Finite Model Theory
Lecture 12: Fixpoints and Infinitary Logic

Spring 2025

Announcements

- Homework 3 due on Friday.
- Today: finish games
- Next week: Weifeiler-Lehman, GNNs, C^k.

Review: Unfolding the Recursion in LFP

$$\boxed{S(x) = (\forall y(E(x,y) \to S(y)))}$$

The ICO: $F(S) = \{x \mid \forall y(E(x,y) \to S(y))\}$

Review: Unfolding the Recursion in LFP

$$\boxed{S(x) = (\forall y(E(x,y) \to S(y)))}$$

The ICO: $F(S) = \{x \mid \forall y(E(x,y) \to S(y))\}$

Elements of the Kleene sequence are computed by unfolded formulas:

Review: Unfolding the Recursion in LFP

$$\boxed{S(x) = (\forall y(E(x,y) \rightarrow S(y)))}$$

The ICO: $F(S) = \{x \mid \forall y(E(x,y) \rightarrow S(y))\}$

Elements of the Kleene sequence are computed by unfolded formulas:

$$\begin{array}{ll} S^0 = \emptyset & \varphi^0(x) = \mathtt{FALSE} \\ S^1 = F(S^0) & \varphi^1(x) = \forall y(E(x,y) \rightarrow \varphi^0(y)) \end{array}$$

Review: Unfolding the Recursion in LFP

$$\boxed{S(x) = (\forall y(E(x,y) \rightarrow S(y)))}$$

The ICO: $F(S) = \{x \mid \forall y(E(x,y) \rightarrow S(y))\}$

Elements of the Kleene sequence are computed by unfolded formulas:

$$\begin{array}{ll} S^0 = \emptyset & \varphi^0(x) = \texttt{FALSE} \\ S^1 = F(S^0) & \varphi^1(x) = \forall y(E(x,y) \rightarrow \varphi^0(y)) \\ S^2 = F(S^1) & \varphi^2(x) = \forall y(E(x,y) \rightarrow \varphi^1(y)) \end{array}$$

Review: Unfolding the Recursion in LFP

$$\boxed{S(x) = (\forall y(E(x,y) \to S(y)))}$$

The ICO: $F(S) = \{x \mid \forall y(E(x,y) \to S(y))\}$

Elements of the Kleene sequence are computed by unfolded formulas:

$$\begin{array}{ll}
S^0 = \varnothing & \varphi^0(x) = \mathtt{FALSE} \\
S^1 = F(S^0) & \varphi^1(x) = \forall y(E(x,y) \to \varphi^0(y)) \\
S^2 = F(S^1) & \varphi^2(x) = \forall y(E(x,y) \to \varphi^1(y)) \\
 & \dots \\
S^{n+1} = F(S^n) & \varphi^{n+1}(x) = \forall y(E(x,y) \to \varphi^n(y)) \\
 & \dots
\end{array}$$

Restricted Variables: FO^k

FO^k

Definition

FO^k is FO restricted to k variables $x_1, x_2, \ldots, x_k$.

Don't confuse with $\mathrm{FO}[k]$, which restricts the quantifier depth.

Examples

"Does the graph have a path of length 2?"

$$\exists x \exists y \exists z (E(x,y) \wedge E(y,z)) \in \mathrm{FO}^3$$

Examples

"Does the graph have a path of length 2?"

$$\exists x \exists y \exists z (E(x,y) \wedge E(y,z)) \in \mathsf{FO}^3$$

$$\exists x \exists y (E(x,y) \wedge \underbrace{\exists x E(y,x)}_{\text{reuse } x}) \in \mathsf{FO}^2$$

Examples

"Does the graph have a path of length 2?"

$$\exists x \exists y \exists z (E(x,y) \wedge E(y,z)) \in \mathrm{FO}^3$$

$$\exists x \exists y (E(x,y) \wedge \underbrace{\exists x E(y,x)}_{\text{reuse } x}) \in \mathrm{FO}^2$$

"Does the graph have a path of length 10?"

$$\exists x (\exists y E(x,y) \wedge \exists x (E(y,x) \wedge \underbrace{\exists y (E(x,y) \wedge \underbrace{\ldots \exists x E(y,x)}_{\text{reuse } x})}_{\text{reuse } y}))) \in \mathrm{FO}^2$$

(the outermost brace, labelled "reuse x", spans from $\exists x(E(y,x) \ldots$ to the end)

Unfolding in FO^k

Proposition

Let $\mathrm{lfp}_{S,\boldsymbol{x}}(\varphi)$ *be a least fixpoint predicate, assume* $\varphi \in FO^k$*.*
The unfolding φ^n *is in* FO^k*, for every* $n \geq 0$*.*

Unfolding in FO^k

Proposition

Let $\mathrm{lfp}_{S,\boldsymbol{x}}(\varphi)$ be a least fixpoint predicate, assume $\varphi \in FO^k$.
The unfolding φ^n is in FO^k, for every $n \geq 0$.

Proof by Example: $\boxed{S(x) = (\forall y(E(x,y) \to S(y)))}$

Unfolding in FO^k

Proposition

Let $\mathrm{lfp}_{S,\mathbf{x}}(\varphi)$ *be a least fixpoint predicate, assume* $\varphi \in FO^k$*.*
The unfolding φ^n *is in* FO^k*, for every* $n \geq 0$*.*

Proof by Example: $\boxed{S(x) = (\forall y(E(x,y) \to S(y)))}$

$$\varphi^1(x) = \forall y(E(x,y) \to \mathtt{FALSE})$$

Unfolding in FO^k

Proposition

Let $\mathrm{lfp}_{S,\boldsymbol{x}}(\varphi)$ *be a least fixpoint predicate, assume* $\varphi \in FO^k$.
The unfolding φ^n *is in* FO^k, *for every* $n \geq 0$.

Proof by Example: $\boxed{S(x) = (\forall y(E(x,y) \to S(y)))}$

$$
\begin{aligned}
\varphi^1(x) =& \forall y(E(x,y) \to \mathtt{FALSE}) \\
\varphi^2(x) =& \forall y(E(x,y) \to \varphi^1(y)) \\
=& \forall y(E(x,y) \to \exists x(x = y \wedge \varphi^1(x))) \\
=& \forall y(E(x,y) \to \exists x(x = y \wedge \forall y(E(x,y) \to \mathtt{FALSE})))
\end{aligned}
$$

Unfolding in FO^k

Proposition

Let $\mathrm{lfp}_{S,\boldsymbol{x}}(\varphi)$ be a least fixpoint predicate, assume $\varphi \in FO^k$.
The unfolding φ^n is in FO^k, for every $n \geq 0$.

Proof by Example: $\boxed{S(x) = (\forall y(E(x,y) \to S(y)))}$

$$
\begin{aligned}
\varphi^1(x) &= \forall y(E(x,y) \to \mathtt{FALSE}) \\
\varphi^2(x) &= \forall y(E(x,y) \to \varphi^1(y)) \\
&= \forall y(E(x,y) \to \exists x(x = y \wedge \varphi^1(x))) \\
&= \forall y(E(x,y) \to \exists x(x = y \wedge \forall y(E(x,y) \to \mathtt{FALSE}))) \\
\varphi^3(x) &= \forall y(E(x,y) \to \varphi^2(y)) \\
&= \forall y(E(x,y) \to \exists x(x = y \wedge \varphi^2(x))) \\
&= \ldots
\end{aligned}
$$

Discussion

- For good programming, we should avoid reusing variables.
- FO^k asks us the opposite: reuse variables when possible.
- Obvious fact: $\mathrm{FO} = \bigcup_{k \geq 0} \mathrm{FO}^k$.

Infinitary Logic and Pebble Games

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

- Let α, β be ordinals. The infinitary logic $L_{\alpha\beta}$ is:
  - Atoms: $x_i = x_j$, $R(\cdots)$
  - Formulas: $\bigvee_{i\in\alpha}\varphi_i$, $(\underbrace{\ldots\exists x_j\ldots}_{j\in\beta})\varphi$, and $\neg\varphi$.
  - No need for $\wedge, \forall$ because we use $\neg$.

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

- Let α, β be ordinals. The infinitary logic $L_{\alpha\beta}$ is:
  - Atoms: $x_i = x_j$, $R(\cdots)$
  - Formulas: $\bigvee_{i\in\alpha}\varphi_i$, $(\underbrace{\ldots\exists x_j\ldots}_{j\in\beta})\varphi$, and $\neg\varphi$.
  - No need for $\wedge, \forall$ because we use $\neg$.
- $L_{\omega\omega} = FO$; finite $\vee$, finite $\exists$.

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

- Let α, β be ordinals. The infinitary logic $L_{\alpha\beta}$ is:
  - Atoms: $x_i = x_j$, $R(\cdots)$
  - Formulas: $\bigvee_{i \in \alpha} \varphi_i$, $(\underbrace{\ldots \exists x_j \ldots}_{j \in \beta})\varphi$, and $\neg\varphi$.
  - No need for $\wedge, \forall$ because we use $\neg$.
- $L_{\omega\omega} = FO$; finite $\vee$, finite $\exists$.
- $L_{\infty\omega}$ = infinite disjunction (no bound!), finite quantifier sequence.

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

- Let α, β be ordinals. The infinitary logic $L_{\alpha\beta}$ is:
  - Atoms: $x_i = x_j$, $R(\cdots)$
  - Formulas: $\bigvee_{i\in\alpha} \varphi_i$, $(\underbrace{\ldots \exists x_j \ldots}_{j\in\beta})\varphi$, and $\neg\varphi$.
  - No need for $\wedge, \forall$ because we use $\neg$.
- $L_{\omega\omega} = FO$; finite $\vee$, finite $\exists$.
- $L_{\infty\omega}$ = infinite disjunction (no bound!), finite quantifier sequence.
- $L^k_{\infty\omega}$ = the restriction to k variables.

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

- Let α, β be ordinals. The infinitary logic $L_{\alpha\beta}$ is:
  - Atoms: $x_i = x_j$, $R(\cdots)$
  - Formulas: $\bigvee_{i\in\alpha} \varphi_i$, $(\underbrace{\ldots \exists x_j \ldots}_{j\in\beta})\varphi$, and $\neg\varphi$.
  - No need for $\wedge, \forall$ because we use $\neg$.
- $L_{\omega\omega} = FO$; finite $\vee$, finite $\exists$.
- $L_{\infty\omega}$ = infinite disjunction (no bound!), finite quantifier sequence.
- $L^{k}_{\infty\omega}$ = the restriction to k variables.
- $L^{\omega}_{\infty\omega} = \bigcup_{k\geq 0} L^{k}_{\infty\omega}$.

Notice: $L^{\omega}_{\infty\omega} \subsetneq L_{\infty\omega}$

Least Fixpoint Logic and $L^{\omega}_{\infty\omega}$

Theorem

Every query Φ in LFP can be expressed in $L^{\omega}_{\infty\omega}$

Least Fixpoint Logic and $L^{\omega}_{\infty\omega}$

Theorem

Every query Φ in LFP can be expressed in $L^{\omega}_{\infty\omega}$

Proof: More precisely, we will prove that, for every $\Phi \in LFP$, $\Phi \in L^{k}_{\infty\omega}$ where $k = |\texttt{Vars}(\Phi)|$.

Least Fixpoint Logic and $L^{\omega}_{\infty\omega}$

Theorem

Every query Φ in LFP can be expressed in $L^{\omega}_{\infty\omega}$

Proof: More precisely, we will prove that, for every $\Phi \in LFP$, $\Phi \in L^{k}_{\infty\omega}$ where $k = |\mathtt{Vars}(\Phi)|$.

Induction on the structure of Φ.

Least Fixpoint Logic and $L^{\omega}_{\infty\omega}$

Theorem

Every query Φ in LFP can be expressed in $L^{\omega}_{\infty\omega}$

Proof: More precisely, we will prove that, for every $\Phi \in LFP$, $\Phi \in L^{k}_{\infty\omega}$ where $k = |\mathtt{Vars}(\Phi)|$.

Induction on the structure of Φ.

If $\Phi = \Phi_1 \vee \Phi_2$, by induction $\Phi_1 \in L^{k}_{\infty\omega}$, $\Phi_2 \in L^{k}_{\infty\omega}$, thus $\Phi_1 \vee \Phi_2 \in L^{k}_{\infty\omega}$.

Least Fixpoint Logic and $L^{\omega}_{\infty\omega}$

Theorem

Every query Φ in LFP can be expressed in $L^{\omega}_{\infty\omega}$

Proof: More precisely, we will prove that, for every $\Phi \in LFP$, $\Phi \in L^{k}_{\infty\omega}$ where $k = |\mathtt{Vars}(\Phi)|$.

Induction on the structure of Φ.

If $\Phi = \Phi_1 \vee \Phi_2$, by induction $\Phi_1 \in L^{k}_{\infty\omega}$, $\Phi_2 \in L^{k}_{\infty\omega}$, thus $\Phi_1 \vee \Phi_2 \in L^{k}_{\infty\omega}$.

The only interesting case is a fixpoint expression: $\Phi = \mathrm{lfp}_{S,\boldsymbol{x}}(\varphi)$.

Least Fixpoint Logic and $L^{\omega}_{\infty\omega}$

Theorem

Every query Φ in LFP can be expressed in $L^{\omega}_{\infty\omega}$

Proof: More precisely, we will prove that, for every $\Phi \in LFP$, $\Phi \in L^{k}_{\infty\omega}$ where $k = |\mathtt{Vars}(\Phi)|$.

Induction on the structure of Φ.

If $\Phi = \Phi_1 \vee \Phi_2$, by induction $\Phi_1 \in L^{k}_{\infty\omega}$, $\Phi_2 \in L^{k}_{\infty\omega}$, thus $\Phi_1 \vee \Phi_2 \in L^{k}_{\infty\omega}$.

The only interesting case is a fixpoint expression: $\Phi = \mathrm{lfp}_{S,\boldsymbol{x}}(\varphi)$.

Then $\Phi = \varphi^0 \vee \varphi^1 \vee \cdots \vee \varphi^n \vee \cdots$; each $\varphi^n \in L^{k}_{\infty\omega}$, thus $\Phi_1 \vee \Phi_2 \in L^{k}_{\infty\omega}$.

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 hand.

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 hand.

Spoiler can do one of the following:

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

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 hand.

Spoiler can do one of the following:

- Place pebble i from his hand on $\boldsymbol{A}$ (or $\boldsymbol{B}$);
  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 $\boldsymbol{A}$).

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 hand.

Spoiler can do one of the following:

- Place pebble i from his hand on $\boldsymbol{A}$ (or $\boldsymbol{B}$);
  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.

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 $\varphi \in FO^k$.

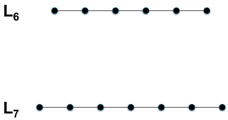
Theorem

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.

Examples

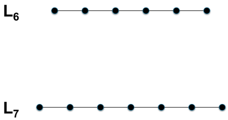
Total Order $L_n = ([n], <)$



Recall what we already know about L_n:

- We cannot express EVEN in FO
- We can express EVEN in Datalog

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

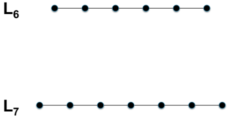


Recall what we already know about L_n:

- We cannot express EVEN in FO
- We can express EVEN in Datalog

Spoiler can win the 2-pebble game on L_m, L_n: $L_m \not\approx^2_{\infty\omega} L_n$. In class

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



Recall what we already know about L_n:

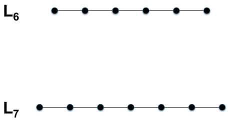
- We cannot express EVEN in FO
- We can express EVEN in Datalog

Spoiler can win the 2-pebble game on L_m, L_n: $L_m \not\approx^2_{\infty\omega} L_n$. In class

Fact

There exists $\varphi \in FO^2$ s.t. $L_7 \vDash \varphi$, $L_6 \vDash \neg\varphi$. Give example φ!

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



Recall what we already know about L_n:

- We cannot express EVEN in FO
- We can express EVEN in Datalog

Spoiler can win the 2-pebble game on L_m, L_n: $L_m \approx^2_{\infty\omega} L_n$. In class

Fact

There exists $\varphi \in FO^2$ s.t. $L_7 \vDash \varphi$, $L_6 \vDash \neg\varphi$. Give example φ!

Theorem

Over an ordered structure $(A, <, R_1^A, R_2^A, \ldots)$ stratified datalog = PTIME.

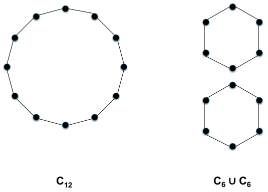
We are interested in what we can express without and order.

Connectivity

Duplicator wins the 2-pebble game on $\boldsymbol{A} = C_{2n}$ and $\boldsymbol{B} = (C_n \cup C_n)$.

Spoiler wins the 3-pebble game.

(In class)



Fact

There exists $\varphi \in FO^3$ s.t. $C_{2n} \models \varphi$, $(C_n \cup C_n) \models \neg\varphi$.

Describe a stratified datalog program that separates C_{2n} from $C_n \cup C_n$.

EVEN Number of Nodes

$\sigma = (E)$ vocabulary of graphs.

Two empty graphs: $G_m = ([m], \varnothing)$, $G_n = ([n], \varnothing)$

If $m, n \geq k$, then duplicator wins the k-pebble game on G_m, G_n.

EVEN Number of Nodes

$\sigma = (E)$ vocabulary of graphs.

Two empty graphs: $G_m = ([m], \emptyset)$, $G_n = ([n], \emptyset)$

If $m, n \geq k$, then duplicator wins the k-pebble game on G_m, G_n.

Fact

"G has an even number of nodes" is not expressible in $L^{\omega}_{\infty\omega}$.
Hence, it is not expressible in stratified datalog

EVEN Number of Edges

$\sigma = (E)$ vocabulary of graphs. Recall: K_n = complete graph.

If $m, n \geq k$, then duplicator wins the k-pebble game on K_m, K_n
Notice: K_m, K_n have m^2, n^2 edges resepctively.

EVEN Number of Edges

$\sigma = (E)$ vocabulary of graphs. Recall: K_n = complete graph.

If $m, n \geq k$, then duplicator wins the k-pebble game on K_m, K_n
Notice: K_m, K_n have m^2, n^2 edges resepctively.

Fact

"G has an even number of edges" is not expressible in $L^{\omega}_{\infty\omega}$.
Hence, it is not expressible in stratified datalog.

Hamiltonean Cycle

$\sigma = (E)$ vocabulary of graphs. Recall: $K_{m,n}$ = complete bipartite graph.

Fact

"G has a Hamiltonean Cycle" is not expressible in $L^{\omega}_{\infty\omega}$.
Hence, it is not expressible in stratified datalog.

Proof: If $m, n \geq k$, then duplicator wins the k-pebble game on $K_{m,n}, K_{n,n}$.

$K_{m,n}$ has a Hamiltonean cycle iff $m = n$.

Proof of the Main Theorem

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. $\boldsymbol{A} \equiv^k_{\mathrm{FO}} \boldsymbol{B}$ denotes: $\boldsymbol{A} \models \varphi$ iff $\boldsymbol{B} \models \varphi$, forall $\varphi \in FO^k$.

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. $\boldsymbol{A} \equiv^k_{\mathrm{FO}} \boldsymbol{B}$ 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.

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. $\boldsymbol{A} \equiv^k_{\infty\omega} \boldsymbol{B}$ implies $\boldsymbol{A} \approx^k_{\infty\omega} \boldsymbol{B}$, and
   $\boldsymbol{A} \equiv^k_{\text{FO}} \boldsymbol{B}$ implies $\boldsymbol{A} \approx^k_{\infty\omega} \boldsymbol{B}$ for finite structures.

The proof is almost identical to the EF-games

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

Induction on k; $k = 0$: same as for EF.

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

Induction on k; $k = 0$: same as for EF.

$k > 0$: We prove $\boldsymbol{A} \vDash \varphi$ implies $\boldsymbol{B} \vDash \varphi$, and vice versa, by induction on φ.

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

Induction on k; $k = 0$: same as for EF.

$k > 0$: We prove $\boldsymbol{A} \vDash \varphi$ implies $\boldsymbol{B} \vDash \varphi$, and vice versa, by induction on φ.

- If $\varphi = \bigvee_{i \in I} \psi_i$ and $\boldsymbol{A} \vDash \varphi$ then exists $i \in I$ s.t. $\boldsymbol{A} \vDash \psi_i$.
  By induction on φ, $\boldsymbol{B} \vDash \psi_i$, hence $\boldsymbol{B} \vDash \varphi$.
- If $\varphi = \neg\psi$, then, by induction $\boldsymbol{B} \vDash \psi$ implies $\boldsymbol{A} \vDash \psi$.
  Thus, $\boldsymbol{A} \vDash \varphi$ implies $\boldsymbol{B} \vDash \varphi$.
- $\varphi = \exists x \psi$. 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

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

Induction on k; $k = 0$: same as for EF.

$k > 0$: We prove $\boldsymbol{A} \models \varphi$ implies $\boldsymbol{B} \models \varphi$, and vice versa, by induction on φ.

- If $\varphi = \bigvee_{i \in I} \psi_i$ and $\boldsymbol{A} \models \varphi$ then exists $i \in I$ s.t. $\boldsymbol{A} \models \psi_i$.
  By induction on φ, $\boldsymbol{B} \models \psi_i$, hence $\boldsymbol{B} \models \varphi$.
- If $\varphi = \neg\psi$, then, by induction $\boldsymbol{B} \models \psi$ implies $\boldsymbol{A} \models \psi$.
  Thus, $\boldsymbol{A} \models \varphi$ implies $\boldsymbol{B} \models \varphi$.
- $\varphi = \exists x \psi$. 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 $(\boldsymbol{A}, c^A) \approx^{k-1}_{\infty\omega} (\boldsymbol{B}, c^B)$ (new constant c)

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

Induction on k; $k = 0$: same as for EF.

$k > 0$: We prove $\boldsymbol{A} \vDash \varphi$ implies $\boldsymbol{B} \vDash \varphi$, and vice versa, by induction on φ.

- If $\varphi = \bigvee_{i \in I} \psi_i$ and $\boldsymbol{A} \vDash \varphi$ then exists $i \in I$ s.t. $\boldsymbol{A} \vDash \psi_i$.
  By induction on φ, $\boldsymbol{B} \vDash \psi_i$, hence $\boldsymbol{B} \vDash \varphi$.
- If $\varphi = \neg\psi$, then, by induction $\boldsymbol{B} \vDash \psi$ implies $\boldsymbol{A} \vDash \psi$.
  Thus, $\boldsymbol{A} \vDash \varphi$ implies $\boldsymbol{B} \vDash \varphi$.
- $\varphi = \exists x \psi$. 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 $(\boldsymbol{A}, c^A) \approx^{k-1}_{\infty\omega} (\boldsymbol{B}, c^B)$ (new constant c)
  $(\boldsymbol{A}, c^A) \vDash \psi(c) (\in L^{k-1}_{\infty\omega})$ implies $(\boldsymbol{B}, c^B) \vDash \psi(c)$ by induction on k.

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

Induction on k; $k = 0$: same as for EF.

$k > 0$: We prove $\boldsymbol{A} \vDash \varphi$ implies $\boldsymbol{B} \vDash \varphi$, and vice versa, by induction on φ.

- If $\varphi = \bigvee_{i \in I} \psi_i$ and $\boldsymbol{A} \vDash \varphi$ then exists $i \in I$ s.t. $\boldsymbol{A} \vDash \psi_i$.
  By induction on φ, $\boldsymbol{B} \vDash \psi_i$, hence $\boldsymbol{B} \vDash \varphi$.

- If $\varphi = \neg\psi$, then, by induction $\boldsymbol{B} \vDash \psi$ implies $\boldsymbol{A} \vDash \psi$.
  Thus, $\boldsymbol{A} \vDash \varphi$ implies $\boldsymbol{B} \vDash \varphi$.

- $\varphi = \exists x \psi$. 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 $(\boldsymbol{A}, c^A) \approx^{k-1}_{\infty\omega} (\boldsymbol{B}, c^B)$ (new constant c)
  $(\boldsymbol{A}, c^A) \vDash \psi(c)(\in L^{k-1}_{\infty\omega})$ implies $(\boldsymbol{B}, c^B) \vDash \psi(c)$ by induction on k.
  Thus, $\boldsymbol{B} \vDash \psi(b)$ and $\boldsymbol{B} \vDash \exists x(\psi(x))$.

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

Induction on k; $k = 0$: same as for EF.

$k > 0$: We prove $\boldsymbol{A} \models \varphi$ implies $\boldsymbol{B} \models \varphi$, and vice versa, by induction on φ.

- If $\varphi = \bigvee_{i \in I} \psi_i$ and $\boldsymbol{A} \models \varphi$ then exists $i \in I$ s.t. $\boldsymbol{A} \models \psi_i$.
  By induction on φ, $\boldsymbol{B} \models \psi_i$, hence $\boldsymbol{B} \models \varphi$.
- If $\varphi = \neg\psi$, then, by induction $\boldsymbol{B} \models \psi$ implies $\boldsymbol{A} \models \psi$.
  Thus, $\boldsymbol{A} \models \varphi$ implies $\boldsymbol{B} \models \varphi$.
- $\varphi = \exists x \psi$. 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 $(\boldsymbol{A}, c^A) \approx^{k-1}_{\infty\omega} (\boldsymbol{B}, c^B)$ (new constant c)
  $(\boldsymbol{A}, c^A) \models \psi(c)(\in L^{k-1}_{\infty\omega})$ implies $(\boldsymbol{B}, c^B) \models \psi(c)$ by induction on k.
  Thus, $\boldsymbol{B} \models \psi(b)$ and $\boldsymbol{B} \models \exists x(\psi(x))$.

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

Obvious: if $\boldsymbol{A}, \boldsymbol{B}$ agree on all $L^k_{\infty\omega}$ sentences, then they also agree on all FO^k sentences, because $\mathrm{FO}^k \subseteq L^k_{\infty\omega}$.

FO^k-Types

Fix k and m.

Definition

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

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

FO^k-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:

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

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

- Both tp^k-types are complete; same as $\mathrm{FO}[k]$
- There are infinitely many tp^k-types of both kinds different from $\mathrm{FO}[k]$

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

Claim

If $\boldsymbol{A} \equiv^k_{\infty\omega} \boldsymbol{B}$, duplicator has a strategy s.t. the pebbles are always in the set $\mathcal{I} \stackrel{\text{def}}{=} \{(\boldsymbol{a}, \boldsymbol{b}) \mid |\boldsymbol{a}| = |\boldsymbol{b}| \leq k, \text{tp}^k_{\infty\omega}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}^k_{\infty\omega}(\boldsymbol{B}, \boldsymbol{b})\}$

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

Claim

If $\boldsymbol{A} \equiv^k_{\infty\omega} \boldsymbol{B}$, duplicator has a strategy s.t. the pebbles are always in the set $\mathcal{I} \stackrel{\text{def}}{=} \{(\boldsymbol{a}, \boldsymbol{b}) \mid |\boldsymbol{a}| = |\boldsymbol{b}| \leq k, \mathrm{tp}^k_{\infty\omega}(\boldsymbol{A}, \boldsymbol{a}) = \mathrm{tp}^k_{\infty\omega}(\boldsymbol{B}, \boldsymbol{b})\}$

Initially: $((), ()) \in \mathcal{I}$ because $\boldsymbol{A} \equiv^k_{\infty\omega} \boldsymbol{B}$.

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

Claim

If $\boldsymbol{A} \equiv^k_{\infty\omega} \boldsymbol{B}$, duplicator has a strategy s.t. the pebbles are always in the set $\mathcal{I} \stackrel{\text{def}}{=} \{(\boldsymbol{a}, \boldsymbol{b}) \mid |\boldsymbol{a}| = |\boldsymbol{b}| \leq k, \mathrm{tp}^k_{\infty\omega}(\boldsymbol{A}, \boldsymbol{a}) = \mathrm{tp}^k_{\infty\omega}(\boldsymbol{B}, \boldsymbol{b})\}$

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

Claim

If $\boldsymbol{A} \equiv^k_{\infty\omega} \boldsymbol{B}$, duplicator has a strategy s.t. the pebbles are always in the set $\mathcal{I} \stackrel{\text{def}}{=} \{(\boldsymbol{a}, \boldsymbol{b}) \mid |\boldsymbol{a}| = |\boldsymbol{b}| \le k, \text{tp}^k_{\infty\omega}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}^k_{\infty\omega}(\boldsymbol{B}, \boldsymbol{b})\}$

Duplicator removes pebble:
$\text{tp}^k_{\infty\omega}(\boldsymbol{A}, (a_1, \ldots, a_m)) = \text{tp}^k_{\infty\omega}(\boldsymbol{B}, (b_1, \ldots, b_m))$,
implies $\text{tp}^k_{\infty\omega}(\boldsymbol{A}, (a_1, \ldots, a_{m-1})) = \text{tp}^k_{\infty\omega}(\boldsymbol{B}, (b_1, \ldots, b_{m-1}))$

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

Claim

If $\boldsymbol{A} \equiv^k_{\infty\omega} \boldsymbol{B}$, duplicator has a strategy s.t. the pebbles are always in the set $\mathcal{I} \stackrel{\text{def}}{=} \{(\boldsymbol{a}, \boldsymbol{b}) \mid |\boldsymbol{a}| = |\boldsymbol{b}| \leq k, \mathrm{tp}^k_{\infty\omega}(\boldsymbol{A}, \boldsymbol{a}) = \mathrm{tp}^k_{\infty\omega}(\boldsymbol{B}, \boldsymbol{b})\}$

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

Claim

If $\boldsymbol{A} \equiv^k_{\infty\omega} \boldsymbol{B}$, duplicator has a strategy s.t. the pebbles are always in the set $\mathcal{I} \stackrel{\text{def}}{=} \{(\boldsymbol{a}, \boldsymbol{b}) \mid |\boldsymbol{a}| = |\boldsymbol{b}| \leq k, \text{tp}^k_{\infty\omega}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}^k_{\infty\omega}(\boldsymbol{B}, \boldsymbol{b})\}$

Duplicator places a pebble: Assume $|\boldsymbol{a}| = |\boldsymbol{b}| < k$
$\text{tp}^k_{\infty\omega}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}^k_{\infty\omega}(\boldsymbol{B}, \boldsymbol{b})$, and duplicator places on $a \in A$.

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

Claim

If $\boldsymbol{A} \equiv^k_{\infty\omega} \boldsymbol{B}$, duplicator has a strategy s.t. the pebbles are always in the set $\mathcal{I} \stackrel{\text{def}}{=} \{(\boldsymbol{a}, \boldsymbol{b}) \mid |\boldsymbol{a}| = |\boldsymbol{b}| \leq k, \text{tp}^k_{\infty\omega}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}^k_{\infty\omega}(\boldsymbol{B}, \boldsymbol{b})\}$

Duplicator places a pebble: Assume $|\boldsymbol{a}| = |\boldsymbol{b}| < k$
$\text{tp}^k_{\infty\omega}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}^k_{\infty\omega}(\boldsymbol{B}, \boldsymbol{b})$, and duplicator places on $a \in A$.
Claim: $\exists b \in B$ s.t. $\text{tp}^k_{\infty\omega}(\boldsymbol{A}, (\boldsymbol{a}, a)) = \text{tp}^k_{\infty\omega}(\boldsymbol{B}, (\boldsymbol{b}, b))$.

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

Claim

If $\boldsymbol{A} \equiv^k_{\infty\omega} \boldsymbol{B}$, duplicator has a strategy s.t. the pebbles are always in the set $\mathcal{I} \stackrel{\text{def}}{=} \{(\boldsymbol{a}, \boldsymbol{b}) \mid |\boldsymbol{a}| = |\boldsymbol{b}| \leq k, \mathrm{tp}^k_{\infty\omega}(\boldsymbol{A}, \boldsymbol{a}) = \mathrm{tp}^k_{\infty\omega}(\boldsymbol{B}, \boldsymbol{b})\}$

Duplicator places a pebble: Assume $|\boldsymbol{a}| = |\boldsymbol{b}| < k$
$\mathrm{tp}^k_{\infty\omega}(\boldsymbol{A}, \boldsymbol{a}) = \mathrm{tp}^k_{\infty\omega}(\boldsymbol{B}, \boldsymbol{b})$, and duplicator places on $a \in A$.
Claim: $\exists b \in B$ s.t. $\mathrm{tp}^k_{\infty\omega}(\boldsymbol{A}, (\boldsymbol{a}, a)) = \mathrm{tp}^k_{\infty\omega}(\boldsymbol{B}, (\boldsymbol{b}, b))$. Otherwise:

$$\forall b \in B, \exists \varphi_b(x_1, \ldots, x_{m-1}, y) \in L^k_{\infty\omega} \text{ s.t.} \quad \boldsymbol{A} \models \varphi_b(\boldsymbol{a}, a) \quad \boldsymbol{B} \not\models \varphi_b(\boldsymbol{b}, b)$$

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

Claim

If $\boldsymbol{A} \equiv^k_{\infty\omega} \boldsymbol{B}$, duplicator has a strategy s.t. the pebbles are always in the set $\mathcal{I} \stackrel{\text{def}}{=} \{(\boldsymbol{a}, \boldsymbol{b}) \mid |\boldsymbol{a}| = |\boldsymbol{b}| \le k, \mathrm{tp}^k_{\infty\omega}(\boldsymbol{A}, \boldsymbol{a}) = \mathrm{tp}^k_{\infty\omega}(\boldsymbol{B}, \boldsymbol{b})\}$

Duplicator places a pebble: Assume $|\boldsymbol{a}| = |\boldsymbol{b}| < k$
$\mathrm{tp}^k_{\infty\omega}(\boldsymbol{A}, \boldsymbol{a}) = \mathrm{tp}^k_{\infty\omega}(\boldsymbol{B}, \boldsymbol{b})$, and duplicator places on $a \in A$.
Claim: $\exists b \in B$ s.t. $\mathrm{tp}^k_{\infty\omega}(\boldsymbol{A}, (\boldsymbol{a}, a)) = \mathrm{tp}^k_{\infty\omega}(\boldsymbol{B}, (\boldsymbol{b}, b))$. Otherwise:

$$\forall b \in B, \exists \varphi_b(x_1, \ldots, x_{m-1}, y) \in L^k_{\infty\omega} \text{ s.t.} \quad \boldsymbol{A} \models \varphi_b(\boldsymbol{a}, a) \quad \boldsymbol{B} \not\models \varphi_b(\boldsymbol{b}, b)$$

$$\psi \stackrel{\text{def}}{=} \exists y \bigwedge_{b' \in B} \varphi_{b'}(x_1, \ldots, x_{m-1}, y) \text{ then} \quad \boldsymbol{A} \models \psi(\boldsymbol{a}) \quad \boldsymbol{B} \not\models \psi(\boldsymbol{b})$$

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

Claim

If $\boldsymbol{A} \equiv^k_{\infty\omega} \boldsymbol{B}$, duplicator has a strategy s.t. the pebbles are always in the set $\mathcal{I} \stackrel{\text{def}}{=} \{(\boldsymbol{a}, \boldsymbol{b}) \mid |\boldsymbol{a}| = |\boldsymbol{b}| \leq k, \text{tp}^k_{\infty\omega}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}^k_{\infty\omega}(\boldsymbol{B}, \boldsymbol{b})\}$

Duplicator places a pebble: Assume $|\boldsymbol{a}| = |\boldsymbol{b}| < k$
$\text{tp}^k_{\infty\omega}(\boldsymbol{A}, \boldsymbol{a}) = \text{tp}^k_{\infty\omega}(\boldsymbol{B}, \boldsymbol{b})$, and duplicator places on $a \in A$.
Claim: $\exists b \in B$ s.t. $\text{tp}^k_{\infty\omega}(\boldsymbol{A}, (\boldsymbol{a}, a)) = \text{tp}^k_{\infty\omega}(\boldsymbol{B}, (\boldsymbol{b}, b))$. Otherwise:

$$\begin{array}{lll} \forall b \in B, \exists \varphi_b(x_1, \ldots, x_{m-1}, y) \in L^k_{\infty\omega} \text{ s.t.} & \boldsymbol{A} \models \varphi_b(\boldsymbol{a}, a) & \boldsymbol{B} \not\models \varphi_b(\boldsymbol{b}, b) \\ \psi \stackrel{\text{def}}{=} \exists y \bigwedge_{b' \in B} \varphi_{b'}(x_1, \ldots, x_{m-1}, y) \text{ then} & \boldsymbol{A} \models \psi(\boldsymbol{a}) & \boldsymbol{B} \not\models \psi(\boldsymbol{b}) \end{array}$$

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

$\boldsymbol{A} \equiv^k_{\mathrm{FO}} \boldsymbol{B}$ and $\boldsymbol{A}, \boldsymbol{B}$ Finite Implies $\boldsymbol{A} \approx^k_{\infty\omega} \boldsymbol{B}$

What changes:

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

The formula $\psi \stackrel{\text{def}}{=} \bigwedge_{b' \in B} \varphi_{b'}(x_1, \ldots, x_{m-1}, y)$ is in FO^k,
because every $\varphi_{b'}$ is in FO^k and the conjunction is finite, since B is finite.

Discussion

- If two finite structures can be distinguished by $L^k_{\infty\omega}$, then they can already be distinguished by FO^k.
- Duplicator's winning positions in the pebble game are captured by FO^k-types; these can be shown to be the same as $L^k_{\infty\omega}$ types.
- Each FO$[k]$ type contains only finitely many formulas: hence their conjunction is a formula that fully characterizes the type.
- Every FO^k, or $L^k_{\infty\omega}$ type contains infinitely many formulas. Still, it can be shown that each type is fully characterized by one formula in FO^k.

Stratified Datalog and LFP

Review: The Win-Move Game

$G = (V, E)$ no more restriction to two children

Player I and Player II take turns, moving a pebble. Player who cannot move (is on a leaf) loses.

Winning positions for Player I:

$$\boxed{P1(x) = \exists y(E(x,y) \wedge \forall z(E(y,z) \to P1(z)))}$$

Review: The Win-Move Game

$G = (V, E)$ no more restriction to two children

Player I and Player II take turns, moving a pebble. Player who cannot move (is on a leaf) loses.

Winning positions for Player I:

$$\boxed{P1(x) = \exists y(E(x,y) \wedge \forall z(E(y,z) \to P1(z)))}$$

Theorem (Kolaitis)

The win-move game is not expressible in stratified datalog.

Recall that this is expressible in datalog when the graph has degree 2 (or any bounded degree).

Identifying the Right Logic and Right Game

Each LFP formula is expressible in $L^k_{\infty\omega}$ for some k.
k-Pebble games capture the expressive power of $L^k_{\infty\omega}$.

Need to find a restricted logic, and a restricted game, to capture stratified datalog programs.

Answer: observe that a stratified datalog program uses a fixed number of negation alternations.
In that case, the duplicator of the k-pebble game can only switch a fixed number of times between structures $\boldsymbol{A}$ and $\boldsymbol{B}$.

Negation Depth

Define $L^{k,\ell}_{\infty\omega}$ to negation depth $\leq \ell$:

- Atoms: $x_i = x_j$, $R(\cdots)$ are in $L^{k,0}_{\infty\omega}$, when the free variables are in $\{x_1, \ldots, x_k\}$.
- If $\varphi_m \in L^{k,\ell}_{\infty\omega}$ for all i, then $\bigvee_m \varphi_m \in L^{k,\ell}_{\infty\omega}$
- If $\varphi \in L^{k,\ell}_{\infty\omega}$ then $\exists x_i \varphi \in L^{k,\ell}_{\infty\omega}$.
- $\neg\varphi$ is in $L^{k,\ell}_{\alpha\beta}$, when φ is in $L^{k,\ell-1}_{\alpha\beta}$.

Notice: $\bigcup_{\ell \geq 0} L^{k,\ell}_{\infty\omega} \subsetneq L^{k}_{\infty\omega}$, but $\bigcup_{k\geq 0} \mathrm{FO}^{k,\ell} = \Sigma_\ell$

Stratified Datalog

Fact

If P is a stratified datalog program, then there exists a formula in $L^{k,\ell}_{\infty\omega}$ that is equivalent to P, for some k, ℓ that depend on P.

Stratified Datalog

Fact

If P is a stratified datalog program, then there exists a formula in $L^{k,\ell}_{\infty\omega}$ that is equivalent to P, for some k, ℓ that depend on P.

Proof sketch $k \stackrel{\text{def}}{=}$ the number of variables in P, $\ell \stackrel{\text{def}}{=}$ number of strata

First stratum: $\varphi_1^0 \vee \varphi_1^1 \vee \varphi_1^2 \vee \cdots \in L^{k,1}_{\infty\omega}$ EDBs might be negated.

Stratified Datalog

Fact

If P is a stratified datalog program, then there exists a formula in $L_{\infty\omega}^{k,\ell}$ that is equivalent to P, for some k,ℓ that depend on P.

Proof sketch $k \stackrel{\text{def}}{=}$ the number of variables in P, $\ell \stackrel{\text{def}}{=}$ number of strata

First stratum: $\varphi_1^0 \vee \varphi_1^1 \vee \varphi_1^2 \vee \cdots \in L_{\infty\omega}^{k,1}$ EDBs might be negated.

Second stratum: $\varphi_2^0 \vee \varphi_2^1 \vee \varphi_2^2 \vee \cdots \in L_{\infty\omega}^{k,2}$

Stratified Datalog

Fact

If P is a stratified datalog program, then there exists a formula in $L^{k,\ell}_{\infty\omega}$ that is equivalent to P, for some k, ℓ that depend on P.

Proof sketch $k \stackrel{\text{def}}{=}$ the number of variables in P, $\ell \stackrel{\text{def}}{=}$ number of strata

First stratum: $\varphi_1^0 \vee \varphi_1^1 \vee \varphi_1^2 \vee \cdots \in L^{k,1}_{\infty\omega}$ EDBs might be negated.

Second stratum: $\varphi_2^0 \vee \varphi_2^1 \vee \varphi_2^2 \vee \cdots \in L^{k,2}_{\infty\omega}$

Third stratum: $\ldots \in L^{k,3}_{\infty\omega}$, etc.

Variation on the Pebble Game

The k, ℓ-pebble game: spoiler can switch between $\boldsymbol{A}, \boldsymbol{B}$ at most ℓ times.

Variation on the Pebble Game

The k, ℓ-pebble game: spoiler can switch between $\boldsymbol{A}, \boldsymbol{B}$ at most ℓ times.

$\boldsymbol{A} \approx^{k,\ell}_{\infty\omega} \boldsymbol{B}$ denotes: duplicator wins.

$\boldsymbol{A} \equiv^{k,\ell}_{\infty\omega} \boldsymbol{B}$ denotes: $\boldsymbol{A} \vDash \varphi$ iff $\boldsymbol{B} \vDash \varphi$ for all $\varphi \in L^{k,\ell}_{\infty\omega}$.

Variation on the Pebble Game

The k, ℓ-pebble game: spoiler can switch between $\boldsymbol{A}, \boldsymbol{B}$ at most ℓ times.

$\boldsymbol{A} \approx_{\infty\omega}^{k,\ell} \boldsymbol{B}$ denotes: duplicator wins.

$\boldsymbol{A} \equiv_{\infty\omega}^{k,\ell} \boldsymbol{B}$ denotes: $\boldsymbol{A} \vDash \varphi$ iff $\boldsymbol{B} \vDash \varphi$ for all $\varphi \in L_{\infty\omega}^{k,\ell}$.

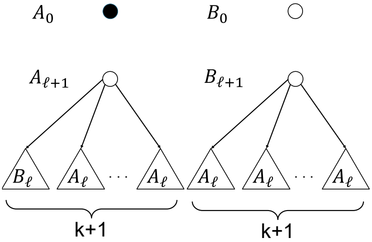
Theorem

$\boldsymbol{A} \approx_{\infty\omega}^{k,\ell} \boldsymbol{B}$ *iff* $\boldsymbol{A} \equiv_{\infty\omega}^{k,\ell} \boldsymbol{B}$

Proof (sketch) For the direction $\Rightarrow$ we only needed to switch $\boldsymbol{A}$ and $\boldsymbol{B}$ once for every negation.

The WinMove Game on Two Graphs

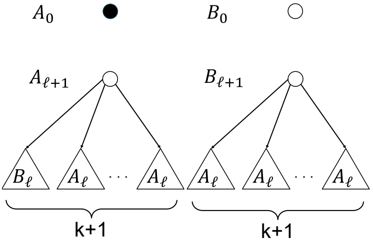
Define the graphs $\boldsymbol{A}_\ell, \boldsymbol{B}_\ell$, $\ell \geq 0$. Black node wins.



The WinMove Game on Two Graphs

Define the graphs $\boldsymbol{A}_\ell, \boldsymbol{B}_\ell$, $\ell \geq 0$. Black node wins.

Player I wins on root(A_ℓ)
Player II wins on root(B_ℓ)

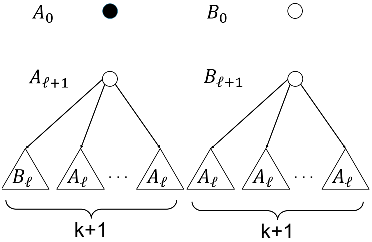


The WinMove Game on Two Graphs

Define the graphs $\boldsymbol{A}_\ell, \boldsymbol{B}_\ell$, $\ell \geq 0$. Black node wins.

Player I wins on root(A_ℓ) Player II wins on root(B_ℓ)

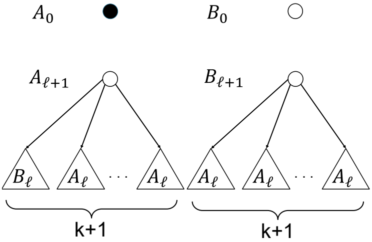
Claim: If spoiler starts on $\boldsymbol{A}_\ell$, then duplicator wins the $k, \ell - 1$ game. If spoiler starts on $\boldsymbol{B}_\ell$, then duplicator wins the k, ℓ game.



The WinMove Game on Two Graphs

Define the graphs $\boldsymbol{A}_\ell, \boldsymbol{B}_\ell$, $\ell \geq 0$. Black node wins.

Player I wins on root(A_ℓ) Player II wins on root(B_ℓ)



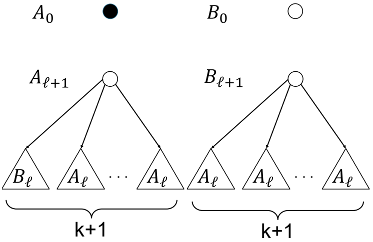
Claim: If spoiler starts on $\boldsymbol{A}_\ell$, then duplicator wins the $k, \ell - 1$ game.
If spoiler starts on $\boldsymbol{B}_\ell$, then duplicator wins the k, ℓ game.

Proof Assume it holds for ℓ, we prove for $\ell + 1$.

The WinMove Game on Two Graphs

Define the graphs $\boldsymbol{A}_\ell, \boldsymbol{B}_\ell$, $\ell \geq 0$. Black node wins.

Player I wins on root(A_ℓ)
Player II wins on root(B_ℓ)



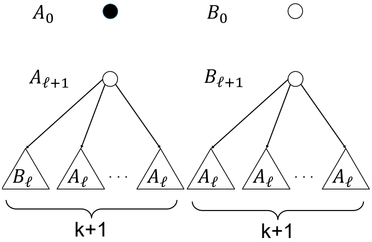
Claim: If spoiler starts on $\boldsymbol{A}_\ell$, then duplicator wins the $k, \ell - 1$ game.
If spoiler starts on $\boldsymbol{B}_\ell$, then duplicator wins the k, ℓ game.

Proof Assume it holds for ℓ, we prove for $\ell + 1$.
If spoiler starts on $\boldsymbol{A}_{\ell+1}$, pebbles in $\boldsymbol{A}_\ell$ are wasted. Only pebbles in $\boldsymbol{B}_\ell$ matter, and here duplicator uses k, ℓ-strategy.

The WinMove Game on Two Graphs

Define the graphs $\boldsymbol{A}_\ell, \boldsymbol{B}_\ell$, $\ell \geq 0$. Black node wins.

Player I wins on root(A_ℓ)
Player II wins on root(B_ℓ)



Claim: If spoiler starts on $\boldsymbol{A}_\ell$, then duplicator wins the $k, \ell - 1$ game.
If spoiler starts on $\boldsymbol{B}_\ell$, then duplicator wins the k, ℓ game.

Proof Assume it holds for ℓ, we prove for $\ell + 1$.
If spoiler starts on $\boldsymbol{A}_{\ell+1}$, pebbles in $\boldsymbol{A}_\ell$ are wasted. Only pebbles in $\boldsymbol{B}_\ell$ matter, and here duplicator uses k, ℓ-strategy.
If spoiler starts on $\boldsymbol{B}_{\ell+1}$, all pebbles are wasted, until he switches to $\boldsymbol{A}_{\ell+1}$.

Discussion

- Games: a powerful and flexible concept that allows us to reason about what is and what is not expressible.
  - EF games for FO;
  - EMSO games to prove s, t-reachability not in MSO;
  - Pebble games to prove EVEN, Hamiltonean path, etc not expressible in datalog
  - Variation: restrict the duplicator's number of alternations to prove limitations of stratified datalog
- Next week: will discuss the Weisfeiler-Lehman refinement procedure, its application to GNNs, and its connection to logics with counting