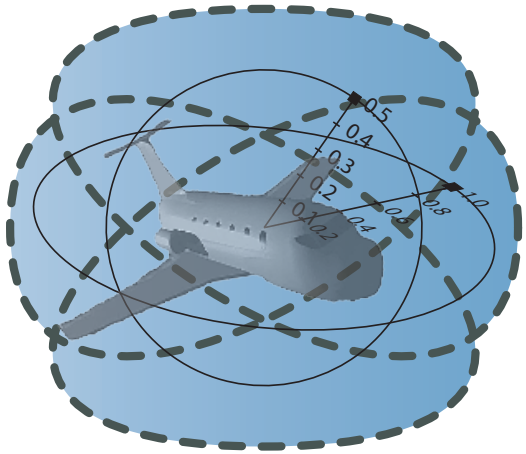
15-819/18-879: Logical Analysis of Hybrid Systems

03: Hybrid Systems

André Platzer

aplatzer@cs.cmu.edu
Carnegie Mellon University, Pittsburgh, PA



Outline

Outline

Labeled Transition Systems

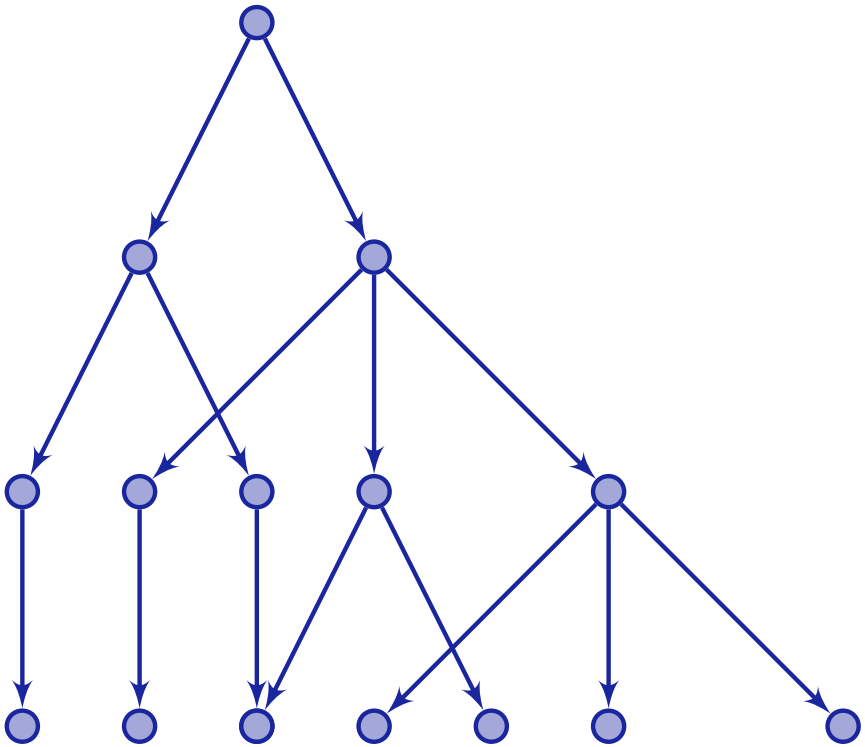
Definition (Labeled Transition System)

- Transition relation on $Q \times A \times Q$, denoted as $q \xrightarrow{a} q^+$, along with
- (possibly infinite) set A of transition actions,
- (possibly infinite) set Q of states.

Labeled Transition Systems

Definition (Labeled Transition System)

- Transition relation on $Q \times A \times Q$, denoted as $q \xrightarrow{a} q^+$, along with
- (possibly infinite) set A of transition actions,
- (possibly infinite) set Q of states.

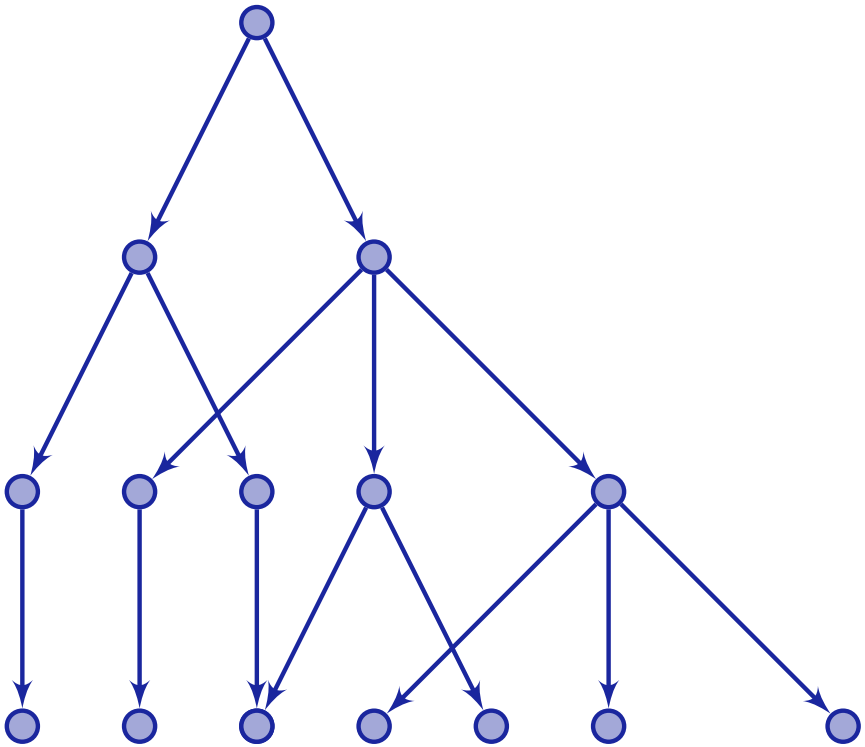


Labeled Transition Systems

Definition (Trace)

(In)Finite series $q_0, q_1, q_2, \ldots \in Q$ such that for some $a_i \in A$

$$q_0 \xrightarrow{a_0} q_1 \xrightarrow{a_1} q_2 \xrightarrow{a_2} q_3 \xrightarrow{a_3} q_4 \xrightarrow{a_4} \ldots$$

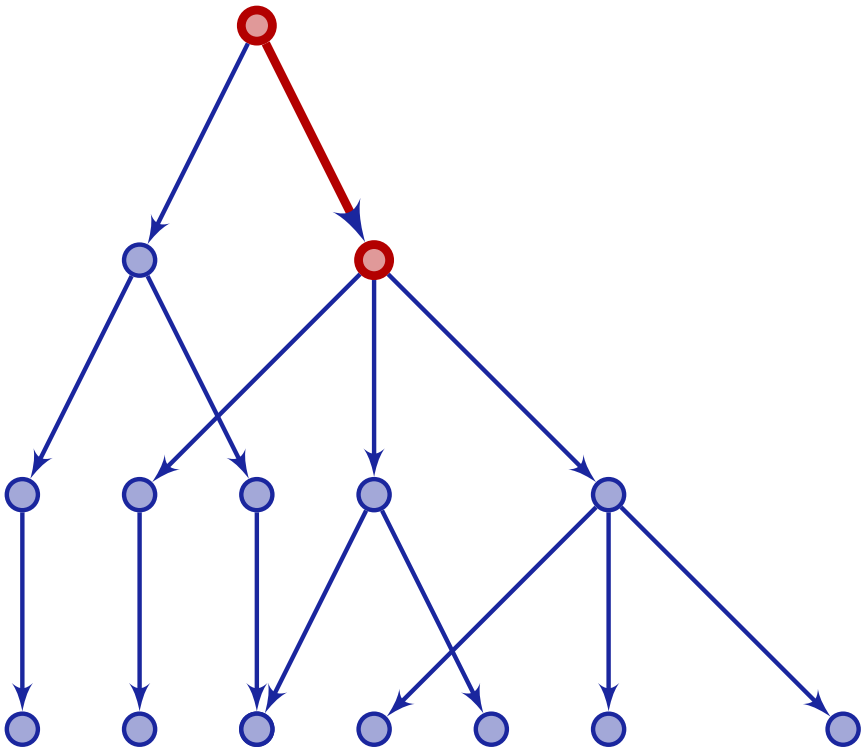


Labeled Transition Systems

Definition (Trace)

(In)Finite series $q_0, q_1, q_2, \ldots \in Q$ such that for some $a_i \in A$

$$q_0 \xrightarrow{a_0} q_1 \xrightarrow{a_1} q_2 \xrightarrow{a_2} q_3 \xrightarrow{a_3} q_4 \xrightarrow{a_4} \ldots$$

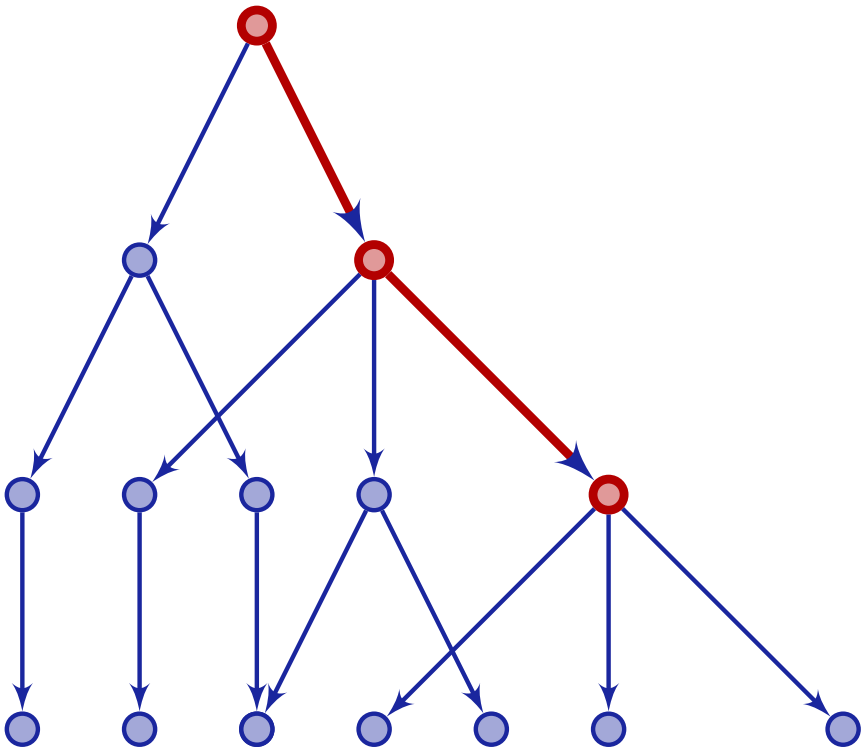


Labeled Transition Systems

Definition (Trace)

(In)Finite series $q_0, q_1, q_2, \ldots \in Q$ such that for some $a_i \in A$

$$q_0 \xrightarrow{a_0} q_1 \xrightarrow{a_1} q_2 \xrightarrow{a_2} q_3 \xrightarrow{a_3} q_4 \xrightarrow{a_4} \ldots$$

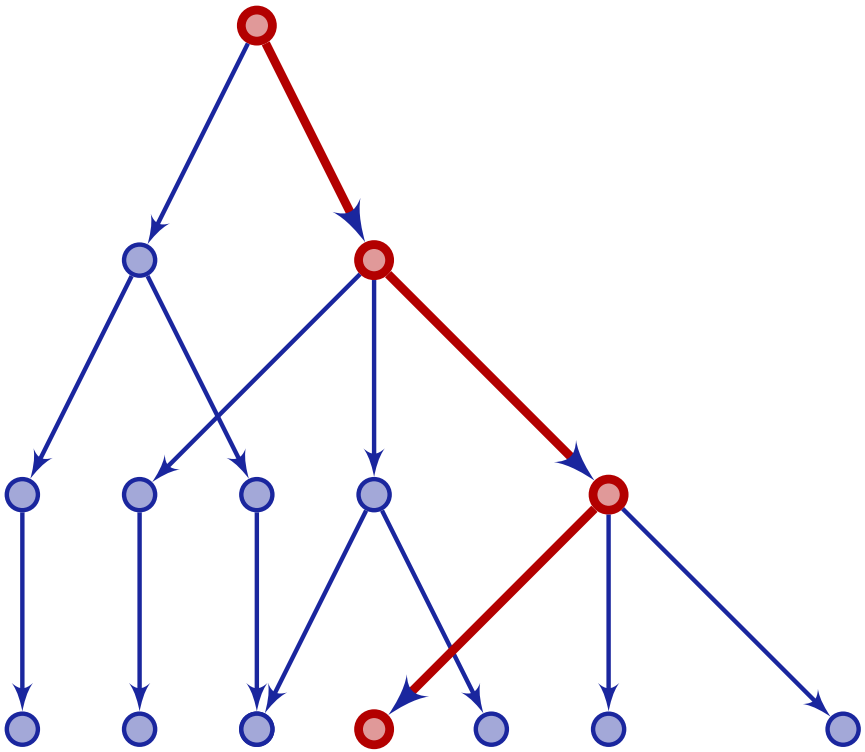


Labeled Transition Systems

Definition (Trace)

(In)Finite series $q_0, q_1, q_2, \ldots \in Q$ such that for some $a_i \in A$

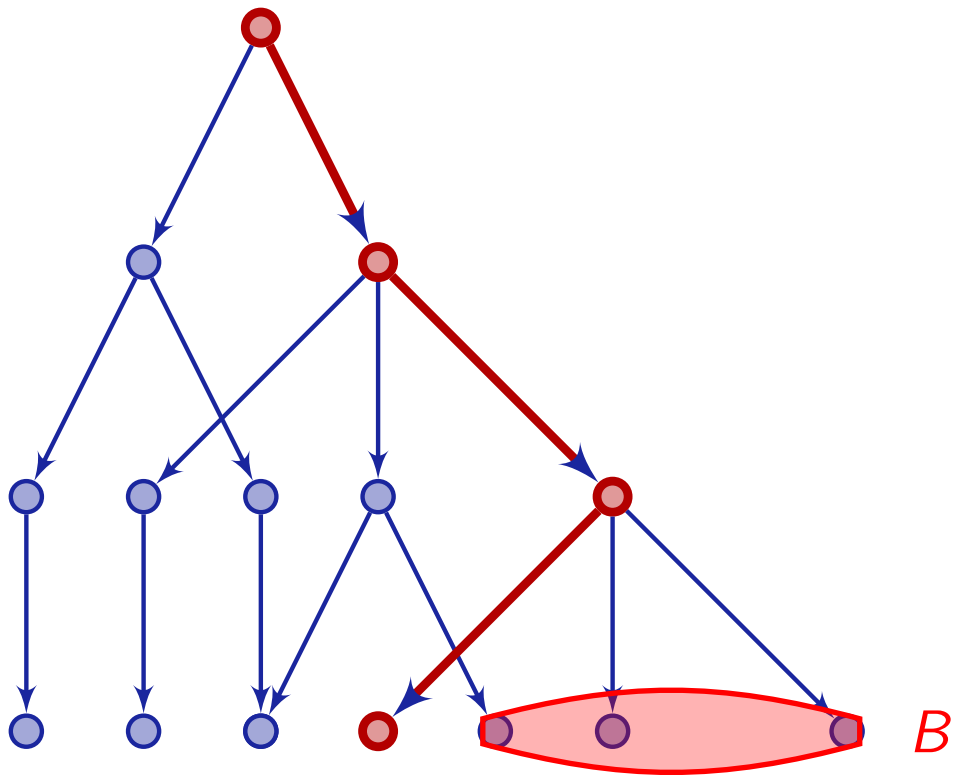
$$q_0 \xrightarrow{a_0} q_1 \xrightarrow{a_1} q_2 \xrightarrow{a_2} q_3 \xrightarrow{a_3} q_4 \xrightarrow{a_4} \ldots$$



Labeled Transition Systems

Definition (Model Checking Problem for Reachability)

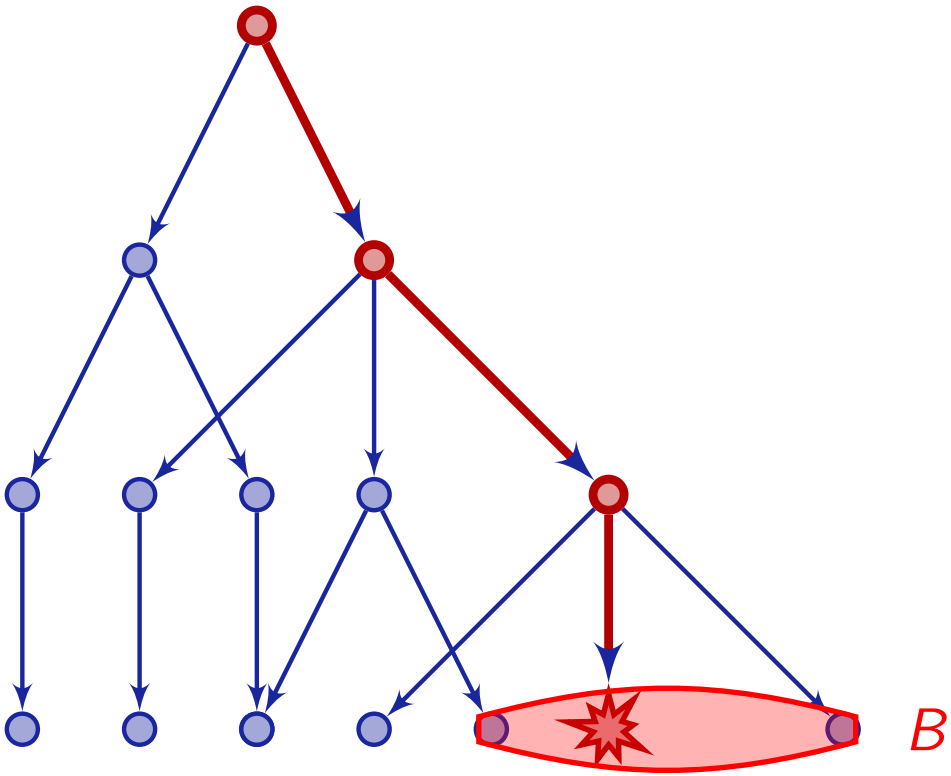
Given initial states $Q_0 \subseteq Q$ and bad states $B \subseteq Q$ for a transition system, check whether there is a trace from some $q_0 \in Q_0$ to some $q_b \in B$.



Labeled Transition Systems

Definition (Model Checking Problem for Reachability)

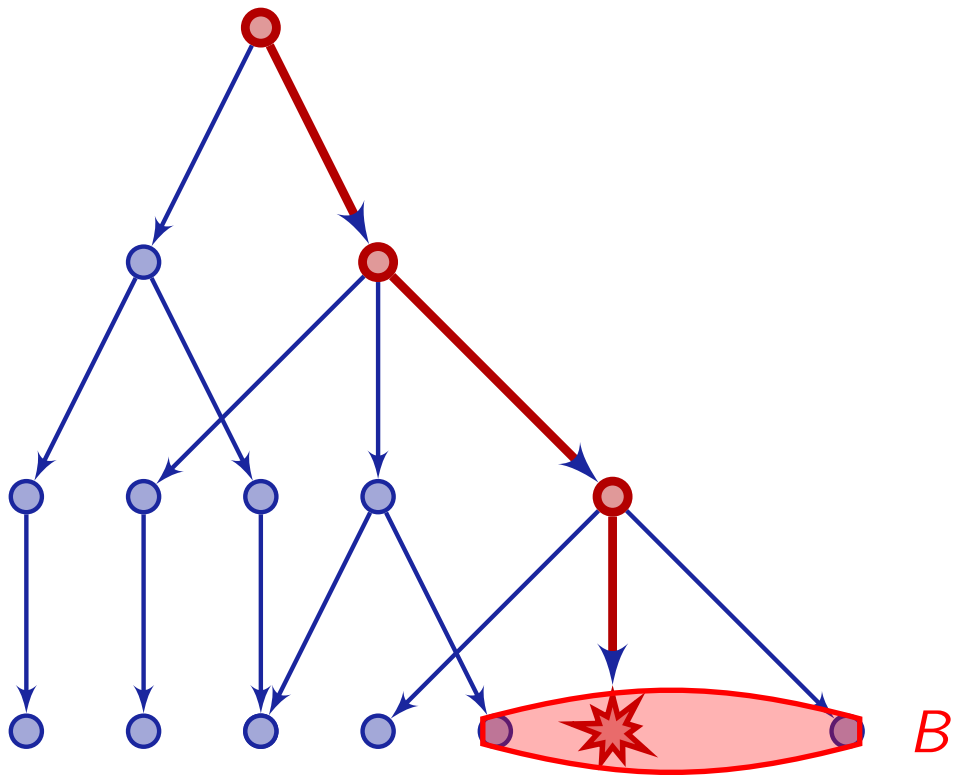
Given initial states $Q_0 \subseteq Q$ and bad states $B \subseteq Q$ for a transition system, check whether there is a trace from some $q_0 \in Q_0$ to some $q_b \in B$.



Labeled Transition Systems

Definition (Image Computation)

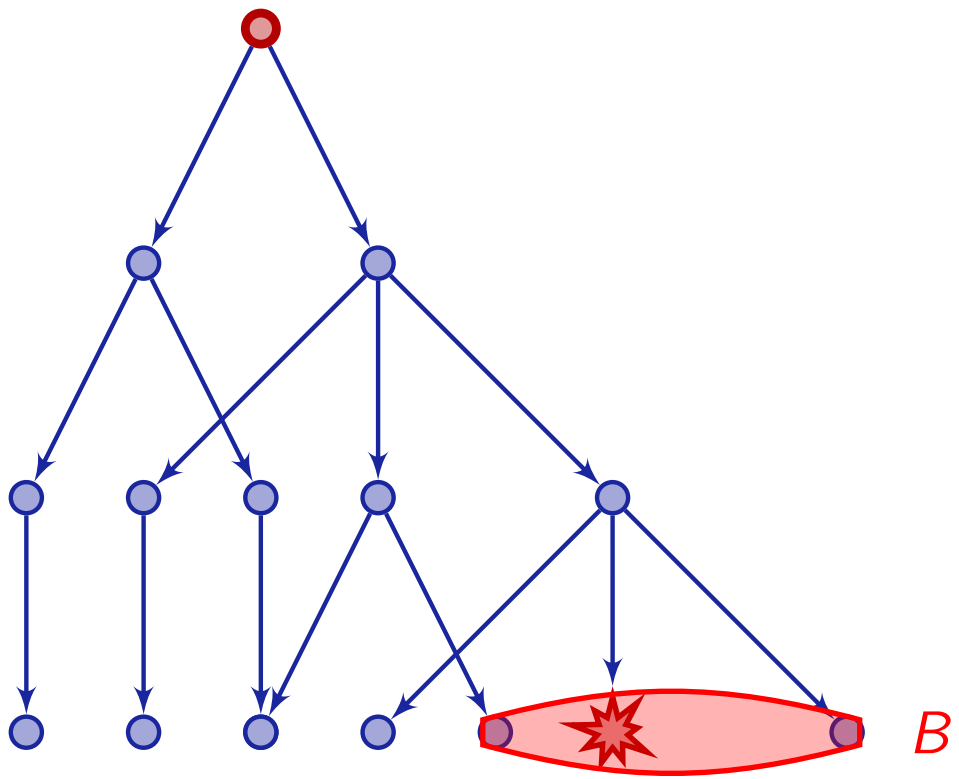
$$Post_A(Y) := \{q^+ \in Q : q \xrightarrow{a} q^+ \text{ for some } q \in Y, a \in A\}$$



Labeled Transition Systems

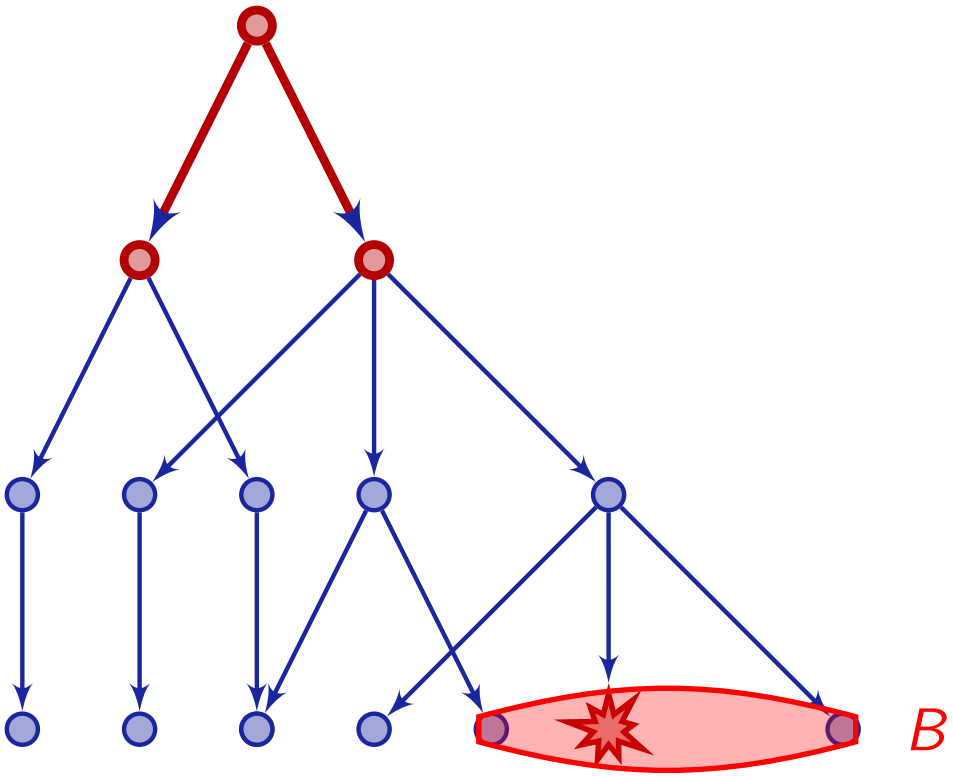
Definition (Image Computation)

$$Post_A(Y) := \{q^+ \in Q : q \xrightarrow{a} q^+ \text{ for some } q \in Y, a \in A\}$$



Definition (Image Computation)

$$Post_A(Y) := \{q^+ \in Q : q \xrightarrow{a} q^+ \text{ for some } q \in Y, a \in A\}$$



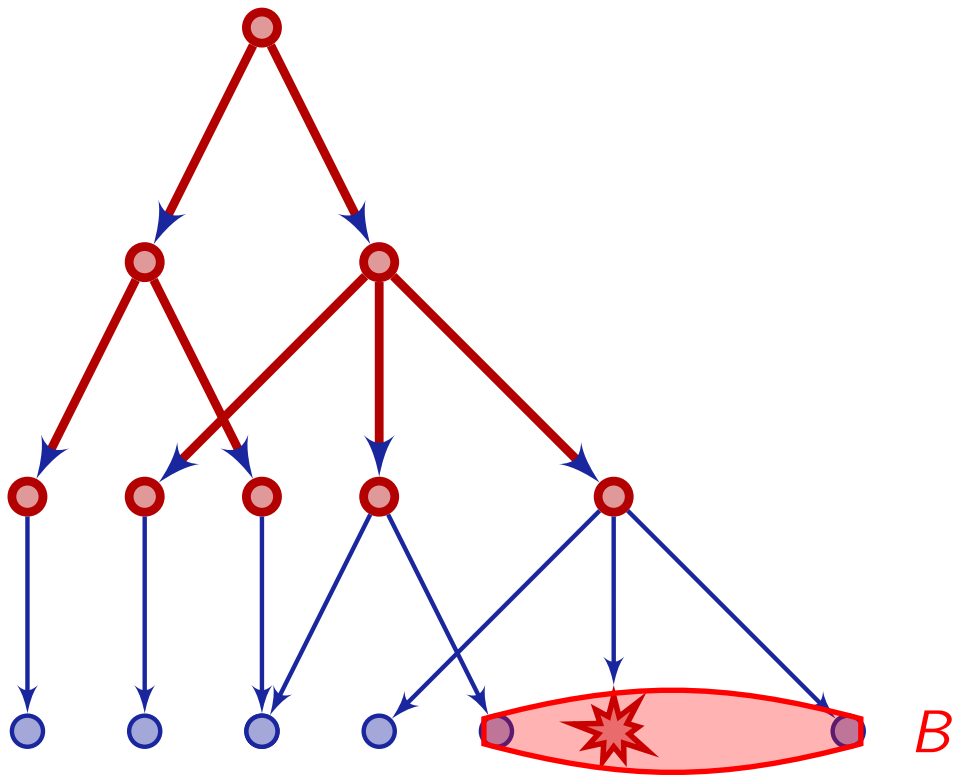
Q_0

$Post_A(Q_0)$

$Q_1 = Post_A(Q_0)$

Definition (Image Computation)

$$Post_A(Y) := \{q^+ \in Q : q \xrightarrow{a} q^+ \text{ for some } q \in Y, a \in A\}$$



$$Q_0$$

$$Post_A(Q_0)$$

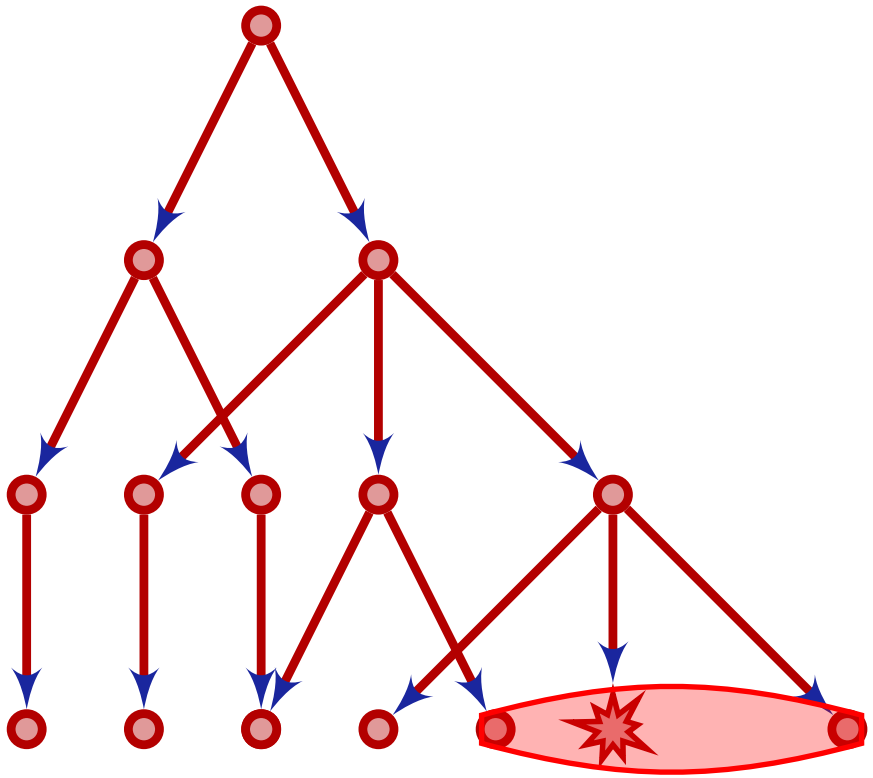
$$Q_1 = Post_A(Q_0)$$

$$Post_A(Q_1)$$

$$Q_2 = Post_A^2(Q_0)$$

Definition (Image Computation)

$$Post_A(Y) := \{q^+ \in Q : q \xrightarrow{a} q^+ \text{ for some } q \in Y, a \in A\}$$



$$Q_0$$

$$Post_A(Q_0)$$

$$Q_1 = Post_A(Q_0)$$

$$Post_A(Q_1)$$

$$Q_2 = Post_A^2(Q_0)$$

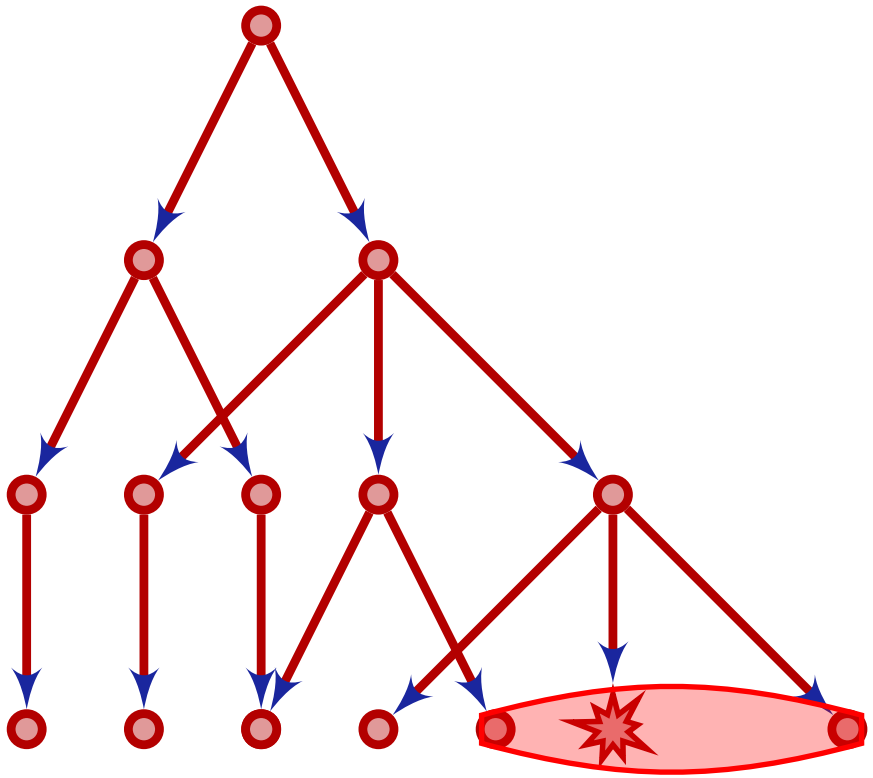
$$Post_A(Q_2)$$

$$Q_3 = Post_A^3(Q_0)$$

Definition (Image Computation)

$$Post_A(Y) := \{q^+ \in Q : q \xrightarrow{a} q^+ \text{ for some } q \in Y, a \in A\}$$

$$Post_A^*(Y) := \bigcup_{n \in \mathbb{N}} Post_A^n(Y) = \mu Z.(Y \cup Z \cup Post_A(Z))$$



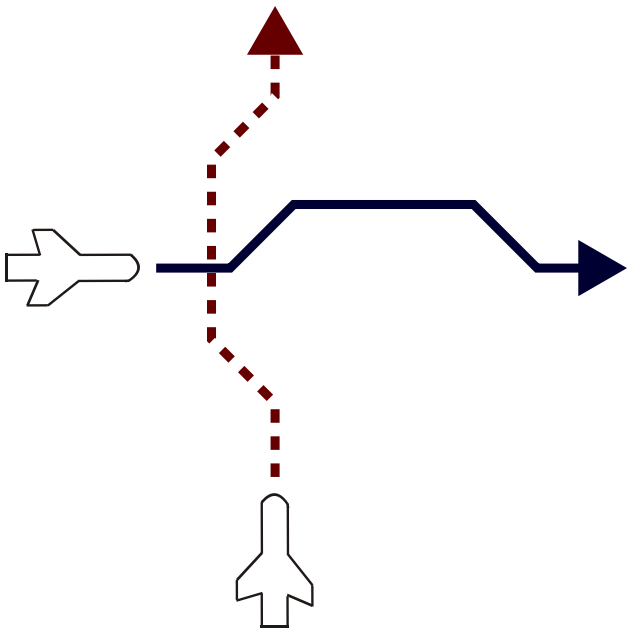
$$Q_0$$
$$Post_A(Q_0)$$
$$Q_1 = Post_A(Q_0)$$
$$Post_A(Q_1)$$
$$Q_2 = Post_A^2(Q_0)$$
$$Post_A(Q_2)$$
$$Q_3 = Post_A^3(Q_0)$$

Finite Automata

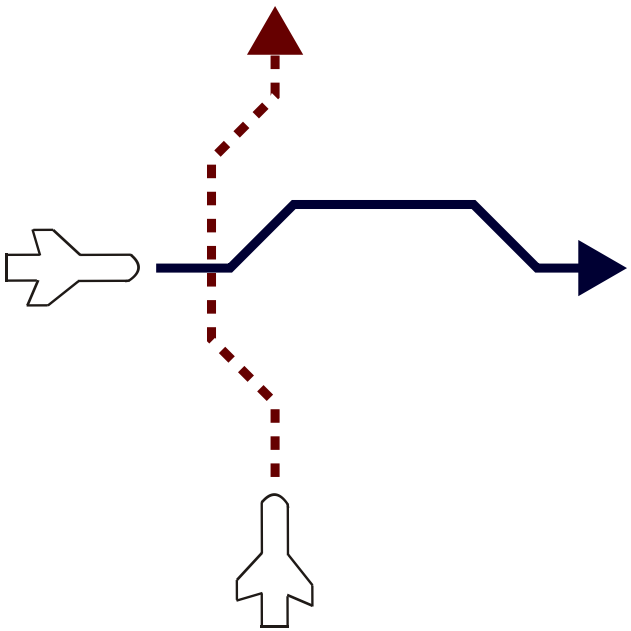
Definition (Nondeterministic Finite Automata $\rightsquigarrow$ LTS)

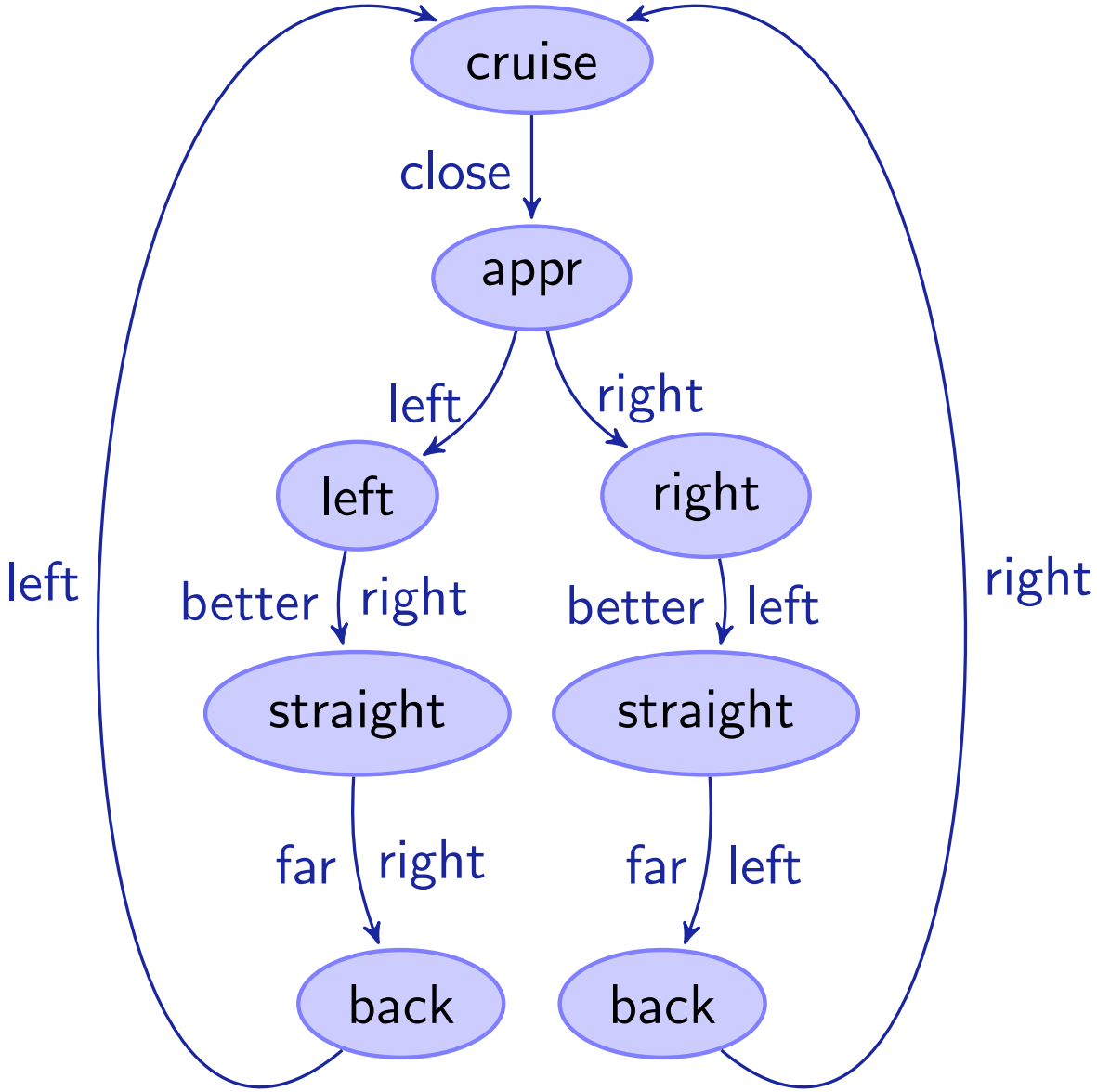
- Transition relation on $Q \times A \times Q$, denoted as $q \xrightarrow{a} q^+$, along with
- finite set A of transition actions,
- finite set Q of states, initial states $Q_0 \subseteq Q$, accepting states $F \subseteq Q$.

Finite Automaton for Collision Avoidance

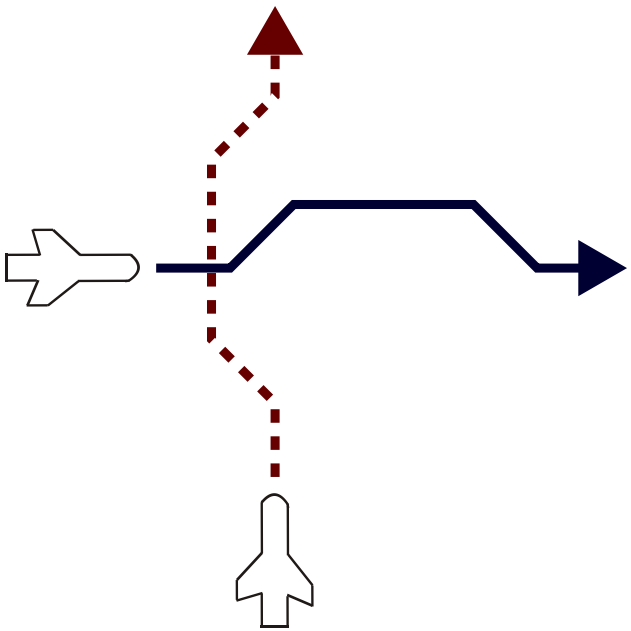


Finite Automaton for Collision Avoidance

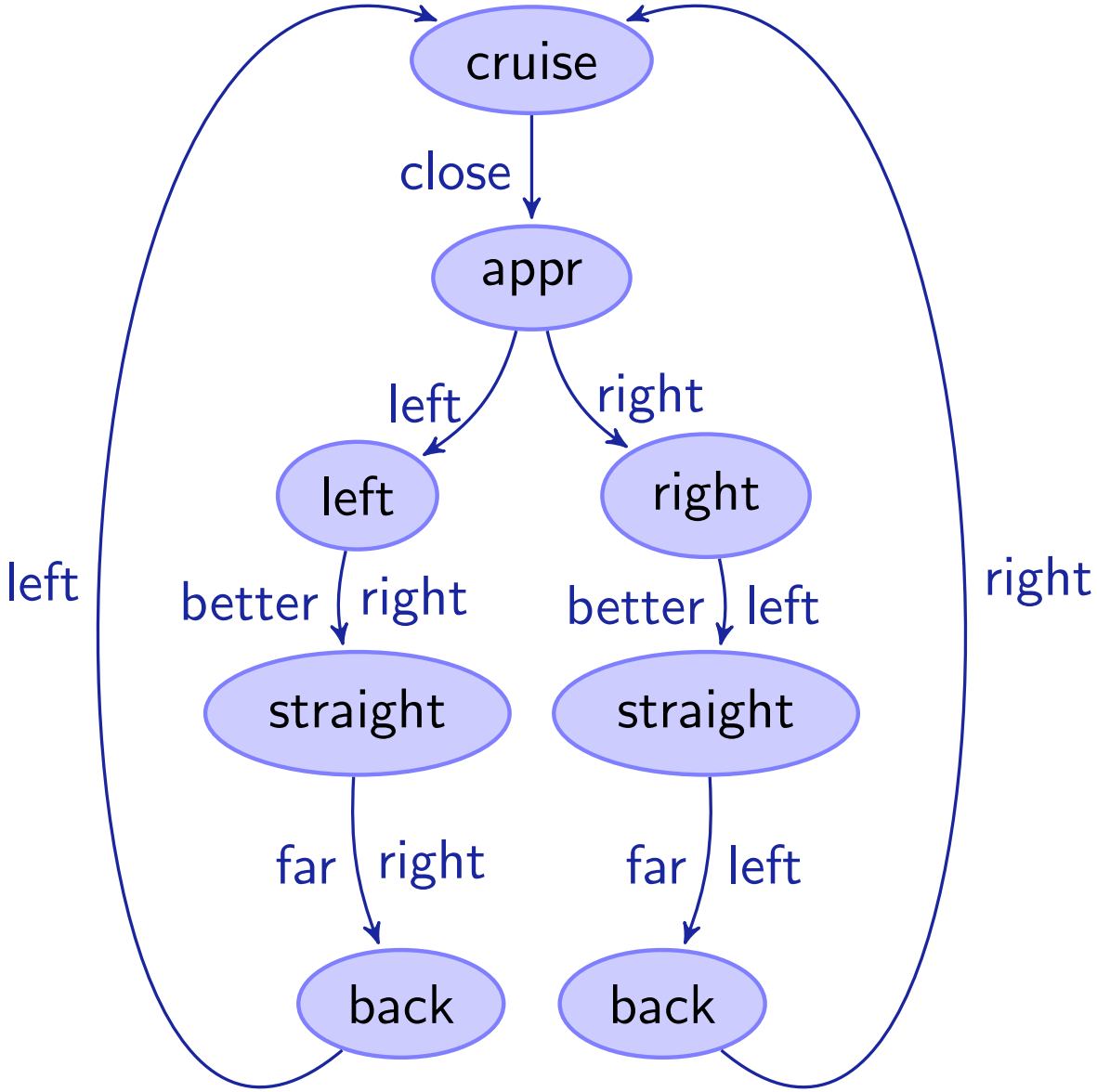




Finite Automaton for Collision Avoidance



good model?



Collision avoidance is a property of controlled movement!

Want Hybrid Transition Systems

Want Hybrid Transition Systems

- continuous state space $\mathbb{R}^n$

Want Hybrid Transition Systems

- continuous state space $\mathbb{R}^n$
- actions $A = D \dot{\cup} \mathbb{R}_{\geq 0}$ that are either discrete (D) or continuous ($\mathbb{R}_{\geq 0}$)

Want Hybrid Transition Systems

- continuous state space $\mathbb{R}^n$
- actions $A = D \dot{\cup} \mathbb{R}_{\geq 0}$ that are either discrete (D) or continuous ($\mathbb{R}_{\geq 0}$)
- Discrete transition somehow

$$x \xrightarrow{a} x^+ \quad \text{justified by discrete action } a \in D$$

Want Hybrid Transition Systems

- continuous state space $\mathbb{R}^n$
- actions $A = D \dot{\cup} \mathbb{R}_{\geq 0}$ that are either discrete (D) or continuous ($\mathbb{R}_{\geq 0}$)
- Discrete transition somehow

$$x \xrightarrow{a} x^+ \quad \text{justified by discrete action } a \in D$$

- Continuous transition somehow

$$x \xrightarrow{a} x^+ \quad \text{by continuous flow of duration } a \in \mathbb{R}_{\geq 0} \text{ from } x \text{ to } x^+$$

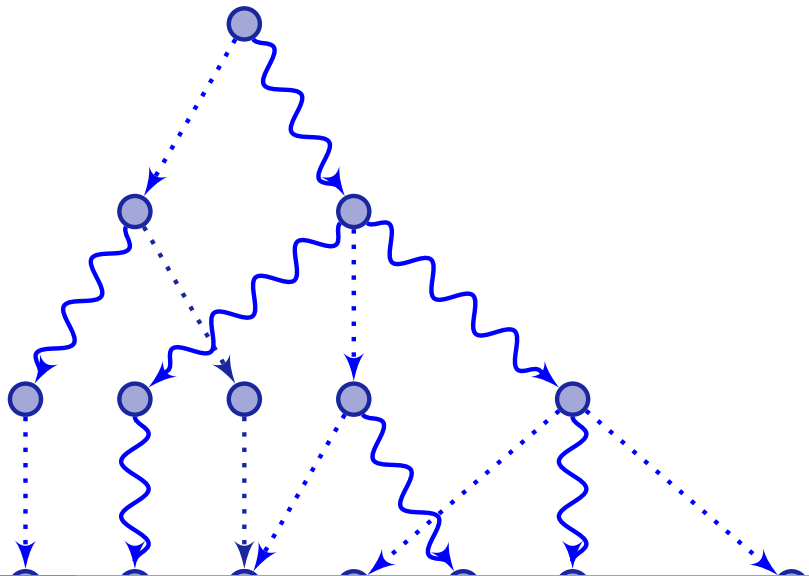
Want Hybrid Transition Systems

- continuous state space $\mathbb{R}^n$
- actions $A = D \dot{\cup} \mathbb{R}_{\geq 0}$ that are either discrete (D) or continuous ($\mathbb{R}_{\geq 0}$)
- Discrete transition somehow

$$x \xrightarrow{a} x^+ \quad \text{justified by discrete action } a \in D$$

- Continuous transition somehow

$$x \xrightarrow{a} x^+ \quad \text{by continuous flow of duration } a \in \mathbb{R}_{\geq 0} \text{ from } x \text{ to } x^+$$



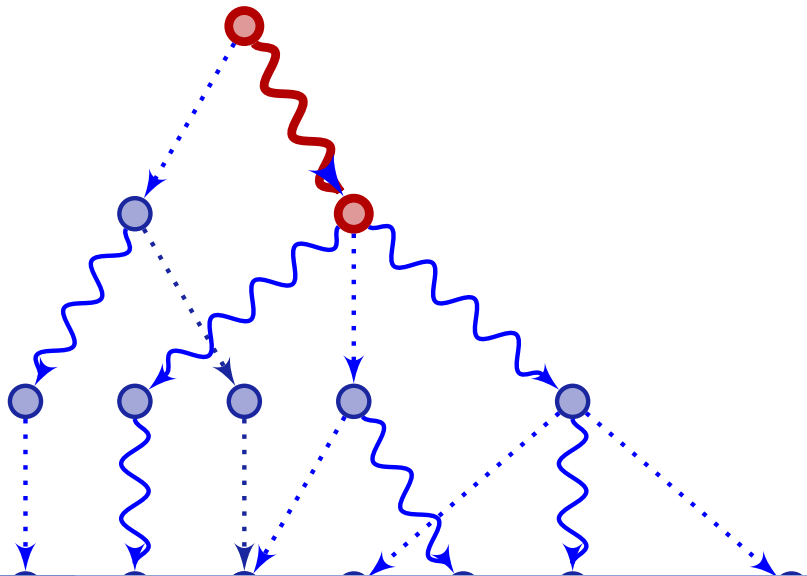
Want Hybrid Transition Systems

- continuous state space $\mathbb{R}^n$
- actions $A = D \dot{\cup} \mathbb{R}_{\geq 0}$ that are either discrete (D) or continuous ($\mathbb{R}_{\geq 0}$)
- Discrete transition somehow

$$x \xrightarrow{a} x^+ \quad \text{justified by discrete action } a \in D$$

- Continuous transition somehow

$$x \xrightarrow{a} x^+ \quad \text{by continuous flow of duration } a \in \mathbb{R}_{\geq 0} \text{ from } x \text{ to } x^+$$



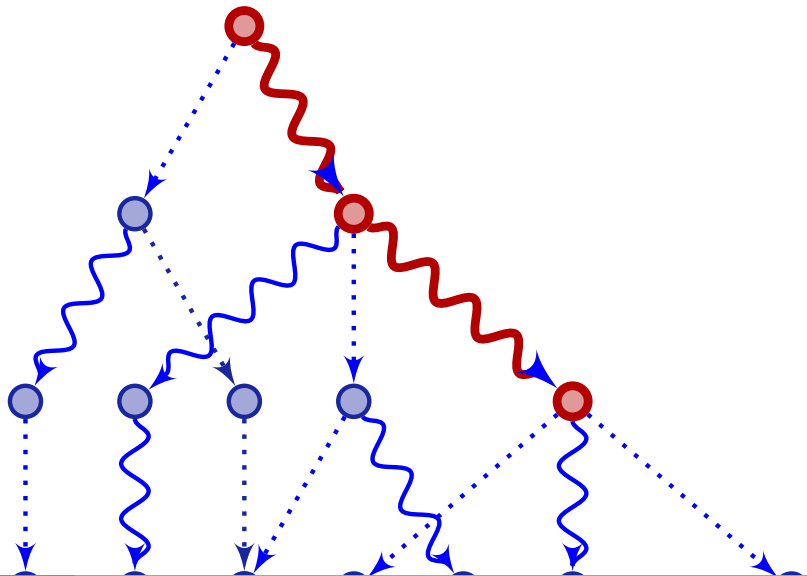
Want Hybrid Transition Systems

- continuous state space $\mathbb{R}^n$
- actions $A = D \dot{\cup} \mathbb{R}_{\geq 0}$ that are either discrete (D) or continuous ($\mathbb{R}_{\geq 0}$)
- Discrete transition somehow

$$x \xrightarrow{a} x^+ \quad \text{justified by discrete action } a \in D$$

- Continuous transition somehow

$$x \xrightarrow{a} x^+ \quad \text{by continuous flow of duration } a \in \mathbb{R}_{\geq 0} \text{ from } x \text{ to } x^+$$



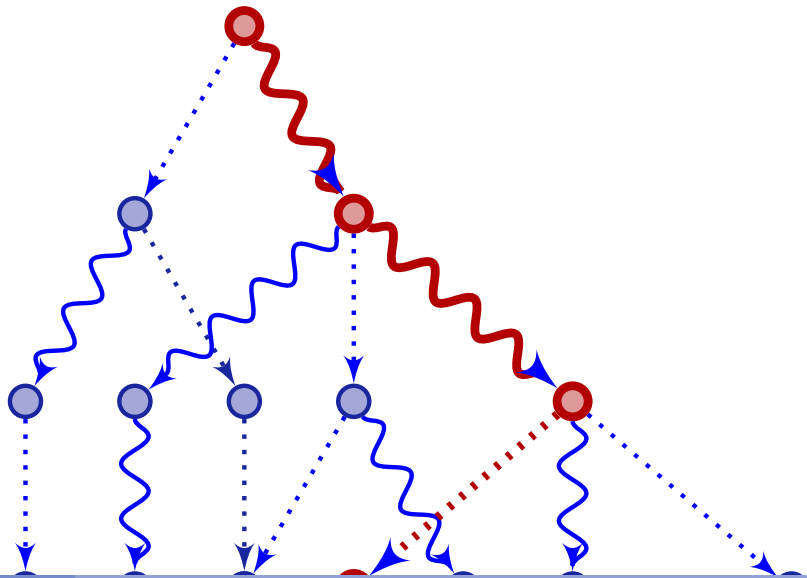
Want Hybrid Transition Systems

- continuous state space $\mathbb{R}^n$
- actions $A = D \dot{\cup} \mathbb{R}_{\geq 0}$ that are either discrete (D) or continuous ($\mathbb{R}_{\geq 0}$)
- Discrete transition somehow

$$x \xrightarrow{a} x^+ \quad \text{justified by discrete action } a \in D$$

- Continuous transition somehow

$$x \xrightarrow{a} x^+ \quad \text{by continuous flow of duration } a \in \mathbb{R}_{\geq 0} \text{ from } x \text{ to } x^+$$



Want Hybrid Transition Systems

- continuous state space $\mathbb{R}^n$
- actions $A = D \dot{\cup} \mathbb{R}_{\geq 0}$ that are either discrete (D) or continuous ($\mathbb{R}_{\geq 0}$)
- Discrete transition somehow

$$x \xrightarrow{a} x^+ \quad \text{justified by discrete action } a \in D$$

- Continuous transition somehow

$$x \xrightarrow{a} x^+ \quad \text{by continuous flow of duration } a \in \mathbb{R}_{\geq 0} \text{ from } x \text{ to } x^+$$

Okay but how do we "justify" the respective transitions from our model??

Hybrid Automata

"Graph switching between continuous dynamical systems"

Definition (Hybrid Automata)

Hybrid Automata

"Graph switching between continuous dynamical systems"

Definition (Hybrid Automata)

- continuous state space $\mathbb{R}^n$

Hybrid Automata

"Graph switching between continuous dynamical systems"

Definition (Hybrid Automata)

- continuous state space $\mathbb{R}^n$
- finite directed graph: vertices Q (*modes*), edges E (*control switches*)

Hybrid Automata

"Graph switching between continuous dynamical systems"

Definition (Hybrid Automata)

- continuous state space $\mathbb{R}^n$
- finite directed graph: vertices Q (*modes*), edges E (*control switches*)
- *flows* φ_q, where $\varphi_q(t; x) \in \mathbb{R}^n$ is the state reached after staying in mode $q \in Q$ for time $t \geq 0$ when continuous evolution starts in state $x \in \mathbb{R}^n$

Hybrid Automata

"Graph switching between continuous dynamical systems"

Definition (Hybrid Automata)

- continuous state space $\mathbb{R}^n$
- finite directed graph: vertices Q (*modes*), edges E (*control switches*)
- *flows* φ_q, where $\varphi_q(t; x) \in \mathbb{R}^n$ is the state reached after staying in mode $q \in Q$ for time $t \geq 0$ when continuous evolution starts in state $x \in \mathbb{R}^n$
- Evolution domain constraints $inv_q \subseteq \mathbb{R}^n$ for $q \in Q$

Hybrid Automata

"Graph switching between continuous dynamical systems"

Definition (Hybrid Automata)

- continuous state space $\mathbb{R}^n$
- finite directed graph: vertices Q (*modes*), edges E (*control switches*)
- *flows* φ_q, where $\varphi_q(t; x) \in \mathbb{R}^n$ is the state reached after staying in mode $q \in Q$ for time $t \geq 0$ when continuous evolution starts in state $x \in \mathbb{R}^n$
- Evolution domain constraints $inv_q \subseteq \mathbb{R}^n$ for $q \in Q$
- jump relations $jump_e \subseteq \mathbb{R}^n \times \mathbb{R}^n$ for edges $e \in E$
  usually comprising guard on current state and reset relations

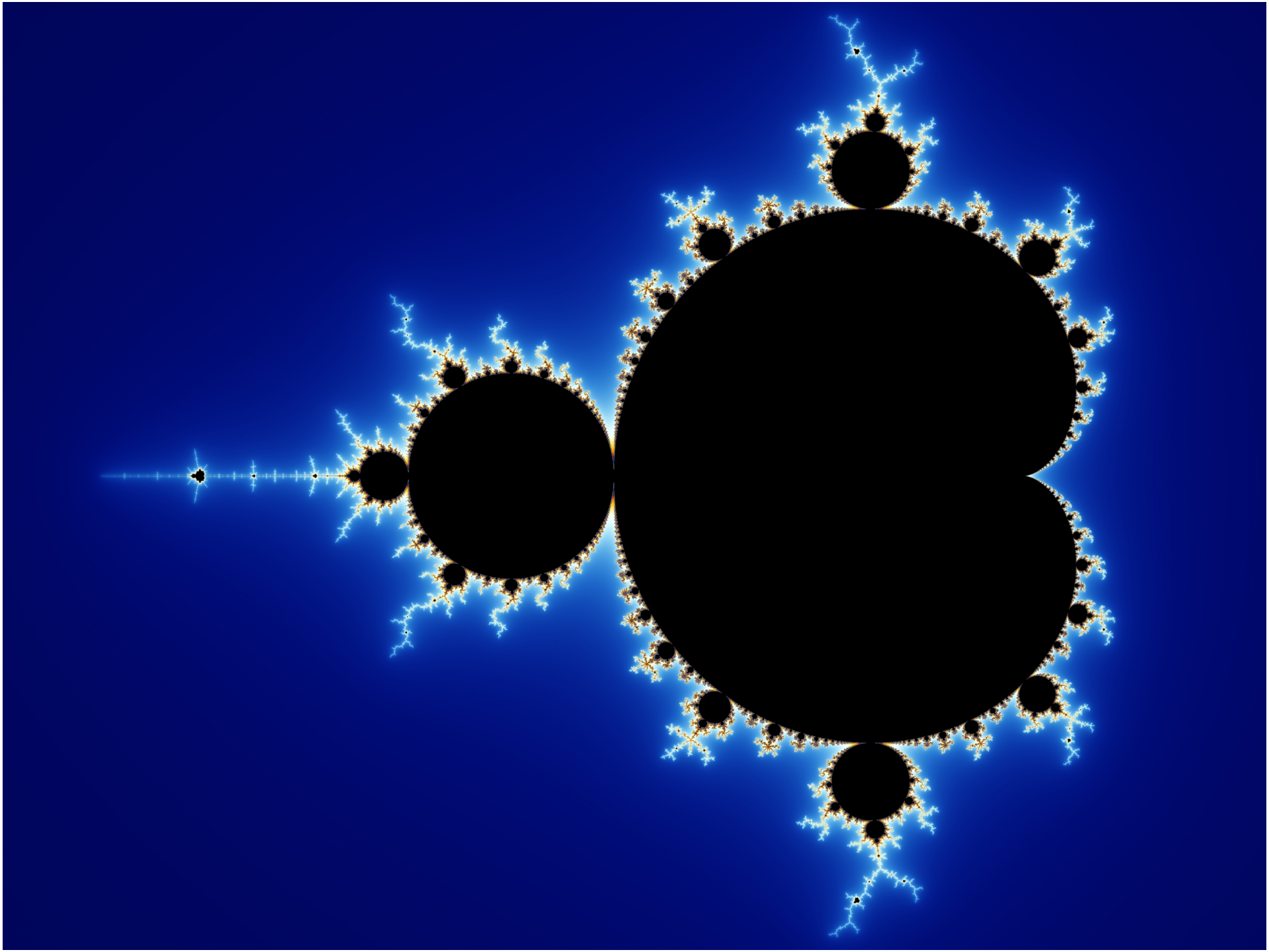
Hybrid Automata

"Graph switching between continuous dynamical systems"

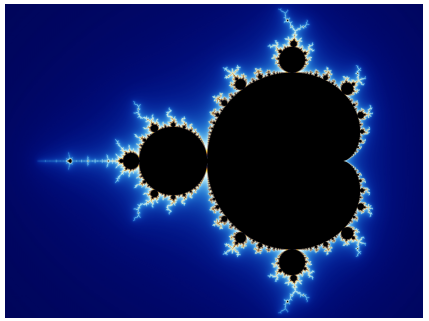
Definition (Hybrid Automata)

- continuous state space $\mathbb{R}^n$
- finite directed graph: vertices Q (*modes*), edges E (*control switches*)
- *flows* φ_q, where $\varphi_q(t; x) \in \mathbb{R}^n$ is the state reached after staying in mode $q \in Q$ for time $t \geq 0$ when continuous evolution starts in state $x \in \mathbb{R}^n$
- Evolution domain constraints $inv_q \subseteq \mathbb{R}^n$ for $q \in Q$
- jump relations $jump_e \subseteq \mathbb{R}^n \times \mathbb{R}^n$ for edges $e \in E$
  usually comprising guard on current state and reset relations

Is this a good definition?



Mandelbrot Fractal

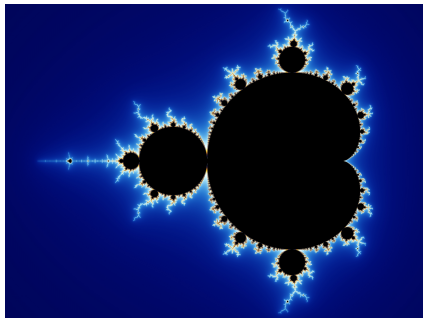


Example (Mandelbrot Set)

For complex numbers $c \in \mathbb{C}$ define $f_0(c) = c$ and $f_{n+1}(c) = f_n(c)^2 + c$. Then the Mandelbrot set is

$$\{c \in \mathbb{C} \ : \ f_n(c) \not\to \infty \text{ as } n \to \infty\}$$

Mandelbrot Fractal



Example (Mandelbrot Set)

For complex numbers $c \in \mathbb{C}$ define $f_0(c) = c$ and $f_{n+1}(c) = f_n(c)^2 + c$.
Then the Mandelbrot set is

$$\{c \in \mathbb{C} \ : \ f_n(c) \not\to \infty \text{ as } n \to \infty\}$$

Theorem (Lenore Blum, Cucker, Shub, Smale'90...98)

"The Mandelbrot set is undecidable over $\mathbb{R}$ / in Real Turing Machines"

Hybrid Automata

"Graph switching between continuous dynamical systems"

Definition (Hybrid Automata)

- continuous state space $\mathbb{R}^n$
- finite directed graph: vertices Q (*modes*), edges E (*control switches*)
- *flows* φ_q, where $\varphi_q(t; x) \in \mathbb{R}^n$ is the state reached after staying in mode $q \in Q$ for time $t \geq 0$ when continuous evolution starts in state $x \in \mathbb{R}^n$
- Evolution domain constraints $inv_q \subseteq \mathbb{R}^n$ for $q \in Q$
- jump relations $jump_e \subseteq \mathbb{R}^n \times \mathbb{R}^n$ for edges $e \in E$
  usually comprising guard on current state and reset relations

Is this a good definition?

Hybrid Automata

"Graph switching between continuous dynamical systems"

Definition (Hybrid Automata)

- continuous state space $\mathbb{R}^n$
- finite directed graph: vertices Q (*modes*), edges E (*control switches*)
- *flows* φ_q, where $\varphi_q(t; x) \in \mathbb{R}^n$ is the state reached after staying in mode $q \in Q$ for time $t \geq 0$ when continuous evolution starts in state $x \in \mathbb{R}^n$
- Evolution domain constraints $inv_q \subseteq \mathbb{R}^n$ for $q \in Q$
- jump relations $jump_e \subseteq \mathbb{R}^n \times \mathbb{R}^n$ for edges $e \in E$
  usually comprising guard on current state and reset relations

What if inv_q is a Mandelbrot set?

Hybrid Automata

"Graph switching between continuous dynamical systems"

Definition (Hybrid Automata)

- continuous state space $\mathbb{R}^n$
- finite directed graph: vertices Q (*modes*), edges E (*control switches*)
- *flows* φ_q, where $\varphi_q(t; x) \in \mathbb{R}^n$ is the state reached after staying in mode $q \in Q$ for time $t \geq 0$ when continuous evolution starts in state $x \in \mathbb{R}^n$
- Evolution domain constraints $inv_q \subseteq \mathbb{R}^n$ for $q \in Q$
- jump relations $jump_e \subseteq \mathbb{R}^n \times \mathbb{R}^n$ for edges $e \in E$
  usually comprising guard on current state and reset relations

And all relations decidable / definable in first-order real arithmetic etc.

Computationally relevant output needs computational input!


M. S. Branicky, V. S. Borkar, and S. K. Mitter.
A unified framework for hybrid control: Model and optimal control theory.
IEEE T. Automat. Contr., 43(1):31–45, 1998.


T. A. Henzinger.
The theory of hybrid automata.
In *LICS*, pages 278–292, Los Alamitos, 1996. IEEE Computer Society.


A. Platzer.
Logical Analysis of Hybrid Systems: Proving Theorems for Complex Dynamics.
Springer, Heidelberg, 2010.