The Keys to Decidable HyperLTL Satisfiability: Small Models or Very Simple Formulas

Corto Mascle Martin Zimmermann

ENS Paris-Saclay University of Liverpool

February 25th - July 26th

Hyperproperties

Properties characterize executions of a system:
→ "The boolean variable b will eventually be true" is a property.

Hyperproperties [Clarkson & Schneider, '08] characterize the set of executions of a system:
→ "For every execution in which b is eventually true, there exists an execution in which b is true later" is a hyperproperty.

A hyperproperty for security

We consider two boolean variables a and b.

"For all executions, there exists another execution with the same behaviour for a but a different one for b"

This hyperproperty expresses that someone having access to the values of a will not be able to infer the value of b from it.

LTL

LTL is a logics on infinite *traces*, i.e., infinite sequences of sets of atomic propositions, like $\{a\}\emptyset\{a\}\{a\}\emptyset\emptyset\cdots$

LTL

LTL is a logics on infinite *traces*, i.e., infinite sequences of sets of atomic propositions, like $\{a\}\emptyset\{a\}\{a\}\emptyset\emptyset\cdots$

It combines:

- Boolean operators $\wedge$, $\vee$, $\neg$
- Temporal operators **F** , **G** , **U** , **X**

LTL semantics

Given a formula φ,

- $\mathbf{F}\,\varphi$ means that φ is satisfied at some further position.

  $\emptyset\emptyset\{a\}\emptyset\emptyset\cdots$ satisfies $\mathbf{F}\,a$
- $\mathbf{G}\,\varphi$ means that φ is satisfied on every further position.

  $\{a\}\{a\}\{a\}\cdots$ satisfies $\mathbf{G}\,a$
- $\varphi\,\mathbf{U}\,\psi$ means that ψ is satisfied at some further position and φ is satisfied at every position in-between.

  $\{b\}\{b\}\{a\}\emptyset\emptyset\cdots$ satisfies $b\,\mathbf{U}\,a$
- $\mathbf{X}\,\varphi$ means that φ is satisfied at the next position.

  $\emptyset\{a\}\emptyset\emptyset\cdots$ satifies $\mathbf{X}\,a$.

Another example

$\mathbf{F}\,\mathbf{G}\,(b \wedge \neg a)$

is satisfied by $\{a\}\{a\}\emptyset\{b\}\{b\}\{b\}\cdots$

but not by $\{a\}\{b\}\{a, b\}\{a, b\}\{a, b\}\cdots$

What we want to express

LTL allows us to express properties about **single executions** of a system, but not about the **set of executions** of a system (hyperproperties).

HyperLTL

Syntax:

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

Formulas are evaluated over sets of infinite traces.

$$\forall\tau.\exists\tau'.\mathbf{F}\,(a_\tau \wedge b_{\tau'})$$

For all τ in the model, there exists τ' in the model such that:

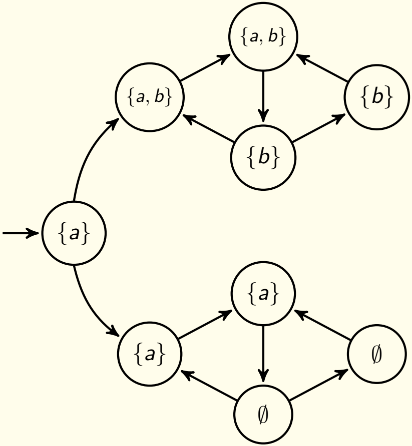
$\tau \;\rightarrow$	$\emptyset$	$\{a\}$	$\cdots$	$\{a\}$	$\cdots$
$\tau' \;\rightarrow$	$\{b\}$	$\emptyset$	$\cdots$	$\{b\}$	$\cdots$

An example

$$\forall \tau . \exists \tau' . \mathbf{G}\,(a_\tau \Leftrightarrow a_{\tau'}) \wedge \mathbf{F}\,\neg(b_\tau \Leftrightarrow b_{\tau'})$$

An example

$$\forall \tau . \exists \tau' . \mathbf{G}\,(a_\tau \Leftrightarrow a_{\tau'}) \wedge \mathbf{F}\,\neg(b_\tau \Leftrightarrow b_{\tau'})$$



Model-checking

Theorem (Clarkson, Finkbeiner, Koleini, Micinski, Rabe, Sánchez)

Model-checking HyperLTL formulas against Kripke structures is decidable, but TOWER*-complete.*

The complexity is a tower of exponentials of height the number of quantifier alternations.

For instance, checking that a Kripke structure satisfies a formula of the form $\forall^*\exists^*\forall^*\psi$ requires space $2^{2^{|\psi|}}$

Satisfiability

Theorem (Finkbeiner & Hahn)

HyperLTL satisfiability is undecidable.

One can encode executions of Turing machines with formulas of the form $\forall\exists$.

This motivates the search for fragments of HyperLTL with decidable satisfiability.

We still want to use this convenient syntax, so we look for decidable syntactical fragments.

Previous results

Theorem (Finkbeiner & Hahn)

Satisfiability is

- PSPACE *for formulas of the form* $\forall^*$ *or* $\exists^*$
- EXPSPACE *for formulas of the form* $\exists^*\forall^*$
- *Undecidable for formulas of the form* $\forall\exists$

Theorem (Demri & Schnoebelen)

The complexity of LTL satisfiability decreases when some bounds are applied on the temporal depth, the set of operators and/or the number of atomic propositions.

Small formulas

Some of the interesting restrictions are temporal depth and alternation depth.

- $\mathrm{td}(a_\pi) = 0$
- $\mathrm{td}(\neg\psi) = \mathrm{td}(\psi)$
- $\mathrm{td}(\psi_1 \vee \psi_2) = \max(\mathrm{td}(\psi_1), \mathrm{td}(\psi_2))$,
- $\mathrm{td}(\mathbf{X}\,\psi) = 1 + \mathrm{td}(\psi)$,
- $\mathrm{td}(\psi_1 \,\mathbf{U}\, \psi_2) = 1 + \max(\mathrm{td}(\psi_1), \mathrm{td}(\psi_2))$,
- $\mathrm{td}(\exists\pi.\varphi) = \mathrm{td}(\forall\pi.\varphi) = \mathrm{td}(\varphi)$

Most examples of security policies expressible in HyperLTL have temporal depth one.

Simplifying formulas

We can reduce the general satisfiability problem to the one on "small" formulas:

For any HyperLTL formula one can compute in polynomial time an equisatisfiable formula with:

- One quantifier alternation,
- Temporal depth two,
- Two universal quantifiers or three atomic propositions.

The decidability border

We look at formulas with temporal depth one, one universal quantifier, and using only **F** and **G**. For example,

$$\forall\tau.\exists\tau'.\mathbf{G}\,(a_\tau \Leftrightarrow a_{\tau'}) \wedge \mathbf{F}\,\neg(b_\tau \Leftrightarrow b_{\tau'})$$

Satisfiability for this fragment is decidable!

An overview

	temporal depth one	arbitrary temporal depth
$\exists^*$ / $\forall^*$	NP-complete	PSPACE-complete
$\exists^*\forall^*$	NEXPTIME-complete	EXPSPACE-complete
$\exists^*\forall\exists^*$	N2EXPTIME (without **U**)	undecidable
$\forall^2\exists^*$	undecidable	undecidable

Small models

Instead of restricting formulas, we can restrict models. Given a formula φ and an integer (in binary) k:

- Models with at most k elements
  $\rightarrow$ EXPSPACE-complete

Small models

Instead of restricting formulas, we can restrict models. Given a formula φ and an integer (in binary) k:

- Models with at most k elements
  $\rightarrow$ EXPSPACE-complete
- Models in which all words are of the form uv^ω with $|u| + |v| \leq k$
  $\rightarrow$ N2EXPTIME-complete

Small models

Instead of restricting formulas, we can restrict models. Given a formula φ and an integer (in binary) k:

- Models with at most k elements
  $\rightarrow$ EXPSPACE-complete
- Models in which all words are of the form uv^ω with $|u| + |v| \leq k$
  $\rightarrow$ N2EXPTIME-complete
- Models represented by a Kripke structure with k states
  $\rightarrow$ TOWER-complete

Further work: Kripke structures

We only consider sets of traces generated by finite Kripke structures.

Satisfiability over Kripke structures is undecidable in general, but semi-decidable.

It is TOWER-hard even for formulas of the form $\forall^*\exists^*$ with temporal depth 1, but may be decidable with suitable restrictions on the formulas.

Conclusion

$\rightarrow$ Better understanding of the expressivity of HyperLTL

$\rightarrow$ More precise decidability border

$\rightarrow$ Many more fundamental problems to explore (other parameters on formulas and models)

Thank you!