Deciding Hyperproperties Combined with Functional Specifications

Raven Beutner\textsuperscript{1}, David Carral\textsuperscript{3}, Bernd Finkbeiner\textsuperscript{1}, Jana Hofmann\textsuperscript{1}, Markus Krötzsch\textsuperscript{2}

\textsuperscript{1}CISPA Helmholtz Center for Information Security, Germany
\textsuperscript{2}TU Dresden, Germany
\textsuperscript{3}LIRMM, Inria, University of Montpellier, CNRS, France
Overview

- **Hyperproperties** describe many information flow policies like noninterference
- **HyperLTL** satisfiability is highly undecidable for $\forall^* \exists^*$ formulas
- 2 new perspectives: temporal safety/liveness + split in functional property and hyperproperty

| Temporal Safety                  | no LTL spec.                  | with LTL spec.                |
|---------------------------------|-------------------------------|-------------------------------|
| complete fragment               | coRE [Thm. 3.7]               | $\Sigma_1^\forall$ [Thm. 3.11]|
| $\forall^* \exists^* . O^*$     | NEXP [Thm. 3.12]              | NEXP [Thm. 3.12]              |
| $\forall^* \exists^* . \Box$    | NEXP [Lem. 3.13]              | $\Sigma_1^\forall$ [Thm. 3.11]|
| $\forall^* \exists^* . \Box(\Box^*)$ | coRE [Lem. 3.10]          | $\Sigma_1^\forall$ [Thm. 3.11]|

| Temporal Liveness               | no LTL spec.                  | with LTL spec.                |
|---------------------------------|-------------------------------|-------------------------------|
| complete fragment               | $\Sigma_1^\forall$ [Thm. 4.2] | $\Sigma_1^\forall$ [Thm. 4.2]|
| $\forall^* . \text{det-liveness}$ | trivial [Prop. 4.15]         | $\Sigma_1^\forall$ [Cor. 4.16]|
| $\forall^* . \Diamond(\Box^*)$ | NP [Lem. 4.4]                | dec. [Thm. 4.6]               |
| $\forall^* . \Diamond \wedge \cdots \wedge \Diamond$ | NP [Lem. 4.4]                | $\Sigma_1^\forall$ [Thm. 4.12]|

- Sound **algorithm** for largest models for $\forall \exists^*$ HyperLTL
Hyperproperties

**Trace property** \( P \): set of traces

\[
\text{system}
\]

\[
\begin{array}{c}
\circ \rightarrow \circ \rightarrow \circ \rightarrow \circ \rightarrow \circ \rightarrow \circ \rightarrow \cdots \\
\circ \rightarrow \circ \rightarrow \circ \rightarrow \circ \rightarrow \circ \rightarrow \circ \rightarrow \cdots \\
\circ \rightarrow \circ \rightarrow \circ \rightarrow \circ \rightarrow \circ \rightarrow \circ \rightarrow \cdots \\
\end{array}
\]

\( \text{trace} \in P? \checkmark \checkmark \checkmark 

Functional properties: safety, liveness, ...

**Hyperproperty** \(^1\) \( H \): set of sets of traces

\[
\text{system}
\]

\[
\begin{array}{c}
\circ \rightarrow \circ \rightarrow \circ \rightarrow \circ \rightarrow \circ \rightarrow \circ \rightarrow \cdots \\
\circ \rightarrow \circ \rightarrow \circ \rightarrow \circ \rightarrow \circ \rightarrow \circ \rightarrow \cdots \\
\circ \rightarrow \circ \rightarrow \circ \rightarrow \circ \rightarrow \circ \rightarrow \circ \rightarrow \cdots \\
\end{array}
\]

\( \text{system} \in H? \checkmark 

Relational properties:

“Does a change of inputs lead to a change of outputs?”

\(^1\) Clarkson, Schneider. *Hyperproperties*. CSF 2008.
HyperLTL\(^1\)

“Secret inputs do not interfere with publicly observable inputs and outputs.”

Generalized noninterference in HyperLTL:

\[
\forall \pi. \forall \pi'. \exists \pi''. \left( (\text{secretIn}_\pi = \text{secretIn}_{\pi''}) \land \right.
\]

\[
\square (\text{observableOut}_\pi' = \text{observableOut}_{\pi''})
\]

LTL with indexed atomic propositions

\(^1\)Clarkson, Finkbeiner, Koleini, Micinski, Rabe, Sánchez. *Temporal Logics for Hyperproperties*. POST 2014.
Satisfiability of $\exists^* \forall^*$ HyperLTL

$\exists^* \forall^*$: decidable\(^1\), in general: highly undecidable (in $\Sigma_1$)\(^2\)

Formulas with $\forall \exists$ quantifier alternation get **undecidable very quickly.** Undecidable are:

- $\forall^2 \exists^*$ + only $\square$ and $\Diamond$, not nested\(^2\)
- $\forall \exists^*$ + arbitrary temporal operators\(^1\)

$\Rightarrow$ No easier fragments obtained from syntactic restrictions

$\Rightarrow$ New perspectives on HyperLTL satisfiability?

\(^2\) Masle, Zimmermann. *The Keys to Decidable HyperLTL Satisfiability: Small Models or Very Simple Formulas.* CSL 2020.

\(^1\) Finkbeiner, Hahn. *Deciding Hyperproperties.* CONCUR 2016.
New Perspectives on $\forall^* \exists^*$ HyperLTL SAT

2 approaches:

1) Split hyperproperty into trace property + hyperproperty

   Trace property in LTL: describes functional behavior, e.g.: safety properties

   Hyperproperty in HyperLTL: simple relational property, e.g.: privacy properties

2) Semantic notion of temporal safety and temporal liveness

   Idea: especially safety properties have algorithmic advantages

   $\forall^* \exists^* . \psi$

   safety / liveness LTL formula
Temporal Safety vs Hypersafety

**Hypersafety**\(^1\): Does every counterexample have a “finite reason” for being a counterexample?

1) Whether HyperLTL formula \( \phi \) is hypersafety is highly undecidable (in \( \Pi_1 \))\(^2\)

2) If we know that \( \phi \) is hypersafety, deciding SAT is in PSPACE - no harder than LTL

---

\(^1\) Finkbeiner, Haas, Torfah. *Canonical Representations of k-Safety Hyperproperties*. CSF 2019.

\(^2\) Clarkson, Schneider. *Hyperproperties*. CSF 2008.
### Results

Reduction to first-order logic

| Temporal Safety | no LTL spec.                  | with LTL spec.                  |
|-----------------|-------------------------------|-------------------------------|
| complete fragment | coRE [Thm. 3.7]               | $\Sigma^1_1$ [Thm. 3.11]     |
| $\forall^* \exists^* \circ^*$ | NEXP [Thm. 3.12]               | NEXP [Thm. 3.12]               |
| $\forall^* \exists^* \square$ | NEXP [Lem. 3.13]               | $\Sigma^1_1$ [Thm. 3.11]     |
| $\forall^* \exists^* \square (\circ^*)$ | coRE [Lem. 3.10]               | $\Sigma^1_1$ [Thm. 3.11]     |

First decidability result for formulas that can enforce models with infinitely many traces

| Temporal Liveness | no LTL spec.                  | with LTL spec.                  |
|-------------------|-------------------------------|-------------------------------|
| complete fragment | $\Sigma^1_1$ [Thm. 4.2]       | $\Sigma^1_1$ [Thm. 4.2]       |
| $\forall^* \exists^* \text{ det-liveness}$ | trivial [Prop. 4.15]         | $\Sigma^1_1$ [Cor. 4.16]       |
| $\forall^* \exists^* \Diamond (\circ^*)$ | NP [Lem. 4.4]               | dec. [Thm. 4.6]              |
| $\forall^* \exists^* \Diamond \land \cdots \land \Diamond$ | NP [Lem. 4.4]               | $\Sigma^1_1$ [Thm. 4.12]     |
Finding Largest Models for $\forall \exists^* \text{ HyperLTL}$

```plaintext
1: procedure findModel($\mathcal{A}$)
2:     if $\mathcal{L}(\mathcal{A}^\forall) = \emptyset$ then
3:         return UNSAT;
4:     if $\mathcal{L}(\mathcal{A}^\exists) \subseteq \mathcal{L}(\mathcal{A}^\forall)$ then
5:         return SAT, model: $\mathcal{L}(\mathcal{A}^\forall)$;
6:     $\mathcal{A}_{\text{new}} := \mathcal{A} \cap \mathcal{A}_{\pi'}^\forall$;
7:     findModel($\mathcal{A}_{\text{new}}$);

$\forall \pi. \exists \pi'. \Box(a_\pi \land (b_\pi \leftrightarrow \bigcirc b_{\pi'}))$
```

Remove $\forall$-traces that produce wrong $\exists$-traces
Finding Largest Models for $\forall \exists^* \text{ HyperLTL}$

1: procedure findModel($\mathcal{A}$)
2:   if $\mathcal{L}(\mathcal{A}^\forall) = \emptyset$ then
3:     return UNSAT;
4:   if $\mathcal{L}(\mathcal{A}^\exists) \subseteq \mathcal{L}(\mathcal{A}^\forall)$ then
5:     return SAT, model: $\mathcal{L}(\mathcal{A}^\forall)$;
6:   $\mathcal{A}_{\text{new}} := \mathcal{A} \cap \mathcal{A}^\forall_{\pi'}$;
7:   findModel($\mathcal{A}_{\text{new}}$);

$\forall \pi. \exists \pi'. \Box (a_\pi \land (b_\pi \leftrightarrow \bigcirc b_{\pi'}))$

- Finds largest models
- Sound but necessarily not complete
- Evaluation: finds models that MGHyper\textsuperscript{1} does not find, can show unsatisfiability

\textsuperscript{1}Finkbeiner, Hahn, Hans. MGHyper: Checking Satisfiability of HyperLTL formulas beyond the $\exists^*\forall^*$ Fragment. ATVA 2018.
Conclusion

• Syntactic fragments of $\forall^* \exists^*$ HyperLTL do not make satisfiability easier

• 2 new perspectives: temporal safety/liveness + split in functional property and hyperproperty

| Temporal Safety | no LTL spec. | with LTL spec. |
|----------------|-------------|--------------|
| $\forall^* \exists^* \cdot O^*$ | coRE [Thm. 3.7] | $\Sigma^1_1$ [Thm. 3.11] |
| $\forall^* \exists^* \cdot \Box$ | NEXP [Thm. 3.12] | NEXP [Thm. 3.12] |
| $\forall^* \exists^* \cdot (\Box^*)$ | coRE [Lem. 3.10] | $\Sigma^1_1$ [Thm. 3.11] |

New $\forall^* \exists^*$ decidability results beyond purely syntactic restrictions

Fixpoint algorithm for SAT and UNSAT

| Temporal Liveness | no LTL spec. | with LTL spec. |
|------------------|--------------|--------------|
| $\exists^* \cdot \text{det-liveness}$ | trivial [Prop. 4.15] | $\Sigma^1_1$ [Cor. 4.16] |
| $\forall^* \cdot (\Diamond^*)$ | NP [Lem. 4.4] | dec. [Thm. 4.6] |
| $\forall^* \exists^* \cdot \Diamond \wedge \cdots \wedge \Diamond$ | NP [Lem. 4.4] | $\Sigma^1_1$ [Thm. 4.12] |

• Sound algorithm for largest models for $\forall^* \exists^*$ HyperLTL