The Complexity of Learning LTL, CTL and ATL Formulas

Benjamin Bordais[1], Daniel Neider[1] and Rajarshi Roy[2]

[1]TU Dortmund, RC-Trust

[2]University of Oxford

STACS 2025

RESEARCH CENTER
TRUSTWORTHY DATA SCIENCE AND SECURITY
UA RUHR

UNIVERSITY OF OXFORD

Context

General setting

- Derive formal specification from observations of system behavior
- Formal specifications: written as **temporal logic** formulas
- Synthesis: in a **passive learning** context
- Goal: find tractable cases for the learning decision problem

Linear Temporal Logic (LTL)

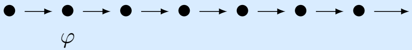
Basic characteristics LTL (Pnueli, 1977)

- Evaluated on infinite words where letters are sets of propositions
- Propositional logic + Temporal operators (X, F, G, U, etc.)

Linear Temporal Logic (LTL)

Basic characteristics LTL (Pnueli, 1977)

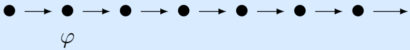
- Evaluated on infinite words where letters are sets of propositions
- Propositional logic + Temporal operators (X, F, G, U, etc.)
- Temporal operators:
  - (NeXt) X φ:



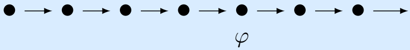
Linear Temporal Logic (LTL)

Basic characteristics LTL (Pnueli, 1977)

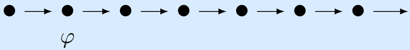
- Evaluated on infinite words where letters are sets of propositions
- Propositional logic + Temporal operators (X, F, G, U, etc.)
- Temporal operators:
  - (NeXt) X φ:

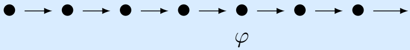


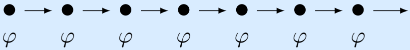
  - (Future) F φ:



Basic characteristics LTL (Pnueli, 1977)

- Evaluated on infinite words where letters are sets of propositions
- Propositional logic + Temporal operators (X, F, G, U, etc.)
- Temporal operators:
  - (NeXt) X φ:


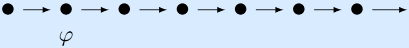
  - (Future) F φ:


  - (Globally) G φ:


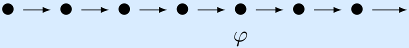
Linear Temporal Logic (LTL)

Basic characteristics LTL (Pnueli, 1977)

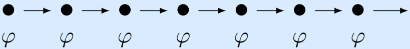
- Evaluated on infinite words where letters are sets of propositions
- Propositional logic + Temporal operators (X, F, G, U, etc.)
- Temporal operators:
  - (NeXt) X φ:



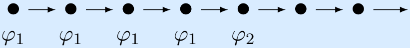
  - (Future) F φ:



  - (Globally) G φ:



  - (Until) φ_1 U φ_2:



Linear Temporal Logic (LTL)

Basic characteristics LTL

- Evaluated on infinite words where letters are sets of propositions
- Propositional logic + Temporal operators
- Set of all operators $\mathsf{Op} := \{\neg, \wedge, \vee, \Rightarrow, \ldots, \mathsf{X}, \mathsf{F}, \mathsf{G}, \mathsf{U}, \ldots\}$

Linear Temporal Logic (LTL)

Basic characteristics LTL

- Evaluated on infinite words where letters are sets of propositions
- Propositional logic + Temporal operators
- Set of all operators $\mathsf{Op} := \{\neg, \wedge, \vee, \Rightarrow, \ldots, \mathsf{X}, \mathsf{F}, \mathsf{G}, \mathsf{U}, \ldots\}$

Set of propositions

$\mathsf{Prop} := \{\mathsf{Ylw}, \mathsf{Red}, \mathsf{Blue}, \mathsf{Grn}, \mathsf{Prpl}, \mathsf{Blck}\}$

Linear Temporal Logic (LTL)

Basic characteristics LTL

- Evaluated on infinite words where letters are sets of propositions
- Propositional logic + Temporal operators
- Set of all operators $\mathsf{Op} := \{\neg, \wedge, \vee, \Rightarrow, \dots, \mathsf{X}, \mathsf{F}, \mathsf{G}, \mathsf{U}, \dots\}$

Set of propositions

$\mathsf{Prop} := \{\mathsf{Ylw}, \mathsf{Red}, \mathsf{Blue}, \mathsf{Grn}, \mathsf{Prpl}, \mathsf{Blck}\}$

An infinite word

$\pi = \{\mathsf{Ylw}, \mathsf{Grn}, \mathsf{Red}\} \cdot \{\mathsf{Blue}, \mathsf{Red}\} \cdot \{\mathsf{Red}\} \cdot \{\mathsf{Blck}, \mathsf{Red}\} \cdot \{\mathsf{Blue}, \mathsf{Prpl}\} \cdots$

Linear Temporal Logic (LTL)

Basic characteristics LTL

- Evaluated on infinite words where letters are sets of propositions
- Propositional logic + Temporal operators
- Set of all operators $\mathsf{Op} := \{\neg, \wedge, \vee, \Rightarrow, \ldots, \mathsf{X}, \mathsf{F}, \mathsf{G}, \mathsf{U}, \ldots\}$

Set of propositions

$\mathsf{Prop} := \{\mathsf{Ylw}, \mathsf{Red}, \mathsf{Blue}, \mathsf{Grn}, \mathsf{Prpl}, \mathsf{Blck}\}$

An infinite word

$\pi = \{\mathsf{Ylw}, \mathsf{Grn}, \mathsf{Red}\} \cdot \{\mathsf{Blue}, \mathsf{Red}\} \cdot \{\mathsf{Red}\} \cdot \{\mathsf{Blck}, \mathsf{Red}\} \cdot \{\mathsf{Blue}, \mathsf{Prpl}\} \cdots$

Example of evaluation

For $\varphi_1 := \mathsf{F}\ (\mathsf{Blck} \wedge \mathsf{X}\ \mathsf{Blue})$:

$$\pi \models \varphi_1$$

For $\varphi_2 := \mathsf{Red}\ \mathsf{U}\ \mathsf{Prpl}$:

$$\pi \models \varphi_2$$

Computation Tree Logic (CTL)

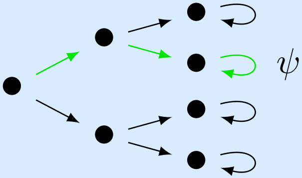
Basic characteristics CTL (Clarke, Emerson, 1981)

- Evaluated on labeled graphs (non-terminating)
- Propositional logic + Quantification temporal operators

Computation Tree Logic (CTL)

Basic characteristics CTL (Clarke, Emerson, 1981)

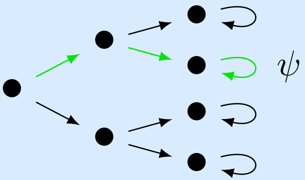
- Evaluated on labeled graphs (non-terminating)
- Propositional logic + Quantification temporal operators
- Quantification:
  - (Existential) $\exists\, \psi$:



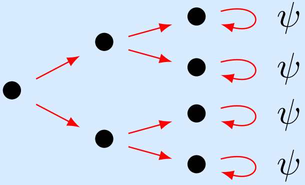
Computation Tree Logic (CTL)

Basic characteristics CTL (Clarke, Emerson, 1981)

- Evaluated on labeled graphs (non-terminating)
- Propositional logic + Quantification temporal operators
- Quantification:
  - (Existential) $\exists\ \psi$:



  - (Universal) $\forall\ \psi$:



Computation Tree Logic (CTL)

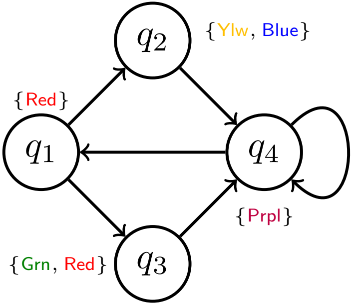
Basic characteristics CTL

- Evaluated on labeled graphs (non-terminating)
- Propositional logic + Quantification temporal operators

Computation Tree Logic (CTL)

Basic characteristics CTL

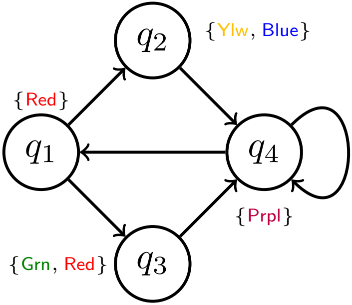
- Evaluated on labeled graphs (non-terminating)
- Propositional logic + Quantification temporal operators



Computation Tree Logic (CTL)

Basic characteristics CTL

- Evaluated on labeled graphs (non-terminating)
- Propositional logic + Quantification temporal operators



Example

- For $\varphi_1 := \exists\, \mathbf{X}$ Blue:
  - $q_1 \models \varphi_1$
  - $q_2, q_3, q_4 \not\models \varphi_1$
- For $\varphi_2 := \forall\, \mathbf{F}$ Prpl:
  - $q_1, q_2, q_3, q_4 \models \varphi_2$

Alternating-time Temporal Logic (ATL)

Basic characteristics ATL_k (Alur, Henzinger, Kupferman, 2002)

- Generalizes CTL
- Evaluated on labeled k-player games
- Propositional logic $+$ (Strategic quantification) temporal operators

Passive learning problem

Logic $\mathcal{L} \in \{\mathsf{LTL}, \mathsf{CTL}, \mathsf{ATL}_k \mid k \in \mathbb{N}_1\}$, subset of operators $\mathsf{O} \subseteq \mathsf{Op}$

Passive learning problem

Logic $\mathcal{L} \in \{\mathsf{LTL}, \mathsf{CTL}, \mathsf{ATL}_k \mid k \in \mathbb{N}_1\}$, subset of operators $\mathsf{O} \subseteq \mathsf{Op}$

The decision problem $\mathsf{PvLn}_{\mathcal{L}}(\mathsf{O})$:

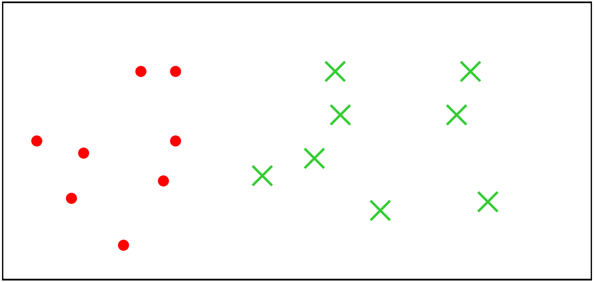
- Input: $\mathcal{L}$-instances in $\mathcal{S}_+$ and $\mathcal{S}_-$, and a bound B (in **unary**)

Passive learning problem

Logic $\mathcal{L} \in \{\mathsf{LTL}, \mathsf{CTL}, \mathsf{ATL}_k \mid k \in \mathbb{N}_1\}$, subset of operators $\mathsf{O} \subseteq \mathsf{Op}$

The decision problem $\mathsf{PvLn}_{\mathcal{L}}(\mathsf{O})$:

- Input: $\mathcal{L}$-instances in $\mathcal{S}_+$ and $\mathcal{S}_-$, and a bound B (in **unary**)

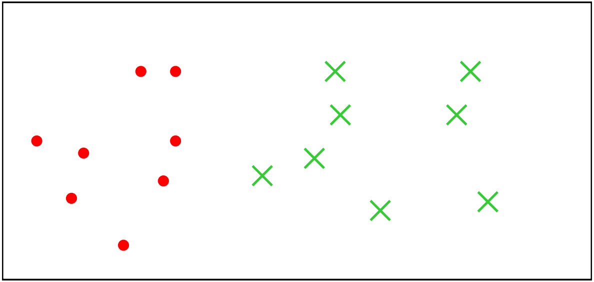


Passive learning problem

Logic $\mathcal{L} \in \{\mathsf{LTL}, \mathsf{CTL}, \mathsf{ATL}_k \mid k \in \mathbb{N}_1\}$, subset of operators $\mathsf{O} \subseteq \mathsf{Op}$

The decision problem $\mathsf{PvLn}_{\mathcal{L}}(\mathsf{O})$:

- Input: $\mathcal{L}$-instances in $\mathcal{S}_+$ and $\mathcal{S}_-$, and a bound B (in **unary**)
- Output: Yes iff there exists an $\mathcal{L}(\mathsf{O})$-formula φ of size at most B:
  - φ accepts all positive instances in $\mathcal{S}_+$
  - φ rejects all negative instances in $\mathcal{S}_-$

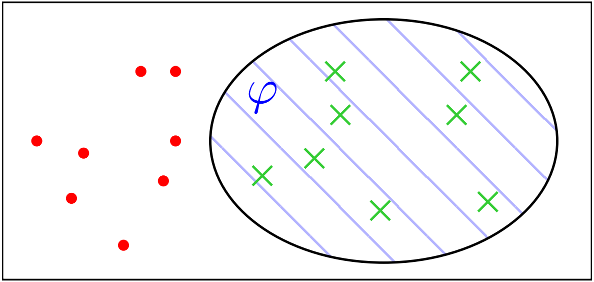


Passive learning problem

Logic $\mathcal{L} \in \{\mathsf{LTL}, \mathsf{CTL}, \mathsf{ATL}_k \mid k \in \mathbb{N}_1\}$, subset of operators $\mathsf{O} \subseteq \mathsf{Op}$

The decision problem $\mathsf{PvLn}_{\mathcal{L}}(\mathsf{O})$:

- Input: $\mathcal{L}$-instances in $\mathcal{S}_+$ and $\mathcal{S}_-$, and a bound B (in **unary**)
- Output: Yes iff there exists an $\mathcal{L}(\mathsf{O})$-formula φ of size at most B:
  - φ accepts all positive instances in $\mathcal{S}_+$
  - φ rejects all negative instances in $\mathcal{S}_-$



Passive learning problem

Logic $\mathcal{L} \in \{\mathsf{LTL}, \mathsf{CTL}, \mathsf{ATL}_k \mid k \in \mathbb{N}_1\}$, subset of operators $\mathsf{O} \subseteq \mathsf{Op}$

The decision problem $\mathsf{PvLn}_{\mathcal{L}}(\mathsf{O})$:

- Input: $\mathcal{L}$-instances in $\mathcal{S}_+$ and $\mathcal{S}_-$, and a bound B (in **unary**)
- Output: Yes iff there exists an $\mathcal{L}(\mathsf{O})$-formula φ of size at most B:
  - φ accepts all positive instances in $\mathcal{S}_+$
  - φ rejects all negative instances in $\mathcal{S}_-$

Bound in unary

- Why a bound? Prevents overfitting
- Why in unary? Generating a formula requires explicitly writing it

Goal

Study the complexity and look for tractable cases

The decision problem $\mathsf{PvLn}_{\mathcal{L}}(\mathsf{O})$ where:

- $\mathcal{L} \in \{\mathsf{LTL}, \mathsf{CTL}, \mathsf{ATL}_k \mid k \in \mathbb{N}_1\}$ ranges over temporal logics
- $\mathsf{O} \subseteq \mathsf{Op}$ ranges over subsets of operators

Goal

Study the complexity and look for tractable cases

The decision problem $\mathsf{PvLn}_{\mathcal{L}}(\mathsf{O})$ where:

- $\mathcal{L} \in \{\mathsf{LTL}, \mathsf{CTL}, \mathsf{ATL}_k \mid k \in \mathbb{N}_1\}$ ranges over temporal logics
- $\mathsf{O} \subseteq \mathsf{Op}$ ranges over subsets of operators

Central related work[1]

The complexity of LTL learning with finite words

[1]Learning temporal formulas from examples is hard, C. Mascle, N. Fijalkow, G. Lagarde, arXiv 2023.

Two straightforward observations

Problems in NP

For all $\mathcal{L} \in \{\mathsf{LTL}, \mathsf{CTL}, \mathsf{ATL}\}$ and subsets of operators $\mathsf{O} \subseteq \mathsf{Op}$:

the problem $\mathsf{PvLn}_{\mathcal{L}}(\mathsf{O})$ is in NP (use guess-and-check sub-routine)

Two straightforward observations

Problems in NP

For all $\mathcal{L} \in \{\mathsf{LTL}, \mathsf{CTL}, \mathsf{ATL}\}$ and subsets of operators $\mathsf{O} \subseteq \mathsf{Op}$:

the problem $\mathsf{PvLn}_{\mathcal{L}}(\mathsf{O})$ is in NP (use guess-and-check sub-routine)

Comparing LTL **and** CTL **learning,** CTL **and** ATL **learning**

For all subsets of operators $\mathsf{O} \subseteq \mathsf{Op}$:

- $\mathsf{PvLn}_{\mathsf{LTL}}(\mathsf{O}) \preceq_{\mathsf{LogSpace}} \mathsf{PvLn}_{\mathsf{CTL}}(\mathsf{O})$

Two straightforward observations

Problems in NP

For all $\mathcal{L} \in \{\mathsf{LTL}, \mathsf{CTL}, \mathsf{ATL}\}$ and subsets of operators $\mathsf{O} \subseteq \mathsf{Op}$:

the problem $\mathsf{PvLn}_{\mathcal{L}}(\mathsf{O})$ is in NP (use guess-and-check sub-routine)

Comparing LTL and CTL learning, CTL and ATL learning

For all subsets of operators $\mathsf{O} \subseteq \mathsf{Op}$:

- $\mathsf{PvLn}_{\mathsf{LTL}}(\mathsf{O}) \preceq_{\mathsf{LogSpace}} \mathsf{PvLn}_{\mathsf{CTL}}(\mathsf{O})$
- For all $k \in \mathbb{N}_1$, $\mathsf{PvLn}_{\mathsf{CTL}}(\mathsf{O}) \preceq_{\mathsf{LogSpace}} \mathsf{PvLn}_{\mathsf{ATL}_k}(\mathsf{O})$

Complexity with binary operators

Set of all binary operators

$\mathsf{BinOp} := \{\wedge, \vee, \Rightarrow, \ldots, \mathsf{U}, \mathsf{R}, \ldots\}$

Complexity with binary operators

Set of all binary operators

$\mathsf{BinOp} := \{\wedge, \vee, \Rightarrow, \dots, \mathsf{U}, \mathsf{R}, \dots\}$

Theorem

For all $\mathsf{O} \subseteq \mathsf{Op}$ s.t. $\mathsf{O} \cap \mathsf{BinOp} \neq \emptyset$:

The decision problem $\mathsf{PvLn}_{\mathsf{LTL}}(\mathsf{O})$ is NP-hard

Complexity with binary operators

Set of all binary operators

$\mathsf{BinOp} := \{\wedge, \vee, \Rightarrow, \ldots, \mathsf{U}, \mathsf{R}, \ldots\}$

Theorem

For all $\mathsf{O} \subseteq \mathsf{Op}$ s.t. $\mathsf{O} \cap \mathsf{BinOp} \neq \emptyset$:

The decision problem $\mathsf{PvLn}_{\mathsf{LTL}}(\mathsf{O})$ is NP-hard

Corollary

For all $\mathsf{O} \subseteq \mathsf{Op}$ s.t. $\mathsf{O} \cap \mathsf{BinOp} \neq \emptyset$ and $\mathcal{L} \in \{\mathsf{LTL}, \mathsf{CTL}, \mathsf{ATL}_k \mid k \in \mathbb{N}_1\}$:

The decision problem $\mathsf{PvLn}_{\mathcal{L}}(\mathsf{O})$ is NP-hard

Complexity with binary operators

Set of all binary operators

$\mathsf{BinOp} := \{\wedge, \vee, \Rightarrow, \ldots, \mathsf{U}, \mathsf{R}, \ldots\}$

Corollary

For all $\mathsf{O} \subseteq \mathsf{Op}$ s.t. $\mathsf{O} \cap \mathsf{BinOp} \neq \emptyset$ and $\mathcal{L} \in \{\mathsf{LTL}, \mathsf{CTL}, \mathsf{ATL}_k \mid k \in \mathbb{N}_1\}$:

The decision problem $\mathsf{PvLn}_{\mathcal{L}}(\mathsf{O})$ is NP-hard

Complexity with binary operators

Set of all binary operators

$\mathsf{BinOp} := \{\wedge, \vee, \Rightarrow, \dots, \mathsf{U}, \mathsf{R}, \dots\}$

Corollary

For all $\mathsf{O} \subseteq \mathsf{Op}$ s.t. $\mathsf{O} \cap \mathsf{BinOp} \neq \emptyset$ and $\mathcal{L} \in \{\mathsf{LTL}, \mathsf{CTL}, \mathsf{ATL}_k \mid k \in \mathbb{N}_1\}$:

The decision problem $\mathsf{PvLn}_{\mathcal{L}}(\mathsf{O})$ is NP-hard

First summary

	Unlimited use of binary operators	
LTL	NP-c	
CTL		
ATL_2		
ATL		

Look for tractable cases

Look for tractable cases

Add restrictions on the use of binary operators

For $n \in \mathbb{N}$, we define the decision problem $\mathsf{PvLn}_{\mathcal{L}}(\mathsf{O}, n)$:

Like $\mathsf{PvLn}_{\mathcal{L}}(\mathsf{O})$, but we consider $\mathcal{L}$-formulas with at most n occurrences of propositional binary operators in O

Look for tractable cases

Add restrictions on the use of binary operators

For $n \in \mathbb{N}$, we define the decision problem $\mathsf{PvLn}_{\mathcal{L}}(\mathsf{O}, n)$:

Like $\mathsf{PvLn}_{\mathcal{L}}(\mathsf{O})$, but we consider $\mathcal{L}$-formulas with at most n occurrences of propositional binary operators in O

Full summary

	Unlimited use of binary operators	Limited use of binary operators		
		With $\mathbf{X}$	Without $\mathbf{X}$	
			With $\mathbf{F}, \mathbf{G}$	Only $\mathbf{F}$ or only $\mathbf{G}$
LTL	NP-c	L		
CTL		NP-c	NL-c	
ATL_2		NP-c		P-c
ATL		NP-c		

Look for tractable cases

Add restrictions on the use of binary operators

For $n \in \mathbb{N}$, we define the decision problem $\mathsf{PvLn}_{\mathcal{L}}(\mathsf{O}, n)$:

Like $\mathsf{PvLn}_{\mathcal{L}}(\mathsf{O})$, but we consider $\mathcal{L}$-formulas with at most n occurrences of propositional binary operators in O

Full summary

<table>
<tr><th rowspan="3"></th><th rowspan="3">Unlimited use of binary operators</th><th colspan="3">Limited use of binary operators</th></tr>
<tr><th rowspan="2">With $\mathbf{X}$</th><th colspan="2">Without $\mathbf{X}$</th></tr>
<tr><th>With $\mathbf{F}, \mathbf{G}$</th><th>Only $\mathbf{F}$ or only $\mathbf{G}$</th></tr>
<tr><td>LTL</td><td rowspan="4">NP-c</td><td colspan="3">L</td></tr>
<tr><td>CTL</td><td>NP-c</td><td colspan="2">NL-c</td></tr>
<tr><td>ATL_2</td><td colspan="2">NP-c</td><td>P-c</td></tr>
<tr><td>ATL</td><td colspan="3">NP-c</td></tr>
</table>

Look for tractable cases

Add restrictions on the use of binary operators

For $n \in \mathbb{N}$, we define the decision problem $\mathsf{PvLn}_{\mathcal{L}}(\mathsf{O}, n)$:

Like $\mathsf{PvLn}_{\mathcal{L}}(\mathsf{O})$, but we consider $\mathcal{L}$-formulas with at most n occurrences of propositional binary operators in O

Full summary

	Unlimited use of binary operators	Limited use of binary operators		
		With $\mathbf{X}$	Without $\mathbf{X}$	
			With $\mathbf{F}, \mathbf{G}$	Only $\mathbf{F}$ or only $\mathbf{G}$
LTL	NP-c	L		
CTL		NP-c	NL-c	
ATL$_2$		NP-c		P-c
ATL		NP-c		

Abstract recipe

Hitting set problem

- Input: $l \in \mathbb{N}_1$, subsets $C = C_1, \ldots, C_n \subseteq [1, \ldots, l]$, $k \leq l$
- Output: Yes iff there is $H \subseteq [1, \ldots, l]$ such that:
  - $|H| \leq k$
  - for all $1 \leq i \leq n$, we have $H \cap C_i \neq \emptyset$

Abstract recipe

Hitting set problem

- Input: $l \in \mathbb{N}_1$, subsets $C = C_1, \ldots, C_n \subseteq [1, \ldots, l]$, $k \leq l$
- Output: Yes iff there is $H \subseteq [1, \ldots, l]$ such that:
  - $|H| \leq k$
  - for all $1 \leq i \leq n$, we have $H \cap C_i \neq \emptyset$

Pick a logic $\mathcal{L}$ and set of operators $\mathsf{O} \subseteq \mathsf{Op}$

From an instance (l, C, k) of the hitting set problem:

Abstract recipe

Hitting set problem

- Input: $l \in \mathbb{N}_1$, subsets $C = C_1, \ldots, C_n \subseteq [1, \ldots, l]$, $k \leq l$
- Output: Yes iff there is $H \subseteq [1, \ldots, l]$ such that:
  - $|H| \leq k$
  - for all $1 \leq i \leq n$, we have $H \cap C_i \neq \emptyset$

Pick a logic $\mathcal{L}$ and set of operators $\mathsf{O} \subseteq \mathsf{Op}$

From an instance (l, C, k) of the hitting set problem:

1. Define a type of $\mathcal{L}(\mathsf{O})$-formulas $\phi^l(H)$, for $H \subseteq [1, \ldots, l]$

Abstract recipe

Hitting set problem

- Input: $l \in \mathbb{N}_1$, subsets $C = C_1, \ldots, C_n \subseteq [1, \ldots, l]$, $k \leq l$
- Output: Yes iff there is $H \subseteq [1, \ldots, l]$ such that:
  - $|H| \leq k$
  - for all $1 \leq i \leq n$, we have $H \cap C_i \neq \emptyset$

Pick a logic $\mathcal{L}$ and set of operators $\mathsf{O} \subseteq \mathsf{Op}$

From an instance (l, C, k) of the hitting set problem:

1. Define a type of $\mathcal{L}(\mathsf{O})$-formulas $\phi^l(H)$, for $H \subseteq [1, \ldots, l]$
2. Define the bound $B \in \mathbb{N}$ and $(+, -)$ instances that a formula separates iff it is equal to some $\phi^l(H)$, for $H \subseteq [1, \ldots, l]$

Abstract recipe

Hitting set problem

- Input: $l \in \mathbb{N}_1$, subsets $C = C_1, \ldots, C_n \subseteq [1, \ldots, l]$, $k \leq l$
- Output: Yes iff there is $H \subseteq [1, \ldots, l]$ such that:
  - $|H| \leq k$
  - for all $1 \leq i \leq n$, we have $H \cap C_i \neq \emptyset$

Pick a logic $\mathcal{L}$ and set of operators $\mathsf{O} \subseteq \mathsf{Op}$

From an instance (l, C, k) of the hitting set problem:

1. Define a type of $\mathcal{L}(\mathsf{O})$-formulas $\phi^l(H)$, for $H \subseteq [1, \ldots, l]$
2. Define the bound $B \in \mathbb{N}$ and $(+, -)$ instances that a formula separates iff it is equal to some $\phi^l(H)$, for $H \subseteq [1, \ldots, l]$
3. Define a $-$instance that $\phi^l(H)$ accepts iff $|H| \geq k + 1$

Abstract recipe

Hitting set problem

- Input: $l \in \mathbb{N}_1$, subsets $C = C_1, \ldots, C_n \subseteq [1, \ldots, l]$, $k \leq l$
- Output: Yes iff there is $H \subseteq [1, \ldots, l]$ such that:
  - $|H| \leq k$
  - for all $1 \leq i \leq n$, we have $H \cap C_i \neq \emptyset$

Pick a logic $\mathcal{L}$ and set of operators O $\subseteq$ Op

From an instance (l, C, k) of the hitting set problem:

1. Define a type of $\mathcal{L}$(O)-formulas $\phi^l(H)$, for $H \subseteq [1, \ldots, l]$
2. Define the bound $B \in \mathbb{N}$ and $(+, -)$ instances that a formula separates iff it is equal to some $\phi^l(H)$, for $H \subseteq [1, \ldots, l]$
3. Define a $-$instance that $\phi^l(H)$ accepts iff $|H| \geq k + 1$
4. For $i \leq n$, define a $+$instance that $\phi^l(H)$ accepts iff $H \cap C_i \neq \emptyset$

Abstract recipe

Hitting set problem

- Input: $l \in \mathbb{N}_1$, subsets $C = C_1, \ldots, C_n \subseteq [1, \ldots, l]$, $k \leq l$
- Output: Yes iff there is $H \subseteq [1, \ldots, l]$ such that:
  - $|H| \leq k$
  - for all $1 \leq i \leq n$, we have $H \cap C_i \neq \emptyset$

Pick a logic $\mathcal{L}$ and set of operators $\mathsf{O} \subseteq \mathsf{Op}$

From an instance (l, C, k) of the hitting set problem:

1. Define a type of $\mathcal{L}(\mathsf{O})$-formulas $\phi^l(H)$, for $H \subseteq [1, \ldots, l]$
2. Define the bound $B \in \mathbb{N}$ and $(+, -)$ instances that a formula separates iff it is equal to some $\phi^l(H)$, for $H \subseteq [1, \ldots, l]$
3. Define a $-$instance that $\phi^l(H)$ accepts iff $|H| \geq k + 1$
4. For $i \leq n$, define a $+$instance that $\phi^l(H)$ accepts iff $H \cap C_i \neq \emptyset$

Application: when $\mathcal{L} = \mathsf{CTL}$, $\mathsf{X} \in \mathsf{O}$

Step 1: type of formulas

For $l \in \mathbb{N}_1$ and $H \subseteq [1, \ldots, l]$:

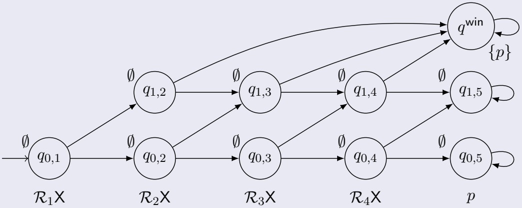
$$\phi^l(H) := \mathcal{R}_1\mathsf{X}\, \mathcal{R}_2\mathsf{X}\ \ldots\ \mathcal{R}_l\mathsf{X}\, p$$

with $\mathcal{R}_i \in \{\exists, \forall\}$ and $\mathcal{R}_i = \exists$ iff $i \in H$.

Application: when $\mathcal{L} = \mathsf{CTL}$, $\mathsf{X} \in \mathsf{O}$

Step 3: bounding the size of H = bounding the number of $\exists$

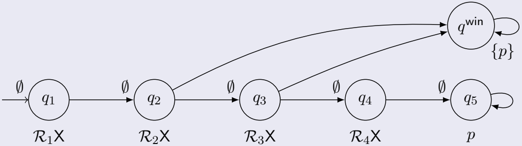
- When $l = 4$ and $k = 1$, define a $-$instance
- Kripke structure $K_{4,2\leq}$:



- We have $K_{4,2\leq} \models \phi^4(H)$ iff $|H| \geq 2$

Application: when $\mathcal{L} = \mathsf{CTL}$, $\mathsf{X} \in \mathsf{O}$

Step 4: H intersects the set C_i = an $\exists$ index is in C_i

- When $l = 4$ and $C_i = \{2, 3\} \subseteq [1, \ldots, 4]$, define a $+$instance
- Kripke structure K_{4,C_i}: 

- We have $K_{4,C_i} \models \phi^4(H)$ iff $H \cap C_i = \emptyset$

Summary of our results

Summary

	Unlimited use of binary operators	Limited use of binary operators		
		With $\mathbf{X}$	Without $\mathbf{X}$	
			With $\mathbf{F}, \mathbf{G}$	Only $\mathbf{F}$ or only $\mathbf{G}$
LTL	NP-c	L		
CTL		NP-c	NL-c	
ATL_2		NP-c		P-c
ATL		NP-c		

Take home message and future work

The passive learning of temporal logic formulas

- NP-hard as soon as there is an unlimited use of binary operators
- By limiting the use of binary operators, we recover some tractable cases

Take home message and future work

The passive learning of temporal logic formulas

- NP-hard as soon as there is an unlimited use of binary operators
- By limiting the use of binary operators, we recover some tractable cases

Various questions could be explored

- Allow a restricted use of binary temporal operators
- Look for tractable approximation algorithms
- What if the bound is written in binary?
- ...