# Optimization-Based Model Checking and Trace Synthesis for Complex STL

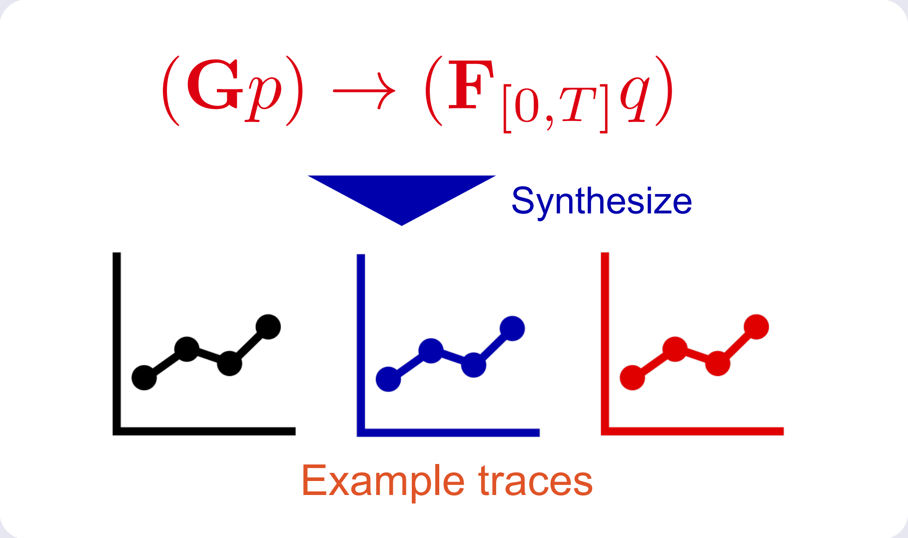
Sota Sato[1,3], Jie An[1,4], Zhenya Zhang[1,2], Ichiro Hasuo[1,3]

[1]National Institute of Informatics, Tokyo, Japan
[2]Kyusyu University, Fukuoka, Japan
[3]The Graduate University for Advances Studies (SOKENDAI), Hayama, Japan
[4]Institute of Software, Chinese Academy of Sciences, Beijing, China

# Summary

## 🌟 What we achieved

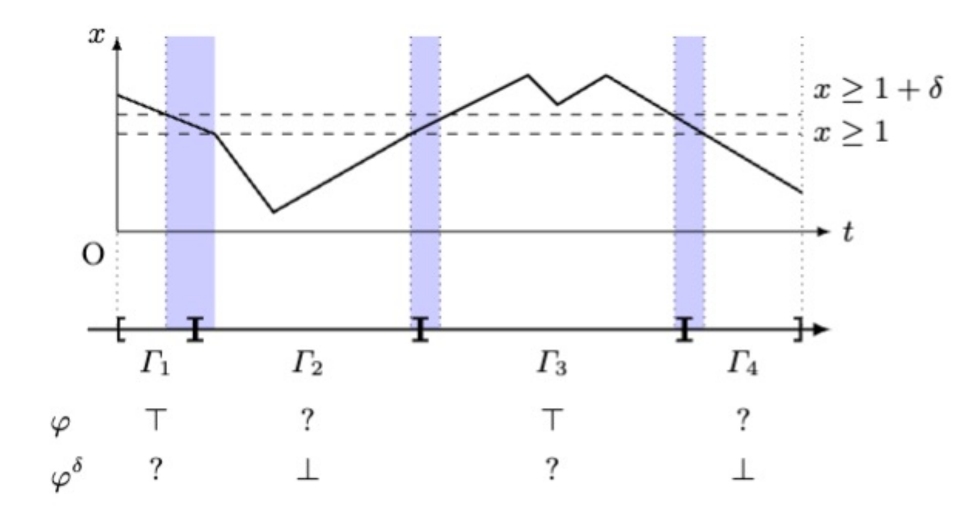
A novel algorithm to quickly **synthesize a trace** with an STL (signal temporal logic) [Maler & Nickovic, 2004] formula




- Can also solve the dual problem (bounded model checking)

## 🛠️ How we did it

The algorithm is fast due to our novel **variable-interval** MILP (mixed-integer linear programming) encoding




- Soundness and (weak) completenss of the result are guaranteed

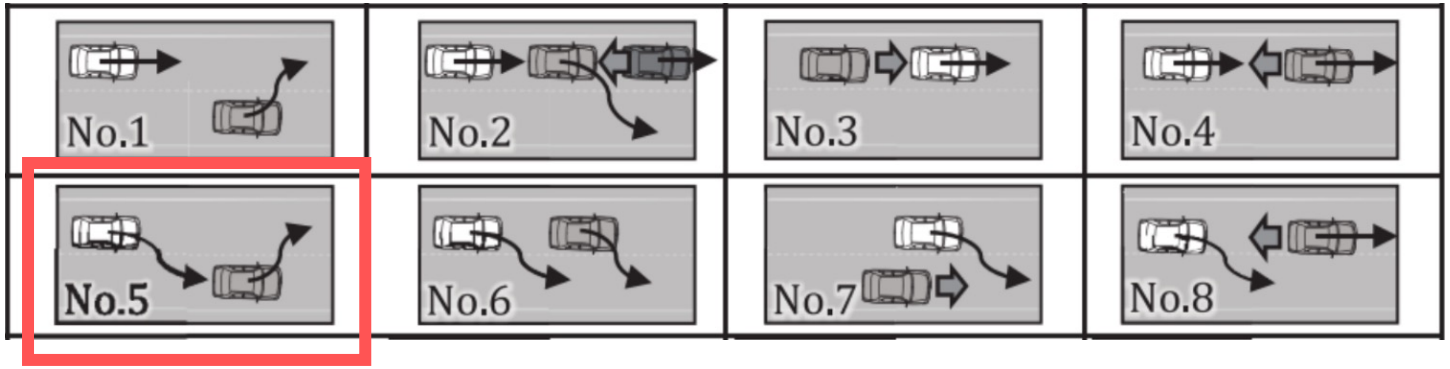
# Outline

👉 1. **Challenges in Trace Synthesis with Complex STL Formulas**
2. Previous Works
3. Our Algorithm: Variable-Interval MILP Encoding
4. Experiments

# STL formula in real-world problem may be large

## Natural-language description (from ISO 34502 critical scenarios)

The subject vehicle cuts in and the preceding other vehicle changes a lane. The traffic situation is initially safe in terms of the RSS safety but eventually gets in danger. (…)




## Formalization in STL [Reimann+, SAC'24]

$$
\begin{aligned}
\texttt{iso}_5 :\equiv\ & \Box_{[0,0.6]}\Big(\neg\big(\boxed{x_b - x_a \leq \mathit{rssDistance}_a + l_b} \land \boxed{x_a - x_b \leq \mathit{rssDistance}_b + l_a} \land \boxed{y_b - y_a \leq \mathit{latRssDistance}_a + w_b} \land \boxed{y_a - y_b \leq \mathit{latRssDistance}_b + w_a}\big)\Big) \\
& \land \boxed{y_a \leq 7.0 + w_a} \land \boxed{y_a \geq 3.5} \land \Diamond\Big(\neg\big(\boxed{y_a \leq 7.0 + w_a} \land \boxed{y_a \geq 3.5}\big)\Big) \land \boxed{y_a \leq 7.0 + w_a} \land \boxed{y_a \geq 3.5} \land \neg\big(\boxed{y_b \leq 7.0 + w_a} \land \boxed{y_b \geq 3.5}\big) \land \\
& \Diamond\Big(\Box_{[0,0.6]}\big(\boxed{x_b - x_a \leq \mathit{rssDistance}_a + l_b} \land \boxed{x_a - x_b \leq \mathit{rssDistance}_b + l_a} \land \boxed{y_b - y_a \leq \mathit{latRssDistance}_a + w_b} \land \boxed{y_a - y_b \leq \mathit{latRssDistance}_b + w_a}\big) \land \\
& \Diamond_{[0,0.6]}\big(\boxed{x_a + l_b \leq x_b} \land \boxed{y_b \leq 7.0 + w_b} \land \boxed{y_b \geq 3.5}\big)\Big)
\end{aligned}
$$

Syntax tree is **quite large**.
Max depth: 5
Leaves: 25
Nodes : 35

# Trace synthesis with STL

Find $\sigma \in \mathcal{L}(\mathcal{M})$ such that $\sigma \vDash \varphi$

Possible traces of the system model $\mathcal{M}$

A trace $\sigma$ satisfies an STL formula $\varphi$

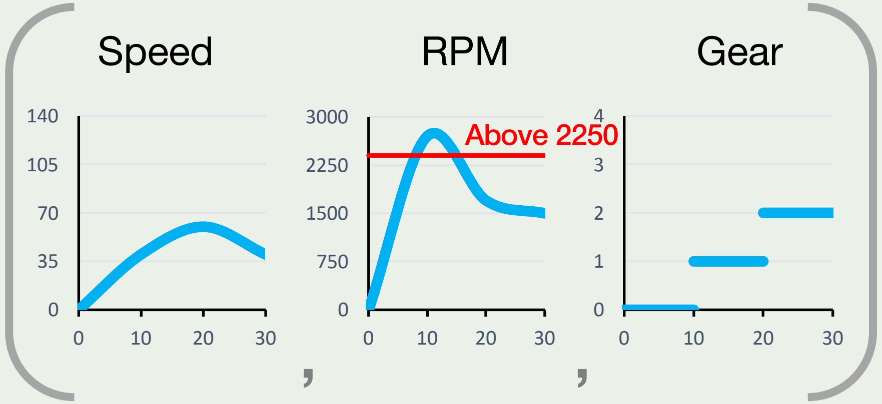
**Example**

$$\sigma \vDash \Diamond_{[0,30]}(\text{RPM} > 2250)$$

*"At some $t \in [0,30]$, RPM is above 2250"*

where $\sigma \in \mathcal{L}$ 

$\sigma$ : satisying trace

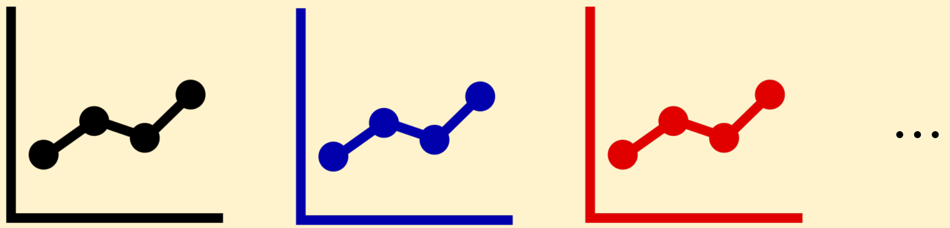


# Trace synthesis provides “debugging” of specs

- There can be a difference between **what one wrote** and **what one intended**
- Automatic generation of an **example trace** would help to recognize mistakes

WRONG $(\mathbf{G}p) \to (\mathbf{F}_{[0,T]}q)$

Synthesize examples

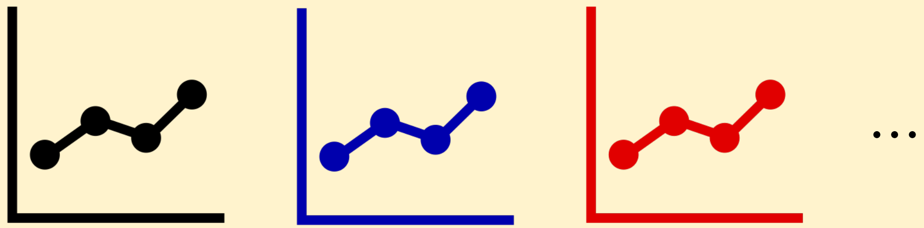


(1) As intended (2) As intended (3) Not as intended …

Human correction

$\mathbf{G}(p \to \mathbf{F}_{[0,T]}q)$

Synthesize examples



(1’) As intended (2’) As intended (3’) As intended …

Odd that the third trace is an example of my specification…

Misplaced parenthesis!

My spec looks correct

# Existing tools suffer from trace synthesis with complex STL formulas

- Small latency is **critical for interactive inspection** of STL formulas
- Our benchmarks showed that existing methods were rarely capable of it

It takes **9-500 seconds** to synthesize a trace

| | Breach | ForeSee | bluSTL | STLmc |
|---|---|---|---|---|
| RNC1 | 59.4 | 546.8 | (¶) | t/o |
| RNC2 | 9.3 | 104.3 | 14.3 | t/o |
| RNC3 | 81.3 | 197.4 | (¶) | t/o |
| NAV1 | (∗) | (∗) | (‡) | 16.5 |
| NAV2 | | | | 10.0 |
| ISO1 | 8.9 | t/o | | |
| ISO3 | t/o | t/o | | |
| ISO4 | t/o | t/o | | |
| ISO5 | 31.2 | 435.8 | (†) | (†) |
| ISO6 | t/o | 58.9 | | |
| ISO7 | 33.6 | 187.2 | | |
| ISO8 | 38.8 | t/o | | |

Our experimental result (we revisit it later)

Breach [Donzé, CAV'10]

ForeSee [Zhang+, FM'21]

BluSTL [Donzé & Raman, ARCH15]

STLmc [Yu+, CAV'22]

# Outline

1. Challenges in Trace Synthesis with Complex STL Formulas
2. 👉 **Previous Works**
3. Our Algorithm: Variable-Interval MILP Encoding
4. Experiments

# Discretization of the STL semantics: constant vs. variable

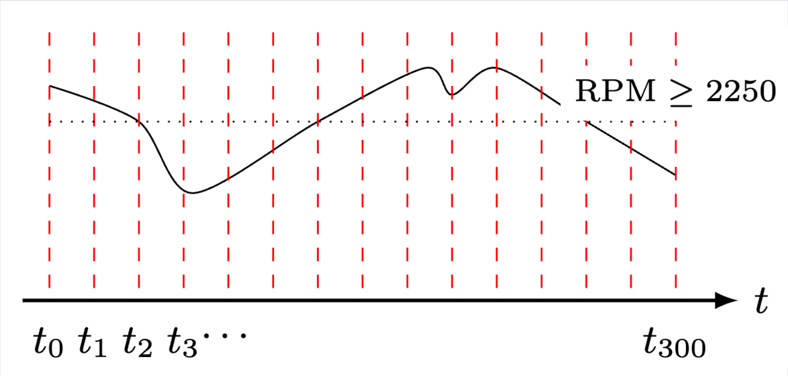
- Due to its **continuous** nature (unlike LTL), discretization is required to compute STL semantics

**Mathematical definition by first-order logic**

$$\sigma \models \Diamond_{[0,30]}(\Box_{[0,5]}\mathrm{RPM} > 2250) \iff \exists \tau_1 \in [0,30]. \forall \tau_2 \in \tau_1 + [0,5]. \sigma_{\mathrm{RPM}}(\tau_2) > 2250$$

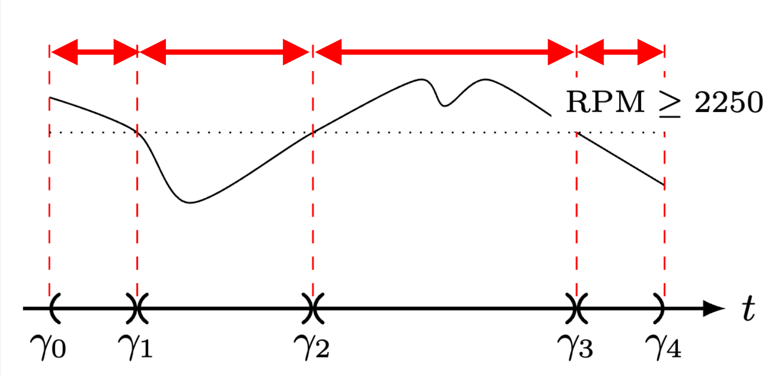
**Constant-step**

[Donzé & Raman, ARCH15]

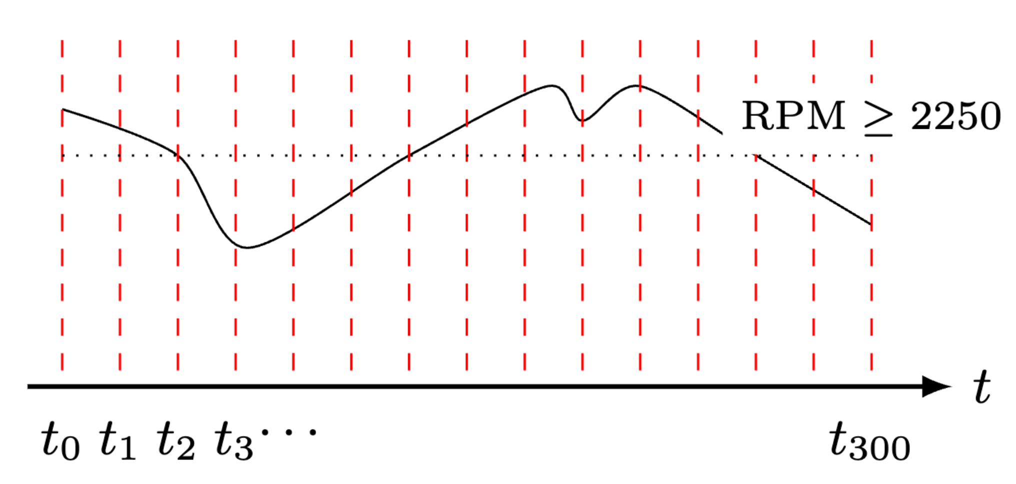



**Variable-intervals (<u>we use this</u>)**

[Yu+, CAV'22]



## Constant-step




Input: $\mathcal{M}, \varphi, t_0, \ldots, t_N$

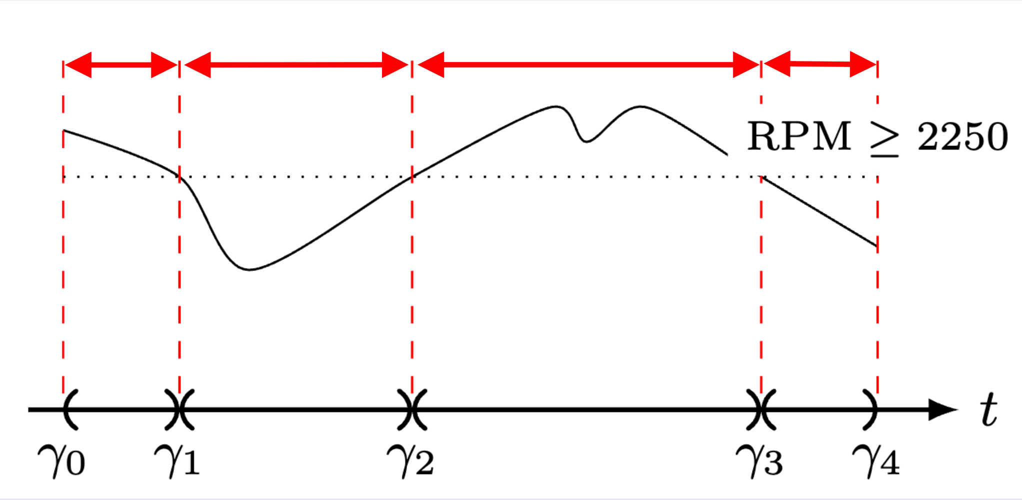
Output: $\vec{x}_1, \ldots, \vec{x}_N$ **(state sequence)**

**Inductive calculation**

| | $t_0$ | $t_2$ | $t_2$ | … | $t_N$ |
|---|---|---|---|---|---|
| state | $\vec{x}_0$ | $\vec{x}_1$ | $\vec{x}_2$ | … | $\vec{x}_N$ |
| $p := \mathrm{RPM} > 2250$ | $\top$ | $\top$ | $\top$ | … | $\bot$ |
| $\square_{[0,5]} p$ | $\bot$ | $\bot$ | $\bot$ | … | $\bot$ |
| $\lozenge_{[0,30]}(\square_{[0,5]} p)$ | $\top$ | $\top$ | $\top$ | … | $\bot$ |

- Straightforward discretization
- $N$ should be **large** for sufficient accuracy => Slowing down SMT/MILP solver
  - E.g., $N = 300$ in our *iso5* benchmark (10 samples every second)

## Variable-interval (we use this)




Input: $\mathcal{M}, \varphi$

Output: $(\gamma_0, \vec{x}_0), \ldots, (\gamma_N, \vec{x}_N)$ (timed-state sequence)

| | $\{\gamma_0\}$ | $(\gamma_0, \gamma_1)$ | $\{\gamma_1\}$ | … | $\{\gamma_N\}$ |
|---|---|---|---|---|---|
| state | $\vec{x}_0$ | $\lambda\vec{x}_0 + \bar{\lambda}\vec{x}_1$ | $\vec{x}_1$ | … | $\vec{x}_N$ |
| p := RPM > 2250 | ⊤ | ⊤ | ⊤ | … | ⊥ |
| $\Box_{[0,5]}\mathrm{p}$ | ⊥ | ⊥ | ⊥ | … | ⊥ |
| $\Diamond_{[0,30]}(\Box_{[0,5]}\mathrm{p})$ | ⊤ | ⊤ | ⊤ | … | ⊥ |

**Inductive calculation**

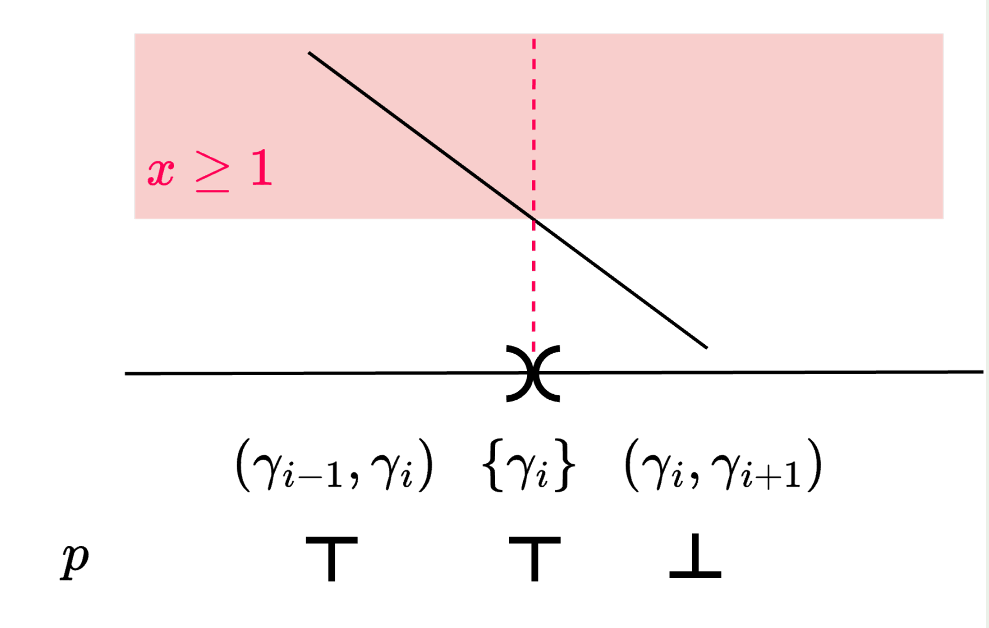
Boolean values must be constant in each **interval** and **subformula**

- Works with **small** $N$ in practice
  - E.g., $N = 4$ in our *iso5* benchmark

# How to find a stable partition for the variable-interval?

[Bae & Lee, POPL'19]

## For atomic proposition




- Put sampling points when crossing the boundary of predicates
- (Assumption: **continuous signal** and a **linear predicate)**

# How to find a stable partition for the variable-interval?

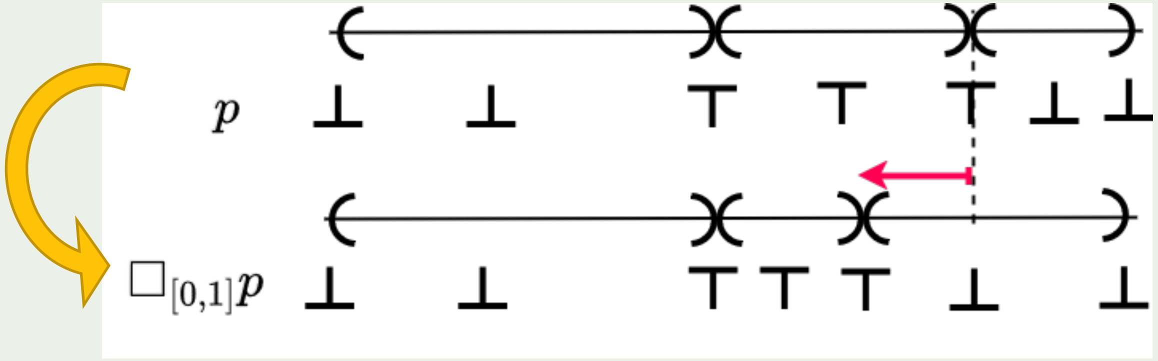
[Bae & Lee, POPL'19]

## For Boolean connectives $\psi_1 \wedge \psi_2$, $\psi_1 \vee \psi_2$

Assuming we have stable partitions for both $\psi_1, \psi_2$, their refinement is a stable partition for $\varphi$

## For "Always for" operator $\Box_{[a,b]}\psi$

Can be formulated based on the **truth values of subformula** and by **interval math**



# Overview of tools

| | Solver | Variable-interval | Why slow with complex STL? |
|---|---|---|---|
| Breach [Donzé, CAV'10] | Stoch. opt. | No | Fall into local solution |
| ForeSee [Zhang+, FM'21] | Stoch. opt. + MCTS | No | Fall into local solution |
| BluSTL [Donzé & Raman, ARCH15] | **MILP** | No | Large sample size |
| STLmc [Yu+, CAV'22] | SMT | **Yes** | Unscalability of SMT |
| STLts (our proposed method) | **MILP** | **Yes (with modification)** | **-** |

We use MILP + Variable-interval encoding

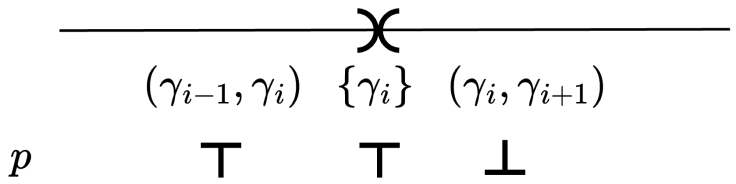
# Outline

1. Challenges in Trace Synthesis with Complex STL Formulas
2. Previous Works
3. **Our Algorithm: Variable-Interval MILP Encoding**
4. Experiments

# MILP (numerical) cannot faithfully express the variable-interval encoding, unlike SMT (symbolic)

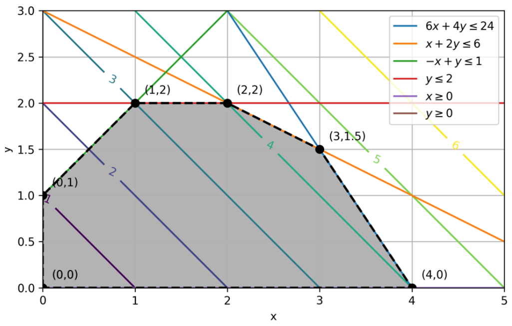
## With variable-interval encoding

- It manages the truth values of open intervals and singular intervals




## With MILP

- Only nonstrict inequality $\leq$ is allowed




https://github.com/BYU-PRISM/GEKKO

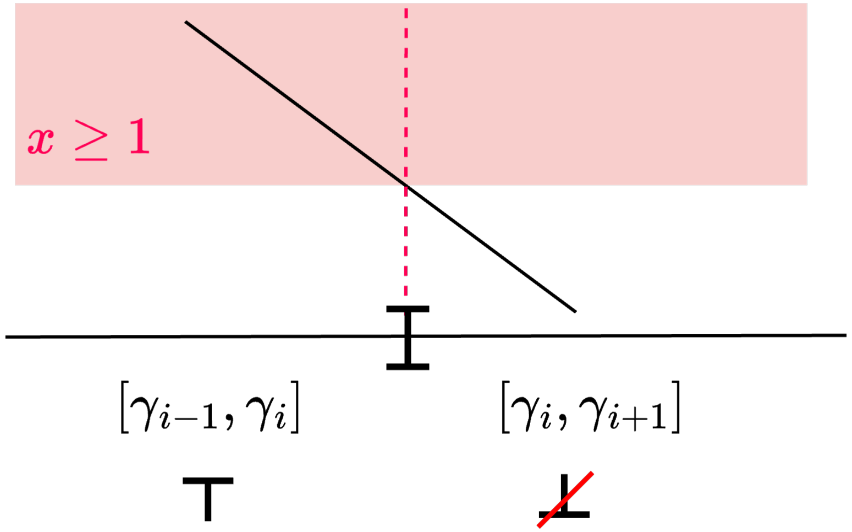
❌ Cannot be combined

# Can we interpret truth values on closed intervals?

## Naive idea

- Given: signal $\sigma$, STL formula $\varphi$
- Find: $\gamma_0 < \cdots < \gamma_N$ such that
  $\underline{\sigma^t \models \varphi \text{ or } \sigma^t \not\models \varphi}$ is constant on each closed interval
  $[\gamma_0, \gamma_1], [\gamma_1, \gamma_2], \ldots, [\gamma_{N-1}, \gamma_N]$

=> **This idea does not work**

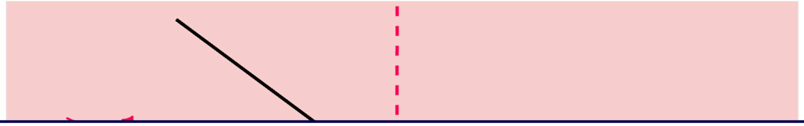



Not constant at its left-edge

# Can we interpret truth values on closed intervals?

## Naiive idea

- Given: signal $\sigma$, STL formula $\varphi$
-



| | Solver | Variable-interval | Why slow with complex STL? |
|---|---|---|---|
| BluSTL [Donzé & Raman, ARCH15] | **MILP** | No | Large sample size |
| STLmc [Yu+, CAV’22] | SMT | **Yes** | Unscalability of SMT |
| STLts (our proposed method) | **MILP** | **Yes (with modification)** | - |

# Introducing $\delta$-*tightening* of formula

- Purely syntactic modification
- Commutative with subformulas ($\psi^{\delta}$ implies $\psi$ for each subformula)

($\delta$: small positive real constant)

$$\varphi :\equiv \Diamond_{[0,30]}(\Box_{[0,5]}\mathrm{RPM} > 2250) \quad \xrightarrow{\text{Tightening}} \quad \varphi^{\delta} :\equiv \Diamond_{[0,30]}(\Box_{[0,5]}\mathrm{RPM} > 2250 + \delta)$$

$$\Box_{[0,5]}\mathrm{RPM} > 2250 \qquad \Box_{[0,5]}\mathrm{RPM} > 2250 + \delta$$

$$\mathrm{RPM} > 2250 \qquad \mathrm{RPM} > 2250 + \delta$$

# Can we interpret truth values on closed intervals?

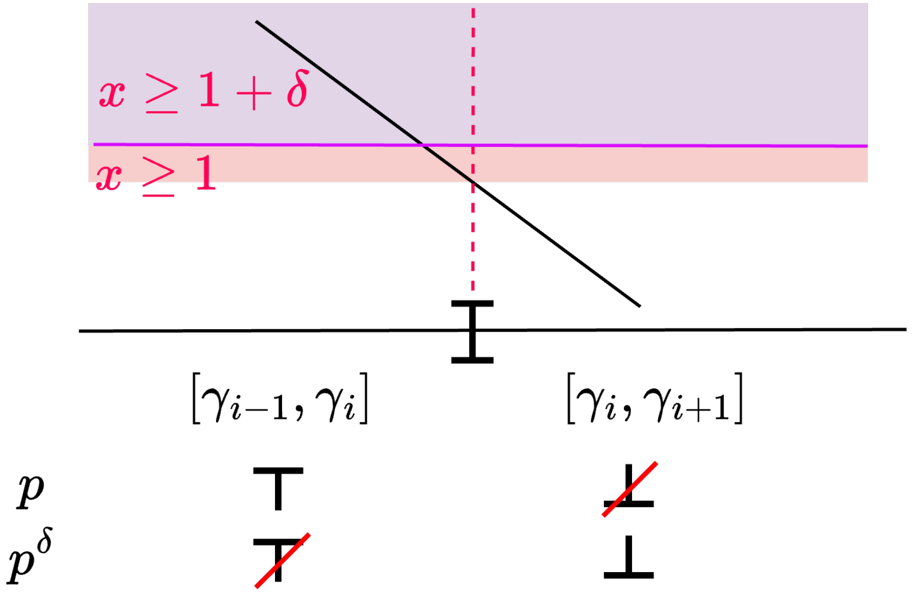
## Our revised idea

- Given: signal $\sigma$, STL formula $\varphi$
- Find: $\gamma_0 < \cdots < \gamma_N$ such that

  $\sigma^t \vDash \varphi$ or $\sigma^t \nvDash \varphi^\delta$ is constant

  on each closed interval

  $[\gamma_0, \gamma_1], [\gamma_1, \gamma_2], \dots, [\gamma_{N-1}, \gamma_N]$




Constantly $\sigma^t \vDash \varphi$ or $\sigma^t \nvDash \varphi^\delta$

# $\delta$-tightening is consistent with STL semantics

**Proposition**

Let $\varphi$ be an STL formula in NNF (*negation-normal form*), $\sigma$ be a continuous signal, $\delta > 0$. There exists a sequence $\gamma_0 < \cdots < \gamma_N$ for some $N$ such that for each $i \in [1, N]$ and $\psi \in sub(\varphi)$ either of $\sigma^t \vDash \psi$ or $\sigma^t \nvDash \psi^\delta$ is constant on $[\gamma_{i-1}, \gamma_i]$.

| | $[\gamma_0, \gamma_1]$ | $[\gamma_1, \gamma_2]$ | … | $[\gamma_{N-1}, \gamma_N]$ |
|---|---|---|---|---|
| state | $\lambda\vec{x}_0 + \bar{\lambda}\vec{x}_1$ | $\lambda\vec{x}_1 + \bar{\lambda}\vec{x}_2$ | … | $\lambda\vec{x}_{N-1} + \bar{\lambda}\vec{x}_N$ |
| p | ⊤ | ⊤ | … | $\bot\ (\delta)$ |
| $\square_{[0,5]}\mathrm{p}$ | $\bot\ (\delta)$ | $\bot\ (\delta)$ | … | $\bot\ (\delta)$ |
| $\Diamond_{[0,30]}(\square_{[0,5]}\mathrm{p})$ | ⊤ | ⊤ | … | $\bot\ (\delta)$ |

Inductive calculation

($\bot\ (\delta)$ is shorthand for $\sigma^t \nvDash \psi^\delta$ for any t $\in [\gamma_{i-1}, \gamma_i]$)

# Our variable-interval MILP algorithm

**Trace synthesis problem**

Find $\sigma \in \mathcal{L}(\mathcal{M})$ such that $\sigma \models \varphi$

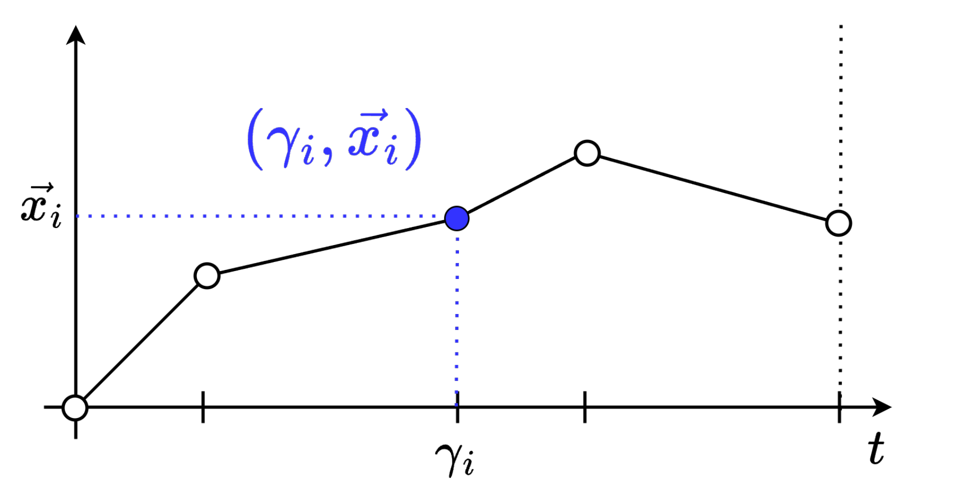
**MILP formulation (used in our algorithm)**

Minimize cost($v$)

subject to $\begin{cases} v \in \text{Enc}(\varphi, \mathcal{M}, N, \delta) \\ v \text{ satisfies the (mixed) integrity condition} \end{cases}$

where $v = [(\gamma_0, \vec{x}_0), \ldots, (\gamma_N, \vec{x}_N), \ldots \text{(other auxiliary variables)}]$

A piecewise-linear trace can be represented by variables $\gamma_0, \ldots, \gamma_N$ and $\vec{x}_0, \ldots, \vec{x}_N$



# Example MILP constraints

## (details are not important)

Note
- We use the Boolean symbols ∧,∨,¬ and the indicator relation ⇒ since they are commonly supported in most MILP solvers
- It is clear that the interval arithmetic in D2.1-D3.3 can be rewritten into MILP constraints

Let $\varphi :\equiv \square_{[0,4]} p$, where $p :\equiv \alpha > 8$:

$\gamma_\mathrm{i}, x_{i,\alpha}$ must represent a trace of given model

[A1] $0 = \gamma_0 < \cdots < \gamma_N = 10$

[A2] $0 \leq x_{i,\alpha} \leq 8 \qquad \forall i \in [0, N]$

[A3] $-1 \leq (x_{i,\alpha} - x_{i-1,\alpha}) \cdot (\gamma_i - \gamma_{i-1}) \leq 8 \qquad \forall i \in [1, N]$

[A4] $x_{0,\alpha} = 0$

Manage truth values of $p$ in each interval $[\gamma_{\mathrm{i}-1}, \gamma_\mathrm{i}]$

[B1] $\zeta_i^p = 1 \Rightarrow x_{i,\alpha} - 3 \leq 0, \zeta_i^p = 0 \Rightarrow x_{i,\alpha} - 3 > 0 \qquad \forall i \in [0, N]$

[B2] $\zeta_i^{\delta,p} = 1 \Rightarrow x_{i,\alpha} - 3 \leq \delta, \zeta_i^p = 0 \Rightarrow x_{i,\alpha} - 3 > \delta \qquad \forall i \in [0, N]$

[C1] $\zeta_i^{\delta,p} = 0 \wedge \zeta_{i+1}^{\delta,p} = 1 \Rightarrow \zeta_i^p = 1, \quad \zeta_i^{\delta,p} = 1 \wedge \zeta_{i+1}^{\delta,p} = 0 \Rightarrow \zeta_i^p = 1 \qquad \forall i \in [0, N]$

[C2] $\langle p \rangle_i = \zeta_{i-1}^{\delta,p} \vee \zeta_i^{\delta,p}$

Manage truth values of $\square_{[0,4]} p$ in each interval $[\gamma_{\mathrm{i}-1}, \gamma_\mathrm{i}]$

[D1] $S_0^p = 0,$

$\langle p \rangle_i = 0 \Rightarrow S_i^p = 0, \langle p \rangle_i = 1 \Rightarrow S_i^p = S_{i-1}^p + (\gamma_i - \gamma_{i-1}) \qquad \forall i \in [1, N]$

[D2.1] $\neg\langle \varphi \rangle_i \vee ([\gamma_{i-1} + 0, \gamma_i + 4] \cap (\gamma_{j-1}, \gamma_j] \neq \emptyset) \vee \langle p \rangle_j \qquad \forall i \in [1, N], j \in [i, N]$

[D2.2] $\neg\langle \varphi \rangle_i \vee (\gamma_i + 4 \leq \gamma_N) \vee \langle p \rangle_j \qquad \forall i \in [1, N]$

[D3.1] $\langle \varphi \rangle_i \vee (\gamma_j \notin (\gamma_{i-1} + 4, \gamma_i + 4)) \vee (S_i^p \leq 4 - 0) \qquad \forall i \in [1, N], j \in [i, N]$

[D3.2] $\langle \varphi \rangle_i \vee (\gamma_{i+b} \notin [\gamma_{j-1}, \gamma_j]) \vee (S_i^p \leq \gamma_j - \gamma_i - 0) \qquad \forall i \in [1, N], j \in [i, N]$

[D3.3] $\langle \varphi \rangle_i \vee (\gamma_{i+b} \leq \gamma_N) \vee S_i^p \leq \gamma_N - \gamma_i - 0 \qquad \forall i \in [1, N]$

$\square_{[0,4]} p$ must be true at $t = 0$

[E1] $\langle \varphi \rangle_1 = 1$

# Correctness of our encoding

- With $\delta$ small enough and $N$ large enough, our algorithm can **always find a satisfying trace** (if exists).
  - In most application, this is not really a limitation.

**Soundness (Theorem 4.17)**

Let $\varphi$ be an STL formula in NNF (*negation-normal form*), $\mathcal{M}$ be a model, $\mathrm{N} \in \mathbb{N}$, and $\delta > 0$. If a feasible solution $v$ lies in our MILP constraints Enc$(\varphi, \mathcal{M}, N, \delta)$, the induced trace $\sigma$ has $\sigma \in \mathcal{L}(\mathcal{M})$ and $\sigma \vDash \varphi$.

**Completeness up to $\delta$ and $N$ (Theorem 4.18)**

Assume the setting above. If there is a piecewise-linear $\sigma \in \mathcal{L}(\mathcal{M})$ such that $[\![\sigma, \varphi]\!] \geq \delta$ (the quantitative sementics is greater than $\delta$), there is an feasible solution $v$ that lies in Enc$(\varphi, \mathcal{M}, N, \delta)$ for some $\mathrm{N} \in \mathbb{N}$.

# Outline

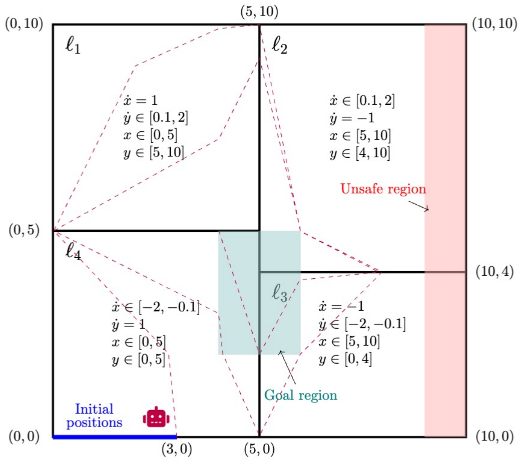
1. Challenges in Trace Synthesis with Complex STL Formulas
2. Previous Works
3. Our Algorithm: Variable-Interval MILP Encoding
4. **Experiments** 👉

# Experimental setting

## Tools

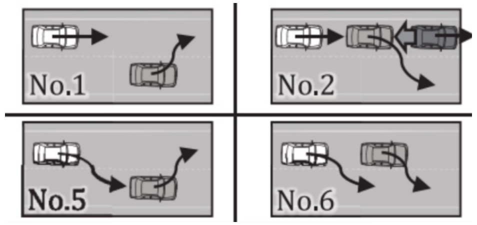
| | Supported class of model | Variable-interval |
|---|---|---|
| Breach [Donzé, CAV'10] | Black-box | No |
| ForeSee [Zhang+, FM'21] | Black-box | No |
| BluSTL [Donzé & Raman, ARCH15] | White-box (MILP) | No |
| STLmc [Yu+, CAV'22] | White-box (SMT) | Yes |
| STLts (our proposed method) | White-box (MILP) | Yes |

## Benchmark models and specs




Navigation of robot in a room (nondeterministic linear dynamics) [Bu+, ARCH22]

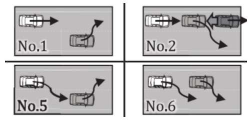
$\Diamond(\Box_{[0,3]}((x,y) \in \mathtt{goalR})) \wedge \Box(x \notin \mathtt{unsafeR})$

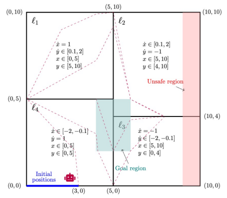


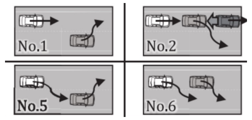

Rear-end near collision & ISO34502 (quadratic dynamics)

$(\Box(x_{\mathrm{f}} - x_{\mathrm{r}} \geq 0)) \wedge$
$\Diamond_{[0,9]}((\Box_{[0,1]}\mathtt{danger}) \wedge (\Box_{[0,1]} a_{\mathrm{r}} \geq 1) \wedge (\Diamond_{[1,5]} \neg \mathtt{danger})$

# Experimental results





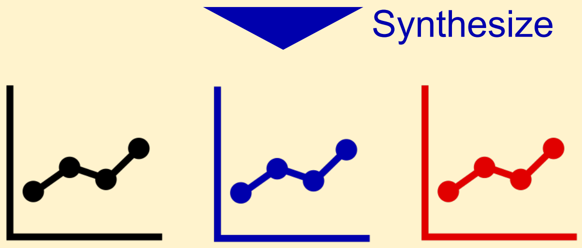

Our tool

| | STLts | | Breach | ForeSee | bluSTL | STLmc |
|---|---|---|---|---|---|---|
| RNC1 | 0.1 | (3) | 59.4 | 546.8 | (¶) | t/o |
| RNC2 | 0.3 | (4) | 9.3 | 104.3 | 14.3 | t/o |
| RNC3 | 0.1 | (3) | 81.3 | 197.4 | (¶) | t/o |
| NAV1 | 32.5 | (17) | (∗) | (∗) | (‡) | 16.5 |
| NAV2 | 2.1 | (11) | | | | 10.0 |
| ISO1 | 0.4 | (3) | 8.9 | t/o | (†) | (†) |
| ISO3 | 0.2 | (2) | t/o | t/o | | |
| ISO4 | 0.4 | (2) | t/o | t/o | | |
| ISO5 | 9.9 | (4) | 31.2 | 435.8 | | |
| ISO6 | 2.4 | (4) | t/o | 58.9 | | |
| ISO7 | 0.6 | (3) | 33.6 | 187.2 | | |
| ISO8 | 1.5 | (3) | 38.8 | t/o | | |

Time (sec) to synthesize a trace

**Latency small enough** (0.1~9.9s) for interactive inspection of STL specs

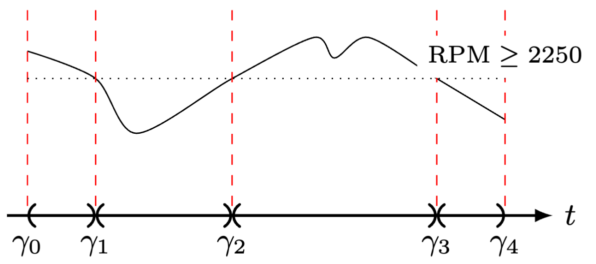
$$(\mathbf{G}p) \rightarrow (\mathbf{F}_{[0,T]}q)$$



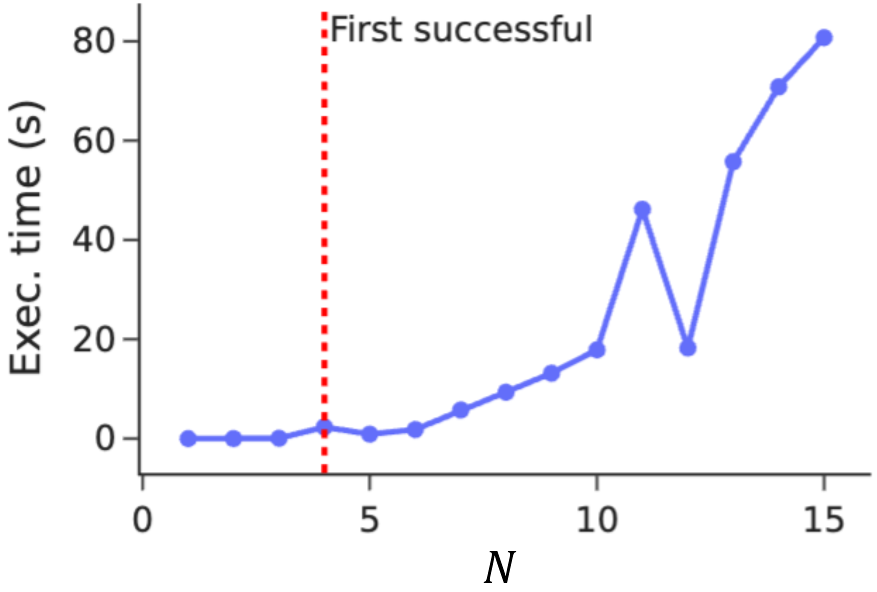

Example traces

# How should we choose the constant $N$?

- $N$ **should be sufficiently large** for completeness
- In practice, start with a small $N$ and incrementally try a larger $N$




$N$ is the number of variable-intervals (Here $N$=4)




ISO6 benchmark

Computational cost is **low if $N$ is small**

## Conclusion:

- First practical algorithm for handling complex STL formulas
  - Demonstrated with a real-world ISO example
- Proposed a consistent framework, $\delta$-stability, to relax the stable partitioning

## Future work:

- Application to falsification problem of black-box models
- Generate diverse examples as desired by users

| | Class of model | Variable-interval |
|---|---|---|
| Breach [Donzé, CAV'10] | Black-box | No |
| ForeSee [Zhang+, FM'21] | Black-box | No |
| BluSTL [Donzé & Raman, ARCH15] | MILP | No |
| STLmc [Yu+, CAV'22] | SMT | Yes |
| STLts (ours) | MILP | Yes |