Finite Model Theory
Unit 2

Dan Suciu

Spring 2018

599c: Finite Model Theory

Unit 2: Expressive Power of Logics on Finite Models

Resources

- Libkin, *Finite Model Theory*, Chapt. 3, 4, 11.
- Grädel, Kolaitis, Libkin, Marx, Spencer, Vardi, Venema, Weinstein: *Finite Model Theory and Its Applications*, Capt. 2 (Expressive Power of Logics on Finite Models).

Where Are We

- Classical model theory is concerned with *truth*, $\boldsymbol{D} \vDash \varphi$, and its implications.
- Finite model theory is concerned with:
  - Expressibility: which classes of finite structures can be expressed in a given logic.
  - Computability: connection between computational complexity and expressibility in that logic.
  - (Asymptotic) probabilities: study the probability (asymptotic or not) of a sentence.

Unit 2: Expressibility

- Ehrenfeuched-Fraisse Games

- Infinitary logics and Pebble Games

The Expressibility Problem

Given a property P, can it be expressed in a logic L?

- Example properties: $\mathrm{CONNECTIVITY}$, EVEN, $\mathrm{PLANARITY}$.
- Example logics: FO, SO, FO+fixpoint, Datalog.

Example 1: EVEN

Find a sentence φ s.t. $G \models \varphi$ iff G has an even number of nodes. In class

Example 1: EVEN

Find a sentence φ s.t. $G \models \varphi$ iff G has an even number of nodes. In class
Impossible! $\mu_n(\varphi) = 0$ when $n =$ odd, $\mu_n(\varphi) = 1$ when $n =$ even, violates 0/1-law.

Example 1: EVEN

Find a sentence φ s.t. $G \models \varphi$ iff G has an even number of nodes. In class
Impossible! $\mu_n(\varphi) = 0$ when $n =$ odd, $\mu_n(\varphi) = 1$ when $n =$ even, violates 0/1-law.

Find a sentence φ s.t. $G \models \varphi$ iff G has an even number of edges.

Example 1: EVEN

Find a sentence φ s.t. $G \vDash \varphi$ iff G has an even number of nodes. In class
Impossible! $\mu_n(\varphi) = 0$ when n = odd, $\mu_n(\varphi) = 1$ when n = even, violates 0/1-law.

Find a sentence φ s.t. $G \vDash \varphi$ iff G has an even number of edges.
The 0/1 law no longer helps.

Example 2: CONNECTED

$G = (V, E)$ is *connected*[1] if forall $a, b \in V$ there exists a path $a \rightarrow^* b$.

Find an FO sentence ψ s.t. $G \vDash \psi$ iff G is connected.

[1] The correct term is *strongly connected*.

Example 2: CONNECTED

$G = (V, E)$ is *connected*[1] if forall $a, b \in V$ there exists a path $a \to^* b$.

Find an FO sentence ψ s.t. $G \models \psi$ iff G is connected.
$\forall x \forall y E(x, y)$?

[1] The correct term is *strongly connected*.

Example 2: CONNECTED

$G = (V, E)$ is *connected*[1] if forall $a, b \in V$ there exists a path $a \to^* b$.

Find an FO sentence ψ s.t. $G \vDash \psi$ iff G is connected.
$\forall x \forall y E(x, y)$?
$\forall x \forall y \exists z (E(x, z) \wedge E(z, y))$?
...

[1] The correct term is *strongly connected*.

Example 2: CONNECTED

$G = (V, E)$ is *connected*[1] if forall $a, b \in V$ there exists a path $a \rightarrow^* b$.

Find an FO sentence ψ s.t. $G \models \psi$ iff G is connected.
$\forall x \forall y E(x, y)$?
$\forall x \forall y \exists z (E(x, z) \wedge E(z, y))$?
...
Impossible! Let's prove that.

[1] The correct term is *strongly connected*.

Example 2: CONNECTED

Suppose $\boldsymbol{G} \vDash \psi$ iff $\boldsymbol{G}$ is connected.

Example 2: CONNECTED

Suppose $\boldsymbol{G} \models \psi$ iff $\boldsymbol{G}$ is connected.

Fix two fresh constants c, d, and, forall $n \geq 1$, define:

$$\varphi_n = (\neg(\exists z_1 \cdots \exists z_n (E(c, z_1) \wedge E(z_1, z_2) \wedge \cdots \wedge E(z_n, d))))$$

It says "c, d are not connected by any path of length n".

Example 2: CONNECTED

Suppose $\boldsymbol{G} \vDash \psi$ iff $\boldsymbol{G}$ is connected.

Fix two fresh constants c, d, and, forall $n \geq 1$, define:

$$\varphi_n = (\neg(\exists z_1 \cdots \exists z_n (E(c, z_1) \wedge E(z_1, z_2) \wedge \cdots \wedge E(z_n, d))))$$

It says "c, d are not connected by any path of length n".

$\Sigma \stackrel{\text{def}}{=} \{\psi\} \cup \{\varphi_n \mid n \geq 1\}$ is finitely satisfiable why?

Example 2: CONNECTED

Suppose $\boldsymbol{G} \vDash \psi$ iff $\boldsymbol{G}$ is connected.

Fix two fresh constants c, d, and, forall $n \geq 1$, define:

$$\varphi_n = (\neg(\exists z_1 \cdots \exists z_n (E(c, z_1) \wedge E(z_1, z_2) \wedge \cdots \wedge E(z_n, d))))$$

It says "c, d are not connected by any path of length n".

$\Sigma \stackrel{\text{def}}{=} \{\psi\} \cup \{\varphi_n \mid n \geq 1\}$ is finitely satisfiable why?

By Compactness, Σ has a model $\boldsymbol{G}$

Example 2: CONNECTED

Suppose $\boldsymbol{G} \vDash \psi$ iff $\boldsymbol{G}$ is connected.

Fix two fresh constants c, d, and, forall $n \geq 1$, define:

$$\varphi_n = (\neg(\exists z_1 \cdots \exists z_n (E(c, z_1) \wedge E(z_1, z_2) \wedge \cdots \wedge E(z_n, d))))$$

It says "c, d are not connected by any path of length n".

$\Sigma \stackrel{\text{def}}{=} \{\psi\} \cup \{\varphi_n \mid n \geq 1\}$ is finitely satisfiable why?

By Compactness, Σ has a model $\boldsymbol{G}$

On one hand $\boldsymbol{G} \vDash \psi$ hence it is connected, on the other hand c, d are not connected in $\boldsymbol{G}$, contradiction.

Example 2: CONNECTED

Suppose $\boldsymbol{G} \models \psi$ iff $\boldsymbol{G}$ is connected.

Fix two fresh constants c, d, and, forall $n \geq 1$, define:

$$\varphi_n = (\neg(\exists z_1 \cdots \exists z_n (E(c, z_1) \wedge E(z_1, z_2) \wedge \cdots \wedge E(z_n, d))))$$

It says "c, d are not connected by any path of length n".

$\Sigma \stackrel{\text{def}}{=} \{\psi\} \cup \{\varphi_n \mid n \geq 1\}$ is finitely satisfiable why?

By Compactness, Σ has a model $\boldsymbol{G}$

On one hand $\boldsymbol{G} \models \psi$ hence it is connected, on the other hand c, d are not connected in $\boldsymbol{G}$, contradiction.

But is CONNECTIVITY expressible over *finite* graphs? This proof does not answer it.

Isomorphism

Assume a *relational vocabulary* $\sigma = (R_1, \dots, R_k, c_1, \dots, c_m)$ (no functions).
Fix $\boldsymbol{A} = (A, R_1^A, \dots, R_k^A, c_1^A, \dots, c_m^A)$, $\boldsymbol{B} = (B, R_1^B, \dots, R_k^B, c_1^B, \dots, c_m^B)$.

Definition

An *isomorphism* $f : \boldsymbol{A} \to \boldsymbol{B}$ is a bijection $A \to B$ such that:

- Forall $R \in \sigma$, $(a_1, \dots, a_k) \in R^A$ iff $(f(a_1), \dots, f(a_k)) \in R^B$.
- Forall $c \in \sigma$, $f(c^A) = c^B$.

We write $\boldsymbol{A} \simeq \boldsymbol{B}$ if there exists an isomorphism $\boldsymbol{A} \to \boldsymbol{B}$.

Remark: if $\boldsymbol{A} \simeq \boldsymbol{B}$ then for any sentence φ in a “reasonable” logics (like FO, or extensions), $\boldsymbol{A} \models \varphi$ iff $\boldsymbol{B} \models \varphi$.

Elementary Equivalence

Definition

$\boldsymbol{A}$ and $\boldsymbol{B}$ are *elementary equivalent* if forall φ, $\boldsymbol{A} \vDash \varphi$ iff $\boldsymbol{B} \vDash \varphi$.

We write $\boldsymbol{A} \equiv \boldsymbol{B}$.

Isomorphisms implies elementary equivalence: if $\boldsymbol{A} \simeq \boldsymbol{B}$ then $\boldsymbol{A} \equiv \boldsymbol{B}$.

Over the finite structures, the converse holds too: if $\boldsymbol{A} \equiv \boldsymbol{B}$, then $\boldsymbol{A} \simeq \boldsymbol{B}$.

We cannot find two finite graphs, one connected and one disconnected, that are elementary equivalent!

Partial Isomorphism

Fix a relational vocabulary σ: relations R_i, constants c_j.
Let $\boldsymbol{A}, \boldsymbol{B}$ be two σ-structures.

Definition

A *partial isomorphism* is a pair $\boldsymbol{a}, \boldsymbol{b}$, where $\boldsymbol{a} = (a_1, , \ldots, a_k) \in A^k$, $\boldsymbol{b} = (b_1, \ldots, b_k) \in B^k$ s.t. the substructures[a] $\boldsymbol{A}|_{\boldsymbol{a}}$, $\boldsymbol{B}|_{\boldsymbol{b}}$ are isomorphic via:

$$\forall i, a_i \mapsto b_i \qquad \forall j, c_j^A \mapsto c_j^B$$

[a] $\boldsymbol{A}|_{\boldsymbol{a}}$ consists of the universe $\{a_1, \ldots, a_k, c_1^A, \ldots, c_m^A\}$.

We write $\boldsymbol{a} \simeq \boldsymbol{b}$.

In other words:

- Forall i, j, $a_i = a_j$ iff $b_i = b_j$. (Equality is preserved.)
- Forall i, j, $a_i = c_j^A$ iff $b_i = c_j^B$. (Constants are preserved.)
- $(t_1, \ldots, t_n) \in R^A$ where each t_i is either some a_j or c_j^A, iff $(t'_1, \ldots, t'_n) \in R^B$ where t'_i is b_j or c_j^B respectively.

Ehrenfeucht-Fraisse Games

There are two players, spoiler and duplicator.
They play on two structures $\boldsymbol{A}, \boldsymbol{B}$ in k rounds, $i = 1, \ldots, k$.

Round i:

- Spoiler places his pebble i on an element $a_i \in A$ or $b_i \in B$.
- Duplicator places her pebble i on an element $b_i \in B$ or $a_i \in A$.

Let $\boldsymbol{a} = (a_1, \ldots, a_k)$, $\boldsymbol{b} = (b_1, \ldots, b_k)$ be the pebbles at the end of the game.
Duplicator wins if $\boldsymbol{a}, \boldsymbol{b}$ forms a partial isomorphism; otherwise Spoiler wins.

Definition

We write $\boldsymbol{A} \sim_k \boldsymbol{B}$ if the duplicator has a winning strategy for k rounds.

Ehrenfeucht-Fraisse Games

There are two players, spoiler and duplicator.
They play on two structures $\boldsymbol{A}, \boldsymbol{B}$ in k rounds, $i = 1, \ldots, k$.

Round i:

- Spoiler places his pebble i on an element $a_i \in A$ or $b_i \in B$.
- Duplicator places her pebble i on an element $b_i \in B$ or $a_i \in A$.

Let $\boldsymbol{a} = (a_1, \ldots, a_k)$, $\boldsymbol{b} = (b_1, \ldots, b_k)$ be the pebbles at the end of the game.
Duplicator wins if $\boldsymbol{a}, \boldsymbol{b}$ forms a partial isomorphism; otherwise Spoiler wins.

Definition

We write $\boldsymbol{A} \sim_k \boldsymbol{B}$ if the duplicator has a winning strategy for k rounds.

Ehrenfeucht-Fraisse Games

There are two players, spoiler and duplicator.
They play on two structures $\boldsymbol{A}, \boldsymbol{B}$ in k rounds, $i = 1, \ldots, k$.

Round i:

- Spoiler places his pebble i on an element $a_i \in A$ or $b_i \in B$.
- Duplicator places her pebble i on an element $b_i \in B$ or $a_i \in A$.

Let $\boldsymbol{a} = (a_1, \ldots, a_k)$, $\boldsymbol{b} = (b_1, \ldots, b_k)$ be the pebbles at the end of the game.
Duplicator wins if $\boldsymbol{a}, \boldsymbol{b}$ forms a partial isomorphism; otherwise Spoiler wins.

Definition

We write $\boldsymbol{A} \sim_k \boldsymbol{B}$ if the duplicator has a winning strategy for k rounds.

Ehrenfeucht-Fraisse Games

There are two players, spoiler and duplicator.
They play on two structures $\boldsymbol{A}, \boldsymbol{B}$ in k rounds, $i = 1, \ldots, k$.

Round i:

- Spoiler places his pebble i on an element $a_i \in A$ or $b_i \in B$.
- Duplicator places her pebble i on an element $b_i \in B$ or $a_i \in A$.

Let $\boldsymbol{a} = (a_1, \ldots, a_k)$, $\boldsymbol{b} = (b_1, \ldots, b_k)$ be the pebbles at the end of the game.
Duplicator wins if $\boldsymbol{a}, \boldsymbol{b}$ forms a partial isomorphism; otherwise Spoiler wins.

Definition

We write $\boldsymbol{A} \sim_k \boldsymbol{B}$ if the duplicator has a winning strategy for k rounds.

Ehrenfeucht-Fraisse Games

There are two players, spoiler and duplicator.
They play on two structures $\boldsymbol{A}, \boldsymbol{B}$ in k rounds, $i = 1, \ldots, k$.

Round i:

- Spoiler places his pebble i on an element $a_i \in A$ or $b_i \in B$.
- Duplicator places her pebble i on an element $b_i \in B$ or $a_i \in A$.

Let $\boldsymbol{a} = (a_1, \ldots, a_k)$, $\boldsymbol{b} = (b_1, \ldots, b_k)$ be the pebbles at the end of the game.
Duplicator wins if $\boldsymbol{a}, \boldsymbol{b}$ forms a partial isomorphism; otherwise Spoiler wins.

Definition

We write $\boldsymbol{A} \sim_k \boldsymbol{B}$ if the duplicator has a winning strategy for k rounds.

Ehrenfeucht-Fraisse Games: Main Result

The *quantifier rank* of a formula φ is defined inductively[2]:

$$\begin{aligned} qr(\boldsymbol{F}) =& qr(t_1 = t_2) = qr(R(t_1, \ldots, t_m)) = 0 \\ qr(\varphi \to \psi) =& \max(qr(\varphi), qr(\psi)) \\ qr(\forall x(\varphi)) =& 1 + qr(\varphi) \end{aligned}$$

$\mathrm{FO}[k] \stackrel{\text{def}}{=}$ FO restricted to formulas with $qr \leq k$.

Theorem (Ehrenfeucht-Fraisse)

$\boldsymbol{A} \equiv_k \boldsymbol{B}$ *(meaning: they agree on* $FO[k]$*) iff* $\boldsymbol{A} \sim_k \boldsymbol{B}$.

We will prove it later. First, let's see examples.

[2] The *number* of quantifiers can be exponentially larger than $qr(\varphi)$ why?

Ehrenfeucht-Fraisse Games: Main Result

The *quantifier rank* of a formula φ is defined inductively[2]:

$$\begin{aligned} qr(\boldsymbol{F}) =& qr(t_1 = t_2) = qr(R(t_1, \ldots, t_m)) = 0 \\ qr(\varphi \rightarrow \psi) =& \max(qr(\varphi), qr(\psi)) \\ qr(\forall x(\varphi)) =& 1 + qr(\varphi) \end{aligned}$$

$\text{FO}[k] \stackrel{\text{def}}{=}$ FO restricted to formulas with $qr \leq k$.

Theorem (Ehrenfeucht-Fraisse)

$\boldsymbol{A} \equiv_k \boldsymbol{B}$ *(meaning: they agree on* $FO[k]$*) iff* $\boldsymbol{A} \sim_k \boldsymbol{B}$.

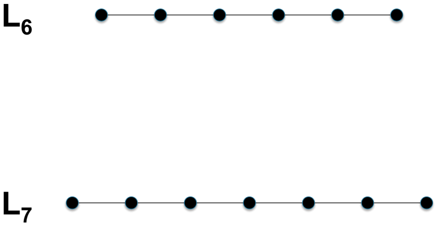
We will prove it later. First, let's see examples.

[2] The *number* of quantifiers can be exponentially larger than $qr(\varphi)$ why?

Ehrenfeucht-Fraisse on Total Orders

Let $L_k = (\{1, 2, \ldots, k\}, <)$.

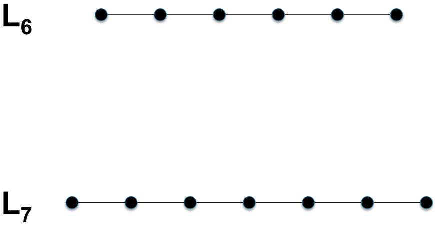
Play the Ehrenfeucht-Fraisse game on L_6, L_7 using $k = 2$ pebbles:



Ehrenfeucht-Fraisse on Total Orders

Let $L_k = (\{1, 2, \ldots, k\}, <)$.

Play the Ehrenfeucht-Fraisse game on L_6, L_7 using $k = 2$ pebbles: $L_6 \sim_2 L_7$

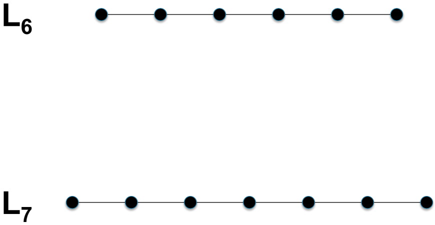


Ehrenfeucht-Fraisse on Total Orders

Let $L_k = (\{1, 2, \ldots, k\}, <)$.

Play the Ehrenfeucht-Fraisse game on L_6, L_7 using $k = 2$ pebbles: $L_6 \sim_2 L_7$

Play the Ehrenfeucht-Fraisse game on L_6, L_7 using $k = 3$ pebbles:

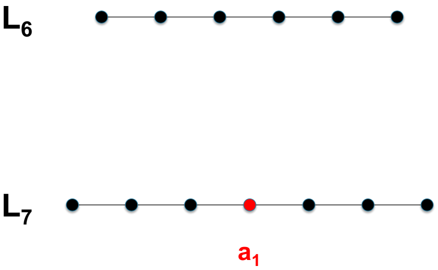


Ehrenfeucht-Fraisse on Total Orders

Let $L_k = (\{1, 2, \ldots, k\}, <)$.

Play the Ehrenfeucht-Fraisse game on L_6, L_7 using $k = 2$ pebbles: $L_6 \sim_2 L_7$

Play the Ehrenfeucht-Fraisse game on L_6, L_7 using $k = 3$ pebbles:

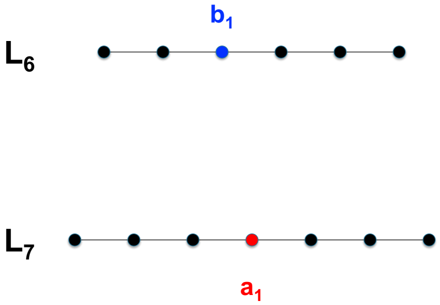


Ehrenfeucht-Fraisse on Total Orders

Let $L_k = (\{1, 2, \ldots, k\}, <)$.

Play the Ehrenfeucht-Fraisse game on L_6, L_7 using $k = 2$ pebbles: $L_6 \sim_2 L_7$

Play the Ehrenfeucht-Fraisse game on L_6, L_7 using $k = 3$ pebbles:

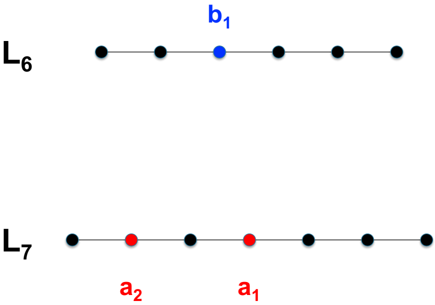


Ehrenfeucht-Fraisse on Total Orders

Let $L_k = (\{1, 2, \ldots, k\}, <)$.

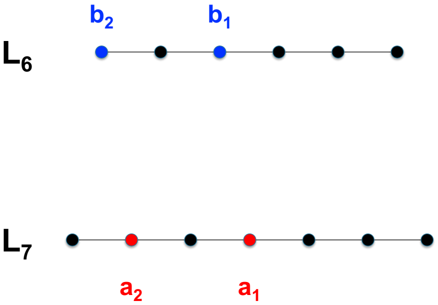
Play the Ehrenfeucht-Fraisse game on L_6, L_7 using $k = 2$ pebbles: $L_6 \sim_2 L_7$

Play the Ehrenfeucht-Fraisse game on L_6, L_7 using $k = 3$ pebbles:



Ehrenfeucht-Fraisse on Total Orders

Let $L_k = (\{1, 2, \ldots, k\}, <)$.
Play the Ehrenfeucht-Fraisse game on L_6, L_7 using $k = 2$ pebbles: $L_6 \sim_2 L_7$
Play the Ehrenfeucht-Fraisse game on L_6, L_7 using $k = 3$ pebbles:

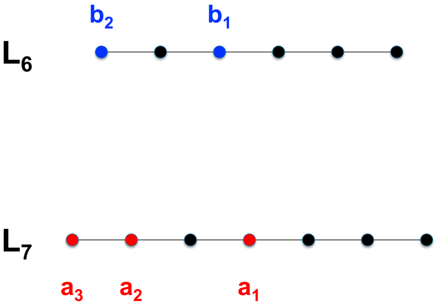


Ehrenfeucht-Fraisse on Total Orders

Let $L_k = (\{1, 2, \ldots, k\}, <)$.

Play the Ehrenfeucht-Fraisse game on L_6, L_7 using $k = 2$ pebbles: $L_6 \sim_2 L_7$

Play the Ehrenfeucht-Fraisse game on L_6, L_7 using $k = 3$ pebbles:

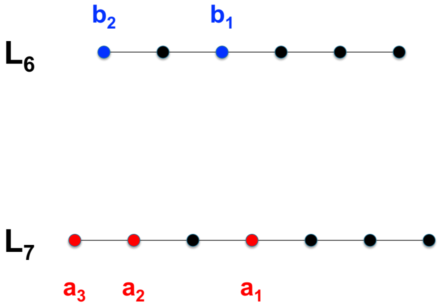


Ehrenfeucht-Fraisse on Total Orders

Let $L_k = (\{1, 2, \ldots, k\}, <)$.

Play the Ehrenfeucht-Fraisse game on L_6, L_7 using $k = 2$ pebbles: $L_6 \sim_2 L_7$

Play the Ehrenfeucht-Fraisse game on L_6, L_7 using $k = 3$ pebbles: $L_6 \not\sim_3 L_7$



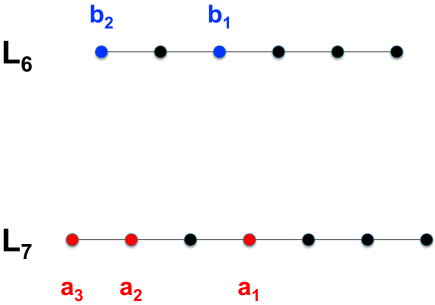
Ehrenfeucht-Fraisse on Total Orders

Let $L_k = (\{1, 2, \ldots, k\}, <)$.

Play the Ehrenfeucht-Fraisse game on L_6, L_7 using $k = 2$ pebbles: $L_6 \sim_2 L_7$

Play the Ehrenfeucht-Fraisse game on L_6, L_7 using $k = 3$ pebbles: $L_6 \not\sim_3 L_7$

Find $\varphi \in FO[3]$ s.t. $L_6 \vDash \varphi, L_7 \nvDash \varphi$



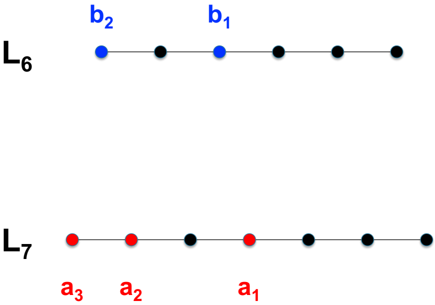
Ehrenfeucht-Fraisse on Total Orders

Let $L_k = (\{1, 2, \ldots, k\}, <)$.

Play the Ehrenfeucht-Fraisse game on L_6, L_7 using $k = 2$ pebbles: $L_6 \sim_2 L_7$

Play the Ehrenfeucht-Fraisse game on L_6, L_7 using $k = 3$ pebbles: $L_6 \not\sim_3 L_7$

Find $\varphi \in FO[3]$ s.t. $L_6 \models \varphi, L_7 \not\models \varphi$



$$\begin{aligned}
&\forall x_1 \forall x_2 (x_2 < x_1 \rightarrow \quad // L_6^{<x_1} \text{ is small}\\
&\quad (\forall x_3 \neg (x_3 < x_2)\\
&\quad \vee \forall x_3 \neg (x_2 < x_3 < x_1)))\\
&\vee \forall x_2 (x_2 > x_1 \rightarrow \quad // L_6^{>x_1} \text{ is small}\\
&\quad (\forall x_3 \neg (x_3 > x_2)\\
&\quad \vee \forall x_3 \neg (x_1 < x_3 < x_2)))
\end{aligned}$$

Ehrenfeucht-Fraisse on Total Orders

Let $L_m = (\{1, 2, \ldots, m\}, <)$.

$$L_m^{<a} \stackrel{\text{def}}{=} \{x \in L_m \mid x < a\} \qquad L_m^{>a} \stackrel{\text{def}}{=} \{x \in L_m \mid x < a\}$$

Lemma

If $L_m^{<a} \sim_k L_n^{<b}$ and $L_m^{>a} \sim_k L_n^{>b}$ (duplicator wins), then $L_m \sim_k L_n$.

Proof.

- If spoiler places pebble in $L_m^{<a}$ then duplicator answers in $L_n^{<b}$.
- If spoiler places pebble in $L_m^{>a}$ then duplicator answers in $L_n^{>b}$.
- If spoiler places pebble on a then duplicator places pebble on b.
- If spoiler plays in the other structure, duplicator answers similarly.

If $L_m^{<a}|_{\mathbf{c}} \simeq L_n^{<b}|_{\mathbf{d}}$ and $L_m^{>a}|_{\mathbf{c}} \simeq L_n^{>b}|_{\mathbf{d}}$ (partial isomorphisms), then $\mathbf{c} \simeq \mathbf{d}$

Ehrenfeucht-Fraisse on Total Orders

Let $L_m = (\{1, 2, \ldots, m\}, <)$.

$$L_m^{<a} \stackrel{\text{def}}{=} \{x \in L_m \mid x < a\} \qquad\qquad L_m^{>a} \stackrel{\text{def}}{=} \{x \in L_m \mid x < a\}$$

Lemma

If $L_m^{<a} \sim_k L_n^{<b}$ and $L_m^{>a} \sim_k L_n^{>b}$ (duplicator wins), then $L_m \sim_k L_n$.

Proof.

- If spoiler places pebble in $L_m^{<a}$ then duplicator answers in $L_n^{<b}$.
- If spoiler places pebble in $L_m^{>a}$ then duplicator answers in $L_n^{>b}$.
- If spoiler places pebble on a then duplicator places pebble on b.
- If spoiler plays in the other structure, duplicator answers similarly.

If $L_m^{<a}|_{\mathbf{c}} \simeq L_n^{<b}|_{\mathbf{d}}$ and $L_m^{>a}|_{\mathbf{c}} \simeq L_n^{>b}|_{\mathbf{d}}$ (partial isomorphisms), then $\mathbf{c} \simeq \mathbf{d}$

Ehrenfeucht-Fraisse on Total Orders

Let $L_m = (\{1, 2, \ldots, m\}, <)$.

$$L_m^{<a} \stackrel{\text{def}}{=} \{x \in L_m \mid x < a\} \qquad L_m^{>a} \stackrel{\text{def}}{=} \{x \in L_m \mid x < a\}$$

Lemma

If $L_m^{<a} \sim_k L_n^{<b}$ and $L_m^{>a} \sim_k L_n^{>b}$ (duplicator wins), then $L_m \sim_k L_n$.

Proof.

- If spoiler places pebble in $L_m^{<a}$ then duplicator answers in $L_n^{<b}$.
- If spoiler places pebble in $L_m^{>a}$ then duplicator answers in $L_n^{>b}$.
- If spoiler places pebble on a then duplicator places pebble on b.
- If spoiler plays in the other structure, duplicator answers similarly.

If $L_m^{<a}|_{\mathbf{c}} \simeq L_n^{<b}|_{\mathbf{d}}$ and $L_m^{>a}|_{\mathbf{c}} \simeq L_n^{>b}|_{\mathbf{d}}$ (partial isomorphisms), then $\mathbf{c} \simeq \mathbf{d}$

Ehrenfeucht-Fraisse on Total Orders

Let $L_m = (\{1, 2, \ldots, m\}, <)$.

$$L_m^{<a} \stackrel{\text{def}}{=} \{x \in L_m \mid x < a\} \qquad L_m^{>a} \stackrel{\text{def}}{=} \{x \in L_m \mid x < a\}$$

Lemma

If $L_m^{<a} \sim_k L_n^{<b}$ and $L_m^{>a} \sim_k L_n^{>b}$ (duplicator wins), then $L_m \sim_k L_n$.

Proof.

- If spoiler places pebble in $L_m^{<a}$ then duplicator answers in $L_n^{<b}$.
- If spoiler places pebble in $L_m^{>a}$ then duplicator answers in $L_n^{>b}$.
- If spoiler places pebble on a then duplicator places pebble on b.
- If spoiler plays in the other structure, duplicator answers similarly.

If $L_m^{<a}|_{\mathbf{c}} \simeq L_n^{<b}|_{\mathbf{d}}$ and $L_m^{>a}|_{\mathbf{c}} \simeq L_n^{>b}|_{\mathbf{d}}$ (partial isomorphisms), then $\mathbf{c} \simeq \mathbf{d}$

Ehrenfeucht-Fraisse on Total Orders

Let $L_m = (\{1, 2, \ldots, m\}, <)$.

$$L_m^{<a} \stackrel{\text{def}}{=} \{x \in L_m \mid x < a\} \qquad L_m^{>a} \stackrel{\text{def}}{=} \{x \in L_m \mid x < a\}$$

Lemma

If $L_m^{<a} \sim_k L_n^{<b}$ and $L_m^{>a} \sim_k L_n^{>b}$ (duplicator wins), then $L_m \sim_k L_n$.

Proof.

- If spoiler places pebble in $L_m^{<a}$ then duplicator answers in $L_n^{<b}$.
- If spoiler places pebble in $L_m^{>a}$ then duplicator answers in $L_n^{>b}$.
- If spoiler places pebble on a then duplicator places pebble on b.
- If spoiler plays in the other structure, duplicator answers similarly.

If $L_m^{<a}|_{\mathbf{c}} \simeq L_n^{<b}|_{\mathbf{d}}$ and $L_m^{>a}|_{\mathbf{c}} \simeq L_n^{>b}|_{\mathbf{d}}$ (partial isomorphisms), then $\mathbf{c} \simeq \mathbf{d}$

Ehrenfeucht-Fraisse on Total Orders

Theorem

Let m, n, k be positive integers. The following are equivalent:

- $L_m \sim_k L_n$
- $m = n$ *or both* $m \geq 2^k - 1$ *and* $n \geq 2^k - 1$

Ehrenfeucht-Fraisse on Total Orders

Theorem

Let m, n, k be positive integers. The following are equivalent:

- $L_m \sim_k L_n$
- $m = n$ *or both* $m \geq 2^k - 1$ *and* $n \geq 2^k - 1$

Proof. If $m, n \geq 2^k - 1$, duplicator has winning strategy. Spoiler plays $a \in L_m$.

- Case 1: $|L_m^{<a}| < 2^{k-1} - 1$ (What do we do?)

Ehrenfeucht-Fraisse on Total Orders

Theorem

Let m, n, k be positive integers. The following are equivalent:

- $L_m \sim_k L_n$
- $m = n$ *or both* $m \geq 2^k - 1$ *and* $n \geq 2^k - 1$

Proof. If $m, n \geq 2^k - 1$, duplicator has winning strategy. Spoiler plays $a \in L_m$.

- Case 1: $|L_m^{<a}| < 2^{k-1} - 1$ (What do we do?)
  Duplicator chooses b s.t. $L_m^{<a} \simeq L_n^{<b}$ (i.e. isomorphic). Then: $|L_m^{>a}|, |L_n^{>b}| > 2^{k-1} - 1$ (why?)

Ehrenfeucht-Fraisse on Total Orders

Theorem

Let m, n, k be positive integers. The following are equivalent:

- $L_m \sim_k L_n$
- $m = n$ *or both* $m \geq 2^k - 1$ *and* $n \geq 2^k - 1$

Proof. If $m, n \geq 2^k - 1$, duplicator has winning strategy. Spoiler plays $a \in L_m$.

- Case 1: $|L_m^{<a}| < 2^{k-1} - 1$ (What do we do?)
  Duplicator chooses b s.t. $L_m^{<a} \simeq L_n^{<b}$ (i.e. isomorphic). Then: $|L_m^{>a}|, |L_n^{>b}| > 2^{k-1} - 1$ (why?), $L_m^{>a} \sim_{k-1} L_n^{>b}$ (why?)

Ehrenfeucht-Fraisse on Total Orders

Theorem

Let m, n, k be positive integers. The following are equivalent:

- $L_m \sim_k L_n$
- $m = n$ *or both* $m \geq 2^k - 1$ *and* $n \geq 2^k - 1$

Proof. If $m, n \geq 2^k - 1$, duplicator has winning strategy. Spoiler plays $a \in L_m$.

- Case 1: $|L_m^{<a}| < 2^{k-1} - 1$ (What do we do?)
  Duplicator chooses b s.t. $L_m^{<a} \simeq L_n^{<b}$ (i.e. isomorphic). Then: $|L_m^{>a}|, |L_n^{>b}| > 2^{k-1} - 1$ (why?), $L_m^{>a} \sim_{k-1} L_n^{>b}$ (why?), $L_m \sim_k L_n$ (lemma).

Ehrenfeucht-Fraisse on Total Orders

Theorem

Let m, n, k be positive integers. The following are equivalent:

- $L_m \sim_k L_n$
- $m = n$ *or both* $m \geq 2^k - 1$ *and* $n \geq 2^k - 1$

Proof. If $m, n \geq 2^k - 1$, duplicator has winning strategy. Spoiler plays $a \in L_m$.

- Case 1: $|L_m^{<a}| < 2^{k-1} - 1$ (What do we do?)
  Duplicator chooses b s.t. $L_m^{<a} \simeq L_n^{<b}$ (i.e. isomorphic). Then:
  $|L_m^{>a}|, |L_n^{>b}| > 2^{k-1} - 1$ (why?), $L_m^{>a} \sim_{k-1} L_n^{>b}$ (why?), $L_m \sim_k L_n$ (lemma).
- Case 2: $|L_m^{>a}| < 2^{k-1} - 1$ Symmetric:
  Duplicator chooses b s.t. $L_m^{>a} \simeq L_n^{>b}$ (i.e. isomorphic). Then:
  $|L_m^{<a}|, |L_n^{<b}| > 2^{k-1} - 1$, $L_m^{<a} \sim_{k-1} L_n^{<b}$, hence $L_m \sim_k L_n$ (lemma).

Ehrenfeucht-Fraisse on Total Orders

Theorem

Let m, n, k be positive integers. The following are equivalent:

- $L_m \sim_k L_n$
- $m = n$ *or both* $m \geq 2^k - 1$ *and* $n \geq 2^k - 1$

Proof. If $m, n \geq 2^k - 1$, duplicator has winning strategy. Spoiler plays $a \in L_m$.

- Case 1: $|L_m^{<a}| < 2^{k-1} - 1$ (What do we do?)
  Duplicator chooses b s.t. $L_m^{<a} \simeq L_n^{<b}$ (i.e. isomorphic). Then: $|L_m^{>a}|, |L_n^{>b}| > 2^{k-1} - 1$ (why?), $L_m^{>a} \sim_{k-1} L_n^{>b}$ (why?), $L_m \sim_k L_n$ (lemma).
- Case 2: $|L_m^{>a}| < 2^{k-1} - 1$ Symmetric:
  Duplicator chooses b s.t. $L_m^{>a} \simeq L_n^{>b}$ (i.e. isomorphic). Then: $|L_m^{<a}|, |L_n^{<b}| > 2^{k-1} - 1$, $L_m^{<a} \sim_{k-1} L_n^{<b}$, hence $L_m \sim_k L_n$ (lemma).
- Case 3: both $|L_m^{<a}|, |L_m^{>a}| \geq 2^{k-1} - 1$ (Is this possible?)

Ehrenfeucht-Fraisse on Total Orders

Theorem

Let m, n, k be positive integers. The following are equivalent:

- $L_m \sim_k L_n$
- $m = n$ *or both* $m \geq 2^k - 1$ *and* $n \geq 2^k - 1$

Proof. If $m, n \geq 2^k - 1$, duplicator has winning strategy. Spoiler plays $a \in L_m$.

- Case 1: $|L_m^{<a}| < 2^{k-1} - 1$ (What do we do?)
  Duplicator chooses b s.t. $L_m^{<a} \simeq L_n^{<b}$ (i.e. isomorphic). Then:
  $|L_m^{>a}|, |L_n^{>b}| > 2^{k-1} - 1$ (why?), $L_m^{>a} \sim_{k-1} L_n^{>b}$ (why?), $L_m \sim_k L_n$ (lemma).
- Case 2: $|L_m^{>a}| < 2^{k-1} - 1$ Symmetric:
  Duplicator chooses b s.t. $L_m^{>a} \simeq L_n^{>b}$ (i.e. isomorphic). Then:
  $|L_m^{<a}|, |L_n^{<b}| > 2^{k-1} - 1$, $L_m^{<a} \sim_{k-1} L_n^{<b}$, hence $L_m \sim_k L_n$ (lemma).
- Case 3: both $|L_m^{<a}|, |L_m^{>a}| \geq 2^{k-1} - 1$ (Is this possible?)
  Duplicator chooses any b s.t. $|L_n^{<b}|, |L_n^{>b}| \geq 2^{k-1} - 1$. Then:
  $|L_m^{<a}|, |L_n^{<b}|, |L_m^{>a}|, |L_n^{>b}| \geq 2^{k-1} - 1$;

Ehrenfeucht-Fraisse on Total Orders

Theorem

Let m, n, k be positive integers. The following are equivalent:

- $L_m \sim_k L_n$
- $m = n$ *or both* $m \geq 2^k - 1$ *and* $n \geq 2^k - 1$

Proof. If $m, n \geq 2^k - 1$, duplicator has winning strategy. Spoiler plays $a \in L_m$.

- Case 1: $|L_m^{<a}| < 2^{k-1} - 1$ (What do we do?)
  Duplicator chooses b s.t. $L_m^{<a} \simeq L_n^{<b}$ (i.e. isomorphic). Then: $|L_m^{>a}|, |L_n^{>b}| > 2^{k-1} - 1$ (why?), $L_m^{>a} \sim_{k-1} L_n^{>b}$ (why?), $L_m \sim_k L_n$ (lemma).
- Case 2: $|L_m^{>a}| < 2^{k-1} - 1$ Symmetric:
  Duplicator chooses b s.t. $L_m^{>a} \simeq L_n^{>b}$ (i.e. isomorphic). Then: $|L_m^{<a}|, |L_n^{<b}| > 2^{k-1} - 1$, $L_m^{<a} \sim_{k-1} L_n^{<b}$, hence $L_m \sim_k L_n$ (lemma).
- Case 3: both $|L_m^{<a}|, |L_m^{>a}| \geq 2^{k-1} - 1$ (Is this possible?)
  Duplicator chooses any b s.t. $|L_n^{<b}|, |L_n^{>b}| \geq 2^{k-1} - 1$. Then: $|L_m^{<a}|, |L_n^{<b}|, |L_m^{>a}|, |L_n^{>b}| \geq 2^{k-1} - 1$; $L_m^{<a} \sim_{k-1} L_n^{<b}$, $L_m^{>a} \sim_{k-1} L_n^{>b}$;

Ehrenfeucht-Fraisse on Total Orders

Theorem

Let m, n, k be positive integers. The following are equivalent:

- $L_m \sim_k L_n$
- $m = n$ *or both* $m \geq 2^k - 1$ *and* $n \geq 2^k - 1$

Proof. If $m, n \geq 2^k - 1$, duplicator has winning strategy. Spoiler plays $a \in L_m$.

- Case 1: $|L_m^{<a}| < 2^{k-1} - 1$ (What do we do?)
  Duplicator chooses b s.t. $L_m^{<a} \simeq L_n^{<b}$ (i.e. isomorphic). Then:
  $|L_m^{>a}|, |L_n^{>b}| > 2^{k-1} - 1$ (why?), $L_m^{>a} \sim_{k-1} L_n^{>b}$ (why?), $L_m \sim_k L_n$ (lemma).
- Case 2: $|L_m^{>a}| < 2^{k-1} - 1$ Symmetric:
  Duplicator chooses b s.t. $L_m^{>a} \simeq L_n^{>b}$ (i.e. isomorphic). Then:
  $|L_m^{<a}|, |L_n^{<b}| > 2^{k-1} - 1$, $L_m^{<a} \sim_{k-1} L_n^{<b}$, hence $L_m \sim_k L_n$ (lemma).
- Case 3: both $|L_m^{<a}|, |L_m^{>a}| \geq 2^{k-1} - 1$ (Is this possible?)
  Duplicator chooses any b s.t. $|L_n^{<b}|, |L_n^{>b}| \geq 2^{k-1} - 1$. Then:
  $|L_m^{<a}|, |L_n^{<b}|, |L_m^{>a}|, |L_n^{>b}| \geq 2^{k-1} - 1$; $L_m^{<a} \sim_{k-1} L_n^{<b}$, $L_m^{>a} \sim_{k-1} L_n^{>b}$; $L_m \sim_k L_n$.

Ehrenfeucht-Fraisse on Total Orders

Corollary

EVEN *is not expressible in FO over total orders.*

More precisely, there is no sentence φ s.t. $(L_n, <) \models \varphi$ iff n is even.

Ehrenfeucht-Fraisse on Total Orders

Corollary

EVEN *is not expressible in FO over total orders.*

More precisely, there is no sentence φ s.t. $(L_n, <) \models \varphi$ iff n is even.

0/1 Law is not useful here why not?

Ehrenfeucht-Fraisse on Total Orders

Corollary

EVEN *is not expressible in FO over total orders.*

More precisely, there is no sentence φ s.t. $(L_n, <) \models \varphi$ iff n is even.

0/1 Law is not useful here why not?

Instead we prove it using EF-games on total orders. how?

Ehrenfeucht-Fraisse on Total Orders

Corollary

EVEN *is not expressible in FO over total orders.*

More precisely, there is no sentence φ s.t. $(L_n, <) \models \varphi$ iff n is even.

0/1 Law is not useful here why not?

Instead we prove it using EF-games on total orders. how?

Let φ be such a sentence, $k \stackrel{\text{def}}{=} qr(\varphi)$. Choose $n \geq 2^k - 1$.

Ehrenfeucht-Fraisse on Total Orders

Corollary

EVEN *is not expressible in FO over total orders.*

More precisely, there is no sentence φ s.t. $(L_n, <) \vDash \varphi$ iff n is even.

0/1 Law is not useful here why not?

Instead we prove it using EF-games on total orders. how?

Let φ be such a sentence, $k \stackrel{\text{def}}{=} qr(\varphi)$. Choose $n \geq 2^k - 1$.

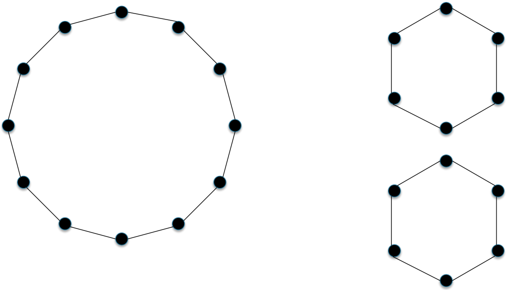
Then $L_n \sim_k L_{n+1}$ hence $L_n \vDash \varphi$ iff $L_{n+1} \vDash \varphi$. Contradiction.

Discussion

- Prove the converse at home: if $m < 2^k - 1 \leq n$ then duplicator has a winning strategy.
- According to the EF theorem, if $m < 2^k - 1 \leq n$ then there exists a sentence $\varphi \in FO[k]$ s.t. $L_m \vDash \varphi$ and $L_n \nvDash \varphi$. What is φ?
- The *Ehrenfeucht-Fraisse method* for showing inexpressibility in FO is this. For each $k > 0$ construct two structures $\boldsymbol{A}_k, \boldsymbol{B}_k$ then:
  - Prove: $\boldsymbol{A}_k \sim_k \boldsymbol{B}_k$.
  - Prove: $\boldsymbol{A}_k$ has the property, $\boldsymbol{B}_k$ does not.
- Proving $\sim_k$: difficult in general. A sufficient condition: Hanf's lemma.

Example: CONNECTIVITY

Prove that duplicator has winning strategy with $k = 3$ pebbles (in class).



$\mathbf{C_{12}}$ $\quad$ $\mathbf{C_6 \cup C_6}$

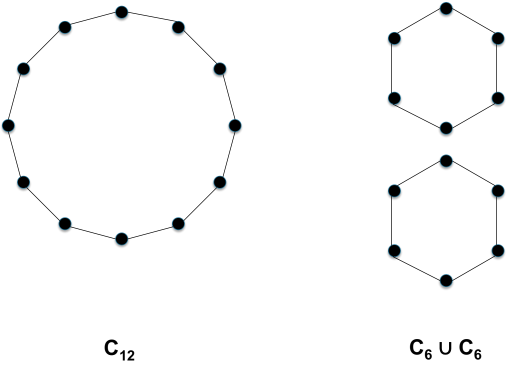
Homework: spoiler has a winning strategy with $k = 4$ pebbles.

Describing and proving a winning strategy in general seems difficult.

Hanf's lemma gives a sufficient condition for a winning strategy.

Example: CONNECTIVITY

Prove that duplicator has winning strategy with $k = 3$ pebbles (in class).



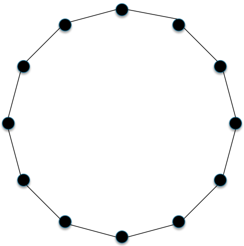
Homework: spoiler has a winning strategy with $k = 4$ pebbles.

Describing and proving a winning strategy in general seems difficult.

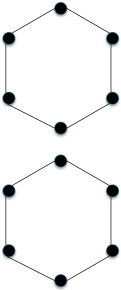
Hanf's lemma gives a sufficient condition for a winning strategy.

Example: CONNECTIVITY

Prove that duplicator has winning strategy with $k = 3$ pebbles (in class).



$\mathbf{C_{12}}$



$\mathbf{C_6 \cup C_6}$

Homework: spoiler has a winning strategy with $k = 4$ pebbles.

Describing and proving a winning strategy in general seems difficult.

Hanf's lemma gives a sufficient condition for a winning strategy.

The Gaifman Graph

Let $\boldsymbol{A} = (A, R_1^A, R_2^A, \ldots, R_m^A, c_1^A, \ldots, c_s^A)$ be a structure.

Definition

The Gaifman graph is $G(\boldsymbol{A}) = (A, E_A)$ where the edges are pairs (c, d) s.t. there exists a tuple $(\ldots, c, \ldots, d, \ldots) \in R_i^A$ or $(\ldots, d, \ldots, c, \ldots) \in R_i^A$.

The Gaifman graph of a graph is obtained by forgetting the directions.

Definition

For $a \in A$ and $d \geq 0$, the d-neighborhood is
$N(a, d) \stackrel{\text{def}}{=} \{b \in A \mid d(a, b) \leq d\} \cup \{c_1^A, \ldots, c_s^A\}$.
The d-type of a is the isomorphism type of the substructure generated by $N(a, d)$ plus the constant a.

Definition

$\boldsymbol{A}, \boldsymbol{B}$ are called d-equivalent if for each d-type they have the same number of elements of that type.

Hanf's Lemma

Fagin, Stockmeyer, Vardi proved the following, building on earlier work by Hanf:

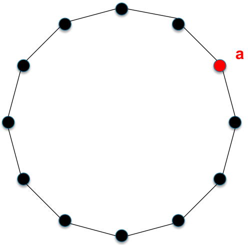
Theorem

Let $d \geq 3^{k-1} - 1$. If $\boldsymbol{A}, \boldsymbol{B}$ are d-equivalent, then $\boldsymbol{A} \sim_k \boldsymbol{B}$.

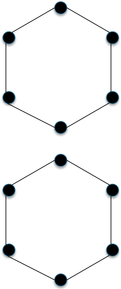
Note 1: Kolaitis requires $d \geq 3^{k-1}$ but defines "distance" s.t. $d(a, a) = 1$.
Note 2: this is only a sufficient condition, not necessary.
The proof exhibits a winning strategy for the duplicator. We omit the proof.

Example: CONNECTIVITY (continued)

Fix $k = 2$ and $d = 2 (= 3^{k-1} - 1)$.
What is $N(a, d)$?



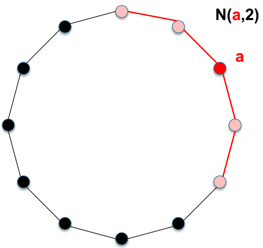
C_{12}



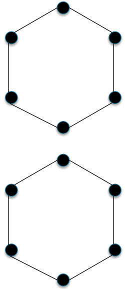
$C_6 \cup C_6$

Example: CONNECTIVITY (continued)

Fix $k = 2$ and $d = 2(= 3^{k-1} - 1)$.
What is $N(a, d)$?



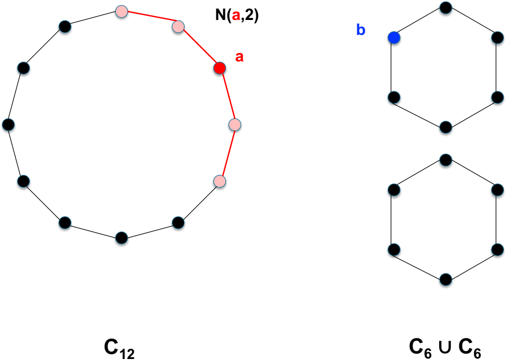
C_{12}



$C_6 \cup C_6$

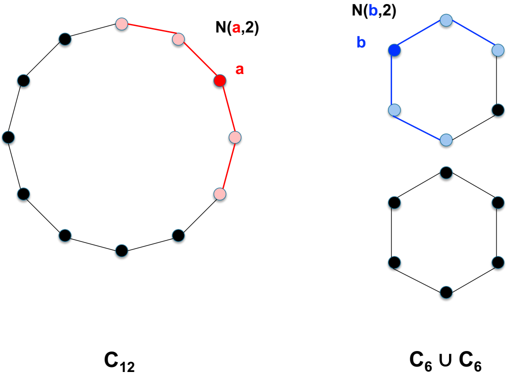
Example: CONNECTIVITY (continued)

Fix $k = 2$ and $d = 2 (= 3^{k-1} - 1)$.
What is $N(a, d)$? What is $N(b, d)$?



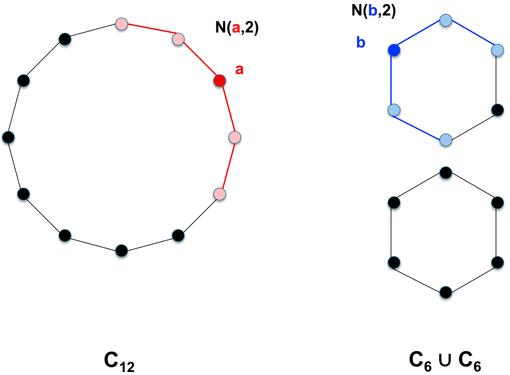
Example: CONNECTIVITY (continued)

Fix $k = 2$ and $d = 2 (= 3^{k-1} - 1)$.
What is $N(a, d)$? What is $N(b, d)$?



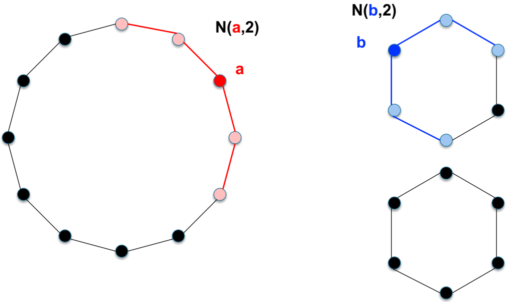
Example: CONNECTIVITY (continued)

Fix $k = 2$ and $d = 2(= 3^{k-1} - 1)$.
What is $N(a, d)$? What is $N(b, d)$?
What is their *type*?



Example: CONNECTIVITY (continued)

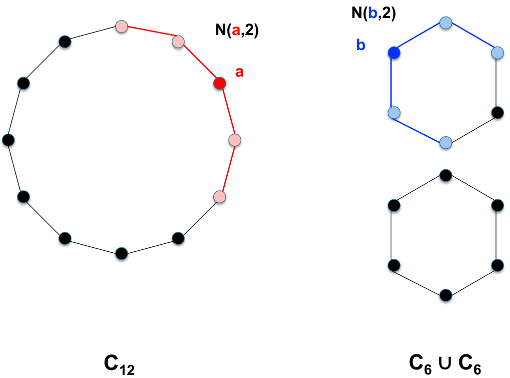
Fix $k = 2$ and $d = 2(= 3^{k-1} - 1)$.
What is $N(a, d)$? What is $N(b, d)$?
What is their *type*? Structures of the form $x - x - * - x - x$



C_{12} $\quad$ $C_6 \cup C_6$

Example: CONNECTIVITY (continued)

Fix $k = 2$ and $d = 2(= 3^{k-1} - 1)$.
What is $N(a, d)$? What is $N(b, d)$?
What is their *type*? Structures of the form $x - x - * - x - x$
How many elements of this type are there in each structure?



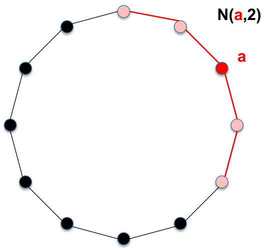
Example: CONNECTIVITY (continued)

Fix $k = 2$ and $d = 2(= 3^{k-1} - 1)$.

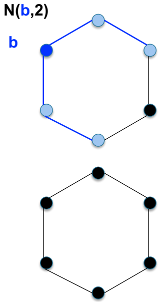
What is $N(a, d)$? What is $N(b, d)$?

What is their *type*? Structures of the form $x - x - * - x - x$

How many elements of this type are there in each structure? 12 in each



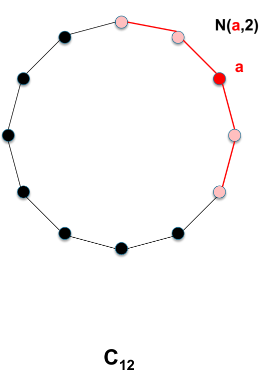
C_{12}

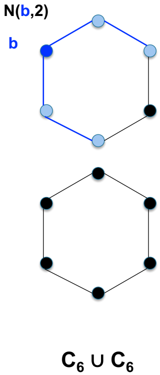


$C_6 \cup C_6$

Example: CONNECTIVITY (continued)

Fix $k = 2$ and $d = 2(= 3^{k-1} - 1)$.
What is $N(a, d)$? What is $N(b, d)$?
What is their *type*? Structures of the form $x - x - * - x - x$
How many elements of this type are there in each structure? 12 in each
Therefore duplicator has winning strategy with $k = 2$ pebbles.





Example: CONNECTIVITY (continued)

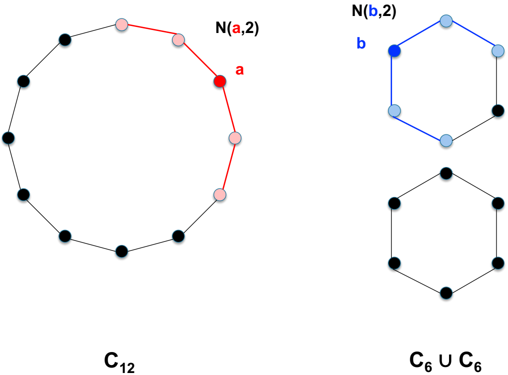
Fix $k = 2$ and $d = 2 (= 3^{k-1} - 1)$.

What is $N(a, d)$? What is $N(b, d)$?

What is their *type*? Structures of the form $x - x - * - x - x$

How many elements of this type are there in each structure? 12 in each

Therefore duplicator has winning strategy with $k = 2$ pebbles.



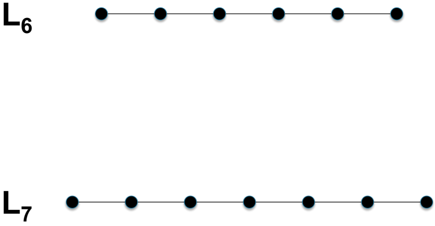
Prove: for every k there exists n s.t. duplicator has a winning strategy on C_{2n} and $C_n \cup C_n$

Example: CONNECTIVITY (continued)

A much simpler proof using an FO-reduction.

Assume φ expresses connectivity of a graph $G = (V, E)$. Then we write a sentence ψ s.t. $(L_n, <) \vDash \psi$ iff $(L_{n+1}, <) \nvDash \psi$.

In $(L_m, <)$ define: $E \stackrel{\text{def}}{=} \{(i, i+2) \mid 1 \le i \le m-2\} \cup \{(m-1, 1), (m, 2)\}$ how?.



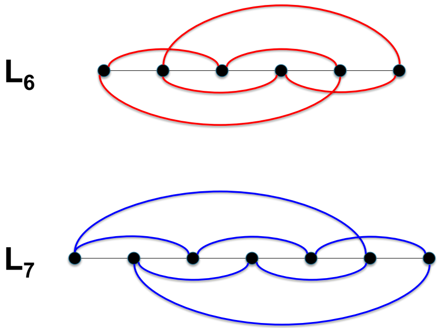
Example: CONNECTIVITY (continued)

A much simpler proof using an FO-reduction.

Assume φ expresses connectivity of a graph $G = (V, E)$. Then we write a sentence ψ s.t. $(L_n, <) \models \psi$ iff $(L_{n+1}, <) \not\models \psi$.

In $(L_m, <)$ define: $E \stackrel{\text{def}}{=} \{(i, i+2) \mid 1 \leq i \leq m-2\} \cup \{(m-1, 1), (m, 2)\}$

how?.



Example: CONNECTIVITY (continued)

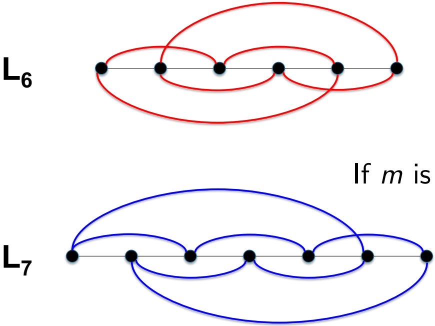
A much simpler proof using an FO-reduction.

Assume φ expresses connectivity of a graph $G = (V, E)$. Then we write a sentence ψ s.t. $(L_n, <) \models \psi$ iff $(L_{n+1}, <) \not\models \psi$.

In $(L_m, <)$ define: $E \stackrel{\text{def}}{=} \{(i, i+2) \mid 1 \leq i \leq m-2\} \cup \{(m-1, 1), (m, 2)\}$ how?.

If m is even then G is disconnected.

If m is odd, then G is connected.



Discussion

- The total oders $(L_m, <)$ are an isolated case when we can completely characterize when the duplicator has a winning strategy. Useful to reduce other problems to total orders, when possible.
- What happends if we replace $(m-1, 1), (m, 2)$ with only $(m-1, 2)$? (Useful in the homework).
- Hanf's lemma is only a sufficient condition; still useful in many cases.
- Next: prove the Ehrenfeucht-Fraisse theorem.

Proof of EF Theorem: Part 1

If $\boldsymbol{A} \sim_k \boldsymbol{B}$ then $\boldsymbol{A} \equiv_k \boldsymbol{B}$. Induction on k.

- $k = 0$. A $\varphi \in FO[0]$ is a Boolean combination of atoms $R(c_1, \ldots, c_k)$.
  $\boldsymbol{A} \equiv_0 \boldsymbol{B}$ implies $R^A(c_1^A, \ldots, c_k^A)$ iff $R^B(c_1^B, \ldots, c_k^B)$.
  Hence $\boldsymbol{A} \models \varphi$ iff $\boldsymbol{B} \models \varphi$.
- $k > 0$. Prove by induction on $\varphi \in FO[k]$ that $\boldsymbol{A} \models \varphi$ iff $\boldsymbol{B} \models \varphi$.
  - Assume $\boldsymbol{A} \models \exists x \psi(x)$, then there exists $a \in \boldsymbol{A}$ s.t. $\boldsymbol{A} \models \psi(a)$.
    When spoiler plays a, duplicator replies with $b \in B$.
    Thus[3], $(\boldsymbol{A}, a) \sim_{k-1} (\boldsymbol{B}, b)$, thus, $(\boldsymbol{A}, a) \equiv_{k-1} (\boldsymbol{B}, b)$ (induction on k).
    This implies $\boldsymbol{B} \models \psi(b)$, and $\boldsymbol{B} \models \exists x \psi(x)$.
  - Assume $\boldsymbol{A} \models \varphi_1 \wedge \varphi_2$. Then $\boldsymbol{A} \models \varphi_1$ and $\boldsymbol{A} \models \varphi_2$,
    hence $\boldsymbol{B} \models \varphi_1$ and $\boldsymbol{B} \models \varphi_2$ (induction on φ).
    This implies $\boldsymbol{B} \models \varphi_1 \wedge \varphi_2$.
  - Etc

[3] Structures extended with one more constant

Proof of EF Theorem: Part 1

If $\boldsymbol{A} \sim_k \boldsymbol{B}$ then $\boldsymbol{A} \equiv_k \boldsymbol{B}$. Induction on k.

- $k = 0$. A $\varphi \in FO[0]$ is a Boolean combination of atoms $R(c_1, \ldots, c_k)$.
  $\boldsymbol{A} \equiv_0 \boldsymbol{B}$ implies $R^A(c_1^A, \ldots, c_k^A)$ iff $R^B(c_1^B, \ldots, c_k^B)$.
  Hence $\boldsymbol{A} \vDash \varphi$ iff $\boldsymbol{B} \vDash \varphi$.
- $k > 0$. Prove by induction on $\varphi \in FO[k]$ that $\boldsymbol{A} \vDash \varphi$ iff $\boldsymbol{B} \vDash \varphi$.
  - Assume $\boldsymbol{A} \vDash \exists x \psi(x)$, then there exists $a \in \boldsymbol{A}$ s.t. $\boldsymbol{A} \vDash \psi(a)$.
    When spoiler plays a, duplicator replies with $b \in B$.
    Thus[3], $(\boldsymbol{A}, a) \sim_{k-1} (\boldsymbol{B}, b)$, thus, $(\boldsymbol{A}, a) \equiv_{k-1} (\boldsymbol{B}, b)$ (induction on k).
    This implies $\boldsymbol{B} \vDash \psi(b)$, and $\boldsymbol{B} \vDash \exists x \psi(x)$.
  - Assume $\boldsymbol{A} \vDash \varphi_1 \wedge \varphi_2$. Then $\boldsymbol{A} \vDash \varphi_1$ and $\boldsymbol{A} \vDash \varphi_2$,
    hence $\boldsymbol{B} \vDash \varphi_1$ and $\boldsymbol{B} \vDash \varphi_2$ (induction on φ).
    This implies $\boldsymbol{B} \vDash \varphi_1 \wedge \varphi_2$.
  - Etc

[3] Structures extended with one more constant

Proof of EF Theorem: Part 1

If $\boldsymbol{A} \sim_k \boldsymbol{B}$ then $\boldsymbol{A} \equiv_k \boldsymbol{B}$. Induction on k.

- $k = 0$. A $\varphi \in FO[0]$ is a Boolean combination of atoms $R(c_1, \ldots, c_k)$.
  $\boldsymbol{A} \equiv_0 \boldsymbol{B}$ implies $R^A(c_1^A, \ldots, c_k^A)$ iff $R^B(c_1^B, \ldots, c_k^B)$.
  Hence $\boldsymbol{A} \models \varphi$ iff $\boldsymbol{B} \models \varphi$.
- $k > 0$. Prove by induction on $\varphi \in FO[k]$ that $\boldsymbol{A} \models \varphi$ iff $\boldsymbol{B} \models \varphi$.
  - Assume $\boldsymbol{A} \models \exists x \psi(x)$, then there exists $a \in \boldsymbol{A}$ s.t. $\boldsymbol{A} \models \psi(a)$.
    When spoiler plays a, duplicator replies with $b \in B$.
    Thus[3], $(\boldsymbol{A}, a) \sim_{k-1} (\boldsymbol{B}, b)$, thus, $(\boldsymbol{A}, a) \equiv_{k-1} (\boldsymbol{B}, b)$ (induction on k).
    This implies $\boldsymbol{B} \models \psi(b)$, and $\boldsymbol{B} \models \exists x \psi(x)$.
  - Assume $\boldsymbol{A} \models \varphi_1 \wedge \varphi_2$. Then $\boldsymbol{A} \models \varphi_1$ and $\boldsymbol{A} \models \varphi_2$,
    hence $\boldsymbol{B} \models \varphi_1$ and $\boldsymbol{B} \models \varphi_2$ (induction on φ).
    This implies $\boldsymbol{B} \models \varphi_1 \wedge \varphi_2$.
  - Etc

[3] Structures extended with one more constant

Proof of EF Theorem: Part 1

If $\boldsymbol{A} \sim_k \boldsymbol{B}$ then $\boldsymbol{A} \equiv_k \boldsymbol{B}$. Induction on k.

- $k = 0$. A $\varphi \in FO[0]$ is a Boolean combination of atoms $R(c_1, \dots, c_k)$. $\boldsymbol{A} \equiv_0 \boldsymbol{B}$ implies $R^A(c_1^A, \dots, c_k^A)$ iff $R^B(c_1^B, \dots, c_k^B)$. Hence $\boldsymbol{A} \models \varphi$ iff $\boldsymbol{B} \models \varphi$.
- $k > 0$. Prove by induction on $\varphi \in FO[k]$ that $\boldsymbol{A} \models \varphi$ iff $\boldsymbol{B} \models \varphi$.
  - Assume $\boldsymbol{A} \models \exists x \psi(x)$, then there exists $a \in \boldsymbol{A}$ s.t. $\boldsymbol{A} \models \psi(a)$. When spoiler plays a, duplicator replies with $b \in B$. Thus[3], $(\boldsymbol{A}, a) \sim_{k-1} (\boldsymbol{B}, b)$, thus, $(\boldsymbol{A}, a) \equiv_{k-1} (\boldsymbol{B}, b)$ (induction on k). This implies $\boldsymbol{B} \models \psi(b)$, and $\boldsymbol{B} \models \exists x \psi(x)$.
  - Assume $\boldsymbol{A} \models \varphi_1 \wedge \varphi_2$. Then $\boldsymbol{A} \models \varphi_1$ and $\boldsymbol{A} \models \varphi_2$, hence $\boldsymbol{B} \models \varphi_1$ and $\boldsymbol{B} \models \varphi_2$ (induction on φ). This implies $\boldsymbol{B} \models \varphi_1 \wedge \varphi_2$.
  - Etc

[3] Structures extended with one more constant

Proof of EF Theorem: Part 1

If $\boldsymbol{A} \sim_k \boldsymbol{B}$ then $\boldsymbol{A} \equiv_k \boldsymbol{B}$. Induction on k.

- $k = 0$. A $\varphi \in FO[0]$ is a Boolean combination of atoms $R(c_1, \ldots, c_k)$.
  $\boldsymbol{A} \equiv_0 \boldsymbol{B}$ implies $R^A(c_1^A, \ldots, c_k^A)$ iff $R^B(c_1^B, \ldots, c_k^B)$.
  Hence $\boldsymbol{A} \models \varphi$ iff $\boldsymbol{B} \models \varphi$.
- $k > 0$. Prove by induction on $\varphi \in FO[k]$ that $\boldsymbol{A} \models \varphi$ iff $\boldsymbol{B} \models \varphi$.
  - Assume $\boldsymbol{A} \models \exists x \psi(x)$, then there exists $a \in A$ s.t. $\boldsymbol{A} \models \psi(a)$.
    When spoiler plays a, duplicator replies with $b \in B$.
    Thus[3], $(\boldsymbol{A}, a) \sim_{k-1} (\boldsymbol{B}, b)$, thus, $(\boldsymbol{A}, a) \equiv_{k-1} (\boldsymbol{B}, b)$ (induction on k).
    This implies $\boldsymbol{B} \models \psi(b)$, and $\boldsymbol{B} \models \exists x \psi(x)$.
  - Assume $\boldsymbol{A} \models \varphi_1 \wedge \varphi_2$. Then $\boldsymbol{A} \models \varphi_1$ and $\boldsymbol{A} \models \varphi_2$,
    hence $\boldsymbol{B} \models \varphi_1$ and $\boldsymbol{B} \models \varphi_2$ (induction on φ).
    This implies $\boldsymbol{B} \models \varphi_1 \wedge \varphi_2$.
  - Etc

[3] Structures extended with one more constant

Proof of EF Theorem: Part 1

If $\boldsymbol{A} \sim_k \boldsymbol{B}$ then $\boldsymbol{A} \equiv_k \boldsymbol{B}$. Induction on k.

- $k = 0$. A $\varphi \in FO[0]$ is a Boolean combination of atoms $R(c_1, \dots, c_k)$.
  $\boldsymbol{A} \equiv_0 \boldsymbol{B}$ implies $R^A(c_1^A, \dots, c_k^A)$ iff $R^B(c_1^B, \dots, c_k^B)$.
  Hence $\boldsymbol{A} \vDash \varphi$ iff $\boldsymbol{B} \vDash \varphi$.
- $k > 0$. Prove by induction on $\varphi \in FO[k]$ that $\boldsymbol{A} \vDash \varphi$ iff $\boldsymbol{B} \vDash \varphi$.
  - Assume $\boldsymbol{A} \vDash \exists x \psi(x)$, then there exists $a \in \boldsymbol{A}$ s.t. $\boldsymbol{A} \vDash \psi(a)$.
    When spoiler plays a, duplicator replies with $b \in B$.
    Thus[3], $(\boldsymbol{A}, a) \sim_{k-1} (\boldsymbol{B}, b)$, thus, $(\boldsymbol{A}, a) \equiv_{k-1} (\boldsymbol{B}, b)$ (induction on k).
    This implies $\boldsymbol{B} \vDash \psi(b)$, and $\boldsymbol{B} \vDash \exists x \psi(x)$.
  - Assume $\boldsymbol{A} \vDash \varphi_1 \wedge \varphi_2$. Then $\boldsymbol{A} \vDash \varphi_1$ and $\boldsymbol{A} \vDash \varphi_2$,
    hence $\boldsymbol{B} \vDash \varphi_1$ and $\boldsymbol{B} \vDash \varphi_2$ (induction on φ).
    This implies $\boldsymbol{B} \vDash \varphi_1 \wedge \varphi_2$.
  - Etc

[3] Structures extended with one more constant

Proof of EF Theorem: Part 1

If $\boldsymbol{A} \sim_k \boldsymbol{B}$ then $\boldsymbol{A} \equiv_k \boldsymbol{B}$. Induction on k.

- $k = 0$. A $\varphi \in FO[0]$ is a Boolean combination of atoms $R(c_1, \ldots, c_k)$.
  $\boldsymbol{A} \equiv_0 \boldsymbol{B}$ implies $R^A(c_1^A, \ldots, c_k^A)$ iff $R^B(c_1^B, \ldots, c_k^B)$.
  Hence $\boldsymbol{A} \vDash \varphi$ iff $\boldsymbol{B} \vDash \varphi$.
- $k > 0$. Prove by induction on $\varphi \in FO[k]$ that $\boldsymbol{A} \vDash \varphi$ iff $\boldsymbol{B} \vDash \varphi$.
  - Assume $\boldsymbol{A} \vDash \exists x \psi(x)$, then there exists $a \in \boldsymbol{A}$ s.t. $\boldsymbol{A} \vDash \psi(a)$.
    When spoiler plays a, duplicator replies with $b \in B$.
    Thus[3], $(\boldsymbol{A}, a) \sim_{k-1} (\boldsymbol{B}, b)$, thus, $(\boldsymbol{A}, a) \equiv_{k-1} (\boldsymbol{B}, b)$ (induction on k).
    This implies $\boldsymbol{B} \vDash \psi(b)$, and $\boldsymbol{B} \vDash \exists x \psi(x)$.
  - Assume $\boldsymbol{A} \vDash \varphi_1 \wedge \varphi_2$. Then $\boldsymbol{A} \vDash \varphi_1$ and $\boldsymbol{A} \vDash \varphi_2$,
    hence $\boldsymbol{B} \vDash \varphi_1$ and $\boldsymbol{B} \vDash \varphi_2$ (induction on φ).
    This implies $\boldsymbol{B} \vDash \varphi_1 \wedge \varphi_2$.
  - Etc

[3] Structures extended with one more constant

Proof of EF Theorem: Part 1

If $\boldsymbol{A} \sim_k \boldsymbol{B}$ then $\boldsymbol{A} \equiv_k \boldsymbol{B}$. Induction on k.

- $k = 0$. A $\varphi \in FO[0]$ is a Boolean combination of atoms $R(c_1, \ldots, c_k)$.
  $\boldsymbol{A} \equiv_0 \boldsymbol{B}$ implies $R^A(c_1^A, \ldots, c_k^A)$ iff $R^B(c_1^B, \ldots, c_k^B)$.
  Hence $\boldsymbol{A} \vDash \varphi$ iff $\boldsymbol{B} \vDash \varphi$.
- $k > 0$. Prove by induction on $\varphi \in FO[k]$ that $\boldsymbol{A} \vDash \varphi$ iff $\boldsymbol{B} \vDash \varphi$.
  - Assume $\boldsymbol{A} \vDash \exists x \psi(x)$, then there exists $a \in \boldsymbol{A}$ s.t. $\boldsymbol{A} \vDash \psi(a)$.
    When spoiler plays a, duplicator replies with $b \in B$.
    Thus[3], $(\boldsymbol{A}, a) \sim_{k-1} (\boldsymbol{B}, b)$, thus, $(\boldsymbol{A}, a) \equiv_{k-1} (\boldsymbol{B}, b)$ (induction on k).
    This implies $\boldsymbol{B} \vDash \psi(b)$, and $\boldsymbol{B} \vDash \exists x \psi(x)$.
  - Assume $\boldsymbol{A} \vDash \varphi_1 \wedge \varphi_2$. Then $\boldsymbol{A} \vDash \varphi_1$ and $\boldsymbol{A} \vDash \varphi_2$,
    hence $\boldsymbol{B} \vDash \varphi_1$ and $\boldsymbol{B} \vDash \varphi_2$ (induction on φ).
    This implies $\boldsymbol{B} \vDash \varphi_1 \wedge \varphi_2$.
  - Etc

[3] Structures extended with one more constant

Proof of EF Theorem: Part 1

If $\boldsymbol{A} \sim_k \boldsymbol{B}$ then $\boldsymbol{A} \equiv_k \boldsymbol{B}$. Induction on k.

- $k = 0$. A $\varphi \in FO[0]$ is a Boolean combination of atoms $R(c_1, \dots, c_k)$.
  $\boldsymbol{A} \equiv_0 \boldsymbol{B}$ implies $R^A(c_1^A, \dots, c_k^A)$ iff $R^B(c_1^B, \dots, c_k^B)$.
  Hence $\boldsymbol{A} \vDash \varphi$ iff $\boldsymbol{B} \vDash \varphi$.
- $k > 0$. Prove by induction on $\varphi \in FO[k]$ that $\boldsymbol{A} \vDash \varphi$ iff $\boldsymbol{B} \vDash \varphi$.
  - Assume $\boldsymbol{A} \vDash \exists x \psi(x)$, then there exists $a \in \boldsymbol{A}$ s.t. $\boldsymbol{A} \vDash \psi(a)$.
    When spoiler plays a, duplicator replies with $b \in B$.
    Thus[3], $(\boldsymbol{A}, a) \sim_{k-1} (\boldsymbol{B}, b)$, thus, $(\boldsymbol{A}, a) \equiv_{k-1} (\boldsymbol{B}, b)$ (induction on k).
    This implies $\boldsymbol{B} \vDash \psi(b)$, and $\boldsymbol{B} \vDash \exists x \psi(x)$.
  - Assume $\boldsymbol{A} \vDash \varphi_1 \wedge \varphi_2$. Then $\boldsymbol{A} \vDash \varphi_1$ and $\boldsymbol{A} \vDash \varphi_2$,
    hence $\boldsymbol{B} \vDash \varphi_1$ and $\boldsymbol{B} \vDash \varphi_2$ (induction on φ).
    This implies $\boldsymbol{B} \vDash \varphi_1 \wedge \varphi_2$.
  - Etc

[3] Structures extended with one more constant

Proof of EF Theorem: Part 1

If $\boldsymbol{A} \sim_k \boldsymbol{B}$ then $\boldsymbol{A} \equiv_k \boldsymbol{B}$. Induction on k.

- $k = 0$. A $\varphi \in FO[0]$ is a Boolean combination of atoms $R(c_1, \dots, c_k)$.
  $\boldsymbol{A} \equiv_0 \boldsymbol{B}$ implies $R^A(c_1^A, \dots, c_k^A)$ iff $R^B(c_1^B, \dots, c_k^B)$.
  Hence $\boldsymbol{A} \vDash \varphi$ iff $\boldsymbol{B} \vDash \varphi$.
- $k > 0$. Prove by induction on $\varphi \in FO[k]$ that $\boldsymbol{A} \vDash \varphi$ iff $\boldsymbol{B} \vDash \varphi$.
  - Assume $\boldsymbol{A} \vDash \exists x \psi(x)$, then there exists $a \in \boldsymbol{A}$ s.t. $\boldsymbol{A} \vDash \psi(a)$.
    When spoiler plays a, duplicator replies with $b \in B$.
    Thus[3], $(\boldsymbol{A}, a) \sim_{k-1} (\boldsymbol{B}, b)$, thus, $(\boldsymbol{A}, a) \equiv_{k-1} (\boldsymbol{B}, b)$ (induction on k).
    This implies $\boldsymbol{B} \vDash \psi(b)$, and $\boldsymbol{B} \vDash \exists x \psi(x)$.
  - Assume $\boldsymbol{A} \vDash \varphi_1 \wedge \varphi_2$. Then $\boldsymbol{A} \vDash \varphi_1$ and $\boldsymbol{A} \vDash \varphi_2$,
    hence $\boldsymbol{B} \vDash \varphi_1$ and $\boldsymbol{B} \vDash \varphi_2$ (induction on φ).
    This implies $\boldsymbol{B} \vDash \varphi_1 \wedge \varphi_2$.
  - Etc

[3] Structures extended with one more constant

Describing Winning Strategies

Fix $\boldsymbol{A}, \boldsymbol{B}$.

What is a "strategy" of the duplicator?

It is precisely a set $\mathcal{I}$ of partial isomorphisms $(\boldsymbol{a}, \boldsymbol{b})$ satisfying:

Definition

$\mathcal{I}$ has the *back-and-forth* property up to k if:

- $((),()) \in \mathcal{I}$ (it contains the empty partial isomorphism).
- Forth: forall $i < k$ if $((a_1, \ldots, a_i), (b_1, \ldots, b_i)) \in \mathcal{I}$ then $\forall a \in A, \exists b \in B$ s.t. $((a_1, \ldots, a_i, a), (b_1, \ldots, b_i, b)) \in \mathcal{I}$
- Back: forall $i < k$ if $((a_1, \ldots, a_i), (b_1, \ldots, b_i)) \in \mathcal{I}$ then $\forall b \in B, \exists a \in A$ s.t. $((a_1, \ldots, a_i, a), (b_1, \ldots, b_i, b)) \in \mathcal{I}$

Fact: a strategy for the duplicator is precisely a set of partial isomorphisms with the back-and-forth property. Proof in class.

Describing Winning Strategies

Fix $\boldsymbol{A}, \boldsymbol{B}$.

What is a "strategy" of the duplicator?

It is precisely a set $\mathcal{I}$ of partial isomorphisms $(\boldsymbol{a}, \boldsymbol{b})$ satisfying:

Definition

$\mathcal{I}$ has the *back-and-forth* property up to k if:

- $((),()) \in \mathcal{I}$ (it contains the empty partial isomorphism).
- Forth: forall $i < k$ if $((a_1, \ldots, a_i), (b_1, \ldots, b_i)) \in \mathcal{I}$ then $\forall a \in A, \exists b \in B$ s.t. $((a_1, \ldots, a_i, a), (b_1, \ldots, b_i, b)) \in \mathcal{I}$
- Back: forall $i < k$ if $((a_1, \ldots, a_i), (b_1, \ldots, b_i)) \in \mathcal{I}$ then $\forall b \in B, \exists a \in A$ s.t. $((a_1, \ldots, a_i, a), (b_1, \ldots, b_i, b)) \in \mathcal{I}$

Fact: a strategy for the duplicator is precisely a set of partial isomorphisms with the back-and-forth property. Proof in class.

Describing Winning Strategies

Fix $\boldsymbol{A}, \boldsymbol{B}$.

What is a "strategy" of the duplicator?

It is precisely a set $\mathcal{I}$ of partial isomorphisms $(\boldsymbol{a}, \boldsymbol{b})$ satisfying:

Definition

$\mathcal{I}$ has the *back-and-forth* property up to k if:

- $((), ()) \in \mathcal{I}$ (it contains the empty partial isomorphism).
- Forth: forall $i < k$ if $((a_1, \dots, a_i), (b_1, \dots, b_i)) \in \mathcal{I}$ then $\forall a \in A, \exists b \in B$ s.t. $((a_1, \dots, a_i, a), (b_1, \dots, b_i, b)) \in \mathcal{I}$
- Back: forall $i < k$ if $((a_1, \dots, a_i), (b_1, \dots, b_i)) \in \mathcal{I}$ then $\forall b \in B, \exists a \in A$ s.t. $((a_1, \dots, a_i, a), (b_1, \dots, b_i, b)) \in \mathcal{I}$

Fact: a strategy for the duplicator is precisely a set of partial isomorphisms with the back-and-forth property. Proof in class.

Types

Fix k and m.

Definition

Let $\boldsymbol{A}$ be a structure, $\boldsymbol{a} \stackrel{\text{def}}{=} (a_1, \dots, a_m) \in A^m$. The *rank k m-type* of $\boldsymbol{a}$ is:

$$\text{tp}_{k,m}(\boldsymbol{A}, \boldsymbol{a}) = \{\varphi(x_1, \dots, x_m) \in FO[k] \mid \boldsymbol{A} \models \varphi(a_1, \dots, a_m)\}$$

Facts:

- $\text{tp}_{k,m}(\boldsymbol{A}, \boldsymbol{a})$ is complete: forall $\varphi \in FO[k]$ either $\varphi \in \text{tp}_{k,m}(\boldsymbol{A}, \boldsymbol{a})$ or $\neg\varphi \in \text{tp}_{k,m}(\boldsymbol{A}, \boldsymbol{a})$ why?
- For all k, m there are only finitely many k, m-types why?
- There exists a single formula $\varphi^{\boldsymbol{A},\boldsymbol{a}}_{k,m}$ (the "type" of $\boldsymbol{a}$) s.t. forall $\boldsymbol{B}, \boldsymbol{b}$, $\text{tp}_{k,m}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}_{k,m}(\boldsymbol{B}, \boldsymbol{b})$ iff $\boldsymbol{B} \models \varphi^{\boldsymbol{A},\boldsymbol{a}}_{k,m}(\boldsymbol{b})$. why?

Types

Fix k and m.

Definition

Let $\boldsymbol{A}$ be a structure, $\boldsymbol{a} \overset{\text{def}}{=} (a_1, \ldots, a_m) \in A^m$. The *rank k m-type* of $\boldsymbol{a}$ is:

$$\mathrm{tp}_{k,m}(\boldsymbol{A}, \boldsymbol{a}) = \{\varphi(x_1, \ldots, x_m) \in FO[k] \mid \boldsymbol{A} \vDash \varphi(a_1, \ldots, a_m)\}$$

Facts:

- $\mathrm{tp}_{k,m}(\boldsymbol{A}, \boldsymbol{a})$ is complete:
  forall $\varphi \in FO[k]$ either $\varphi \in \mathrm{tp}_{k,m}(\boldsymbol{A}, \boldsymbol{a})$ or $\neg\varphi \in \mathrm{tp}_{k,m}(\boldsymbol{A}, \boldsymbol{a})$ why?
- For all k, m there are only finitely many k, m-types why?
- There exists a single formula $\varphi^{\boldsymbol{A},\boldsymbol{a}}_{k,m}$ (the "type" of $\boldsymbol{a}$) s.t. forall $\boldsymbol{B}, \boldsymbol{b}$, $\mathrm{tp}_{k,m}(\boldsymbol{A}, \boldsymbol{a}) = \mathrm{tp}_{k,m}(\boldsymbol{B}, \boldsymbol{b})$ iff $\boldsymbol{B} \vDash \varphi^{\boldsymbol{A},\boldsymbol{a}}_{k,m}(\boldsymbol{b})$. why?

Types

Fix k and m.

Definition

Let $\boldsymbol{A}$ be a structure, $\boldsymbol{a} \stackrel{\text{def}}{=} (a_1, \ldots, a_m) \in A^m$. The *rank k m-type* of $\boldsymbol{a}$ is:

$$\text{tp}_{k,m}(\boldsymbol{A}, \boldsymbol{a}) = \{\varphi(x_1, \ldots, x_m) \in FO[k] \mid \boldsymbol{A} \vDash \varphi(a_1, \ldots, a_m)\}$$

Facts:

- $\text{tp}_{k,m}(\boldsymbol{A}, \boldsymbol{a})$ is complete:
  forall $\varphi \in FO[k]$ either $\varphi \in \text{tp}_{k,m}(\boldsymbol{A}, \boldsymbol{a})$ or $\neg\varphi \in \text{tp}_{k,m}(\boldsymbol{A}, \boldsymbol{a})$ why?
- For all k, m there are only finitely many k, m-types why?
- There exists a single formula $\varphi^{\boldsymbol{A},\boldsymbol{a}}_{k,m}$ (the "type" of $\boldsymbol{a}$) s.t. forall $\boldsymbol{B}, \boldsymbol{b}$, $\text{tp}_{k,m}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}_{k,m}(\boldsymbol{B}, \boldsymbol{b})$ iff $\boldsymbol{B} \vDash \varphi^{\boldsymbol{A},\boldsymbol{a}}_{k,m}(\boldsymbol{b})$. why?

Types

Fix k and m.

Definition

Let $\boldsymbol{A}$ be a structure, $\boldsymbol{a} \stackrel{\text{def}}{=} (a_1, \ldots, a_m) \in A^m$. The *rank k m-type* of $\boldsymbol{a}$ is:

$$\text{tp}_{k,m}(\boldsymbol{A}, \boldsymbol{a}) = \{\varphi(x_1, \ldots, x_m) \in FO[k] \mid \boldsymbol{A} \models \varphi(a_1, \ldots, a_m)\}$$

Facts:

- $\text{tp}_{k,m}(\boldsymbol{A}, \boldsymbol{a})$ is complete:
  forall $\varphi \in FO[k]$ either $\varphi \in \text{tp}_{k,m}(\boldsymbol{A}, \boldsymbol{a})$ or $\neg\varphi \in \text{tp}_{k,m}(\boldsymbol{A}, \boldsymbol{a})$ why?
- For all k, m there are only finitely many k, m-types why?
- There exists a single formula $\varphi^{\boldsymbol{A},\boldsymbol{a}}_{k,m}$ (the "type" of $\boldsymbol{a}$) s.t. forall $\boldsymbol{B}, \boldsymbol{b}$, $\text{tp}_{k,m}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}_{k,m}(\boldsymbol{B}, \boldsymbol{b})$ iff $\boldsymbol{B} \models \varphi^{\boldsymbol{A},\boldsymbol{a}}_{k,m}(\boldsymbol{b})$. why?

Proof of EF Theorem: Part 2

If $\boldsymbol{A} \equiv_k \boldsymbol{B}$ then $\boldsymbol{A} \sim_k \boldsymbol{B}$.

Define $\mathcal{I} = \{(\boldsymbol{a}, \boldsymbol{b}) \mid \text{tp}_{k-i,i}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}_{k-i,i}(\boldsymbol{B}, \boldsymbol{b}),\ \text{where } i \stackrel{\text{def}}{=} |\boldsymbol{a}| = |\boldsymbol{b}|\}$

Then $((),()) \in \mathcal{I}$ why? Because $\boldsymbol{A} \equiv_k \boldsymbol{B}$, hence $\text{tp}_{k,0}(\boldsymbol{A},()) = \text{tp}_{k,0}(\boldsymbol{B},())$.
Let $i < k$ and suppose $\boldsymbol{a} = (a_1, \ldots, a_i)$, $\boldsymbol{b} = (b_1, \ldots, b_i)$ are s.t. $(\boldsymbol{a}, \boldsymbol{b}) \in \mathcal{I}$.

- Forth property. Let $a \in A$ and $\boldsymbol{a}' \stackrel{\text{def}}{=} (a_1, \ldots, a_i, a)$.
  For any $b \in B$, define $\boldsymbol{b}' \stackrel{\text{def}}{=} (b_1, \ldots, b_i, b)$.
  Suppose $\text{tp}_{k-i-1,i+1}(\boldsymbol{A}, \boldsymbol{a}') \neq \text{tp}_{k-i-1,i+1}(\boldsymbol{B}, \boldsymbol{b}')$.
  Let $\varphi_b(x_1, \ldots, x_i, y) \in FO[k-i-1]$ be s.t.

  $$\boldsymbol{A} \models \varphi_b(a_1, \ldots, a_i, a) \qquad\qquad \boldsymbol{B} \not\models \varphi_b(b_1, \ldots, b_i, b)$$

  Then $\boldsymbol{A} \models \psi(\boldsymbol{a})$ and $\boldsymbol{B} \not\models \psi(\boldsymbol{b})$ for $\psi \stackrel{\text{def}}{=} \exists y \bigwedge_b \varphi_b(x_1, \ldots, x_i, y)$.
  Since $\psi \in FO[k-i]$, it contradicts $\text{tp}_{k-i,i}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}_{k-i,i}(\boldsymbol{B}, \boldsymbol{b})$.
- Back property. Similar.

Proof of EF Theorem: Part 2

If $\boldsymbol{A} \equiv_k \boldsymbol{B}$ then $\boldsymbol{A} \sim_k \boldsymbol{B}$.

Define $\mathcal{I} = \{(\boldsymbol{a}, \boldsymbol{b}) \mid \text{tp}_{k-i,i}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}_{k-i,i}(\boldsymbol{B}, \boldsymbol{b}), \text{ where } i \stackrel{\text{def}}{=} |\boldsymbol{a}| = |\boldsymbol{b}|\}$

Then $((), ()) \in \mathcal{I}$ why? Because $\boldsymbol{A} \equiv_k \boldsymbol{B}$, hence $\text{tp}_{k,0}(\boldsymbol{A}, ()) = \text{tp}_{k,0}(\boldsymbol{B}, ())$.
Let $i < k$ and suppose $\boldsymbol{a} = (a_1, \ldots, a_i)$, $\boldsymbol{b} = (b_1, \ldots, b_i)$ are s.t. $(\boldsymbol{a}, \boldsymbol{b}) \in \mathcal{I}$.

- Forth property. Let $a \in A$ and $\boldsymbol{a}' \stackrel{\text{def}}{=} (a_1, \ldots, a_i, a)$.
  For any $b \in B$, define $\boldsymbol{b}' \stackrel{\text{def}}{=} (b_1, \ldots, b_i, b)$.
  Suppose $\text{tp}_{k-i-1,i+1}(\boldsymbol{A}, \boldsymbol{a}') \neq \text{tp}_{k-i-1,i+1}(\boldsymbol{B}, \boldsymbol{b}')$.
  Let $\varphi_b(x_1, \ldots, x_i, y) \in FO[k-i-1]$ be s.t.

  $$\boldsymbol{A} \models \varphi_b(a_1, \ldots, a_i, a) \qquad \boldsymbol{B} \not\models \varphi_b(b_1, \ldots, b_i, b)$$

  Then $\boldsymbol{A} \models \psi(\boldsymbol{a})$ and $\boldsymbol{B} \not\models \psi(\boldsymbol{b})$ for $\psi \stackrel{\text{def}}{=} \exists y \bigwedge_b \varphi_b(x_1, \ldots, x_i, y)$.
  Since $\psi \in FO[k-i]$, it contradicts $\text{tp}_{k-i,i}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}_{k-i,i}(\boldsymbol{B}, \boldsymbol{b})$.
- Back property. Similar.

Proof of EF Theorem: Part 2

If $\boldsymbol{A} \equiv_k \boldsymbol{B}$ then $\boldsymbol{A} \sim_k \boldsymbol{B}$.

Define $\mathcal{I} = \{(\boldsymbol{a}, \boldsymbol{b}) \mid \text{tp}_{k-i,i}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}_{k-i,i}(\boldsymbol{B}, \boldsymbol{b}), \text{ where } i \stackrel{\text{def}}{=} |\boldsymbol{a}| = |\boldsymbol{b}|\}$

Then $((), ()) \in \mathcal{I}$ why? Because $\boldsymbol{A} \equiv_k \boldsymbol{B}$, hence $\text{tp}_{k,0}(\boldsymbol{A}, ()) = \text{tp}_{k,0}(\boldsymbol{B}, ())$.

Let $i < k$ and suppose $\boldsymbol{a} = (a_1, \ldots, a_i)$, $\boldsymbol{b} = (b_1, \ldots, b_i)$ are s.t. $(\boldsymbol{a}, \boldsymbol{b}) \in \mathcal{I}$.

- Forth property. Let $a \in A$ and $\boldsymbol{a}' \stackrel{\text{def}}{=} (a_1, \ldots, a_i, a)$.
  For any $b \in B$, define $\boldsymbol{b}' \stackrel{\text{def}}{=} (b_1, \ldots, b_i, b)$.
  Suppose $\text{tp}_{k-i-1,i+1}(\boldsymbol{A}, \boldsymbol{a}') \neq \text{tp}_{k-i-1,i+1}(\boldsymbol{B}, \boldsymbol{b}')$.
  Let $\varphi_b(x_1, \ldots, x_i, y) \in FO[k-i-1]$ be s.t.

  $$\boldsymbol{A} \models \varphi_b(a_1, \ldots, a_i, a) \qquad \boldsymbol{B} \not\models \varphi_b(b_1, \ldots, b_i, b)$$

  Then $\boldsymbol{A} \models \psi(\boldsymbol{a})$ and $\boldsymbol{B} \not\models \psi(\boldsymbol{b})$ for $\psi \stackrel{\text{def}}{=} \exists y \bigwedge_b \varphi_b(x_1, \ldots, x_i, y)$.
  Since $\psi \in FO[k-i]$, it contradicts $\text{tp}_{k-i,i}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}_{k-i,i}(\boldsymbol{B}, \boldsymbol{b})$.
- Back property. Similar.

Proof of EF Theorem: Part 2

If $\boldsymbol{A} \equiv_k \boldsymbol{B}$ then $\boldsymbol{A} \sim_k \boldsymbol{B}$.

Define $\mathcal{I} = \{(\boldsymbol{a}, \boldsymbol{b}) \mid \text{tp}_{k-i,i}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}_{k-i,i}(\boldsymbol{B}, \boldsymbol{b}), \text{ where } i \stackrel{\text{def}}{=} |\boldsymbol{a}| = |\boldsymbol{b}|\}$

Then $((),()) \in \mathcal{I}$ why? Because $\boldsymbol{A} \equiv_k \boldsymbol{B}$, hence $\text{tp}_{k,0}(\boldsymbol{A},()) = \text{tp}_{k,0}(\boldsymbol{B},())$.

Let $i < k$ and suppose $\boldsymbol{a} = (a_1, \ldots, a_i)$, $\boldsymbol{b} = (b_1, \ldots, b_i)$ are s.t. $(\boldsymbol{a}, \boldsymbol{b}) \in \mathcal{I}$.

- Forth property. Let $a \in A$ and $\boldsymbol{a}' \stackrel{\text{def}}{=} (a_1, \ldots, a_i, a)$.
  For any $b \in B$, define $\boldsymbol{b}' \stackrel{\text{def}}{=} (b_1, \ldots, b_i, b)$.
  Suppose $\text{tp}_{k-i-1,i+1}(\boldsymbol{A}, \boldsymbol{a}') \neq \text{tp}_{k-i-1,i+1}(\boldsymbol{B}, \boldsymbol{b}')$.
  Let $\varphi_b(x_1, \ldots, x_i, y) \in FO[k-i-1]$ be s.t.

  $$\boldsymbol{A} \models \varphi_b(a_1, \ldots, a_i, a) \qquad \boldsymbol{B} \not\models \varphi_b(b_1, \ldots, b_i, b)$$

  Then $\boldsymbol{A} \models \psi(\boldsymbol{a})$ and $\boldsymbol{B} \not\models \psi(\boldsymbol{b})$ for $\psi \stackrel{\text{def}}{=} \exists y \bigwedge_b \varphi_b(x_1, \ldots, x_i, y)$.
  Since $\psi \in FO[k-i]$, it contradicts $\text{tp}_{k-i,i}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}_{k-i,i}(\boldsymbol{B}, \boldsymbol{b})$.
- Back property. Similar.

Proof of EF Theorem: Part 2

If $\boldsymbol{A} \equiv_k \boldsymbol{B}$ then $\boldsymbol{A} \sim_k \boldsymbol{B}$.

Define $\mathcal{I} = \{(\boldsymbol{a}, \boldsymbol{b}) \mid \text{tp}_{k-i,i}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}_{k-i,i}(\boldsymbol{B}, \boldsymbol{b}), \text{ where } i \stackrel{\text{def}}{=} |\boldsymbol{a}| = |\boldsymbol{b}|\}$

Then $((), ()) \in \mathcal{I}$ why? Because $\boldsymbol{A} \equiv_k \boldsymbol{B}$, hence $\text{tp}_{k,0}(\boldsymbol{A}, ()) = \text{tp}_{k,0}(\boldsymbol{B}, ())$.

Let $i < k$ and suppose $\boldsymbol{a} = (a_1, \ldots, a_i)$, $\boldsymbol{b} = (b_1, \ldots, b_i)$ are s.t. $(\boldsymbol{a}, \boldsymbol{b}) \in \mathcal{I}$.

- Forth property. Let $a \in A$ and $\boldsymbol{a}' \stackrel{\text{def}}{=} (a_1, \ldots, a_i, a)$.
  For any $b \in B$, define $\boldsymbol{b}' \stackrel{\text{def}}{=} (b_1, \ldots, b_i, b)$.
  Suppose $\text{tp}_{k-i-1,i+1}(\boldsymbol{A}, \boldsymbol{a}') \neq \text{tp}_{k-i-1,i+1}(\boldsymbol{B}, \boldsymbol{b}')$.
  Let $\varphi_b(x_1, \ldots, x_i, y) \in FO[k-i-1]$ be s.t.

  $$\boldsymbol{A} \models \varphi_b(a_1, \ldots, a_i, a) \qquad \boldsymbol{B} \not\models \varphi_b(b_1, \ldots, b_i, b)$$

  Then $\boldsymbol{A} \models \psi(\boldsymbol{a})$ and $\boldsymbol{B} \not\models \psi(\boldsymbol{b})$ for $\psi \stackrel{\text{def}}{=} \exists y \bigwedge_b \varphi_b(x_1, \ldots, x_i, y)$.
  Since $\psi \in FO[k-i]$, it contradicts $\text{tp}_{k-i,i}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}_{k-i,i}(\boldsymbol{B}, \boldsymbol{b})$.
- Back property. Similar.

Proof of EF Theorem: Part 2

If $\boldsymbol{A} \equiv_k \boldsymbol{B}$ then $\boldsymbol{A} \sim_k \boldsymbol{B}$.

Define $\mathcal{I} = \{(\boldsymbol{a}, \boldsymbol{b}) \mid \text{tp}_{k-i,i}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}_{k-i,i}(\boldsymbol{B}, \boldsymbol{b}), \text{ where } i \stackrel{\text{def}}{=} |\boldsymbol{a}| = |\boldsymbol{b}|\}$

Then $((), ()) \in \mathcal{I}$ why? Because $\boldsymbol{A} \equiv_k \boldsymbol{B}$, hence $\text{tp}_{k,0}(\boldsymbol{A}, ()) = \text{tp}_{k,0}(\boldsymbol{B}, ())$.

Let $i < k$ and suppose $\boldsymbol{a} = (a_1, \ldots, a_i)$, $\boldsymbol{b} = (b_1, \ldots, b_i)$ are s.t. $(\boldsymbol{a}, \boldsymbol{b}) \in \mathcal{I}$.

- Forth property. Let $a \in A$ and $\boldsymbol{a}' \stackrel{\text{def}}{=} (a_1, \ldots, a_i, a)$.
  For any $b \in B$, define $\boldsymbol{b}' \stackrel{\text{def}}{=} (b_1, \ldots, b_i, b)$.
  Suppose $\text{tp}_{k-i-1,i+1}(\boldsymbol{A}, \boldsymbol{a}') \neq \text{tp}_{k-i-1,i+1}(\boldsymbol{B}, \boldsymbol{b}')$.
  Let $\varphi_b(x_1, \ldots, x_i, y) \in FO[k-i-1]$ be s.t.

$$\boldsymbol{A} \models \varphi_b(a_1, \ldots, a_i, a) \qquad \boldsymbol{B} \not\models \varphi_b(b_1, \ldots, b_i, b)$$

  Then $\boldsymbol{A} \models \psi(\boldsymbol{a})$ and $\boldsymbol{B} \not\models \psi(\boldsymbol{b})$ for $\psi \stackrel{\text{def}}{=} \exists y \bigwedge_b \varphi_b(x_1, \ldots, x_i, y)$.
  Since $\psi \in FO[k-i]$, it contradicts $\text{tp}_{k-i,i}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}_{k-i,i}(\boldsymbol{B}, \boldsymbol{b})$.
- Back property. Similar.

Proof of EF Theorem: Part 2

If $\boldsymbol{A} \equiv_k \boldsymbol{B}$ then $\boldsymbol{A} \sim_k \boldsymbol{B}$.

Define $\mathcal{I} = \{(\boldsymbol{a}, \boldsymbol{b}) \mid \text{tp}_{k-i,i}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}_{k-i,i}(\boldsymbol{B}, \boldsymbol{b}),\ \text{where } i \stackrel{\text{def}}{=} |\boldsymbol{a}| = |\boldsymbol{b}|\}$

Then $((),()) \in \mathcal{I}$ why? Because $\boldsymbol{A} \equiv_k \boldsymbol{B}$, hence $\text{tp}_{k,0}(\boldsymbol{A},()) = \text{tp}_{k,0}(\boldsymbol{B},())$.

Let $i < k$ and suppose $\boldsymbol{a} = (a_1, \ldots, a_i)$, $\boldsymbol{b} = (b_1, \ldots, b_i)$ are s.t. $(\boldsymbol{a}, \boldsymbol{b}) \in \mathcal{I}$.

- Forth property. Let $a \in A$ and $\boldsymbol{a}' \stackrel{\text{def}}{=} (a_1, \ldots, a_i, a)$.
  For any $b \in B$, define $\boldsymbol{b}' \stackrel{\text{def}}{=} (b_1, \ldots, b_i, b)$.
  Suppose $\text{tp}_{k-i-1,i+1}(\boldsymbol{A}, \boldsymbol{a}') \neq \text{tp}_{k-i-1,i+1}(\boldsymbol{B}, \boldsymbol{b}')$.
  Let $\varphi_b(x_1, \ldots, x_i, y) \in FO[k-i-1]$ be s.t.

  $$\boldsymbol{A} \models \varphi_b(a_1, \ldots, a_i, a) \qquad \boldsymbol{B} \not\models \varphi_b(b_1, \ldots, b_i, b)$$

  Then $\boldsymbol{A} \models \psi(\boldsymbol{a})$ and $\boldsymbol{B} \not\models \psi(\boldsymbol{b})$ for $\psi \stackrel{\text{def}}{=} \exists y \bigwedge_b \varphi_b(x_1, \ldots, x_i, y)$.
  Since $\psi \in FO[k-i]$, it contradicts $\text{tp}_{k-i,i}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}_{k-i,i}(\boldsymbol{B}, \boldsymbol{b})$.
- Back property. Similar.

Proof of EF Theorem: Part 2

If $\boldsymbol{A} \equiv_k \boldsymbol{B}$ then $\boldsymbol{A} \sim_k \boldsymbol{B}$.

Define $\mathcal{I} = \{(\boldsymbol{a}, \boldsymbol{b}) \mid \text{tp}_{k-i,i}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}_{k-i,i}(\boldsymbol{B}, \boldsymbol{b}), \text{ where } i \stackrel{\text{def}}{=} |\boldsymbol{a}| = |\boldsymbol{b}|\}$

Then $((), ()) \in \mathcal{I}$ why? Because $\boldsymbol{A} \equiv_k \boldsymbol{B}$, hence $\text{tp}_{k,0}(\boldsymbol{A}, ()) = \text{tp}_{k,0}(\boldsymbol{B}, ())$.

Let $i < k$ and suppose $\boldsymbol{a} = (a_1, \ldots, a_i)$, $\boldsymbol{b} = (b_1, \ldots, b_i)$ are s.t. $(\boldsymbol{a}, \boldsymbol{b}) \in \mathcal{I}$.

- Forth property. Let $a \in A$ and $\boldsymbol{a}' \stackrel{\text{def}}{=} (a_1, \ldots, a_i, a)$.
  For any $b \in B$, define $\boldsymbol{b}' \stackrel{\text{def}}{=} (b_1, \ldots, b_i, b)$.
  Suppose $\text{tp}_{k-i-1,i+1}(\boldsymbol{A}, \boldsymbol{a}') \neq \text{tp}_{k-i-1,i+1}(\boldsymbol{B}, \boldsymbol{b}')$.
  Let $\varphi_b(x_1, \ldots, x_i, y) \in FO[k-i-1]$ be s.t.

  $$\boldsymbol{A} \models \varphi_b(a_1, \ldots, a_i, a) \qquad \boldsymbol{B} \not\models \varphi_b(b_1, \ldots, b_i, b)$$

  Then $\boldsymbol{A} \models \psi(\boldsymbol{a})$ and $\boldsymbol{B} \not\models \psi(\boldsymbol{b})$ for $\psi \stackrel{\text{def}}{=} \exists y \bigwedge_b \varphi_b(x_1, \ldots, x_i, y)$.
  Since $\psi \in FO[k-i]$, it contradicts $\text{tp}_{k-i,i}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}_{k-i,i}(\boldsymbol{B}, \boldsymbol{b})$.
- Back property. Similar.

Proof of EF Theorem: Part 2

If $\boldsymbol{A} \equiv_k \boldsymbol{B}$ then $\boldsymbol{A} \sim_k \boldsymbol{B}$.

Define $\mathcal{I} = \{(\boldsymbol{a}, \boldsymbol{b}) \mid \text{tp}_{k-i,i}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}_{k-i,i}(\boldsymbol{B}, \boldsymbol{b}),\ \text{where } i \stackrel{\text{def}}{=} |\boldsymbol{a}| = |\boldsymbol{b}|\}$

Then $((), ()) \in \mathcal{I}$ why? Because $\boldsymbol{A} \equiv_k \boldsymbol{B}$, hence $\text{tp}_{k,0}(\boldsymbol{A}, ()) = \text{tp}_{k,0}(\boldsymbol{B}, ())$.

Let $i < k$ and suppose $\boldsymbol{a} = (a_1, \ldots, a_i)$, $\boldsymbol{b} = (b_1, \ldots, b_i)$ are s.t. $(\boldsymbol{a}, \boldsymbol{b}) \in \mathcal{I}$.

- Forth property. Let $a \in A$ and $\boldsymbol{a}' \stackrel{\text{def}}{=} (a_1, \ldots, a_i, a)$.
  For any $b \in B$, define $\boldsymbol{b}' \stackrel{\text{def}}{=} (b_1, \ldots, b_i, b)$.
  Suppose $\text{tp}_{k-i-1,i+1}(\boldsymbol{A}, \boldsymbol{a}') \neq \text{tp}_{k-i-1,i+1}(\boldsymbol{B}, \boldsymbol{b}')$.
  Let $\varphi_b(x_1, \ldots, x_i, y) \in FO[k - i - 1]$ be s.t.

  $$\boldsymbol{A} \models \varphi_b(a_1, \ldots, a_i, a) \qquad \boldsymbol{B} \not\models \varphi_b(b_1, \ldots, b_i, b)$$

  Then $\boldsymbol{A} \models \psi(\boldsymbol{a})$ and $\boldsymbol{B} \not\models \psi(\boldsymbol{b})$ for $\psi \stackrel{\text{def}}{=} \exists y \bigwedge_b \varphi_b(x_1, \ldots, x_i, y)$.
  Since $\psi \in FO[k - i]$, it contradicts $\text{tp}_{k-i,i}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}_{k-i,i}(\boldsymbol{B}, \boldsymbol{b})$.
- Back property. Similar.

Discussion

- Ehrenfeucht-Fraisse games can be applied to infinite structures as well! If $\boldsymbol{A} \equiv_k \boldsymbol{B}$ forall $k \geq 0$, then $\boldsymbol{A} \equiv \boldsymbol{B}$.
- EF games generalize to other logics to prove inexpressibility results. We will discuss two:
  - Inexpressibility for ∃MSO
  - Inexpressibility for logics with recursion.

Second Order Logic

Second Order Logic, SO, extends FO with *2nd order variables*, which range over relations.

Example[4]:

$$\text{EVEN} \equiv \exists U(\forall x \exists! y (x \neq y) \wedge U(x,y) \wedge U(y,x))$$

Note: can always assume that 2nd order quantifiers come *before* 1st order quantifiers why?

[4] ∃! means "exists and is unique". write it in FO.

Second Order Logic

Second Order Logic, SO, extends FO with *2nd order variables*, which range over relations.

Example[4]:

$$\text{EVEN} \equiv \exists U(\forall x \exists! y(x \neq y) \wedge U(x,y) \wedge U(y,x))$$

Note: can always assume that 2nd order quantifiers come *before* 1st order quantifiers why?

[4] ∃! means "exists and is unique". write it in FO.

Second Order Logic

Second Order Logic, SO, extends FO with *2nd order variables*, which range over relations.

Example[4]:

$$\text{EVEN} \equiv \exists U (\forall x \exists ! y (x \neq y) \wedge U(x,y) \wedge U(y,x))$$

Note: can always assume that 2nd order quantifiers come *before* 1st order quantifiers why?

[4] ∃! means "exists and is unique". write it in FO.

Fragments of SO

Monadic Second Order Logic, MSO, restricts the 2nd order variables to be unary relations.

∃MSO and ∀MSO further restrict the 2-nd order quantifiers to ∃ or to ∀ respectively.

Example:

$$\begin{aligned}\text{3-COLORABILITY} \equiv \exists R \exists B \exists G \forall x (R(x) \vee B(x) \vee G(x)) \\ \wedge \forall x \forall y (E(x,y) \rightarrow \neg(R(x) \wedge R(y))) \\ \wedge \forall x \forall y (E(x,y) \rightarrow \neg(G(x) \wedge G(y))) \\ \wedge \forall x \forall y (E(x,y) \rightarrow \neg(B(x) \wedge B(y)))\end{aligned}$$

Fragments of SO

Monadic Second Order Logic, MSO, restricts the 2nd order variables to be unary relations.

∃MSO and ∀MSO further restrict the 2-nd order quantifiers to ∃ or to ∀ respectively.

Example:

$$\text{3-COLORABILITY} \equiv \exists R \exists B \exists G \forall x (R(x) \vee B(x) \vee G(x)) \\ \wedge \forall x \forall y (E(x,y) \rightarrow \neg(R(x) \wedge R(y))) \\ \wedge \forall x \forall y (E(x,y) \rightarrow \neg(G(x) \wedge G(y))) \\ \wedge \forall x \forall y (E(x,y) \rightarrow \neg(B(x) \wedge B(y)))$$

Fragments of SO

Monadic Second Order Logic, MSO, restricts the 2nd order variables to be unary relations.

∃MSO and ∀MSO further restrict the 2-nd order quantifiers to ∃ or to ∀ respectively.

Example:

$$\begin{aligned}
\text{3-COLORABILITY} \equiv \exists R \exists B \exists G \forall x (R(x) \vee B(x) \vee G(x)) \\
\wedge \forall x \forall y (E(x,y) \rightarrow \neg(R(x) \wedge R(y))) \\
\wedge \forall x \forall y (E(x,y) \rightarrow \neg(G(x) \wedge G(y))) \\
\wedge \forall x \forall y (E(x,y) \rightarrow \neg(B(x) \wedge B(y)))
\end{aligned}$$

MSO

Theorem

CONNECTIVITY *is expressible in* $\forall MSO$*.*

how??

$\forall U \forall x \forall y ((U(x) \wedge \neg U(y)) \rightarrow \exists u \exists v E(u,v) \wedge U(u) \wedge \neg U(v))$

Theorem (Fagin)

CONNECTIVITY *is not expressible in* $\exists MSO$*.*

We will prove it next, using games.

MSO

Theorem

CONNECTIVITY *is expressible in* $\forall MSO$*.*

how??

$\forall U \forall x \forall y ((U(x) \wedge \neg U(y)) \rightarrow \exists u \exists v E(u,v) \wedge U(u) \wedge \neg U(v))$

Theorem (Fagin)

CONNECTIVITY *is not expressible in* $\exists MSO$*.*

We will prove it next, using games.

MSO

Theorem

CONNECTIVITY *is expressible in* $\forall MSO$.

how??

$$\forall U \forall x \forall y \, ((U(x) \wedge \neg U(y)) \rightarrow \exists u \exists v E(u, v) \wedge U(u) \wedge \neg U(v))$$

Theorem (Fagin)

CONNECTIVITY *is not expressible in* $\exists MSO$.

We will prove it next, using games.

Games for ∃MSO

The (r, k)-Ajtai-Fagin game for ∃MSO and a problem P is the following:

- Duplicator picks a structure $\boldsymbol{A}$ that satisfies P.
- Spoiler picks r unary relations $U_1^A, \ldots, U_r^A$ on $\boldsymbol{A}$.
- Duplicator picks a structure $\boldsymbol{B}$ that does not satisfy P.
- Duplicator picks $U_1^B, \ldots, U_r^B$ in $\boldsymbol{B}$.
- Spoiler and Duplicator play an EF game with k pebbles on the structures $(\boldsymbol{A}, U_1^A, \ldots, U_r^A)$ and $(\boldsymbol{B}, U_1^B, \ldots, U_r^B)$.

Games for ∃MSO

Lemma

If Duplicator wins the (r, k) game, then no EMSO sentence with r 2-nd order quantifiers and k 1-st order quantifiers can express P.

Proof: Suppose $\varphi = \exists U_1 \cdots \exists U_r \psi$ is such a sentence. Then:

$$\boldsymbol{A} \models \exists U_1 \cdots \exists U_r \psi$$

exists sets $U_1^A, \ldots, U_r^A$ $\quad (\boldsymbol{A}, U_1^A, \ldots, U_r^A) \models \psi$

$$(\boldsymbol{B}, U_1^B, \ldots, U_r^B) \models \psi$$

$$\boldsymbol{B} \models \exists U_1 \cdots \exists U_r \psi$$

where $(\boldsymbol{B}, U_1^B, \ldots, U_r^B)$ is the structure chosen by the duplicator. This is a contradiction, since $\boldsymbol{B}$ does not satisfy P.

Games for ∃MSO

Lemma

If Duplicator wins the (r, k) game, then no EMSO sentence with r 2-nd order quantifiers and k 1-st order quantifiers can express P.

Proof: Suppose $\varphi = \exists U_1 \cdots \exists U_r \psi$ is such a sentence. Then:

$$\boldsymbol{A} \models \exists U_1 \cdots \exists U_r \psi$$

exists sets $U_1^A, \ldots, U_r^A$ $\quad (\boldsymbol{A}, U_1^A, \ldots, U_r^A) \models \psi$

$$(\boldsymbol{B}, U_1^B, \ldots, U_r^B) \models \psi$$

$$\boldsymbol{B} \models \exists U_1 \cdots \exists U_r \psi$$

where $(\boldsymbol{B}, U_1^B, \ldots, U_r^B)$ is the structure chosen by the duplicator. This is a contradiction, since $\boldsymbol{B}$ does not satisfy P.

Proof of Fagin's Theorem

CONNECTIVITY is not expressible in ∃MSO.

Fix r, k. Let $\boldsymbol{A}$ be a cycle C_n; will choose n later "big enough".

There are r unary relations, hence each $v \in C_n$ has one of 2^r colors.

For $d = 3^{k-1} - 1$, there are "a small number" of isomorphism types $N(a, d)$
Details: the number of types t is $t \leq (2^r)^{2d+1} = 2^{r(2d+1)}$.

If n is big, then we can find two elements u, v of the same type, at distance $d(u, v) \geq 2d + 2$.
Details: at least one type must occur $\geq n/t$ times; the first and the middle one are at distance $d(u, v) \geq n/(2t)$. Simply choose $n \geq 2t(2d + 2)$

"Cut" C_n at u,v and construct two cycles C_{n_1} (containing u) and C_{n_2} (containing v). Both $n_1, n_2 > 2d + 1$.

Finally: C_n is d-equivalent with $C_{n_1} \cup C_{n_2}$, hence use Hanf's lemma to derive $C_n \sim_k (C_{n_1} \cup C_{n_2})$.

Proof of Fagin's Theorem

CONNECTIVITY is not expressible in ∃MSO.

Fix r, k. Let $\boldsymbol{A}$ be a cycle C_n; will choose n later "big enough".

There are r unary relations, hence each $v \in C_n$ has one of 2^r colors.

For $d = 3^{k-1} - 1$, there are "a small number" of isomorphism types $N(a, d)$
Details: the number of types t is $t \leq (2^r)^{2d+1} = 2^{r(2d+1)}$.

If n is big, then we can find two elements u, v of the same type, at distance $d(u, v) \geq 2d + 2$.
Details: at least one type must occur $\geq n/t$ times; the first and the middle one are at distance $d(u, v) \geq n/(2t)$. Simply choose $n \geq 2t(2d + 2)$

"Cut" C_n at u,v and construct two cycles C_{n_1} (containing u) and C_{n_2} (containing v). Both $n_1, n_2 > 2d + 1$.

Finally: C_n is d-equivalent with $C_{n_1} \cup C_{n_2}$, hence use Hanf's lemma to derive $C_n \sim_k (C_{n_1} \cup C_{n_2})$.

Proof of Fagin's Theorem

CONNECTIVITY is not expressible in ∃MSO.

Fix r, k. Let $\boldsymbol{A}$ be a cycle C_n; will choose n later "big enough".

There are r unary relations, hence each $v \in C_n$ has one of 2^r colors.

For $d = 3^{k-1} - 1$, there are "a small number" of isomorphism types $N(a, d)$
Details: the number of types t is $t \leq (2^r)^{2d+1} = 2^{r(2d+1)}$.

If n is big, then we can find two elements u, v of the same type, at distance $d(u, v) \geq 2d + 2$.
Details: at least one type must occur $\geq n/t$ times; the first and the middle one are at distance $d(u, v) \geq n/(2t)$. Simply choose $n \geq 2t(2d + 2)$

"Cut" C_n at u,v and construct two cycles C_{n_1} (containing u) and C_{n_2} (containing v). Both $n_1, n_2 > 2d + 1$.

Finally: C_n is d-equivalent with $C_{n_1} \cup C_{n_2}$, hence use Hanf's lemma to derive $C_n \sim_k (C_{n_1} \cup C_{n_2})$.

Proof of Fagin's Theorem

CONNECTIVITY is not expressible in ∃MSO.

Fix r, k. Let $\boldsymbol{A}$ be a cycle C_n; will choose n later "big enough".

There are r unary relations, hence each $v \in C_n$ has one of 2^r colors.

For $d = 3^{k-1} - 1$, there are "a small number" of isomorphism types $N(a, d)$
Details: the number of types t is $t \le (2^r)^{2d+1} = 2^{r(2d+1)}$.

If n is big, then we can find two elements u, v of the same type, at distance $d(u, v) \ge 2d + 2$.
Details: at least one type must occur $\ge n/t$ times; the first and the middle one are at distance $d(u, v) \ge n/(2t)$. Simply choose $n \ge 2t(2d + 2)$

"Cut" C_n at u,v and construct two cycles C_{n_1} (containing u) and C_{n_2} (containing v). Both $n_1, n_2 > 2d + 1$.

Finally: C_n is d-equivalent with $C_{n_1} \cup C_{n_2}$, hence use Hanf's lemma to derive $C_n \sim_k (C_{n_1} \cup C_{n_2})$.

Proof of Fagin's Theorem

CONNECTIVITY is not expressible in ∃MSO.

Fix r, k. Let $\boldsymbol{A}$ be a cycle C_n; will choose n later "big enough".

There are r unary relations, hence each $v \in C_n$ has one of 2^r colors.

For $d = 3^{k-1} - 1$, there are "a small number" of isomorphism types $N(a, d)$
Details: the number of types t is $t \leq (2^r)^{2d+1} = 2^{r(2d+1)}$.

If n is big, then we can find two elements u, v of the same type, at distance $d(u, v) \geq 2d + 2$.
Details: at least one type must occur $\geq n/t$ times; the first and the middle one are at distance $d(u, v) \geq n/(2t)$. Simply choose $n \geq 2t(2d + 2)$

"Cut" C_n at u,v and construct two cycles C_{n_1} (containing u) and C_{n_2} (containing v). Both $n_1, n_2 > 2d + 1$.

Finally: C_n is d-equivalent with $C_{n_1} \cup C_{n_2}$, hence use Hanf's lemma to derive $C_n \sim_k (C_{n_1} \cup C_{n_2})$.

Proof of Fagin's Theorem

CONNECTIVITY is not expressible in ∃MSO.

Fix r, k. Let $\boldsymbol{A}$ be a cycle C_n; will choose n later "big enough".

There are r unary relations, hence each $v \in C_n$ has one of 2^r colors.

For $d = 3^{k-1} - 1$, there are "a small number" of isomorphism types $N(a, d)$
Details: the number of types t is $t \leq (2^r)^{2d+1} = 2^{r(2d+1)}$.

If n is big, then we can find two elements u, v of the same type, at distance $d(u, v) \geq 2d + 2$.
Details: at least one type must occur $\geq n/t$ times; the first and the middle one are at distance $d(u, v) \geq n/(2t)$. Simply choose $n \geq 2t(2d + 2)$

"Cut" C_n at u,v and construct two cycles C_{n_1} (containing u) and C_{n_2} (containing v). Both $n_1, n_2 > 2d + 1$.

Finally: C_n is d-equivalent with $C_{n_1} \cup C_{n_2}$, hence use Hanf's lemma to derive $C_n \sim_k (C_{n_1} \cup C_{n_2})$.

Recursion

Several logics add recursion to FO, in order to express CONNECTIVITY and similar queries.

The nicest way to describe these logics is using datalog.

Datalog

The vocabulary consists of two kinds of relation names:

- EDB predicates = input relations $R_1, R_2, \ldots$
- IDB predicates = computed relations $P_1, P_2, \ldots$

A datalog program is a set of rules of the form:

$$P(x, y, z, \ldots) \leftarrow \texttt{Body}$$

where the `Body` is a conjunction of literals.
The rule is safe if every variable in the head occurs in some positive relational literal.

Datalog by Example

Transitive closure:

$$
\begin{aligned}
T(x,y) &\leftarrow R(x,y)\\
T(x,y) &\leftarrow R(x,z), T(z,y)
\end{aligned}
$$

Equivalent formulation in FO:

$$
\begin{aligned}
\forall x \forall y T(x,y) &\leftarrow R(x,y)\\
\forall x \forall y \forall z T(x,y) &\leftarrow R(x,z) \wedge T(z,y)
\end{aligned}
$$

Also:

$$
\begin{aligned}
\forall x \forall y T(x,y) &\leftarrow R(x,y)\\
\forall x \forall y T(x,y) &\leftarrow \exists z (R(x,z) \wedge T(z,y))
\end{aligned}
$$

A non-head variable is called an existential variable; e.g. z

Datalog by Example

Transitive closure:

$$T(x,y) \leftarrow R(x,y)$$
$$T(x,y) \leftarrow R(x,z), T(z,y)$$

Equivalent formulation in FO:

$$\forall x \forall y T(x,y) \leftarrow R(x,y)$$
$$\forall x \forall y \forall z T(x,y) \leftarrow R(x,z) \wedge T(z,y)$$

Also:

$$\forall x \forall y T(x,y) \leftarrow R(x,y)$$
$$\forall x \forall y T(x,y) \leftarrow \exists z (R(x,z) \wedge T(z,y))$$

A non-head variable is called an existential variable; e.g. z

Datalog by Example

Transitive closure:

$$T(x,y) \leftarrow R(x,y)$$
$$T(x,y) \leftarrow R(x,z), T(z,y)$$

Equivalent formulation in FO:

$$\forall x \forall y T(x,y) \leftarrow R(x,y)$$
$$\forall x \forall y \forall z T(x,y) \leftarrow R(x,z) \wedge T(z,y)$$

Also:

$$\forall x \forall y T(x,y) \leftarrow R(x,y)$$
$$\forall x \forall y T(x,y) \leftarrow \exists z (R(x,z) \wedge T(z,y))$$

A non-head variable is called an existential variable; e.g. z

Datalog by Example

Transitive closure:

$$T(x,y) \leftarrow R(x,y)$$
$$T(x,y) \leftarrow R(x,z), T(z,y)$$

Equivalent formulation in FO:

$$\forall x \forall y T(x,y) \leftarrow R(x,y)$$
$$\forall x \forall y \forall z T(x,y) \leftarrow R(x,z) \wedge T(z,y)$$

Also:

$$\forall x \forall y T(x,y) \leftarrow R(x,y)$$
$$\forall x \forall y T(x,y) \leftarrow \exists z (R(x,z) \wedge T(z,y))$$

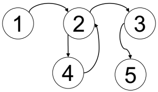
A non-head variable is called an existential variable; e.g. z

Fixpoint Semantics of Datalog

Informally, the fixpoint semantics is this. Start with the IDB = $\emptyset$, compute iteratively until fixpoint.

Fixpoint Semantics of Datalog

Informally, the fixpoint semantics is this. Start with the IDB = ∅, compute iteratively until fixpoint.
E.g. Transitive closure:

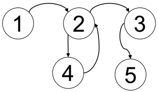


$$T_0 = \emptyset$$
$$T_{i+1} = \{(x,y) \mid R(x,y) \vee (\exists z(R(x,z) \wedge T_i(z,y)))\}$$

Fixpoint Semantics of Datalog

Informally, the fixpoint semantics is this. Start with the IDB = ∅, compute iteratively until fixpoint.

E.g. Transitive closure:



$$T_0 = \emptyset$$
$$T_{i+1} = \{(x,y) \mid R(x,y) \vee (\exists z(R(x,z) \wedge T_i(z,y)))\}$$

i	T_i
0	∅
1	(1,2), (2,3), (2,4), (4,2), (3,5)
2	(1,2), (2,3), (2,4), (4,2), (3,5), (1,3), (1,4), (4,3), (2,5)
2	(1,2), (2,3), (2,4), (4,2), (3,5), (1,3), (1,4), (4,3), (2,5), (1,5), (4,5)
3	(1,2), (2,3), (2,4), (4,2), (3,5), (1,3), (1,4), (4,3), (2,5), (1,5), (4,5)

Discussion

- Datalog can express some cool queries (try at home; may need $\neg$):
  - Same generation: if $G = (V, E)$ is a tree, find pairs of nodes x, y in the same generation (same distance to the root)
  - Given G find tuples (x, y, u, v) s.t. $d(x, y) = d(u, v)$ (same distance).
  - Check if G is a totally balanced tree.
- But it cannot express some trivial queries:
  - Is $|E|$ even?
  - Is $|A| \leq |B|$? (Homework)
- To prove inexpressibility results for datalog we will show that it is a subset of a much more powerful logic, $L^{\omega}_{\infty\omega}$, then describe pebble games for it.

FO^k

- FO^k is FO restricted to k variables $x_1, x_2, \ldots, x_k$.
- Example "there exists two nodes connected by 10 edges" in FO^3

$$\exists x \exists z (\exists y E(x,y) \wedge \underbrace{\exists x (E(y,x) \wedge \underbrace{\exists y (E(x,y) \wedge \underbrace{\ldots \exists x (E(y,x) \wedge E(x,z)))}_{\text{reuse } x}}_{\text{reuse } y}}_{\text{reuse } x}$$

Proposition

Consider a datalog program using k variables. Let T_n be an IDB relation after n iterations. Then $T_n \in FO^k$. why?

The datatlog program is equivalent to $T_0 \vee T_1 \vee T_2 \vee \cdots$

FO^k

- FO^k is FO restricted to k variables $x_1, x_2, \ldots, x_k$.
- Example "there exists two nodes connected by 10 edges" in FO^3

$$\exists x \exists z(\exists y E(x,y) \wedge \underbrace{\exists x(E(y,x) \wedge \underbrace{\exists y(E(x,y) \wedge \underbrace{\ldots \exists x(E(y,x) \wedge E(x,z)))}_{\text{reuse } x}}_{\text{reuse } y}}_{\text{reuse } x}$$

Proposition

Consider a datalog program using k variables. Let T_n be an IDB relation after n iterations. Then $T_n \in FO^k$. why?

The datatlog program is equivalent to $T_0 \vee T_1 \vee T_2 \vee \cdots$

FO^k

- FO^k is FO restricted to k variables $x_1, x_2, \ldots, x_k$.
- Example "there exists two nodes connected by 10 edges" in FO^3

$$\exists x \exists z(\exists y E(x,y) \wedge \underbrace{\exists x(E(y,x) \wedge \underbrace{\exists y(E(x,y) \wedge \underbrace{\ldots \exists x(E(y,x) \wedge E(x,z)))}_{\text{reuse } x}}_{\text{reuse } y}}_{\text{reuse } x}$$

Proposition

Consider a datalog program using k variables. Let T_n be an IDB relation after n iterations. Then $T_n \in FO^k$. why?

The datatlog program is equivalent to $T_0 \vee T_1 \vee T_2 \vee \cdots$

FO^k

- FO^k is FO restricted to k variables $x_1, x_2, \ldots, x_k$.
- Example "there exists two nodes connected by 10 edges" in FO^3

$$\exists x \exists z (\exists y E(x,y) \wedge \underbrace{\exists x (E(y,x) \wedge \underbrace{\exists y (E(x,y) \wedge \underbrace{\ldots \exists x (E(y,x) \wedge E(x,z)))}_{\text{reuse } x}}_{\text{reuse } y}}_{\text{reuse } x}$$

Proposition

Consider a datalog program using k variables. Let T_n be an IDB relation after n iterations. Then $T_n \in FO^k$. why?

The datatlog program is equivalent to $T_0 \vee T_1 \vee T_2 \vee \cdots$

FO^k

- FO^k is FO restricted to k variables $x_1, x_2, \ldots, x_k$.
- Example "there exists two nodes connected by 10 edges" in FO^3

$$\exists x \exists z (\exists y E(x,y) \wedge \underbrace{\exists x (E(y,x) \wedge \underbrace{\exists y (E(x,y) \wedge \underbrace{\ldots \exists x (E(y,x) \wedge E(x,z)))}_{\text{reuse } x}}_{\text{reuse } y}}_{\text{reuse } x}$$

Proposition

Consider a datalog program using k variables. Let T_n be an IDB relation after n iterations. Then $T_n \in FO^k$. *why?*

The datatlog program is equivalent to $T_0 \vee T_1 \vee T_2 \vee \cdots$

$L^{\omega}_{\infty\omega}$

- Let α, β be ordinals[5]. The infinitary logic $L_{\alpha\beta}$ is:

$$\text{Atoms: } x_i = x_j,\ R(\cdots); \qquad \bigvee_{i\in I} \varphi_i; \qquad \underbrace{(\ldots \exists x_j \ldots)}_{j\in J}\varphi; \qquad \neg\varphi$$

where $|I| < \alpha$, $|J| < \beta$.

- $L_{\omega\omega} = FO$; finite disjunctions, finite quantifier sequence.
- $L_{\infty\omega}$ = infinite disjunction (no bound!), finite quantifier sequence. Note: the quantifier rank may be any ordinal, e.g. $\omega + 1$ in class
- $L^k_{\infty\omega}$ = the restriction to k variables.
- $L^{\omega}_{\infty\omega} = \bigcup_{k\geq 0} L^k_{\infty\omega}$. What is $\bigcup_{k\geq 0} FO^k$?

[5] An *ordinal*= isomorphism type of a well order. E.g. $\omega = \{1, 2, 3, \ldots\}$.

$L^{\omega}_{\infty\omega}$

- Let α, β be ordinals[5]. The infinitary logic $L_{\alpha\beta}$ is:

$$\text{Atoms: } x_i = x_j,\ R(\cdots); \qquad \bigvee_{i \in I} \varphi_i; \qquad \underbrace{(\ldots \exists x_j \ldots)}_{j \in J}\varphi; \qquad \neg\varphi$$

  where $|I| < \alpha$, $|J| < \beta$.

- $L_{\omega\omega} = FO$; finite disjunctions, finite quantifier sequence.
- $L_{\infty\omega}$ = infinite disjunction (no bound!), finite quantifier sequence. Note: the quantifier rank may be any ordinal, e.g. $\omega + 1$ in class
- $L^k_{\infty\omega}$ = the restriction to k variables.
- $L^{\omega}_{\infty\omega} = \bigcup_{k \geq 0} L^k_{\infty\omega}$. What is $\bigcup_{k \geq 0} FO^k$?

[5] An *ordinal*= isomorphism type of a well order. E.g. $\omega = \{1, 2, 3, \ldots\}$.

$L^{\omega}_{\infty\omega}$

- Let α, β be ordinals[5]. The infinitary logic $L_{\alpha\beta}$ is:

$$\text{Atoms: } x_i = x_j,\ R(\cdots); \qquad \bigvee_{i \in I} \varphi_i; \qquad \underbrace{(\ldots \exists x_j \ldots)}_{j \in J}\varphi; \qquad \neg\varphi$$

  where $|I| < \alpha$, $|J| < \beta$.

- $L_{\omega\omega} = FO$; finite disjunctions, finite quantifier sequence.
- $L_{\infty\omega}$ = infinite disjunction (no bound!), finite quantifier sequence. Note: the quantifier rank may be any ordinal, e.g. $\omega + 1$ in class
- $L^k_{\infty\omega}$ = the restriction to k variables.
- $L^{\omega}_{\infty\omega} = \bigcup_{k \geq 0} L^k_{\infty\omega}$. What is $\bigcup_{k \geq 0} FO^k$?

[5] An *ordinal*= isomorphism type of a well order. E.g. $\omega = \{1, 2, 3, \ldots\}$.

$L^{\omega}_{\infty\omega}$

- Let α, β be ordinals[5]. The infinitary logic $L_{\alpha\beta}$ is:

$$\text{Atoms: } x_i = x_j,\ R(\cdots); \qquad \bigvee_{i \in I} \varphi_i; \qquad \underbrace{(\ldots \exists x_j \ldots)}_{j \in J}\varphi; \qquad \neg\varphi$$

  where $|I| < \alpha$, $|J| < \beta$.

- $L_{\omega\omega} = FO$; finite disjunctions, finite quantifier sequence.
- $L_{\infty\omega}$ = infinite disjunction (no bound!), finite quantifier sequence. Note: the quantifier rank may be any ordinal, e.g. $\omega + 1$ in class
- $L^k_{\infty\omega}$ = the restriction to k variables.
- $L^{\omega}_{\infty\omega} = \bigcup_{k \geq 0} L^k_{\infty\omega}$. What is $\bigcup_{k \geq 0} FO^k$?

[5] An *ordinal*= isomorphism type of a well order. E.g. $\omega = \{1, 2, 3, \ldots\}$.

$L^{\omega}_{\infty\omega}$

- Let α, β be ordinals[5]. The infinitary logic $L_{\alpha\beta}$ is:

$$\text{Atoms: } x_i = x_j,\ R(\cdots); \qquad \bigvee_{i \in I} \varphi_i; \qquad \underbrace{(\ldots \exists x_j \ldots)}_{j \in J}\varphi; \qquad \neg\varphi$$

  where $|I| < \alpha$, $|J| < \beta$.

- $L_{\omega\omega} = FO$; finite disjunctions, finite quantifier sequence.
- $L_{\infty\omega}$ = infinite disjunction (no bound!), finite quantifier sequence. Note: the quantifier rank may be any ordinal, e.g. $\omega + 1$ in class
- $L^k_{\infty\omega}$ = the restriction to k variables.
- $L^{\omega}_{\infty\omega} = \bigcup_{k \geq 0} L^k_{\infty\omega}$. What is $\bigcup_{k \geq 0} FO^k$?

[5] An *ordinal*= isomorphism type of a well order. E.g. $\omega = \{1, 2, 3, \ldots\}$.

$L^{\omega}_{\infty\omega}$

- Let α, β be ordinals[5]. The infinitary logic $L_{\alpha\beta}$ is:

$$\text{Atoms: } x_i = x_j,\ R(\cdots); \qquad \bigvee_{i \in I} \varphi_i; \qquad \underbrace{(\ldots \exists x_j \ldots)}_{j \in J}\varphi; \qquad \neg\varphi$$

  where $|I| < \alpha$, $|J| < \beta$.

- $L_{\omega\omega} = FO$; finite disjunctions, finite quantifier sequence.
- $L_{\infty\omega}$ = infinite disjunction (no bound!), finite quantifier sequence. Note: the quantifier rank may be any ordinal, e.g. $\omega + 1$ in class
- $L^k_{\infty\omega}$ = the restriction to k variables.
- $L^{\omega}_{\infty\omega} = \bigcup_{k \geq 0} L^k_{\infty\omega}$.

What is $\bigcup_{k \geq 0} FO^k$?

[5] An *ordinal*= isomorphism type of a well order. E.g. $\omega = \{1, 2, 3, \ldots\}$.

Discussion

- Any property P on finite structures can be expressed by in $L_{\infty\omega}$ why?
  Let $\varphi_{\boldsymbol{A}}$ fully describes $\boldsymbol{A}$. Then P is expressed by $\bigvee_{\boldsymbol{A} \models P} \varphi_{\boldsymbol{A}}$.
- Thus, $L_{\infty\omega}$ is too powerful to prove inexpressibility.
- $L^{\omega}_{\infty\omega}$ is much weaker. We will show it cannot express EVEN.
- Datalog $\subseteq L^{\omega}_{\infty\omega}$ why? Hence it cannot express EVEN.
- $L^{k}_{\infty\omega}$ admits a normal form on finite structures: $\varphi' = \bigvee_{i \in \mathbb{N}} \psi_i$ where
  - $\psi_i \in FO^k$, for $i = 1, 2, \ldots$
  - For any finite structure, $\boldsymbol{A} \models \varphi$ iff $\boldsymbol{A} \models \varphi'$.

Discussion

- Any property P on finite structures can be expressed by in $L_{\infty\omega}$ why?
  Let $\varphi_{\boldsymbol{A}}$ fully describes $\boldsymbol{A}$. Then P is expressed by $\bigvee_{\boldsymbol{A} \models P} \varphi_{\boldsymbol{A}}$.
- Thus, $L_{\infty\omega}$ is too powerful to prove inexpressibility.
- $L^{\omega}_{\infty\omega}$ is much weaker. We will show it cannot express EVEN.
- Datalog $\subseteq L^{\omega}_{\infty\omega}$ why? Hence it cannot express EVEN.
- $L^{k}_{\infty\omega}$ admits a normal form on finite structures: $\varphi' = \bigvee_{i \in \mathbb{N}} \psi_i$ where
  - $\psi_i \in FO^k$, for $i = 1, 2, \ldots$
  - For any finite structure, $\boldsymbol{A} \models \varphi$ iff $\boldsymbol{A} \models \varphi'$.

Discussion

- Any property P on finite structures can be expressed by in $L_{\infty\omega}$ why?
  Let $\varphi_{\boldsymbol{A}}$ fully describes $\boldsymbol{A}$. Then P is expressed by $\bigvee_{\boldsymbol{A} \models P} \varphi_{\boldsymbol{A}}$.
- Thus, $L_{\infty\omega}$ is too powerful to prove inexpressibility.
- $L^{\omega}_{\infty\omega}$ is much weaker. We will show it cannot express EVEN.
- Datalog $\subseteq L^{\omega}_{\infty\omega}$ why? Hence it cannot express EVEN.
- $L^{k}_{\infty\omega}$ admits a normal form on finite structures: $\varphi' = \bigvee_{i \in \mathbb{N}} \psi_i$ where
  - $\psi_i \in FO^k$, for $i = 1, 2, \ldots$
  - For any finite structure, $\boldsymbol{A} \models \varphi$ iff $\boldsymbol{A} \models \varphi'$.

Discussion

- Any property P on finite structures can be expressed by in $L_{\infty\omega}$ why?
  Let $\varphi_{\boldsymbol{A}}$ fully describes $\boldsymbol{A}$. Then P is expressed by $\bigvee_{\boldsymbol{A} \models P} \varphi_{\boldsymbol{A}}$.
- Thus, $L_{\infty\omega}$ is too powerful to prove inexpressibility.
- $L^{\omega}_{\infty\omega}$ is much weaker. We will show it cannot express EVEN.
- Datalog $\subseteq L^{\omega}_{\infty\omega}$ why? Hence it cannot express EVEN.
- $L^{k}_{\infty\omega}$ admits a normal form on finite structures: $\varphi' = \bigvee_{i \in \mathbb{N}} \psi_i$ where
  - $\psi_i \in FO^k$, for $i = 1, 2, \ldots$
  - For any finite structure, $\boldsymbol{A} \models \varphi$ iff $\boldsymbol{A} \models \varphi'$.

Discussion

- Any property P on finite structures can be expressed by in $L_{\infty\omega}$ why?
  Let $\varphi_{\boldsymbol{A}}$ fully describes $\boldsymbol{A}$. Then P is expressed by $\bigvee_{\boldsymbol{A} \models P} \varphi_{\boldsymbol{A}}$.
- Thus, $L_{\infty\omega}$ is too powerful to prove inexpressibility.
- $L^{\omega}_{\infty\omega}$ is much weaker. We will show it cannot express EVEN.
- Datalog $\subseteq L^{\omega}_{\infty\omega}$ why? Hence it cannot express EVEN.
- $L^{k}_{\infty\omega}$ admits a normal form on finite structures: $\varphi' = \bigvee_{i \in \mathbb{N}} \psi_i$ where
  - $\psi_i \in FO^k$, for $i = 1, 2, \ldots$
  - For any finite structure, $\boldsymbol{A} \models \varphi$ iff $\boldsymbol{A} \models \varphi'$.

Discussion

- Any property P on finite structures can be expressed by in $L_{\infty\omega}$ why?
  Let $\varphi_{\boldsymbol{A}}$ fully describes $\boldsymbol{A}$. Then P is expressed by $\bigvee_{\boldsymbol{A} \models P} \varphi_{\boldsymbol{A}}$.
- Thus, $L_{\infty\omega}$ is too powerful to prove inexpressibility.
- $L^{\omega}_{\infty\omega}$ is much weaker. We will show it cannot express EVEN.
- Datalog $\subseteq L^{\omega}_{\infty\omega}$ why? Hence it cannot express EVEN.
- $L^{k}_{\infty\omega}$ admits a normal form on finite structures: $\varphi' = \bigvee_{i \in \mathbb{N}} \psi_i$ where
  - $\psi_i \in FO^k$, for $i = 1, 2, \ldots$
  - For any finite structure, $\boldsymbol{A} \models \varphi$ iff $\boldsymbol{A} \models \varphi'$.

Discussion

- Any property P on finite structures can be expressed by in $L_{\infty\omega}$ why?
  Let $\varphi_{\boldsymbol{A}}$ fully describes $\boldsymbol{A}$. Then P is expressed by $\bigvee_{\boldsymbol{A} \models P} \varphi_{\boldsymbol{A}}$.
- Thus, $L_{\infty\omega}$ is too powerful to prove inexpressibility.
- $L^{\omega}_{\infty\omega}$ is much weaker. We will show it cannot express EVEN.
- Datalog $\subseteq L^{\omega}_{\infty\omega}$ why? Hence it cannot express EVEN.
- $L^{k}_{\infty\omega}$ admits a normal form on finite structures: $\varphi' = \bigvee_{i \in \mathbb{N}} \psi_i$ where
  - $\psi_i \in FO^k$, for $i = 1, 2, \ldots$
  - For any finite structure, $\boldsymbol{A} \models \varphi$ iff $\boldsymbol{A} \models \varphi'$.

The k-Pebble Games

There are two structures $\boldsymbol{A}, \boldsymbol{B}$ and $2k$ pebbles, labeled $1, 1, 2, 2, \ldots, k, k$.

Initially both spoiler and duplicator have k pebbles in their hands; one of each label. At each round, spoiler chooses one of these moves:

- Place pebble i from his hand on $\boldsymbol{A}$ (or $\boldsymbol{B}$); the duplicator must reply by placing her pebble i on $\boldsymbol{B}$ (or $\boldsymbol{A}$).
- Remove pebble i from $\boldsymbol{A}$ (or $\boldsymbol{B}$); duplicator must reply by removing pebble i from $\boldsymbol{B}$ (or $\boldsymbol{A}$).

There are infinitely many rounds. Duplicator wins if at each round the set of pebbles on $\boldsymbol{A}$ and on $\boldsymbol{B}$ forms a partial isomorphism.

The k-Pebble Games

There are two structures $\boldsymbol{A}, \boldsymbol{B}$ and $2k$ pebbles, labeled $1, 1, 2, 2, \ldots, k, k$.

Initially both spoiler and duplicator have k pebbles in their hands; one of each label. At each round, spoiler chooses one of these moves:

- Place pebble i from his hand on $\boldsymbol{A}$ (or $\boldsymbol{B}$); the duplicator must reply by placing her pebble i on $\boldsymbol{B}$ (or $\boldsymbol{A}$).
- Remove pebble i from $\boldsymbol{A}$ (or $\boldsymbol{B}$); duplicator must reply by removing pebble i from $\boldsymbol{B}$ (or $\boldsymbol{A}$).

There are infinitely many rounds. Duplicator wins if at each round the set of pebbles on $\boldsymbol{A}$ and on $\boldsymbol{B}$ forms a partial isomorphism.

The k-Pebble Games

There are two structures $\boldsymbol{A}, \boldsymbol{B}$ and $2k$ pebbles, labeled $1, 1, 2, 2, \ldots, k, k$.

Initially both spoiler and duplicator have k pebbles in their hands; one of each label. At each round, spoiler chooses one of these moves:

- Place pebble i from his hand on $\boldsymbol{A}$ (or $\boldsymbol{B}$); the duplicator must reply by placing her pebble i on $\boldsymbol{B}$ (or $\boldsymbol{A}$).
- Remove pebble i from $\boldsymbol{A}$ (or $\boldsymbol{B}$); duplicator must reply by removing pebble i from $\boldsymbol{B}$ (or $\boldsymbol{A}$).

There are infinitely many rounds. Duplicator wins if at each round the set of pebbles on $\boldsymbol{A}$ and on $\boldsymbol{B}$ forms a partial isomorphism.

The k-Pebble Games

There are two structures $\boldsymbol{A}, \boldsymbol{B}$ and $2k$ pebbles, labeled $1, 1, 2, 2, \ldots, k, k$.

Initially both spoiler and duplicator have k pebbles in their hands; one of each label. At each round, spoiler chooses one of these moves:

- Place pebble i from his hand on $\boldsymbol{A}$ (or $\boldsymbol{B}$); the duplicator must reply by placing her pebble i on $\boldsymbol{B}$ (or $\boldsymbol{A}$).
- Remove pebble i from $\boldsymbol{A}$ (or $\boldsymbol{B}$); duplicator must reply by removing pebble i from $\boldsymbol{B}$ (or $\boldsymbol{A}$).

There are infinitely many rounds. Duplicator wins if at each round the set of pebbles on $\boldsymbol{A}$ and on $\boldsymbol{B}$ forms a partial isomorphism.

The k-Pebble Games

There are two structures $\boldsymbol{A}, \boldsymbol{B}$ and $2k$ pebbles, labeled $1, 1, 2, 2, \ldots, k, k$.

Initially both spoiler and duplicator have k pebbles in their hands; one of each label. At each round, spoiler chooses one of these moves:

- Place pebble i from his hand on $\boldsymbol{A}$ (or $\boldsymbol{B}$); the duplicator must reply by placing her pebble i on $\boldsymbol{B}$ (or $\boldsymbol{A}$).
- Remove pebble i from $\boldsymbol{A}$ (or $\boldsymbol{B}$); duplicator must reply by removing pebble i from $\boldsymbol{B}$ (or $\boldsymbol{A}$).

There are infinitely many rounds. Duplicator wins if at each round the set of pebbles on $\boldsymbol{A}$ and on $\boldsymbol{B}$ forms a partial isomorphism.

The k-Pebble Games: Discussion

- An equivalent formulation is that the spoiler never removes, but instead "moves" a pebble from one position to another (possibly on the other structure).
- It suffices to check partial isomorphism only when all k pebbles are placed on the structures why?

Main Theorem of Pebble Games

1. $\boldsymbol{A} \approx^k_{\infty\omega} \boldsymbol{B}$ denotes: duplicator wins the k-pebble game.
2. $\boldsymbol{A} \equiv^k_{\infty\omega} \boldsymbol{B}$ denotes: $\boldsymbol{A} \vDash \varphi$ iff $\boldsymbol{B} \vDash \varphi$, forall $\varphi \in L^k_{\infty\omega}$
3. $\boldsymbol{A} \equiv^k_{\text{FO}} \boldsymbol{B}$ denotes: $\boldsymbol{A} \vDash \varphi$ iff $\boldsymbol{B} \vDash \varphi$, forall $\varphi \in FO^k$.

Theorem

1 and 2 are equivalent. When $\boldsymbol{A}, \boldsymbol{B}$ are finite, then 1, 2, 3 are equivalent.

We will prove shortly, but first some examples.

Example: Total Order $L_n = ([n], <)$

We cannot distinguish L_m, L_n in $FO[r]$ (quantifier rank r), when $m, n \geq 2^r - 1$. But we can in FO^2 (two variables).

Proposition

If $m \neq n$ then $L_m \not\equiv^2_{FO} L_n$.

Proof. Define[6] $\varphi_0(x) \stackrel{\text{def}}{=} \mathbf{T}$, $\varphi_{p+1}(x) \stackrel{\text{def}}{=} \exists y((x < y) \wedge \varphi_p(y))$.

$$\varphi_1(x) = \exists y(x < y) \quad \varphi_2(x) = \exists y(x < y \wedge (\exists x(y < x)))$$

$$\varphi_3(x) = \exists y(x < y \wedge (\exists x(y < x \wedge \exists y(x < y)))) \quad \ldots$$

what does $\varphi_p(x)$ say?

Let $\psi_p \stackrel{\text{def}}{=} \exists x \varphi_p(x) \wedge \neg \exists x \varphi_{p+1}(x)$. Then $L_m \models \psi_m$, $L_n \not\models \psi_m$, $\psi_m \in FO^2$.

[6] Switching x and y is a bit informal. Formally, we could set $\varphi_{p+1}(x) \stackrel{\text{def}}{=} \exists y(x < y \wedge \exists x(x = y \wedge \varphi_p(x)))$. Others ways are possible (without using =).

Example: Total Order $L_n = ([n], <)$

We cannot distinguish L_m, L_n in $FO[r]$ (quantifier rank r), when $m, n \geq 2^r - 1$. But we can in FO^2 (two variables).

Proposition

If $m \neq n$ then $L_m \not\equiv^2_{FO} L_n$.

Proof. Define[6] $\varphi_0(x) \stackrel{\text{def}}{=} \boldsymbol{T}$, $\varphi_{p+1}(x) \stackrel{\text{def}}{=} \exists y((x < y) \wedge \varphi_p(y))$.

$$\varphi_1(x) = \exists y(x < y) \quad \varphi_2(x) = \exists y(x < y \wedge (\exists x(y < x)))$$

$$\varphi_3(x) = \exists y(x < y \wedge (\exists x(y < x \wedge \exists y(x < y)))) \quad \ldots$$

what does $\varphi_p(x)$ say?

Let $\psi_p \stackrel{\text{def}}{=} \exists x \varphi_p(x) \wedge \neg \exists x \varphi_{p+1}(x)$. Then $L_m \models \psi_m$, $L_n \not\models \psi_m$, $\psi_m \in FO^2$.

---

[6] Switching x and y is a bit informal. Formally, we could set $\varphi_{p+1}(x) \stackrel{\text{def}}{=} \exists y(x < y \wedge \exists x(x = y \wedge \varphi_p(x)))$. Others ways are possible (without using $=$).

Example: Total Order $L_n = ([n], <)$

We cannot distinguish L_m, L_n in $FO[r]$ (quantifier rank r), when $m, n \geq 2^r - 1$. But we can in FO^2 (two variables).

Proposition

If $m \neq n$ then $L_m \not\equiv^2_{FO} L_n$.

Proof. Define[6] $\varphi_0(x) \stackrel{\text{def}}{=} \boldsymbol{T}$, $\varphi_{p+1}(x) \stackrel{\text{def}}{=} \exists y((x < y) \wedge \varphi_p(y))$.

$$\varphi_1(x) = \exists y(x < y) \quad \varphi_2(x) = \exists y(x < y \wedge (\exists x(y < x)))$$
$$\varphi_3(x) = \exists y(x < y \wedge (\exists x(y < x \wedge \exists y(x < y)))) \quad \ldots$$

what does $\varphi_p(x)$ say?

Let $\psi_p \stackrel{\text{def}}{=} \exists x \varphi_p(x) \wedge \neg \exists x \varphi_{p+1}(x)$. Then $L_m \models \psi_m$, $L_n \not\models \psi_m$, $\psi_m \in FO^2$.

[6] Switching x and y is a bit informal. Formally, we could set $\varphi_{p+1}(x) \stackrel{\text{def}}{=} \exists y(x < y \wedge \exists x(x = y \wedge \varphi_p(x)))$. Others ways are possible (without using =).

Example: Total Order $L_n = ([n], <)$

We cannot distinguish L_m, L_n in $FO[r]$ (quantifier rank r), when $m, n \geq 2^r - 1$. But we can in FO^2 (two variables).

Proposition

If $m \neq n$ then $L_m \not\equiv^2_{FO} L_n$.

Proof. Define[6] $\varphi_0(x) \stackrel{\text{def}}{=} \boldsymbol{T}$, $\varphi_{p+1}(x) \stackrel{\text{def}}{=} \exists y((x < y) \wedge \varphi_p(y))$.

$$\varphi_1(x) = \exists y(x < y) \quad \varphi_2(x) = \exists y(x < y \wedge (\exists x(y < x)))$$

$$\varphi_3(x) = \exists y(x < y \wedge (\exists x(y < x \wedge \exists y(x < y)))) \quad \ldots$$

what does $\varphi_p(x)$ say?

Let $\psi_p \stackrel{\text{def}}{=} \exists x \varphi_p(x) \wedge \neg \exists x \varphi_{p+1}(x)$. Then $L_m \vDash \psi_m$, $L_n \nvDash \psi_m$, $\psi_m \in FO^2$.

[6] Switching x and y is a bit informal. Formally, we could set $\varphi_{p+1}(x) \stackrel{\text{def}}{=} \exists y(x < y \wedge \exists x(x = y \wedge \varphi_p(x)))$. Others ways are possible (without using =).

Example: EVEN

- "Graph G has an EVEN number of nodes" is not expressible in $L^{\omega}_{\infty\omega}$.
  Proof. Suppose $\varphi \in L^k_{\infty\omega}$ expresses it; let[7] $G_n \stackrel{\text{def}}{=} ([n], \varnothing)$.
  Prove (in class): if $n \geq k$ then $G_n \sim^k_{\infty\omega} G_{n+1}$.
- "Graph G has an EVEN number of edges" is not expressible in $L^{\omega}_{\infty\omega}$.
  Proof. Suppose $\varphi \in L^k_{\infty\omega}$ expresses it; let[8] $K_n \stackrel{\text{def}}{=} ([n], [n] \times [n])$.
  Prove in class: if $n \geq k$ then $K_n \sim^k_{\infty\omega} K_{n+1}$.

[7] Empty graph.
[8] Complete graph.

Example: EVEN

- "Graph G has an EVEN number of nodes" is not expressible in $L^{\omega}_{\infty\omega}$.
  Proof. Suppose $\varphi \in L^k_{\infty\omega}$ expresses it; let[7] $G_n \stackrel{\text{def}}{=} ([n], \emptyset)$.
  Prove (in class): if $n \geq k$ then $G_n \sim^k_{\infty\omega} G_{n+1}$.

- "Graph G has an EVEN number of edges" is not expressible in $L^{\omega}_{\infty\omega}$.
  Proof. Suppose $\varphi \in L^k_{\infty\omega}$ expresses it; let[8] $K_n \stackrel{\text{def}}{=} ([n], [n] \times [n])$.
  Prove in class: if $n \geq k$ then $K_n \sim^k_{\infty\omega} K_{n+1}$.

---

[7] Empty graph.

[8] Complete graph.

Example: EVEN

- "Graph G has an EVEN number of nodes" is not expressible in $L^\omega_{\infty\omega}$.
  Proof. Suppose $\varphi \in L^k_{\infty\omega}$ expresses it; let[7] $G_n \stackrel{\text{def}}{=} ([n], \emptyset)$.
  Prove (in class): if $n \geq k$ then $G_n \sim^k_{\infty\omega} G_{n+1}$.

- "Graph G has an EVEN number of edges" is not expressible in $L^\omega_{\infty\omega}$.
  Proof. Suppose $\varphi \in L^k_{\infty\omega}$ expresses it; let[8] $K_n \stackrel{\text{def}}{=} ([n], [n] \times [n])$.
  Prove in class: if $n \geq k$ then $K_n \sim^k_{\infty\omega} K_{n+1}$.

[7] Empty graph.

[8] Complete graph.

Example: EVEN

- "Graph G has an EVEN number of nodes" is not expressible in $L^{\omega}_{\infty\omega}$.
  Proof. Suppose $\varphi \in L^k_{\infty\omega}$ expresses it; let[7] $G_n \stackrel{\text{def}}{=} ([n], \emptyset)$.
  Prove (in class): if $n \geq k$ then $G_n \sim^k_{\infty\omega} G_{n+1}$.

- "Graph G has an EVEN number of edges" is not expressible in $L^{\omega}_{\infty\omega}$.
  Proof. Suppose $\varphi \in L^k_{\infty\omega}$ expresses it; let[8] $K_n \stackrel{\text{def}}{=} ([n], [n] \times [n])$.
  Prove in class: if $n \geq k$ then $K_n \sim^k_{\infty\omega} K_{n+1}$.

[7] Empty graph.
[8] Complete graph.

Main Theorem of Pebble Games

1. $\boldsymbol{A} \approx^k_{\infty\omega} \boldsymbol{B}$ denotes: duplicator wins the k-pebble game.
2. $\boldsymbol{A} \equiv^k_{\infty\omega} \boldsymbol{B}$ denotes: $\boldsymbol{A} \models \varphi$ iff $\boldsymbol{B} \models \varphi$, forall $\varphi \in L^k_{\infty\omega}$
3. $\boldsymbol{A} \equiv^k_{\text{FO}} \boldsymbol{B}$ denotes: $\boldsymbol{A} \models \varphi$ iff $\boldsymbol{B} \models \varphi$, forall $\varphi \in FO^k$.

Theorem

1 and 2 are equivalent. When $\boldsymbol{A}, \boldsymbol{B}$ are finite, then all are equivalent.

We will prove:

1. $\boldsymbol{A} \approx^k_{\infty\omega} \boldsymbol{B}$ implies $\boldsymbol{A} \equiv^k_{\infty\omega} \boldsymbol{B}$.
2. $\boldsymbol{A} \equiv^k_{\infty\omega} \boldsymbol{B}$ implies $\boldsymbol{A} \equiv^k_{\text{FO}} \boldsymbol{B}$ (this is obvious!).
3. $\boldsymbol{A} \equiv^k_{\text{FO}} \boldsymbol{B}$ implies $\boldsymbol{A} \approx^k_{\infty\omega} \boldsymbol{B}$.

The proof is almost identical to the EF-games! (Good that we covered that.)

$A \approx^k_{\infty\omega} B$ implies $A \equiv^k_{\infty\omega} B$

Induction on k.

$k = 0$: same as for EF.

$k > 0$: same as for EF. We prove $A \models \varphi$ iff $B \models \varphi$ by induction[9] on φ.

- $\varphi = \exists x \psi$. If $A \models \varphi$, there is $a \in A$ s.t. $A \models \psi(a)$.
  We ask duplicator "what do you answer to a?". She says b
  Then $(A, c^A) \approx^{k-1}_{\infty\omega} (B, c^B)$ (structures with a new constant c) WHY?
  $(A, c^A) \models \psi(c) (\in L^{k-1}_{\infty\omega})$ implies $(B, c^B) \models \psi(c)$ by induction on k.
  Thus, $B \models \psi(b)$ and $B \models \exists x(\psi(x))$.
- If $\varphi = \bigvee_{i \in I} \psi_i$, then $A \models \varphi$ implies exists $i \in I$ s.t. $A \models \psi_i$.
  By induction on φ, $B \models \psi_i$, hence $B \models \varphi$.
- Etc.

[9] Transfinite induction! since $\varphi \in L^k_{\infty\omega}$

$A \approx^k_{\infty\omega} B$ implies $A \equiv^k_{\infty\omega} B$

Induction on k.

$k = 0$: same as for EF.

$k > 0$: same as for EF. We prove $\boldsymbol{A} \models \varphi$ iff $\boldsymbol{B} \models \varphi$ by induction[9] on φ.

- $\varphi = \exists x \psi$. If $\boldsymbol{A} \models \varphi$, there is $a \in A$ s.t. $\boldsymbol{A} \models \psi(a)$.
  We ask duplicator "what do you answer to a?". She says b
  Then $(\boldsymbol{A}, c^A) \approx^{k-1}_{\infty\omega} (\boldsymbol{B}, c^B)$ (structures with a new constant c) WHY?
  $(\boldsymbol{A}, c^A) \models \psi(c) (\in L^{k-1}_{\infty\omega})$ implies $(\boldsymbol{B}, c^B) \models \psi(c)$ by induction on k.
  Thus, $\boldsymbol{B} \models \psi(b)$ and $\boldsymbol{B} \models \exists x(\psi(x))$.
- If $\varphi = \bigvee_{i \in I} \psi_i$, then $\boldsymbol{A} \models \varphi$ implies exists $i \in I$ s.t. $\boldsymbol{A} \models \psi_i$.
  By induction on φ, $\boldsymbol{B} \models \psi_i$, hence $\boldsymbol{B} \models \varphi$.
- Etc.

[9] Transfinite induction! since $\varphi \in L^k_{\infty\omega}$

$\boldsymbol{A} \approx^k_{\infty\omega} \boldsymbol{B}$ implies $\boldsymbol{A} \equiv^k_{\infty\omega} \boldsymbol{B}$

Induction on k.

$k = 0$: same as for EF.

$k > 0$: same as for EF. We prove $\boldsymbol{A} \vDash \varphi$ iff $\boldsymbol{B} \vDash \varphi$ by induction[9] on φ.

- $\varphi = \exists x \psi$. If $\boldsymbol{A} \vDash \varphi$, there is $a \in A$ s.t. $\boldsymbol{A} \vDash \psi(a)$.
  We ask duplicator "what do you answer to a?". She says b
  Then $(\boldsymbol{A}, c^A) \approx^{k-1}_{\infty\omega} (\boldsymbol{B}, c^B)$ (structures with a new constant c) WHY?
  $(\boldsymbol{A}, c^A) \vDash \psi(c) (\in L^{k-1}_{\infty\omega})$ implies $(\boldsymbol{B}, c^B) \vDash \psi(c)$ by induction on k.
  Thus, $\boldsymbol{B} \vDash \psi(b)$ and $\boldsymbol{B} \vDash \exists x(\psi(x))$.
- If $\varphi = \bigvee_{i \in I} \psi_i$, then $\boldsymbol{A} \vDash \varphi$ implies exists $i \in I$ s.t. $\boldsymbol{A} \vDash \psi_i$.
  By induction on φ, $\boldsymbol{B} \vDash \psi_i$, hence $\boldsymbol{B} \vDash \varphi$.
- Etc.

[9] Transfinite induction! since $\varphi \in L^k_{\infty\omega}$

$A \approx^k_{\infty\omega} B$ implies $A \equiv^k_{\infty\omega} B$

Induction on k.

$k = 0$: same as for EF.

$k > 0$: same as for EF. We prove $\boldsymbol{A} \vDash \varphi$ iff $\boldsymbol{B} \vDash \varphi$ by induction[9] on φ.

- $\varphi = \exists x \psi$. If $\boldsymbol{A} \vDash \varphi$, there is $a \in A$ s.t. $\boldsymbol{A} \vDash \psi(a)$.
  We ask duplicator "what do you answer to a?". She says b
  Then $(\boldsymbol{A}, c^A) \approx^{k-1}_{\infty\omega} (\boldsymbol{B}, c^B)$ (structures with a new constant c) WHY?
  $(\boldsymbol{A}, c^A) \vDash \psi(c) (\in L^{k-1}_{\infty\omega})$ implies $(\boldsymbol{B}, c^B) \vDash \psi(c)$ by induction on k.
  Thus, $\boldsymbol{B} \vDash \psi(b)$ and $\boldsymbol{B} \vDash \exists x(\psi(x))$.
- If $\varphi = \bigvee_{i \in I} \psi_i$, then $\boldsymbol{A} \vDash \varphi$ implies exists $i \in I$ s.t. $\boldsymbol{A} \vDash \psi_i$.
  By induction on φ, $\boldsymbol{B} \vDash \psi_i$, hence $\boldsymbol{B} \vDash \varphi$.
- Etc.

[9] Transfinite induction! since $\varphi \in L^k_{\infty\omega}$

$A \approx^k_{\infty\omega} B$ implies $A \equiv^k_{\infty\omega} B$

Induction on k.

$k = 0$: same as for EF.

$k > 0$: same as for EF. We prove $\boldsymbol{A} \models \varphi$ iff $\boldsymbol{B} \models \varphi$ by induction[9] on φ.

- $\varphi = \exists x\psi$. If $\boldsymbol{A} \models \varphi$, there is $a \in A$ s.t. $\boldsymbol{A} \models \psi(a)$.
  We ask duplicator "what do you answer to a?". She says b
  Then $(\boldsymbol{A}, c^A) \approx^{k-1}_{\infty\omega} (\boldsymbol{B}, c^B)$ (structures with a new constant c) WHY?
  $(\boldsymbol{A}, c^A) \models \psi(c) (\in L^{k-1}_{\infty\omega})$ implies $(\boldsymbol{B}, c^B) \models \psi(c)$ by induction on k.
  Thus, $\boldsymbol{B} \models \psi(b)$ and $\boldsymbol{B} \models \exists x(\psi(x))$.
- If $\varphi = \bigvee_{i \in I} \psi_i$, then $\boldsymbol{A} \models \varphi$ implies exists $i \in I$ s.t. $\boldsymbol{A} \models \psi_i$.
  By induction on φ, $\boldsymbol{B} \models \psi_i$, hence $\boldsymbol{B} \models \varphi$.
- Etc.

[9] Transfinite induction! since $\varphi \in L^k_{\infty\omega}$

$\boldsymbol{A} \approx^k_{\infty\omega} \boldsymbol{B}$ implies $\boldsymbol{A} \equiv^k_{\infty\omega} \boldsymbol{B}$

Induction on k.

$k = 0$: same as for EF.

$k > 0$: same as for EF. We prove $\boldsymbol{A} \vDash \varphi$ iff $\boldsymbol{B} \vDash \varphi$ by induction[9] on φ.

- $\varphi = \exists x \psi$. If $\boldsymbol{A} \vDash \varphi$, there is $a \in A$ s.t. $\boldsymbol{A} \vDash \psi(a)$.
  We ask duplicator "what do you answer to a?". She says b
  Then $(\boldsymbol{A}, c^A) \approx^{k-1}_{\infty\omega} (\boldsymbol{B}, c^B)$ (structures with a new constant c) WHY?
  $(\boldsymbol{A}, c^A) \vDash \psi(c) (\in L^{k-1}_{\infty\omega})$ implies $(\boldsymbol{B}, c^B) \vDash \psi(c)$ by induction on k.
  Thus, $\boldsymbol{B} \vDash \psi(b)$ and $\boldsymbol{B} \vDash \exists x(\psi(x))$.
- If $\varphi = \bigvee_{i \in I} \psi_i$, then $\boldsymbol{A} \vDash \varphi$ implies exists $i \in I$ s.t. $\boldsymbol{A} \vDash \psi_i$.
  By induction on φ, $\boldsymbol{B} \vDash \psi_i$, hence $\boldsymbol{B} \vDash \varphi$.
- Etc.

[9] Transfinite induction! since $\varphi \in L^k_{\infty\omega}$

$A \approx^k_{\infty\omega} B$ implies $A \equiv^k_{\infty\omega} B$

Induction on k.

$k = 0$: same as for EF.

$k > 0$: same as for EF. We prove $\boldsymbol{A} \models \varphi$ iff $\boldsymbol{B} \models \varphi$ by induction[9] on φ.

- $\varphi = \exists x \psi$. If $\boldsymbol{A} \models \varphi$, there is $a \in A$ s.t. $\boldsymbol{A} \models \psi(a)$.
  We ask duplicator "what do you answer to a?". She says b
  Then $(\boldsymbol{A}, c^A) \approx^{k-1}_{\infty\omega} (\boldsymbol{B}, c^B)$ (structures with a new constant c) WHY?
  $(\boldsymbol{A}, c^A) \models \psi(c)(\in L^{k-1}_{\infty\omega})$ implies $(\boldsymbol{B}, c^B) \models \psi(c)$ by induction on k.
  Thus, $\boldsymbol{B} \models \psi(b)$ and $\boldsymbol{B} \models \exists x(\psi(x))$.
- If $\varphi = \bigvee_{i \in I} \psi_i$, then $\boldsymbol{A} \models \varphi$ implies exists $i \in I$ s.t. $\boldsymbol{A} \models \psi_i$.
  By induction on φ, $\boldsymbol{B} \models \psi_i$, hence $\boldsymbol{B} \models \varphi$.
- Etc.

[9] Transfinite induction! since $\varphi \in L^k_{\infty\omega}$

$A \equiv^k_{\infty\omega} B$ implies $A \equiv^k_{\text{FO}} B$

(obvious)

Describing Winning Strategies

A *winning strategy* for the duplicator is precisely a set $\mathcal{I}$ of partial isomorphisms $(\boldsymbol{a}, \boldsymbol{b})$ satisfying:

Definition

$\mathcal{I}$ has the *back-and-forth* property up to k if $\mathcal{I} \neq \emptyset$ and:

- (Stronger than in EF games!) If $((a_1, \ldots, a_i), (b_1, \ldots, b_i)) \in \mathcal{I}$ then removing any pebble j still leaves them in $\mathcal{I}$:

$$((a_1, \ldots, a_{j-1}, a_{j+1}, \ldots, a_i), (b_1, \ldots, b_{j-1}, b_{j+1}, \ldots, b_i)) \in \mathcal{I}$$

- Forth: forall $i < k$ if $((a_1, \ldots, a_i), (b_1, \ldots, b_i)) \in \mathcal{I}$ then $\forall a \in A, \exists b \in B$ s.t. $((a_1, \ldots, a_i, a), (b_1, \ldots, b_i, b)) \in \mathcal{I}$
- Back: forall $i < k$ if $((a_1, \ldots, a_i), (b_1, \ldots, b_i)) \in \mathcal{I}$ then $\forall b \in B, \exists a \in A$ s.t. $((a_1, \ldots, a_i, a), (b_1, \ldots, b_i, b)) \in \mathcal{I}$

Fact: a strategy for the duplicator is precisely a set of partial isomorphisms with the back-and-forth property. Proof in class.

Describing Winning Strategies

A *winning strategy* for the duplicator is precisely a set $\mathcal{I}$ of partial isomorphisms $(\boldsymbol{a}, \boldsymbol{b})$ satisfying:

Definition

$\mathcal{I}$ has the *back-and-forth* property up to k if $\mathcal{I} \neq \emptyset$ and:

- (Stronger than in EF games!) If $((a_1, \ldots, a_i), (b_1, \ldots, b_i)) \in \mathcal{I}$ then removing any pebble j still leaves them in $\mathcal{I}$:

$$((a_1, \ldots, a_{j-1}, a_{j+1}, \ldots, a_i), (b_1, \ldots, b_{j-1}, b_{j+1}, \ldots, b_i)) \in \mathcal{I}$$

- Forth: forall $i < k$ if $((a_1, \ldots, a_i), (b_1, \ldots, b_i)) \in \mathcal{I}$ then $\forall a \in A, \exists b \in B$ s.t. $((a_1, \ldots, a_i, a), (b_1, \ldots, b_i, b)) \in \mathcal{I}$
- Back: forall $i < k$ if $((a_1, \ldots, a_i), (b_1, \ldots, b_i)) \in \mathcal{I}$ then $\forall b \in B, \exists a \in A$ s.t. $((a_1, \ldots, a_i, a), (b_1, \ldots, b_i, b)) \in \mathcal{I}$

Fact: a strategy for the duplicator is precisely a set of partial isomorphisms with the back-and-forth property. Proof in class.

Types

Fix k and m.

Definition

Fix $\boldsymbol{A}$ and $\boldsymbol{a} = (a_1, \ldots, a_m) \in A^m$. The $L^k_{\infty\omega}$ and the FO^k types are:

$$\mathrm{tp}^k_{\infty\omega}(\boldsymbol{A}, \boldsymbol{a}) = \{\varphi(x_1, \ldots, x_m) \in L^k_{\infty\omega} \mid \boldsymbol{A} \vDash \varphi(a_1, \ldots, a_m)\}$$
$$\mathrm{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \{\varphi(x_1, \ldots, x_m) \in FO^k \mid \boldsymbol{A} \vDash \varphi(a_1, \ldots, a_m)\}$$

Facts:

- Both sets are complete same as for EF
- There are infinitely many types of both kinds different from EF
- The pebble-games theorem implies: on finite structures, $\mathrm{tp}^k_{\infty\omega}(\boldsymbol{A}, \boldsymbol{a}) = \mathrm{tp}^k_{\infty\omega}(\boldsymbol{B}, \boldsymbol{b})$ iff $\mathrm{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \mathrm{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b})$ surprising!

Types

Fix k and m.

Definition

Fix $\boldsymbol{A}$ and $\boldsymbol{a} = (a_1, \ldots, a_m) \in A^m$. The $L^k_{\infty\omega}$ and the FO^k types are:

$$\text{tp}^k_{\infty\omega}(\boldsymbol{A}, \boldsymbol{a}) = \{\varphi(x_1, \ldots, x_m) \in L^k_{\infty\omega} \mid \boldsymbol{A} \models \varphi(a_1, \ldots, a_m)\}$$

$$\text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \{\varphi(x_1, \ldots, x_m) \in FO^k \mid \boldsymbol{A} \models \varphi(a_1, \ldots, a_m)\}$$

Facts:

- Both sets are complete same as for EF
- There are infinitely many types of both kinds different from EF
- The pebble-games theorem implies: on finite structures, $\text{tp}^k_{\infty\omega}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}^k_{\infty\omega}(\boldsymbol{B}, \boldsymbol{b})$ iff $\text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b})$ surprising!

$A \equiv^k_{\mathrm{FO}} B$ implies $A \approx^k_{\infty\omega} B$

Define $\mathcal{I} = \{(\boldsymbol{a}, \boldsymbol{b}) \mid |\boldsymbol{a}| = |\boldsymbol{b}| \leq k, \mathrm{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \mathrm{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b})\}$

Then $((), ()) \in \mathcal{I}$ same as for EF hence $\mathcal{I} \neq \emptyset$.

Removing pebbles: Suppose $\mathrm{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \mathrm{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b})$.
Let $\boldsymbol{a}', \boldsymbol{b}'$ be $\boldsymbol{a}, \boldsymbol{b}$ witout position j: then $\mathrm{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}') = \mathrm{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b}')$
why? Because a formula $\varphi(x_1, \ldots, x_i)$ does not need to use x_j.

Forth: Suppose $\mathrm{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \mathrm{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b})$, $|\boldsymbol{a}| = |\boldsymbol{b}| < k$. Let $a \in A$.
Claim: $\exists b \in B$ s.t. $\mathrm{tp}^k_{FO}(\boldsymbol{A}, (\boldsymbol{a}, a)) = \mathrm{tp}^k_{FO}(\boldsymbol{B}, (\boldsymbol{b}, b))$. Otherwise:

$\forall b \in B, \exists \varphi_b(x_1, \ldots, x_i, y) \in FO^k$ s.t. $\quad \boldsymbol{A} \models \varphi_b(\boldsymbol{a}, a) \quad \boldsymbol{B} \not\models \varphi_b(\boldsymbol{b}, b)$

$\forall b \in B, \quad \boldsymbol{A} \models \bigwedge_{b' \in B} \varphi_{b'}(\boldsymbol{a}, a) \quad \boldsymbol{B} \not\models \bigwedge_{b' \in B} \varphi_{b'}(\boldsymbol{b}, b)$

$\psi \stackrel{\text{def}}{=} \exists y \bigwedge_{b' \in B} \varphi_{b'}(x_1, \ldots, x_i, y)$ then $\quad \boldsymbol{A} \models \psi(\boldsymbol{a}) \quad \boldsymbol{B} \not\models \psi(\boldsymbol{b})$

$\psi \in L^k_{\infty\omega}$ or $\in FO^k$ when $\boldsymbol{B}$ is finite. Contradicts $\mathrm{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \mathrm{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b})$.

Back property: Similar.

$A \equiv_{FO}^k B$ implies $A \approx_{\infty\omega}^k B$

Define $\mathcal{I} = \{(\boldsymbol{a}, \boldsymbol{b}) \mid |\boldsymbol{a}| = |\boldsymbol{b}| \leq k, \text{tp}_{FO}^k(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}_{FO}^k(\boldsymbol{B}, \boldsymbol{b})\}$

Then $((), ()) \in \mathcal{I}$ same as for EF hence $\mathcal{I} \neq \varnothing$.

Removing pebbles: Suppose $\text{tp}_{FO}^k(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}_{FO}^k(\boldsymbol{B}, \boldsymbol{b})$.
Let $\boldsymbol{a}', \boldsymbol{b}'$ be $\boldsymbol{a}, \boldsymbol{b}$ witout position j: then $\text{tp}_{FO}^k(\boldsymbol{A}, \boldsymbol{a}') = \text{tp}_{FO}^k(\boldsymbol{B}, \boldsymbol{b}')$
why? Because a formula $\varphi(x_1, \ldots, x_i)$ does not need to use x_j.

Forth: Suppose $\text{tp}_{FO}^k(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}_{FO}^k(\boldsymbol{B}, \boldsymbol{b})$, $|\boldsymbol{a}| = |\boldsymbol{b}| < k$. Let $a \in A$.
Claim: $\exists b \in B$ s.t. $\text{tp}_{FO}^k(\boldsymbol{A}, (\boldsymbol{a}, a)) = \text{tp}_{FO}^k(\boldsymbol{B}, (\boldsymbol{b}, b))$. Otherwise:

$\forall b \in B, \exists \varphi_b(x_1, \ldots, x_i, y) \in FO^k$ s.t. $\boldsymbol{A} \models \varphi_b(\boldsymbol{a}, a)$ $\quad \boldsymbol{B} \not\models \varphi_b(\boldsymbol{b}, b)$

$\forall b \in B,$ $\quad \boldsymbol{A} \models \bigwedge_{b' \in B} \varphi_{b'}(\boldsymbol{a}, a)$ $\quad \boldsymbol{B} \not\models \bigwedge_{b' \in B} \varphi_{b'}(\boldsymbol{b}, b)$

$\psi \stackrel{\text{def}}{=} \exists y \bigwedge_{b' \in B} \varphi_{b'}(x_1, \ldots, x_i, y)$ then $\quad \boldsymbol{A} \models \psi(\boldsymbol{a})$ $\quad \boldsymbol{B} \not\models \psi(\boldsymbol{b})$

$\psi \in L_{\infty\omega}^k$ or $\in FO^k$ when $\boldsymbol{B}$ is finite. Contradicts $\text{tp}_{FO}^k(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}_{FO}^k(\boldsymbol{B}, \boldsymbol{b})$.

Back property: Similar.

$A \equiv^k_{\text{FO}} B$ implies $A \approx^k_{\infty\omega} B$

Define $\mathcal{I} = \{(\boldsymbol{a}, \boldsymbol{b}) \mid |\boldsymbol{a}| = |\boldsymbol{b}| \le k, \text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b})\}$

Then $((), ()) \in \mathcal{I}$ same as for EF hence $\mathcal{I} \neq \emptyset$.

Removing pebbles: Suppose $\text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b})$.

Let $\boldsymbol{a}', \boldsymbol{b}'$ be $\boldsymbol{a}, \boldsymbol{b}$ witout position j: then $\text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}') = \text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b}')$

why? Because a formula $\varphi(x_1, \ldots, x_i)$ does not need to use x_j.

Forth: Suppose $\text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b})$, $|\boldsymbol{a}| = |\boldsymbol{b}| < k$. Let $a \in A$.

Claim: $\exists b \in B$ s.t. $\text{tp}^k_{FO}(\boldsymbol{A}, (\boldsymbol{a}, a)) = \text{tp}^k_{FO}(\boldsymbol{B}, (\boldsymbol{b}, b))$. Otherwise:

$\forall b \in B, \exists \varphi_b(x_1, \ldots, x_i, y) \in FO^k$ s.t. $\quad \boldsymbol{A} \models \varphi_b(\boldsymbol{a}, a) \quad \boldsymbol{B} \not\models \varphi_b(\boldsymbol{b}, b)$

$\forall b \in B, \quad \boldsymbol{A} \models \bigwedge_{b' \in B} \varphi_{b'}(\boldsymbol{a}, a) \quad \boldsymbol{B} \not\models \bigwedge_{b' \in B} \varphi_{b'}(\boldsymbol{b}, b)$

$\psi \stackrel{\text{def}}{=} \exists y \bigwedge_{b' \in B} \varphi_{b'}(x_1, \ldots, x_i, y)$ then $\quad \boldsymbol{A} \models \psi(\boldsymbol{a}) \quad \boldsymbol{B} \not\models \psi(\boldsymbol{b})$

$\psi \in L^k_{\infty\omega}$ or $\in FO^k$ when $\boldsymbol{B}$ is finite. Contradicts $\text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b})$.

Back property: Similar.

$\boldsymbol{A} \equiv^k_{\text{FO}} \boldsymbol{B}$ implies $\boldsymbol{A} \approx^k_{\infty\omega} \boldsymbol{B}$

Define $\mathcal{I} = \{(\boldsymbol{a}, \boldsymbol{b}) \mid |\boldsymbol{a}| = |\boldsymbol{b}| \leq k, \text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b})\}$

Then $((), ()) \in \mathcal{I}$ same as for EF hence $\mathcal{I} \neq \emptyset$.

Removing pebbles: Suppose $\text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b})$.

Let $\boldsymbol{a}', \boldsymbol{b}'$ be $\boldsymbol{a}, \boldsymbol{b}$ witout position j: then $\text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}') = \text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b}')$

why? Because a formula $\varphi(x_1, \ldots, x_i)$ does not need to use x_j.

Forth: Suppose $\text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b})$, $|\boldsymbol{a}| = |\boldsymbol{b}| < k$. Let $a \in A$.

Claim: $\exists b \in B$ s.t. $\text{tp}^k_{FO}(\boldsymbol{A}, (\boldsymbol{a}, a)) = \text{tp}^k_{FO}(\boldsymbol{B}, (\boldsymbol{b}, b))$. Otherwise:

$\forall b \in B, \exists \varphi_b(x_1, \ldots, x_i, y) \in FO^k$ s.t. $\quad \boldsymbol{A} \models \varphi_b(\boldsymbol{a}, a) \quad \boldsymbol{B} \not\models \varphi_b(\boldsymbol{b}, b)$

$\forall b \in B, \quad \boldsymbol{A} \models \bigwedge_{b' \in B} \varphi_{b'}(\boldsymbol{a}, a) \quad \boldsymbol{B} \not\models \bigwedge_{b' \in B} \varphi_{b'}(\boldsymbol{b}, b)$

$\psi \stackrel{\text{def}}{=} \exists y \bigwedge_{b' \in B} \varphi_{b'}(x_1, \ldots, x_i, y)$ then $\quad \boldsymbol{A} \models \psi(\boldsymbol{a}) \quad \boldsymbol{B} \not\models \psi(\boldsymbol{b})$

$\psi \in L^k_{\infty\omega}$ or $\in FO^k$ when $\boldsymbol{B}$ is finite. Contradicts $\text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b})$.

Back property: Similar.

$\boldsymbol{A} \equiv^k_{\text{FO}} \boldsymbol{B}$ implies $\boldsymbol{A} \approx^k_{\infty\omega} \boldsymbol{B}$

Define $\mathcal{I} = \{(\boldsymbol{a}, \boldsymbol{b}) \mid |\boldsymbol{a}| = |\boldsymbol{b}| \leq k, \text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b})\}$

Then $((), ()) \in \mathcal{I}$ same as for EF hence $\mathcal{I} \neq \varnothing$.

Removing pebbles: Suppose $\text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b})$.

Let $\boldsymbol{a}', \boldsymbol{b}'$ be $\boldsymbol{a}, \boldsymbol{b}$ witout position j: then $\text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}') = \text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b}')$

why? Because a formula $\varphi(x_1, \ldots, x_i)$ does not need to use x_j.

Forth: Suppose $\text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b})$, $|\boldsymbol{a}| = |\boldsymbol{b}| < k$. Let $a \in A$.

Claim: $\exists b \in B$ s.t. $\text{tp}^k_{FO}(\boldsymbol{A}, (\boldsymbol{a}, a)) = \text{tp}^k_{FO}(\boldsymbol{B}, (\boldsymbol{b}, b))$. Otherwise:

$\forall b \in B, \exists \varphi_b(x_1, \ldots, x_i, y) \in FO^k$ s.t. $\quad \boldsymbol{A} \models \varphi_b(\boldsymbol{a}, a) \quad \boldsymbol{B} \not\models \varphi_b(\boldsymbol{b}, b)$

$\forall b \in B, \quad \boldsymbol{A} \models \bigwedge_{b' \in B} \varphi_{b'}(\boldsymbol{a}, a) \quad \boldsymbol{B} \not\models \bigwedge_{b' \in B} \varphi_{b'}(\boldsymbol{b}, b)$

$\psi \stackrel{\text{def}}{=} \exists y \bigwedge_{b' \in B} \varphi_{b'}(x_1, \ldots, x_i, y)$ then $\quad \boldsymbol{A} \models \psi(\boldsymbol{a}) \quad \boldsymbol{B} \not\models \psi(\boldsymbol{b})$

$\psi \in L^k_{\infty\omega}$ or $\in FO^k$ when $\boldsymbol{B}$ is finite. Contradicts $\text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b})$.

Back property: Similar.

$A \equiv^k_{FO} B$ implies $A \approx^k_{\infty\omega} B$

Define $\mathcal{I} = \{(\boldsymbol{a}, \boldsymbol{b}) \mid |\boldsymbol{a}| = |\boldsymbol{b}| \leq k, \text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b})\}$

Then $((),()) \in \mathcal{I}$ same as for EF hence $\mathcal{I} \neq \emptyset$.

Removing pebbles: Suppose $\text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b})$.
Let $\boldsymbol{a}', \boldsymbol{b}'$ be $\boldsymbol{a}, \boldsymbol{b}$ witout position j: then $\text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}') = \text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b}')$
why? Because a formula $\varphi(x_1, \ldots, x_i)$ does not need to use x_j.

Forth: Suppose $\text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b})$, $|\boldsymbol{a}| = |\boldsymbol{b}| < k$. Let $a \in A$.
Claim: $\exists b \in B$ s.t. $\text{tp}^k_{FO}(\boldsymbol{A}, (\boldsymbol{a}, a)) = \text{tp}^k_{FO}(\boldsymbol{B}, (\boldsymbol{b}, b))$. Otherwise:

$\forall b \in B, \exists \varphi_b(x_1, \ldots, x_i, y) \in FO^k$ s.t. $\boldsymbol{A} \models \varphi_b(\boldsymbol{a}, a)$ $\quad \boldsymbol{B} \not\models \varphi_b(\boldsymbol{b}, b)$

$\forall b \in B,$ $\quad \boldsymbol{A} \models \bigwedge_{b' \in B} \varphi_{b'}(\boldsymbol{a}, a)$ $\quad \boldsymbol{B} \not\models \bigwedge_{b' \in B} \varphi_{b'}(\boldsymbol{b}, b)$

$\psi \overset{\text{def}}{=} \exists y \bigwedge_{b' \in B} \varphi_{b'}(x_1, \ldots, x_i, y)$ then $\quad \boldsymbol{A} \models \psi(\boldsymbol{a})$ $\quad \boldsymbol{B} \not\models \psi(\boldsymbol{b})$

$\psi \in L^k_{\infty\omega}$ or $\in FO^k$ when $\boldsymbol{B}$ is finite. Contradicts $\text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b})$.

Back property: Similar.

$\boldsymbol{A} \equiv^k_{\text{FO}} \boldsymbol{B}$ implies $\boldsymbol{A} \approx^k_{\infty\omega} \boldsymbol{B}$

Define $\mathcal{I} = \{(\boldsymbol{a}, \boldsymbol{b}) \mid |\boldsymbol{a}| = |\boldsymbol{b}| \leq k, \text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b})\}$

Then $((), ()) \in \mathcal{I}$ same as for EF hence $\mathcal{I} \neq \emptyset$.

Removing pebbles: Suppose $\text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b})$.
Let $\boldsymbol{a}', \boldsymbol{b}'$ be $\boldsymbol{a}, \boldsymbol{b}$ witout position j: then $\text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}') = \text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b}')$
why? Because a formula $\varphi(x_1, \ldots, x_i)$ does not need to use x_j.

Forth: Suppose $\text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b})$, $|\boldsymbol{a}| = |\boldsymbol{b}| < k$. Let $a \in A$.
Claim: $\exists b \in B$ s.t. $\text{tp}^k_{FO}(\boldsymbol{A}, (\boldsymbol{a}, a)) = \text{tp}^k_{FO}(\boldsymbol{B}, (\boldsymbol{b}, b))$. Otherwise:

$\forall b \in B, \exists \varphi_b(x_1, \ldots, x_i, y) \in FO^k$ s.t. $\quad \boldsymbol{A} \models \varphi_b(\boldsymbol{a}, a) \quad \boldsymbol{B} \not\models \varphi_b(\boldsymbol{b}, b)$

$\forall b \in B, \quad \boldsymbol{A} \models \bigwedge_{b' \in B} \varphi_{b'}(\boldsymbol{a}, a) \quad \boldsymbol{B} \not\models \bigwedge_{b' \in B} \varphi_{b'}(\boldsymbol{b}, b)$

$\psi \stackrel{\text{def}}{=} \exists y \bigwedge_{b' \in B} \varphi_{b'}(x_1, \ldots, x_i, y)$ then $\quad \boldsymbol{A} \models \psi(\boldsymbol{a}) \quad \boldsymbol{B} \not\models \psi(\boldsymbol{b})$

$\psi \in L^k_{\infty\omega}$ or $\in FO^k$ when $\boldsymbol{B}$ is finite. Contradicts $\text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b})$.

Back property: Similar.

$\boldsymbol{A} \equiv^k_{FO} \boldsymbol{B}$ implies $\boldsymbol{A} \approx^k_{\infty\omega} \boldsymbol{B}$

Define $\mathcal{I} = \{(\boldsymbol{a}, \boldsymbol{b}) \mid |\boldsymbol{a}| = |\boldsymbol{b}| \leq k, \text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b})\}$

Then $((),()) \in \mathcal{I}$ same as for EF hence $\mathcal{I} \neq \emptyset$.

Removing pebbles: Suppose $\text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b})$.
Let $\boldsymbol{a}', \boldsymbol{b}'$ be $\boldsymbol{a}, \boldsymbol{b}$ witout position j: then $\text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}') = \text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b}')$
why? Because a formula $\varphi(x_1, \ldots, x_i)$ does not need to use x_j.

Forth: Suppose $\text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b})$, $|\boldsymbol{a}| = |\boldsymbol{b}| < k$. Let $a \in A$.
Claim: $\exists b \in B$ s.t. $\text{tp}^k_{FO}(\boldsymbol{A}, (\boldsymbol{a}, a)) = \text{tp}^k_{FO}(\boldsymbol{B}, (\boldsymbol{b}, b))$. Otherwise:

$\forall b \in B, \exists \varphi_b(x_1, \ldots, x_i, y) \in FO^k$ s.t. $\quad \boldsymbol{A} \models \varphi_b(\boldsymbol{a}, a) \quad \boldsymbol{B} \not\models \varphi_b(\boldsymbol{b}, b)$

$\forall b \in B, \quad \boldsymbol{A} \models \bigwedge_{b' \in B} \varphi_{b'}(\boldsymbol{a}, a) \quad \boldsymbol{B} \not\models \bigwedge_{b' \in B} \varphi_{b'}(\boldsymbol{b}, b)$

$\psi \stackrel{\text{def}}{=} \exists y \bigwedge_{b' \in B} \varphi_{b'}(x_1, \ldots, x_i, y)$ then $\quad \boldsymbol{A} \models \psi(\boldsymbol{a}) \quad \boldsymbol{B} \not\models \psi(\boldsymbol{b})$

$\psi \in L^k_{\infty\omega}$ or $\in FO^k$ when $\boldsymbol{B}$ is finite. Contradicts $\text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b})$.

Back property: Similar.

Discussion

- If two finite structures can be distinguished by $L^k_{\infty\omega}$, then they can already be distinguished by FO^k.
- Positions in the pebble game are captured by FO^k-types, which are the same as $L^k_{\infty\omega}$ types.
- Don't confuse FO^k m-types tp^k_{FO} with rank r m-types $\text{tp}_{r,m}$, which refer to $FO[r]$. (Notation sucks.)
- Every type $\text{tp}_{r,m}$ contains a finite number of formulas: hence their conjunction is a formula that fully characterizes the type.
- Every type tp^k_{FO} has infinitely many formulas. Still, we will prove (next) that each type is fully described by one formula in FO^k.

FO^k-Type Formula

Recall: an FO^k m-type is:
$\text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) \stackrel{\text{def}}{=} \{\varphi(x_1, \ldots, x_m) \in FO^k \mid \boldsymbol{A} \vDash \varphi(a_1, \ldots, a_m)\}$.

Theorem

For every FO^k type m-type τ, there exist a formula $\psi^\tau \in FO^k$ s.t., for any finite structure $\boldsymbol{A}$, $(\boldsymbol{A}, \boldsymbol{a}) \vDash \psi^\tau$ iff $tp^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \tau$.

If τ were finite, then could take $\psi^\tau = \bigwedge_{\varphi \in \tau} \varphi$
But τ is finite, and the proof is much more subtle.

Before the proof, an application.

Application: Normal Form for $L^k_{\infty\omega}$

Corollary

Let $\varphi \in L^k_{\infty\omega}$. Then there exists a sequence of formulas $\psi_i \in FO^k$, $i = 1, 2, \ldots$ s.t. $\varphi \equiv_{fin} \psi_1 \vee \psi_2 \vee \psi_3 \vee \cdots$

In other words, only one single countable $\bigvee$ suffices to capture $L^k_{\infty\omega}$.

Proof Let $(\boldsymbol{A}_i, \boldsymbol{a}_i)$, $i = 1, 2, 3, \ldots$ be all finite structures s.t. $\boldsymbol{A}_i \models \varphi(\boldsymbol{a}_i)$
why only countably many?

Let $\tau_i = \text{tp}^k_{FO}(\boldsymbol{A}_i, \boldsymbol{a}_i)$. Notice: $\varphi \in \tau_i$ forall i.

Claim: $\varphi \equiv_{\text{fin}} \bigvee_i \psi^{\tau_i}$.
(1) if $\boldsymbol{B} \models \varphi(\boldsymbol{b})$ then $\exists i$ s.t. $(\boldsymbol{B}, \boldsymbol{b}) = (\boldsymbol{A}_i, \boldsymbol{a}_i)$, hence $\boldsymbol{B} \models \psi^{\tau_i}(\boldsymbol{b})$.
(2) if $\boldsymbol{B} \models \bigvee_i \psi^{\tau_i}(\boldsymbol{b})$ then $\exists i$ s.t. $\boldsymbol{B} \models \psi^{\tau_i}(\boldsymbol{b})$,
hence, by the Theorem, $\text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b}) = \text{tp}^k_{FO}(\boldsymbol{A}_i, \boldsymbol{a}_i)$,
hence $\varphi \in \text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b})$, hence $\boldsymbol{B} \models \varphi(\boldsymbol{b})$.

Application: Normal Form for $L^k_{\infty\omega}$

Corollary

Let $\varphi \in L^k_{\infty\omega}$. Then there exists a sequence of formulas $\psi_i \in FO^k$, $i = 1, 2, \ldots$ s.t. $\varphi \equiv_{fin} \psi_1 \vee \psi_2 \vee \psi_3 \vee \cdots$

In other words, only one single countable $\bigvee$ suffices to capture $L^k_{\infty\omega}$.

Proof Let $(\boldsymbol{A}_i, \boldsymbol{a}_i)$, $i = 1, 2, 3, \ldots$ be all finite structures s.t. $\boldsymbol{A}_i \models \varphi(\boldsymbol{a}_i)$

why only countably many?

Let $\tau_i = \text{tp}^k_{FO}(\boldsymbol{A}_i, \boldsymbol{a}_i)$. Notice: $\varphi \in \tau_i$ forall i.

Claim: $\varphi \equiv_{\text{fin}} \bigvee_i \psi^{\tau_i}$.

(1) if $\boldsymbol{B} \models \varphi(\boldsymbol{b})$ then $\exists i$ s.t. $(\boldsymbol{B}, \boldsymbol{b}) = (\boldsymbol{A}_i, \boldsymbol{a}_i)$, hence $\boldsymbol{B} \models \psi^{\tau_i}(\boldsymbol{b})$.

(2) if $\boldsymbol{B} \models \bigvee_i \psi^{\tau_i}(\boldsymbol{b})$ then $\exists i$ s.t. $\boldsymbol{B} \models \psi^{\tau_i}(\boldsymbol{b})$,
hence, by the Theorem, $\text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b}) = \text{tp}^k_{FO}(\boldsymbol{A}_i, \boldsymbol{a}_i)$,
hence $\varphi \in \text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b})$, hence $\boldsymbol{B} \models \varphi(\boldsymbol{b})$.

Application: Normal Form for $L^k_{\infty\omega}$

Corollary

Let $\varphi \in L^k_{\infty\omega}$. Then there exists a sequence of formulas $\psi_i \in FO^k$, $i = 1, 2, \ldots$ s.t. $\varphi \equiv_{fin} \psi_1 \vee \psi_2 \vee \psi_3 \vee \cdots$

In other words, only one single countable $\bigvee$ suffices to capture $L^k_{\infty\omega}$.

Proof Let $(\boldsymbol{A}_i, \boldsymbol{a}_i)$, $i = 1, 2, 3, \ldots$ be all finite structures s.t. $\boldsymbol{A}_i \vDash \varphi(\boldsymbol{a}_i)$
why only countably many?

Let $\tau_i = \text{tp}^k_{FO}(\boldsymbol{A}_i, \boldsymbol{a}_i)$. Notice: $\varphi \in \tau_i$ forall i.

Claim: $\varphi \equiv_{\text{fin}} \bigvee_i \psi^{\tau_i}$.
(1) if $\boldsymbol{B} \vDash \varphi(\boldsymbol{b})$ then $\exists i$ s.t. $(\boldsymbol{B}, \boldsymbol{b}) = (\boldsymbol{A}_i, \boldsymbol{a}_i)$, hence $\boldsymbol{B} \vDash \psi^{\tau_i}(\boldsymbol{b})$.
(2) if $\boldsymbol{B} \vDash \bigvee_i \psi^{\tau_i}(\boldsymbol{b})$ then $\exists i$ s.t. $\boldsymbol{B} \vDash \psi^{\tau_i}(\boldsymbol{b})$,
hence, by the Theorem, $\text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b}) = \text{tp}^k_{FO}(\boldsymbol{A}_i, \boldsymbol{a}_i)$,
hence $\varphi \in \text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b})$, hence $\boldsymbol{B} \vDash \varphi(\boldsymbol{b})$.

Application: Normal Form for $L^k_{\infty\omega}$

Corollary

Let $\varphi \in L^k_{\infty\omega}$. Then there exists a sequence of formulas $\psi_i \in FO^k$, $i = 1, 2, \ldots$ s.t. $\varphi \equiv_{fin} \psi_1 \vee \psi_2 \vee \psi_3 \vee \cdots$

In other words, only one single countable $\bigvee$ suffices to capture $L^k_{\infty\omega}$.

Proof Let $(\boldsymbol{A}_i, \boldsymbol{a}_i)$, $i = 1, 2, 3, \ldots$ be all finite structures s.t. $\boldsymbol{A}_i \vDash \varphi(\boldsymbol{a}_i)$

why only countably many?

Let $\tau_i = \text{tp}^k_{FO}(\boldsymbol{A}_i, \boldsymbol{a}_i)$. Notice: $\varphi \in \tau_i$ forall i.

Claim: $\varphi \equiv_{\text{fin}} \bigvee_i \psi^{\tau_i}$.

(1) if $\boldsymbol{B} \vDash \varphi(\boldsymbol{b})$ then $\exists i$ s.t. $(\boldsymbol{B}, \boldsymbol{b}) = (\boldsymbol{A}_i, \boldsymbol{a}_i)$, hence $\boldsymbol{B} \vDash \psi^{\tau_i}(\boldsymbol{b})$.

(2) if $\boldsymbol{B} \vDash \bigvee_i \psi^{\tau_i}(\boldsymbol{b})$ then $\exists i$ s.t. $\boldsymbol{B} \vDash \psi^{\tau_i}(\boldsymbol{b})$,

hence, by the Theorem, $\text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b}) = \text{tp}^k_{FO}(\boldsymbol{A}_i, \boldsymbol{a}_i)$,

hence $\varphi \in \text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b})$, hence $\boldsymbol{B} \vDash \varphi(\boldsymbol{b})$.

Application: Normal Form for $L^k_{\infty\omega}$

Corollary

Let $\varphi \in L^k_{\infty\omega}$. Then there exists a sequence of formulas $\psi_i \in FO^k$, $i = 1, 2, \ldots$ s.t. $\varphi \equiv_{fin} \psi_1 \vee \psi_2 \vee \psi_3 \vee \cdots$

In other words, only one single countable $\bigvee$ suffices to capture $L^k_{\infty\omega}$.

Proof Let $(\boldsymbol{A}_i, \boldsymbol{a}_i)$, $i = 1, 2, 3, \ldots$ be all finite structures s.t. $\boldsymbol{A}_i \vDash \varphi(\boldsymbol{a}_i)$
why only countably many?

Let $\tau_i = \text{tp}^k_{FO}(\boldsymbol{A}_i, \boldsymbol{a}_i)$. Notice: $\varphi \in \tau_i$ forall i.

Claim: $\varphi \equiv_{\text{fin}} \bigvee_i \psi^{\tau_i}$.

(1) if $\boldsymbol{B} \vDash \varphi(\boldsymbol{b})$ then $\exists i$ s.t. $(\boldsymbol{B}, \boldsymbol{b}) = (\boldsymbol{A}_i, \boldsymbol{a}_i)$, hence $\boldsymbol{B} \vDash \psi^{\tau_i}(\boldsymbol{b})$.

(2) if $\boldsymbol{B} \vDash \bigvee_i \psi^{\tau_i}(\boldsymbol{b})$ then $\exists i$ s.t. $\boldsymbol{B} \vDash \psi^{\tau_i}(\boldsymbol{b})$,
hence, by the Theorem, $\text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b}) = \text{tp}^k_{FO}(\boldsymbol{A}_i, \boldsymbol{a}_i)$,
hence $\varphi \in \text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b})$, hence $\boldsymbol{B} \vDash \varphi(\boldsymbol{b})$.

Application: Normal Form for $L^k_{\infty\omega}$

Corollary

Let $\varphi \in L^k_{\infty\omega}$. Then there exists a sequence of formulas $\psi_i \in FO^k$, $i = 1, 2, \ldots$ s.t. $\varphi \equiv_{fin} \psi_1 \vee \psi_2 \vee \psi_3 \vee \cdots$

In other words, only one single countable $\bigvee$ suffices to capture $L^k_{\infty\omega}$.

Proof Let $(\boldsymbol{A}_i, \boldsymbol{a}_i)$, $i = 1, 2, 3, \ldots$ be all finite structures s.t. $\boldsymbol{A}_i \vDash \varphi(\boldsymbol{a}_i)$
why only countably many?

Let $\tau_i = \text{tp}^k_{FO}(\boldsymbol{A}_i, \boldsymbol{a}_i)$. Notice: $\varphi \in \tau_i$ forall i.

Claim: $\varphi \equiv_{\text{fin}} \bigvee_i \psi^{\tau_i}$.
(1) if $\boldsymbol{B} \vDash \varphi(\boldsymbol{b})$ then $\exists i$ s.t. $(\boldsymbol{B}, \boldsymbol{b}) = (\boldsymbol{A}_i, \boldsymbol{a}_i)$, hence $\boldsymbol{B} \vDash \psi^{\tau_i}(\boldsymbol{b})$.
(2) if $\boldsymbol{B} \vDash \bigvee_i \psi^{\tau_i}(\boldsymbol{b})$ then $\exists i$ s.t. $\boldsymbol{B} \vDash \psi^{\tau_i}(\boldsymbol{b})$,
hence, by the Theorem, $\text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b}) = \text{tp}^k_{FO}(\boldsymbol{A}_i, \boldsymbol{a}_i)$,
hence $\varphi \in \text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b})$, hence $\boldsymbol{B} \vDash \varphi(\boldsymbol{b})$.

Discussion

- Theorem says: every FO^k type τ, is described (on finite structures) by one formula $\psi^\tau \in FO^k$.
- If we restricted the quantifier rank, then τ is finite and we take $\psi^\tau = \bigwedge_{\varphi \in \tau} \varphi$.
- But quantifier rank of formulas in τ is unbounded (and τ is infinite).
- Yet τ is described by one formula, with some fixed quantifier rank. What is $qr(\psi^\tau)$?
  (How do we get from the infinite τ a finite bound for $qr(\psi^\tau)$?)

Discussion

- Theorem says: every FO^k type τ, is described (on finite structures) by one formula $\psi^\tau \in FO^k$.
- If we restricted the quantifier rank, then τ is finite and we take $\psi^\tau = \bigwedge_{\varphi \in \tau} \varphi$.
- But quantifier rank of formulas in τ is unbounded (and τ is infinite).
- Yet τ is described by one formula, with some fixed quantifier rank. What is $qr(\psi^\tau)$?
  (How do we get from the infinite τ a finite bound for $qr(\psi^\tau)$?)
- Answer: we assume τ is satisfied by some *finite structure* $(\boldsymbol{B}, \boldsymbol{b})$; this will give us the desired finite rank.
- If τ is not satisfiable in the finite, then simply take $\psi^\tau = \boldsymbol{F}$.
  We assume $\boldsymbol{F}$ is an FO^k type.

FO^k-Type Formula

Theorem

For every FO^k type m-type τ, there exist a formula $\psi^\tau \in FO^k$ s.t., for any finite structure $\boldsymbol{A}$, $(\boldsymbol{A}, \boldsymbol{a}) \models \psi^\tau$ iff $tp^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \tau$.

Proof plan. Fix a structure $(\boldsymbol{B}, \boldsymbol{b})$ s.t. $\tau = \text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b})$.

- Types of quantifier-rank $r = 1, 2, 3, \ldots$ reach a fixpoint on $\boldsymbol{B}$ for $r = R$.
- Then $\psi^\tau(\boldsymbol{x})$ will says two things:
  1. $\text{TYPE}_R(\boldsymbol{x})$: "$\boldsymbol{x}$ has the R, m-type of $(\boldsymbol{B}, \boldsymbol{b})$" and,
  2. DONE_R: "<u>every</u> $R + 1, m$-type is some R, m type"

Defining $\text{TYPE}_R(\boldsymbol{x})$

For each quantifier rank r, there are finitely many, say n_r, types.
Each is described by one formula: $\varphi_{1,r}, \varphi_{2,r}, \ldots, \ldots, \varphi_{n_r,r} \in FO^k[r]$.
(Note: every $\varphi \in FO^k[r]$ is a union of types $\varphi = \bigvee_i \varphi_{i,r}$.)

[10] Some equivalence classes are empty.

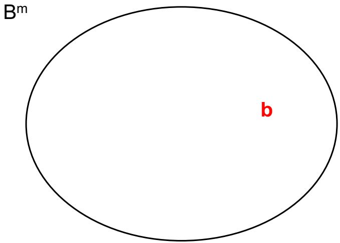
Defining $\text{TYPE}_R(\boldsymbol{x})$

For each quantifier rank r, there are finitely many, say n_r, types.
Each is described by one formula: $\varphi_{1,r}, \varphi_{2,r}, \ldots, \ldots, \varphi_{n_r,r} \in FO^k[r]$.
(Note: every $\varphi \in FO^k[r]$ is a union of types $\varphi = \bigvee_i \varphi_{i,r}$.)
Each $\varphi_{i,r}$ defines the equivalence class[10] $\{\boldsymbol{c} \in B^m \mid \boldsymbol{B} \vDash \varphi_{i,r}(\boldsymbol{c})\}$.

[10] Some equivalence classes are empty.

Defining $\text{TYPE}_R(\boldsymbol{x})$

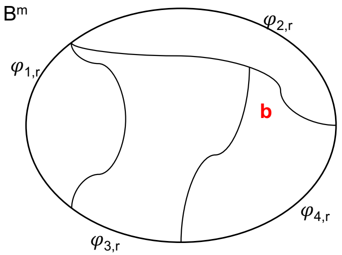
For each quantifier rank r, there are finitely many, say n_r, types.
Each is described by one formula: $\varphi_{1,r}, \varphi_{2,r}, \ldots, \ldots, \varphi_{n_r,r} \in FO^k[r]$.
(Note: every $\varphi \in FO^k[r]$ is a union of types $\varphi = \bigvee_i \varphi_{i,r}$.)
Each $\varphi_{i,r}$ defines the equivalence class[10] $\{\boldsymbol{c} \in B^m \mid \boldsymbol{B} \models \varphi_{i,r}(\boldsymbol{c})\}$.



[10] Some equivalence classes are empty.

Defining $\mathrm{TYPE}_R(\boldsymbol{x})$

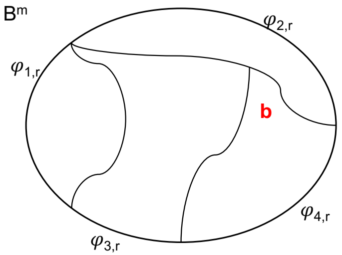
For each quantifier rank r, there are finitely many, say n_r, types.
Each is described by one formula: $\varphi_{1,r}, \varphi_{2,r}, \ldots, \ldots, \varphi_{n_r,r} \in FO^k[r]$.
(Note: every $\varphi \in FO^k[r]$ is a union of types $\varphi = \bigvee_i \varphi_{i,r}$.)
Each $\varphi_{i,r}$ defines the equivalence class[10] $\{\boldsymbol{c} \in B^m \mid \boldsymbol{B} \vDash \varphi_{i,r}(\boldsymbol{c})\}$.



[10] Some equivalence classes are empty.

Defining $\text{TYPE}_R(\boldsymbol{x})$

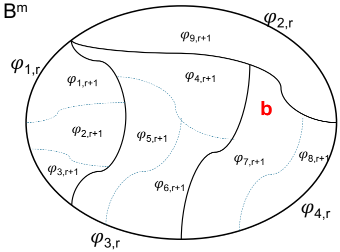
For each quantifier rank r, there are finitely many, say n_r, types.
Each is described by one formula: $\varphi_{1,r}, \varphi_{2,r}, \ldots, \ldots, \varphi_{n_r,r} \in FO^k[r]$.
(Note: every $\varphi \in FO^k[r]$ is a union of types $\varphi = \bigvee_i \varphi_{i,r}$.)
Each $\varphi_{i,r}$ defines the equivalence class[10] $\{\boldsymbol{c} \in B^m \mid \boldsymbol{B} \models \varphi_{i,r}(\boldsymbol{c})\}$.
The equivalence classes for $r+1$ are a refinement of those for r.



[10] Some equivalence classes are empty.

Defining $\text{TYPE}_R(\boldsymbol{x})$

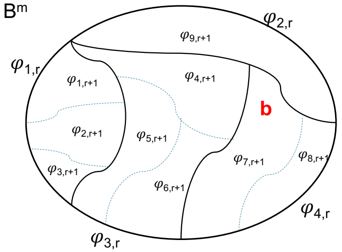
For each quantifier rank r, there are finitely many, say n_r, types.
Each is described by one formula: $\varphi_{1,r}, \varphi_{2,r}, \ldots, \ldots, \varphi_{n_r,r} \in FO^k[r]$.
(Note: every $\varphi \in FO^k[r]$ is a union of types $\varphi = \bigvee_i \varphi_{i,r}$.)
Each $\varphi_{i,r}$ defines the equivalence class[10] $\{\boldsymbol{c} \in B^m \mid \boldsymbol{B} \models \varphi_{i,r}(\boldsymbol{c})\}$.
The equivalence classes for $r+1$ are a refinement of those for r.



[10]Some equivalence classes are empty.

Defining $\text{TYPE}_R(\boldsymbol{x})$

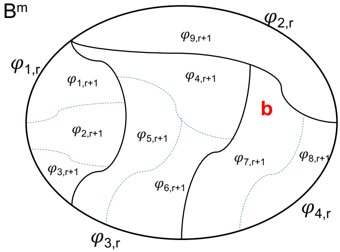
For each quantifier rank r, there are finitely many, say n_r, types.
Each is described by one formula: $\varphi_{1,r}, \varphi_{2,r}, \ldots, \ldots, \varphi_{n_r,r} \in FO^k[r]$.
(Note: every $\varphi \in FO^k[r]$ is a union of types $\varphi = \bigvee_i \varphi_{i,r}$.)
Each $\varphi_{i,r}$ defines the equivalence class[10] $\{\boldsymbol{c} \in B^m \mid \boldsymbol{B} \models \varphi_{i,r}(\boldsymbol{c})\}$.
The equivalence classes for $r+1$ are a refinement of those for r.
Since $\boldsymbol{B}$ is finite, the refinement stops at some R.



[10] Some equivalence classes are empty.

Defining $\text{TYPE}_R(\boldsymbol{x})$

For each quantifier rank r, there are finitely many, say n_r, types.
Each is described by one formula: $\varphi_{1,r}, \varphi_{2,r}, \ldots, \ldots, \varphi_{n_r,r} \in FO^k[r]$.
(Note: every $\varphi \in FO^k[r]$ is a union of types $\varphi = \bigvee_i \varphi_{i,r}$.)
Each $\varphi_{i,r}$ defines the equivalence class[10] $\{\boldsymbol{c} \in B^m \mid \boldsymbol{B} \vDash \varphi_{i,r}(\boldsymbol{c})\}$.
The equivalence classes for $r+1$ are a refinement of those for r.
Since $\boldsymbol{B}$ is finite, the refinement stops at some R.



Define: $\text{TYPE}_R(\boldsymbol{x}) \stackrel{\text{def}}{=} \varphi_{i,R}(\boldsymbol{x})$
where i = "the R-type of $\boldsymbol{b}$"

Note: all types reach a fixpoint at rank R, not just $\boldsymbol{b}$

[10] Some equivalence classes are empty.

Defining DONE_R

Every rank $r+1$ type refines some rank r type: $\forall j \exists i_j$,

$$\models \forall \boldsymbol{x}(\varphi_{j,r+1}(\boldsymbol{x}) \rightarrow \varphi_{i_j,r}(\boldsymbol{x}))$$

In $\boldsymbol{B}$, this becomes an equivalence at rank R:

$$\boldsymbol{B} \models \forall \boldsymbol{x}(\varphi_{j,R+1}(\boldsymbol{x}) \leftrightarrow \varphi_{i_j,R}(\boldsymbol{x}))$$

Define: $$\text{DONE}_R \stackrel{\text{def}}{=} \bigwedge_{j=1,n_{R+1}} \forall \boldsymbol{x}(\varphi_{j,R+1}(\boldsymbol{x}) \leftrightarrow \varphi_{i_j,R}(\boldsymbol{x})).$$

Assuming DONE_R, *every* rank $r > R$ is equivalent to some rank R:

Lemma

If $r > R$, then $\forall j \exists i_j$ s.t. $DONE_R \models \bigwedge_{j=1,n_r} \forall \boldsymbol{x}(\varphi_{j,r}(\boldsymbol{x}) \leftrightarrow \varphi_{i_j,R}(\boldsymbol{x}))$

proof in class (also on next slide)

Defining DONE_R

Every rank $r+1$ type refines some rank r type: $\forall j \exists i_j$,

$$\models \forall \boldsymbol{x}(\varphi_{j,r+1}(\boldsymbol{x}) \rightarrow \varphi_{i_j,r}(\boldsymbol{x}))$$

In $\boldsymbol{B}$, this becomes an equivalence at rank R:

$$\boldsymbol{B} \models \forall \boldsymbol{x}(\varphi_{j,R+1}(\boldsymbol{x}) \leftrightarrow \varphi_{i_j,R}(\boldsymbol{x}))$$

Define: $\text{DONE}_R \stackrel{\text{def}}{=} \bigwedge_{j=1,n_{R+1}} \forall \boldsymbol{x}(\varphi_{j,R+1}(\boldsymbol{x}) \leftrightarrow \varphi_{i_j,R}(\boldsymbol{x}))$.

Assuming DONE_R, *every* rank $r > R$ is equivalent to some rank R:

Lemma

If $r > R$, then $\forall j \exists i_j$ s.t. $DONE_R \models \bigwedge_{j=1,n_r} \forall \boldsymbol{x}(\varphi_{j,r}(\boldsymbol{x}) \leftrightarrow \varphi_{i_j,R}(\boldsymbol{x}))$

proof in class (also on next slide)

Defining DONE_R

Every rank $r+1$ type refines some rank r type: $\forall j \exists i_j$,

$$\models \forall \boldsymbol{x}(\varphi_{j,r+1}(\boldsymbol{x}) \rightarrow \varphi_{i_j,r}(\boldsymbol{x}))$$

In $\boldsymbol{B}$, this becomes an equivalence at rank R:

$$\boldsymbol{B} \models \forall \boldsymbol{x}(\varphi_{j,R+1}(\boldsymbol{x}) \leftrightarrow \varphi_{i_j,R}(\boldsymbol{x}))$$

Define: $$\text{DONE}_R \stackrel{\text{def}}{=} \bigwedge_{j=1,n_{R+1}} \forall \boldsymbol{x}(\varphi_{j,R+1}(\boldsymbol{x}) \leftrightarrow \varphi_{i_j,R}(\boldsymbol{x})) .$$

Assuming DONE_R, *every* rank $r > R$ is equivalent to some rank R:

Lemma

If $r > R$, then $\forall j \exists i_j$ s.t. $DONE_R \models \bigwedge_{j=1,n_r} \forall \boldsymbol{x}(\varphi_{j,r}(\boldsymbol{x}) \leftrightarrow \varphi_{i_j,R}(\boldsymbol{x}))$

proof in class (also on next slide)

Defining DONE_R

Every rank $r+1$ type refines some rank r type: $\forall j \exists i_j$,

$$\models \forall \boldsymbol{x}(\varphi_{j,r+1}(\boldsymbol{x}) \rightarrow \varphi_{i_j,r}(\boldsymbol{x}))$$

In $\boldsymbol{B}$, this becomes an equivalence at rank R:

$$\boldsymbol{B} \models \forall \boldsymbol{x}(\varphi_{j,R+1}(\boldsymbol{x}) \leftrightarrow \varphi_{i_j,R}(\boldsymbol{x}))$$

Define: $$\text{DONE}_R \stackrel{\text{def}}{=} \bigwedge_{j=1,n_{R+1}} \forall \boldsymbol{x}(\varphi_{j,R+1}(\boldsymbol{x}) \leftrightarrow \varphi_{i_j,R}(\boldsymbol{x})) \quad .$$

Assuming DONE_R, *every* rank $r > R$ is equivalent to some rank R:

Lemma

If $r > R$, then $\forall j \exists i_j$ s.t. $DONE_R \models \bigwedge_{j=1,n_r} \forall \boldsymbol{x}(\varphi_{j,r}(\boldsymbol{x}) \leftrightarrow \varphi_{i_j,R}(\boldsymbol{x}))$

proof in class (also on next slide)

Will show: every $R+2$ type is equivalent to some R type; induction follows.

$$\varphi_{j,R+2} \equiv \varphi_{j_0,R+1} \wedge \underbrace{F(\cdots \exists x_\ell \varphi_{j,R+1}, \cdots)}_{\substack{\text{Boolean combination } F \\ \text{of all } R+1 \text{ types } \varphi_{j,R+1} \\ \text{plus one extra } \exists x_\ell}}$$

DONE_R asserts that each $\varphi_{j,R+1}$ is equivalent to some $\varphi_{i_j,R}$:

$$\varphi_{j,R+2} \equiv \underbrace{\varphi_{j_0,R+1} \wedge F(\cdots \exists x_\ell \varphi_{i_j,R}, \cdots)}_{\text{quantifier rank } R+1}$$

$\varphi_{j,R+2} \equiv \varphi_{j_0,R+1}$ or $\varphi_{j,R+2} \equiv \boldsymbol{F}$ why?

Assuming DONE_R, we have $\varphi_{j_0,R+1} \equiv \varphi_{i_{j_0},R}$.

Proof of the Theorem

Theorem

For every FO^k type m-type τ, there exist a formula $\psi^\tau \in FO^k$ s.t., for any finite structure $\boldsymbol{A}$, $(\boldsymbol{A}, \boldsymbol{a}) \models \psi^\tau$ iff $tp^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \tau$.

Recall:

$$\tau = \text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b})$$
$$\psi^\tau(\boldsymbol{x}) = \text{TYPE}_R(\boldsymbol{x}) \wedge \text{DONE}_R$$

Assume $\text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \tau$; by construction $\psi^\tau \in \tau$, hence $(\boldsymbol{A}, \boldsymbol{a}) \models \psi^\tau$.
Assume $(\boldsymbol{A}, \boldsymbol{a}) \models \psi^\tau$. Let $\varphi \in \text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a})$ and $r = \max(qr(\varphi), R)$:

$\varphi(\boldsymbol{x}) = \bigvee_j \varphi_{j,r}(\boldsymbol{x})$ disjunction of some r-types

$\varphi(\boldsymbol{x}) = \bigvee_i \varphi_{i,R}(\boldsymbol{x})$ disjunction of some R-types (because $\boldsymbol{A} \models \text{DONE}_r$)

$\varphi(\boldsymbol{x}) \leftarrow \text{TYPE}_R(\boldsymbol{x})$ TYPE_R is an R-type

$\boldsymbol{B} \models \varphi(\boldsymbol{b})$ because the type of $(\boldsymbol{B}, \boldsymbol{b})$ is TYPE_R

Proof of the Theorem

Theorem

For every FO^k type m-type τ, there exist a formula $\psi^\tau \in FO^k$ s.t., for any finite structure $\boldsymbol{A}$, $(\boldsymbol{A}, \boldsymbol{a}) \models \psi^\tau$ iff $tp^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \tau$.

Recall:

$$\tau = \text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b})$$

$$\psi^\tau(\boldsymbol{x}) = \text{TYPE}_R(\boldsymbol{x}) \wedge \text{DONE}_R$$

Assume $\text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \tau$; by construction $\psi^\tau \in \tau$, hence $(\boldsymbol{A}, \boldsymbol{a}) \models \psi^\tau$.

Assume $(\boldsymbol{A}, \boldsymbol{a}) \models \psi^\tau$. Let $\varphi \in \text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a})$ and $r = \max(qr(\varphi), R)$:

$\varphi(\boldsymbol{x}) = \bigvee_j \varphi_{j,r}(\boldsymbol{x})$ disjunction of some r-types

$\varphi(\boldsymbol{x}) = \bigvee_i \varphi_{i,R}(\boldsymbol{x})$ disjunction of some R-types (because $\boldsymbol{A} \models \text{DONE}_r$)

$\varphi(\boldsymbol{x}) \leftarrow \text{TYPE}_R(\boldsymbol{x})$ TYPE_R is an R-type

$\boldsymbol{B} \models \varphi(\boldsymbol{b})$ because the type of $(\boldsymbol{B}, \boldsymbol{b})$ is TYPE_R

Proof of the Theorem

Theorem

For every FO^k type m-type τ, there exist a formula $\psi^\tau \in FO^k$ s.t., for any finite structure $\boldsymbol{A}$, $(\boldsymbol{A}, \boldsymbol{a}) \models \psi^\tau$ iff $tp^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \tau$.

Recall:

$$\tau = \mathrm{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b})$$
$$\psi^\tau(\boldsymbol{x}) = \mathrm{TYPE}_R(\boldsymbol{x}) \wedge \mathrm{DONE}_R$$

Assume $\mathrm{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \tau$; by construction $\psi^\tau \in \tau$, hence $(\boldsymbol{A}, \boldsymbol{a}) \models \psi^\tau$.
Assume $(\boldsymbol{A}, \boldsymbol{a}) \models \psi^\tau$. Let $\varphi \in \mathrm{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a})$ and $r = \max(qr(\varphi), R)$:

$\varphi(\boldsymbol{x}) = \bigvee_j \varphi_{j,r}(\boldsymbol{x})$ disjunction of some r-types

$\varphi(\boldsymbol{x}) = \bigvee_i \varphi_{i,R}(\boldsymbol{x})$ disjunction of some R-types (because $\boldsymbol{A} \models \mathrm{DONE}_r$)

$\varphi(\boldsymbol{x}) \leftarrow \mathrm{TYPE}_R(\boldsymbol{x})$ TYPE_R is an R-type

$\boldsymbol{B} \models \varphi(\boldsymbol{b})$ because the type of $(\boldsymbol{B}, \boldsymbol{b})$ is TYPE_R

Proof of the Theorem

Theorem

For every FO^k type m-type τ, there exist a formula $\psi^\tau \in FO^k$ s.t., for any finite structure $\boldsymbol{A}$, $(\boldsymbol{A}, \boldsymbol{a}) \models \psi^\tau$ iff $tp^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \tau$.

Recall:

$$\tau = \text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b})$$
$$\psi^\tau(\boldsymbol{x}) = \text{TYPE}_R(\boldsymbol{x}) \wedge \text{DONE}_R$$

Assume $\text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \tau$; by construction $\psi^\tau \in \tau$, hence $(\boldsymbol{A}, \boldsymbol{a}) \models \psi^\tau$.
Assume $(\boldsymbol{A}, \boldsymbol{a}) \models \psi^\tau$. Let $\varphi \in \text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a})$ and $r = \max(qr(\varphi), R)$:

$\varphi(\boldsymbol{x}) = \bigvee_j \varphi_{j,r}(\boldsymbol{x})$ disjunction of some r-types

$\varphi(\boldsymbol{x}) = \bigvee_i \varphi_{i,R}(\boldsymbol{x})$ disjunction of some R-types (because $\boldsymbol{A} \models \text{DONE}_r$)

$\varphi(\boldsymbol{x}) \leftarrow \text{TYPE}_R(\boldsymbol{x})$ TYPE_R is an R-type

$\boldsymbol{B} \models \varphi(\boldsymbol{b})$ because the type of $(\boldsymbol{B}, \boldsymbol{b})$ is TYPE_R

Proof of the Theorem

Theorem

For every FO^k type m-type τ, there exist a formula $\psi^\tau \in FO^k$ s.t., for any finite structure $\boldsymbol{A}$, $(\boldsymbol{A}, \boldsymbol{a}) \models \psi^\tau$ iff $tp^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \tau$.

Recall:

$$\tau = \text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b})$$
$$\psi^\tau(\boldsymbol{x}) = \text{TYPE}_R(\boldsymbol{x}) \wedge \text{DONE}_R$$

Assume $\text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \tau$; by construction $\psi^\tau \in \tau$, hence $(\boldsymbol{A}, \boldsymbol{a}) \models \psi^\tau$.
Assume $(\boldsymbol{A}, \boldsymbol{a}) \models \psi^\tau$. Let $\varphi \in \text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a})$ and $r = \max(qr(\varphi), R)$:

$\varphi(\boldsymbol{x}) = \bigvee_j \varphi_{j,r}(\boldsymbol{x})$ disjunction of some r-types

$\varphi(\boldsymbol{x}) = \bigvee_i \varphi_{i,R}(\boldsymbol{x})$ disjunction of some R-types (because $\boldsymbol{A} \models \text{DONE}_r$)

$\varphi(\boldsymbol{x}) \leftarrow \text{TYPE}_R(\boldsymbol{x})$ TYPE_R is an R-type

$\boldsymbol{B} \models \varphi(\boldsymbol{b})$ because the type of $(\boldsymbol{B}, \boldsymbol{b})$ is TYPE_R

Proof of the Theorem

Theorem

For every FO^k type m-type τ, there exist a formula $\psi^\tau \in FO^k$ s.t., for any finite structure $\boldsymbol{A}$, $(\boldsymbol{A}, \boldsymbol{a}) \vDash \psi^\tau$ iff $tp^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \tau$.

Recall:
$$\tau = \text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b})$$
$$\psi^\tau(\boldsymbol{x}) = \text{TYPE}_R(\boldsymbol{x}) \wedge \text{DONE}_R$$

Assume $\text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \tau$; by construction $\psi^\tau \in \tau$, hence $(\boldsymbol{A}, \boldsymbol{a}) \vDash \psi^\tau$.
Assume $(\boldsymbol{A}, \boldsymbol{a}) \vDash \psi^\tau$. Let $\varphi \in \text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a})$ and $r = \max(qr(\varphi), R)$:

$\varphi(\boldsymbol{x}) = \bigvee_j \varphi_{j,r}(\boldsymbol{x})$ disjunction of some r-types

$\varphi(\boldsymbol{x}) = \bigvee_i \varphi_{i,R}(\boldsymbol{x})$ disjunction of some R-types (because $\boldsymbol{A} \vDash \text{DONE}_r$)

$\varphi(\boldsymbol{x}) \leftarrow \text{TYPE}_R(\boldsymbol{x})$ TYPE_R is an R-type

$\boldsymbol{B} \vDash \varphi(\boldsymbol{b})$ because the type of $(\boldsymbol{B}, \boldsymbol{b})$ is TYPE_R

Proof of the Theorem

Theorem

For every FO^k type m-type τ, there exist a formula $\psi^\tau \in FO^k$ s.t., for any finite structure $\boldsymbol{A}$, $(\boldsymbol{A}, \boldsymbol{a}) \vDash \psi^\tau$ iff $tp^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \tau$.

Recall:
$$\tau = \text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b})$$
$$\psi^\tau(\boldsymbol{x}) = \text{TYPE}_R(\boldsymbol{x}) \wedge \text{DONE}_R$$

Assume $\text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \tau$; by construction $\psi^\tau \in \tau$, hence $(\boldsymbol{A}, \boldsymbol{a}) \vDash \psi^\tau$.
Assume $(\boldsymbol{A}, \boldsymbol{a}) \vDash \psi^\tau$. Let $\varphi \in \text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a})$ and $r = \max(qr(\varphi), R)$:

$\varphi(\boldsymbol{x}) = \bigvee_j \varphi_{j,r}(\boldsymbol{x})$ disjunction of some r-types

$\varphi(\boldsymbol{x}) = \bigvee_i \varphi_{i,R}(\boldsymbol{x})$ disjunction of some R-types (because $\boldsymbol{A} \vDash \text{DONE}_r$)

$\varphi(\boldsymbol{x}) \leftarrow \text{TYPE}_R(\boldsymbol{x})$ TYPE_R is an R-type

$\boldsymbol{B} \vDash \varphi(\boldsymbol{b})$ because the type of $(\boldsymbol{B}, \boldsymbol{b})$ is TYPE_R

Proof of the Theorem

Theorem

For every FO^k type m-type τ, there exist a formula $\psi^\tau \in FO^k$ s.t., for any finite structure $\boldsymbol{A}$, $(\boldsymbol{A}, \boldsymbol{a}) \vDash \psi^\tau$ iff $tp^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \tau$.

Recall:

$$\tau = \text{tp}^k_{FO}(\boldsymbol{B}, \boldsymbol{b})$$

$$\psi^\tau(\boldsymbol{x}) = \text{TYPE}_R(\boldsymbol{x}) \wedge \text{DONE}_R$$

Assume $\text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a}) = \tau$; by construction $\psi^\tau \in \tau$, hence $(\boldsymbol{A}, \boldsymbol{a}) \vDash \psi^\tau$.
Assume $(\boldsymbol{A}, \boldsymbol{a}) \vDash \psi^\tau$. Let $\varphi \in \text{tp}^k_{FO}(\boldsymbol{A}, \boldsymbol{a})$ and $r = \max(qr(\varphi), R)$:

$\varphi(\boldsymbol{x}) = \bigvee_j \varphi_{j,r}(\boldsymbol{x})$	disjunction of some r-types
$\varphi(\boldsymbol{x}) = \bigvee_i \varphi_{i,R}(\boldsymbol{x})$	disjunction of some R-types (because $\boldsymbol{A} \vDash \text{DONE}_r$)
$\varphi(\boldsymbol{x}) \leftarrow \text{TYPE}_R(\boldsymbol{x})$	TYPE_R is an R-type
$\boldsymbol{B} \vDash \varphi(\boldsymbol{b})$	because the type of $(\boldsymbol{B}, \boldsymbol{b})$ is TYPE_R

Recap

- Recap: a "type" τ is a maximally consistent set of formulas with m free variables, from some language (e.g. $FO[r]$ or FO^k or $FO^k[r]$).

- Equivalently, a "type" τ is the set of formulas that satisfy some $(\boldsymbol{A}, \boldsymbol{a})$ (where $|\boldsymbol{a}| = m$).

Discussion

Can we describe a type τ using a single formula?

- $FO[r]$ has finitely many formulas. Hence, a type is uniquely described by their conjunction, $\varphi_{r,m}$.
- FO^k has infinitely many formulas. The theorem says that, surprisingly(!), we can still describe it by a single formula ψ^τ, but only on finite structures.
- What is the quantifier rank of ψ^τ? Since τ is satisfied by some finite structure, its rank r is the smallest needed to express it in that structure.
- ψ^τ is $\varphi_{r,m}$ AND the assertion that this rank is sufficient.