# How undecidable are HyperLTL and HyperCTL$^*$?

**Marie Fortin**, Louwe Kuijer, Patrick Totzke, Martin Zimmermann
University of Liverpool

Logic & Semantics Seminar, Cambridge – March 4, 2022

# Hyperproperties [Clarkson, Schneider 2010]

Specifications that relate multiple executions of a system, such as in information-flow security policies.

# Hyperproperties [Clarkson, Schneider 2010]

Specifications that relate multiple executions of a system, such as in information-flow security policies.

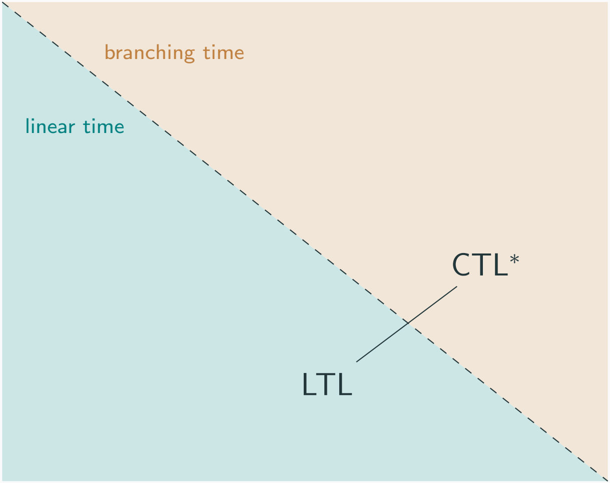

e.g. "no secret information should leak to low-level users"

# Hyperproperties [Clarkson, Schneider 2010]

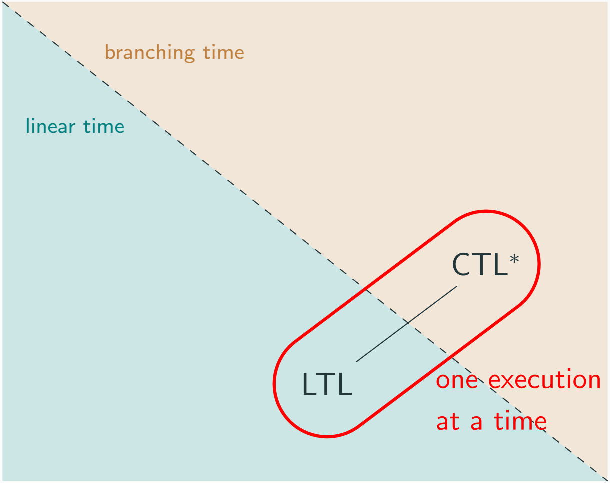
Specifications that relate multiple executions of a system, such as in information-flow security policies.

- Noninterference
- Observational determinism
- Declassification
- $\cdots$

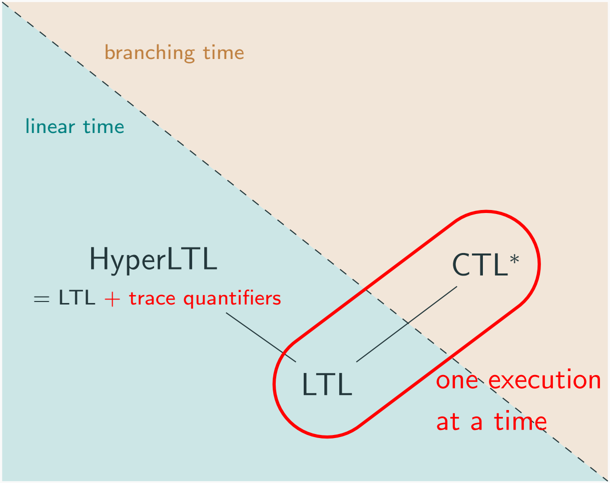
# HyperLTL and HyperCTL*



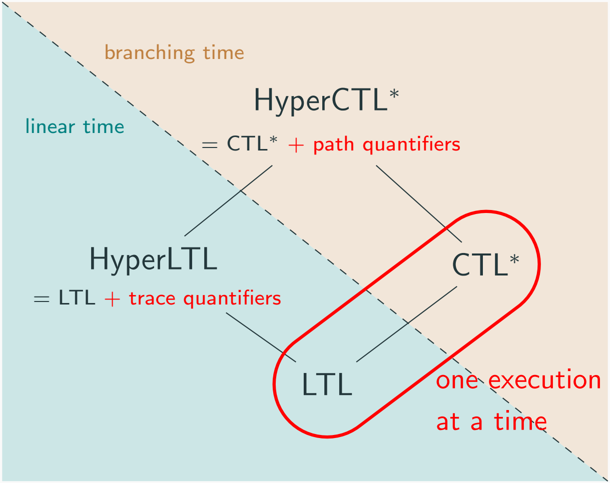
# HyperLTL and HyperCTL*



# HyperLTL and HyperCTL$^*$



# HyperLTL and HyperCTL*



# Linear Temporal Logic (LTL)

**LTL**

$$\psi ::= a \mid \neg\psi \mid \psi \lor \psi \mid \psi \land \psi \mid \mathsf{X}\,\psi \mid \psi\,\mathsf{U}\,\psi \mid \mathsf{F}\,\psi \mid \mathsf{G}\,\psi$$

# Linear Temporal Logic (LTL)

**LTL**

$$\psi ::= a \mid \neg\psi \mid \psi \vee \psi \mid \psi \wedge \psi \mid \mathsf{X}\,\psi \mid \psi\,\mathsf{U}\,\psi \mid \mathsf{F}\,\psi \mid \mathsf{G}\,\psi$$

$a \in \mathsf{AP}$
(atomic proposition)

# Linear Temporal Logic (LTL)

## LTL

$$\psi ::= a \mid \neg\psi \mid \psi \vee \psi \mid \psi \wedge \psi \mid \mathsf{X}\,\psi \mid \psi\,\mathsf{U}\,\psi \mid \mathsf{F}\,\psi \mid \mathsf{G}\,\psi$$

$a \in \mathsf{AP}$
(atomic proposition)

"Next $\psi$"

# Linear Temporal Logic (LTL)

**LTL**

$$\psi ::= a \mid \neg\psi \mid \psi \vee \psi \mid \psi \wedge \psi \mid \mathsf{X}\,\psi \mid \psi\,\mathsf{U}\,\psi \mid \mathsf{F}\,\psi \mid \mathsf{G}\,\psi$$

$a \in$ AP
(atomic proposition)

"Next $\psi$"

"$\psi$ Until $\psi$"

# Linear Temporal Logic (LTL)

**LTL**

$$\psi ::= a \mid \neg\psi \mid \psi \vee \psi \mid \psi \wedge \psi \mid \mathsf{X}\,\psi \mid \psi\,\mathsf{U}\,\psi \mid \mathsf{F}\,\psi \mid \mathsf{G}\,\psi$$

$a \in$ AP (atomic proposition)

"Next $\psi$" "$\psi$ Until $\psi$" "Eventually $\psi$"

# Linear Temporal Logic (LTL)

**LTL**

$$\psi ::= a \mid \neg\psi \mid \psi \vee \psi \mid \psi \wedge \psi \mid \mathsf{X}\,\psi \mid \psi\,\mathsf{U}\,\psi \mid \mathsf{F}\,\psi \mid \mathsf{G}\,\psi$$

$a \in \mathsf{AP}$
(atomic proposition)

"Next $\psi$" "$\psi$ Until $\psi$" "Eventually $\psi$" "Globally $\psi$"

# Linear Temporal Logic (LTL)

**LTL**

$$\psi ::= a \mid \neg\psi \mid \psi \vee \psi \mid \psi \wedge \psi \mid \mathsf{X}\,\psi \mid \psi\,\mathsf{U}\,\psi \mid \mathsf{F}\,\psi \mid \mathsf{G}\,\psi$$

$a \in \mathsf{AP}$
(atomic proposition)

"Next $\psi$" "$\psi$ Until $\psi$" "Eventually $\psi$" "Globally $\psi$"

## Examples

# Linear Temporal Logic (LTL)

**LTL**

$$\psi ::= a \mid \neg\psi \mid \psi \vee \psi \mid \psi \wedge \psi \mid \mathsf{X}\,\psi \mid \psi\,\mathsf{U}\,\psi \mid \mathsf{F}\,\psi \mid \mathsf{G}\,\psi$$

$a \in \mathsf{AP}$ (atomic proposition)

"Next $\psi$" "$\psi$ Until $\psi$" "Eventually $\psi$" "Globally $\psi$"

## Examples

- Safety: $\mathsf{G}\,\neg\texttt{bad}$

# Linear Temporal Logic (LTL)

## LTL

$$\psi ::= a \mid \neg\psi \mid \psi \vee \psi \mid \psi \wedge \psi \mid \mathsf{X}\,\psi \mid \psi\,\mathsf{U}\,\psi \mid \mathsf{F}\,\psi \mid \mathsf{G}\,\psi$$

$a \in \mathsf{AP}$
(atomic proposition)

"Next $\psi$" "$\psi$ Until $\psi$" "Eventually $\psi$" "Globally $\psi$"

**Examples**

- Safety: $\mathsf{G}\,\neg\texttt{bad}$
- Liveness: $\mathsf{G}\,\mathsf{F}\,\texttt{active}$

# Linear Temporal Logic (LTL)

**LTL**

$$\psi ::= a \mid \neg\psi \mid \psi \vee \psi \mid \psi \wedge \psi \mid \mathsf{X}\,\psi \mid \psi\,\mathsf{U}\,\psi \mid \mathsf{F}\,\psi \mid \mathsf{G}\,\psi$$

$a \in \mathsf{AP}$
(atomic proposition)

"Next $\psi$" "$\psi$ Until $\psi$" "Eventually $\psi$" "Globally $\psi$"

## Examples

- Safety: $\mathsf{G}\,\neg\mathtt{bad}$
- Liveness: $\mathsf{G}\,\mathsf{F}\,\mathtt{active}$
- $\mathsf{G}(\mathtt{request} \to \mathsf{X}(\neg\mathtt{request}\,\mathsf{U}\,\mathtt{grant}))$

  *"every request is eventually granted, and there can be no other request in the meantime"*

# Linear Temporal Logic (LTL)

**LTL**

$$\psi ::= a \mid \neg\psi \mid \psi \vee \psi \mid \psi \wedge \psi \mid \mathsf{X}\,\psi \mid \psi\,\mathsf{U}\,\psi \mid \mathsf{F}\,\psi \mid \mathsf{G}\,\psi$$

$a \in \mathsf{AP}$
(atomic proposition)

"Next $\psi$" "$\psi$ Until $\psi$" "Eventually $\psi$" "Globally $\psi$"

## Examples

- Safety: $\mathsf{G}\,\neg\texttt{bad}$
- Liveness: $\mathsf{G}\,\mathsf{F}\,\texttt{active}$
- $\mathsf{G}(\texttt{request} \rightarrow \mathsf{X}(\neg\texttt{request}\;\mathsf{U}\;\texttt{grant}))$

  *"every request is eventually granted, and there can be no other request in the meantime"*

Properties of individual traces $\mathbb{N} \rightarrow 2^{\mathsf{AP}}$

Cannot compare executions

# HyperLTL [Clarkson et al. 2014]

## Syntax of HyperLTL

$$\varphi ::= \exists \pi . \varphi \mid \forall \pi . \varphi \mid \psi$$

$$\psi ::= a_\pi \mid \neg \psi \mid \psi \vee \psi \mid \psi \wedge \psi \mid \mathsf{X}\, \psi \mid \psi\, \mathsf{U}\, \psi \mid \mathsf{F}\, \psi \mid \mathsf{G}\, \psi$$

"Next $\psi$" "$\psi$ Until $\psi$" "Eventually $\psi$" "Globally $\psi$"

# HyperLTL [Clarkson et al. 2014]

## Syntax of HyperLTL

$\varphi ::= \exists\pi.\varphi \mid \forall\pi.\varphi \mid \psi$

$\psi ::= a_\pi \mid \neg\psi \mid \psi \vee \psi \mid \psi \wedge \psi \mid \mathsf{X}\,\psi \mid \psi\,\mathsf{U}\,\psi \mid \mathsf{F}\,\psi \mid \mathsf{G}\,\psi$

"Next $\psi$" "$\psi$ Until $\psi$" "Eventually $\psi$" "Globally $\psi$"

# HyperLTL [Clarkson et al. 2014]

## Syntax of HyperLTL

$$\varphi ::= \exists\pi.\varphi \mid \forall\pi.\varphi \mid \psi$$

$$\psi ::= a_\pi \mid \neg\psi \mid \psi \vee \psi \mid \psi \wedge \psi \mid \mathsf{X}\,\psi \mid \psi\,\mathsf{U}\,\psi \mid \mathsf{F}\,\psi \mid \mathsf{G}\,\psi$$

"$p$ holds at the current position on trace $\pi$"

"Next $\psi$" "$\psi$ Until $\psi$" "Eventually $\psi$" "Globally $\psi$"

# HyperLTL [Clarkson et al. 2014]

## Syntax of HyperLTL

$$\varphi ::= \exists\pi.\varphi \mid \forall\pi.\varphi \mid \psi$$

$$\psi ::= a_\pi \mid \neg\psi \mid \psi \vee \psi \mid \psi \wedge \psi \mid \mathsf{X}\,\psi \mid \psi\,\mathsf{U}\,\psi \mid \mathsf{F}\,\psi \mid \mathsf{G}\,\psi$$

"$p$ holds at the current position on trace $\pi$"

"Next $\psi$" "$\psi$ Until $\psi$" "Eventually $\psi$" "Globally $\psi$"

**Example:**

$$\forall\pi.\forall\pi'.\ \mathsf{G}(\mathtt{in\_public}_\pi \leftrightarrow \mathtt{in\_public}_{\pi'}) \rightarrow \mathsf{G}(\mathtt{out\_public}_\pi \leftrightarrow \mathtt{out\_public}_{\pi'})$$

*"Any two traces with the same public input have the same public output"*

# HyperLTL [Clarkson et al. 2014]

## Syntax of HyperLTL

$$\varphi ::= \exists\pi.\varphi \mid \forall\pi.\varphi \mid \psi$$

$$\psi ::= a_\pi \mid \neg\psi \mid \psi \vee \psi \mid \psi \wedge \psi \mid \mathsf{X}\,\psi \mid \psi\,\mathsf{U}\,\psi \mid \mathsf{F}\,\psi \mid \mathsf{G}\,\psi$$

"$p$ holds at the current position on trace $\pi$"

"Next $\psi$" "$\psi$ Until $\psi$" "Eventually $\psi$" "Globally $\psi$"

**Example:**

$$\forall\pi.\forall\pi'.\ \mathsf{G}(\mathtt{in\_public}_\pi \leftrightarrow \mathtt{in\_public}_{\pi'}) \rightarrow \mathsf{G}(\mathtt{out\_public}_\pi \leftrightarrow \mathtt{out\_public}_{\pi'})$$

*"Any two traces with the same public input have the same public output"*

Always in prenex normal form: $\underbrace{Q_1\pi_1.Q_n\pi_n\ldots}_{\text{trace quantifiers}}\ \underbrace{\varphi}_{\text{LTL formula}}$

# HyperLTL [Clarkson et al. 2014]

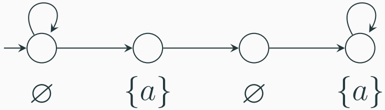
A transition system $\mathcal{T}$ satisfies a HyperLTL formula $\varphi$ if $Traces(\mathcal{T}) \models \varphi$.

# HyperLTL [Clarkson et al. 2014]

A transition system $\mathcal{T}$ satisfies a HyperLTL formula $\varphi$ if $Traces(\mathcal{T}) \models \varphi$.

**Example**

$\mathcal{T} =$

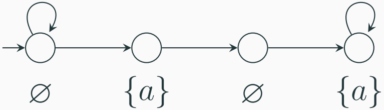


# HyperLTL [Clarkson et al. 2014]

A transition system $\mathcal{T}$ satisfies a HyperLTL formula $\varphi$ if $Traces(\mathcal{T}) \models \varphi$.

**Example**

$\mathcal{T} =$



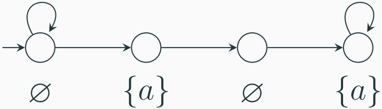

$$
\begin{array}{rllllllll}
Traces(\mathcal{T}) = \{ & \varnothing & \{a\} & \varnothing & \{a\} & \{a\} & \{a\} & \{a\} & \cdots, \\
& \varnothing & \varnothing & \{a\} & \varnothing & \{a\} & \{a\} & \{a\} & \cdots, \\
& \varnothing & \varnothing & \varnothing & \{a\} & \varnothing & \{a\} & \{a\} & \cdots, \\
& \cdots & \} & & & & & &
\end{array}
$$

# HyperLTL [Clarkson et al. 2014]

A transition system $\mathcal{T}$ satisfies a HyperLTL formula $\varphi$ if $Traces(\mathcal{T}) \models \varphi$.

**Example**

$\mathcal{T} =$




$$
\begin{array}{llllllllll}
Traces(\mathcal{T}) = \{ & \varnothing & \{a\} & \varnothing & \{a\} & \{a\} & \{a\} & \{a\} & \cdots, \\
& \varnothing & \varnothing & \{a\} & \varnothing & \{a\} & \{a\} & \{a\} & \cdots, \\
& \varnothing & \varnothing & \varnothing & \{a\} & \varnothing & \{a\} & \{a\} & \cdots, \\
& \cdots & \} & & & & & &
\end{array}
$$

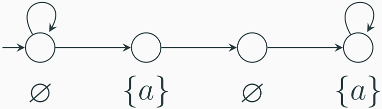
$$\mathcal{T} \models \forall\pi.\ \exists\pi'.\ \mathsf{G}\,(a_\pi \leftrightarrow \mathsf{X}\,a_{\pi'})$$

# HyperLTL [Clarkson et al. 2014]

A transition system $\mathcal{T}$ satisfies a HyperLTL formula $\varphi$ if $\mathit{Traces}(\mathcal{T}) \models \varphi$.

**Example**

$\mathcal{T} =$




$$
\begin{array}{lcccccccc}
\mathit{Traces}(\mathcal{T}) = \{ & \varnothing & \{a\} & \varnothing & \{a\} & \{a\} & \{a\} & \{a\} & \cdots, \\
& \varnothing & \varnothing & \{a\} & \varnothing & \{a\} & \{a\} & \{a\} & \cdots, \\
& \varnothing & \varnothing & \varnothing & \{a\} & \varnothing & \{a\} & \{a\} & \cdots, \\
& \cdots & \} & & & & & &
\end{array}
$$

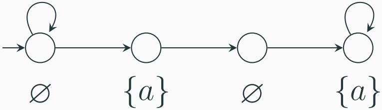
$$\mathcal{T} \models \forall \pi.\ \exists \pi'.\ \mathsf{G}\,(a_\pi \leftrightarrow \mathsf{X}\, a_{\pi'})$$

# HyperLTL [Clarkson et al. 2014]

A transition system $\mathcal{T}$ satisfies a HyperLTL formula $\varphi$ if $Traces(\mathcal{T}) \models \varphi$.

**Example**

$\mathcal{T} =$




$$
\begin{aligned}
Traces(\mathcal{T}) = \{\ & \varnothing \quad \{a\} \quad \varnothing \quad \{a\} \quad \{a\} \quad \{a\} \quad \{a\} \quad \cdots, \\
& \varnothing \quad \varnothing \quad \{a\} \quad \varnothing \quad \{a\} \quad \{a\} \quad \{a\} \quad \cdots, \\
& \varnothing \quad \varnothing \quad \varnothing \quad \{a\} \quad \varnothing \quad \{a\} \quad \{a\} \quad \cdots, \\
& \cdots \quad \}
\end{aligned}
$$

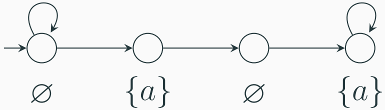
$\mathcal{T} \models \forall \pi.\ \exists \pi'.\ \mathsf{G}\,(a_\pi \leftrightarrow \mathsf{X}\, a_{\pi'})$

# HyperLTL [Clarkson et al. 2014]

A transition system $\mathcal{T}$ satisfies a HyperLTL formula $\varphi$ if $\mathit{Traces}(\mathcal{T}) \models \varphi$.

**Example**

$\mathcal{T} =$




$$
\begin{array}{llllllllll}
\mathit{Traces}(\mathcal{T}) = \{ & \varnothing & \{a\} & \varnothing & \{a\} & \{a\} & \{a\} & \{a\} & \cdots, \\
 & \varnothing & \varnothing & \{a\} & \varnothing & \{a\} & \{a\} & \{a\} & \cdots, \\
 & \varnothing & \varnothing & \varnothing & \{a\} & \varnothing & \{a\} & \{a\} & \cdots, \\
 & \cdots & \} & & & & & &
\end{array}
$$

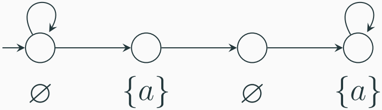
$\mathcal{T} \models \forall \pi.\ \exists \pi'.\ \mathsf{G}\,(a_\pi \leftrightarrow \mathsf{X}\, a_{\pi'})$

# HyperLTL [Clarkson et al. 2014]

A transition system $\mathcal{T}$ satisfies a HyperLTL formula $\varphi$ if $\mathit{Traces}(\mathcal{T}) \models \varphi$.

**Example**

$\mathcal{T} =$



$$
\begin{array}{lllllllll}
\mathit{Traces}(\mathcal{T}) = \{ & \varnothing & \{a\} & \varnothing & \{a\} & \{a\} & \{a\} & \{a\} & \cdots, \\
 & \varnothing & \varnothing & \{a\} & \varnothing & \{a\} & \{a\} & \{a\} & \cdots, \\
 & \varnothing & \varnothing & \varnothing & \{a\} & \varnothing & \{a\} & \{a\} & \cdots, \\
 & \cdots & \} & & & & & &
\end{array}
$$

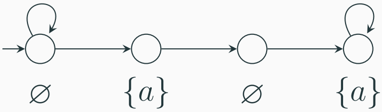
$\mathcal{T} \models \forall \pi.\ \exists \pi'.\ \mathsf{G}\,(a_\pi \leftrightarrow \mathsf{X}\, a_{\pi'})$

# HyperLTL [Clarkson et al. 2014]

A transition system $\mathcal{T}$ satisfies a HyperLTL formula $\varphi$ if $\mathit{Traces}(\mathcal{T}) \models \varphi$.

**Example**

$\mathcal{T} =$




$$\begin{array}{llllllllll}
\mathit{Traces}(\mathcal{T}) = \{ & \varnothing & \{a\} & \varnothing & \{a\} & \{a\} & \{a\} & \{a\} & \cdots, \\
& \varnothing & \varnothing & \{a\} & \varnothing & \{a\} & \{a\} & \{a\} & \cdots, \\
& \varnothing & \varnothing & \varnothing & \{a\} & \varnothing & \{a\} & \{a\} & \cdots, \\
& \cdots & \} & & & & & & 
\end{array}$$

$\mathcal{T} \models \forall \pi.\ \exists \pi'.\ \mathsf{G}\,(a_\pi \leftrightarrow \mathsf{X}\, a_{\pi'})$

# HyperCTL$^*$ [Clarkson et al. 2014]

# HyperCTL$^*$ [Clarkson et al. 2014]

**Syntax of HyperCTL$^*$**

$$\varphi ::= \exists \pi . \varphi \mid \forall \pi . \varphi \mid a_\pi \mid \neg \varphi \mid \varphi \vee \varphi \mid \mathsf{X}\,\varphi \mid \varphi\,\mathsf{U}\,\varphi \mid \mathsf{F}\,\varphi \mid \mathsf{G}\,\varphi$$

# HyperCTL$^*$ [Clarkson et al. 2014]

**Syntax of HyperCTL$^*$**

$$\varphi ::= \exists \pi . \varphi \mid \forall \pi . \varphi \mid a_\pi \mid \neg \varphi \mid \varphi \vee \varphi \mid \mathsf{X}\, \varphi \mid \varphi\, \mathsf{U}\, \varphi \mid \mathsf{F}\, \varphi \mid \mathsf{G}\, \varphi$$

- No prenex normal form

# HyperCTL$^*$ [Clarkson et al. 2014]

**Syntax of HyperCTL$^*$**

$$\varphi ::= \exists \pi. \varphi \mid \forall \pi. \varphi \mid a_\pi \mid \neg \varphi \mid \varphi \vee \varphi \mid \mathsf{X}\, \varphi \mid \varphi\, \mathsf{U}\, \varphi \mid \mathsf{F}\, \varphi \mid \mathsf{G}\, \varphi$$

- No prenex normal form
- Branching-time semantics:
  - evaluation over transition systems/computation trees rather than sets of traces
  - quantifiers range over paths in the transition system, starting in the latest state

# HyperCTL* [Clarkson et al. 2014]

**Syntax of HyperCTL***

$$\varphi ::= \exists\pi.\varphi \mid \forall\pi.\varphi \mid a_\pi \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{X}\,\varphi \mid \varphi\,\mathsf{U}\,\varphi \mid \mathsf{F}\,\varphi \mid \mathsf{G}\,\varphi$$

- No prenex normal form
- Branching-time semantics:
  - evaluation over transition systems/computation trees rather than sets of traces
  - quantifiers range over paths in the transition system, starting in the latest state
- Time progresses synchronously in all branches when evaluating temporal operators

# HyperCTL$^*$ [Clarkson et al. 2014]

**Syntax of HyperCTL$^*$**

$$\varphi ::= \exists\pi.\varphi \mid \forall\pi.\varphi \mid a_\pi \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{X}\,\varphi \mid \varphi\,\mathsf{U}\,\varphi \mid \mathsf{F}\,\varphi \mid \mathsf{G}\,\varphi$$

- No prenex normal form
- Branching-time semantics:
  - evaluation over transition systems/computation trees rather than sets of traces
  - quantifiers range over paths in the transition system, starting in the latest state
- Time progresses synchronously in all branches when evaluating temporal operators
- Strict generalization of both HyperLTL and CTL$^*$

# HyperCTL$^*$ [Clarkson et al. 2014]

## Example

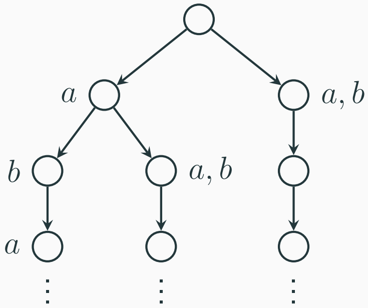
$$\varphi = \forall\pi.\ \mathsf{G}(a_\pi \to \forall\pi'.\ \mathsf{G}(b_\pi \leftrightarrow b_{\pi'}))$$

# HyperCTL$^*$ [Clarkson et al. 2014]

## Example

$$\varphi = \forall \pi.\ \mathsf{G}(a_\pi \rightarrow \forall \pi'.\ \mathsf{G}(b_\pi \leftrightarrow b_{\pi'}))$$

"All paths starting from an $a$-labeled state coincide on $b$"

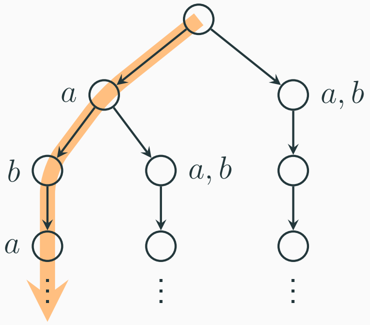


# HyperCTL* [Clarkson et al. 2014]

## Example

$$\varphi = \boxed{\forall \pi.}\ \mathsf{G}(a_\pi \rightarrow \forall \pi'.\ \mathsf{G}(b_\pi \leftrightarrow b_{\pi'}))$$

"All paths starting from an $a$-labeled state coincide on $b$"

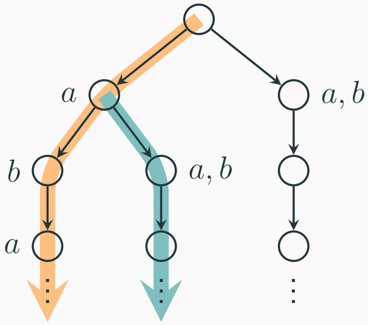


# HyperCTL* [Clarkson et al. 2014]

## Example

$$\varphi = \boxed{\forall \pi.}\ \mathsf{G}(a_\pi \rightarrow \boxed{\forall \pi'.}\ \mathsf{G}(b_\pi \leftrightarrow b_{\pi'}))$$

"All paths starting from an $a$-labeled state coincide on $b$"



# Undecidability

HyperLTL and HyperCTL* **model-checking** problems are decidable . . .

. . . but their satisfiability problems are undecidable.

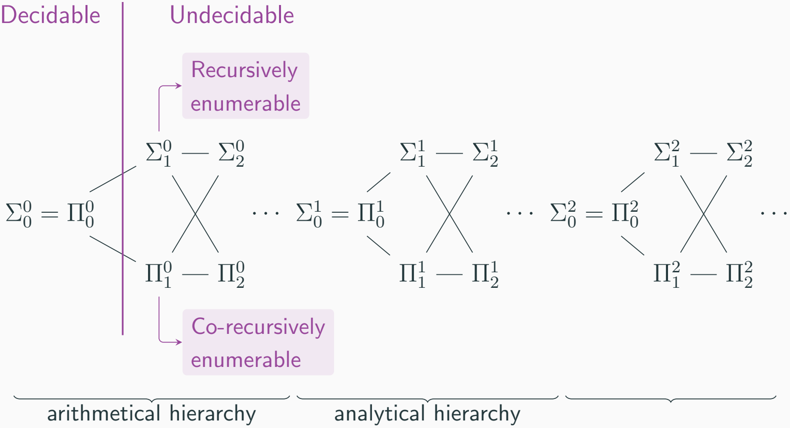
# Undecidability

HyperLTL and HyperCTL* **model-checking** problems are decidable . . .

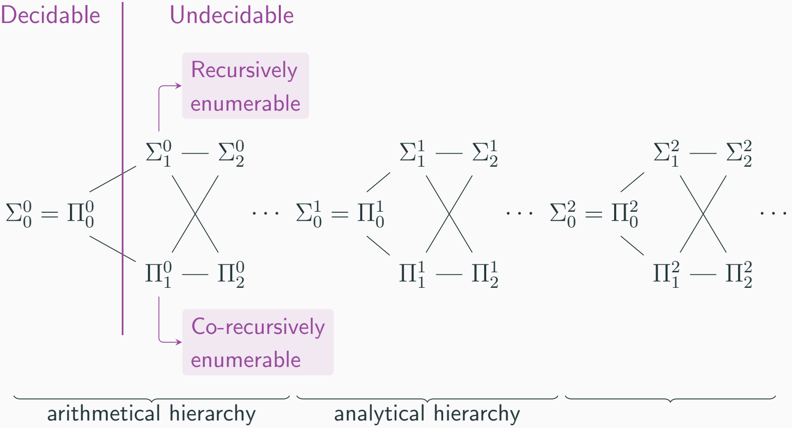
. . . but their satisfiability problems are undecidable.

**How** undecidable is HyperLTL or HyperCTL* satisfiability?

# Levels of Undecidability

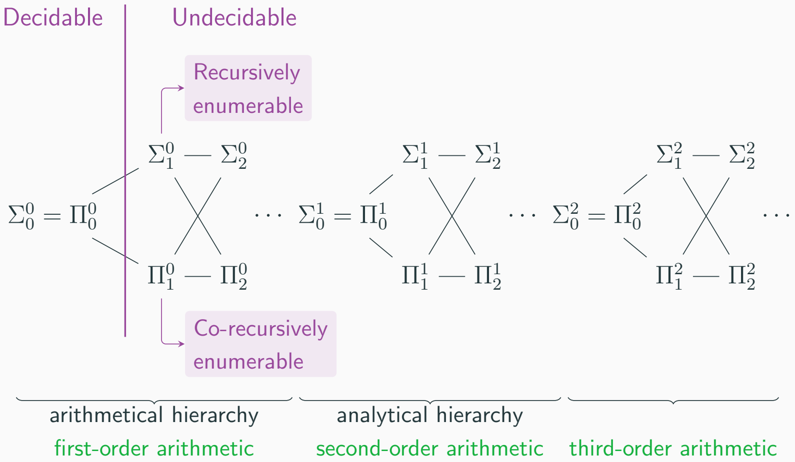


# Levels of Undecidability




$\Sigma_i^j$: definable by a formula of the form $\underbrace{\exists^*\forall^*\exists^*\forall^*\cdots}_{\substack{\text{i blocks}\\ (j+1)\text{th order}\\ \text{variables}}}\ \underbrace{\varphi}_{\substack{j\text{-th order}\\ \text{arithmetic}}}$

# Levels of Undecidability




$\Sigma^j_i$: definable by a formula of the form $\underbrace{\exists^*\forall^*\exists^*\forall^*\cdots}_{\substack{\text{i blocks}\\ (j+1)\text{th order}\\ \text{variables}}} \underbrace{\varphi}_{\substack{j\text{-th order}\\ \text{arithmetic}}}$

# What Was Known

# What Was Known

- HyperLTL satisfiability is $\Sigma_1^0$-hard. [Finkbeiner, Hahn 2016]

# What Was Known

- HyperLTL satisfiability is $\Sigma_1^0$-hard. [Finkbeiner, Hahn 2016]
- HyperLTL satisfiability **restricted to finite sets of traces** is $\Sigma_1^0$-complete (complete for the class of recursively enumerable problems).
  [Finkbeiner, Hahn, Hans 2018], [Mascle, Zimmermann, 2020]

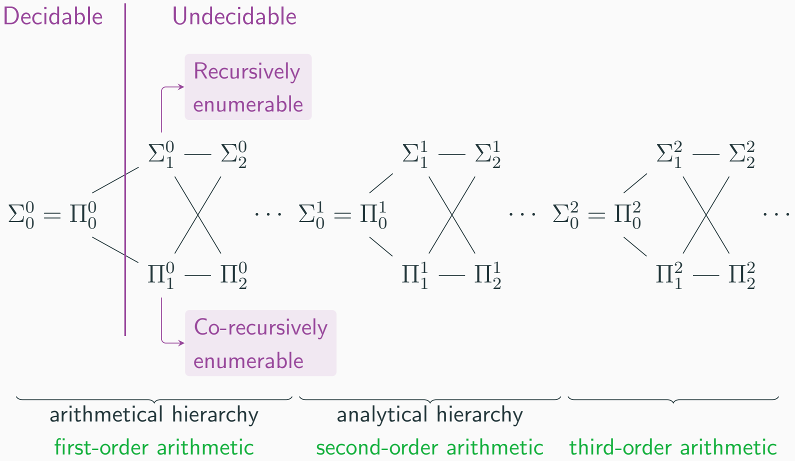
# What Was Known

- HyperLTL satisfiability is $\Sigma_1^0$-hard. [Finkbeiner, Hahn 2016]
- HyperLTL satisfiability **restricted to finite sets of traces** is $\Sigma_1^0$-complete (complete for the class of recursively enumerable problems).
  [Finkbeiner, Hahn, Hans 2018], [Mascle, Zimmermann, 2020]
- HyperCTL$^*$ satisfiability is $\Sigma_1^1$-hard.
  [Clarkson, Finkbeiner, Koleini, Micinski, Rabe, Sánchez 2014]

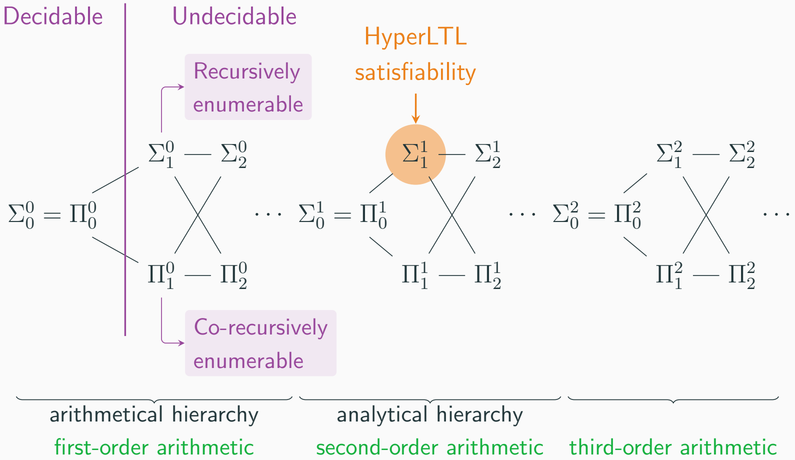
# What Was Known

- HyperLTL satisfiability is $\Sigma_1^0$-hard. [Finkbeiner, Hahn 2016]
- HyperLTL satisfiability **restricted to finite sets of traces** is $\Sigma_1^0$-complete (complete for the class of recursively enumerable problems).
  [Finkbeiner, Hahn, Hans 2018], [Mascle, Zimmermann, 2020]
- HyperCTL$^*$ satisfiability is $\Sigma_1^1$-hard.
  [Clarkson, Finkbeiner, Koleini, Micinski, Rabe, Sánchez 2014]
- **Finite-state** HyperCTL$^*$ satisfiability is $\Sigma_1^0$-complete.
  [Clarkson, Finkbeiner, Koleini, Micinski, Rabe, Sánchez 2014]

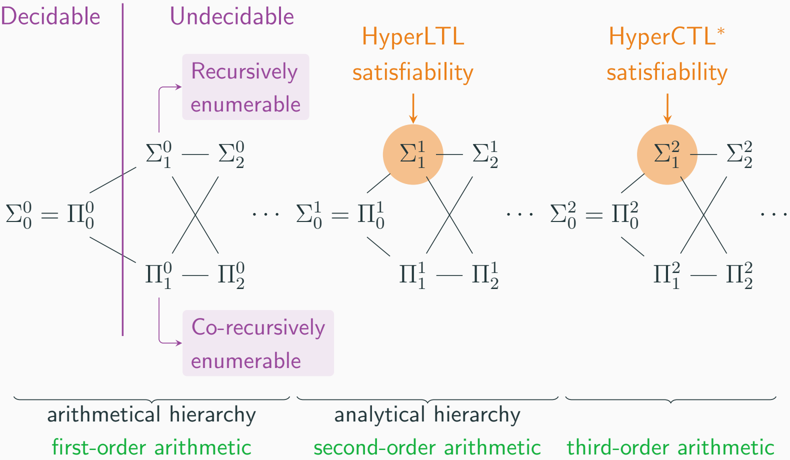
# Completeness Results



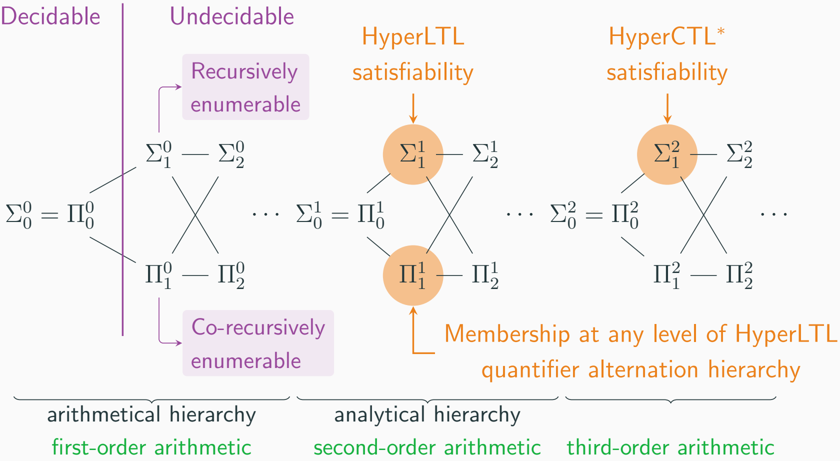
# Completeness Results



# Completeness Results



# Completeness Results



# HyperLTL Satisfiability – Membership

**Lemma**

HyperLTL satisfiability is in $\Sigma_1^1$

# HyperLTL Satisfiability – Membership

**Lemma**

HyperLTL satisfiability is in $\Sigma^1_1$, i.e., there is a formula $\Phi(x) = \underbrace{\exists x_1, \ldots, \exists x_n.}_{\text{second-order variables}} \underbrace{\Phi_0(x, x_1, \ldots, x_n)}_{\text{first-order arithmetic}}$ such that $\psi \in$ HyperLTL is satisfiable iff $\Phi(\lceil \psi \rceil)$ holds.

encoding of $\psi$ as a natural number

# HyperLTL Satisfiability – Membership

**Lemma**

HyperLTL satisfiability is in $\Sigma_1^1$, i.e., there is a formula $\Phi(x) = \underbrace{\exists x_1, \ldots, \exists x_n.}_{\text{second-order variables}} \underbrace{\Phi_0(x, x_1, \ldots, x_n)}_{\text{first-order arithmetic}}$ such that $\psi \in$ HyperLTL is satisfiable iff $\Phi(\lceil \psi \rceil)$ holds.

encoding of $\psi$ as a natural number

**Some intuitions:**

- Minimal size of a model: every satisfiable HyperLTL formula has a countable model [Finkbeiner, Zimmermann '17]

# HyperLTL Satisfiability – Membership

**Lemma**

HyperLTL satisfiability is in $\Sigma^1_1$, i.e., there is a formula $\Phi(x) = \underbrace{\exists x_1, \ldots, \exists x_n.}_{\text{second-order variables}} \underbrace{\Phi_0(x, x_1, \ldots, x_n)}_{\text{first-order arithmetic}}$ such that $\psi \in$ HyperLTL is satisfiable iff $\Phi(\lceil \psi \rceil)$ holds.

encoding of $\psi$ as a natural number

**Some intuitions:**

- Minimal size of a model: every satisfiable HyperLTL formula has a countable model [Finkbeiner, Zimmermann '17]
- Countable models can be seen as functions from $\mathbb{N} \times \mathbb{N} \to \mathbb{N}$ mapping a pair (trace, position) to a label

# HyperLTL Satisfiability – Membership

**Lemma**

HyperLTL satisfiability is in $\Sigma_1^1$, i.e., there is a formula $\Phi(x) = \underbrace{\exists x_1, \ldots, \exists x_n.}_{\text{second-order variables}} \underbrace{\Phi_0(x, x_1, \ldots, x_n)}_{\text{first-order arithmetic}}$ such that $\psi \in$ HyperLTL is satisfiable iff $\Phi(\lceil \psi \rceil)$ holds.

encoding of $\psi$ as a natural number

**Some intuitions:**

- Minimal size of a model: every satisfiable HyperLTL formula has a countable model [Finkbeiner, Zimmermann '17]
- Countable models can be seen as functions from $\mathbb{N} \times \mathbb{N} \to \mathbb{N}$ mapping a pair (trace, position) to a label
- Existential second-order quantification is used to encode the existence of a model

# HyperLTL Satisfiability – Hardness

**Lemma**

HyperLTL satisfiability is $\Sigma_1^1$-hard.

**Proof idea:** by reduction from the recurring tiling problem: given a set of tiles, is there a tiling of $\mathbb{N} \times \mathbb{N}$ such that a specific tile occurs infinitely often on the $\mathbb{N} \times 0$ border?

# HyperLTL Satisfiability – Hardness

**Lemma**

HyperLTL satisfiability is $\Sigma_1^1$-hard.

**Proof idea:** by reduction from the recurring tiling problem: given a set of tiles, is there a tiling of $\mathbb{N} \times \mathbb{N}$ such that a specific tile occurs infinitely often on the $\mathbb{N} \times 0$ border?

- Each tile is encoded by an atomic proposition

# HyperLTL Satisfiability – Hardness

**Lemma**

HyperLTL satisfiability is $\Sigma_1^1$-hard.

**Proof idea:** by reduction from the recurring tiling problem: given a set of tiles, is there a tiling of $\mathbb{N} \times \mathbb{N}$ such that a specific tile occurs infinitely often on the $\mathbb{N} \times 0$ border?

- Each tile is encoded by an atomic proposition
- Each row is encoded by a trace

# HyperLTL Satisfiability – Hardness

**Lemma**

HyperLTL satisfiability is $\Sigma_1^1$-hard.

**Proof idea:** by reduction from the recurring tiling problem: given a set of tiles, is there a tiling of $\mathbb{N} \times \mathbb{N}$ such that a specific tile occurs infinitely often on the $\mathbb{N} \times 0$ border?

- Each tile is encoded by an atomic proposition
- Each row is encoded by a trace
- Rows/traces are ordered vertically using a special atomic proposition $y$ true exactly once in each trace: $y$ is true at time $i$ on the trace representing row $\mathbb{N} \times i$

# HyperLTL Satisfiability – Hardness

**Proof idea (continued):**

Define $\varphi \in$ HyperLTL expressing that:

# HyperLTL Satisfiability – Hardness

**Proof idea (continued):**

Define $\varphi \in$ HyperLTL expressing that:

1. there is exactly one tile at every position in every trace

# HyperLTL Satisfiability – Hardness

**Proof idea (continued):**

Define $\varphi \in$ HyperLTL expressing that:

1. there is exactly one tile at every position in every trace
2. $y$ is true exactly once on each trace

$$\forall \pi.\, (\neg y_\pi \,\mathsf{U}\, (y_\pi \wedge \mathsf{X}\,\mathsf{G}\, \neg y_\pi))$$

# HyperLTL Satisfiability – Hardness

**Proof idea (continued):**

Define $\varphi \in$ HyperLTL expressing that:

1. there is exactly one tile at every position in every trace
2. $y$ is true exactly once on each trace
3. for every $i$, there is a trace with $y$ at position $i$

$$(\exists \pi.\, y_\pi) \wedge (\forall \pi.\, \exists \pi'.\, \mathsf{F}(y_\pi \wedge X y'_\pi))$$

# HyperLTL Satisfiability – Hardness

**Proof idea (continued):**

Define $\varphi \in$ HyperLTL expressing that:

1. there is exactly one tile at every position in every trace
2. $y$ is true exactly once on each trace
3. for every $i$, there is a trace with $y$ at position $i$
4. two traces with the same position for $y$ are identical

$$\forall \pi, \pi'.\ \mathsf{F}(y_\pi \wedge y_{\pi'}) \rightarrow \mathsf{G}\left(\bigwedge_\tau \tau_\pi \leftrightarrow \tau_{\pi'}\right)$$

# HyperLTL Satisfiability – Hardness

**Proof idea (continued):**

Define $\varphi \in$ HyperLTL expressing that:

1. there is exactly one tile at every position in every trace
2. $y$ is true exactly once on each trace
3. for every $i$, there is a trace with $y$ at position $i$
4. two traces with the same position for $y$ are identical
5. tiles match horizontally

$$\forall \pi.\ \mathsf{G} \left( \bigvee_{(\tau,\tau') \in H} \tau_\pi \land \mathsf{X}\, \tau_{\pi'} \right)$$

# HyperLTL Satisfiability – Hardness

**Proof idea (continued):**

Define $\varphi \in$ HyperLTL expressing that:

1. there is exactly one tile at every position in every trace
2. $y$ is true exactly once on each trace
3. for every $i$, there is a trace with $y$ at position $i$
4. two traces with the same position for $y$ are identical
5. tiles match horizontally
6. tiles match vertically

$$\forall \pi, \pi'.\ \mathsf{F}(y_\pi \wedge \mathsf{X}\, y_{\pi'}) \rightarrow \mathsf{G}\left( \bigvee_{(\tau,\tau') \in V} \tau_\pi \wedge \tau_{\pi'} \right)$$

# HyperLTL Satisfiability – Hardness

**Proof idea (continued):**

Define $\varphi \in$ HyperLTL expressing that:

1. there is exactly one tile at every position in every trace
2. $y$ is true exactly once on each trace
3. for every $i$, there is a trace with $y$ at position $i$
4. two traces with the same position for $y$ are identical
5. tiles match horizontally
6. tiles match vertically
7. the specified tile occurs infinitely often on the trace where $y$ is true at $0$

$$\exists \pi.\, (y_\pi \wedge \mathsf{G}\,\mathsf{F}(\tau_0)_\pi)$$

# HyperLTL vs. HyperCTL$^*$

**Theorem**

HyperLTL satisfiability is $\Sigma_1^1$-complete.

# HyperLTL vs. HyperCTL$^*$

**Theorem**

HyperLTL satisfiability is $\Sigma_1^1$-complete.

**Theorem**

HyperCTL$^*$ satisfiability is $\Sigma_1^2$-complete.

# HyperLTL vs. HyperCTL$^*$

**Theorem**

HyperLTL satisfiability is $\Sigma_1^1$-complete.

**Theorem**

HyperCTL$^*$ satisfiability is $\Sigma_1^2$-complete.

existential second-order $\rightarrow$ existential third-order arithmetic

# HyperLTL vs. HyperCTL$^*$

**Theorem**

HyperLTL satisfiability is $\Sigma_1^1$-complete.

**Theorem**

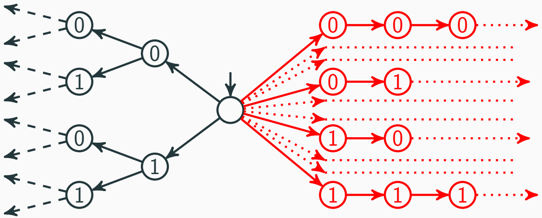
HyperCTL$^*$ satisfiability is $\Sigma_1^2$-complete.

existential second-order $\rightarrow$ existential third-order arithmetic

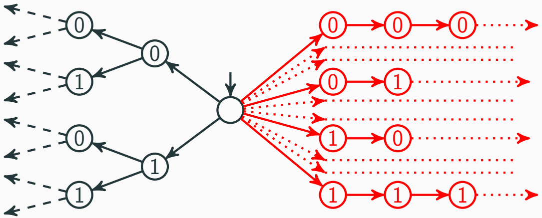
**Why?**

- Every satisfiable HyperLTL formula has a countable model
- Some formulas of HyperCTL$^*$ require models of cardinality $\mathfrak{c} = |2^{\mathbb{N}}|$ (and this bound is optimal)

# Models of Cardinality at Least $\mathfrak{c}$



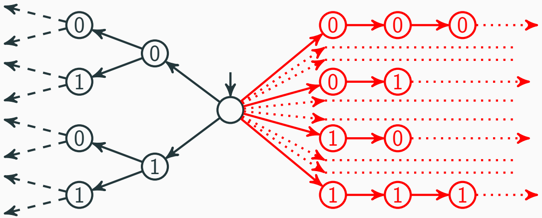
# Models of Cardinality at Least $\mathfrak{c}$




The following can be expressed in HyperCTL$^*$:

- Every state is labeled red or black, and 1 or 0

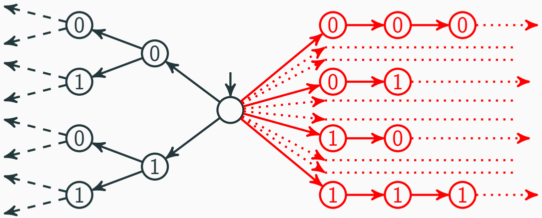
# Models of Cardinality at Least $\mathfrak{c}$




The following can be expressed in HyperCTL$^*$:

- Every state is labeled red or black, and 1 or 0
- Every black state has a (black) 0- and 1-successor

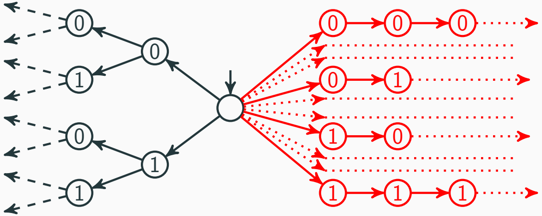
# Models of Cardinality at Least $\mathfrak{c}$




The following can be expressed in HyperCTL$^*$:

- Every state is labeled red or black, and 1 or 0
- Every black state has a (black) 0- and 1-successor
- Every black path has a copy in the red part
  $\forall\pi.\left(\mathsf{X}\,\mathtt{black}_\pi \rightarrow \exists\pi'.\,\mathsf{X}(\mathtt{red}_{\pi'} \wedge \mathsf{G}(0_\pi \leftrightarrow 0_{\pi'}))\right)$

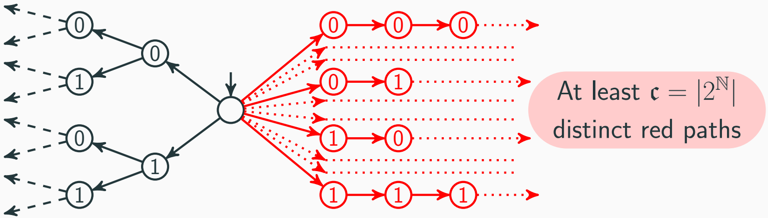
# Models of Cardinality at Least $\mathfrak{c}$




The following can be expressed in HyperCTL$^*$:

- Every state is labeled red or black, and 1 or 0
- Every black state has a (black) 0- and 1-successor
- Every black path has a copy in the red part
  $\forall\pi.\ \big(\mathsf{X}\,\mathtt{black}_\pi \rightarrow \exists\pi'.\ \mathsf{X}(\mathtt{red}_{\pi'} \wedge \mathsf{G}(0_\pi \leftrightarrow 0_{\pi'}))\big)$
- Two red paths starting in the same state have the same sequence of labels (red paths are initially disjoint)
  $\forall\pi.\ \mathsf{G}\,\big(\mathtt{red}_\pi \rightarrow \forall\pi'.\ \mathsf{G}(0_\pi \leftrightarrow 0_{\pi'})\big)$

# Models of Cardinality at Least $\mathfrak{c}$




The following can be expressed in HyperCTL$^*$:

- Every state is labeled red or black, and 1 or 0
- Every black state has a (black) 0- and 1-successor
- Every black path has a copy in the red part
  $\forall \pi. \left( \mathsf{X}\, \texttt{black}_\pi \rightarrow \exists \pi'. \, \mathsf{X}(\texttt{red}_{\pi'} \wedge \mathsf{G}(0_\pi \leftrightarrow 0_{\pi'}))\right)$
- Two red paths starting in the same state have the same sequence of labels (red paths are initially disjoint)
  $\forall \pi. \, \mathsf{G} \left( \texttt{red}_\pi \rightarrow \forall \pi'. \, \mathsf{G}(0_\pi \leftrightarrow 0_{\pi'})\right)$

# Matching upper bound

Every satisfiable HyperCTL* formula $\varphi$ has a model of cardinality at most $\mathfrak{c}$.

# Matching upper bound

Every satisfiable HyperCTL* formula $\varphi$ has a model of cardinality at most $\mathfrak{c}$.

- Start with an arbitrary model $\mathcal{T}$ of $\varphi$, and Skolem functions witnessing satisfaction.

# Matching upper bound

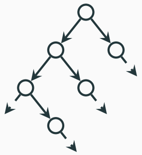
Every satisfiable HyperCTL* formula $\varphi$ has a model of cardinality at most $\mathfrak{c}$.

- Start with an arbitrary model $\mathcal{T}$ of $\varphi$, and Skolem functions witnessing satisfaction.
- Saturation procedure:

$$\mathcal{T}_0 = \{s_0 \to s_1 \to \cdots\}$$

$$\mathcal{T}_{\alpha+1} = \mathcal{T}_\alpha \cup \bigcup_{\substack{\bar{x} \text{ inputs from } \mathcal{T}_\alpha \\ f \text{ a skolem function}}} f(\bar{x})$$

$$\mathcal{T}_\alpha = \bigcup_{\alpha' < \alpha} \mathcal{T}_{\alpha'} \quad \text{for limit ordinals}$$



# Matching upper bound

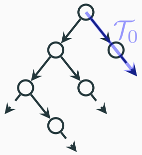
Every satisfiable HyperCTL* formula $\varphi$ has a model of cardinality at most $\mathfrak{c}$.

- Start with an arbitrary model $\mathcal{T}$ of $\varphi$, and Skolem functions witnessing satisfaction.
- Saturation procedure:

$$\mathcal{T}_0 = \{s_0 \to s_1 \to \cdots\}$$

$$\mathcal{T}_{\alpha+1} = \mathcal{T}_\alpha \cup \bigcup_{\substack{\bar{x} \text{ inputs from } \mathcal{T}_\alpha \\ f \text{ a skolem function}}} f(\bar{x})$$

$$\mathcal{T}_\alpha = \bigcup_{\alpha' < \alpha} \mathcal{T}_{\alpha'} \quad \text{for limit ordinals}$$



# Matching upper bound

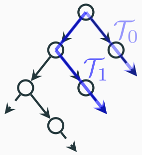
Every satisfiable HyperCTL* formula $\varphi$ has a model of cardinality at most $\mathfrak{c}$.

- Start with an arbitrary model $\mathcal{T}$ of $\varphi$, and Skolem functions witnessing satisfaction.
- Saturation procedure:

$$\mathcal{T}_0 = \{s_0 \to s_1 \to \cdots\}$$

$$\mathcal{T}_{\alpha+1} = \mathcal{T}_\alpha \cup \bigcup_{\substack{\bar{x} \text{ inputs from } \mathcal{T}_\alpha \\ f \text{ a skolem function}}} f(\bar{x})$$

$$\mathcal{T}_\alpha = \bigcup_{\alpha' < \alpha} \mathcal{T}_{\alpha'} \quad \text{for limit ordinals}$$



# Matching upper bound

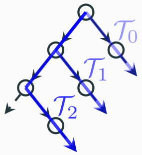
Every satisfiable HyperCTL* formula $\varphi$ has a model of cardinality at most $\mathfrak{c}$.

- Start with an arbitrary model $\mathcal{T}$ of $\varphi$, and Skolem functions witnessing satisfaction.
- Saturation procedure:

$$\mathcal{T}_0 = \{s_0 \to s_1 \to \cdots\}$$

$$\mathcal{T}_{\alpha+1} = \mathcal{T}_\alpha \cup \bigcup_{\substack{\bar{x} \text{ inputs from } \mathcal{T}_\alpha \\ f \text{ a skolem function}}} f(\bar{x})$$

$$\mathcal{T}_\alpha = \bigcup_{\alpha' < \alpha} \mathcal{T}_{\alpha'} \quad \text{for limit ordinals}$$



# Matching upper bound

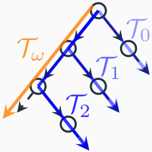
Every satisfiable HyperCTL* formula $\varphi$ has a model of cardinality at most $\mathfrak{c}$.

- Start with an arbitrary model $\mathcal{T}$ of $\varphi$, and Skolem functions witnessing satisfaction.
- Saturation procedure:

$$\mathcal{T}_0 = \{s_0 \to s_1 \to \cdots\}$$

$$\mathcal{T}_{\alpha+1} = \mathcal{T}_\alpha \cup \bigcup_{\substack{\bar{x} \text{ inputs from } \mathcal{T}_\alpha \\ f \text{ a skolem function}}} f(\bar{x})$$

$$\mathcal{T}_\alpha = \bigcup_{\alpha' < \alpha} \mathcal{T}_{\alpha'} \quad \text{for limit ordinals}$$



# Matching upper bound

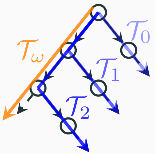
Every satisfiable HyperCTL* formula $\varphi$ has a model of cardinality at most $\mathfrak{c}$.

- Start with an arbitrary model $\mathcal{T}$ of $\varphi$, and Skolem functions witnessing satisfaction.
- Saturation procedure:

$$\mathcal{T}_0 = \{s_0 \to s_1 \to \cdots\}$$

$$\mathcal{T}_{\alpha+1} = \mathcal{T}_\alpha \cup \bigcup_{\substack{\bar{x} \text{ inputs from } \mathcal{T}_\alpha \\ f \text{ a skolem function}}} f(\bar{x})$$

$$\mathcal{T}_\alpha = \bigcup_{\alpha' < \alpha} \mathcal{T}_{\alpha'} \quad \text{for limit ordinals}$$




- Fixpoint at the first uncountable ordinal: $\mathcal{T}_{\omega_1} = \mathcal{T}_{\omega_1+1}$.

# Matching upper bound

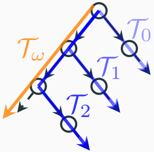
Every satisfiable HyperCTL* formula $\varphi$ has a model of cardinality at most $\mathfrak{c}$.

- Start with an arbitrary model $\mathcal{T}$ of $\varphi$, and Skolem functions witnessing satisfaction.
- Saturation procedure:

$$\mathcal{T}_0 = \{s_0 \to s_1 \to \cdots\}$$

$$\mathcal{T}_{\alpha+1} = \mathcal{T}_\alpha \cup \bigcup_{\substack{\bar{x} \text{ inputs from } \mathcal{T}_\alpha \\ f \text{ a skolem function}}} f(\bar{x})$$

$$\mathcal{T}_\alpha = \bigcup_{\alpha' < \alpha} \mathcal{T}_{\alpha'} \quad \text{for limit ordinals}$$




- Fixpoint at the first uncountable ordinal: $\mathcal{T}_{\omega_1} = \mathcal{T}_{\omega_1+1}$.
- $\mathcal{T}_{\omega_1}$ contains at most $\mathfrak{c}$ vertices, and $\mathcal{T}_{\omega_1} \models \varphi$.

# Size of Minimal Models

**Theorem**

- Every satisfiable HyperCTL$^*$ formula has a model of cardinality at most $\mathfrak{c} = |2^{\mathbb{N}}|$.
- There is a satisfiable HyperCTL$^*$ formula that does not have any model of cardinality less than $\mathfrak{c}$.

# HyperCTL$^*$ Satisfiability

**Theorem**

HyperCTL$^*$ satisfiability is $\Sigma_1^2$-complete.

$$\underbrace{\exists x_1, \ldots, \exists x_n.}_{\text{third-order variables}} \underbrace{\Phi_0(x, x_1, \ldots, x_n)}_{\text{second-order arithmetic}}$$

# HyperCTL* Satisfiability

**Theorem**

HyperCTL* satisfiability is $\Sigma_1^2$-complete.

$$\underbrace{\exists x_1, \dots, \exists x_n.}_{\text{third-order variables}} \underbrace{\Phi_0(x, x_1, \dots, x_n)}_{\text{second-order arithmetic}}$$

## Upper bound

- Every satisfiable HyperCTL* formula has a model of cardinality at most $\mathfrak{c}$
  - set of states $= 2^{\mathbb{N}}$
  - transitions $=$ subset of $2^{\mathbb{N}} \times 2^{\mathbb{N}}$
  - labeling function $2^{\mathbb{N}} \to \mathbb{N}$

# HyperCTL$^*$ Satisfiability

**Theorem**

HyperCTL$^*$ satisfiability is $\Sigma_1^2$-complete.

$$\underbrace{\exists x_1, \ldots, \exists x_n.}_{\text{third-order variables}} \underbrace{\Phi_0(x, x_1, \ldots, x_n)}_{\text{second-order arithmetic}}$$

## Upper bound

- Every satisfiable HyperCTL$^*$ formula has a model of cardinality at most $\mathfrak{c}$
  - set of states $= 2^{\mathbb{N}}$
  - transitions $=$ subset of $2^{\mathbb{N}} \times 2^{\mathbb{N}}$
  - labeling function $2^{\mathbb{N}} \to \mathbb{N}$
- Use third-order quantifiers to express the existence of a model

# HyperCTL$^*$ Satisfiability

**Theorem**

HyperCTL$^*$ satisfiability is $\Sigma_1^2$-complete.

$$\underbrace{\exists x_1, \ldots, \exists x_n.}_{\text{third-order variables}} \underbrace{\Phi_0(x, x_1, \ldots, x_n)}_{\text{second-order arithmetic}}$$

**Lower bound**

From an existential third-order arithmetic formula $\Phi(x)$ and $n$, construct $\psi$ such that $\mathbb{N} \models \Phi(n)$ iff $\psi$ is satisfiable:

# HyperCTL$^*$ Satisfiability

**Theorem**

HyperCTL$^*$ satisfiability is $\Sigma^2_1$-complete.

$$\underbrace{\exists x_1, \ldots, \exists x_n.}_{\text{third-order variables}} \underbrace{\Phi_0(x, x_1, \ldots, x_n)}_{\text{second-order arithmetic}}$$

## Lower bound

From an existential third-order arithmetic formula $\Phi(x)$ and $n$, construct $\psi$ such that $\mathbb{N} \models \Phi(n)$ iff $\psi$ is satisfiable:

- $n \in \mathbb{N} \rightsquigarrow$ path with $1$ in $n$-th position only;

# HyperCTL* Satisfiability

**Theorem**

HyperCTL* satisfiability is $\Sigma_1^2$-complete.

$$\underbrace{\exists x_1, \ldots, \exists x_n.}_{\text{third-order variables}} \underbrace{\Phi_0(x, x_1, \ldots, x_n)}_{\text{second-order arithmetic}}$$

## Lower bound

From an existential third-order arithmetic formula $\Phi(x)$ and $n$, construct $\psi$ such that $\mathbb{N} \models \Phi(n)$ iff $\psi$ is satisfiable:

- $n \in \mathbb{N} \rightsquigarrow$ path with $1$ in $n$-th position only;
- $A \subseteq \mathbb{N} \rightsquigarrow$ path labeled by characteristic sequence of $A$;

  First and second-order quantifiers $\rightsquigarrow$ path quantifiers

# HyperCTL* Satisfiability

**Theorem**

HyperCTL* satisfiability is $\Sigma^2_1$-complete.

$$\underbrace{\exists x_1, \ldots, \exists x_n.}_{\text{third-order variables}} \underbrace{\Phi_0(x, x_1, \ldots, x_n)}_{\text{second-order arithmetic}}$$

## Lower bound

From an existential third-order arithmetic formula $\Phi(x)$ and $n$, construct $\psi$ such that $\mathbb{N} \models \Phi(n)$ iff $\psi$ is satisfiable:

- $n \in \mathbb{N} \rightsquigarrow$ path with $1$ in $n$-th position only;
- $A \subseteq \mathbb{N} \rightsquigarrow$ path labeled by characteristic sequence of $A$;
  First and second-order quantifiers $\rightsquigarrow$ path quantifiers
- One atomic proposition $p_i$ for each third-order $x_i$.
  Existential third-order quantifiers $\rightsquigarrow$ satisfiability

# Conclusion

# Conclusion

- HyperLTL satisfiability problem is $\Sigma_1^1$-complete, thus highly undecidable

# Conclusion

- HyperLTL satisfiability problem is $\Sigma_1^1$-complete, thus highly undecidable
- HyperCTL$^*$ satisfiability problem is $\Sigma_1^2$-complete, which is infinitely higher than $\Sigma_1^1$ in the hierarchy

# Conclusion

- HyperLTL satisfiability problem is $\Sigma_1^1$-complete, thus highly undecidable
- HyperCTL$^*$ satisfiability problem is $\Sigma_1^2$-complete, which is infinitely higher than $\Sigma_1^1$ in the hierarchy
- First (and optimal) bound on the minimal size of models for HyperCTL$^*$:
  - every satisfiable formula has a model with at most $\mathfrak{c}$ many states
  - there is satisfiable formula that requires $\mathfrak{c}$ many states

# Conclusion

Other results:

# Conclusion

Other results:

- HyperLTL satisfiability is still $\Sigma_1^1$-complete when restricted to ultimately periodic traces.

# Conclusion

Other results:

- HyperLTL satisfiability is still $\Sigma_1^1$-complete when restricted to ultimately periodic traces.
- HyperCTL$^*$ satisfiability restricted to countable or finitely branching transition systems is equivalent to the problem of evaluating a second-order arithmetic formula.

# Conclusion

Other results:

- HyperLTL satisfiability is still $\Sigma_1^1$-complete when restricted to ultimately periodic traces.
- HyperCTL$^*$ satisfiability restricted to countable or finitely branching transition systems is equivalent to the problem of evaluating a second-order arithmetic formula.
- deciding if a HyperLTL formula is equivalent to one with $n$ quantifier alternations is exactly as hard unsatisfiability, i.e. $\Pi_1^1$-complete.

**Thank you!**