Temporal Logics for Information-flow Policies

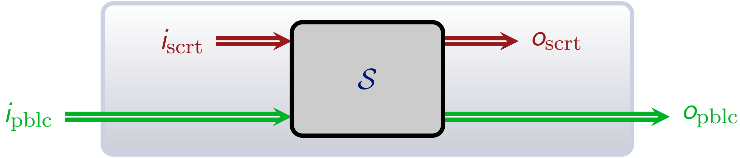
Martin Zimmermann

University of Liverpool

January 14th, 2020

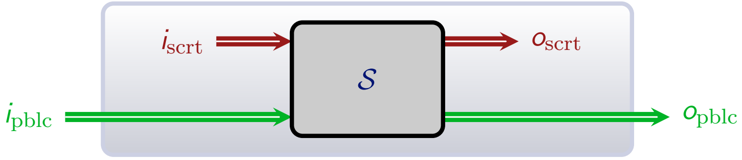
Royal Holloway, London, UK

Motivation



i_{scrt}
o_{scrt}
$\mathcal{S}$
i_{pblc}
o_{pblc}

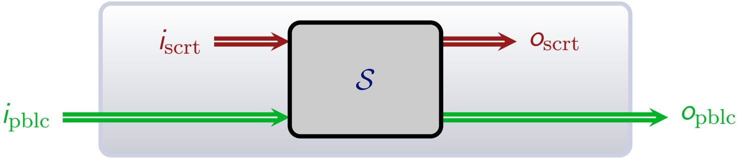
Motivation



Trace-based view on $\mathcal{S}$: observe execution traces, i.e., infinite sequences over 2^{AP} for some set AP of atomic propositions.

$$\{\mathtt{init}, \mathtt{i}_{\mathsf{pblc}}\} \quad \{\mathtt{i}_{\mathsf{scrt}}\} \quad \{\mathtt{i}_{\mathsf{pblc}}\} \quad \{\mathtt{i}_{\mathsf{scrt}}, \mathtt{o}_{\mathsf{pblc}}, \mathtt{term}\} \quad \cdots$$

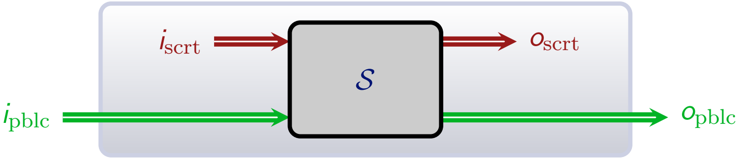
Motivation



Typical requirements:

- $\mathcal{S}$ terminates

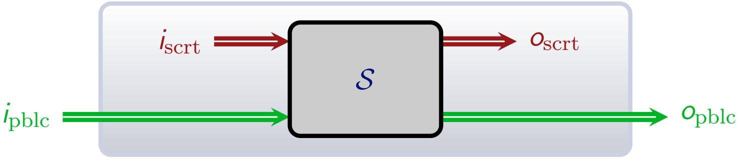
Motivation



Typical requirements:

- $\mathcal{S}$ terminates
- $\mathcal{S}$ terminates within a uniform time bound

Motivation

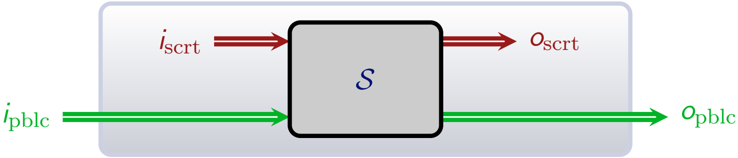


Typical requirements:

- $\mathcal{S}$ terminates
- $\mathcal{S}$ terminates within a uniform time bound
- $\mathcal{S}$ is input-deterministic: for all traces t, t' of $\mathcal{S}$

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

Motivation



Typical requirements:

- $\mathcal{S}$ terminates
- $\mathcal{S}$ terminates within a uniform time bound
- $\mathcal{S}$ 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 $\mathcal{S}$

$$t =_{i_{\mathrm{pblc}}} t' \quad \text{implies} \quad t =_{o_{\mathrm{pblc}}} t'$$

Trace Properties vs. Hyperproperties

Definition

A trace property $T \subseteq (2^{\mathrm{AP}})^\omega$ is a set of traces. A system $\mathcal{S}$ satisfies T, if $\mathrm{Traces}(\mathcal{S}) \subseteq T$.

Example: The set of traces where `term` holds at least once.

Trace Properties vs. Hyperproperties

Definition

A trace property $T \subseteq (2^{\mathrm{AP}})^\omega$ is a set of traces. A system $\mathcal{S}$ satisfies T, if $\mathrm{Traces}(\mathcal{S}) \subseteq T$.

Example: The set of traces where `term` holds at least once.

Definition

A hyperproperty $H \subseteq 2^{(2^{\mathrm{AP}})^\omega}$ is a set of sets of traces. A system $\mathcal{S}$ satisfies H if $\mathrm{Traces}(\mathcal{S}) \in H$.

Example: The set $\{T \subseteq T_n \mid n \in \mathbb{N}\}$ where T_n is the trace property containing the traces where `term` holds at least once within the first n positions.

Outline

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

Outline

LTL in One Slide

Syntax

$$\varphi ::= a \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathbf{X}\,\varphi \mid \varphi\,\mathbf{U}\,\varphi \qquad \text{where } a \in \mathrm{AP}$$

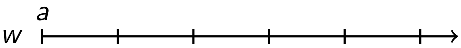
LTL in One Slide

Syntax

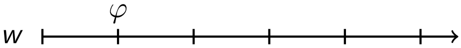
$$\varphi ::= a \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathbf{X}\,\varphi \mid \varphi\,\mathbf{U}\,\varphi \qquad \text{where } a \in \mathrm{AP}$$

Semantics

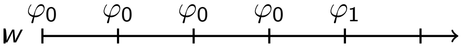
- $w \models a$:



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



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



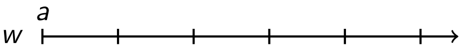
LTL in One Slide

Syntax

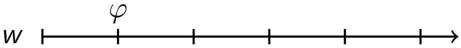
$$\varphi ::= a \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathbf{X}\,\varphi \mid \varphi\,\mathbf{U}\,\varphi \qquad \text{where } a \in \mathrm{AP}$$

Semantics

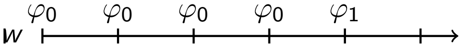
- $w \models a$:



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



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



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

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

Examples

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

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

In HyperLTL: $\forall\pi\forall\pi'.\ \mathbf{G}\,(i_\pi \leftrightarrow i_{\pi'}) \rightarrow \mathbf{G}\,(o_\pi \leftrightarrow o_{\pi'})$

Examples

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

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

In HyperLTL: $\forall\pi\forall\pi'.\ \mathbf{G}\,(i_\pi \leftrightarrow i_{\pi'}) \rightarrow \mathbf{G}\,(o_\pi \leftrightarrow o_{\pi'})$

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

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

In HyperLTL:
$\forall\pi\forall\pi'.\ \mathbf{G}\,((i_{\mathsf{pblc}})_\pi \leftrightarrow (i_{\mathsf{pblc}})_{\pi'}) \rightarrow \mathbf{G}\,((o_{\mathsf{pblc}})_\pi \leftrightarrow (o_{\mathsf{pblc}})_{\pi'})$

Examples

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

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

In HyperLTL: $\forall\pi\forall\pi'.\ \mathbf{G}\,(i_\pi \leftrightarrow i_{\pi'}) \rightarrow \mathbf{G}\,(o_\pi \leftrightarrow o_{\pi'})$

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

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

In HyperLTL:
$\forall\pi\forall\pi'.\ \mathbf{G}\,((i_{\mathsf{pblc}})_\pi \leftrightarrow (i_{\mathsf{pblc}})_{\pi'}) \rightarrow \mathbf{G}\,((o_{\mathsf{pblc}})_\pi \leftrightarrow (o_{\mathsf{pblc}})_{\pi'})$

- $\mathcal{S}$ terminates within a uniform time bound.
Not expressible in HyperLTL.

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

There are prototype tools for model checking, satisfiability checking, runtime verification, and synthesis.

The Virtues of LTL

LTL is the most important specification language for reactive systems and has many desirable properties:

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

Which properties does HyperLTL retain?

Outline

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

What about Finite Models?

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

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

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 (Finkbeiner & Z. '17)

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

More Results

Theorem (Finkbeiner & Z. '17)

Every satisfiable HyperLTL sentence has a countable model.

More Results

Theorem (Finkbeiner & Z. '17)

Every satisfiable HyperLTL sentence has a countable model.

What about ω-regular models?

Theorem (Finkbeiner & Z. '17)

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

More Results

Theorem (Finkbeiner & Z. '17)

Every satisfiable HyperLTL sentence has a countable model.

What about ω-regular models?

Theorem (Finkbeiner & Z. '17)

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

What about ultimately periodic models?

Theorem (Finkbeiner & Z. '17)

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

Outline

1. HyperLTL
2. The Models Of HyperLTL
3. **The First-order Logic of Hyperproperties**
4. HyperLTL Satisfiability
5. Team Semantics
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

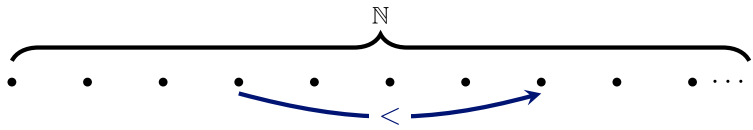
$$\forall x(P_q(x) \land \neg P_p(x)) \to \exists y(x < y \land P_p(y))$$

and

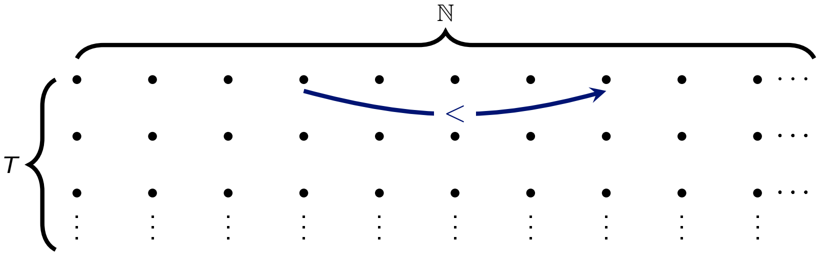
$$\mathbf{G}\,(q \to \mathbf{F}\,p)$$

are equivalent.

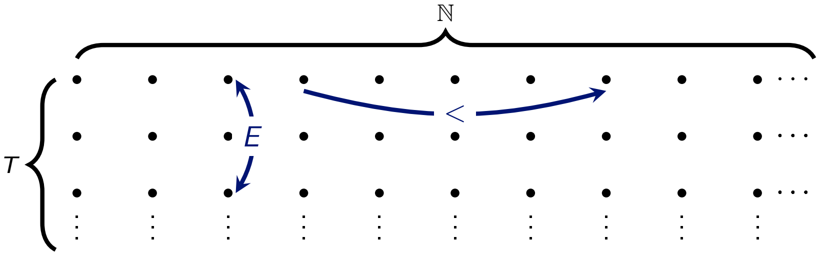
First-order Logic for Hyperproperties



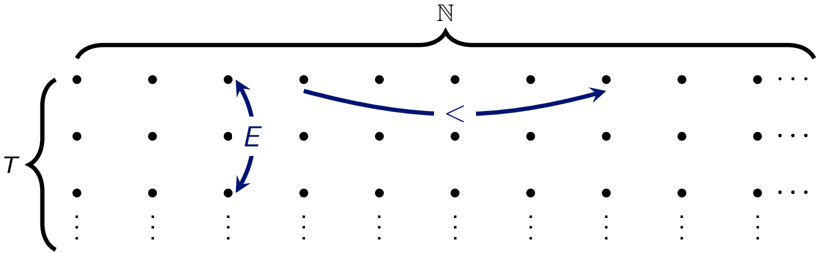
First-order Logic for Hyperproperties



First-order Logic for Hyperproperties



First-order Logic for Hyperproperties

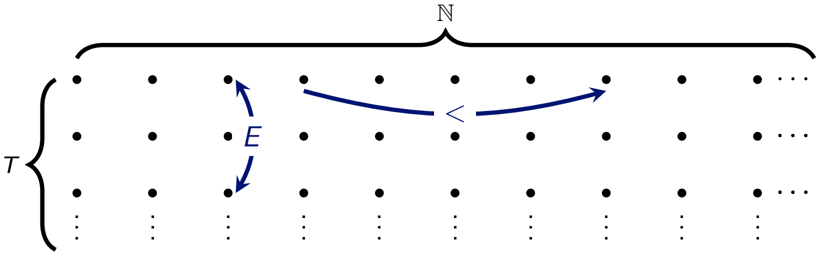


- FO[$<$, E]: first-order logic with equality over the signature $\{<, E\} \cup \{P_a \mid a \in \mathrm{AP}\}$ over structures with universe $T \times \mathbb{N}$.

Example

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

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$[<, E]$:

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

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 (Finkbeiner & Z. '17)

HyperLTL and HyperFO are equally expressive.

Equivalence

Theorem (Finkbeiner & Z. '17)

HyperLTL and HyperFO are equally expressive.

Proof

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

Outline

Undecidability

The HyperLTL satisfiability problem:

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

Theorem (Finkbeiner & Hahn '16)

$\forall\exists$-*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 (Finkbeiner & Hahn '16)

$\forall\exists$-HyperLTL satisfiability is undecidable.

Proof:

Express the mortality problem for Turing machines: Given a Turing machine, decide whether it has an infinite run starting in some (not necessarily initial) configuration:

$$\forall \pi \exists \pi'.\ \varphi$$

where φ expresses that π' encodes a successor configuration of the configuration encoded by π.

Decidability

Theorem (Finkbeiner & Hahn '16)

1. $\exists^*$-*HyperLTL satisfiability is* PSpace-*complete.*
2. $\forall^*$-*HyperLTL satisfiability is* PSpace-*complete.*
3. $\exists^*\forall^*$-*HyperLTL satisfiability is* ExpSpace-*complete.*

Decidability

Theorem (Finkbeiner & Hahn '16)

1. $\exists^*$-*HyperLTL satisfiability is* PSpace-*complete.*
2. $\forall^*$-*HyperLTL satisfiability is* PSpace-*complete.*
3. $\exists^*\forall^*$-*HyperLTL satisfiability is* ExpSpace-*complete.*

Theorem (Mascle & Zimmermann '20)

1. *"Is there a model with $\leq k$ traces?" is* ExpSpace-*complete.*
2. *"Is there a model with ultimately periodic traces of length $\leq k$?" is* N2ExpTime-*complete.*
3. *"Is there a model represented by a transition system with $\leq k$ states?" is* Tower-*complete.*

Also: Decidability/better complexity for restricted nesting of temporal operators.

Model-Checking

The HyperLTL model-checking problem:

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

Theorem (Clarkson et al. '14)

The HyperLTL model-checking problem is decidable.

Corollary (Mascle & Z. '20)

The HyperLTL model-checking problem is TOWER-hard, even for a fixed transition system with 5 states and formulas without nested operators.

Outline

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

Team Semantics for LTL

Team semantics have been introduced to capture notions like dependence and independence in first-order logic.

- Novelty: evaluate formulas on sets (called teams) of variable assignments instead of a single assignment.

Team Semantics for LTL

Team semantics have been introduced to capture notions like dependence and independence in first-order logic.

- Novelty: evaluate formulas on sets (called teams) of variable assignments instead of a single assignment.

What about team semantics for (classical) LTL, i.e., evaluate formulas on sets of traces instead of traces?

Team Semantics for LTL

Team semantics have been introduced to capture notions like dependence and independence in first-order logic.

- Novelty: evaluate formulas on sets (called teams) of variable assignments instead of a single assignment.

What about team semantics for (classical) LTL, i.e., evaluate formulas on sets of traces instead of traces?

Theorem (Krebs, Meier, Virtema, Z. '18)

1. *TeamLTL satisfiability is decidable.*
2. *TeamLTL and HyperLTL are incomparable. In particluar, TeamLTL can express "There is an n such that $p \notin t(n)$ for every $t \in T$".*

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?
- Is there a temporal logic that is expressively equivalent to FO[$<$, E]?
- What about HyperCTL*?
- Quantitative hyperproperties
- Is TeamLTL model checking decidable?

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?
- Is there a temporal logic that is expressively equivalent to FO[$<$, E]?
- What about HyperCTL*?
- Quantitative hyperproperties
- Is TeamLTL model checking decidable?

Thank you