Game-based Model-Checking of HyperLTL

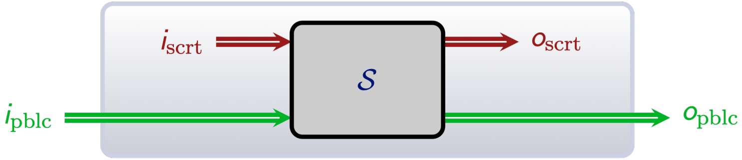
Martin Zimmermann

Aalborg University

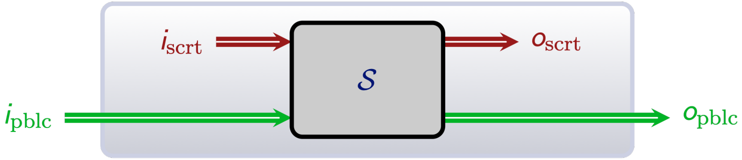
July 2025

TU Dortmund

Reactive Systems

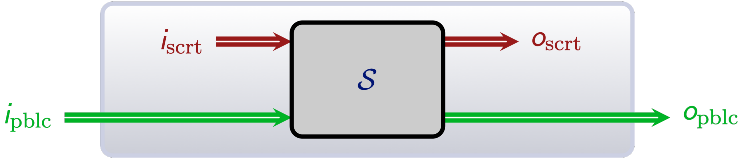


Reactive Systems



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

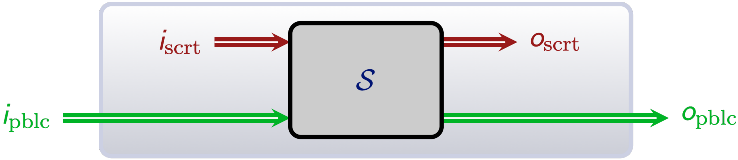
Reactive Systems



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

$\{\mathtt{init}, \mathtt{i}_{\mathsf{pblc}}\}$

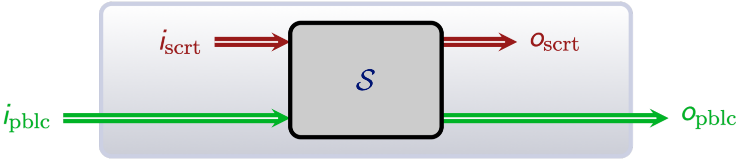
Reactive Systems



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

$\{\mathtt{init}, \mathtt{i}_{\mathsf{pblc}}\}$ $\{\mathtt{i}_{\mathsf{scrt}}\}$

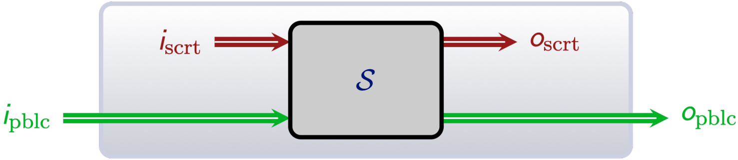
Reactive Systems



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

$\{\mathtt{init}, \mathtt{i}_{\mathsf{pblc}}\}$ $\{\mathtt{i}_{\mathsf{scrt}}\}$ $\{\mathtt{i}_{\mathsf{pblc}}\}$

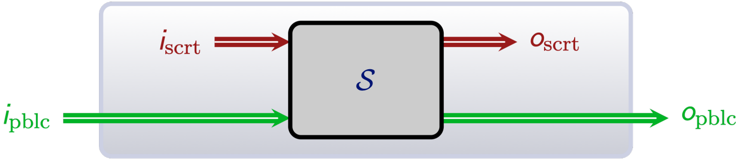
Reactive Systems



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

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

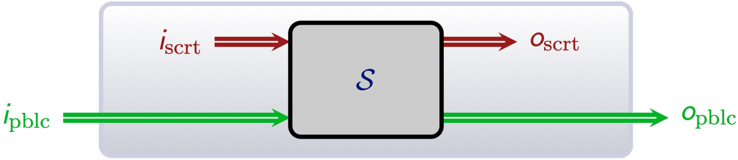
Reactive Systems



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

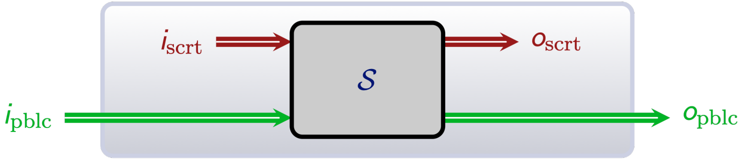
$\{\mathtt{init}, \mathtt{i}_{\mathsf{pblc}}\} \quad \{\mathtt{i}_{\mathsf{scrt}}\} \quad \{\mathtt{i}_{\mathsf{pblc}}\} \quad \{\mathtt{i}_{\mathsf{scrt}}, \mathtt{o}_{\mathsf{pblc}}, \mathtt{term}\} \quad \emptyset \cdots$

Reactive Systems



Typical specifications:

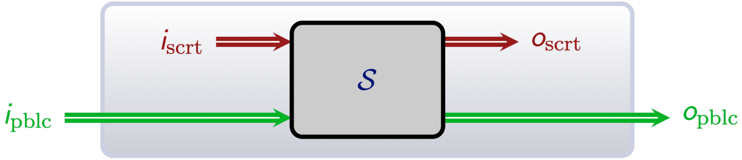
Reactive Systems



Typical specifications:

- $\mathcal{S}$ terminates

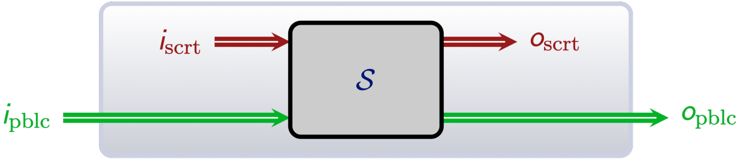
Reactive Systems



Typical specifications:

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

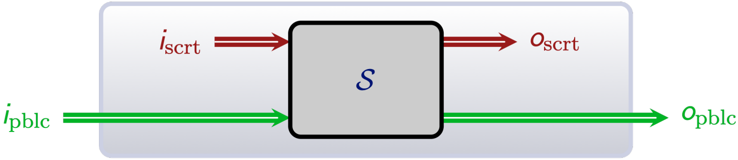
Reactive Systems



Typical specifications:

- Noninterference: for all traces t, t' of $\mathcal{S}$, if t and t' coincide on their projection to their public inputs, then they also coincide on their projection to the public outputs.

Reactive Systems



Typical specifications:

- Noninterference: for all traces t, t' of $\mathcal{S}$, if t and t' coincide on their projection to their public inputs, then they also coincide on their projection to the public outputs.
- Noninterference for nondeterministic systems: for all traces t, t' of $\mathcal{S}$ there exists a trace t'' of $\mathcal{S}$ such that t'' and t coincide on their projection to public inputs and outputs and t'' and t' coincide on their projection to secret inputs.

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

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

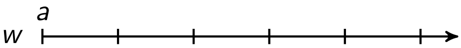
LTL in One Slide

Syntax

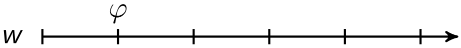
$$\varphi ::= a \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathbf{X}\,\varphi \mid \varphi\,\mathbf{U}\,\varphi \qquad \text{where } a \in \mathrm{AP}$$

Semantics

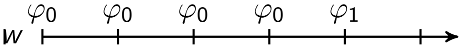
- $w \models a$:



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



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



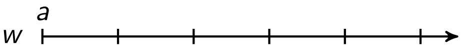
LTL in One Slide

Syntax

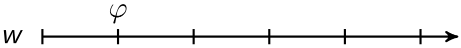
$$\varphi ::= a \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathbf{X}\,\varphi \mid \varphi\,\mathbf{U}\,\varphi \qquad \text{where } a \in \mathrm{AP}$$

Semantics

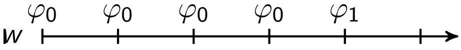
- $w \models a$:



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



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



Syntactic Sugar

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

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

- Noninterference:

$$\forall\pi\forall\pi'.\ \mathbf{G}((i_{\mathsf{pblc}})_\pi \leftrightarrow (i_{\mathsf{pblc}})_{\pi'}) \rightarrow \mathbf{G}((o_{\mathsf{pblc}})_\pi \leftrightarrow (o_{\mathsf{pblc}})_{\pi'})$$

Examples

- Noninterference:

$$\forall\pi\forall\pi'.\ \mathbf{G}((i_{\mathsf{pblc}})_\pi \leftrightarrow (i_{\mathsf{pblc}})_{\pi'}) \rightarrow \mathbf{G}((o_{\mathsf{pblc}})_\pi \leftrightarrow (o_{\mathsf{pblc}})_{\pi'})$$

- Noninterference for nondeterministic systems:

$$\begin{aligned}\forall\pi\forall\pi'\exists\pi''.\ &\mathbf{G}((i_{\mathsf{pblc}})_\pi \leftrightarrow (i_{\mathsf{pblc}})_{\pi''})\wedge\\ &\mathbf{G}((o_{\mathsf{pblc}})_\pi \leftrightarrow (o_{\mathsf{pblc}})_{\pi''})\wedge\\ &\mathbf{G}((i_{\mathsf{scrt}})_{\pi'} \leftrightarrow (i_{\mathsf{scrt}})_{\pi''})\end{aligned}$$

Examples

- Noninterference:

$$\forall\pi\forall\pi'.\ \mathbf{G}((i_{\mathsf{pblc}})_\pi \leftrightarrow (i_{\mathsf{pblc}})_{\pi'}) \rightarrow \mathbf{G}((o_{\mathsf{pblc}})_\pi \leftrightarrow (o_{\mathsf{pblc}})_{\pi'})$$

- Noninterference for nondeterministic systems:

$$\begin{aligned}\forall\pi\forall\pi'\exists\pi''.\ & \mathbf{G}((i_{\mathsf{pblc}})_\pi \leftrightarrow (i_{\mathsf{pblc}})_{\pi''})\wedge\\ & \mathbf{G}((o_{\mathsf{pblc}})_\pi \leftrightarrow (o_{\mathsf{pblc}})_{\pi''})\wedge\\ & \mathbf{G}((i_{\mathsf{scrt}})_{\pi'} \leftrightarrow (i_{\mathsf{scrt}})_{\pi''})\end{aligned}$$

- $\mathcal{S}$ 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 the automotive industry
- Network verification
  - Latency and congestion of computer networks

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

Model-Checking

The HyperLTL model-checking problem:

Given a finite transition system $\mathcal{S}$ and φ, does $\mathrm{Traces}(\mathcal{S}) \models \varphi$?

Model-Checking

The HyperLTL model-checking problem:

Given a finite transition system $\mathcal{S}$ and φ, does $\mathrm{Traces}(\mathcal{S}) \models \varphi$?

Recall: The LTL model-checking problem is PSPACE-complete.

Model-Checking

The HyperLTL model-checking problem:

Given a finite transition system $\mathcal{S}$ and φ, does $\mathrm{Traces}(\mathcal{S}) \models \varphi$?

Recall: The LTL model-checking problem is PSPACE-complete.

Theorem (Clarkson et al. '14, Rabe '16, Mascle & Z. '20)
The HyperLTL model-checking problem is TOWER-complete, 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 $\varphi = \exists\pi_1. \forall\pi_2. \dots \exists\pi_{k-1}. \forall\pi_k. \psi$.
- Rewrite as $\exists\pi_1. \neg\exists\pi_2. \neg \dots \exists\pi_{k-1}. \neg\exists\pi_k. \neg\psi$.
- We construct, by induction over the quantifier prefix, non-determinstic Büchi automata accepting exactly the variable assignments satisfying the subformulas of φ.
- Then, we obtain an automaton $\mathcal{A}$ with $L(\mathcal{A}) \neq \emptyset$ iff $\mathrm{Traces}(\mathcal{S}) \models \varphi$.

Model-Checking

Proof:

- Consider $\varphi = \exists \pi_1. \forall \pi_2. \dots \exists \pi_{k-1}. \forall \pi_k. \psi$.
- Rewrite as $\exists \pi_1. \neg \exists \pi_2. \neg \dots \exists \pi_{k-1}. \neg \exists \pi_k. \neg \psi$.
- We construct, by induction over the quantifier prefix, non-determinstic Büchi automata accepting exactly the variable assignments satisfying the subformulas of φ.
- Then, we obtain an automaton $\mathcal{A}$ with $L(\mathcal{A}) \neq \emptyset$ iff $\mathrm{Traces}(\mathcal{S}) \models \varphi$.
  - Induction start: build automaton for the LTL formula obtained from $\neg\psi$ by replacing a_{π_j} by a_j.

Model-Checking

Proof:

- Consider $\varphi = \exists\pi_1. \forall\pi_2. \dots \exists\pi_{k-1}. \forall\pi_k. \psi$.
- Rewrite as $\exists\pi_1. \neg\exists\pi_2. \neg \dots \exists\pi_{k-1}. \neg\exists\pi_k. \neg\psi$.
- We construct, by induction over the quantifier prefix, non-determinstic Büchi automata accepting exactly the variable assignments satisfying the subformulas of φ.
- Then, we obtain an automaton $\mathcal{A}$ with $L(\mathcal{A}) \neq \emptyset$ iff $\mathrm{Traces}(\mathcal{S}) \models \varphi$.
  - Induction start: build automaton for the 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 $\mathcal{S}$ (involves product with $\mathcal{S}$).

Model-Checking

Proof:

- Consider $\varphi = \exists\pi_1. \forall\pi_2. \ldots \exists\pi_{k-1}. \forall\pi_k. \psi$.
- Rewrite as $\exists\pi_1. \neg\exists\pi_2. \neg \ldots \exists\pi_{k-1}. \neg\exists\pi_k. \neg\psi$.
- We construct, by induction over the quantifier prefix, non-determinstic Büchi automata accepting exactly the variable assignments satisfying the subformulas of φ.
- Then, we obtain an automaton $\mathcal{A}$ with $L(\mathcal{A}) \neq \emptyset$ iff $\mathrm{Traces}(\mathcal{S}) \models \varphi$.
  - Induction start: build automaton for the 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 $\mathcal{S}$ (involves product with $\mathcal{S}$).
  - For $\neg\theta$ complement automaton for θ.

Skolem Functions

$\mathcal{S}$ satisfies a formula of the form $\forall\pi.\ \exists\pi'.\ \psi$ iff there is a (Skolem) function $f\colon \mathrm{Traces}(\mathcal{S}) \to \mathrm{Traces}(\mathcal{S})$ such that the assignment

$$[\pi \mapsto t, \pi' \mapsto f(t)]$$

satisfies ψ for all $t \in \mathrm{Tr}(\mathcal{S})$.

Skolem Functions

$\mathcal{S}$ satisfies a formula of the form $\forall\pi.\ \exists\pi'.\ \psi$ iff there is a (Skolem) function $f\colon \mathrm{Traces}(\mathcal{S}) \to \mathrm{Traces}(\mathcal{S})$ such that the assignment

$$[\pi \mapsto t, \pi' \mapsto f(t)]$$

satisfies ψ for all $t \in \mathrm{Tr}(\mathcal{S})$.

- Thus, f "explains" why $\mathcal{S} \models \forall\pi.\ \exists\pi'.\ \psi$.

Skolem Functions

$\mathcal{S}$ satisfies a formula of the form $\forall \pi.\ \exists \pi'.\ \psi$ iff there is a (Skolem) function $f\colon \mathrm{Traces}(\mathcal{S}) \to \mathrm{Traces}(\mathcal{S})$ such that the assignment

$$[\pi \mapsto t, \pi' \mapsto f(t)]$$

satisfies ψ for all $t \in \mathrm{Tr}(\mathcal{S})$.

- Thus, f "explains" why $\mathcal{S} \models \forall \pi.\ \exists \pi'.\ \psi$.
- In general, if $\mathcal{S} \models \varphi$, then Skolem functions for the existentially quantified variables in φ explain why $\mathcal{S} \models \varphi$.

Skolem Functions

$\mathcal{S}$ satisfies a formula of the form $\forall\pi.\ \exists\pi'.\ \psi$ iff there is a (Skolem) function $f\colon \mathrm{Traces}(\mathcal{S}) \to \mathrm{Traces}(\mathcal{S})$ such that the assignment

$$[\pi \mapsto t, \pi' \mapsto f(t)]$$

satisfies ψ for all $t \in \mathrm{Tr}(\mathcal{S})$.

- Thus, f "explains" why $\mathcal{S} \models \forall\pi.\ \exists\pi'.\ \psi$.
- In general, if $\mathcal{S} \models \varphi$, then Skolem functions for the existentially quantified variables in φ explain why $\mathcal{S} \models \varphi$.
- Dually, if $\mathcal{S} \not\models \varphi$, then $\mathcal{S} \models \neg\varphi$ and Skolem functions for the existentially quantified variables in $\neg\psi$ are a "counterexample" for $\mathcal{S} \not\models \varphi$.

Computable Skolem Functions

To interpret and algorithmically handle Skolem functions, we represent them by finite automata with output (transducers).

Computable Skolem Functions

To interpret and algorithmically handle Skolem functions, we represent them by finite automata with output (transducers).

Example

Consider $\mathcal{S}$ with $\mathrm{Traces}(\mathcal{S}) = (2^{\{a\}})^\omega$, which satisfies

$$\varphi = \forall\pi.\ \exists\pi'.\ (\mathbf{X}\, a_\pi) \leftrightarrow a_{\pi'}.$$

Computable Skolem Functions

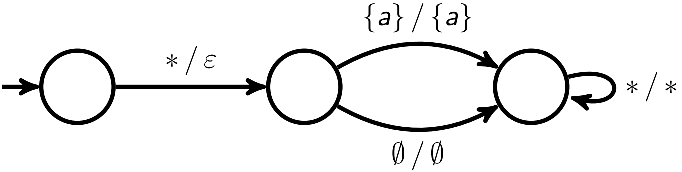
To interpret and algorithmically handle Skolem functions, we represent them by finite automata with output (transducers).

Example

Consider $\mathcal{S}$ with $\mathrm{Traces}(\mathcal{S}) = (2^{\{a\}})^\omega$, which satisfies

$$\varphi = \forall\pi.\ \exists\pi'.\ (\mathbf{X}\, a_\pi) \leftrightarrow a_{\pi'}.$$

The following transducer represents a Skolem function for π':



Another Example

Consider the formula

$$\forall\pi.\ \exists\pi'.\ (\mathbf{F}\, a_\pi) \leftrightarrow a_{\pi'}.$$

Another Example

Consider the formula

$$\forall \pi.\ \exists \pi'.\ (\mathbf{F}\, a_\pi) \leftrightarrow a_{\pi'}.$$

- $\mathcal{S}$ with $\mathrm{Traces}(\mathcal{S}) = (2^{\{a\}})^\omega$ satisfies it, witnesssed e.g., by the Skolem function

$$f(t) = \begin{cases} \{a\}\emptyset^\omega & \text{if } t \text{ contains an } a \text{ somewhere,} \\ \emptyset^\omega & \text{if } t \text{ does not contain an } a \text{ anywhere.} \end{cases}$$

Another Example

Consider the formula

$$\forall \pi.\ \exists \pi'.\ (\mathbf{F}\, a_\pi) \leftrightarrow a_{\pi'}.$$

- $\mathcal{S}$ with $\mathrm{Traces}(\mathcal{S}) = (2^{\{a\}})^\omega$ satisfies it, witnesssed e.g., by the Skolem function

$$f(t) = \begin{cases} \{a\}\emptyset^\omega & \text{if } t \text{ contains an } a \text{ somewhere,} \\ \emptyset^\omega & \text{if } t \text{ does not contain an } a \text{ anywhere.} \end{cases}$$

- However, this, and any other Skolem function, is not representable by a transducer.

Our Goal

Given $\mathcal{S}$ and φ such that $\mathcal{S} \models \varphi$, is $\mathcal{S} \models \varphi$ witnessed by Skolem functions representable by finite transducers?

Our Goal

Given $\mathcal{S}$ and φ such that $\mathcal{S} \models \varphi$, is $\mathcal{S} \models \varphi$ witnessed by Skolem functions representable by finite transducers?

- We characterize their existence by a game.

Our Goal

Given $\mathcal{S}$ and φ such that $\mathcal{S} \models \varphi$, is $\mathcal{S} \models \varphi$ witnessed by Skolem functions representable by finite transducers?

- We characterize their existence by a game.
- (Incomplete) intuition for $\forall \pi_0. \exists \pi_1. \forall \pi_2. \exists \pi_3. \forall \pi_4. \exists \pi_5. \psi$:

Our Goal

Given $\mathcal{S}$ and φ such that $\mathcal{S} \models \varphi$, is $\mathcal{S} \models \varphi$ witnessed by Skolem functions representable by finite transducers?

- We characterize their existence by a game.
- (Incomplete) intuition for $\forall\pi_0.\ \exists\pi_1.\ \forall\pi_2.\ \exists\pi_3.\ \forall\pi_4.\ \exists\pi_5.\ \psi$:

π_0 t_0^0

π_1

π_2

π_3

π_4

π_5

Our Goal

Given $\mathcal{S}$ and φ such that $\mathcal{S} \models \varphi$, is $\mathcal{S} \models \varphi$ witnessed by Skolem functions representable by finite transducers?

- We characterize their existence by a game.
- (Incomplete) intuition for $\forall\pi_0.\ \exists\pi_1.\ \forall\pi_2.\ \exists\pi_3.\ \forall\pi_4.\ \exists\pi_5.\ \psi$:

π_0 t_0^0

π_1 t_1^0

π_2

π_3

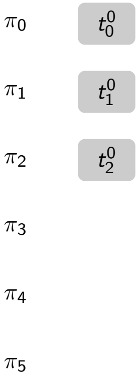
π_4

π_5

Our Goal

Given $\mathcal{S}$ and φ such that $\mathcal{S} \models \varphi$, is $\mathcal{S} \models \varphi$ witnessed by Skolem functions representable by finite transducers?

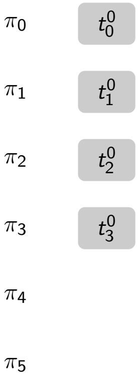
- We characterize their existence by a game.
- (Incomplete) intuition for $\forall \pi_0.\ \exists \pi_1.\ \forall \pi_2.\ \exists \pi_3.\ \forall \pi_4.\ \exists \pi_5.\ \psi$:



Our Goal

Given $\mathcal{S}$ and φ such that $\mathcal{S} \models \varphi$, is $\mathcal{S} \models \varphi$ witnessed by Skolem functions representable by finite transducers?

- We characterize their existence by a game.
- (Incomplete) intuition for $\forall\pi_0.\ \exists\pi_1.\ \forall\pi_2.\ \exists\pi_3.\ \forall\pi_4.\ \exists\pi_5.\ \psi$:



Our Goal

Given $\mathcal{S}$ and φ such that $\mathcal{S} \models \varphi$, is $\mathcal{S} \models \varphi$ witnessed by Skolem functions representable by finite transducers?

- We characterize their existence by a game.
- (Incomplete) intuition for $\forall\pi_0.\ \exists\pi_1.\ \forall\pi_2.\ \exists\pi_3.\ \forall\pi_4.\ \exists\pi_5.\ \psi$:

π_0	t_0^0
π_1	t_1^0
π_2	t_2^0
π_3	t_3^0
π_4	t_4^0
π_5	

Our Goal

Given $\mathcal{S}$ and φ such that $\mathcal{S} \models \varphi$, is $\mathcal{S} \models \varphi$ witnessed by Skolem functions representable by finite transducers?

- We characterize their existence by a game.
- (Incomplete) intuition for $\forall \pi_0. \exists \pi_1. \forall \pi_2. \exists \pi_3. \forall \pi_4. \exists \pi_5. \psi$:

π_0	t_0^0
π_1	t_1^0
π_2	t_2^0
π_3	t_3^0
π_4	t_4^0
π_5	t_5^0

Our Goal

Given $\mathcal{S}$ and φ such that $\mathcal{S} \models \varphi$, is $\mathcal{S} \models \varphi$ witnessed by Skolem functions representable by finite transducers?

- We characterize their existence by a game.
- (Incomplete) intuition for $\forall\pi_0.\ \exists\pi_1.\ \forall\pi_2.\ \exists\pi_3.\ \forall\pi_4.\ \exists\pi_5.\ \psi$:

π_0	t_0^0	t_0^1
π_1	t_1^0	
π_2	t_2^0	
π_3	t_3^0	
π_4	t_4^0	
π_5	t_5^0	

Our Goal

Given $\mathcal{S}$ and φ such that $\mathcal{S} \models \varphi$, is $\mathcal{S} \models \varphi$ witnessed by Skolem functions representable by finite transducers?

- We characterize their existence by a game.
- (Incomplete) intuition for $\forall \pi_0.\ \exists \pi_1.\ \forall \pi_2.\ \exists \pi_3.\ \forall \pi_4.\ \exists \pi_5.\ \psi$:

π_0	t_0^0	t_0^1
π_1	t_1^0	t_1^1
π_2	t_2^0	
π_3	t_3^0	
π_4	t_4^0	
π_5	t_5^0	

Our Goal

Given $\mathcal{S}$ and φ such that $\mathcal{S} \models \varphi$, is $\mathcal{S} \models \varphi$ witnessed by Skolem functions representable by finite transducers?

- We characterize their existence by a game.
- (Incomplete) intuition for $\forall\pi_0.\ \exists\pi_1.\ \forall\pi_2.\ \exists\pi_3.\ \forall\pi_4.\ \exists\pi_5.\ \psi$:

π_0	t_0^0	t_0^1
π_1	t_1^0	t_1^1
π_2	t_2^0	t_2^1
π_3	t_3^0	
π_4	t_4^0	
π_5	t_5^0	

Our Goal

Given $\mathcal{S}$ and φ such that $\mathcal{S} \models \varphi$, is $\mathcal{S} \models \varphi$ witnessed by Skolem functions representable by finite transducers?

- We characterize their existence by a game.
- (Incomplete) intuition for $\forall\pi_0.\ \exists\pi_1.\ \forall\pi_2.\ \exists\pi_3.\ \forall\pi_4.\ \exists\pi_5.\ \psi$:

π_0	t_0^0	t_0^1
π_1	t_1^0	t_1^1
π_2	t_2^0	t_2^1
π_3	t_3^0	t_3^1
π_4	t_4^0	
π_5	t_5^0	

Our Goal

Given $\mathcal{S}$ and φ such that $\mathcal{S} \models \varphi$, is $\mathcal{S} \models \varphi$ witnessed by Skolem functions representable by finite transducers?

- We characterize their existence by a game.
- (Incomplete) intuition for $\forall \pi_0.\ \exists \pi_1.\ \forall \pi_2.\ \exists \pi_3.\ \forall \pi_4.\ \exists \pi_5.\ \psi$:

π_0	t_0^0	t_0^1
π_1	t_1^0	t_1^1
π_2	t_2^0	t_2^1
π_3	t_3^0	t_3^1
π_4	t_4^0	t_4^1
π_5	t_5^0	

Our Goal

Given $\mathcal{S}$ and φ such that $\mathcal{S} \models \varphi$, is $\mathcal{S} \models \varphi$ witnessed by Skolem functions representable by finite transducers?

- We characterize their existence by a game.
- (Incomplete) intuition for $\forall \pi_0.\ \exists \pi_1.\ \forall \pi_2.\ \exists \pi_3.\ \forall \pi_4.\ \exists \pi_5.\ \psi$:

π_0	t_0^0	t_0^1
π_1	t_1^0	t_1^1
π_2	t_2^0	t_2^1
π_3	t_3^0	t_3^1
π_4	t_4^0	t_4^1
π_5	t_5^0	t_5^1

Our Goal

Given $\mathcal{S}$ and φ such that $\mathcal{S} \models \varphi$, is $\mathcal{S} \models \varphi$ witnessed by Skolem functions representable by finite transducers?

- We characterize their existence by a game.
- (Incomplete) intuition for $\forall\pi_0.\ \exists\pi_1.\ \forall\pi_2.\ \exists\pi_3.\ \forall\pi_4.\ \exists\pi_5.\ \psi$:

π_0	t_0^0	t_0^1	t_0^2
π_1	t_1^0	t_1^1	
π_2	t_2^0	t_2^1	
π_3	t_3^0	t_3^1	
π_4	t_4^0	t_4^1	
π_5	t_5^0	t_5^1	

Our Goal

Given $\mathcal{S}$ and φ such that $\mathcal{S} \models \varphi$, is $\mathcal{S} \models \varphi$ witnessed by Skolem functions representable by finite transducers?

- We characterize their existence by a game.
- (Incomplete) intuition for $\forall \pi_0.\ \exists \pi_1.\ \forall \pi_2.\ \exists \pi_3.\ \forall \pi_4.\ \exists \pi_5.\ \psi$:

π_0	t_0^0	t_0^1	t_0^2
π_1	t_1^0	t_1^1	t_1^2
π_2	t_2^0	t_2^1	
π_3	t_3^0	t_3^1	
π_4	t_4^0	t_4^1	
π_5	t_5^0	t_5^1	

Our Goal

Given $\mathcal{S}$ and φ such that $\mathcal{S} \models \varphi$, is $\mathcal{S} \models \varphi$ witnessed by Skolem functions representable by finite transducers?

- We characterize their existence by a game.
- (Incomplete) intuition for $\forall\pi_0.\ \exists\pi_1.\ \forall\pi_2.\ \exists\pi_3.\ \forall\pi_4.\ \exists\pi_5.\ \psi$:

π_0	t_0^0	t_0^1	t_0^2
π_1	t_1^0	t_1^1	t_1^2
π_2	t_2^0	t_2^1	t_2^2
π_3	t_3^0	t_3^1	
π_4	t_4^0	t_4^1	
π_5	t_5^0	t_5^1	

Our Goal

Given $\mathcal{S}$ and φ such that $\mathcal{S} \models \varphi$, is $\mathcal{S} \models \varphi$ witnessed by Skolem functions representable by finite transducers?

- We characterize their existence by a game.
- (Incomplete) intuition for $\forall \pi_0.\ \exists \pi_1.\ \forall \pi_2.\ \exists \pi_3.\ \forall \pi_4.\ \exists \pi_5.\ \psi$:

π_0	t_0^0	t_0^1	t_0^2
π_1	t_1^0	t_1^1	t_1^2
π_2	t_2^0	t_2^1	t_2^2
π_3	t_3^0	t_3^1	t_3^2
π_4	t_4^0	t_4^1	
π_5	t_5^0	t_5^1	

Our Goal

Given $\mathcal{S}$ and φ such that $\mathcal{S} \models \varphi$, is $\mathcal{S} \models \varphi$ witnessed by Skolem functions representable by finite transducers?

- We characterize their existence by a game.
- (Incomplete) intuition for $\forall\pi_0.\ \exists\pi_1.\ \forall\pi_2.\ \exists\pi_3.\ \forall\pi_4.\ \exists\pi_5.\ \psi$:

π_0	t_0^0	t_0^1	t_0^2
π_1	t_1^0	t_1^1	t_1^2
π_2	t_2^0	t_2^1	t_2^2
π_3	t_3^0	t_3^1	t_3^2
π_4	t_4^0	t_4^1	t_4^2
π_5	t_5^0	t_5^1	

Our Goal

Given $\mathcal{S}$ and φ such that $\mathcal{S} \models \varphi$, is $\mathcal{S} \models \varphi$ witnessed by Skolem functions representable by finite transducers?

- We characterize their existence by a game.
- (Incomplete) intuition for $\forall \pi_0.\ \exists \pi_1.\ \forall \pi_2.\ \exists \pi_3.\ \forall \pi_4.\ \exists \pi_5.\ \psi$:

π_0	t_0^0	t_0^1	t_0^2
π_1	t_1^0	t_1^1	t_1^2
π_2	t_2^0	t_2^1	t_2^2
π_3	t_3^0	t_3^1	t_3^2
π_4	t_4^0	t_4^1	t_4^2
π_5	t_5^0	t_5^1	t_5^2

Our Goal

Given $\mathcal{S}$ and φ such that $\mathcal{S} \models \varphi$, is $\mathcal{S} \models \varphi$ witnessed by Skolem functions representable by finite transducers?

- We characterize their existence by a game.
- (Incomplete) intuition for $\forall\pi_0.\ \exists\pi_1.\ \forall\pi_2.\ \exists\pi_3.\ \forall\pi_4.\ \exists\pi_5.\ \psi$:

π_0	t_0^0	t_0^1	t_0^2	$\cdots$
π_1	t_1^0	t_1^1	t_1^2	$\cdots$
π_2	t_2^0	t_2^1	t_2^2	$\cdots$
π_3	t_3^0	t_3^1	t_3^2	$\cdots$
π_4	t_4^0	t_4^1	t_4^2	$\cdots$
π_5	t_5^0	t_5^1	t_5^2	$\cdots$

Problem 1: Information

$$\forall\pi_0.\ \exists\pi_1.\ \forall\pi_2.\ \exists\pi_3.\ \forall\pi_4.\ \exists\pi_5.\ \psi$$

- The Skolem function for π_1 may only depend on the trace assigned to π_0, but not those assigned to π_2 and π_4.

Problem 1: Information

$$\forall \pi_0.\ \exists \pi_1.\ \forall \pi_2.\ \exists \pi_3.\ \forall \pi_4.\ \exists \pi_5.\ \psi$$

- The Skolem function for π_1 may only depend on the trace assigned to π_0, but not those assigned to π_2 and π_4.
- Thus, our game needs to be one of imperfect information:
  - A coalition of players, one for each existentially quantified variable against
  - a (single) player for the universally quantified variables.
  - Player i for odd i has only access to the choices for $\pi_0, \pi_1, \dots, \pi_{i-1}$.

Problem 1: Information

$$\forall \pi_0.\ \exists \pi_1.\ \forall \pi_2.\ \exists \pi_3.\ \forall \pi_4.\ \exists \pi_5.\ \psi$$

- The Skolem function for π_1 may only depend on the trace assigned to π_0, but not those assigned to π_2 and π_4.
- Thus, our game needs to be one of imperfect information:
  - A coalition of players, one for each existentially quantified variable against
  - a (single) player for the universally quantified variables.
  - Player *i* for odd *i* has only access to the choices for $\pi_0, \pi_1, \ldots, \pi_{i-1}$.
- The information is hierarchical $\Rightarrow$ solving games with ω-regular winning conditions is decidable.

Problem 2: Order of Moves

$$\varphi = \forall \pi.\ \exists \pi'.\ (\mathbf{X}\, a_\pi) \leftrightarrow a_{\pi'}$$

- To pick the first letter of the trace for π', the player needs to know the second letter of the trace for π.

Problem 2: Order of Moves

$$\varphi = \forall \pi.\ \exists \pi'.\ (\mathbf{X}\, a_\pi) \leftrightarrow a_{\pi'}$$

- To pick the first letter of the trace for π', the player needs to know the second letter of the trace for π.
- Thus, our game needs to allow to "delay" moves.

Problem 2: Order of Moves

$$\varphi = \forall \pi.\ \exists \pi'.\ (\mathbf{X}\, a_\pi) \leftrightarrow a_{\pi'}$$

- To pick the first letter of the trace for π', the player needs to know the second letter of the trace for π.
- Thus, our game needs to allow to "delay" moves.

π_0

π_1

π_2

π_3

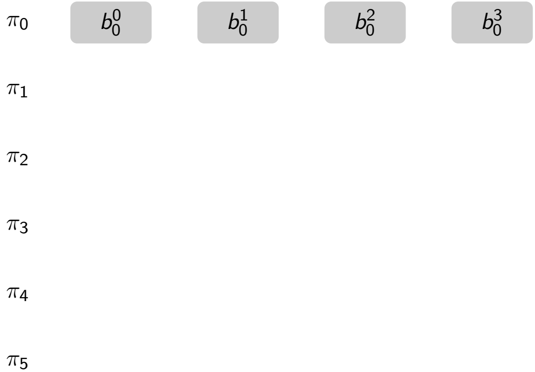
π_4

π_5

Problem 2: Order of Moves

$$\varphi = \forall \pi.\ \exists \pi'.\ (\mathbf{X}\, a_\pi) \leftrightarrow a_{\pi'}$$

- To pick the first letter of the trace for π', the player needs to know the second letter of the trace for π.
- Thus, our game needs to allow to "delay" moves.



Problem 2: Order of Moves

$$\varphi = \forall \pi.\ \exists \pi'.\ (\mathbf{X}\, a_\pi) \leftrightarrow a_{\pi'}$$

- To pick the first letter of the trace for π', the player needs to know the second letter of the trace for π.
- Thus, our game needs to allow to "delay" moves.

π_0	b_0^0	b_0^1	b_0^2	b_0^3
π_1	b_1^0	b_1^1	b_1^2	
π_2				
π_3				
π_4				
π_5				

Problem 2: Order of Moves

$$\varphi = \forall \pi.\ \exists \pi'.\ (\mathbf{X}\, a_\pi) \leftrightarrow a_{\pi'}$$

- To pick the first letter of the trace for π', the player needs to know the second letter of the trace for π.
- Thus, our game needs to allow to “delay” moves.

π_0	b_0^0	b_0^1	b_0^2	b_0^3
π_1	b_1^0	b_1^1	b_1^2	
π_2	b_2^0	b_2^1	b_2^2	
π_3				
π_4				
π_5				

Problem 2: Order of Moves

$$\varphi = \forall \pi.\ \exists \pi'.\ (\mathbf{X}\, a_\pi) \leftrightarrow a_{\pi'}$$

- To pick the first letter of the trace for π', the player needs to know the second letter of the trace for π.
- Thus, our game needs to allow to “delay” moves.

π_0	b_0^0	b_0^1	b_0^2	b_0^3
π_1	b_1^0	b_1^1	b_1^2	
π_2	b_2^0	b_2^1	b_2^2	
π_3	b_3^0	b_3^1		
π_4				
π_5				

Problem 2: Order of Moves

$$\varphi = \forall\pi.\ \exists\pi'.\ (\mathbf{X}\, a_\pi) \leftrightarrow a_{\pi'}$$

- To pick the first letter of the trace for π', the player needs to know the second letter of the trace for π.
- Thus, our game needs to allow to "delay" moves.

π_0	b_0^0	b_0^1	b_0^2	b_0^3
π_1	b_1^0	b_1^1	b_1^2	
π_2	b_2^0	b_2^1	b_2^2	
π_3	b_3^0	b_3^1		
π_4	b_4^0	b_4^1		
π_5				

Problem 2: Order of Moves

$$\varphi = \forall\pi.\ \exists\pi'.\ (\mathbf{X}\, a_\pi) \leftrightarrow a_{\pi'}$$

- To pick the first letter of the trace for π', the player needs to know the second letter of the trace for π.
- Thus, our game needs to allow to "delay" moves.

π_0	b_0^0	b_0^1	b_0^2	b_0^3
π_1	b_1^0	b_1^1	b_1^2	
π_2	b_2^0	b_2^1	b_2^2	
π_3	b_3^0	b_3^1		
π_4	b_4^0	b_4^1		
π_5	b_5^0			

Problem 2: Order of Moves

$$\varphi = \forall \pi.\ \exists \pi'.\ (\mathbf{X}\, a_\pi) \leftrightarrow a_{\pi'}$$

- To pick the first letter of the trace for π', the player needs to know the second letter of the trace for π.
- Thus, our game needs to allow to "delay" moves.

π_0	b_0^0	b_0^1	b_0^2	b_0^3	b_0^4
π_1	b_1^0	b_1^1	b_1^2		
π_2	b_2^0	b_2^1	b_2^2		
π_3	b_3^0	b_3^1			
π_4	b_4^0	b_4^1			
π_5	b_5^0				

Problem 2: Order of Moves

$$\varphi = \forall\pi.\ \exists\pi'.\ (\mathbf{X}\, a_\pi) \leftrightarrow a_{\pi'}$$

- To pick the first letter of the trace for π', the player needs to know the second letter of the trace for π.
- Thus, our game needs to allow to "delay" moves.

π_0	b_0^0	b_0^1	b_0^2	b_0^3	b_0^4
π_1	b_1^0	b_1^1	b_1^2	b_1^3	
π_2	b_2^0	b_2^1	b_2^2		
π_3	b_3^0	b_3^1			
π_4	b_4^0	b_4^1			
π_5	b_5^0				

Problem 2: Order of Moves

$$\varphi = \forall\pi.\ \exists\pi'.\ (\mathbf{X}\, a_\pi) \leftrightarrow a_{\pi'}$$

- To pick the first letter of the trace for π', the player needs to know the second letter of the trace for π.
- Thus, our game needs to allow to "delay" moves.

π_0	b_0^0	b_0^1	b_0^2	b_0^3	b_0^4
π_1	b_1^0	b_1^1	b_1^2	b_1^3	
π_2	b_2^0	b_2^1	b_2^2	b_2^3	
π_3	b_3^0	b_3^1			
π_4	b_4^0	b_4^1			
π_5	b_5^0				

Problem 2: Order of Moves

$$\varphi = \forall \pi.\ \exists \pi'.\ (\mathbf{X}\, a_\pi) \leftrightarrow a_{\pi'}$$

- To pick the first letter of the trace for π', the player needs to know the second letter of the trace for π.
- Thus, our game needs to allow to "delay" moves.

π_0	b_0^0	b_0^1	b_0^2	b_0^3	b_0^4
π_1	b_1^0	b_1^1	b_1^2	b_1^3	
π_2	b_2^0	b_2^1	b_2^2	b_2^3	
π_3	b_3^0	b_3^1	b_3^2		
π_4	b_4^0	b_4^1			
π_5	b_5^0				

Problem 2: Order of Moves

$$\varphi = \forall \pi.\ \exists \pi'.\ (\mathbf{X}\, a_\pi) \leftrightarrow a_{\pi'}$$

- To pick the first letter of the trace for π', the player needs to know the second letter of the trace for π.
- Thus, our game needs to allow to “delay” moves.

π_0	b_0^0	b_0^1	b_0^2	b_0^3	b_0^4
π_1	b_1^0	b_1^1	b_1^2	b_1^3	
π_2	b_2^0	b_2^1	b_2^2	b_2^3	
π_3	b_3^0	b_3^1	b_3^2		
π_4	b_4^0	b_4^1	b_4^2		
π_5	b_5^0				

Problem 2: Order of Moves

$$\varphi = \forall \pi.\ \exists \pi'.\ (\mathbf{X}\, a_\pi) \leftrightarrow a_{\pi'}$$

- To pick the first letter of the trace for π', the player needs to know the second letter of the trace for π.
- Thus, our game needs to allow to “delay” moves.

π_0	b_0^0	b_0^1	b_0^2	b_0^3	b_0^4
π_1	b_1^0	b_1^1	b_1^2	b_1^3	
π_2	b_2^0	b_2^1	b_2^2	b_2^3	
π_3	b_3^0	b_3^1	b_3^2		
π_4	b_4^0	b_4^1	b_4^2		
π_5	b_5^0	b_5^1			

Problem 2: Order of Moves

$$\varphi = \forall \pi.\ \exists \pi'.\ (\mathbf{X}\, a_\pi) \leftrightarrow a_{\pi'}$$

- To pick the first letter of the trace for π', the player needs to know the second letter of the trace for π.
- Thus, our game needs to allow to “delay” moves.

π_0	b_0^0	b_0^1	b_0^2	b_0^3	b_0^4	$\cdots$
π_1	b_1^0	b_1^1	b_1^2	b_1^3	$\cdots$	
π_2	b_2^0	b_2^1	b_2^2	b_2^3	$\cdots$	
π_3	b_3^0	b_3^1	b_3^2	$\cdots$		
π_4	b_4^0	b_4^1	b_4^2	$\cdots$		
π_5	b_5^0	b_5^1	$\cdots$			

Results

Theorem (Winter & Z. '24)

There is a block size (effectively computable from $\mathcal{S}$ and φ) such that the following are equivalent:

1. *The coalition of players for the existentially quantified variables in φ has a collection of winning strategies.*
2. *$\mathcal{S} \models \varphi$ is witnessed by Skolem functions implemented by transducers.*

Furthermore, the game is effectively solvable and the transducers can be effectively computed.

Prophecies

- So, we can determine the existence of computable Skolem functions.
- But $\forall\pi.\ \exists\pi'.\ (\mathbf{F}\, a_\pi) \leftrightarrow a_{\pi'}$ does not have computable Skolem functions.

Prophecies

- So, we can determine the existence of computable Skolem functions.
- But $\forall\pi.\ \exists\pi'.\ (\mathbf{F}\, a_\pi) \leftrightarrow a_{\pi'}$ does not have computable Skolem functions.
- Beutner and Finkbeiner have shown that model-checking of $\forall^*\exists^*$-formulas can be characterized by a two-player perfect information game using "prophecies".

Prophecies

- So, we can determine the existence of computable Skolem functions.
- But $\forall \pi.\ \exists \pi'.\ (\mathbf{F}\, a_\pi) \leftrightarrow a_{\pi'}$ does not have computable Skolem functions.
- Beutner and Finkbeiner have shown that model-checking of $\forall^*\exists^*$-formulas can be characterized by a two-player perfect information game using "prophecies".
- A prophecy is an ω-language and the player in charge of the universal variables has to specify in each round whether the traces he will pick are in the prophecy or not.
- If he cheats, he loses.

Prophecies

- So, we can determine the existence of computable Skolem functions.
- But $\forall\pi.\ \exists\pi'.\ (\mathbf{F}\, a_\pi) \leftrightarrow a_{\pi'}$ does not have computable Skolem functions.
- Beutner and Finkbeiner have shown that model-checking of $\forall^*\exists^*$-formulas can be characterized by a two-player perfect information game using "prophecies".
- A prophecy is an ω-language and the player in charge of the universal variables has to specify in each round whether the traces he will pick are in the prophecy or not.
- If he cheats, he loses.
- In the example above, the prophecy is the language of words containing an a somewhere.

What about arbitrary quantifier prefixes?

Theorem (Winter & Z. '25)

Given $\mathcal{S}$ and φ, there is an effectively computable and solvable imperfect information game such that the following are equivalent:

1. *The coalition of players for the existentially quantified variables in φ has a collection of winning strategies.*
2. $\mathcal{S} \models \varphi$.

Conclusion

- HyperLTL model-checking can be characterized by games of imperfect information, another manifestation of the tight connection between logic and games.
- Skolem functions yield explanations.

Conclusion

- HyperLTL model-checking can be characterized by games of imperfect information, another manifestation of the tight connection between logic and games.
- Skolem functions yield explanations.

- Key ingredients:
  - Construct traces on-the-fly and in alternation for decidability of the games.
  - Imperfect information.
  - An element of delay/lookahead.
  - Automata for the quantifier-free part of the formula.

Conclusion

- HyperLTL model-checking can be characterized by games of imperfect information, another manifestation of the tight connection between logic and games.
- Skolem functions yield explanations.

- Key ingredients:
  - Construct traces on-the-fly and in alternation for decidability of the games.
  - Imperfect information.
  - An element of delay/lookahead.
  - Automata for the quantifier-free part of the formula.

- Future work:
  - More expressive logics
  - Infinite-state systems
  - Complexity analysis