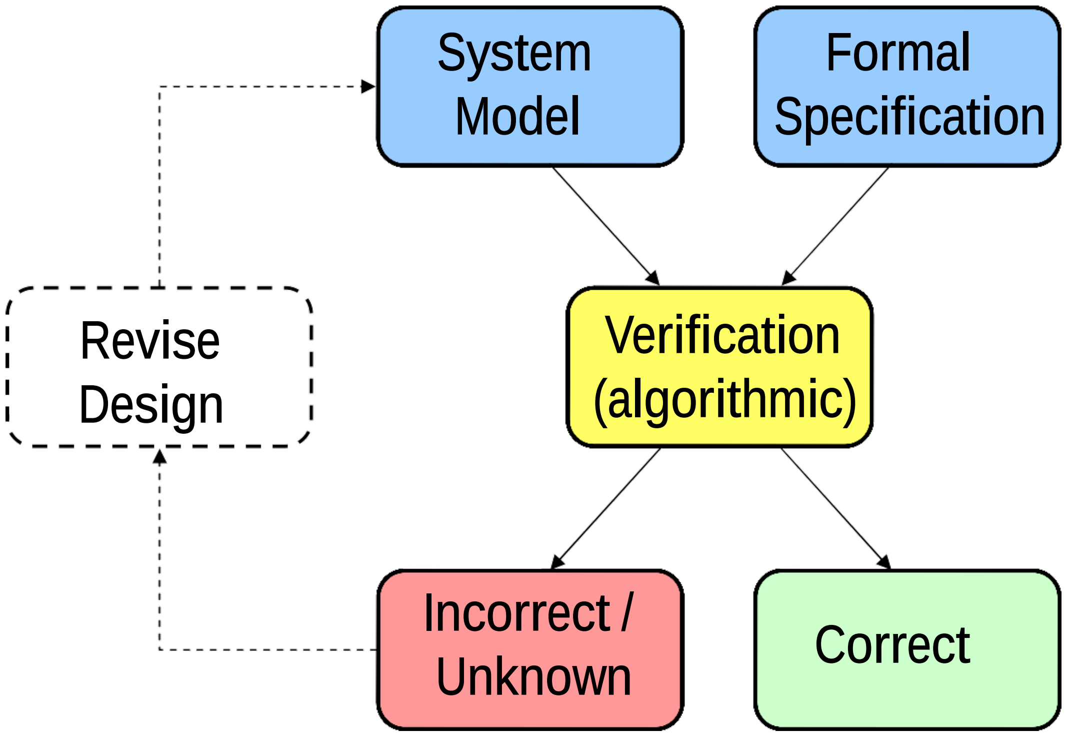
Safety Standards recommending Formal Methods in Verification

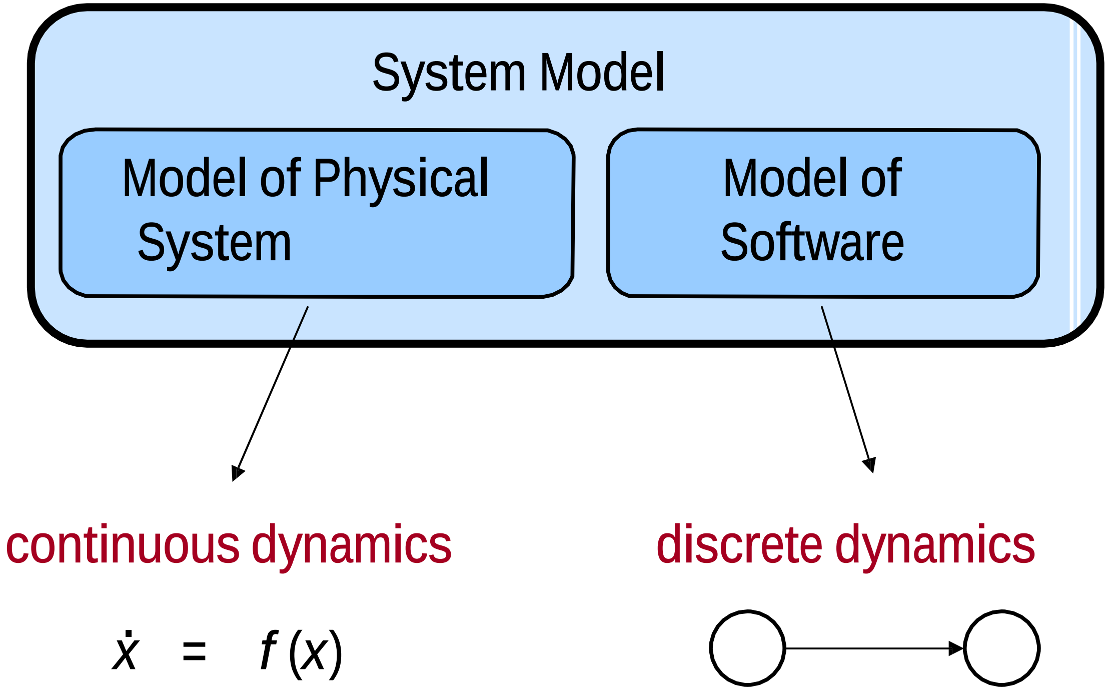
- Aeronautics (DO-178C)
- Automotive (ISO 26262)
- Industrial process automation (IEC 61508)
- Nuclear (IEC 60880)
- Railway (EN 50128)
- Space (ECSS-Q-ST-80C)

Development Cycle of Embedded Control



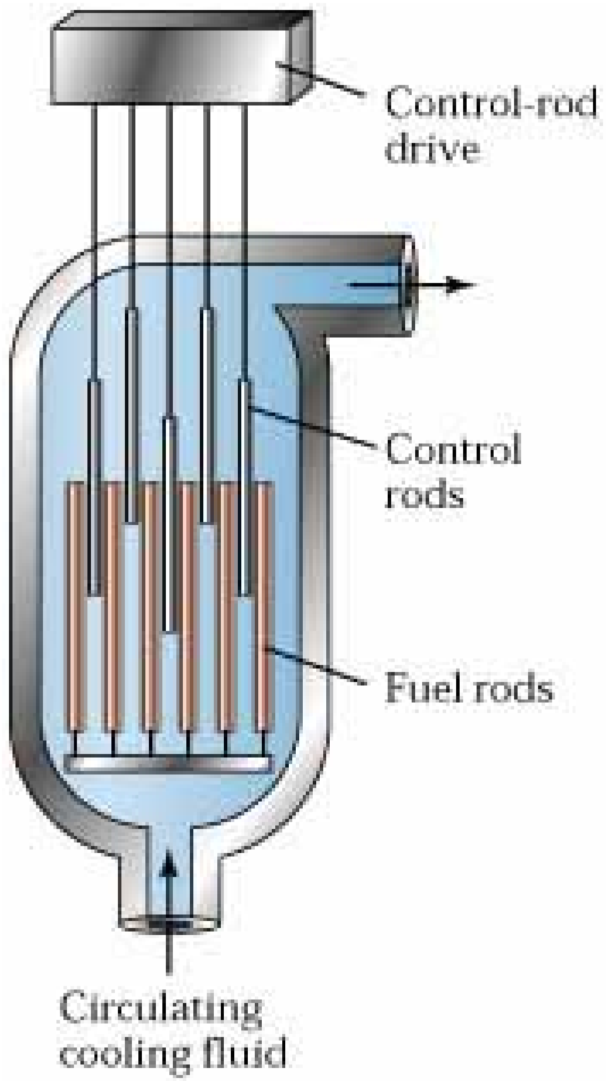
System Modeling and Verification





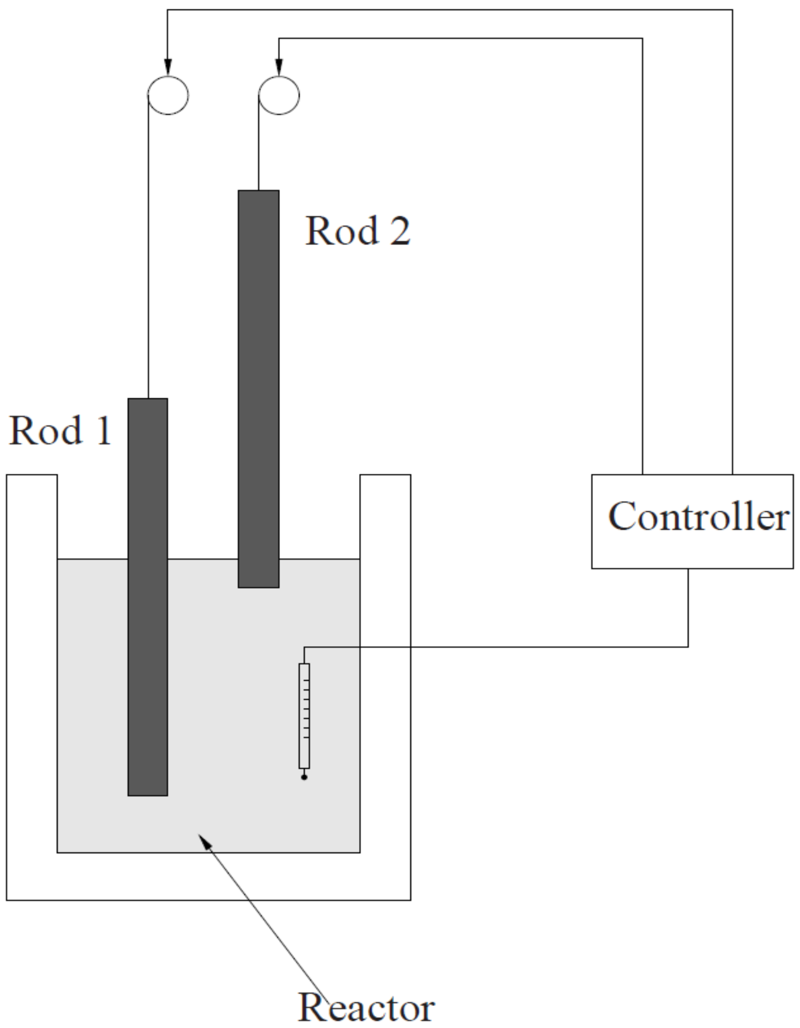
$$\dot{x} = f(x)$$

Example: Cooling of a reactor



- One or more control rods are inserted between fuel rods
  - Controls the neutron flux
  - .. that is, the number of neutrons that split further uranium atoms
- Keeping them inserted for too long slows down the reactor
- Allowing the temperature to rise beyond a level is dangerous
  - There exists points of no return – leads to meltdown

A system with two rods



<u>Three discrete states:</u>

- State-1: None of the two rods are in the reactor
- State-2: Only Rod-1 is in the reactor
- State-3: Only Rod-2 is in the reactor

$x \equiv$ temperature of coolant

Temperature changes in State-1 as per the following equation:

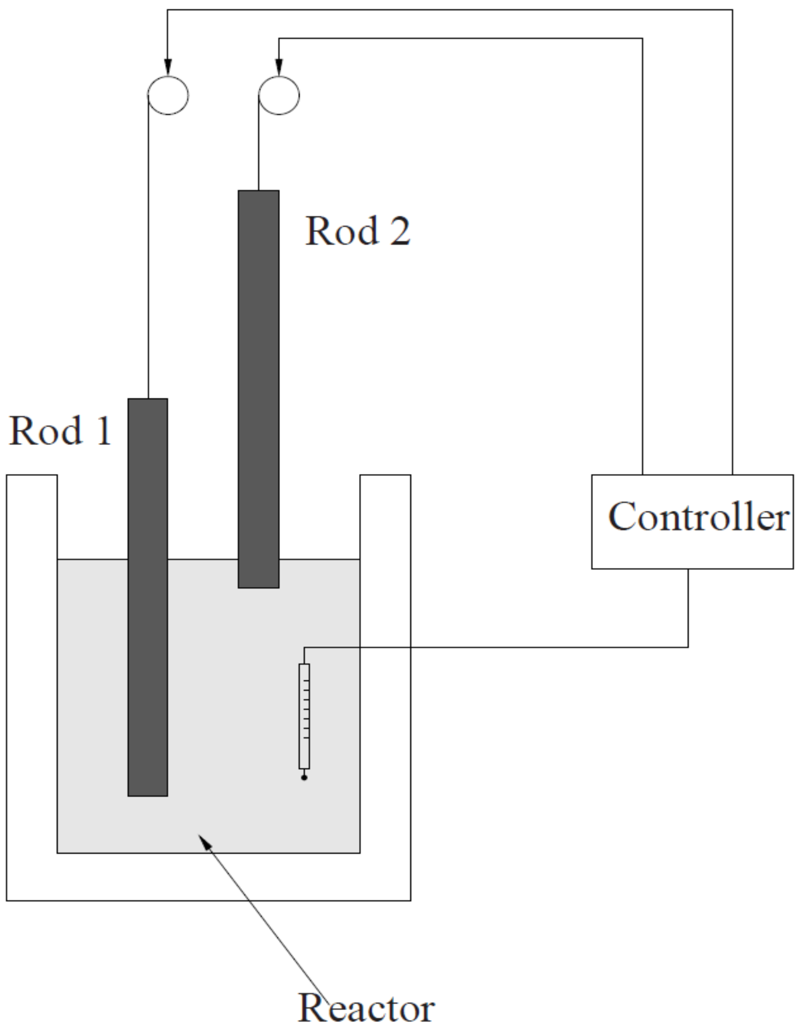
$$\dot{x} = 0.1x - 50$$

Note that:

- When x is above 500, temperature continues to rise
- When x is below 500, temperature continues to fall

If the temperature is allowed to fall below 500, the reactor will shut down.

Effects of control rods



<u>Cooling states:</u>

- State-2: Only Rod-1 is in the reactor
- State-3: Only Rod-2 is in the reactor

Temperature changes in State-2 as per the following equation:

$$\dot{x} = 0.1x - 56$$

Temperature changes in State-3 as per the following equation:

$$\dot{x} = 0.1x - 60$$

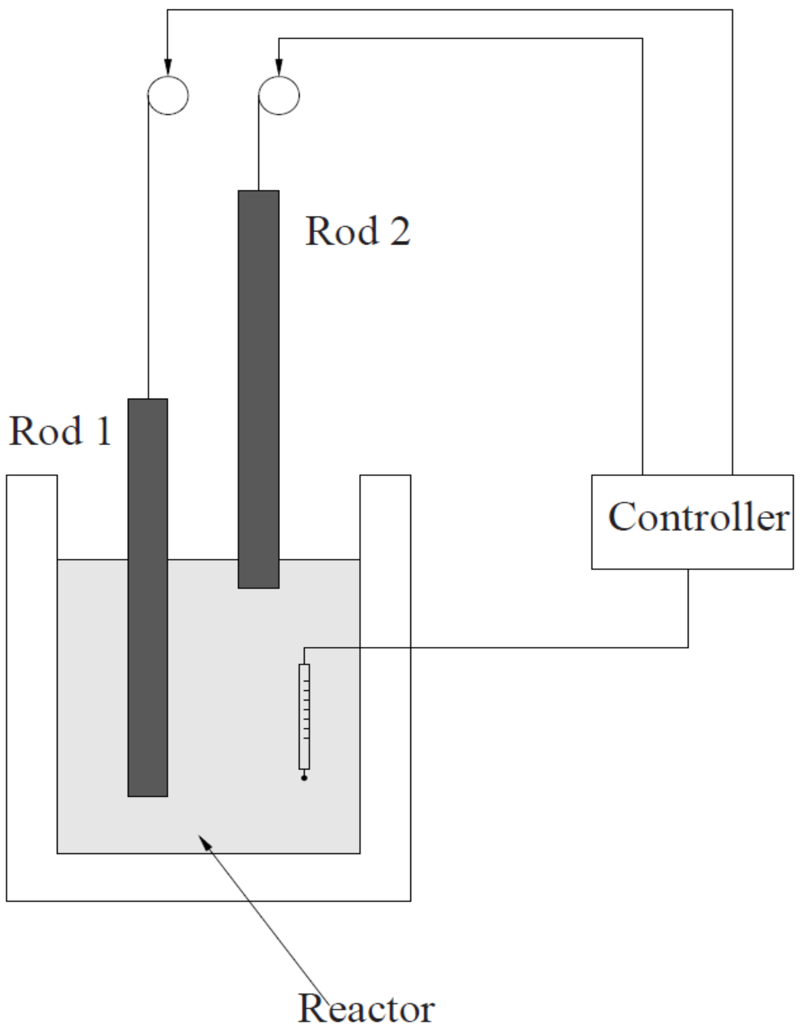
Note that:

- Rod-1 cannot bring the temperature down if it crosses 560
- Rod-2 cannot bring the temperature down if it crosses 600

If temperature crosses 600, meltdown is inevitable

Rod-1 may be inserted when $x < 560$
Rod-2 may be inserted when $x < 600$
.. and they have to be taken out sometime when $x > 500$

Restrictions on control rods



For mechanical reasons, the rods can be lowered into the core only if it has not been there for at least 20 seconds

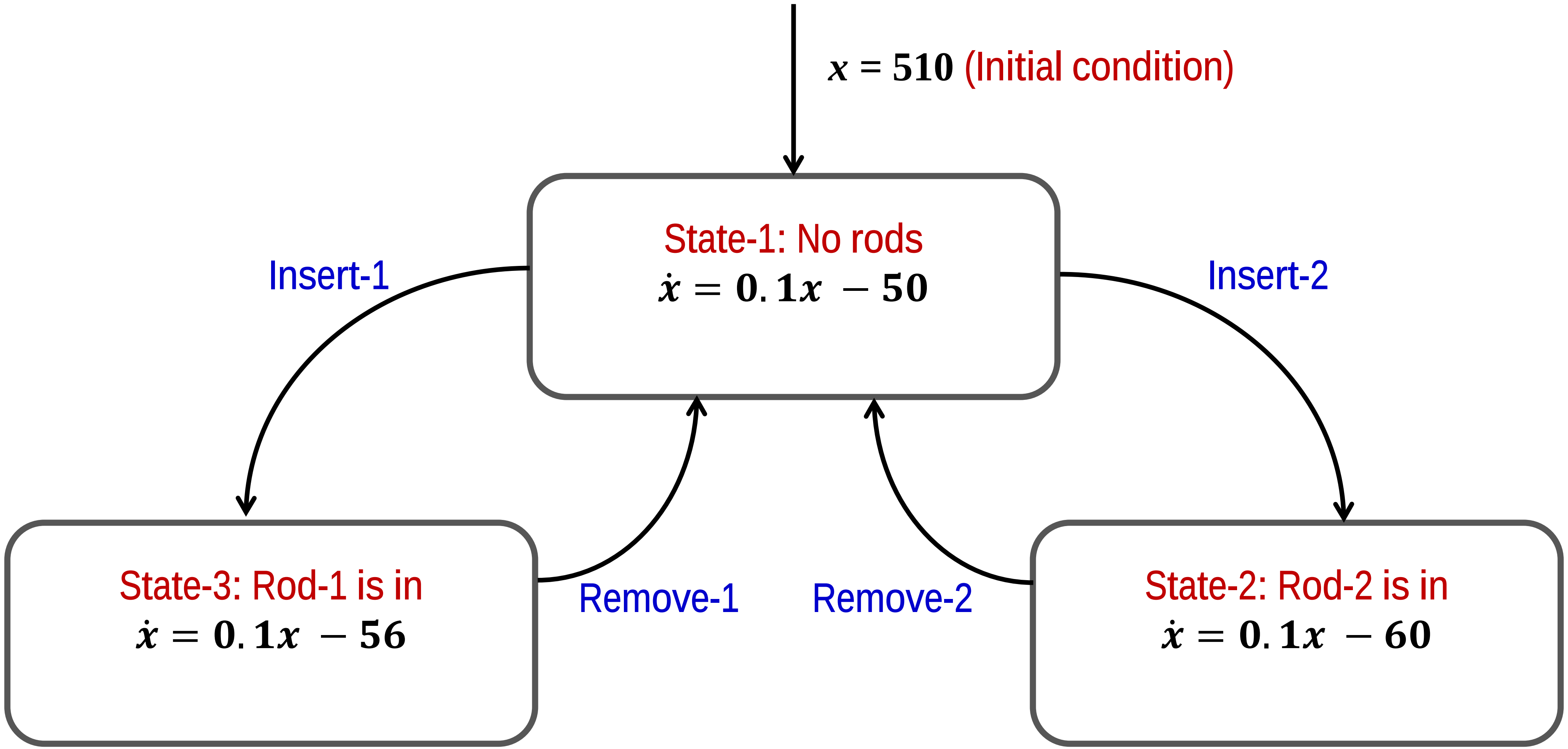
Which means:

- Both rods may become unavailable at some time
- If temperature crosses 600 in the meantime, then consequences are catastrophic

Control problem: Develop a strategy to operate the rods.

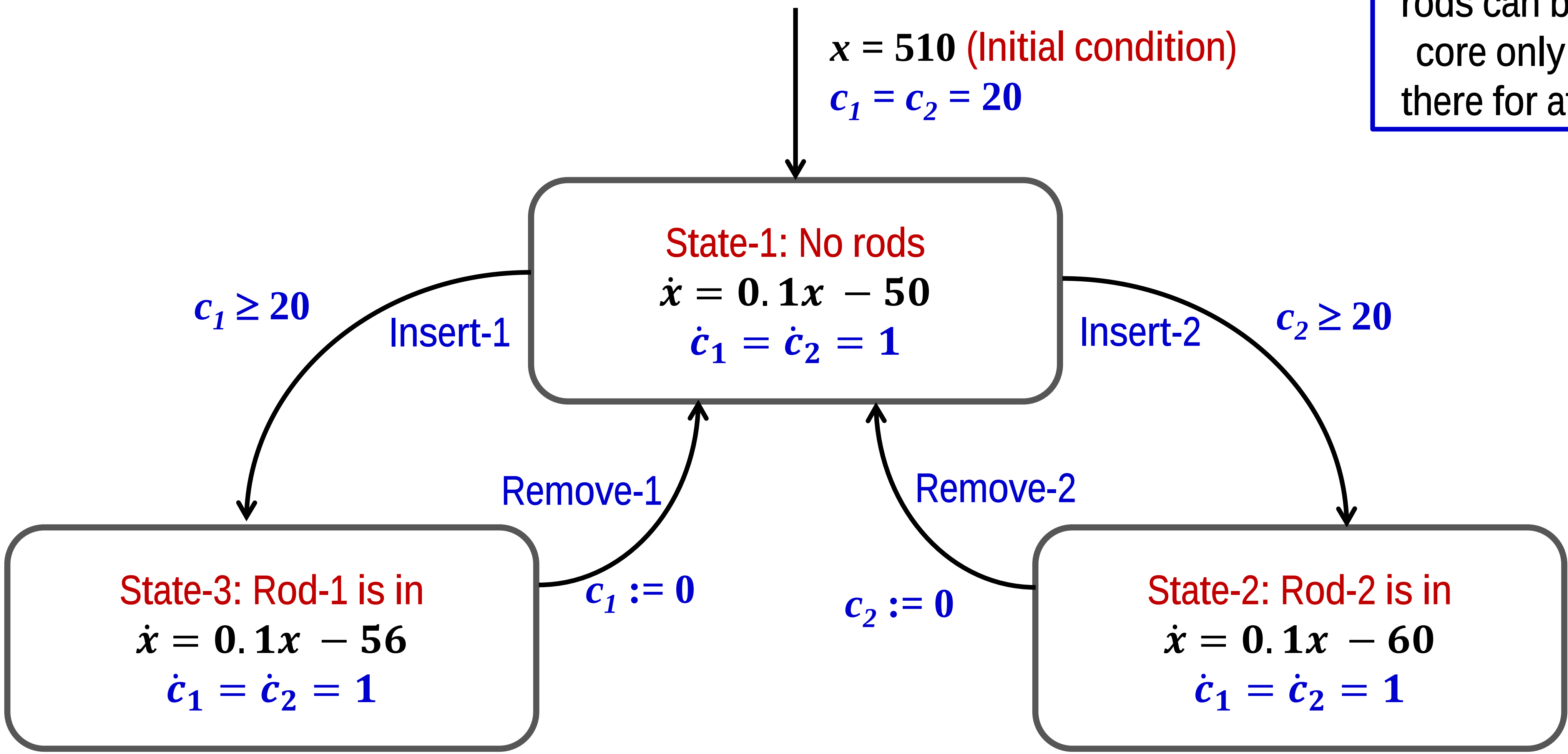
Safety validation problem: Prove that meltdown is never possible under *any* application of that strategy

Hybrid Automaton (Skeleton)



Introducing constraints on rod movement

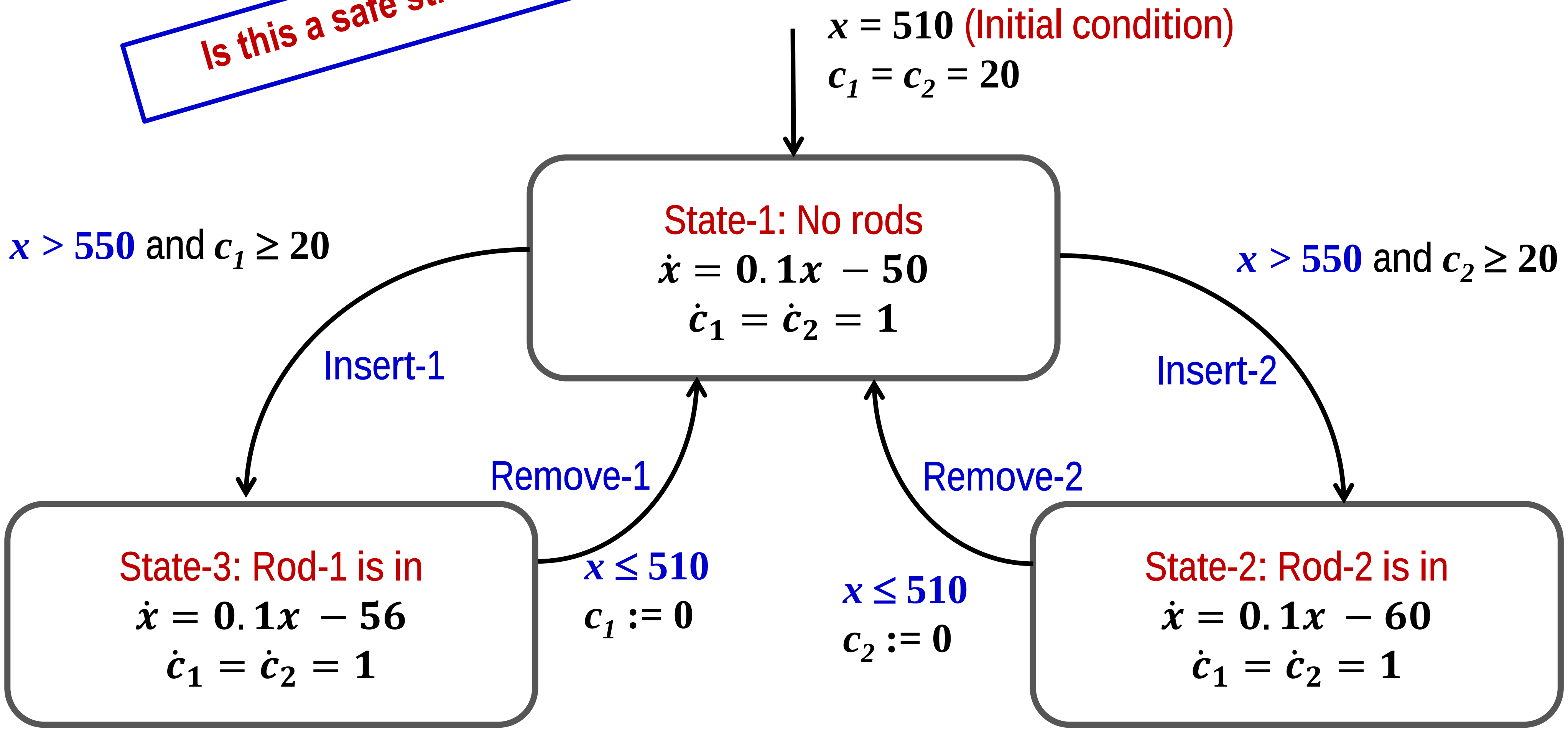
For mechanical reasons, the rods can be lowered into the core only if it has not been there for at least 20 seconds



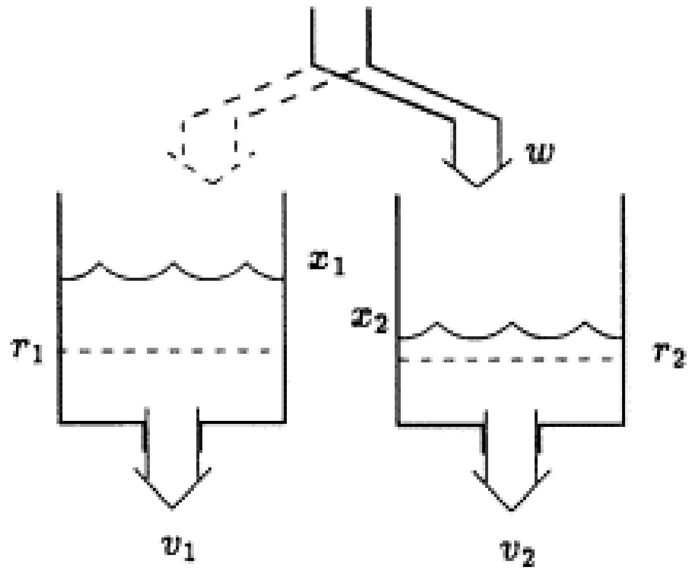
First-cut strategy on Hybrid Automaton

Move an available rod in after temperature rises above 550.
Remove the rod after the temperature drops to 510 or below.

Is this a safe strategy?



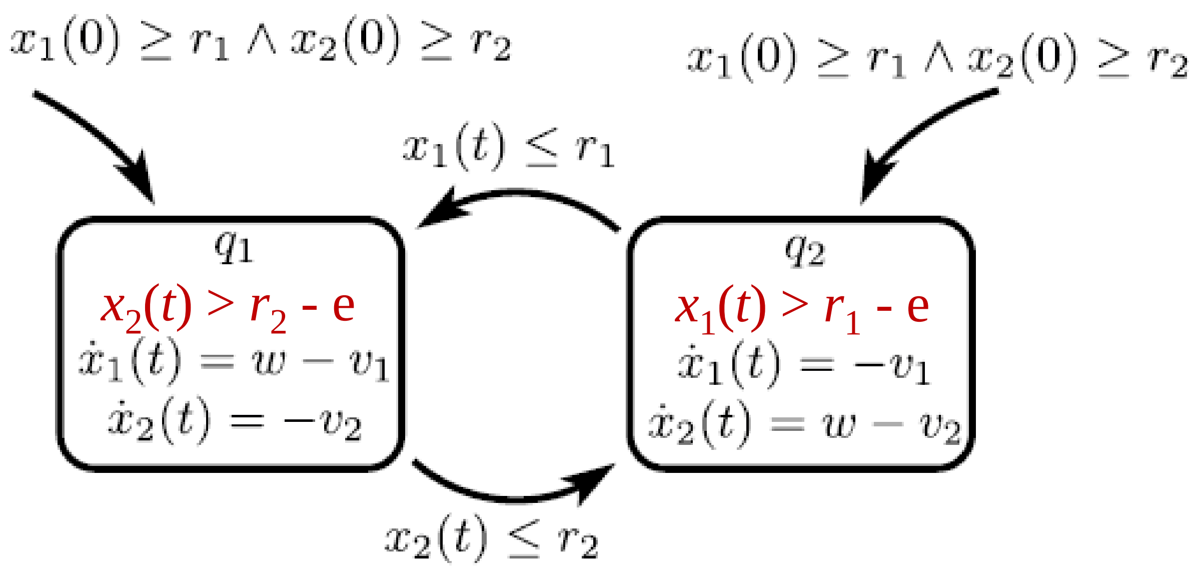
Modeling Exercise-1

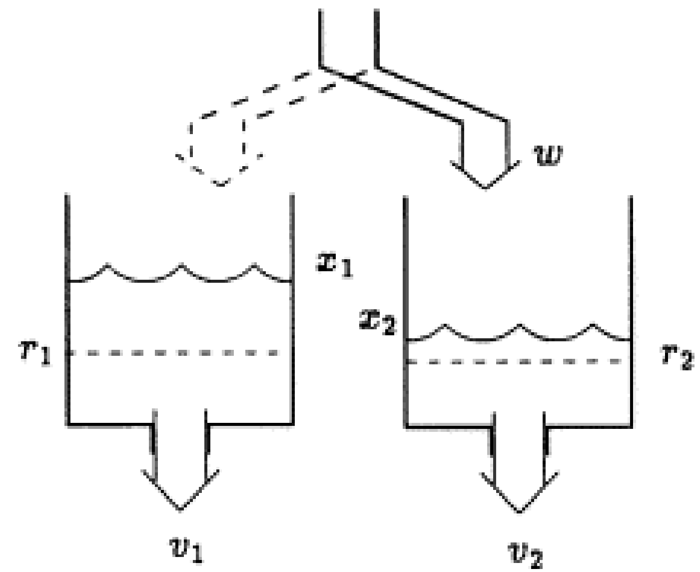


Consider a hybrid system consisting of two tanks containing water.

- Each tank is leaking at a constant rate.
- Water is added at a constant rate, *w*, to the system through a hose, which at any point of time is filling either one tank or the other. It is assumed that the hose can switch between the tanks instantaneously.
- Let x_1 and x_2 denote the volume of water in Tank-1 and Tank-2 respectively.
- Let v_1 and v_2 denote the constant flow of water out of Tank-1 and Tank-2 respectively.
- The objective is to keep the water volumes above r_1 and r_2 respectively, assuming that the water volumes are above r_1 and r_2 initially.
- This is achieved by a controller that switches the inflow to Tank-1 whenever $x_1(t) \leq r_1$ and to Tank-2 whenever $x_2(t) \leq r_2$
- Draw a Hybrid Automaton representing this strategy.

Partial Solution for Modeling Exercise-1

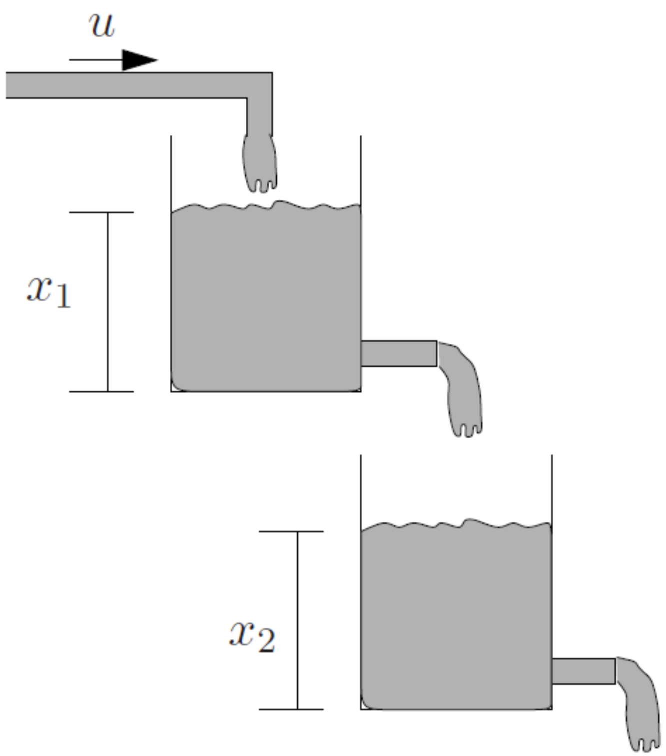




- The controller can switch the inflow to Tank-1 when $\boldsymbol{x_1(t) \leq r_1}$ and to Tank-2 when $\boldsymbol{x_2(t) \leq r_2}$
- Does this automaton admit zeno behaviors? If so, how shall we eliminate them?
- Do we need location invariants?

The automaton will have no (infinite) run if $\boldsymbol{w < v_1 + v_2}$

Modeling Exercise-2



There are three taps in the system, namely:

- Tap-1 having a flow rate of $u = 5$
- Tap-2 having a flow capacity of $v = 2$
- Tap-3 having a flow capacity of $w = 4$
- Tap-2 and Tap-3 are always on.
- Tap-1 is switched on when $x_1 + x_2$ falls below 10 and is switched off when x_1 exceeds 80.
- Initially, we have $x_1 = 50$ and $x_2 = 50$.

Draw a hybrid automaton for the system. Explain the dynamics of the system.

[Hint: Note that the outflow of Tank-2 changes discretely if it becomes empty before Tank-1.]

Modeling Exercise-3

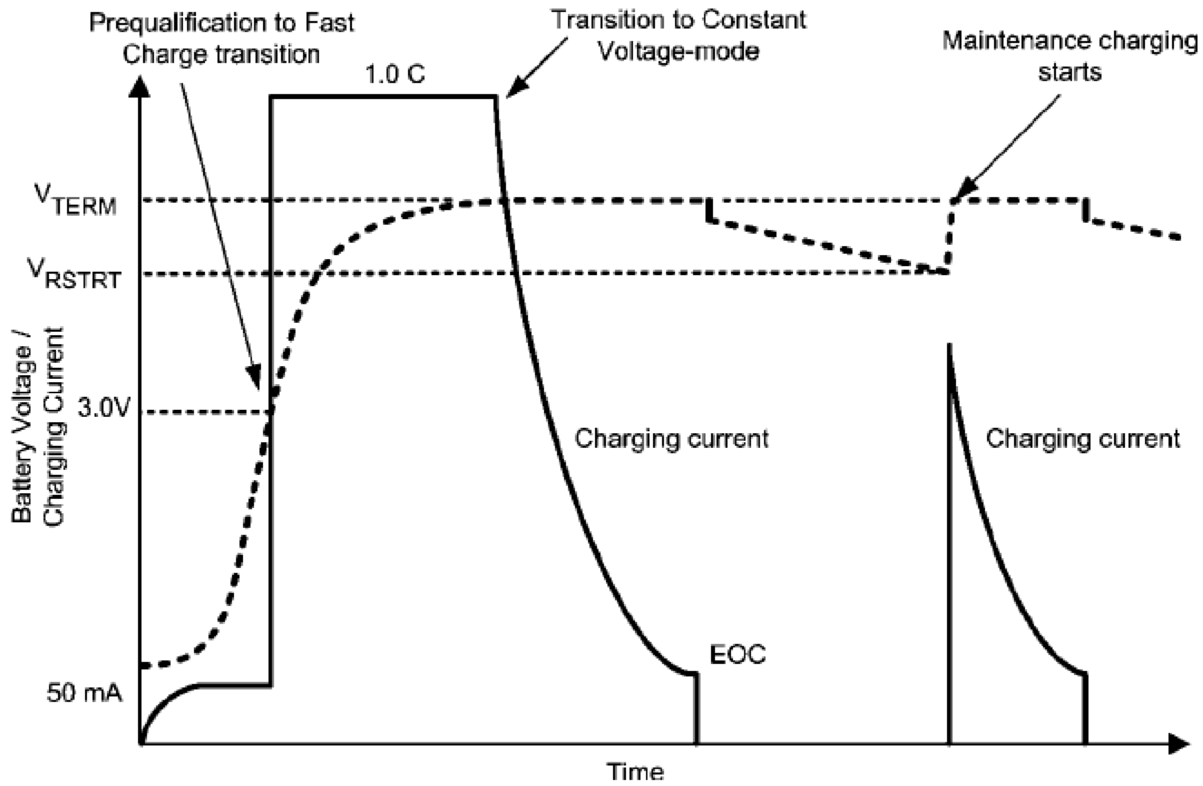
Two trains are heading toward each other on a single track at constant speeds:

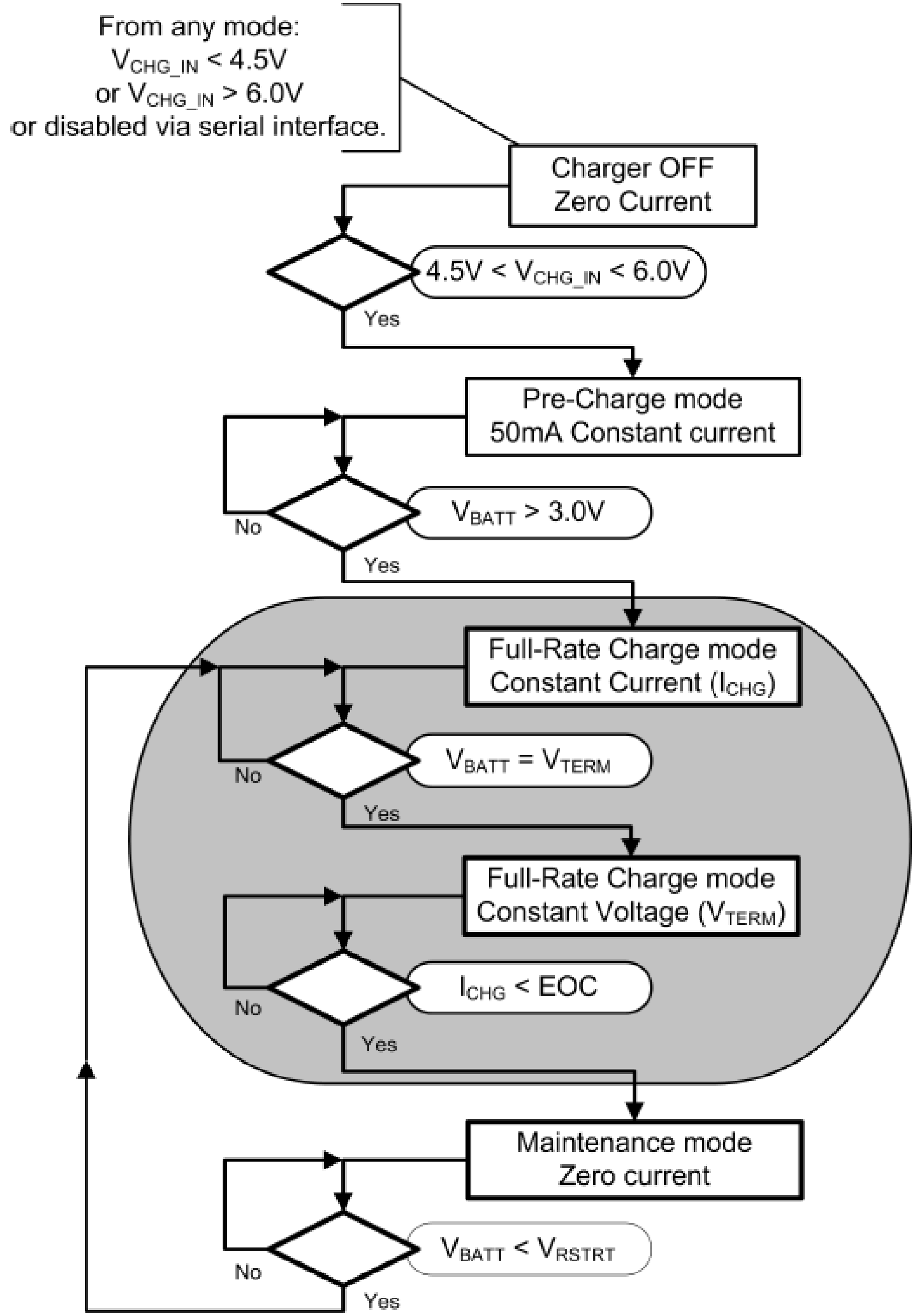
- Train *E* is travelling east at a fixed speed v_e, and the train *W* is travelling west at a fixed speed v_w.
- A bird *B* is initially travelling east at a fixed speed v_b along the line joining the two trains.
  - When the bird reaches the train *W*, it reverses its direction, heads west at the same speed v_b, and reverses its direction again when it reaches the train *E*. This cycle repeats.

Model the scenario as a hybrid automaton.

- The hybrid automaton can have two locations, one each corresponding to the direction in which the bird is travelling, and three state variables that capture the positions of the train E, the train W, and the bird B.
- Draw the hybrid automaton showing the transition guards, location invariants, and the location dynamics.

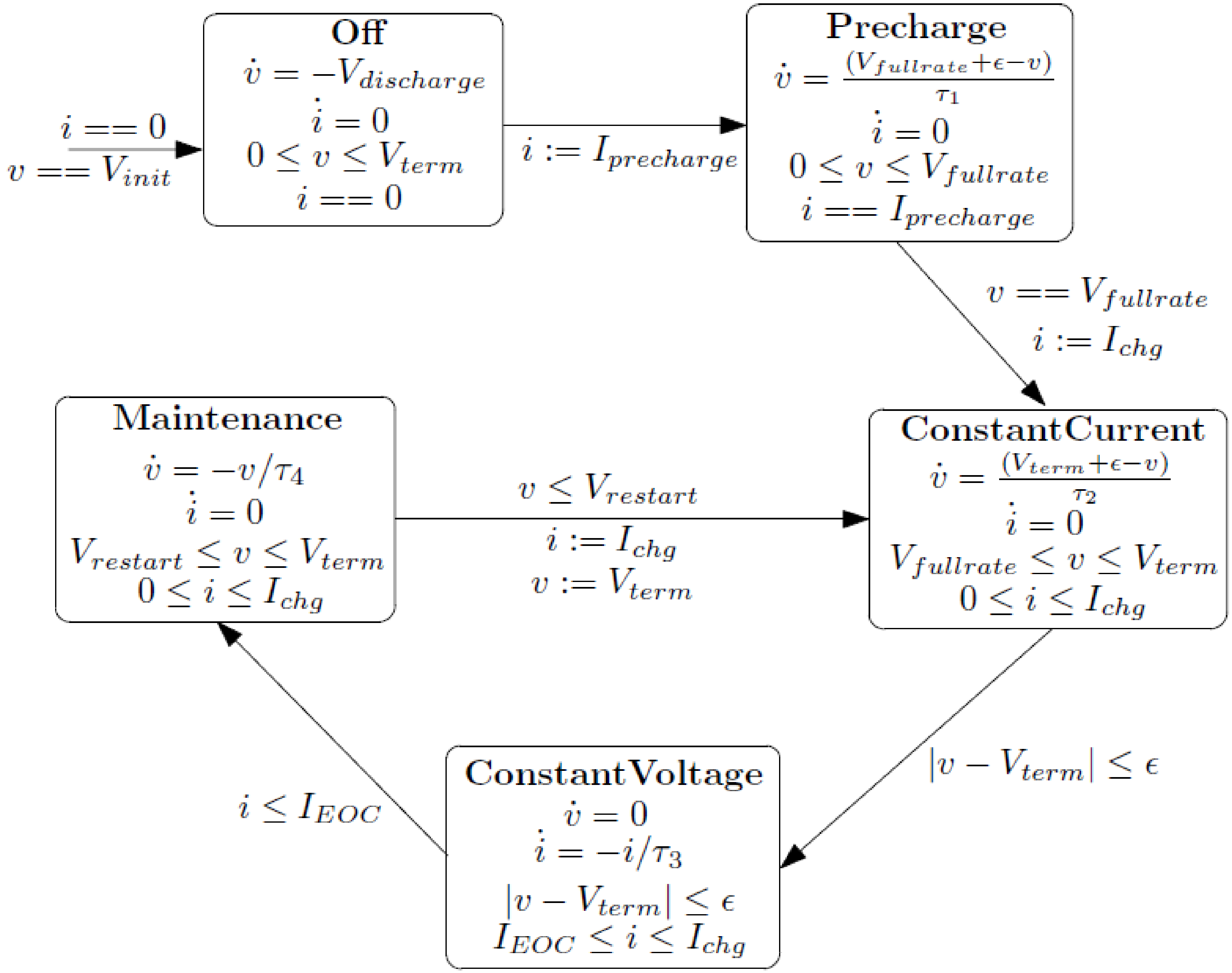
Another Example: Battery Charger





Hybrid Automaton of a Battery Charger

- Modes of Operation:
  - Off
  - Precharge
  - Constant Current
  - Constant Voltage
  - Maintenance
- Model must specify:
  - Dynamics of each mode
  - Transition guards



Formal Model for Hybrid System

Hybrid automaton: $H = (Loc, Var, Lab, Edg, Act, Inv)$ consists of:

- A finite set *Loc* of *locations*.
- A finite set *Var* of real valued *variables*.
  - We write V for the set of valuations. A valuation $\boldsymbol{v}$ is a function that assigns a real-value $\boldsymbol{v}(x) \in \mathrm{R}$ to each variable $x \in Var$.
  - A state is a pair $(\boldsymbol{\ell}, \boldsymbol{v})$ consisting of a location $\boldsymbol{\ell} \in Loc$ and a valuation $\boldsymbol{v} \in V$.
- A finite set *Lab* of *synchronization labels*.
  - *Lab* necessarily contains the stutter label τ, i.e. $\tau \in Lab$.
- A finite set *Edg* of edges called *transitions*. The next slide elaborates the types of transitions.
- A labeling function *Act* that assigns to each location $\boldsymbol{\ell} \in Loc$ a set of *activities*.
  - Each activity is a function from the nonnegative reals $R^{\geq 0}$ to V.
  - The activities of at location are *time-invariant*.
- A labeling function *Inv* that assigns to each location $\boldsymbol{\ell} \in Loc$ an *invariant* $Inv(\boldsymbol{\ell}) \subseteq V$.
  - The system may stay at a location only if the location invariant is true; that is, some discrete transition must be taken before the invariant becomes false.

Transitions of a Hybrid Automaton

- Each transition $e = (\ell, a, \mu, \ell')$ consists of :
  - A source location $\ell \in Loc$,
  - *A target location* $\ell' \in Loc$,
  - A synchronization label $a \in Lab$
  - A transition relation $\mu \subseteq V^2$

- For each location $\ell \in Loc$ there is a set $con \subseteq Var$ of control variables and a stutter transition of the form $(\ell, \tau, ID_{con}, \ell)$, where $(\nu, \nu') \in ID_{con}$ iff for all variables $x \in Var$, either $x \notin con$ or $\nu(x) = \nu'(x)$. In other words, time may pass without any update on the control variables.

- The transition e is enabled in a state (ℓ, ν) if for some valuation $\nu' \in V$, $(\nu, \nu') \in \mu$. The state (ℓ', ν') is then said to be a *transition successor* of (ℓ, ν).

Time Deterministic Hybrid System

- A hybrid system H is *time-deterministic* if for every location $\ell \in Loc$ and every valuation $\nu \in V$, there is at most one activity $f \in \boldsymbol{Act(\ell)}$ with $f(0) = \nu$.
- The activity f, then, is denoted by $\varphi_\ell[\nu]$.

The runs of a hybrid system

The state of a hybrid system can change in two ways:

- By a *discrete* and *instantaneous* transition that changes both the control location and the values of the variables according the transition relation
- By a *time delay* that changes only the values of the variables according to the activities of the current location.

A run of the hybrid system *H*, then, is a finite or infinite sequence, $\rho : \sigma_0 \mapsto^{t_0}_{f_0} \sigma_1 \mapsto^{t_1}_{f_1} \sigma_2 \mapsto^{t_2}_{f_2} \dots$

of states $\sigma_i = (\ell_i, \nu_i) \in \Sigma$, nonnegative reals $t_i \in R^{\geq 0}$, and activities $f \in Act(\ell_i)$, such that for all $i \geq 0$:

1. $f_i(0) = \nu_i$
2. For all $0 \leq t \leq t_i$, $f_i(t) \in Inv(\ell_i)$
3. The state σ_{i+1} is a transition successor of the state, $\sigma'_i = (l_i, f_i(t_i))$
   - The state σ'_i is called a *time successor* of the state σ_i
   - The state σ_{i+1} is called a *successor* of σ_i.

We write [*H*] for the set of runs of the hybrid system *H*.

Hybrid Systems as Transition Systems

With a hybrid system H, we associate the labeled transition system $\tau_H = (\Sigma, Lab'' \cup \mathbb{R}^{\geq 0}, \rightarrow)$, where the *step relation* $\rightarrow$ is the union of the following two:

- The transition-step relations $\rightarrow^a$, for $a \in Lab$,

$$\frac{(\ell, a, \mu, \ell') \in Edg \qquad (v, v') \in \mu \qquad v, v' \in Inv(\ell)}{(\ell, v) \rightarrow^a (\ell', v')}$$

- The time-step relations $\rightarrow^t$, for $t \in \mathbb{R}^{\geq 0}$

$$\frac{f \in Act(\ell) \qquad f(0) = v \qquad \forall 0 \leq t' \leq t.\, f(t') \in Inv(\ell)}{(\ell, v) \rightarrow^a (\ell, f(t))}$$

Hybrid Systems as Transition Systems

- The stutter transitions ensure that the transition system τ_H is reflexive. For all states σ, σ', $\in$, Σ, where $\sigma = (\ell, v)$ and for all $t \in R^{\geq 0}$,

$$\exists f \in Act(\ell), \boldsymbol{\sigma} \mapsto^t_f \boldsymbol{\sigma}' \quad \textit{iff} \quad \exists \boldsymbol{\sigma}'' \in \Sigma, \; a \in \textit{lab}. \boldsymbol{\sigma} \to^t \boldsymbol{\sigma}'' \to^a \boldsymbol{\sigma}'$$

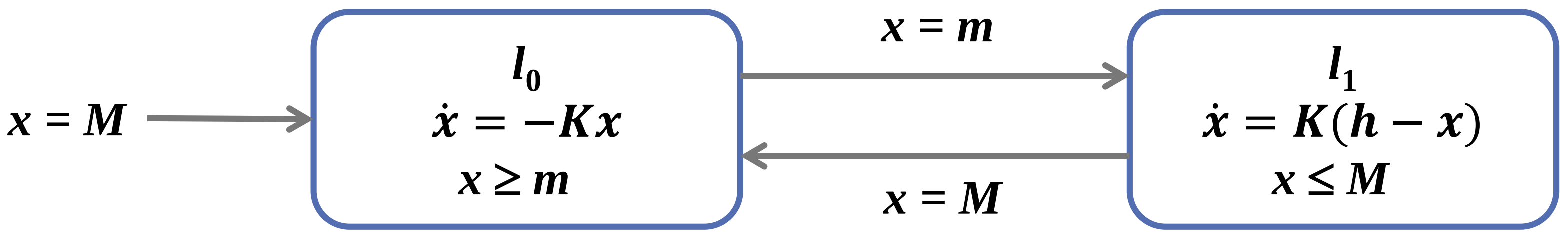
- It follows that for every hybrid system, the set of runs is closed under prefixes, suffixes, stuttering, and fusion.
- For time-deterministic hybrid systems, Time can progress by the amount $t \in R^{\geq 0}$ from the state (ℓ, v) if this is permitted by the invariant of location ℓ; that is :

$$tcp_\ell[v](t) \;\; \text{iff} \;\; \forall 0 \leq \mathrm{t}' \leq t.\, \phi_\ell[v](t') \in Inv(\ell)$$

- We can rewrite the time-step rule for time-deterministic systems as: $\dfrac{tcp_\ell[v](t)}{(\ell, v) \to^t (\ell, \varphi_\ell[v](t))}$

Example: Thermostat

- When the heater is off, the temperature: $x(t) = \theta e^{-Kt}$
- When the heater is on: $x(t) = \theta e^{-Kt} + h\left(1 - e^{-Kt}\right)$
- The resulting time-deterministic hybrid system is shown below:



Linear Hybrid Systems

- A *linear term* over the set *Var* of variables is linear combination of the variables in *Var* with integer coefficients.
- A *linear* formula over *Var* is a Boolean combination of inequalities between linear terms over *Var.*
- The time-deterministic hybrid system *H = (Loc Var, Lab, Edg, Act, Inv)* is linear if its activities, invariants, and transition relations can be defined by linear expressions over the set *Var* of variables:
  - For all locations $\ell \in Loc$, the activities Act(ℓ) are defined by a set of differential equations of the form $\dot{x} = k_x$, one for each variable $x \in Var$, where $k_x \in \mathbb{Z}$ is an integer constant
  - For all valuations $v \in V$, variables $x \in Var$, and nonnegative reals $t \in \mathrm{R}^{\geq 0}$

$$\phi_\ell^x[v](t) = v(x) + k_x \cdot t$$

Linear Hybrid Systems

- For all location $\ell \in Loc$ the invariant $Inv(\ell)$ is defined by a linear formula ψ over Var.

$$v \in Inv(\ell) \quad iff \quad v(\psi)$$

- For all transitions $e \in Edg$ the transition relation μ is defined by a guarded set of nondeterministic assignments.

$$\psi \Rightarrow \{x := [\alpha_x, \beta_x] \mid x \in \mathrm{Var}\}.$$

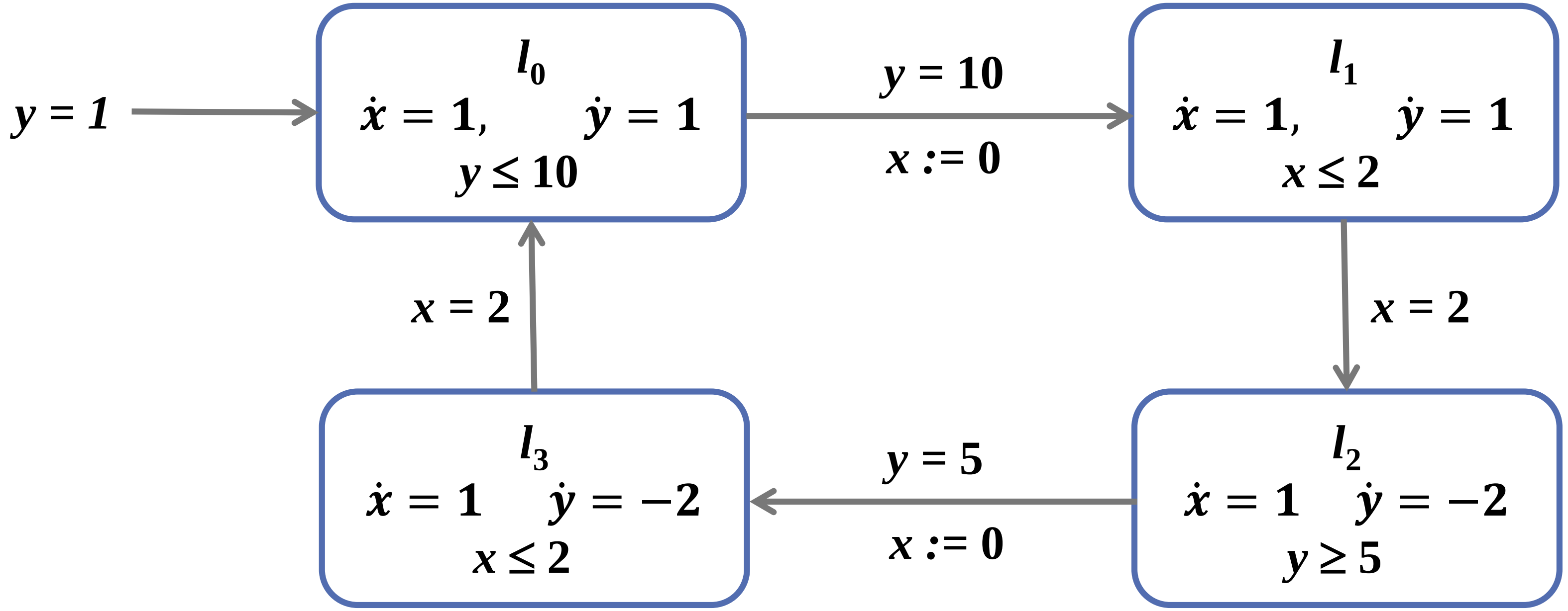
Here, the guard ψ is a linear formula, and both interval boundaries α_x and β_x are linear terms for each variable $x \in Var$:

$$(v, v') \in \mu \quad iff \quad v(\psi) \wedge \forall x \in Var.v(\alpha_x) \leq v'(x) \leq v(\beta_x)$$

Examples: A Water-Level Monitor

The water level in a tank is controlled through a monitor, which continuously senses the water level and turns a pump on and off. The water level changes as a piecewise-linear function over time.

- When the pump is off, the water level, denoted by the variable y, falls by 2 inches per second
- When the pump is on, the water level rises by 1 inch per second.



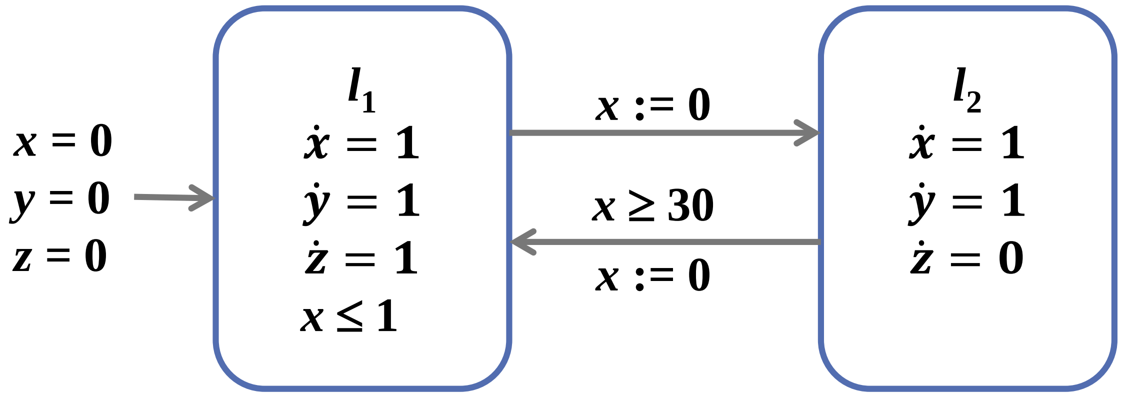
Suppose that initially the water level is 1 inch and the pump is turned on. We wish to keep the water level between 1 and 12 inches. But from the time that the monitor signals to change the status of the pump to the time that the change becomes effective, there is a delay of 2 seconds. Thus the monitor must signal to turn the pump on before the water level falls to 1 inch, and it must signal to turn the pump off before the water level reaches 12 inches.

A Leaking Gas Burner

The hybrid automaton models a leaking gas burner. It is assumed that:

- Any leakage can be detected and stopped within 1 second and
- The gas burner will not leak for 30 seconds after a leakage has been stopped.

We wish to prove that the accumulated time of leakage is at most one twentieth of the time in any interval of at least 60 seconds.



- In location l_1, the gas burner leaks. Location l_2 is the non-leaking location
- The integrator z records the cumulative leakage time – the accumulated amount of time that the system has spent in location l_1.
- The clock x records the time the system has spent in the current location
- The clock y records the total elapsed time
- We wish to prove that $y \geq 60 \Rightarrow 20z \leq y$ is an invariant of the system.

Formal Verification

Key Problems

- computable (decidable) only for simple dynamics
- computationally expensive
- representation of / computation with continuous sets

Formal Verification

Fighting complexity with overapproximations

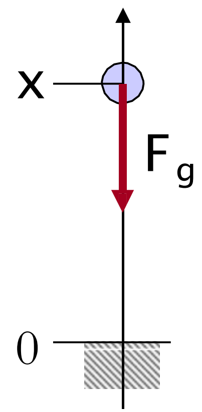
- simplify dynamics
- set representations
- set computations

Overapproximations should be

- conservative
- easy to derive and compute with
- accurate (not too many false positives)

Modeling Hybrid Systems: Bouncing Ball

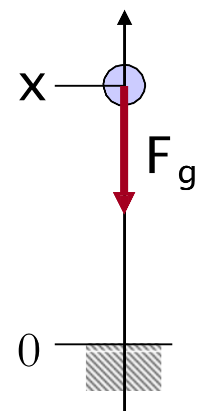
- ball with mass m and position x in free fall
- bounces when it hits the ground at $x = 0$
- initially at position x_0 and at rest



Part I – Free Fall

Condition for Free Fall

– ball above ground: $x \geq 0$



First Principles (physical laws)

- gravitational force :

$$F_g = -mg$$

$$g = 9.81\, m/s^2$$

- Newton's law of motion :

$$m\ddot{x} = F_g$$

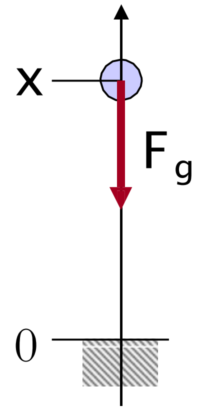
Part I – Free Fall

Obtaining 1st Order ODE System

- ordinary differential equation $\dot{x} = f(x)$
- transform to 1st order by introducing variables for higher derivatives
- here: $v = \dot{x}$

$$\begin{aligned} \dot{x} &= v \\ \dot{v} &= -g \end{aligned}$$

$$\begin{aligned} F_g &= -mg \\ m\ddot{x} &= F_g \end{aligned}$$



Part II – Bouncing

Conditions for "Bouncing"

- ball at ground position: $x = 0$
- downward motion: $v < 0$

Action for "Bouncing"

- velocity changes direction
- loss of velocity (deformation, friction)
- $v := -cv$, $0 \leq c \leq 1$

Combining Part I and II

Free Fall

- while $x \geq 0$,

$$\begin{aligned} \dot{x} &= v \\ \dot{v} &= -g \end{aligned}$$

Bouncing

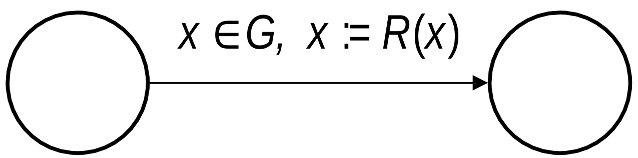
- if $x = 0$ and $v < 0$

$$v := -cv$$

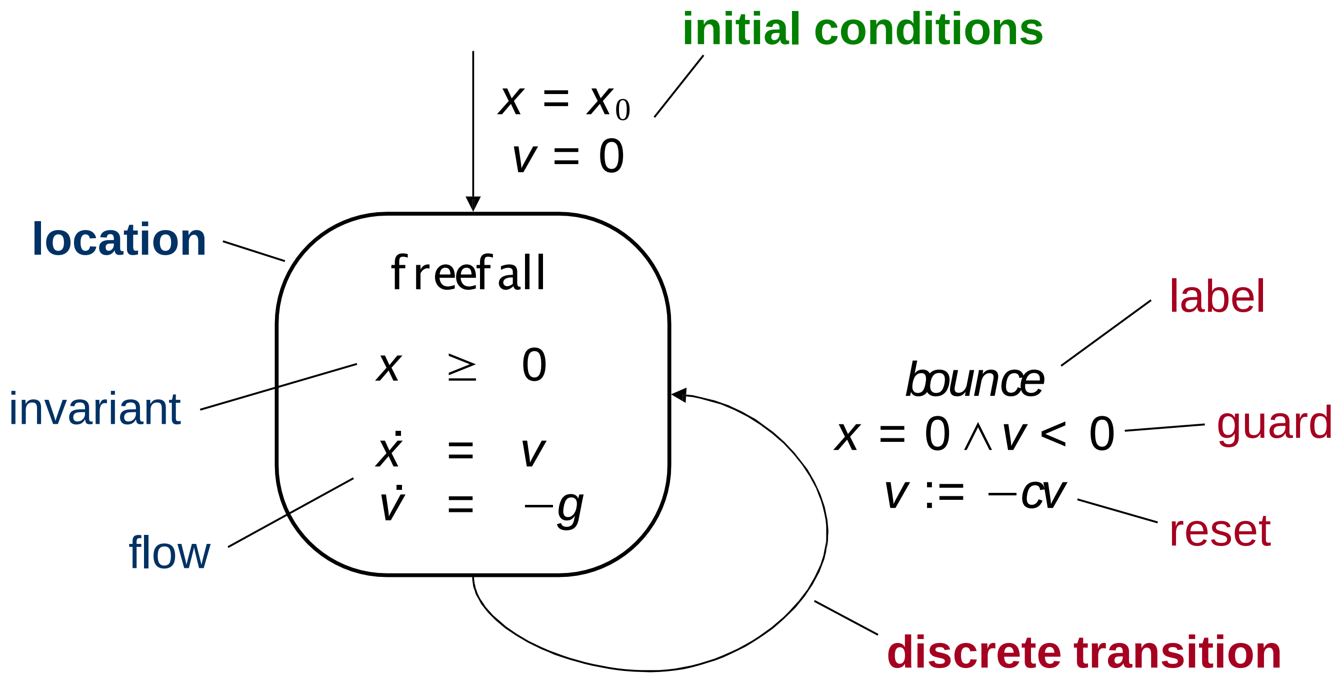
continuous dynamics

$$\dot{x} = f(x)$$

discrete dynamics



Hybrid Automaton Model



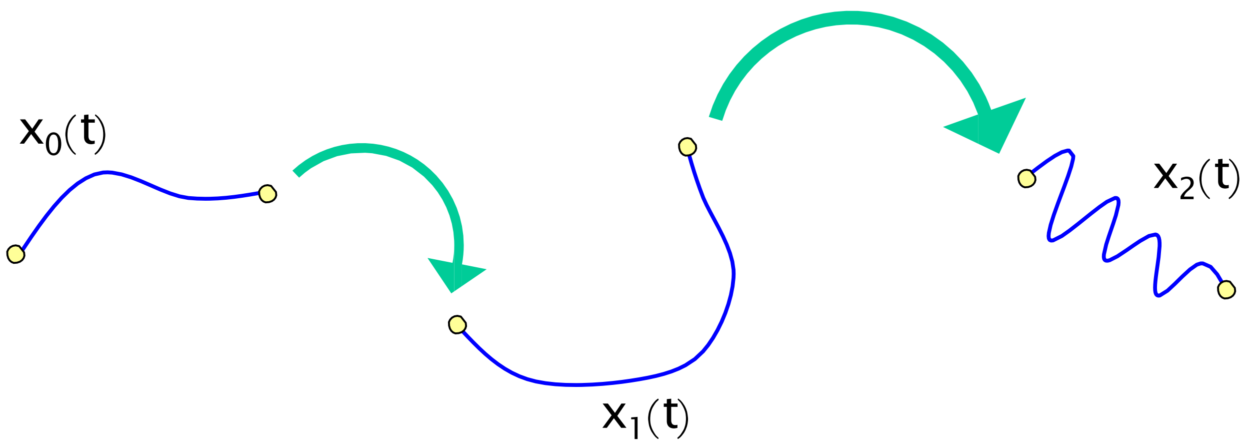
Hybrid Automata - Semantics

Run

– sequence of discrete transitions and time elapse

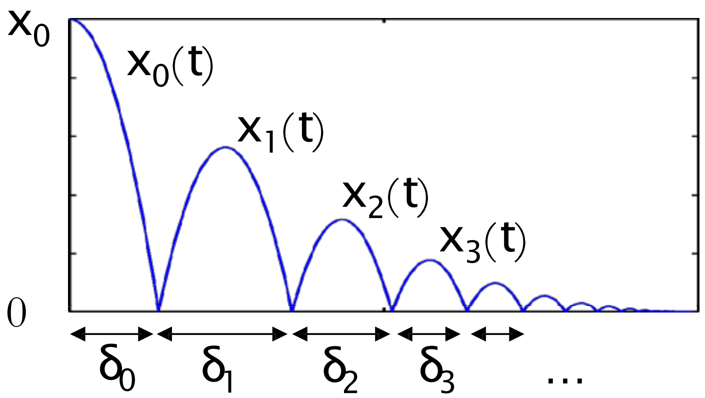
Execution

– run that starts in the initial states



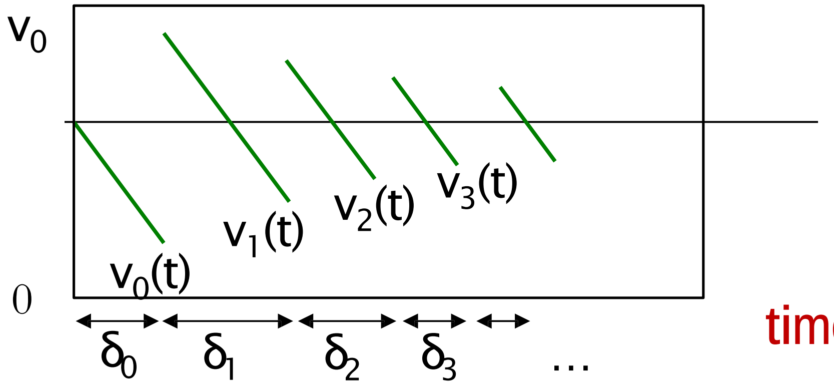
Execution of Bouncing Ball

position x



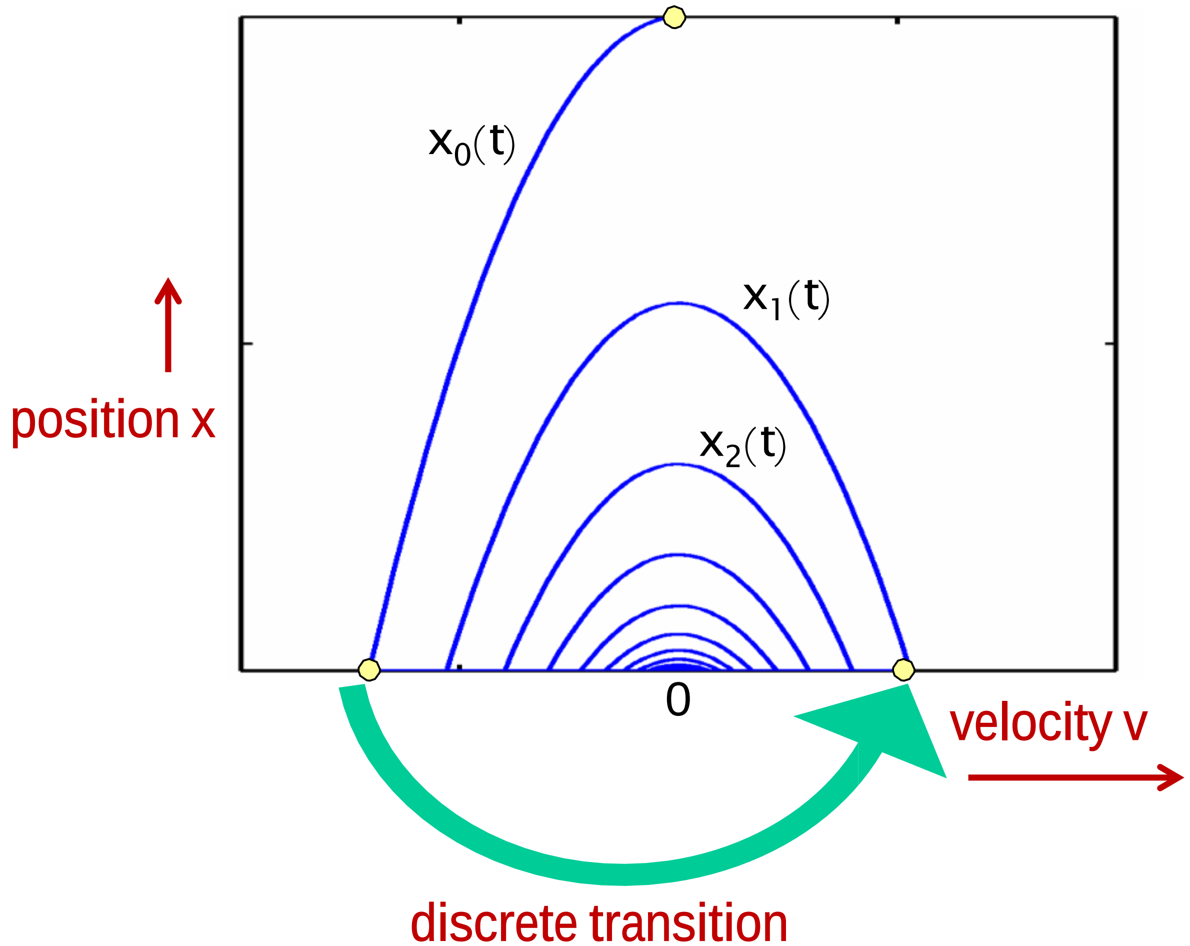
time t

velocity v



time t

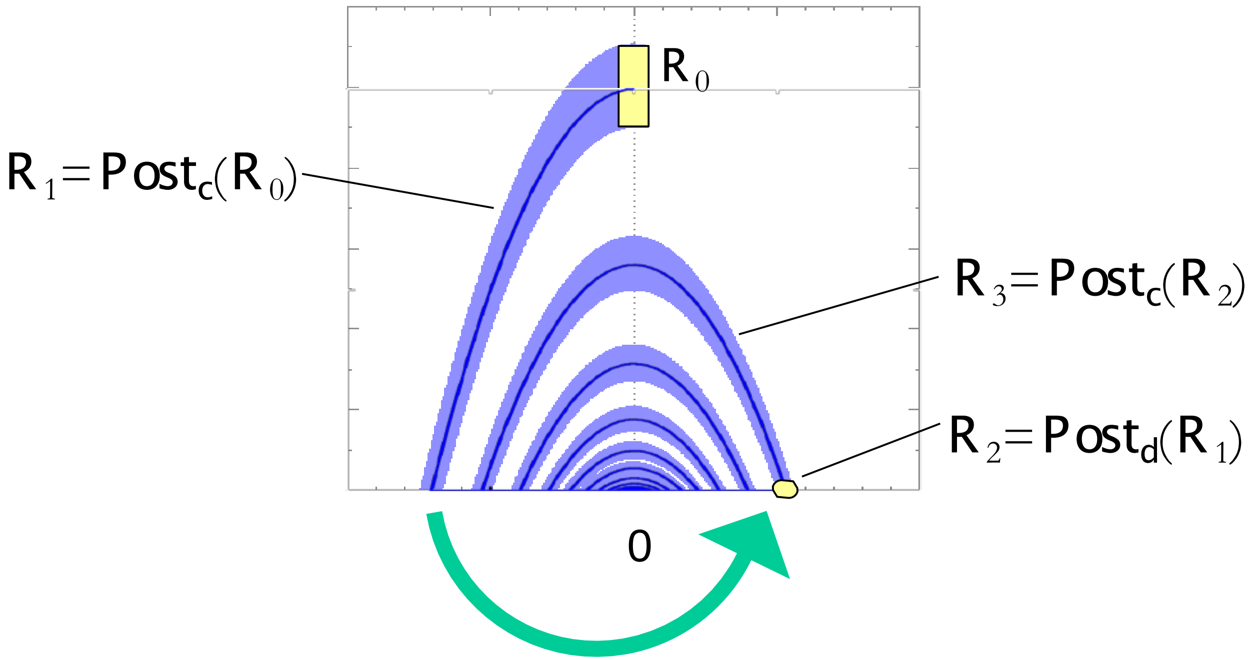
State-Space View (infinite time range)



Computing Reachable States

Successor functions

- discrete transitions : $Post_d(R)$
- time elapse : $Post_c(R)$



Computing Reachable States

Fixpoint computation

- Initialization: $R_0 = Ini$
- Recurrence: $R_{k+1} = R_k \cup Post_d(R_k) \cup Post_c(R_k)$
- Termination: $R_{k+1} = R_k \Rightarrow Reach = R_k$.

Problems

– in general termination not guaranteed

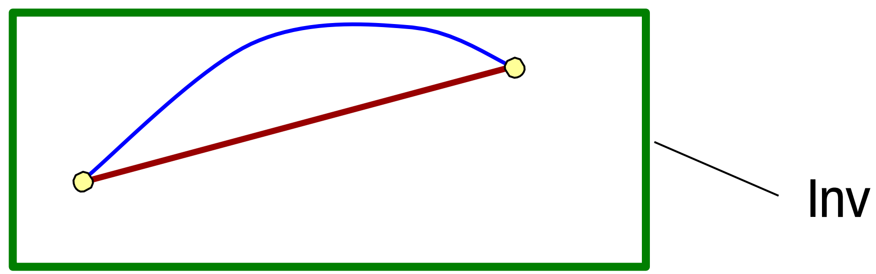
– time-elapse very hard to compute with sets

Reachability with Linear Hybrid Automata

Compute time elapse states $\mathrm{Post}_C(S)$

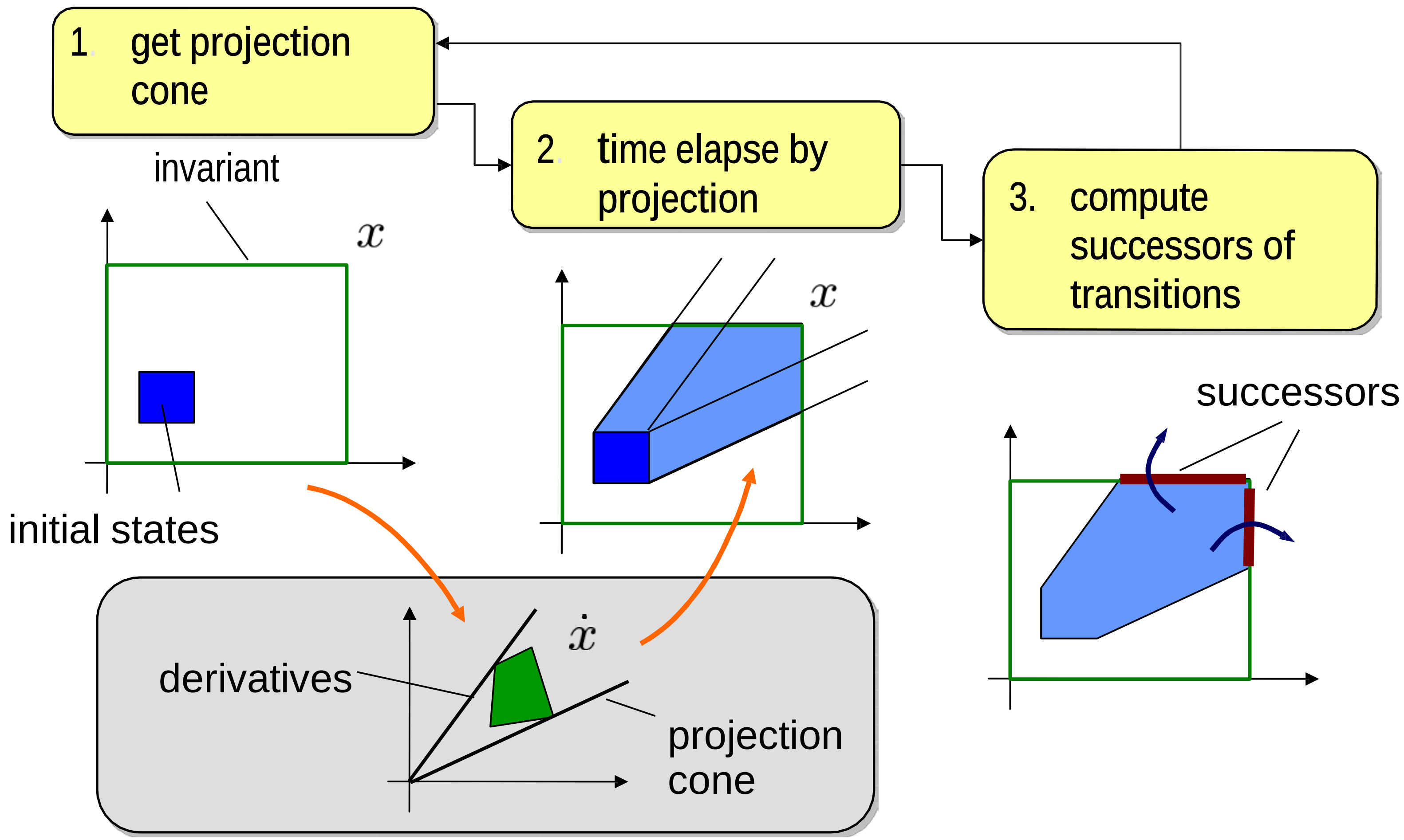
Theorem [Alur et al.]

- Time elapse along arbitrary trajectory iff time elapse along straight line (convex invariant).



- Time elapse along straight line can be computed as projection along cone [Halbwachs et al.]

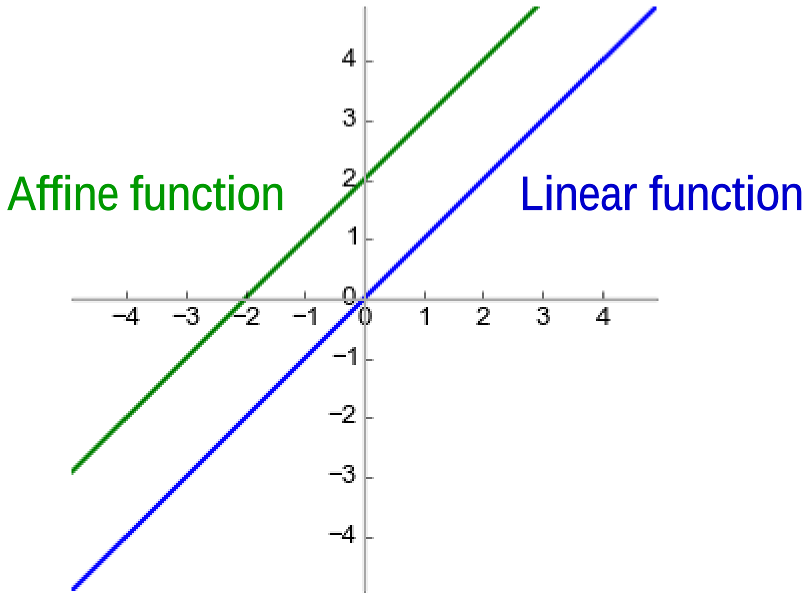
Reachability with Linear Hybrid Automata [Halbwachs, Henzinger, 93-97]



Piecewise Affine Hybrid Systems

- Another class of (not quite so) simple dynamics

- Exact computation of time elapse only at discrete points in time
  - used to overapproximate continuous time



Piecewise Affine Hybrid Systems

Linear Dynamics

Let's begin with "autonomous" part of the dynamics:

$$\dot{x} = Ax, \quad x \in R^n$$

Known solutions:

- analytic solution in continuous time
- explicit solution at discrete points in time (up to arbitrary accuracy)

Approach for Reachability:

– Compute reachable states over finite time: $Reach_{[0,T]}(X_{ini})$

– Use time-discretization, but with care!

Affine dynamics

– Flow:

$$\dot{x} = Ax + b \qquad \text{(deterministic)}$$

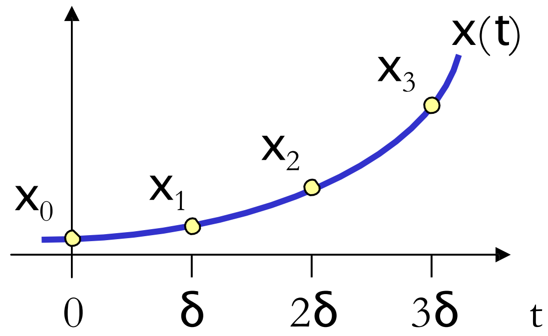
$$\dot{x} \in Ax + B, \qquad \text{with } B \text{ a set (nondeterministic)}$$

– For time elapse it's enough to look at a single location.

Time-Discretization for an Initial Point

- Analytic solution: $\boldsymbol{x(t) = e^{At}x_{Ini}}$
  - With $\boldsymbol{t = \delta k}$:

$$\boldsymbol{x\big(\delta(k+1)\big) = e^{A\delta}x(\delta k)}$$

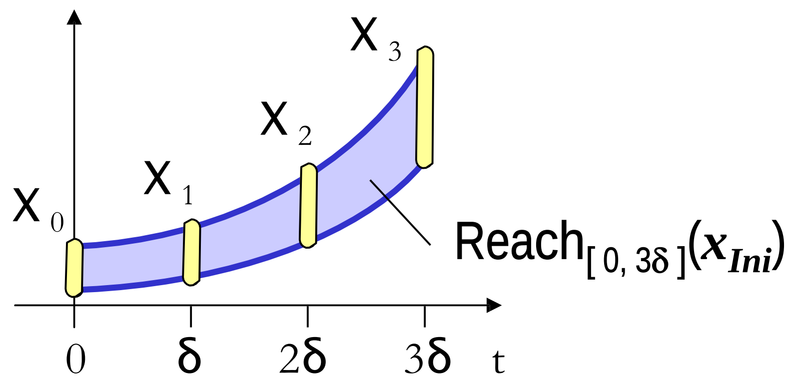


- Explicit solution in discretized time:
  - $\boldsymbol{x_0 = x_{Ini}}$
  - $\boldsymbol{x_{k+1} = e^{A\delta}x_k}$

Multiplication with constant matrix $\boldsymbol{x^{A\delta}}$ = linear transform

Time-Discretization for an Initial Set

- Explicit solution in discretized time:
  - $x_0 = x_{Ini}$
  - $x_{k+1} = e^{A\delta} x_k$



- Acceptable solution for purely continuous systems
  - $x(t)$ is in $\varepsilon(\delta)$-neighborhood of some X_k

- Unacceptable for hybrid systems
  - discrete transitions might "*fire*" between sampling times
  - if transitions are "*missed*", then $x(t)$ is not in $\varepsilon(\delta)$-neighborhood

Reachability by Time-Discretization

Goal:

- Compute sequence Ω_k over bounded time $[\,0, N\delta\,]$ such that:

$$\text{Reach}_{[\,0,\,N\delta\,]}(\,x_{Ini}\,) \subseteq \Omega_0 \cup \Omega_1 \cup \ldots \cup \Omega_N$$

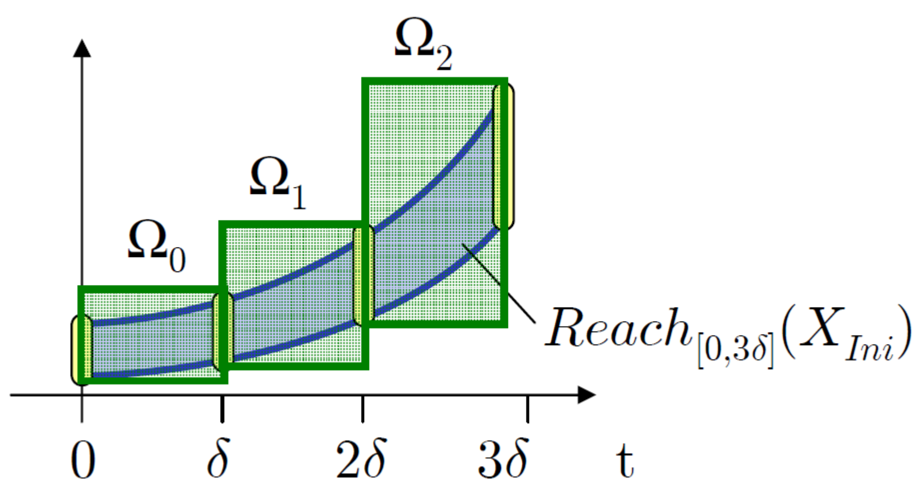
Approach:

- Refine Ω_k by recurrence

$$\Omega_{k+1} = e^{A\delta}\Omega_k$$

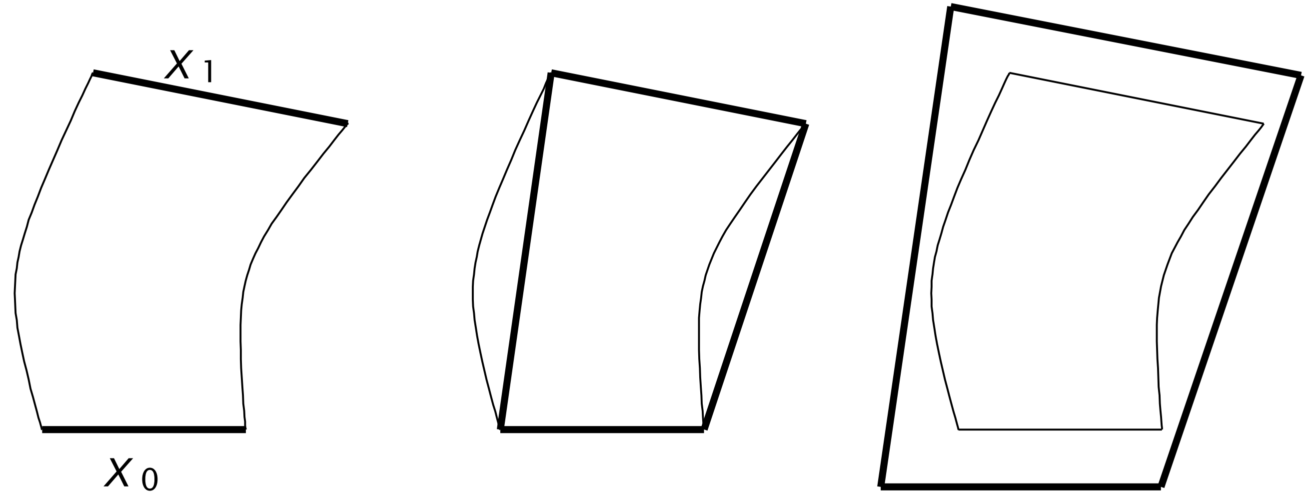
- Condition for Ω_0 :

$$\text{Reach}_{[\,0,\,\delta\,]}(\,x_{Ini}\,) \subseteq \Omega_0$$



Time-Discretization with Convex Hull

Over-approximating Reach$_{[0,\delta]}$:



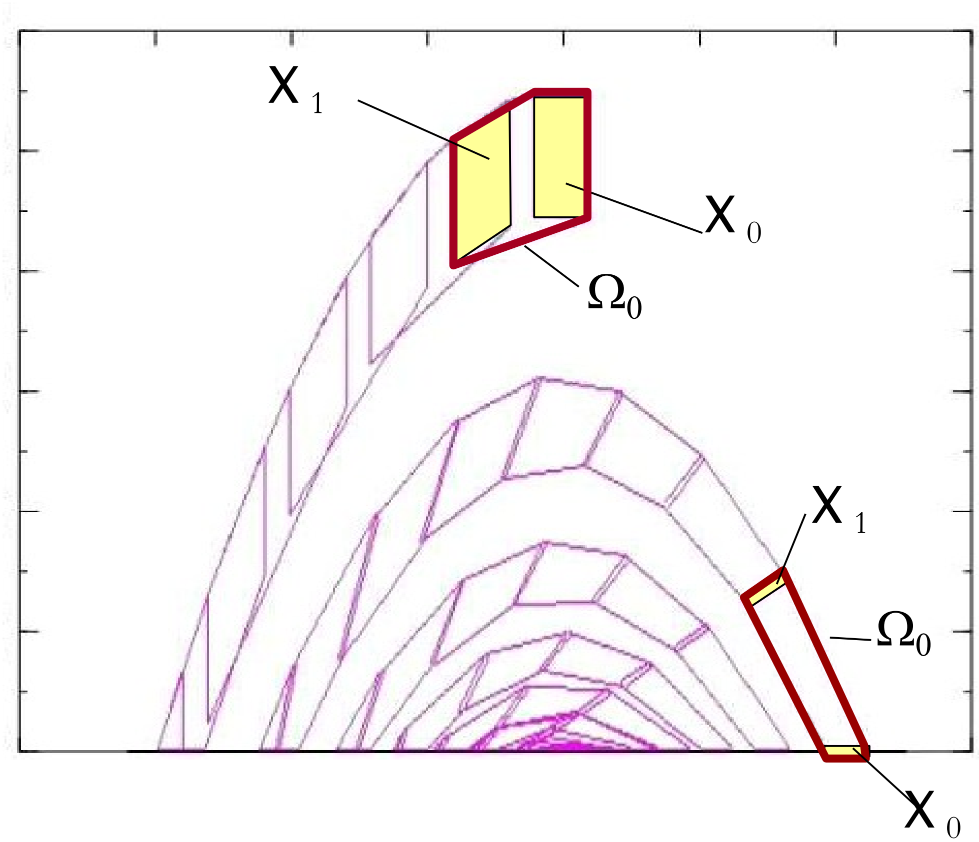
Reach$_{[0,\delta]}(\boldsymbol{x_{Ini}})$

Conv$(\boldsymbol{x_0}, \boldsymbol{x_1})$
(convex hull)

Bloat(Conv$(\boldsymbol{x_0}, \boldsymbol{x_1})$)

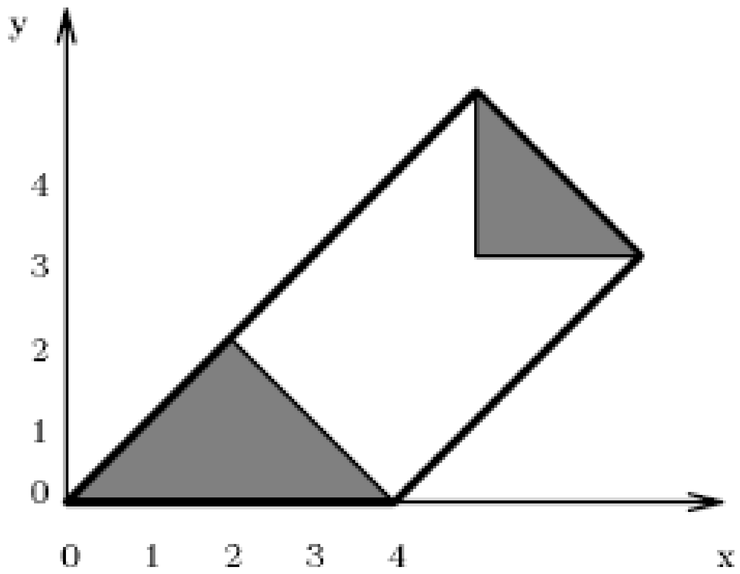
Time-Discretization with Convex Hull

Bouncing Ball:



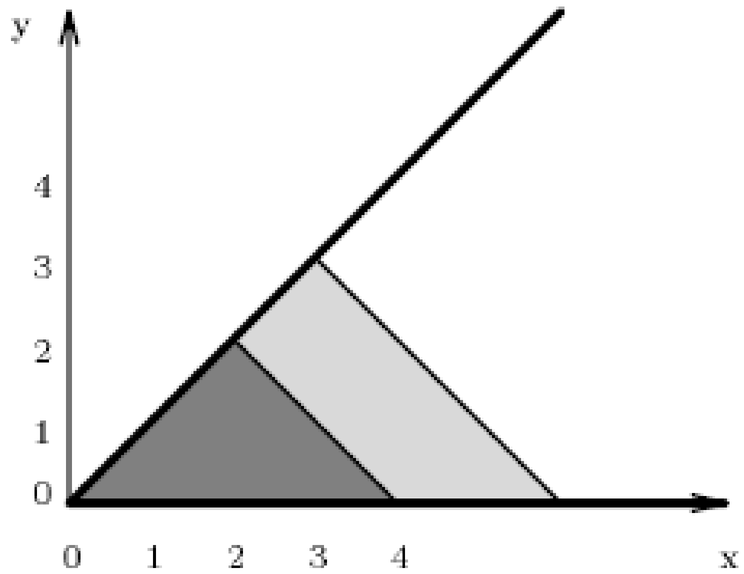
Approximate Analysis

Approximation Operators:



$$\{0 \le y \le x \le 4 - y\}$$
$$\sqcup\{x \ge 5 \wedge y \ge 3 \wedge x + y \le 10\}$$
$$= \{0 \le y \le x \le y + 4 \wedge x + y \le 10\}$$

(a). Convex hull



$$\{0 \le y \le x \le 4 - y\}$$
$$\nabla\{0 \le y \le x \le 6 - y\}$$
$$= \{0 \le y \le x\}$$

(b). Widening

Analysis of Leaking Gas Burner

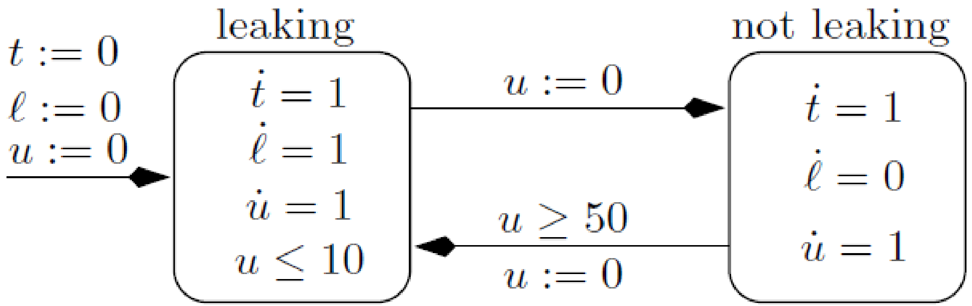


Fig. 1. Hybrid automaton of the gas burner

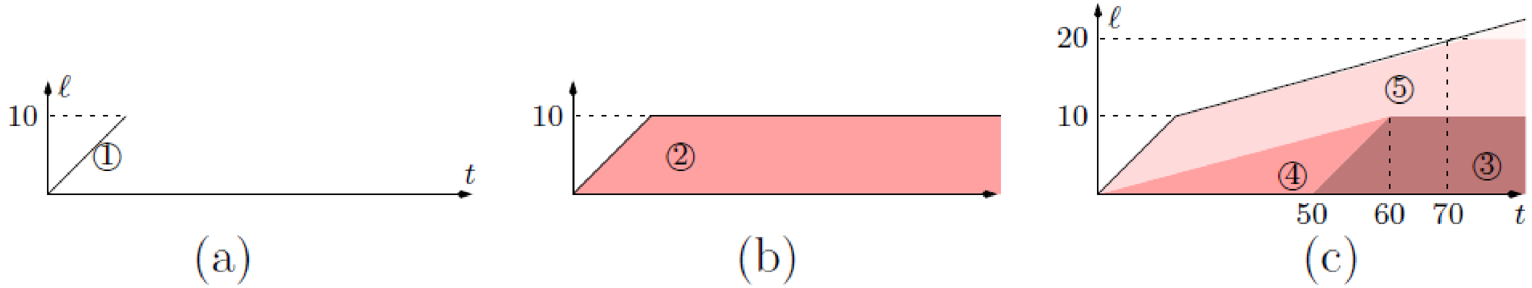


Fig. 2. Analysis of the hybrid gas burner

Step-1: Leaking location reached with $\{\, t = l = 0 \,\}$, and as time elapses we get the polyhedron $\{\, 0 \leq t = l \leq 10 \,\}$ (Region (1) in Fig. 2.a)

Step-2: Non-leaking location is reached with $\{\, 0 \leq t = l \leq 10 \,\}$. As time elapses, we get $\{\, 0 \leq l \leq 10, t \geq l \,\}$. (Region (2) in Fig. 2.b)

Step-3: We go back to leaking location with $\{\, 0 \leq l \leq 10, t \geq l + 50 \,\}$. (Region (3) in Fig. 2.c).
Convex hull with $\{\, t = l = 0 \,\}$ gives $\{\, 0 \leq l \leq 10, t \geq 6l \,\}$. (Region (4) in Fig. 2.c)

Source: [Gonnord, Halbwachs, LNCS 4134] Combining widening and acceleration in linear relation analysis.

Step-3 (contd): Time passage yields $\{\, 0 \leq l \leq 20, t \geq l, t \geq 6l - 50 \,\}$.
Now standard widening yields $\{\, 0 \leq l \leq t, t \geq 6l - 50 \,\}$.
(Region (5) in Fig. 2.c)