

# Deciding Hyperproperties

---

Bernd Finkbeiner and Christopher Hahn

Reactive Systems Group

Saarland University, Germany

Highlights of Logic, Games and Automata

Brussels, 06.-09. September 2016

# Information Leakage



- **Heartbleed** - 4.5m patient information leaked
- **Goto Fail** - encryption of >300m devices broken
- **Shellshock** - web servers attackable for 22 years

# HyperLTL - A Logic for Information-flow Control

[Clarkson, Finkbeiner, Koleini, Micinski, Rabe, Sánchez, '14]

- **Observational Determinism:** “Program appears deterministic to low security users.”

$$\forall \pi. \forall \pi'. \Box(I_{\pi} = I_{\pi'}) \rightarrow \Box(O_{\pi} = O_{\pi'})$$

- **Generalized Noninterference:** “. . . additionally low-security outputs may not be altered by injection of high-security inputs.”

$$\forall \pi. \forall \pi'. \exists \pi''. \Box(HighI_{\pi} = HighI_{\pi''}) \wedge \Box(O_{\pi'} = O_{\pi''})$$

# HyperLTL - An Extension of LTL

## LTL

- logical connectives:  $\vee, \neg$   
temporal connectives:  
 $\square$  - *globally*  
 $\bigcirc$  - *next*
- $\square a$  is satisfied by  $\{a\}^\omega$  as well as  $\{a, b\}^\omega$
- $\square a \wedge \square \neg a$  is unsatisfiable.
- defines a set of computation traces  
(trace property)

## HyperLTL

- LTL + explicit trace quantifiers:  
Observational Determinism:  
$$\forall \pi. \forall \pi'. \square (I_\pi = I_{\pi'}) \rightarrow \square (O_\pi = O_{\pi'})$$
- $\exists \pi. \exists \pi'. \square a_\pi \wedge \square \neg a_{\pi'}$   
is satisfiable by  $\{\{a\}^\omega, \{b\}^\omega\}$ .
- defines a **set of sets** of computation traces  
(hyperproperty)

# Satisfiability of HyperLTL

## Definition (HyperLTL-SAT)

Let  $\varphi$  be an HyperLTL formula. HyperLTL-SAT is the problem to decide whether there exists a **non-empty** trace set  $T$  satisfying  $\varphi$ .

## Example (Application)

Two versions of Observational Determinism:

- $\forall \pi. \forall \pi'. \Box(I_\pi = I_{\pi'}) \rightarrow \Box(O_\pi = O_{\pi'})$
- $\forall \pi. \forall \pi'. (I_\pi = I_{\pi'}) \rightarrow \Box(O_\pi = O_{\pi'})$

Which variation is stronger?

# Challenge

## LTL Satisfiability Solving

- Translate LTL formula into Büchi automaton
- Check the automaton for emptiness
- PSPACE-complete

## HyperLTL Satisfiability Solving

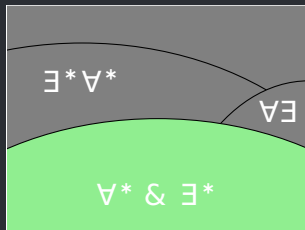
- A Hyperproperty is not necessarily  $\omega$ -regular
- Standard automata approach cannot be applied

## Key Results

[Finkbeiner, H., '16]

- HyperLTL-SAT is PSPACE-complete for alternation-free formulas
- HyperLTL-SAT is EXPSPACE-complete for  $\exists^* \forall^*$  formulas
- HyperLTL-SAT is undecidable for  $\forall \exists$  formulas

## Outline - Solving HyperLTL-SAT



1. Alternation-free fragments ( $\forall^* \& \exists^*$ )
2. Alternation starting with existential quantifier ( $\exists^* \forall^*$ )
3. Alternation starting with universal quantifier ( $\forall^* \exists^*$ )



# Existential Fragment

## Theorem

$\exists^*$  HyperLTL-SAT is PSPACE-complete.

## Example

$\exists \pi_0 \exists \pi_1. \Box a_{\pi_0} \wedge \Box b_{\pi_0} \wedge \Box c_{\pi_0} \wedge \Box a_{\pi_1} \wedge \Box \neg c_{\pi_1}$

Idea:

Replace indexed atomic propositions with fresh atomic propositions.

$\Box a_0 \wedge \Box b_0 \wedge \Box c_0 \wedge \Box a_1 \wedge \Box \neg c_1$

# Existential Fragment

## Theorem

$\exists^*$  HyperLTL-SAT is PSPACE-complete.

## Example

$\exists \pi_0 \exists \pi_1. \Box a_{\pi_0} \wedge \Box b_{\pi_0} \wedge \Box c_{\pi_0} \wedge \Box a_{\pi_1} \wedge \Box \neg c_{\pi_1}$

Idea:

Replace indexed atomic propositions with fresh atomic propositions.

$$\Box a_0 \wedge \Box b_0 \wedge \Box c_0 \wedge \Box a_1 \wedge \Box \neg c_1$$
$$t : \{a_0, b_0, c_0, a_1\}^\omega$$

# Existential Fragment

## Theorem

$\exists^*$  HyperLTL-SAT is PSPACE-complete.

## Example

$\exists \pi_0 \exists \pi_1. \Box a_{\pi_0} \wedge \Box b_{\pi_0} \wedge \Box c_{\pi_0} \wedge \Box a_{\pi_1} \wedge \Box \neg c_{\pi_1}$

Idea:

Replace indexed atomic propositions with fresh atomic propositions.

$$\Box a_0 \wedge \Box b_0 \wedge \Box c_0 \wedge \Box a_1 \wedge \Box \neg c_1$$

$$t : \{a_0, b_0, c_0, a_1\}^\omega$$

$$T = \{\{a, b, c\}^\omega, \{a\}^\omega\}$$

# Universal Fragment

## Theorem

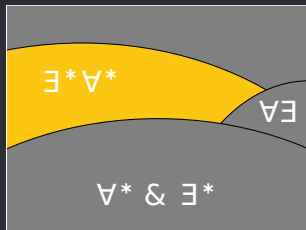
$\forall^*$  HyperLTL-SAT is PSPACE-complete.

## Example

$$\begin{array}{ccc} \forall \pi \forall \pi'. \Box b_\pi \wedge \Box \neg b_{\pi'} & \equiv & \Box b \wedge \Box \neg b \\ \Downarrow \quad \Downarrow & & \Downarrow \\ t \quad t & & \text{unsatisfiable} \end{array}$$

Idea: Discard indexes from indexed propositions

## Outline - Solving HyperLTL-SAT



1. Alternation-free fragments ( $\forall^* \exists^*$ )
2. Alternation starting with existential quantifier ( $\exists^* \forall^*$ )
3. Alternation starting with universal quantifier ( $\forall^* \exists^*$ )

# $\exists^* \forall^*$ HyperLTL-SAT

## Lemma

For every  $\exists \pi_1 \dots \exists \pi_n \forall \pi'_1 \dots \forall \pi'_m . \varphi$  HyperLTL formula, there exists an equisatisfiable  $\exists^*$  HyperLTL formula.

## Example

$$\exists \pi_0 \exists \pi_1 \forall \pi'_0 \forall \pi'_1. (\Box a_{\pi'_0} \wedge \Box b_{\pi'_1}) \wedge (\Box c_{\pi_0} \wedge \Box d_{\pi_1})$$

Idea: Unroll universal quantifiers

$$\begin{aligned} \exists \pi_0 \exists \pi_1. & (\Box a_{\pi_0} \wedge \Box b_{\pi_0}) \wedge (\Box c_{\pi_0} \wedge \Box d_{\pi_1}) \\ & \wedge (\Box a_{\pi_1} \wedge \Box b_{\pi_0}) \wedge (\Box c_{\pi_0} \wedge \Box d_{\pi_1}) \\ & \wedge (\Box a_{\pi_0} \wedge \Box b_{\pi_1}) \wedge (\Box c_{\pi_0} \wedge \Box d_{\pi_1}) \\ & \wedge (\Box a_{\pi_1} \wedge \Box b_{\pi_1}) \wedge (\Box c_{\pi_0} \wedge \Box d_{\pi_1}) \end{aligned}$$

## Complexity of $\exists^* \forall^*$ HyperLTL-SAT

### Theorem

*Let  $n$  be the number of existential quantifier and  $m$  be the number of universal quantifier.  $\exists^* \forall^*$  HyperLTL-SAT is EXPSPACE-complete.*

- Unrolling results in formula of size  $O(n^m)$ .
- Hardness follows from an encoding of an EXPSPACE-bounded Turing machine in this fragment.

# $\exists^* \forall^b$ HyperLTL-SAT

## Theorem

*Bounded  $\exists^* \forall^b$  HyperLTL-SAT is PSPACE-complete.*

Observation: In practice, many properties of interest quantify universally over pairs of traces

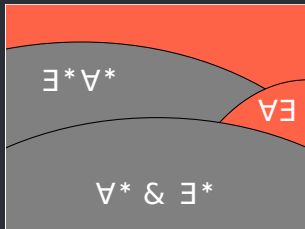
$$\forall \pi. \forall \pi'. \square(I_\pi = I_{\pi'}) \rightarrow \square(O_\pi = O_{\pi'})$$

$$\forall \pi. \forall \pi'. (I_\pi = I_{\pi'}) \rightarrow \square(O_\pi = O_{\pi'})$$

$$\forall \pi. \forall \pi'. \exists \pi''. \square(\text{High}I_\pi = \text{High}I_{\pi''}) \wedge \square(O_{\pi'} = O_{\pi''})$$



## Outline - Solving HyperLTL-SAT



1. Alternation-free fragments ( $\forall^* \& \exists^*$ )
2. Alternation starting with existential quantifier ( $\exists^* \forall^*$ )
3. Alternation starting with universal quantifier ( $\forall^* \exists^*$ )

## The Power of $\forall\exists$ - Encoding of PCP

- Can give a  $\forall\exists$  HyperLTL formula, which is only satisfied by an infinite trace set:

$$\forall\pi\exists\pi'. a_{\pi'} \tag{1}$$

$$\wedge \Box(a_{\pi} \rightarrow \bigcirc\Box\neg a_{\pi}) \tag{2}$$

$$\wedge \Box(a_{\pi} \rightarrow \bigcirc a_{\pi'}) \tag{3}$$

- Encoding of Posts Correspondence Problem (PCP) in this fragment.

### Theorem

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

## Summary & Conclusion

$\exists^*$	$\forall^*$	$\exists^* \forall^*$	$b$ -Bounded $\exists^* \forall^*$	$\forall \exists$
<b>PSpace- complete</b>	<b>PSpace- complete</b>	<b>EXPSpace- complete</b>	<b>PSpace- complete</b>	undecidable

- Satisfiability of alternation-free formulas is decidable
- Implication and equivalence of alternation-free formulas are decidable
- Full logic is undecidable:  
HyperLTL is much more powerful than LTL

Christopher Hahn: [hahn@react.uni-saarland.de](mailto:hahn@react.uni-saarland.de)

# Appendix

## Bibliography

[Clarkson, Schneider, '10] Clarkson, M. R., and F. B. Schneider. "Hyperproperties." *Journal of Computer Security* 18.6 (2010): 1157-1210.

[Clarkson, Finkbeiner, Koleini, Micinski, Rabe, Sánchez, '14]

Clarkson, M. R., Finkbeiner, B., Koleini, M., Micinski, K. K., Rabe, M. N., & Sánchez, C. (2014, April). Temporal logics for hyperproperties. In *International Conference on Principles of Security and Trust* (pp. 265-284).

[Finkbeiner, H., '16] Bernd Finkbeiner and Christopher Hahn. Deciding hyperproperties. In *International Conference on Concurrency Theory* (2016).

Picture: [http://russia-insider.com/sites/insider/files/20110226\\_bbd001\\_0.jpg](http://russia-insider.com/sites/insider/files/20110226_bbd001_0.jpg)

# HyperLTL Syntax

## Syntax

$$\psi ::= \exists \pi. \psi \mid \forall \pi. \psi \mid \varphi$$
$$\varphi ::= a_\pi \mid \neg \varphi \mid \varphi \vee \varphi \mid \bigcirc \varphi \mid \varphi \mathcal{U} \varphi$$

- Quantifier Prefix with arbitrary alternation
- Then quantifier-free LTL formula with trace variables
- $\wedge, \rightarrow, \leftrightarrow, \square, \diamond$  derived in the usual way
- $X_\pi = X_{\pi'}$  syntactic sugar for  $\bigwedge_{x \in X} (x_\pi \leftrightarrow x_{\pi'})$

## Example

“All executions have the light on at the same time.”

$$\forall \pi. \forall \pi'. \square (on_\pi \leftrightarrow on_{\pi'})$$

# HyperLTL Semantics

Semantics w.r.t. Trace Environment  $\Pi : \text{Var} \mapsto \text{TR}$

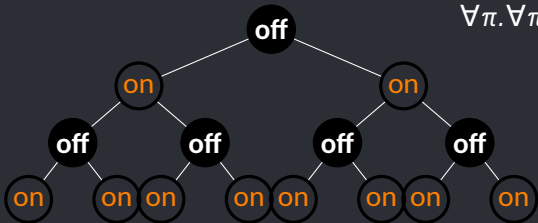
$\Pi \models_T \forall \pi. \varphi$  iff for all  $t \in T$ , s.t.  $\Pi[\pi \mapsto t] \models_T \varphi$

$\Pi \models_T a_\pi$  iff  $a \in \Pi(\pi)[0]$

$\Pi \models_T \Box \varphi$  iff  $\forall i \geq 0 : \Pi[i, \infty] \models_T \varphi$

“All executions have the light on at the same time.”

$\forall \pi. \forall \pi'. \Box (on_\pi \leftrightarrow on_{\pi'})$



# HyperLTL Semantics

Semantics w.r.t. Trace Environment  $\Pi : \text{Var} \mapsto \text{TR}$

$\Pi \models_T \forall \pi. \varphi$  iff for all  $t \in T$ , s.t.  $\Pi[\pi \mapsto t] \models_T \varphi$

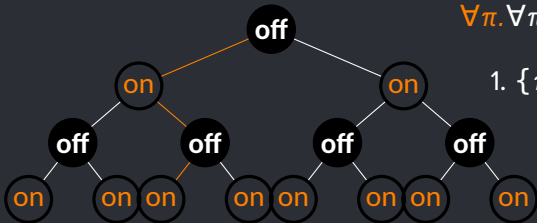
$\Pi \models_T a_\pi$  iff  $a \in \Pi(\pi)[0]$

$\Pi \models_T \Box \varphi$  iff  $\forall i \geq 0 : \Pi[i, \infty] \models_T \varphi$

“All executions have the light on at the same time.”

$\forall \pi. \forall \pi'. \Box (on_\pi \leftrightarrow on_{\pi'})$

1.  $\{\pi \mapsto t\} \models_M \forall \pi'. \Box(\dots)$





# HyperLTL Semantics

Semantics w.r.t. Trace Environment  $\Pi : Var \mapsto TR$

$\Pi \models_T \forall \pi. \varphi$  iff for all  $t \in T$ , s.t.  $\Pi[\pi \mapsto t] \models_T \varphi$

$\Pi \models_T a_\pi$  iff  $a \in \Pi(\pi)[0]$

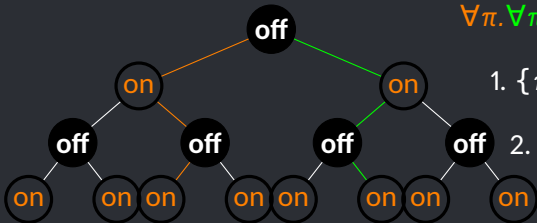
$\Pi \models_T \Box \varphi$  iff  $\forall i \geq 0 : \Pi[i, \infty] \models_T \varphi$

“All executions have the light on at the same time.”

$\forall \pi. \forall \pi'. \Box (on_\pi \leftrightarrow on_{\pi'})$

1.  $\{\pi \mapsto t\} \models_M \forall \pi'. \Box(\dots)$

2.  $\{\pi \mapsto t, \pi' \mapsto t'\} \models_T \Box(\dots)$



# HyperLTL Semantics

Semantics w.r.t. Trace Environment  $\Pi : Var \mapsto TR$

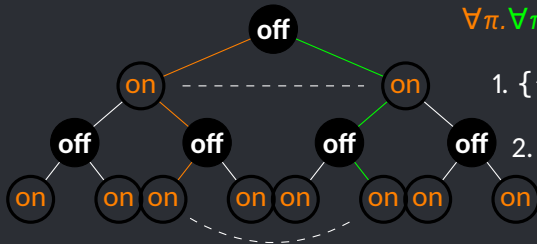
$\Pi \models_T \forall \pi. \varphi$  iff for all  $t \in T$ , s.t.  $\Pi[\pi \mapsto t] \models_T \varphi$

$\Pi \models_T a_\pi$  iff  $a \in \Pi(\pi)[0]$

$\Pi \models_T \Box \varphi$  iff  $\forall i \geq 0 : \Pi[i, \infty] \models_T \varphi$

“All executions have the light on at the same time.”

$\forall \pi. \forall \pi'. \Box(on_\pi \leftrightarrow on_{\pi'})$



1.  $\{\pi \mapsto t\} \models_M \forall \pi'. \Box(\dots)$

2.  $\{\pi \mapsto t, \pi' \mapsto t'\} \models_T \Box(\dots)$

3.  $\forall i \geq 0 : \{\pi \mapsto t[i, \infty], \pi' \mapsto t'[i, \infty]\} \models_T on_\pi \leftrightarrow on_{\pi'}$

# Encoding of Posts Correspondence Problem

Example PCP instance:

I	II	III
a	ab	bba
baa	aa	bb

# Encoding of Posts Correspondence Problem

Example PCP solution:

III	II	III	I
bba	ab	bba	a
bb	aa	bb	baa

HyperLTL encoding:

1. exists a “solution”-trace  $\pi_s$ ,  
where top matches bottom

# Encoding of Posts Correspondence Problem

Example PCP solution:

III	II	III	I
bba	ab	bba	a
bb	aa	bb	baa

$\pi_S$

b b a a b b b a a # # ...  
b b a a b b b a a # # ...

HyperLTL encoding:

1. exists a “solution”-trace  $\pi_S$ ,  
where top matches bottom

# Encoding of Posts Correspondence Problem

Example PCP solution:

III	II	III	I
bba	ab	bba	a
bb	aa	bb	baa

$\pi_S$ 

b	b	a		a	b	b	b	a	a	#	#	...
b	b	a		a	b	b	b	a	a	#	#	...
b	b	a		a	b	b	b	a	a	#	#	...

HyperLTL encoding:

1. exists a “solution”-trace  $\pi_S$ , where top matches bottom
2. every trace starts with a valid stone

## Encoding of Posts Correspondence Problem

Example PCP solution:

III	II	III	I
bba	ab	bba	a
bb	aa	bb	baa

$\pi_s$ 

b	b	a		a	b	b	b	a	a	#	#	...
b	b	a		a	b	b	b	a	a	#	#	...
b	b	a		a	b	b	b	a	a	#	#	...

HyperLTL encoding:

1. exists a “solution”-trace  $\pi_s$ , where top matches bottom
2. every trace starts with a valid stone
3. for every trace, there exists another without the first stone

# Encoding of Posts Correspondence Problem

Example PCP solution:

III	II	III	I
bba	ab	bba	a
bb	aa	bb	baa

HyperLTL encoding:

1. exists a “solution”-trace  $\pi_s, \pi'_s$   
where top matches bottom
2. every trace starts with a valid stone
3. for every trace, there exists another without the first stone

$\pi_s$ 

b	b	a		a	b	b	b	a	a	#	#	...
b	b	a		a	b	b	b	a	a	#	#	...
			III									

a	b	b	b	a	a	#	#	...
a	a	b	b	b	a	a	#	...



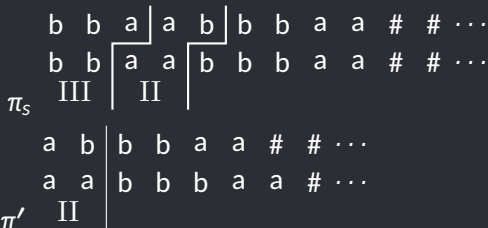
# Encoding of Posts Correspondence Problem

Example PCP solution:

III	II	III	I
bba	ab	bba	a
bb	aa	bb	baa

HyperLTL encoding:

1. exists a “solution”-trace  $\pi_s$ , where top matches bottom
2. every trace starts with a valid stone
3. for every trace, there exists another without the first stone



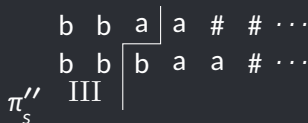
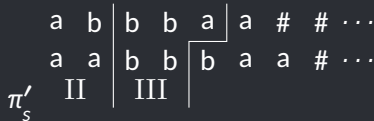
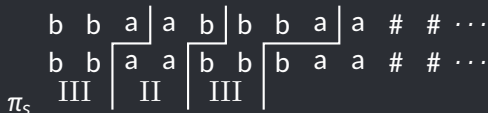
# Encoding of Posts Correspondence Problem

Example PCP solution:

III	II	III	I
bba	ab	bba	a
bb	aa	bb	baa

HyperLTL encoding:

1. exists a “solution”-trace  $\pi_s$ , where top matches bottom
2. every trace starts with a valid stone
3. for every trace, there exists another without the first stone



# Encoding of Posts Correspondence Problem

Example PCP solution:

III	II	III	I
bba	ab	bba	a
bb	aa	bb	baa

HyperLTL encoding:

1. exists a “solution”-trace  $\pi_s$ , where top matches bottom
2. every trace starts with a valid stone
3. for every trace, there exists another without the first stone

