Controller Synthesis for Hyperproperties

Borzoo Bonakdarpour

MICHIGAN STATE UNIVERSITY

This is joint work with

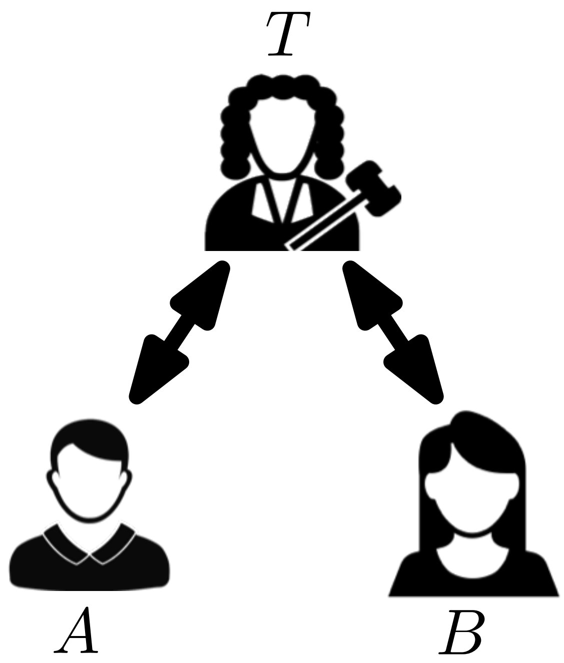
Bernd Finkbeiner

- Borzoo Bonakdarpour, Bernd Finkbeiner, *Controller Synthesis for Hyperproperties* The 33rd IEEE International Symposium on Computer Security Foundations (CSF), 2020

1. Motivation

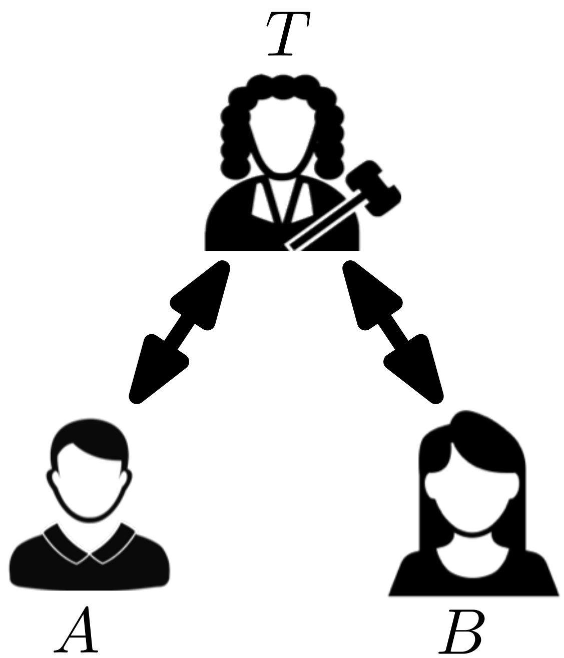
Motivation

- The purpose of a *non-repudiation* protocol is to allow two parties A and B to exchange messages through a *trusted third party* T without any party being able to deny having participated in the exchange.



Motivation

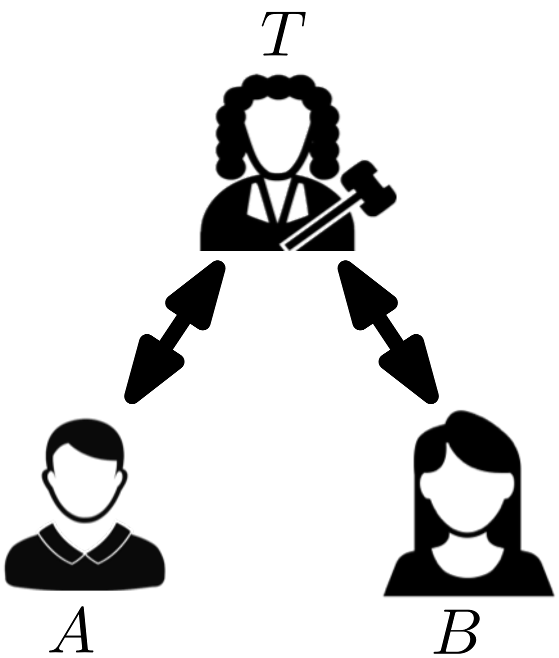
- The purpose of a *non-repudiation* protocol is to allow two parties A and B to exchange messages through a *trusted third party* T without any party being able to deny having participated in the exchange.



- The recipient of the message obtains an *NRO* evidence.

Motivation

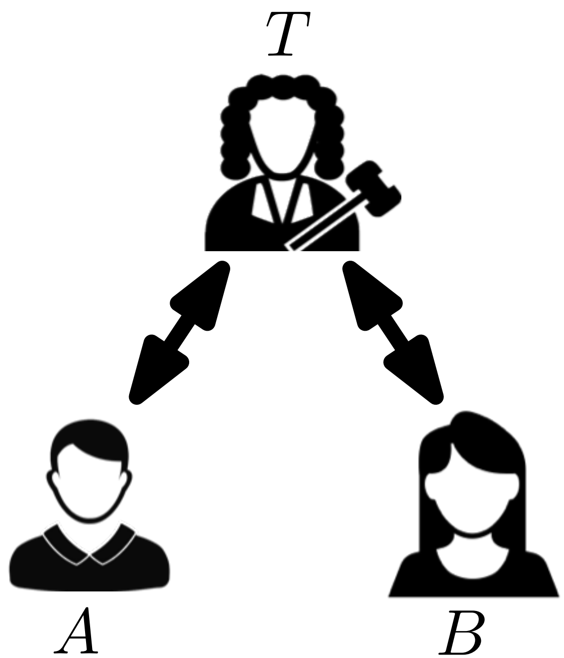
- The purpose of a *non-repudiation* protocol is to allow two parties A and B to exchange messages through a *trusted third party* T without any party being able to deny having participated in the exchange.



- The recipient of the message obtains an *NRO* evidence.
- The sender of the message obtains an *NRR* evidence.

Motivation

- The purpose of a *non-repudiation* protocol is to allow two parties A and B to exchange messages through a *trusted third party* T without any party being able to deny having participated in the exchange.

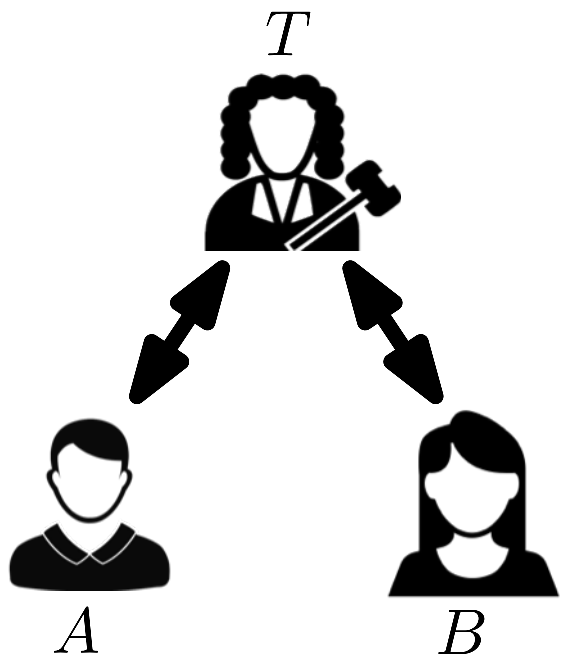


- The recipient of the message obtains an *NRO* evidence.
- The sender of the message obtains an *NRR* evidence.

- The protocol is *effective* if it is possible to successfully transmit the message to the recipient and the evidence to both parties.

Motivation

- The purpose of a *non-repudiation* protocol is to allow two parties A and B to exchange messages through a *trusted third party* T without any party being able to deny having participated in the exchange.



- The recipient of the message obtains an *NRO* evidence.
- The sender of the message obtains an *NRR* evidence.

- The protocol is *effective* if it is possible to successfully transmit the message to the recipient and the evidence to both parties.
- The protocol is *fair* if it is *impossible* for one party to obtain the evidence without the other party *also* receiving the evidence.

Motivation

Actions of participants

Motivation

Actions of participants

$$Act_A = \{A \rightarrow B{:}m, A \rightarrow T{:}m, A \rightarrow B{:}NRO, A \rightarrow T{:}NRO, A{:}skip\}$$

$$Act_B = \{B \rightarrow A{:}NRR, B \rightarrow T{:}NRR, B{:}skip\}$$

$$Act_T = \{T \rightarrow A{:}NRR, T \rightarrow B{:}NRO, T \rightarrow B{:}m, T{:}skip\}$$

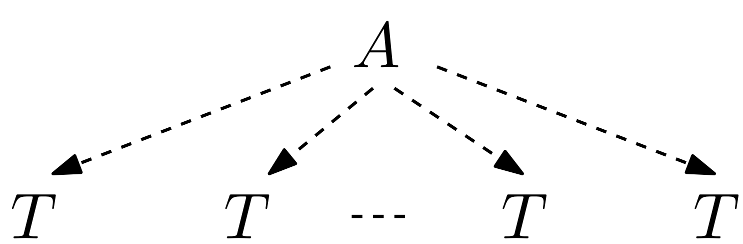
Motivation

Actions of participants

$$Act_A = \{A{\rightarrow}B{:}m, A{\rightarrow}T{:}m, A{\rightarrow}B{:}NRO, A{\rightarrow}T{:}NRO, A{:}skip\}$$

$$Act_B = \{B{\rightarrow}A{:}NRR, B{\rightarrow}T{:}NRR, B{:}skip\}$$

$$Act_T = \{T{\rightarrow}A{:}NRR, T{\rightarrow}B{:}NRO, T{\rightarrow}B{:}m, T{:}skip\}$$



Controllable transition

Uncontrollable transition

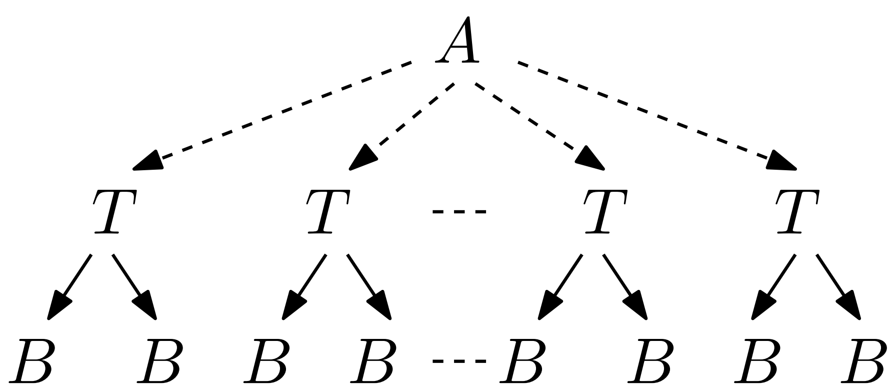
Motivation

Actions of participants

$$Act_A = \{A \to B{:}m, A \to T{:}m, A \to B{:}NRO, A \to T{:}NRO, A{:}skip\}$$

$$Act_B = \{B \to A{:}NRR, B \to T{:}NRR, B{:}skip\}$$

$$Act_T = \{T \to A{:}NRR, T \to B{:}NRO, T \to B{:}m, T{:}skip\}$$



Controllable transition

Uncontrollable transition

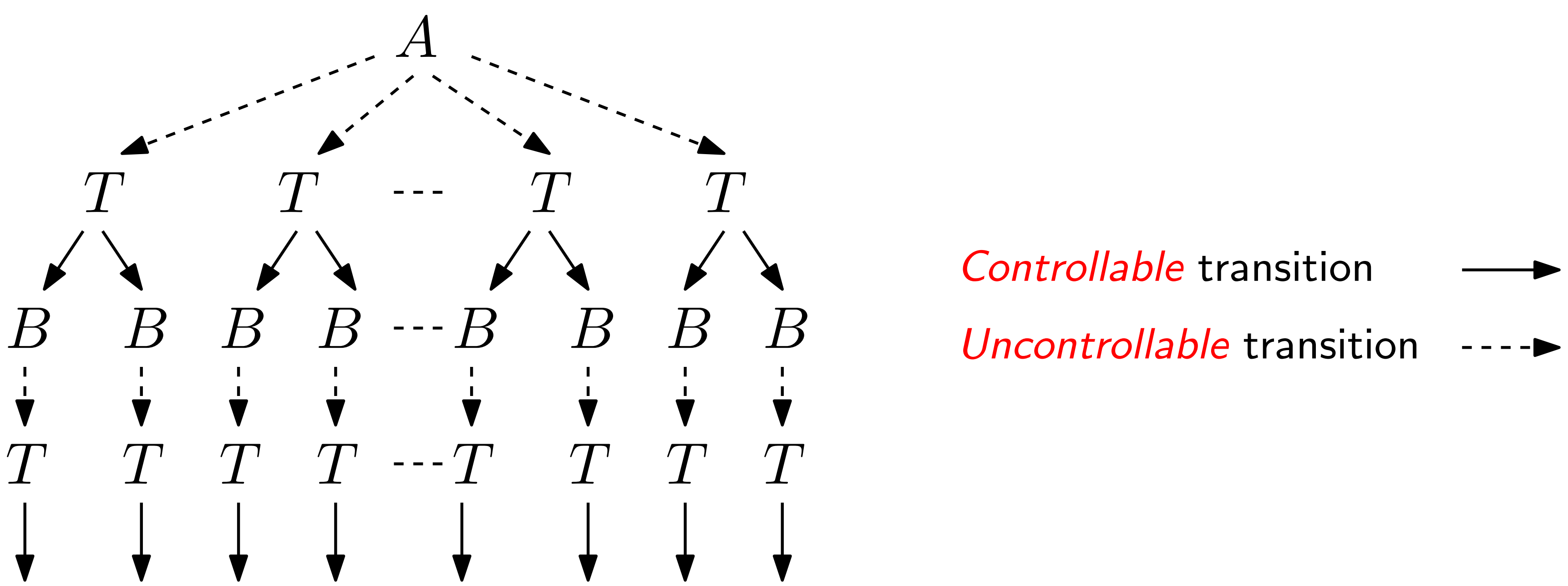
Motivation

Actions of participants

$$Act_A = \{A{\rightarrow}B{:}m, A{\rightarrow}T{:}m, A{\rightarrow}B{:}NRO, A{\rightarrow}T{:}NRO, A{:}skip\}$$

$$Act_B = \{B{\rightarrow}A{:}NRR, B{\rightarrow}T{:}NRR, B{:}skip\}$$

$$Act_T = \{T{\rightarrow}A{:}NRR, T{\rightarrow}B{:}NRO, T{\rightarrow}B{:}m, T{:}skip\}$$



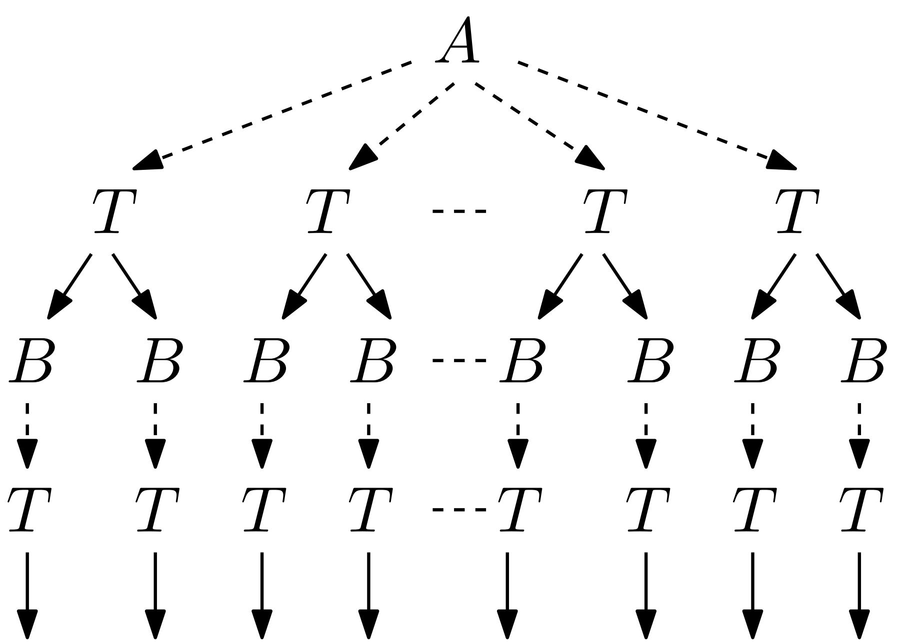
Motivation

Actions of participants

$$Act_A = \{A{\rightarrow}B{:}m, A{\rightarrow}T{:}m, A{\rightarrow}B{:}NRO, A{\rightarrow}T{:}NRO, A{:}skip\}$$

$$Act_B = \{B{\rightarrow}A{:}NRR, B{\rightarrow}T{:}NRR, B{:}skip\}$$

$$Act_T = \{T{\rightarrow}A{:}NRR, T{\rightarrow}B{:}NRO, T{\rightarrow}B{:}m, T{:}skip\}$$



Controllable transition $\longrightarrow$

Uncontrollable transition $\dashrightarrow$

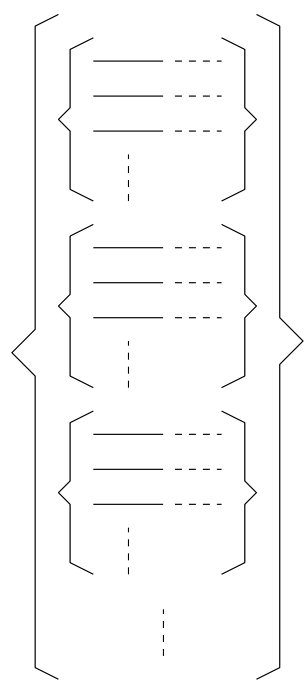
How can we *synthesize* the behavior of T, ensuring arbitrary behavior for A and B?

Motivation

- *Specification* of the protocol:
  - there should *exist* a sequence of actions, such that the message m, the NRR, and the NRO get received, such that
  - *for all* similar executions of A and B, it must still hold that the NRR gets received if and only if the NRO gets received.

Motivation

- *Specification* of the protocol:
  - there should *exist* a sequence of actions, such that the message m, the NRR, and the NRO get received, such that
  - *for all* similar executions of A and B, it must still hold that the NRR gets received if and only if the NRO gets received.
- This is a *hyperproperty*, i.e., a set of sets of traces.



Motivation

Prominent security breaches related to *information flow security* (hyperproperties)

Motivation

Prominent security breaches related to *information flow security* (hyperproperties)

MELTDOWN

Motivation

Prominent security breaches related to *information flow security* (hyperproperties)

MELTDOWN

SPECTRE

Motivation

Prominent security breaches related to *information flow security* (hyperproperties)

MELTDOWN

SPECTRE

Informal Problem Statement

Informal Problem Statement

(Controllable)
Plant $\mathcal{P}$

Informal Problem Statement

(Controllable)
Plant $\mathcal{P}$

Uncontrollable
transitions $\mathfrak{u}$

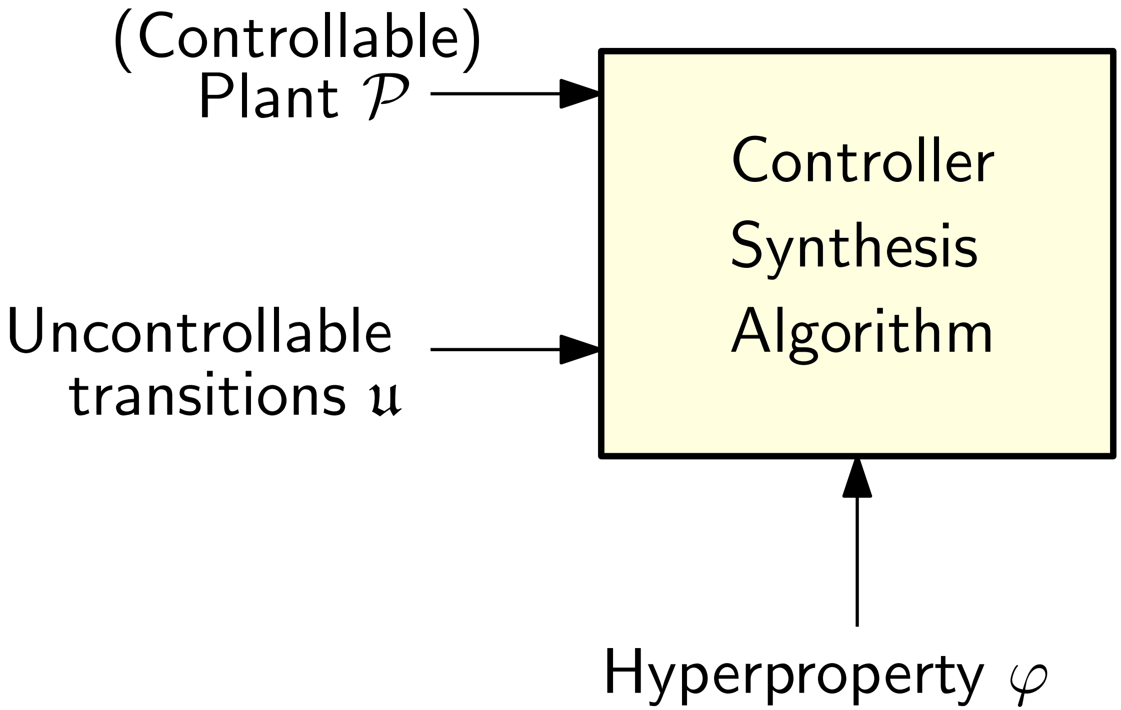
Informal Problem Statement

(Controllable)
Plant $\mathcal{P}$

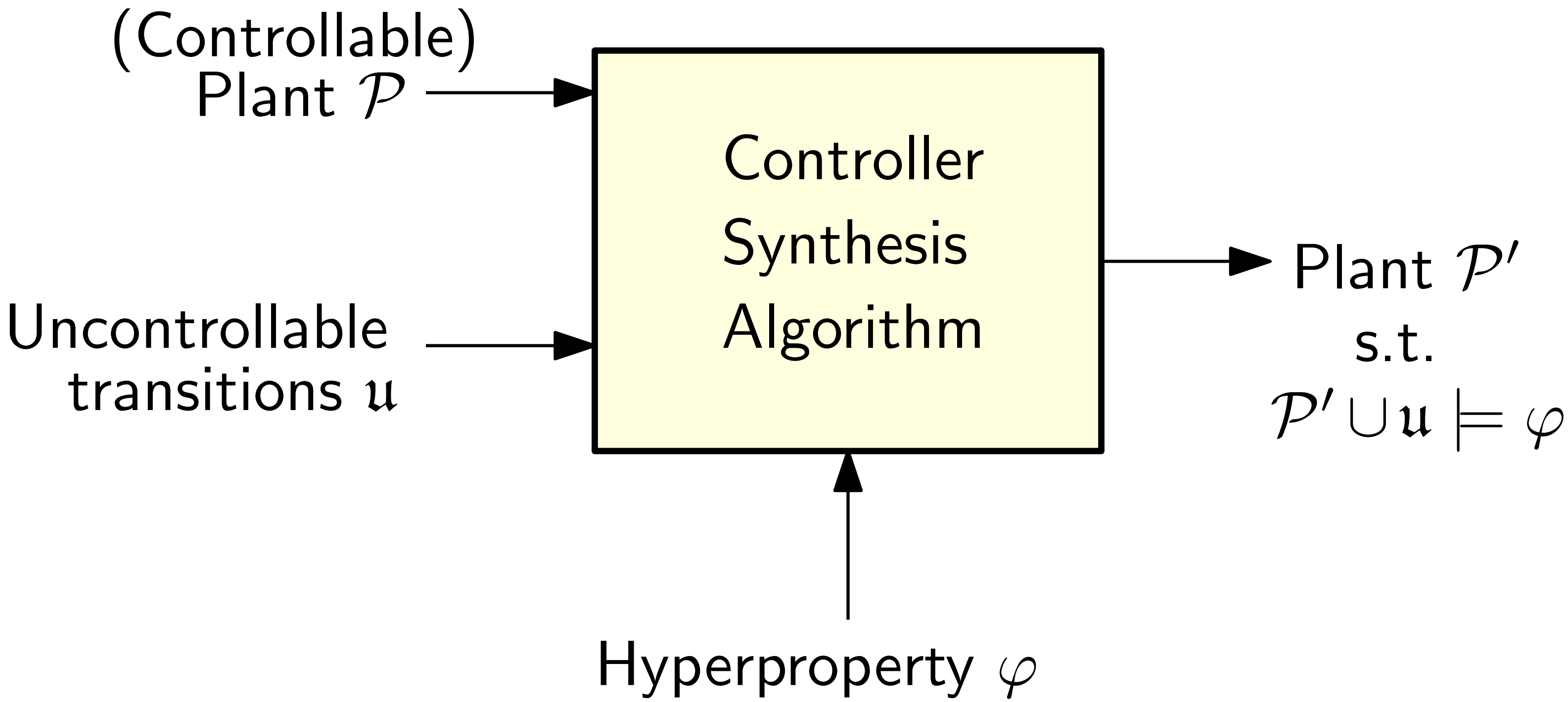
Uncontrollable
transitions $\mathfrak{u}$

Hyperproperty φ

Informal Problem Statement



Informal Problem Statement



2. HyperLTL

M. R. Clarkson, B. Finkbeiner, M. Koleini, K. K. Micinski, M. N. Rabe, C. Sánchez:
Temporal Logics for Hyperproperties. POST 2014: 265-284

Preliminaries – HyperLTL

Syntax

Semantics

Preliminaries – HyperLTL

Syntax

$$\alpha ::= \exists \pi.\alpha \;\;\Big|\;\; \forall \pi.\alpha \;\;\Big|\;\; \varphi$$

Semantics

Preliminaries – HyperLTL

Syntax

$$\alpha ::= \exists \pi . \alpha \quad \Big| \quad \forall \pi . \alpha \quad \Big| \quad \varphi$$

$$\varphi ::= a_\pi \quad \Big| \quad \varphi \vee \varphi \quad \Big| \quad \neg \varphi \quad \Big| \quad \bigcirc \varphi \quad \Big| \quad \varphi \, \mathcal{U} \, \varphi$$

Semantics

Preliminaries – HyperLTL

Syntax

$$\alpha ::= \exists \pi . \alpha \quad \big| \quad \forall \pi . \alpha \quad \big| \quad \varphi$$

$$\varphi ::= a_\pi \quad \big| \quad \varphi \vee \varphi \quad \big| \quad \neg \varphi \quad \big| \quad \bigcirc \varphi \quad \big| \quad \varphi \, \mathcal{U} \, \varphi$$

Semantics

(W, Π)	$\models \exists \pi . \alpha$	iff	for some $\sigma \in W$, $(W, \Pi[\pi \mapsto (\sigma, 0)]) \models \alpha$
(W, Π)	$\models \forall \pi . \alpha$	iff	for all $\sigma \in W$, $(W, \Pi[\pi \mapsto (\sigma, 0)]) \models \alpha$

Preliminaries – HyperLTL

Syntax

$$\alpha ::= \exists \pi.\alpha \quad \Big| \quad \forall \pi.\alpha \quad \Big| \quad \varphi$$

$$\varphi ::= a_\pi \quad \Big| \quad \varphi \vee \varphi \quad \Big| \quad \neg\varphi \quad \Big| \quad \bigcirc\varphi \quad \Big| \quad \varphi\,\mathcal{U}\,\varphi$$

Semantics

$$\begin{array}{lll} (W,\Pi) \models \exists \pi.\alpha & \text{iff} & \text{for some } \sigma \in W,\ (W,\Pi[\pi \mapsto (\sigma,0)]) \models \alpha \\ (W,\Pi) \models \forall \pi.\alpha & \text{iff} & \text{for all } \sigma \in W,\ (W,\Pi[\pi \mapsto (\sigma,0)]) \models \alpha \end{array}$$

$$\begin{array}{rlll} (W,\Pi) & \models \varphi & \text{iff} & \Pi \models \varphi \\ \Pi & \models a_\pi & \text{iff} & a \in \sigma(p),\ \text{where } (\sigma,p) = \Pi(\pi) \\ \Pi & \models \varphi_1 \vee \varphi_2 & \text{iff} & \Pi \models \varphi_1 \text{ or } \Pi \models \varphi_2 \\ \Pi & \models \neg\varphi & \text{iff} & \Pi \not\models \varphi \end{array}$$

Preliminaries – HyperLTL

Syntax

$$\alpha ::= \exists\pi.\alpha \quad \big| \quad \forall\pi.\alpha \quad \big| \quad \varphi$$

$$\varphi ::= a_\pi \quad \big| \quad \varphi \vee \varphi \quad \big| \quad \neg\varphi \quad \big| \quad \bigcirc\varphi \quad \big| \quad \varphi\,\mathcal{U}\,\varphi$$

Semantics

(W,Π)	$\models \exists\pi.\alpha$	iff	for some $\sigma \in W$, $(W,\Pi[\pi \mapsto (\sigma,0)]) \models \alpha$
(W,Π)	$\models \forall\pi.\alpha$	iff	for all $\sigma \in W$, $(W,\Pi[\pi \mapsto (\sigma,0)]) \models \alpha$

(W,Π)	$\models \varphi$	iff	$\Pi \models \varphi$
Π	$\models a_\pi$	iff	$a \in \sigma(p)$, where $(\sigma,p) = \Pi(\pi)$
Π	$\models \varphi_1 \vee \varphi_2$	iff	$\Pi \models \varphi_1$ or $\Pi \models \varphi_2$
Π	$\models \neg\varphi$	iff	$\Pi \not\models \varphi$

Π	$\models \bigcirc\varphi$	iff	$(\Pi+1) \models \varphi$
Π	$\models \varphi_1\,\mathcal{U}\,\varphi_2$	iff	for some $j \geq 0$ $(\Pi+j) \models \varphi_2$ and for all $0 \leq i < j, (\Pi+i) \models \varphi_1$

Preliminaries – HyperLTL

Preliminaries – HyperLTL

- The meaning of *HyperLTL* formula

$$\varphi = \forall\pi.\forall\pi'.\Box(a_\pi \leftrightarrow a_{\pi'})$$

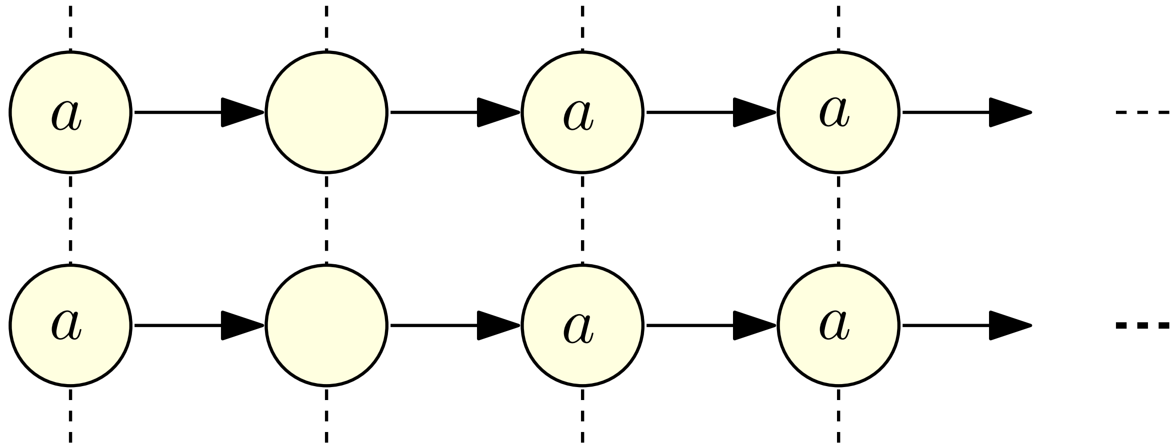
is that any pair of traces should agree on the value of a at every position.

Preliminaries – HyperLTL

- The meaning of *HyperLTL* formula

$$\varphi = \forall \pi . \forall \pi' . \Box (a_\pi \leftrightarrow a_{\pi'})$$

is that any pair of traces should agree on the value of a at every position.

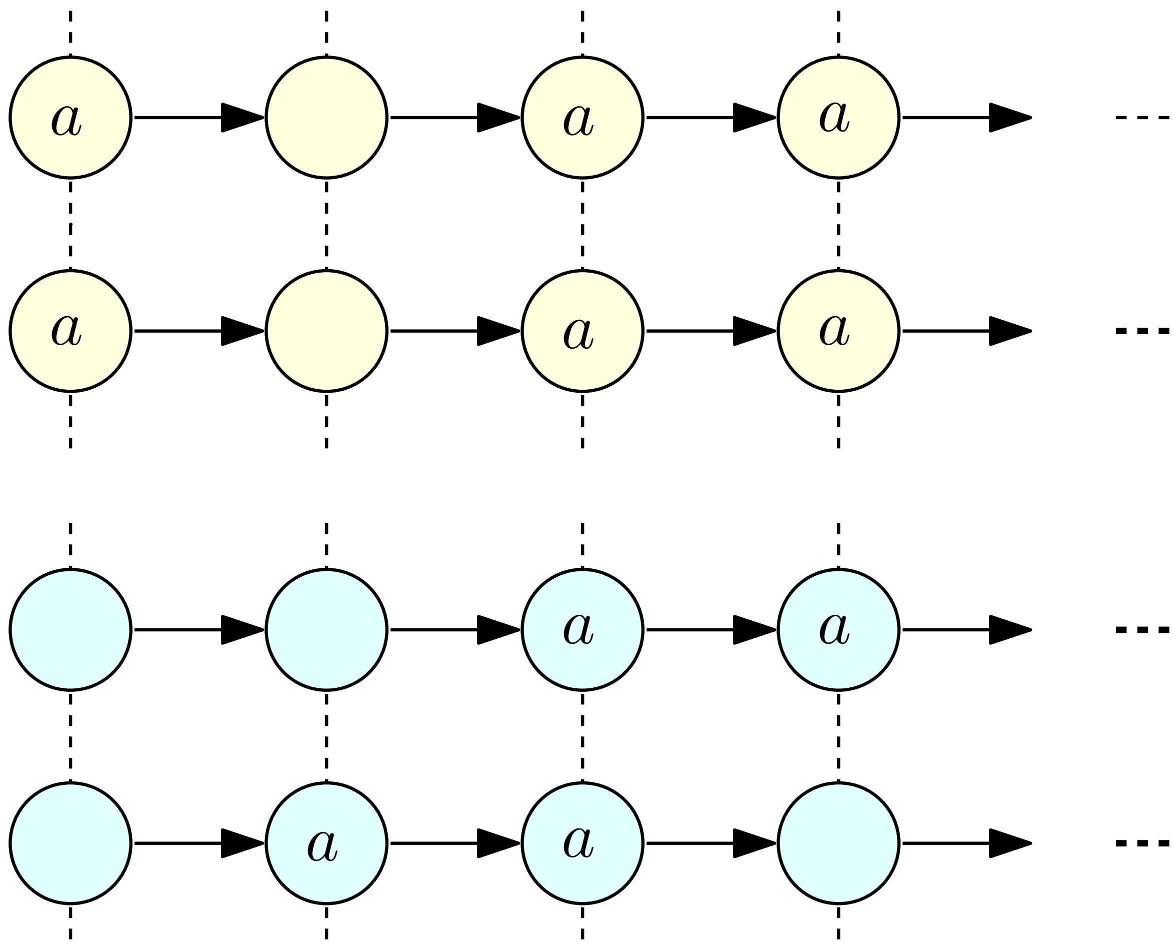


Preliminaries – HyperLTL

- The meaning of *HyperLTL* formula

$$\varphi = \forall \pi . \forall \pi' . \Box (a_\pi \leftrightarrow a_{\pi'})$$

is that any pair of traces should agree on the value of a at every position.

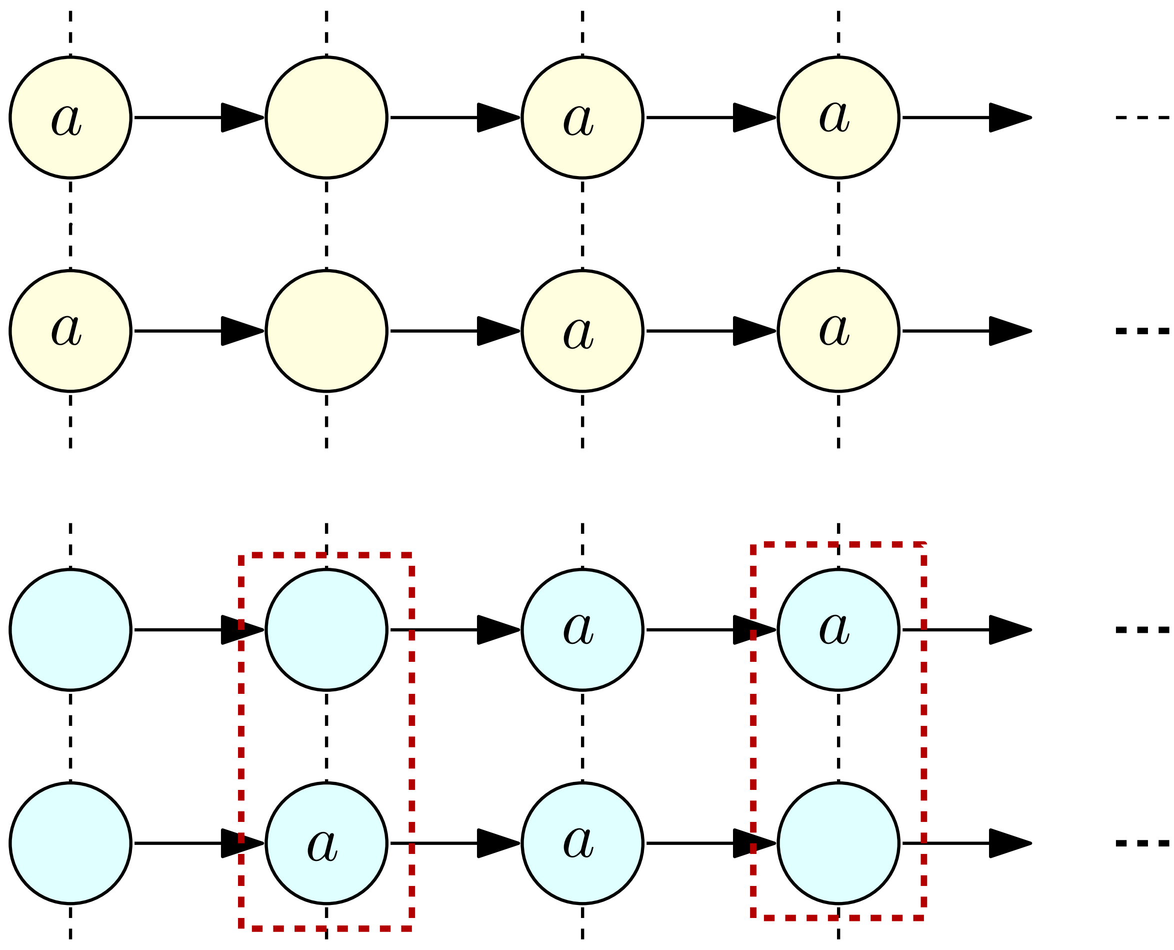


Preliminaries – HyperLTL

- The meaning of *HyperLTL* formula

$$\varphi = \forall \pi . \forall \pi' . \square (a_\pi \leftrightarrow a_{\pi'})$$

is that any pair of traces should agree on the value of a at every position.



Preliminaries – HyperLTL

Preliminaries – HyperLTL

- *Observational determinism* [Zdancewich, Meyers 2003]:

$$\forall \pi. \forall \pi'. (i_\pi \leftrightarrow i_{\pi'}) \;\rightarrow\; \Box(o_\pi \leftrightarrow o_{\pi'})$$

Preliminaries – HyperLTL

- *Observational determinism* [Zdancewich, Meyers 2003]:

$$\forall \pi . \forall \pi' . (i_\pi \leftrightarrow i_{\pi'}) \;\rightarrow\; \Box (o_\pi \leftrightarrow o_{\pi'})$$

- *Non-inference* [McLean 1994]

$$\forall \pi . \exists \pi' . \Box (hi_\pi) \wedge \Box (li_\pi \leftrightarrow li_{\pi'} \;\wedge\; lo_\pi \leftrightarrow lo_{\pi'})$$

Preliminaries – HyperLTL

Preliminaries – HyperLTL

- *Non-repudiation*:

$$\varphi = \exists\pi.\forall\pi'.\ (\Diamond m_\pi) \wedge (\Diamond NRR_\pi) \wedge (\Diamond NRO_\pi)$$

(effectiveness)

$$\wedge \Big((\Box \textstyle\bigwedge_{a \in Act_A} a_\pi \Leftrightarrow a_{\pi'}) \Rightarrow \big((\Diamond NRR_{\pi'}) \Leftrightarrow (\Diamond NRO_{\pi'}) \big) \Big)$$

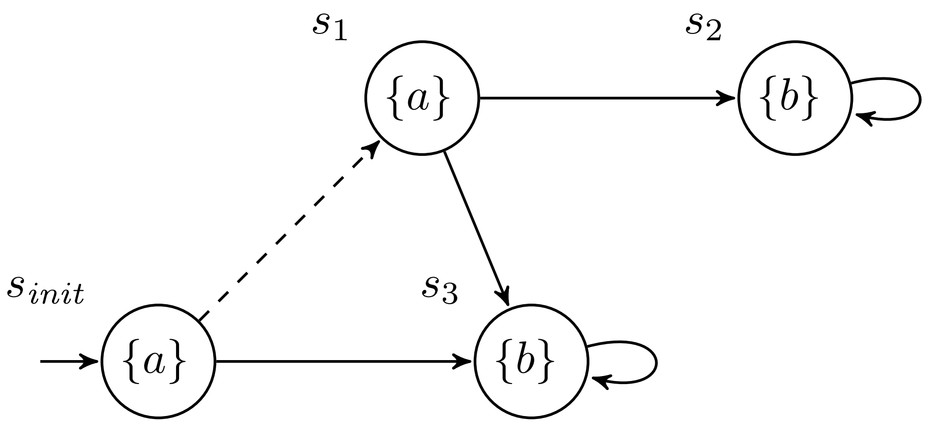
(fairness for A)

$$\wedge \Big((\Box \textstyle\bigwedge_{a \in Act_B} a_\pi \Leftrightarrow a_{\pi'}) \Rightarrow \big((\Diamond NRR_{\pi'}) \Leftrightarrow (\Diamond NRO_{\pi'}) \big) \Big)$$

(fairness for B)

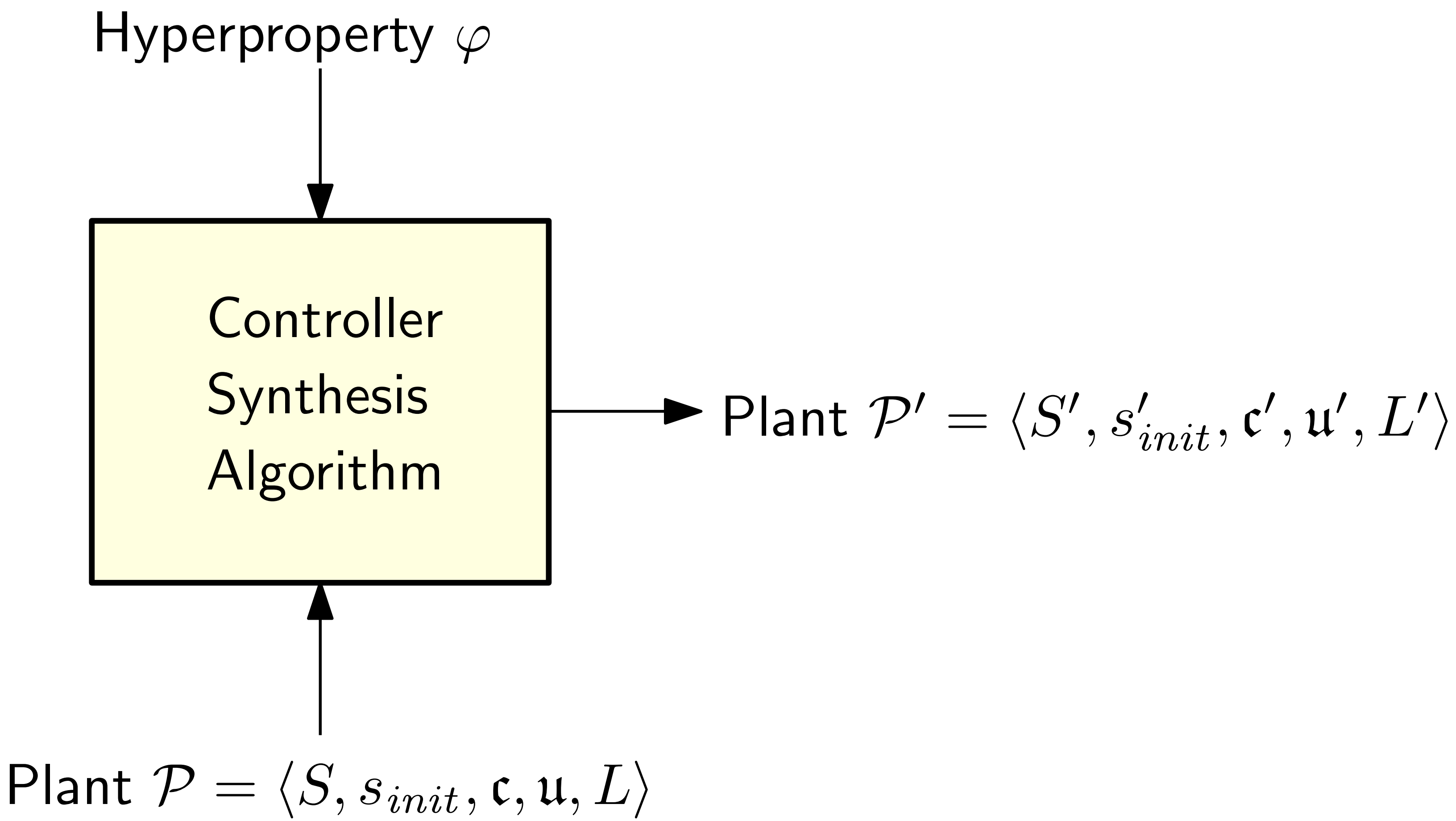
Preliminaries – Plants

- A *plant* is a tuple $\mathcal{P} = \langle S, s_{init}, \mathfrak{c}, \mathfrak{u}, L \rangle$, where
  - S is a finite set of *states*;
  - $s_{init} \in S$ is the *initial state*;
  - $\mathfrak{c}, \mathfrak{u} \subseteq S \times S$ are respectively sets of of *controllable* and *uncontrollable* transitions, where $\mathfrak{c} \cap \mathfrak{u} = \{\}$, and
  - $L : S \to \Sigma$ is a *labeling function* on the states of $\mathcal{P}$.

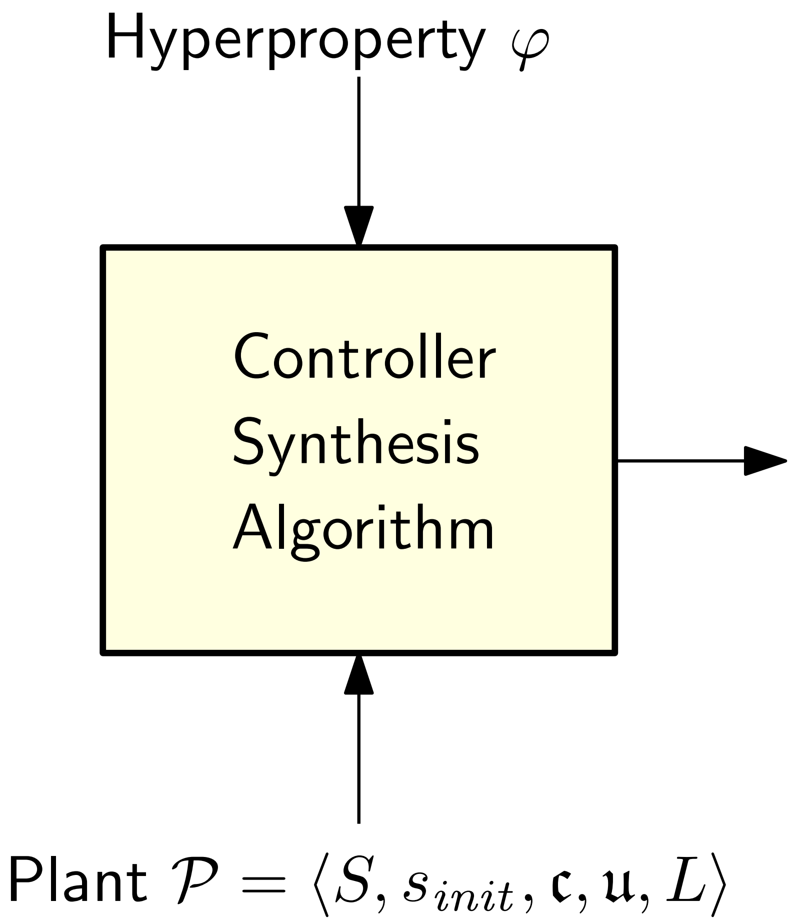


3. Problem Statement

Formal Problem Statement



Formal Problem Statement

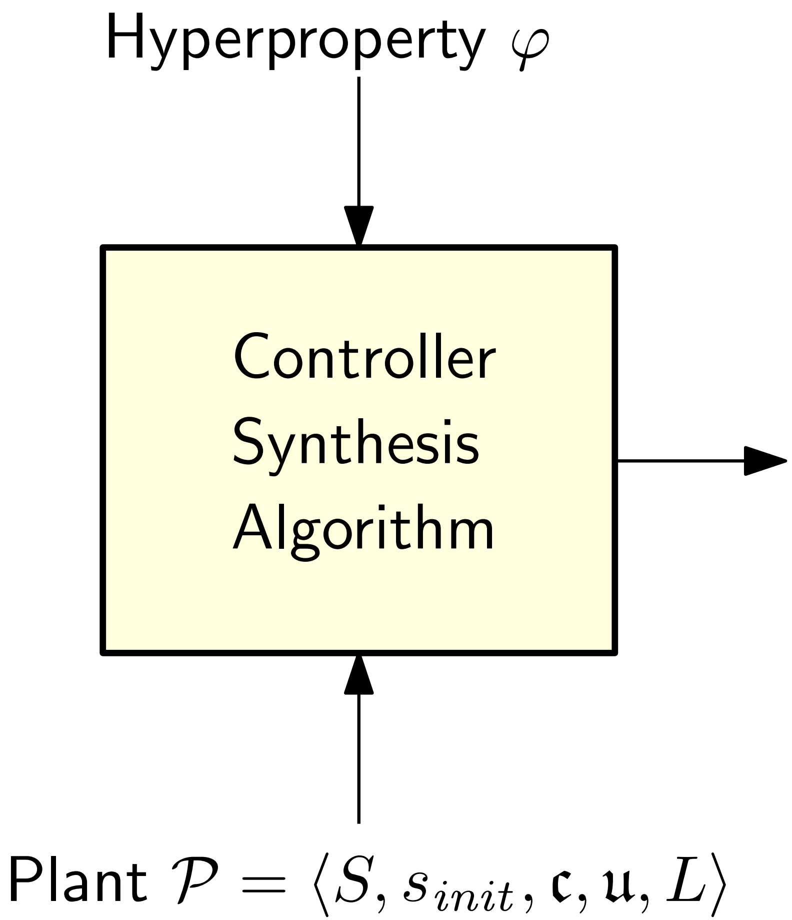


Plant $\mathcal{P}' = \langle S', s'_{init}, \mathfrak{c}', \mathfrak{u}', L' \rangle$

s.t.

- $S' = S$
- $s'_{init} = s_{init}$
- $\mathfrak{c}' \subseteq \mathfrak{c}$
- $\mathfrak{u}' = \mathfrak{u}$
- $L' = L$
- $\mathcal{P}' \models \varphi$

Formal Problem Statement



Plant $\mathcal{P}' = \langle S', s'_{init}, \mathfrak{c}', \mathfrak{u}', L' \rangle$

s.t.

- $S' = S$
- $s'_{init} = s_{init}$
- $\mathfrak{c}' \subseteq \mathfrak{c}$
- $\mathfrak{u}' = \mathfrak{u}$
- $L' = L$
- $\mathcal{P}' \models \varphi$

Formal Problem Statement

Hyperproperty φ $\begin{cases} \forall\forall \text{ (observational determinism)} \\ \forall\exists \text{ (non-inference)} \\ \exists\forall \text{ (non-repudiation)} \\ \forall\forall\exists \text{ (generalized non-interference)} \end{cases}$

Controller Synthesis Algorithm

Plant $\mathcal{P} = \langle S, s_{init}, \mathfrak{c}, \mathfrak{u}, L \rangle$

Plant $\mathcal{P}' = \langle S', s'_{init}, \mathfrak{c}', \mathfrak{u}', L' \rangle$

s.t.

- $S' = S$
- $s'_{init} = s_{init}$
- $\mathfrak{c}' \subseteq \mathfrak{c}$
- $\mathfrak{u}' = \mathfrak{u}$
- $L' = L$
- $\mathcal{P}' \models \varphi$

Formal Problem Statement

Hyperproperty φ

- $\forall\forall$ (observational determinism)
- $\forall\exists$ (non-inference)
- $\exists\forall$ (non-repudiation)
- $\forall\forall\exists$ (generalized non-interference)

Controller Synthesis Algorithm

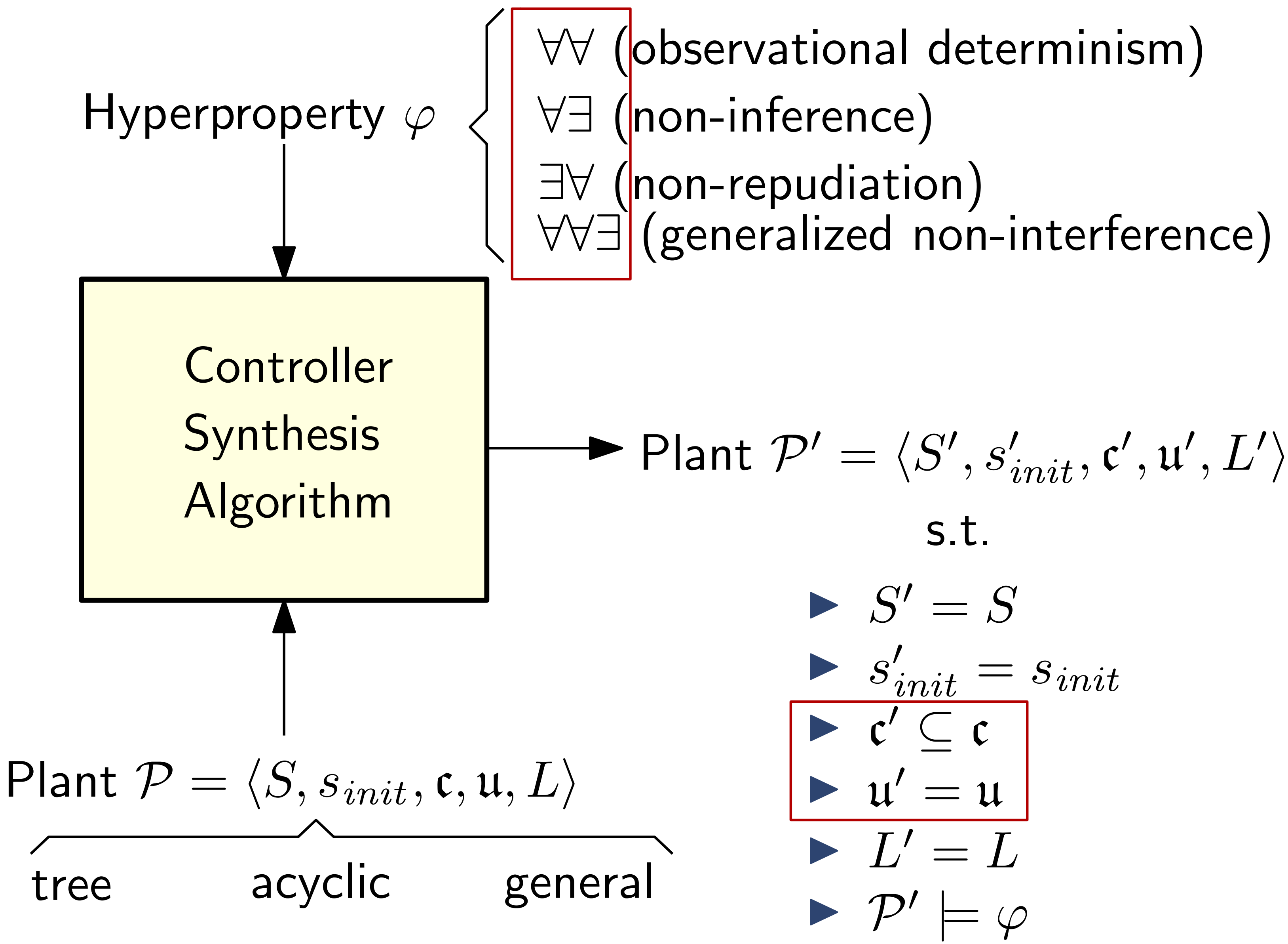
Plant $\mathcal{P} = \langle S, s_{init}, \mathfrak{c}, \mathfrak{u}, L \rangle$

Plant $\mathcal{P}' = \langle S', s'_{init}, \mathfrak{c}', \mathfrak{u}', L' \rangle$

s.t.

- $S' = S$
- $s'_{init} = s_{init}$
- $\mathfrak{c}' \subseteq \mathfrak{c}$
- $\mathfrak{u}' = \mathfrak{u}$
- $L' = L$
- $\mathcal{P}' \models \varphi$

Formal Problem Statement



Formal Problem Statement

Hyperproperty φ

- $\forall\forall$ (observational determinism)
- $\forall\exists$ (non-inference)
- $\exists\forall$ (non-repudiation)
- $\forall\forall\exists$ (generalized non-interference)

Controller Synthesis Algorithm

Plant $\mathcal{P}' = \langle S', s'_{init}, \mathfrak{c}', \mathfrak{u}', L' \rangle$

s.t.

- $S' = S$
- $s'_{init} = s_{init}$
- $\mathfrak{c}' \subseteq \mathfrak{c}$
- $\mathfrak{u}' = \mathfrak{u}$
- $L' = L$
- $\mathcal{P}' \models \varphi$

Plant $\mathcal{P} = \langle S, s_{init}, \mathfrak{c}, \mathfrak{u}, L \rangle$

tree acyclic general

Session-based and terminating protocols

Summary of Results

<table>
<tr><th>HyperLTL fragment</th><th>Tree</th><th>Acyclic</th><th>General</th></tr>
<tr><td>E^*</td><td rowspan="2">L-complete
(Theorem 1)</td><td>NL-complete
(Theorem 5)</td><td>NL-complete
(Theorem 9)</td></tr>
<tr><td>$\mathsf{E}^*\mathsf{A}$</td><td>Σ_2^p</td><td rowspan="2">PSPACE-complete
(Theorem 11)</td></tr>
<tr><td>AE^*</td><td>P-complete
(Theorem 2)</td><td>Σ_2^p-complete
(Theorem 8)</td></tr>
<tr><td>AA^+</td><td rowspan="4">NP-complete
(Corollary 1)</td><td>NP-complete
(Theorem 6)</td><td>NP-complete
(Theorem 10)</td></tr>
<tr><td>$(\mathsf{E}^*\mathsf{A}^*)^k$,
$k \geq 2$</td><td>Σ_k^p-complete
(Theorem 8)</td><td rowspan="2">$(k-1)$-EXPSPACE-complete
(Theorem 11)</td></tr>
<tr><td>$(\mathsf{A}^*\mathsf{E}^*)^k$,
$k \geq 1$</td><td>Σ_{k+1}^p-complete
(Theorem 8)</td></tr>
<tr><td>$(\mathsf{A}^*\mathsf{E}^*)^*$</td><td>PSPACE
(Corollary 3)</td><td>NONELEMENTARY
(Corollary 4)</td></tr>
</table>

Summary of Results

<table>
<tr><th>HyperLTL fragment</th><th>Tree</th><th>Acyclic</th><th>General</th></tr>
<tr><td>E^*</td><td rowspan="2">L-complete
(Theorem 1)</td><td>NL-complete
(Theorem 5)</td><td>NL-complete
(Theorem 9)</td></tr>
<tr><td>$\mathsf{E}^*\mathsf{A}$</td><td>Σ_2^p</td><td rowspan="2">PSPACE-complete
(Theorem 11)</td></tr>
<tr><td>$\mathsf{A}\mathsf{E}^*$</td><td>P-complete
(Theorem 2)</td><td>Σ_2^p-complete
(Theorem 8)</td></tr>
<tr><td>$\mathsf{A}\mathsf{A}^+$</td><td rowspan="4">NP-complete
(Corollary 1)</td><td>NP-complete
(Theorem 6)</td><td>NP-complete
(Theorem 10)</td></tr>
<tr><td>$(\mathsf{E}^*\mathsf{A}^*)^k$,
$k \geq 2$</td><td>Σ_k^p-complete
(Theorem 8)</td><td rowspan="2">$(k-1)$-EXPSPACE-complete
(Theorem 11)</td></tr>
<tr><td>$(\mathsf{A}^*\mathsf{E}^*)^k$,
$k \geq 1$</td><td>Σ_{k+1}^p-complete
(Theorem 8)</td></tr>
<tr><td>$(\mathsf{A}^*\mathsf{E}^*)^*$</td><td>PSPACE
(Corollary 3)</td><td>NONELEMENTARY
(Corollary 4)</td></tr>
</table>

Summary of Results

HyperLTL fragment	**Tree**	**Acyclic**	**General**
E*	L-complete *(Theorem 1)*	NL-complete *(Theorem 5)*	NL-complete *(Theorem 9)*
E*A		Σ_2^p	PSPACE-complete *(Theorem 11)*
AE*	P-complete *(Theorem 2)*	Σ_2^p-complete *(Theorem 8)*	
AA$^+$	NP-complete *(Corollary 1)*	NP-complete *(Theorem 6)*	NP-complete *(Theorem 10)*
(E*A*)^k, $k \geq 2$		Σ_k^p-complete *(Theorem 8)*	$(k-1)$-EXPSPACE-complete *(Theorem 11)*
(A*E*)^k, $k \geq 1$		Σ_{k+1}^p-complete *(Theorem 8)*	
(A*E*)*		PSPACE *(Corollary 3)*	NONELEMENTARY *(Corollary 4)*

Summary of Results

HyperLTL fragment	Tree	Acyclic	General
E^*	L-complete *(Theorem 1)*	NL-complete *(Theorem 5)*	NL-complete *(Theorem 9)*
$\mathsf{E}^*\mathsf{A}$		Σ_2^p	PSPACE-complete *(Theorem 11)*
$\mathsf{A}\mathsf{E}^*$	P-complete *(Theorem 2)*	Σ_2^p-complete *(Theorem 8)*	
$\mathsf{A}\mathsf{A}^+$	NP-complete *(Corollary 1)*	NP-complete *(Theorem 6)*	NP-complete *(Theorem 10)*
$(\mathsf{E}^*\mathsf{A}^*)^k$, $k \geq 2$		Σ_k^p-complete *(Theorem 8)*	$(k-1)$-EXPSPACE-complete *(Theorem 11)*
$(\mathsf{A}^*\mathsf{E}^*)^k$, $k \geq 1$		Σ_{k+1}^p-complete *(Theorem 8)*	
$(\mathsf{A}^*\mathsf{E}^*)^*$		PSPACE *(Corollary 3)*	NONELEMENTARY *(Corollary 4)*

4. Controller Synthesis for Trees

Controller Synthesis for Trees

HyperLTL fragment	**Tree**
E*	L-complete *(Theorem 1)*
E*A	
AE*	P-complete *(Theorem 2)*
AA$^+$	NP-complete *(Corollary 1)*
(E*A*)^k, $k \geq 2$	
(A*E*)^k, $k \geq 1$	
(A*E*)*	

Controller Synthesis for Trees

HyperLTL fragment	Tree
E^*	L-complete *(Theorem 1)*
$\mathsf{E}^*\mathsf{A}$	
$\mathsf{A}\mathsf{E}^*$	P-complete *(Theorem 2)*
$\mathsf{A}\mathsf{A}^+$	NP-complete *(Corollary 1)*
$(\mathsf{E}^*\mathsf{A}^*)^k$, $k \geq 2$	
$(\mathsf{A}^*\mathsf{E}^*)^k$, $k \geq 1$	
$(\mathsf{A}^*\mathsf{E}^*)^*$	

Controller Synthesis for Trees

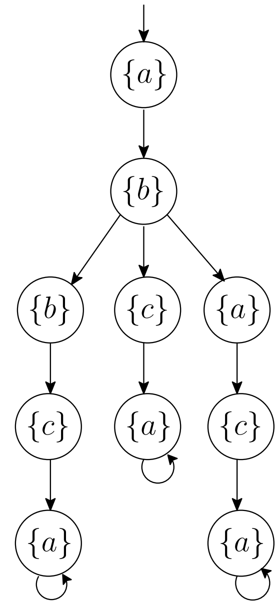
HyperLTL fragment	Tree
E^*	L-complete *(Theorem 1)*
$\mathsf{E}^*\mathsf{A}$	
$\mathsf{A}\mathsf{E}^*$	P-complete *(Theorem 2)*
$\mathsf{A}\mathsf{A}^+$	NP-complete *(Corollary 1)*
$(\mathsf{E}^*\mathsf{A}^*)^k$, $k \geq 2$	
$(\mathsf{A}^*\mathsf{E}^*)^k$, $k \geq 1$	
$(\mathsf{A}^*\mathsf{E}^*)^*$	

Upper bound

Controller Synthesis for Trees

HyperLTL fragment	Tree
E*	L-complete *(Theorem 1)*
E*A	
AE*	P-complete *(Theorem 2)*
AA$^+$	NP-complete *(Corollary 1)*
(E*A*)^k, $k \geq 2$	
(A*E*)^k, $k \geq 1$	
(A*E*)*	

Upper bound



|paths| $\leq$ |states|
trace length $\leq$ |states|

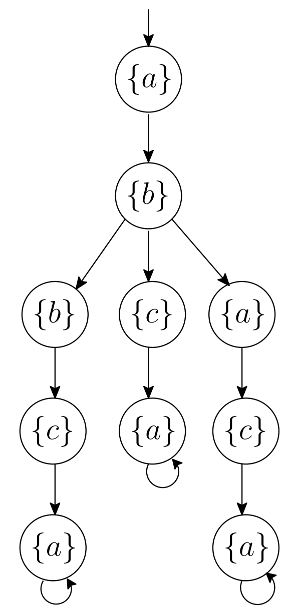
Controller Synthesis for Trees

HyperLTL fragment	Tree
E*	L-complete *(Theorem 1)*
E*A	
AE*	P-complete *(Theorem 2)*
AA$^+$	NP-complete *(Corollary 1)*
(E*A*)^k, $k \geq 2$	
(A*E*)^k, $k \geq 1$	
(A*E*)*	

Upper bound

We only need one path per $\exists$

1. *Find path assignment:*
   - go through all path assignments for $\exists^*$ using logarithmic counters
   - Go through all path assignments for $\forall^*$ to one of the $\exists$-paths



|paths| $\leq$ |states|
trace length $\leq$ |states|

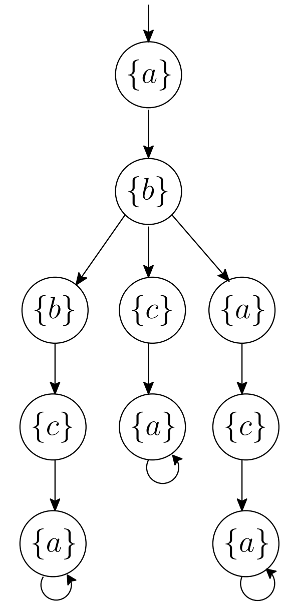
Controller Synthesis for Trees

HyperLTL fragment	**Tree**
E*	L-complete *(Theorem 1)*
E*A	
AE*	P-complete *(Theorem 2)*
AA$^+$	NP-complete *(Corollary 1)*
(E*A*)^k, $k \geq 2$	
(A*E*)^k, $k \geq 1$	
(A*E*)*	

Upper bound

We only need one path per $\exists$

1. *Find path assignment:*
   - go through all path assignments for $\exists^*$ using logarithmic counters
   - Go through all path assignments for $\forall^*$ to one of the $\exists$-paths



|paths| $\leq$ |states|
trace length $\leq$ |states|

2. *Verify correctness:*
   - check each temporal operator with a logarithmic counter

Controller Synthesis for Trees

HyperLTL fragment	**Tree**
E^*	L-complete *(Theorem 1)*
$\mathsf{E}^*\mathsf{A}$	
AE^*	P-complete *(Theorem 2)*
AA^+	NP-complete *(Corollary 1)*
$(\mathsf{E}^*\mathsf{A}^*)^k$, $k \geq 2$	
$(\mathsf{A}^*\mathsf{E}^*)^k$, $k \geq 1$	
$(\mathsf{A}^*\mathsf{E}^*)^*$	

Controller Synthesis for Trees

HyperLTL fragment	**Tree**
E^*	L-complete *(Theorem 1)*
$\mathsf{E}^*\mathsf{A}$	
$\mathsf{A}\mathsf{E}^*$	P-complete *(Theorem 2)*
$\mathsf{A}\mathsf{A}^+$	NP-complete *(Corollary 1)*
$(\mathsf{E}^*\mathsf{A}^*)^k$, $k \geq 2$	
$(\mathsf{A}^*\mathsf{E}^*)^k$, $k \geq 1$	
$(\mathsf{A}^*\mathsf{E}^*)^*$	

Controller Synthesis for Trees

HyperLTL fragment	Tree
E^*	L-complete *(Theorem 1)*
$\mathsf{E}^*\mathsf{A}$	
AE^*	P-complete *(Theorem 2)*
AA^+	NP-complete *(Corollary 1)*
$(\mathsf{E}^*\mathsf{A}^*)^k$, $k \geq 2$	
$(\mathsf{A}^*\mathsf{E}^*)^k$, $k \geq 1$	
$(\mathsf{A}^*\mathsf{E}^*)^*$	

Upper bound $(\forall \pi_1 . \exists \pi_2 . \psi)$

Controller Synthesis for Trees

HyperLTL fragment	Tree
E*	L-complete *(Theorem 1)*
E*A	
AE*	P-complete *(Theorem 2)*
AA$^+$	NP-complete *(Corollary 1)*
(E*A*)^k, $k \geq 2$	
(A*E*)^k, $k \geq 1$	
(A*E*)*	

Upper bound $(\forall \pi_1 . \exists \pi_2 . \psi)$

1. We begin by *marking* all leaves and proceed in several rounds, in which at least one mark is removed (*linear* rounds).

Controller Synthesis for Trees

HyperLTL fragment	**Tree**
E*	L-complete *(Theorem 1)*
E*A	
AE*	P-complete *(Theorem 2)*
AA$^+$	NP-complete *(Corollary 1)*
(E*A*)^k, $k \geq 2$	
(A*E*)^k, $k \geq 1$	
(A*E*)*	

Upper bound $(\forall \pi_1 . \exists \pi_2 . \psi)$

1. We begin by *marking* all leaves and proceed in several rounds, in which at least one mark is removed (*linear* rounds).

2. In each *round*, we go through all marked leaves v_1 and instantiate π_1 with the trace leading to v_1. We then again go through all marked leaves v_2 and instantiate π_2 with the trace leading to v_2, and check ψ on the pair of traces (*linear* time).

Controller Synthesis for Trees

HyperLTL fragment	Tree
E^*	L-complete (*Theorem 1*)
$\mathsf{E}^*\mathsf{A}$	
AE^*	P-complete (*Theorem 2*)
AA^+	NP-complete (*Corollary 1*)
$(\mathsf{E}^*\mathsf{A}^*)^k$, $k \geq 2$	
$(\mathsf{A}^*\mathsf{E}^*)^k$, $k \geq 1$	
$(\mathsf{A}^*\mathsf{E}^*)^*$	

Upper bound $(\forall \pi_1 . \exists \pi_2 . \psi)$

1. We begin by *marking* all leaves and proceed in several rounds, in which at least one mark is removed (*linear* rounds).

2. In each *round*, we go through all marked leaves v_1 and instantiate π_1 with the trace leading to v_1. We then again go through all marked leaves v_2 and instantiate π_2 with the trace leading to v_2, and check ψ on the pair of traces (*linear* time).

3. If successful for some instantiation of π_2, we leave v_1 marked, otherwise we remove the mark. If no mark was removed by the end of the round, we terminate.

Controller Synthesis for Trees

HyperLTL fragment	**Tree**
E^*	L-complete *(Theorem 1)*
$\mathsf{E}^*\mathsf{A}$	
$\mathsf{A}\mathsf{E}^*$	P-complete *(Theorem 2)*
$\mathsf{A}\mathsf{A}^+$	NP-complete *(Corollary 1)*
$(\mathsf{E}^*\mathsf{A}^*)^k$, $k \geq 2$	
$(\mathsf{A}^*\mathsf{E}^*)^k$, $k \geq 1$	
$(\mathsf{A}^*\mathsf{E}^*)^*$	

Controller Synthesis for Trees

Lower bound

HyperLTL fragment	Tree
E^*	L-complete *(Theorem 1)*
$\mathsf{E}^*\mathsf{A}$	
AE^*	P-complete *(Theorem 2)*
AA^+	NP-complete *(Corollary 1)*
$(\mathsf{E}^*\mathsf{A}^*)^k$, $k \geq 2$	
$(\mathsf{A}^*\mathsf{E}^*)^k$, $k \geq 1$	
$(\mathsf{A}^*\mathsf{E}^*)^*$	

Controller Synthesis for Trees

Lower bound

Reduction from the *horn SAT* problem

HyperLTL fragment	Tree
E^*	L-complete *(Theorem 1)*
$\mathsf{E}^*\mathsf{A}$	
AE^*	P-complete *(Theorem 2)*
AA^+	NP-complete *(Corollary 1)*
$(\mathsf{E}^*\mathsf{A}^*)^k$, $k \geq 2$	
$(\mathsf{A}^*\mathsf{E}^*)^k$, $k \geq 1$	
$(\mathsf{A}^*\mathsf{E}^*)^*$	

Controller Synthesis for Trees

Lower bound

HyperLTL fragment	**Tree**
E*	L-complete *(Theorem 1)*
E*A	
AE*	P-complete *(Theorem 2)*
AA$^+$	NP-complete *(Corollary 1)*
(E*A*)^k, $k \geq 2$	
(A*E*)^k, $k \geq 1$	
(A*E*)*	

Reduction from the *horn SAT* problem

$(\neg x_1 \vee \neg x_2 \vee \neg x_3 \vee x_4) \wedge (\neg x_2 \vee x_4) \wedge (\neg x_1)$

Controller Synthesis for Trees

Lower bound

HyperLTL fragment	**Tree**
E*	L-complete *(Theorem 1)*
E*A	
AE*	P-complete *(Theorem 2)*
AA+	NP-complete *(Corollary 1)*
(E*A*)^k, $k \geq 2$	
(A*E*)^k, $k \geq 1$	
(A*E*)*	

Reduction from the *horn SAT* problem

$$(\neg x_1 \vee \neg x_2 \vee \neg x_3 \vee x_4) \wedge (\neg x_2 \vee x_4) \wedge (\neg x_1)$$

$$= (\neg x_1 \vee \neg x_2 \vee f) \wedge (\neg x_3 \vee \neg f \vee x_4) \wedge (\neg x_2 \vee \neg x_2 \vee x_4) \wedge (\neg x_1 \vee \neg x_1 \vee \bot)$$

$$X = \{\bot, x_1, x_2, x_3, x_4, f, \top\}$$

Controller Synthesis for Trees

Lower bound

HyperLTL fragment	Tree
E*	L-complete *(Theorem 1)*
E*A	
AE*	P-complete *(Theorem 2)*
AA+	NP-complete *(Corollary 1)*
(E*A*)^k, $k \geq 2$	
(A*E*)^k, $k \geq 1$	
(A*E*)*	

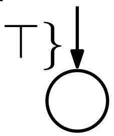
Reduction from the *horn SAT* problem

$$(\neg x_1 \vee \neg x_2 \vee \neg x_3 \vee x_4) \wedge (\neg x_2 \vee x_4) \wedge (\neg x_1)$$

$$= (\neg x_1 \vee \neg x_2 \vee f) \wedge (\neg x_3 \vee \neg f \vee x_4) \wedge$$

$$(\neg x_2 \vee \neg x_2 \vee x_4) \wedge (\neg x_1 \vee \neg x_1 \vee \bot)$$

$$X = \{\bot, x_1, x_2, x_3, x_4, f, \top\}$$



Controller Synthesis for Trees

Lower bound

HyperLTL fragment	**Tree**
E*	L-complete *(Theorem 1)*
E*A	
AE*	P-complete *(Theorem 2)*
AA$^+$	NP-complete *(Corollary 1)*
(E*A*)^k, $k \geq 2$	
(A*E*)^k, $k \geq 1$	
(A*E*)*	

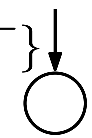
Reduction from the *horn SAT* problem

$$(\neg x_1 \vee \neg x_2 \vee \neg x_3 \vee x_4) \wedge (\neg x_2 \vee x_4) \wedge (\neg x_1)$$

$$= (\neg x_1 \vee \neg x_2 \vee \boxed{f}) \wedge (\neg x_3 \vee \neg f \vee \boxed{x_4}) \wedge$$

$$(\neg x_2 \vee \neg x_2 \vee x_4) \wedge (\neg x_1 \vee \neg x_1 \vee \boxed{\perp})$$

$$X = \{\perp, x_1, x_2, x_3, x_4, f, \top\}$$



Controller Synthesis for Trees

Lower bound

HyperLTL fragment	Tree
E^*	L-complete *(Theorem 1)*
$\mathsf{E}^*\mathsf{A}$	
$\mathsf{A}\mathsf{E}^*$	P-complete *(Theorem 2)*
$\mathsf{A}\mathsf{A}^+$	NP-complete *(Corollary 1)*
$(\mathsf{E}^*\mathsf{A}^*)^k$, $k \geq 2$	
$(\mathsf{A}^*\mathsf{E}^*)^k$, $k \geq 1$	
$(\mathsf{A}^*\mathsf{E}^*)^*$	

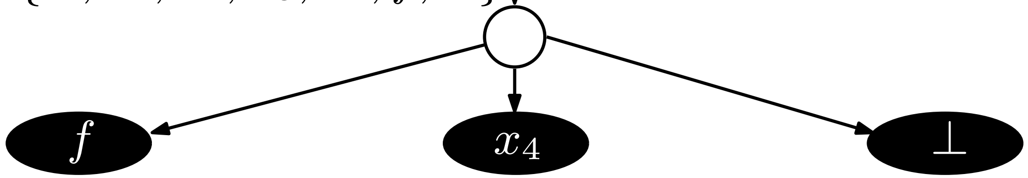
Reduction from the *horn SAT* problem

$$(\neg x_1 \vee \neg x_2 \vee \neg x_3 \vee x_4) \wedge (\neg x_2 \vee x_4) \wedge (\neg x_1)$$

$$= (\neg x_1 \vee \neg x_2 \vee \boxed{f}) \wedge (\neg x_3 \vee \neg f \vee \boxed{x_4}) \wedge$$

$$(\neg x_2 \vee \neg x_2 \vee x_4) \wedge (\neg x_1 \vee \neg x_1 \vee \boxed{\perp})$$

$$X = \{\perp, x_1, x_2, x_3, x_4, f, \top\}$$



Controller Synthesis for Trees

Lower bound

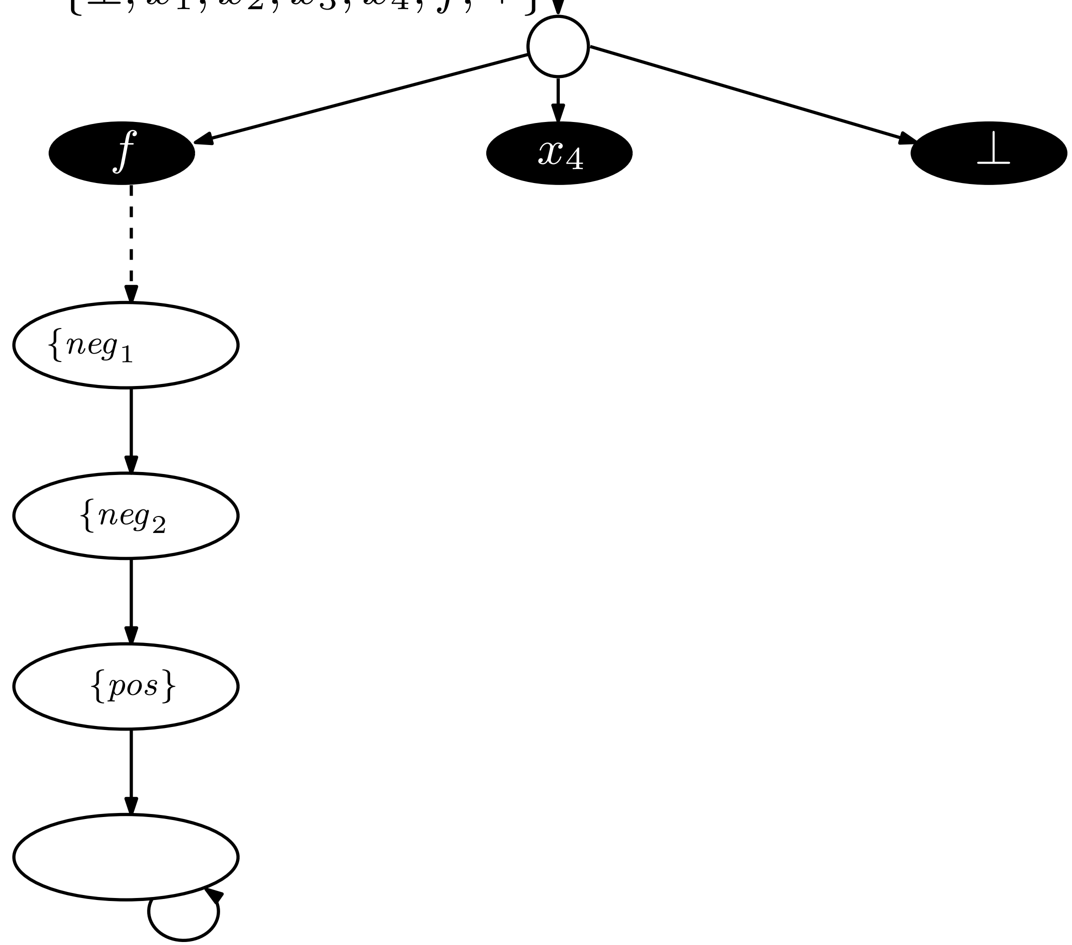
HyperLTL fragment	Tree
E*	L-complete *(Theorem 1)*
E*A	
AE*	P-complete *(Theorem 2)*
AA$^+$	NP-complete *(Corollary 1)*
(E*A*)^k, $k \geq 2$	
(A*E*)^k, $k \geq 1$	
(A*E*)*	

Reduction from the *horn SAT* problem

$$(\neg x_1 \vee \neg x_2 \vee \neg x_3 \vee x_4) \wedge (\neg x_2 \vee x_4) \wedge (\neg x_1)$$

$$= (\neg x_1 \vee \neg x_2 \vee \boxed{f}) \wedge (\neg x_3 \vee \neg f \vee \boxed{x_4}) \wedge$$
$$(\neg x_2 \vee \neg x_2 \vee x_4) \wedge (\neg x_1 \vee \neg x_1 \vee \boxed{\bot})$$

$X = \{\bot, x_1, x_2, x_3, x_4, f, \top\}$



Controller Synthesis for Trees

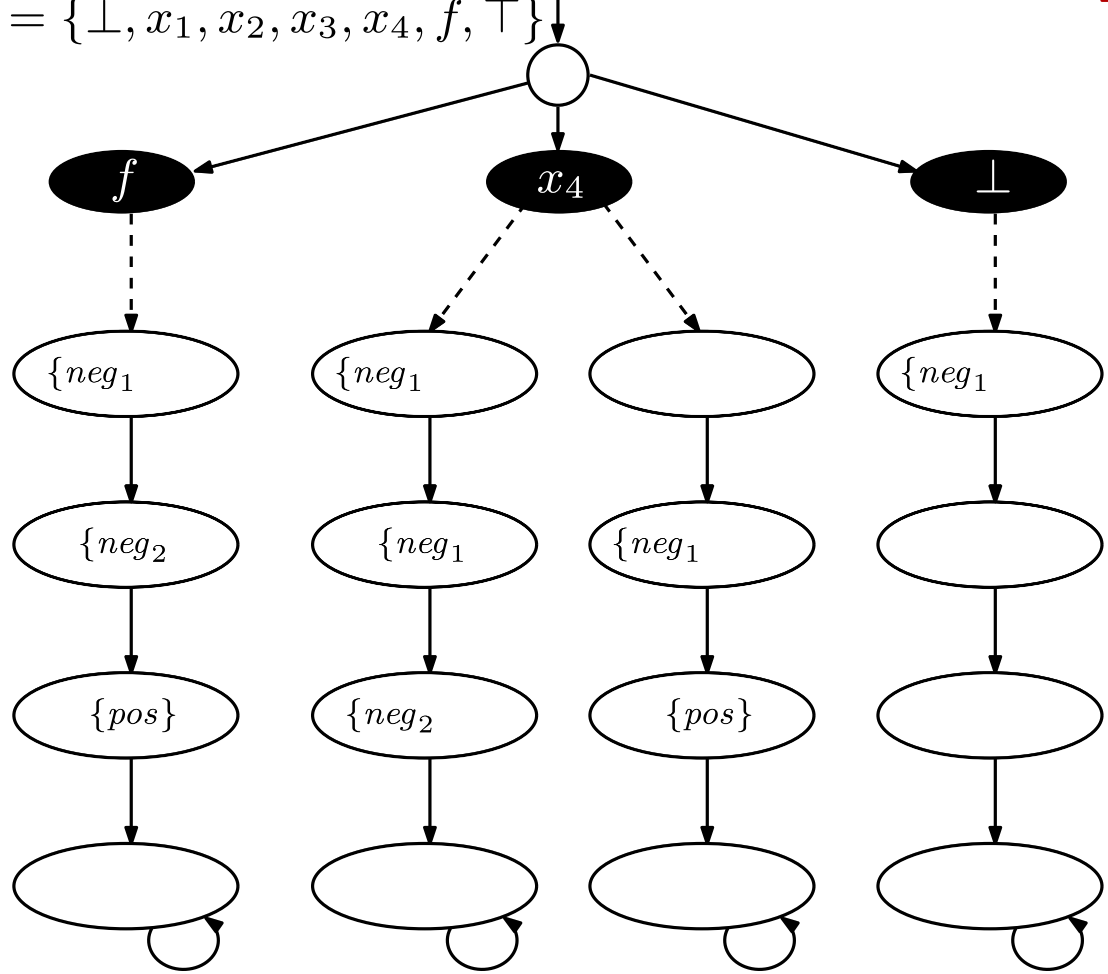
Lower bound

HyperLTL fragment	Tree
E*	L-complete *(Theorem 1)*
E*A	
AE*	P-complete *(Theorem 2)*
AA$^+$	NP-complete *(Corollary 1)*
(E*A*)^k, $k \geq 2$	
(A*E*)^k, $k \geq 1$	
(A*E*)*	

Reduction from the *horn SAT* problem

$$(\neg x_1 \vee \neg x_2 \vee \neg x_3 \vee x_4) \wedge (\neg x_2 \vee x_4) \wedge (\neg x_1)$$
$$= (\neg x_1 \vee \neg x_2 \vee \boxed{f}) \wedge (\neg x_3 \vee \neg f \vee \boxed{x_4}) \wedge (\neg x_2 \vee \neg x_2 \vee x_4) \wedge (\neg x_1 \vee \neg x_1 \vee \boxed{\bot})$$

$X = \{\bot, x_1, x_2, x_3, x_4, f, \top\}$



Controller Synthesis for Trees

Lower bound

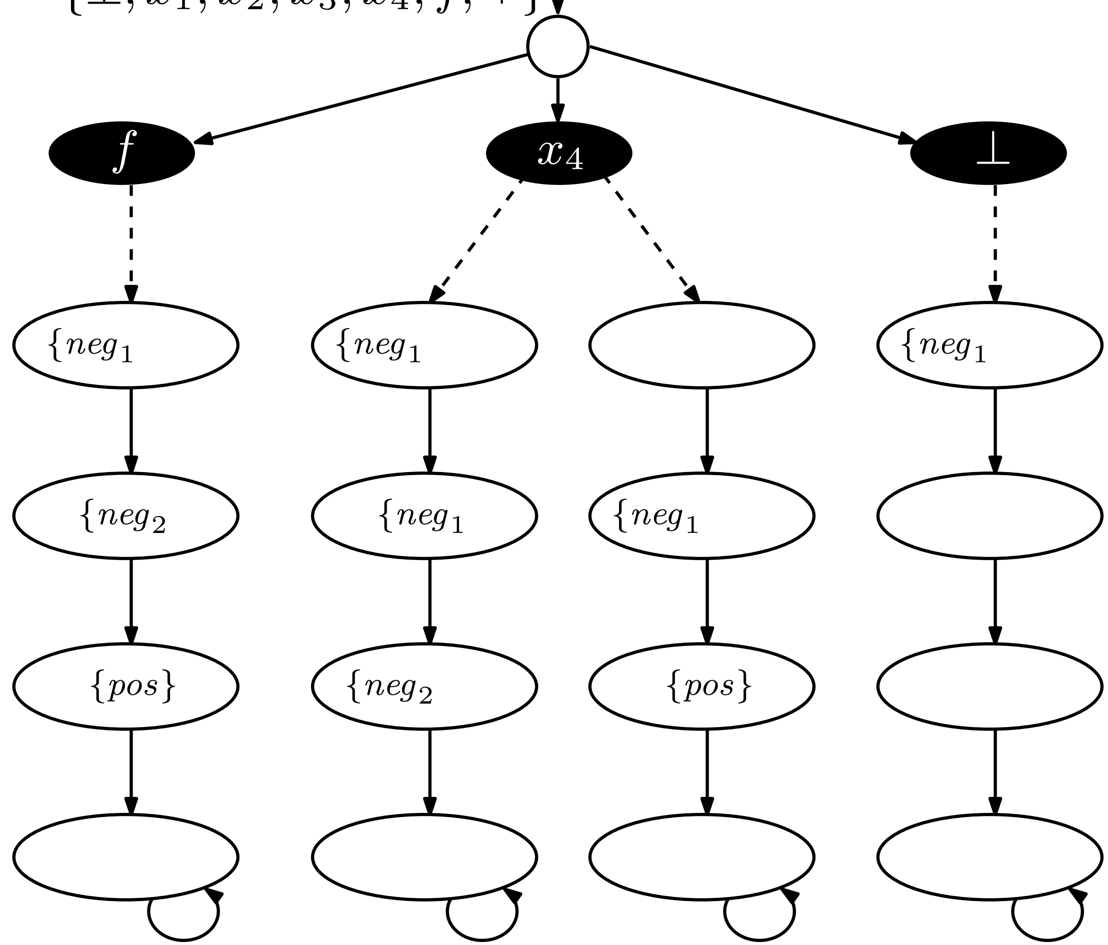
HyperLTL fragment	Tree
E*	L-complete *(Theorem 1)*
E*A	
AE*	P-complete *(Theorem 2)*
AA$^+$	
(E*A*)^k, $k \geq 2$	NP-complete *(Corollary 1)*
(A*E*)^k, $k \geq 1$	
(A*E*)*	

Reduction from the *horn SAT* problem

$$(\neg x_1 \vee \neg x_2 \vee \neg x_3 \vee x_4) \wedge (\neg x_2 \vee x_4) \wedge (\neg x_1)$$

$$= (\neg x_1 \vee \neg x_2 \vee \boxed{f}) \wedge (\neg x_3 \vee \neg f \vee \boxed{x_4}) \wedge (\neg x_2 \vee \neg x_2 \vee x_4) \wedge (\neg x_1 \vee \neg x_1 \vee \boxed{\bot})$$

$X = \{\bot, x_1, x_2, x_3, x_4, f, \top\}$



$$\varphi_{\text{map}} = \forall \pi_1. \exists \pi_2. \Diamond(\neg pos_{\pi_1}) \wedge \Box(\neg pos_{\pi_2}) \wedge \Box\Big((neg_{1_{\pi_1}} \leftrightarrow pos_{\pi_2}) \vee (neg_{2_{\pi_1}} \leftrightarrow pos_{\pi_2})\Big)$$

Controller Synthesis for Trees

HyperLTL fragment	**Tree**
E*	L-complete *(Theorem 1)*
E*A	
AE*	P-complete *(Theorem 2)*
AA$^+$	NP-complete *(Corollary 1)*
(E*A*)^k, $k \geq 2$	
(A*E*)^k, $k \geq 1$	
(A*E*)*	

Controller Synthesis for Trees

HyperLTL fragment	Tree
E^*	L-complete *(Theorem 1)*
$\mathsf{E}^*\mathsf{A}$	
$\mathsf{A}\mathsf{E}^*$	P-complete *(Theorem 2)*
$\mathsf{A}\mathsf{A}^+$	NP-complete *(Corollary 1)*
$(\mathsf{E}^*\mathsf{A}^*)^k$, $k \geq 2$	
$(\mathsf{A}^*\mathsf{E}^*)^k$, $k \geq 1$	
$(\mathsf{A}^*\mathsf{E}^*)^*$	

Controller Synthesis for Trees

HyperLTL fragment	Tree
E^*	L-complete *(Theorem 1)*
$\mathsf{E}^*\mathsf{A}$	
AE^*	P-complete *(Theorem 2)*
AA^+	NP-complete *(Corollary 1)*
$(\mathsf{E}^*\mathsf{A}^*)^k$, $k \geq 2$	
$(\mathsf{A}^*\mathsf{E}^*)^k$, $k \geq 1$	
$(\mathsf{A}^*\mathsf{E}^*)^*$	

Upper bound

1. *Guess* a solution to the synthesis problem
2. *Verify* the correctness of the solution

(using logarithmic counters for path assignments and temporal operators as before)

Controller Synthesis for Trees

HyperLTL fragment	Tree
E^*	L-complete *(Theorem 1)*
$\mathsf{E}^*\mathsf{A}$	
AE^*	P-complete *(Theorem 2)*
AA^+	NP-complete *(Corollary 1)*
$(\mathsf{E}^*\mathsf{A}^*)^k$, $k \geq 2$	
$(\mathsf{A}^*\mathsf{E}^*)^k$, $k \geq 1$	
$(\mathsf{A}^*\mathsf{E}^*)^*$	

Controller Synthesis for Trees

Lower bound

HyperLTL fragment	**Tree**
E^*	L-complete *(Theorem 1)*
$\mathsf{E}^*\mathsf{A}$	
AE^*	P-complete *(Theorem 2)*
AA^+	NP-complete *(Corollary 1)*
$(\mathsf{E}^*\mathsf{A}^*)^k$, $k \geq 2$	
$(\mathsf{A}^*\mathsf{E}^*)^k$, $k \geq 1$	
$(\mathsf{A}^*\mathsf{E}^*)^*$	

Controller Synthesis for Trees

Lower bound

HyperLTL fragment	Tree
E^*	L-complete *(Theorem 1)*
$\mathsf{E}^*\mathsf{A}$	
AE^*	P-complete *(Theorem 2)*
AA^+	NP-complete *(Corollary 1)*
$(\mathsf{E}^*\mathsf{A}^*)^k$, $k \geq 2$	
$(\mathsf{A}^*\mathsf{E}^*)^k$, $k \geq 1$	
$(\mathsf{A}^*\mathsf{E}^*)^*$	

Reduction from the *3SAT* problem

Controller Synthesis for Trees

Lower bound

HyperLTL fragment	Tree
E*	L-complete *(Theorem 1)*
E*A	
AE*	P-complete *(Theorem 2)*
AA$^+$	NP-complete *(Corollary 1)*
(E*A*)^k, $k \geq 2$	
(A*E*)^k, $k \geq 1$	
(A*E*)*	

Reduction from the *3SAT* problem

$$(\neg x_1 \vee \neg x_2 \vee x_3) \;\wedge\; (x_1 \vee x_2 \vee \neg x_4)$$

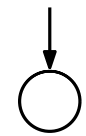
Controller Synthesis for Trees

Lower bound

HyperLTL fragment	Tree
E*	L-complete *(Theorem 1)*
E*A	
AE*	P-complete *(Theorem 2)*
AA+	NP-complete *(Corollary 1)*
(E*A*)^k, $k \geq 2$	
(A*E*)^k, $k \geq 1$	
(A*E*)*	

Reduction from the *3SAT* problem

$$(\neg x_1 \vee \neg x_2 \vee x_3) \ \wedge \ (x_1 \vee x_2 \vee \neg x_4)$$



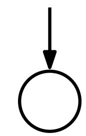
Controller Synthesis for Trees

Lower bound

HyperLTL fragment	Tree
E*	L-complete *(Theorem 1)*
E*A	
AE*	P-complete *(Theorem 2)*
AA$^+$	NP-complete *(Corollary 1)*
(E*A*)^k, $k \geq 2$	
(A*E*)^k, $k \geq 1$	
(A*E*)*	

Reduction from the *3SAT* problem

$$\underbrace{(\neg x_1 \vee \neg x_2 \vee x_3)}_{y_1} \wedge (x_1 \vee x_2 \vee \neg x_4)$$



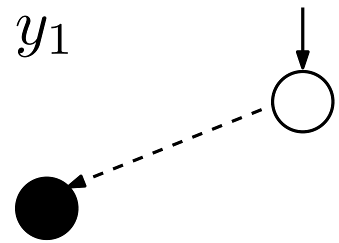
Controller Synthesis for Trees

Lower bound

HyperLTL fragment	**Tree**
E*	L-complete *(Theorem 1)*
E*A	
AE*	P-complete *(Theorem 2)*
AA$^+$	NP-complete *(Corollary 1)*
(E*A*)^k, $k \geq 2$	
(A*E*)^k, $k \geq 1$	
(A*E*)*	

Reduction from the *3SAT* problem

$$\underbrace{(\neg x_1 \vee \neg x_2 \vee x_3)}_{y_1} \wedge (x_1 \vee x_2 \vee \neg x_4)$$



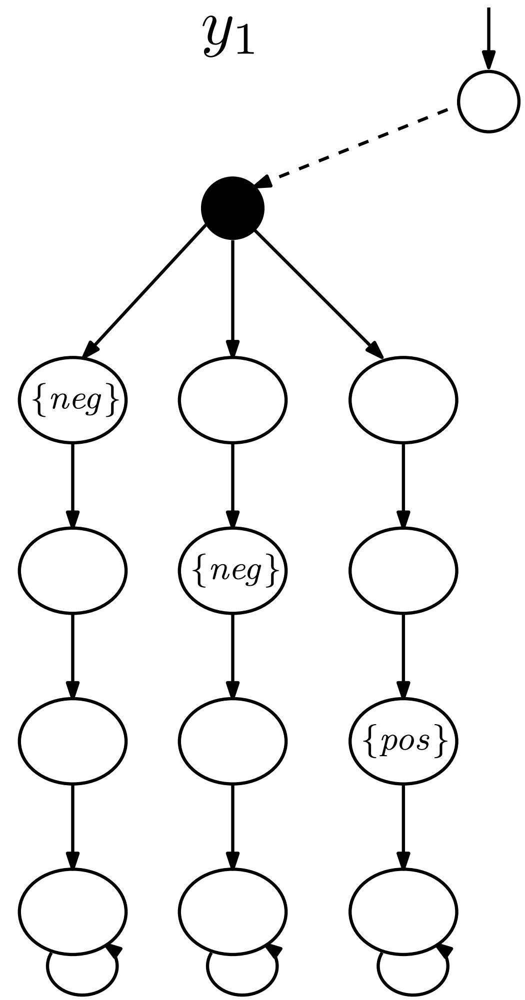
Controller Synthesis for Trees

Lower bound

HyperLTL fragment	Tree
E^*	L-complete *(Theorem 1)*
$\mathsf{E}^*\mathsf{A}$	
$\mathsf{A}\mathsf{E}^*$	P-complete *(Theorem 2)*
$\mathsf{A}\mathsf{A}^+$	NP-complete *(Corollary 1)*
$(\mathsf{E}^*\mathsf{A}^*)^k$, $k \geq 2$	
$(\mathsf{A}^*\mathsf{E}^*)^k$, $k \geq 1$	
$(\mathsf{A}^*\mathsf{E}^*)^*$	

Reduction from the *3SAT* problem

$$\underbrace{(\neg x_1 \vee \neg x_2 \vee x_3)}_{y_1} \wedge (x_1 \vee x_2 \vee \neg x_4)$$



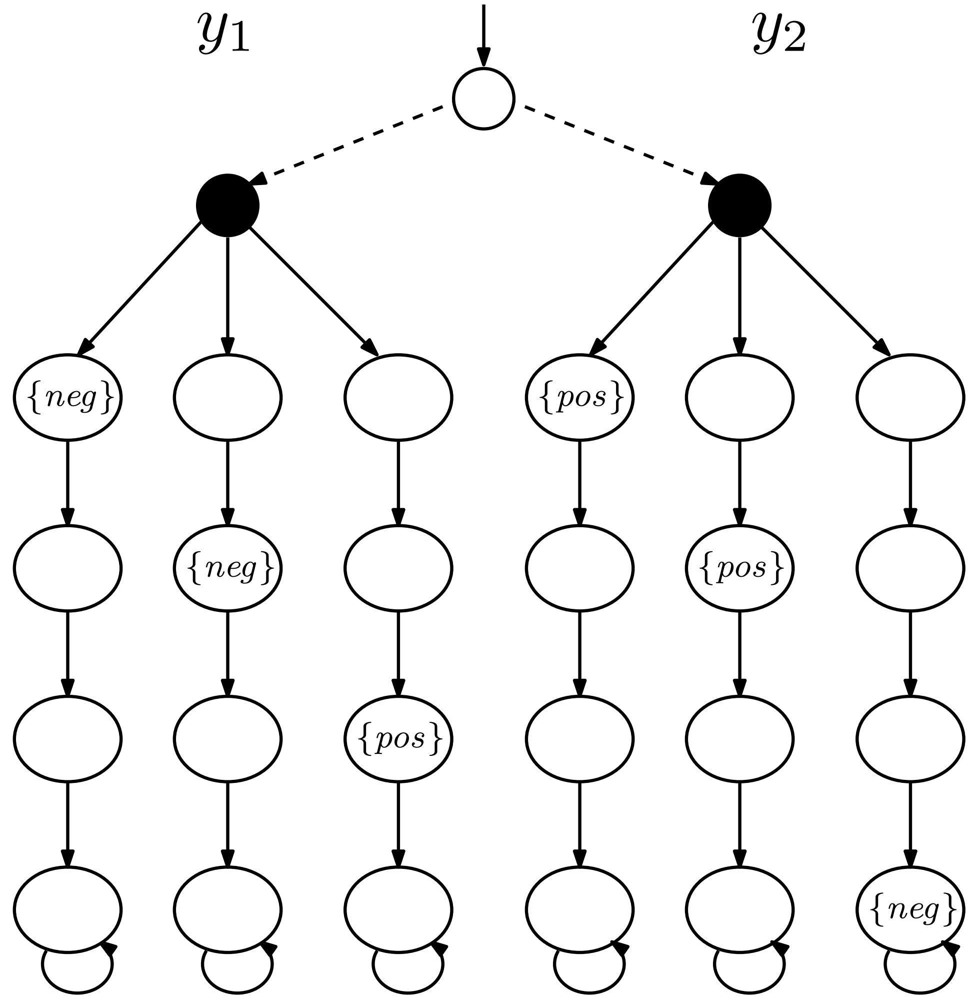
Controller Synthesis for Trees

Lower bound

HyperLTL fragment	Tree
E*	L-complete *(Theorem 1)*
E*A	
AE*	P-complete *(Theorem 2)*
AA$^+$	NP-complete *(Corollary 1)*
(E*A*)^k, $k \geq 2$	
(A*E*)^k, $k \geq 1$	
(A*E*)*	

Reduction from the *3SAT* problem

$$\underbrace{(\neg x_1 \vee \neg x_2 \vee x_3)}_{y_1} \wedge \underbrace{(x_1 \vee x_2 \vee \neg x_4)}_{y_2}$$



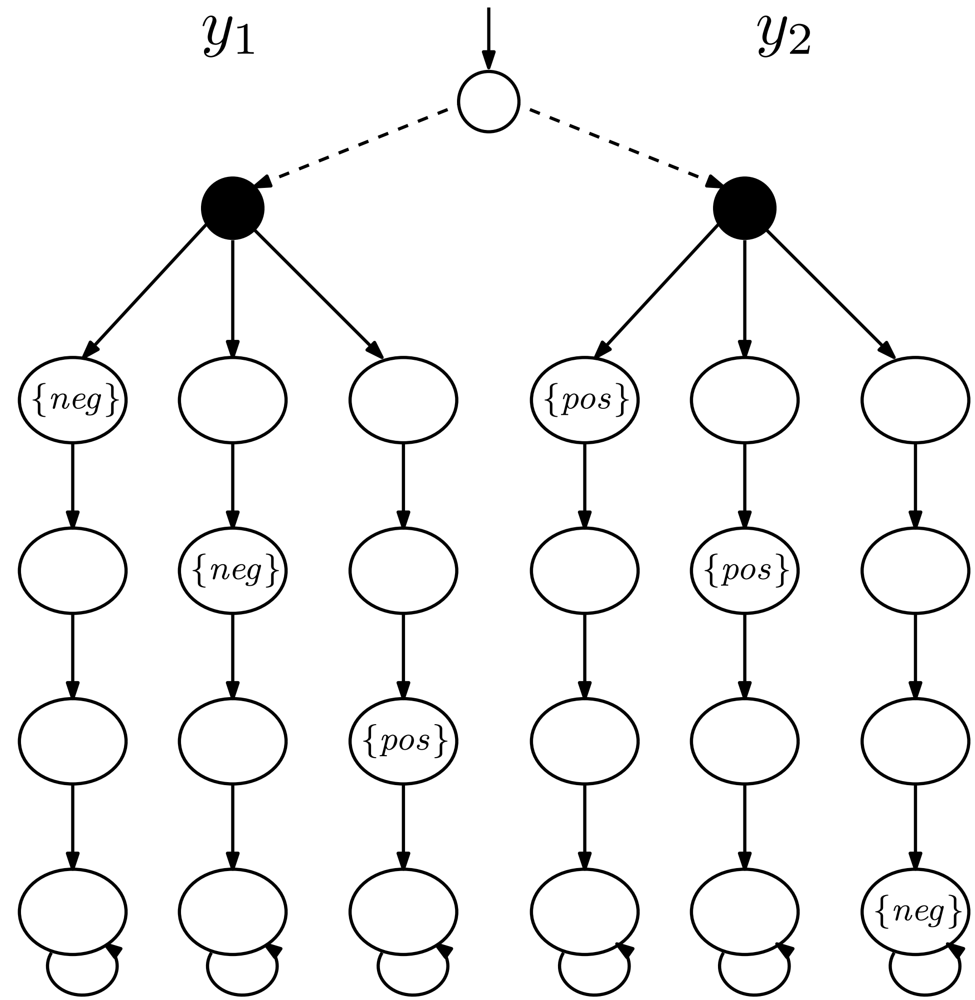
Controller Synthesis for Trees

Lower bound

HyperLTL fragment	**Tree**
E*	L-complete *(Theorem 1)*
E*A	
AE*	P-complete *(Theorem 2)*
AA$^+$	NP-complete *(Corollary 1)*
(E*A*)^k, $k \geq 2$	
(A*E*)^k, $k \geq 1$	
(A*E*)*	

Reduction from the *3SAT* problem

$$\underbrace{(\neg x_1 \lor \neg x_2 \lor x_3)}_{y_1} \land \underbrace{(x_1 \lor x_2 \lor \neg x_4)}_{y_2}$$



$$\varphi_{\mathsf{map}} = \forall \pi_1 . \forall \pi_2 . \Box \left(\neg pos_{\pi_1} \lor \neg neg_{\pi_2} \right)$$

Controller Synthesis for Trees

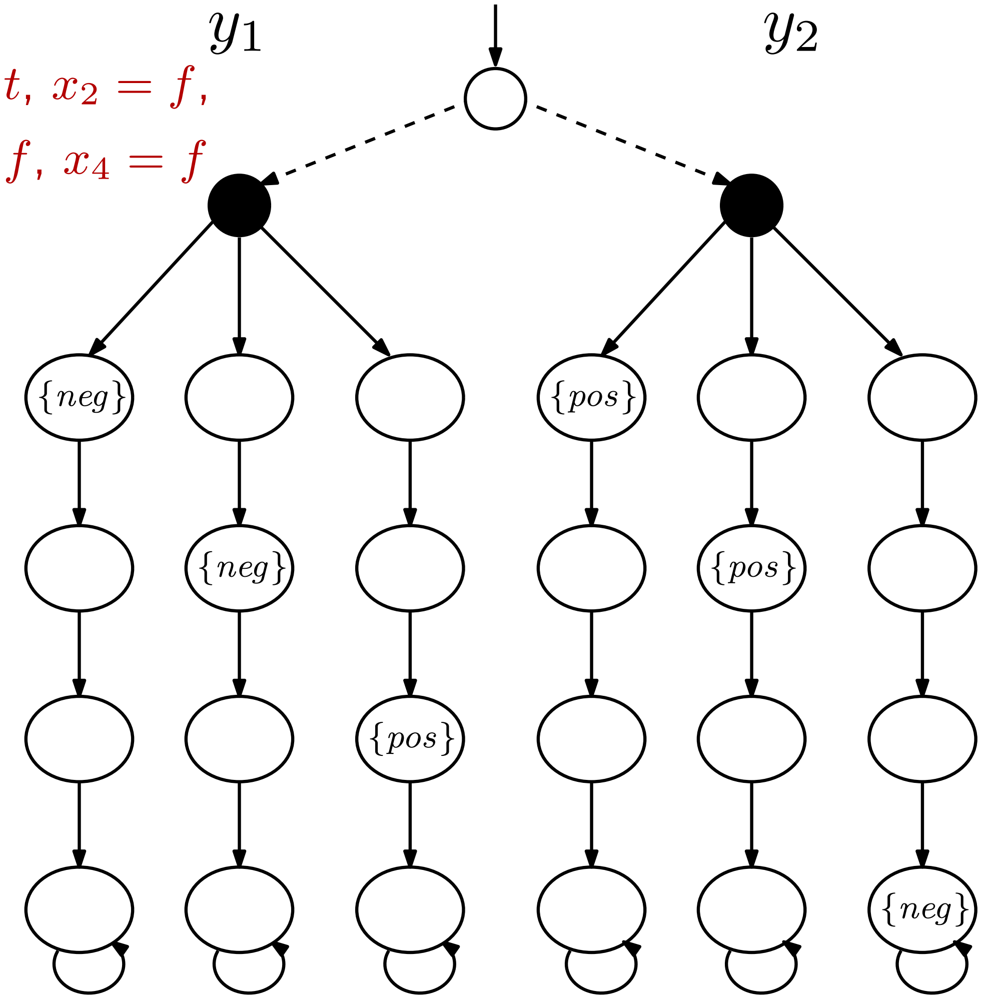
Lower bound

HyperLTL fragment	Tree
E*	L-complete *(Theorem 1)*
E*A	
AE*	P-complete *(Theorem 2)*
AA$^+$	NP-complete *(Corollary 1)*
(E*A*)^k, $k \geq 2$	
(A*E*)^k, $k \geq 1$	
(A*E*)*	

Reduction from the *3SAT* problem

$$\underbrace{(\neg x_1 \vee \neg x_2 \vee x_3)}_{y_1} \wedge \underbrace{(x_1 \vee x_2 \vee \neg x_4)}_{y_2}$$

$x_1 = t,\ x_2 = f,$
$x_3 = f,\ x_4 = f$



$$\varphi_{\mathsf{map}} = \forall \pi_1 . \forall \pi_2 . \square \left(\neg pos_{\pi_1} \ \vee \ \neg neg_{\pi_2} \right)$$

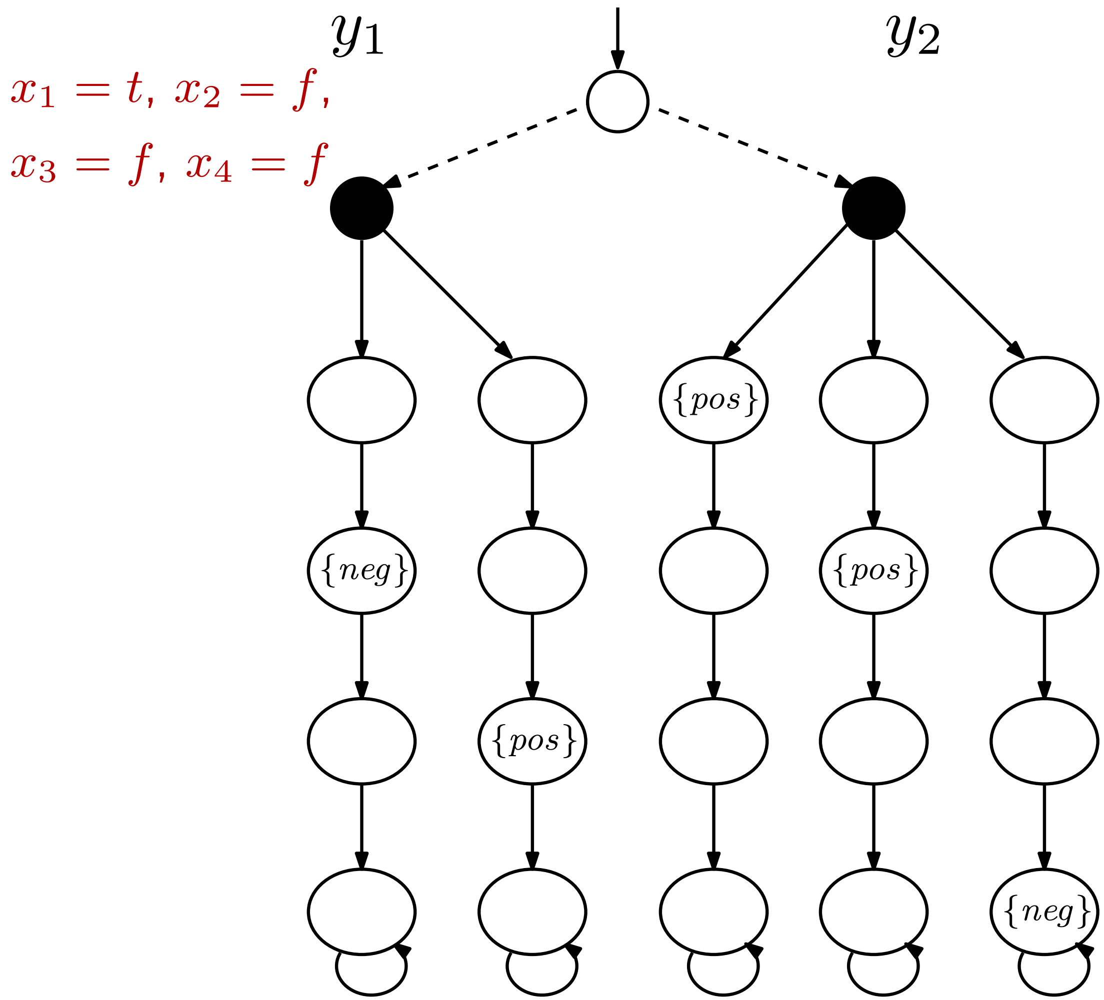
Controller Synthesis for Trees

Lower bound

HyperLTL fragment	Tree
E*	L-complete *(Theorem 1)*
E*A	
AE*	P-complete *(Theorem 2)*
AA+	NP-complete *(Corollary 1)*
(E*A*)^k, $k \geq 2$	
(A*E*)^k, $k \geq 1$	
(A*E*)*	

Reduction from the *3SAT* problem

$$\underbrace{(\neg x_1 \vee \neg x_2 \vee x_3)}_{y_1} \;\wedge\; \underbrace{(x_1 \vee x_2 \vee \neg x_4)}_{y_2}$$



$$\varphi_{\mathsf{map}} = \forall \pi_1. \forall \pi_2. \square \left(\neg pos_{\pi_1} \;\vee\; \neg neg_{\pi_2} \right)$$

Controller Synthesis for Trees

Lower bound

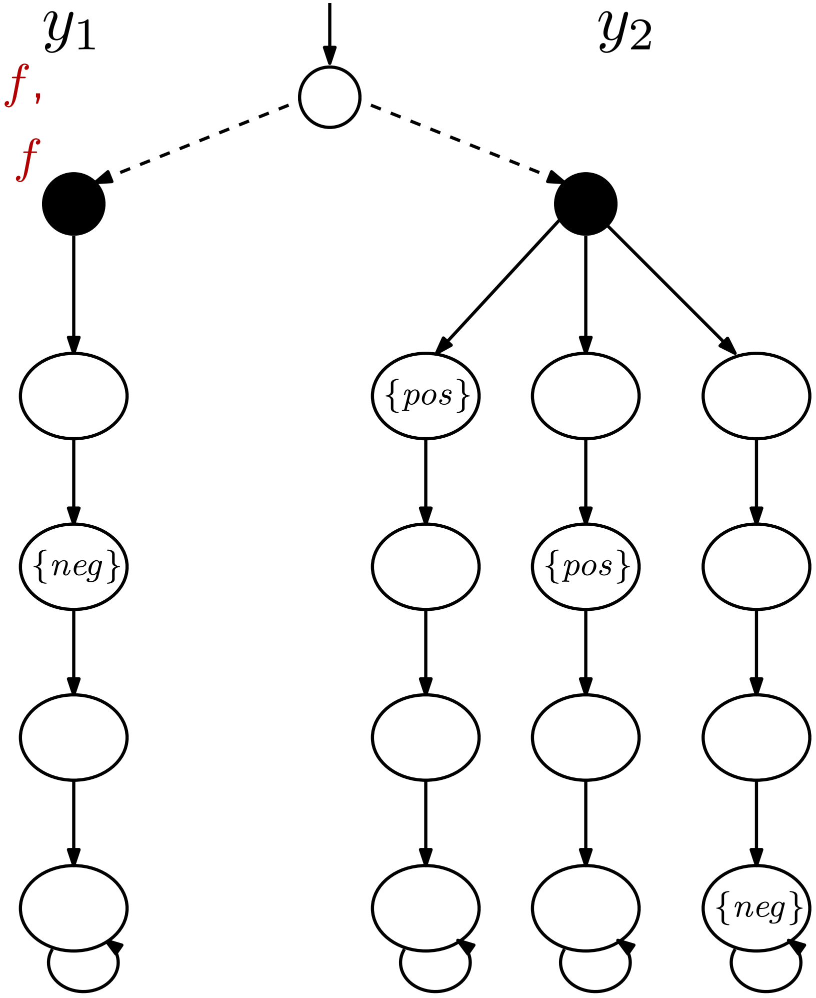
HyperLTL fragment	**Tree**
E^*	L-complete (*Theorem 1*)
$\mathsf{E}^*\mathsf{A}$	
AE^*	P-complete (*Theorem 2*)
AA^+	NP-complete (*Corollary 1*)
$(\mathsf{E}^*\mathsf{A}^*)^k$, $k \geq 2$	
$(\mathsf{A}^*\mathsf{E}^*)^k$, $k \geq 1$	
$(\mathsf{A}^*\mathsf{E}^*)^*$	

Reduction from the *3SAT* problem

$$\underbrace{(\neg x_1 \vee \neg x_2 \vee x_3)}_{y_1} \wedge \underbrace{(x_1 \vee x_2 \vee \neg x_4)}_{y_2}$$

$x_1 = t,\ x_2 = f,$

$x_3 = f,\ x_4 = f$



$$\varphi_{\mathsf{map}} = \forall \pi_1 . \forall \pi_2 . \Box \left(\neg pos_{\pi_1} \vee \neg neg_{\pi_2} \right)$$

Controller Synthesis for Trees

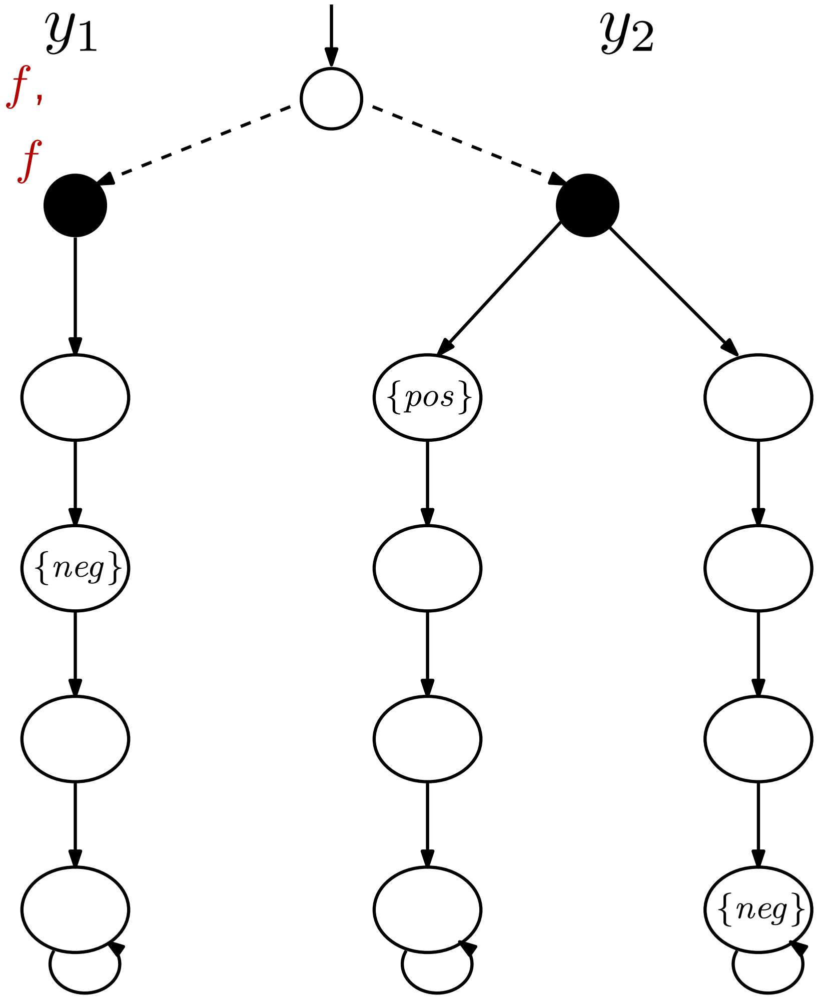
Lower bound

HyperLTL fragment	Tree
E*	L-complete *(Theorem 1)*
E*A	
AE*	P-complete *(Theorem 2)*
AA$^+$	NP-complete *(Corollary 1)*
(E*A*)^k, $k \geq 2$	
(A*E*)^k, $k \geq 1$	
(A*E*)*	

Reduction from the *3SAT* problem

$$\underbrace{(\neg x_1 \vee \neg x_2 \vee x_3)}_{y_1} \wedge \underbrace{(x_1 \vee x_2 \vee \neg x_4)}_{y_2}$$

$x_1 = t,\ x_2 = f,$
$x_3 = f,\ x_4 = f$



$$\varphi_{\mathsf{map}} = \forall \pi_1 . \forall \pi_2 . \Box \left(\neg pos_{\pi_1} \ \vee \ \neg neg_{\pi_2} \right)$$

Controller Synthesis for Trees

<table>
<tr><th>HyperLTL fragment</th><th>Tree (Controller Synthesis)</th><th>Tree [BF18] (Verification)</th></tr>
<tr><td>E^*</td><td rowspan="2">L-complete (Theorem 1)</td><td rowspan="7">L-complete</td></tr>
<tr><td>$\mathsf{E}^*\mathsf{A}$</td></tr>
<tr><td>AE^*</td><td>P-complete (Theorem 2)</td></tr>
<tr><td>AA^+</td><td rowspan="4">NP-complete (Corollary 1)</td></tr>
<tr><td>$(\mathsf{E}^*\mathsf{A}^*)^k$, $k \geq 2$</td></tr>
<tr><td>$(\mathsf{A}^*\mathsf{E}^*)^k$, $k \geq 1$</td></tr>
<tr><td>$(\mathsf{A}^*\mathsf{E}^*)^*$</td></tr>
</table>

- (BF18) Borzoo Bonakdarpour, Bernd Finkbeiner, *The complexity of monitoring hyperproperties*. CSF 2018.

Controller Synthesis for Trees

<table>
<tr><th>HyperLTL fragment</th><th>Tree (Controller Synthesis)</th><th>Tree [BF18] (Verification)</th></tr>
<tr><td>E^*</td><td rowspan="2">L-complete (Theorem 1)</td><td rowspan="7">L-complete</td></tr>
<tr><td>$\mathsf{E}^*\mathsf{A}$</td></tr>
<tr><td>AE^*</td><td>P-complete (Theorem 2)</td></tr>
<tr><td>AA^+</td><td rowspan="4">NP-complete (Corollary 1)</td></tr>
<tr><td>$(\mathsf{E}^*\mathsf{A}^*)^k$, $k \geq 2$</td></tr>
<tr><td>$(\mathsf{A}^*\mathsf{E}^*)^k$, $k \geq 1$</td></tr>
<tr><td>$(\mathsf{A}^*\mathsf{E}^*)^*$</td></tr>
</table>

- (BF18) Borzoo Bonakdarpour, Bernd Finkbeiner, *The complexity of monitoring hyperproperties*. CSF 2018.

4. Controller Synthesis for Acyclic Graphs

Controller Synthesis for Acyclic Graphs

Upper bound

HyperLTL fragment	Acyclic
E*	NL-complete *(Theorem 5)*
E*A	Σ_2^p
AE*	Σ_2^p-complete *(Theorem 8)*
AA$^+$	NP-complete *(Theorem 6)*
(E*A*)^k, $k \geq 2$	Σ_k^p-complete *(Theorem 8)*
(A*E*)^k, $k \geq 1$	Σ_{k+1}^p-complete *(Theorem 8)*
(A*E*)*	PSPACE *(Corollary 3)*

Controller Synthesis for Acyclic Graphs

HyperLTL fragment	Acyclic
E^*	NL-complete *(Theorem 5)*
$\mathsf{E}^*\mathsf{A}$	Σ_2^p
$\mathsf{A}\mathsf{E}^*$	Σ_2^p-complete *(Theorem 8)*
$\mathsf{A}\mathsf{A}^+$	NP-complete *(Theorem 6)*
$(\mathsf{E}^*\mathsf{A}^*)^k$, $k \geq 2$	Σ_k^p-complete *(Theorem 8)*
$(\mathsf{A}^*\mathsf{E}^*)^k$, $k \geq 1$	Σ_{k+1}^p-complete *(Theorem 8)*
$(\mathsf{A}^*\mathsf{E}^*)^*$	PSPACE *(Corollary 3)*

Upper bound

1. *Guess* a solution to synthesis + path assignment for leading $\exists^*$

2. *Verify* the remaining formula (model checking is in Π_k^p)

Controller Synthesis for Acyclic Graphs

HyperLTL fragment	**Acyclic** *(Controller Synthesis)*	**Acyclic [BF18]** *(Verification)*
E^*	NL-complete *(Theorem 5)*	NL-complete
$\mathsf{E}^*\mathsf{A}$		
AA^+	NP-complete *(Theorem 6)*	
AE^*	Σ_2^p-complete *(Theorem 8)*	Π_2^p-complete
$(\mathsf{E}^*\mathsf{A}^*)^k$, $k \geq 2$	Σ_k^p-complete *(Theorem 8)*	Σ_k^p-complete
$(\mathsf{A}^*\mathsf{E}^*)^k$, $k \geq 1$	Σ_{k+1}^p-complete *(Theorem 8)*	Π_k^p-complete
$(\mathsf{A}^*\mathsf{E}^*)^*$	PSPACE *(Corollary 3)*	PSPACE

Controller Synthesis for Acyclic Graphs

HyperLTL fragment	**Acyclic** *(Controller Synthesis)*	**Acyclic [BF18]** *(Verification)*
E*	NL-complete *(Theorem 5)*	NL-complete
E*A		
AA$^+$	NP-complete *(Theorem 6)*	
AE*	Σ_2^p-complete *(Theorem 8)*	Π_2^p-complete
(E*A*)^k, $k \geq 2$	Σ_k^p-complete *(Theorem 8)*	Σ_k^p-complete
(A*E*)^k, $k \geq 1$	Σ_{k+1}^p-complete *(Theorem 8)*	Π_k^p-complete
(A*E*)*	PSPACE *(Corollary 3)*	PSPACE

- (BF18) Borzoo Bonakdarpour, Bernd Finkbeiner, *The complexity of monitoring hyperproperties*. CSF 2018.

5. Controller Synthesis for General Graphs

Controller Synthesis for General Graphs

HyperLTL fragment	**General**
E*	NL-complete *(Theorem 9)*
E*A	PSPACE-complete *(Theorem 11)*
AE*	
AA$^+$	NP-complete *(Theorem 10)*
(E*A*)^k, $k \geq 2$	$(k-1)$-EXPSPACE-complete *(Theorem 11)*
(A*E*)^k, $k \geq 1$	
(A*E*)*	NONELEMENTARY *(Corollary 4)*

Upper bound

Controller Synthesis for General Graphs

HyperLTL fragment	General
E^*	NL-complete *(Theorem 9)*
$\mathsf{E}^*\mathsf{A}$	PSPACE-complete *(Theorem 11)*
AE^*	
AA^+	NP-complete *(Theorem 10)*
$(\mathsf{E}^*\mathsf{A}^*)^k$, $k \geq 2$	$(k-1)$-EXPSPACE-complete *(Theorem 11)*
$(\mathsf{A}^*\mathsf{E}^*)^k$, $k \geq 1$	
$(\mathsf{A}^*\mathsf{E}^*)^*$	NONELEMENTARY *(Corollary 4)*

Upper bound

Synthesis is *dominated* by verification

1. Guess a solution to the synthesis problem

2. Verify

Controller Synthesis for General Graphs

HyperLTL fragment	**General** *(Controller Synthesis)*	**General [BF18]** *(Verification)*
E^*	NL-complete *(Theorem 9)*	NL-complete
AA^+	NP-complete *(Theorem 10)*	
$\mathsf{E}^*\mathsf{A}$	PSPACE-complete *(Theorem 11)*	PSPACE-complete
AE^*		
$(\mathsf{E}^*\mathsf{A}^*)^k$, $k \geq 2$	$(k-1)$-EXPSPACE-complete *(Theorem 11)*	$(k-1)$-EXPSPACE-complete
$(\mathsf{A}^*\mathsf{E}^*)^k$, $k \geq 1$		
$(\mathsf{A}^*\mathsf{E}^*)^*$	NONELEMENTARY *(Corollary 4)*	NONELEMENTARY

Controller Synthesis for General Graphs

HyperLTL fragment	General *(Controller Synthesis)*	General [BF18] *(Verification)*
E*	NL-complete *(Theorem 9)*	NL-complete
AA$^+$	NP-complete *(Theorem 10)*	
E*A	PSPACE-complete *(Theorem 11)*	PSPACE-complete
AE*		
(E*A*)^k, $k \geq 2$	$(k-1)$-EXPSPACE-complete *(Theorem 11)*	$(k-1)$-EXPSPACE-complete
(A*E*)^k, $k \geq 1$		
(A*E*)*	NONELEMENTARY *(Corollary 4)*	NONELEMENTARY

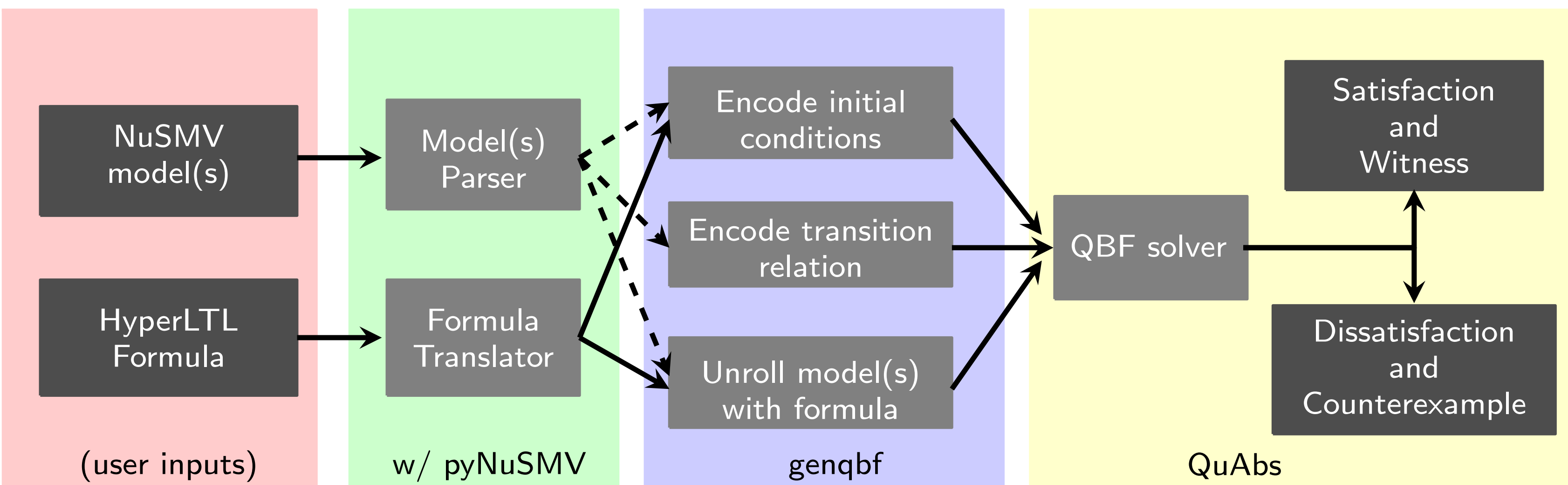
5. The Tool HyperQube

- Tzu-Han Hsu, César Sánchez, and Borzoo Bonakdarpour: *Bounded Model Checking for Hyperproperties*. TACAS 2021: 94-112
- Tzu-Han Hsu, Borzoo Bonakdarpour, and César Sánchez: *HyperQube: A QBF-Based Bounded Model Checker for Hyperproperties*. CoRR abs/2109.12989 (2021)

HyperQube

- HyperQube is a push-button *QBF-based* bounded model checker for hyperproperties.

- Unlike the existing similar tools, the QBF-based technique allows HyperQube to seamlessly deal with *quantifier alternations*.



- Tzu-Han Hsu, César Sánchez, and Borzoo Bonakdarpour: *Bounded Model Checking for Hyperproperties*. TACAS 2021: 94-112
- Tzu-Han Hsu, Borzoo Bonakdarpour, and César Sánchez: *HyperQube: A QBF-Based Bounded Model Checker for Hyperproperties*. CoRR abs/2109.12989 (2021)

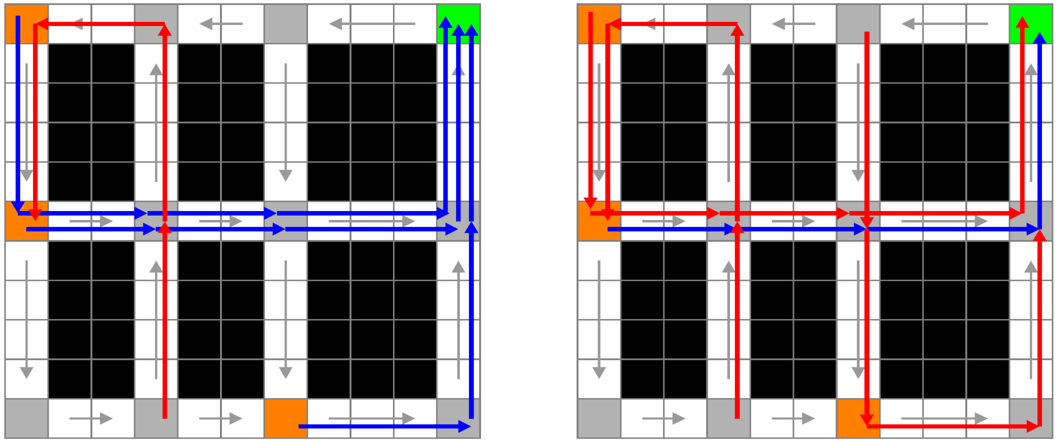
HyperQube Performance

	#	Model K	Formula	QBF	sem	states	k	parseSMV (sec)	genqbf (sec)	QuAbS (sec)	**Total** (sec)	
Symmetry in HW	0.1	Bakery.3proc	φ_{S1}	SAT	*pes*	167	10	0.33	0.84	0.33	**1.50**	✗
	0.2	Bakery.3proc	φ_{S2}	SAT	*pes*	167	10	0.32	0.94	0.38	**1.64**	✗
	0.3	Bakery.3proc	φ_{S3}	UNSAT	*opt*	167	10	0.34	0.84	0.36	**1.54**	✓
	1.1	Bakery.3proc	φ_{sym1}	SAT	*pes*	167	10	0.36	0.85	0.36	**1.57**	✗
	1.2	Bakery.3proc	φ_{sym2}	SAT	*pes*	167	10	0.53	0.83	0.48	**1.84**	✗
	1.3	Bakery.5proc	φ_{sym1}	SAT	*pes*	996	10	1.73	11.88	8.17	**21.78**	✗
	1.4	Bakery.5proc	φ_{sym2}	SAT	*pes*	996	10	1.52	12.40	7.66	**21.58**	✗
Linearizability	2.1	SNARK-bug1	φ_{lin}	SAT	*pes*	4914/548	18	49.13	119.90	429.16	**598.19**	✗
	2.2	SNARK-bug2	φ_{lin}	SAT	*pes*	3405/664	30	50.57	407.54	327.02	**785.13**	✗
Information-flow Security	3.1	$3\text{-}Thread_{incorrect}$	φ_{NI}	SAT	*h-pes*	368	50	0.50	8.61	5.47	**14.58**	✗
	3.2	$3\text{-}Thread_{correct}$	φ_{NI}	UNSAT	*h-opt*	64	50	0.24	1.45	0.68	**2.37**	✓
	4.1	$NRP : T_{incorrect}$	φ_{fair}	SAT	*h-pes*	55	15	0.23	0.39	0.28	**0.90**	✗
	4.2	$NRP : T_{correct}$	φ_{fair}	UNSAT	*h-opt*	54	15	0.24	0.41	0.49	**1.14**	✓
	5.1	Shortest Path	(see Table 5)									synthesis
	5.2	Initial State Robustness										
Mutation Testing	6.1	Mutant	φ_{mut}	SAT	*h-pes*	32	10	0.20	0.17	0.09	**0.46**	

Synthesis using HyperQube

- Adversarial multi-agent path planning

Prop.	# adv.	# agents	QS	$size$	h	Total[s]
φ_{react}	1	1	$\forall\exists$	10^2	20	**3.78**
				20^2	40	**95.50**
				30^2	60	**1597.70**
	2	1	$\forall\forall\exists$	10^2	20	**10.13**
				20^2	40	**597.86**
				30^2	60	**5627.61**
	3	1	$\forall\forall\forall\exists$	10^2	20	**13.66**
				20^2	40	**407.62**
				30^2	60	**5370.74**
	1	2	$\forall\exists\exists$	10^2	20	**14.41**
				20^2	40	**973.98**
				30^2	60	**16785.63**
	1	3	$\forall\exists\exists\exists$	10^2	20	**17.65**
				20^2	40	**1559.10**
				30^2	60	**68059.38**



(a) Multi-agent vs one Adversary

(b) One agent vs Multi-adversary

Controller Synthesis using HyperQube

- *Non-repudiation*:

$$\varphi = \exists\pi.\forall\pi'.\ (\Diamond\, m_\pi) \land (\Diamond\, NRR_\pi) \land (\Diamond\, NRO_\pi)$$

(effectiveness)

$$\land \Big((\Box \textstyle\bigwedge_{a \in Act_A} a_\pi \Leftrightarrow a_{\pi'}) \Rightarrow \big((\Diamond\, NRR_{\pi'}) \Leftrightarrow (\Diamond\, NRO_{\pi'}) \big) \Big)$$

(fairness for A)

$$\land \Big((\Box \textstyle\bigwedge_{a \in Act_B} a_\pi \Leftrightarrow a_{\pi'}) \Rightarrow \big((\Diamond\, NRR_{\pi'}) \Leftrightarrow (\Diamond\, NRO_{\pi'}) \big) \Big)$$

(fairness for B)

Controller Synthesis using HyperQube

- *Non-repudiation*:

$$\varphi = \exists \pi . \forall \pi' . \ (\Diamond m_{\pi}) \wedge (\Diamond NRR_{\pi}) \wedge (\Diamond NRO_{\pi})$$

(effectiveness)

$$\wedge \Big((\Box \bigwedge_{a \in Act_A} a_{\pi} \Leftrightarrow a_{\pi'}) \Rightarrow \big((\Diamond NRR_{\pi'}) \Leftrightarrow (\Diamond NRO_{\pi'}) \big) \Big)$$

(fairness for A)

$$\wedge \Big((\Box \bigwedge_{a \in Act_B} a_{\pi} \Leftrightarrow a_{\pi'}) \Rightarrow \big((\Diamond NRR_{\pi'}) \Leftrightarrow (\Diamond NRO_{\pi'}) \big) \Big)$$

(fairness for B)

- We ran HyperQube *iteratively*, where each round finds a new witness to the existential quantifier in formula until there is no more such trace.

- We synthesized the correct non-repudiation protocol in only $0.8s$.

Controller Synthesis using HyperQube

- *Non-repudiation*:

$$\varphi = \exists \pi . \forall \pi' . \ (\Diamond m_\pi) \wedge (\Diamond NRR_\pi) \wedge (\Diamond NRO_\pi)$$

(effectiveness)

$$\wedge \Big((\Box \textstyle\bigwedge_{a \in Act_A} a_\pi \Leftrightarrow a_{\pi'}) \Rightarrow \big((\Diamond NRR_{\pi'}) \Leftrightarrow (\Diamond NRO_{\pi'}) \big) \Big)$$

(fairness for A)

$$\wedge \Big((\Box \textstyle\bigwedge_{a \in Act_B} a_\pi \Leftrightarrow a_{\pi'}) \Rightarrow \big((\Diamond NRR_{\pi'}) \Leftrightarrow (\Diamond NRO_{\pi'}) \big) \Big)$$

(fairness for B)

- We ran HyperQube *iteratively*, where each round finds a new witness to the existential quantifier in formula until there is no more such trace.

- We synthesized the correct non-repudiation protocol in only $0.8s$.

(1) skip until $A{:}m{\to}T$; (2) skip until $A{:}NRO{\to}T$; (3) $T{\to}B{:}m$;
(4) skip until $B{\to}T{:}NRR$; (5) $T{\to}B{:}NRO$; (6) $T{\to}A{:}NRR$;

6. Conclusion

Conclusion

- **Summary**
  - *Controller synthesis* is a promising approach to synthesize secure systems
  - Similar complexity to *verification*
  - Potential for *scalable* algorithms and tools for relevant fragments

- **Future work**
  - Hyperlogics beyond HyperLTL (e.g., HyperCTL*, FO/SO hyperlogics)
  - Controller synthesis beyond finite state spaces
  - *Syntax-guided* synthesis for hyperproperties

Thanks!