Hereditariness in the finite and prefix classes of first order logic

Abhisekh Sankaran
University of Cambridge

Joint work with Anuj Dawar

Verification Seminar
University of Oxford, UK

Nov 7, 2019

Introduction

- Hereditariness is a well realized property in computer science. E.g. cliques, bounded degree graphs, 3-colorable graphs, graphs of bounded clique-width, etc.
- The Łoś-Tarski theorem characterizes FO definable hereditary properties in terms of universal sentences.
- Historically significant: among the earliest applications of Gödel's Compactness theorem and opened the area of preservation theorems in model theory.
- Fails in the finite: there is a hereditary FO sentence that is not equivalent to any universal sentence over all finite structures (Gurevich-Shelah, 1984).
- But already in 1959, Tait gave a different counterexample, that turns out to be more powerful than known so far.

Main results

Let $\Sigma_n := \underbrace{\exists \bar{x}_1 \forall \bar{x}_2 \exists \bar{x}_3 \ldots}_{n \text{ blocks}} \alpha(\bar{x}_1, \ldots, \bar{x}_n)$ where α is quantifier-free.

Theorem

Tait's counterexample is an FO sentence that is hereditary over all finite structures, but is not equivalent over this class to any Σ_3 sentence. Further, the negation of the counterexample can be expressed in Datalog$(\neq, \neg)$.

Theorem

For every n, there is a vocabulary τ_n and an FO(τ_n) sentence φ_n that is hereditary over all finite structures, but that is not equivalent over this class to any Σ_n sentence. Further, $\neg\varphi_n$ can be expressed in Datalog$(\neq, \neg)$.

Main results

Let $\Sigma_n := \underbrace{\exists \bar{x}_1 \forall \bar{x}_2 \exists \bar{x}_3 \ldots}_{n \text{ blocks}} \alpha(\bar{x}_1, \ldots, \bar{x}_n)$ where α is quantifier-free.

Theorem

Tait's counterexample is an FO sentence that is hereditary over all finite structures, but is not equivalent over this class to any Σ_3 sentence. Further, the negation of the counterexample can be expressed in Datalog$(\neq, \neg)$.

Theorem

No prefix classes of FO is expressive enough to capture:

- FO-hereditariness in the finite
- FO $\cap$ Datalog$(\neq, \neg)$ queries in the finite

Analysing Tait's sentence

Analysing Tait's sentence: overview

- The sentence
- Showing hereditariness
- Construction of a suitable class of models and non-models
- Showing inexpressibility in Σ_2, and then in Σ_3
- Showing expressibility of negation of the sentence in Datalog($\neq, \neg$)

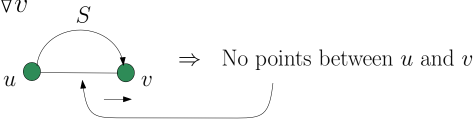
The sentence

Tait's sentence

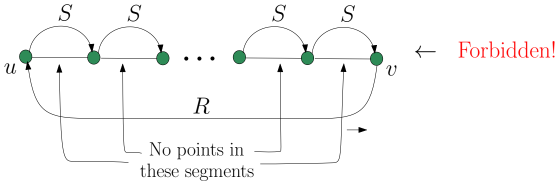
$$\Psi := \xi_1 \wedge \xi_2 \wedge \xi_3 \quad \in \mathrm{FO}(\sigma) \text{ where } \sigma = \{\leq, R, S\}$$

$\xi_1 :=$ " $\leq$ is a linear order"

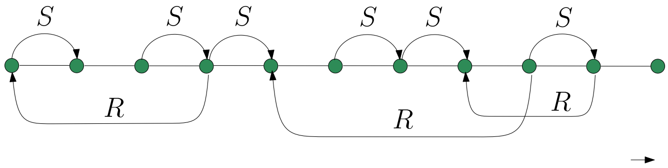
$\xi_2 := \forall u \forall v$



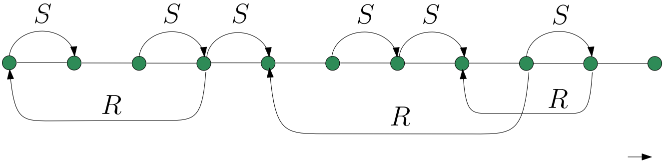
$\xi_3 := \forall u \forall v$



A model for Ψ

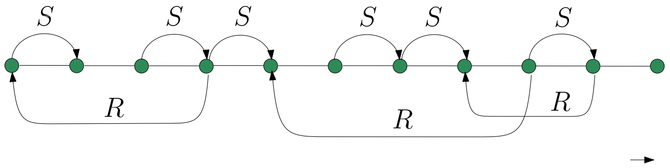


A model for Ψ

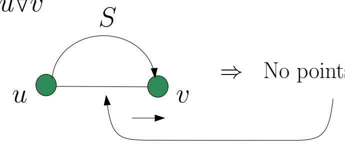


$\xi_1 :=$ " $\leq$ is a linear order" ✓

A model for Ψ

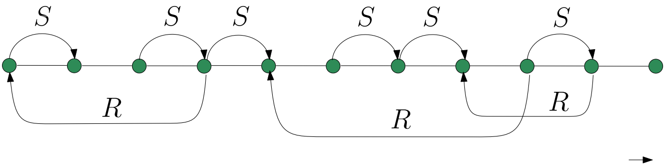


$\xi_2 := \forall u \forall v$

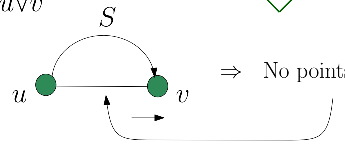


$\Rightarrow$ No points between u and v

A model for Ψ

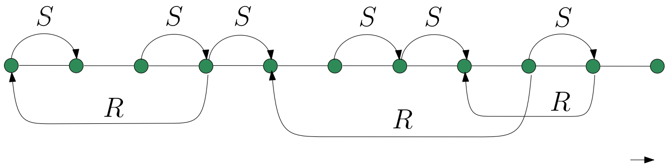


$\xi_2 := \forall u \forall v$

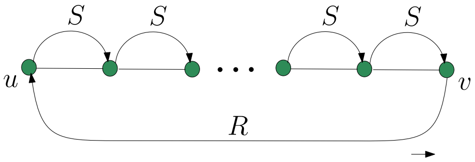


$\Rightarrow$ No points between u and v

A model for Ψ

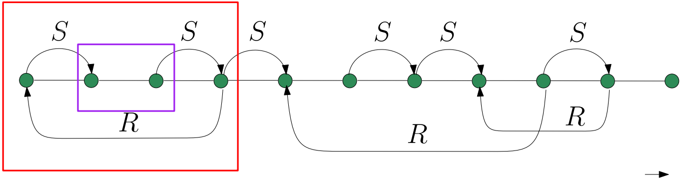


$\xi_3 := \forall u \forall v$

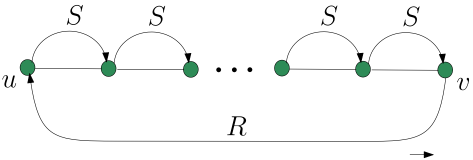


$\leftarrow$ Forbidden!

A model for Ψ

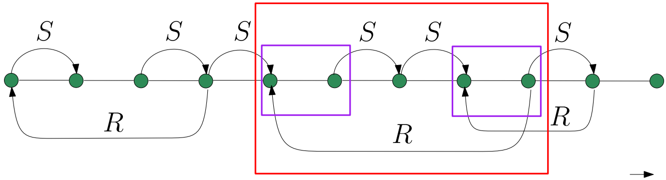


$\xi_3 := \forall u \forall v$

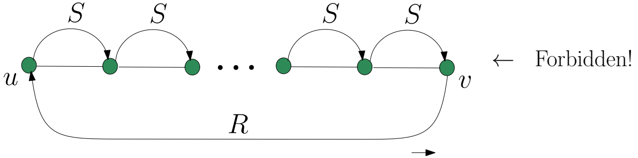


$\leftarrow$ Forbidden!

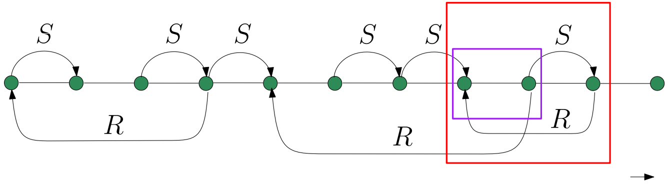
A model for Ψ



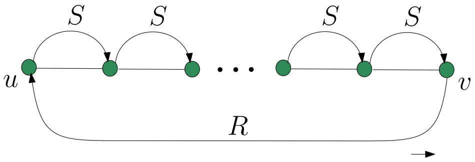
$\xi_3 := \forall u \forall v$



A model for Ψ

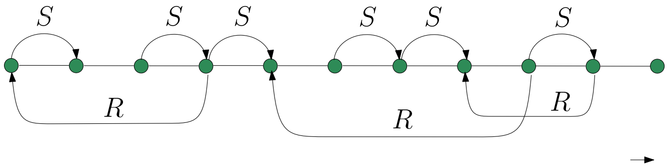


$\xi_3 := \forall u \forall v$

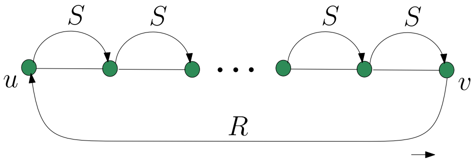


$\leftarrow$ Forbidden!

A model for Ψ



$\xi_3 := \forall u \forall v$



$\leftarrow$ Forbidden!

Hereditariness of Tait's sentence

Ψ is hereditary

- Suppose $\mathcal{A} \models \Psi$ where $\Psi := \xi_1 \wedge \xi_2 \wedge \xi_3$, and $\mathcal{B} \subseteq \mathcal{A}$.

Ψ is hereditary

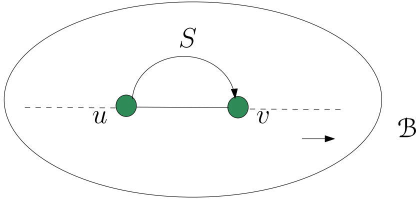
- Suppose $\mathcal{A} \models \Psi$ where $\Psi := \xi_1 \wedge \xi_2 \wedge \xi_3$, and $\mathcal{B} \subseteq \mathcal{A}$.
- Since ξ_1 is universal, it is hereditary, so $\mathcal{B} \models \xi_1$.

Ψ is hereditary

- Suppose $\mathcal{A} \models \Psi$ where $\Psi := \xi_1 \wedge \xi_2 \wedge \xi_3$, and $\mathcal{B} \subseteq \mathcal{A}$.
- Since ξ_1 is universal, it is hereditary, so $\mathcal{B} \models \xi_1$.
- Consider ξ_2.

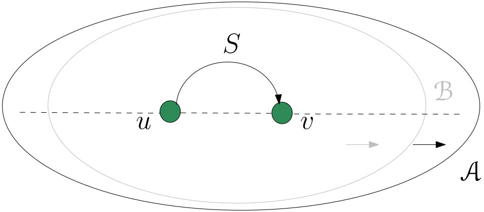
Ψ is hereditary

- Suppose $\mathcal{A} \models \Psi$ where $\Psi := \xi_1 \wedge \xi_2 \wedge \xi_3$, and $\mathcal{B} \subseteq \mathcal{A}$.
- Since ξ_1 is universal, it is hereditary, so $\mathcal{B} \models \xi_1$.
- Consider ξ_2.



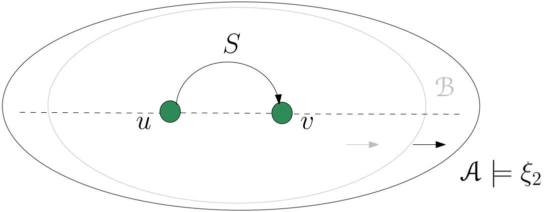
Ψ is hereditary

- Suppose $\mathcal{A} \models \Psi$ where $\Psi := \xi_1 \wedge \xi_2 \wedge \xi_3$, and $\mathcal{B} \subseteq \mathcal{A}$.
- Since ξ_1 is universal, it is hereditary, so $\mathcal{B} \models \xi_1$.
- Consider ξ_2.



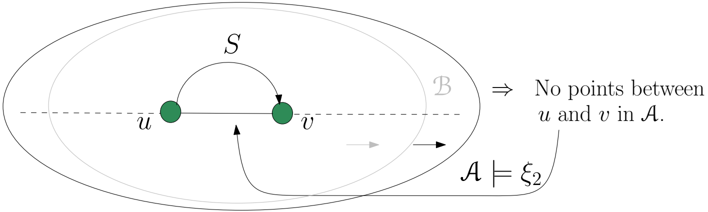
Ψ is hereditary

- Suppose $\mathcal{A} \models \Psi$ where $\Psi := \xi_1 \wedge \xi_2 \wedge \xi_3$, and $\mathcal{B} \subseteq \mathcal{A}$.
- Since ξ_1 is universal, it is hereditary, so $\mathcal{B} \models \xi_1$.
- Consider ξ_2.



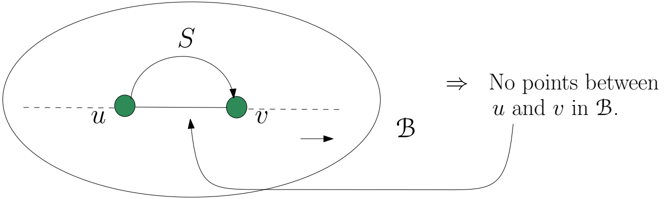
Ψ is hereditary

- Suppose $\mathcal{A} \models \Psi$ where $\Psi := \xi_1 \wedge \xi_2 \wedge \xi_3$, and $\mathcal{B} \subseteq \mathcal{A}$.
- Since ξ_1 is universal, it is hereditary, so $\mathcal{B} \models \xi_1$.
- Consider ξ_2.



Ψ is hereditary

- Suppose $\mathcal{A} \models \Psi$ where $\Psi := \xi_1 \wedge \xi_2 \wedge \xi_3$, and $\mathcal{B} \subseteq \mathcal{A}$.
- Since ξ_1 is universal, it is hereditary, so $\mathcal{B} \models \xi_1$.
- Consider ξ_2.



Ψ is hereditary

- Suppose $\mathcal{A} \models \Psi$ where $\Psi := \xi_1 \wedge \xi_2 \wedge \xi_3$, and $\mathcal{B} \subseteq \mathcal{A}$.
- Since ξ_1 is universal, it is hereditary, so $\mathcal{B} \models \xi_1$.
- As explained, $\mathcal{B} \models \xi_2$.

Ψ is hereditary

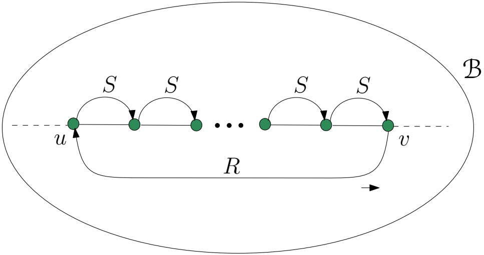
- Suppose $\mathcal{A} \models \Psi$ where $\Psi := \xi_1 \wedge \xi_2 \wedge \xi_3$, and $\mathcal{B} \subseteq \mathcal{A}$.
- Since ξ_1 is universal, it is hereditary, so $\mathcal{B} \models \xi_1$.
- As explained, $\mathcal{B} \models \xi_2$.
- Consider ξ_3. Suppose $\mathcal{B} \not\models \xi_3$.

Ψ is hereditary

- Suppose $\mathcal{A} \models \Psi$ where $\Psi := \xi_1 \wedge \xi_2 \wedge \xi_3$, and $\mathcal{B} \subseteq \mathcal{A}$.
- Since ξ_1 is universal, it is hereditary, so $\mathcal{B} \models \xi_1$.
- As explained, $\mathcal{B} \models \xi_2$.
- Consider ξ_3. Suppose $\mathcal{B} \not\models \xi_3$.

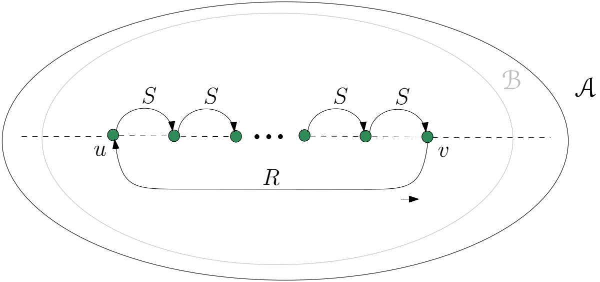
Ψ is hereditary

- Suppose $\mathcal{A} \models \Psi$ where $\Psi := \xi_1 \wedge \xi_2 \wedge \xi_3$, and $\mathcal{B} \subseteq \mathcal{A}$.
- Since ξ_1 is universal, it is hereditary, so $\mathcal{B} \models \xi_1$.
- As explained, $\mathcal{B} \models \xi_2$.
- Consider ξ_3. Suppose $\mathcal{B} \not\models \xi_3$.



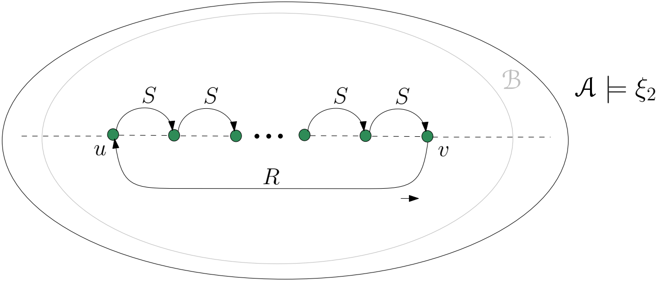
Ψ is hereditary

- Suppose $\mathcal{A} \models \Psi$ where $\Psi := \xi_1 \wedge \xi_2 \wedge \xi_3$, and $\mathcal{B} \subseteq \mathcal{A}$.
- Since ξ_1 is universal, it is hereditary, so $\mathcal{B} \models \xi_1$.
- As explained, $\mathcal{B} \models \xi_2$.
- Consider ξ_3. Suppose $\mathcal{B} \not\models \xi_3$.



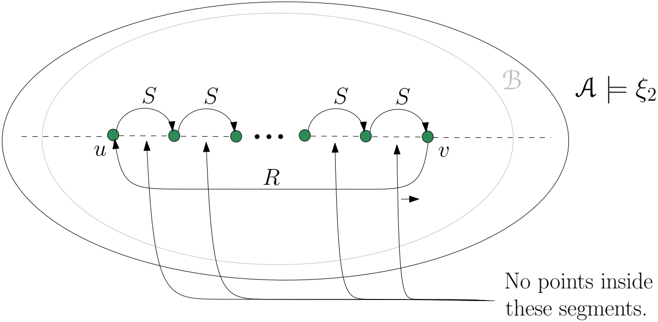
Ψ is hereditary

- Suppose $\mathcal{A} \models \Psi$ where $\Psi := \xi_1 \wedge \xi_2 \wedge \xi_3$, and $\mathcal{B} \subseteq \mathcal{A}$.
- Since ξ_1 is universal, it is hereditary, so $\mathcal{B} \models \xi_1$.
- As explained, $\mathcal{B} \models \xi_2$.
- Consider ξ_3. Suppose $\mathcal{B} \not\models \xi_3$.



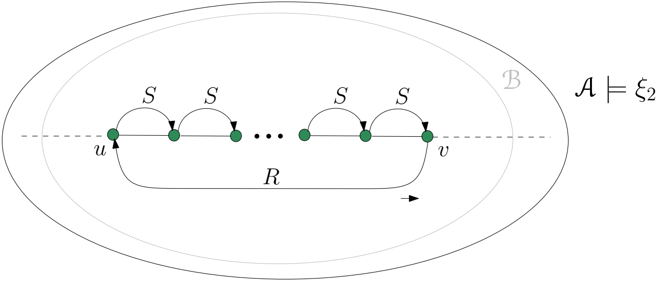
Ψ is hereditary

- Suppose $\mathcal{A} \models \Psi$ where $\Psi := \xi_1 \wedge \xi_2 \wedge \xi_3$, and $\mathcal{B} \subseteq \mathcal{A}$.
- Since ξ_1 is universal, it is hereditary, so $\mathcal{B} \models \xi_1$.
- As explained, $\mathcal{B} \models \xi_2$.
- Consider ξ_3. Suppose $\mathcal{B} \not\models \xi_3$.



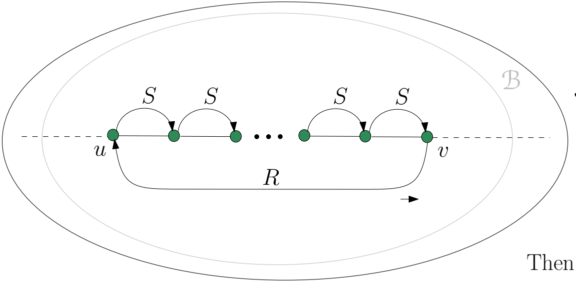
Ψ is hereditary

- Suppose $\mathcal{A} \models \Psi$ where $\Psi := \xi_1 \wedge \xi_2 \wedge \xi_3$, and $\mathcal{B} \subseteq \mathcal{A}$.
- Since ξ_1 is universal, it is hereditary, so $\mathcal{B} \models \xi_1$.
- As explained, $\mathcal{B} \models \xi_2$.
- Consider ξ_3. Suppose $\mathcal{B} \not\models \xi_3$.



Ψ is hereditary

- Suppose $\mathcal{A} \models \Psi$ where $\Psi := \xi_1 \wedge \xi_2 \wedge \xi_3$, and $\mathcal{B} \subseteq \mathcal{A}$.
- Since ξ_1 is universal, it is hereditary, so $\mathcal{B} \models \xi_1$.
- As explained, $\mathcal{B} \models \xi_2$.
- Consider ξ_3. Suppose $\mathcal{B} \not\models \xi_3$.

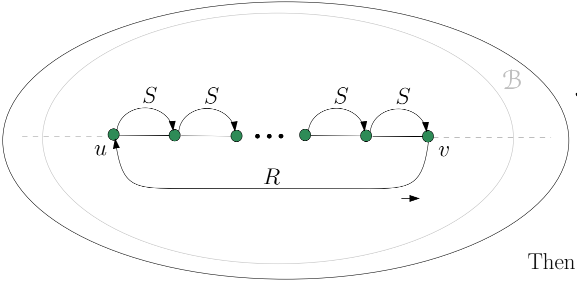


$\mathcal{A} \models \xi_2$

Then $\mathcal{A} \not\models \xi_3$
- a contradiction.

Ψ is hereditary

- Suppose $\mathcal{A} \models \Psi$ where $\Psi := \xi_1 \wedge \xi_2 \wedge \xi_3$, and $\mathcal{B} \subseteq \mathcal{A}$.
- Since ξ_1 is universal, it is hereditary, so $\mathcal{B} \models \xi_1$.
- As explained, $\mathcal{B} \models \xi_2$ and $\mathcal{B} \models \xi_3$; then $\mathfrak{B} \models \Psi$. $\square$



$\mathcal{A} \models \xi_2$

Then $\mathcal{A} \not\models \xi_3$
- a contradiction.

Stronger failure of Łoś-Tarski theorem in the finite

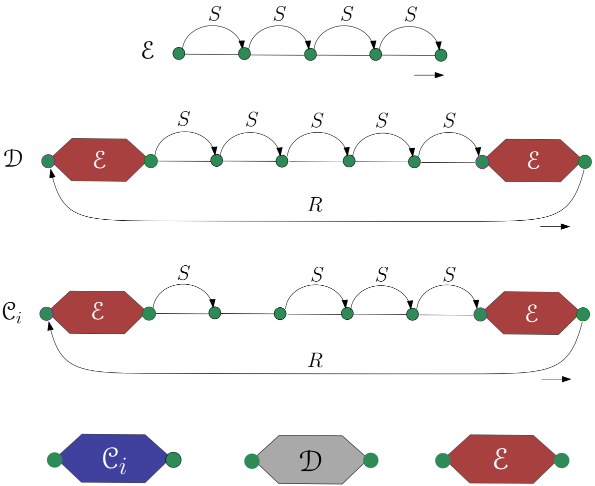
Proposition

The sentence Ψ, which is hereditary over all finite structures, is not equivalent over this class to any Σ_2 sentence.

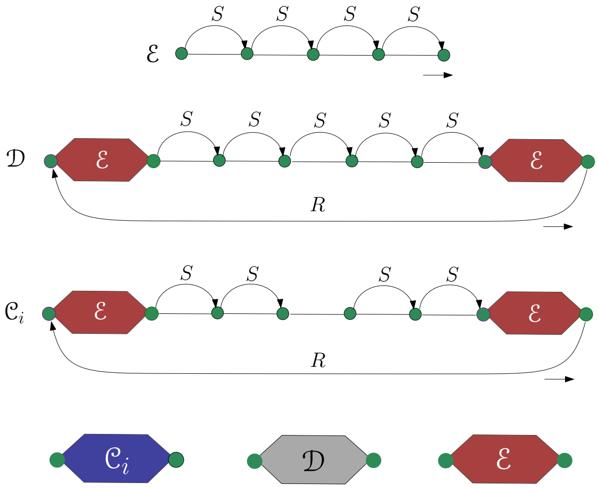
- Let $\Sigma_{2,k}$ = class of Σ_2 sentences in which each block of quantifiers has size k.
- For each k, we construct a class $\mathbf{A}$ of models and a class $\mathbf{B}$ of non-models of Ψ such that for each $\Sigma_{2,k}$ sentence θ, if there is a model of θ in $\mathbf{A}$, then there is a model of θ in $\mathbf{B}$ as well.
- Denote the above condition as $\mathbf{A} \Rrightarrow_{2,k} \mathbf{B}$.
- We illustrate our constructions for $k = 3$.

Construction of
a class of models and a class of non-models

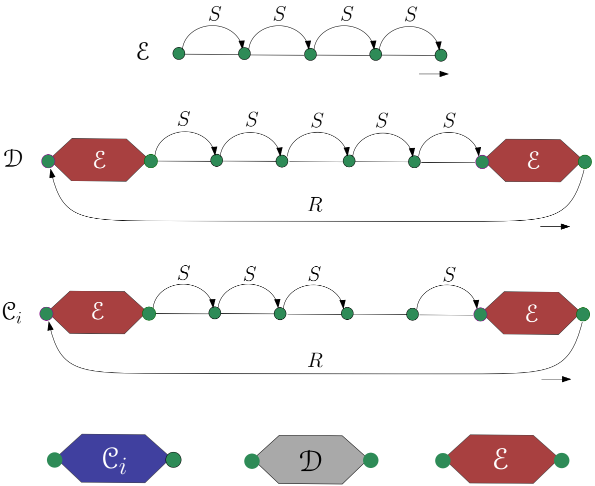
Construction of classes $\mathbf{A}$ and $\mathbf{B}$



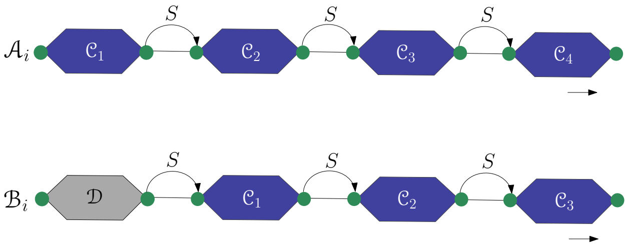
Construction of classes $\mathbb{A}$ and $\mathbb{B}$



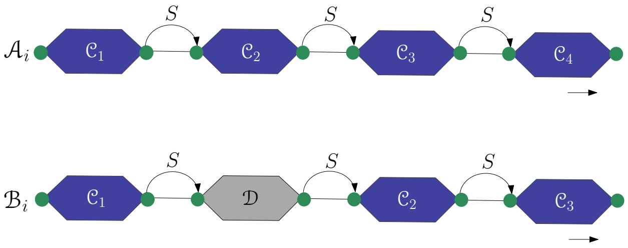
Construction of classes $\mathbf{A}$ and $\mathbf{B}$



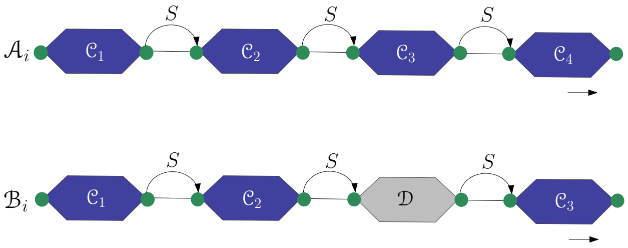
Construction of classes $\mathbf{A}$ and $\mathbf{B}$



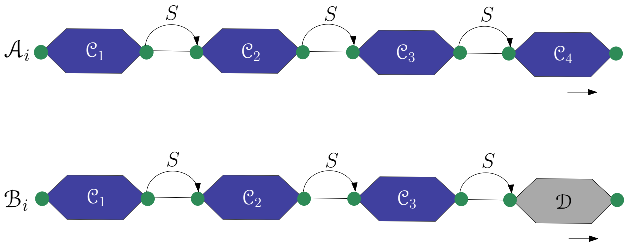
Construction of classes $\mathbf{A}$ and $\mathbf{B}$



Construction of classes $\mathbf{A}$ and $\mathbf{B}$

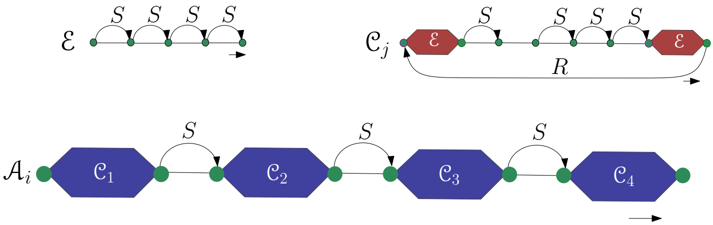


Construction of classes $\mathbf{A}$ and $\mathbf{B}$

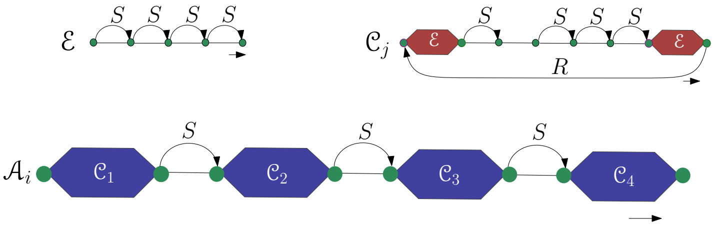


$\mathbf{A}$ = Class of all $\mathcal{A}_i$s; $\mathbf{B}$ = Class of all $\mathcal{B}_i$s

A as a class of models of Ψ

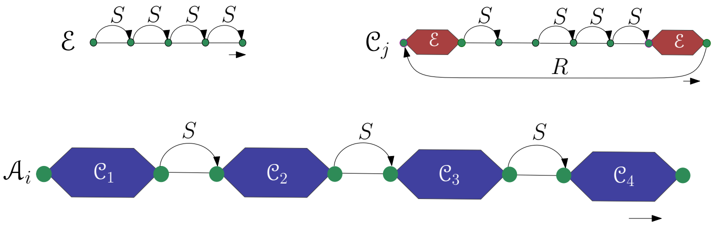


$\mathbb{A}$ as a class of models of Ψ



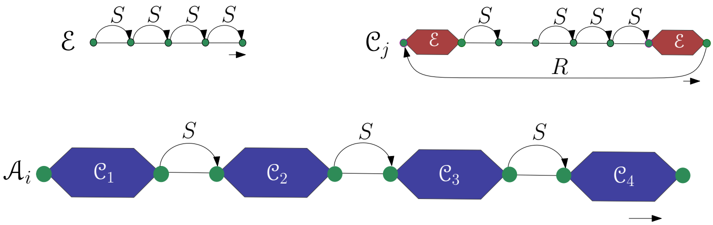
$\xi_1 :=$ " $\leq$ is a linear order"

$\mathbb{A}$ as a class of models of Ψ

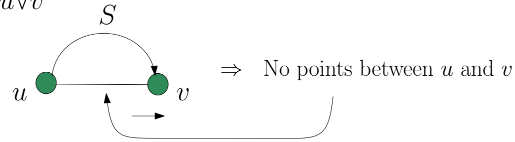


$\xi_1 :=$ " $\leq$ is a linear order" ✓

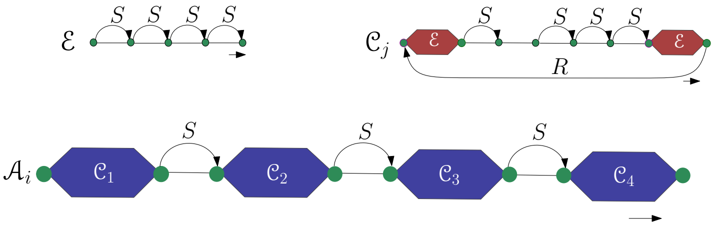
$\mathbb{A}$ as a class of models of Ψ



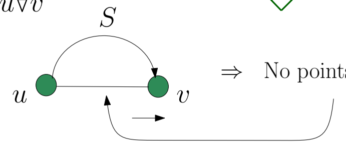
$\xi_2 := \forall u \forall v$



$\mathcal{A}$ as a class of models of Ψ

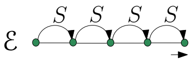


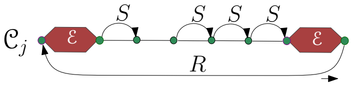
$\xi_2 := \forall u \forall v$

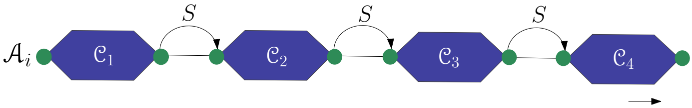


$\Rightarrow$ No points between u and v

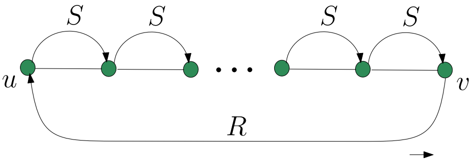
$\mathbb{A}$ as a class of models of Ψ





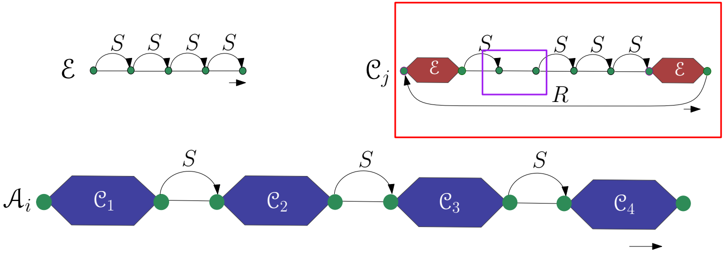


$\xi_3 := \forall u \forall v$

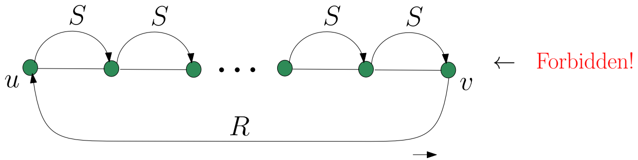


$\leftarrow$ Forbidden!

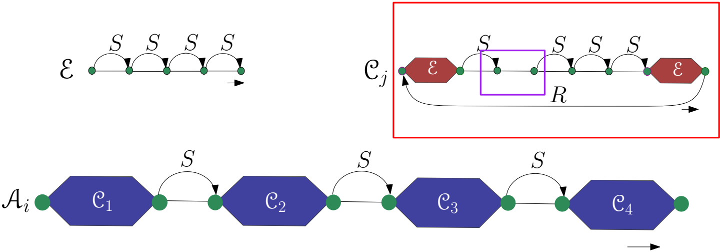
A as a class of models of Ψ



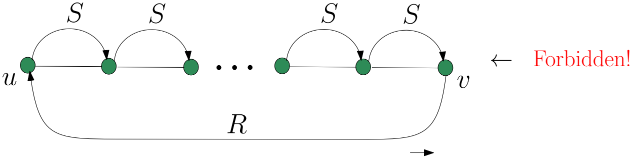
$\xi_3 := \forall u \forall v$



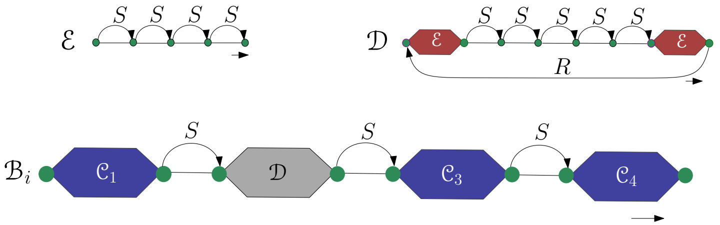
$\mathbb{A}$ as a class of models of Ψ



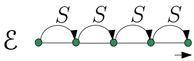
$\xi_3 := \forall u \forall v$

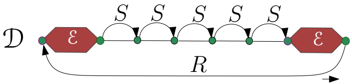


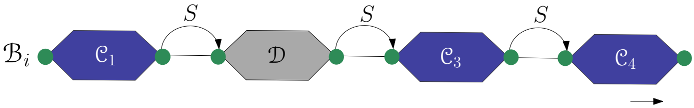
$\mathbb{B}$ as a class of non-models of Ψ



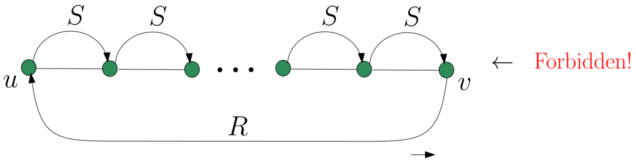
B as a class of non-models of Ψ



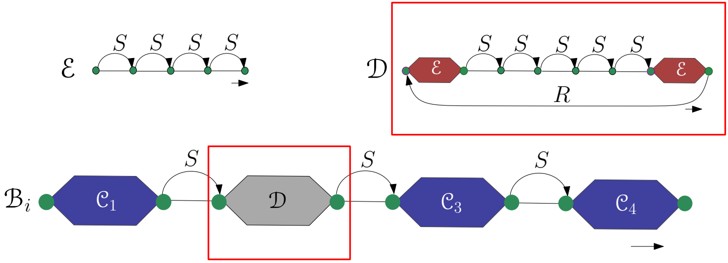




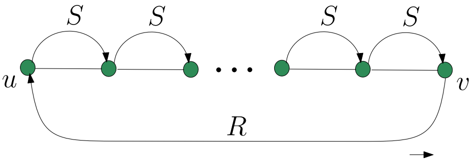
$\xi_3 := \forall u \forall v$



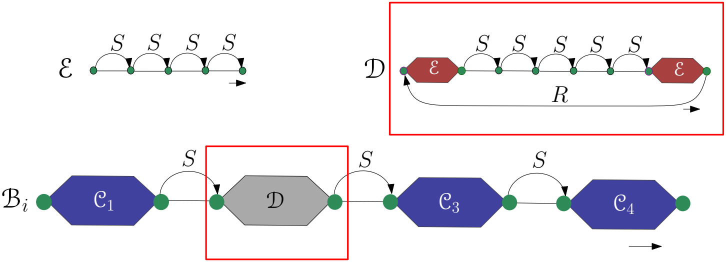
B as a class of non-models of Ψ



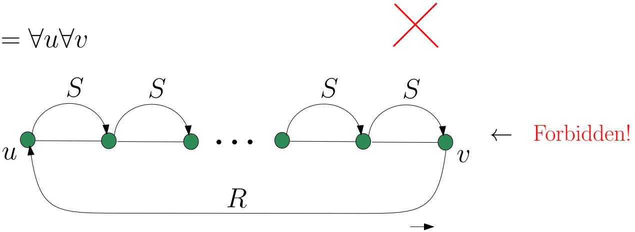
$\xi_3 := \forall u \forall v$



$\leftarrow$ Forbidden!



$\xi_3 := \forall u \forall v$



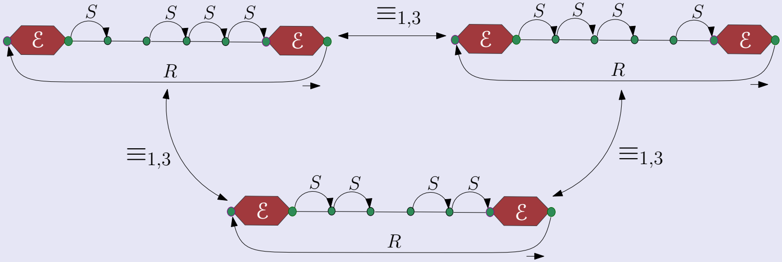
Inexpressibility in Σ_2

Towards showing $\mathbf{A} \Rrightarrow_{2,3} \mathbf{B}$

- For structures $\mathcal{M}_1$ and $\mathcal{M}_2$, denote by $\mathcal{M}_1 \equiv_{1,3} \mathcal{M}_2$ that $\mathcal{M}_1$ and $\mathcal{M}_2$ agree on all sentences of $\Sigma_{1,3}$.
- The relation $\equiv_{1,3}$ is an equivalence relation (of finite index).
- E.g.: any two linear orders of length ≥ 3 are $\equiv_{1,3}$-equivalent.
- One can build pairs of $\equiv_{1,3}$-equivalent structures from given such pairs using operators on structures that satisfy the Feferman-Vaught (FV) composition property.
- A binary operator $\oplus$ satisfies FV composition w.r.t. $\equiv_{1,3}$ if the $\equiv_{1,3}$-class of $\mathcal{M} \oplus \mathcal{N}$ is completely determined by the $\equiv_{1,3}$-classes of $\mathcal{M}$ and $\mathcal{N}$.
- E.g.: the ordered sum of two (ordered) structures satisfies FV-composition w.r.t. $\equiv_{1,3}$.

Some $\equiv_{1,3}$-equivalences

Lemma

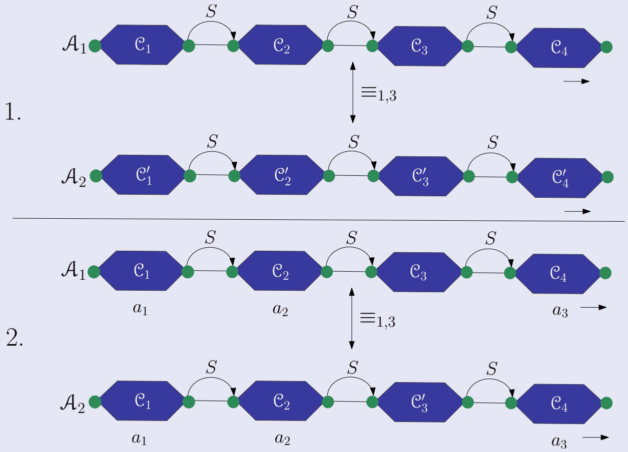


Proof Sketch.

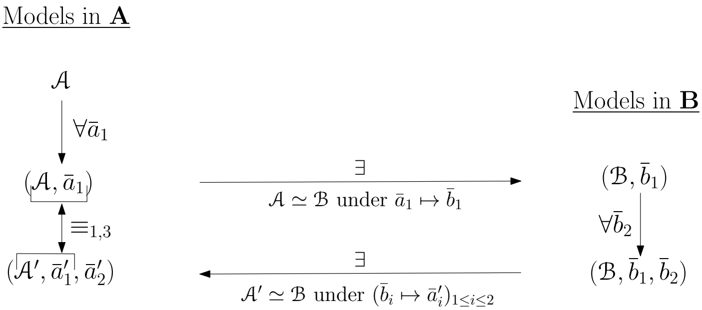
Using FV-composition for the ordered sum of linear orders equipped with a full successor relation and colored endpoints, and the fact that such linear orders of length ≥ 4 are $\equiv_{1,3}$-equivalent. □

Some $\equiv_{1,3}$-equivalences

Corollary

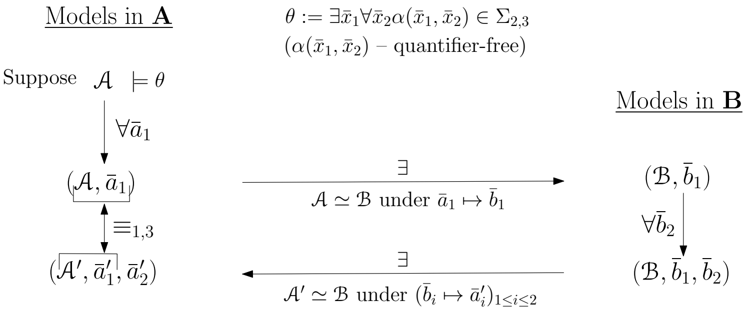


Overview of proof approach for showing $\mathbf{A} \Rrightarrow_{2,3} \mathbf{B}$



"$\mathcal{A} \simeq \mathcal{B}$ under $\bar{a} \mapsto \bar{b}$" $:=$ $\bar{a} \mapsto \bar{b}$ is a partial isomorphism between $\mathcal{A}$ and $\mathcal{B}$.

Overview of proof approach for showing $\mathbf{A} \Rrightarrow_{2,3} \mathbf{B}$

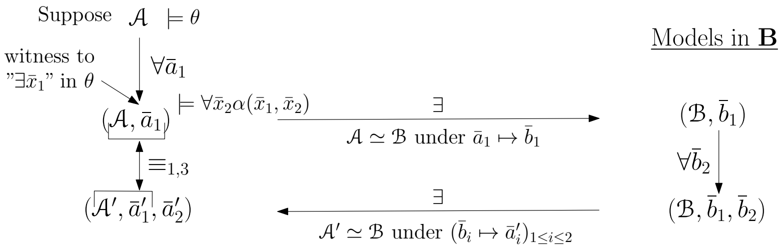


"$\mathcal{A} \simeq \mathcal{B}$ under $\bar{a} \mapsto \bar{b}$" $:= \bar{a} \mapsto \bar{b}$ is a partial isomorphism between $\mathcal{A}$ and $\mathcal{B}$.

Overview of proof approach for showing $\mathbf{A} \Rrightarrow_{2,3} \mathbf{B}$

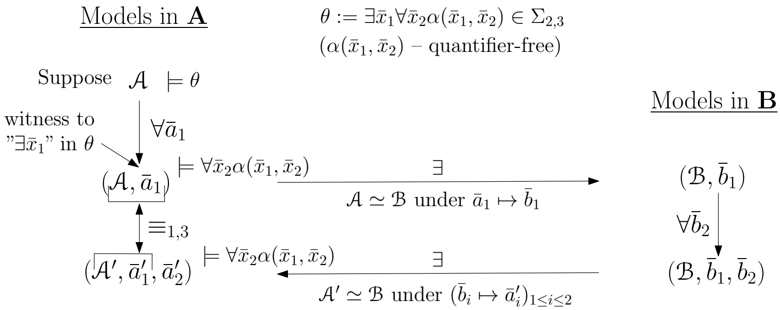
Models in $\mathbf{A}$

$\theta := \exists \bar{x}_1 \forall \bar{x}_2 \alpha(\bar{x}_1, \bar{x}_2) \in \Sigma_{2,3}$
$(\alpha(\bar{x}_1, \bar{x}_2)$ – quantifier-free)



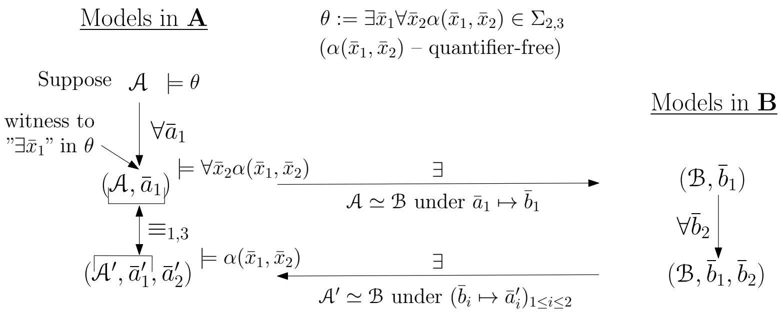
"$\mathcal{A} \simeq \mathcal{B}$ under $\bar{a} \mapsto \bar{b}$" $:= \bar{a} \mapsto \bar{b}$ is a partial isomorphism between $\mathcal{A}$ and $\mathcal{B}$.

Overview of proof approach for showing $\mathbf{A} \Rrightarrow_{2,3} \mathbf{B}$



"$\mathcal{A} \simeq \mathcal{B}$ under $\bar{a} \mapsto \bar{b}$" $:= \bar{a} \mapsto \bar{b}$ is a partial isomorphism between $\mathcal{A}$ and $\mathcal{B}$.

Overview of proof approach for showing $\mathbf{A} \Rrightarrow_{2,3} \mathbf{B}$

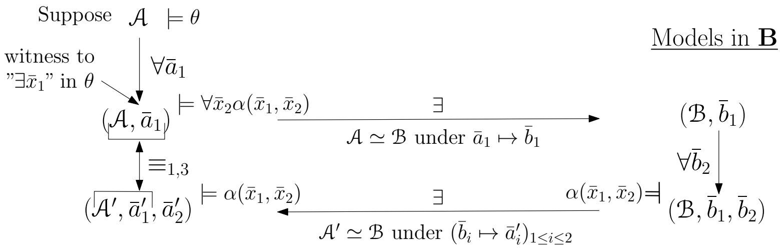


"$\mathcal{A} \simeq \mathcal{B}$ under $\bar{a} \mapsto \bar{b}$" $:= \bar{a} \mapsto \bar{b}$ is a partial isomorphism between $\mathcal{A}$ and $\mathcal{B}$.

Overview of proof approach for showing $\mathbf{A} \Rrightarrow_{2,3} \mathbf{B}$

<u>Models in $\mathbf{A}$</u>

$\theta := \exists \bar{x}_1 \forall \bar{x}_2 \alpha(\bar{x}_1, \bar{x}_2) \in \Sigma_{2,3}$
$(\alpha(\bar{x}_1, \bar{x}_2)$ – quantifier-free$)$

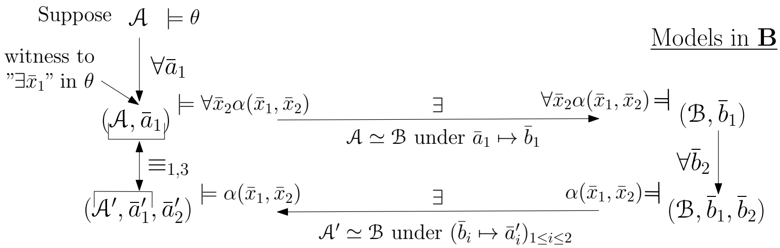


"$\mathcal{A} \simeq \mathcal{B}$ under $\bar{a} \mapsto \bar{b}$" $:= \bar{a} \mapsto \bar{b}$ is a partial isomorphism between $\mathcal{A}$ and $\mathcal{B}$.

Overview of proof approach for showing $\mathbf{A} \Rrightarrow_{2,3} \mathbf{B}$

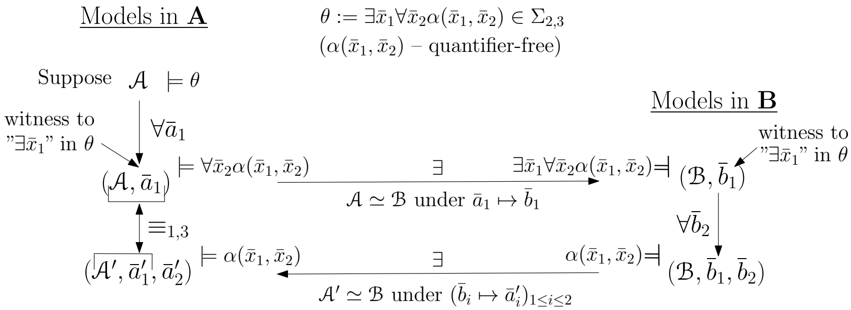
Models in **A**

$\theta := \exists \bar{x}_1 \forall \bar{x}_2 \alpha(\bar{x}_1, \bar{x}_2) \in \Sigma_{2,3}$
$(\alpha(\bar{x}_1, \bar{x}_2)$ – quantifier-free$)$



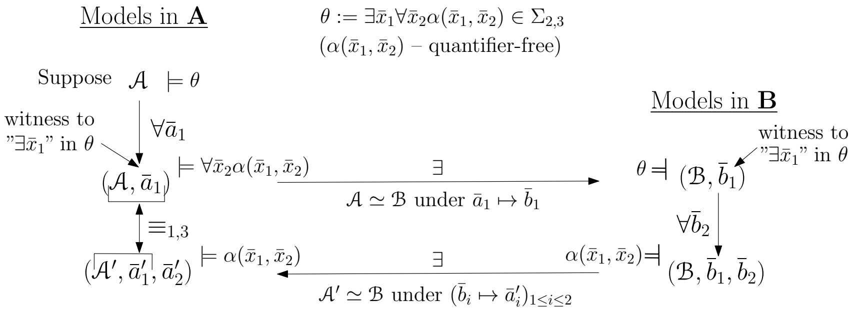
"$\mathcal{A} \simeq \mathcal{B}$ under $\bar{a} \mapsto \bar{b}$" $:= \bar{a} \mapsto \bar{b}$ is a partial isomorphism between $\mathcal{A}$ and $\mathcal{B}$.

Overview of proof approach for showing $\mathbf{A} \Rrightarrow_{2,3} \mathbf{B}$



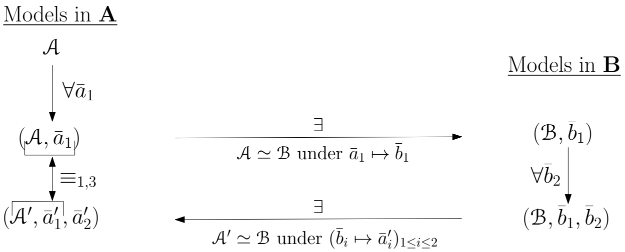
"$\mathcal{A} \simeq \mathcal{B}$ under $\bar{a} \mapsto \bar{b}$" $:= \bar{a} \mapsto \bar{b}$ is a partial isomorphism between $\mathcal{A}$ and $\mathcal{B}$.

Overview of proof approach for showing $\mathbf{A} \Rrightarrow_{2,3} \mathbf{B}$



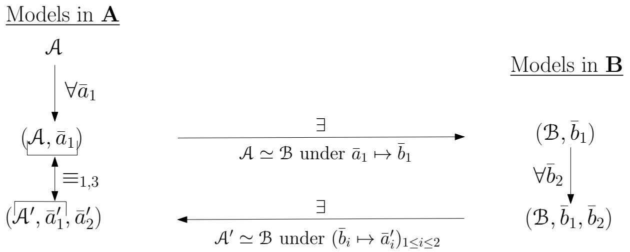
"$\mathcal{A} \simeq \mathcal{B}$ under $\bar{a} \mapsto \bar{b}$" $:= \bar{a} \mapsto \bar{b}$ is a partial isomorphism between $\mathcal{A}$ and $\mathcal{B}$.

Proof approach as an Ehrenfeucht-Fräissé game



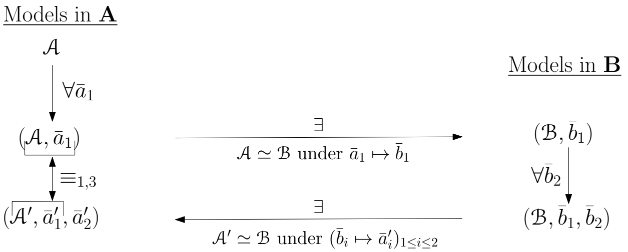
- Players: Spoiler, Duplicator; Game arena: just structure $\mathcal{A}$ initially.
- Round 1: Spoiler picks a 3-tuple $\bar{a}_1$ from $\mathcal{A}$. In response, Duplicator first chooses $\mathcal{B} \in \mathbf{B}$; then picks a 3-tuple $\bar{b}_1$ from $\mathcal{B}$.
- Winning condition: Duplicator wins the round if $\mathcal{A} \simeq \mathcal{B}$ under the map $\bar{a}_1 \mapsto \bar{b}_1$. Else Spoiler wins (this play of) the game.

Proof approach as an Ehrenfeucht-Fräissé game



- Round 2: Spoiler picks a 3-tuple $\bar{b}_2$ from $\mathcal{B}$. In response, Duplicator first chooses $\mathcal{A}' \in \mathbf{A}$; then picks 3-tuples $\bar{a}_1', \bar{a}_2'$ from $\mathcal{A}'$.
- Winning condition: Duplicator wins the round and (this play of) the game if (i) $(\mathcal{A}', \bar{a}_1') \equiv_{1,3} (\mathcal{A}, \bar{a}_1)$; (ii) $\mathcal{A}' \simeq \mathcal{B}$ under $(\bar{a}_i' \mapsto \bar{b}_i)_{1 \leq i \leq 2}$. (Else Spoiler wins.)

Proof approach as an Ehrenfeucht-Fräissé game

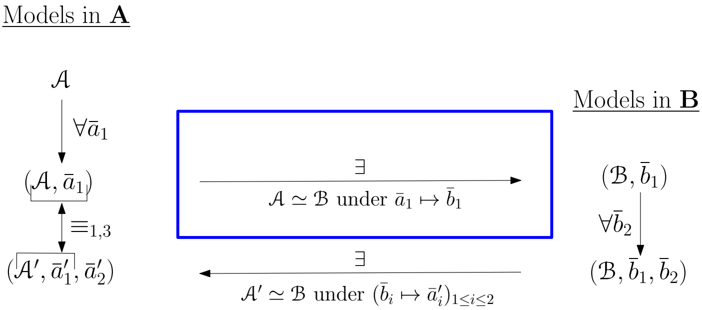


- Duplicator has a winning strategy in the game described if she wins every play of the game.

Proposition

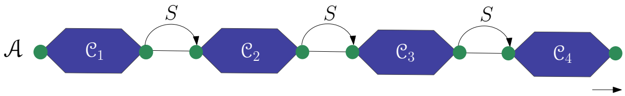
If Duplicator has winning strategy in the described game, then $\mathbf{A} \Rrightarrow_{2,3} \mathbf{B}$.

Executing proof approach for $\mathbf{A} \Rrightarrow_{2,3} \mathbf{B}$: From $\mathbf{A}$ to $\mathbf{B}$

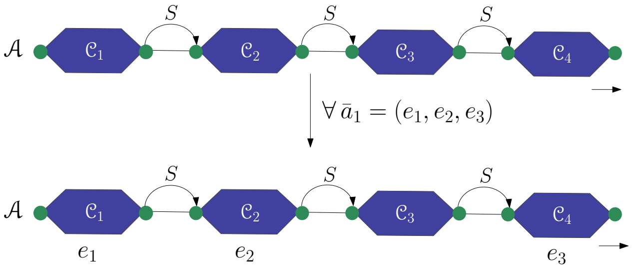


"$\mathcal{A} \simeq \mathcal{B}$ under $\bar{a} \mapsto \bar{b}$" $:=$ $\bar{a} \mapsto \bar{b}$ is a partial isomorphism between $\mathcal{A}$ and $\mathcal{B}$.

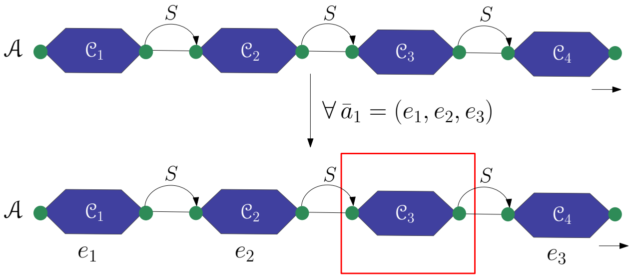
Executing proof approach for $\mathbf{A} \Rrightarrow_{2,3} \mathbf{B}$: From $\mathbf{A}$ to $\mathbf{B}$



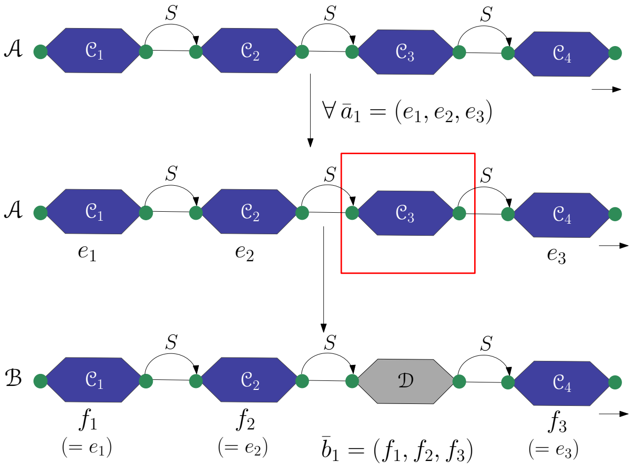
Executing proof approach for $\mathbf{A} \Rrightarrow_{2,3} \mathbf{B}$: From $\mathbf{A}$ to $\mathbf{B}$



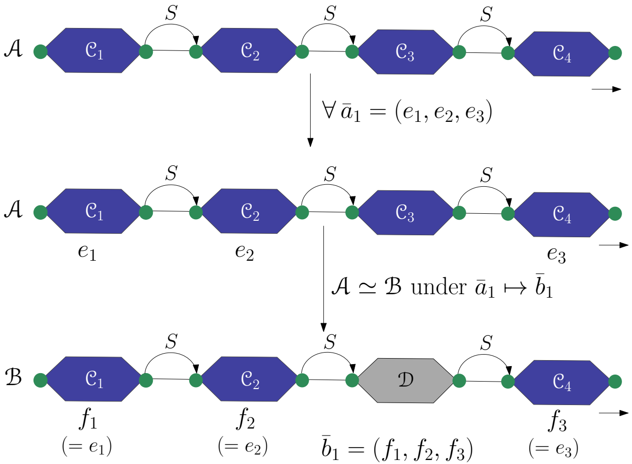
Executing proof approach for $\mathbf{A} \Rrightarrow_{2,3} \mathbf{B}$: From $\mathbf{A}$ to $\mathbf{B}$



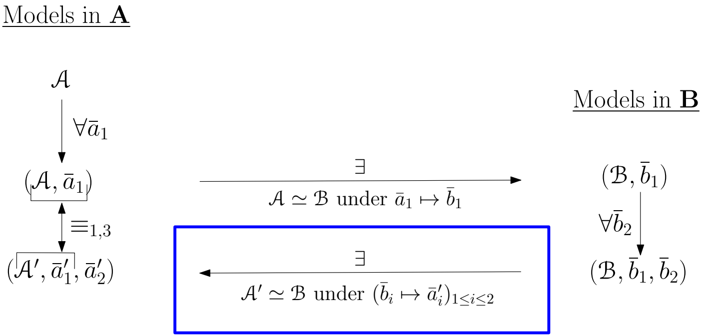
Executing proof approach for $\mathbf{A} \Rrightarrow_{2,3} \mathbf{B}$: From $\mathbf{A}$ to $\mathbf{B}$



Executing proof approach for $\mathbf{A} \Rrightarrow_{2,3} \mathbf{B}$: From $\mathbf{A}$ to $\mathbf{B}$

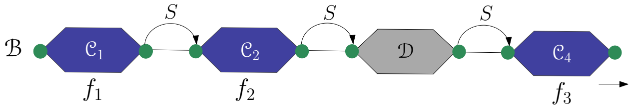


Executing proof approach for $\mathbf{A} \Rrightarrow_{2,3} \mathbf{B}$: From $\mathbf{B}$ to $\mathbf{A}$

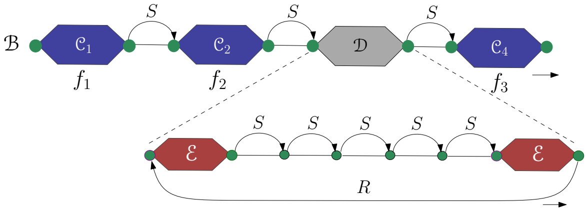


"$\mathcal{A} \simeq \mathcal{B}$ under $\bar{a} \mapsto \bar{b}$" $:=$ $\bar{a} \mapsto \bar{b}$ is a partial isomorphism between $\mathcal{A}$ and $\mathcal{B}$.

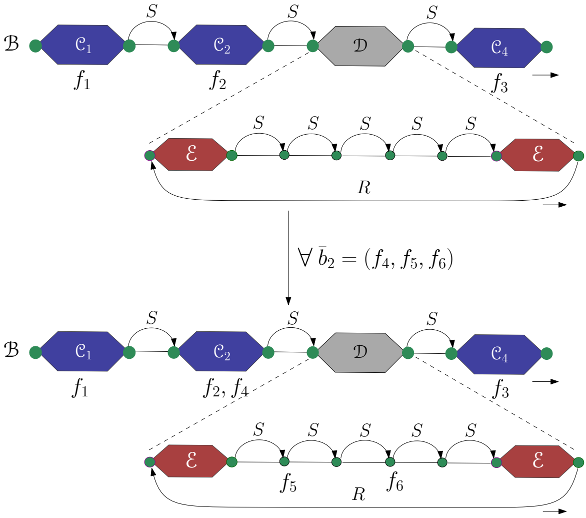
Executing proof approach for $\mathbf{A} \Rrightarrow_{2,3} \mathbf{B}$: From $\mathbf{B}$ to $\mathbf{A}$



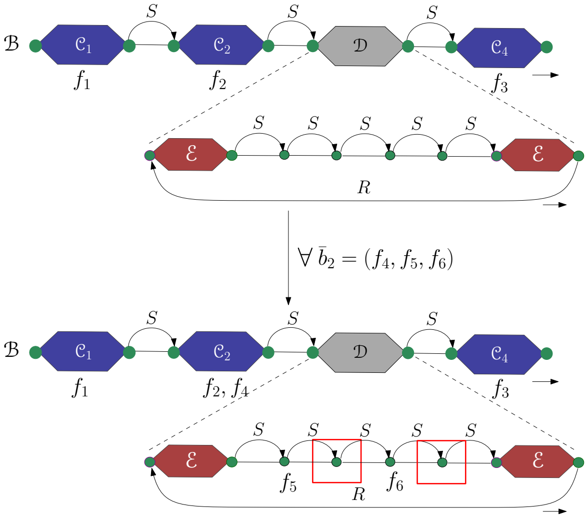
Executing proof approach for $\mathbf{A} \Rrightarrow_{2,3} \mathbf{B}$: From $\mathbf{B}$ to $\mathbf{A}$



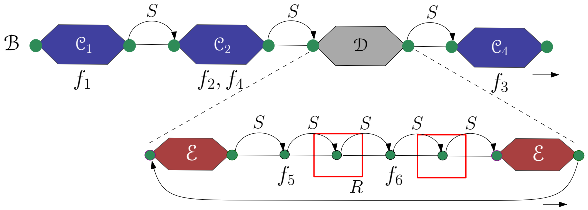
Executing proof approach for $\mathbf{A} \Rrightarrow_{2,3} \mathbf{B}$: From $\mathbf{B}$ to $\mathbf{A}$



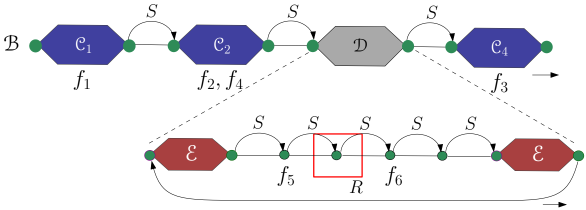
Executing proof approach for $\mathbf{A} \Rrightarrow_{2,3} \mathbf{B}$: From $\mathbf{B}$ to $\mathbf{A}$



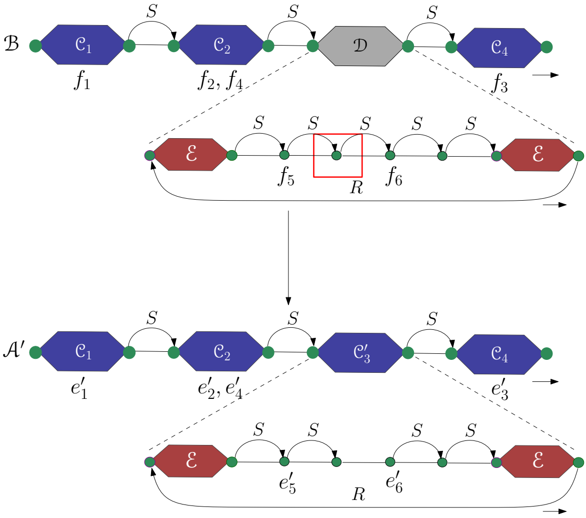
Executing proof approach for $\mathbf{A} \Rrightarrow_{2,3} \mathbf{B}$: From $\mathbf{B}$ to $\mathbf{A}$



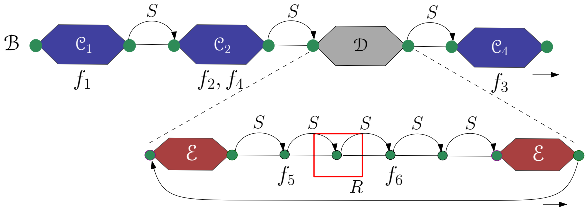
Executing proof approach for $\mathbf{A} \Rrightarrow_{2,3} \mathbf{B}$: From $\mathbf{B}$ to $\mathbf{A}$



Executing proof approach for $\mathbf{A} \Rrightarrow_{2,3} \mathbf{B}$: From $\mathbf{B}$ to $\mathbf{A}$

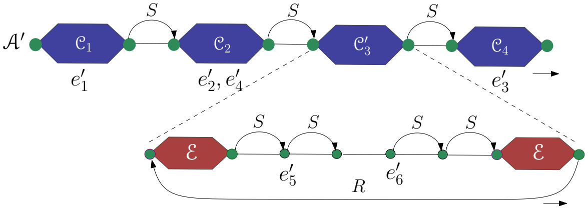


Executing proof approach for $\mathbf{A} \Rrightarrow_{2,3} \mathbf{B}$: From $\mathbf{B}$ to $\mathbf{A}$

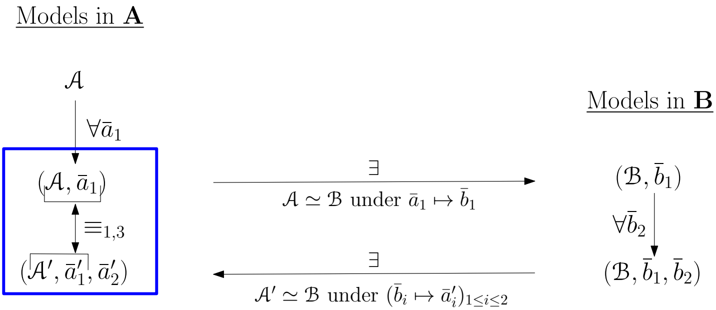


$\bar{a}'_1 = (e'_1, e'_2, e'_3) = \bar{b}_1$
$\bar{a}'_2 = (e'_4, e'_5, e'_6) = \bar{b}_2$

$B \simeq A'$ under $(\bar{b}_i \mapsto \bar{a}'_i)_{1 \leq i \leq 2}$

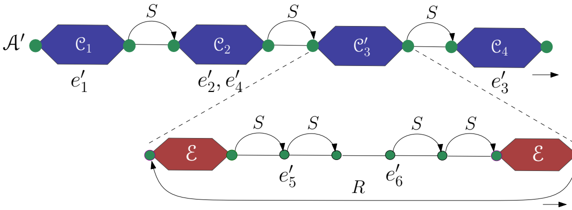


Executing proof approach for $\mathbf{A} \Rrightarrow_{2,3} \mathbf{B}$: Showing $(\mathcal{A}, \bar{a}_1) \equiv_{1,3} (\mathcal{A}', \bar{a}'_1)$

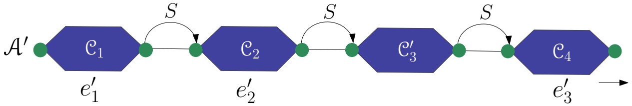


"$\mathcal{A} \simeq \mathcal{B}$ under $\bar{a} \mapsto \bar{b}$" $:=$ $\bar{a} \mapsto \bar{b}$ is a partial isomorphism between $\mathcal{A}$ and $\mathcal{B}$.

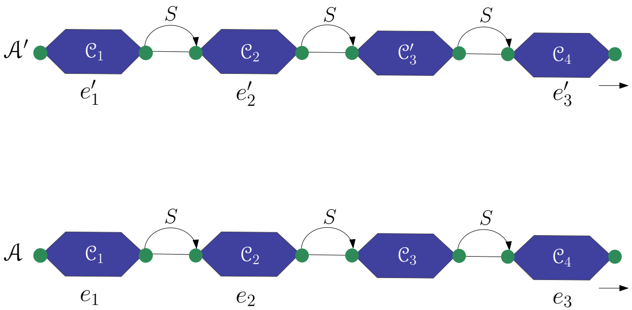
Executing proof approach for $\mathbf{A} \Rrightarrow_{2,3} \mathbf{B}$: Showing $(\mathcal{A}, \bar{a}_1) \equiv_{1,3} (\mathcal{A}', \bar{a}'_1)$



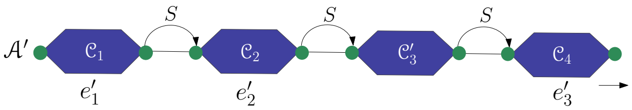
Executing proof approach for $\mathbf{A} \Rrightarrow_{2,3} \mathbf{B}$: Showing $(\mathcal{A}, \bar{a}_1) \equiv_{1,3} (\mathcal{A}', \bar{a}_1')$



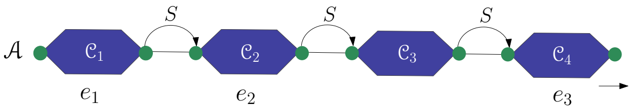
Executing proof approach for $\mathbf{A} \Rrightarrow_{2,3} \mathbf{B}$: Showing $(\mathcal{A}, \bar{a}_1) \equiv_{1,3} (\mathcal{A}', \bar{a}'_1)$



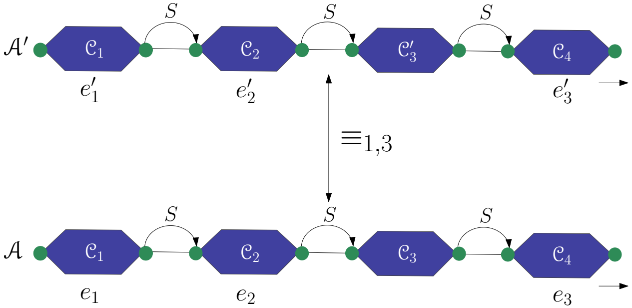
Executing proof approach for $\mathbf{A} \Rrightarrow_{2,3} \mathbf{B}$: Showing $(\mathcal{A}, \bar{a}_1) \equiv_{1,3} (\mathcal{A}', \bar{a}'_1)$



$$e'_1 = e_1, e'_2 = e_2, e'_3 = e_3$$



Executing proof approach for $\mathbf{A} \Rrightarrow_{2,3} \mathbf{B}$: Showing $(\mathcal{A}, \bar{a}_1) \equiv_{1,3} (\mathcal{A}', \bar{a}'_1)$



Inexpressibility in Σ_3

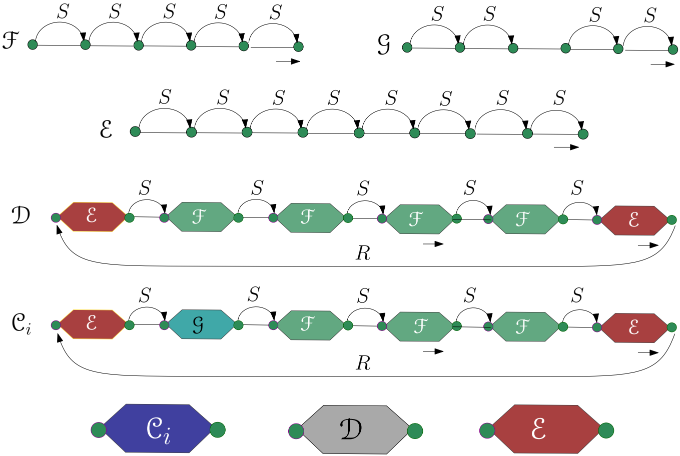
Even stronger failure of Łoś-Tarski theorem in the finite

Proposition

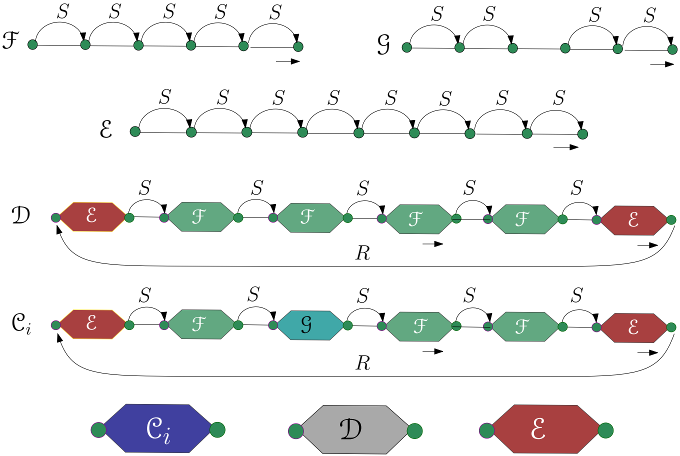
The sentence Ψ, which is hereditary over all finite structures, is not equivalent over this class to any Σ_3 sentence.

- Let $\Sigma_{3,k}$ = class of Σ_3 sentences in which each block of quantifiers has size k.
- For each k, we construct a class $\mathbf{A}$ of models and a class $\mathbf{B}$ of non-models of Ψ such that $\mathbf{A} \Rrightarrow_{3,k} \mathbf{B}$ holds: for each $\Sigma_{3,k}$ sentence θ, if $\mathbf{A}$ contains a model of θ, then so does $\mathbf{B}$.
- We illustrate our constructions for $k = 3$.

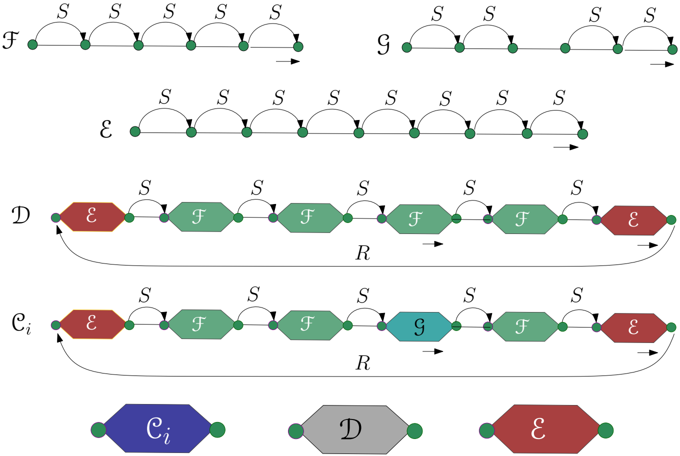
Construction of classes $\mathbf{A}$ and $\mathbf{B}$



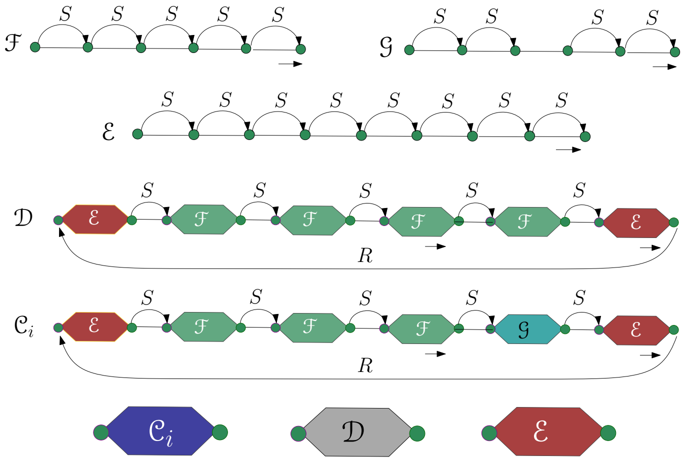
Construction of classes $\mathbf{A}$ and $\mathbf{B}$



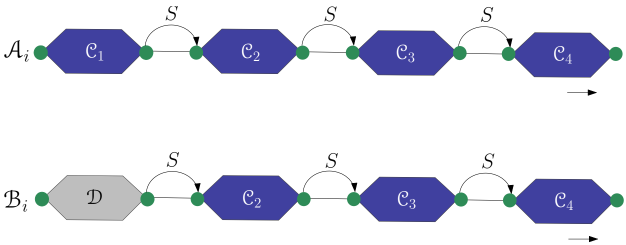
Construction of classes $\mathbf{A}$ and $\mathbf{B}$



Construction of classes $\mathbf{A}$ and $\mathbf{B}$

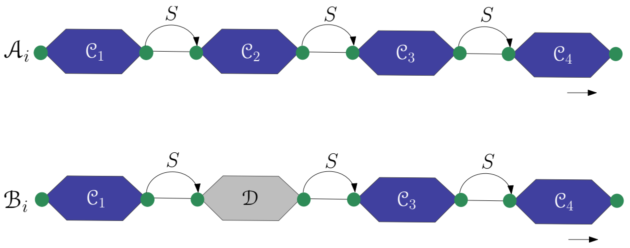


Construction of classes $\mathbf{A}$ and $\mathbf{B}$



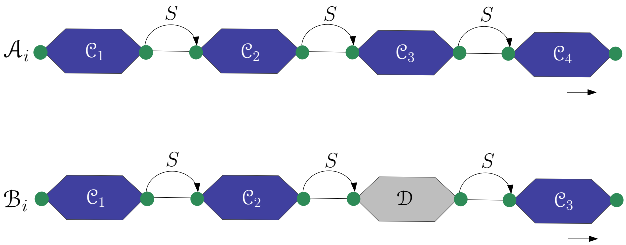
$\mathbf{A}$ = Class of all $\mathcal{A}_i$s; $\mathbf{B}$ = Class of all $\mathcal{B}_i$s

Construction of classes $\mathbf{A}$ and $\mathbf{B}$



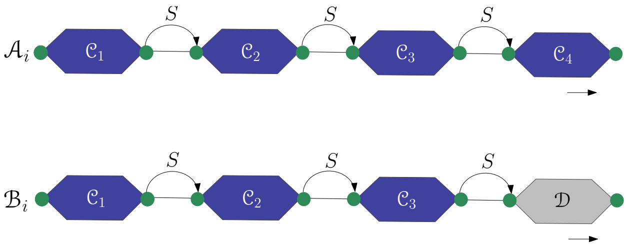
$\mathbf{A}$ = Class of all $\mathcal{A}_i$s; $\mathbf{B}$ = Class of all $\mathcal{B}_i$s

Construction of classes $\mathbf{A}$ and $\mathbf{B}$



$\mathbf{A}$ = Class of all $\mathcal{A}_i$s; $\mathbf{B}$ = Class of all $\mathcal{B}_i$s

Construction of classes $\mathbf{A}$ and $\mathbf{B}$



$\mathbf{A}$ = Class of all $\mathcal{A}_i$s; $\mathbf{B}$ = Class of all $\mathcal{B}_i$s

Similarly as before, it can be shown that $\mathcal{A}_i \models \Psi$ and $\mathcal{B}_i \models \neg\Psi$.

Towards showing $\mathbf{A} \Rrightarrow_{3,3} \mathbf{B}$

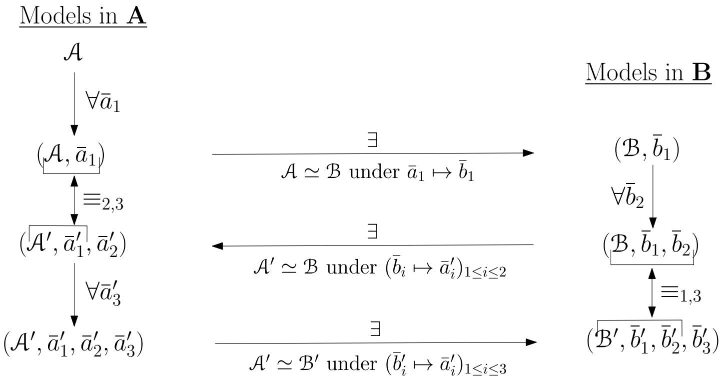
- For structures $\mathcal{M}_1$ and $\mathcal{M}_2$, denote by $\mathcal{M}_1 \equiv_{2,3} \mathcal{M}_2$ that $\mathcal{M}_1$ and $\mathcal{M}_2$ agree on all sentences of $\Sigma_{2,3}$.

Lemma

The following equivalences hold for any i, j:

- $\mathcal{C}_i \equiv_{2,3} \mathcal{C}_j$
- $\mathcal{A}_i \equiv_{2,3} \mathcal{A}_j$
- $(\mathcal{A}, \bar{a}) \equiv_{2,3} (\mathcal{A}', \bar{a})$ where $\bar{a}$ is a 3-tuple and $\mathcal{A}'$ is obtained from $\mathcal{A}$ by replacing the $\mathcal{C}_i$ segment not touched by $\bar{a}$, with $\mathcal{C}_j$

Overview of proof approach for showing $\mathbf{A} \Rrightarrow_{3,3} \mathbf{B}$



Expressibility in Datalog($\neq, \neg$)

Datalog syntax

- A Datalog($\neq, \neg$) rule is of one of the foll. forms:

$$\begin{array}{rcl} R(\bar{x}) & \longleftarrow & A(\bar{x}_1) \\ R(\bar{x}) & \longleftarrow & R_1(\bar{x}_1), \ldots, R_n(\bar{x}_n) \end{array}$$

- In the first rule above, $A(\bar{x}_1)$ is an atom that can appear negated. Also A can be equality or its negation.
- In the second rule above, all predicates R_i that are not atoms appear un-negated. Also, R can be one of the R_is.
- In both rules, the variables appearing in the LHS are a subset of the variables appearing in the RHS.
- A Datalog($\neq, \neg$) program is a finite set of Datalog rules.

Datalog model-theoretic semantics

- Consider the following Datalog program:

$$\begin{array}{rcl} R(x,y) & \longleftarrow & A(x,z), B(z,y) \\ R(x,y) & \longleftarrow & \neg A(x,z), R(x,y) \end{array}$$

- The first rule as a program by itself corresponds to

$$\alpha(x,y) := \exists z (A(x,z) \wedge B(z,y))$$

- With both rules, the program corresponds to the existential least fixpoint logic sentence $\beta(x,y)$ given as below:

$$\begin{array}{rcl} \beta(x,y) & := & \mathsf{LFP}_{R,u,v}\varphi(R,u,v)](x,y) \\ \varphi(R,u,v) & := & \alpha(u,v) \vee \exists z(\neg A(u,z) \wedge R(u,v)) \end{array}$$

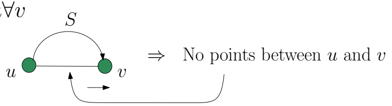
- Datalog($\neq, \neg$) corresponds exactly to existential least fixpoint logic, and thus any Datalog($\neq, \neg$) program is extension closed.

$\neg\Psi$ as a Datalog program

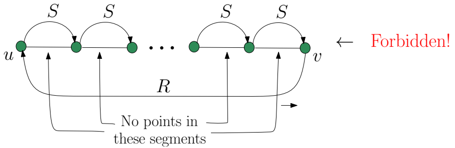
$$\Psi := \xi_1 \wedge \xi_2 \wedge \xi_3 \quad \in \mathrm{FO}(\sigma) \text{ where } \sigma = \{\leq, R, S\}$$

$\xi_1 :=$ " $\leq$ is a linear order"

$\xi_2 := \forall u \forall v$



$\xi_3 := \forall u \forall v$



- Express $\neg\xi_1, \neg\xi_2, \neg\xi_3$ as Datalog programs $\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3$ with "start symbols" T_1, T_2, T_3 resp. Then the Datalog program for $\neg\Psi$ is

$$T \quad \longleftarrow \quad T_1 \mid T_2 \mid T_3$$

$\neg\xi_1$, $\neg\xi_2$ and $\neg\xi_3$ as Datalog programs

$\xi_1 :=$ " $\leq$ is a linear order"

$$\xi_1 := \forall x \forall y \forall z \qquad \begin{pmatrix} x \leq x \;\; \wedge \\ (x \leq y \;\; \wedge \;\; y \leq x) \rightarrow \;\; x = y \;\; \wedge \\ (x \leq y \;\; \wedge \;\; y \leq z) \rightarrow \;\; x \leq z \end{pmatrix}$$

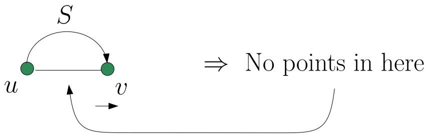
$$\neg\xi_1 := \exists x \exists y \exists z \qquad \begin{pmatrix} \neg\; x \leq x & \vee \\ (x \leq y \wedge y \leq x \wedge x \neq y) & \vee \\ (x \leq y \wedge y \leq z \wedge \neg\; x \leq z) & \end{pmatrix}$$

Datalog program for $\neg\xi_1$:

$$\begin{aligned} T_1 \longleftarrow \; & \neg x \leq x \mid \\ & x \leq y, \; y \leq x, \; x \neq y \;\mid \\ & x \leq y, \; y \leq z, \; \neg\; x \leq z \end{aligned}$$

$\neg\xi_1$, $\neg\xi_2$ and $\neg\xi_3$ as Datalog programs

$\xi_2 := \forall u \forall v$



$$\xi_2 := \forall u \forall v \; S(u,v) \rightarrow \neg \exists z \begin{pmatrix} u \leq z \wedge z \leq v \;\; \wedge \\ u \neq z \wedge z \neq v \end{pmatrix}$$

$$\neg\xi_2 := \exists u \exists v \; S(u,v) \wedge \exists z \begin{pmatrix} u \leq z \wedge z \leq v \;\; \wedge \\ u \neq z \wedge z \neq v \end{pmatrix}$$

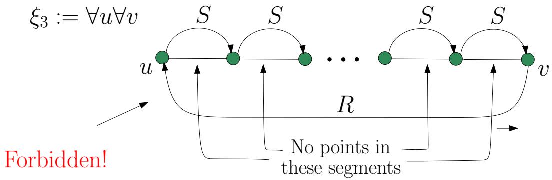
Datalog program for $\neg\xi_2$:

$$T_2 \longleftarrow S(u,v), X(u,v)$$

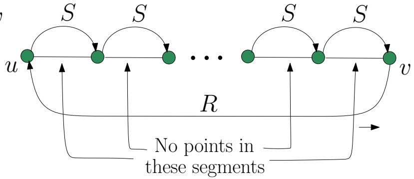
$$X(u,v) \longleftarrow u \leq z, z \leq v, u \neq z, z \neq v$$

$\neg\xi_1$, $\neg\xi_2$ and $\neg\xi_3$ as Datalog programs

$\xi_3 := \forall u \forall v$



$\neg\xi_3 := \exists u \exists v$



Datalog program for $\neg\xi_3$:

$$T_3 \longleftarrow R(v,u), S\text{-reach}(u,v)$$

$$S\text{-reach}(u,v) \longleftarrow S(u,v) \;\mid\; S(u,z), S\text{-reach}(z,v)$$

Generalizing Tait's sentence

Overview

Theorem

For every n, there is a vocabulary τ_n and an FO(τ_n) sentence φ_n such that the following hold:

1. φ_n is hereditary over all finite structures, but is not equivalent over this class to any Σ_n sentence.
2. $\neg\varphi_n$ can be expressed in Datalog$(\neq, \neg)$.

- Construction of φ_n
- Inexpressibility of φ_n in Σ_n using a suitable class of models and non-models

Construction of φ_n

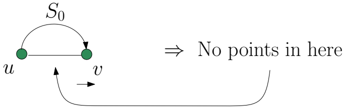
Construction of φ_n

Consider $\Psi_1(x, y)$ over $\sigma_1 = \{\leq, R_1, S_0, S_1\}$ as below.

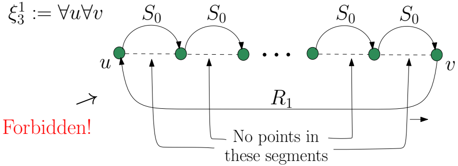
$$\Psi_1(x, y) := S_1(x, y) \to \left(\xi_1^1(x, y) \wedge \xi_2^1 \wedge \xi_3^1\right)$$

$\xi_1^1(x, y) :=$ " $\leq$ is a linear order" $\wedge$
"x is min and y is max under $\leq$"

$\xi_2^1 := \forall u \forall v$



$\xi_3^1 := \forall u \forall v$



Construction of φ_n

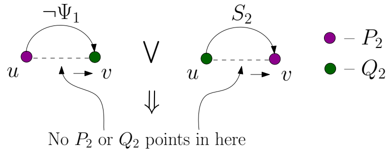
Consider $\Psi_2(x, y)$ over $\sigma_2 = \sigma_1 \cup \{P_2, Q_2, R_2, S_2\}$ as below.

$$\Psi_2(x, y) := S_2(x, y) \rightarrow \left(\xi_1^2(x, y) \wedge \xi_2^2 \wedge \xi_3^2\right)$$

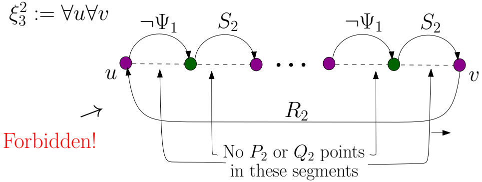
$\xi_1^2(x, y) :=$ " $\leq$ is a linear order" $\wedge$

"x is min and y is max under $\leq$ " $\wedge\ P_2(x) \wedge Q_2(y)$

$\xi_2^2 := \forall u \forall v$



$\xi_3^2 := \forall u \forall v$



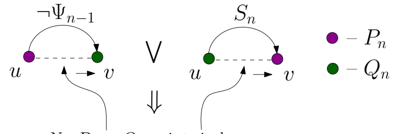
Construction of φ_n

Consider $\Psi_n(x, y)$ over $\sigma_n = \sigma_{n-1} \cup \{P_n, Q_n, R_n, S_n\}$ as below.

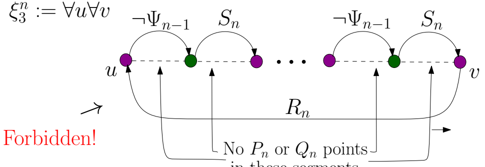
$$\Psi_n(x,y) := S_n(x,y) \rightarrow \left(\xi_1^n(x,y) \wedge \xi_2^n \wedge \xi_3^n\right)$$

$\xi_1^n(x,y) :=$ " $\leq$ is a linear order" $\wedge$
"x is min and y is max under $\leq$" $\wedge P_n(x) \wedge Q_n(y)$

$\xi_2^n := \forall u \forall v$



$\xi_3^n := \forall u \forall v$



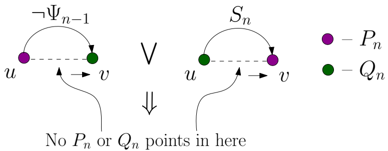
Construction of φ_n

$$\varphi_n := \Psi_n(x,y)[x \mapsto c; y \mapsto d] \text{ for constants } c, d$$

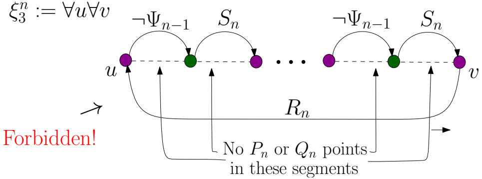
$$\Psi_n(x,y) := S_n(x,y) \to \left(\xi_1^n(x,y) \wedge \xi_2^n \wedge \xi_3^n\right)$$

$$\xi_1^n(x,y) := \text{“} \leq \text{ is a linear order”} \wedge \text{“}x \text{ is min and } y \text{ is max under } \leq \text{”} \wedge P_n(x) \wedge Q_n(y)$$

$\xi_2^n := \forall u \forall v$



$\xi_3^n := \forall u \forall v$



Inexpressibility of φ_n in Σ_{2n}

Theorem

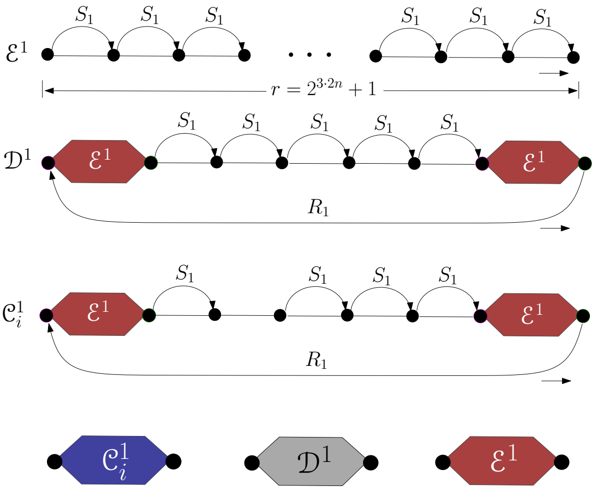
The following are true of $\Psi_n(x, y)$, and therefore also of φ_n:

1. $\neg\Psi_n(x, y)$ is expressible in Datalog$(\neq, \neg)$ and hence $\Psi_n(x, y)$ is hereditary over all finite structures.
2. $\Psi_n(x, y)$ is not equivalent to any Σ_{2n} formula.

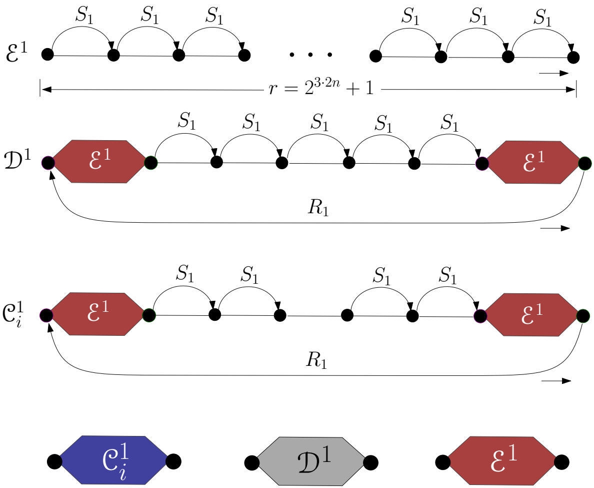
- Define $\Sigma_{2n,k}$ analogously to $\Sigma_{2,k}$ and for formulae.
- For each k, we construct a class $\mathbf{A}^n$ of models and a class $\mathbf{B}^n$ of non-models of $\Psi_n(x, y)$ such that $\mathbf{A}^n \Rrightarrow_{2n,k} \mathbf{B}^n$ holds: for each $\Sigma_{2n,k}$ formula $\theta(x, y)$, if $\mathbf{A}^n$ contains a model of $\theta(x, y)$, then so does $\mathbf{B}^n$.
- We again illustrate our constructions for $k = 3$.

Construction of $\mathbf{A}^n$ and $\mathbf{B}^n$

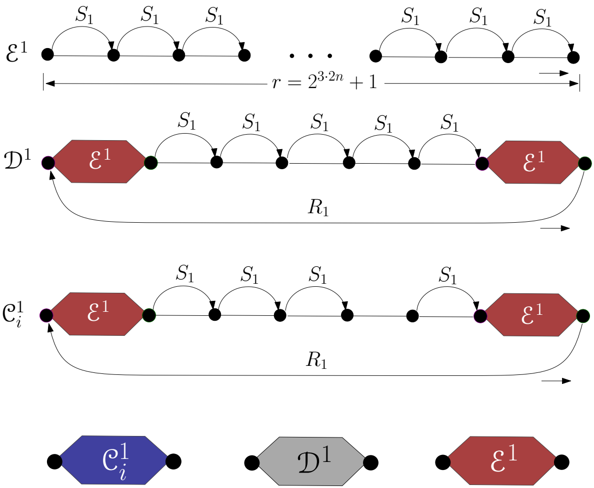
Construction of $\mathbf{A}^n$ and $\mathbf{B}^n$: Structures at level 1



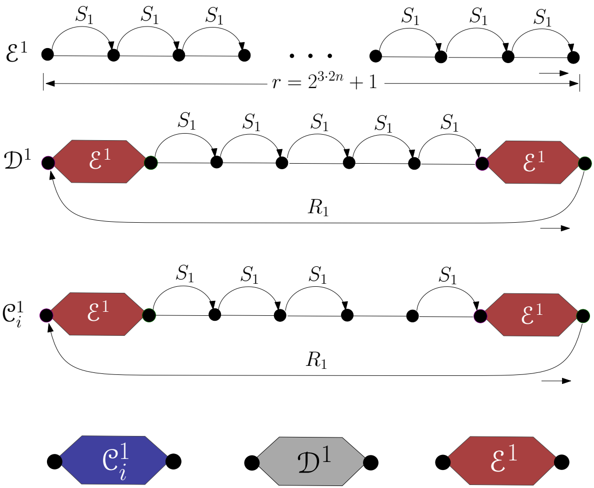
Construction of $\mathbf{A}^n$ and $\mathbf{B}^n$: Structures at level 1



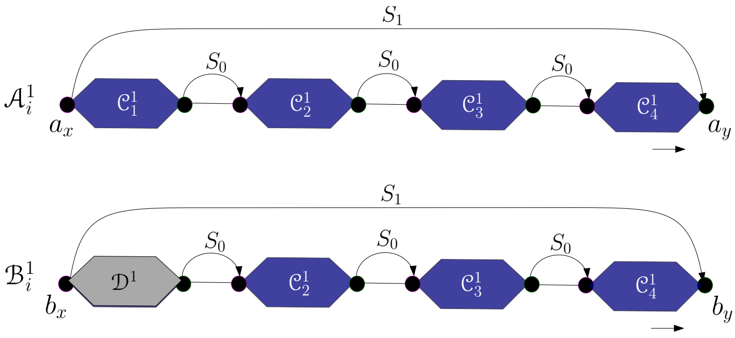
Construction of $\mathbf{A}^n$ and $\mathbf{B}^n$: Structures at level 1



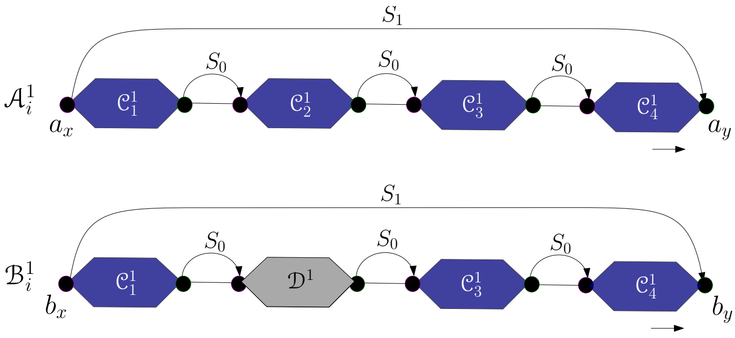
Construction of $\mathbf{A}^n$ and $\mathbf{B}^n$: Structures at level 1



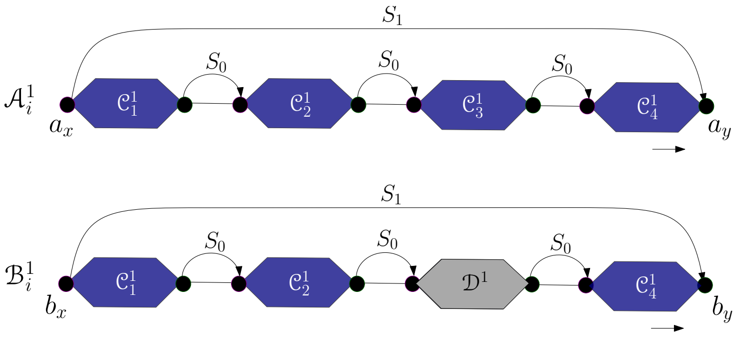
Construction of $\mathbf{A}^n$ and $\mathbf{B}^n$: Structures at level 1



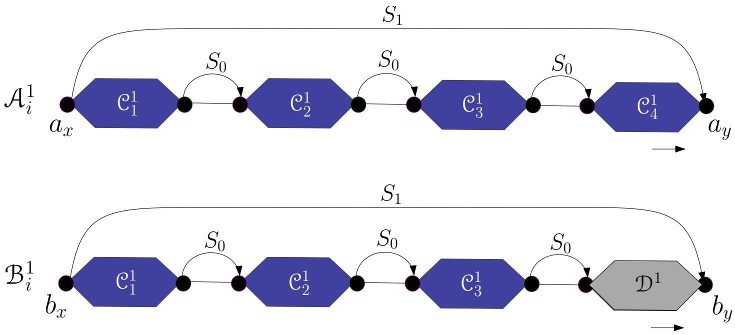
Construction of $\mathbf{A}^n$ and $\mathbf{B}^n$: Structures at level 1



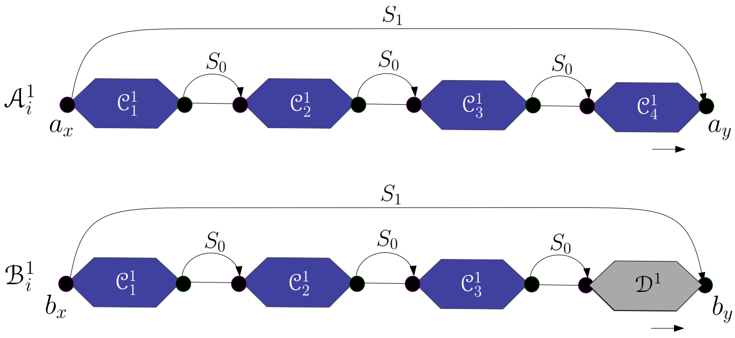
Construction of $\mathbf{A}^n$ and $\mathbf{B}^n$: Structures at level 1



Construction of $\mathbf{A}^n$ and $\mathbf{B}^n$: Structures at level 1

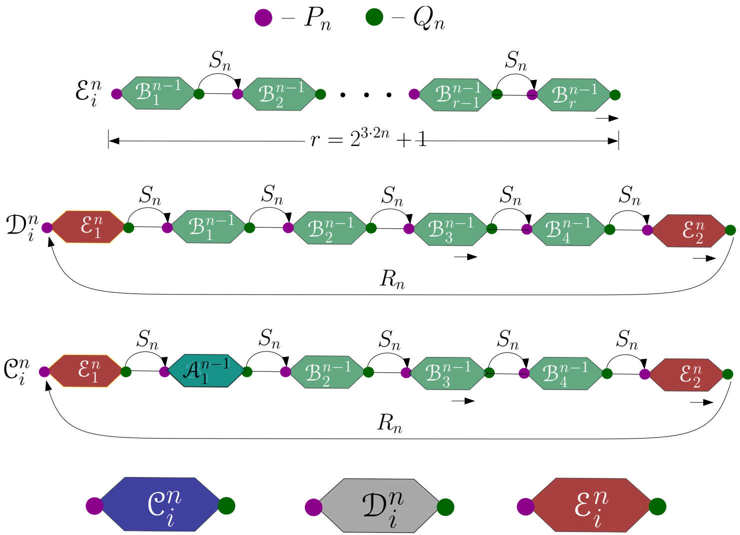


Construction of $\mathbf{A}^n$ and $\mathbf{B}^n$: Structures at level 1

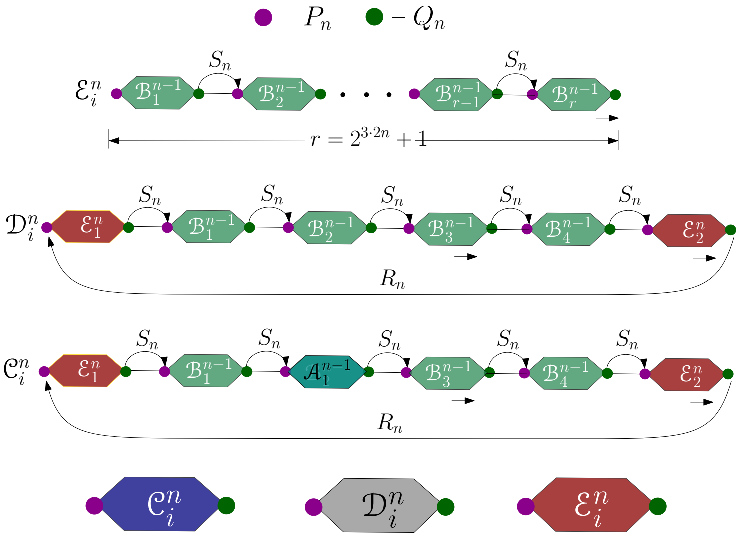


$\mathbf{A}^1$ = Class of all $\mathcal{A}_i^1$s; $\mathbf{B}^1$ = Class of all $\mathcal{B}_i^1$s

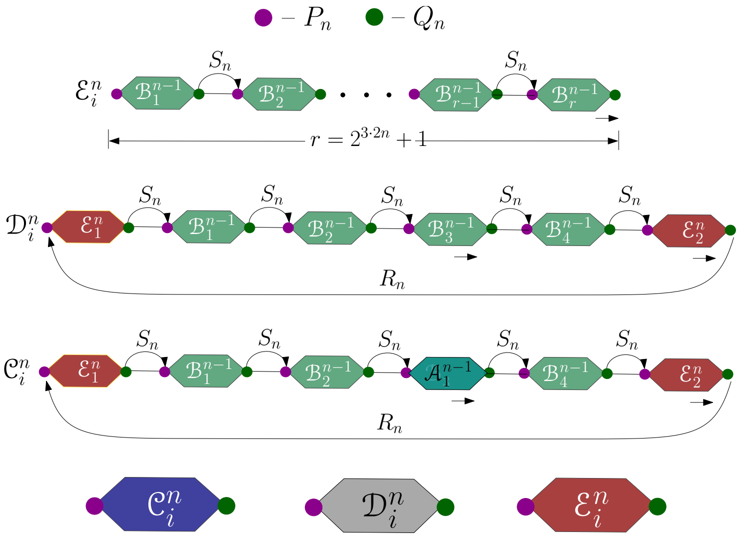
Construction of $\mathbf{A}^n$ and $\mathbf{B}^n$: Structures at level n



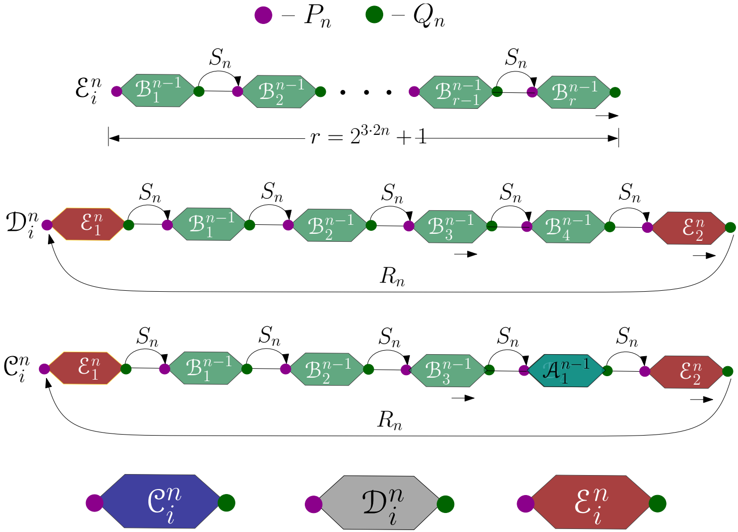
Construction of $\mathbf{A}^n$ and $\mathbf{B}^n$: Structures at level n



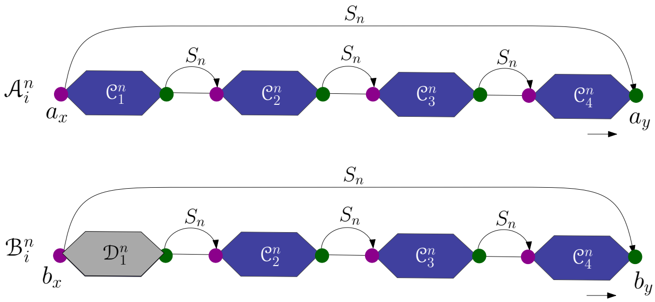
Construction of $\mathbf{A}^n$ and $\mathbf{B}^n$: Structures at level n



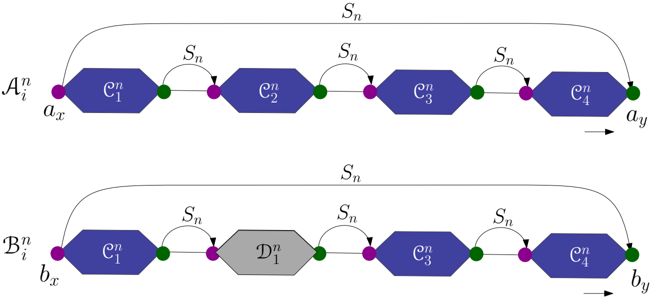
Construction of $\mathbf{A}^n$ and $\mathbf{B}^n$: Structures at level n



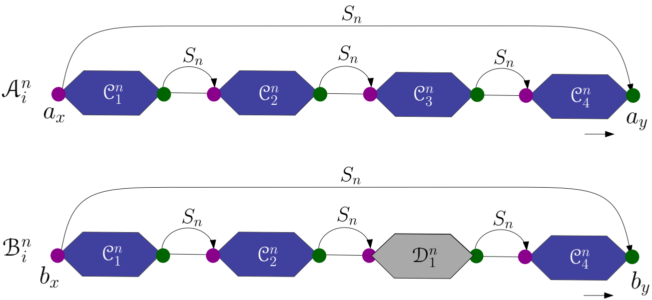
Construction of $\mathbf{A}^n$ and $\mathbf{B}^n$: Structures at level n



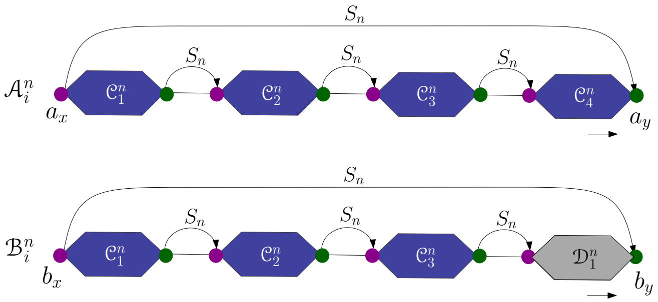
Construction of $\mathbf{A}^n$ and $\mathbf{B}^n$: Structures at level n



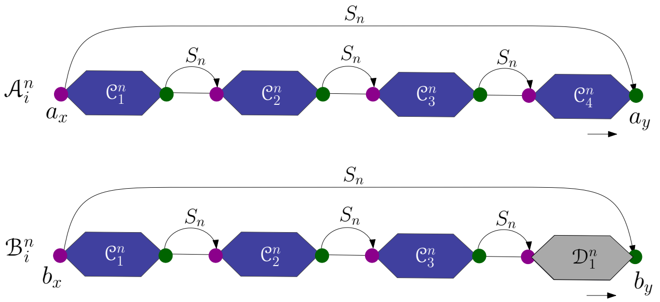
Construction of $\mathbf{A}^n$ and $\mathbf{B}^n$: Structures at level n



Construction of $\mathbf{A}^n$ and $\mathbf{B}^n$: Structures at level n



Construction of $\mathbf{A}^n$ and $\mathbf{B}^n$: Structures at level n



$\mathbf{A}^n$ = Class of all $\mathcal{A}_i^n$s; $\mathbf{B}^n$ = Class of all $\mathcal{B}_i^n$s

It can be shown that

$$(\mathcal{A}_i^n, a_x, a_y) \models \Psi_n(x, y)$$
$$(\mathcal{B}_i^n, b_x, b_y) \models \neg\Psi_n(x, y)$$

Conclusion

Future directions

- The formula Ψ_n is over a vocabulary σ_n that grows with n.
- Further, σ_n can be seen as the vocabulary of ordered vertex colored graphs with multiple edge relations.

Open question 1.

Is there a fixed (finite) vocabulary σ^* such that prefix classes fail to capture FO(σ^*) expressible hereditary properties?

Open question 2.

Do prefix classes fail to capture hereditary properties of undirected graphs (possibly vertex colored)? In particular, does Łoś-Tarski theorem continue to fail over these graphs?

Thank you!