

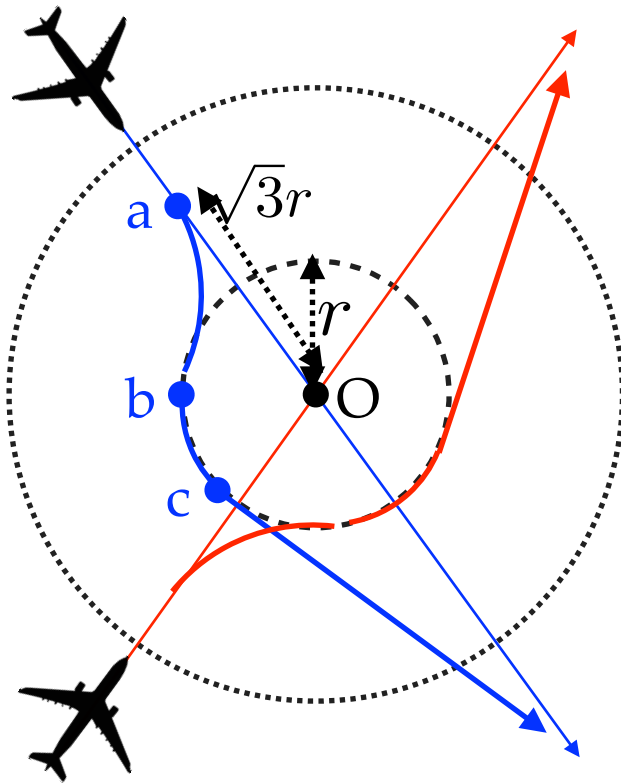
# Verification of Cyber-Physical Systems

Pavithra Prabhakar  
Kansas State University

Lecture 5 & 6: Hybrid System Safety Analysis

Global Initiative of Academic Networks  
Indian Institute of Science

# Air traffic collision avoidance protocol



$\mathbf{x} = (x_1, x_2)$ : position of the airplane

$\mathbf{d} = (d_1, d_2)$ : velocity of the airplane

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{d}_1 \\ \dot{d}_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\omega \\ 0 & 0 & \omega & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ d_1 \\ d_2 \end{bmatrix}$$

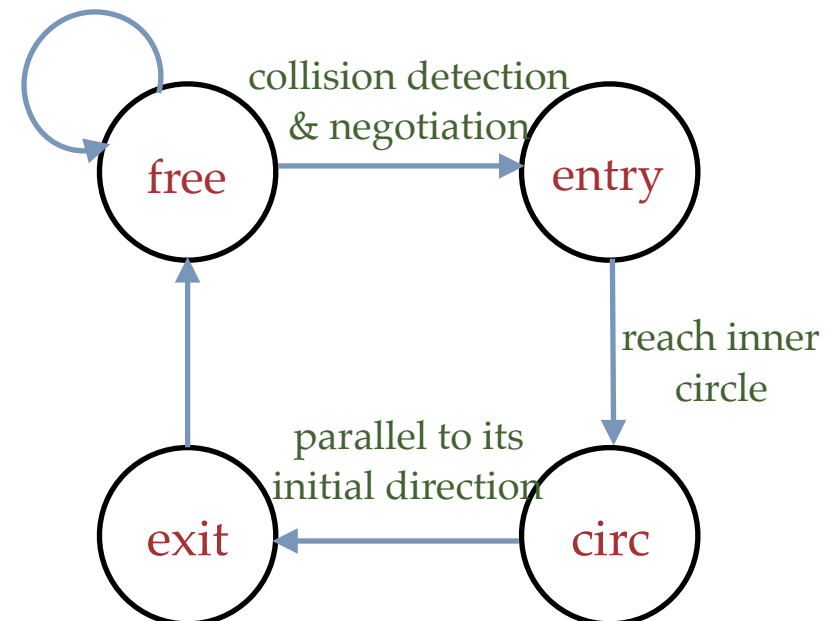
$\omega$ : the angular velocity

## Minimum separation

*The aircraft maintain a minimum distance between them always*

$$\begin{aligned} \|x - y\| &\leq p & c &= x + \lambda d = y + \lambda e \\ \|x - c\| &= \sqrt{3}r & (r\omega)^2 &= \|d\|^2 & x^0 &:= x, d^0 := d \end{aligned}$$

$$\omega := *$$

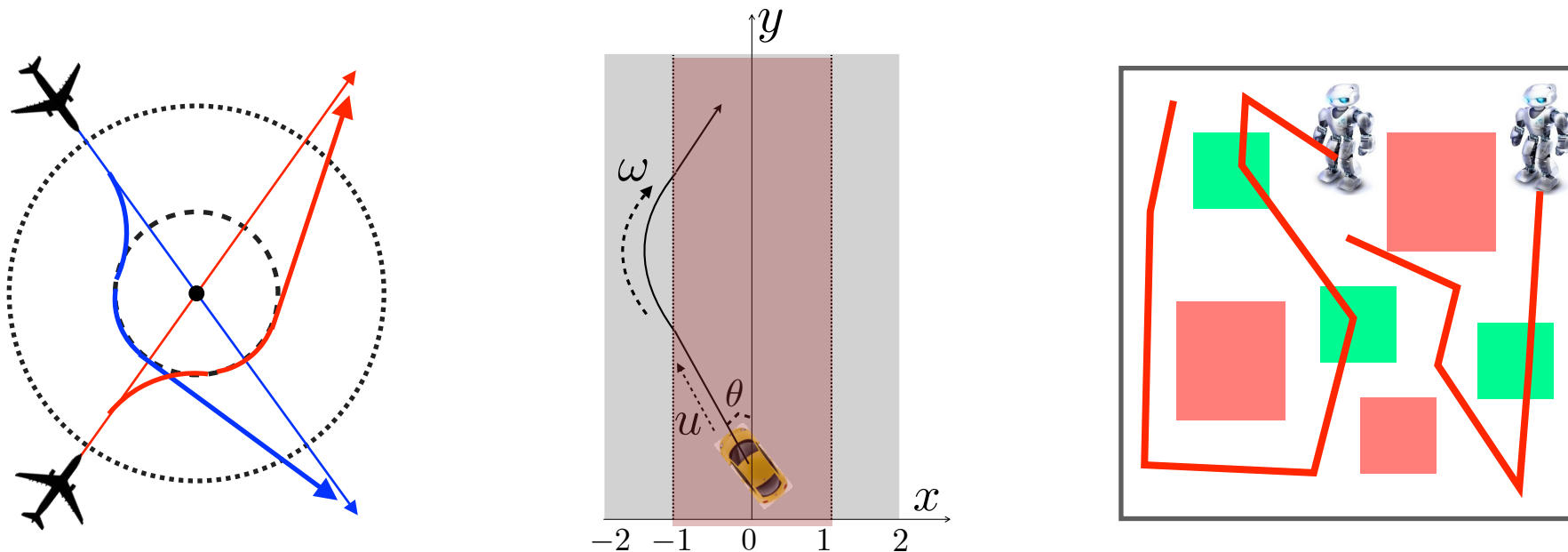


$$\begin{aligned} \|x - c\| &\leq r \\ \omega &:= -\omega \end{aligned}$$

$$\omega := 0 \quad x + \lambda_2 d = x^0 + \lambda_1 d^0$$

# Correctness Specification: Safety

Every execution of the system is error free



- ✧ Air-traffic control: collision avoidance
- ✧ Autonomous cars: vehicle always remains in the lane
- ✧ Multi-robot navigation: collision avoidance

---

# Hybrid System Syntax and Semantics

- ❖ Systems with mixed discrete and continuous behaviors
- ❖ Combine finite state automata and differential equations



# Hybrid Automaton Model

---

A hybrid automaton  $\mathcal{H} = (Q, X, Act, Prop, q_0, X_0, F, I, E, Lab)$

- $Q$  is a set of discrete location;
- $X = \mathbb{R}^n$  is a set of continuous state space;
- $Act$  is a set of actions;
- $Prop$  is a set of propositions;
- $q_0 \in Q$  is the initial location;
- $X_0 \subseteq X$  is a set of initial continuous states;
- $F : Q \times X \rightarrow X$  specifies the vector field for each location;
- $I : Q \rightarrow 2^X$  specifies the invariant for each location;
- $E \subseteq Q \times Act \times 2^{X \times X} \times Q$  is a set of edges;
- $Lab : Q \rightarrow 2^{Prop}$  is the labeling function.

# Two vehicles at an intersection

$(x_1(t), y_1(t))$ : vehicle 1 position at time  $t$

$(x_2(t), y_2(t))$ : vehicle 2 position at time  $t$

**Vehicle 1 dynamics moving east**


$$\dot{x}_1 = r$$

$$\dot{y}_1 = 0$$

**Vehicle 1 dynamics moving north**

$$\dot{x}_1 = 0$$

$$\dot{y}_1 = r$$


$(x_{01}, y_{01})$  

$(0, 0)$

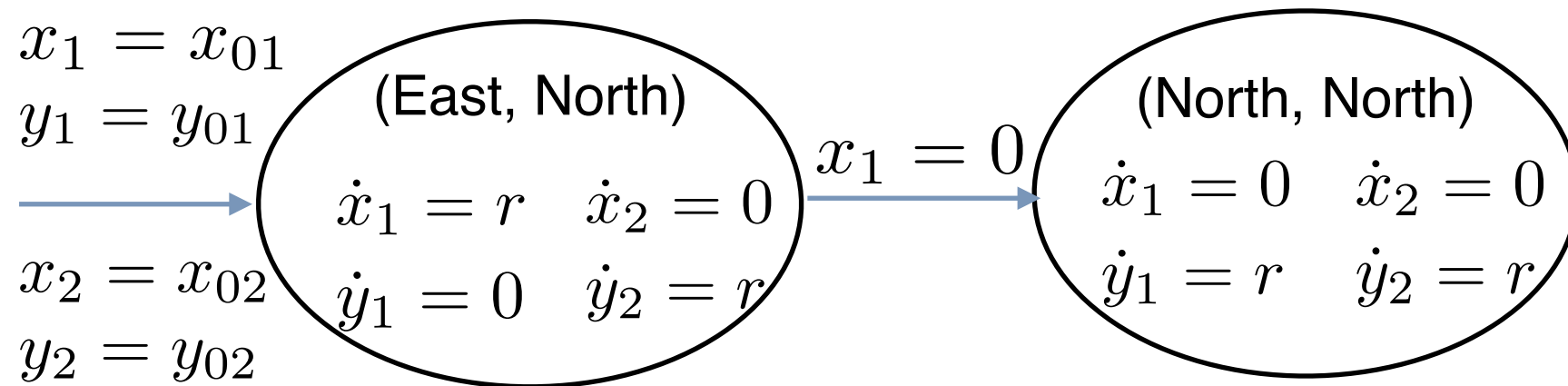
**Vehicle 2 dynamics moving north**

$$\dot{x}_2 = 0$$

$$\dot{y}_2 = r$$

$(x_{02}, y_{02})$  

# Hybrid Automaton Model



Locations  $Q = \{(East, North), (North, North)\}$

Initial Location =  $(East, North)$

Continuous statespace  $X = \mathbb{R}^4$

Initial continuous states  $X_0 = \{(x_{01}, y_{01}, x_{02}, y_{02})\}$

$F((East, North), (x_1, y_1, x_2, y_2)) = (r, 0, 0, r)$

$F((North, North), (x_1, y_1, x_2, y_2)) = (0, r, 0, r)$

$I((North, North)) = I((North, North)) = \mathbb{R}^4$

Edge  $E = \{((East, North), J, (North, North))\}$ ,

$J = \{((x_1, y_1, x_2, y_2), (x_1, y_1, x_2, y_2)) \mid x_1 = 0\}$

# Hybrid Automaton Semantics

The semantics of a hybrid automaton  $\mathcal{H} = (Q, X, Act, Prop, q_0, X_0, F, I, J, Lab)$  is the transition system  $\mathcal{T}_{\mathcal{H}} = (S, S_0, A, Prop, \rightarrow, L)$ , where:

- $S = Q \times X;$

Statespace

- $S_0 = \{q_0\} \times X_0;$

Initial states

- $A = E \cup \mathbb{R}_{\geq 0};$

Actions

- $\rightarrow$  consists of continuous and discrete transitions:

- Continuous transition:  $(q, x) \xrightarrow{t} (q', x'), t \in \mathbb{R}_{\geq 0};$

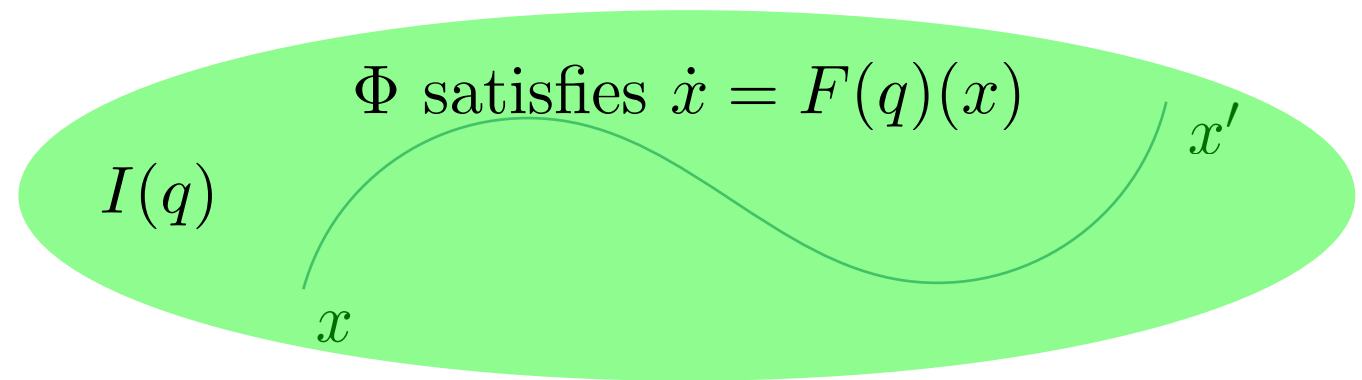
- Discrete transition:  $(q, x) \xrightarrow{e} (q', x'), e \in E;$

- $L : Q \times X \rightarrow Prop$  given by  $L(q, x) = Lab(q).$

Labeling function

# Continuous transitions

Capture the state change  
due to time evolution



$(q, x) \xrightarrow{t} (q', x')$  if  $q = q'$  and there exists a function  $\Phi : [0, t] \rightarrow \mathbb{R}^n$  such that

- $\Phi$  satisfies the differential equation corresponding to location  $q$

$$\frac{d\Phi}{dt}(t) = F(q)(\Phi(t))$$

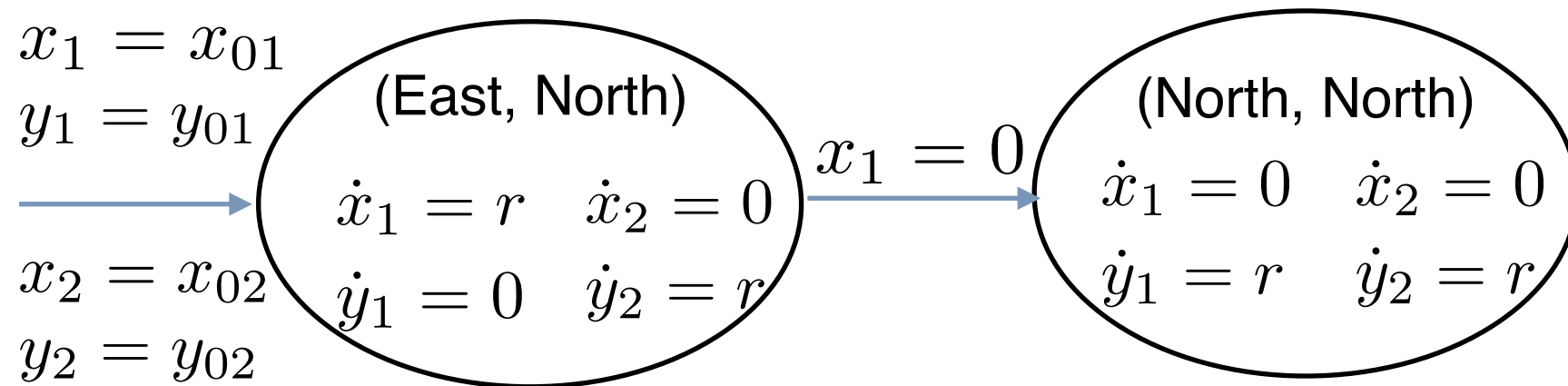
- $\Phi$  evolves from  $x$  at time 0 to  $x'$  at time  $t$

$$\Phi(0) = x, \Phi(t) = x'$$

- $\Phi$  remains in the invariant of  $q$  all along the evolution

$$\Phi(t') \in I(q), \forall t' \in [0, t]$$

# Continuous transition example



Let  $r = 2$  and continuous state variables be  $(x_1, y_1, x_2, y_2)$

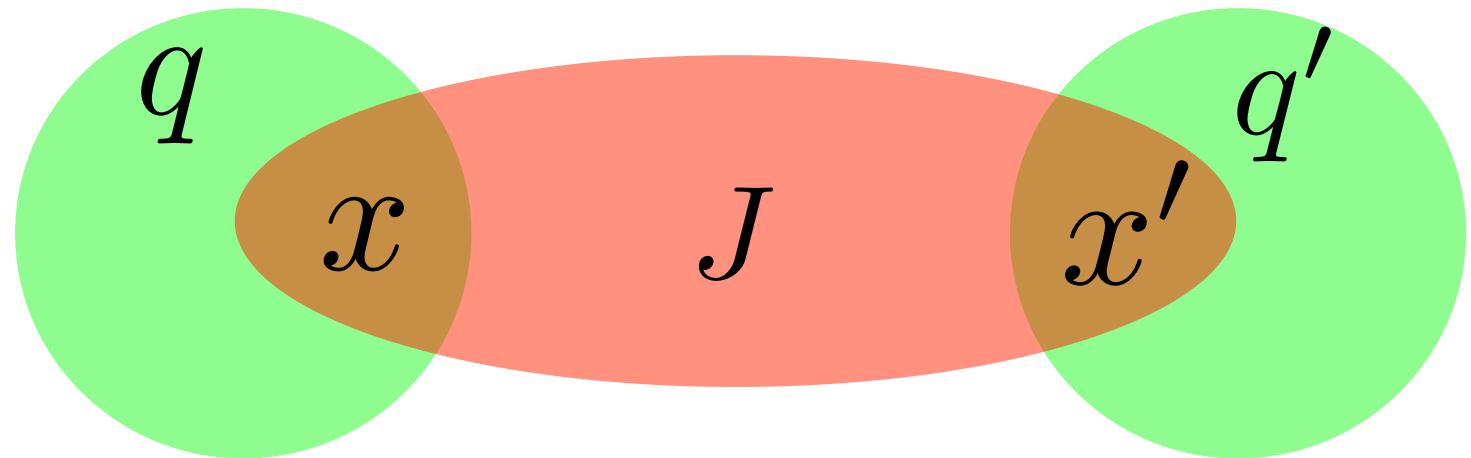
$((\text{East, North}), (-2, 0, 0, -4)) \xrightarrow{3} ((\text{East, North}), (4, 0, 0, 2))$   
is a continuous transition, since

the function  $\Phi(t) = (-2 + 2t, 0, 0, -4 + 2t)$  satisfies:

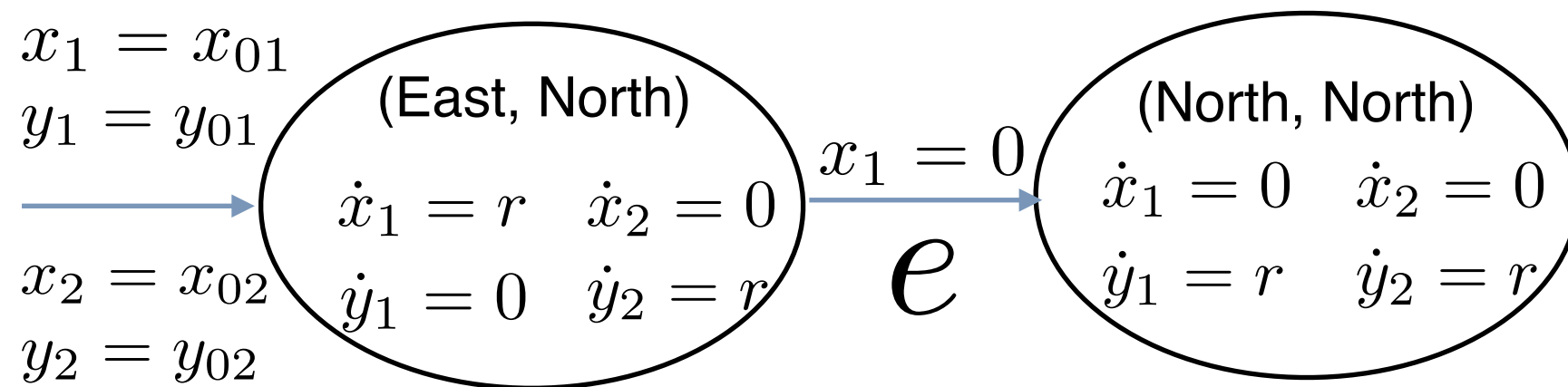
- $\frac{d\Phi}{dt}(t) = (2, 0, 0, 2) = F((\text{East, North}))(\Phi(t))$
- $\Phi(0) = (-2, 0, 0, -4)$ ,  $\Phi(3) = (4, 0, 0, 2)$
- $\Phi(t') = (-2 + 2t', 0, 0, -4 + 2t') \in \mathbb{R}^4 = I((\text{East, North}))$  for all  $t' \in [0, 3]$

# Discrete transitions

Capture the state change  
due to a mode change



$$(q, x) \xrightarrow{e} (q', x') \text{ if } e = (q, J, q') \in E \text{ and } (x, x') \in J$$



$$((\text{East, North}), (0, 0, 0, -2)) \xrightarrow{e} ((\text{North, North}), (0, 0, 0, -2)) \quad \text{Yes}$$

$$((\text{East, North}), (4, 0, 0, 2)) \xrightarrow{e} ((\text{North, North}), (4, 0, 0, 2)) \quad \text{No}$$

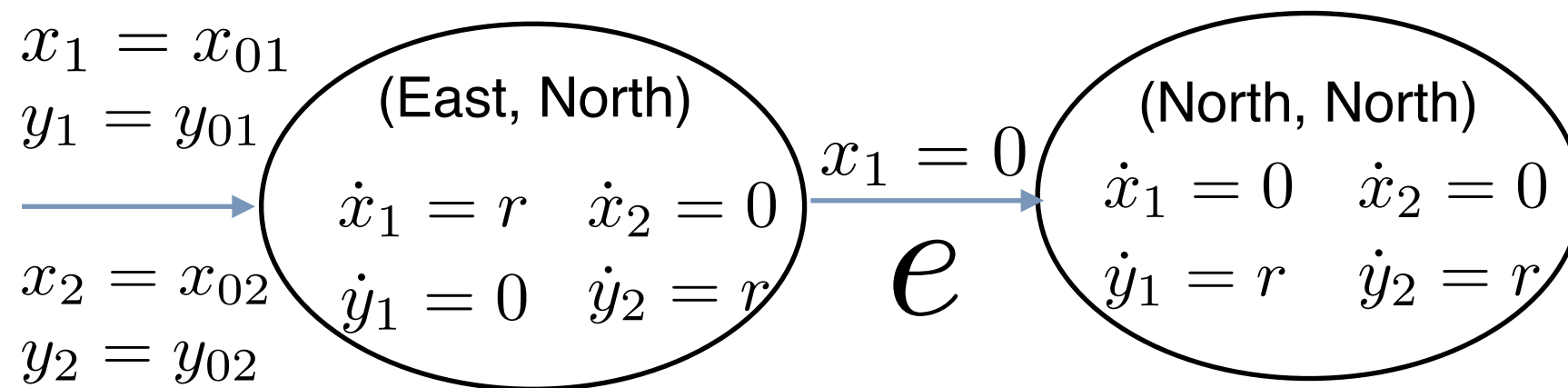
# Executions

Captures the state evolution of a hybrid system through time elapse and mode changes.

An **execution** of the hybrid system is a (finite or infinite) path in its transition system.

$$\sigma = (q_0, x_0) \xrightarrow{a_1} (q_1, x_1) \xrightarrow{a_2} (q_2, x_2) \xrightarrow{a_3} (q_3, x_3) \dots$$

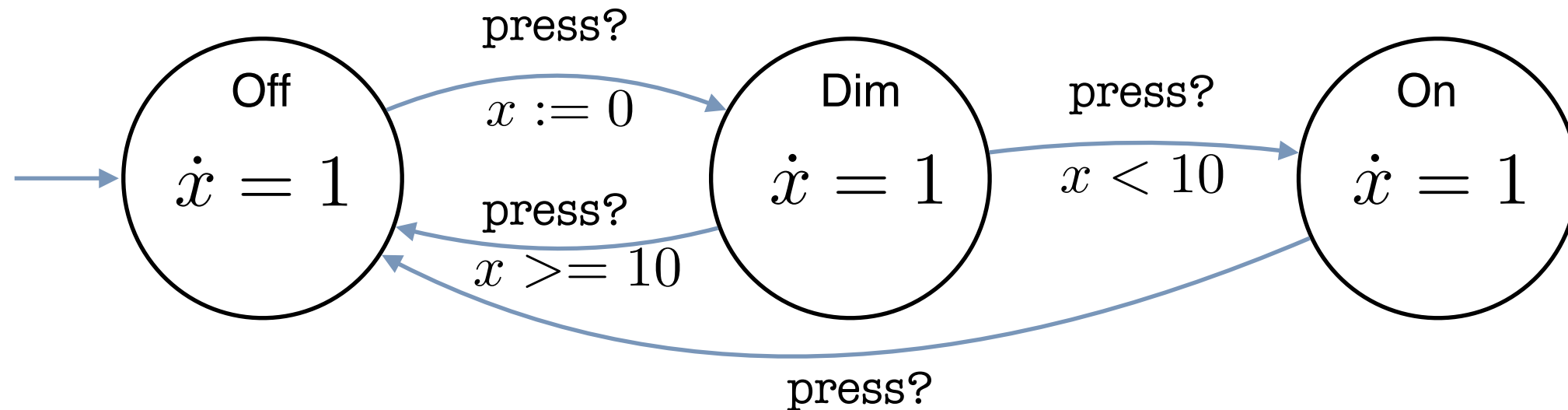
where each of the transitions is either discrete or continuous.



$$\begin{aligned} & ((\text{East, North}), (-2, 0, 0, -4)) \xrightarrow{1} ((\text{East, North}), (0, 0, 0, -2)) \\ & \xrightarrow{e} ((\text{North, North}), (0, 0, 0, -2)) \xrightarrow{2} ((\text{North, North}), (0, 2, 0, 0)) \end{aligned}$$



# Executions



$$\sigma = (Off, 0) \xrightarrow{5} (Off, 5) \xrightarrow{press?} (Dim, 0) \xrightarrow{5} (Dim, 5)$$

$$\xrightarrow{press?} (On, 5) \xrightarrow{2} (On, 7) \xrightarrow{press?} (Off, 7) \xrightarrow{5} (Off, 12)$$

$$\xrightarrow{press?} (Dim, 0) \xrightarrow{12} (Dim, 12) \xrightarrow{press?} (Off, 12) \dots$$

# Reachability

---

- A state  $(q', x')$  is reachable from a state  $(q, x)$  in a hybrid system  $\mathcal{H}$ , if there is an execution of  $\mathcal{H}$  that starts at  $(q, x)$  and reaches  $(q', x')$ , that is,  $(q, x) = (q_0, x_0) \xrightarrow{a_0} (q_1, x_1) \xrightarrow{a_1} \dots (q_n, x_n) = (q', x')$  is an execution of  $\mathcal{H}$ .
- Given a set of states  $S_0$  of  $\mathcal{H}$ ,  $Reach_{\mathcal{H}}(S_0)$  is the set of all states reachable from some state in  $S_0$ .
- The time elapsed during the execution (duration) is the sum of all  $a_i$ s which correspond to continuous transitions.
- The number of (discrete) steps of the execution is the number of  $a_i$ s which correspond to discrete transitions.

$$\begin{aligned} & ((\text{East, North}), (-2, 0, 0, -4)) \xrightarrow{1} ((\text{East, North}), (0, 0, 0, -2)) \\ & \xrightarrow{e} ((\text{North, North}), (0, 0, 0, -2)) \xrightarrow{2} ((\text{North, North}), (0, 2, 0, 0)) \end{aligned}$$

The duration of the above execution is 3 and the number of discrete steps is 1.

# Trajectories

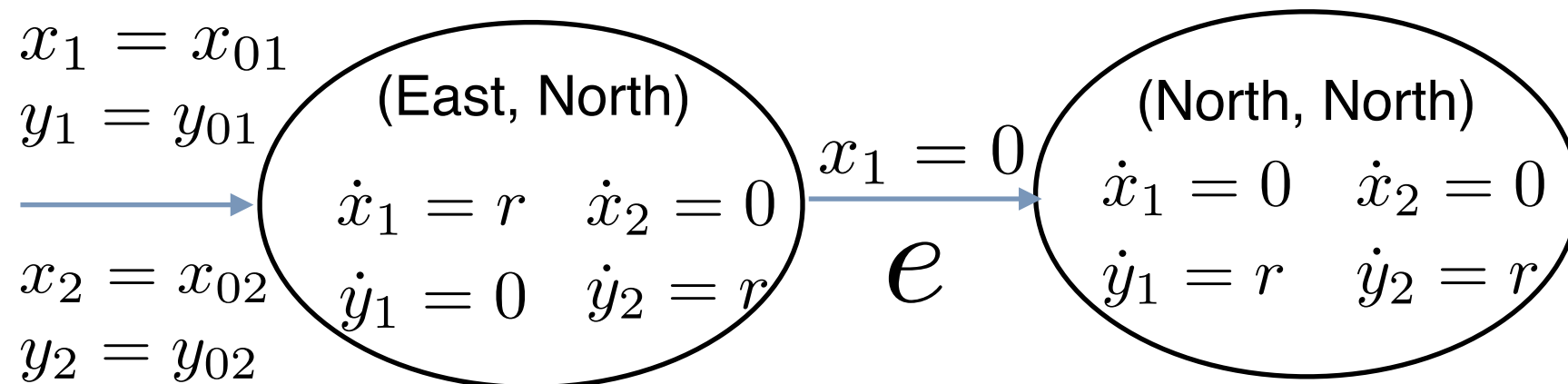
Need the value of continuous state at all time

Consider systems where continuous state remains same during mode switch

- An trajectory is a function  $\tau : [0, \infty) \rightarrow \mathbb{R}^n$ .
- A trajectory  $\tau$  of a hybrid system captures the continuous states along an execution.
- More precisely,  $\tau$  corresponding to an execution  $\sigma$  is such that  $\tau(t)$  captures the value of the continuous state in  $\sigma$  reached after total time  $t$  elapse.

We will use trajectories and executions as well as symbols representing them interchangeably

# Trajectories



**Execution:**

$$((\text{East, North}), (-2, 0, 0, -4)) \xrightarrow{1} ((\text{East, North}), (0, 0, 0, -2)) \xrightarrow{e} ((\text{North, North}), (0, 0, 0, -2)) \xrightarrow{2} ((\text{North, North}), (0, 2, 0, 0))$$

**Trajectory:**

$$\sigma(t) = \begin{cases} (-2 + 2t, & 0, & 0, & -4 + 2t) & \text{for } t \in [0, 1] \\ (0, & 2(t - 1), & 0, & -2 + 2(t - 1)) & \text{for } t \in [1, 3] \end{cases}$$

---

# Safety Problem

# Two vehicles at an intersection


$(x_1(t), y_1(t))$ : vehicle 1 position at time  $t$

$(x_2(t), y_2(t))$ : vehicle 2 position at time  $t$

**Vehicle 1 dynamics**

$$\dot{x}_1 = r$$

$$\dot{y}_1 = 0$$

$(x_{01}, y_{01})$    $r$

**Vehicle 1 dynamics moving**

$$\dot{x}_1 = 0$$


$$\dot{y}_1 = r$$

$(0, 0)$

**Vehicle 2 dynamics moving**

$$\dot{x}_2 = 0$$

$$\dot{y}_2 = r$$

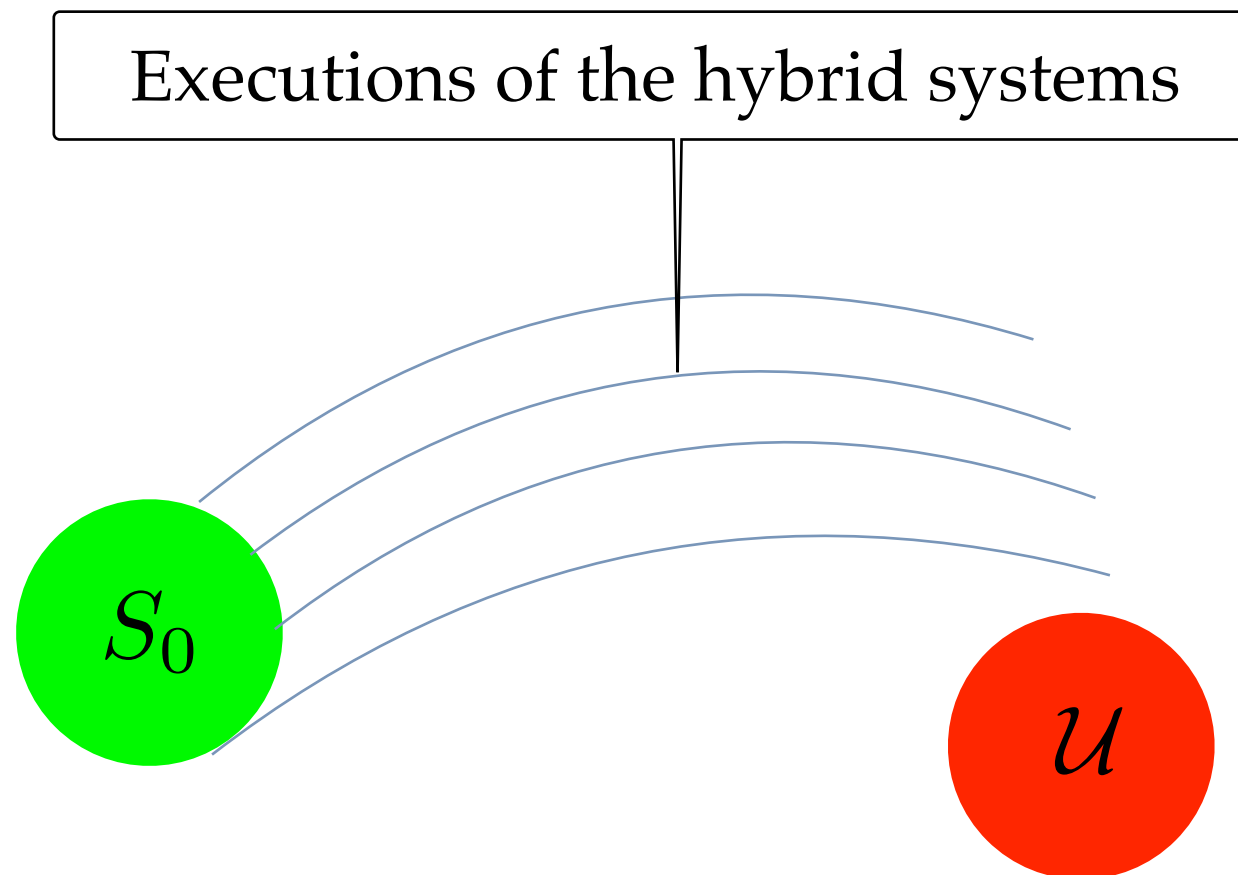
$(x_{02}, y_{02})$    $r$

Do the two vehicles collide?

# Safety Problem

---

Given a hybrid automaton  $\mathcal{H}$ , and a set of unsafe states  $\mathcal{U}$ , is any state of  $\mathcal{U}$  reachable from  $S_0$ , the initial states of  $\mathcal{H}$ ?

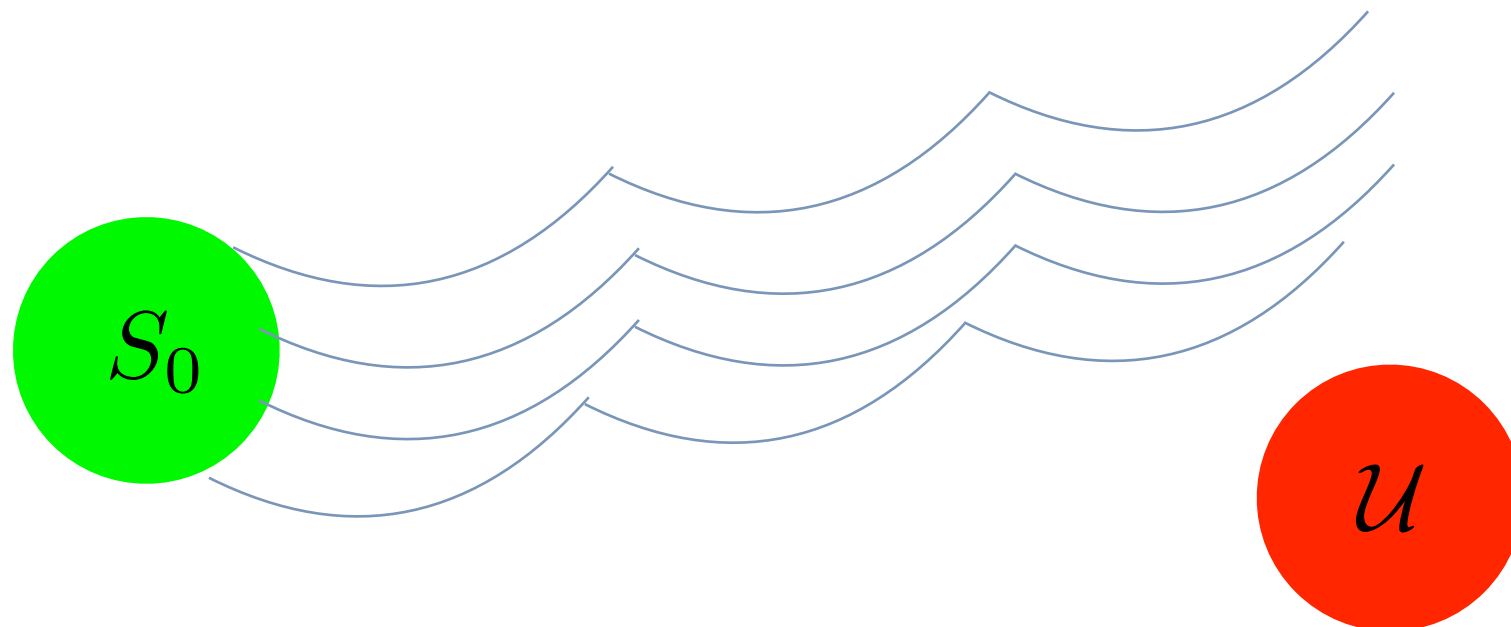


Equivalently, is  $Reach_{\mathcal{H}}(S_0) \cap \mathcal{U} \neq \emptyset$ ?

# Bounded Safety Problem

---

Given a hybrid automaton  $\mathcal{H}$ , a set of unsafe states  $\mathcal{U}$ , a positive integer  $k$  and a positive real number  $T$ , does there exist an execution with at most  $k$  discrete transitions and duration at most  $T$ , that reaches a state of  $\mathcal{U}$  starting from  $S_0$ ?





# Bounded Safety Analysis

---

- ❖ We will encode the executions of bounded duration and bounded number of discrete transitions as an SMT formula
- ❖ Every satisfiable instance of the formula will correspond to an execution and vice versa
- ❖ The SMT formula along with the unsafe set is satisfiable if and only if the bounded safety is violated

# Encoding executions — Components

---

- ✦ Encode continuous and discrete transitions

$\varphi_C(s, t, s')$  if and only if  $s \xrightarrow{t} s'$

Formula encoding continuous transitions

$\varphi_D(s, s')$  if and only if  $s \xrightarrow{e} s'$  for some  $e \in E$

Formula encoding discrete transitions

- ✦ Encoding initial and unsafe states

$\varphi_0(s)$  if and only if  $s \in S_0$

$\varphi_U(s)$  if and only if  $s \in \mathcal{U}$

# Encoding executions

---

- ✧ To encode executions with  $k$  discrete transitions, create  $2k + 2$  state variables and  $k+1$  time variables to capture the states and time evolutions in an execution of length  $k$

$$s_1 \xrightarrow{t_1} s'_1 \xrightarrow{a_1} s_2 \xrightarrow{t_2} s'_2 \xrightarrow{a_2} \dots s_k \xrightarrow{t_k} s'_k \xrightarrow{a_k} s_{k+1} \xrightarrow{t_{k+1}} s'_{k+1}$$

- ✧ The following formula encodes executions with  $k$  discrete transitions and at most duration  $T$

$$\begin{aligned} \varphi_{\sigma}^{k,T}(s_1, t_1, s'_1, a_1, s_2, \dots, s_{k+1}) &:= \varphi_0(s_1) \wedge \bigwedge_{i=1}^k \varphi_D(s'_i, s_{i+1}) \\ &\wedge \bigwedge_{i=1}^{k+1} \varphi_C(s_i, t_i, s'_i) \wedge t_1 + t_2 + \dots + t_k \leq T \end{aligned}$$

# Encoding bounded safety problem

---

- ✦ The following formula encodes executions with at most  $k$  discrete transitions and at most duration  $T$

$$\varphi_{\sigma}^{\leq k, T}(s_1, t_1, s'_1, a_1, s_2, \dots, s_{k+1}) :=$$

$$\bigvee_{j=0}^k \varphi_{\sigma}^{j, T}(s_1, t_1, s'_1, a_1, s_2, \dots, s_{j+1})$$

Executions with 0 transitions or 1 transition or 2 transitions .....

- ✦ Unsafe set is reachable in at most  $k$  discrete transitions and at most duration  $T$  if and only if the following formula is satisfiable:

$$\bigvee_{j=0}^k [\varphi_{\sigma}^{j, T}(s_1, t_1, s'_1, a_1, s_2, \dots, s_{j+1}) \wedge \varphi_U(s_{j+1})]$$

---

# Bounded Safety Analysis: Illustration

# Two vehicles at an intersection


$(x_1(t), y_1(t))$ : vehicle 1 position at time  $t$

**Vehicle 1 dynamics moving east**

$$\begin{aligned}\dot{x}_1 &= r \\ \dot{y}_1 &= 0\end{aligned}$$

**Vehicle 1 dynamics moving north**


$$\begin{aligned}\dot{x}_1 &= 0 \\ \dot{y}_1 &= r\end{aligned}$$

$(x_{01}, y_{01})$  

$(0, 0)$

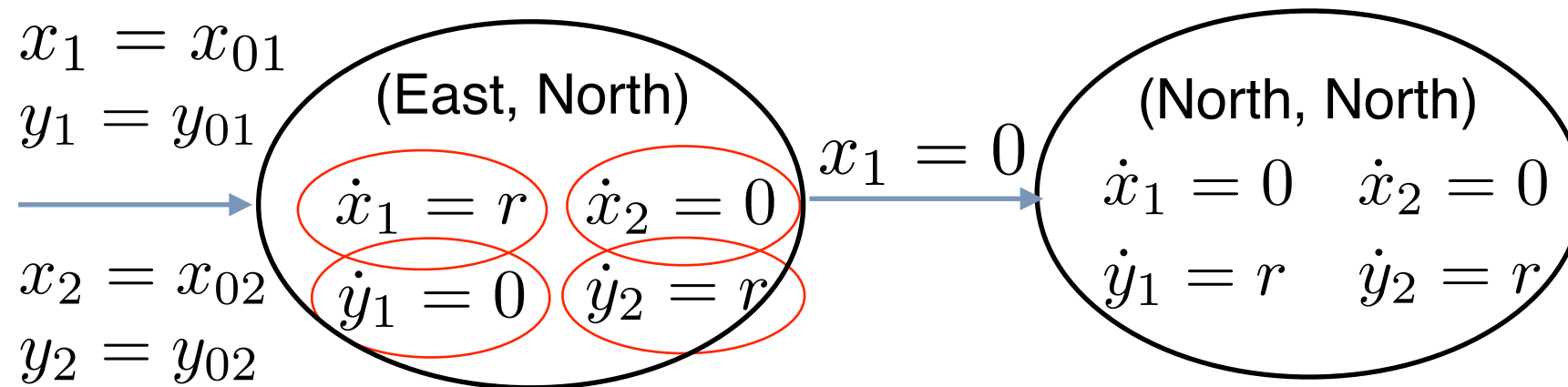
**Vehicle 2 dynamics moving north**

$$\begin{aligned}\dot{x}_2 &= 0 \\ \dot{y}_2 &= r\end{aligned}$$

$(x_{02}, y_{02})$  

$(x_2(t), y_2(t))$ : vehicle 2 position at time  $t$

# Encoding continuous transitions



Encodes (East, North) dynamics

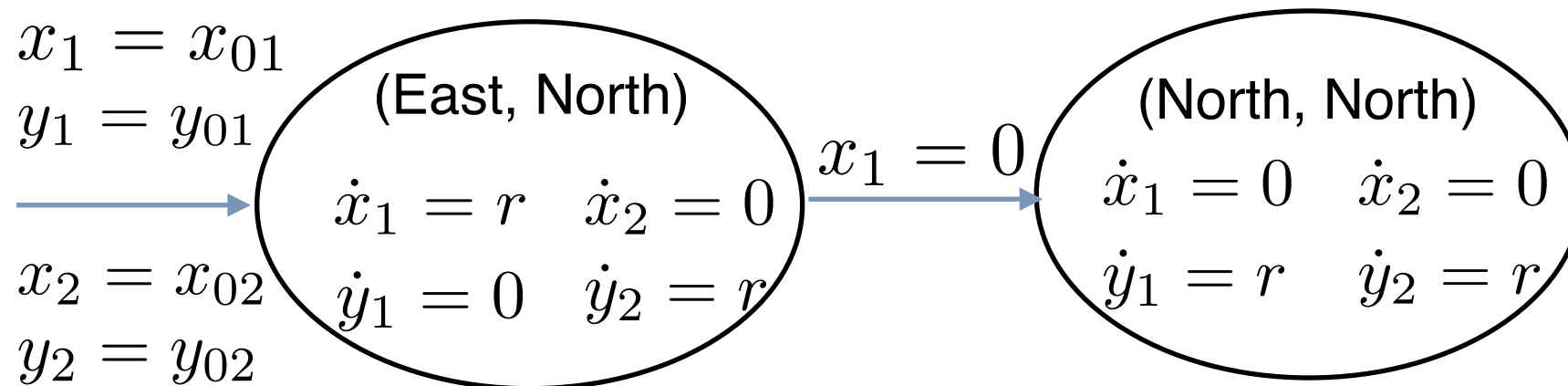
Encodes (North, North) dynamics

$$\varphi_C(l, x_1, y_1, x_2, y_2, t, l', x'_1, y'_1, x'_2, y'_2) :=$$

$$[l = l' = 1 \wedge x'_1 = x_1 + rt \wedge y'_1 = y_1 \wedge x'_2 = x_2 \wedge y'_2 = y_2 + rt]$$

$$\vee [l = l' = 2 \wedge x'_1 = x_1 \wedge y'_1 = y_1 + rt \wedge x'_2 = x_2 \wedge y'_2 = y_2 + rt]$$

# Encoding discrete transitions



$$\varphi_D(l, x_1, y_1, x_2, y_2, l', x'_1, y'_1, x'_2, y'_2) :=$$

$$l = 1 \wedge l' = 2 \wedge x_1 = 0 \wedge x'_1 = x_1 \wedge y'_1 = y_1 \wedge x'_2 = x_2 \wedge y'_2 = y_2$$

The vehicles collide either

- ❖ before Vehicle 1 enters the intersection (0 discrete transitions), or
- ❖ after it turns at the intersection (1 discrete transition)

Let us say the vehicles start at  $(-5, 0)$  and  $(0, -4)$  with speed 1



# Can the vehicles collide with 0 discrete transitions?

Initial condition

$$l^1 = 1 \wedge x_1^1 = -5 \wedge y_1^1 = 0 \wedge x_2^1 = 0 \wedge y_2^1 = -4$$

Continuous transition

$$[l^1 = l'^1 = 1 \wedge x_1'^1 = x_1^1 + t \wedge y_1'^1 = y_1^1 \wedge x_2'^1 = x_2^1 \wedge y_2'^1 = y_2^1 + t]$$

Unsafe set (collision)

$$\begin{aligned} &[l = l' = 1 \wedge x_1' = x_1 + rt \wedge y_1' = y_1 \wedge x_2' = x_2 \wedge y_2' = y_2 + rt] \\ &\vee [l = l' = 2 \wedge x_1' = x_1 \wedge y_1' = y_1 + rt \wedge x_2' = x_2 \wedge y_2' = y_2 + rt] \end{aligned}$$

Can these constraints be satisfied simultaneously?

# Can the vehicles collide with 0 discrete transitions?

$$l^1 = 1 \wedge x_1^1 = -5 \wedge y_1^1 = 0 \wedge x_2^1 = 0 \wedge y_2^1 = -4$$

$$[l^1 = l'^1 = 1 \wedge x_1'^1 = x_1^1 + t \wedge y_1'^1 = y_1^1 \wedge x_2'^1 = x_2^1 \wedge y_2'^1 = y_2^1 + t]$$

$$x_1'^1 = x_2'^1 \wedge y_1'^1 = y_2'^1$$

$$-5 + t = x_1^1 + t = x_1'^1 = x_2'^1 = x_2^1 = 0 \Rightarrow t = 5$$

Vehicle 1 reaches the intersection at  $t = 5$

Vehicle 2 reaches the intersection at  $t = 4$

The two vehicles do not collide!

If Vehicle 2 starts at  $(-5, 0)$ , it will collide with Vehicle 1

# Can the vehicles collide with 0 discrete transitions?

$$l^1 = 1 \wedge x_1^1 = -5 \wedge y_1^1 = 0 \wedge x_2^1 = 0 \wedge y_2^1 = -4$$

$$[l^1 = l'^1 = 1 \wedge x_1'^1 = x_1^1 + t \wedge y_1'^1 = y_1^1 \wedge x_2'^1 = x_2^1 \wedge y_2'^1 = y_2^1 + t]$$

$$x_1'^1 = x_2'^1 \wedge y_1'^1 = y_2'^1$$

$$-5 + t = x_1^1 + t = x_1'^1 = x_2'^1 = x_2^1 = 0 \Rightarrow t = 5$$

Vehicle 1 reaches the intersection at  $t = 5$

Vehicle 2 reaches the intersection at  $t = 4$

If Vehicles are not points, we say they collide if they are, say, within distance 1 of each other. In this case, the two vehicles collide.

# Can the vehicles collide with 1 discrete transition?

$$l^1 = 1 \wedge x_1^1 = -5 \wedge y_1^1 = 0 \wedge x_2^1 = 0 \wedge y_2^1 = -4$$

$$[l^1 = l'^1 = 1 \wedge x_1'^1 = x_1^1 + t \wedge y_1'^1 = y_1^1 \wedge x_2'^1 = x_2^1 \wedge y_2'^1 = y_2^1 + t]$$

Initial state and first continuous transition

First discrete transition

$$l'^1 = 1 \wedge l^2 = 2 \wedge x_1^2 = 0 \wedge x_1^2 = x_1'^1 \wedge y_1^2 = y_1'^1 \wedge x_2^2 = x_2'^1 \wedge y_2^2 = y_2'^1$$

Second continuous transition

$$[l^2 = l'^2 = 1 \wedge x_1'^2 = x_1^2 \wedge y_1'^2 = y_1^2 + t \wedge x_2'^2 = x_2^2 \wedge y_2'^2 = y_2^2 + t]$$

Collision if they are within distance 1

$$-1 \leq x_1'^2 - x_2'^2 \leq 1 \wedge -1 \leq y_1'^2 - y_2'^2 \leq 1$$

# Satisfiability Module Theory Solvers

---

- ❖ The constraints can be solved using SMT solvers
- ❖ Input is a quantifier free first order logic formulas
- ❖ Check if there exists an assignments for the variables that satisfies the formula
- ❖ The formula essentially consists of constraints (linear arithmetic, non-linear arithmetic) that are combined using boolean operators

# SMT solving examples

---

$x \geq 0 \wedge x < 1$  in the theory  $(\mathbb{R}, \geq)$

Linear real arithmetic formula, satisfiable

$x > 0 \wedge x < 1$  in the theory  $(\mathbb{Z}, \geq)$

Linear integer arithmetic formula, unsatisfiable

$x > 2 \vee (x + 2y \leq 2)$  in the theory  $(\mathbb{R}, \geq, +)$

Linear real arithmetic formula, satisfiable

$x + y \cdot t < 3 \wedge t \geq 0$  in the theory  $(\mathbb{R}, \geq, +, \cdot)$

Non-linear real arithmetic formula, satisfiable

# SMT solvers

---

- ❖ Linear arithmetic - Z3 ([rise4fun.com](http://rise4fun.com)), Yices
- ❖ Non-linear arithmetic - iSAT, MiniSmt

# Z3 SMT

---

$x > 0 \wedge x < 1$  in the theory  $(\mathbb{R}, \geq)$

```
(declare-fun x () Real)
(assert (> x 0))
(assert (< x 1))
(check-sat)
(get-model)

sat
(model
  (define-fun x () Real
    (/ 1.0 2.0))
)
```

$x > 0 \wedge x < 1$  in the theory  $(\mathbb{Z}, \geq)$

```
(declare-fun x () Int)
(assert (> x 0))
(assert (< x 1))
(check-sat)
(get-model)

unsat
Z3(5, 10): ERROR: model is not available
```



---

# Bounded Safety Analysis: Approximation

# Two vehicles at an intersection


$(x_1(t), y_1(t))$ : vehicle 1 position at time  $t$

**Vehicle 1 dynamics moving east**

$$\begin{aligned}\dot{x}_1 &= r \\ \dot{y}_1 &= 0\end{aligned}$$

**Vehicle 1 dynamics moving north**

$$\begin{aligned}\dot{x}_1 &= 0 \\ \dot{y}_1 &= r\end{aligned}$$


$(x_{01}, y_{01})$  

$(0, 0)$

**Vehicle 1 dynamics  
at the intersection?**

**Vehicle 2 dynamics moving north**

$$\begin{aligned}\dot{x}_2 &= 0 \\ \dot{y}_2 &= r\end{aligned}$$

$(x_{02}, y_{02})$  

$(x_2(t), y_2(t))$ : vehicle 2 position at time  $t$

# Dubin's car dynamics

---

$(x(t), y(t))$ : Position of the Vehicle at time  $t$

$(v^x(t), v^y(t))$ : Velocity of the Vehicle at time  $t$

$$\dot{x} = v^x$$

$$\dot{y} = v^y$$

$$\dot{v}^x = -\omega v^y$$

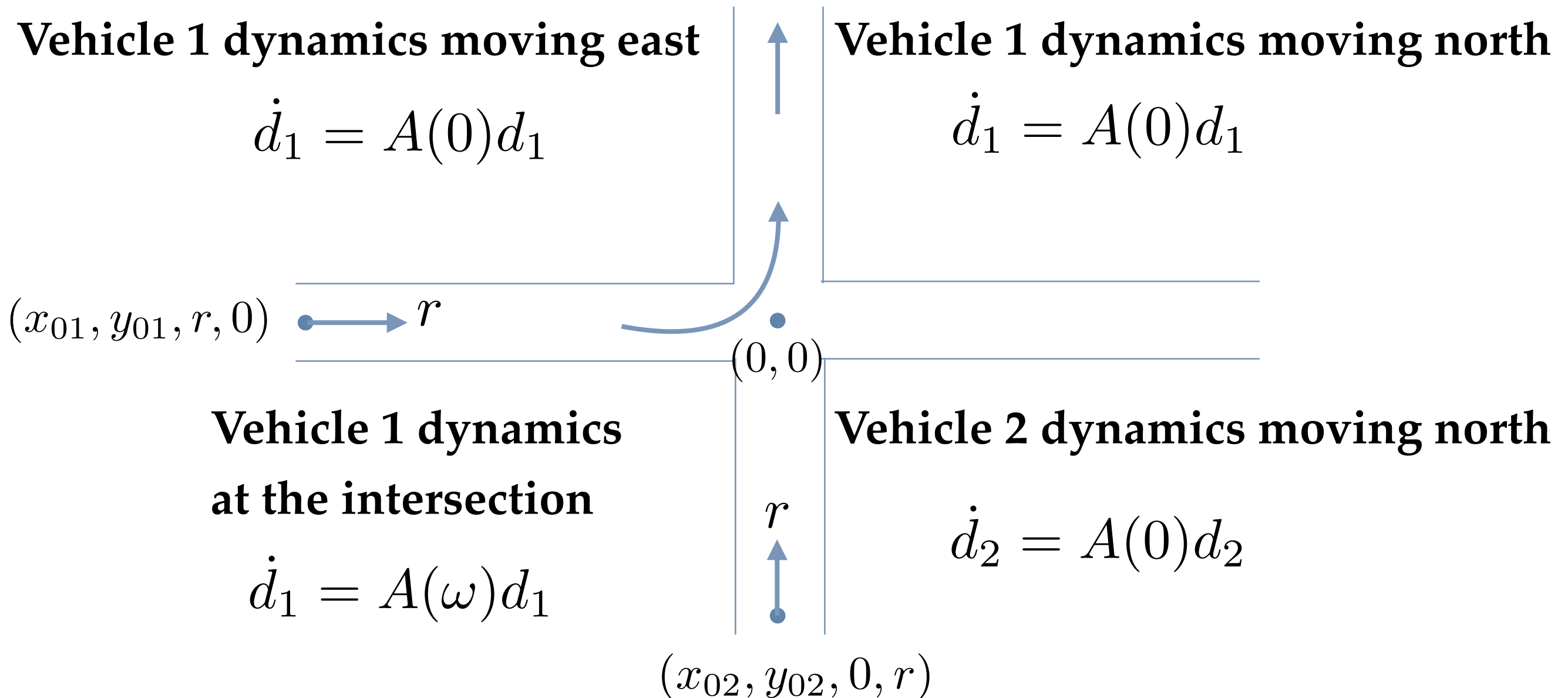
$$\dot{v}^y = \omega v^x$$

Represent succinctly as  $\dot{d} = A(\omega)d$ , where  $d = (x, y, v^x, v^y)$  and

$$A(\omega) = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\omega \\ 0 & 0 & \omega & 0 \end{pmatrix}$$

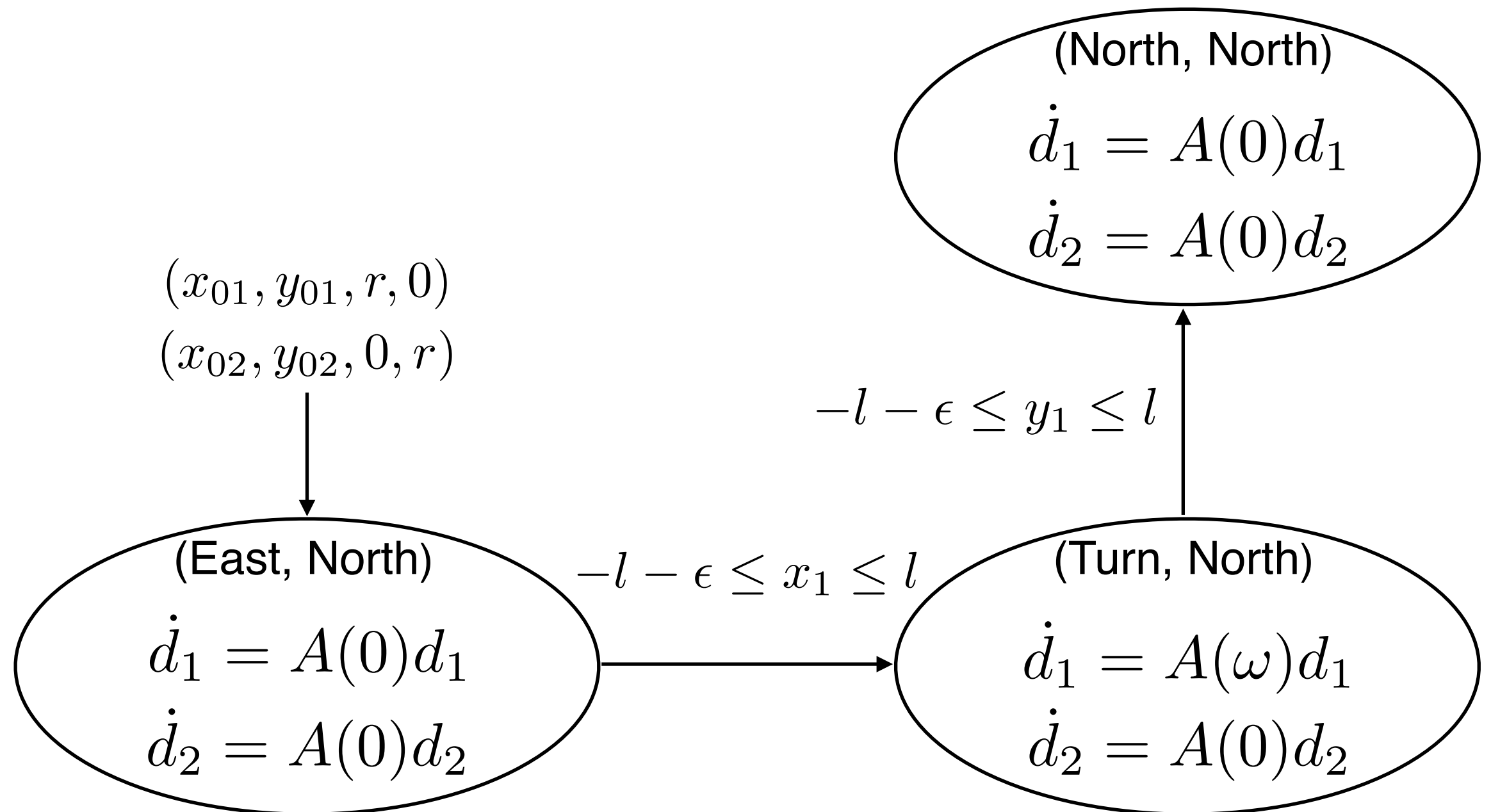
# Two vehicles at an intersection

$d_1(t), d_2(t)$ : the state of vehicle 1 and 2, respectively at time  $t$



# Two vehicles at an intersection

$d_1(t)$ ,  $d_2(t)$ : the state of vehicle 1 and 2, respectively at time  $t$



# Bounded Safety Analysis

---

- ❖ Broad approach — Encode executions as SMT formulas, solve the formulas for satisfiability
- ❖ Satisfiability of SMT formulas is decidable only when the constraints are in a certain theory — linear, nonlinear (polynomial constraints)
- ❖ However, continuous transitions of complex dynamics cannot be encoded in these theories.

# Linear dynamical systems

---

Linear Dynamical System  $\dot{x}(t) = ax(t)$

Closed form solution  $x(t) = e^{at}x(0)$

$$\frac{d}{dt}x(t) = ae^{at}x(0) = ax(t)$$

Linear Dynamical System

$$\dot{\bar{x}}(t) = A\bar{x}(t), \bar{x}_0 \in X \subseteq \mathbb{R}^n$$

Closed form solution

$$\bar{x}(t) = e^{At}\bar{x}(0)$$

$$e^y = 1 + y + \frac{y^2}{2!} + \frac{y^3}{3!} + \dots$$

$$e^B = 1 + B + \frac{B^2}{2!} + \frac{B^3}{3!} + \dots$$

# Continuous transitions

---

Linear Dynamical System  $\dot{x}(t) = ax(t)$

Closed form solution  $x(t) = e^{at}x(0)$

$x_1 \xrightarrow{t} x_2$  iff  $x_2 = e^{at}x_1$  Not a polynomial constraint

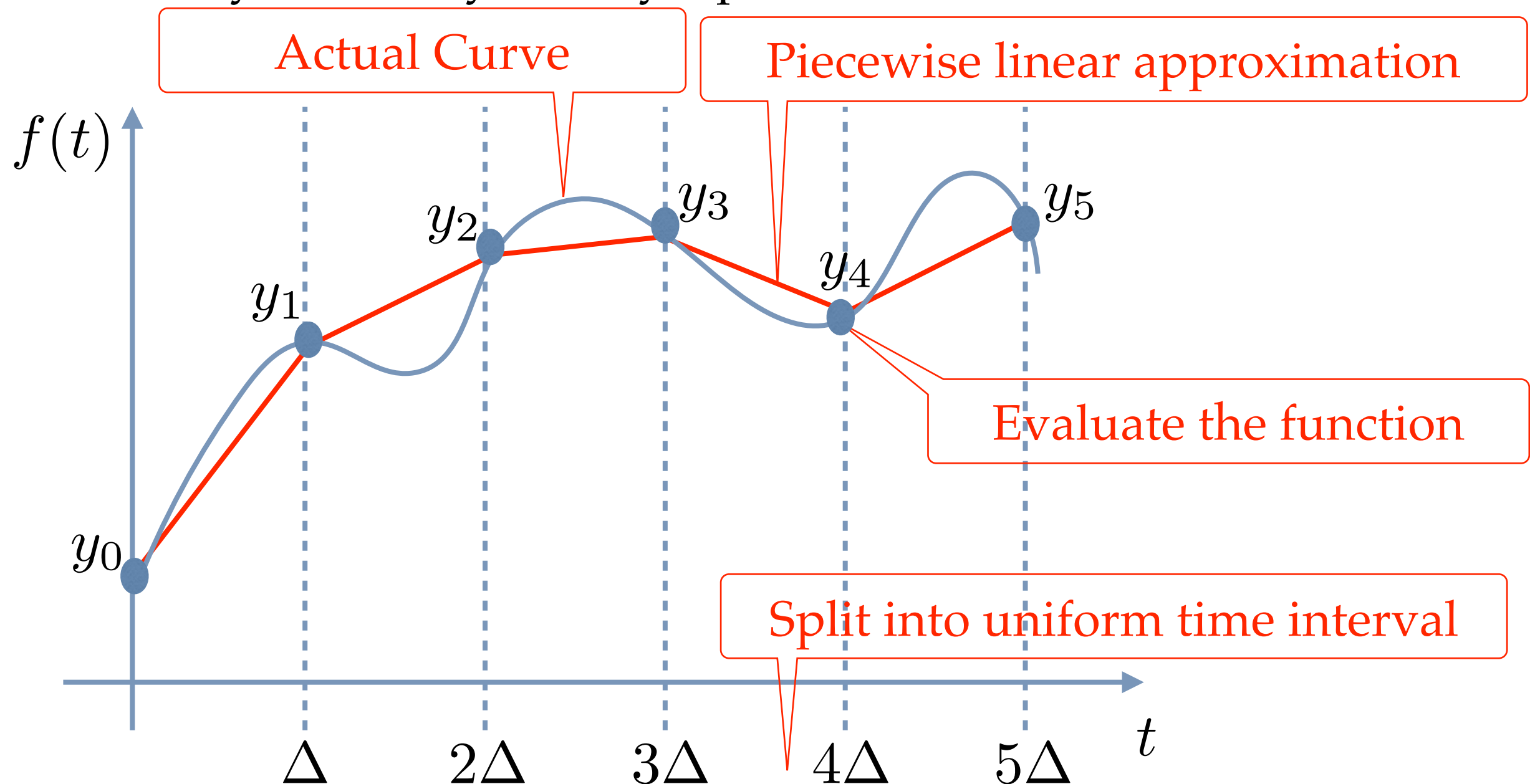
The decidability of the theory of reals with exponential functions is an open problem

We cannot directly encode the continuous transition, we will approximate!



# Sampling based approximation

- Approximate the exponential function that arises as the solution of a linear dynamical system by a piecewise linear curve



# Sampling based approximation

$$\varphi_C(x_1, t, x_2) := x_2 = e^{at} x_1$$

Formula for the actual continuous transition

Let  $y_i = e^{i\Delta} x_1$  be the  $i$ -th sample point

Computed function values at sample point

Formula for the approximate continuous transition

First piece

$$\hat{\varphi}_C(x_1, t, x_2) := [0 \leq t \leq \Delta \implies x_2 = y_0 + \frac{y_1 - y_0}{\Delta} t]$$

Second piece

$$\wedge [\Delta \leq t \leq 2\Delta \implies x_2 = y_1 + \frac{y_2 - y_1}{\Delta} t]$$

$$\vdots$$

$$\wedge [(k-1)\Delta \leq t \leq k\Delta \implies x_2 = y_{k-1} + \frac{y_k - y_{k-1}}{\Delta} t]$$

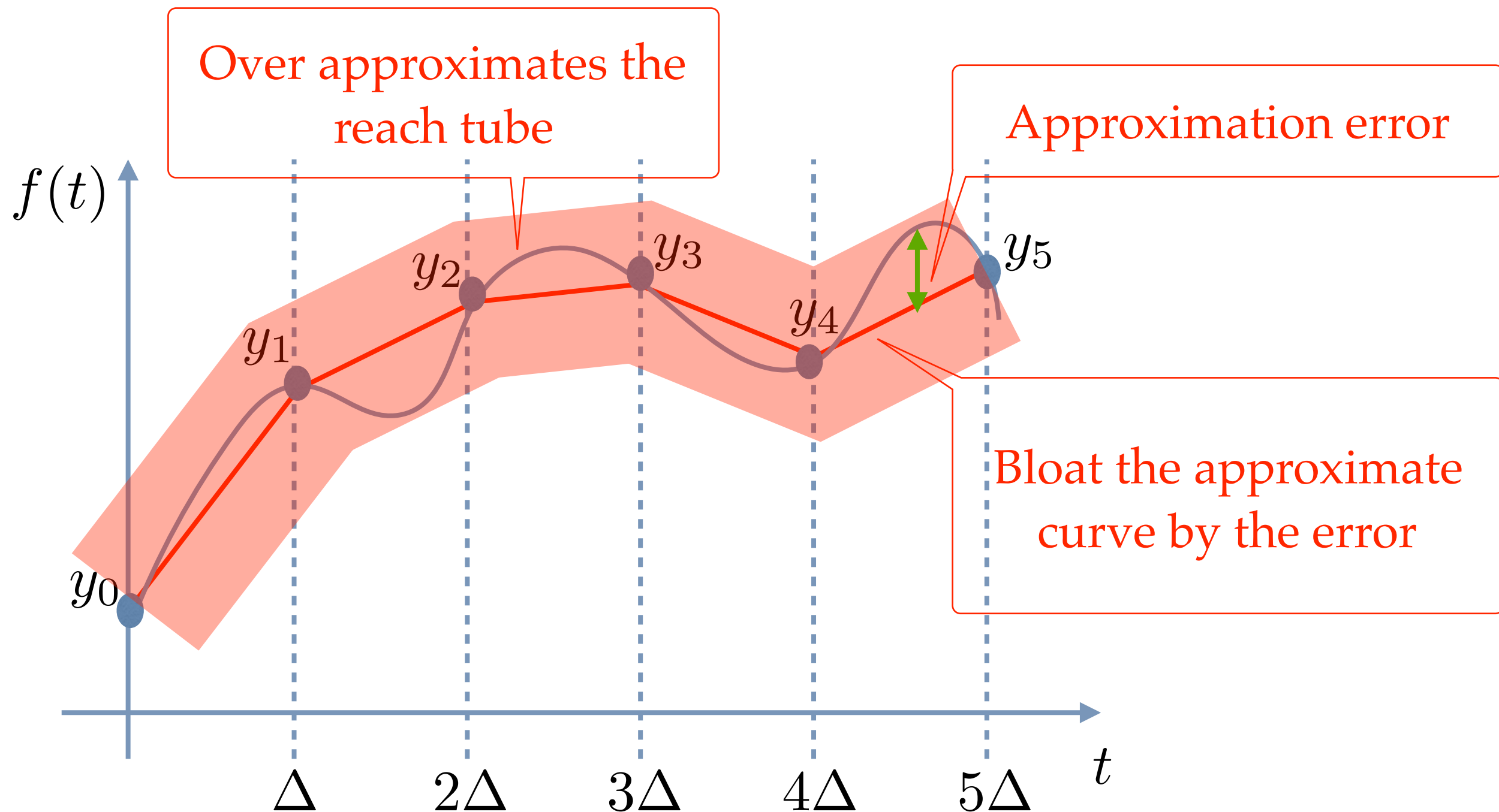
# Sampling based approximation

---

$$\begin{aligned}\hat{\varphi}_C(x_1, t, x_2) := & [0 \leq t \leq \Delta \implies x_2 = y_0 + \frac{y_1 - y_0}{\Delta} t] \\ & \wedge [\Delta \leq t \leq 2\Delta \implies x_2 = y_1 + \frac{y_2 - y_1}{\Delta} t] \\ & \vdots \\ & \wedge [(k-1)\Delta \leq t \leq k\Delta \implies x_2 = y_{k-1} + \frac{y_k - y_{k-1}}{\Delta} t]\end{aligned}$$

Note that all constraints are linear (t is multiplied by a constant)

# Bounded error approximation



# Sampling based approximation

---

$$\begin{aligned}\hat{\varphi}_C(x_1, t, x_2) &:= [0 \leq t \leq \Delta \implies x_2 = y_0 + \frac{y_1 - y_0}{\Delta} t] \\ &\quad \wedge [\Delta \leq t \leq 2\Delta \implies x_2 = y_1 + \frac{y_2 - y_1}{\Delta} t] \\ &\quad \vdots \\ &\quad \wedge [(k-1)\Delta \leq t \leq k\Delta \implies x_2 = y_{k-1} + \frac{y_k - y_{k-1}}{\Delta} t]\end{aligned}$$

Let  $\epsilon$  be an error bound

$$\hat{\varphi}_C^\epsilon(x, t, x') := \hat{\varphi}_C(x, t, x'') \wedge -\epsilon \leq x'' - x' \leq \epsilon$$

Bloated reach tube

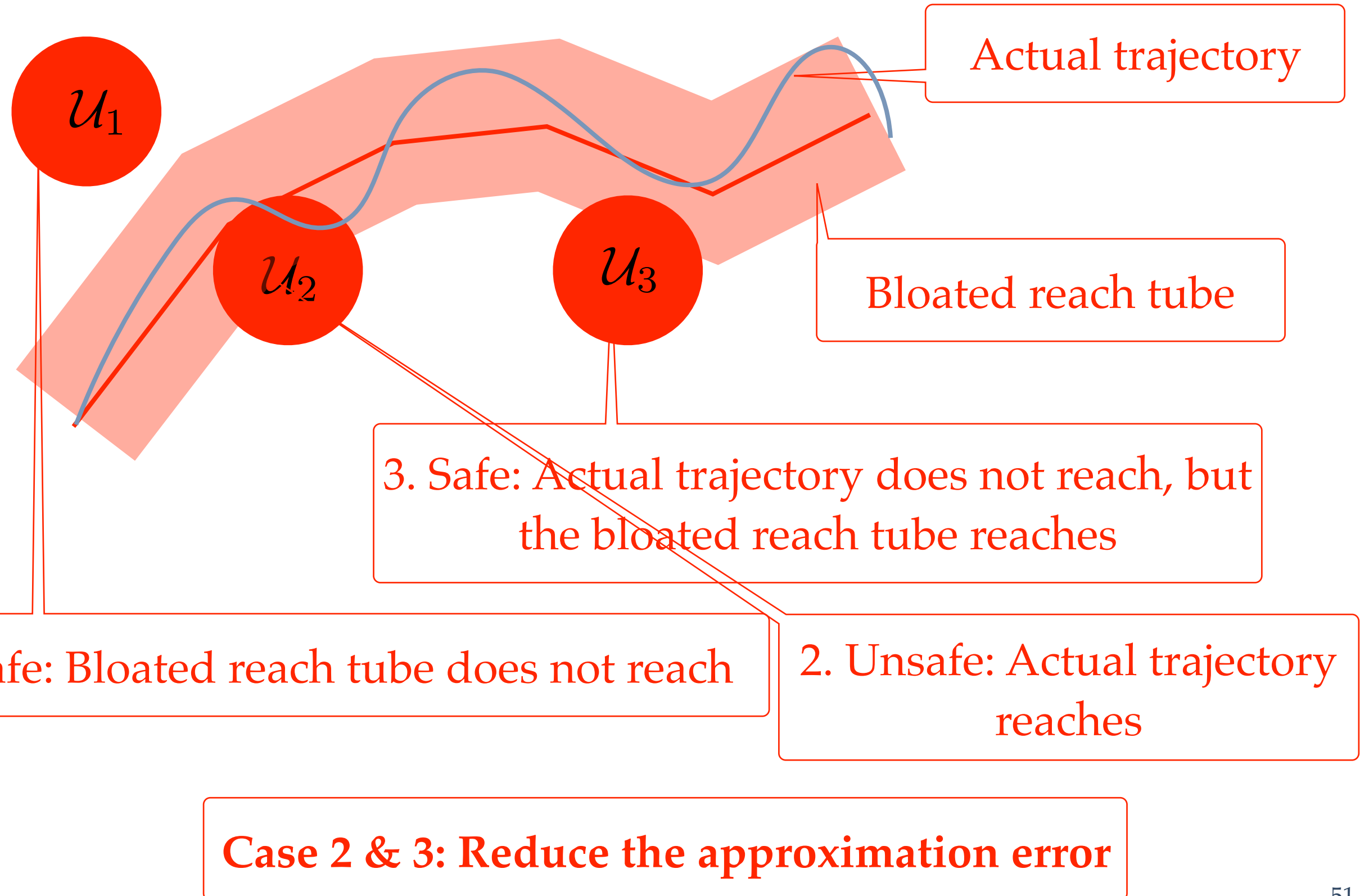
All points close to the points in the  
approximate trajectory

# Bounded Safety Analysis using Bloated Reach Tubes

---

- ❖ Note that if the bloated reach tube does not intersect an unsafe set, then the original trajectory also does not intersect the unsafe set
- ❖ If the bloated reach tube intersects the unsafe set, then
  - ❖ either the actual trajectory reaches the unsafe set, or
  - ❖ the precision of approximation is too coarse

# Bounded Safety Analysis using Bloated Reach Tubes



# Bounded Safety Analysis using Bloated Reach Tubes

---

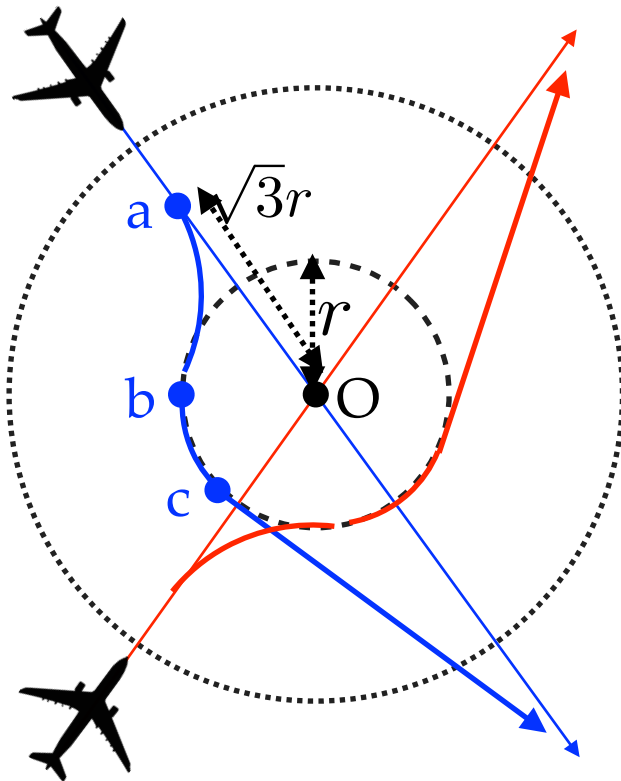
- ❖ Note that if the bloated reach tube does not intersect an unsafe set, then the original trajectory also does not intersect the unsafe set
- ❖ If the bloated reach tube intersects the unsafe set, then
  - ❖ either the actual trajectory reaches the unsafe set, or
  - ❖ the precision of approximation is too coarse
- ❖ We iteratively reduce the approximation error
- ❖ If the bloated reach tube does not intersect the unsafe set, we can conclude safety
- ❖ However, we will not be able to conclude that the system is unsafe
- ❖ Need to under-approximate (very hard in general)



---

# Bounded Safety Analysis: Approximation based Approach Illustration

# Air traffic collision avoidance protocol



$\mathbf{x} = (x_1, x_2)$ : position of the airplane

$\mathbf{d} = (d_1, d_2)$ : velocity of the airplane

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{d}_1 \\ \dot{d}_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\omega \\ 0 & 0 & \omega & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ d_1 \\ d_2 \end{bmatrix}$$

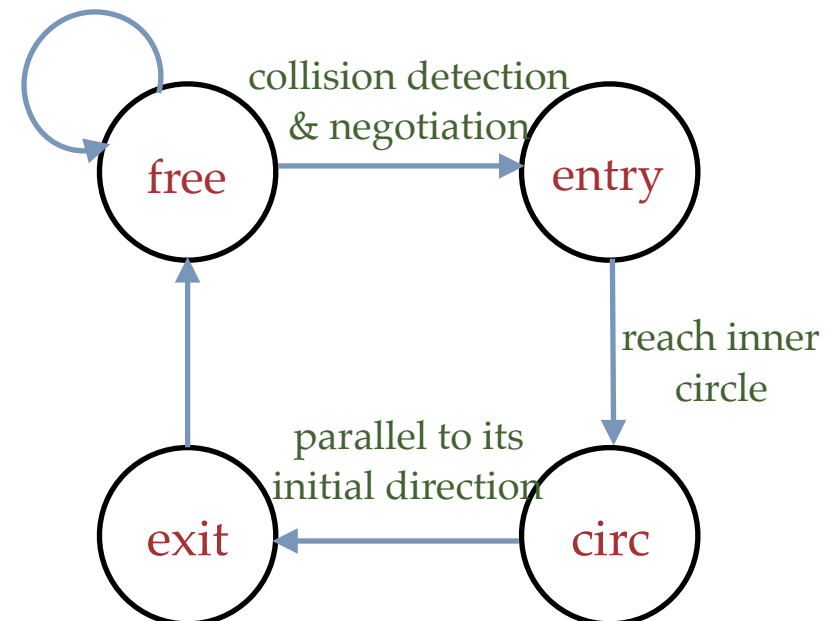
$\omega$ : the angular velocity

## Minimum separation

*The aircraft maintain a minimum distance between them always*

$$\begin{aligned} \|x - y\| &\leq p & c &= x + \lambda d = y + \lambda e \\ \|x - c\| &= \sqrt{3}r & (r\omega)^2 &= \|d\|^2 & x^0 &:= x, d^0 := d \end{aligned}$$

$$\omega := *$$



$$\begin{aligned} \|x - c\| &\leq r \\ \omega &:= -\omega \end{aligned}$$

$$\omega := 0 \quad x + \lambda_2 d = x^0 + \lambda_1 d^0$$

# Parameterized linear systems

## Parameterized linear system

$$\begin{aligned}\dot{x} &= Ax \\ x_0 &\in X_0, t \in [0, T] \\ A &\in \Omega\end{aligned}$$

## Related work:

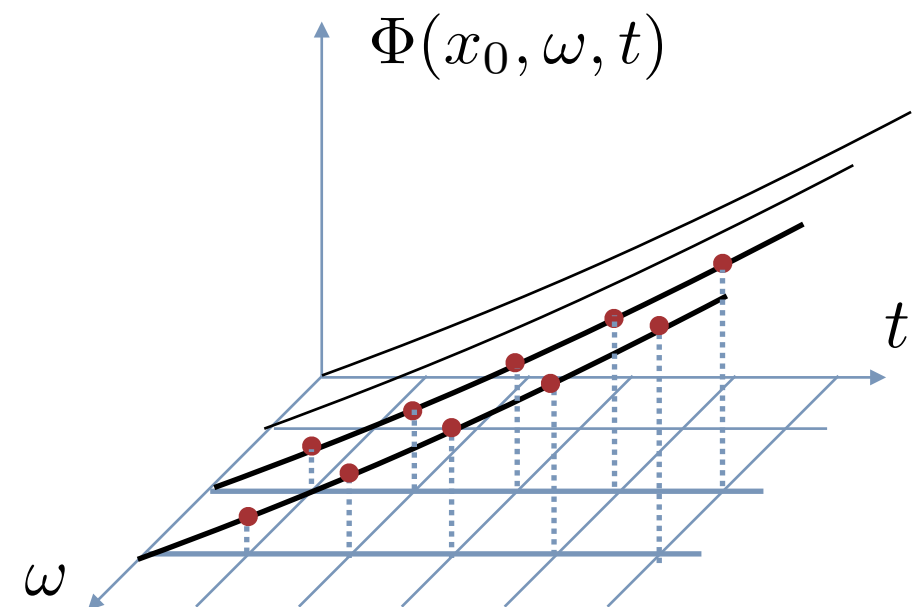
Approximate the state transition matrices [Althoff et al]:

$$\mathcal{M}(\delta) = \{e^{A\delta} \mid A \in \Omega\}$$

Not straightforward to compute the sampling interval for a given error tolerance

## Main idea:

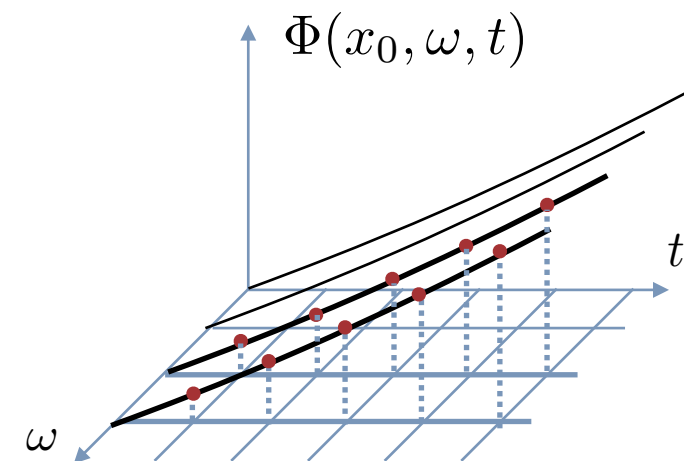
- ❖ Sample both the parameter space and the time domain
- ❖ Construct a piecewise bilinear function interpolating the values at the sample points



# Parameterized linear systems

## Main idea:

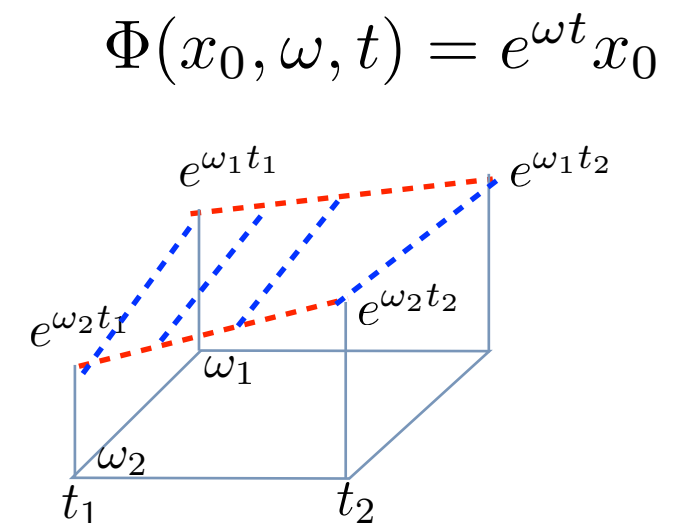
- ❖ Sample both the parameter space and the time domain
- ❖ Construct a piecewise bilinear function interpolating the values at the sample points



For  $\omega \in [\omega_1, \omega_2]$  and  $t \in [t_1, t_2]$ ,

$$\hat{\Phi}(x_0, \omega, t) = [\beta \{ \alpha e^{\omega_1 t_1} + (1 - \alpha) e^{\omega_1 t_2} \} + (1 - \beta) \{ \alpha e^{\omega_2 t_1} + (1 - \alpha) e^{\omega_2 t_2} \}] x_0$$

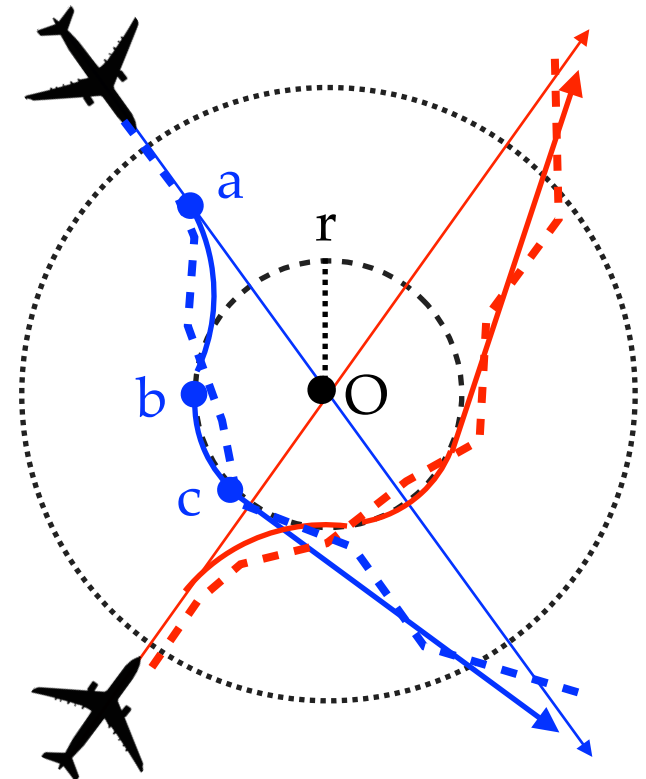
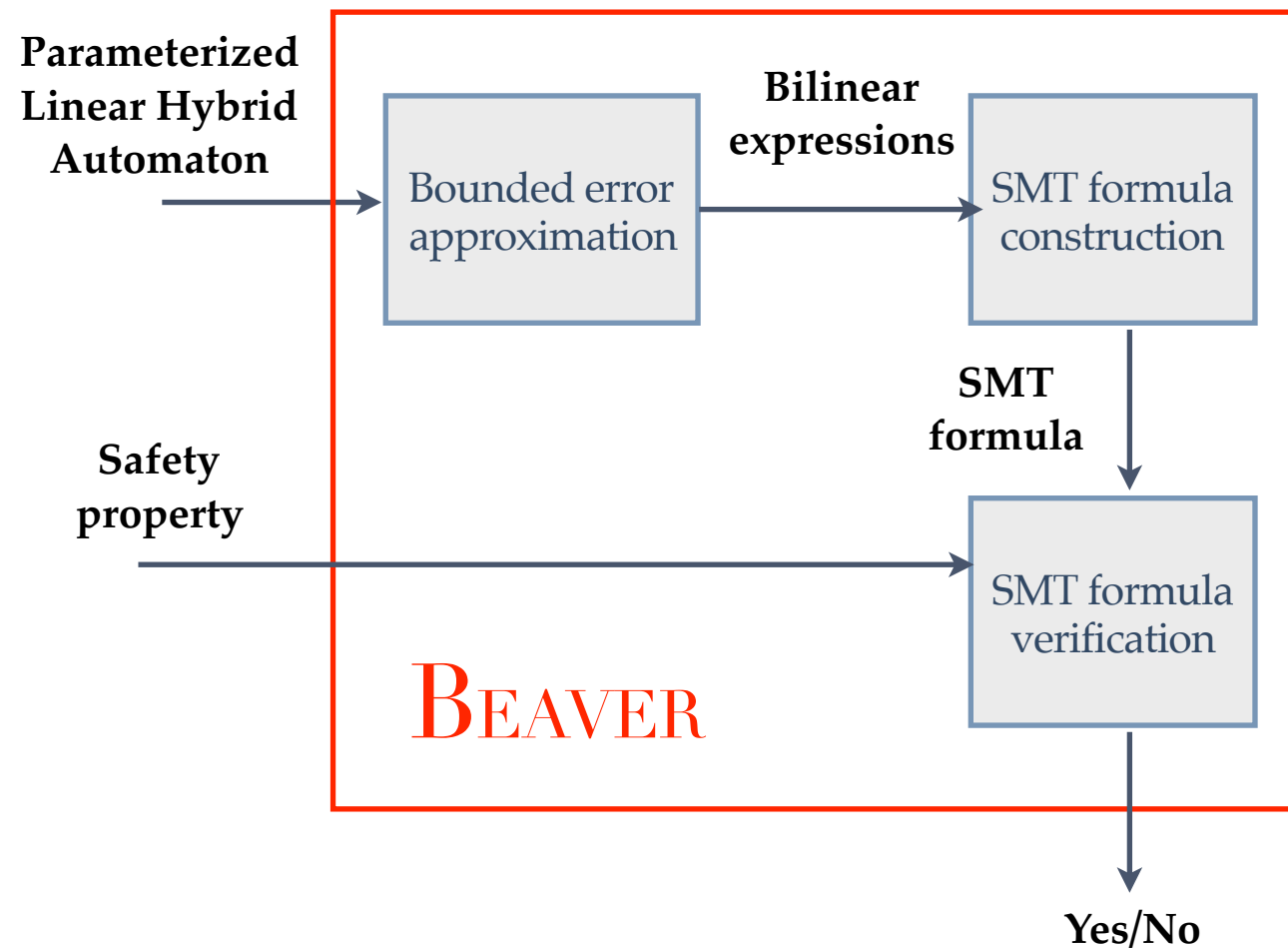
where  $\alpha = \frac{t - t_2}{t_1 - t_2}$  and  $\beta = \frac{\omega - \omega_1}{\omega_1 - \omega_2}$



## Bound the precision of approximation:

- ❖ Finding the  $\delta$  corresponding to an  $\epsilon$   $\max\{\delta \|\Omega\| e^{\delta \|\Omega\|}, \delta T e^{\delta T}\} \leq \frac{\epsilon}{4e^{\|\Omega\|T}}$

# BEAVER: Bounded Error Approximation based VERification



$$\varphi_{exec}^{i,\epsilon}(\mathbf{x}_i, t_i) = \varphi_{free}^{i,\epsilon} \wedge \varphi_{entry}^{i,\epsilon} \wedge \varphi_{circ}^{i,\epsilon} \wedge \varphi_{exit}^{i,\epsilon}$$

$$\varphi_{safe}^{\epsilon} = \neg \exists t [\varphi_{exec}^{1,\epsilon}(\mathbf{x}_1, t) \wedge \varphi_{exec}^{2,\epsilon}(\mathbf{x}_2, t) \wedge \|\mathbf{x}_1 - \mathbf{x}_2\| \leq d_{sep} + 2\epsilon]$$

**Main highlight of BEAVER — can perform compositional verification**

# Analysis results

| # Aircraft | epsilon | Size of formula<br>(Chars) ( $10^{+6}$ ) | Time Create<br>SMT( $10^{-1}$ ) | Time Verify (in<br>sec) ( $10^{+1}$ ) | Total Time (in<br>sec) ( $10^{+1}$ ) | SMT result |
|------------|---------|--|---------------------------------|---------------------------------------|--------------------------------------|------------|
| 2          | 2       | 0.072                                    | 2.084                           | 0.035                                 | 0.056                                | Sat        |
| 2          | 1       | 2.09                                     | 5.066                           | 0.301                                 | 0.351                                | Unsat      |
| 4          | 2       | 1.44                                     | 3.729                           | 0.152                                 | 0.189                                | Sat        |
| 4          | 1       | 3.37                                     | 8.514                           | 1.280                                 | 1.360                                | Unsat      |
| 6          | 2       | 1.81                                     | 4.764                           | 0.384                                 | 0.431                                | Sat        |
| 6          | 1       | 3.92                                     | 9.731                           | 4.310                                 | 4.410                                | Unsat      |
| 8          | 2       | 2.55                                     | 6.646                           | 2.850                                 | 2.920                                | Sat        |
| 8          | 1       | 5.21                                     | 14.74                           | 29.50                                 | 30.00                                | Unsat      |

Safety analysis of aircraft collision avoidance protocol for  $p=1$ ,  $T=0.2$

- ❖ We start from some reasonable value of error, and we reduce it gradually until we get safety
- ❖ The size of formula increases slowly as we increase the number of aircraft
- ❖ Total time for safety analysis grows slowly as we increase the number of aircraft

---

# Overview of other safety analysis techniques

# Safety Analysis

---

- ❖ So far, we saw bounded safety analysis using bounded error approximation
- ❖ How about unbounded safety?
- ❖ Two broad techniques based on state-space exploration
  - ❖ Symbolic reach set computation
  - ❖ Abstractions



# State-space exploration

---

- ❖ Start with the initial set of states
- ❖ Iteratively compute the set of states reached by traversing a discrete or a continuous transition
- ❖ Until a fix point is reached

Discrete Post Operator

$$DPost(S) = \{s' \mid s \in S, e \in E, s \xrightarrow{e} s'\}$$

Continuous Post Operator

$$CPost(S) = \{s' \mid s \in S, t \in \mathbb{R}_{\geq 0}, s \xrightarrow{t} s'\}$$

# Reach Set Computation

---

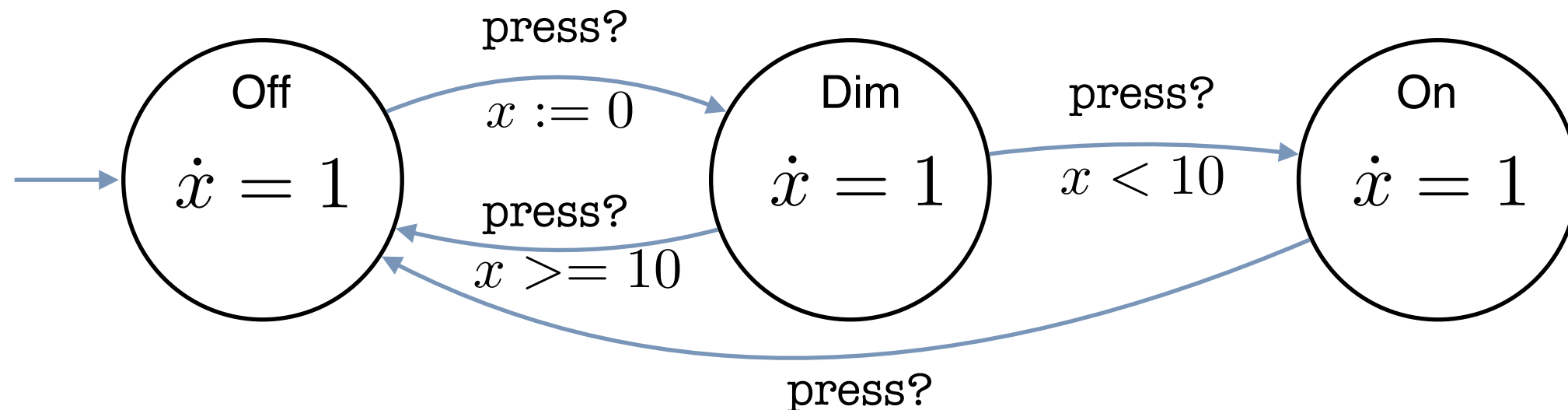
Compute  $Reach(\mathcal{H}, I)$

1. Let  $R_0 := I$
2. Compute  $R_{i+1} := R_i \cup DPost(CPost(R_i))$
3. Stop when  $R_{i+1} = R_i$

The reach sets computed are infinite sets,  
need efficient representation

The shape of the reach set and the appropriate  
representation depends on dynamics

# Illustration of symbolic computation



- ❖ For constant dynamics, we can use polyhedral set
- ❖ In one dimension, a polyhedral set is an interval

$$(Dim, 0) = \{(Dim, 0)\}$$

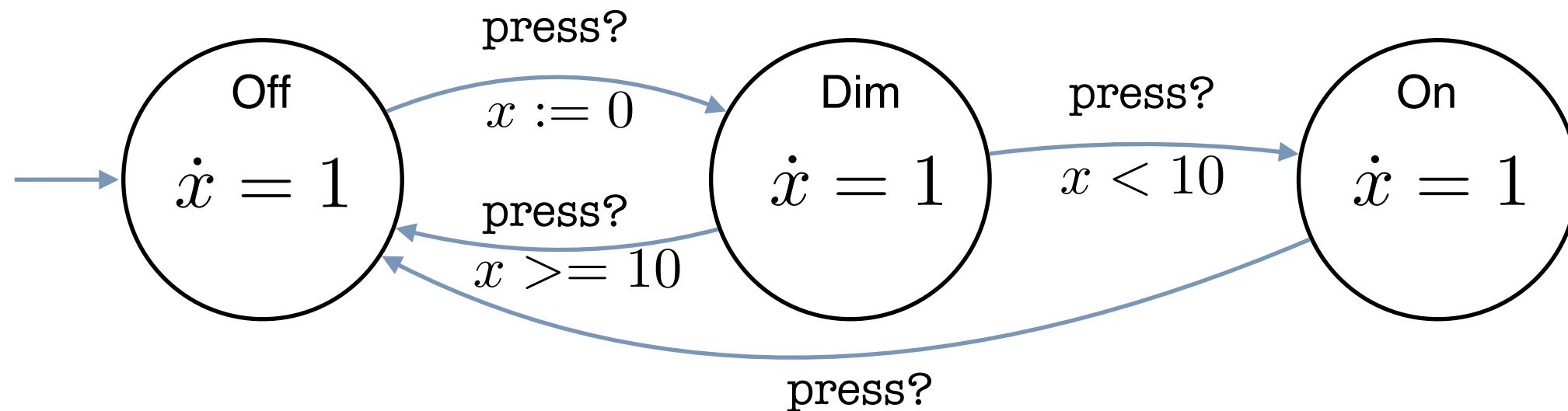
$$(Off, \mathbb{R}_{\geq 0}) = \{(Off, x) \mid x \in \mathbb{R}_{\geq 0}\}$$

$$(On, \{x < 10\}) = \{(On, x) \mid x < 10\}$$

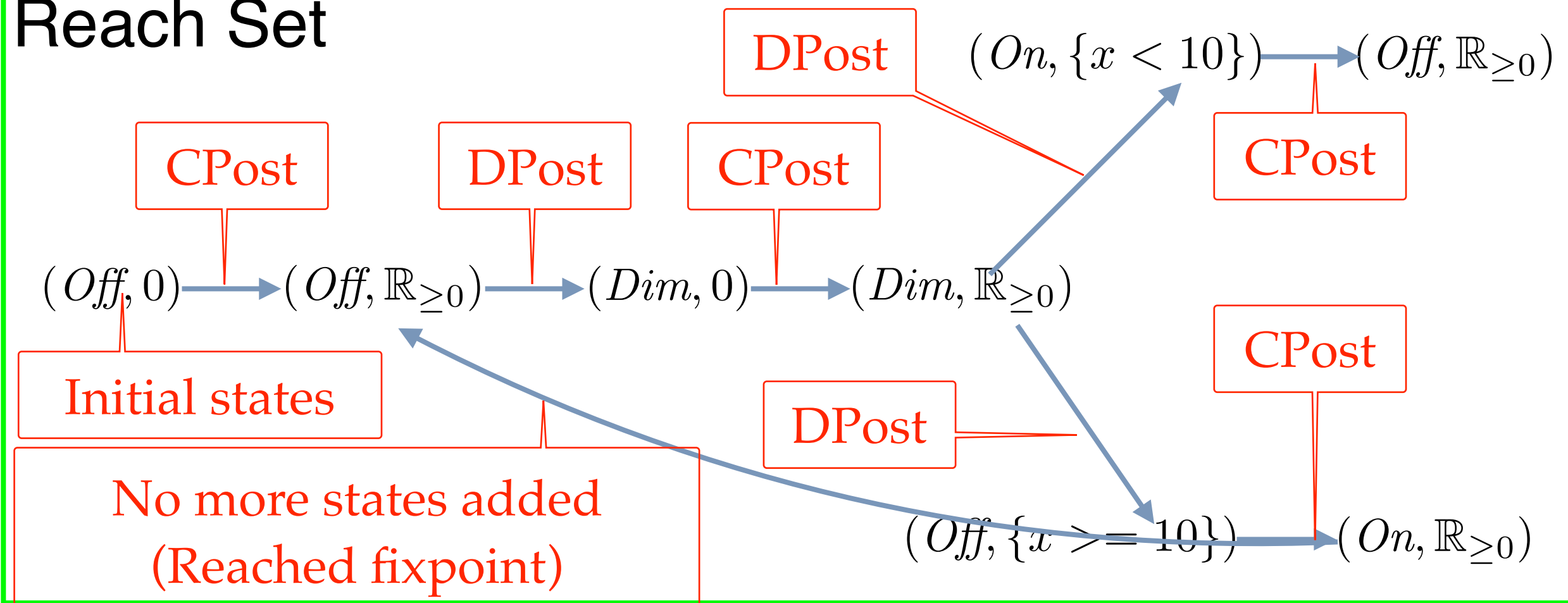
Symbolic state

Concrete state

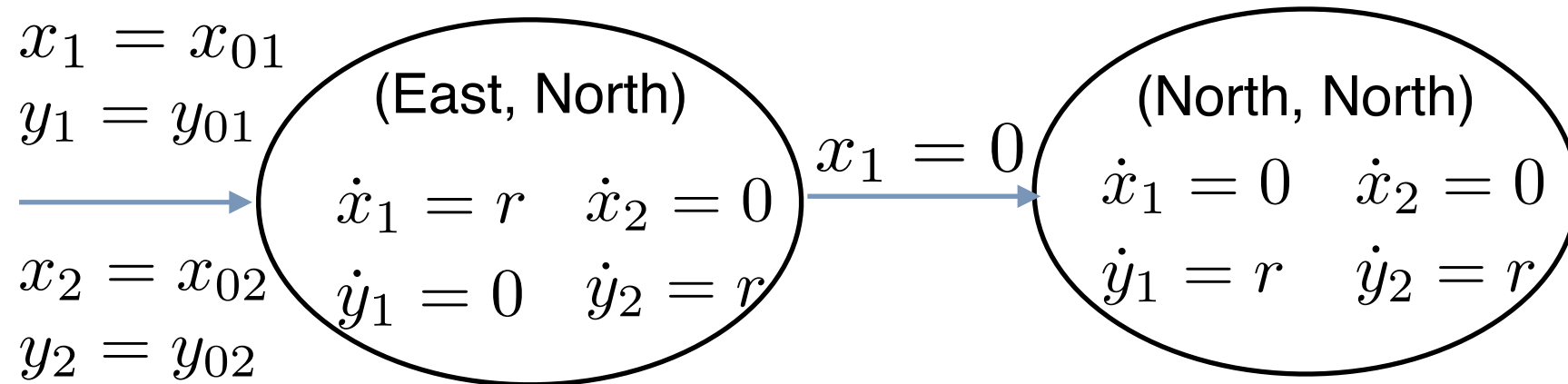
# Illustration of symbolic computation



## Reach Set



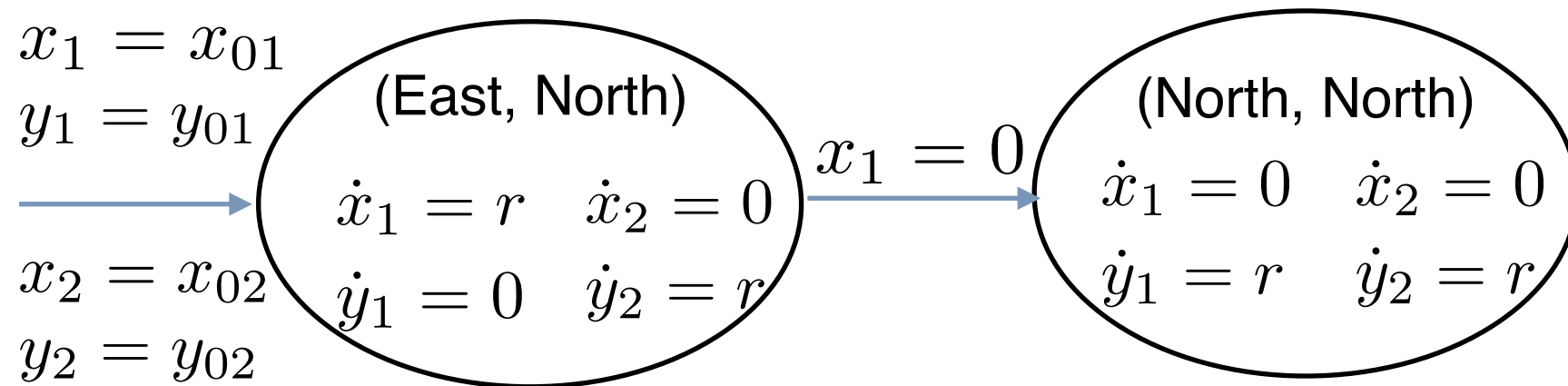
# Illustration of symbolic computation



$$\{(-5, 0)\} \xrightarrow{CPost} \{(x, 0) \mid x \in (-5, \infty)\} \xrightarrow{DPost} \{(0, 0)\} \xrightarrow{CPost} \{(0, y) \mid y \in (0, \infty)\}$$

- ❖ Each of the intermediate set of states can be represented as a polyhedron and the CPost and DPost as operations on polyhedra.
  - ❖ For instance, DPost operation above corresponds to intersection with the y-axis (a polyhedron)

# Illustration of symbolic computation



$$\{(-5, 0)\} \xrightarrow{CP_{ost}} \{(x, 0) \mid x \in (-5, \infty)\} \xrightarrow{DP_{ost}} \{(0, 0)\} \xrightarrow{CP_{ost}} \{(0, y) \mid y \in (0, \infty)\}$$

- ✧ Alternately, we can represent them as SMT formula and CPost and DPost operations would correspond to quantifier elimination

$$\begin{aligned} x = -5 &\xrightarrow{CP_{ost}} \exists t, x, t \geq 0, x \geq -5, x' = x + 2t \\ &\equiv x' \geq -5 \end{aligned}$$

# Challenges in symbolic exploration

---

- ❖ Symbolic computation relies on being able to represent the sets obtained by CPost and DPost
- ❖ CPost can be a complex set, e.g., CPost for linear dynamics systems requires exponential functions
- ❖ Again, we need to approximate the reach set by data structures for which operations such as intersection and emptiness checking are computationally possible

Challenge 1: Efficient data structures for representing and manipulating the intermediate reach sets or precise over approximations

# Related work

---

- ❖ Complexity of verification is affected by the number of sample points and the data structures used to represent the reach sets
- ❖ Data structure investigated — Polyhedra [Dang,Maler], [Chutinan, Krogh], Ellipsoids [Kurzhanski, Varaiya], Zonotopes, Support functions [Girard, Guernic]
- ❖ Previous work: a dynamic algorithm which samples non-uniformly, and provides an approximation with orders of magnitude smaller number of sample points, and takes orders of magnitude smaller time [Prabhakar, Viswanathan]



# Challenges in symbolic exploration

---

Challenge 2: In practice, a straight forward state space exploration does not ensure fixpoint

- ❖ Alternate technique: Use abstractions to simplify the system, so that state space exploration terminates

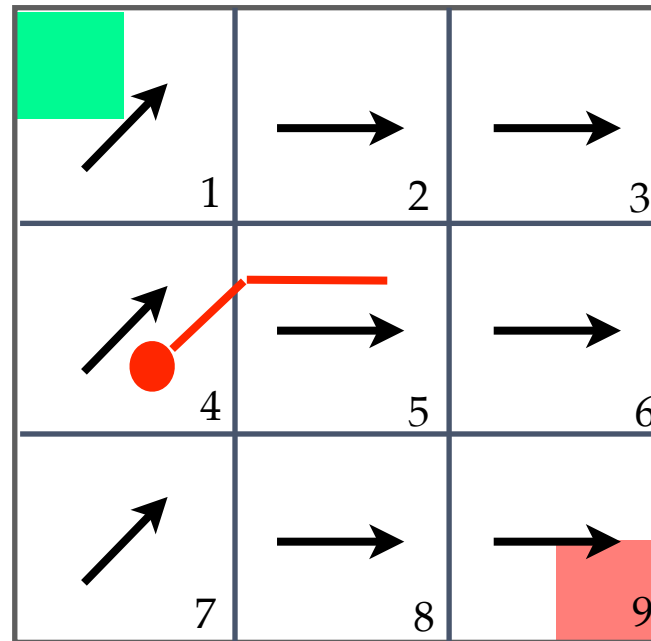
# Predicate Abstraction

---

- ❖ Construct a finite abstract system from a given concrete hybrid automaton such that if the abstract system is safe, we can conclude that the hybrid automaton is safe
- ❖ The safety verification of the finite abstract system is efficient
- ❖ However, finite abstract system does not provide a bound on the error of approximation
- ❖ Hence, abstractions are often coupled with a refinement loop to assist the safety proof search

# Robot Navigation Protocol

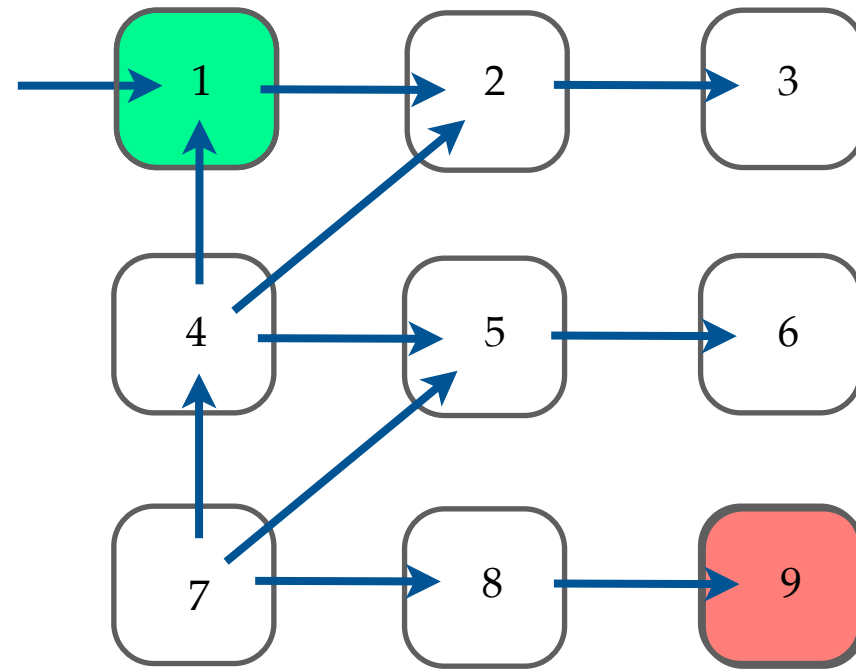
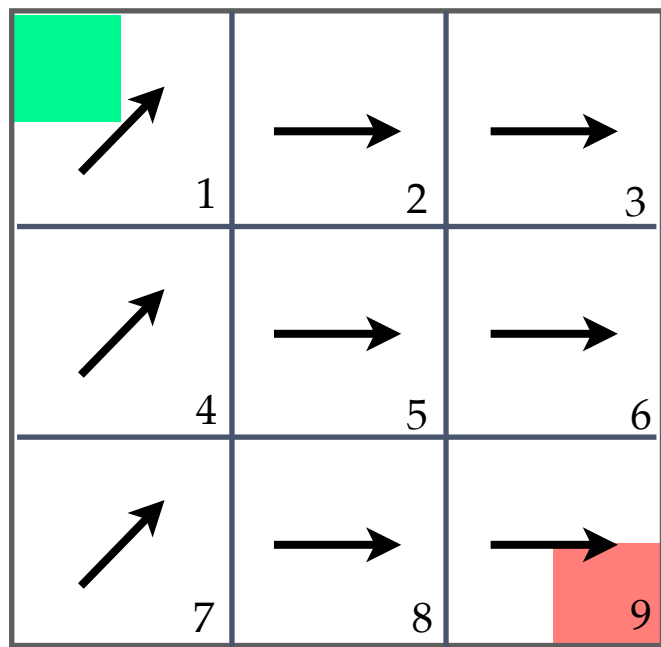
---



## Safety Problem

- ❖ Can the robot reach the red region starting from the green region?
- ❖ There is no bound on the number of cells the robot crosses — unbounded safety analysis problem.

# Abstraction



- ✦ Construct a finite graph where the nodes correspond to cells and edges between them to trajectories between the corresponding cells
- ✦ Every trajectory corresponds to a path in the graph
- ✦ Absence of a path from green to red node implies safety

# Predicate Abstraction

---

- ❖ A technique for constructing a finite state abstraction from a finite set of predicates [Graf & Saidi 97]
- ❖ The abstract system simulates the concrete system

- ❖ Predicate  
 $P \subseteq \mathcal{S}$

- ❖ Fix a set of predicates

$$\Pi = \{P_1, \dots, P_k\}$$

- ❖ Abstraction function

$$\alpha_\Pi : \mathcal{S} \rightarrow \{0, 1\}^k$$

$$s \mapsto (P_1(s), \dots, P_k(s))$$

- ❖ Concretization function

$$\gamma_\Pi : \{0, 1\}^k \rightarrow 2^{\mathcal{S}}$$

$$(b_1, \dots, b_k) \mapsto \bigcap_{i:b_i=1} P_i \cap \bigcap_{i:b_i=0} \mathcal{S} \setminus P_i$$

# Predicate Abstraction

---

- ✧ Set of Predicates

$$\Pi = \{P_1, \dots, P_k\}$$

- ✧ Abstract state-space

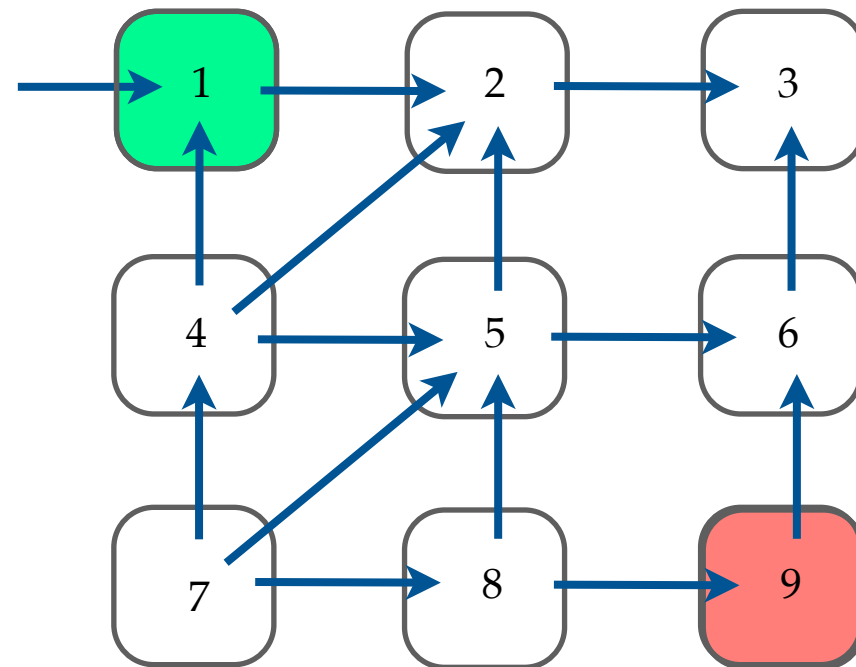
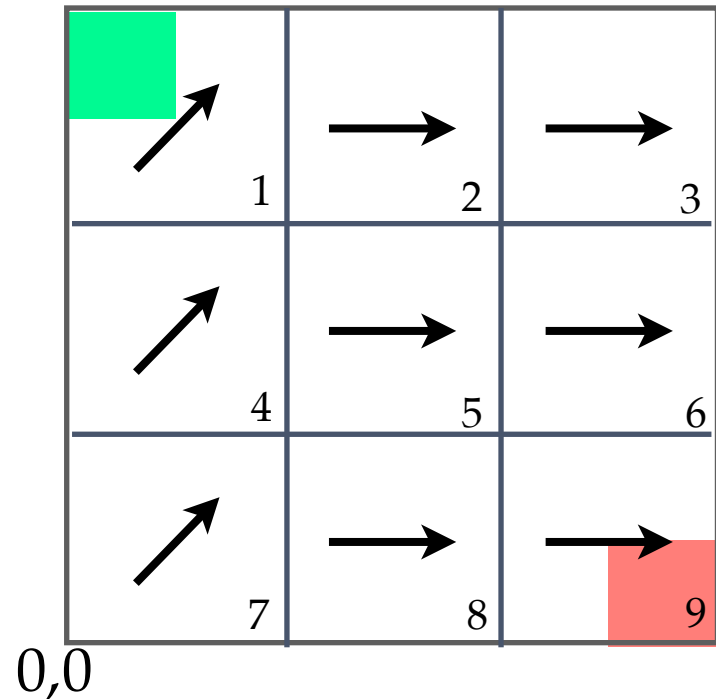
$$\{0, 1\}^k$$

- ✧ Abstract transitions

$$b_1 \rightarrow_A b_2$$

$$\exists s_1 \in \gamma_\Pi(b_1), s_2 \in \gamma_\Pi(b_2) : s_1 \rightarrow_C s_2$$

# Predicate Abstraction: Example

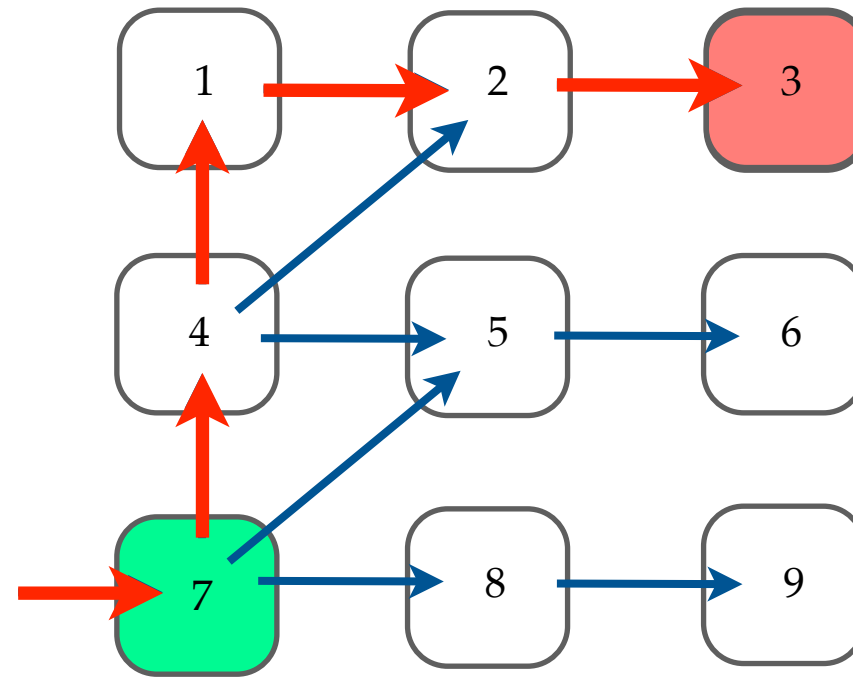
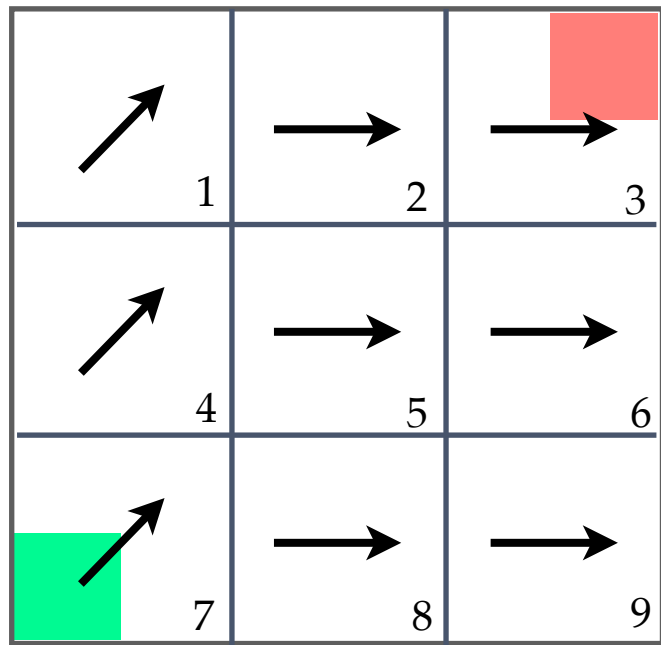


Predicates:  $x \leq 0$ ,  $x \leq 1$ ,  $x \leq 2$ ,  $x \leq 3$ ,  $y \leq 0$ ,  $y \leq 1$ ,  $y \leq 2$ ,  $y \leq 3$

Abstraction Function:  $s \mapsto i$  if  $s \in C_i$

Concretization Function:  $i \mapsto C_i$

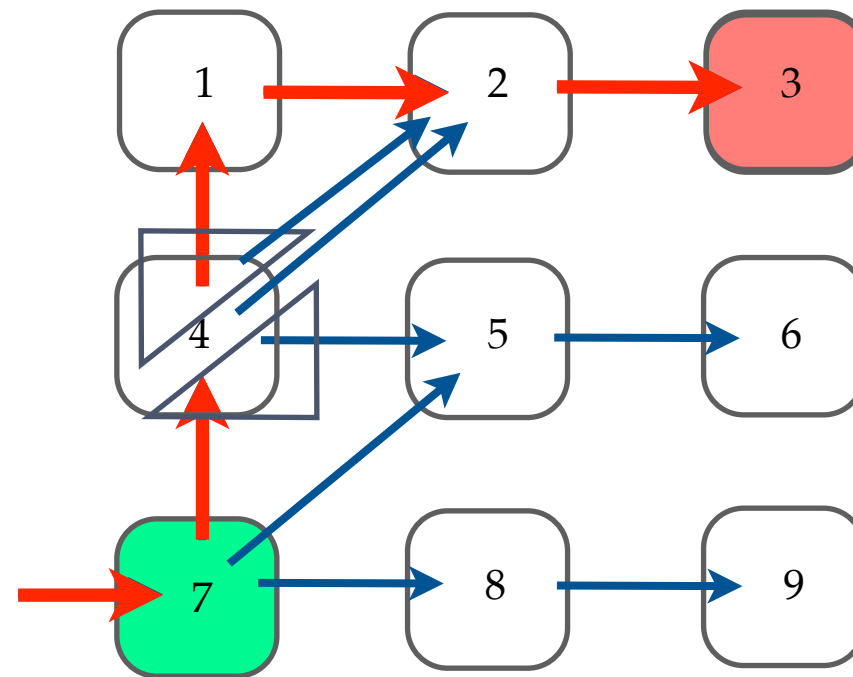
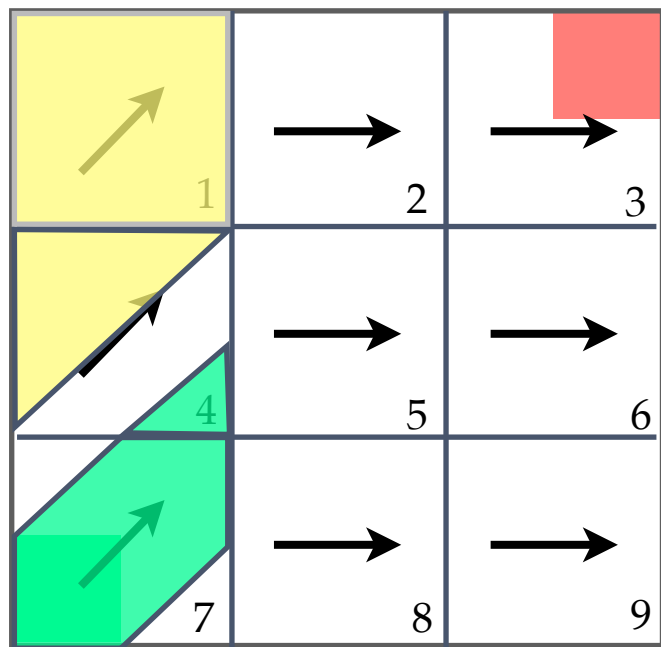
# Abstraction



- ❖ The above system is safe
- ❖ The abstract graph has a counter-example
- ❖ Right abstractions are hard to find!

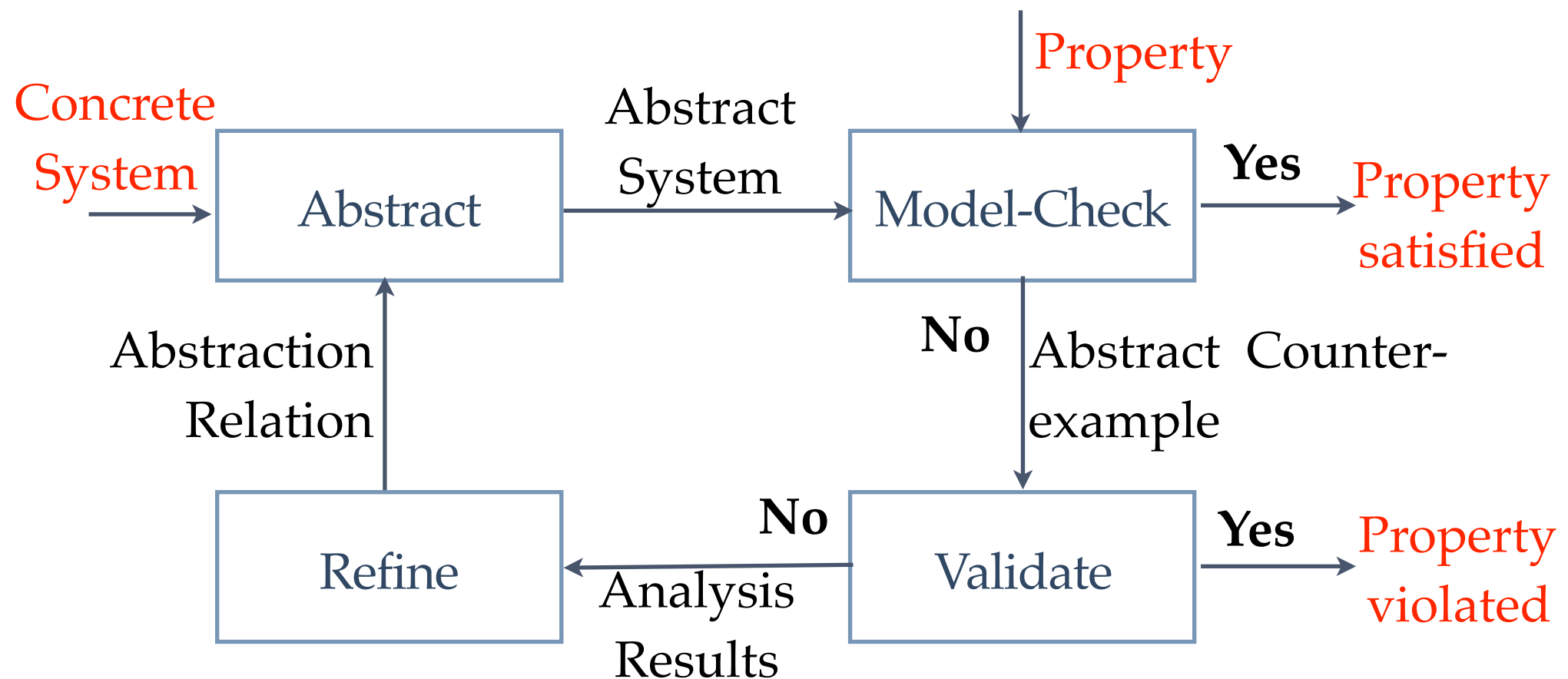


# Refinement



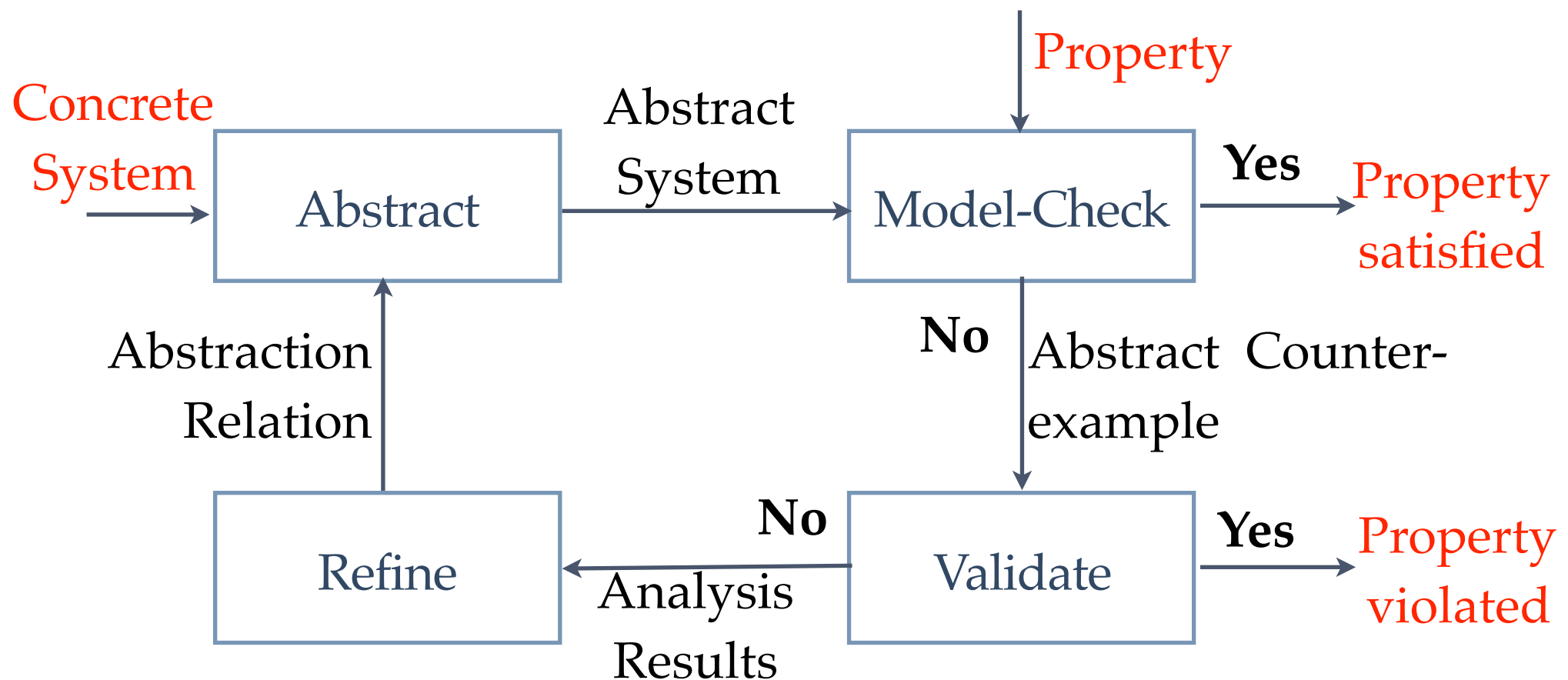
- ❖ Refine by analyzing the abstract counter-example
- ❖ Check if counter-example corresponds to an actual trajectory
- ❖ If yes, then the system is unsafe
- ❖ Otherwise, it is a spurious counter-example and we use the knowledge from the analysis to refine the abstraction

# Counter-example guided abstraction refinement



- ❖ **CEGAR for discrete systems** [Kurshan et al. 93, Clarke et al. 00, Ball et al. 02]
- ❖ **CEGAR for hybrid systems safety verification** [Alur et al 03, Clarke et al 03, Prabhakar et al 13]

# Challenges with CEGAR



- ❖ Finite abstraction construction involves CPost computation
- ❖ Validation is a bounded model-checking problem and can only be performed exactly for limited dynamics (so there is no guarantee of exhibiting an unsafe trajectory even if the counter-example is valid)

# Hybridization

---

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

$$x \in X_0 \subseteq \mathbb{R}^n$$

- ❖ Divide the state-space into a finite number of regions
- ❖ Approximate the dynamics on the right hand side by simple dynamics solving optimization problems
- ❖ Hybridization techniques consider different simpler abstract dynamics including rectangular, linear [Puri, Borkar, Varaiya], [Asarin,Dang,Girard]

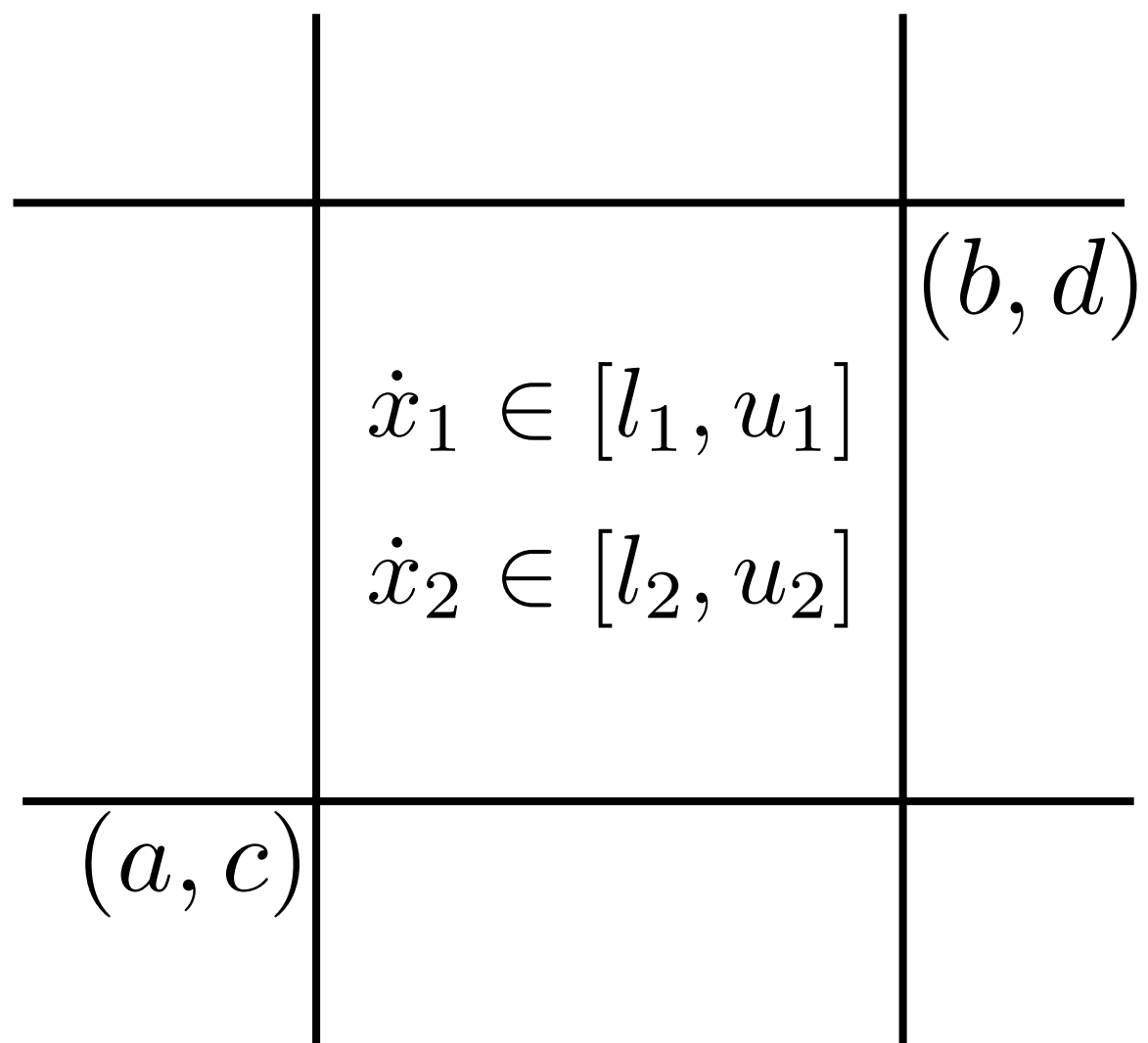
# Hybridization — Rectangular Approximation

---

$$\dot{x}_1 = f_1(x_1, x_2)$$

$$\dot{x}_2 = f_2(x_1, x_2)$$

Find a rectangular approximation  
of  $f(x)$  in each cell



Value of  $u_1$

*Maximize*  $f_1(x_1, x_2)$

$$a \leq x_1 \leq b$$

$$c \leq x_2 \leq d$$

# Hybridization

---

- ❖ Can bound the error of approximation between the right hand sides of the differential equation
- ❖ However, it does not provide a global bound on the error between the solutions
- ❖ Abstraction construction is simpler
- ❖ Model-checking is more complex
- ❖ The problems with validation remain

# Summary and Research Challenges

| Verification technique | Problems that can be solved          | Precision of abstraction | Computational challenges                    |
|------------------------|--------------------------------------|--------------------------|---|
| SMT based verification | Bounded safety                       | Provide error bound      | Need to solve differential equations        |
| Flowpipe construction  | Bounded safety (sometimes unbounded) | Provide error bound      | Need to solve differential equations        |
| Predicate abstraction  | Unbounded safety                     | No error bound           | Require (overapproximate) CPost computation |
| Hybridization          | Unbounded safety                     | Provide error bound      | Rely only on optimization                   |

# Summary and Research Challenges

| Class of systems    | Form of solutions                  | Bounded safety analysis                                | Unbounded safety                                  |
|---------------------|------------------------------------|--|---|
| Timed / Rectangular | Solutions are linear               | Decidable  | Decidable under some constraints on the switching |
| Linear              | Solutions are exponential          | Not known<br>(Bounded error approximations computable) | In general, undecidable                           |
| Nonlinear           | Closed form solutions do not exist | Not known  | In general, undecidable                           |



# Summary and Research Challenges

---

- ❖ How to compute approximations of CPost that are
  - ❖ Precise
  - ❖ Have efficient representation
  - ❖ Low computation overhead
- ❖ How to design an abstract refinement framework that provides
  - ❖ Abstractions that have efficient analysis algorithms
  - ❖ Abstractions that are efficiently computable
  - ❖ Better refinement strategies for the guiding the proof search