Logics for Hyperproperties

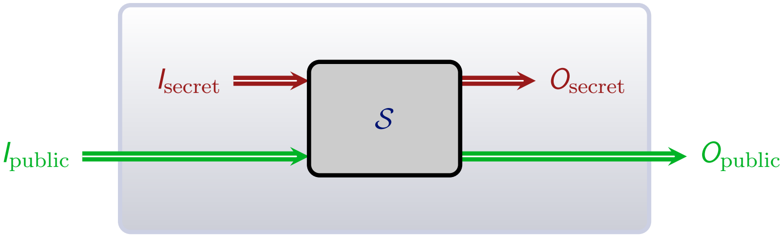
Martin Zimmermann

Saarland University

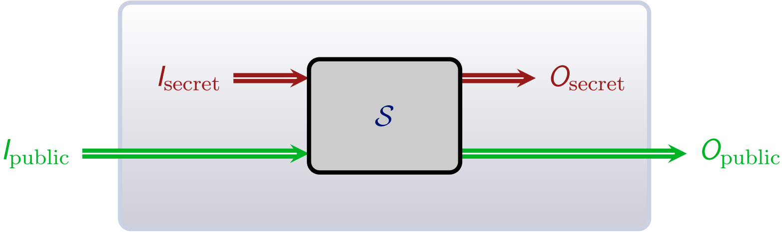
May, 19th 2017

Centre Fédéré en Vérification, Brussels, Belgium

Hyperproperties



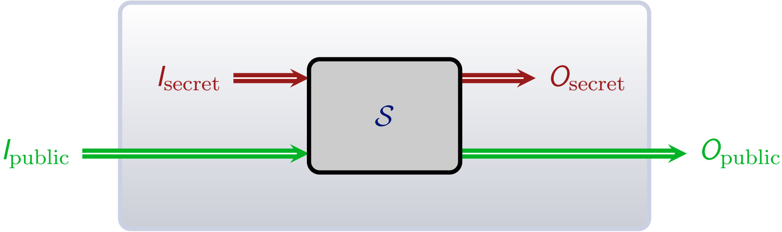
Hyperproperties



- The system $\mathcal{S}$ is input-deterministic: for all traces t, t' of $\mathcal{S}$

$$t =_I t' \quad \text{implies} \quad t =_O t'$$

Hyperproperties



- The system $\mathcal{S}$ is input-deterministic: for all traces t, t' of $\mathcal{S}$

$$t =_I t' \quad \text{implies} \quad t =_O t'$$

- Noninterference: for all traces t, t' of $\mathcal{S}$

$$t =_{I_{\text{public}}} t' \quad \text{implies} \quad t =_{O_{\text{public}}} t'$$

Hyperproperties

- Both properties are not trace properties, i.e., sets $T \subseteq \mathrm{Traces}(\mathcal{S})$ of traces, but
- hyperproperties, i.e., sets $H \subseteq 2^{\mathrm{Traces}(\mathcal{S})}$ of sets of traces.
- A system $\mathcal{S}$ satisfies a hyperproperty H, if $\mathrm{Traces}(\mathcal{S}) \in H$.

Example: Noninterference as trace property:

$$\{T \subseteq \mathrm{Traces}(\mathcal{S}) \mid \forall t, t' \in T : t =_{I_{\mathrm{public}}} t' \Rightarrow t =_{O_{\mathrm{public}}} t'\}$$

Hyperproperties

- Both properties are not trace properties, i.e., sets $T \subseteq \mathrm{Traces}(\mathcal{S})$ of traces, but
- hyperproperties, i.e., sets $H \subseteq 2^{\mathrm{Traces}(\mathcal{S})}$ of sets of traces.
- A system $\mathcal{S}$ satisfies a hyperproperty H, if $\mathrm{Traces}(\mathcal{S}) \in H$.

Example: Noninterference as trace property:

$$\{T \subseteq \mathrm{Traces}(\mathcal{S}) \mid \forall t, t' \in T : t =_{I_{\mathrm{public}}} t' \Rightarrow t =_{O_{\mathrm{public}}} t'\}$$

Specification languages for hyperproperties

HyperLTL: Extend LTL by trace quantifiers.

HyperCTL*: Extend CTL* by trace quantifiers.

Outline

1. HyperLTL
2. The Models Of HyperLTL
3. HyperLTL Satisfiability
4. HyperLTL Model-checking
5. The First-order Logic of Hyperproperties
6. Conclusion

Outline

1. **HyperLTL**
2. The Models Of HyperLTL
3. HyperLTL Satisfiability
4. HyperLTL Model-checking
5. The First-order Logic of Hyperproperties
6. Conclusion

LTL in One Slide

Syntax

$$\varphi ::= a \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \mathbf{X}\,\varphi \mid \varphi\,\mathbf{U}\,\varphi$$

where $a \in \mathrm{AP}$ (atomic propositions).

LTL in One Slide

Syntax

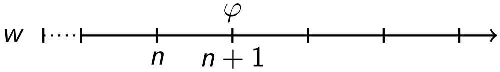
$$\varphi ::= a \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \mathbf{X}\,\varphi \mid \varphi\,\mathbf{U}\,\varphi$$

where $a \in \mathrm{AP}$ (atomic propositions).

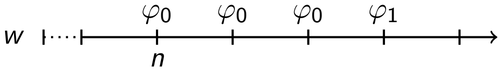
Semantics

$w, n \models \varphi$ for a trace $w \in (2^{\mathrm{AP}})^\omega$ and a position $n \in \mathbb{N}$:

- $w, n \models \mathbf{X}\,\varphi$:



- $w, n \models \varphi_0\,\mathbf{U}\,\varphi_1$:



LTL in One Slide

Syntax

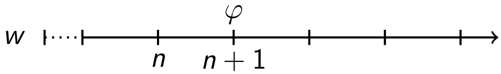
$$\varphi ::= a \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \mathbf{X}\,\varphi \mid \varphi\,\mathbf{U}\,\varphi$$

where $a \in \mathrm{AP}$ (atomic propositions).

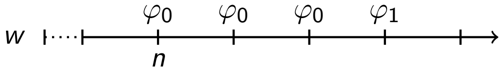
Semantics

$w, n \models \varphi$ for a trace $w \in (2^{\mathrm{AP}})^\omega$ and a position $n \in \mathbb{N}$:

- $w, n \models \mathbf{X}\,\varphi$:



- $w, n \models \varphi_0\,\mathbf{U}\,\varphi_1$:



Syntactic Sugar

- $\mathbf{F}\,\psi = \mathbf{true}\,\mathbf{U}\,\psi$
- $\mathbf{G}\,\psi = \neg\mathbf{F}\,\neg\psi$

HyperLTL

HyperLTL = LTL + trace quantification

$$\varphi ::= \exists\pi.\ \varphi \mid \forall\pi.\ \varphi \mid \psi$$
$$\psi ::= a_\pi \mid \neg\psi \mid \psi \vee \psi \mid \mathbf{X}\,\psi \mid \psi\,\mathbf{U}\,\psi$$

where $a \in \mathrm{AP}$ (atomic propositions) and $\pi \in \mathcal{V}$ (trace variables).

HyperLTL

HyperLTL = LTL + trace quantification

$$\varphi ::= \exists \pi.\ \varphi \mid \forall \pi.\ \varphi \mid \psi$$
$$\psi ::= a_\pi \mid \neg \psi \mid \psi \vee \psi \mid \mathbf{X}\,\psi \mid \psi\,\mathbf{U}\,\psi$$

where $a \in \mathrm{AP}$ (atomic propositions) and $\pi \in \mathcal{V}$ (trace variables).

- Prenex normal form, but
- closed under boolean combinations.

Semantics

$$\varphi = \forall \pi. \forall \pi'. \mathbf{G}\, \mathtt{on}_\pi \leftrightarrow \mathtt{on}_{\pi'}$$

$T \subseteq (2^{\mathrm{AP}})^\omega$ is a model of φ iff

Semantics

$$\varphi = \forall \pi.\, \forall \pi'.\, \mathbf{G}\, \mathtt{on}_{\pi} \leftrightarrow \mathtt{on}_{\pi'}$$

$T \subseteq (2^{\mathrm{AP}})^{\omega}$ is a model of φ iff

$$\{\} \models \forall \pi.\, \forall \pi'.\, \mathbf{G}\, \mathtt{on}_{\pi} \leftrightarrow \mathtt{on}_{\pi'}$$

Semantics

$$\varphi = \forall \pi.\, \forall \pi'.\, \mathbf{G}\, \mathtt{on}_{\pi} \leftrightarrow \mathtt{on}_{\pi'}$$

$T \subseteq (2^{\mathrm{AP}})^{\omega}$ is a model of φ iff

$$\{\} \models \forall \pi.\, \forall \pi'.\, \mathbf{G}\, \mathtt{on}_{\pi} \leftrightarrow \mathtt{on}_{\pi'}$$

$$\{\pi \mapsto t\} \models \forall \pi'.\, \mathbf{G}\, \mathtt{on}_{\pi} \leftrightarrow \mathtt{on}_{\pi'} \qquad \text{for all } t \in T$$

Semantics

$$\varphi = \forall\pi.\,\forall\pi'.\,\mathbf{G}\,\mathtt{on}_\pi \leftrightarrow \mathtt{on}_{\pi'}$$

$T \subseteq (2^{\mathrm{AP}})^\omega$ is a model of φ iff

$$\{\} \models \forall\pi.\,\forall\pi'.\,\mathbf{G}\,\mathtt{on}_\pi \leftrightarrow \mathtt{on}_{\pi'}$$

$$\{\pi \mapsto t\} \models \forall\pi'.\,\mathbf{G}\,\mathtt{on}_\pi \leftrightarrow \mathtt{on}_{\pi'} \qquad \text{for all } t \in T$$

$$\{\pi \mapsto t, \pi' \mapsto t'\} \models \mathbf{G}\,\mathtt{on}_\pi \leftrightarrow \mathtt{on}_{\pi'} \qquad \text{for all } t' \in T$$

Semantics

$$\varphi = \forall\pi.\,\forall\pi'.\,\mathbf{G}\,\mathtt{on}_\pi \leftrightarrow \mathtt{on}_{\pi'}$$

$T \subseteq (2^{\mathrm{AP}})^\omega$ is a model of φ iff

$$\{\} \models \forall\pi.\,\forall\pi'.\,\mathbf{G}\,\mathtt{on}_\pi \leftrightarrow \mathtt{on}_{\pi'}$$

$$\{\pi \mapsto t\} \models \forall\pi'.\,\mathbf{G}\,\mathtt{on}_\pi \leftrightarrow \mathtt{on}_{\pi'} \qquad \text{for all } t \in T$$

$$\{\pi \mapsto t, \pi' \mapsto t'\} \models \mathbf{G}\,\mathtt{on}_\pi \leftrightarrow \mathtt{on}_{\pi'} \qquad \text{for all } t' \in T$$

$$\{\pi \mapsto t[n,\infty), \pi' \mapsto t'[n,\infty)\} \models \mathtt{on}_\pi \leftrightarrow \mathtt{on}_{\pi'} \qquad \text{for all } n \in \mathbb{N}$$

Semantics

$$\varphi = \forall \pi.\, \forall \pi'.\, \mathbf{G}\, \mathtt{on}_\pi \leftrightarrow \mathtt{on}_{\pi'}$$

$T \subseteq (2^{\mathrm{AP}})^\omega$ is a model of φ iff

$$\{\} \models \forall \pi.\, \forall \pi'.\, \mathbf{G}\, \mathtt{on}_\pi \leftrightarrow \mathtt{on}_{\pi'}$$

$$\{\pi \mapsto t\} \models \forall \pi'.\, \mathbf{G}\, \mathtt{on}_\pi \leftrightarrow \mathtt{on}_{\pi'} \qquad \text{for all } t \in T$$

$$\{\pi \mapsto t, \pi' \mapsto t'\} \models \mathbf{G}\, \mathtt{on}_\pi \leftrightarrow \mathtt{on}_{\pi'} \qquad \text{for all } t' \in T$$

$$\{\pi \mapsto t[n,\infty), \pi' \mapsto t'[n,\infty)\} \models \mathtt{on}_\pi \leftrightarrow \mathtt{on}_{\pi'} \qquad \text{for all } n \in \mathbb{N}$$

$$\mathtt{on} \in t(n) \Leftrightarrow \mathtt{on} \in t'(n)$$

Applications

- Uniform framework for information-flow control
  - Does a system leak information?
- Symmetries in distributed systems
  - Are clients treated symmetrically?
- Error resistant codes
  - Do codes for distinct inputs have at least Hamming distance d?
- Software doping
  - Think emission scandal in automotive industry

The Virtues of LTL

LTL has many desirables properties:

1. Every satisfiable LTL formula is satisfied by an ultimately periodic trace, i.e., by a finite and finitely-represented model.
2. LTL satisfiability and model-checking are PSpace-complete.
3. LTL and FO[$<$] are expressively equivalent.

Which properties does HyperLTL retain ?

References

- Michael R. Clarkson and Fred B. Schneider. **Hyperproperties.** *Journal of Computer Security* (2010).
- Michael R. Clarkson, Bernd Finkbeiner, Masoud Koleini, Kristopher K. Micinski, Markus N. Rabe, and César Sánchez. **Temporal logics for hyperproperties.** In *Proceedings of POST 2014*.
- Bernd Finkbeiner and Markus N. Rabe. **The Linear-Hyper-Branching Spectrum of Temporal Logics.** it-Information Technology (2014).
- Markus N. Rabe. **A Temporal Logic Approach to Information-flow Control.** PhD thesis, Saarland University (2016).

Outline

What about Finite Models?

Fix $\mathrm{AP} = \{a\}$ and consider the conjunction φ of

- $\forall\pi.\ (\neg a_\pi)\,\mathbf{U}\,(a_\pi \wedge \mathbf{X}\,\mathbf{G}\,\neg a_\pi)$

What about Finite Models?

Fix $\mathrm{AP} = \{a\}$ and consider the conjunction φ of

- $\forall \pi.\ (\neg a_\pi)\, \mathbf{U}\, (a_\pi \wedge \mathbf{X}\, \mathbf{G}\, \neg a_\pi)$
- $\exists \pi.\ a_\pi$

What about Finite Models?

Fix $\mathrm{AP} = \{a\}$ and consider the conjunction φ of

- $\forall \pi.\ (\neg a_\pi)\,\mathbf{U}\,(a_\pi \wedge \mathbf{X}\,\mathbf{G}\,\neg a_\pi)$
- $\exists \pi.\ a_\pi$

$\{a\} \quad \emptyset \quad \emptyset \quad \emptyset \quad \emptyset \quad \emptyset \quad \emptyset \quad \emptyset \quad \cdots$

What about Finite Models?

Fix $\mathrm{AP} = \{a\}$ and consider the conjunction φ of

- $\forall \pi.\ (\neg a_\pi)\,\mathbf{U}\,(a_\pi \wedge \mathbf{X}\,\mathbf{G}\,\neg a_\pi)$
- $\exists \pi.\ a_\pi$
- $\forall \pi.\ \exists \pi'.\ \mathbf{F}\,(a_\pi \wedge \mathbf{X}\,a_{\pi'})$

$\{a\} \quad \emptyset \quad \emptyset \quad \emptyset \quad \emptyset \quad \emptyset \quad \emptyset \quad \emptyset \quad \cdots$

What about Finite Models?

Fix $\mathrm{AP} = \{a\}$ and consider the conjunction φ of

- $\forall \pi.\ (\neg a_\pi)\,\mathbf{U}\,(a_\pi \wedge \mathbf{X}\,\mathbf{G}\,\neg a_\pi)$
- $\exists \pi.\ a_\pi$
- $\forall \pi.\ \exists \pi'.\ \mathbf{F}\,(a_\pi \wedge \mathbf{X}\,a_{\pi'})$

$$\begin{array}{ccccccccc} \{a\} & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \cdots \\ \emptyset & \{a\} & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \cdots \end{array}$$

What about Finite Models?

Fix $\mathrm{AP} = \{a\}$ and consider the conjunction φ of

- $\forall\pi.\ (\neg a_\pi)\,\mathbf{U}\,(a_\pi \wedge \mathbf{X}\,\mathbf{G}\,\neg a_\pi)$
- $\exists\pi.\ a_\pi$
- $\forall\pi.\ \exists\pi'.\ \mathbf{F}\,(a_\pi \wedge \mathbf{X}\,a_{\pi'})$

$$\begin{array}{ccccccccc}
\{a\} & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \cdots \\
\emptyset & \{a\} & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \cdots \\
\emptyset & \emptyset & \{a\} & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots &
\end{array}$$

The unique model of φ is $\{\emptyset^n\,\{a\}\,\emptyset^\omega \mid n \in \mathbb{N}\}$.

What about Finite Models?

Fix $\mathrm{AP} = \{a\}$ and consider the conjunction φ of

- $\forall \pi.\ (\neg a_\pi)\,\mathbf{U}\,(a_\pi \wedge \mathbf{X}\,\mathbf{G}\,\neg a_\pi)$
- $\exists \pi.\ a_\pi$
- $\forall \pi.\ \exists \pi'.\ \mathbf{F}\,(a_\pi \wedge \mathbf{X}\, a_{\pi'})$

$$\begin{array}{ccccccccc}
\{a\} & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \cdots \\
\emptyset & \{a\} & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \cdots \\
\emptyset & \emptyset & \{a\} & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots &
\end{array}$$

The unique model of φ is $\{\emptyset^n \{a\} \emptyset^\omega \mid n \in \mathbb{N}\}$.

Theorem

There is a satisfiable HyperLTL sentence that is not satisfied by any finite set of traces.

What about Countable Models?

Theorem

Every satisfiable HyperLTL sentence has a countable model.

What about Countable Models?

Theorem

Every satisfiable HyperLTL sentence has a countable model.

Proof

- W.l.o.g. $\varphi = \forall \pi_0.\ \exists \pi'_0. \cdots \forall \pi_k.\ \exists \pi'_k.\ \psi$ with quantifier-free ψ.
- Fix a Skolem function f_j for every existentially quantified π'_j.

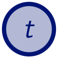
What about Countable Models?

Theorem

Every satisfiable HyperLTL sentence has a countable model.

Proof

- W.l.o.g. $\varphi = \forall \pi_0.\ \exists \pi'_0. \cdots \forall \pi_k.\ \exists \pi'_k.\ \psi$ with quantifier-free ψ.
- Fix a Skolem function f_j for every existentially quantified π'_j.



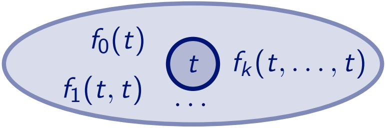
What about Countable Models?

Theorem

Every satisfiable HyperLTL sentence has a countable model.

Proof

- W.l.o.g. $\varphi = \forall\pi_0.\ \exists\pi'_0.\cdots\forall\pi_k.\ \exists\pi'_k.\ \psi$ with quantifier-free ψ.
- Fix a Skolem function f_j for every existentially quantified π'_j.



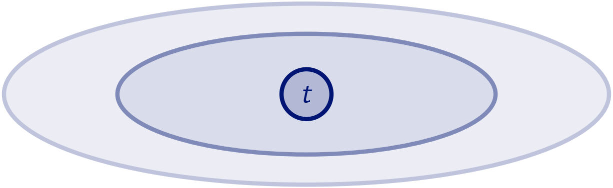
What about Countable Models?

Theorem

Every satisfiable HyperLTL sentence has a countable model.

Proof

- W.l.o.g. $\varphi = \forall\pi_0.\ \exists\pi'_0.\cdots\forall\pi_k.\ \exists\pi'_k.\ \psi$ with quantifier-free ψ.
- Fix a Skolem function f_j for every existentially quantified π'_j.



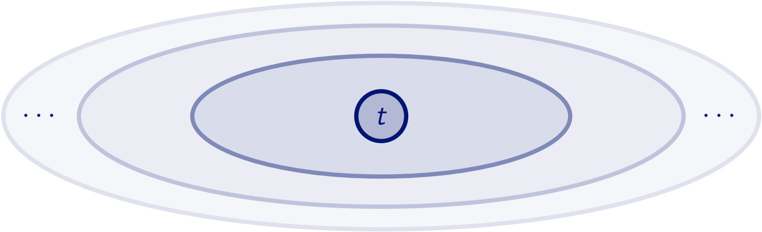
What about Countable Models?

Theorem

Every satisfiable HyperLTL sentence has a countable model.

Proof

- W.l.o.g. $\varphi = \forall \pi_0.\ \exists \pi'_0. \cdots \forall \pi_k.\ \exists \pi'_k.\ \psi$ with quantifier-free ψ.
- Fix a Skolem function f_j for every existentially quantified π'_j.



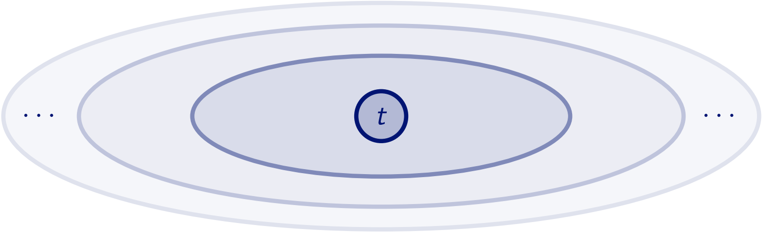
What about Countable Models?

Theorem

Every satisfiable HyperLTL sentence has a countable model.

Proof

- W.l.o.g. $\varphi = \forall\pi_0.\ \exists\pi'_0.\cdots\forall\pi_k.\ \exists\pi'_k.\ \psi$ with quantifier-free ψ.
- Fix a Skolem function f_j for every existentially quantified π'_j.



The limit is a model of φ and countable.

What about Regular Models?

Theorem

There is a satisfiable HyperLTL sentence that is not satisfied by any ω-regular set of traces.

What about Regular Models?

Theorem

There is a satisfiable HyperLTL sentence that is not satisfied by any ω-regular set of traces.

Proof

Express that a model T contains..

1. .. $(\{a\}\{b\})^n\emptyset^\omega$ for every n.

What about Regular Models?

Theorem

There is a satisfiable HyperLTL sentence that is not satisfied by any ω-regular set of traces.

Proof

Express that a model T contains..

$\{a\}\ \{b\}\ \{a\}\ \{b\}\ \{a\}\ \{b\}\ \emptyset^\omega$

1. .. $(\{a\}\{b\})^n\emptyset^\omega$ for every n.

What about Regular Models?

Theorem

There is a satisfiable HyperLTL sentence that is not satisfied by any ω-regular set of traces.

Proof

Express that a model T contains..

1. .. $(\{a\}\{b\})^n\emptyset^\omega$ for every n.
2. .. for every trace of the form $x\{b\}\{a\}y$ in T, also the trace $x\{a\}\{b\}y$.

$\{a\}\ \{b\}\ \{a\}\ \{b\}\ \{a\}\ \{b\}\ \emptyset^\omega$

What about Regular Models?

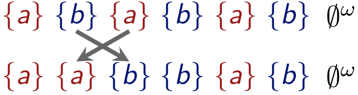
Theorem

There is a satisfiable HyperLTL sentence that is not satisfied by any ω-regular set of traces.

Proof

Express that a model T contains..

1. .. $(\{a\}\{b\})^n\emptyset^\omega$ for every n.
2. .. for every trace of the form $x\{b\}\{a\}y$ in T, also the trace $x\{a\}\{b\}y$.



What about Regular Models?

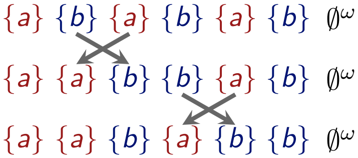
Theorem

There is a satisfiable HyperLTL sentence that is not satisfied by any ω-regular set of traces.

Proof

Express that a model T contains..

1. .. $(\{a\}\{b\})^n\emptyset^\omega$ for every n.
2. .. for every trace of the form $x\{b\}\{a\}y$ in T, also the trace $x\{a\}\{b\}y$.



What about Regular Models?

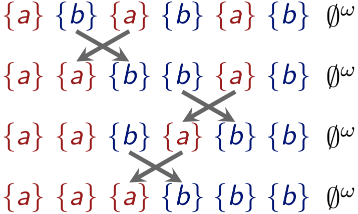
Theorem

There is a satisfiable HyperLTL sentence that is not satisfied by any ω-regular set of traces.

Proof

Express that a model T contains..

1. .. $(\{a\}\{b\})^n\emptyset^\omega$ for every n.
2. .. for every trace of the form $x\{b\}\{a\}y$ in T, also the trace $x\{a\}\{b\}y$.



What about Regular Models?

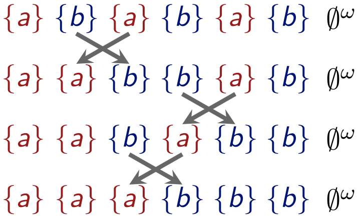
Theorem

There is a satisfiable HyperLTL sentence that is not satisfied by any ω-regular set of traces.

Proof

Express that a model T contains..

1. .. $(\{a\}\{b\})^n\emptyset^\omega$ for every n.
2. .. for every trace of the form $x\{b\}\{a\}y$ in T, also the trace $x\{a\}\{b\}y$.



Then, $T \cap \{a\}^*\{b\}^*\emptyset^\omega = \{\{a\}^n\{b\}^n\emptyset^\omega \mid n \in \mathbb{N}\}$ is not ω-regular.

What about Ultimately Periodic Models?

Theorem

There is a satisfiable HyperLTL sentence that is not satisfied by any set of traces that contains an ultimately periodic trace.

What about Ultimately Periodic Models?

Theorem

There is a satisfiable HyperLTL sentence that is not satisfied by any set of traces that contains an ultimately periodic trace.

One can even encode the prime numbers in HyperLTL!

References

- Bernd Finkbeiner and Martin Zimmermann. **The first-order logic of hyperproperties.** In *Proceedings of STACS 2017*.

Outline

Undecidability

The HyperLTL satisfiability problem:

Given φ, is there a non-empty set T of traces with $T \models \varphi$?

Theorem

HyperLTL satisfiability is undecidable.

Undecidability

The HyperLTL satisfiability problem:

Given φ, is there a non-empty set T of traces with $T \models \varphi$?

Theorem

HyperLTL satisfiability is undecidable.

Proof:

By a reduction from Post's correspondence problem.

Example

Blocks (a, baa) (ab, aa) (bba, bb)

Undecidability

The HyperLTL satisfiability problem:

Given φ, is there a non-empty set T of traces with $T \models \varphi$?

Theorem

HyperLTL satisfiability is undecidable.

Proof:

By a reduction from Post's correspondence problem.

Example

Blocks (a, baa) (ab, aa) (bba, bb)

A solution:

b	b	a	a	b	b	b	a	a
b	b	a	a	b	b	b	a	a

Undecidability

The HyperLTL satisfiability problem:

Given φ, is there a non-empty set T of traces with $T \models \varphi$?

Theorem

HyperLTL satisfiability is undecidable.

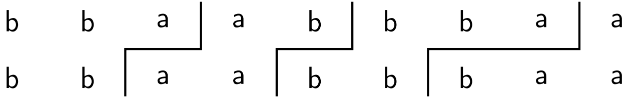
Proof:

By a reduction from Post's correspondence problem.

Example

Blocks (a, baa) (ab, aa) (bba, bb)

A solution:



Undecidability

1. There is a (solution) trace where top matches bottom.

Undecidability

1. There is a (solution) trace where top matches bottom.

$$\{b\}\ \{b\}\ \{a\}\ \{a\}\ \{b\}\ \{b\}\ \{b\}\ \{a\}\ \{a\}\ \emptyset^\omega$$
$$\{b\}\ \{b\}\ \{a\}\ \{a\}\ \{b\}\ \{b\}\ \{b\}\ \{a\}\ \{a\}\ \emptyset^\omega$$

Undecidability

$\{b\}\ \{b\}\ \{a\}\ \{a\}\ \{b\}\ \{b\}\ \{b\}\ \{a\}\ \{a\}\ \emptyset^\omega$

$\{b\}\ \{b\}\ \{a\}\ \{a\}\ \{b\}\ \{b\}\ \{b\}\ \{a\}\ \{a\}\ \emptyset^\omega$

1. There is a (solution) trace where top matches bottom.
2. Every trace is *finite* and starts with a block or is *empty*.

Undecidability

1. There is a (solution) trace where top matches bottom.
2. Every trace is *finite* and starts with a block or is *empty*.

$$\{b\}\ \{b\}\ \{a\}\ \{a\}\ \{b\}\ \{b\}\ \{b\}\ \{a\}\ \{a\}\ \emptyset^\omega$$
$$\{b\}\ \{b\}\ \{a\}\ \{a\}\ \{b\}\ \{b\}\ \{b\}\ \{a\}\ \{a\}\ \emptyset^\omega$$

Undecidability

$$\{b\}\ \ \{b\}\ \ \{a\}\ \big|\ \{a\}\ \ \{b\}\ \ \{b\}\ \ \{b\}\ \ \{a\}\ \ \{a\}\ \ \emptyset^\omega$$
$$\{b\}\ \ \{b\}\ \big|\ \{a\}\ \ \{a\}\ \ \{b\}\ \ \{b\}\ \ \{b\}\ \ \{a\}\ \ \{a\}\ \ \emptyset^\omega$$

1. There is a (solution) trace where top matches bottom.
2. Every trace is *finite* and starts with a block or is *empty*.
3. For every non-empty trace, the trace obtained by removing the first block also exists.

Undecidability

1. There is a (solution) trace where top matches bottom.
2. Every trace is *finite* and starts with a block or is *empty*.
3. For every non-empty trace, the trace obtained by removing the first block also exists.

$$\{b\}\ \{b\}\ \{a\}\ |\ \{a\}\ \{b\}\ \{b\}\ \{b\}\ \{a\}\ \{a\}\ \emptyset^\omega$$
$$\{b\}\ \{b\}\ |\ \{a\}\ \{a\}\ \{b\}\ \{b\}\ \{b\}\ \{a\}\ \{a\}\ \emptyset^\omega$$

$$\{a\}\ \{b\}\ \{b\}\ \{b\}\ \{a\}\ \{a\}\ \emptyset\ \emptyset\ \emptyset\ \emptyset^\omega$$
$$\{a\}\ \{a\}\ \{b\}\ \{b\}\ \{b\}\ \{a\}\ \{a\}\ \emptyset\ \emptyset\ \emptyset^\omega$$

Undecidability

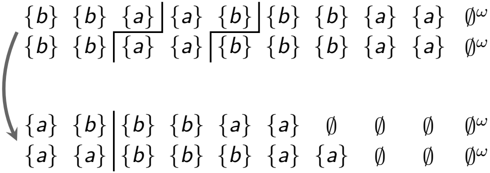
1. There is a (solution) trace where top matches bottom.
2. Every trace is *finite* and starts with a block or is *empty*.
3. For every non-empty trace, the trace obtained by removing the first block also exists.

$$\{b\}\ \{b\}\ \{a\} \mid \{a\}\ \{b\}\ \{b\}\ \{b\}\ \{a\}\ \{a\}\ \emptyset^\omega$$
$$\{b\}\ \{b\} \mid \{a\}\ \{a\}\ \{b\}\ \{b\}\ \{b\}\ \{a\}\ \{a\}\ \emptyset^\omega$$

$$\{a\}\ \{b\} \mid \{b\}\ \{b\}\ \{a\}\ \{a\}\ \emptyset\ \emptyset\ \emptyset\ \emptyset^\omega$$
$$\{a\}\ \{a\} \mid \{b\}\ \{b\}\ \{b\}\ \{a\}\ \{a\}\ \emptyset\ \emptyset\ \emptyset^\omega$$

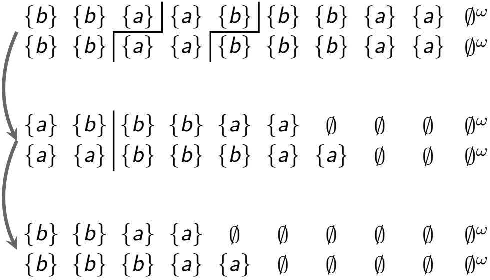
Undecidability

1. There is a (solution) trace where top matches bottom.
2. Every trace is *finite* and starts with a block or is *empty*.
3. For every non-empty trace, the trace obtained by removing the first block also exists.



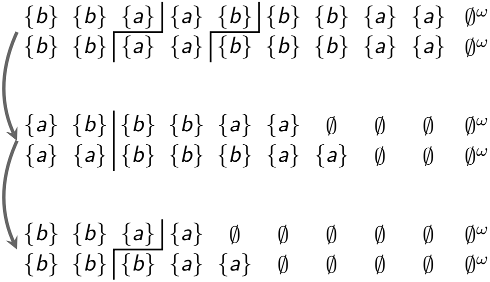
Undecidability

1. There is a (solution) trace where top matches bottom.
2. Every trace is *finite* and starts with a block or is *empty*.
3. For every non-empty trace, the trace obtained by removing the first block also exists.



Undecidability

1. There is a (solution) trace where top matches bottom.
2. Every trace is *finite* and starts with a block or is *empty*.
3. For every non-empty trace, the trace obtained by removing the first block also exists.



Undecidability

1. There is a (solution) trace where top matches bottom.
2. Every trace is *finite* and starts with a block or is *empty*.
3. For every non-empty trace, the trace obtained by removing the first block also exists.

$\{b\}\ \{b\}\ \{a\} \mid \{a\}\ \{b\} \mid \{b\}\ \{b\}\ \{a\} \mid \{a\}\ \emptyset^\omega$

$\{b\}\ \{b\} \mid \{a\}\ \{a\} \mid \{b\}\ \{b\} \mid \{b\}\ \{a\}\ \{a\}\ \emptyset^\omega$

$\{a\}\ \{b\} \mid \{b\}\ \{b\}\ \{a\}\ \{a\}\ \emptyset\ \emptyset\ \emptyset\ \emptyset^\omega$

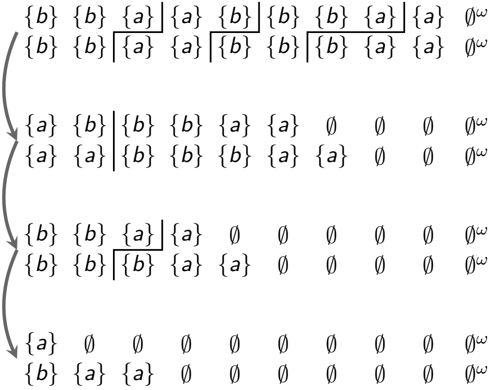
$\{a\}\ \{a\} \mid \{b\}\ \{b\}\ \{b\}\ \{a\}\ \{a\}\ \emptyset\ \emptyset\ \emptyset^\omega$

$\{b\}\ \{b\}\ \{a\} \mid \{a\}\ \emptyset\ \emptyset\ \emptyset\ \emptyset\ \emptyset\ \emptyset^\omega$

$\{b\}\ \{b\} \mid \{b\}\ \{a\}\ \{a\}\ \emptyset\ \emptyset\ \emptyset\ \emptyset\ \emptyset^\omega$

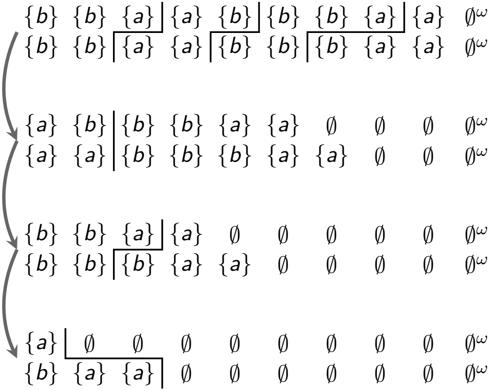
Undecidability

1. There is a (solution) trace where top matches bottom.
2. Every trace is *finite* and starts with a block or is *empty*.
3. For every non-empty trace, the trace obtained by removing the first block also exists.



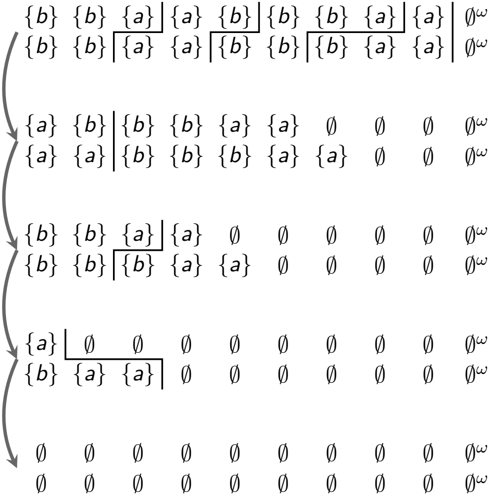
Undecidability

1. There is a (solution) trace where top matches bottom.
2. Every trace is *finite* and starts with a block or is *empty*.
3. For every non-empty trace, the trace obtained by removing the first block also exists.



Undecidability

1. There is a (solution) trace where top matches bottom.
2. Every trace is *finite* and starts with a block or is *empty*.
3. For every non-empty trace, the trace obtained by removing the first block also exists.



Decidability

Theorem

$\exists^*$*-HyperLTL satisfiability is* PSpace*-complete.*

Decidability

Theorem

$\exists^*$-*HyperLTL satisfiability is* PSpace-*complete.*

Proof:

- Membership:
  - Consider $\varphi = \exists\pi_0 \ldots \exists\pi_k.\, \psi$.
  - Obtain ψ' from ψ by replacing each a_{π_j} by a fresh proposition a_j.
  - Then: φ and the LTL formula ψ' are equi-satisfiable.
- Hardness: trivial reduction from LTL satisfiability

Decidability

Theorem

$\forall^*$*-HyperLTL satisfiability is* PSpace*-complete.*

Decidability

Theorem

$\forall^*$-*HyperLTL satisfiability is* PSpace-*complete.*

Proof:

- Membership:
  - Consider $\varphi = \forall \pi_0 \ldots \forall \pi_k. \psi$.
  - Obtain ψ' from ψ by replacing each a_{π_j} by a.
  - Then: φ and the LTL formula ψ' are equi-satisfiable.
- Hardness: trivial reduction from LTL satisfiability

Decidability

Theorem

$\exists^*\forall^*$-*HyperLTL satisfiability is* ExpSpace-*complete.*

Decidability

Theorem

$\exists^*\forall^*$-*HyperLTL satisfiability is* ExpSpace-*complete.*

Proof:

- Membership:
  - Consider $\varphi = \exists\pi_0 \ldots \exists\pi_k. \forall\pi'_0 \ldots \forall\pi'_\ell. \psi$.
  - Let

$$\varphi' = \exists\pi_0 \ldots \exists\pi_k \bigwedge_{j_0=0}^{k} \cdots \bigwedge_{j_\ell=0}^{k} \psi_{j_0,\ldots,j_\ell}$$

    where $\psi_{j_0,\ldots,j_\ell}$ is obtained from ψ by replacing each occurrence of π'_i by π_{j_i}.
  - Then: φ and φ' are equi-satisfiable.
- Hardness: encoding of exponential-space Turing machines.

Further Results

HyperLTL implication checking: given φ and φ', does, for every T, $T \models \varphi$ imply $T \models \varphi'$?

Lemma

φ does not imply φ' iff $(\varphi \wedge \neg\varphi')$ is satisfiable.

Further Results

HyperLTL implication checking: given φ and φ', does, for every T, $T \models \varphi$ imply $T \models \varphi'$?

Lemma

φ does not imply φ' iff $(\varphi \wedge \neg\varphi')$ is satisfiable.

Corollary

Implication checking for alternation-free HyperLTL formulas is ExpSpace-*complete.*

Tool **EAHyper**:

- satisfiability, implication, and equivalence checking for HyperLTL

References

- Bernd Finkbeiner and Christopher Hahn. **Deciding Hyperproperties.** In *Proceedings of CONCUR 2016*.
- Bernd Finkbeiner, Christopher Hahn, and Marvin Stenger. **EAHyper: Satisfiability, Implication, and Equivalence Checking of Hyperproperties.** In *Proceedings of CAV 2017*.

Outline

Model-Checking

The HyperLTL model-checking problem:

Given a transition system $\mathcal{S}$ and φ, does $\mathrm{Traces}(\mathcal{S}) \models \varphi$?

Theorem

The HyperLTL model-checking problem is decidable.

Model-Checking

Proof:

- Consider $\varphi = \exists\pi_1.\, \forall\pi_2.\, \ldots \exists\pi_{k-1}.\, \forall\pi_k.\, \psi$.
- Rewrite as $\exists\pi_1.\, \neg\exists\pi_2.\, \neg \ldots \exists\pi_{k-1}.\, \neg\exists\pi_k.\, \neg\psi$.

Model-Checking

Proof:

- Consider $\varphi = \exists\pi_1.\, \forall\pi_2.\, \ldots \exists\pi_{k-1}.\, \forall\pi_k.\, \psi$.
- Rewrite as $\exists\pi_1.\, \neg\exists\pi_2.\, \neg \ldots \exists\pi_{k-1}.\, \neg\exists\pi_k.\, \neg\psi$.
- By induction over quantifier prefix construct non-determinstic Büchi automaton $\mathcal{A}$ with $L(\mathcal{A}) \neq \emptyset$ iff $\mathrm{Traces}(\mathcal{S}) \models \varphi$.
  - Induction start: build automaton for LTL formula obtained from $\neg\psi$ by replacing a_{π_j} by a_j.
  - For $\exists\pi_j\theta$ restrict automaton for θ in dimension j to traces of $\mathcal{S}$.
  - For $\neg\theta$ complement automaton for θ.

Model-Checking

Proof:

- Consider $\varphi = \exists\pi_1.\,\forall\pi_2.\,\ldots\exists\pi_{k-1}.\,\forall\pi_k.\,\psi$.
- Rewrite as $\exists\pi_1.\,\neg\exists\pi_2.\,\neg\ldots\exists\pi_{k-1}.\,\neg\exists\pi_k.\,\neg\psi$.
- By induction over quantifier prefix construct non-determinstic Büchi automaton $\mathcal{A}$ with $L(\mathcal{A}) \neq \emptyset$ iff $\mathrm{Traces}(\mathcal{S}) \models \varphi$.
  - Induction start: build automaton for LTL formula obtained from $\neg\psi$ by replacing a_{π_j} by a_j.
  - For $\exists\pi_j\theta$ restrict automaton for θ in dimension j to traces of $\mathcal{S}$.
  - For $\neg\theta$ complement automaton for θ.

$\Rightarrow$ Non-elementary complexity, but alternation-free fragments are as hard as LTL.

References

- Bernd Finkbeiner, Markus N. Rabe, and César Sánchez. **Algorithms for Model Checking HyperLTL and HyperCTL*.** In *Proceedings of CAV 2015*.

Outline

1. HyperLTL
2. The Models Of HyperLTL
3. HyperLTL Satisfiability
4. HyperLTL Model-checking
5. **The First-order Logic of Hyperproperties**
6. Conclusion

First-order Logic vs. LTL

FO$[<]$: first-order order logic over signature $\{<\} \cup \{P_a \mid a \in \mathrm{AP}\}$ over structures with universe $\mathbb{N}$.

Theorem (Kamp '68, Gabbay et al. '80)

LTL and FO$[<]$ are expressively equivalent.

First-order Logic vs. LTL

FO[$<$]: first-order order logic over signature $\{<\} \cup \{P_a \mid a \in \mathrm{AP}\}$ over structures with universe $\mathbb{N}$.

Theorem (Kamp '68, Gabbay et al. '80)

LTL and FO[$<$] *are expressively equivalent.*

Example

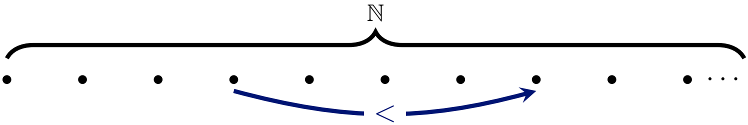
$$\forall x(P_q(x) \wedge \neg P_p(x)) \rightarrow \exists y(x < y \wedge P_p(y))$$

and

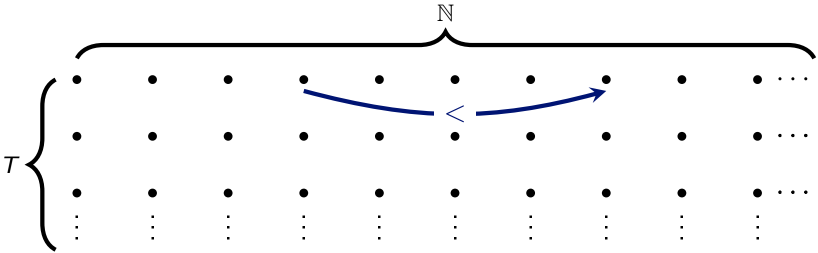
$$\mathbf{G}\,(q \rightarrow \mathbf{F}\,p)$$

are equivalent.

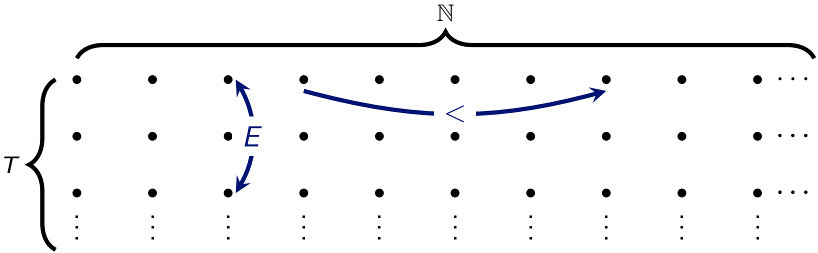
First-order Logic for Hyperproperties



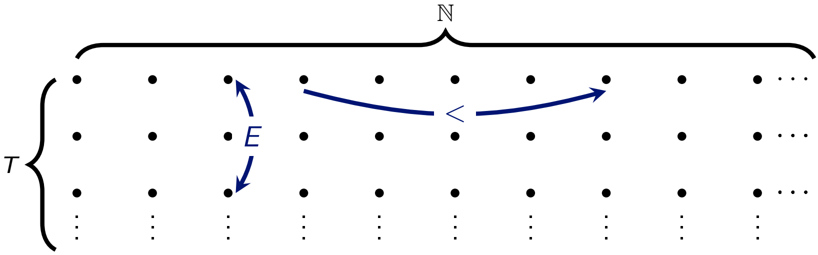
First-order Logic for Hyperproperties



First-order Logic for Hyperproperties



First-order Logic for Hyperproperties

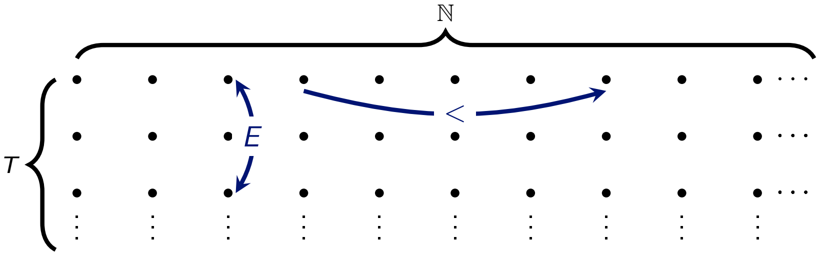


- FO[$<$, E]: first-order logic with equality over the signature $\{<, E\} \cup \{P_a \mid a \in \mathrm{AP}\}$ over structures with universe $T \times \mathbb{N}$.

Example

$$\forall x \forall x' \; E(x, x') \rightarrow (P_{\texttt{on}}(x) \leftrightarrow P_{\texttt{on}}(x'))$$

First-order Logic for Hyperproperties



- FO$[<, E]$: first-order logic with equality over the signature $\{<, E\} \cup \{P_a \mid a \in \mathrm{AP}\}$ over structures with universe $T \times \mathbb{N}$.

Proposition

For every HyperLTL sentence there is an equivalent FO$[<, E]$ sentence.

A Setback

- Let φ be the following property of sets $T \subseteq (2^{\{p\}})^\omega$:

  There is an n such that $p \notin t(n)$ for every $t \in T$.

Theorem (Bozzelli et al. '15)

φ is not expressible in HyperLTL.

A Setback

- Let φ be the following property of sets $T \subseteq (2^{\{p\}})^\omega$:

There is an n such that $p \notin t(n)$ for every $t \in T$.

Theorem (Bozzelli et al. '15)

φ is not expressible in HyperLTL.

- But, φ is easily expressible in FO$[<, E]$:

$$\exists x \, \forall y \, E(x, y) \rightarrow \neg P_p(y)$$

Corollary

FO$[<, E]$ strictly subsumes HyperLTL.

HyperFO

- $\exists^M x$ and $\forall^M x$: quantifiers restricted to initial positions.
- $\exists^G y \geq x$ and $\forall^G y \geq x$: if x is initial, then quantifiers restricted to positions on the same trace as x.

HyperFO

- $\exists^M x$ and $\forall^M x$: quantifiers restricted to initial positions.
- $\exists^G y \geq x$ and $\forall^G y \geq x$: if x is initial, then quantifiers restricted to positions on the same trace as x.

HyperFO: sentences of the form

$$\varphi = Q_1^M x_1. \cdots Q_k^M x_k.\ Q_1^G y_1 \geq x_{g_1}. \cdots Q_\ell^G y_\ell \geq x_{g_\ell}.\ \psi$$

- $Q \in \{\exists, \forall\}$,
- $\{x_1, \ldots, x_k\}$ and $\{y_1, \ldots, y_\ell\}$ are disjoint,
- every guard x_{g_j} is in$\{x_1, \ldots, x_k\}$, and
- ψ is quantifier-free over signature $\{<, E\} \cup \{P_a \mid a \in \mathrm{AP}\}$ with free variables in $\{y_1, \ldots, y_\ell\}$.

Equivalence

Theorem

HyperLTL and HyperFO are equally expressive.

Equivalence

Theorem

HyperLTL and HyperFO are equally expressive.

Proof

- From HyperLTL to HyperFO: structural induction.
- From HyperFO to HyperLTL: reduction to Kamp's theorem.

From HyperFO to HyperLTL

$$\forall x \forall x' \quad E(x, x') \rightarrow (P_{\texttt{on}}(x) \leftrightarrow P_{\texttt{on}}(x'))$$

From HyperFO to HyperLTL

$$\forall x \forall x' \quad E(x, x') \to (P_{\texttt{on}}(x) \leftrightarrow P_{\texttt{on}}(x'))$$

$$\forall^M x_1 \, \forall^M x_2 \quad \forall^G y_1 \geq x_1 \, \forall^G y_2 \geq x_2 E(y_1, y_2) \to (P_{\texttt{on}}(y_1) \leftrightarrow P_{\texttt{on}}(y_2))$$

From HyperFO to HyperLTL

$$\forall x \forall x' \quad E(x, x') \rightarrow (P_{\mathtt{on}}(x) \leftrightarrow P_{\mathtt{on}}(x'))$$

$$\forall^M x_1 \, \forall^M x_2 \quad \forall^G y_1 \geq x_1 \, \forall^G y_2 \geq x_2 E(y_1, y_2) \rightarrow (P_{\mathtt{on}}(y_1) \leftrightarrow P_{\mathtt{on}}(y_2))$$

$x_1 \mapsto$	$\{\mathtt{on}\}$	$\{\mathtt{on}\}$	$\emptyset$	$\{\mathtt{on}\}$	$\cdots$
$x_2 \mapsto$	$\{\mathtt{on}\}$	$\emptyset$	$\emptyset$	$\{\mathtt{on}\}$	$\cdots$

From HyperFO to HyperLTL

$$\forall x \forall x' \quad E(x,x') \rightarrow (P_{\mathtt{on}}(x) \leftrightarrow P_{\mathtt{on}}(x'))$$

$$\forall^G y_1 \geq x_1 \, \forall^G y_2 \geq x_2 E(y_1,y_2) \rightarrow (P_{\mathtt{on}}(y_1) \leftrightarrow P_{\mathtt{on}}(y_2))$$

$x_1 \mapsto$	$\{\mathtt{on}\}$	$\{\mathtt{on}\}$	$\emptyset$	$\{\mathtt{on}\}$	$\cdots$
$x_2 \mapsto$	$\{\mathtt{on}\}$	$\emptyset$	$\emptyset$	$\{\mathtt{on}\}$	$\cdots$

From HyperFO to HyperLTL

$$\forall x \forall x' \quad E(x, x') \rightarrow (P_{\texttt{on}}(x) \leftrightarrow P_{\texttt{on}}(x'))$$

$$\forall^G y_1 \geq x_1 \, \forall^G y_2 \geq x_2 E(y_1, y_2) \rightarrow (P_{\texttt{on}}(y_1) \leftrightarrow P_{\texttt{on}}(y_2))$$

$$\forall y_1 \, \forall y_2 \; (y_1 = y_2) \rightarrow (P_{(\texttt{on},1)}(y_1) \leftrightarrow P_{(\texttt{on},2)}(y_2))$$

$\{(\mathrm{on}, 1), (\mathrm{on}, 2)\}$	$\{(\mathrm{on}, 1)\}$	$\emptyset$	$\{(\mathrm{on}, 1), (\mathrm{on}, 2)\}$	$\cdots$

From HyperFO to HyperLTL

$$\forall x \forall x' \quad E(x, x') \rightarrow (P_{\mathrm{on}}(x) \leftrightarrow P_{\mathrm{on}}(x'))$$

$$\forall^G y_1 \geq x_1 \, \forall^G y_2 \geq x_2 E(y_1, y_2) \rightarrow (P_{\mathrm{on}}(y_1) \leftrightarrow P_{\mathrm{on}}(y_2))$$

$$\forall y_1 \, \forall y_2 \, (y_1 = y_2) \rightarrow (P_{(\mathrm{on},1)}(y_1) \leftrightarrow P_{(\mathrm{on},2)}(y_2))$$

$$\mathbf{G}\,((\mathrm{on}, 1) \leftrightarrow (\mathrm{on}, 2))$$

$\{(\mathrm{on}, 1), (\mathrm{on}, 2)\}$	$\{(\mathrm{on}, 1)\}$	$\emptyset$	$\{(\mathrm{on}, 1), (\mathrm{on}, 2)\}$	$\cdots$

From HyperFO to HyperLTL

$$\forall x \forall x' \quad E(x, x') \rightarrow (P_{\mathrm{on}}(x) \leftrightarrow P_{\mathrm{on}}(x'))$$

$$\forall^M x_1 \forall^M x_2 \quad \forall^G y_1 \geq x_1 \forall^G y_2 \geq x_2 E(y_1, y_2) \rightarrow (P_{\mathrm{on}}(y_1) \leftrightarrow P_{\mathrm{on}}(y_2))$$

$$\forall y_1 \forall y_2 \ (y_1 = y_2) \rightarrow (P_{(\mathrm{on},1)}(y_1) \leftrightarrow P_{(\mathrm{on},2)}(y_2))$$

$$\mathbf{G}\,((\mathrm{on}, 1) \leftrightarrow (\mathrm{on}, 2))$$

$\{(\mathrm{on}, 1), (\mathrm{on}, 2)\}$	$\{(\mathrm{on}, 1)\}$	$\emptyset$	$\{(\mathrm{on}, 1), (\mathrm{on}, 2)\}$	$\cdots$

From HyperFO to HyperLTL

$$\forall x \forall x' \quad E(x, x') \rightarrow (P_{\mathtt{on}}(x) \leftrightarrow P_{\mathtt{on}}(x'))$$

$$\forall^M x_1 \, \forall^M x_2 \quad \forall^G y_1 \geq x_1 \, \forall^G y_2 \geq x_2 E(y_1, y_2) \rightarrow (P_{\mathtt{on}}(y_1) \leftrightarrow P_{\mathtt{on}}(y_2))$$

$$\forall y_1 \, \forall y_2 \; (y_1 = y_2) \rightarrow (P_{(\mathtt{on},1)}(y_1) \leftrightarrow P_{(\mathtt{on},2)}(y_2))$$

$$\mathbf{G}\,((\mathtt{on}, 1) \leftrightarrow (\mathtt{on}, 2))$$

$$\forall \pi_1 \, \forall \pi_2 \quad \mathbf{G}\,(\mathtt{on}_{\pi_1} \leftrightarrow \mathtt{on}_{\pi_2})$$

$\pi_1 \mapsto$	$\{\mathtt{on}\}$	$\{\mathtt{on}\}$	$\emptyset$	$\{\mathtt{on}\}$	$\cdots$
$\pi_2 \mapsto$	$\{\mathtt{on}\}$	$\emptyset$	$\emptyset$	$\{\mathtt{on}\}$	$\cdots$

References

- Bernd Finkbeiner and Martin Zimmermann. **The first-order logic of hyperproperties.** In *Proceedings of STACS 2017*.

Outline

Conclusion

HyperLTL behaves quite differently than LTL:

- The models of HyperLTL are rather not well-behaved, i.e., in general (countably) infinite, non-regular, and non-periodic.
- Satisfiability is in general undecidable.
- Model-checking is decidable, but non-elementary.

Conclusion

HyperLTL behaves quite differently than LTL:

- The models of HyperLTL are rather not well-behaved, i.e., in general (countably) infinite, non-regular, and non-periodic.
- Satisfiability is in general undecidable.
- Model-checking is decidable, but non-elementary.

But with the feasible problems, you can do exciting things:
HyperLTL is a powerful tool for information security and beyond

- Information-flow control
- Symmetries in distributed systems
- Error resistant codes
- Software doping

Open Problems

- Is there a class of languages $\mathcal{L}$ such that every satisfiable HyperLTL sentence has a model from $\mathcal{L}$?
- Is the quantifier alternation hierarchy strict?
- HyperLTL synthesis
- Is there a temporal logic that is expressively equivalent to FO[$<$, E]?
- What about HyperCTL*?
- Software model-checking
- Quantitative hyperproperties

Open Problems

- Is there a class of languages $\mathcal{L}$ such that every satisfiable HyperLTL sentence has a model from $\mathcal{L}$?
- Is the quantifier alternation hierarchy strict?
- HyperLTL synthesis
- Is there a temporal logic that is expressively equivalent to $\mathrm{FO}[<, E]$?
- What about HyperCTL*?
- Software model-checking
- Quantitative hyperproperties

Thank you