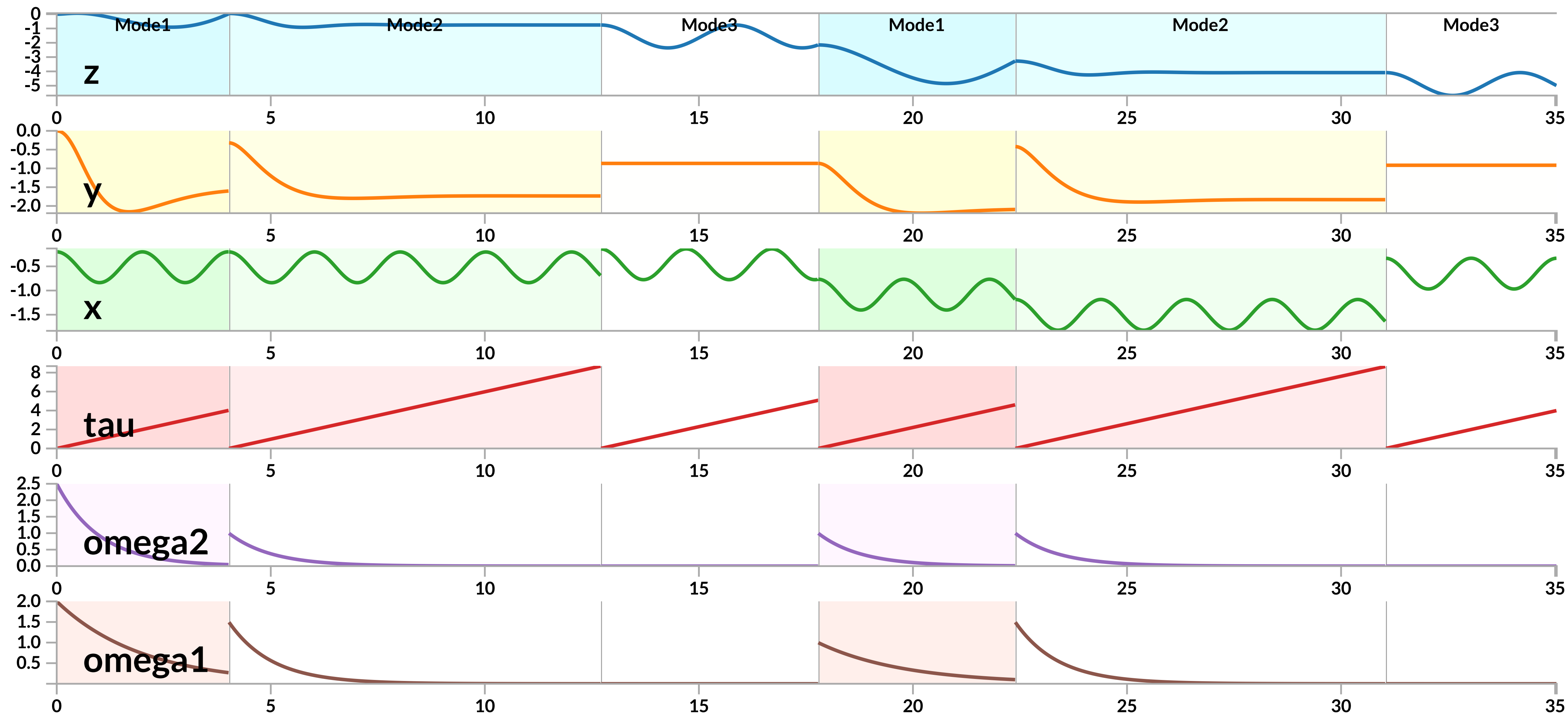
dReach: δ-Reachability Analysis for Hybrid Systems*

Soonho Kong
soonhok@cs.cmu.edu
Carnegie Mellon University

*Joint work with Sicun Gao(MIT), Wei Chen(CMU), and Edmund Clarke(CMU)

Hybrid Systems

Discrete Control + Continuous Dynamics



3-mode Oscillator

Hybrid Systems

Discrete Control + Continuous Dynamics

Cardiac Cell Model

Mode 1

$\text{flow}_1:$

$$\frac{du}{dt} = \varepsilon - \frac{u}{\tau_{o1}}$$

$$\frac{dv}{dt} = \frac{1-v}{\tau_{v1}^-}$$

$$\frac{dw}{dt} = \frac{1 - \frac{u}{\tau_w^\infty} - w}{\tau_{w1}^- + \frac{\tau_{w2}^- - \tau_{w1}^-}{1+e^{-2k_w^-(u-u_w^-)}}}$$

$$\frac{ds}{dt} = \left(\frac{1}{1+e^{-2k_s(u-u_s)}} - s\right)\frac{1}{\tau_{s1}}$$

Mode 1 $\xrightarrow{u \geq \theta_o}$ Mode 2; Mode 2 $\xrightarrow{u < \theta_o}$ Mode 1

Mode 2

$\text{flow}_2:$

$$\frac{du}{dt} = \varepsilon - \frac{u}{\tau_{o2}}$$

$$\frac{dv}{dt} = -\frac{v}{\tau_{v2}^-}$$

$$\frac{dw}{dt} = \frac{w_\infty^* - w}{\tau_{w1}^- + \frac{\tau_{w2}^- - \tau_{w1}^-}{1+e^{-2k_w^-(u-u_w^-)}}}$$

$$\frac{ds}{dt} = \left(\frac{1}{1+e^{-2k_s(u-u_s)}} - s\right)\frac{1}{\tau_{s1}}$$

Mode 2 $\xrightarrow{u \geq \theta_w}$ Mode 3; Mode 3 $\xrightarrow{u < \theta_w}$ Mode 2

Mode 3

$\text{flow}_3:$

$$\frac{du}{dt} = \varepsilon - \frac{1}{\tau_{so1} + \frac{\tau_{so2} - \tau_{so1}}{1+e^{-2k_{so}(u-u_{so})}}} + \frac{w \cdot s}{\tau_{si}}$$

$$\frac{dv}{dt} = -\frac{v}{\tau_{v2}^-}$$

$$\frac{dw}{dt} = -\frac{w}{\tau_w^+}$$

$$\frac{ds}{dt} = \left(\frac{1}{1+e^{-2k_s(u-u_s)}} - s\right)\frac{1}{\tau_{s1}}$$

Mode 3 $\xrightarrow{u \geq \theta_v}$ Mode 4; Mode 4 $\xrightarrow{u < \theta_v}$ Mode 3

Mode 4

$\text{flow}_4:$

$$\frac{du}{dt} = \varepsilon + \frac{v(u-\theta_v)(u_u - u)}{\tau_{fi}} - \frac{1}{\tau_{so1} + \frac{\tau_{so2} - \tau_{so1}}{1+e^{-2k_{so}(u-u_{so})}}} + \frac{w \cdot s}{\tau_{si}}$$

$$\frac{dv}{dt} = -\frac{v}{\tau_{v2}^-}$$

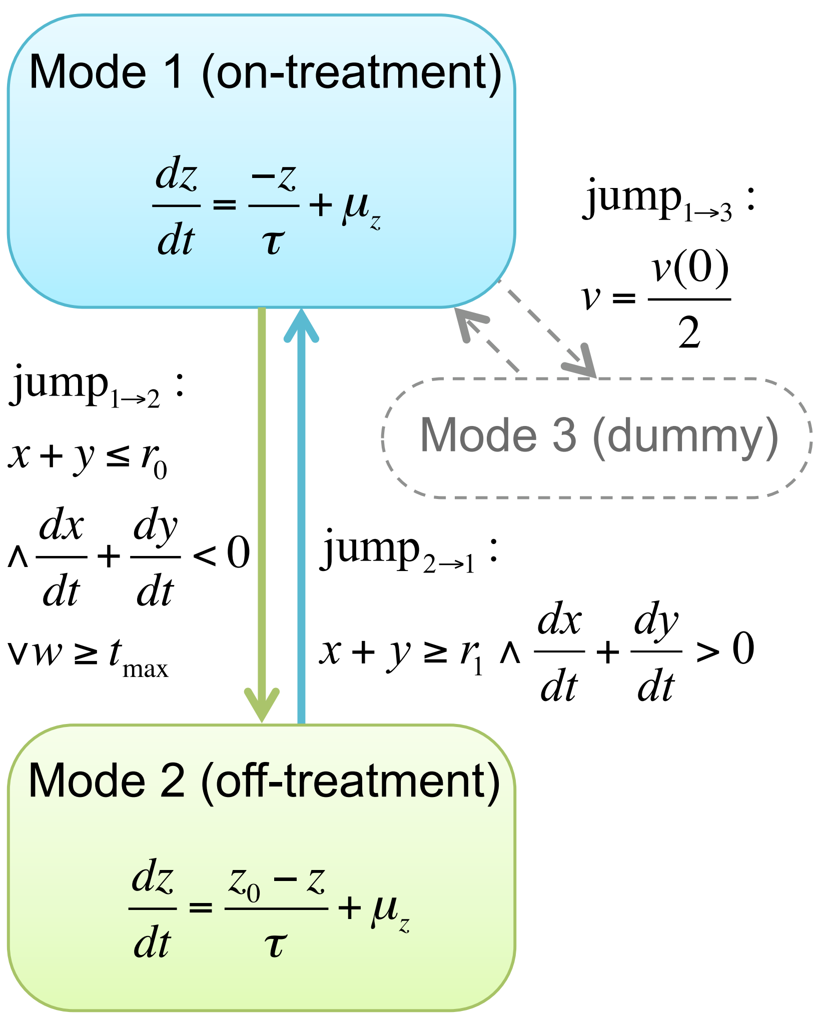
$$\frac{dw}{dt} = -\frac{w}{\tau_w^+}$$

$$\frac{ds}{dt} = \left(\frac{1}{1+e^{-2k_s(u-u_s)}} - s\right)\frac{1}{\tau_{s1}}$$

Hybrid Systems

Discrete Control + Continuous Dynamics

Prostate Cancer Model



$\|$

$$\frac{dx}{dt} = \left(\alpha_x \left(\frac{1}{1+e^{-(z-k_1)k_2}} \right) - \beta_x \left(\frac{1}{1+e^{-(z-k_3)k_4}} \right) - m_1 \left(1 - \frac{z}{z_0}\right) - \lambda_x \right) x + \mu_x$$

$$\frac{dy}{dt} = m_1 \left(1 - \frac{z}{z_0}\right) x + \left(\alpha_y \left(1 - d\,\frac{z}{z_0}\right) - \beta_y \right) y$$

$$\frac{dv}{dt} = \left(\alpha_x \left(\frac{1}{1+e^{-(z-k_1)k_2}} \right) - \beta_x \left(\frac{1}{1+e^{-(z-k_3)k_4}} \right) - m_1 \left(1 - \frac{z}{z_0}\right) - \lambda_x \right) x + \mu_x$$
$$+ m_1 \left(1 - \frac{z}{z_0}\right) x + \left(\alpha_y \left(1 - d\,\frac{z}{z_0}\right) - \beta_y \right) y$$

Control (cancer therapy)

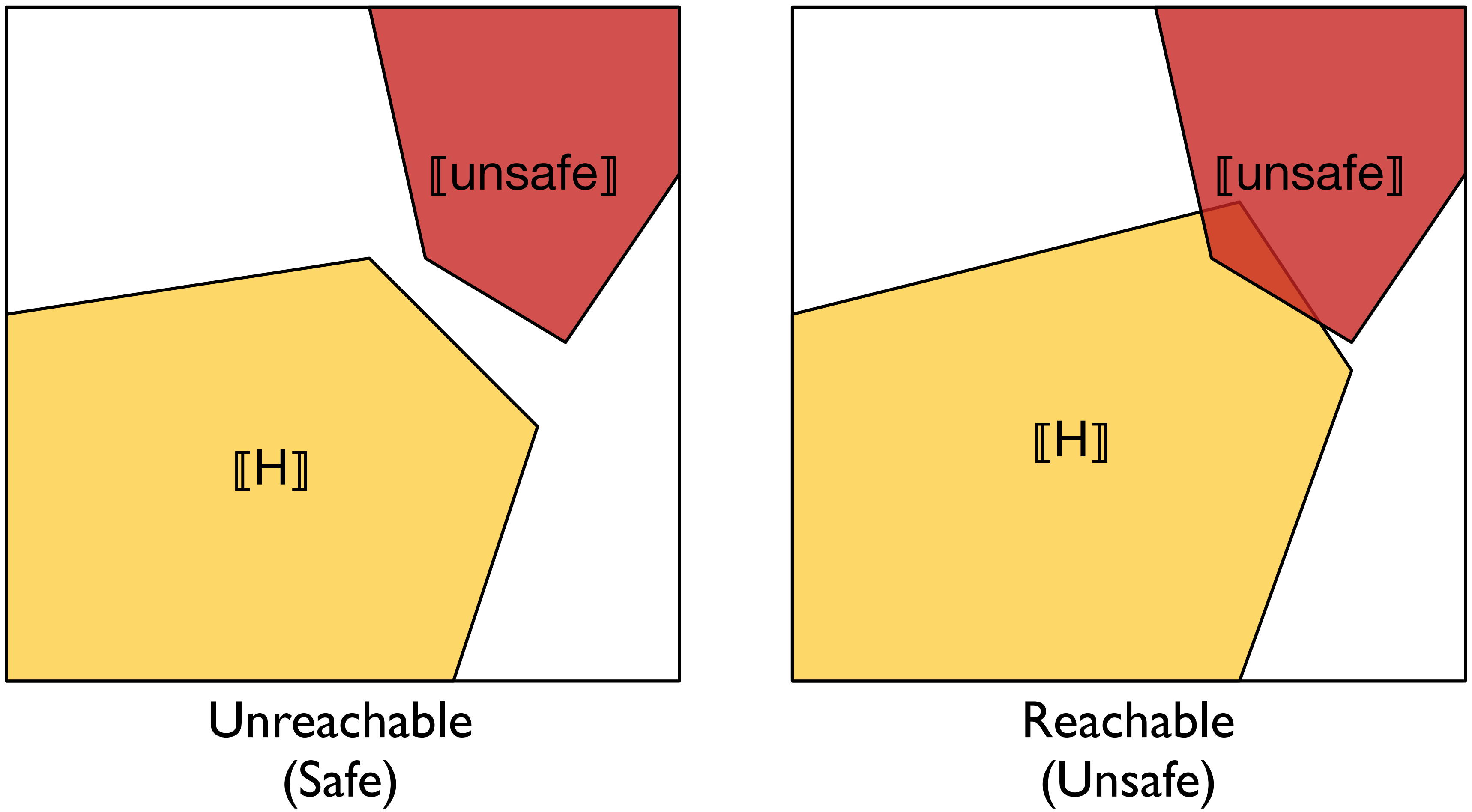
Plant (cancer progression)

Reachability Analysis of Hybrid Systems

Can a hybrid system run into an **unsafe** region of its state space?

Reachability Analysis of Hybrid Systems

Can a hybrid system run into an **unsafe** region of its state space?



The standard bounded reachability problems for simple hybrid systems are **undecidable**.

Reachability Analysis of Hybrid Systems

The standard bounded reachability problems for simple hybrid systems are **undecidable**.

Reachability Analysis of Hybrid Systems

The standard bounded reachability problems for simple hybrid systems are **undecidable**.

1. Give up

2. Don't give Up

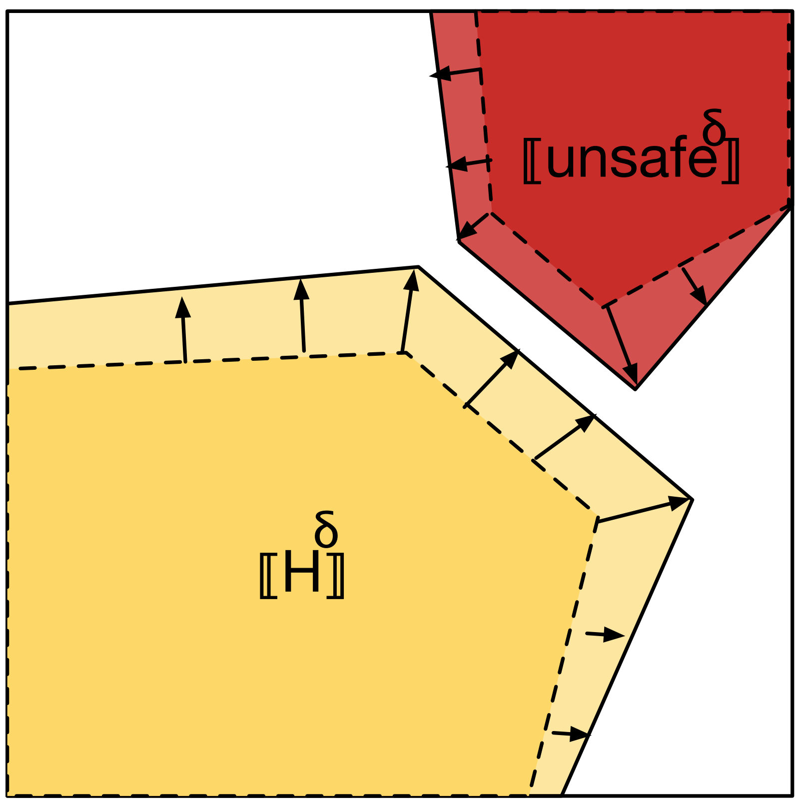
   A. Find a decidable fragment and solve it

   B. Use approximation

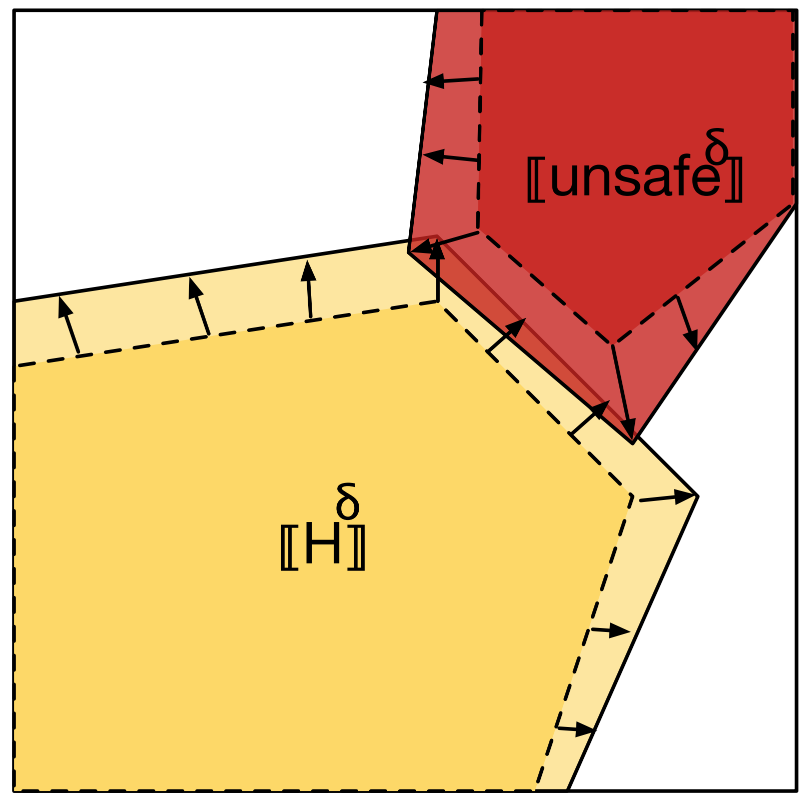
δ-Reachability Analysis of Hybrid Systems

Given $\delta \in \mathbb{Q}^+$, $[\![H]\!]^\delta$ and $[\![unsafe]\!]^\delta$ **over-approximate** $[\![H]\!]$ and $[\![unsafe]\!]$

δ-reachability problem asks for one of the following answers:



Unreachable
(Safe)



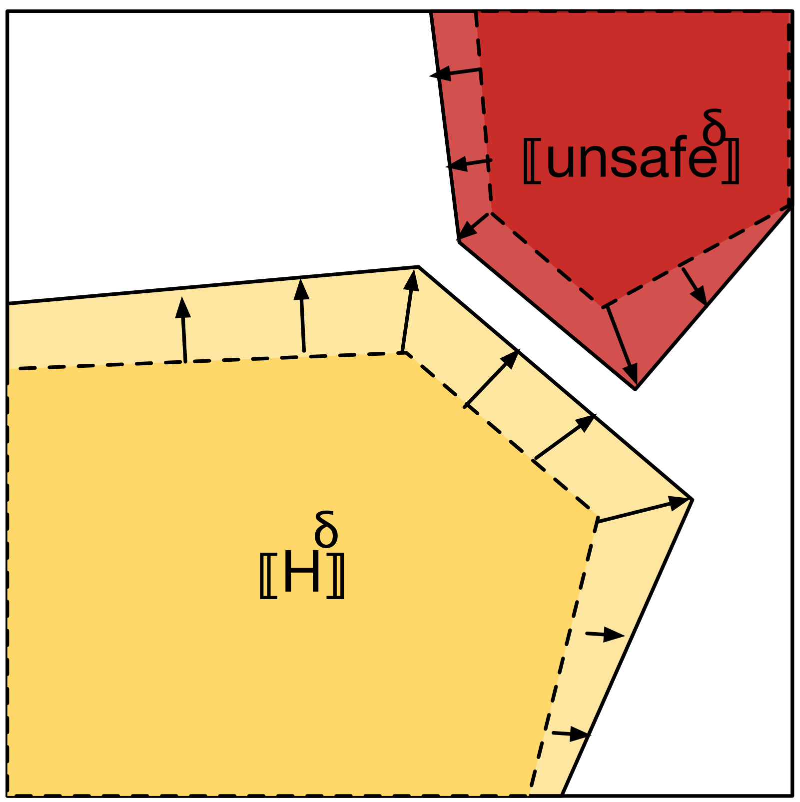
δ-reachable
(Unsafe)

- **Decidable** for a wide range of **nonlinear** hybrid systems
    - polynomials, log, exp, trigonometric functions, …

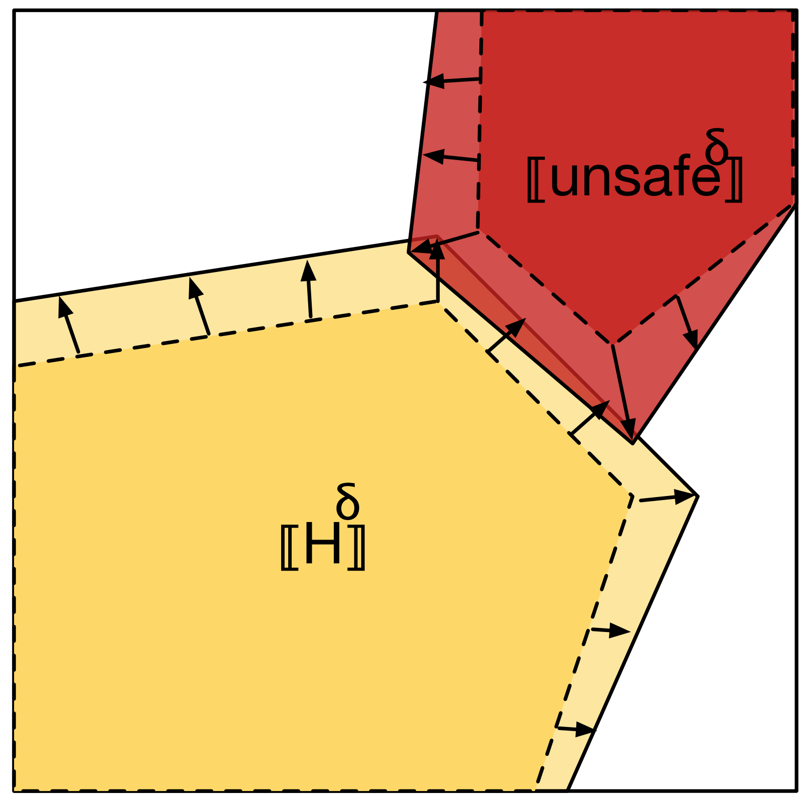
δ-Reachability Analysis of Hybrid Systems

Given $\delta \in \mathbb{Q}^+$, $[\![H]\!]^\delta$ and $[\![unsafe]\!]^\delta$ **over-approximate** $[\![H]\!]$ and $[\![unsafe]\!]$

δ-reachability problem asks for one of the following answers:



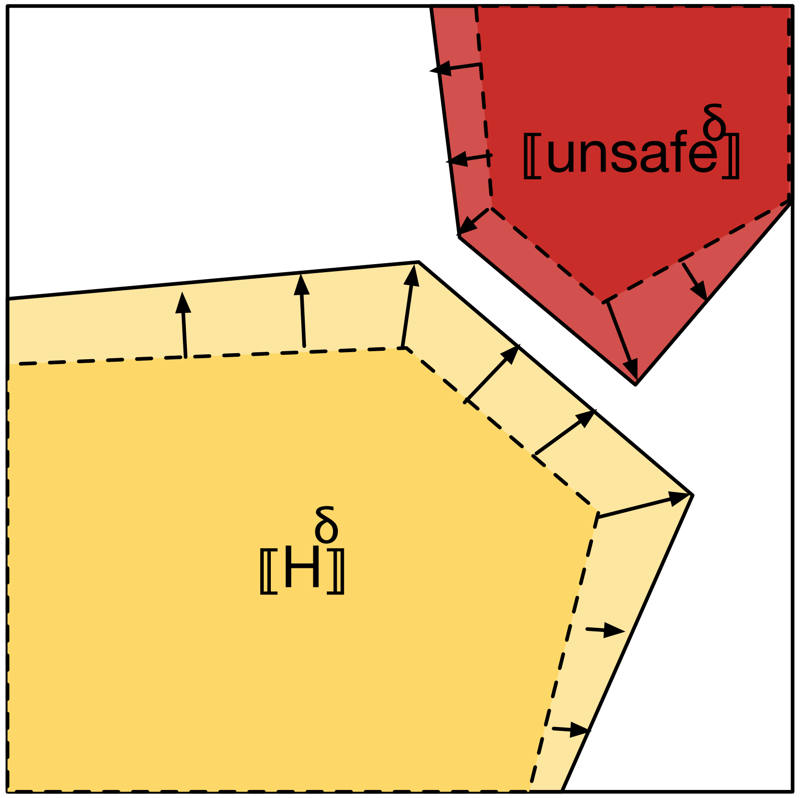
Unreachable
(Safe)



δ-reachable
(Unsafe)

- **Decidable** for a wide range of **nonlinear** hybrid systems
- **Reasonable** complexity bound (PSPACE-complete)

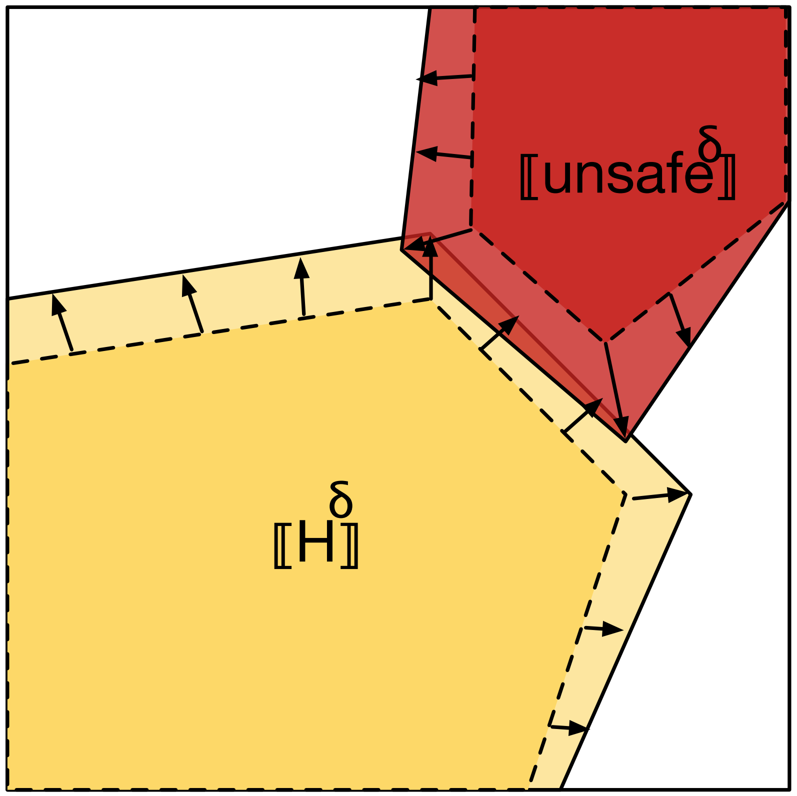
δ-Reachability Analysis of Hybrid Systems



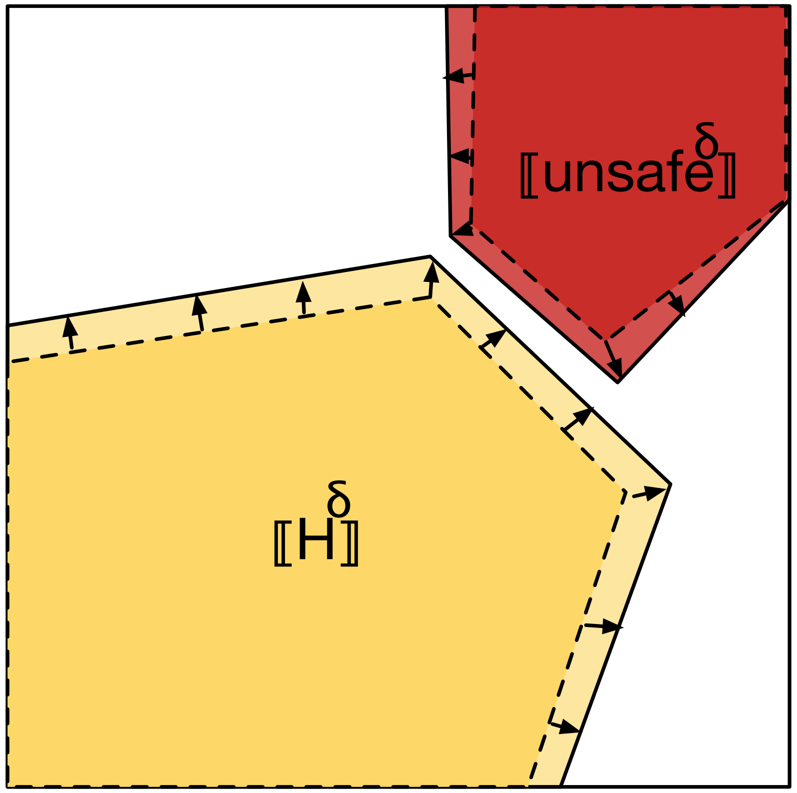
Unreachable
(Safe)

1. "Unreachable" answers is **sound**.

δ-Reachability Analysis of Hybrid Systems



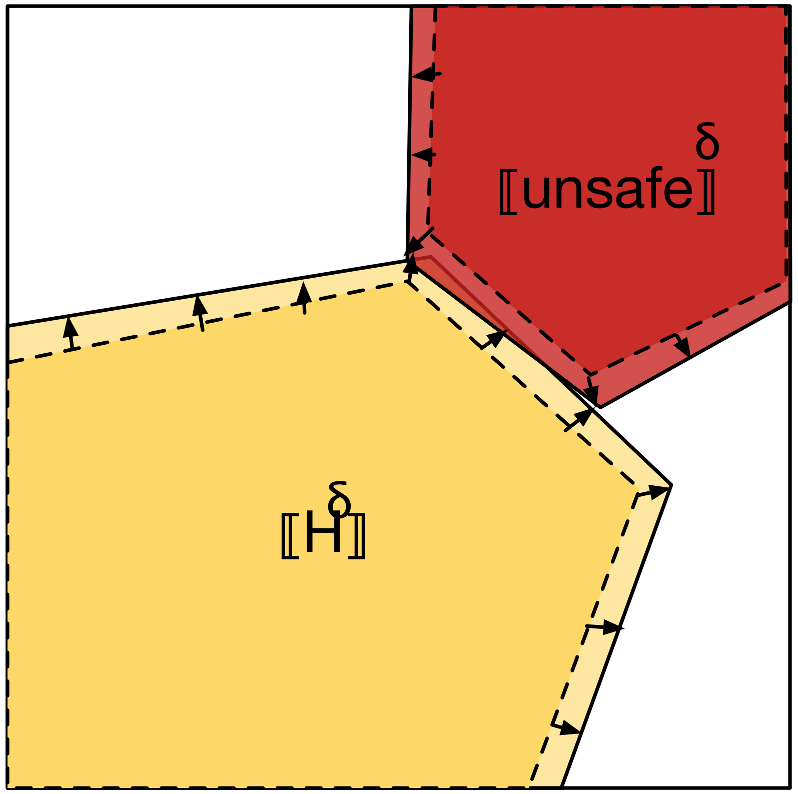
δ-reachable
(Unsafe)



Unreachable with smaller δ
(Safe)

2. Analysis is parameterized with **δ**

δ-Reachability Analysis of Hybrid Systems



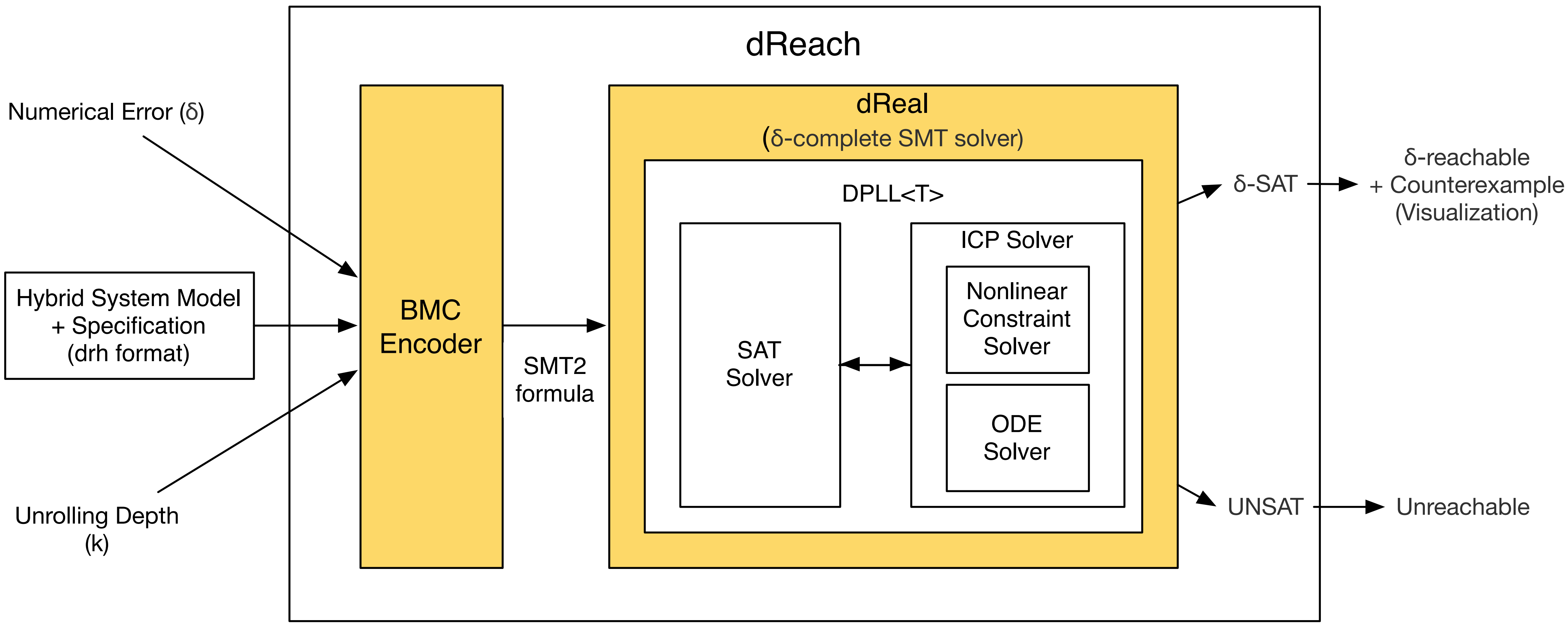
Unreachable
(Unsafe)

3. Robustness:
   If your system is **δ-reachable** under a reasonably small δ, then a small error can lead your system to an **unsafe** state

δ-Reachability Analysis of Hybrid Systems

"δ-reachability analysis checks robustness which implies safety."

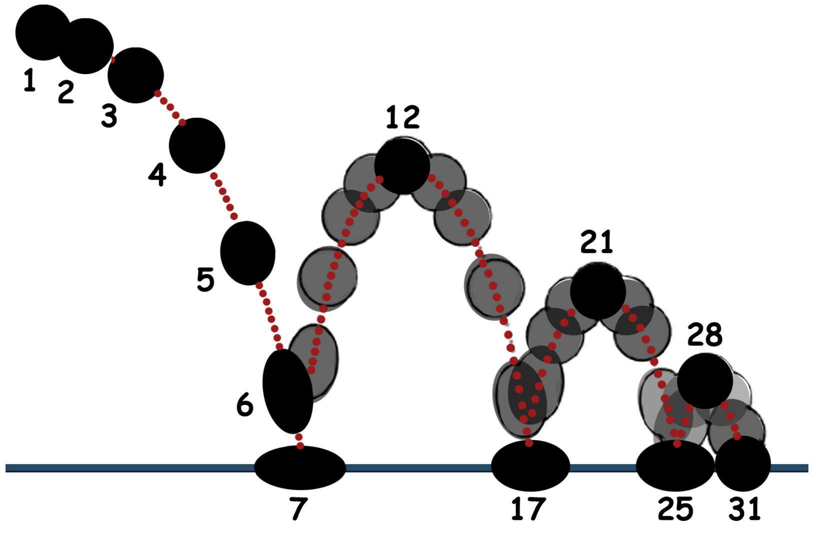
dReach: δ-Reachability Analysis of Hybrid Systems



- Open Source (GPL3), available at https://dreal.github.io
- Support polynomials, transcendental functions and nonlinear ODEs
- Formulas with 100+ ODEs have been solved.

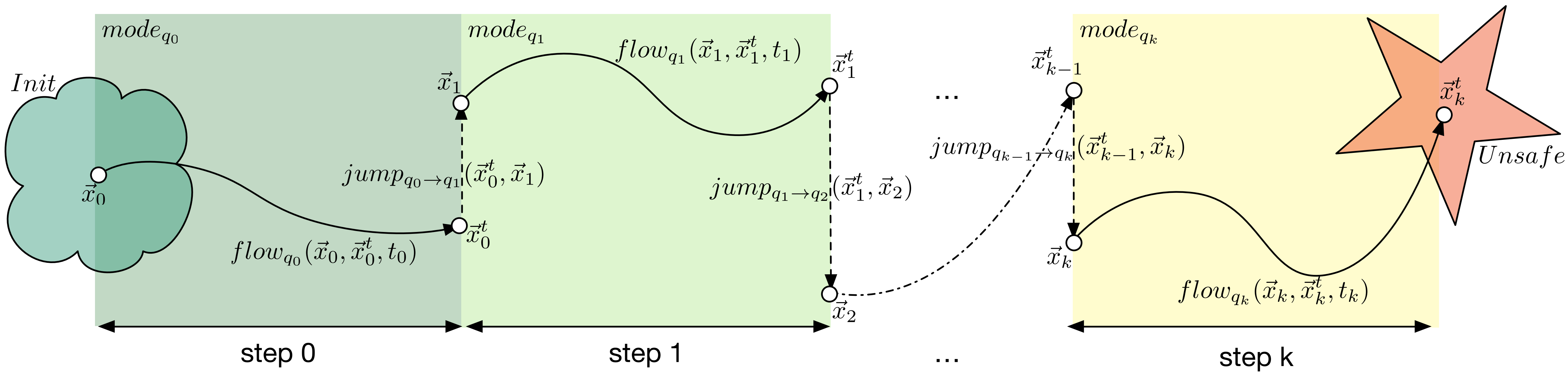
Input Format (drh) for Hybrid System

```
#define D 0.45
#define K 0.9
[0, 15] x;
[9.8] g;
[-18, 18] v;
[0, 3] time;

{   mode 1;
    invt: (v <= 0);
          (x >= 0);
    flow: d/dt[x] = v;
          d/dt[v] = -g - (D * v ^ 2);
    jump: (x = 0) ==> @2 (and (x' = x) (v' = - K * v)); }

{   mode 2;
    invt: (v >= 0);
          (x >= 0);
    flow: d/dt[x] = v;
          d/dt[v] = -g + (D * v ^ 2);
    jump: (v = 0) ==> @1 (and (x' = x) (v' = v)); }

init: @1 (and (x >= 5) (v = 0));
goal: @1 (and (x >= 0.45));
```



Inelastic bouncing ball with air resistance

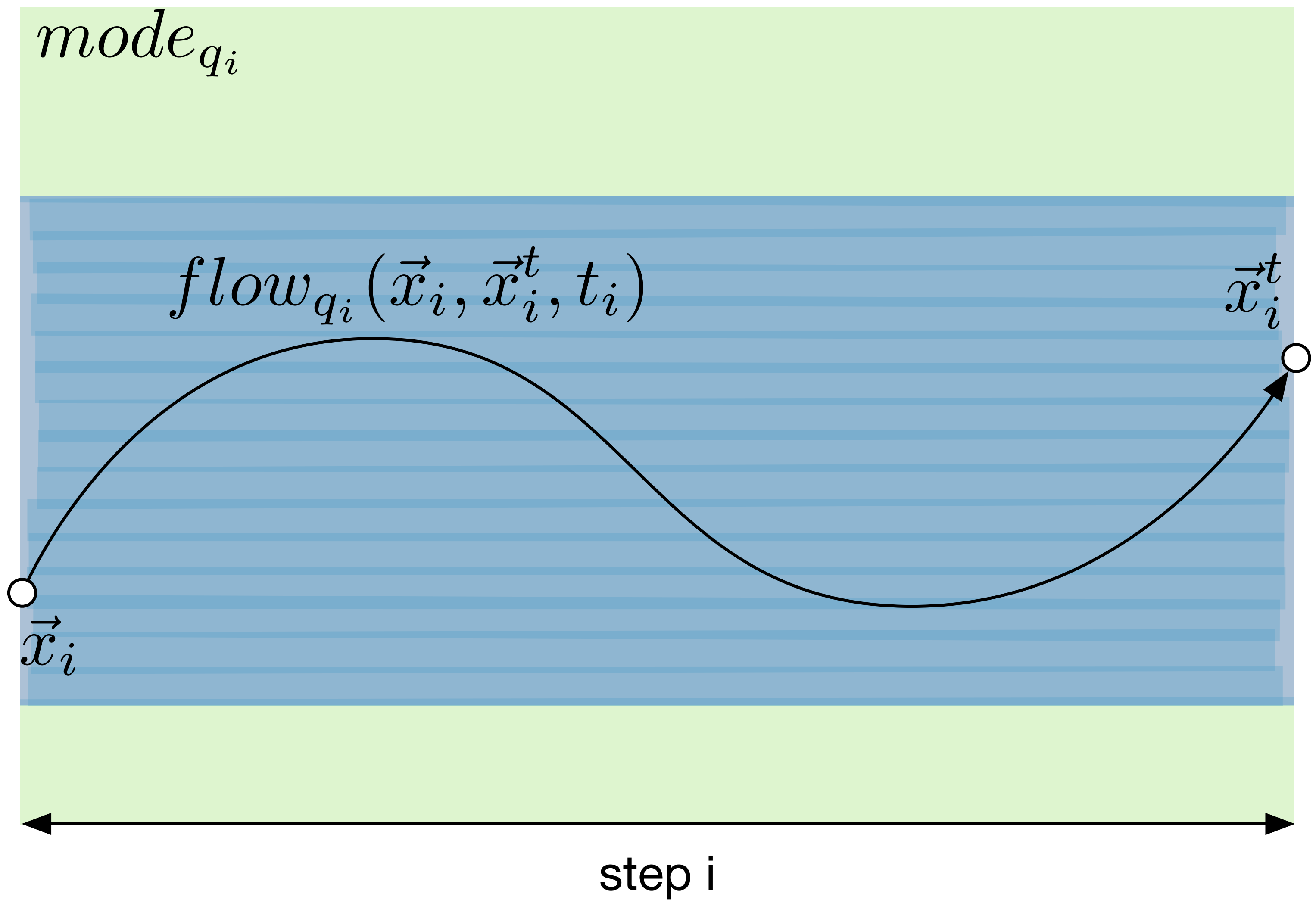
Logical Encoding of Reachability Problem



$$
\begin{aligned}
&\exists \vec{x}_0, \vec{x}_1, \ldots, \vec{x}_k \exists \vec{x}^t_0, \vec{x}^t_1, \ldots, \vec{x}^t_k \exists t_0, t_1, \ldots, t_k \\
&Init(\vec{x}_0) \wedge flow_{q_0}(\vec{x}_0, \vec{x}^t_0, t_0) \wedge jump_{q_0 \to q_1}(\vec{x}^t_0, \vec{x}_1) \wedge \\
&\qquad flow_{q_1}(\vec{x}_1, \vec{x}^t_1, t_1) \wedge jump_{q_1 \to q_2}(\vec{x}^t_1, \vec{x}_2) \wedge \\
&\qquad \ldots \\
&\qquad flow_{q_k}(\vec{x}_k, \vec{x}^t_k, t_k) \wedge Unsafe(\vec{x}_k)
\end{aligned}
$$

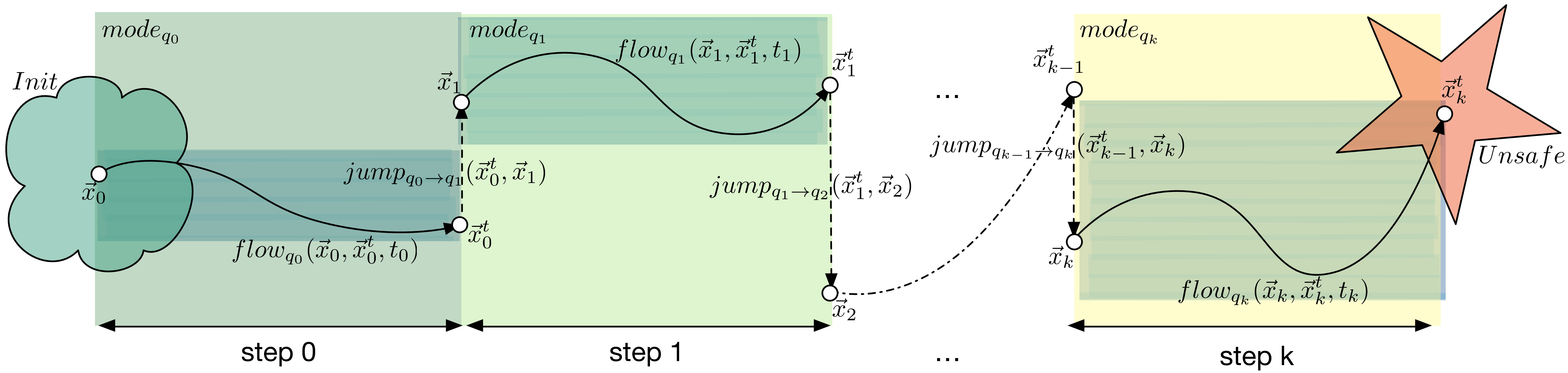
Logical Encoding of Reachability Problem

How to encode a mode invariant



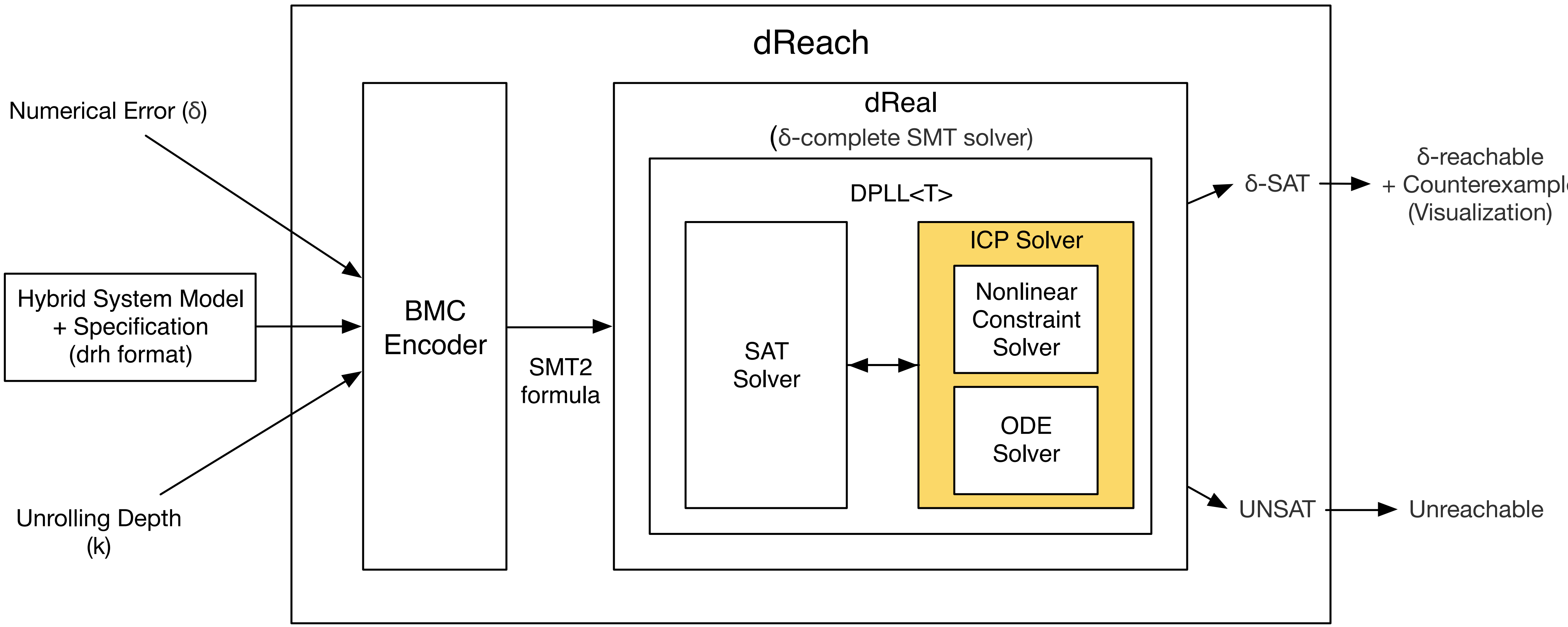
$$\forall t \in [0, t_i]\ \forall \vec{x} \in X\ \ flow_{q_i}(\vec{x}_i, \vec{x}, t) \implies inv_{q_i}(\vec{x})$$

Logical Encoding of Reachability Problem



$$
\begin{aligned}
&\exists \vec{x}_0, \vec{x}_1, \ldots, \vec{x}_k \exists \vec{x}_0^t, \vec{x}_1^t, \ldots, \vec{x}_k^t \exists t_0, t_1, \ldots, t_k \\
&Init(\vec{x}_0) \wedge flow_{q_0}(\vec{x}_0, \vec{x}_0^t, t_0) \wedge \underline{\forall t \in [0, t_0]\ \forall \vec{x} \in X\ flow_{q_0}(\vec{x}_0, \vec{x}, t) \implies inv_{q_0}(\vec{x})} \wedge jump_{q_0 \to q_1}(\vec{x}_0^t, \vec{x}_1) \wedge \\
&\qquad flow_{q_1}(\vec{x}_1, \vec{x}_1^t, t_1) \wedge \underline{\forall t \in [0, t_1]\ \forall \vec{x} \in X\ flow_{q_1}(\vec{x}_1, \vec{x}, t) \implies inv_{q_1}(\vec{x})} \wedge jump_{q_1 \to q_2}(\vec{x}_1^t, \vec{x}_2) \wedge \\
&\qquad \ldots \\
&\qquad flow_{q_k}(\vec{x}_k, \vec{x}_k^t, t_k) \wedge \underline{\forall t \in [0, t_k]\ \forall \vec{x} \in X\ flow_{q_k}(\vec{x}_k, \vec{x}, t) \implies inv_{q_k}(\vec{x})} \wedge Unsafe(\vec{x}_k)
\end{aligned}
$$

How to Solve



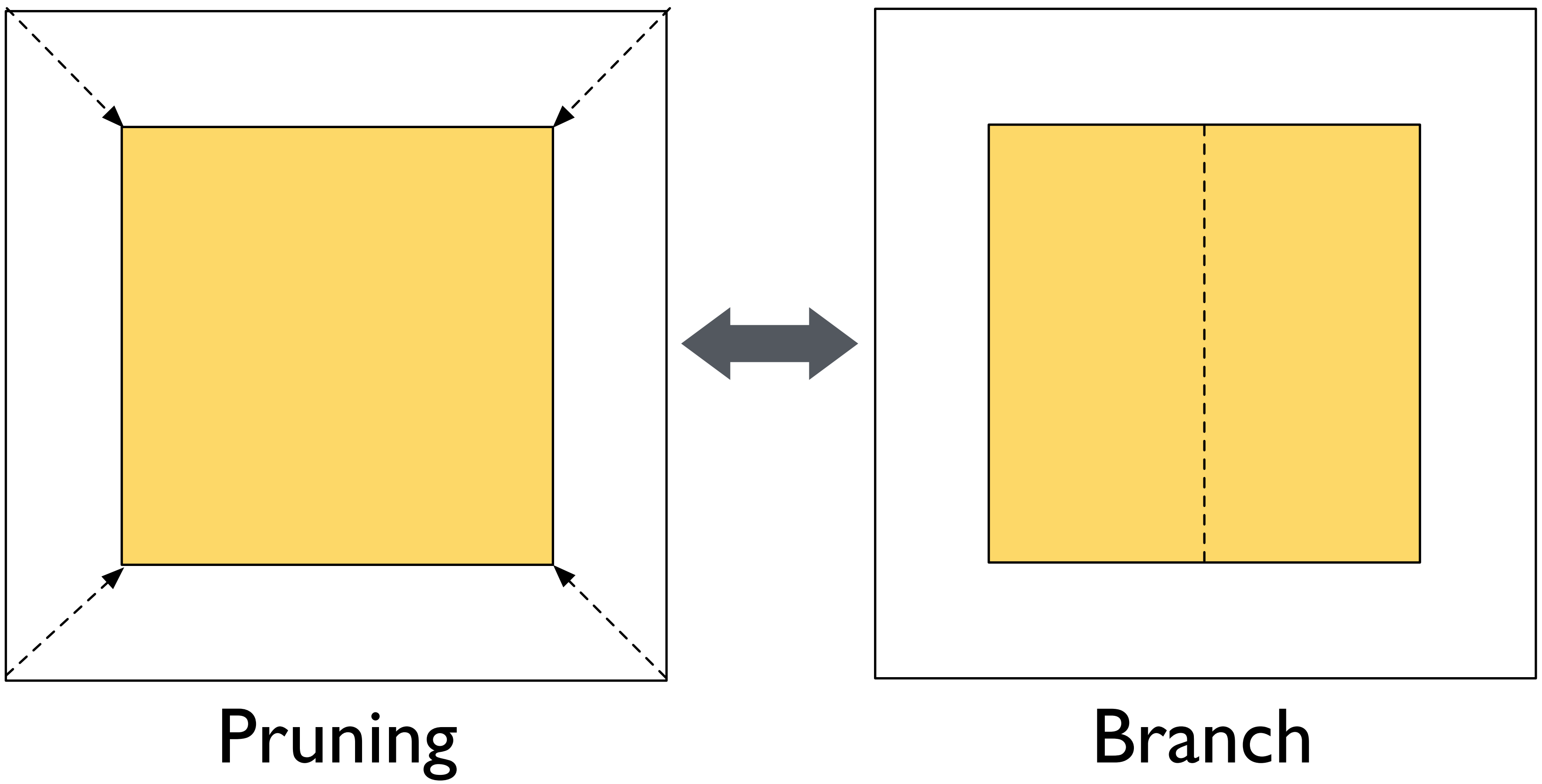
Theory Solver

Input:

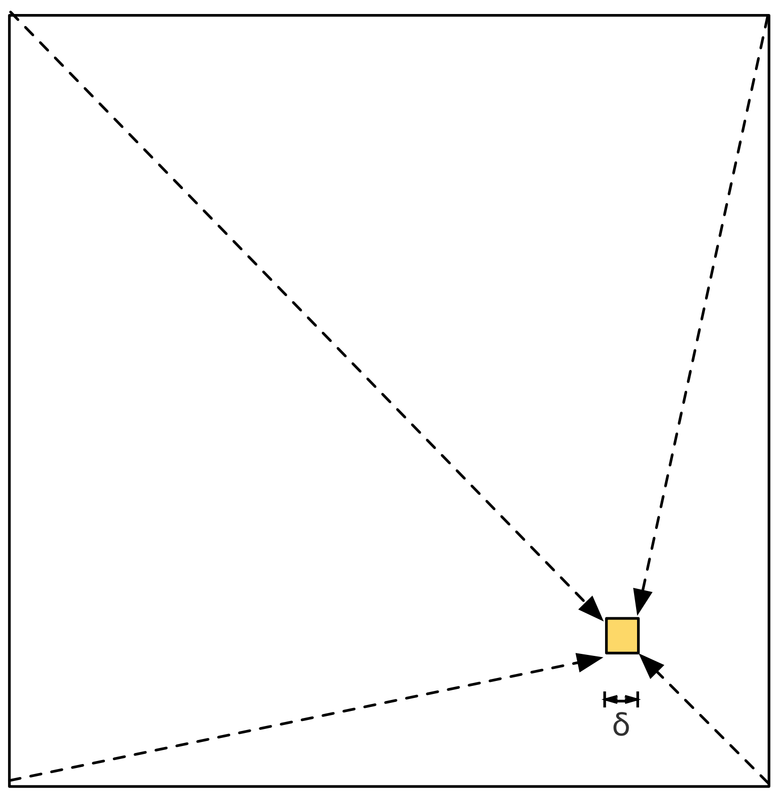
- Box (search space)
- List of constraints $l_1 \wedge l_2 \wedge \cdots \wedge l_n$

Output: δ-sat or Unsat

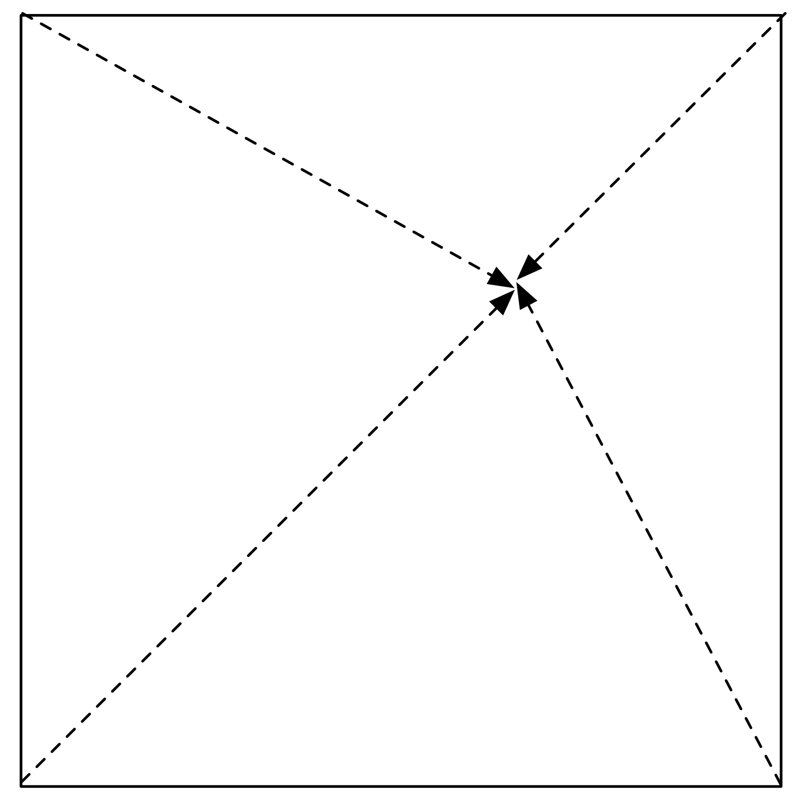
Main Algorithm: Interval Constraint Propagation



Main Algorithm: Interval Constraint Propagation

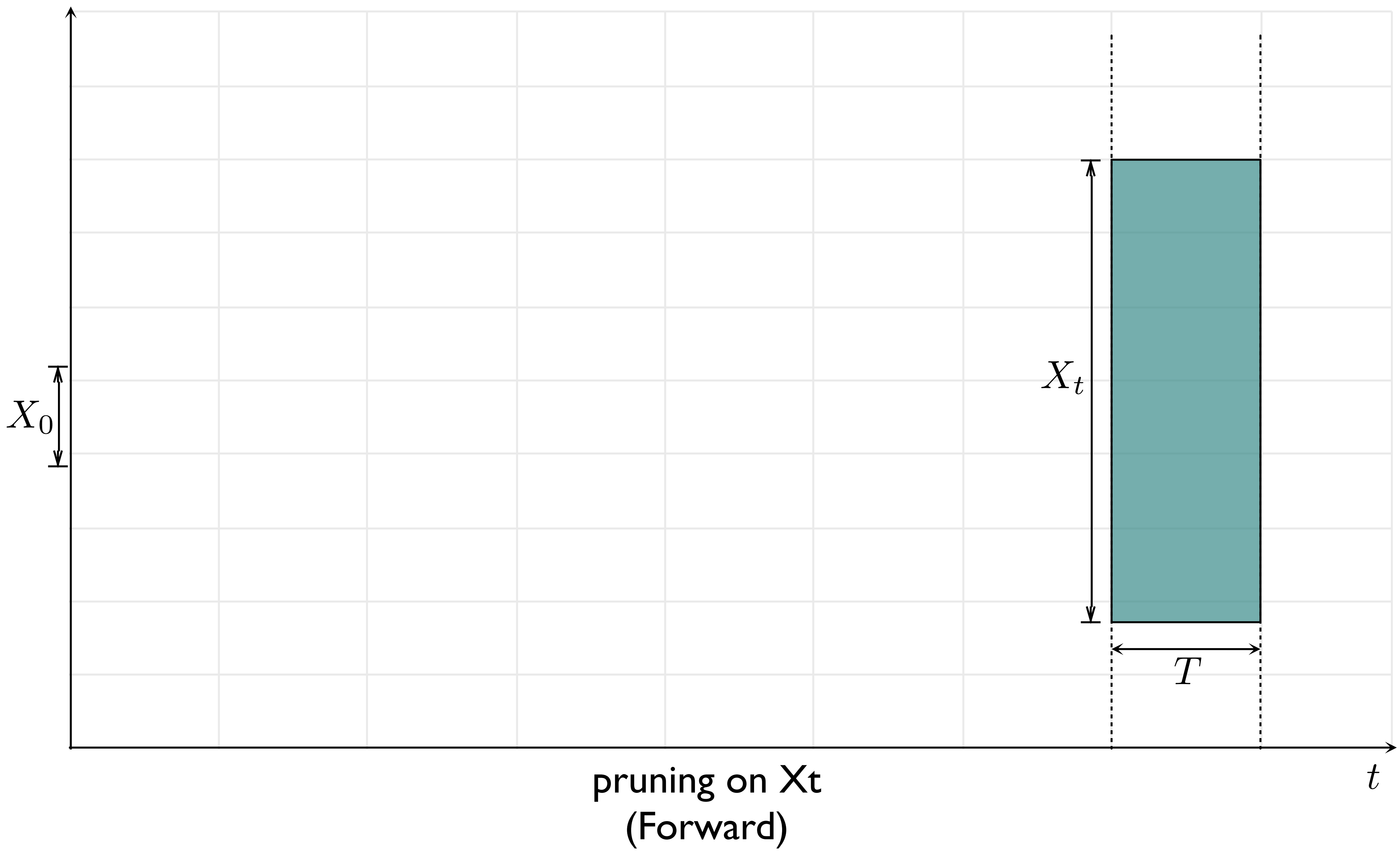


δ-sat

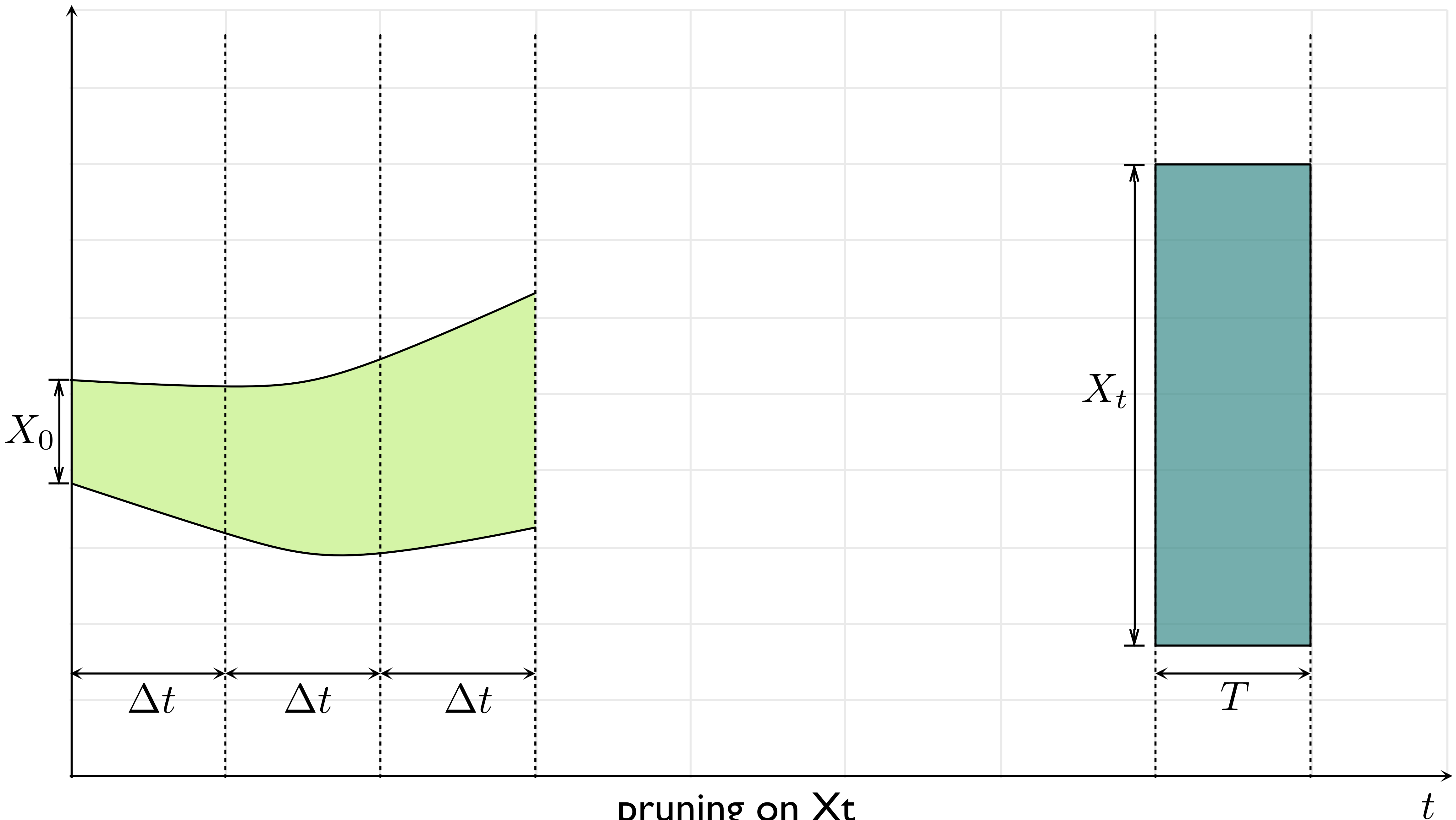


Unsat

Pruning using ODEs

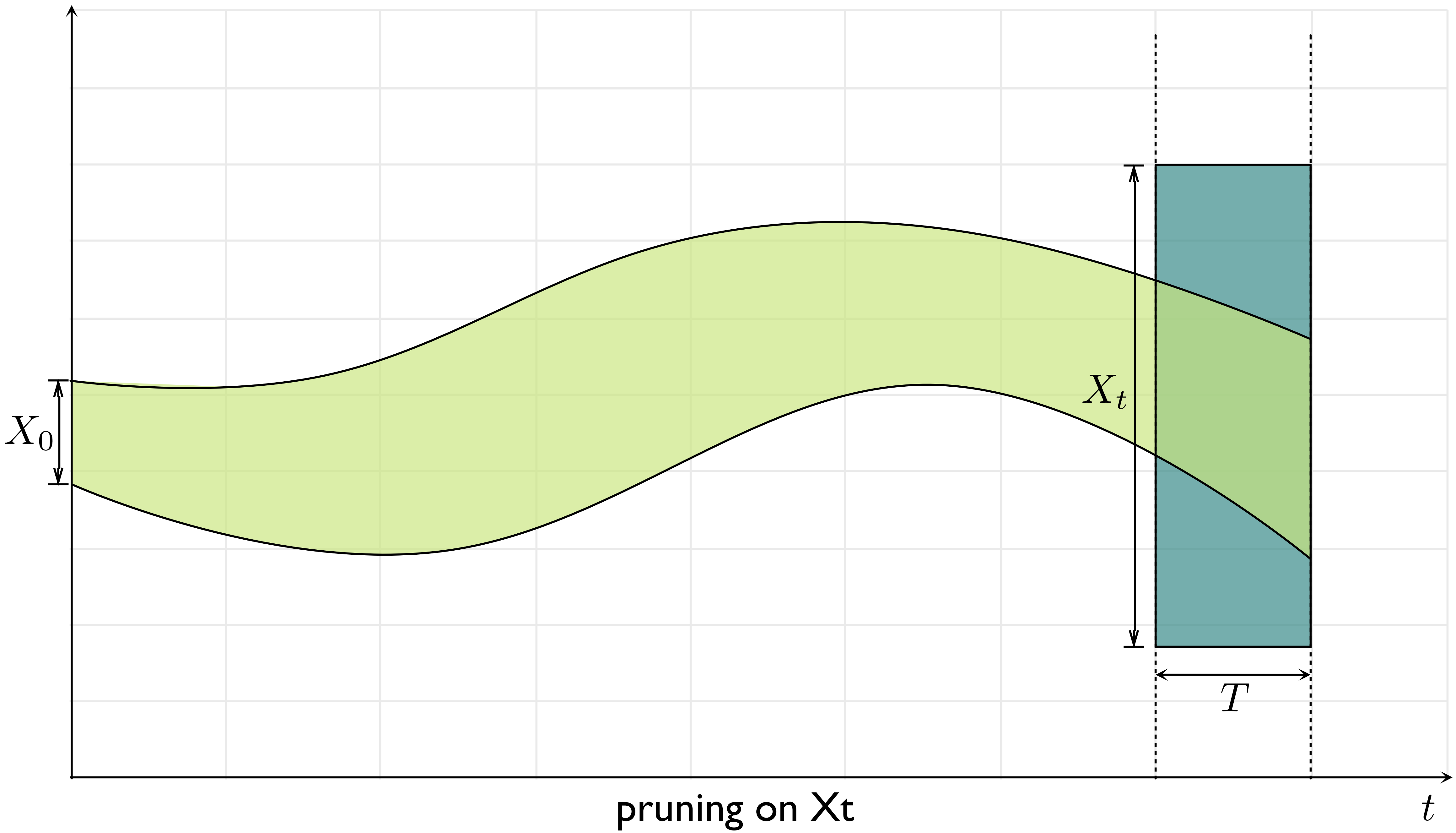


Pruning using ODEs



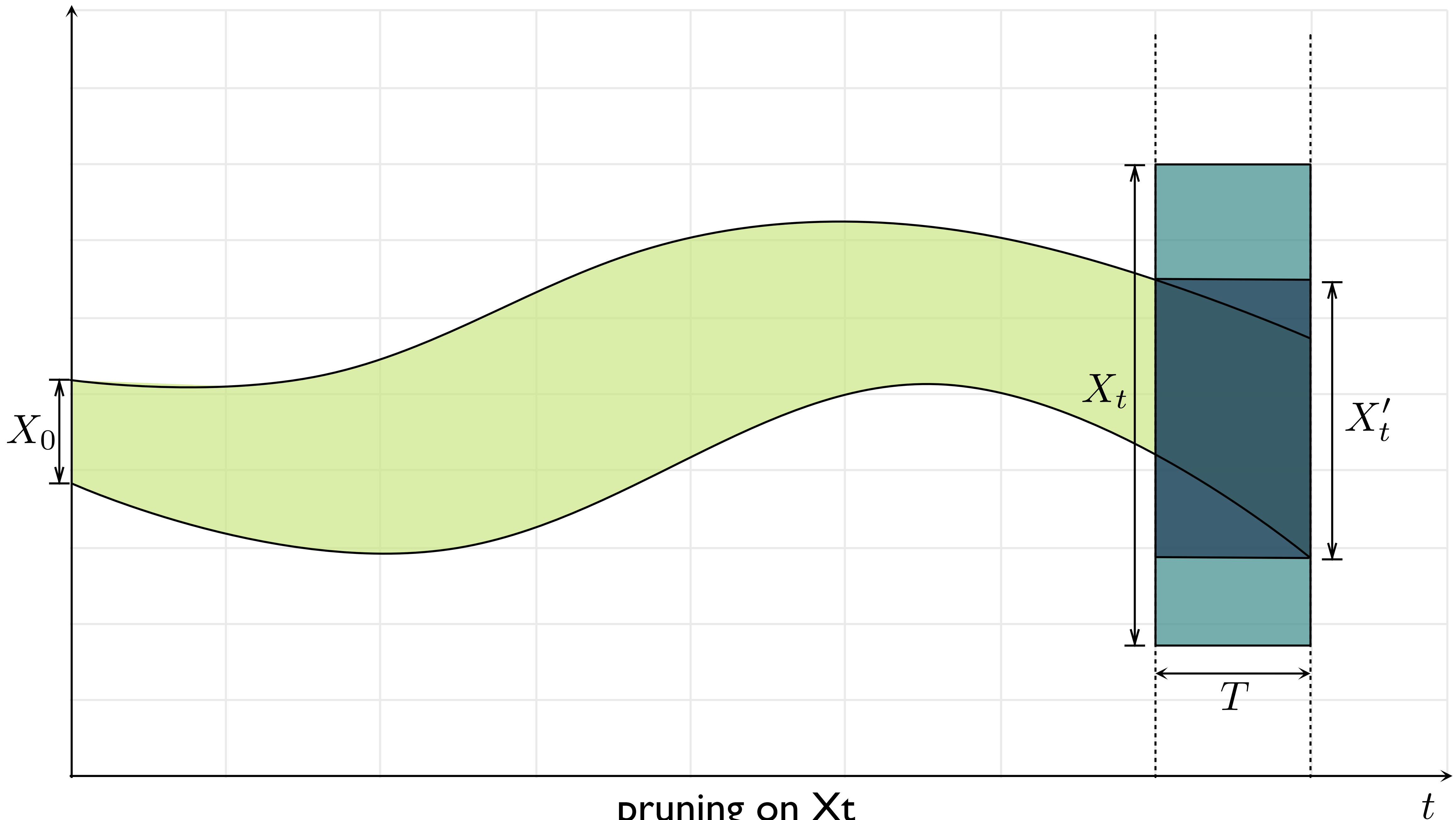
pruning on Xt
(Forward)

Pruning using ODEs



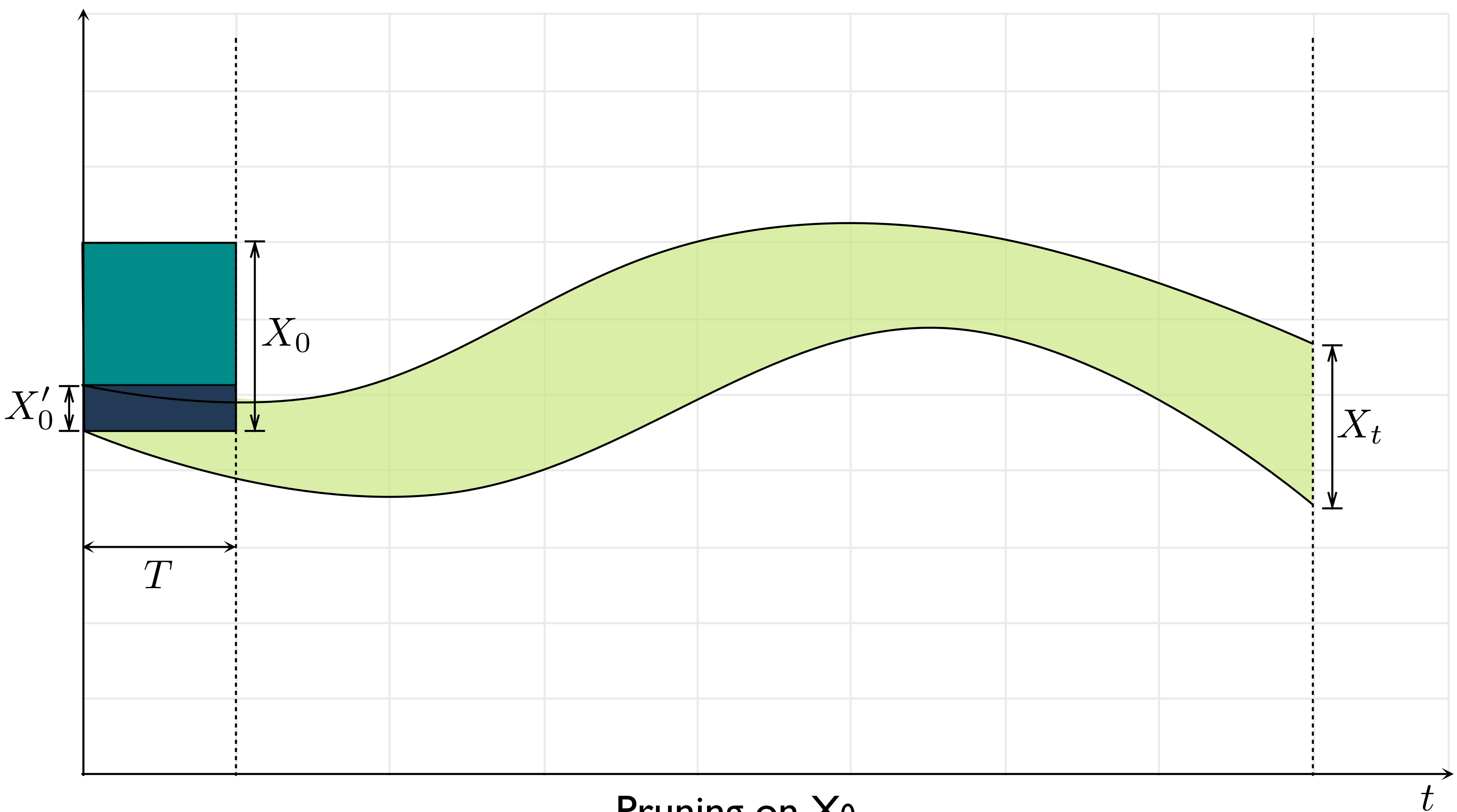
pruning on Xt
(Forward)

Pruning using ODEs



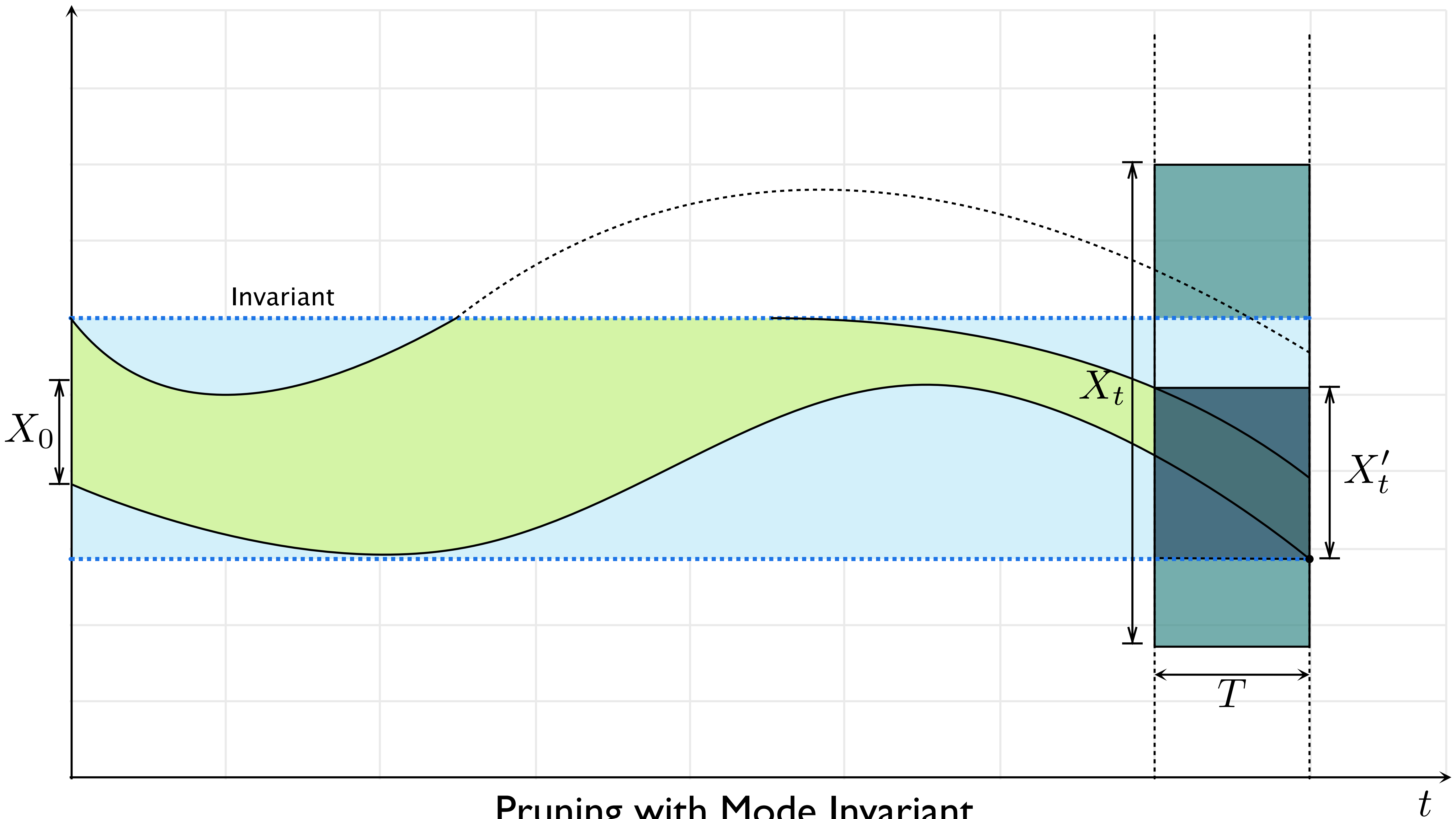
pruning on Xt
(Forward)

Pruning using ODEs



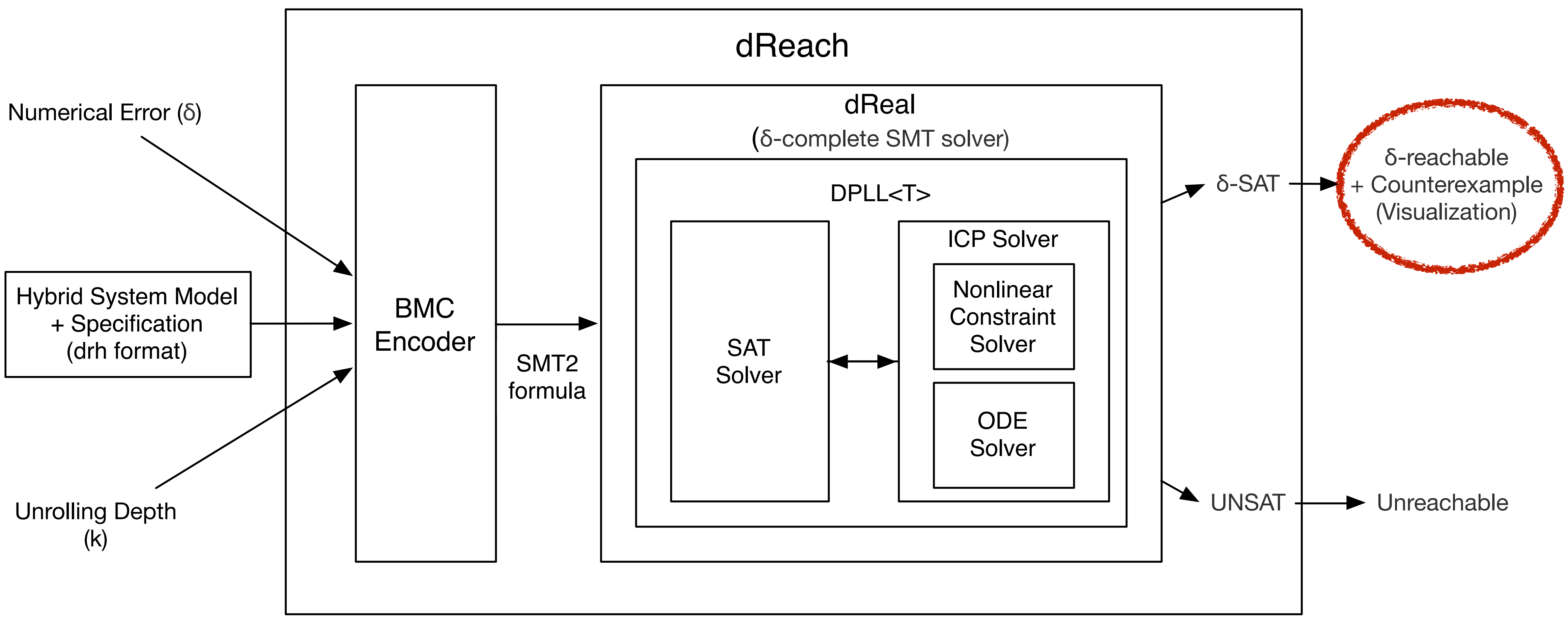
Pruning on X0
(Backward)

Pruning using ODEs

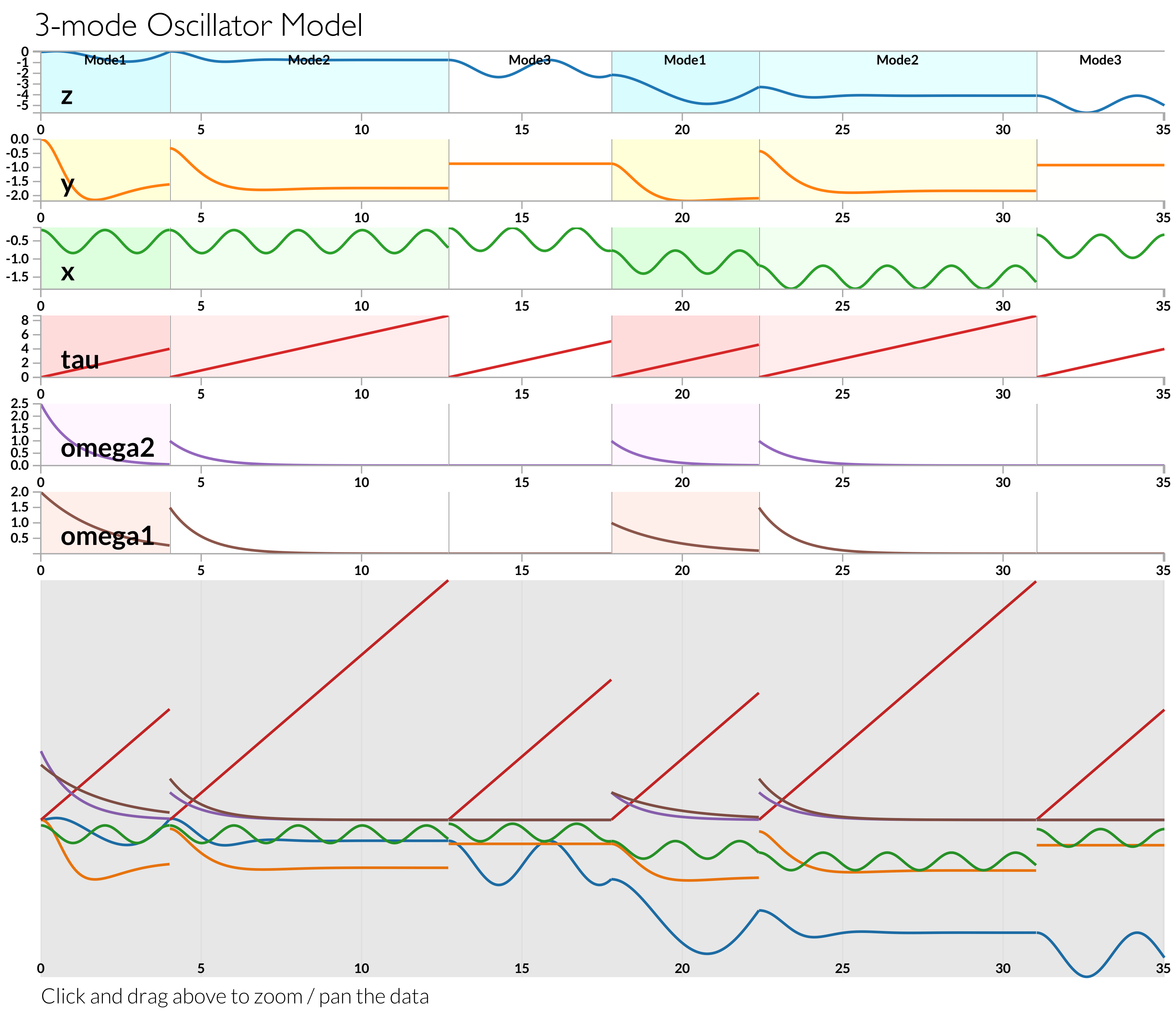


Pruning with Mode Invariant

Visualization of Counterexample



Visualization of Counterexample



Demo
(1 min)

Fork me on GitHub

Thank You

See You @ Tool Market (16:30-18:00, Octagon)

http://dreal.github.io