Temporal Logics for the Specification of Hyperproperties

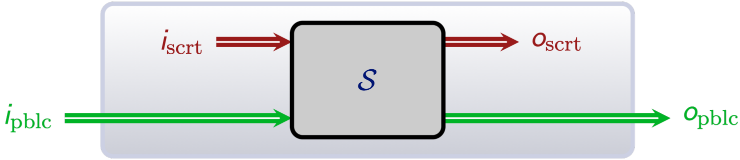
Martin Zimmermann

Aalborg University

October 18th, 2022

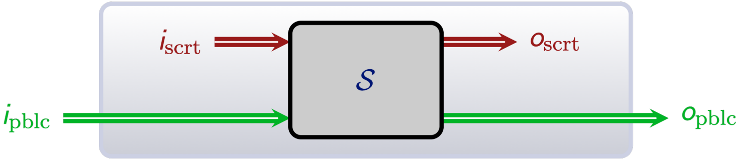
NetVAS 22

Motivation



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

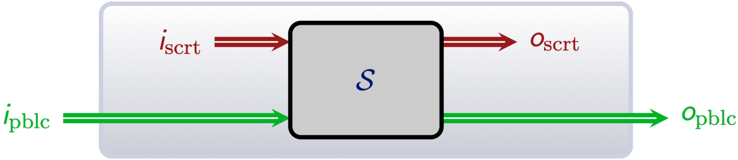
Motivation



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

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

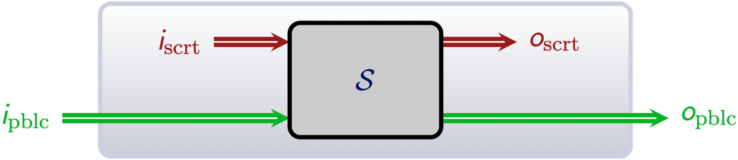
Motivation



Typical requirements:

- $\mathcal{S}$ terminates

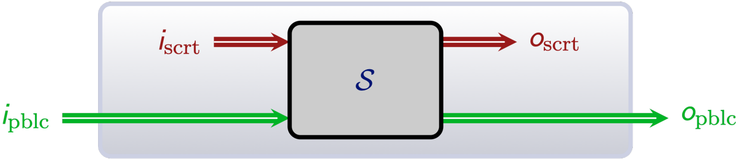
Motivation



Typical requirements:

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

Motivation

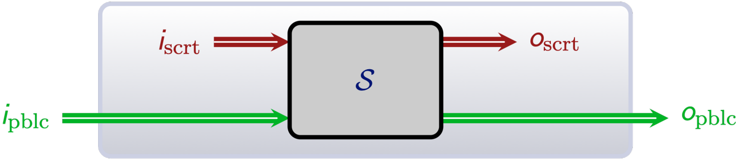


Typical requirements:

- $\mathcal{S}$ 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.

LTL in One Slide

Syntax

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

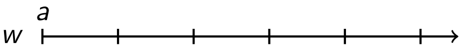
LTL in One Slide

Syntax

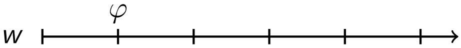
$$\varphi ::= a \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathbf{X}\,\varphi \mid \varphi\,\mathbf{U}\,\varphi \qquad \text{where } a \in \mathrm{AP}$$

Semantics

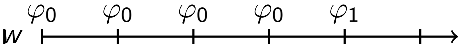
- $w \models a$:



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



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



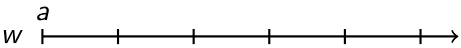
LTL in One Slide

Syntax

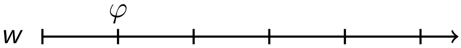
$$\varphi ::= a \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathbf{X}\,\varphi \mid \varphi\,\mathbf{U}\,\varphi \qquad \text{where } a \in \mathrm{AP}$$

Semantics

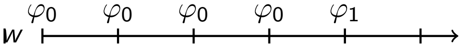
- $w \models a$:



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



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



Syntactic Sugar

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

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.

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 $\mathcal{S}$

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

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

- $\mathcal{S}$ 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
- Network verification?

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

Another Example

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

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

Another Example

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

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

Another Example

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

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

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

Another Example

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

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

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

Another Example

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

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

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

Another Example

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

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

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

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

Another Example

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 $\{\emptyset^n \{a\} \emptyset^\omega \mid n \in \mathbb{N}\}$.

Consequence:

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

Undecidability

The HyperLTL satisfiability problem:

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

Theorem (Fortin et. al '21)

HyperLTL satisfiability is Σ^1_1-complete (i.e., highly undecidable).

Undecidability

The HyperLTL satisfiability problem:

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

Theorem (Fortin et. al '21)

HyperLTL satisfiability is Σ_1^1-complete (i.e., highly undecidable).

Fine-grained analysis:

Theorem (Finkbeiner & Hahn '16)

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

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.

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.

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
- Soon: Network verification