**Highlights 2022**
of Logic, Games and Automata
Paris, June 28–July 1st, 2022

# Scalable Anytime Algorithms for Learning Fragments of Linear Temporal Logic (SCARLET)

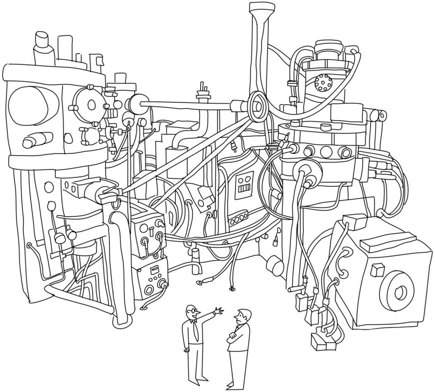

**Ritam Raha** Rajarshi Roy Nathanaël Fijalkow Daniel Neider

Universiteit Antwerpen
LaBRI
MAX PLANCK INSTITUTE FOR SOFTWARE SYSTEMS



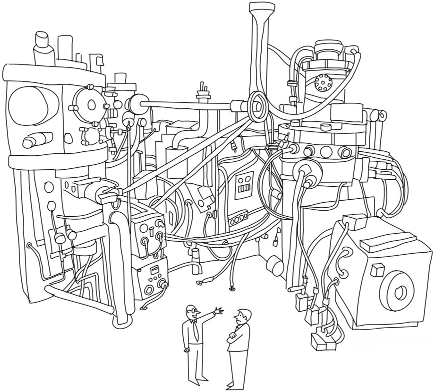
Appeared at TACAS'22

# Explainable AI



*Well, it looks too complicated*

# Explainable AI

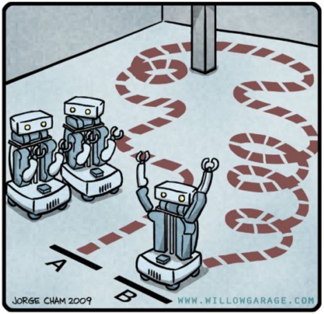


*Well, it looks too complicated*

**Goal:** Learn simple (human interpretable) models by observing complex systems

# Robot Motion-Planning

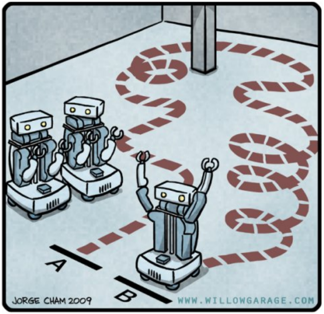
R.O.B.O.T. Comics




"HIS PATH-PLANNING MAY BE SUB-OPTIMAL, BUT IT'S GOT FLAIR."

# Robot Motion-Planning

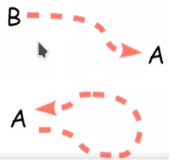
R.O.B.O.T. Comics



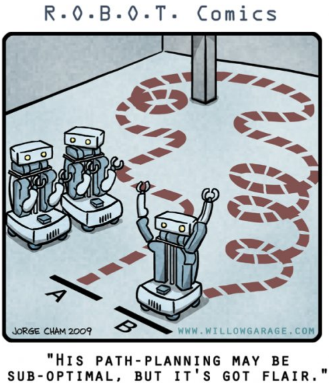

"HIS PATH-PLANNING MAY BE SUB-OPTIMAL, BUT IT'S GOT FLAIR."

*Positive*


*Negative*

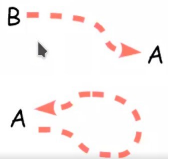


# Robot Motion-Planning




*Positive*


*Negative*




$A \wedge$ Finally $B$

# LTL as a descriptive model

**Linear Temporal Logic**
Eg. *Globally, Finally, Next*

**Syntax:**

$$\varphi ::= p \in \Sigma \mid \neg p \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \wedge \varphi_2 \mid \mathbf{X}\varphi \mid \mathbf{F}\varphi \mid \mathbf{G}\varphi \mid \varphi_1 \mathbf{U} \varphi_2$$

# LTL as a descriptive model

**Linear Temporal Logic** **<u>on finite words</u>** (Vardi & Giacomo '13)
Eg. *Globally, Finally, Next*

**Syntax:**

$$\varphi ::= p \in \Sigma \mid \neg p \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \wedge \varphi_2 \mid \mathbf{X}\varphi \mid \mathbf{F}\varphi \mid \mathbf{G}\varphi \mid \varphi_1 \mathbf{U} \varphi_2$$

# LTL as a descriptive model

**Linear Temporal Logic** **<u>on finite words</u>** (Vardi & Giacomo '13)
Eg. *Globally, Finally, Next*

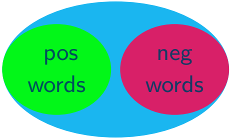
**Syntax:**

$$\varphi ::= p \in \Sigma \mid \neg p \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \wedge \varphi_2 \mid \mathbf{X}\varphi \mid \mathbf{F}\varphi \mid \mathbf{G}\varphi \mid \cancel{\varphi_1 \mathbf{U} \varphi_2}$$

The fragment: <u>LTL(F, X, G, ∧, ∨)</u>

# The learning problem

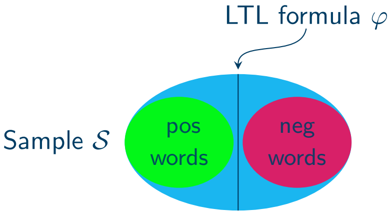
Sample $\mathcal{S}$




LTL Learning on Finite Words

**Input:** A set of positive words $P$ & negative words $N$

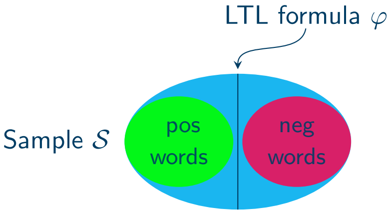
# The learning problem




**LTL LEARNING ON FINITE WORDS**

**Input:** A set of positive words $P$ & negative words $N$

# The learning problem




**LTL Learning on Finite Words**

**Input:** A set of positive words $P$ & negative words $N$

**Question:** Find a minimal LTL formula $\varphi$ such that, $\forall w \in P, w \models \varphi$ and $\forall w \in N, w \not\models \varphi$?

# State-of-the-Art

**Theorem (Fijalkow & Lagarde '21)**

*The learning problem for the fragments of LTL: LTL(X, $\wedge$), LTL(F, $\wedge$) and LTL(F, X, $\wedge$, $\vee$) is NP-complete.*

# State-of-the-Art

**Theorem (Fijalkow & Lagarde '21)**

*The learning problem for the fragments of LTL: LTL(X, $\wedge$), LTL(F, $\wedge$) and LTL(F, X, $\wedge$, $\vee$) is NP-complete.*

**Existing approaches:**

- SAT-Solvers - `FLIE` (Neider & Gavran '18)
- SyGuS solvers - `SYSLITE` (Arif et al. '20)

# State-of-the-Art

**Theorem (Fijalkow & Lagarde '21)**

*The learning problem for the fragments of LTL: LTL(X, $\wedge$), LTL(F, $\wedge$) and LTL(F, X, $\wedge$, $\vee$) is NP-complete.*

**Existing approaches:**

- SAT-Solvers - `FLIE` (Neider & Gavran '18)
- SyGuS solvers - `SYSLITE` (Arif et al. '20)

**Overview**

- For all LTL formulas of size $k$, check if separating.
- Increase $k$ and repeat.

# Towards Approximation

# Towards Approximation

**Overview.**

- Extract LTL patterns of increasing complexity from sample
  (Technique used: Dynamic Programming)

# Towards Approximation

**Overview.**

- Extract LTL patterns of increasing complexity from sample
  (Technique used: Dynamic Programming)

- Generate their Boolean combinations to find the (minimal) formula by solving **Boolean Set Cover** problem
  (Technique used: Greedy approximation or Decision Tree)

# Finding LTL patterns

**Sample $S$**

| **Positive Words** | **Negative Words** |
| --- | --- |
| pqqp | qqqq |
| qqpp | ppqp |

**Idea:**

**Candidate:**

**Formula:**

# Finding LTL patterns

**Sample** $S$

| **Positive Words** | **Negative Words** |
|---|---|
| pqqp | qqqq |
| qqpp | ppqp |

**Idea:** Find *separating patterns* with intervals

**Candidate:**

**Formula:**

# Finding LTL patterns

**Sample $S$**

| Positive Words | Negative Words |
| --- | --- |
| pqqp ✓ | qqqq |
| qqpp ✓ | ppqp |

**Idea:** Find *separating patterns* with intervals

**Candidate:** (1,q,>0,p)

**Formula:**

# Finding LTL patterns

**Sample** $S$

| **Positive Words** | **Negative Words** |
| --- | --- |
| pqqp ✓ | qqqq ✗ |
| qqpp ✓ | ppqp ✗ |

**Idea:** Find *separating patterns* with intervals

**Candidate:** (1,q,>0,p)

**Formula:**

# Finding LTL patterns

**Sample $S$**

| **Positive Words** | **Negative Words** |
| --- | --- |
| pqqp ✓ | qqqq ✘ |
| qqpp ✓ | ppqp ✘ |

**Idea:** Find *separating patterns* with intervals

**Candidate:** (1,q,>0,p)

**Formula:** $X(q \wedge Fp)$

# Directed LTL

LTL patterns that arise from the following grammar:

$$\varphi :\, = X^n p \quad | \quad FX^n p \quad | \quad X^n(p \wedge \varphi) \quad | \quad FX^n(p \wedge \varphi),$$

**Theorem**

*The boolean combination of dLTL formulas is as expressive as $LTL(F, X, \wedge, \vee)$*

# Directed LTL

LTL patterns that arise from the following grammar:

$$\varphi :\, = X^n p \quad | \quad FX^n p \quad | \quad X^n(p \wedge \varphi) \quad | \quad FX^n(p \wedge \varphi),$$

**Theorem**

*The boolean combination of dLTL formulas is as expressive as LTL(F, X, $\wedge$, $\vee$)*

**Dual:** $\neg$F $\neg\varphi$ =G $\varphi$ : swap positive and negative words!

# Directed LTL

LTL patterns that arise from the following grammar:

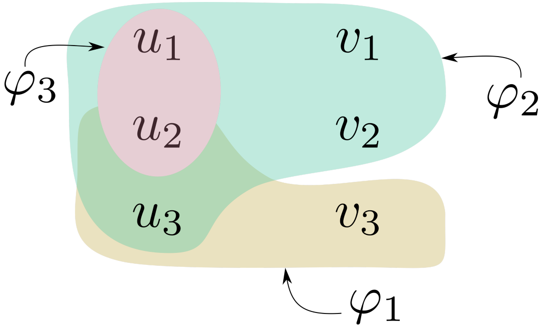
$$\varphi := X^n p \quad | \quad FX^n p \quad | \quad X^n(p \wedge \varphi) \quad | \quad FX^n(p \wedge \varphi),$$

**Theorem**

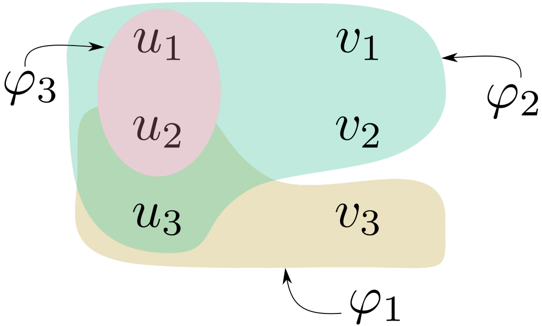
*The boolean combination of dLTL formulas is as expressive as LTL(F, X, $\wedge$, $\vee$)*

**Dual:** $\neg$F $\neg\varphi$ =G $\varphi$ : swap positive and negative words!

# Boolean Set Cover

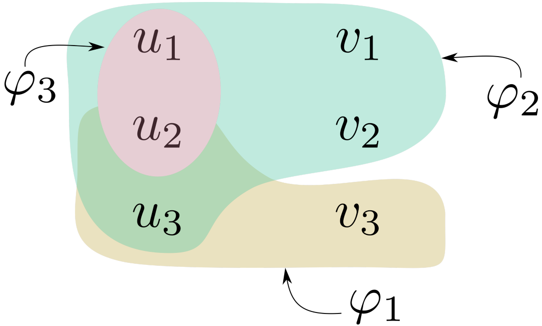


# Boolean Set Cover




**Problem:** Find the minimal boolean combination of formulas that separates the sample

# Boolean Set Cover




**Problem:** Find the minimal boolean combination of formulas that separates the sample

**Sol:** $(\varphi_1 \wedge \varphi_2) \vee \varphi_3$

# Boolean Set Cover

- A similar greedy approximation algorithm to classical set cover

# Boolean Set Cover

- A similar greedy approximation algorithm to classical set cover
- Another approach: Decision Trees

# Advantages of our approach

- **Anytime algorithm**

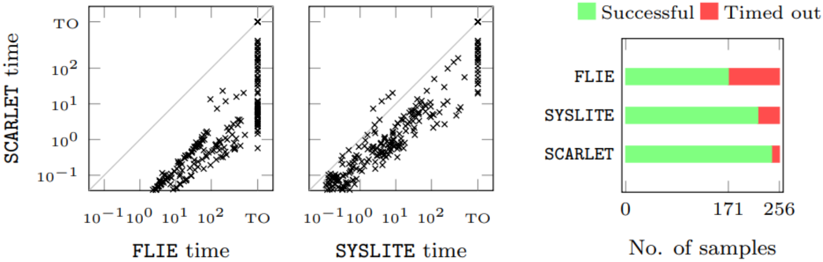
# Advantages of our approach

- **Anytime algorithm**
- **Optimized according to the Sample**

# Advantages of our approach

- **Anytime algorithm**
- **Optimized according to the Sample**
- **Noisy Data Setting**

# SCARLET



TO
$10^2$
$10^1$
$10^0$
$10^{-1}$
SCARLET time
$10^{-1}$ $10^0$ $10^1$ $10^2$ TO
FLIE time
$10^{-1}$ $10^0$ $10^1$ $10^2$ TO
SYSLITE time
Successful
Timed out
FLIE
SYSLITE
SCARLET
0
171
256
No. of samples

# Future Work/ Open Questions

# Future Work/ Open Questions

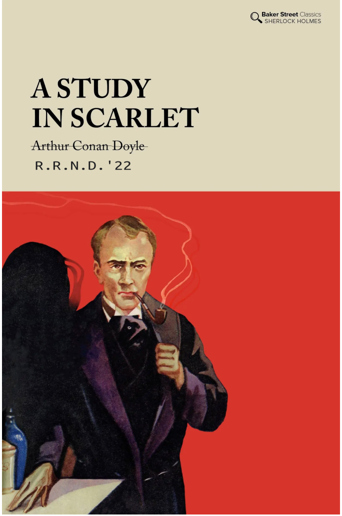
- Exact approximation factor of the algorithm

# Future Work/ Open Questions

- Exact approximation factor of the algorithm
- Capture more expressive power: learn formulas with U-operator

# Future Work/ Open Questions

- Exact approximation factor of the algorithm
- Capture more expressive power: learn formulas with U-operator
- Towards real-valued traces: learn formulas in STL


Baker Street Classics
SHERLOCK HOLMES
A STUDY IN SCARLET
~~Arthur Conan Doyle~~
R.R.N.D.'22

*Thank you!*