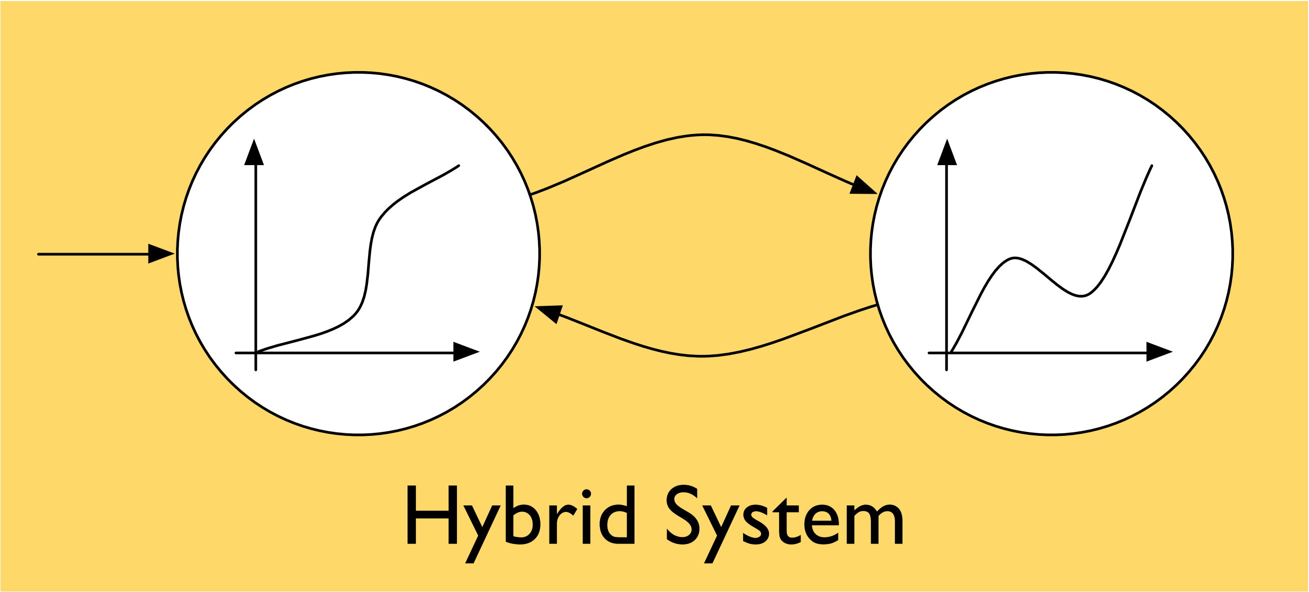
δ-Reachability Analysis for Hybrid Systems

Soonho Kong

soonhok@cs.cmu.edu
Carnegie Mellon University
Computer Science Department

Hybrid Systems

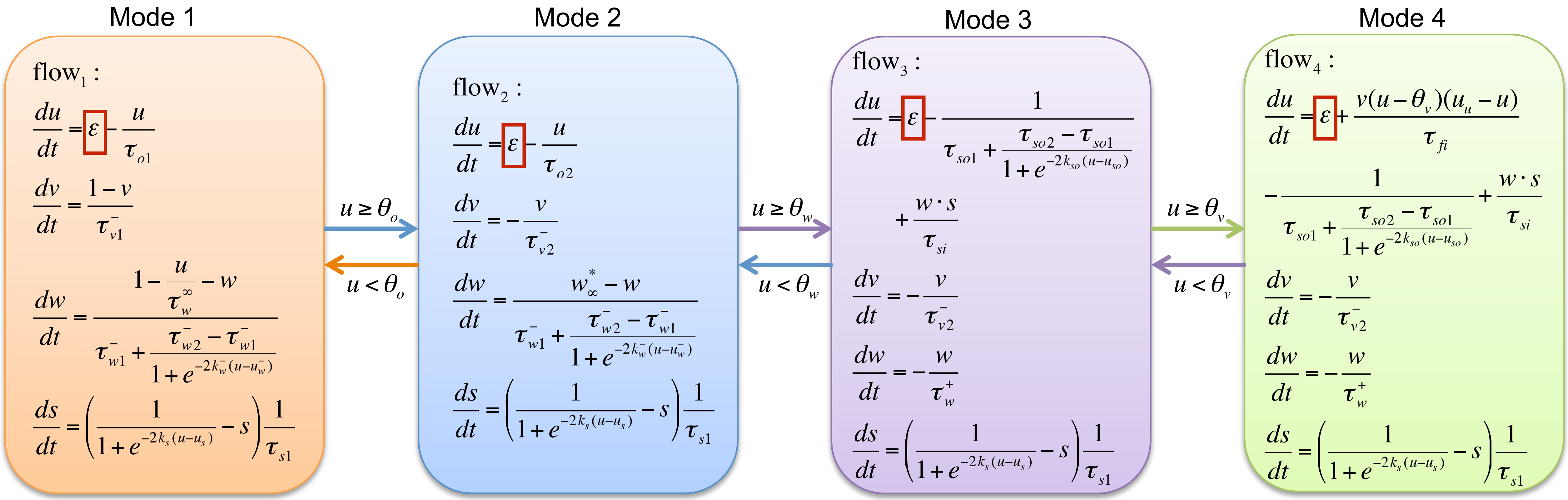


Discrete Control + Continuous Dynamics

Hybrid Systems

Discrete Control + Continuous Dynamics

Cardiac Cell Action Potential Model (BCF Model[Bueno2008])



Mode 1

$\text{flow}_1:$

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

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

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

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

$u \geq \theta_o$

$u < \theta_o$

Mode 2

$\text{flow}_2:$

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

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

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

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

$u \geq \theta_w$

$u < \theta_w$

Mode 3

$\text{flow}_3:$

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

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

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

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

$u \geq \theta_v$

$u < \theta_v$

Mode 4

$\text{flow}_4:$

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

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

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

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

State Variables

- u: cell's transmembrane potential
- v: current at fast channel gate
- w, s: currents at two slow channel gate

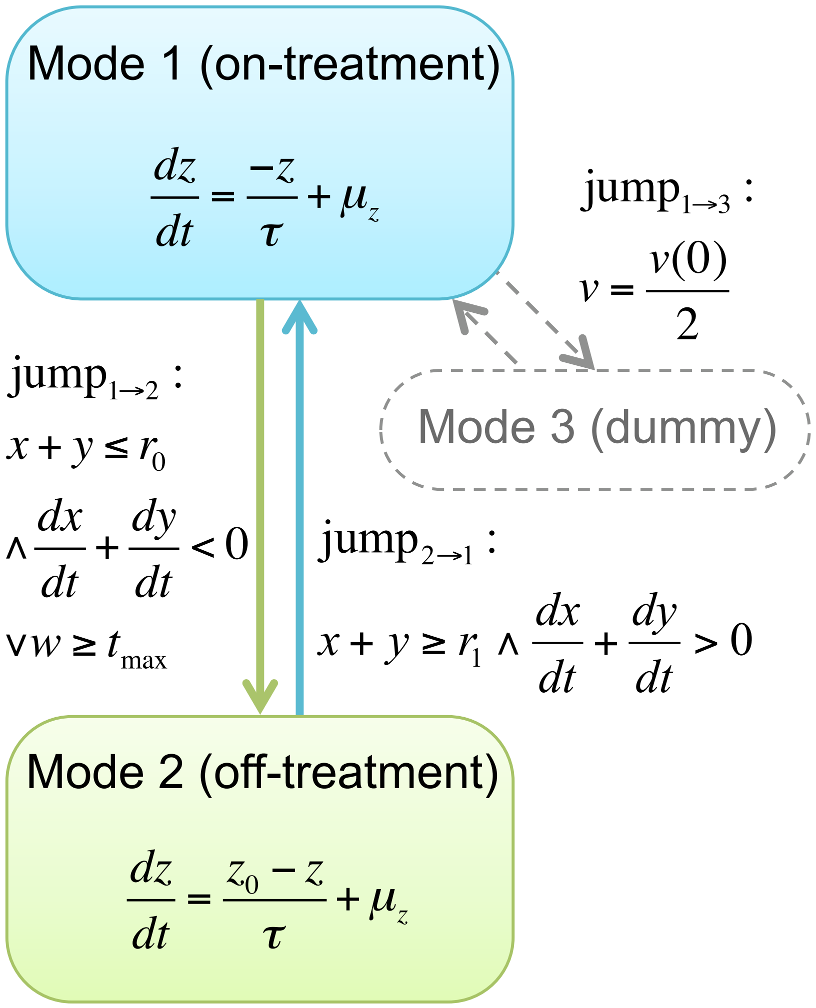
Parameters

- **ε**: external stimulus current to cell

Hybrid Systems

Discrete Control + Continuous Dynamics

Prostate Cancer Treatment Model [Bing2015]



Mode 1 (on-treatment)

$$\frac{dz}{dt} = \frac{-z}{\tau} + \mu_z$$

$\text{jump}_{1\to 3}$:

$$v = \frac{v(0)}{2}$$

Mode 3 (dummy)

$\text{jump}_{1\to 2}$:

$$x + y \le r_0$$

$$\wedge \frac{dx}{dt} + \frac{dy}{dt} < 0$$

$$\vee w \ge t_{\max}$$

$\text{jump}_{2\to 1}$:

$$x + y \ge r_1 \wedge \frac{dx}{dt} + \frac{dy}{dt} > 0$$

Mode 2 (off-treatment)

$$\frac{dz}{dt} = \frac{z_0 - z}{\tau} + \mu_z$$

$\|$

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

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

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

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

Control (cancer therapy)

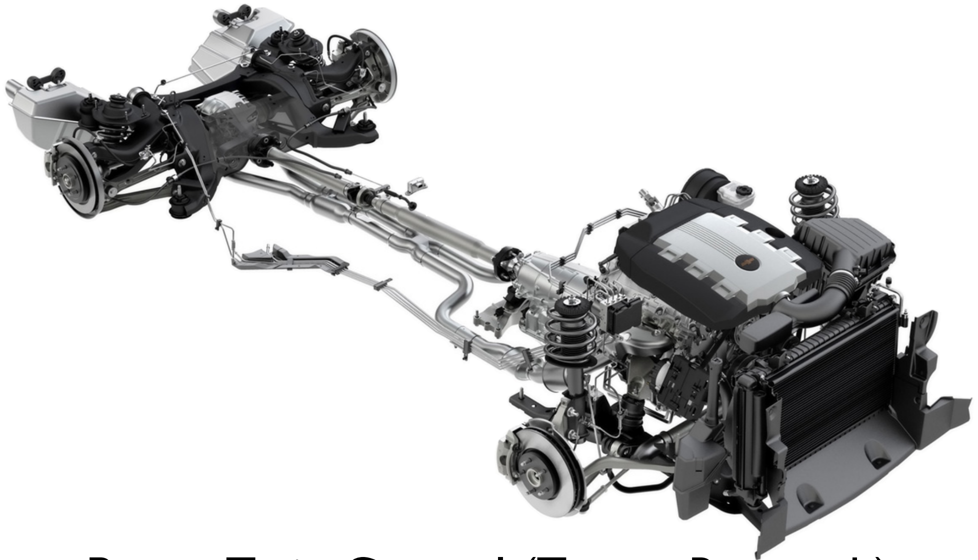
Plant (cancer progression)

x: Population of HSCs (Hormone Sensitive Cells)
y: Population of CRCs (Cancer Resistant Cells)

z: Androgen concentration
v: PSV level, biomarker for the total population of cancer cells

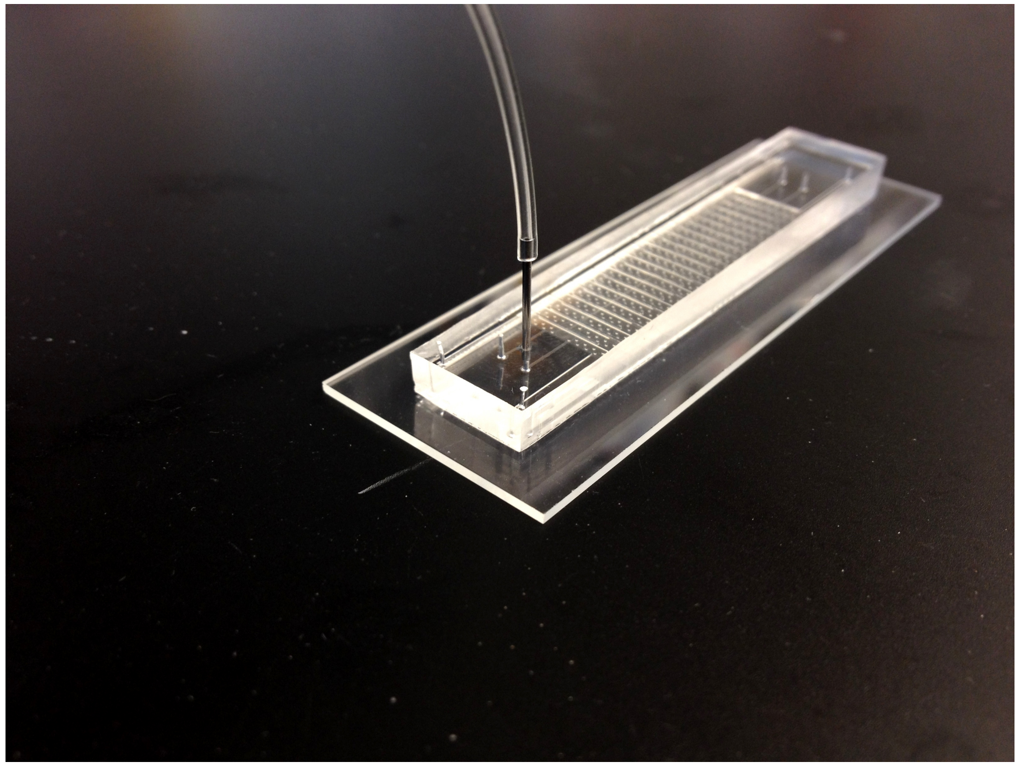
Hybrid Systems

Quadcopter Control (CMU)



Power Train Control (Toyota Research)

Autonomous Vehicle (CMU ECE)



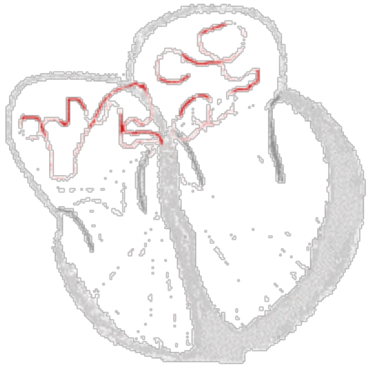
Microfluidic Chip Design (Univ. of Waterloo)

Reachability Analysis of Hybrid Systems

Reachability Analysis of Hybrid Systems

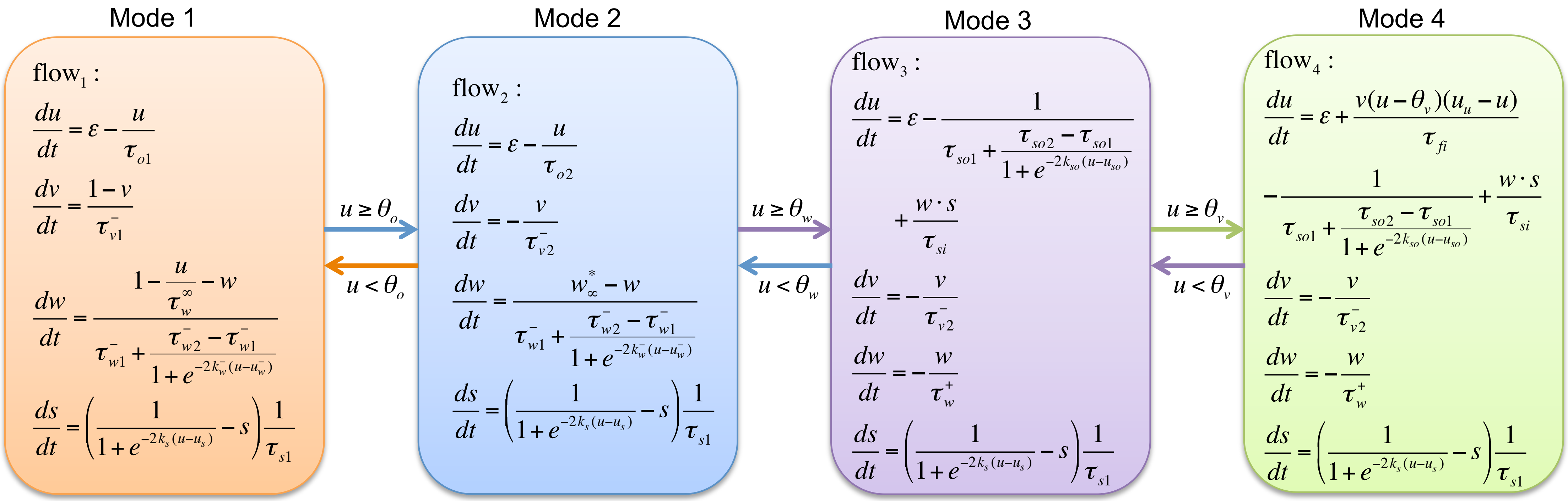
Improper functioning of cardiac cell ionic channels can cause the cells to lose excitability, which disorders electric wave propagation and leads to cardiac abnormalities such as ventricular *tachycardia* or *fibrillation*.

Normal Condition



Atrial Fibrillation

Cardiac Cell Action Potential



Mode 1

flow$_1$:

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

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

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

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

$u \geq \theta_o$

$u < \theta_o$

Mode 2

flow$_2$:

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

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

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

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

$u \geq \theta_w$

$u < \theta_w$

Mode 3

flow$_3$:

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

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

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

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

$u \geq \theta_v$

$u < \theta_v$

Mode 4

flow$_4$:

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

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

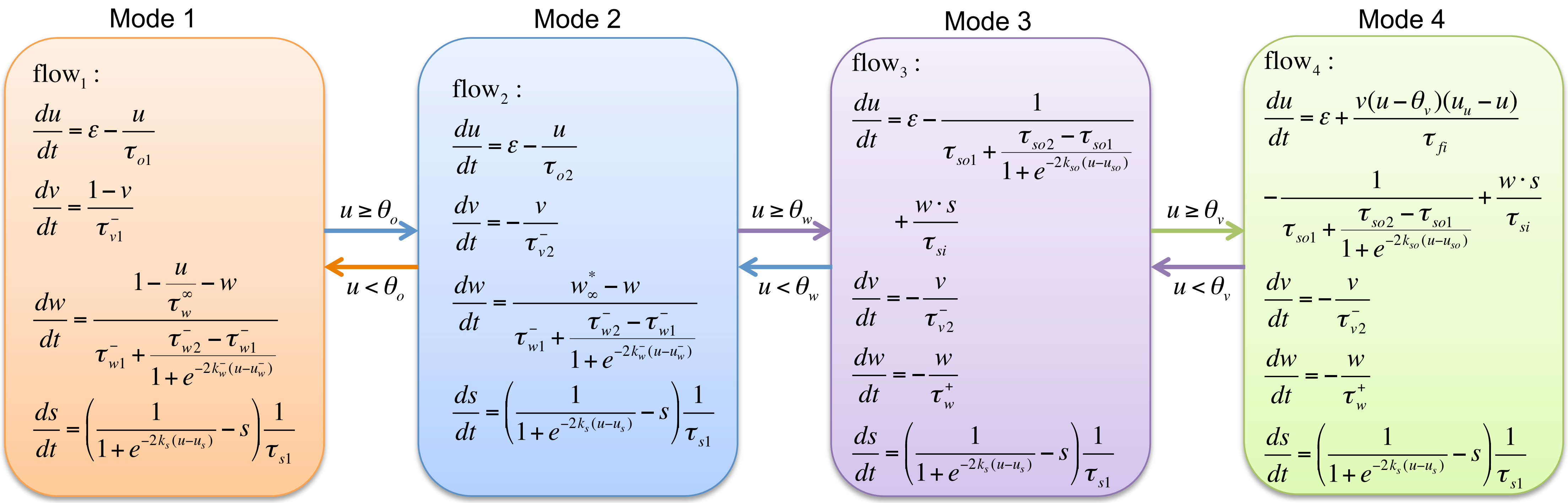
$$\frac{dw}{dt} = -\frac{w}{\tau_w^{+}}$$

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

Reachability Analysis of Hybrid Systems

Can we find a set of initial values/parameters for which a cardiac cell loses excitability? (= can it reach mode 4?)

Cardiac Cell Action Potential



Mode 1

flow$_1$:

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

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

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

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

$u \geq \theta_o$

$u < \theta_o$

Mode 2

flow$_2$:

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

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

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

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

$u \geq \theta_w$

$u < \theta_w$

Mode 3

flow$_3$:

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

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

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

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

$u \geq \theta_v$

$u < \theta_v$

Mode 4

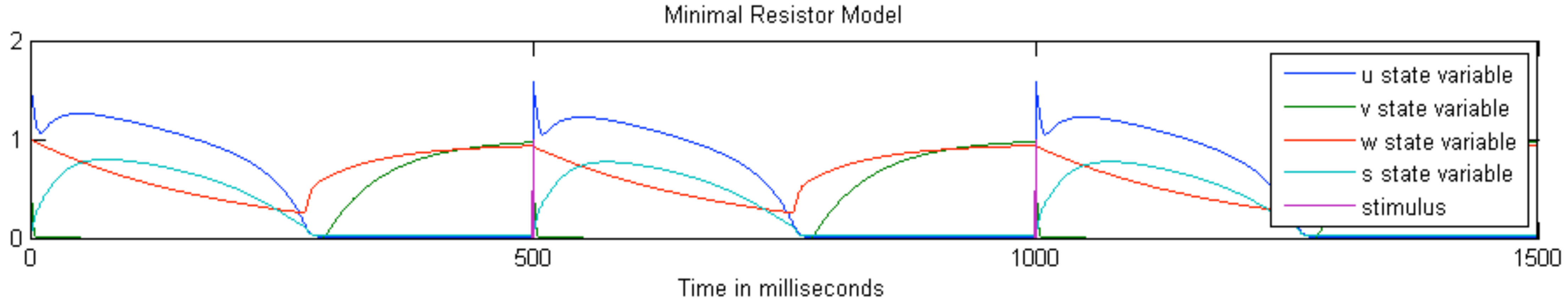
flow$_4$:

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

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

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

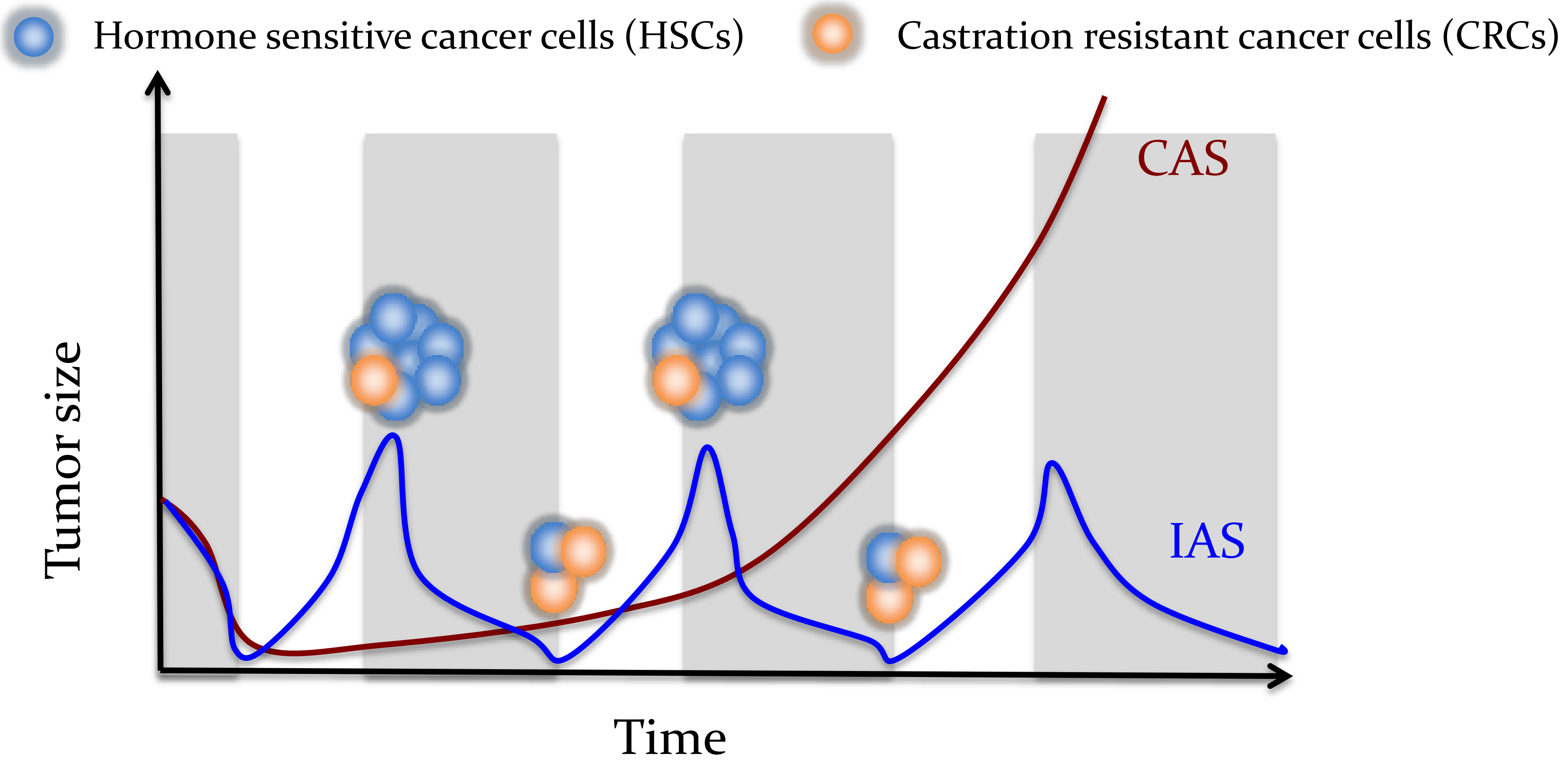
$$\frac{ds}{dt} = \left(\frac{1}{1+e^{-2k_s(u-u_s)}} - s\right)\frac{1}{\tau_{s1}}$$



Minimal Resistor Model

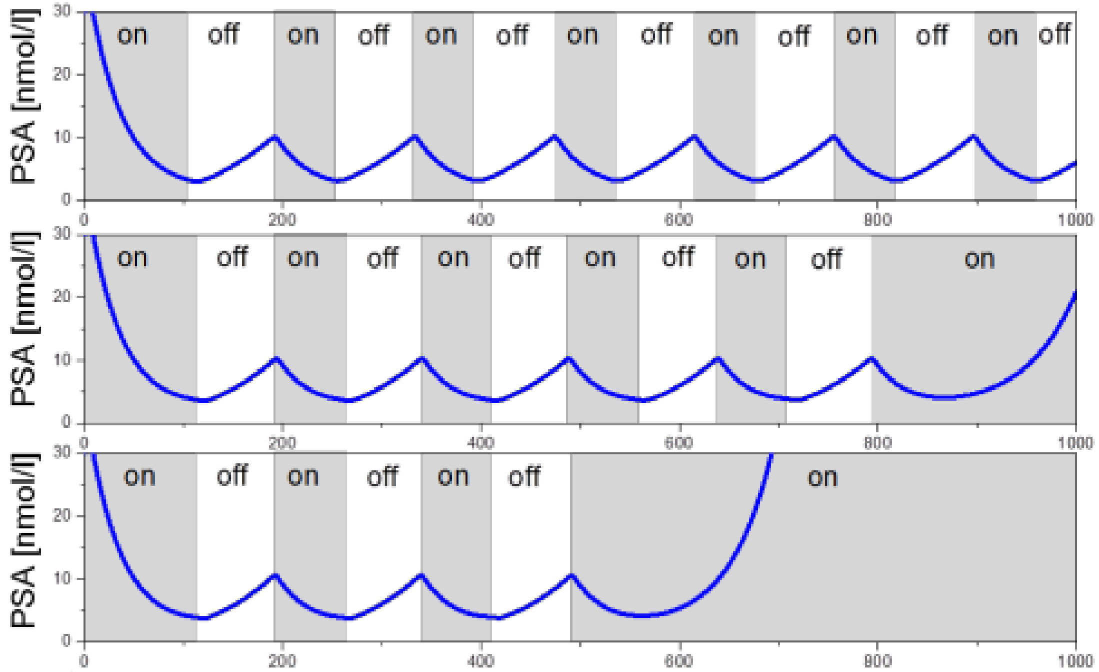
Reachability Analysis of Hybrid Systems

- CAS(Continuous Androgen Suppression) is known to fail.
- IAS(Intermittent Androgen Suppression) works better



Reachability Analysis of Hybrid Systems

- CAS(Continuous Androgen Suppression) is known to fail.
- IAS(Intermittent Androgen Suppression) works better
- However, we need to identify patient-specific schedules.



Patient#A avoided ($r_0 = 4$ and $r_1 = 10$)

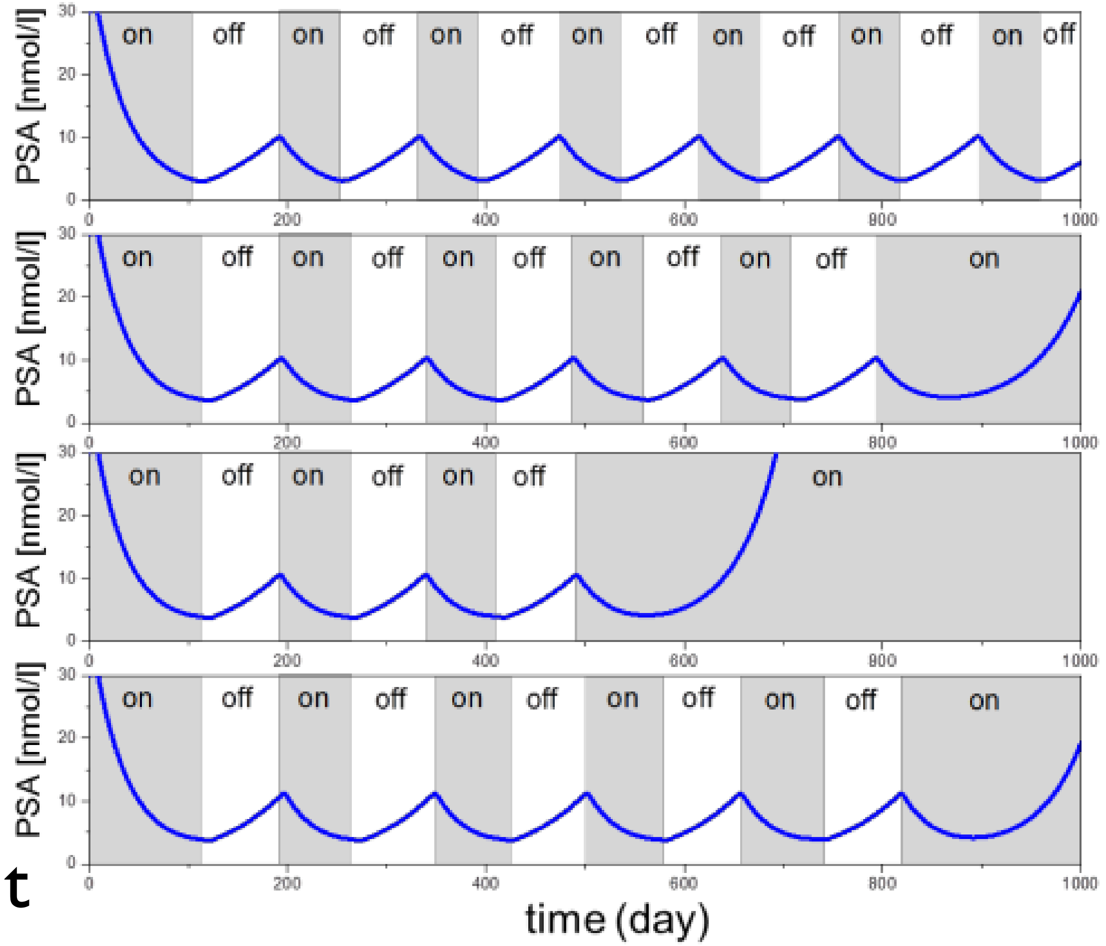
Patient#B delayed ($r_0 = 4$ and $r_1 = 10$)

Patient#C relapse ($r_0 = 4$ and $r_1 = 10$)

Same schedule

Reachability Analysis of Hybrid Systems

- CAS(Continuous Androgen Suppression) is known to fail.
- IAS(Intermittent Androgen Suppression) works better
- However, we need to identify patient-specific schedules.



Patient#A avoided ($r_0 = 4$ and $r_1 = 10$)

Patient#B delayed ($r_0 = 4$ and $r_1 = 10$)

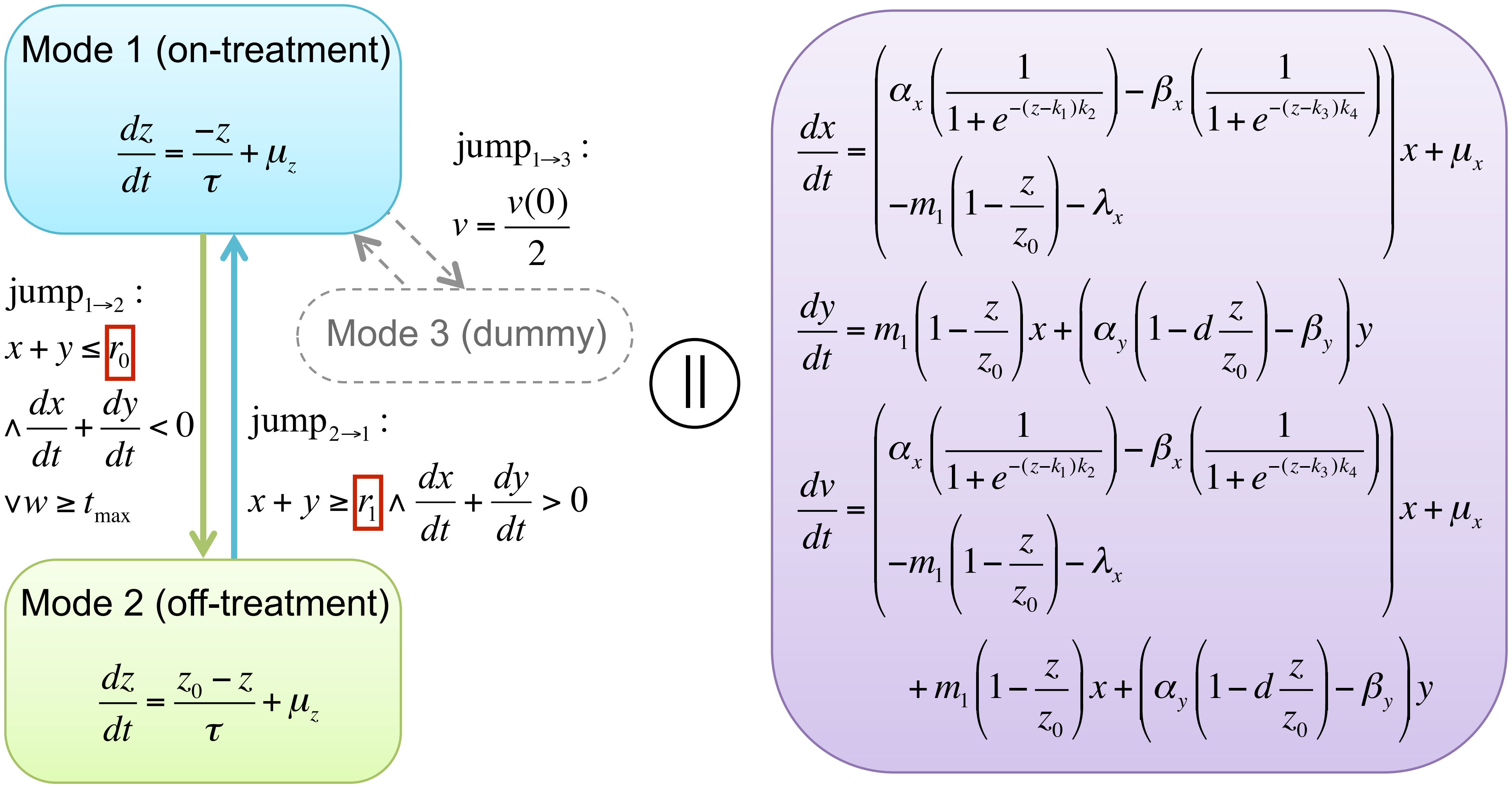
Patient#C relapse ($r_0 = 4$ and $r_1 = 10$)

Patient#C delayed ($r_0 = 5$ and $r_1 = 16$)

Same patient

Reachability Analysis of Hybrid Systems

Can we find a personalized treatment schedule which prevents the cancer recurrence in N days?



Mode 1 (on-treatment)

$$\frac{dz}{dt} = \frac{-z}{\tau} + \mu_z$$

$$\text{jump}_{1\to 3}: \quad v = \frac{v(0)}{2}$$

Mode 3 (dummy)

$$\text{jump}_{1\to 2}: \quad x + y \leq \boxed{r_0} \wedge \frac{dx}{dt} + \frac{dy}{dt} < 0 \vee w \geq t_{\max}$$

$$\text{jump}_{2\to 1}: \quad x + y \geq \boxed{r_1} \wedge \frac{dx}{dt} + \frac{dy}{dt} > 0$$

Mode 2 (off-treatment)

$$\frac{dz}{dt} = \frac{z_0 - z}{\tau} + \mu_z$$

$$\|$$

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

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

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

Control (cancer therapy)

Plant (cancer progression)

x: Population of HSCs (Hormone Sensitive Cells)
y: Population of CRCs (Cancer Resistant Cells)

z: Androgen concentration
v: PSV level, biomarker for the total population of cancer cells

Reachability Analysis of Hybrid Systems

Can we find a personalized treatment schedule which prevents the cancer recurrence in N days?

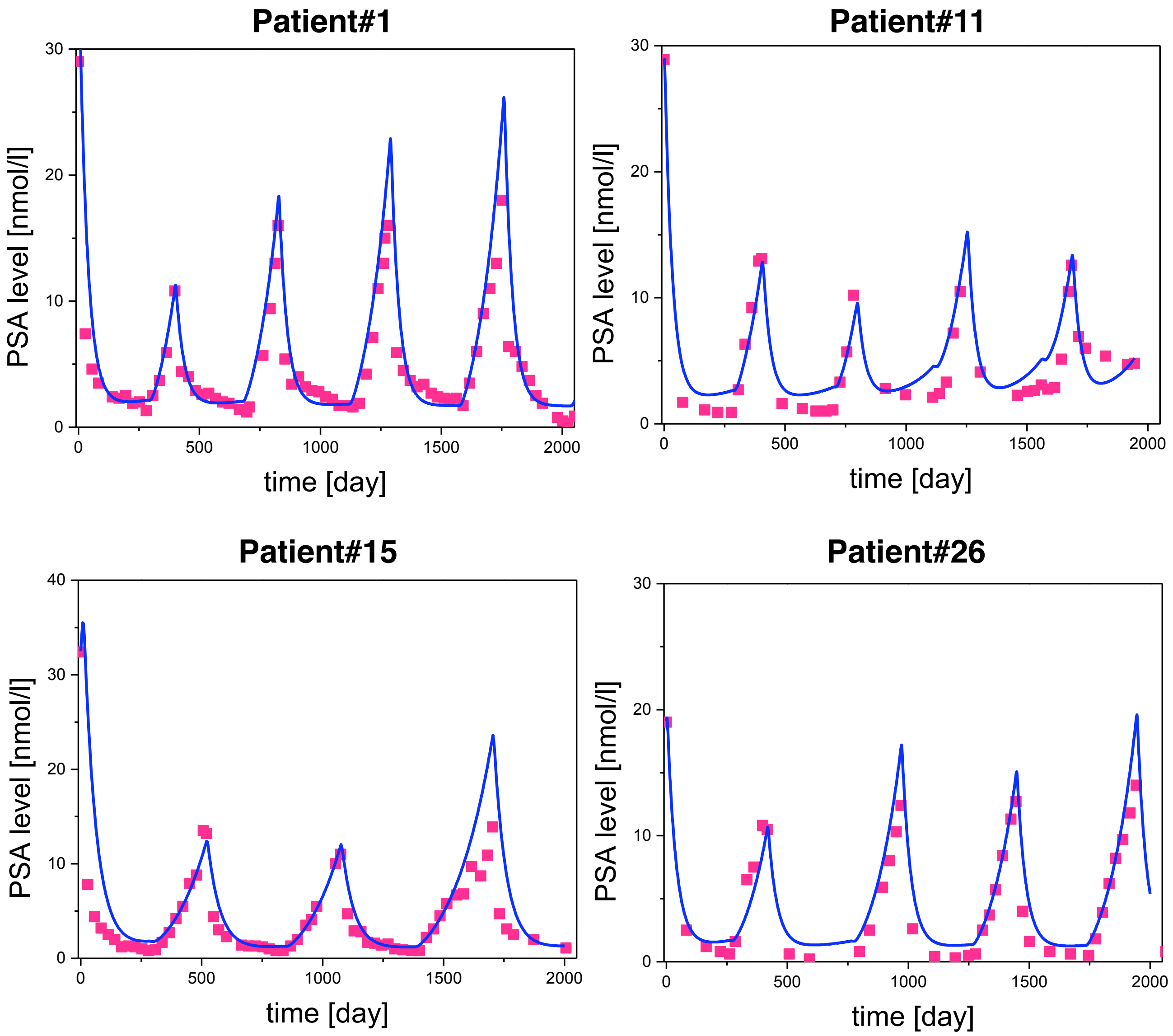


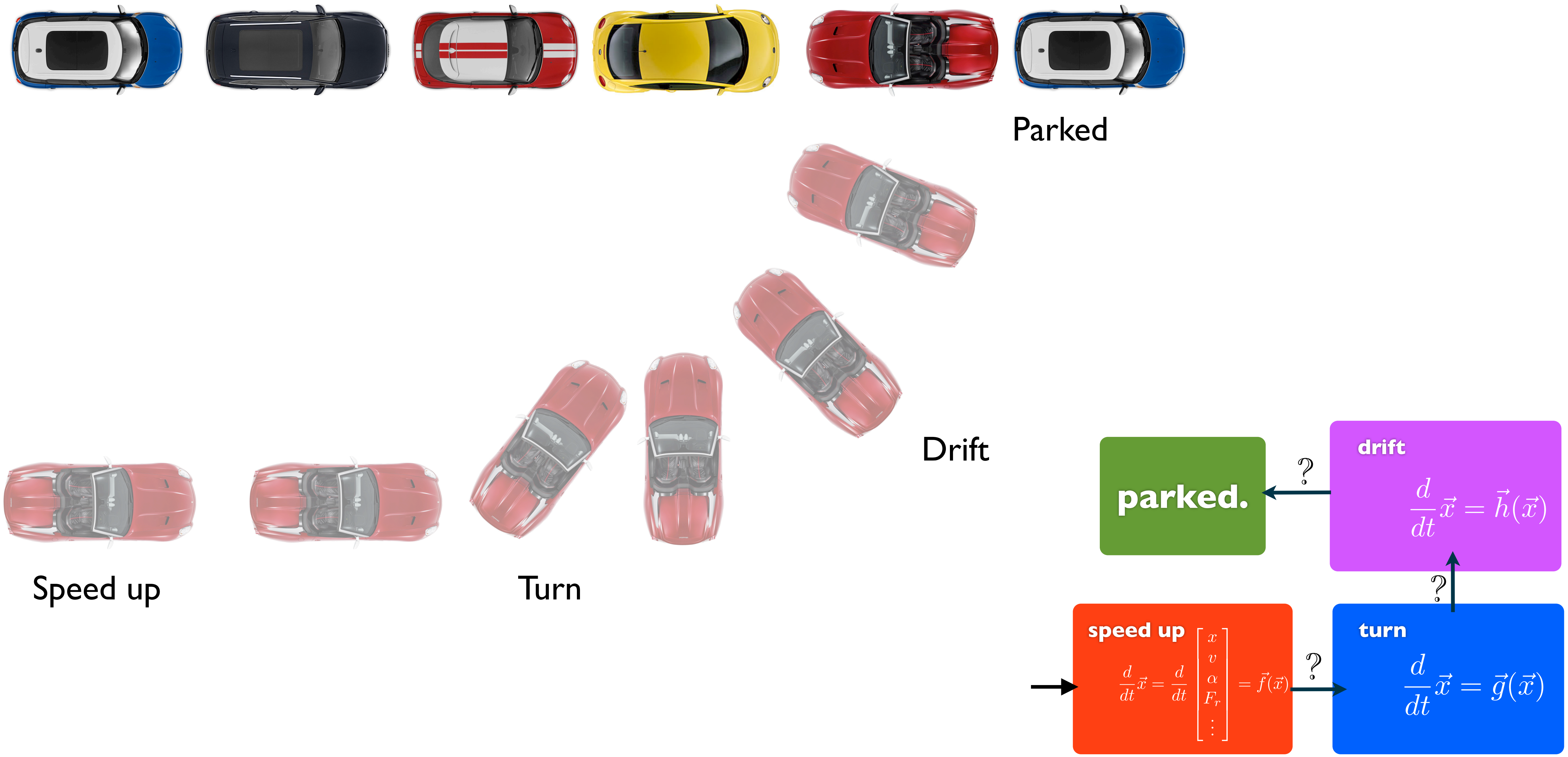
Figure 7: Model prediction vs. experimental data.

Reachability Analysis of Hybrid Systems

Can we automate a non-trivial parking?

Reachability Analysis of Hybrid Systems

Can we automate a non-trivial parking?



Reachability Analysis of Hybrid Systems

Can we automate a non-trivial parking?

Reachability Analysis of Hybrid Systems

Reachability Analysis of Hybrid Systems

"**Tesla Motors** will be rolling out an **automated car passing feature,** **but** which is **initiated by the human driver** as a way to make the latter **ultimately responsible** for the outcomes."

Matthew O'Kelly, Houssam Abbas, Sicun Gao, Shin'ichi Shiraishi, Shinpei Kato, and Rahul Mangharam ,
"APEX: A Tool for Autonomous Vehicle Plan Verification and Execution", In Society of Automotive Engineers (SAE) World Congress and Exhibition 2016

Reachability Analysis of Hybrid Systems

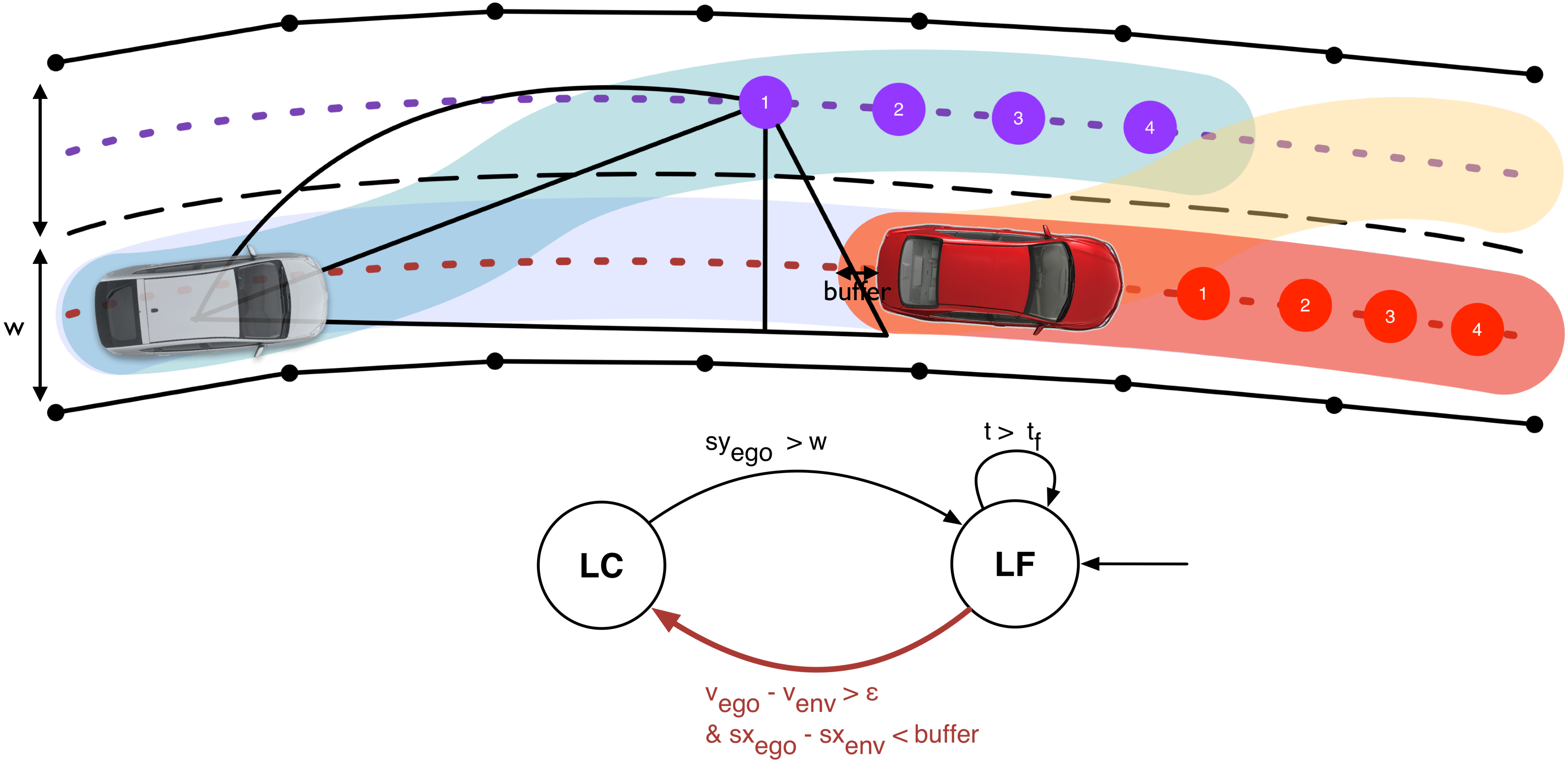
“**Tesla Motors** will be rolling out an **automated car passing feature**, but which is **initiated by the human driver** as a way to make the latter **ultimately responsible** for the outcomes.”

“**Randomized testing**, where the configurations are sampled from hypercubes of parameters, is **not a scalable solution**: suppose we decide to sample only 10 points in the range of every state variable. For our 7D model, and with 2 cars, this yields a total of 10^{14} simulations. Say we wish to simulate 10 seconds. Even if a simulation runs in real-time, this still requires $10 * 10^{14}$ seconds = 30 million years to complete.”

Matthew O'Kelly, Houssam Abbas, Sicun Gao, Shin'ichi Shiraishi, Shinpei Kato, and Rahul Mangharam ,
“APEX: A Tool for Autonomous Vehicle Plan Verification and Execution”, In Society of Automotive Engineers (SAE) World Congress and Exhibition 2016

Reachability Analysis of Hybrid Systems

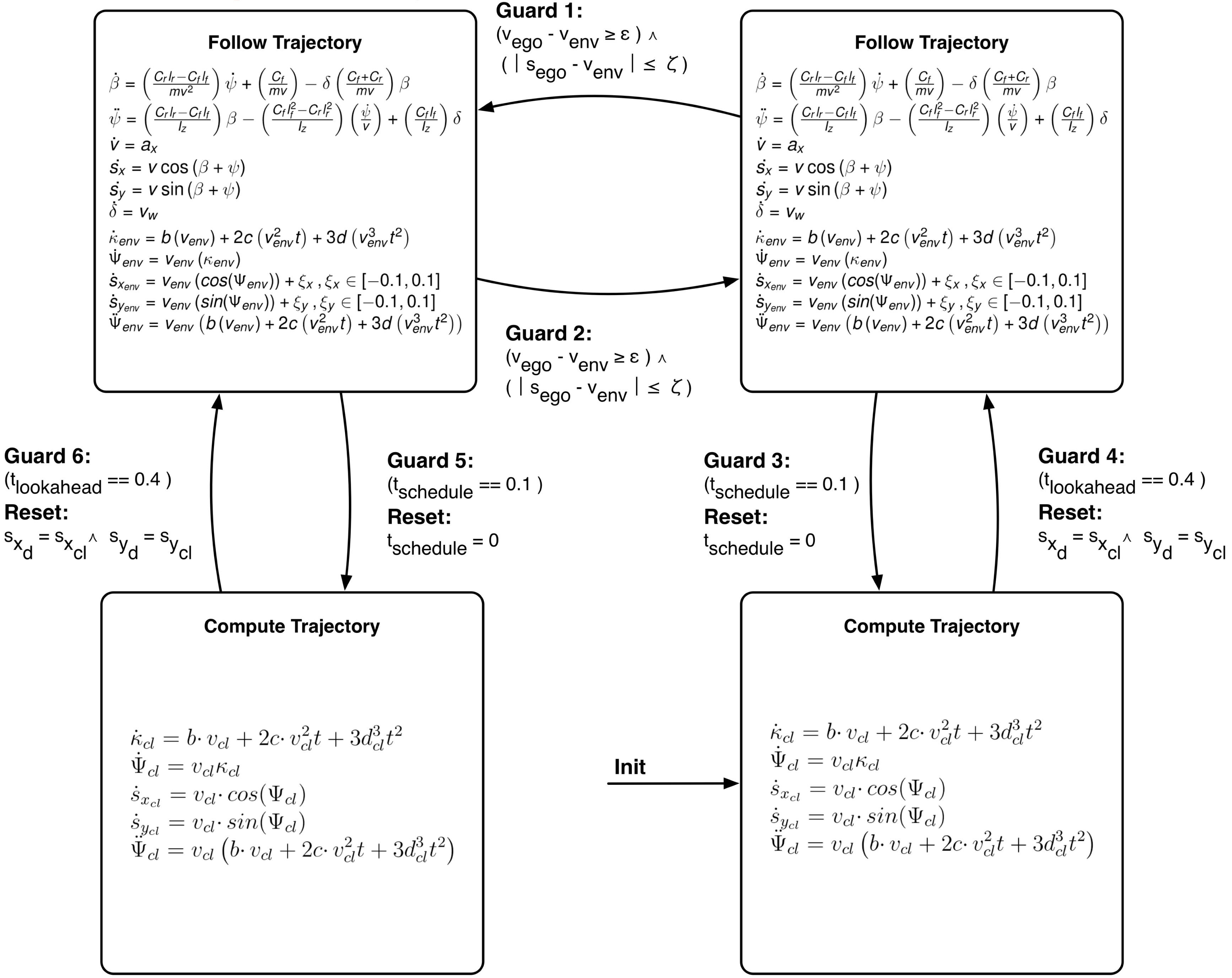
Can we do validated planning?



Matthew O'Kelly, Houssam Abbas, Sicun Gao, Shin'ichi Shiraishi, Shinpei Kato, and Rahul Mangharam ,
“APEX: A Tool for Autonomous Vehicle Plan Verification and Execution”, In Society of Automotive Engineers (SAE) World Congress and Exhibition 2016

Reachability Analysis of Hybrid Systems

Can we do validated planning?



Follow Trajectory

$$\dot{\beta} = \left(\frac{C_r l_r - C_f l_f}{mv^2}\right)\dot{\psi} + \left(\frac{C_f}{mv}\right) - \delta\left(\frac{C_f + C_r}{mv}\right)\beta$$
$$\ddot{\psi} = \left(\frac{C_r l_r - C_f l_f}{I_z}\right)\beta - \left(\frac{C_f l_f^2 - C_r l_r^2}{I_z}\right)\left(\frac{\dot{\psi}}{v}\right) + \left(\frac{C_f l_f}{I_z}\right)\delta$$
$$\dot{v} = a_x$$
$$\dot{s}_x = v\cos(\beta + \psi)$$
$$\dot{s}_y = v\sin(\beta + \psi)$$
$$\dot{\delta} = v_w$$
$$\dot{\kappa}_{env} = b(v_{env}) + 2c\left(v_{env}^2 t\right) + 3d\left(v_{env}^3 t^2\right)$$
$$\dot{\Psi}_{env} = v_{env}(\kappa_{env})$$
$$\dot{s}_{x_{env}} = v_{env}(cos(\Psi_{env})) + \xi_x, \xi_x \in [-0.1, 0.1]$$
$$\dot{s}_{y_{env}} = v_{env}(sin(\Psi_{env})) + \xi_y, \xi_y \in [-0.1, 0.1]$$
$$\ddot{\Psi}_{env} = v_{env}\left(b(v_{env}) + 2c\left(v_{env}^2 t\right) + 3d\left(v_{env}^3 t^2\right)\right)$$

Guard 1:
$(v_{ego} - v_{env} \geq \varepsilon) \wedge$
$(| s_{ego} - v_{env} | \leq \zeta)$

Follow Trajectory

$$\dot{\beta} = \left(\frac{C_r l_r - C_f l_f}{mv^2}\right)\dot{\psi} + \left(\frac{C_f}{mv}\right) - \delta\left(\frac{C_f + C_r}{mv}\right)\beta$$
$$\ddot{\psi} = \left(\frac{C_r l_r - C_f l_f}{I_z}\right)\beta - \left(\frac{C_f l_f^2 - C_r l_r^2}{I_z}\right)\left(\frac{\dot{\psi}}{v}\right) + \left(\frac{C_f l_f}{I_z}\right)\delta$$
$$\dot{v} = a_x$$
$$\dot{s}_x = v\cos(\beta + \psi)$$
$$\dot{s}_y = v\sin(\beta + \psi)$$
$$\dot{\delta} = v_w$$
$$\dot{\kappa}_{env} = b(v_{env}) + 2c\left(v_{env}^2 t\right) + 3d\left(v_{env}^3 t^2\right)$$
$$\dot{\Psi}_{env} = v_{env}(\kappa_{env})$$
$$\dot{s}_{x_{env}} = v_{env}(cos(\Psi_{env})) + \xi_x, \xi_x \in [-0.1, 0.1]$$
$$\dot{s}_{y_{env}} = v_{env}(sin(\Psi_{env})) + \xi_y, \xi_y \in [-0.1, 0.1]$$
$$\ddot{\Psi}_{env} = v_{env}\left(b(v_{env}) + 2c\left(v_{env}^2 t\right) + 3d\left(v_{env}^3 t^2\right)\right)$$

Guard 2:
$(v_{ego} - v_{env} \geq \varepsilon) \wedge$
$(| s_{ego} - v_{env} | \leq \zeta)$

Guard 6:
$(t_{lookahead} == 0.4)$
Reset:
$s_{x_d} = s_{x_{cl}} \wedge s_{y_d} = s_{y_{cl}}$

Guard 5:
$(t_{schedule} == 0.1)$
Reset:
$t_{schedule} = 0$

Guard 3:
$(t_{schedule} == 0.1)$
Reset:
$t_{schedule} = 0$

Guard 4:
$(t_{lookahead} == 0.4)$
Reset:
$s_{x_d} = s_{x_{cl}} \wedge s_{y_d} = s_{y_{cl}}$

Compute Trajectory

$$\dot{\kappa}_{cl} = b\cdot v_{cl} + 2c\cdot v_{cl}^2 t + 3d_{cl}^3 t^2$$
$$\dot{\Psi}_{cl} = v_{cl}\kappa_{cl}$$
$$\dot{s}_{x_{cl}} = v_{cl}\cdot cos(\Psi_{cl})$$
$$\dot{s}_{y_{cl}} = v_{cl}\cdot sin(\Psi_{cl})$$
$$\ddot{\Psi}_{cl} = v_{cl}\left(b\cdot v_{cl} + 2c\cdot v_{cl}^2 t + 3d_{cl}^3 t^2\right)$$

Init

Compute Trajectory

$$\dot{\kappa}_{cl} = b\cdot v_{cl} + 2c\cdot v_{cl}^2 t + 3d_{cl}^3 t^2$$
$$\dot{\Psi}_{cl} = v_{cl}\kappa_{cl}$$
$$\dot{s}_{x_{cl}} = v_{cl}\cdot cos(\Psi_{cl})$$
$$\dot{s}_{y_{cl}} = v_{cl}\cdot sin(\Psi_{cl})$$
$$\ddot{\Psi}_{cl} = v_{cl}\left(b\cdot v_{cl} + 2c\cdot v_{cl}^2 t + 3d_{cl}^3 t^2\right)$$

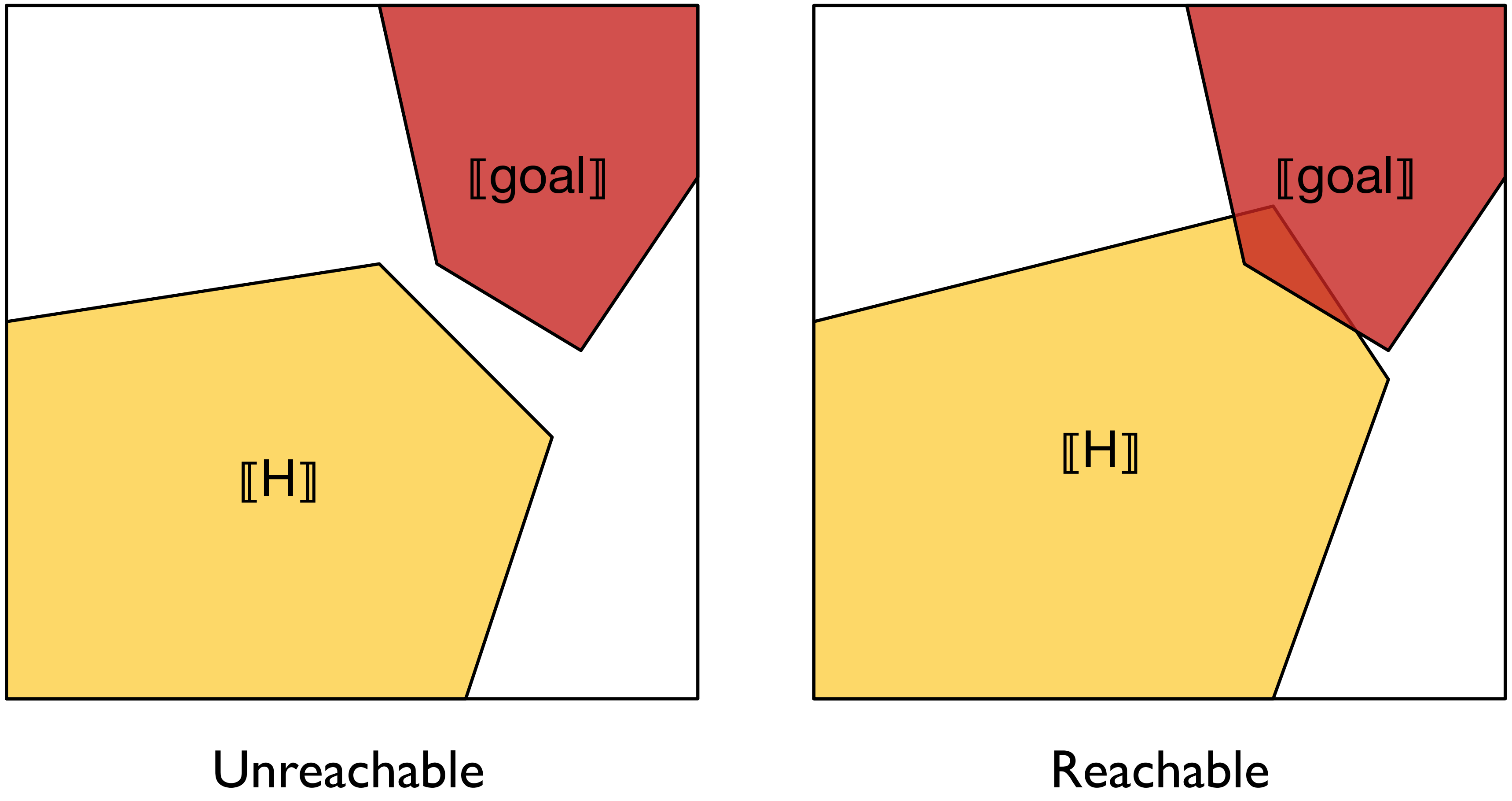
Matthew O'Kelly, Houssam Abbas, Sicun Gao, Shin'ichi Shiraishi, Shinpei Kato, and Rahul Mangharam ,
"APEX: A Tool for Autonomous Vehicle Plan Verification and Execution", In Society of Automotive Engineers (SAE) World Congress and Exhibition 2016

Reachability Analysis of Hybrid Systems

Can a hybrid system reach a **goal** region of its state space?

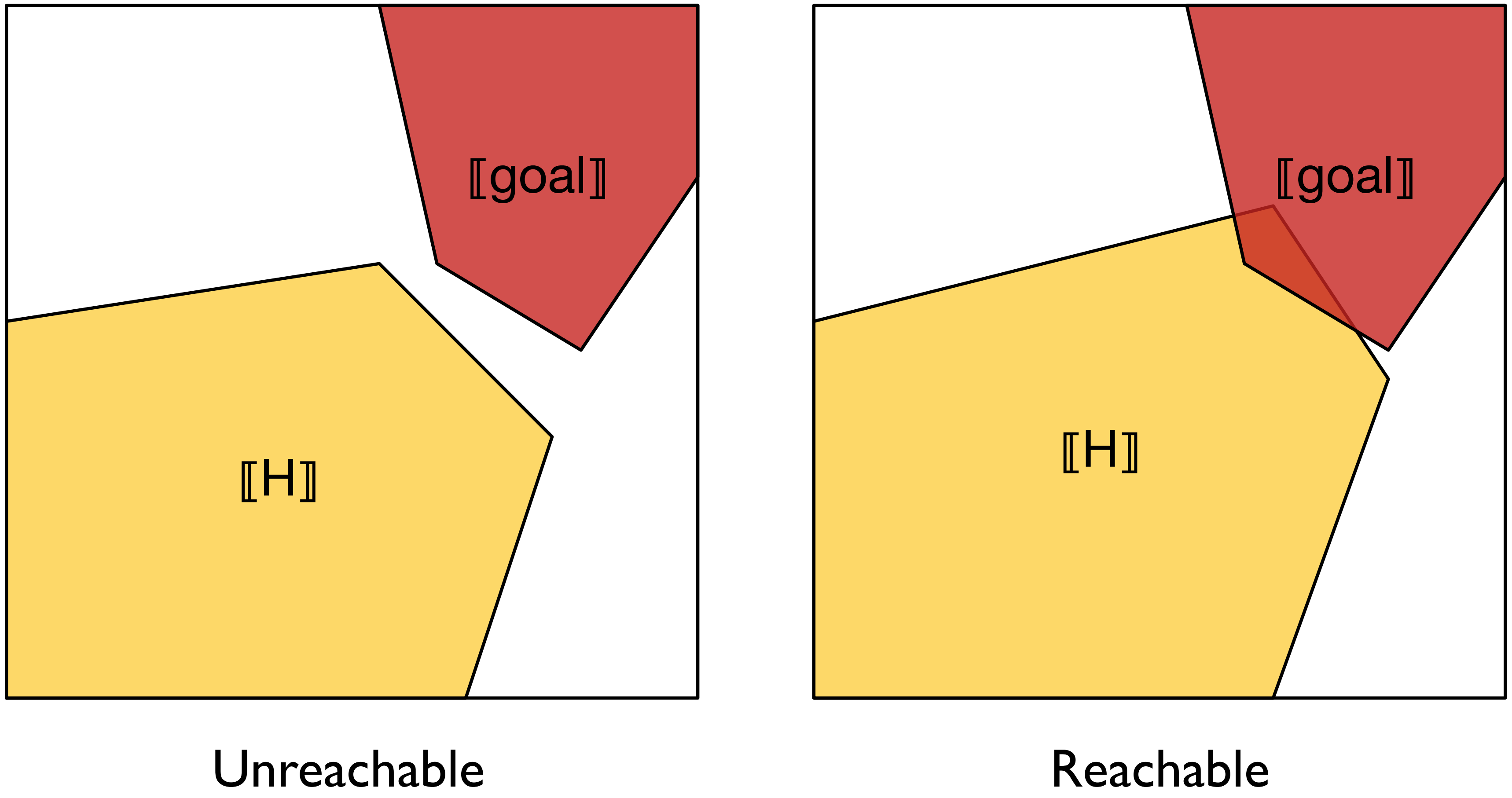
Reachability Analysis of Hybrid Systems

Can a hybrid system reach a **goal** region of its state space?



Reachability Analysis of Hybrid Systems

Can a hybrid system reach a **goal** region of its state space?



The standard bounded reachability problems for simple hybrid systems are **undecidable**[Alur et al, 1992].

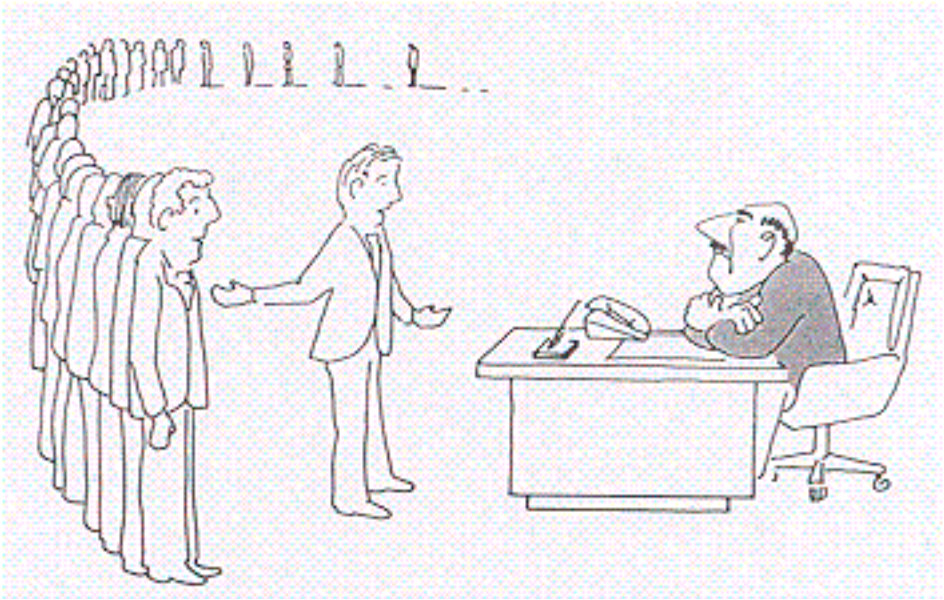
Reachability Analysis of Hybrid Systems

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



"I can't find an algorithm,
but neither can all these famous people."

Reachability Analysis of Hybrid Systems

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

1. Give up

2. Don't give Up

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

Reachability Analysis of Hybrid Systems

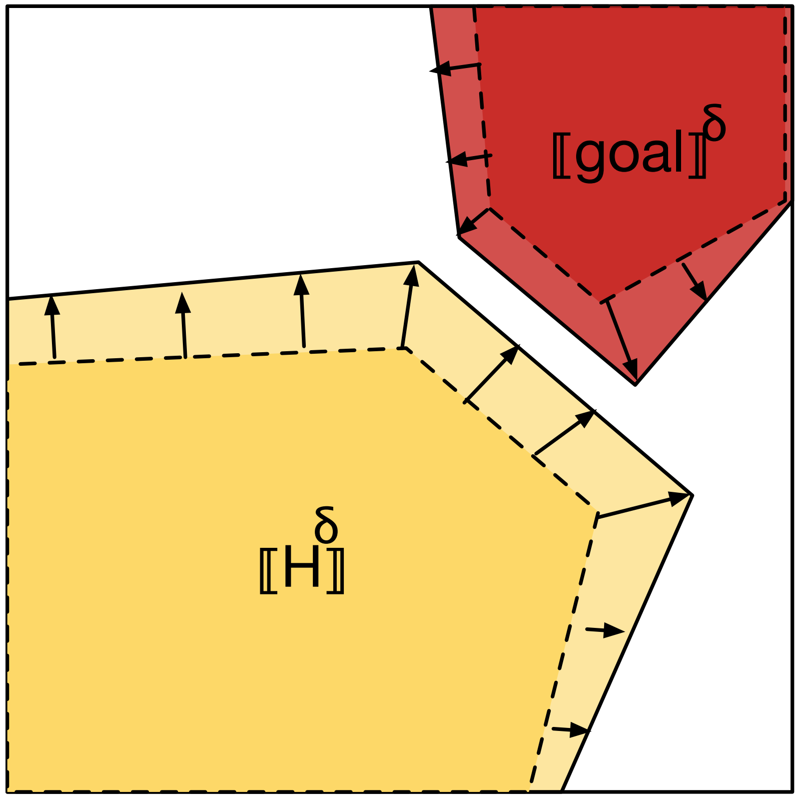
The standard bounded reachability problems for simple hybrid systems are **undecidable**.

1. Give up
2. Don't give Up
   - A. Find a decidable fragment and solve it
   - B. Use approximation

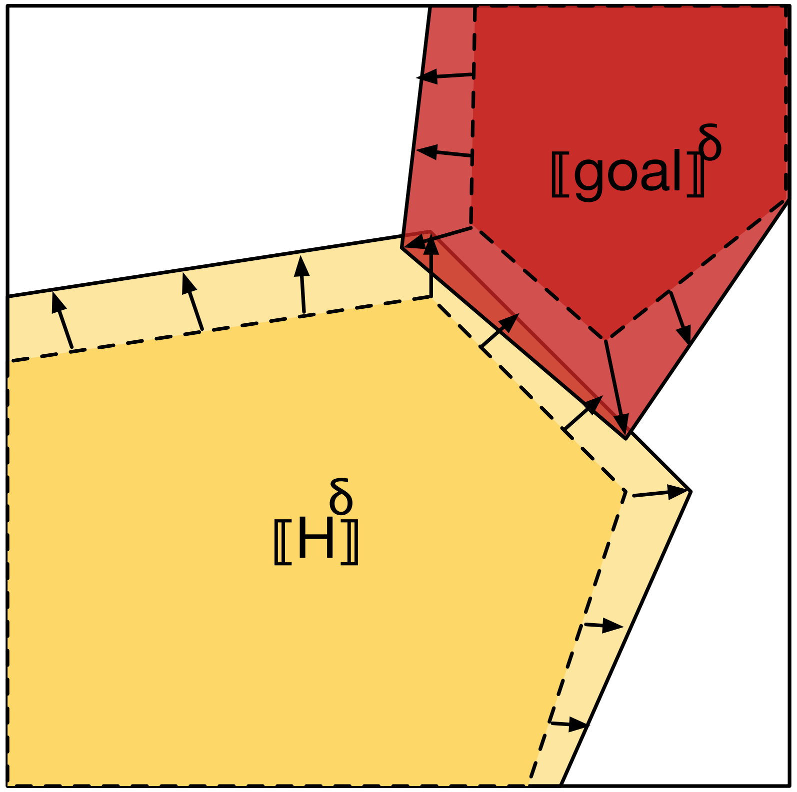
δ-Reachability Analysis of Hybrid Systems

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

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



Unreachable

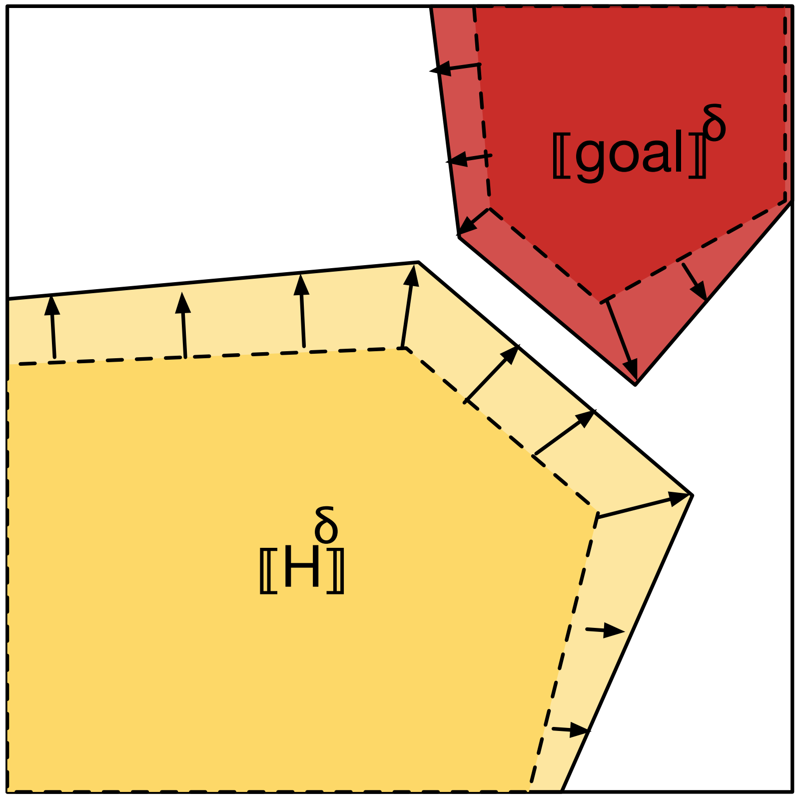


δ-reachable

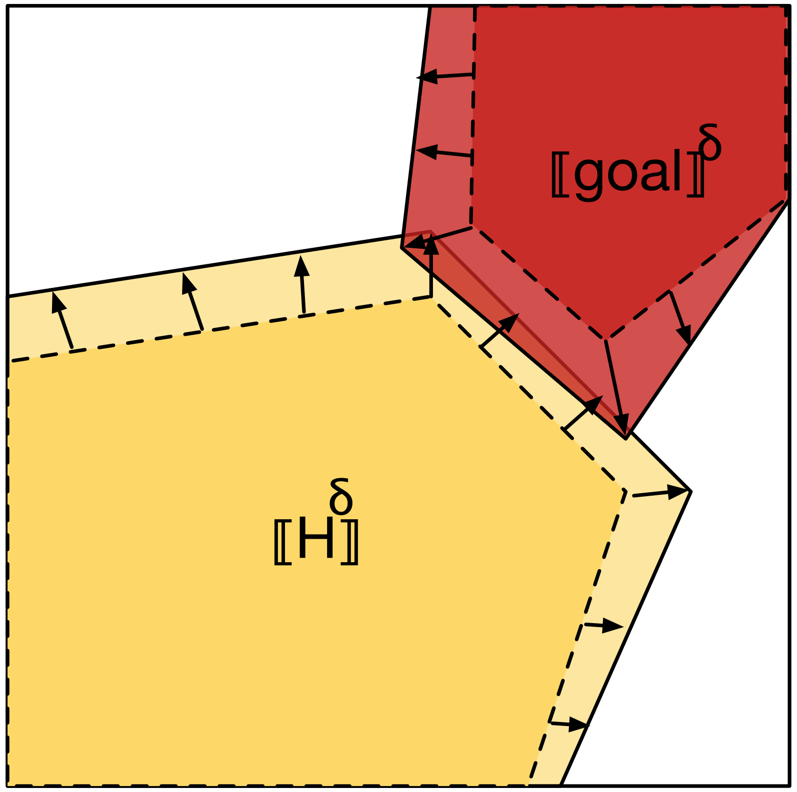
δ-Reachability Analysis of Hybrid Systems

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

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



Unreachable



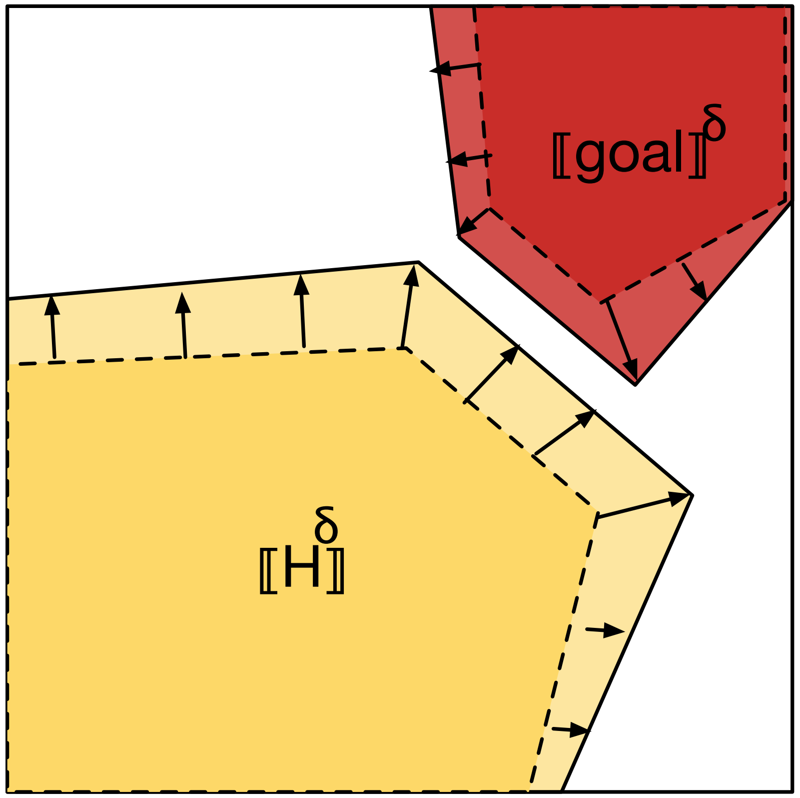
δ-reachable

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

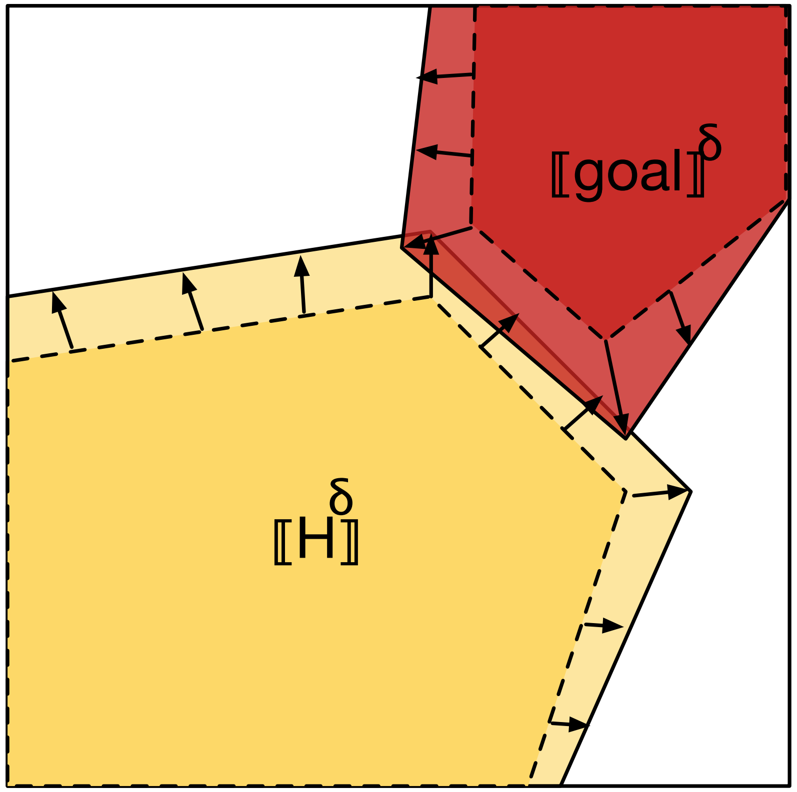
δ-Reachability Analysis of Hybrid Systems

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

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



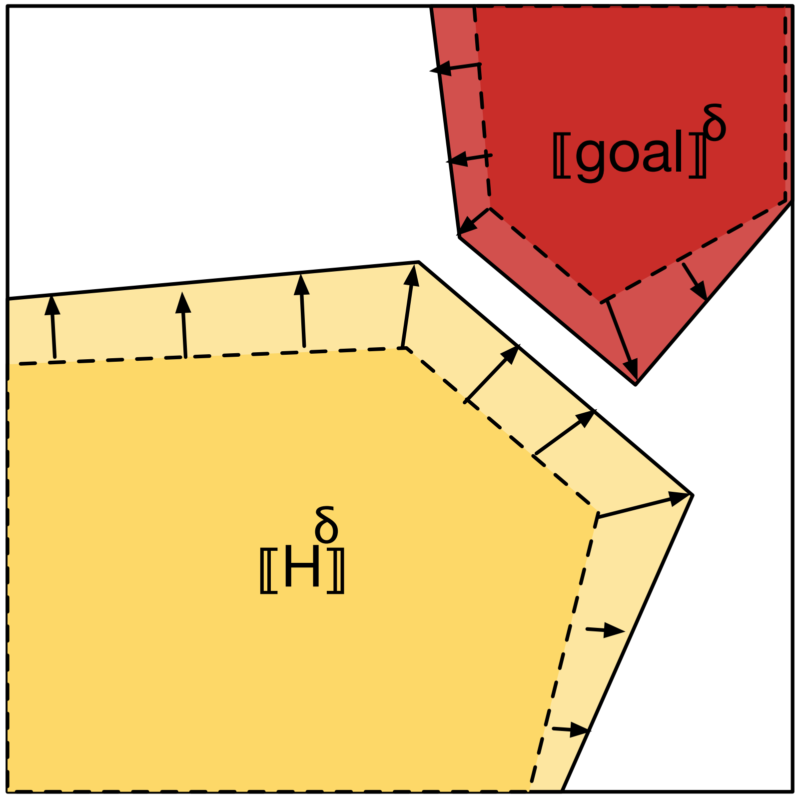
Unreachable



δ-reachable

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

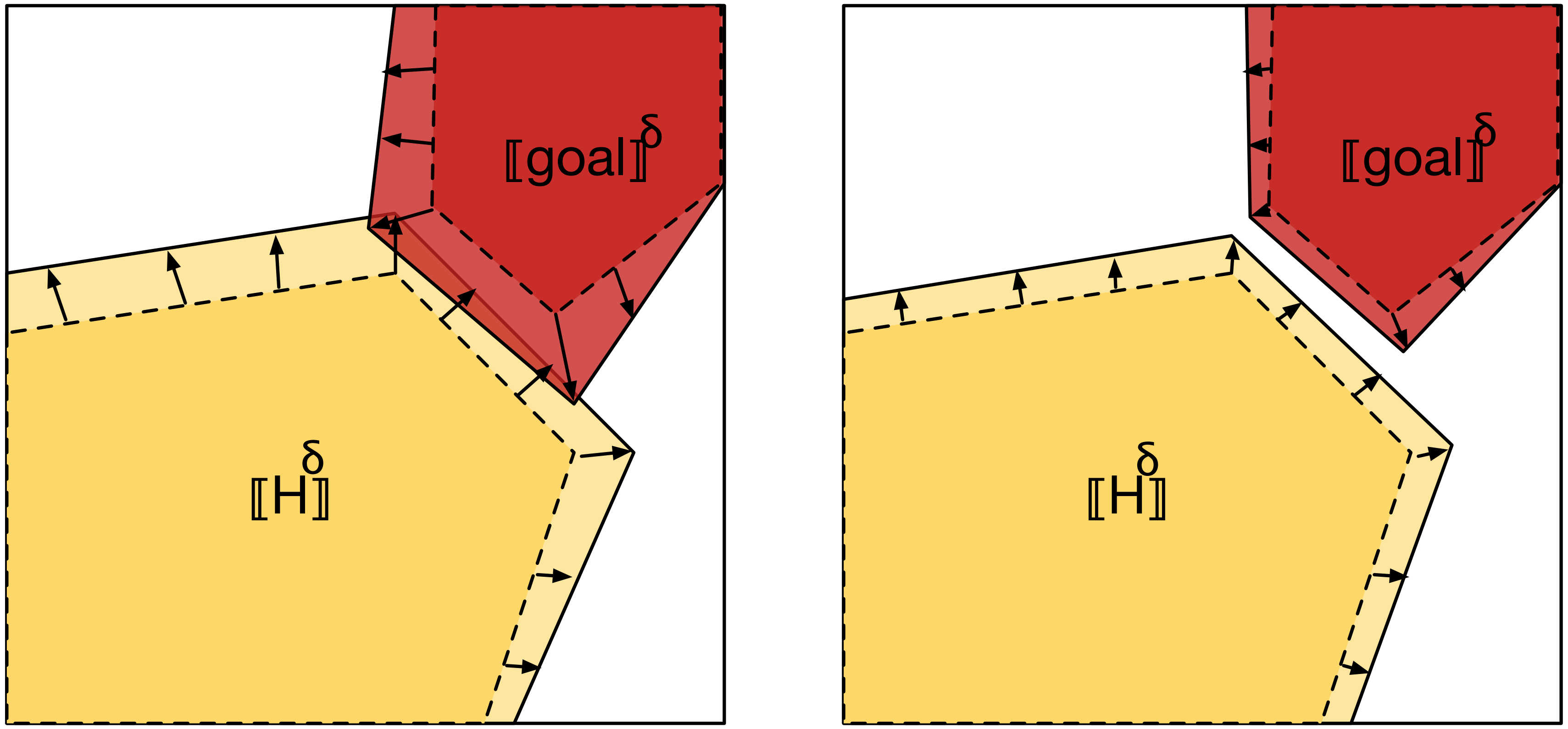
δ-Reachability Analysis of Hybrid Systems



Unreachable

1. "Unreachable" answer is **sound**.

δ-Reachability Analysis of Hybrid Systems



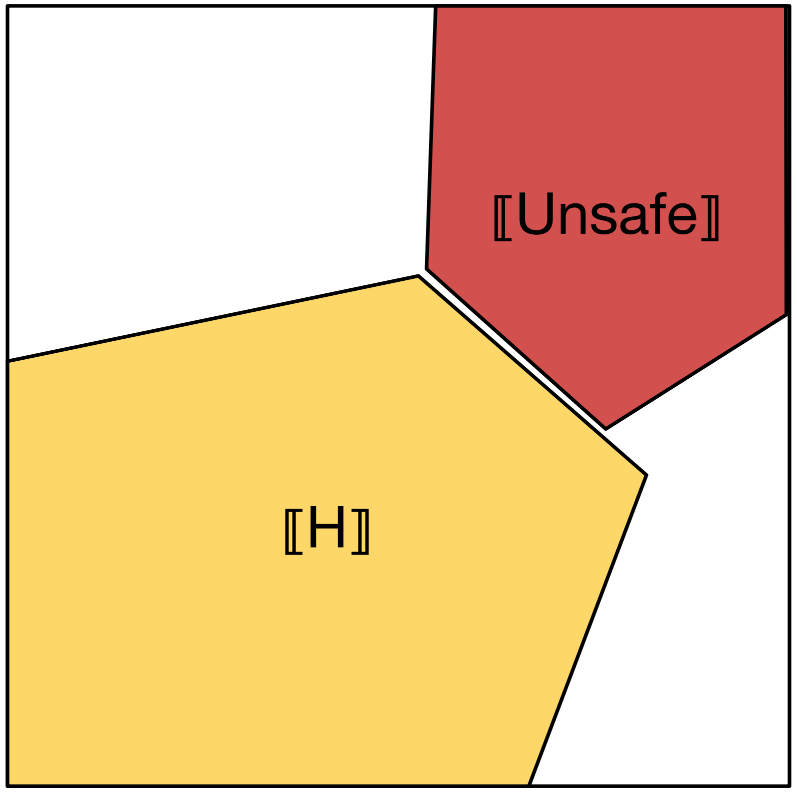
δ-reachable

Unreachable
(analyzed with smaller δ')

2. Analysis is parameterized by **δ**

If using a delta (δ) leads to an **infeasible** counterexample, you may try a **smaller delta** (δ') and possibly get rid of it.

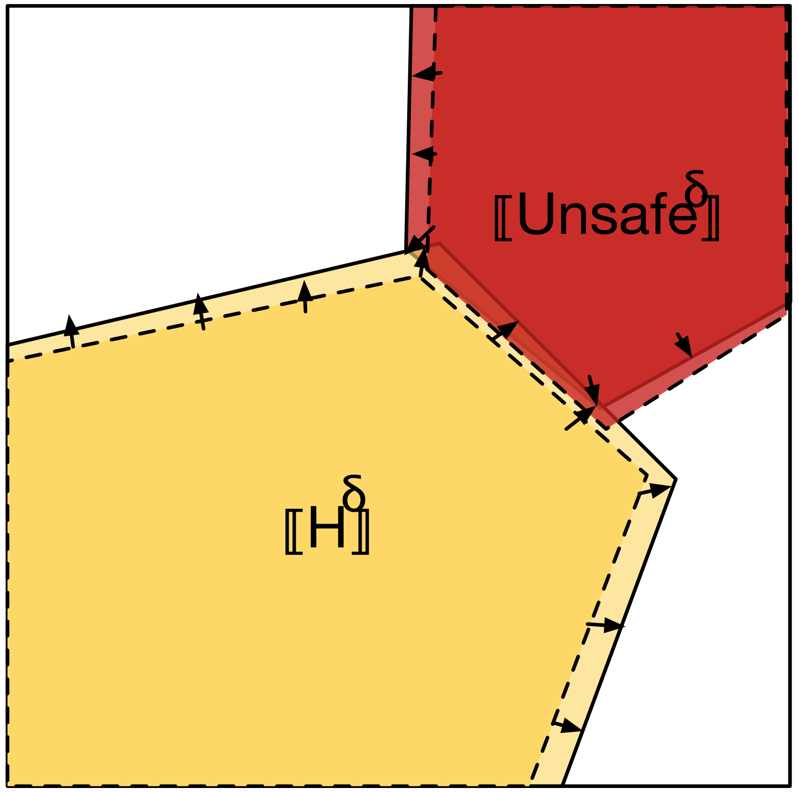
δ-Reachability Analysis of Hybrid Systems



Unreachable
(Safe)

3. Robustness (in verification context)

δ-Reachability Analysis of Hybrid Systems



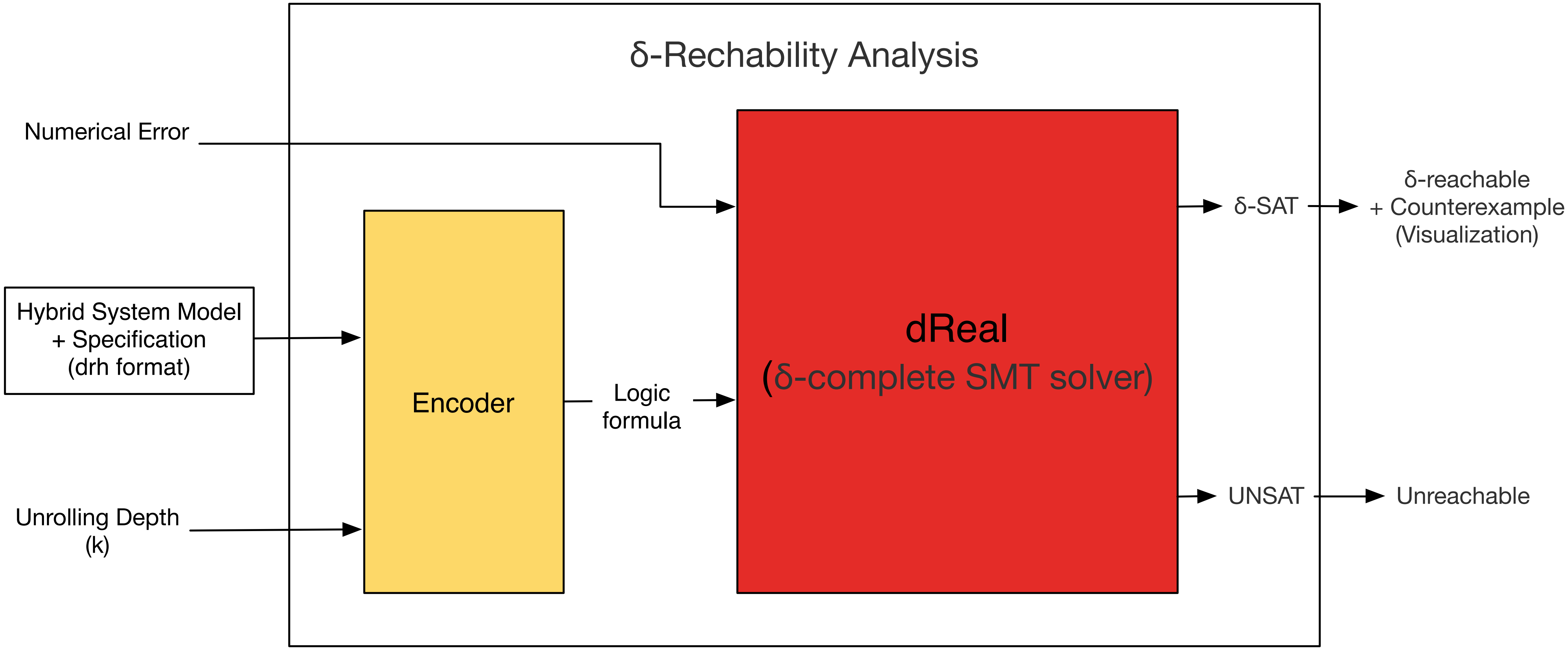
δ-reachable
(δ-unsafe)

3. Robustness (in verification context)
If your safe system is **δ-unsafe** under a reasonably small δ, then it indicates that your system is not **robust**.

δ-Reachability Analysis of Hybrid Systems

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

δ-Reachability Analysis of Hybrid Systems

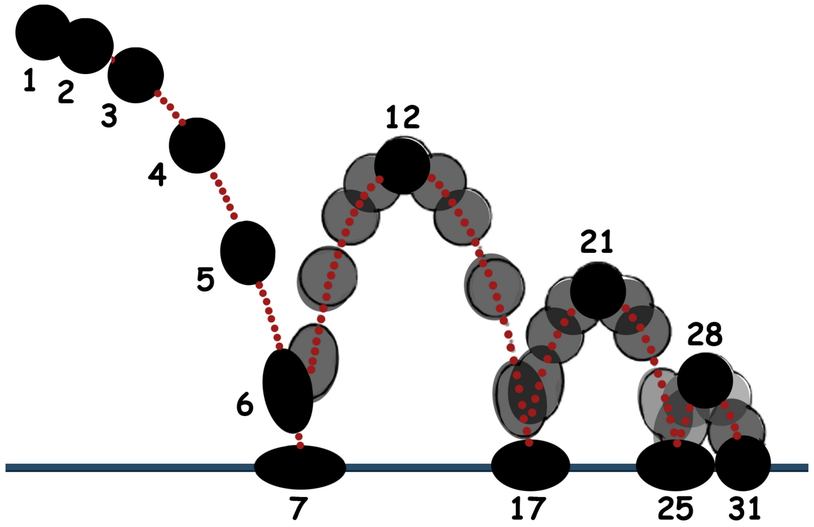


dReach: Bounded delta-reachability analysis tool for Hybrid Systems

Encode reachability problems of hybrid systems to first-order formulas over real numbers, which are solved by a delta-decision procedure, dReal.

Input Format (drh) for Hybrid System

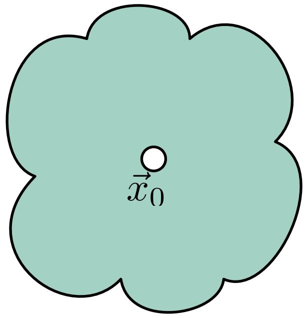
```
#define D 0.45
#define K 0.9
[0, 15] x;
[9.8] g;
[-18, 18] v;
[0, 3] time;

{   mode 1;
    invt: (v <= 0);
          (x >= 0);
    flow: d/dt[x] = v;
          d/dt[v] = -g - (D ⋆ v ˆ 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 ˆ 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

Can a system run into an **unsafe** region after making k steps?

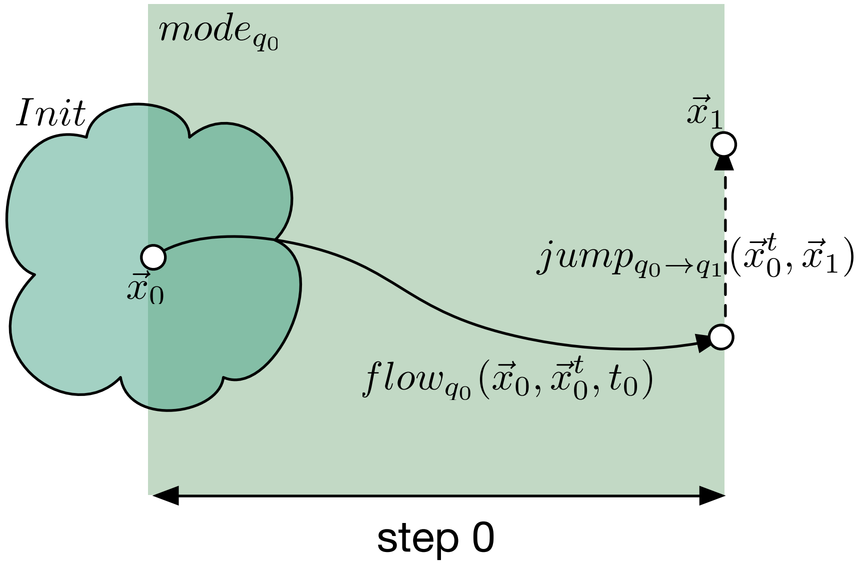


$$x_0 \geq 5 \wedge v = 0$$

$$Init(\vec{x}_0)$$

Logical Encoding of Reachability Problem

Can a system run into an **unsafe** region after making k steps?



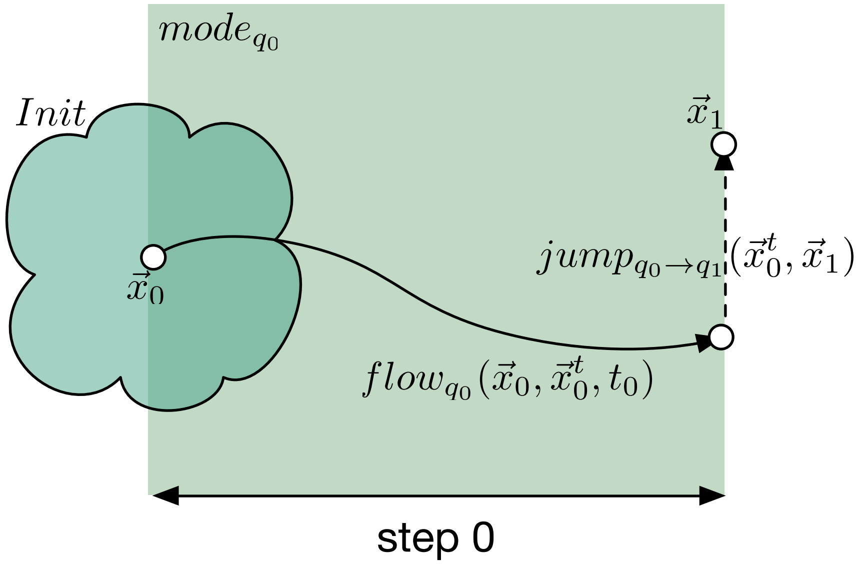
$$Init(\vec{x}_0) \wedge flow_{q_0}(\vec{x}_0, \vec{x}_0^t, t_0) \wedge jump_{q_0 \to q_1}(\vec{x}_0^t, \vec{x}_1)$$

$$x_0^t = x_0 + \int_0^{t_0} v \ \mathrm{dt}$$

$$v_0^t = v_0 + \int_0^{t_0} -9.8 - 0.45v^2 \ \mathrm{dt}$$

Logical Encoding of Reachability Problem

Can a system run into an **unsafe** region after making k steps?



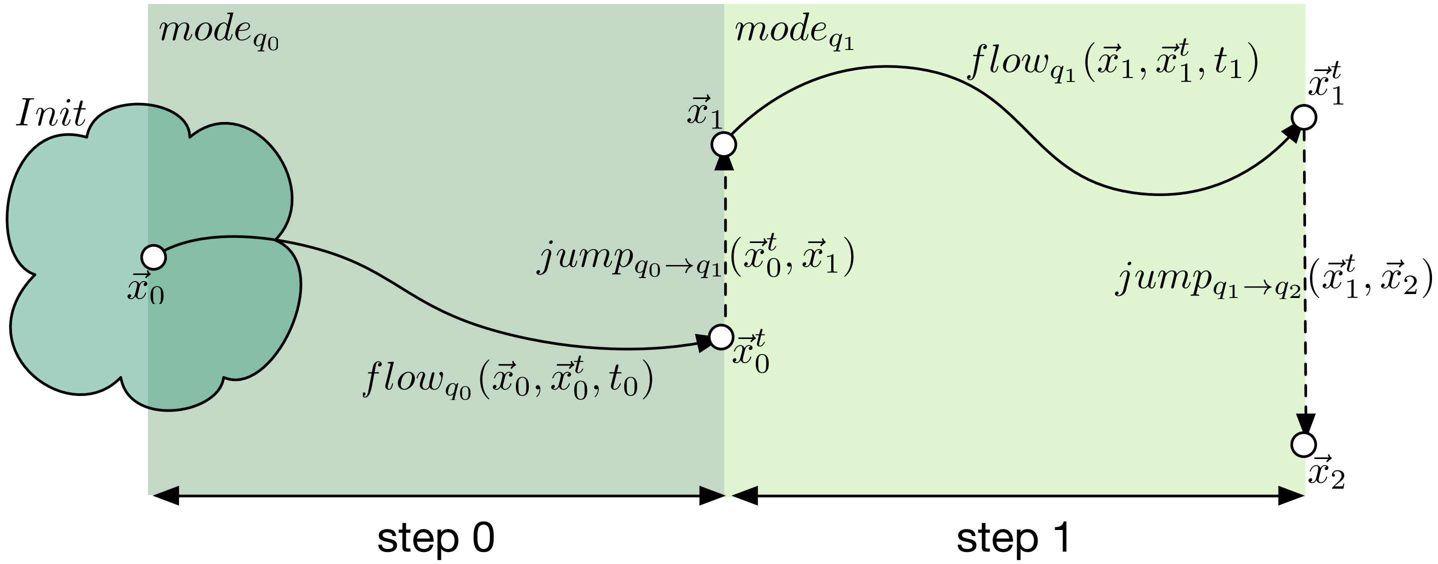
$$x_1 = x_0^t$$
$$v_1 = 0.9 * v_0^t$$

$$Init(\vec{x}_0) \wedge flow_{q_0}(\vec{x}_0, \vec{x}_0^t, t_0) \wedge jump_{q_0\to q_1}(\vec{x}_0^t, \vec{x}_1)$$

$$x_0^t = x_0 + \int_0^{t_0} v \text{ dt}$$
$$v_0^t = v_0 + \int_0^{t_0} -9.8 - 0.45v^2 \text{ dt}$$

Logical Encoding of Reachability Problem

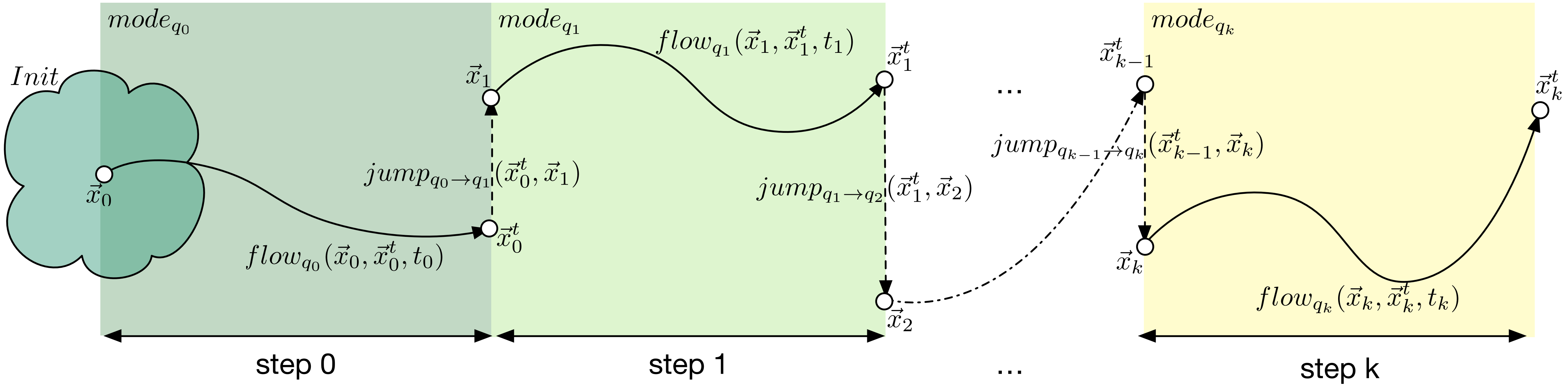
Can a system run into an **unsafe** region after making k steps?



$$Init(\vec{x}_0) \wedge flow_{q_0}(\vec{x}_0, \vec{x}^t_0, t_0) \wedge jump_{q_0 \to q_1}(\vec{x}^t_0, \vec{x}_1) \wedge$$
$$flow_{q_1}(\vec{x}_1, \vec{x}^t_1, t_1) \wedge jump_{q_1 \to q_2}(\vec{x}^t_1, \vec{x}_2)$$

Logical Encoding of Reachability Problem

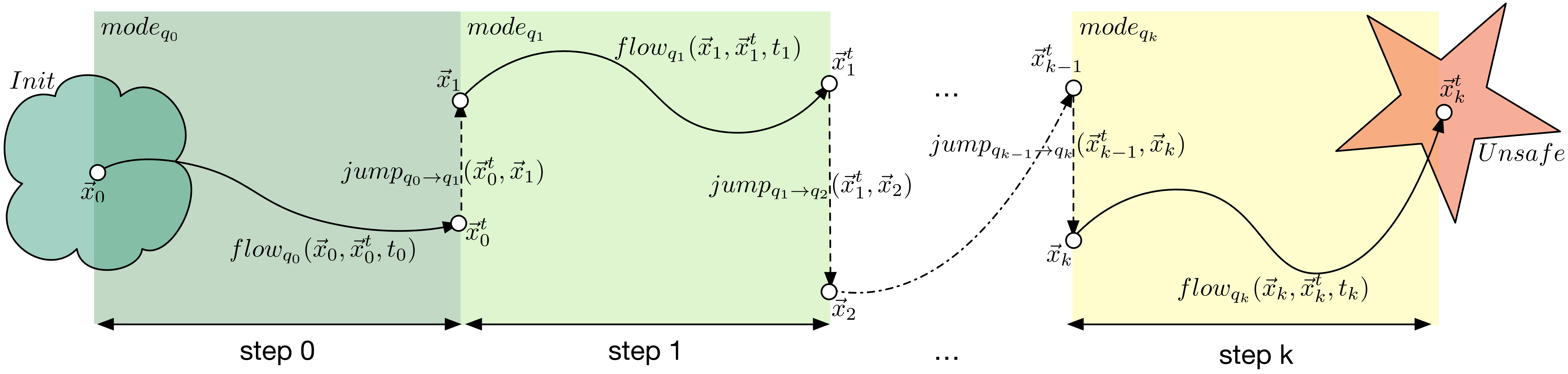
Can a system run into an **unsafe** region after making k steps?



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

Logical Encoding of Reachability Problem

Can a system run into an **unsafe** region after making k steps?



$$Init(\vec{x}_0) \wedge flow_{q_0}(\vec{x}_0, \vec{x}_0^t, t_0) \wedge jump_{q_0 \to q_1}(\vec{x}_0^t, \vec{x}_1) \wedge$$

$$flow_{q_1}(\vec{x}_1, \vec{x}_1^t, t_1) \wedge jump_{q_1 \to q_2}(\vec{x}_1^t, \vec{x}_2) \wedge$$

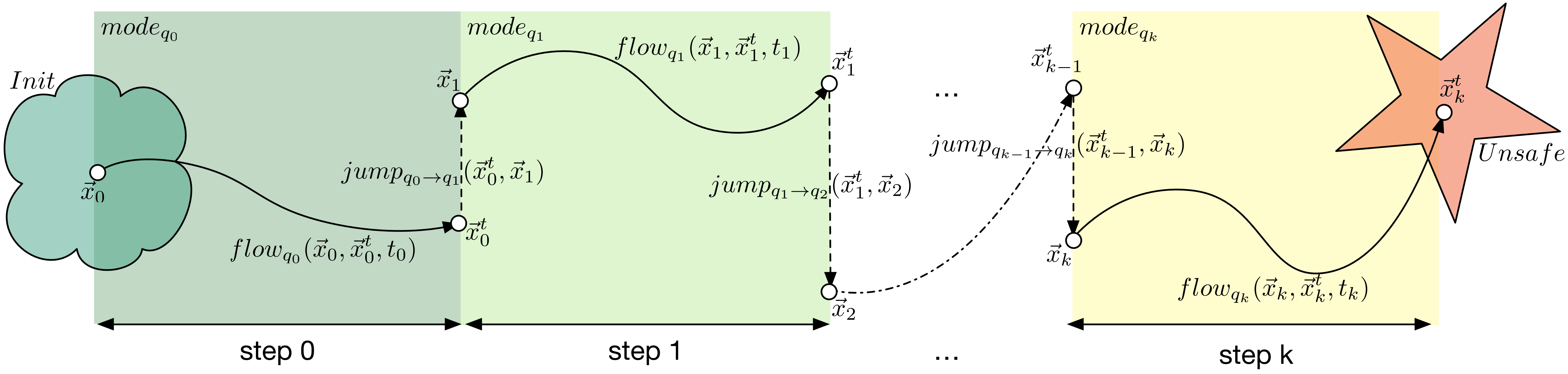
$$\dots$$

$$flow_{q_k}(\vec{x}_k, \vec{x}_k^t, t_k) \wedge Unsafe(\vec{x}_k^t)$$

$$x_n^t \geq 0.45$$

Logical Encoding of Reachability Problem

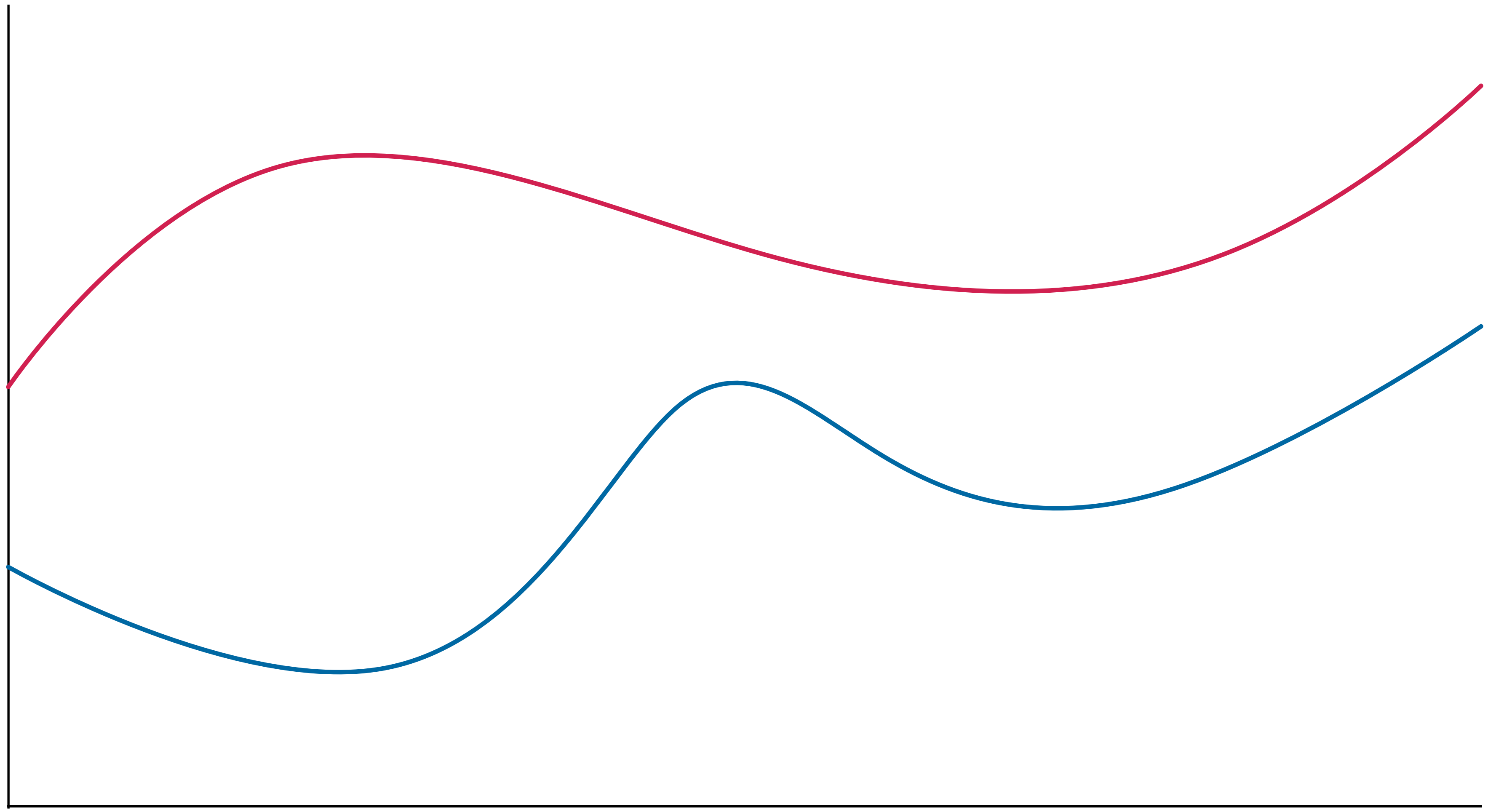
Can a system run into an **unsafe** region after making k steps?



$$\exists \vec{x}_0, \vec{x}_1, \ldots, \vec{x}_k \exists \vec{x}_0^t, \vec{x}_1^t, \ldots, \vec{x}_k^t \exists t_0, t_1, \ldots, t_k$$

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

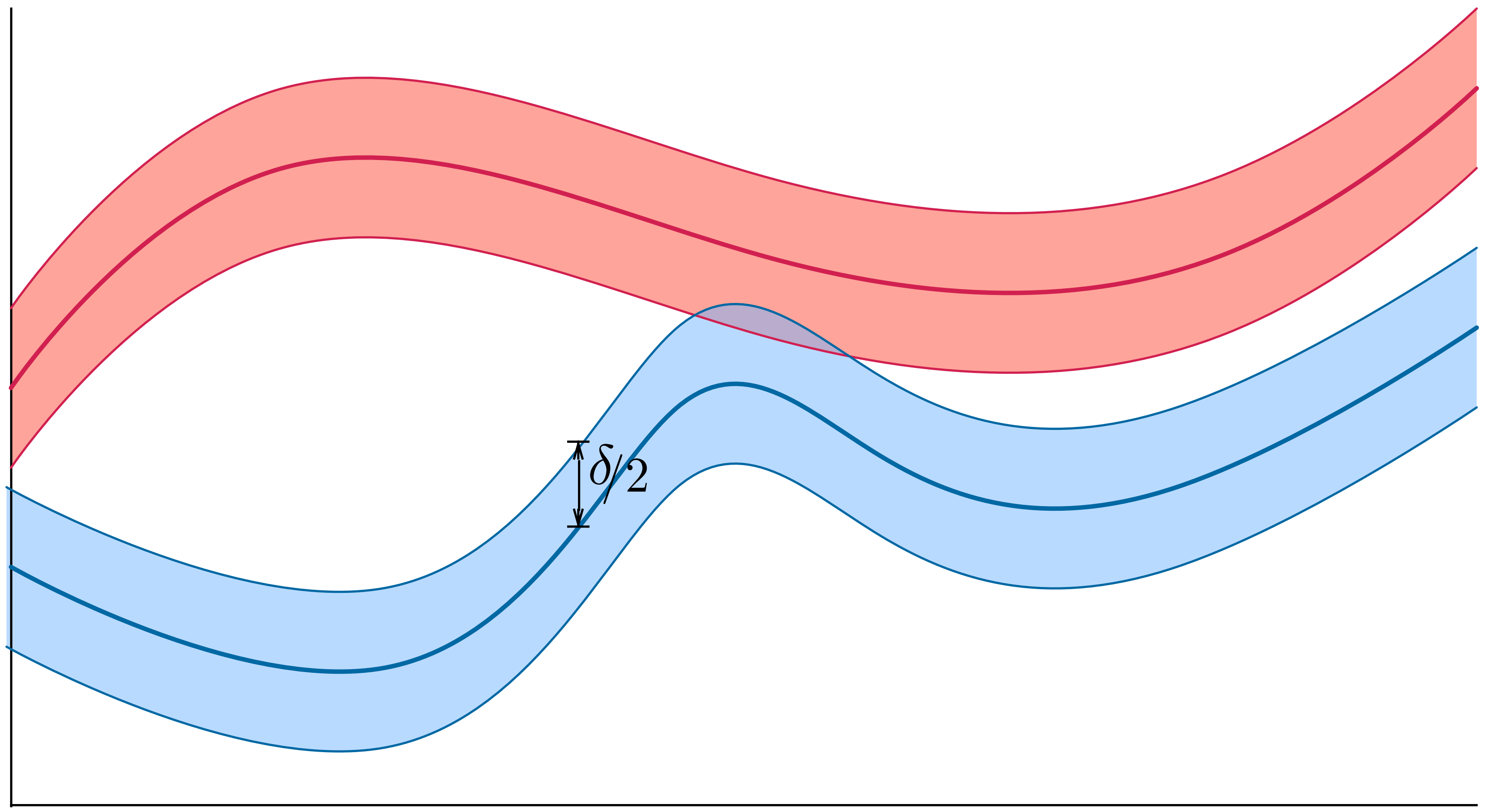
Decision Problem



Standard Form

$$\phi := \exists^{\mathbf{I}}\mathbf{x} \bigvee_i \bigwedge_j f_{i,j}(\mathbf{x}) = 0$$

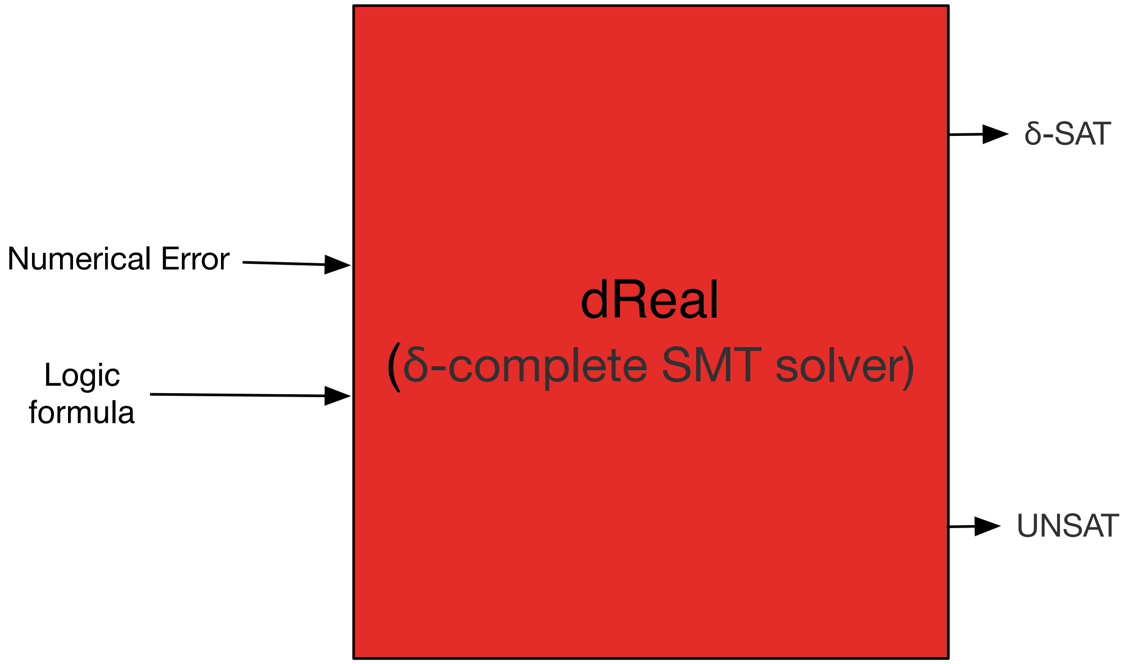
δ-Decision Problem



δ-Weakening of φ

$$\phi^{\delta} := \exists^{\mathbf{I}}\mathbf{x} \bigvee_{i} \bigwedge_{j} |f_{i,j}(\mathbf{x})| \leq \delta$$

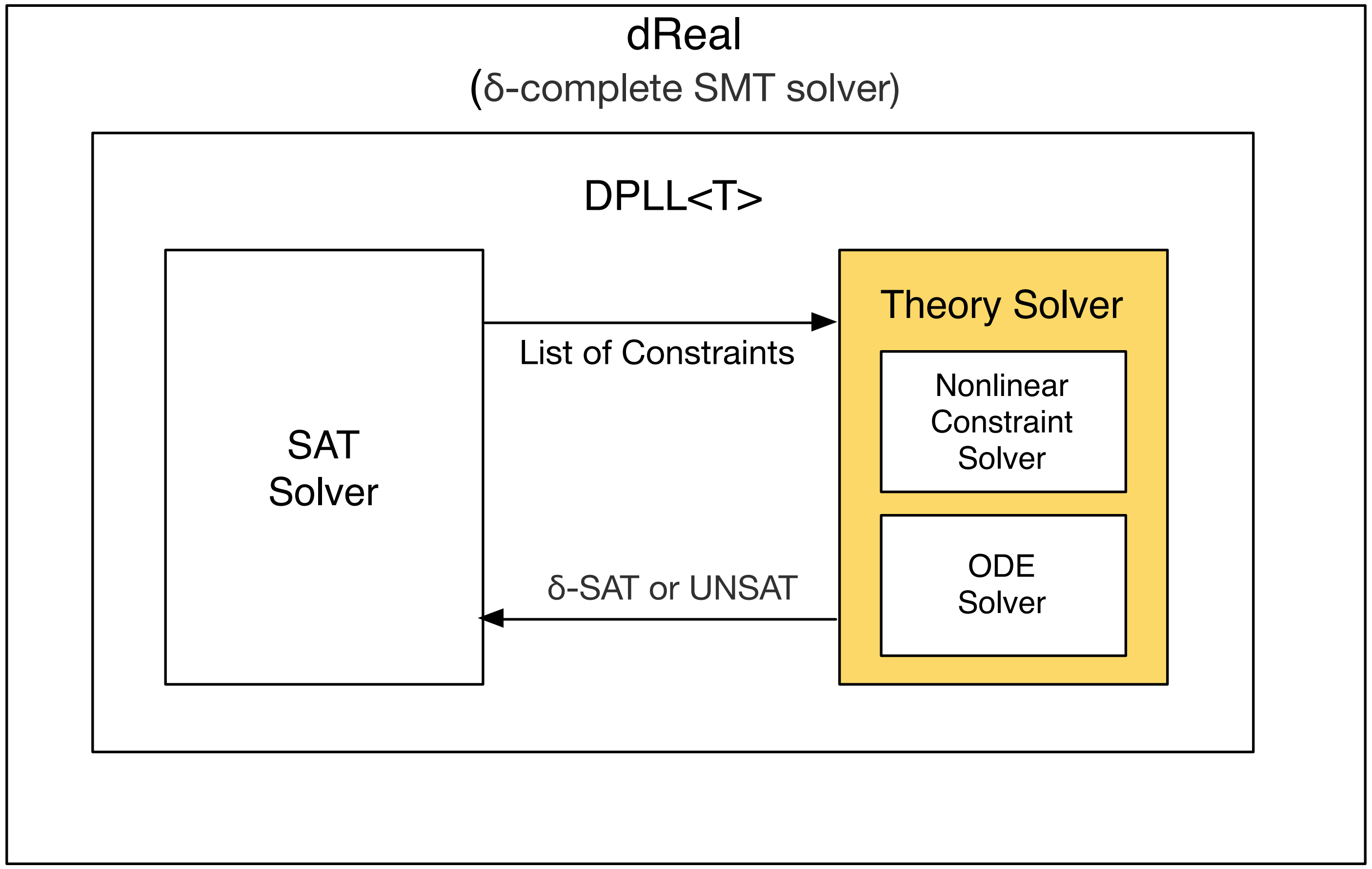
Solving Logic Formula



DPLL<T> Framework

- **SAT solver** finds a satisfying **Boolean** assignment
- **Theory solver** checks whether the assignment is feasible under the first order theory of **Real**

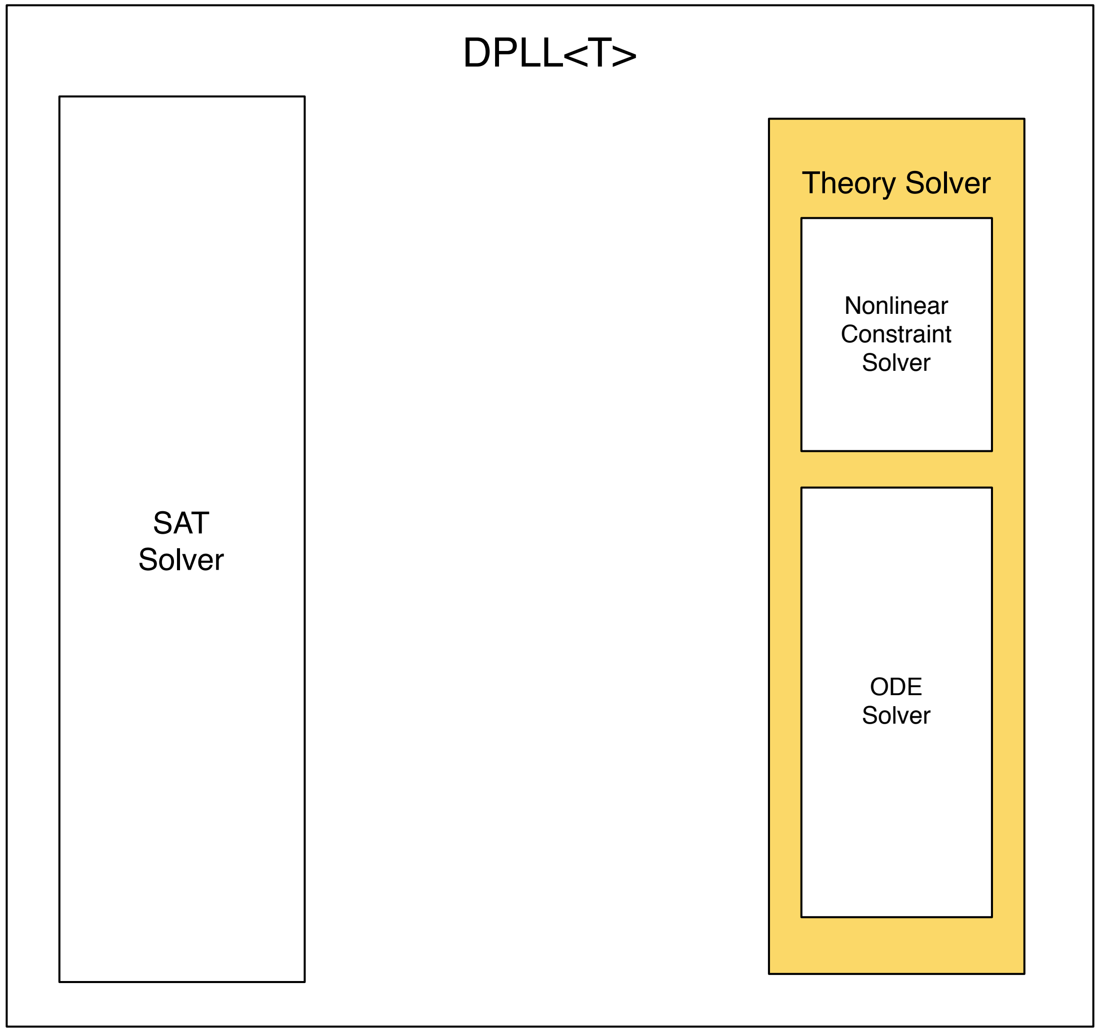
Solving Logic Formula



DPLL<T> Framework

- **SAT solver** finds a satisfying **Boolean** assignment
- **Theory solver** checks whether the assignment is feasible under the first order theory of **Real**

Solving Logic Formula



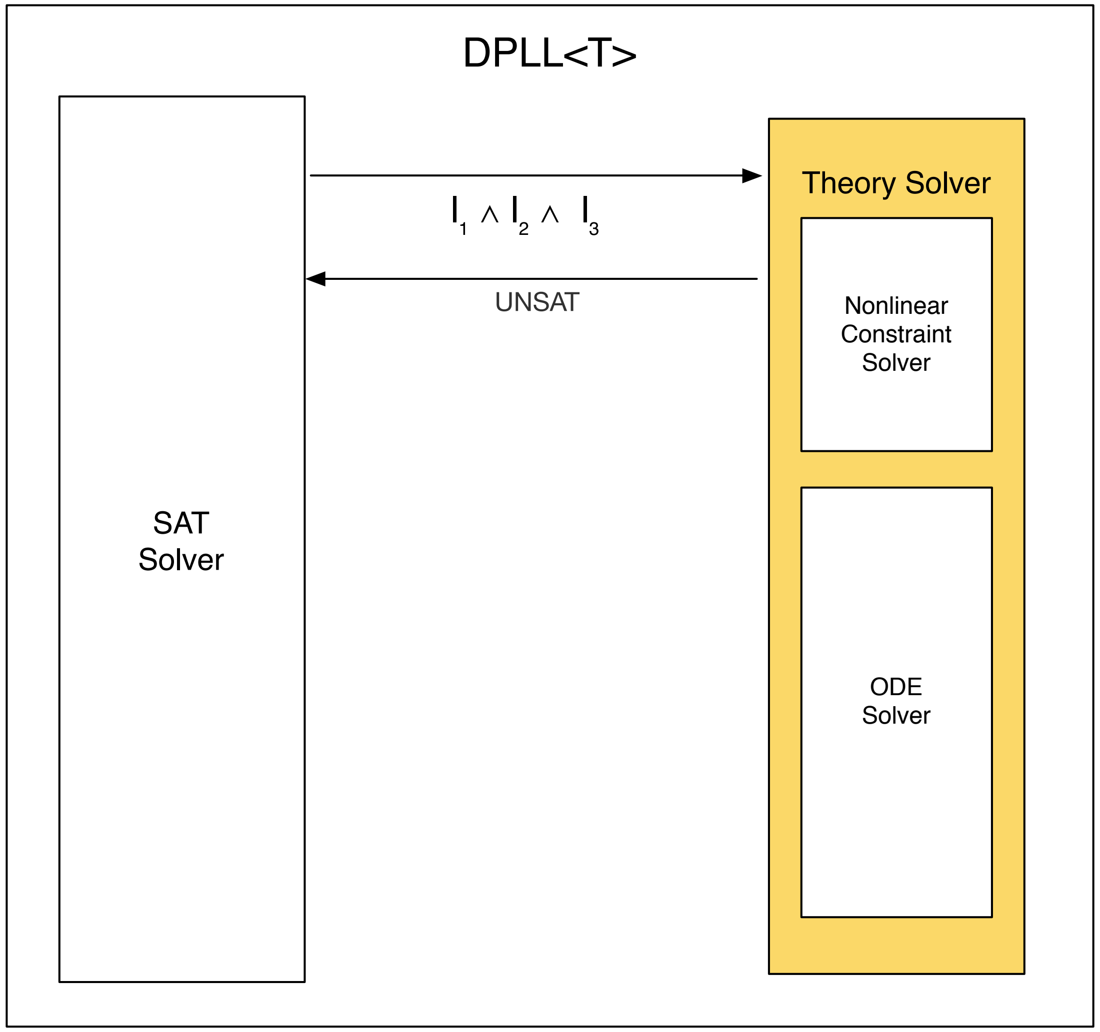
$$\exists x.\ (l_1 \wedge l_2) \implies (l_3 \vee l_4 \vee l_5)$$

$$l_1 := x > 4$$
$$l_2 := x < 10$$
$$l_3 := x^2 < 10$$
$$l_4 := x^2 - 6x + 9 = 0$$
$$l_5 := \cos(x) < 0.5$$

DPLL<T> Framework

- **SAT solver** finds a satisfying **Boolean** assignment
- **Theory solver** checks whether the assignment is feasible under the first order theory of **Real**

Solving Logic Formula



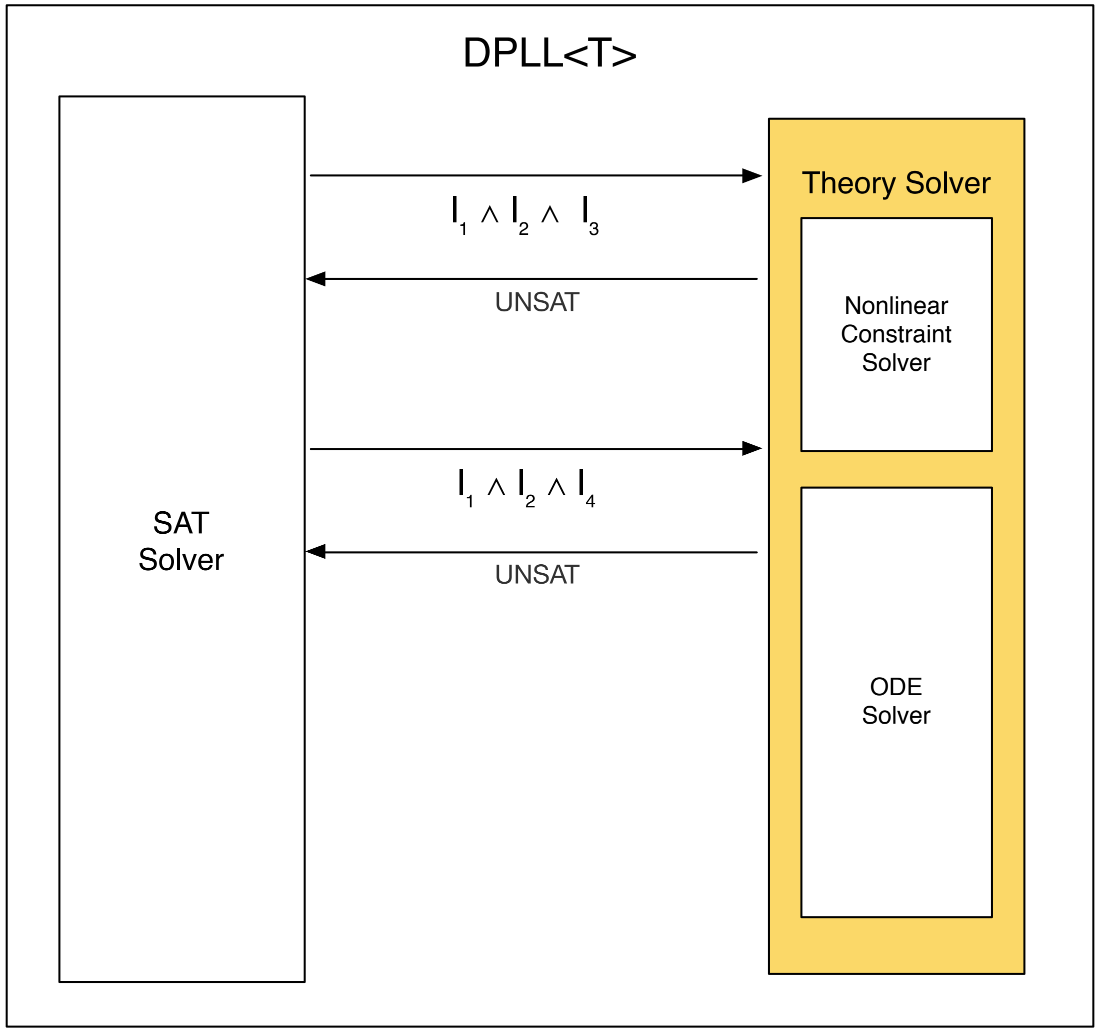
$$\exists x.\ (l_1 \wedge l_2) \implies (l_3 \vee l_4 \vee l_5)$$

$$\boxed{l_1 := x > 4}$$
$$\boxed{l_2 := x < 10}$$
$$\boxed{l_3 := x^2 < 10}$$
$$l_4 := x^2 - 6x + 9 = 0$$
$$l_5 := \cos(x) < 0.5$$

DPLL<T> Framework

- **SAT solver** finds a satisfying **Boolean** assignment
- **Theory solver** checks whether the assignment is feasible under the first order theory of **Real**

Solving Logic Formula



$$\exists x.\ (l_1 \wedge l_2) \implies (l_3 \vee l_4 \vee l_5)$$

$$\boxed{l_1 := x > 4}$$

$$\boxed{l_2 := x < 10}$$

$$l_3 := x^2 < 10$$

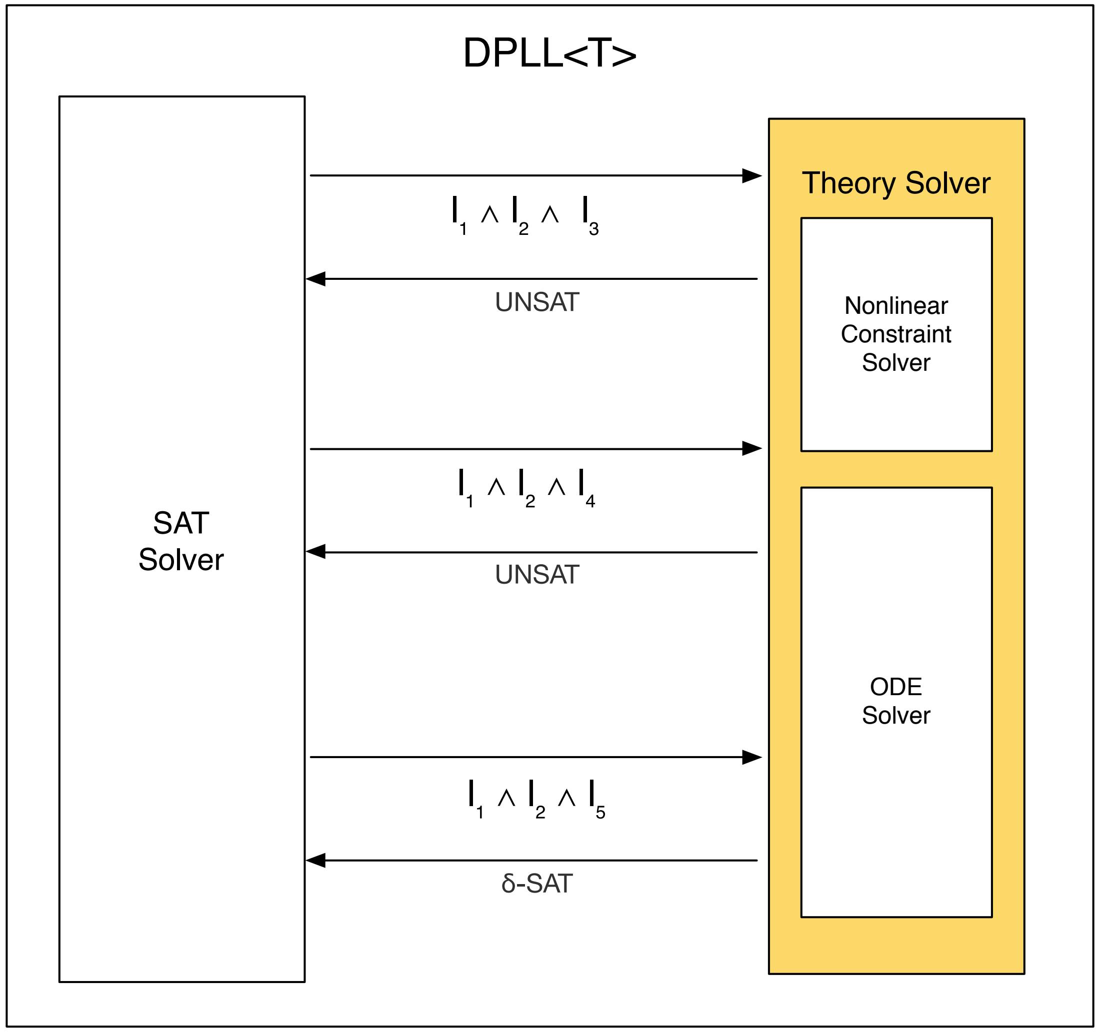
$$\boxed{l_4 := x^2 - 6x + 9 = 0}$$

$$l_5 := \cos(x) < 0.5$$

DPLL<T> Framework

- **SAT solver** finds a satisfying **Boolean** assignment
- **Theory solver** checks whether the assignment is feasible under the first order theory of **Real**

Solving Logic Formula



$$\exists x.\ (l_1 \wedge l_2) \implies (l_3 \vee l_4 \vee l_5)$$

$$\boxed{l_1 := x > 4}$$

$$\boxed{l_2 := x < 10}$$

$$l_3 := x^2 < 10$$

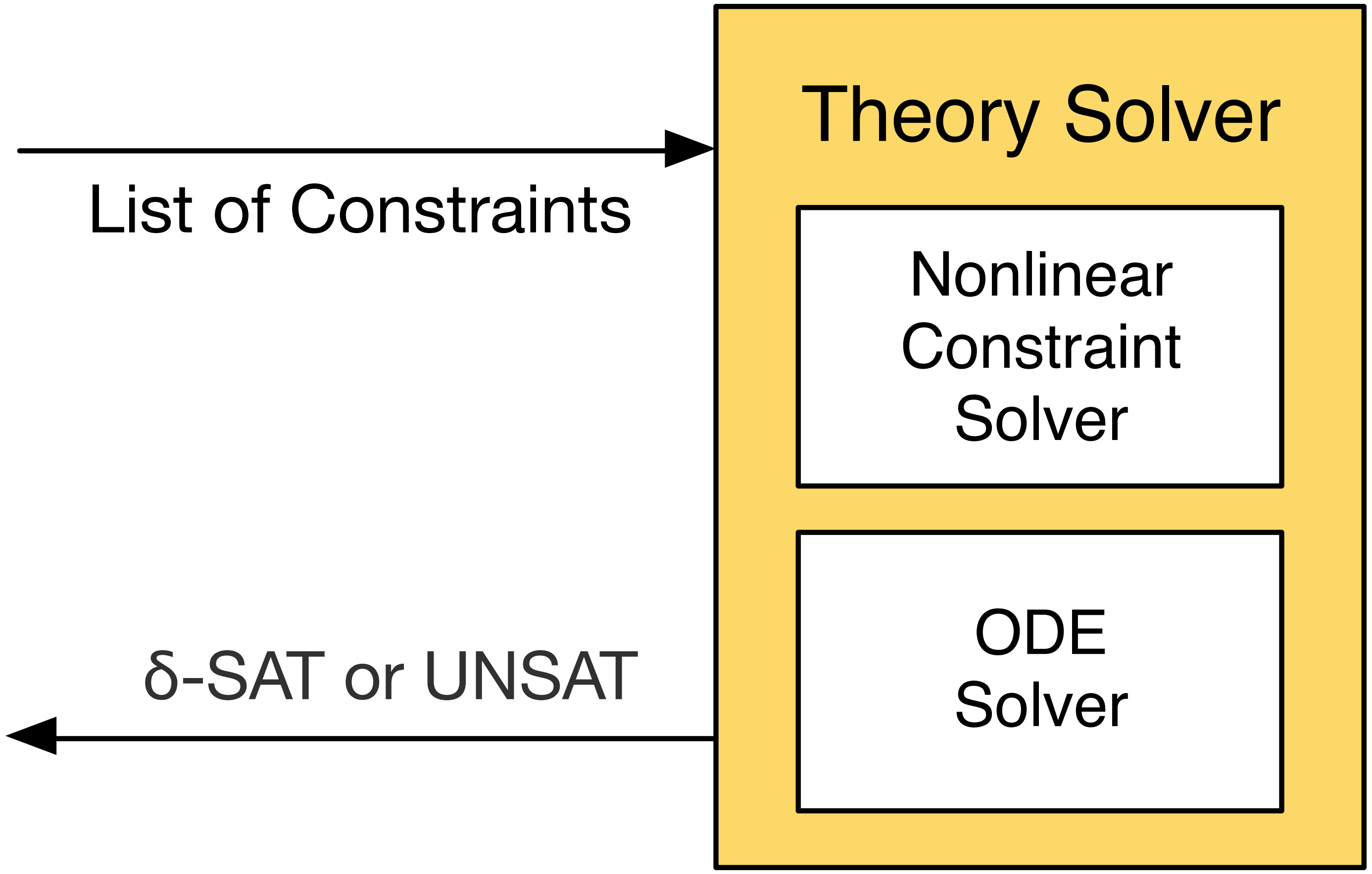
$$l_4 := x^2 - 6x + 9 = 0$$

$$\boxed{l_5 := \cos(x) < 0.5}$$

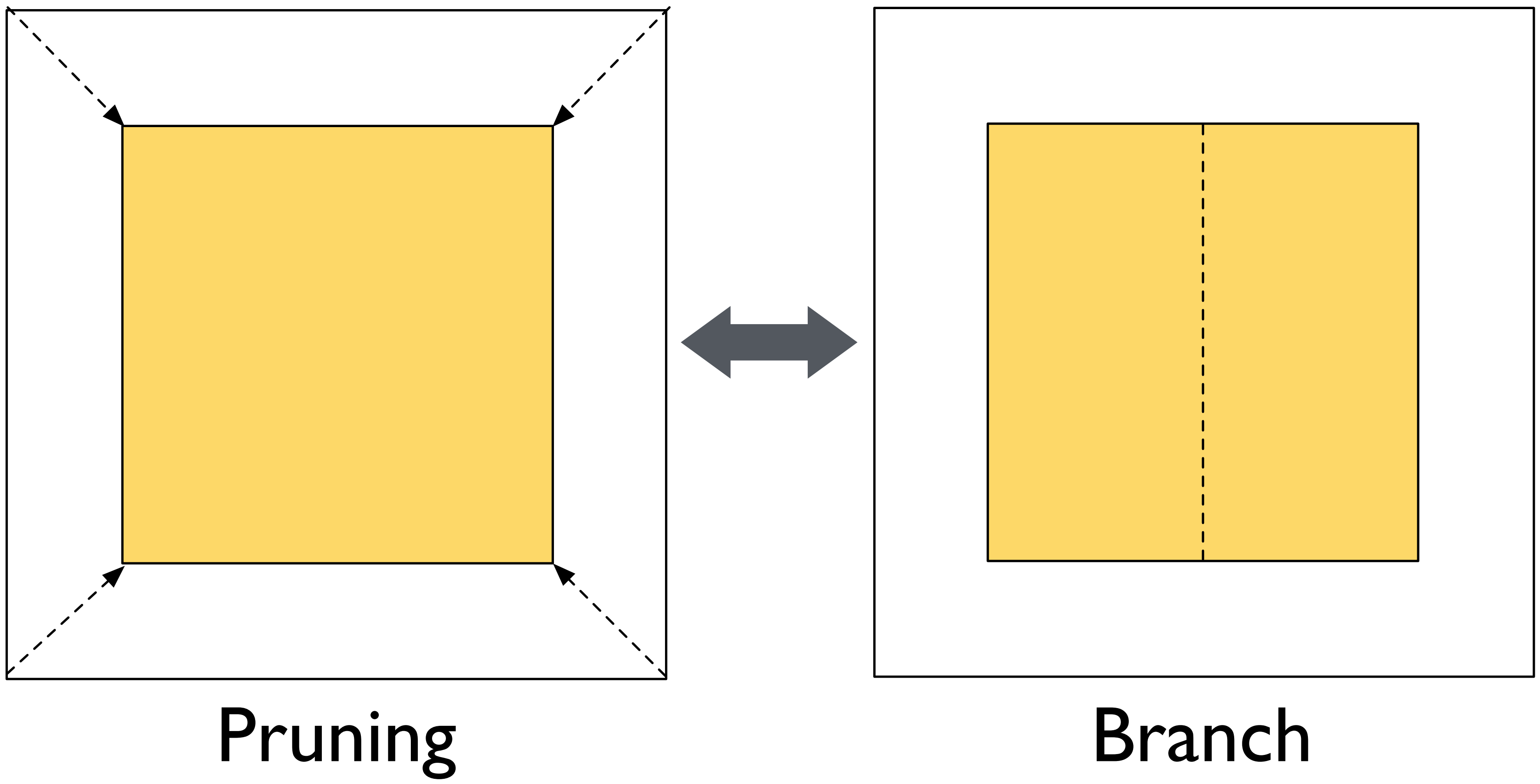
DPLL<T> Framework

- **SAT solver** finds a satisfying **Boolean** assignment
- **Theory solver** checks whether the assignment is feasible under the first order theory of **Real**

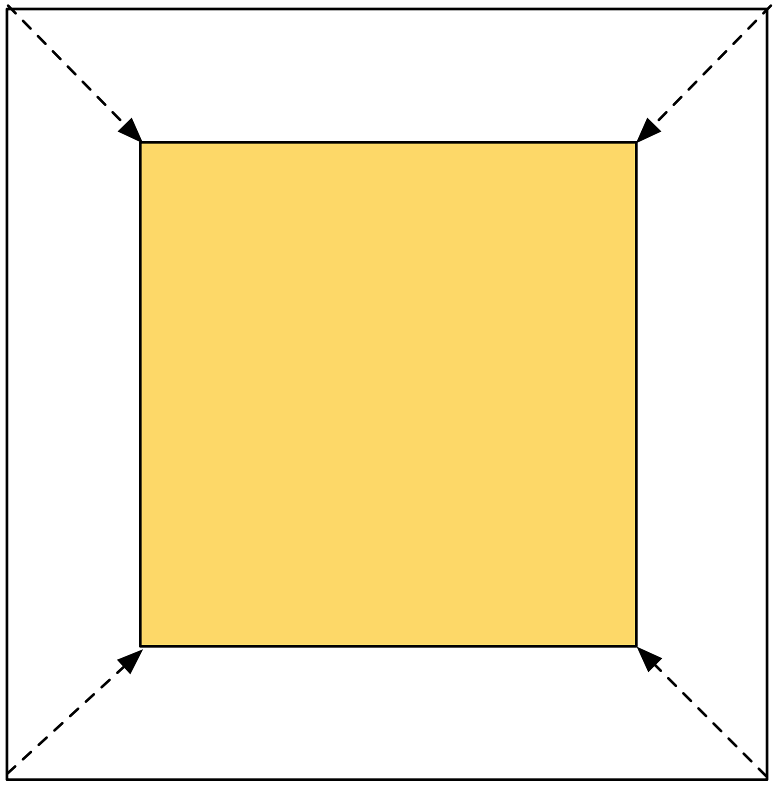
Main Algorithm of Theory Solver



Main Algorithm of Theory Solver: ICP(Interval Constraint Propagation)



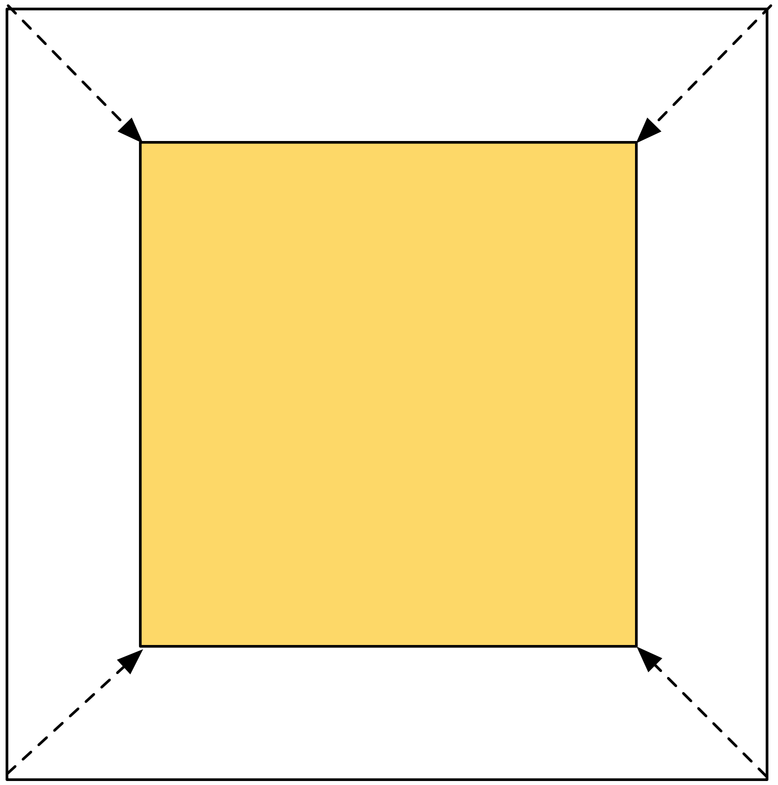
Main Algorithm of Theory Solver: ICP(Interval Constraint Propagation)



Pruning

Contracting interval domains associated to variables of Real without removing any value that is consistent with a set of constraints

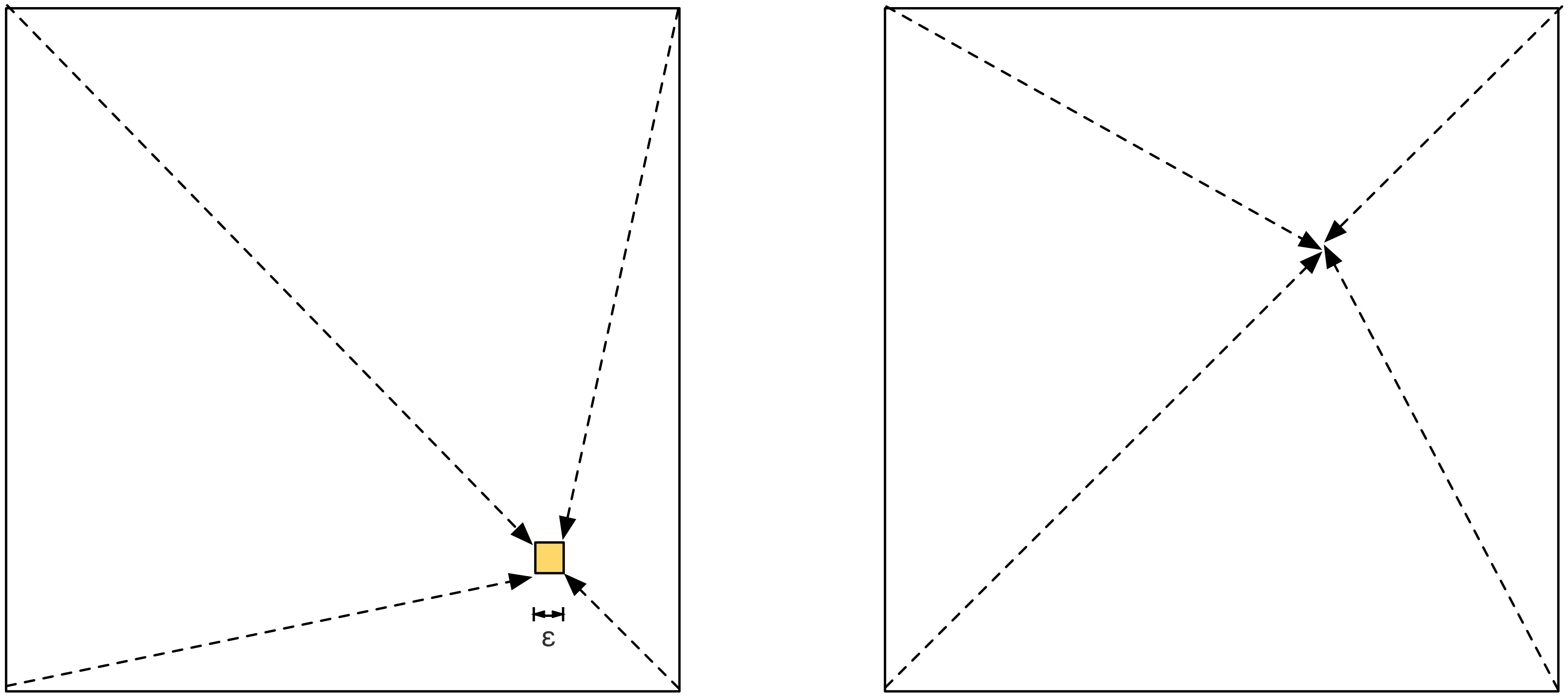
Main Algorithm of Theory Solver: ICP(Interval Constraint Propagation)



Pruning

Monotone	$B_1 \subseteq B_2 \implies f(B_1) \subseteq f(B_2)$	(Termination)
Reductive	$f(B) \subseteq B$	(Termination)
Solution-Preserving	$x \in B \wedge x \in Sol(f) \implies x \in f(B)$	(Soundness)

Two Termination Conditions of ICP

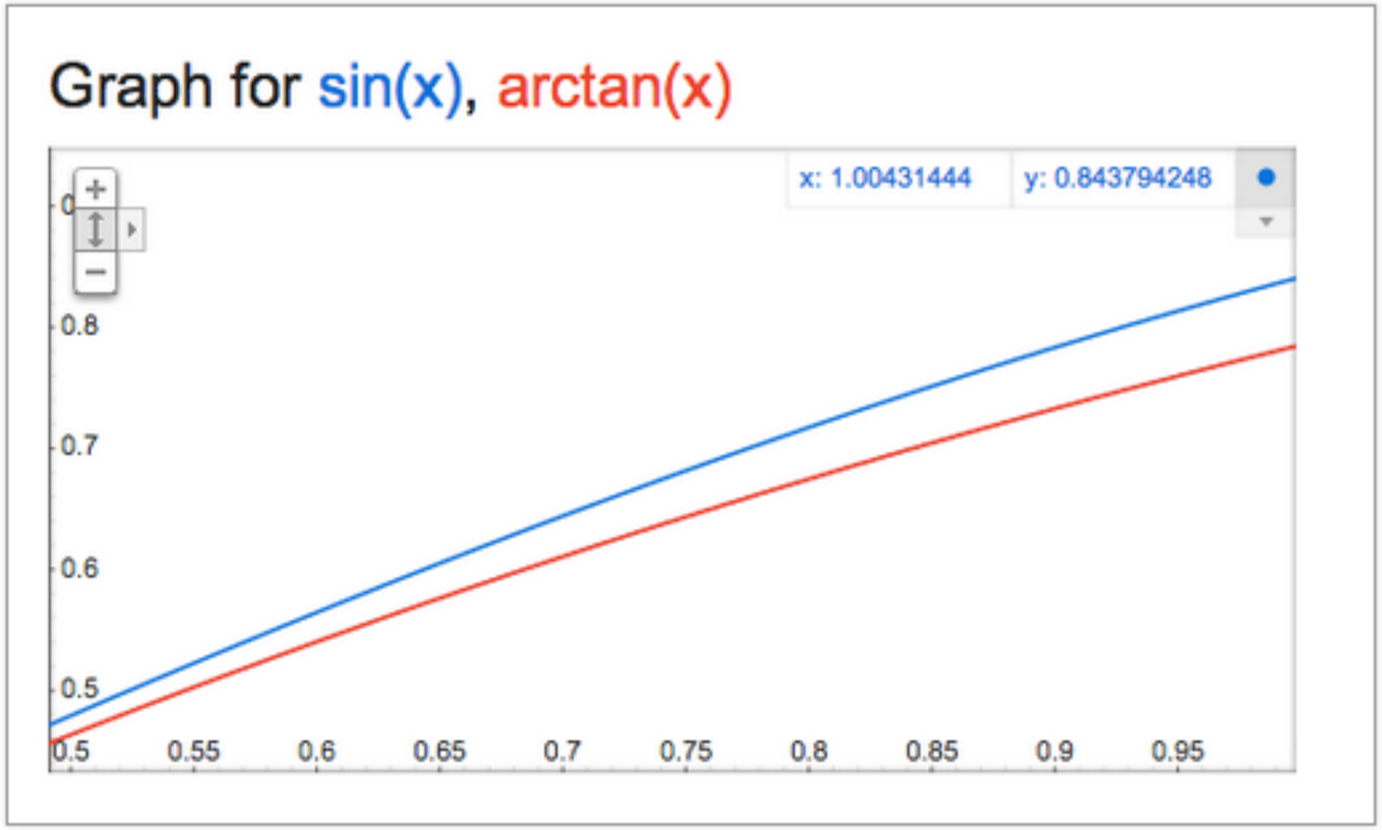


δ-sat

Unsat

Example of Pruning Operations

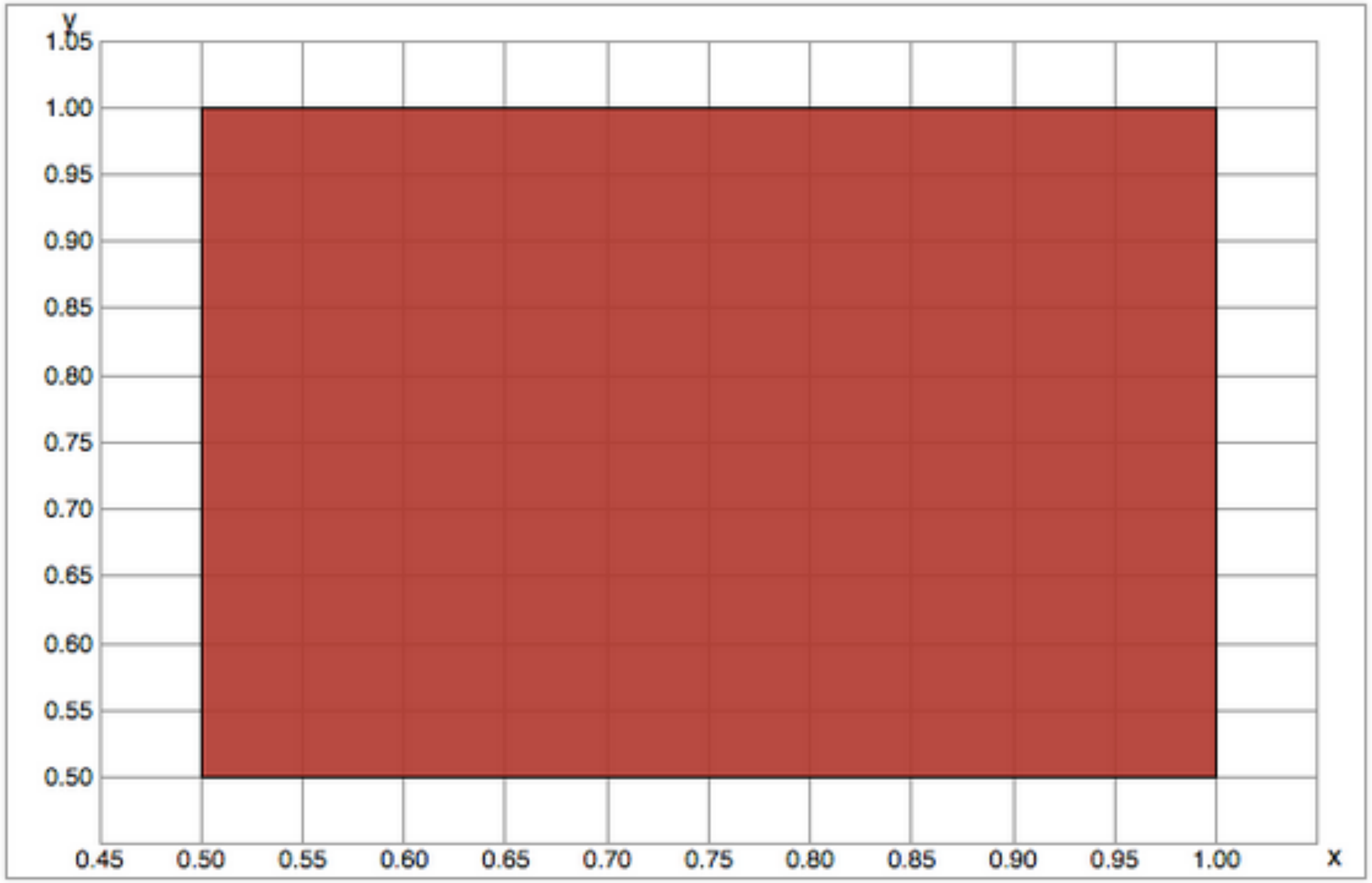
$$\exists x, y \in [0.5, 1.0] : y = \sin(x) \wedge y = \operatorname{atan}(x)$$



ANSWER: **UNSAT**

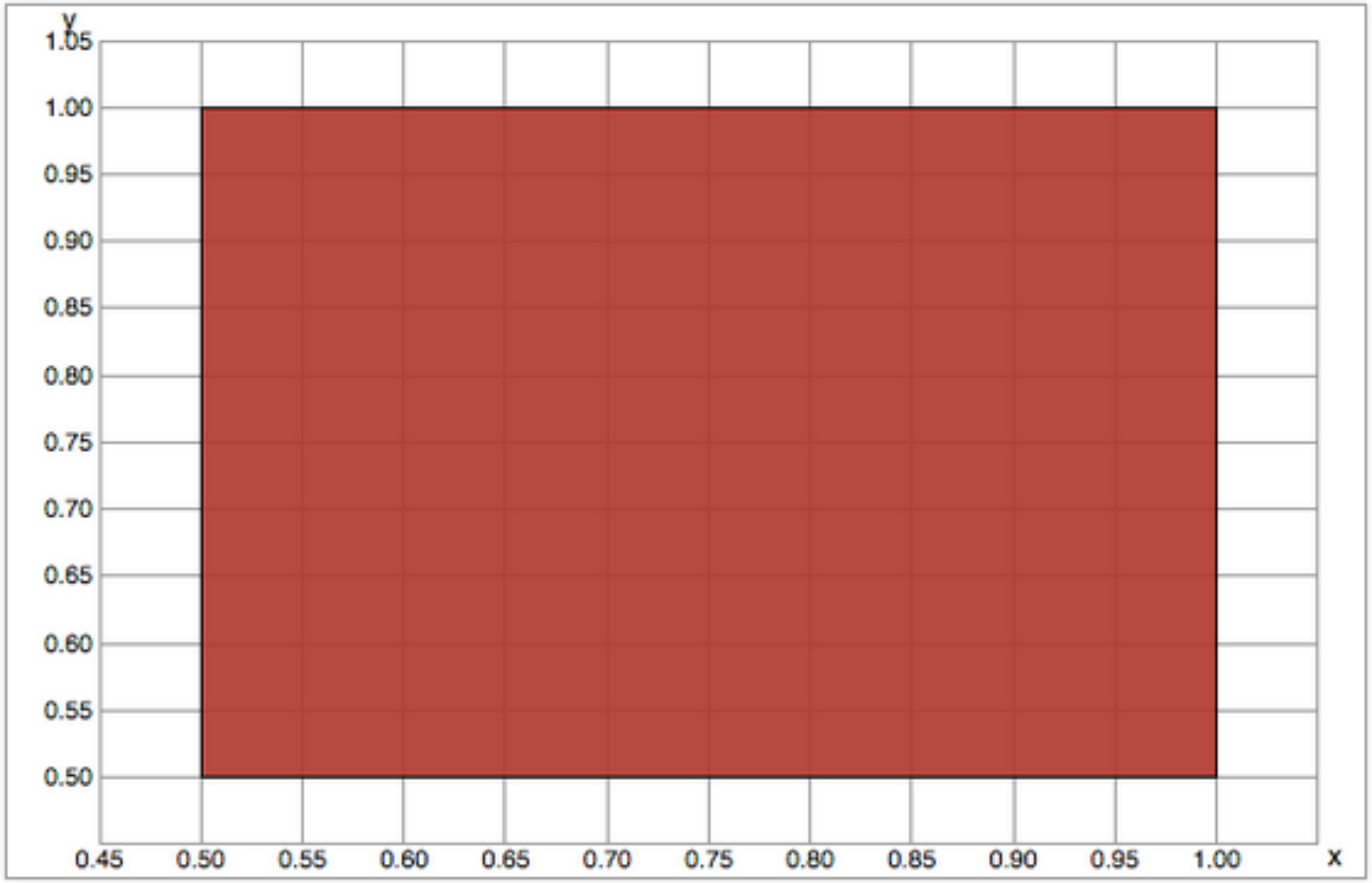
Example of Pruning Operations

$$\exists x, y \in [0.5, 1.0] : y = \sin(x) \wedge y = \operatorname{atan}(x)$$



Example of Pruning Operations

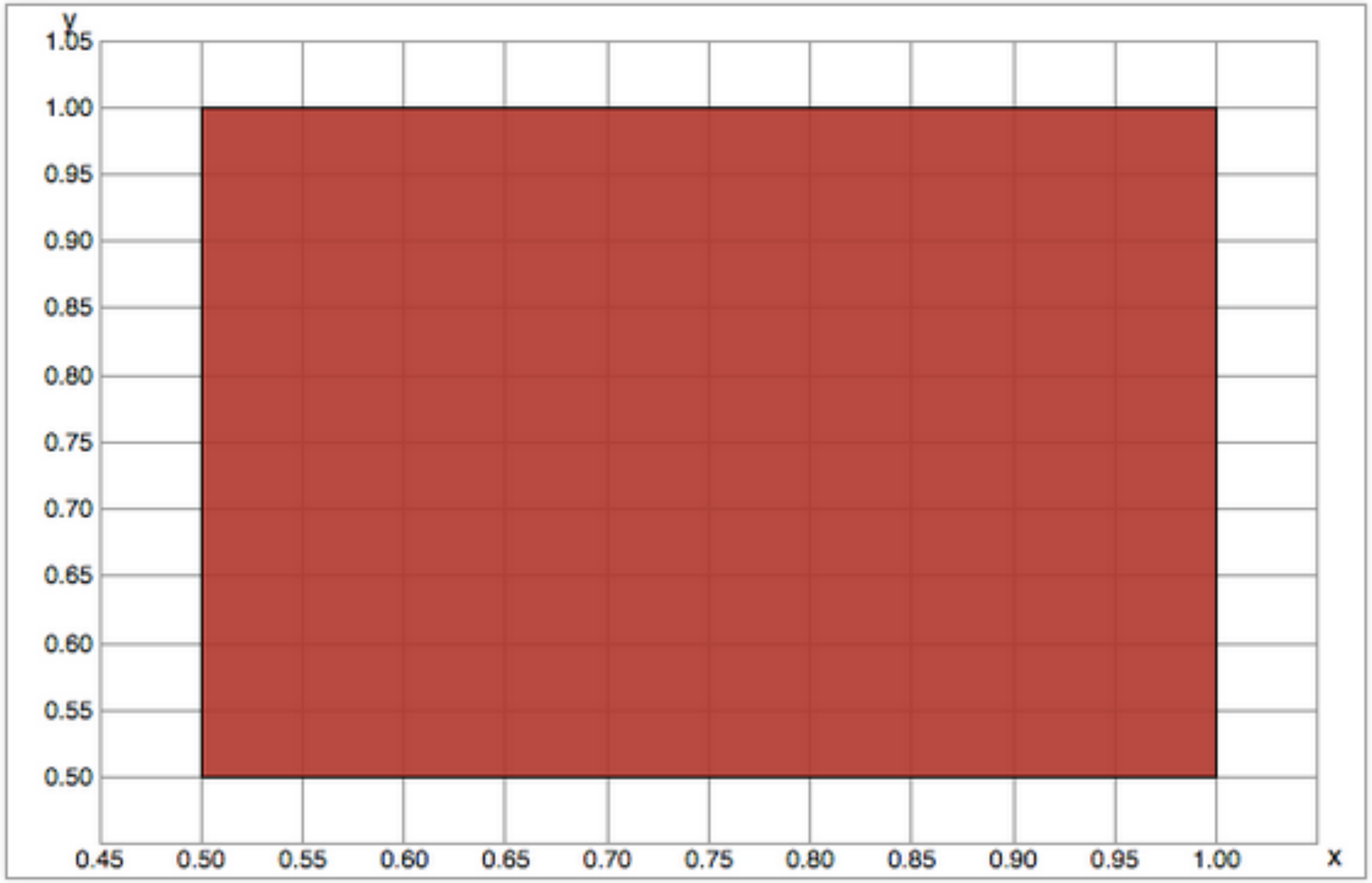
$$\exists x, y \in [0.5, 1.0] : y = \sin(x) \wedge y = \mathrm{atan}(x)$$



$$\begin{aligned} x' &= x \cap \sin^{-1}(y) \\ &= [0.5, 1.0] \cap \sin^{-1}([0.5, 1.0]) \\ &= [0.5, 1.0] \cap [0.524, 1.570] \\ &= [0.524, 1.0] \end{aligned}$$

Example of Pruning Operations

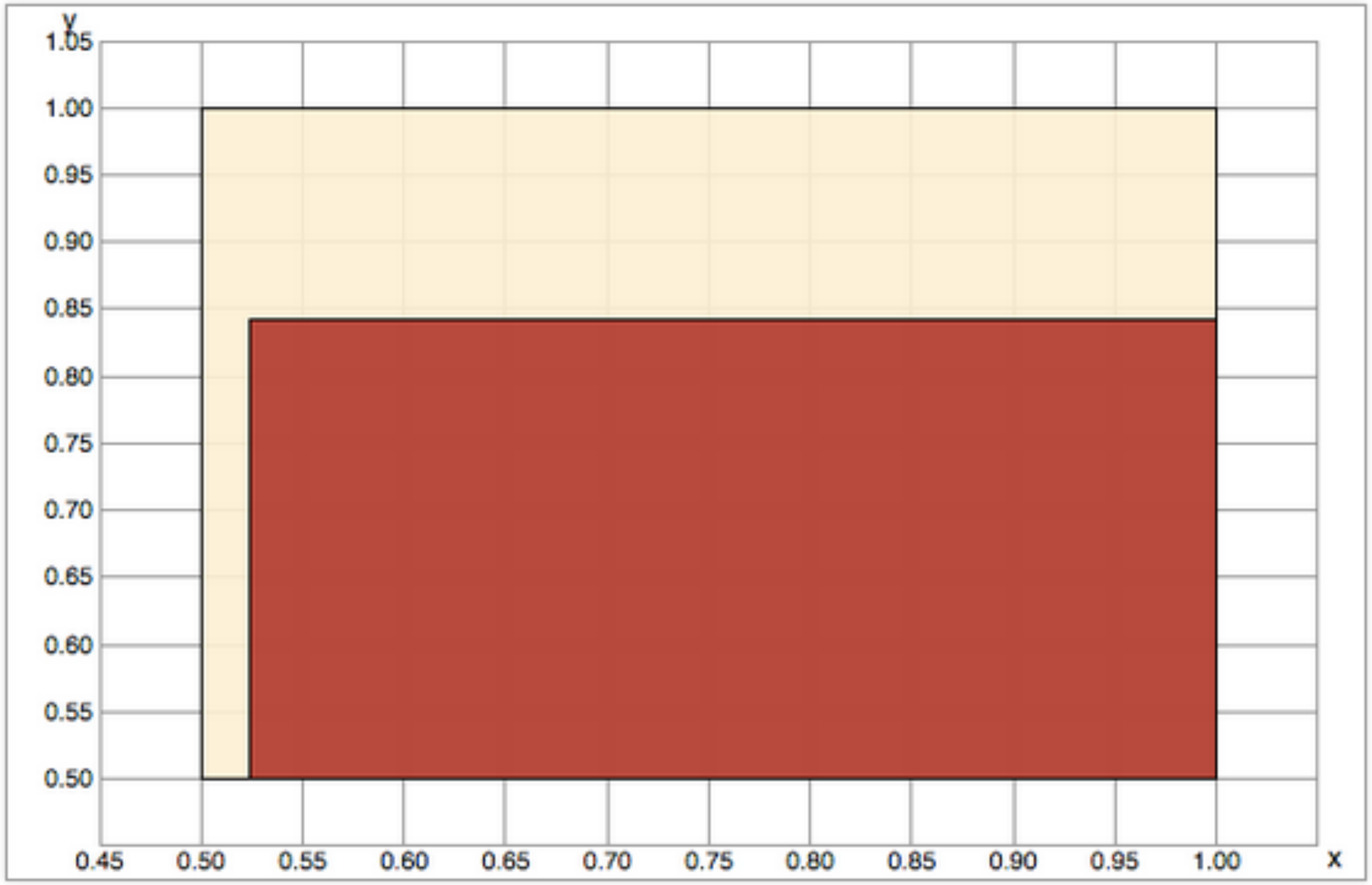
$$\exists x, y \in [0.5, 1.0] : y = \sin(x) \wedge y = \operatorname{atan}(x)$$



$$\begin{aligned} y' &= y \cap \sin(x) \\ &= [0.5, 1.0] \cap \sin([0.524, 1.0]) \\ &= [0.5, 1.0] \cap [0.5, 0.841] \\ &= [0.5, 0.841] \end{aligned}$$

Example of Pruning Operations

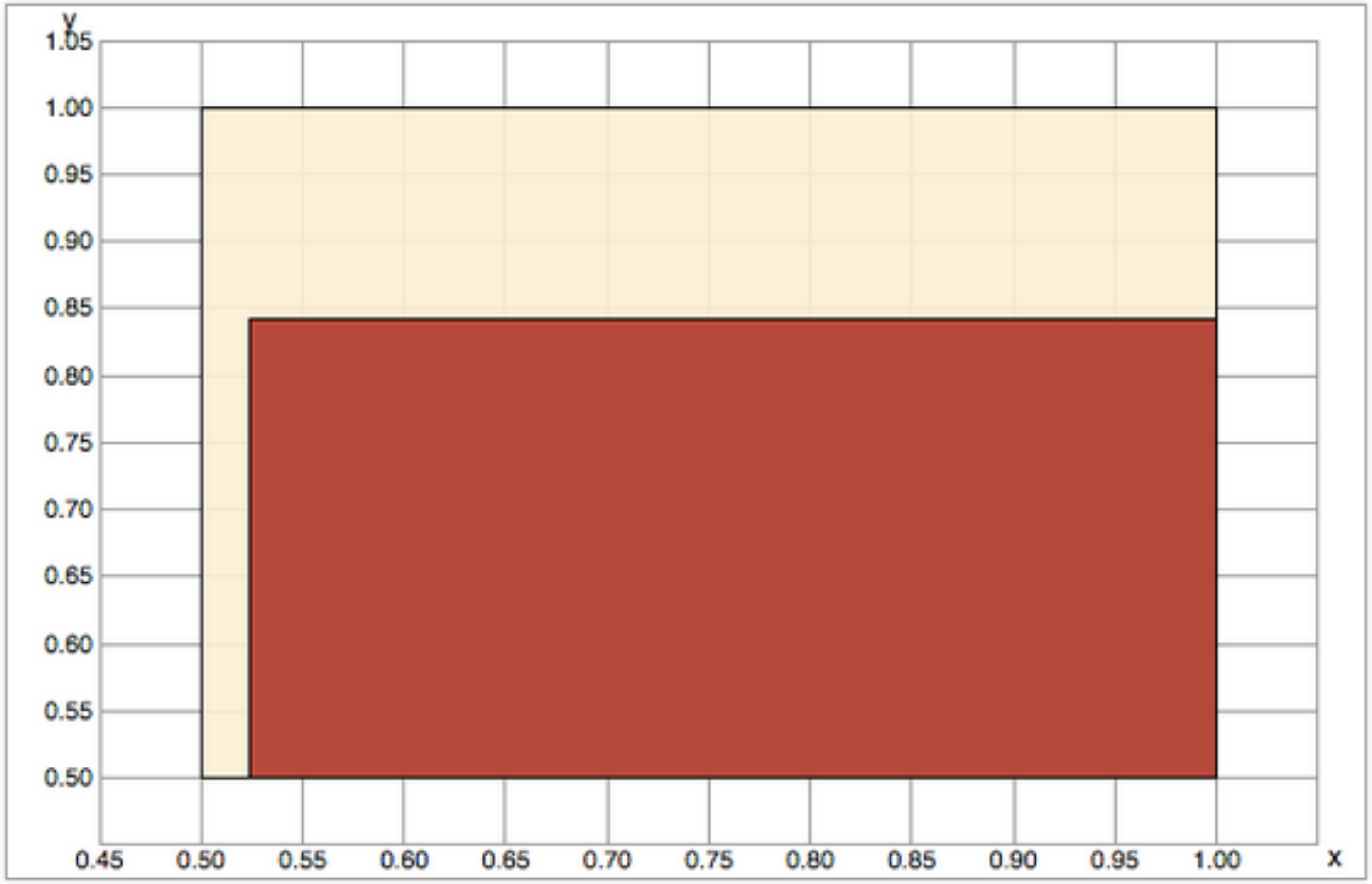
$$\exists x, y \in [0.5, 1.0] : y = \sin(x) \wedge y = \mathrm{atan}(x)$$



$$x : [0.524, 1.0], y : [0.5, 0.841]$$

Example of Pruning Operations

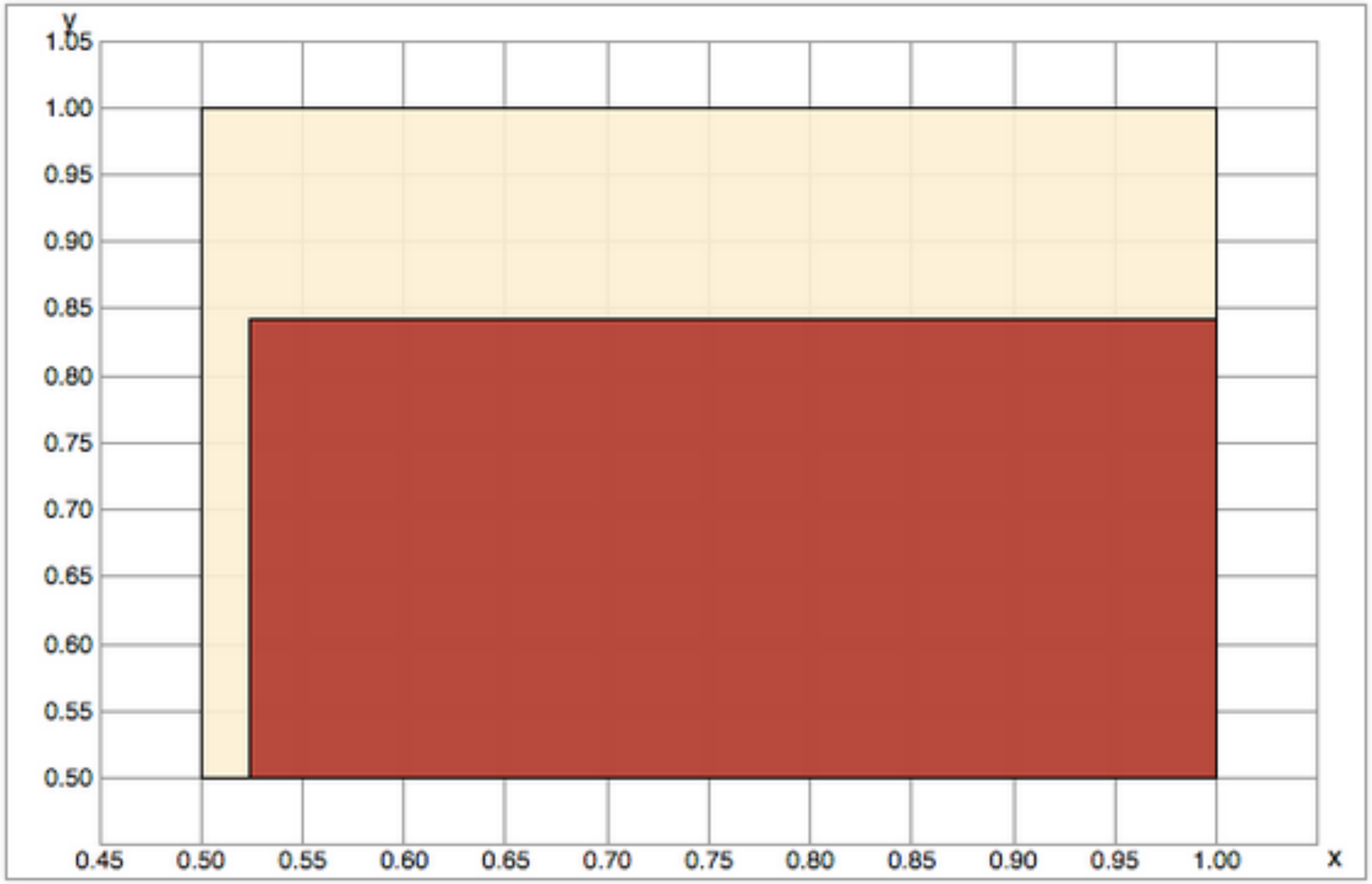
$$\exists x, y \in [0.5, 1.0] : y = \sin(x) \wedge y = \mathrm{atan}(x)$$



$$\begin{aligned} x' &= x \cap \mathrm{atan}^{-1}(y) \\ &= [0.524, 1] \cap \mathrm{atan}^{-1}([0.5, 0.841]) \\ &= [0.524, 1] \cap [0.546, 1.117] \\ &= [0.546, 1.0] \end{aligned}$$

Example of Pruning Operations

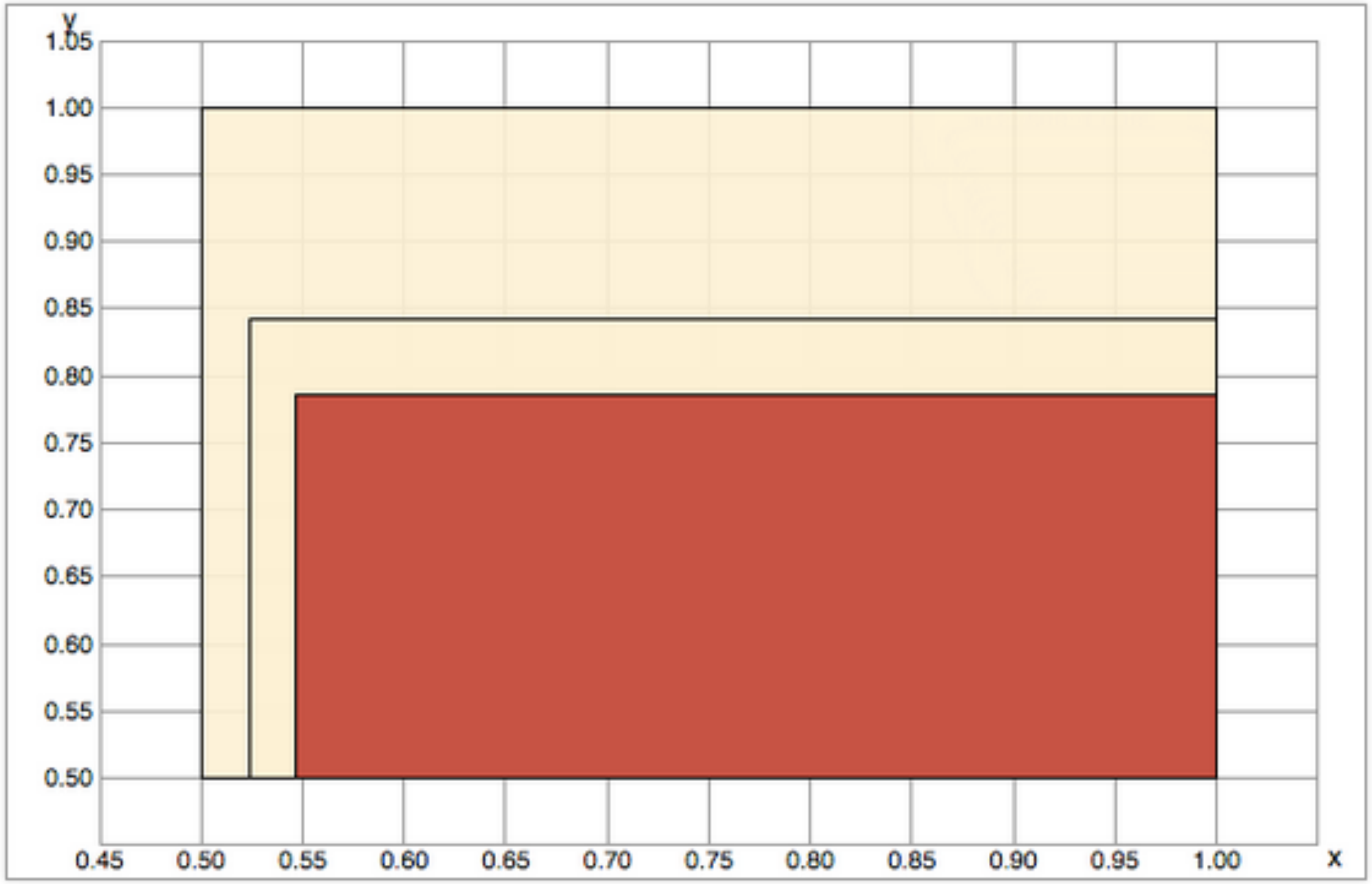
$$\exists x, y \in [0.5, 1.0] : y = \sin(x) \wedge y = \mathrm{atan}(x)$$



$$\begin{aligned} y' &= y \cap \mathrm{atan}(x) \\ &= [0.5, 0.841] \cap \mathrm{atan}([0.546, 1.0]) \\ &= [0.5, 0.841] \cap [0.5, 1.0] \\ &= [0.5, 0.785] \end{aligned}$$

Example of Pruning Operations

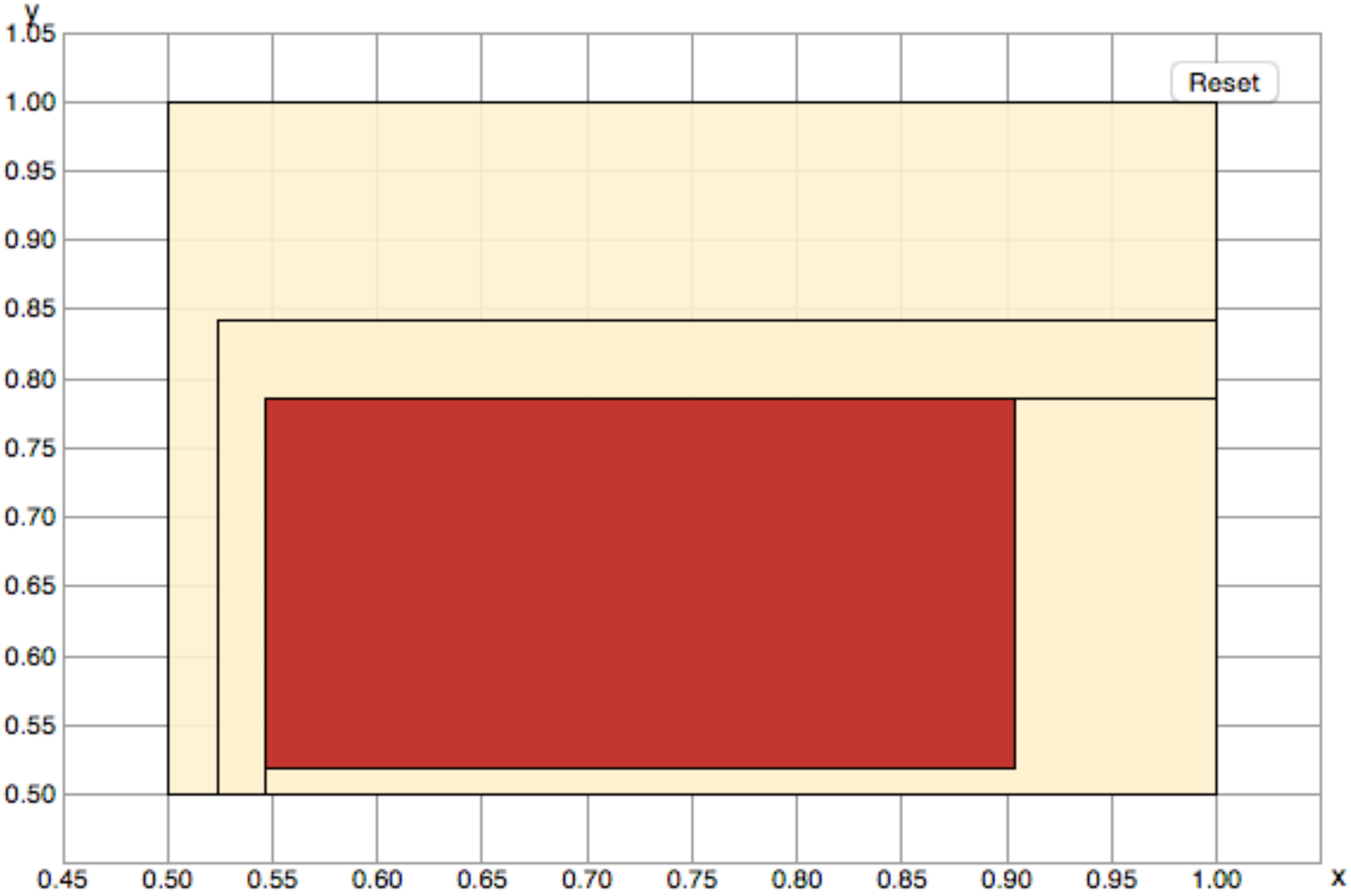
$$\exists x, y \in [0.5, 1.0] : y = \sin(x) \wedge y = \mathrm{atan}(x)$$



$$x : [0.524, 1.0], y : [0.5, 0.785]$$

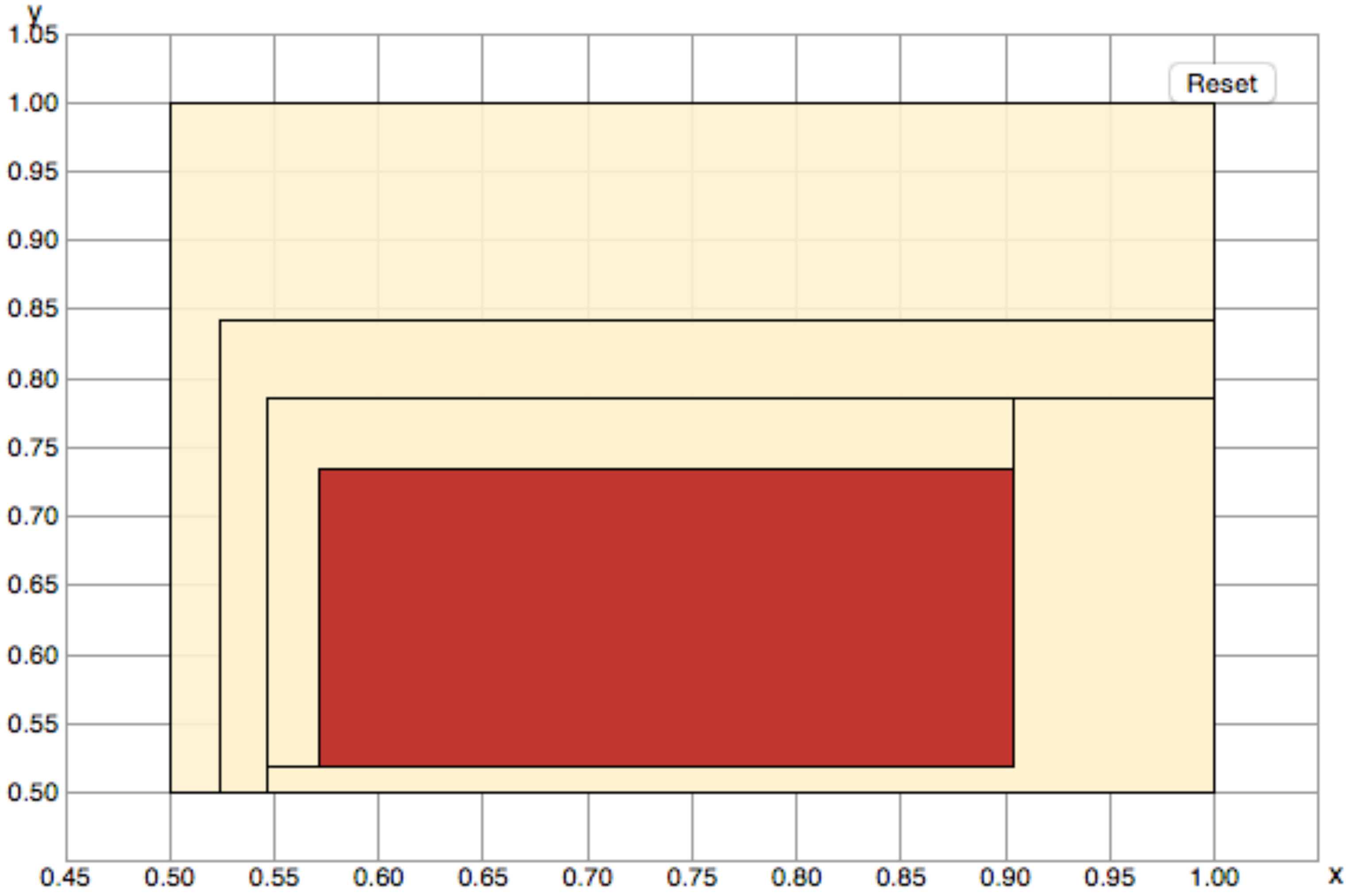
Example of Pruning Operations

$$\exists x, y \in [0.5, 1.0] : y = \sin(x) \wedge y = \operatorname{atan}(x)$$



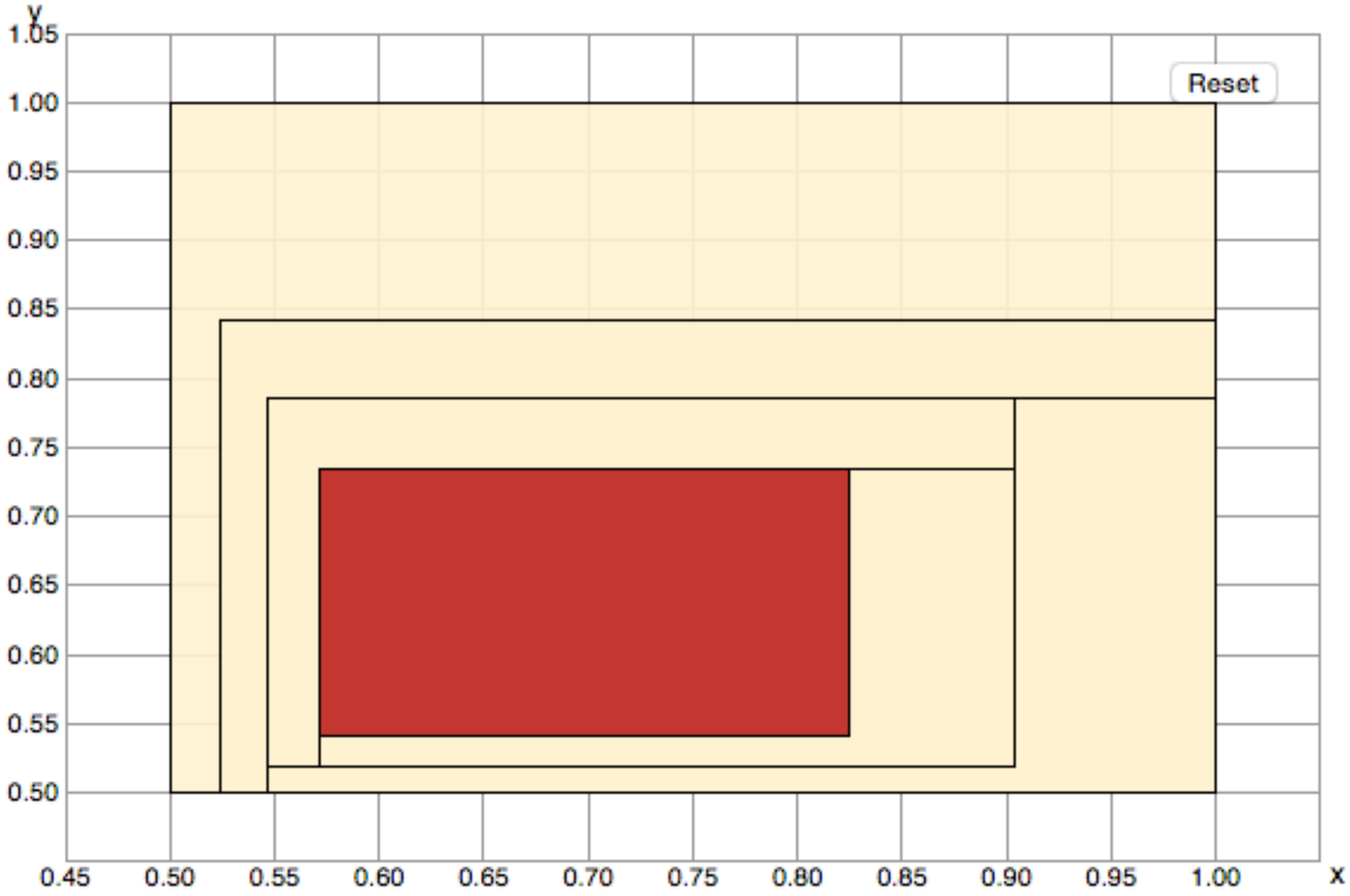
Example of Pruning Operations

$$\exists x, y \in [0.5, 1.0] : y = \sin(x) \wedge y = \operatorname{atan}(x)$$



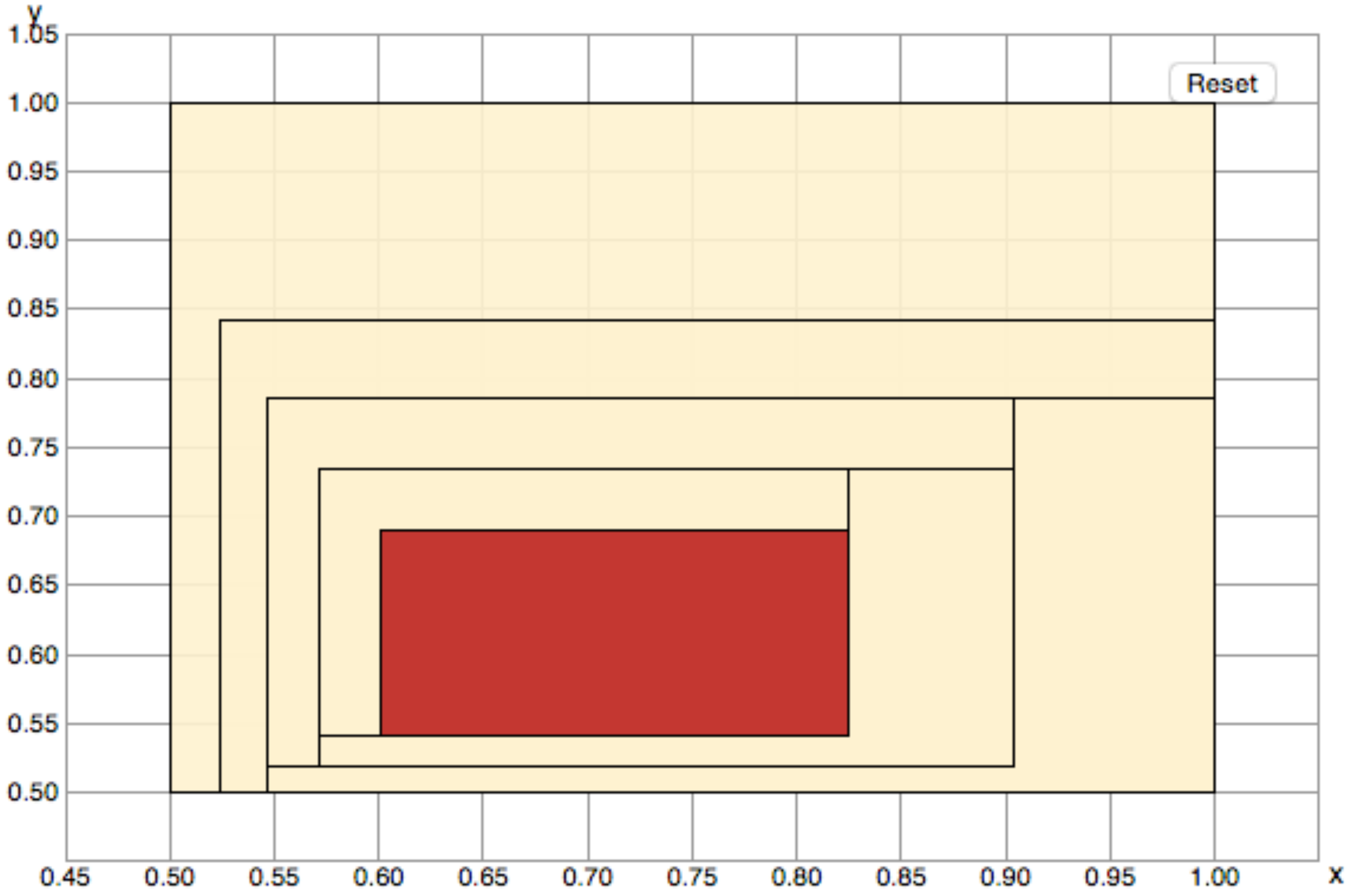
Example of Pruning Operations

$$\exists x, y \in [0.5, 1.0] : y = \sin(x) \wedge y = \operatorname{atan}(x)$$



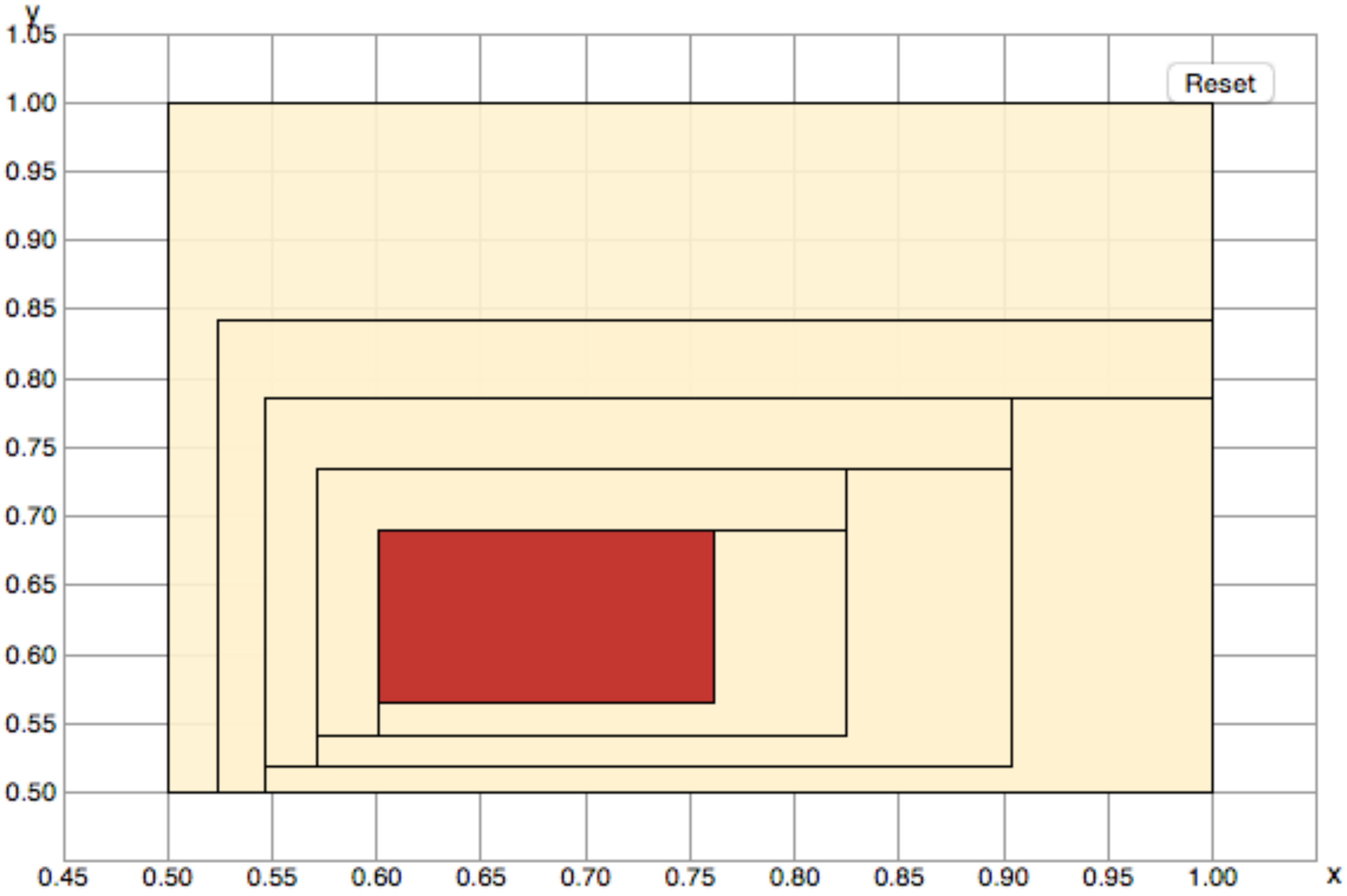
Example of Pruning Operations

$$\exists x, y \in [0.5, 1.0] : y = \sin(x) \wedge y = \operatorname{atan}(x)$$



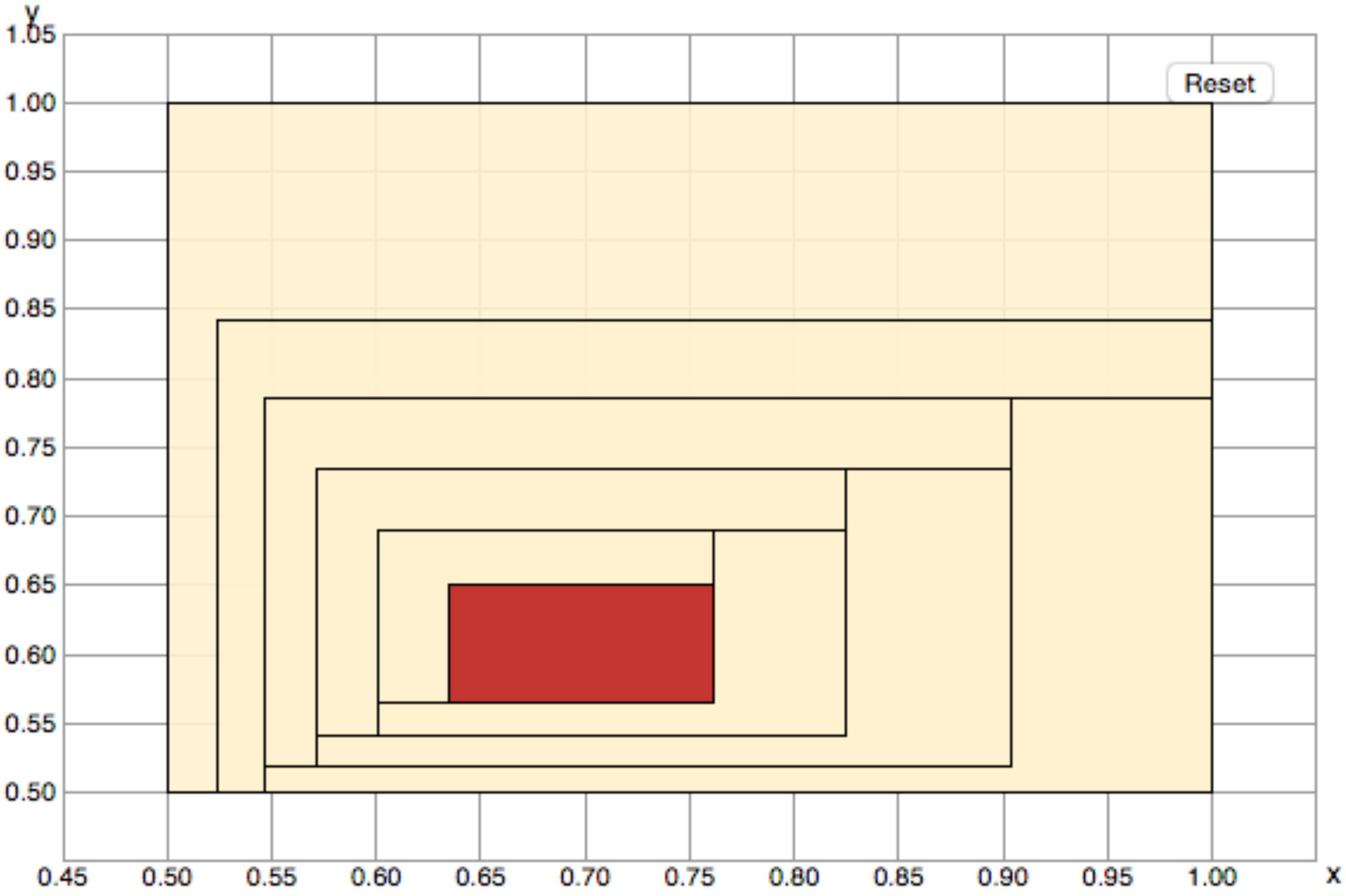
Example of Pruning Operations

$$\exists x, y \in [0.5, 1.0] : y = \sin(x) \wedge y = \mathrm{atan}(x)$$



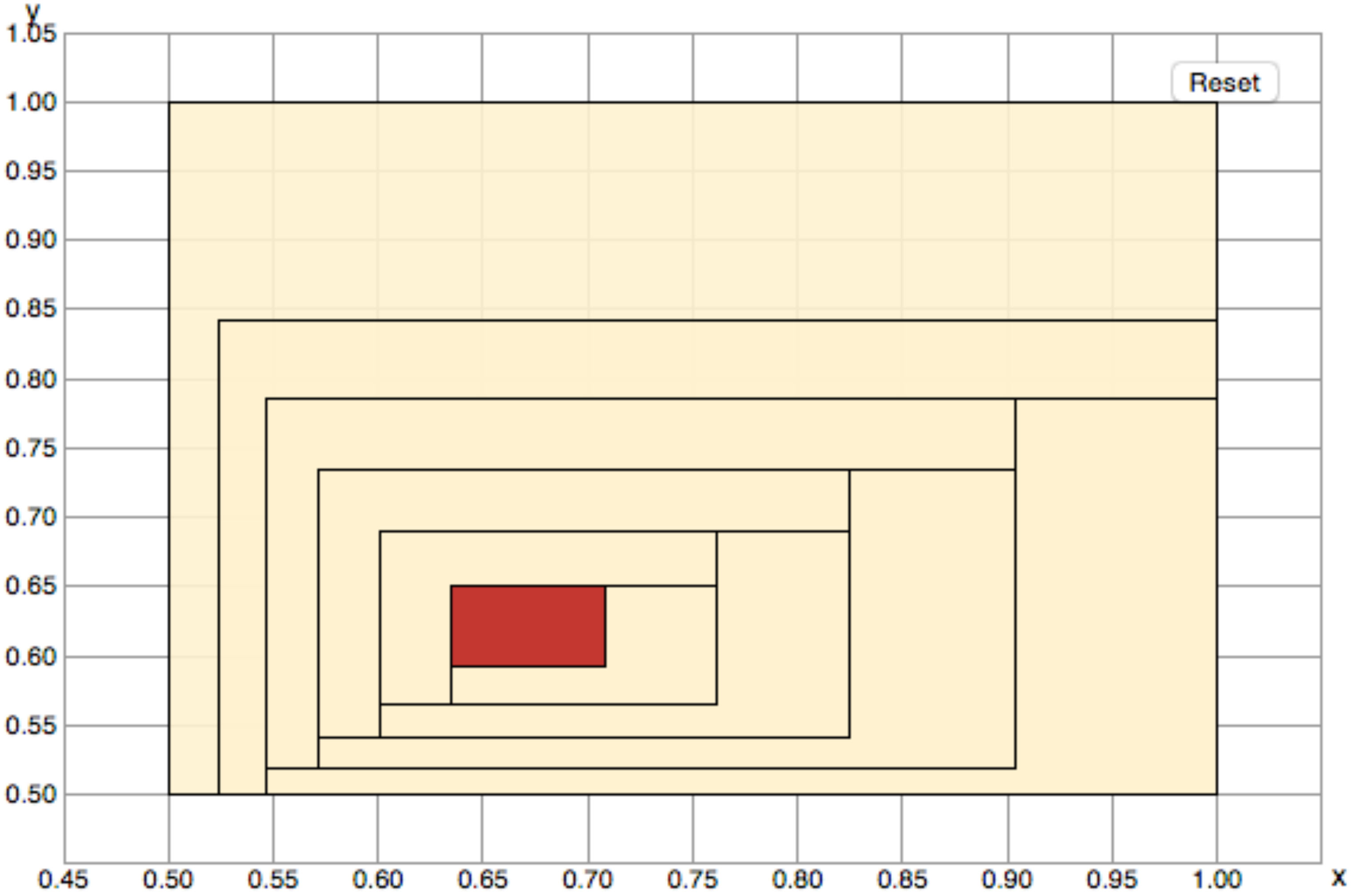
Example of Pruning Operations

$$\exists x, y \in [0.5, 1.0] : y = \sin(x) \land y = \operatorname{atan}(x)$$



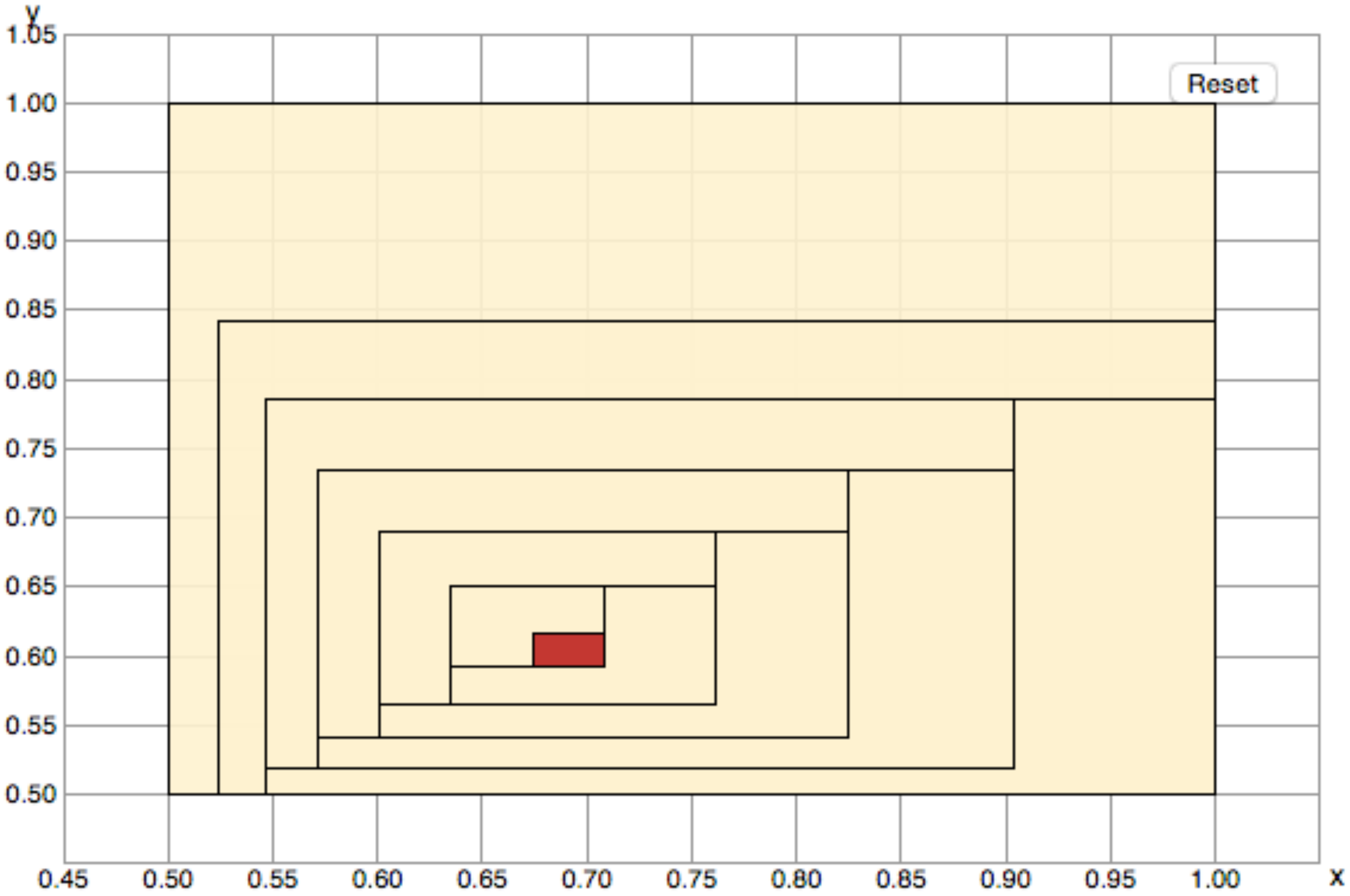
Example of Pruning Operations

$$\exists x, y \in [0.5, 1.0] : y = \sin(x) \land y = \mathrm{atan}(x)$$



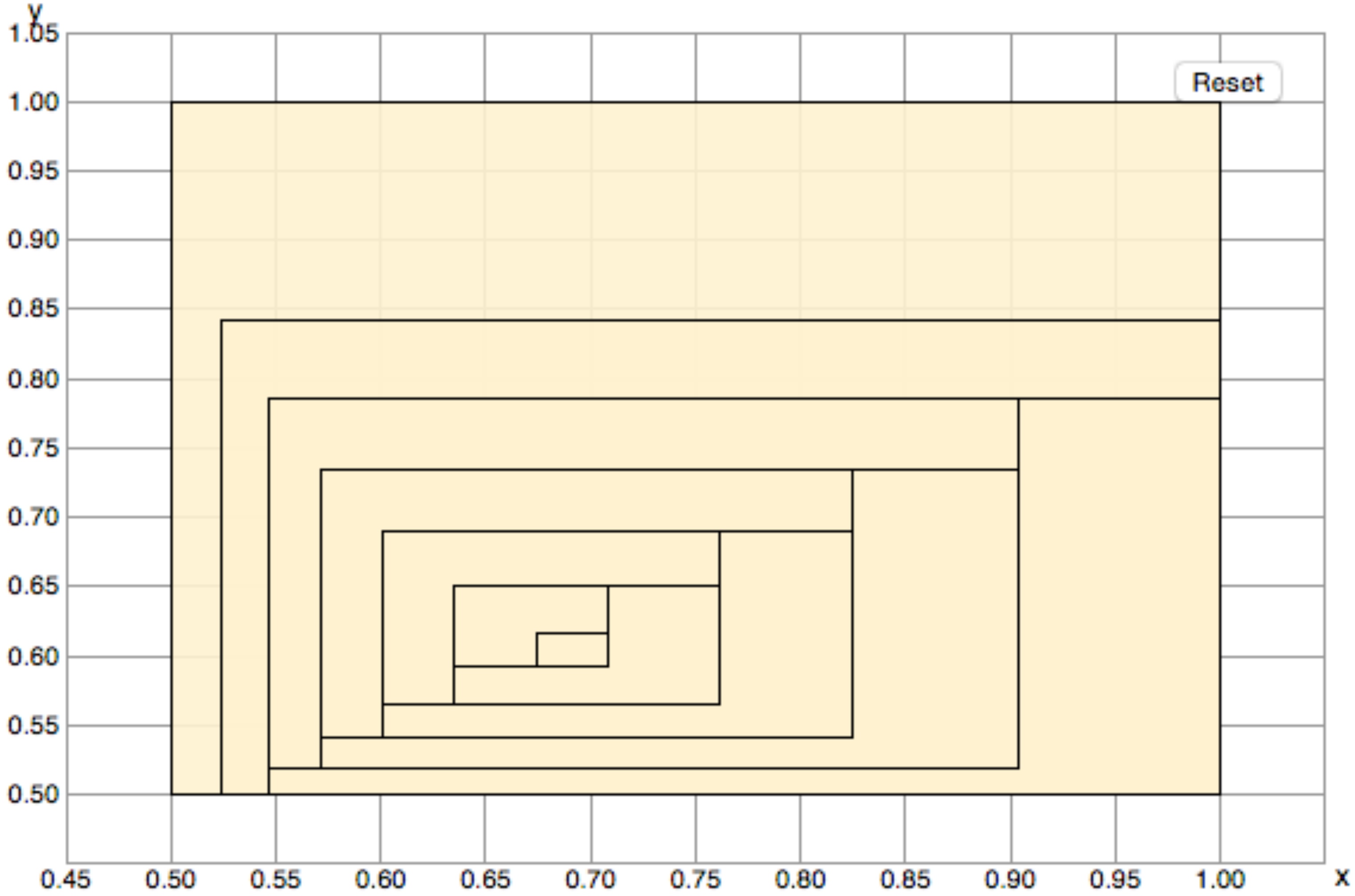
Example of Pruning Operations

$$\exists x, y \in [0.5, 1.0] : y = \sin(x) \wedge y = \operatorname{atan}(x)$$



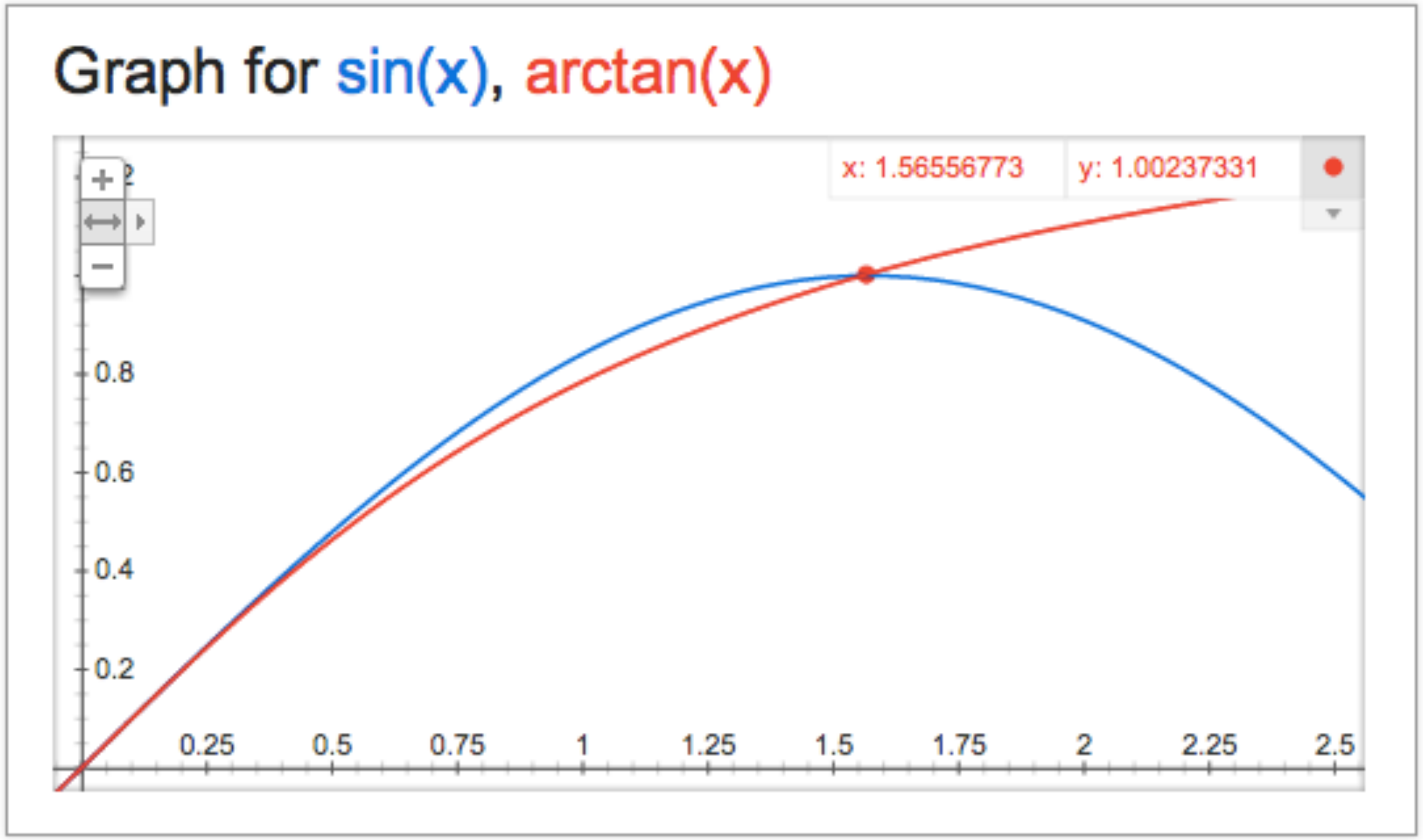
Example of Pruning Operations

$$\exists x, y \in [0.5, 1.0] : y = \sin(x) \wedge y = \operatorname{atan}(x)$$



Unsat

Example of ICP



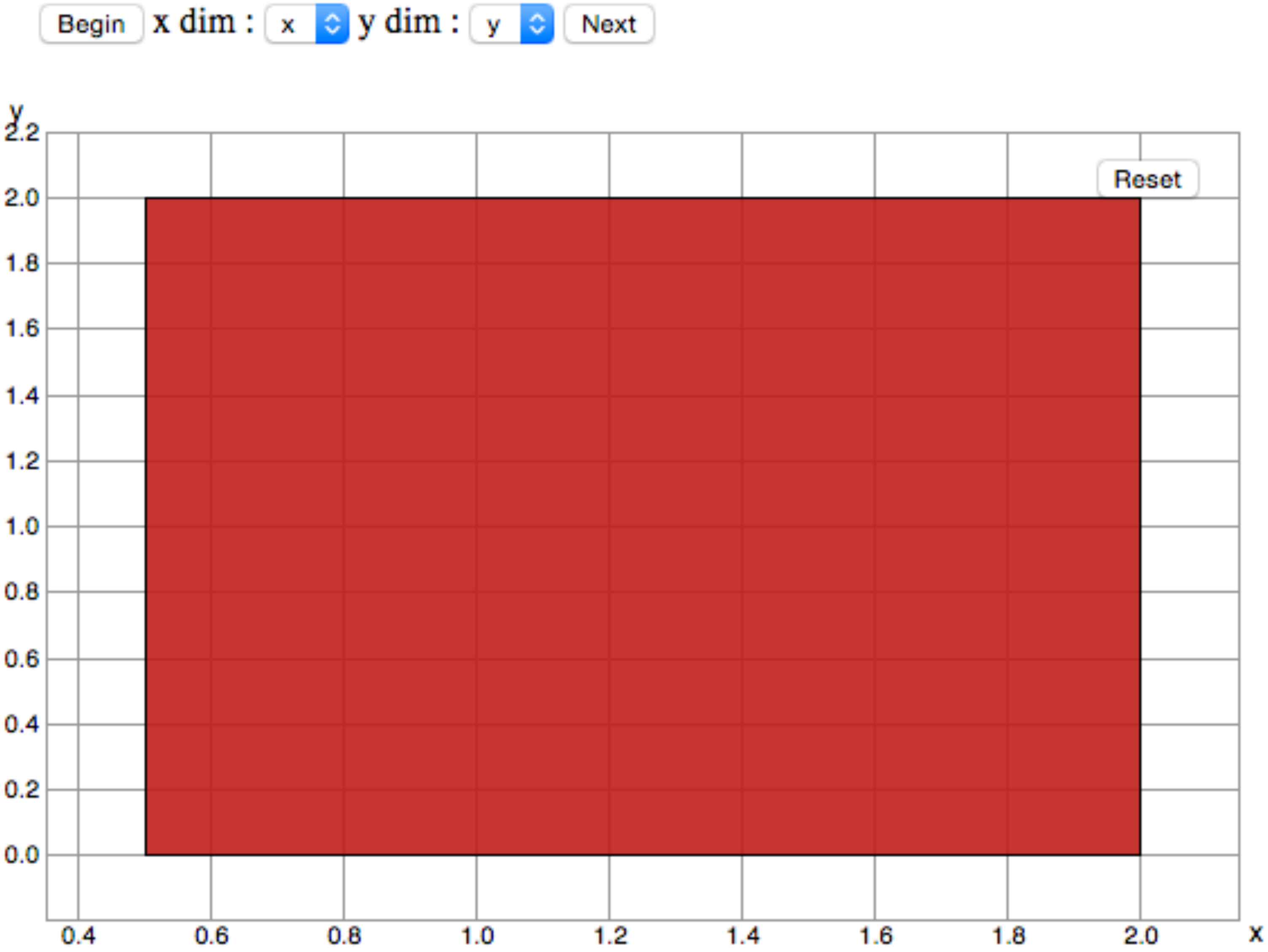
Graph for sin(x), arctan(x)

x: 1.56556773 y: 1.00237331

ANSWER: **SAT**

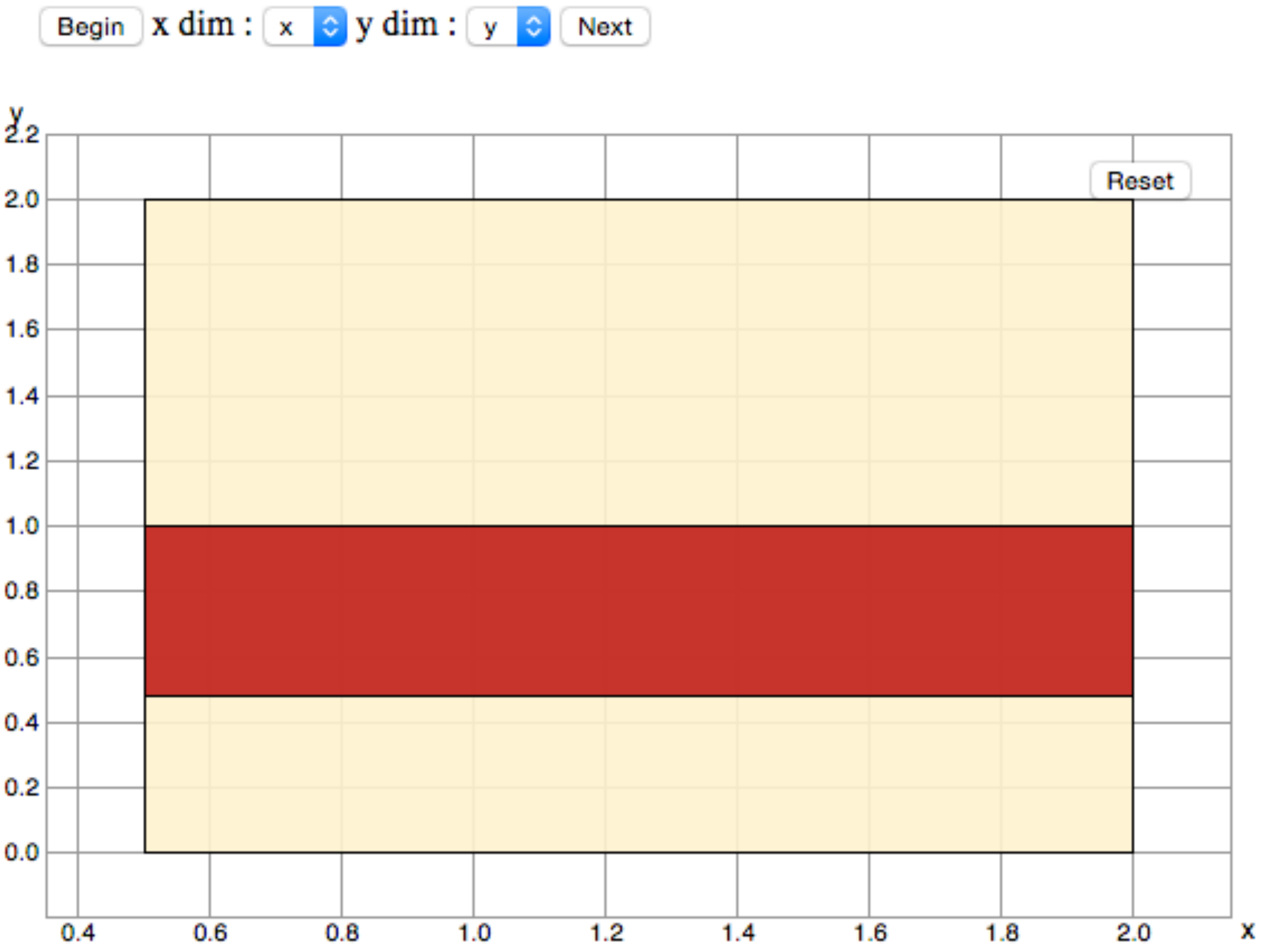
Example of ICP

$$\exists x \in [0.5, 2.0], y \in [0.0, 2.0] : y = \sin(x) \wedge y = \operatorname{atan}(x)$$



Example of ICP

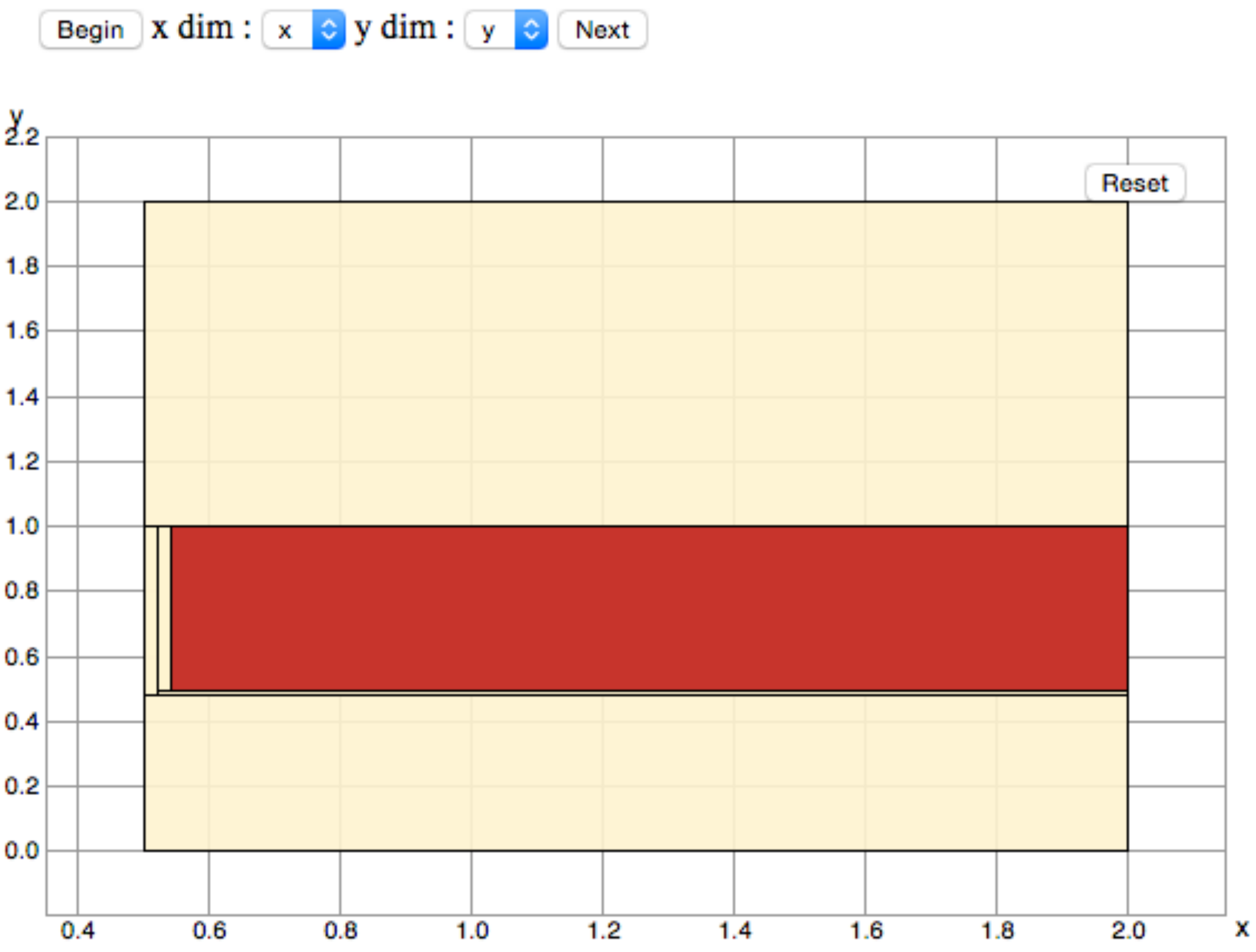
$$\exists x \in [0.5, 2.0], y \in [0.0, 2.0] : \boxed{y = \sin(x)} \wedge y = \mathrm{atan}(x)$$



Pruning Applied

Example of ICP

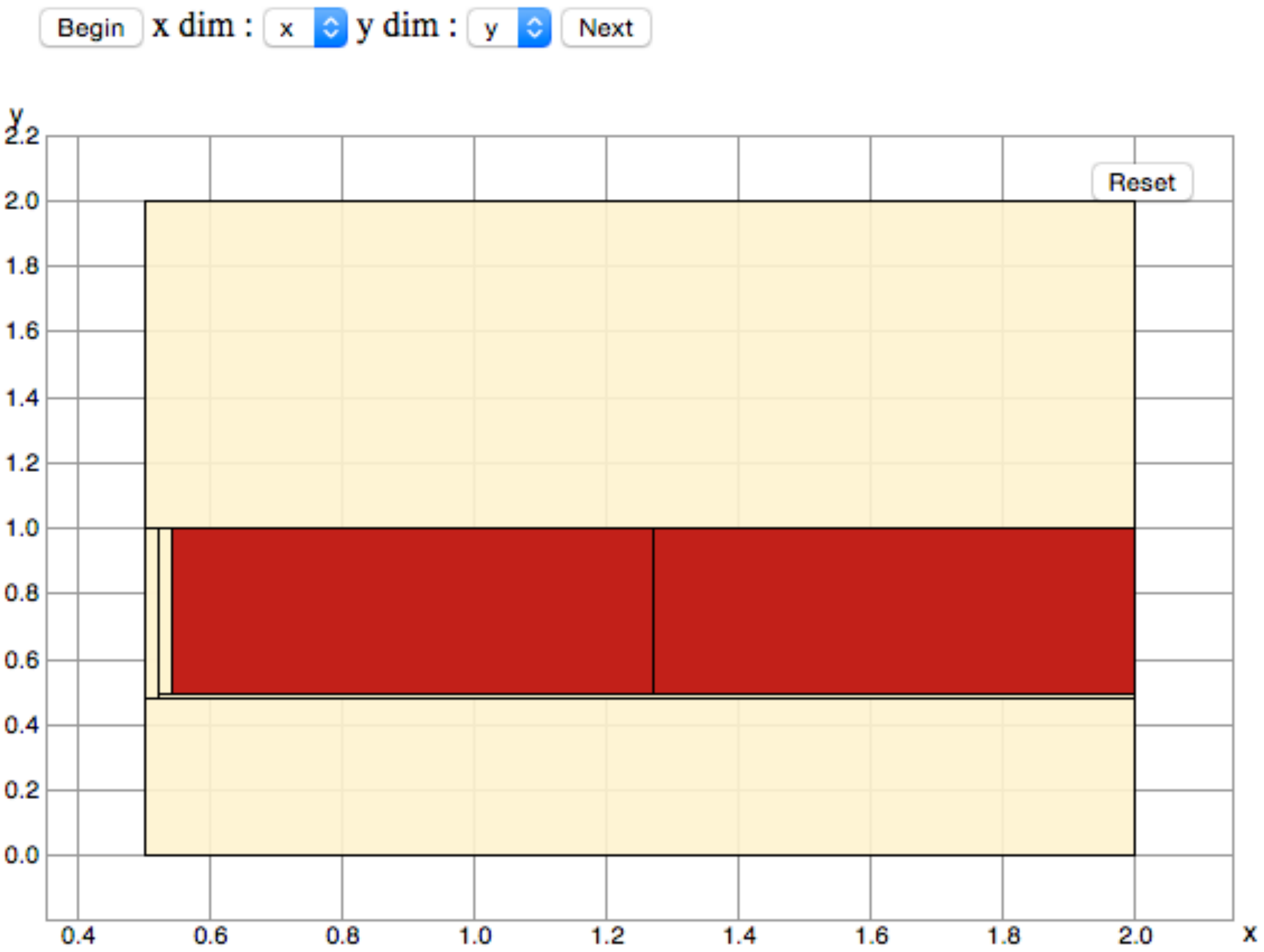
$$\exists x \in [0.5, 2.0], y \in [0.0, 2.0] : y = \sin(x) \wedge y = \operatorname{atan}(x)$$



After steps, pruning reaches a fixed point.

Example of ICP

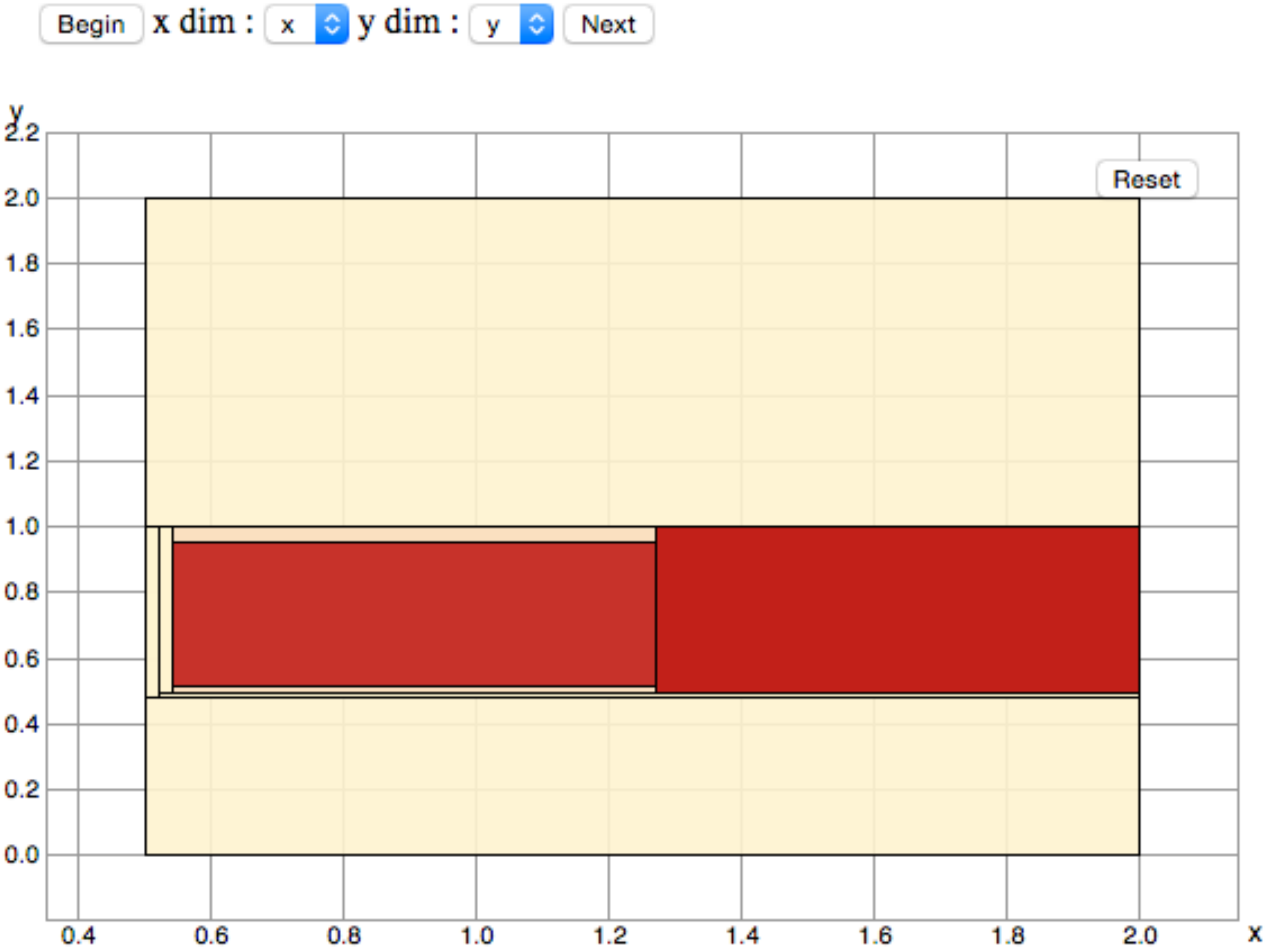
$$\exists x \in [0.5, 2.0], y \in [0.0, 2.0] : y = \sin(x) \wedge y = \operatorname{atan}(x)$$



Branching on X

Example of ICP

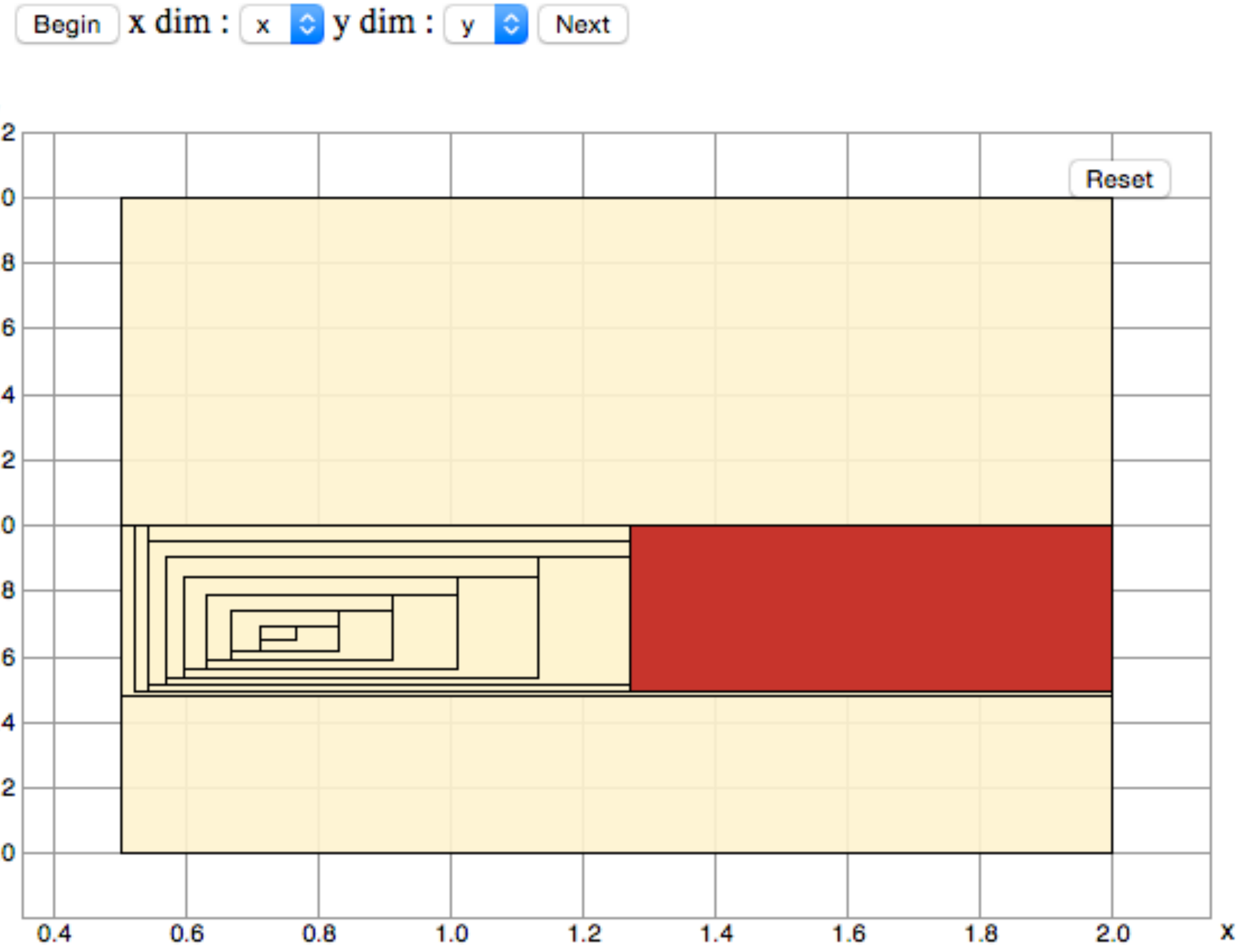
$$\exists x \in [0.5, 2.0], y \in [0.0, 2.0] : y = \sin(x) \land y = \operatorname{atan}(x)$$



Apply Pruning on the Left-hand Box

Example of ICP

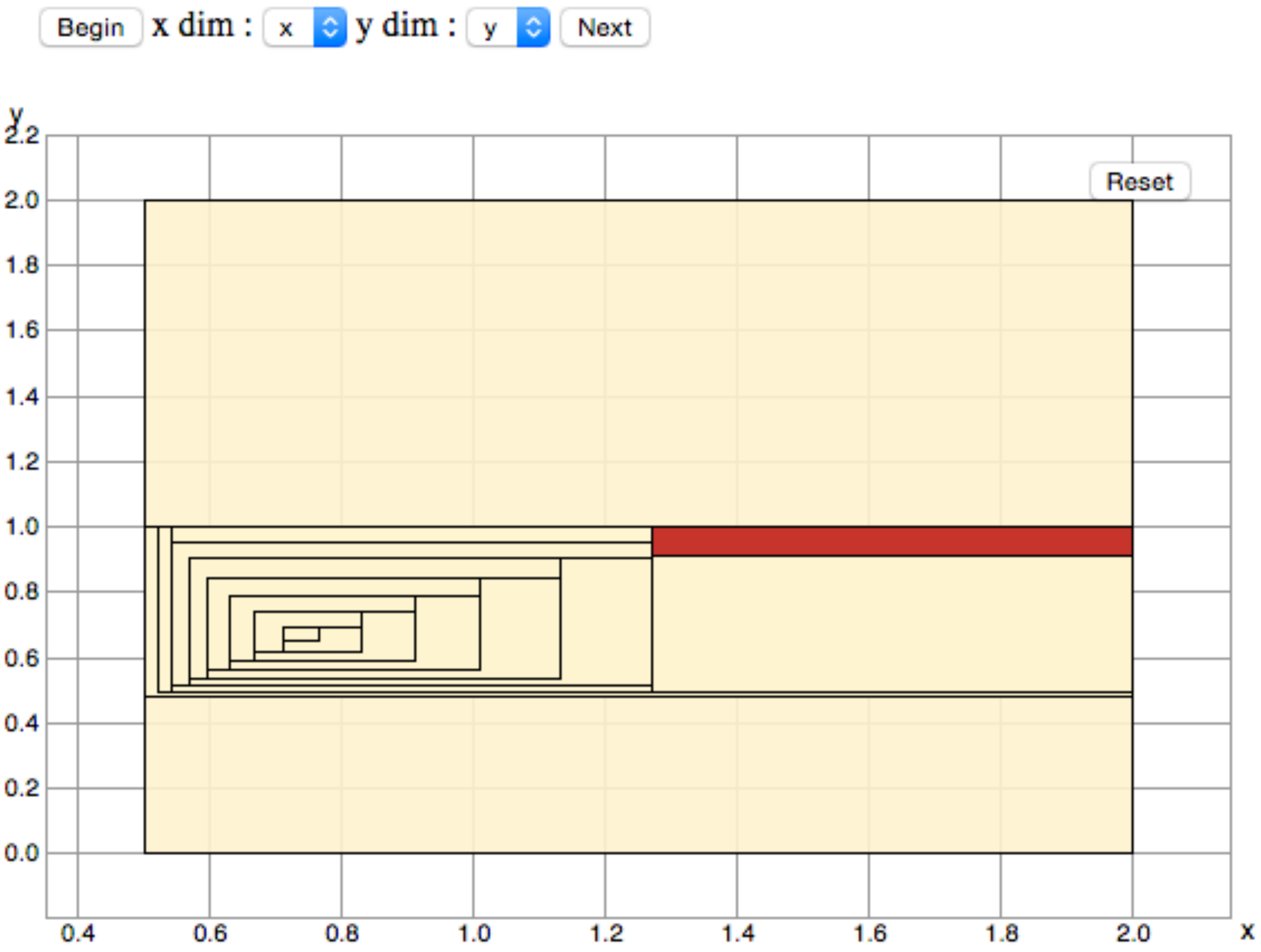
$$\exists x \in [0.5, 2.0], y \in [0.0, 2.0] : y = \sin(x) \wedge y = \operatorname{atan}(x)$$



After pruning steps,
it shows that left-hand box contains NO solution.

Example of ICP

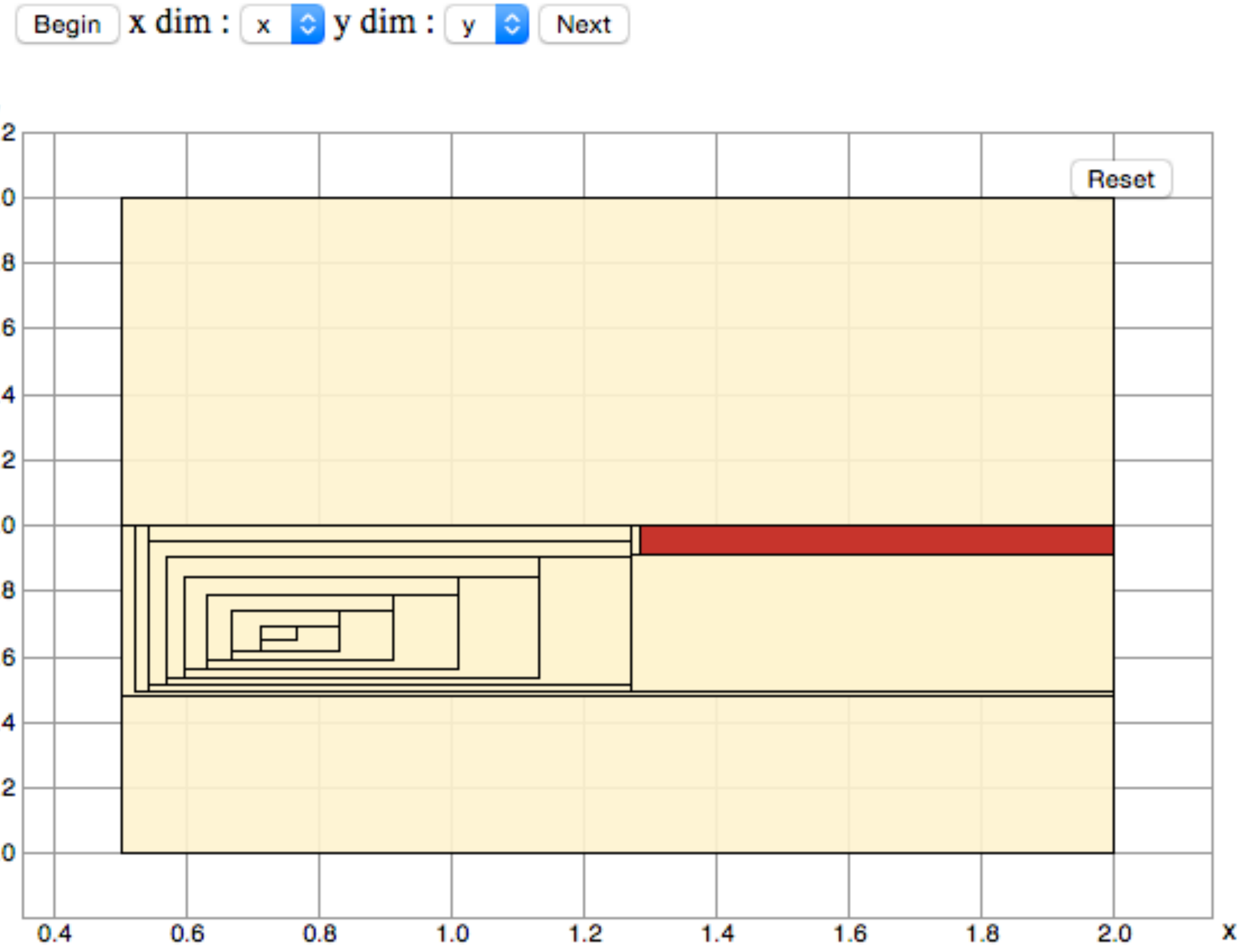
$$\exists x \in [0.5, 2.0], y \in [0.0, 2.0] : y = \sin(x) \wedge y = \operatorname{atan}(x)$$



Apply Pruning on the Right-hand Box

Example of ICP

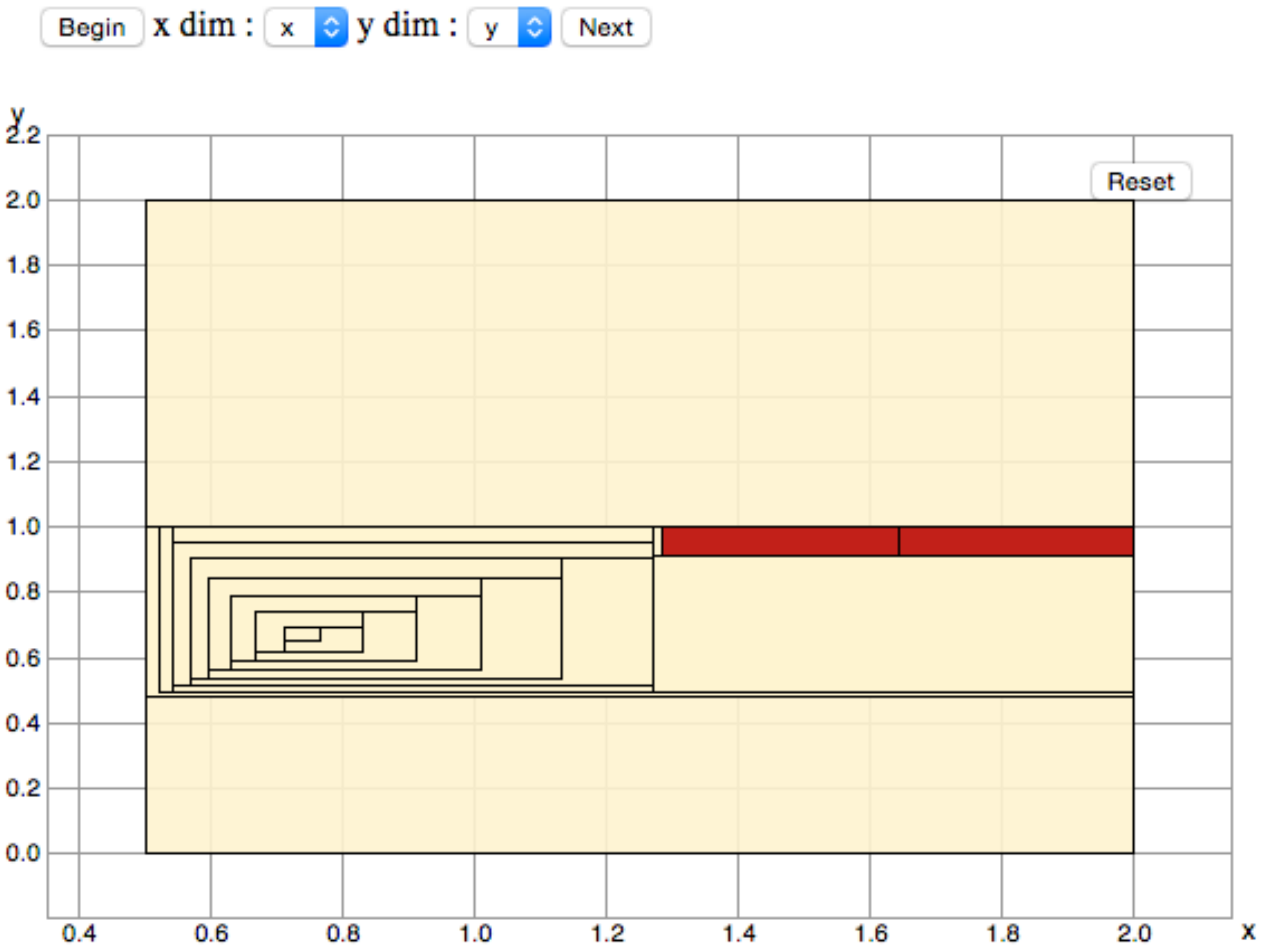
$$\exists x \in [0.5, 2.0], y \in [0.0, 2.0] : y = \sin(x) \land y = \operatorname{atan}(x)$$



Apply Pruning on the Right-hand Box

Example of ICP

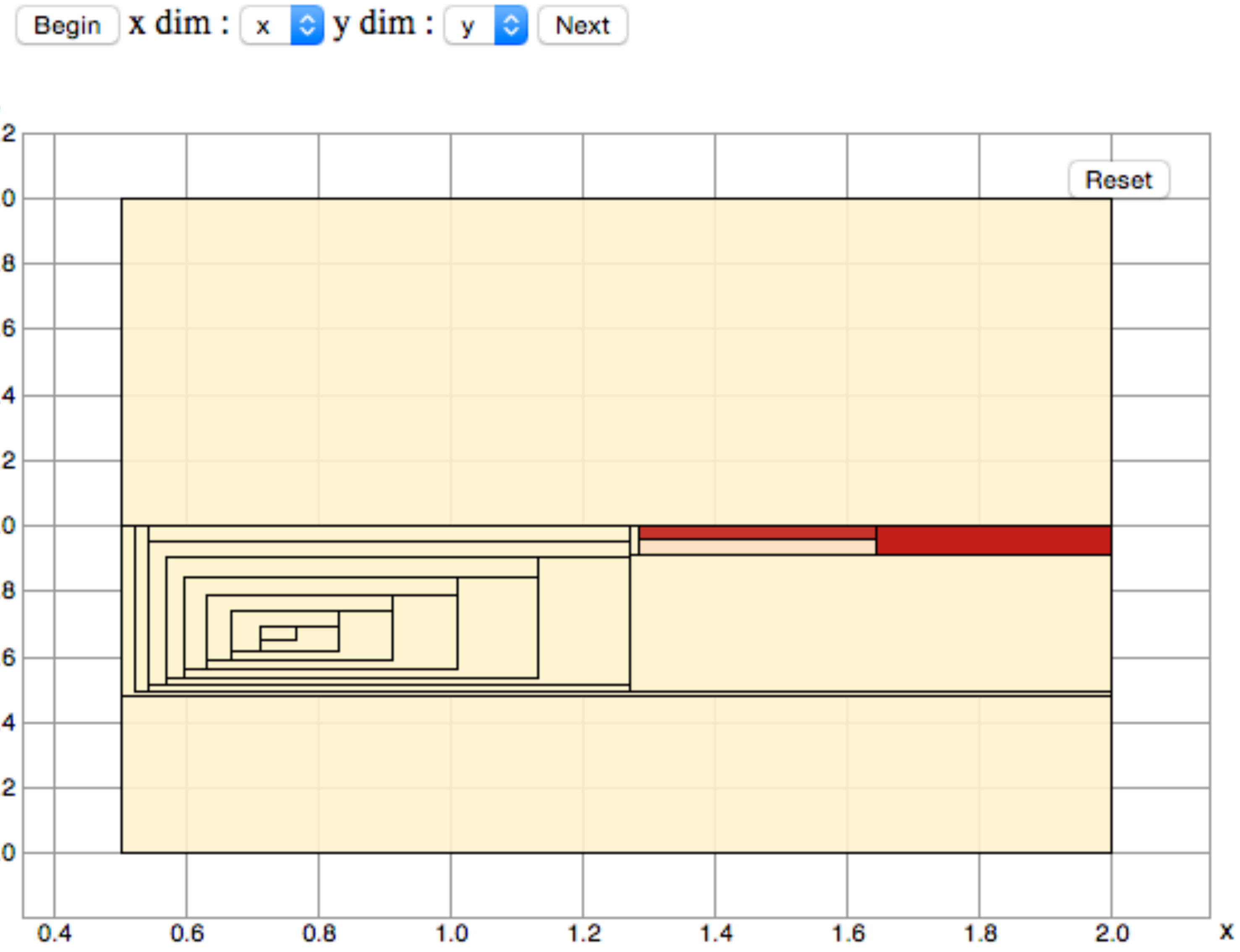
$$\exists x \in [0.5, 2.0], y \in [0.0, 2.0] : y = \sin(x) \wedge y = \operatorname{atan}(x)$$



Branching on X

Example of ICP

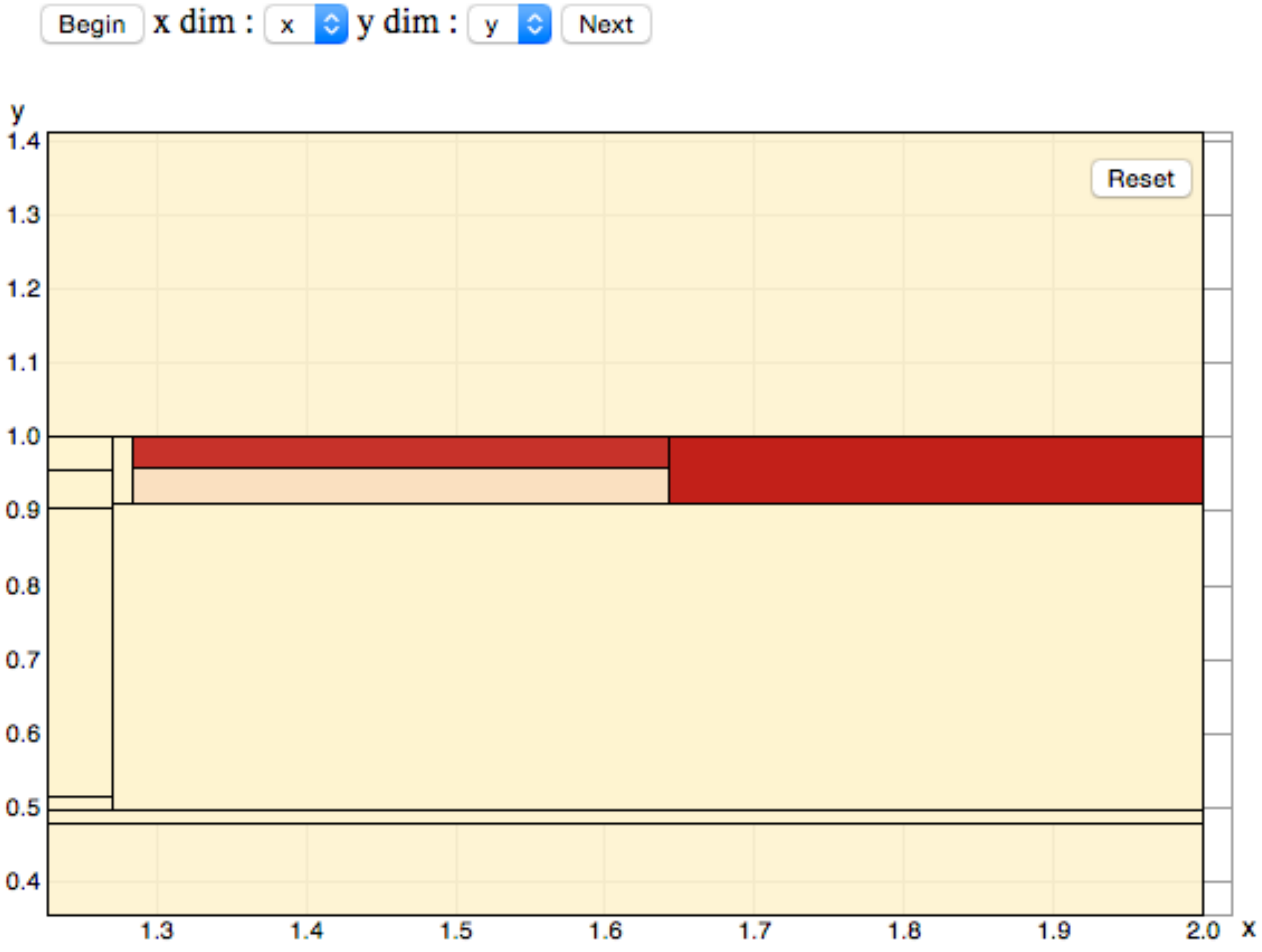
$$\exists x \in [0.5, 2.0], y \in [0.0, 2.0] : y = \sin(x) \wedge y = \mathrm{atan}(x)$$



Apply Pruning on the Right-hand Box

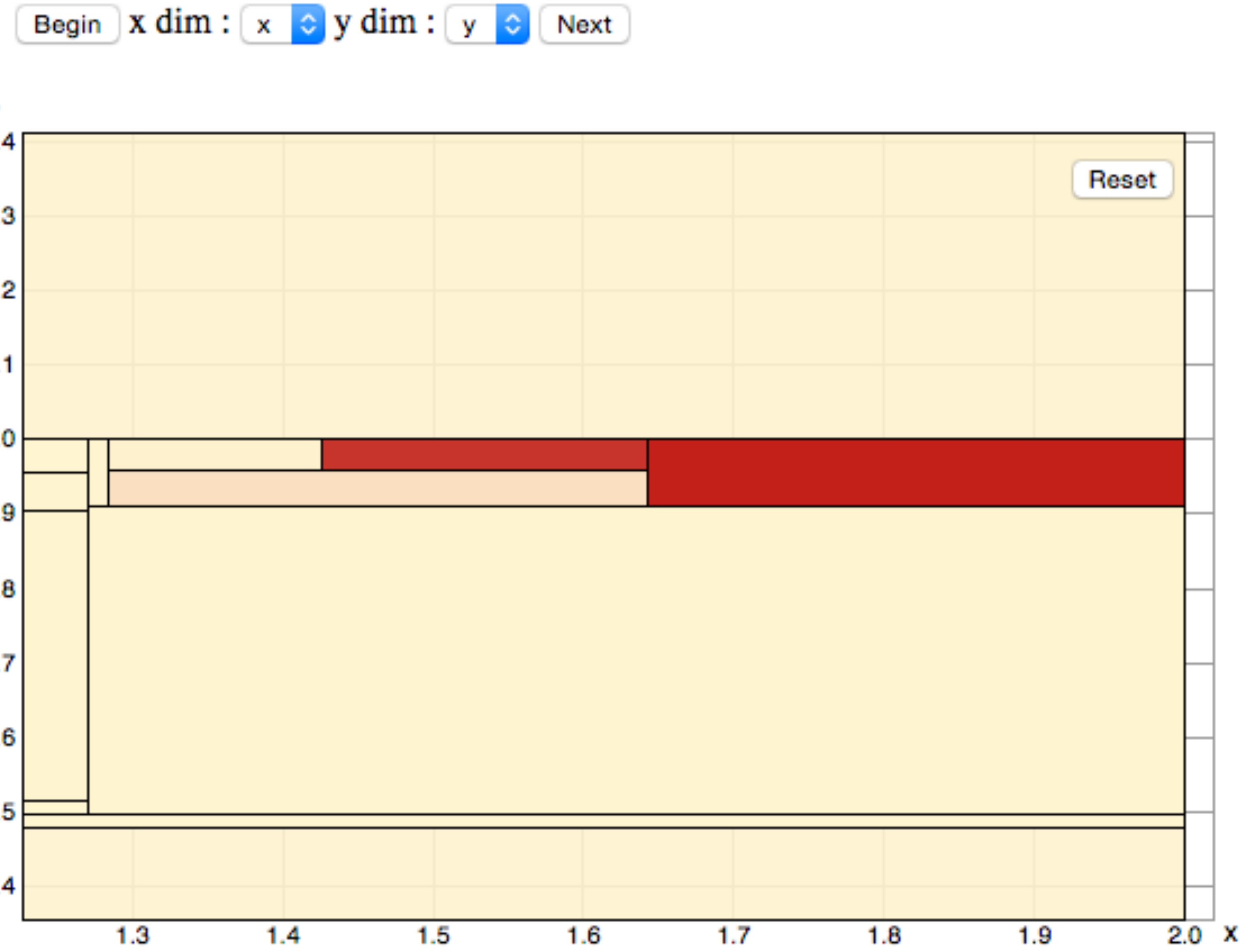
Example of ICP

$$\exists x \in [0.5, 2.0], y \in [0.0, 2.0] : y = \sin(x) \wedge y = \operatorname{atan}(x)$$



Example of ICP

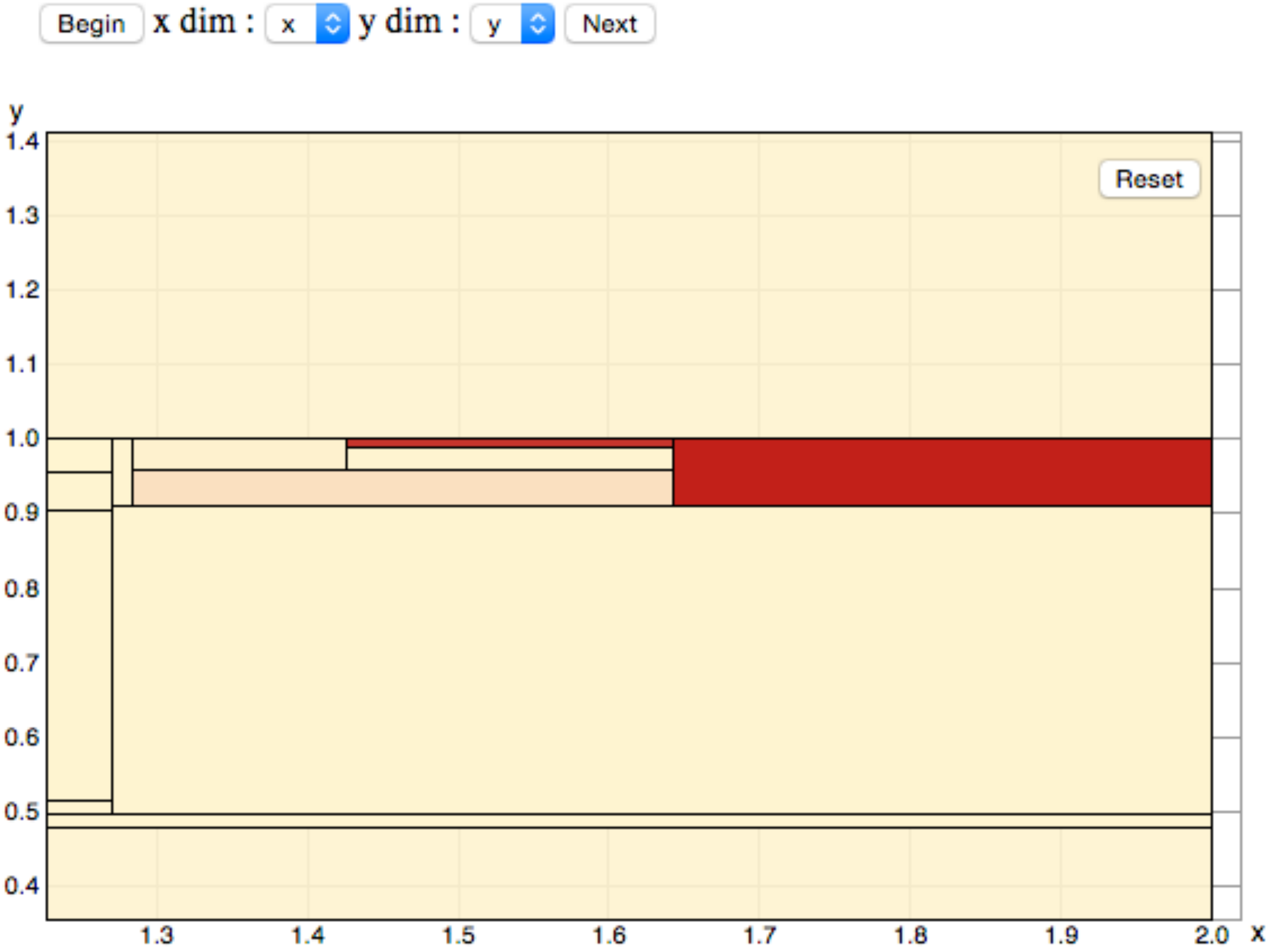
$$\exists x \in [0.5, 2.0], y \in [0.0, 2.0] : y = \sin(x) \wedge y = \operatorname{atan}(x)$$



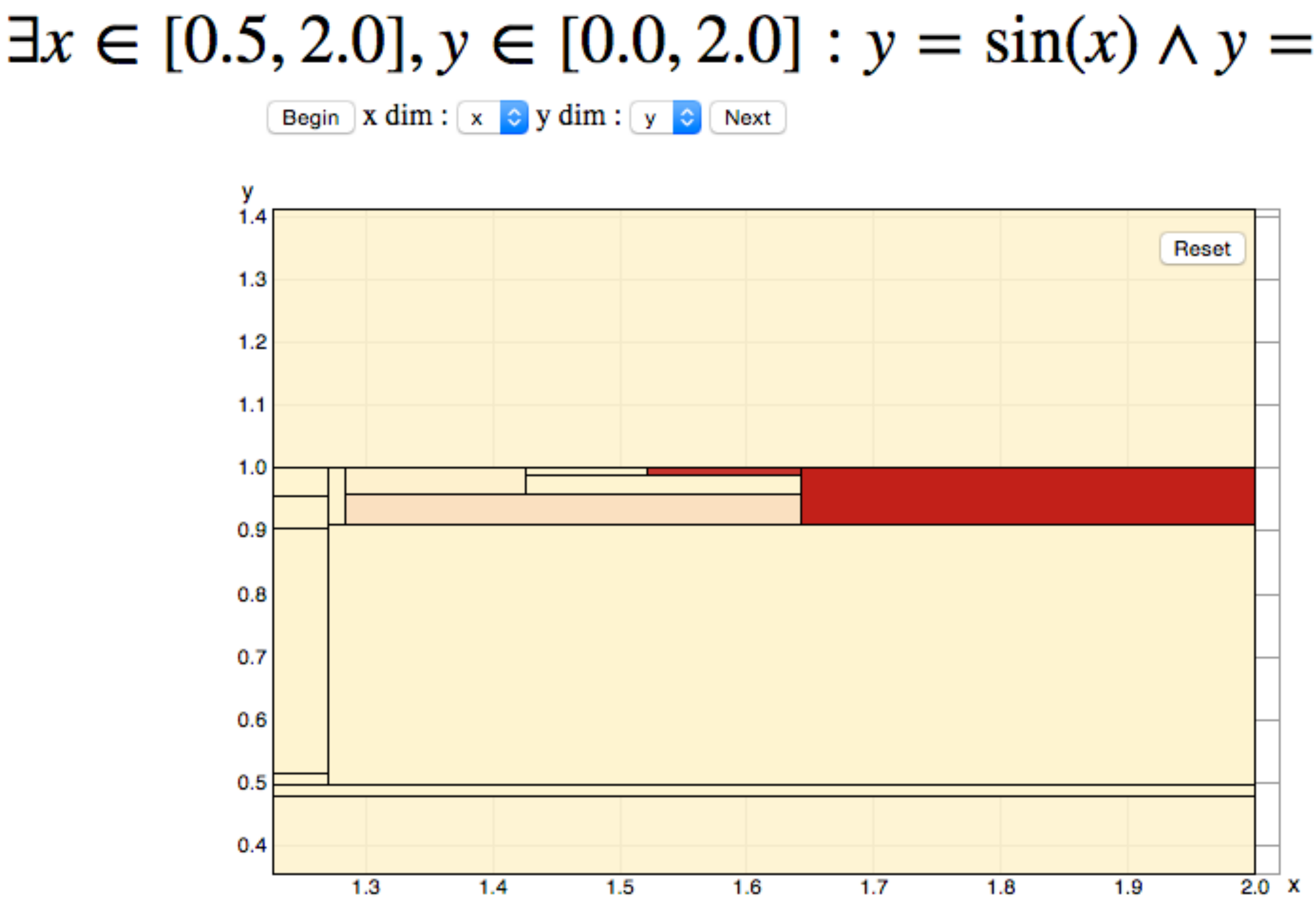
x dim : x y dim : y

Example of ICP

$$\exists x \in [0.5, 2.0], y \in [0.0, 2.0] : y = \sin(x) \wedge y = \mathrm{atan}(x)$$



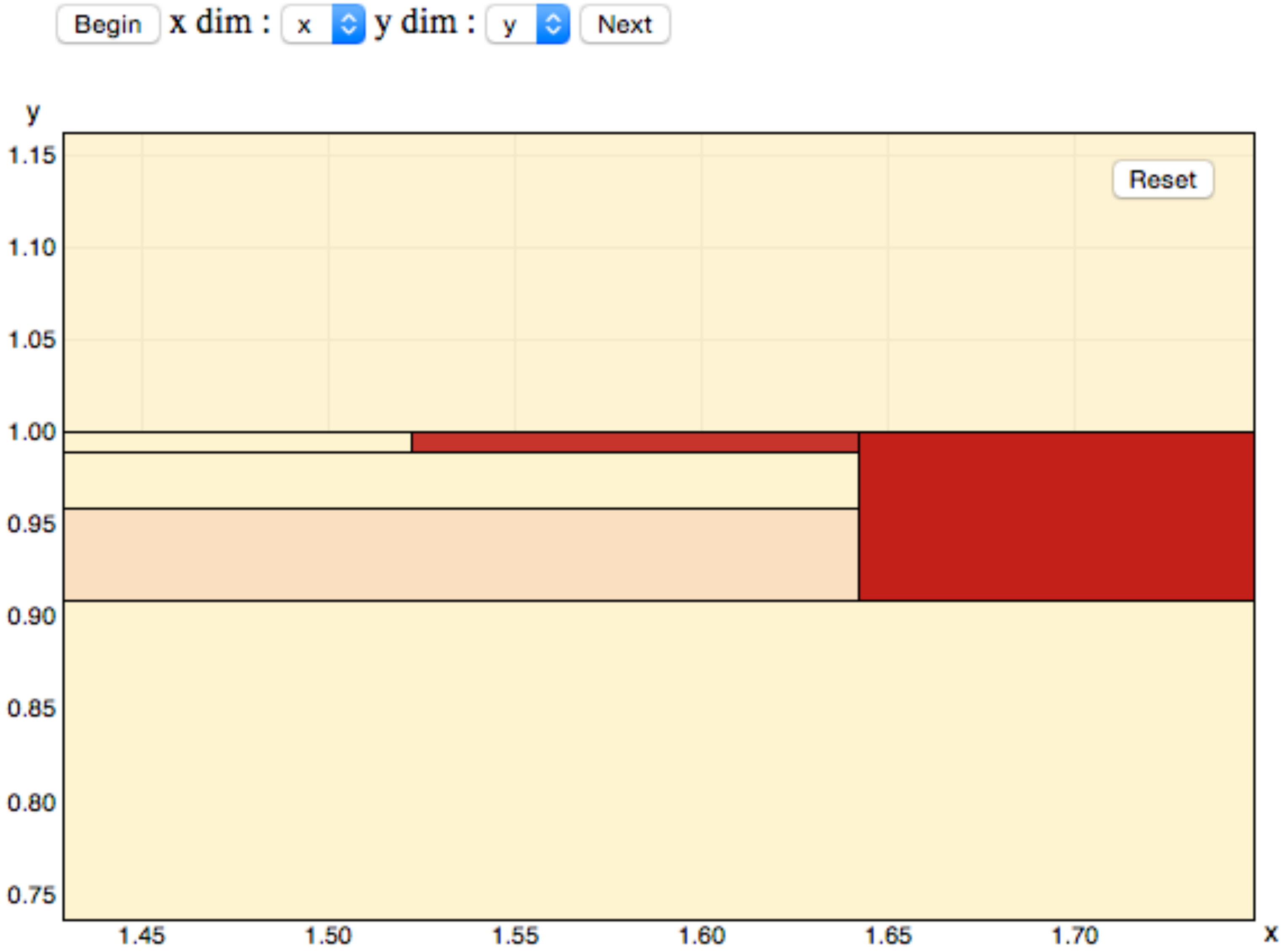
Example of ICP



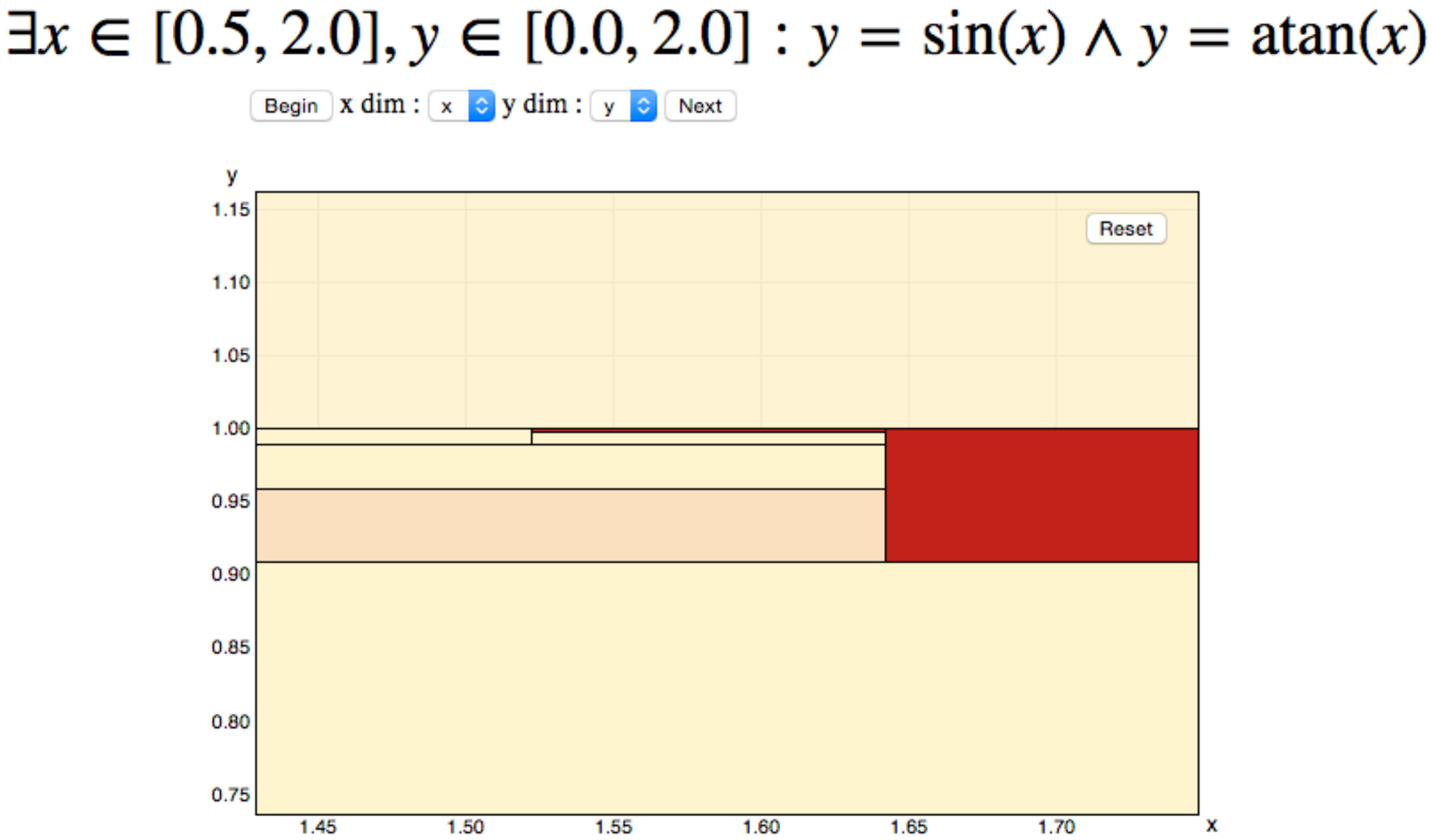
$$\exists x \in [0.5, 2.0], y \in [0.0, 2.0] : y = \sin(x) \land y = \operatorname{atan}(x)$$

Example of ICP

$$\exists x \in [0.5, 2.0], y \in [0.0, 2.0] : y = \sin(x) \wedge y = \mathrm{atan}(x)$$



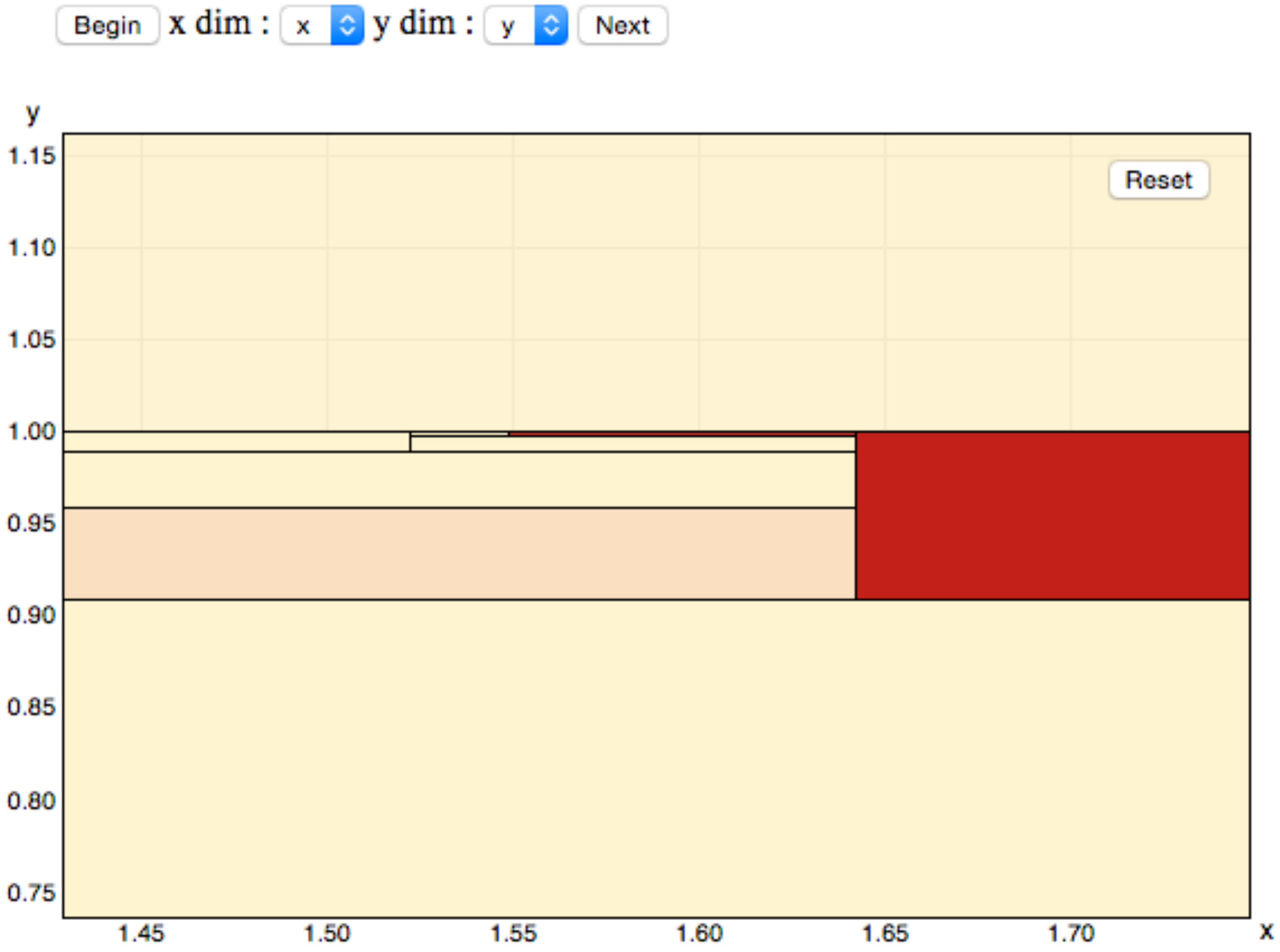
Example of ICP



$$\exists x \in [0.5, 2.0], y \in [0.0, 2.0] : y = \sin(x) \wedge y = \operatorname{atan}(x)$$

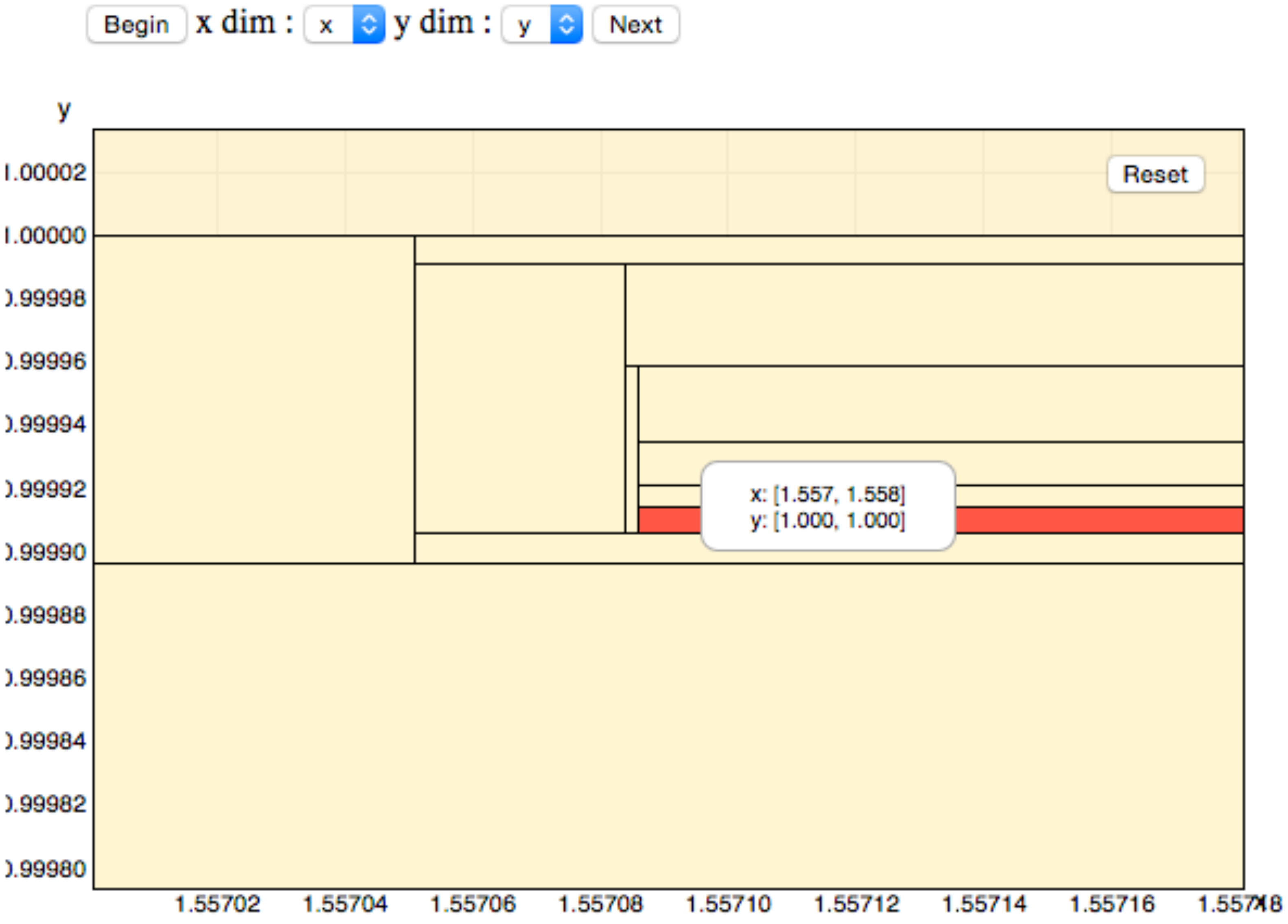
Example of ICP

$$\exists x \in [0.5, 2.0], y \in [0.0, 2.0] : y = \sin(x) \wedge y = \operatorname{atan}(x)$$



Example of ICP

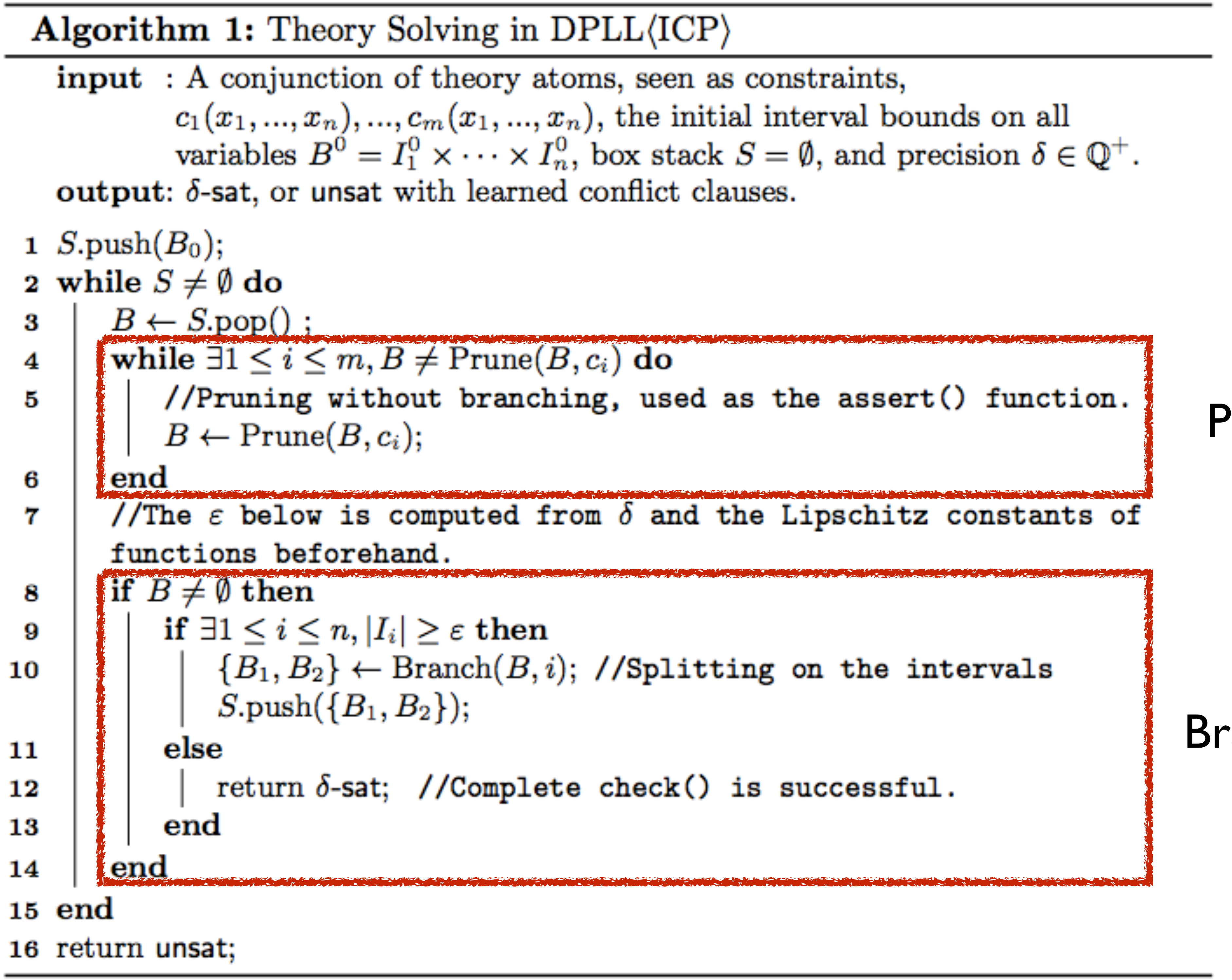
$$\exists x \in [0.5, 2.0], y \in [0.0, 2.0] : y = \sin(x) \land y = \mathrm{atan}(x)$$



Found a small enough Box (width <= 0.001)

Answer: delta-SAT

Main Algorithm of ICP



Algorithm 1: Theory Solving in DPLL$\langle$ICP$\rangle$

input : A conjunction of theory atoms, seen as constraints, $c_1(x_1, ..., x_n), ..., c_m(x_1, ..., x_n)$, the initial interval bounds on all variables $B^0 = I_1^0 \times \cdots \times I_n^0$, box stack $S = \emptyset$, and precision $\delta \in \mathbb{Q}^+$.

output: δ-sat, or unsat with learned conflict clauses.

```
$S$.push($B_0$);
while $S \neq \emptyset$ do
    $B \leftarrow S$.pop() ;
    while $\exists 1 \leq i \leq m, B \neq \mathrm{Prune}(B, c_i)$ do
        //Pruning without branching, used as the assert() function.
        $B \leftarrow \mathrm{Prune}(B, c_i)$;
    end
    //The $\varepsilon$ below is computed from $\delta$ and the Lipschitz constants of
    functions beforehand.
    if $B \neq \emptyset$ then
        if $\exists 1 \leq i \leq n, |I_i| \geq \varepsilon$ then
            $\{B_1, B_2\} \leftarrow \mathrm{Branch}(B, i)$; //Splitting on the intervals
            $S$.push($\{B_1, B_2\}$);
        else
            return $\delta$-sat; //Complete check() is successful.
        end
    end
end
return unsat;
```

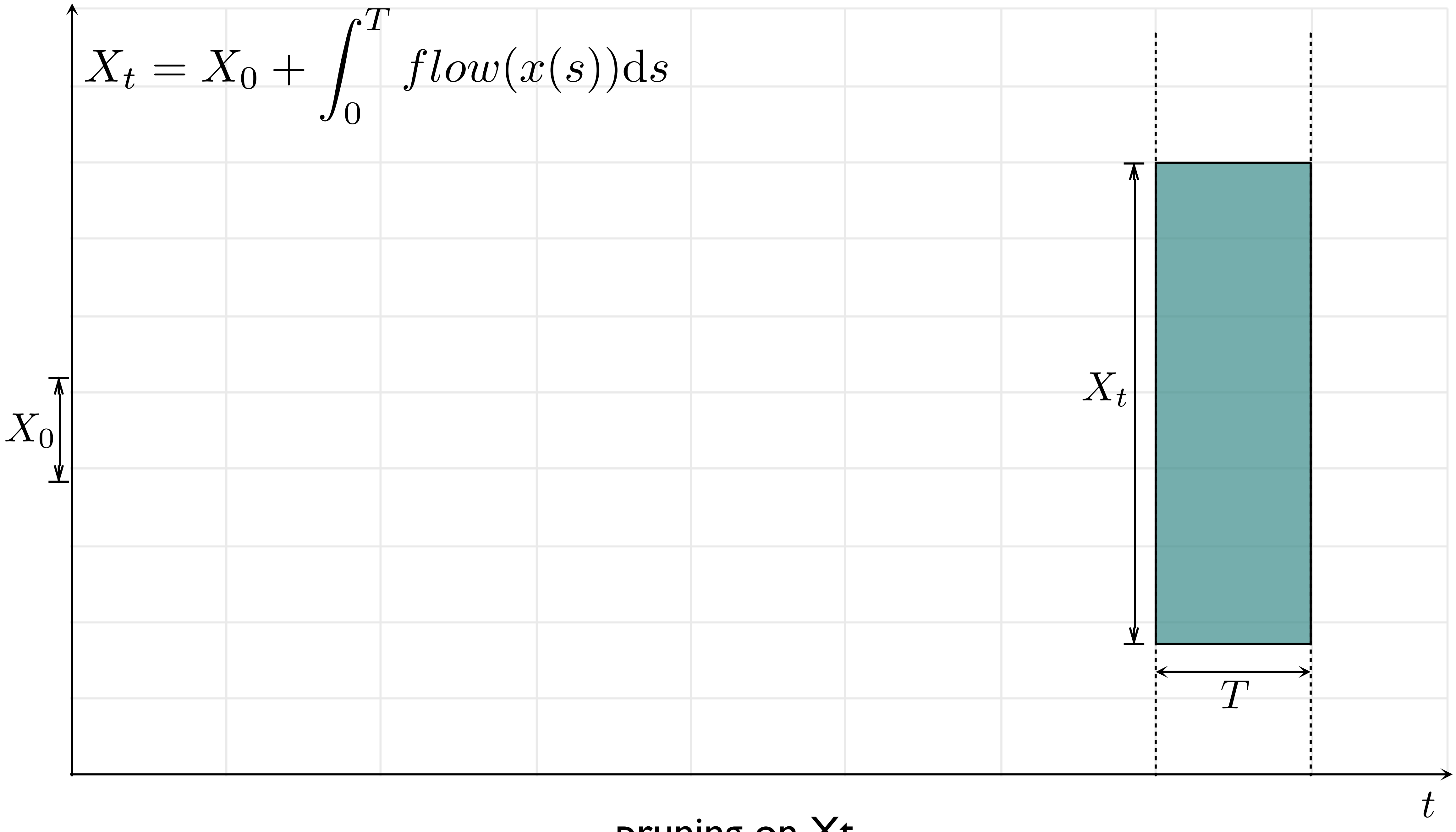
Pruning

Branching

Pruning using ODEs

$$X_t = X_0 + \int_0^T flow(x(s))\mathrm{d}s$$

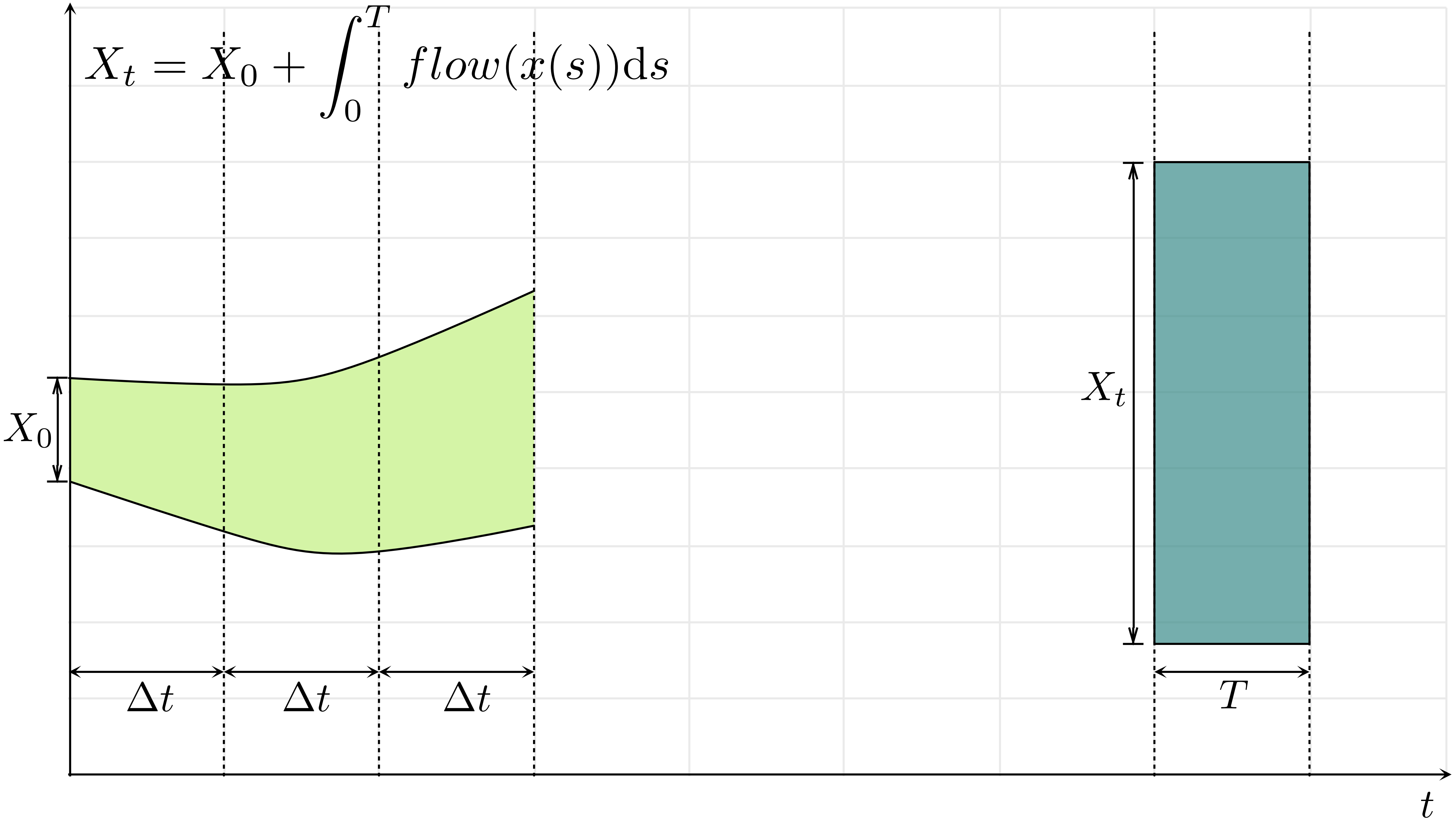
Pruning using ODEs
(Forward)



pruning on Xt

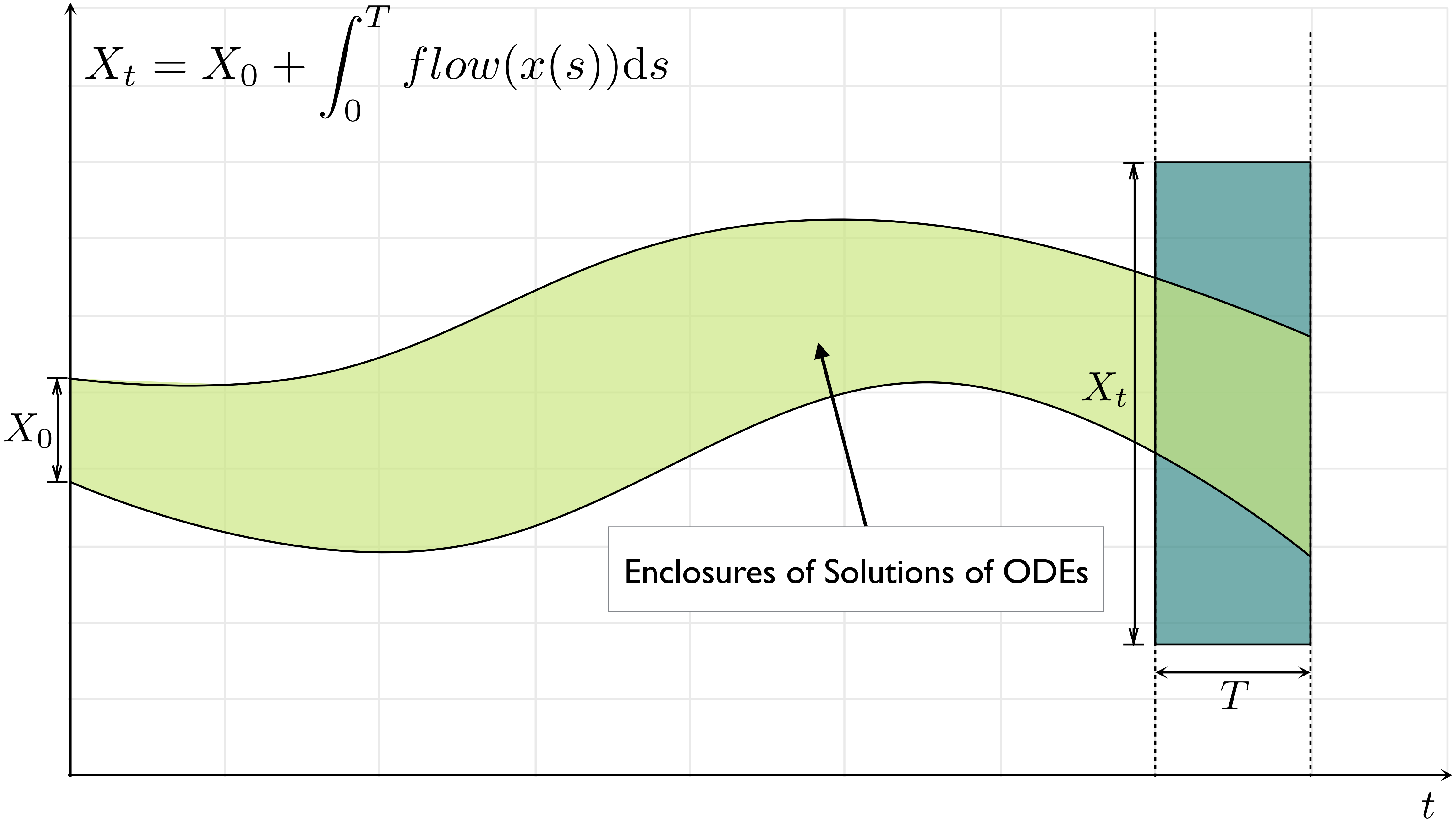
How can we prune X_t?

Pruning using ODEs (Forward)



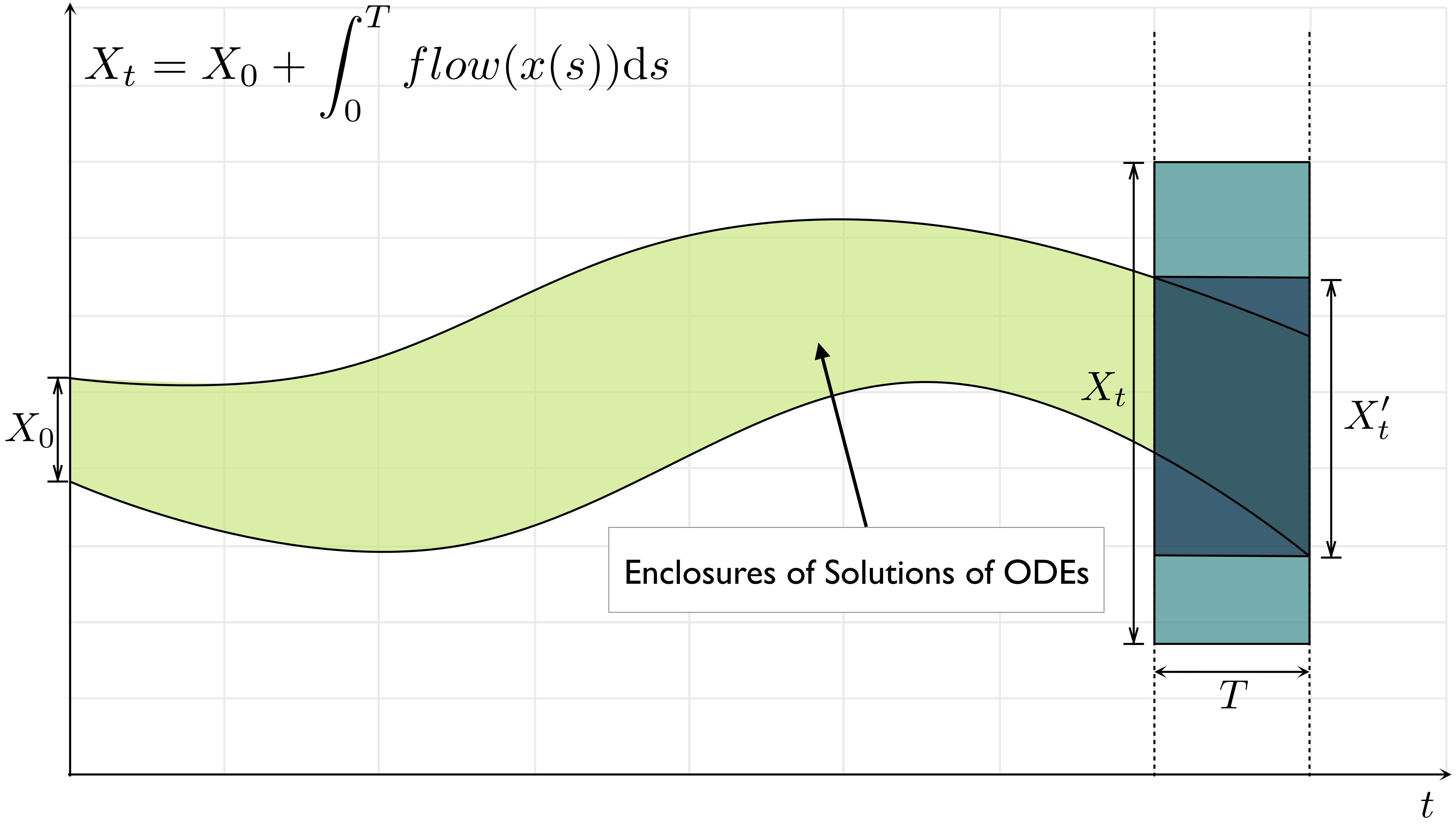
(numerically) Compute the enclosures of the solutions of ODE

Pruning using ODEs
(Forward)



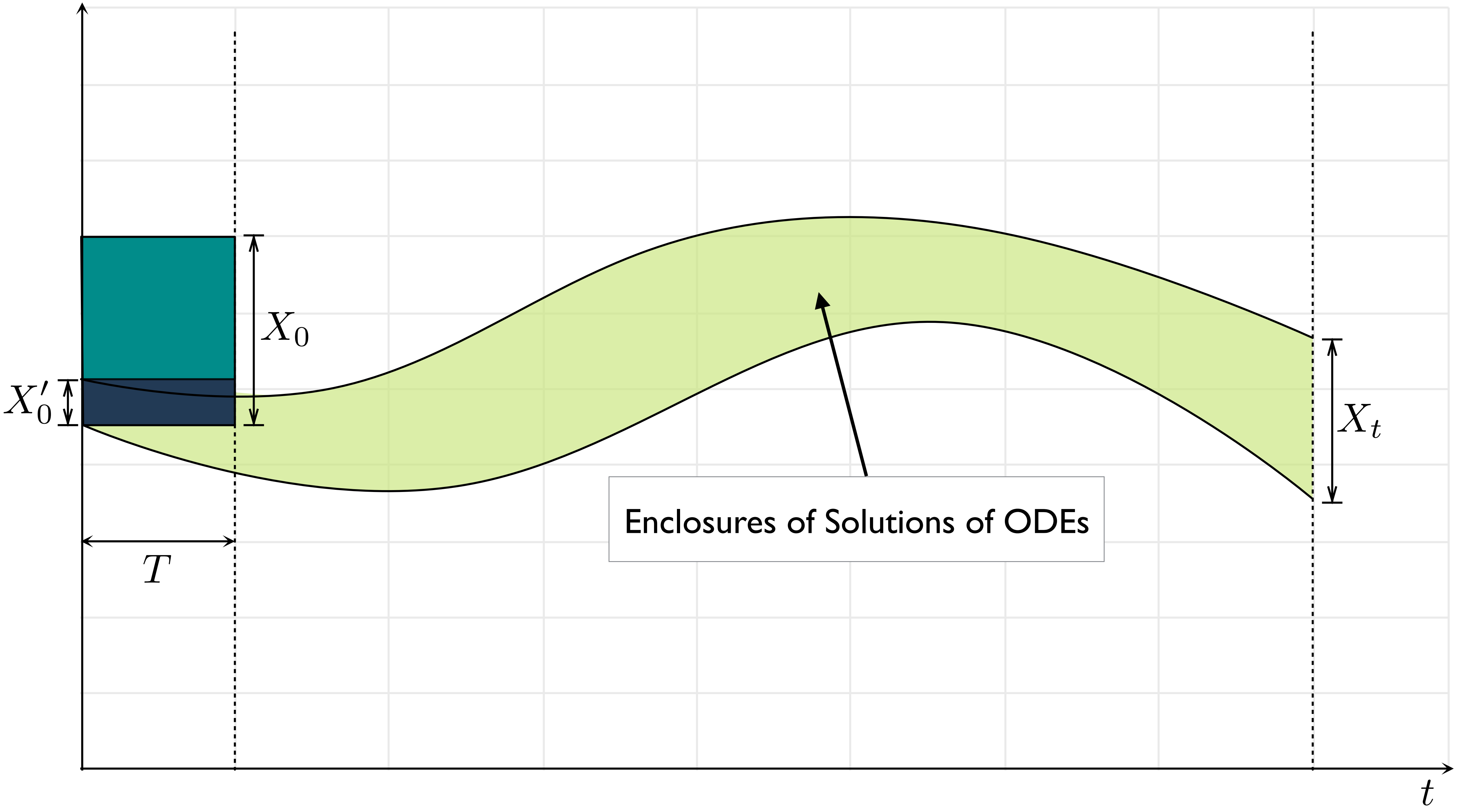
(numerically) Compute the enclosures of the solutions of ODE

Pruning using ODEs (Forward)



Take the intersection between the Enclosure and Xt

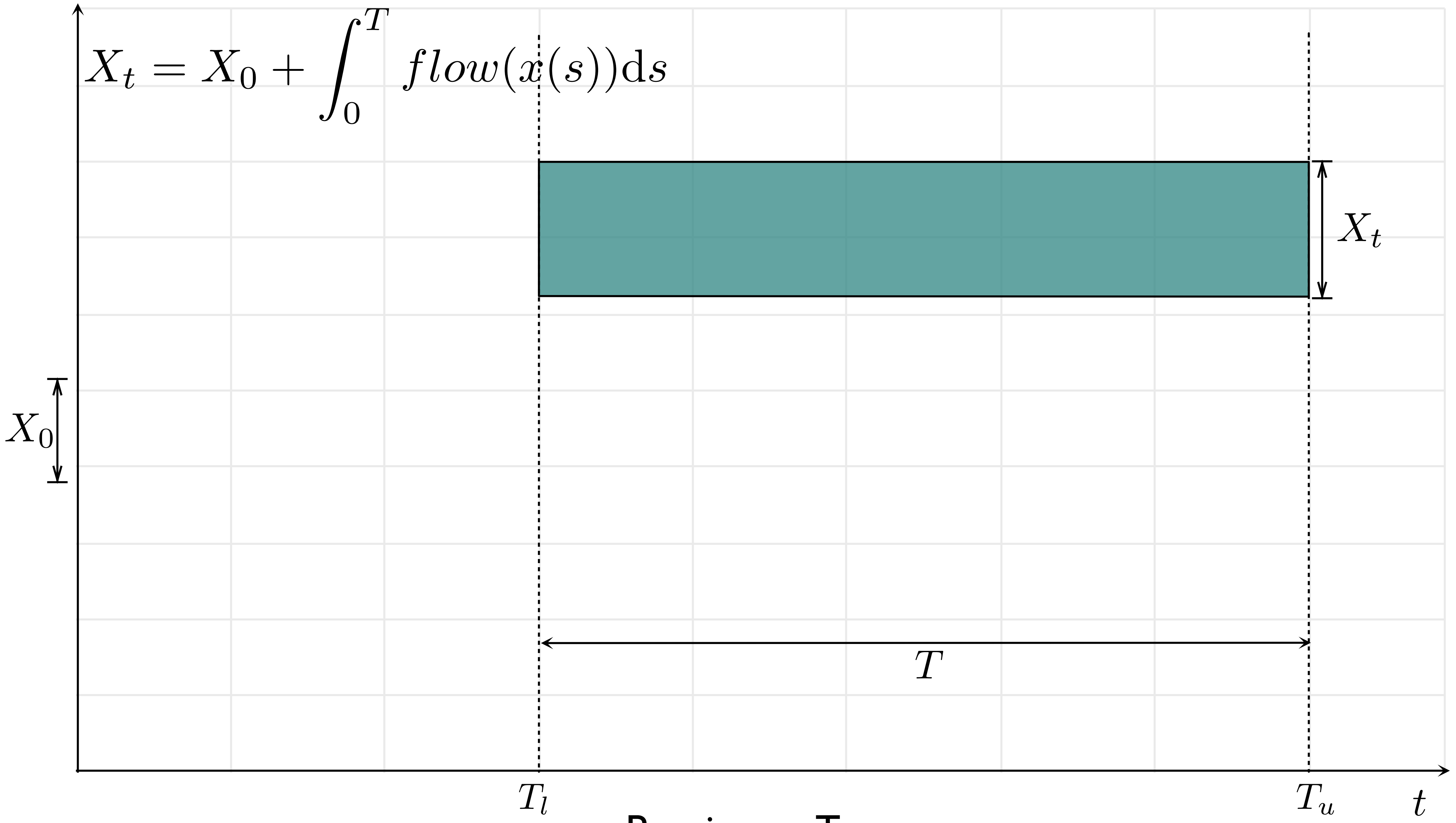
Pruning using ODEs
(Backward)



Enclosures of Solutions of ODEs

Pruning on X0

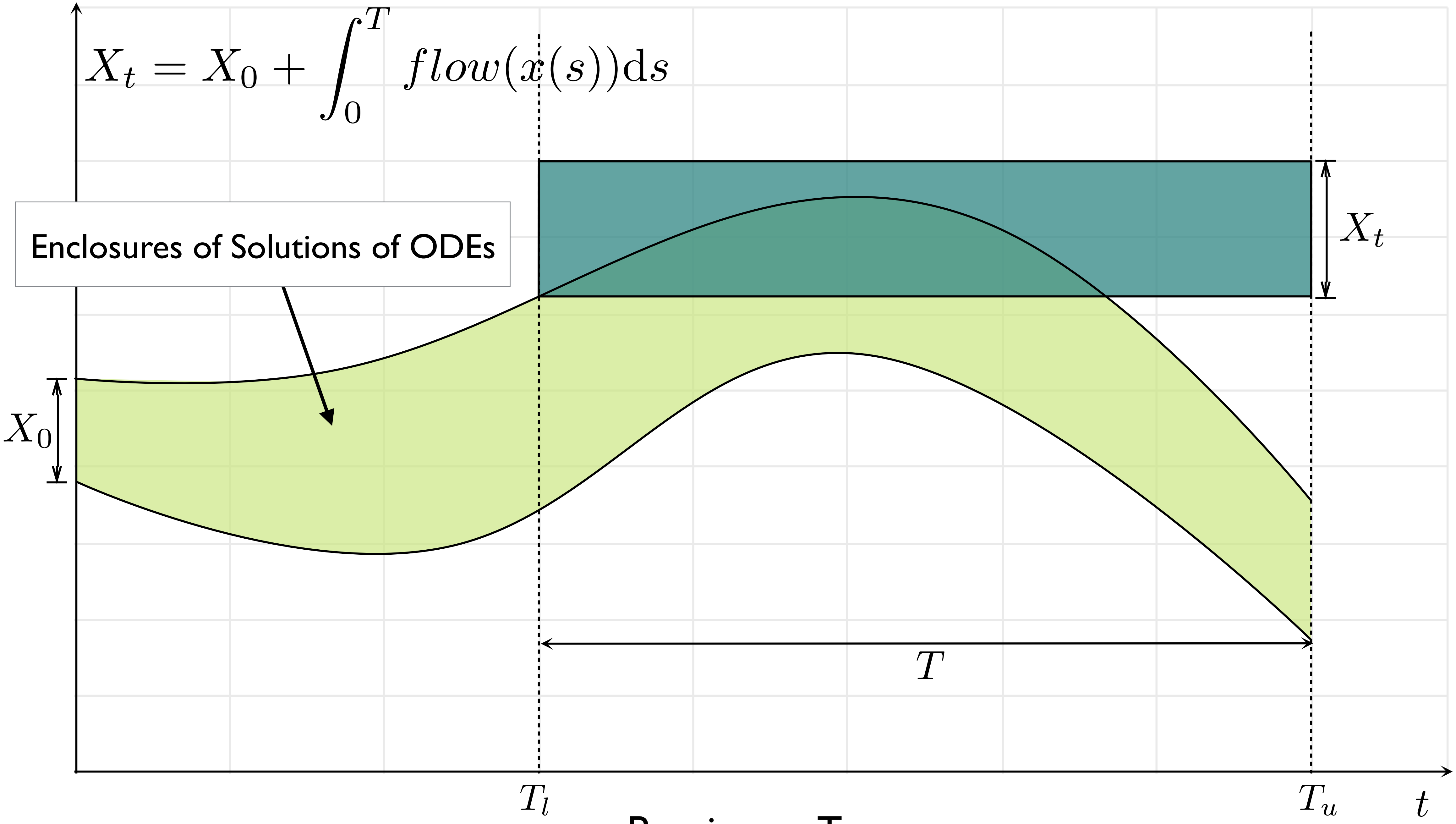
Pruning using ODEs (on Time)



$$X_t = X_0 + \int_0^T flow(x(s))\mathrm{d}s$$

Pruning on T

Pruning using ODEs
(on Time)

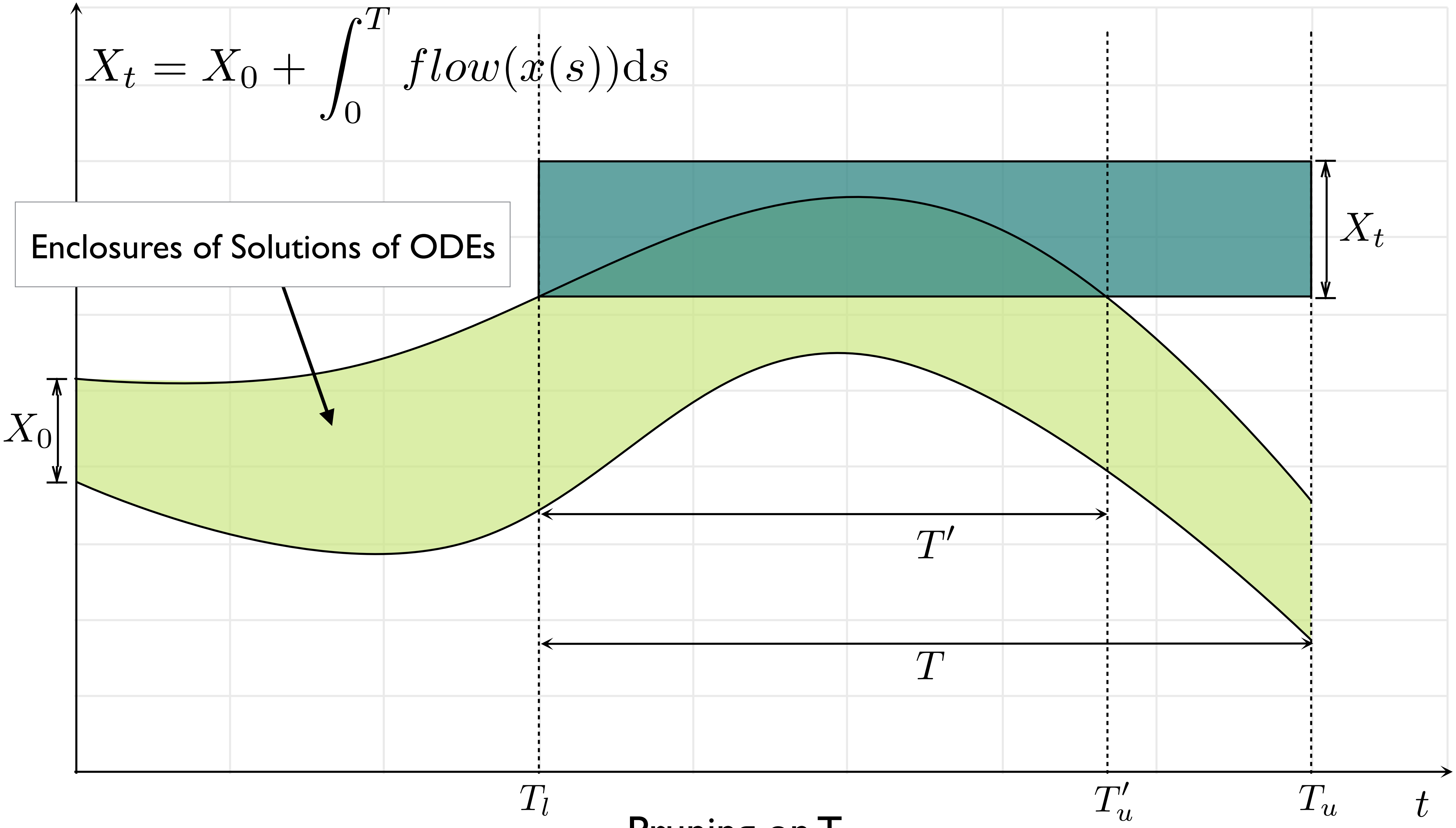


$$X_t = X_0 + \int_0^T flow(x(s))\mathrm{d}s$$

Enclosures of Solutions of ODEs

Pruning on T

Pruning using ODEs (on Time)

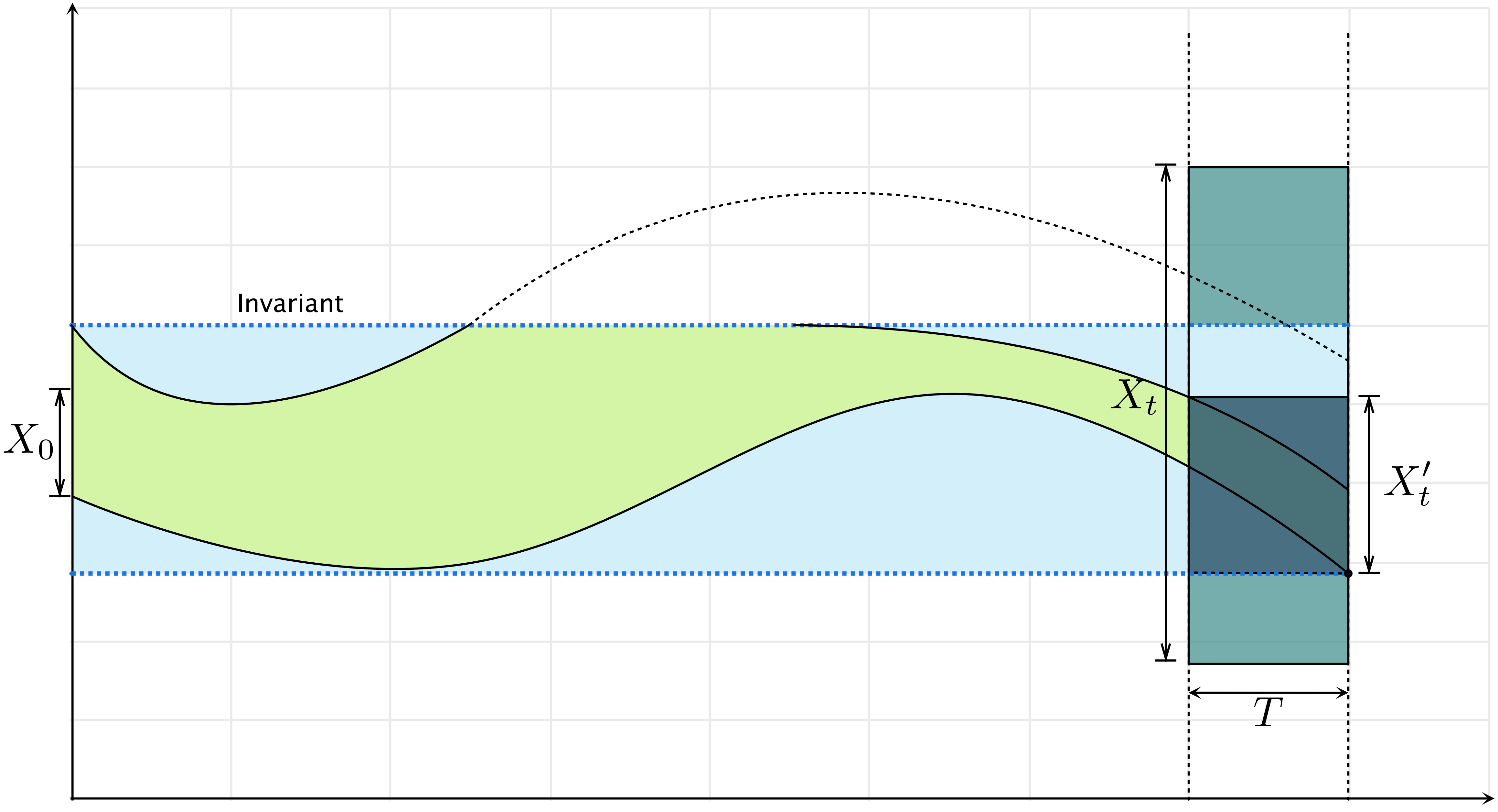


$$X_t = X_0 + \int_0^T flow(x(s))\mathrm{d}s$$

Enclosures of Solutions of ODEs

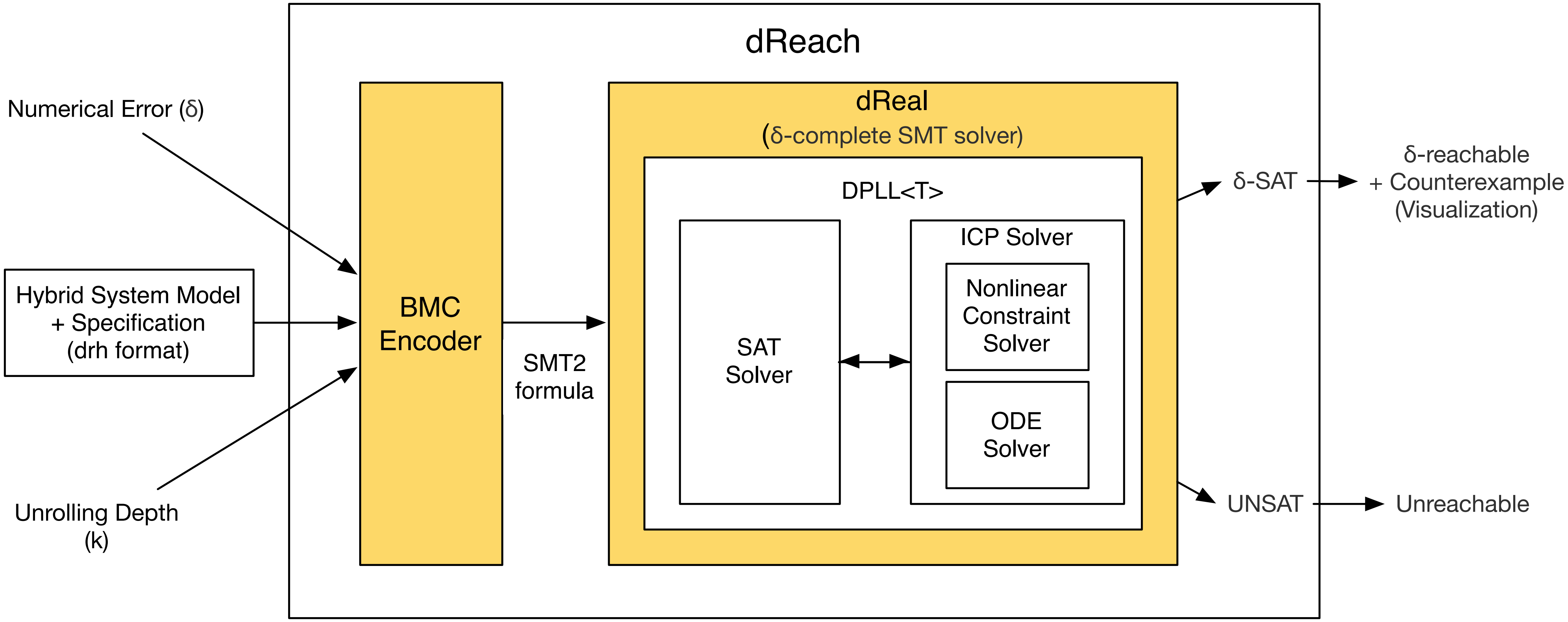
Pruning on T

Pruning using ODEs (using Invariant)



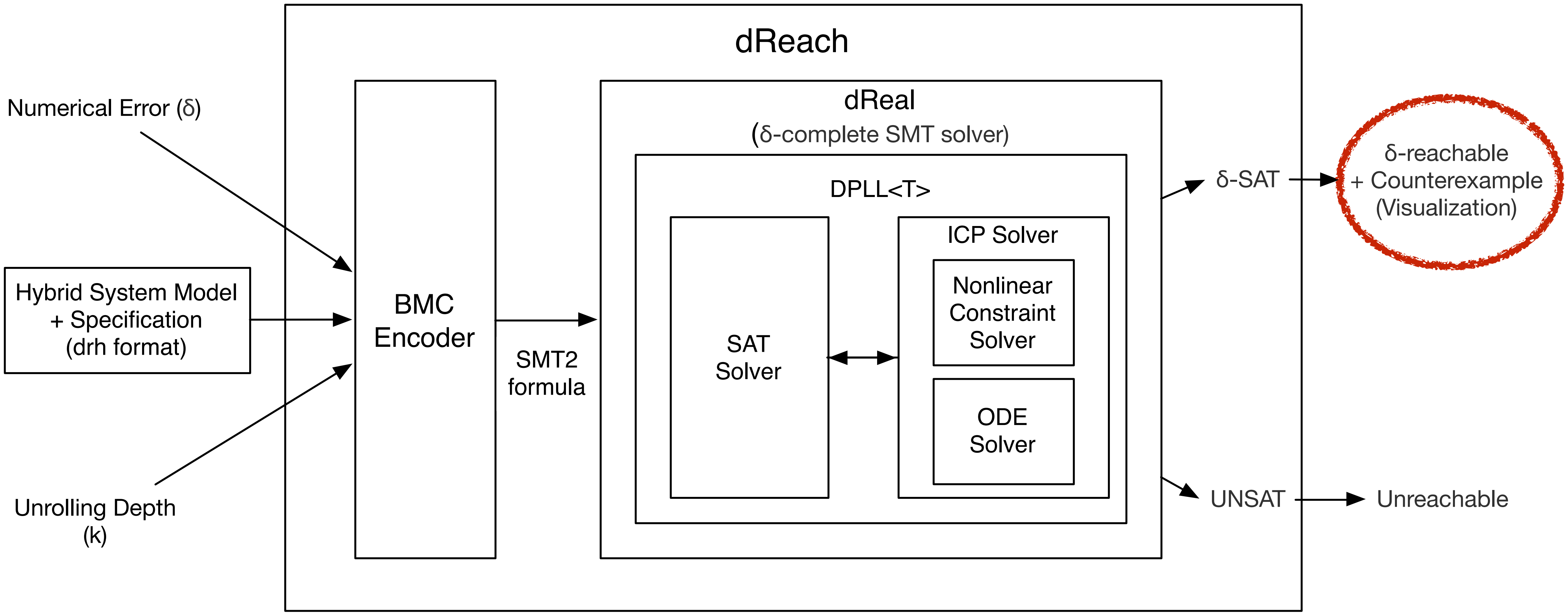
Pruning with Mode Invariant

dReach: δ-Reachability Analysis of Hybrid Systems

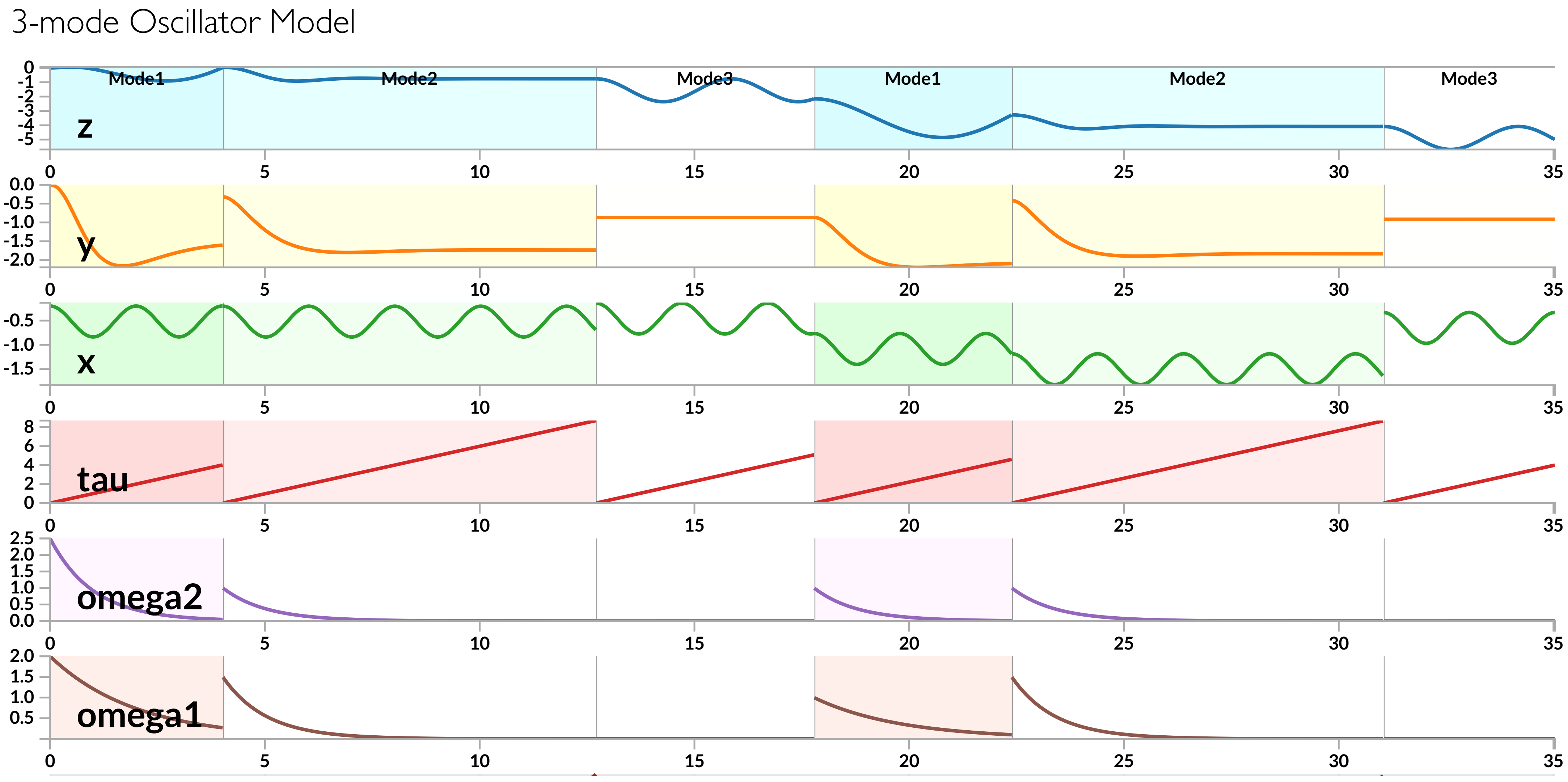


- Open Source, available at https://dreal.github.io
- Support polynomials, transcendental functions and nonlinear ODEs
- Formulas with 100+ ODEs have been solved.

Visualization of Counterexample



Visualization of Counterexample



Applications

* Cardiac Cells, Prostate Cancer (CMU, GIT, TU Vienna)

* Prostate Cancer (CMU, UPITT)

* Power-train Control, Validated Planning (Toyota Research Labs)

* Microfluidic Chip Designs (Univ. of Waterloo)

* Analog Circuits (City University London)

* Quadcopter Control, Autonomous Driving (CMU)

* FDA-accepted non-linear hybrid physiological model for diabetes, (UPENN)

Tools based on dReal/dReach

- APEX: A Tool for Autonomous Vehicle Plan Verification and Execution (Toyota)
- ProbReach: Probabilistic reachability analysis of hybrid systems (Univ. of Newcastle)
- BioPSy: Parameter set synthesis on biological models (Univ. of Newcastle)
- SReach: Bounded model checker for stochastic hybrid systems (CMU)
- Osmosis: Semantic importance sampling for statistical model checking (CMU SEI)
- Sigma: Probabilistic programming language (MIT)

Conclusion

- δ-reachability analysis checks robustness of hybrid systems which implies safety.
- Decidable (PSPACE-Complete)
- Use dReal, δ-SMT solver supporting nonlinear functions/ODEs
- Based on DPLL⟨ICP⟩ framework
- Open-source: available at http://dreal.github.io

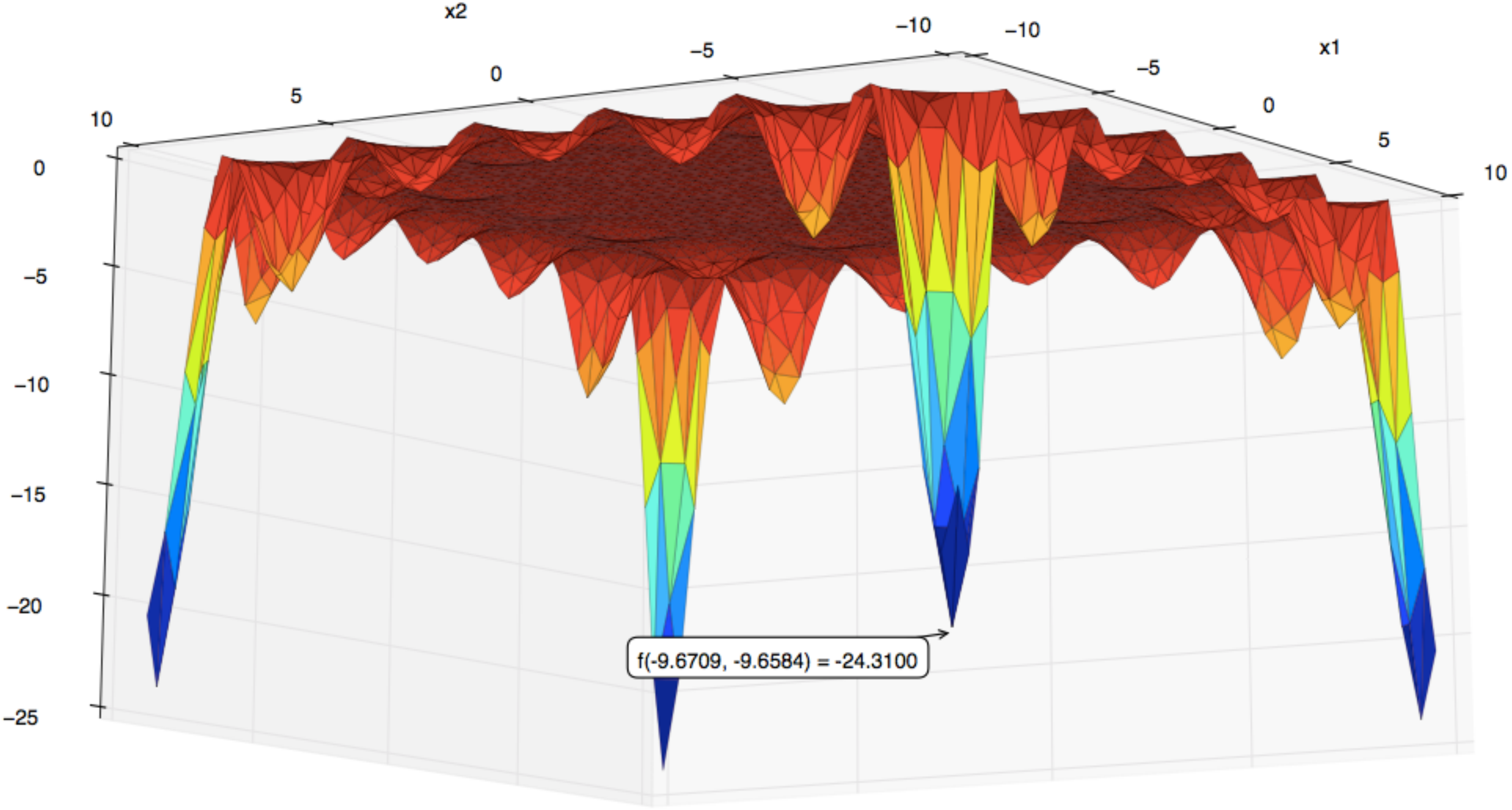
Future Work

- Scalability
  - <u>Learning & Non-chronological Backtracking in ICP</u>
  - Parallelization
- Expressiveness
  - <u>Support Exist-Forall formulas (for optimization problems)</u>
  - Support PDEs (Partial Differential Equations)
- Generating Certificate for UNSAT cases
  - Already have a prototype independent type-checker
  - Plan to generate Lean/Coq proofs

147. **Table 3 / Carrom Table Function** [58] (Continuous, Differentiable, Non-Separable, Non-Scalable, Multimodal)

$$f_{147}(\mathbf{x}) = -[(\cos(x_1)\cos(x_2) \\ \exp|1 - [(x_1^2 + x_2^2)^{0.5}]/\pi|)^2]/30$$

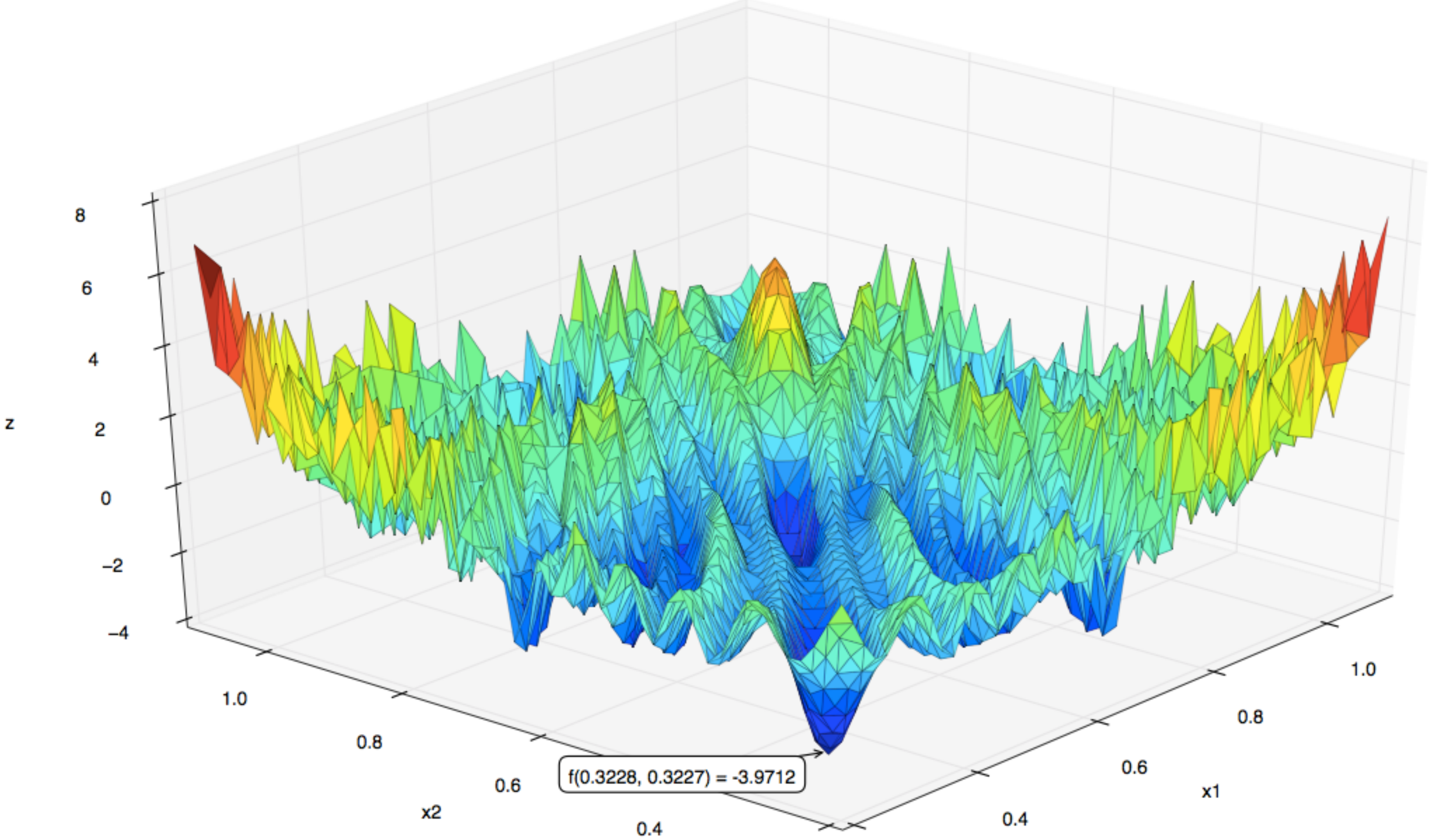
subject to $-10 \le x_i \le 10$.



Momin Jamil and Xin-She Yang, A literature survey of benchmark functions for global optimization problems, Int. Journal of Mathematical Modelling and Numerical Optimisation

167. **Whitley Function** [86] (Continuous, Differentiable, Non-Separable, Scalable, Multimodal)

$$f_{167}(\mathbf{x}) = \sum_{i=1}^{D}\sum_{j=1}^{D}\left[\frac{(100(x_i^2-x_j)^2+(1-x_j)^2)^2}{4000} - \cos\left(100(x_i^2-x_j)^2+(1-x_j)^2+1\right)\right]$$



Momin Jamil and Xin-She Yang, A literature survey of benchmark functions for global optimization problems, Int. Journal of Mathematical Modelling and Numerical Optimisation

Any Questions?

FAQs

Q1. How to pick ε from a given δ∈ $\mathbb{Q}^+$ in ICP Algorithm?

A1:

- For all f_i, find ε_i such that

$$\forall \vec{x}, \vec{y} \in B, ||x - y|| < \epsilon_i \implies |f_i(\vec{x}) - f_i(\vec{y})| < \delta$$

- Fix ε be the minimum of ε_is

$$\epsilon = \min(\epsilon_1, \ldots, \epsilon_n)$$

FAQs

Q2. Lipschitz continuity?

A2: A Lipschitz continuous function is limited in how fast it can change: there exists a definite real number K such that, for every pair of points on the graph of this function, the absolute value of the slope of the line connecting them is not greater than this real number; this bound is called a "Lipschitz constant" of the function (or "modulus of uniform continuity").

$$\frac{d_Y(f(x_1), f(x_2))}{d_X(x_1, x_2)} \leq K.$$