Finite Model Theory
Lecture 8: Ehrenfeucht-Fraisse Games

Spring 2025

Announcements

- Homework 2 due on Friday.
- HW3 will be about EF games: will release it soon.
- Today: we finish EF games.
- Next week: Second order logic, strings, trees, regular languages.

Review: Expressibility of FO

Goal: describe what properties P we can express in FO.

Each finite structure $\boldsymbol{A}$ is uniquely identified by a sentence φ^A.

- We cannot find $\boldsymbol{A}, \boldsymbol{B}$ such that $P(\boldsymbol{A}) = 1$, $P(\boldsymbol{B}) = 0$ and $\boldsymbol{A}, \boldsymbol{B}$ satisfy exactly the same sentences.

Workaround: restrict the quantifier depth k, FO[k]:

- For each k find $\boldsymbol{A}_k, \boldsymbol{B}_k$ such that $P(\boldsymbol{A}_k) = 1$, $P(\boldsymbol{B}_k) = 0$ and $\boldsymbol{A}_k, \boldsymbol{B}_k$ satisfy exactly the same sentences in FO[k].

Lots of notations!

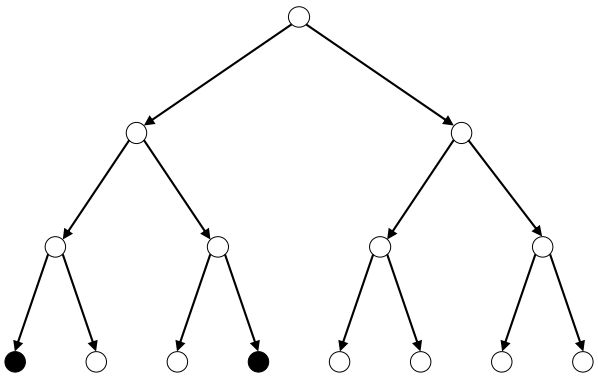
- $\boldsymbol{A} \simeq \boldsymbol{B}$: isomorphic
- $\boldsymbol{A} \equiv \boldsymbol{B}$: elementary equivalent, i.e. satisfy the same FO sentences
  when $\boldsymbol{A}, \boldsymbol{B}$ are finite, then this implies $\boldsymbol{A} \simeq \boldsymbol{B}$.
- $\boldsymbol{A} \equiv_k \boldsymbol{B}$: equivalent for all sentences in FO$[k]$.
- $\boldsymbol{A} \sim_k \boldsymbol{B}$: duplicator wins the EF game with k pebbles

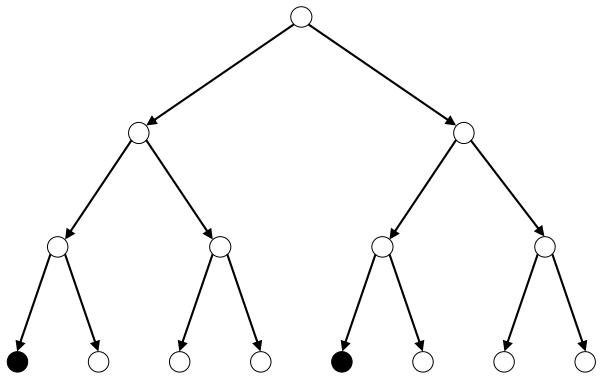
Ehrenfeucht-Fraisse theorem: $\boxed{\boldsymbol{A} \equiv_k \boldsymbol{B}}$ iff $\boxed{\boldsymbol{A} \sim_k \boldsymbol{B}}$.

Review: Ehrenfeucht-Fraisse Games

$\sigma = (E, B)$ $E(x, y)$ =edge, $B(x)$ = "black"

Prove that the Spoiler has a winning strategy with $k = 3$ pebbles.

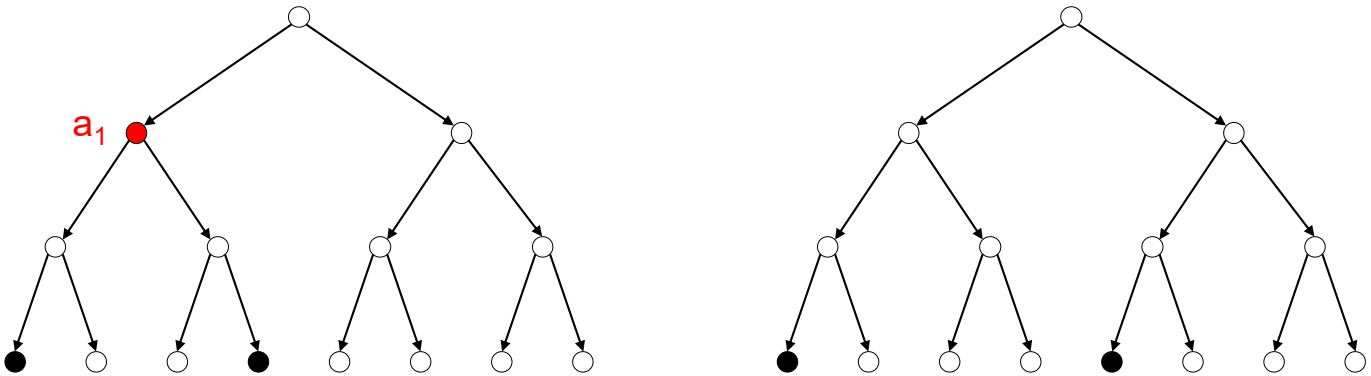




Review: Ehrenfeucht-Fraisse Games

$\sigma = (E, B)$ $E(x, y)$ =edge, $B(x)$ = "black"

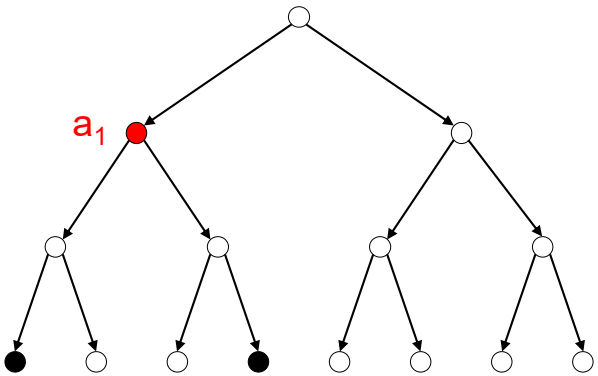
Prove that the Spoiler has a winning strategy with $k = 3$ pebbles.

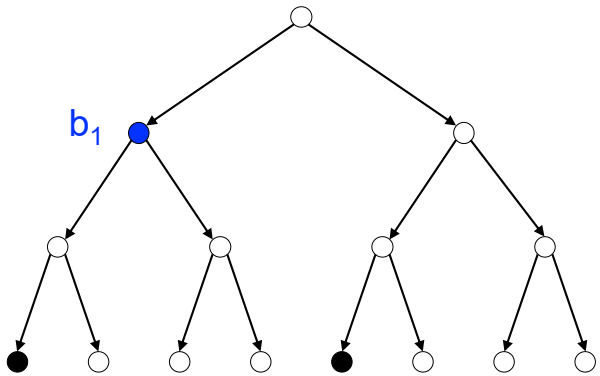


Review: Ehrenfeucht-Fraisse Games

$\sigma = (E, B)$ $E(x, y)$ =edge, $B(x)$ = "black"

Prove that the Spoiler has a winning strategy with $k = 3$ pebbles.

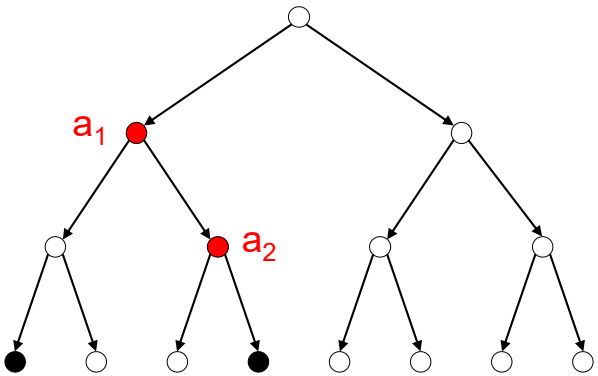


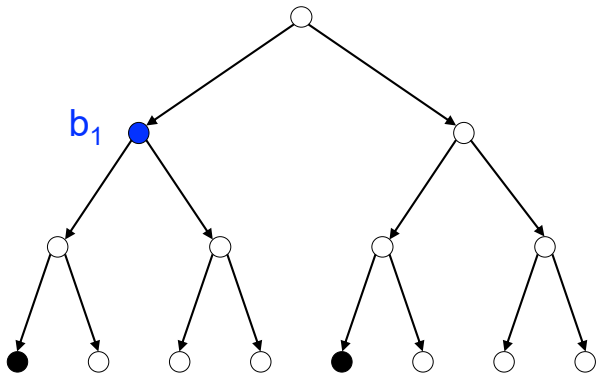


Review: Ehrenfeucht-Fraisse Games

$\sigma = (E, B)$ $E(x, y)$ =edge, $B(x)$ = "black"

Prove that the Spoiler has a winning strategy with $k = 3$ pebbles.

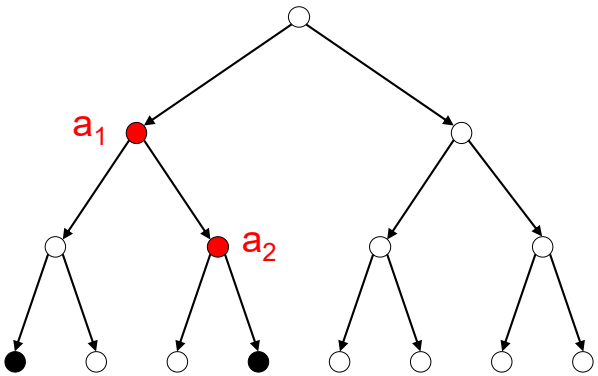


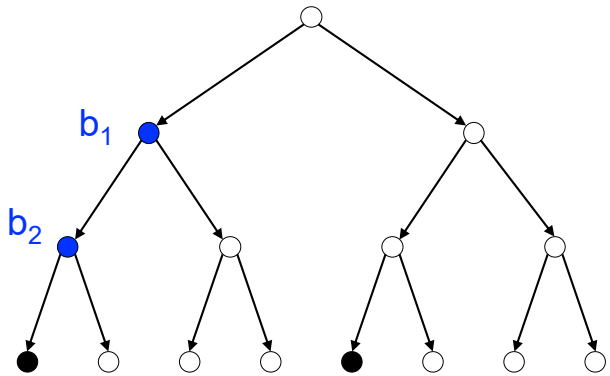


Review: Ehrenfeucht-Fraisse Games

$\sigma = (E, B)$ $E(x, y)$ =edge, $B(x)$ = "black"

Prove that the Spoiler has a winning strategy with $k = 3$ pebbles.



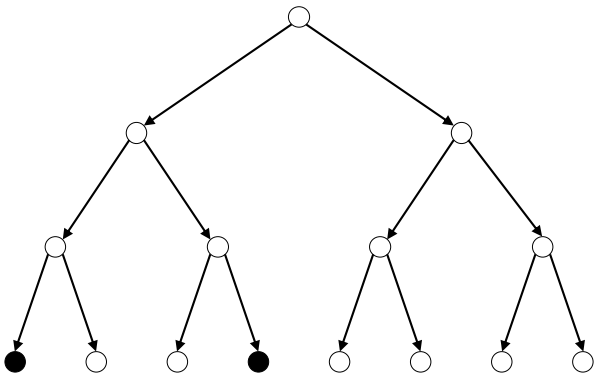


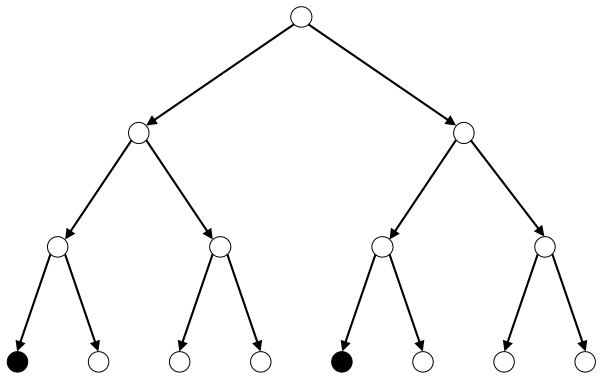
Whatever spoiler does at round 3, duplicator wins.
Let's try a better strategy for spoiler

Review: Ehrenfeucht-Fraisse Games

$\sigma = (E, B)$ $E(x, y)$ =edge, $B(x)$ = "black"

Prove that the Spoiler has a winning strategy with $k = 3$ pebbles.

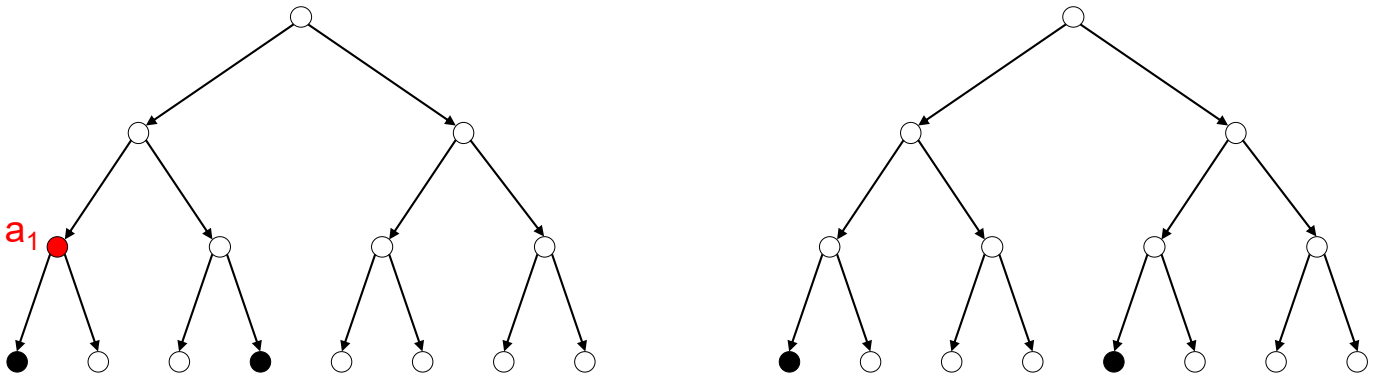




Review: Ehrenfeucht-Fraisse Games

$\sigma = (E, B)$ $E(x, y)$ =edge, $B(x)$ = "black"

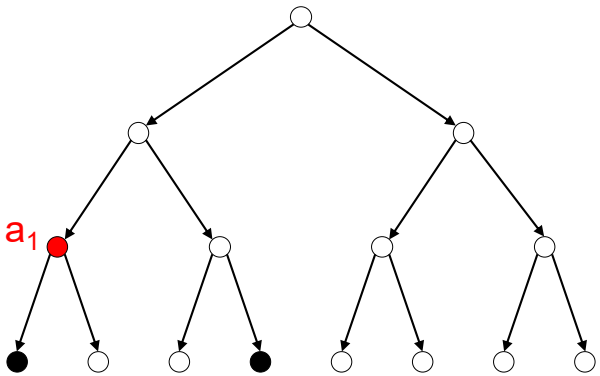
Prove that the Spoiler has a winning strategy with $k = 3$ pebbles.

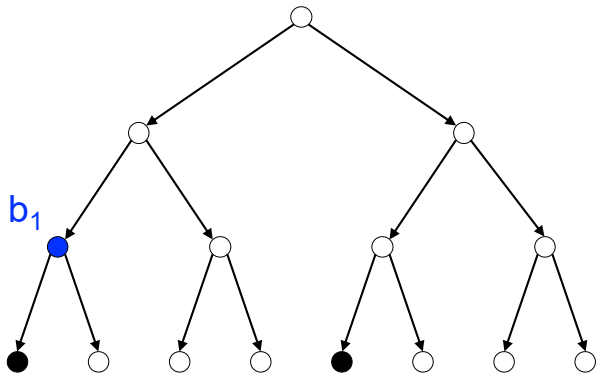


Review: Ehrenfeucht-Fraisse Games

$\sigma = (E, B)$ $E(x, y)$ =edge, $B(x)$ = "black"

Prove that the Spoiler has a winning strategy with $k = 3$ pebbles.

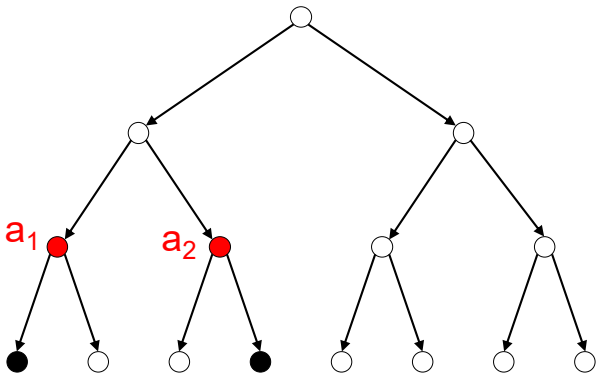


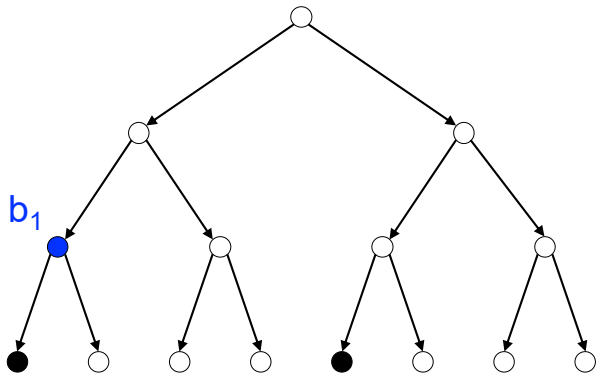


Review: Ehrenfeucht-Fraisse Games

$\sigma = (E, B)$ $E(x, y)$ =edge, $B(x)$ = "black"

Prove that the Spoiler has a winning strategy with $k = 3$ pebbles.

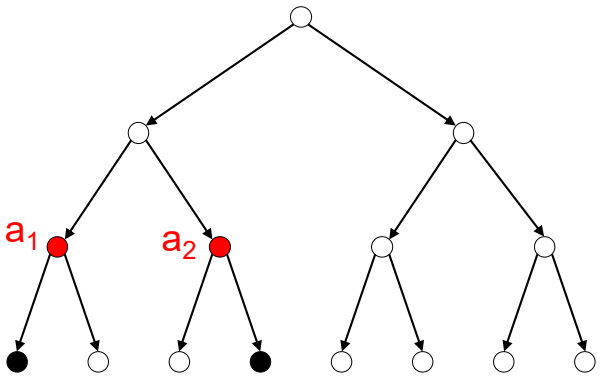


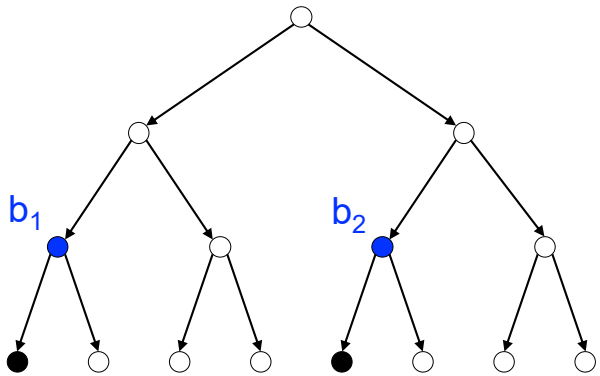


Review: Ehrenfeucht-Fraisse Games

$\sigma = (E, B)$ $E(x, y)$ =edge, $B(x)$ = "black"

Prove that the Spoiler has a winning strategy with $k = 3$ pebbles.

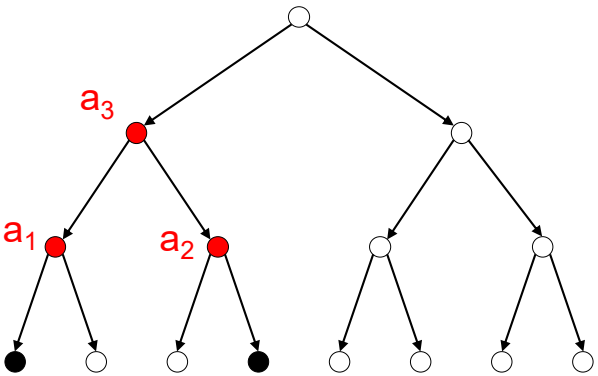


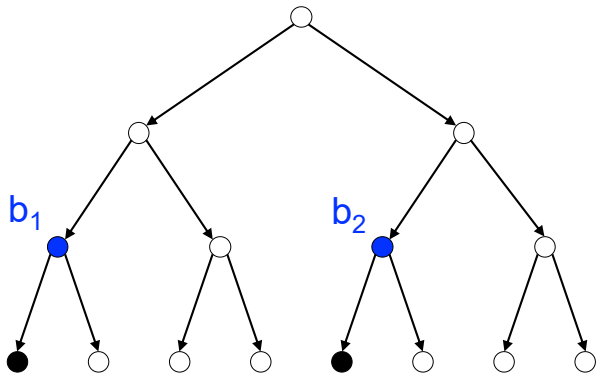


Review: Ehrenfeucht-Fraisse Games

$\sigma = (E, B)$ $E(x, y)$ =edge, $B(x)$ = "black"

Prove that the Spoiler has a winning strategy with $k = 3$ pebbles.

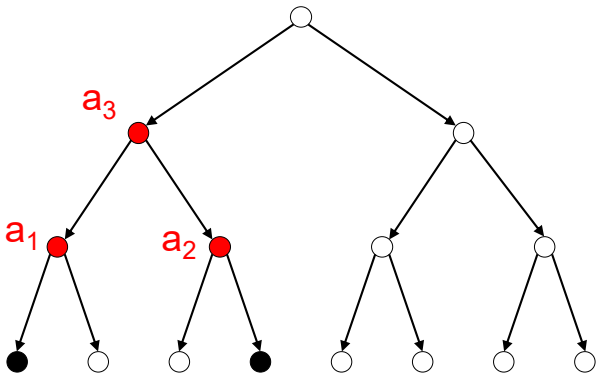


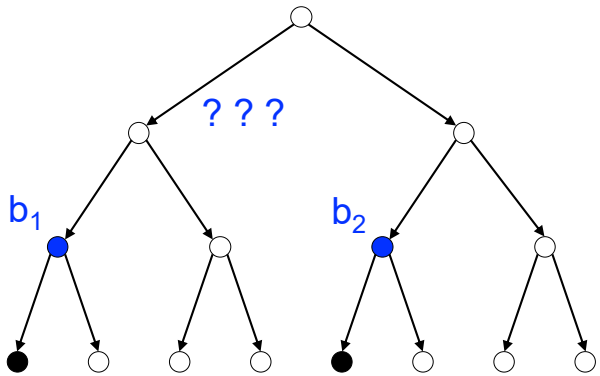


Review: Ehrenfeucht-Fraisse Games

$\sigma = (E, B)$ $E(x, y)$ =edge, $B(x)$ = "black"

Prove that the Spoiler has a winning strategy with $k = 3$ pebbles.

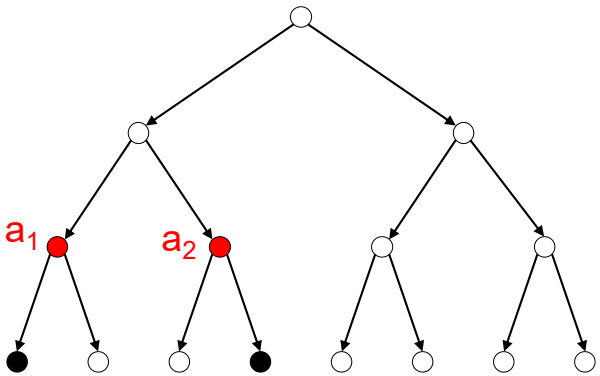


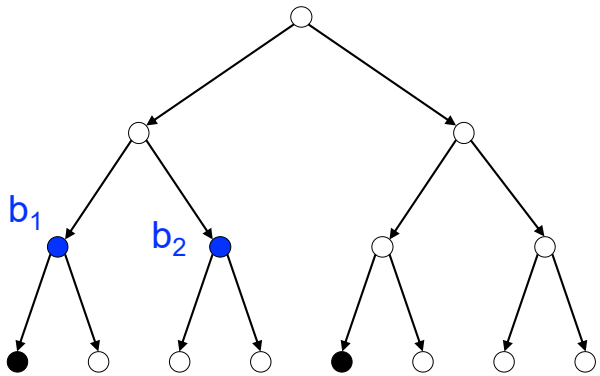


Review: Ehrenfeucht-Fraisse Games

$\sigma = (E, B)$ $E(x, y)$ =edge, $B(x)$ = "black"

Prove that the Spoiler has a winning strategy with $k = 3$ pebbles.

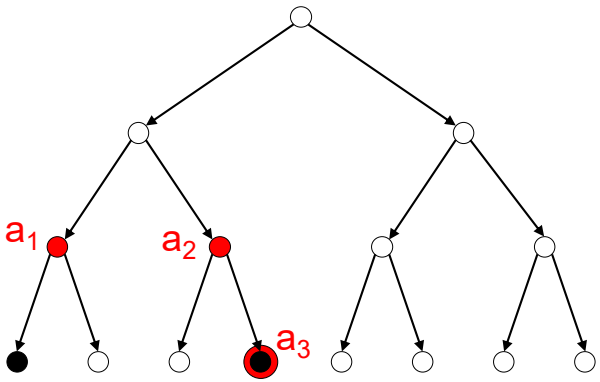


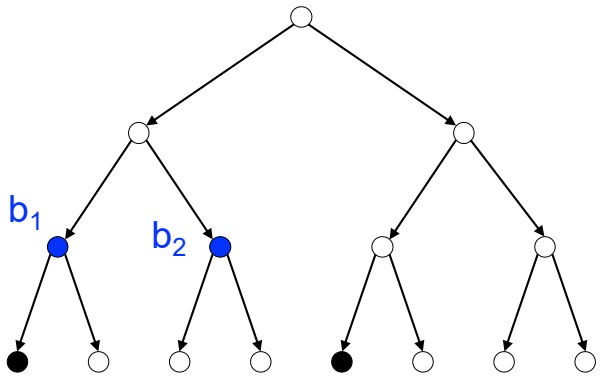


Review: Ehrenfeucht-Fraisse Games

$\sigma = (E, B)$ $E(x, y)$ =edge, $B(x)$ = "black"

Prove that the Spoiler has a winning strategy with $k = 3$ pebbles.

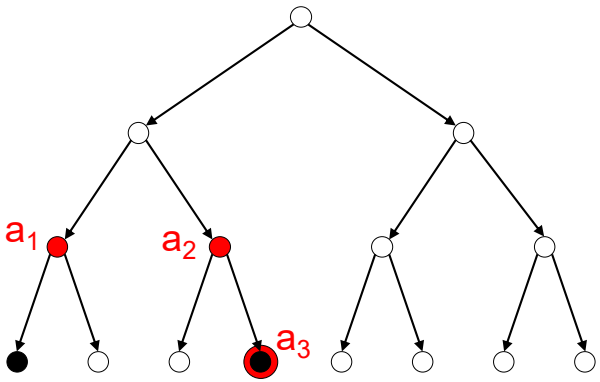


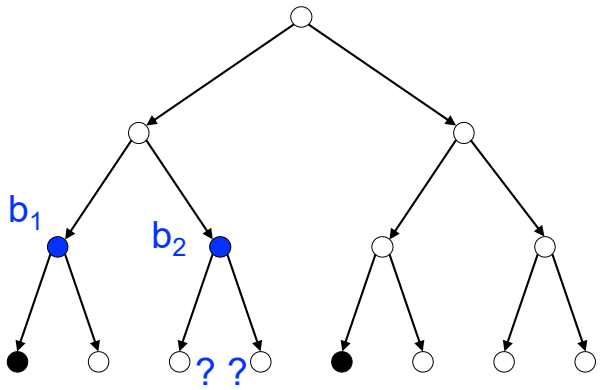


Review: Ehrenfeucht-Fraisse Games

$\sigma = (E, B)$ $E(x, y)$ =edge, $B(x)$ = "black"

Prove that the Spoiler has a winning strategy with $k = 3$ pebbles.

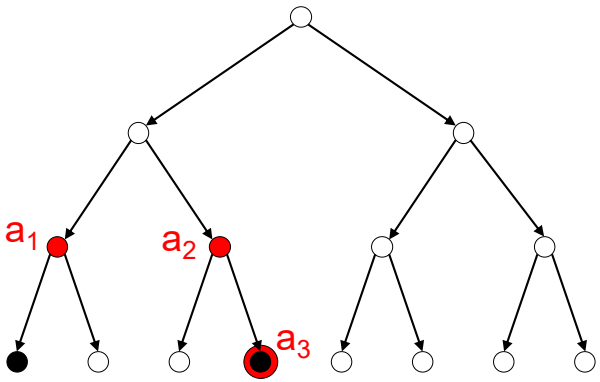


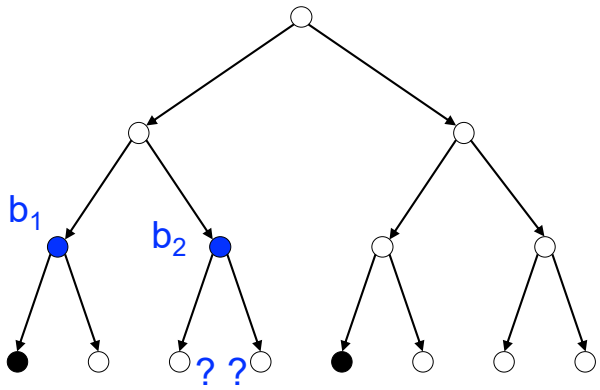


Review: Ehrenfeucht-Fraisse Games

$\sigma = (E, B)$ $E(x, y)$ =edge, $B(x)$ = "black"

Prove that the Spoiler has a winning strategy with $k = 3$ pebbles.





At home: find $\varphi \in \mathrm{FO}[3]$ that separates the two trees!

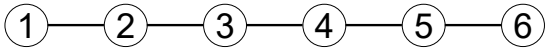
Discussion

- Number of rounds k correspond to quantifier depth FO[k]
- When duplicator plays on $\boldsymbol{A}$ it corresponds to $\exists$
  When duplicator plays on $\boldsymbol{B}$ it corresponds to $\forall$.
- We will re-examine this shortly, but first let's see two applications.

EVEN and CONNECTIVITY

Review: Linear Order

$\boldsymbol{L}_n = ([n], <)$



Theorem

If $m, n \geq 2^k - 1$, then $\boldsymbol{L}_m \sim_k \boldsymbol{L}_n$. *Otherwise $\boldsymbol{L}_m \sim_k \boldsymbol{L}_n$ iff $m = n$.*

Corollary

EVEN is not expressible over linear orders.

Proof Suppose φ is s.t. $\boldsymbol{L}_n \models \varphi$ iff n is even.
Choose $m \stackrel{\text{def}}{=} 2^k - 1$ and $n \stackrel{\text{def}}{=} 2^k$, where $k = qr(\varphi)$.
Then $\boldsymbol{L}_n \models \varphi$, hence $\boldsymbol{L}_m \models \varphi$, contradiction.

Application: CONNECTIVITY is not expressible in FO

CONNECTIVITY: given a graph $\boldsymbol{G} = (V, E)$, check if it is connected.

Corollary

CONNECTIVITY is not expressible in FO

Application: CONNECTIVITY is not expressible in FO

CONNECTIVITY: given a graph $\boldsymbol{G} = (V, E)$, check if it is connected.

Corollary

CONNECTIVITY is not expressible in FO

Proof Reduction from EVEN. Assume ψ expresses CONNECTIVITY.

Application: CONNECTIVITY is not expressible in FO

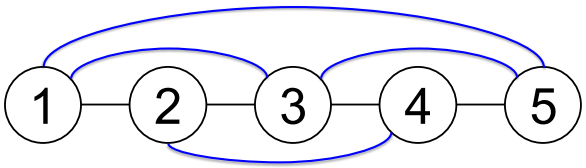
CONNECTIVITY: given a graph $\boldsymbol{G} = (V, E)$, check if it is connected.

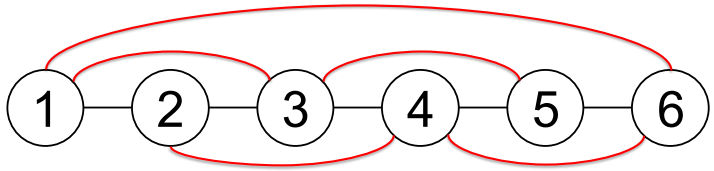
Corollary

CONNECTIVITY is not expressible in FO

Proof Reduction from EVEN. Assume ψ expresses CONNECTIVITY.

Given $\boldsymbol{L_n}$, construct $\boldsymbol{G}$ with edges $(i, i+2)$ and $(1, n)$





Application: CONNECTIVITY is not expressible in FO

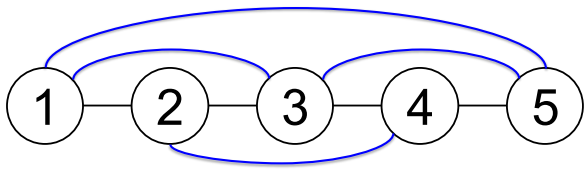
CONNECTIVITY: given a graph $\boldsymbol{G} = (V, E)$, check if it is connected.

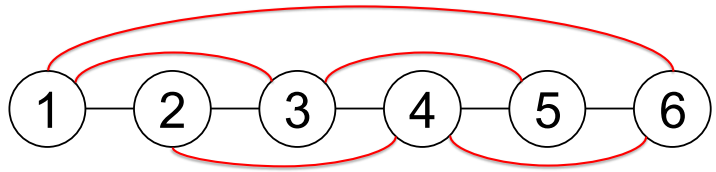
Corollary

CONNECTIVITY is not expressible in FO

Proof Reduction from EVEN. Assume ψ expresses CONNECTIVITY.

Given $\boldsymbol{L_n}$, construct $\boldsymbol{G}$ with edges $(i, i+2)$ and $(1, n)$





Then n is even iff $\boldsymbol{G}$ is connected.

Application: CONNECTIVITY is not expressible in FO

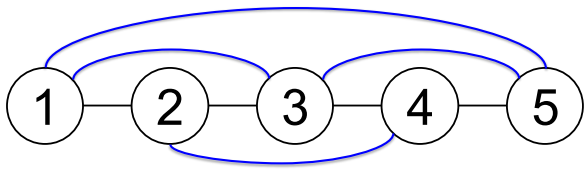
CONNECTIVITY: given a graph $\boldsymbol{G} = (V, E)$, check if it is connected.

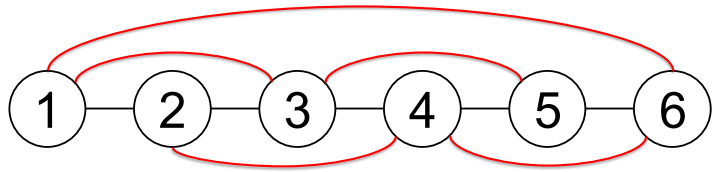
Corollary

CONNECTIVITY is not expressible in FO

Proof Reduction from EVEN. Assume ψ expresses CONNECTIVITY.

Given $\boldsymbol{L_n}$, construct $\boldsymbol{G}$ with edges $(i, i+2)$ and $(1, n)$





$$E(x,y) \stackrel{\text{def}}{=} (\exists z(\texttt{succ}(x,z) \wedge \texttt{succ}(z,y))) \vee (\texttt{isMin}(x) \wedge \texttt{isMax}(y))$$

Then n is even iff $\boldsymbol{G}$ is connected.

Discussion

EF games are a powerful tool for proving inexpressibility. Examples:

- EVEN, CONNECTIVITY.
- Check if a graph is a tree.
- Check if a graph is planar.

Deep Dive into FO[k]

The Constants

Vocabulary $\sigma = (R_1, \ldots, R_n, c_1, \ldots, c_m)$.

EF games add the constants $(a_1, \ldots, a_k)$ and $(b_1, \ldots, b_k)$; unavoidable.

What is FO[0]?

The Constants

Vocabulary $\sigma = (R_1, \ldots, R_n, c_1, \ldots, c_m)$.

EF games add the constants $(a_1, \ldots, a_k)$ and $(b_1, \ldots, b_k)$; unavoidable.

What is FO[0]?

- Sentences with constants only, e.g. $R_1(c_1, c_2) \wedge \neg R_2(c_2, c_2)$.

The Constants

Vocabulary $\sigma = (R_1, \ldots, R_n, c_1, \ldots, c_m)$.

EF games add the constants $(a_1, \ldots, a_k)$ and $(b_1, \ldots, b_k)$; unavoidable.

What is FO[0]?

- Sentences with constants only, e.g. $R_1(c_1, c_2) \wedge \neg R_2(c_2, c_2)$.
- If σ has no constants, FO[0] is empty, and $\boldsymbol{A} \equiv_0 \boldsymbol{B}$ for any $\boldsymbol{A}, \boldsymbol{B}$.

The Constants

Vocabulary $\sigma = (R_1, \ldots, R_n, c_1, \ldots, c_m)$.

EF games add the constants $(a_1, \ldots, a_k)$ and $(b_1, \ldots, b_k)$; unavoidable.

What is FO[0]?

- Sentences with constants only, e.g. $R_1(c_1, c_2) \wedge \neg R_2(c_2, c_2)$.
- If σ has no constants, FO[0] is empty, and $\boldsymbol{A} \equiv_0 \boldsymbol{B}$ for any $\boldsymbol{A}, \boldsymbol{B}$.

Can the duplicator lose in 0-rounds?

The Constants

Vocabulary $\sigma = (R_1, \ldots, R_n, c_1, \ldots, c_m)$.

EF games add the constants $(a_1, \ldots, a_k)$ and $(b_1, \ldots, b_k)$; unavoidable.

What is FO[0]?

- Sentences with constants only, e.g. $R_1(c_1, c_2) \wedge \neg R_2(c_2, c_2)$.
- If σ has no constants, FO[0] is empty, and $\boldsymbol{A} \equiv_0 \boldsymbol{B}$ for any $\boldsymbol{A}, \boldsymbol{B}$.

Can the duplicator lose in 0-rounds?

- YES: e.g. when $\boldsymbol{A} \models R_1(c_1, c_2)$ and $\boldsymbol{B} \models \neg R_1(c_1, c_2)$.

The Constants

Vocabulary $\sigma = (R_1, \ldots, R_n, c_1, \ldots, c_m)$.

EF games add the constants $(a_1, \ldots, a_k)$ and $(b_1, \ldots, b_k)$; unavoidable.

What is FO[0]?

- Sentences with constants only, e.g. $R_1(c_1, c_2) \wedge \neg R_2(c_2, c_2)$.
- If σ has no constants, FO[0] is empty, and $\boldsymbol{A} \equiv_0 \boldsymbol{B}$ for any $\boldsymbol{A}, \boldsymbol{B}$.

Can the duplicator lose in 0-rounds?

- YES: e.g. when $\boldsymbol{A} \models R_1(c_1, c_2)$ and $\boldsymbol{B} \models \neg R_1(c_1, c_2)$.
- If σ has no constants, duplicator always wins in 0 rounds.

Counting the Number of Sentences in FO[k]

Fix n Boolean variables: $X_1, X_2, \ldots, X_n$.

How many Boolean functions exists? E.g. $X_1 \wedge X_2$; or $X_1 \vee \neg X_3 \wedge X_4$, ...

Counting the Number of Sentences in FO[k]

Fix n Boolean variables: $X_1, X_2, \ldots, X_n$.

How many Boolean functions exists? E.g. $X_1 \wedge X_2$; or $X_1 \vee \neg X_3 \wedge X_4$, ...

Answer: 2^{2^n}

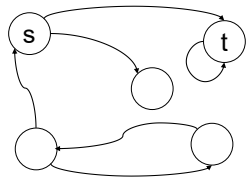
Counting the Number of Sentences in FO[k]

Fix n Boolean variables: $X_1, X_2, \ldots, X_n$.

How many Boolean functions exists? E.g. $X_1 \wedge X_2$; or $X_1 \vee \neg X_3 \wedge X_4$, ...

Answer: 2^{2^n}

Graphs with source and target: $\sigma = (E, s, t)$.



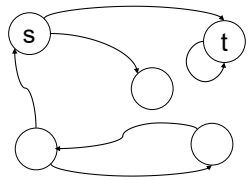
Counting the Number of Sentences in FO[k]

Fix n Boolean variables: $X_1, X_2, \ldots, X_n$.

How many Boolean functions exists? E.g. $X_1 \wedge X_2$; or $X_1 \vee \neg X_3 \wedge X_4$, ...

Answer: 2^{2^n}

Graphs with source and target: $\sigma = (E, s, t)$.



FO[0] atoms without variables: $E(s,s), E(s,t), E(t,s), E(t,t)$. 2^2

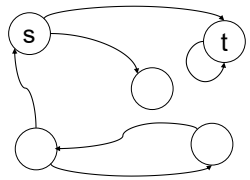
Counting the Number of Sentences in FO[k]

Fix n Boolean variables: $X_1, X_2, \ldots, X_n$.

How many Boolean functions exists? E.g. $X_1 \wedge X_2$; or $X_1 \vee \neg X_3 \wedge X_4$, ...

Answer: 2^{2^n}

Graphs with source and target: $\sigma = (E, s, t)$.



FO[0] atoms without variables: $E(s,s), E(s,t), E(t,s), E(t,t)$. 2^2

FO[0] sentences: e.g. $E(s,s) \wedge \neg E(t,s)$.

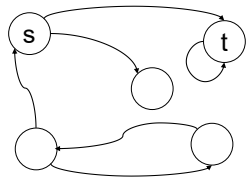
Counting the Number of Sentences in FO[k]

Fix n Boolean variables: $X_1, X_2, \ldots, X_n$.

How many Boolean functions exists? E.g. $X_1 \wedge X_2$; or $X_1 \vee \neg X_3 \wedge X_4$, ...

Answer: 2^{2^n}

Graphs with source and target: $\sigma = (E, s, t)$.



FO[0] atoms without variables: $E(s,s), E(s,t), E(t,s), E(t,t)$. 2^2

FO[0] sentences: e.g. $E(s,s) \wedge \neg E(t,s)$. $2^{2^{2^2}}$

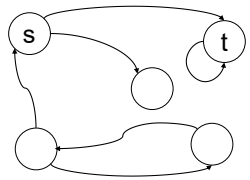
Counting the Number of Sentences in FO[k]

Fix n Boolean variables: $X_1, X_2, \ldots, X_n$.

How many Boolean functions exists? E.g. $X_1 \wedge X_2$; or $X_1 \vee \neg X_3 \wedge X_4$, ...

Answer: 2^{2^n}

Graphs with source and target: $\sigma = (E, s, t)$.



FO[0] atoms without variables: $E(s,s), E(s,t), E(t,s), E(t,t)$. 2^2

FO[0] sentences: e.g. $E(s,s) \wedge \neg E(t,s)$. $2^{2^{2^2}}$

FO[0] atoms with 1 variable x: $E(x,x) \wedge \neg E(x,s), \ldots$.

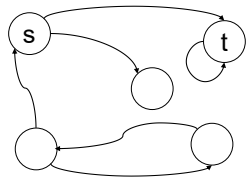
Counting the Number of Sentences in FO[k]

Fix n Boolean variables: $X_1, X_2, \ldots, X_n$.

How many Boolean functions exists? E.g. $X_1 \wedge X_2$; or $X_1 \vee \neg X_3 \wedge X_4$, ...

Answer: 2^{2^n}

Graphs with source and target: $\sigma = (E, s, t)$.



FO[0] atoms without variables: $E(s,s), E(s,t), E(t,s), E(t,t)$. 2^2

FO[0] sentences: e.g. $E(s,s) \wedge \neg E(t,s)$. $2^{2^{2^2}}$

FO[0] atoms with 1 variable x: $E(x,x) \wedge \neg E(x,s), \ldots$. 3^2

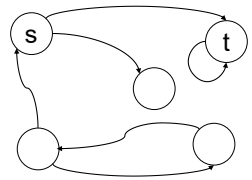
Counting the Number of Sentences in FO[k]

Fix n Boolean variables: $X_1, X_2, \ldots, X_n$.

How many Boolean functions exists? E.g. $X_1 \wedge X_2$; or $X_1 \vee \neg X_3 \wedge X_4$, ...

Answer: 2^{2^n}

Graphs with source and target: $\sigma = (E, s, t)$.



FO[0] atoms without variables: $E(s,s), E(s,t), E(t,s), E(t,t)$. 2^2

FO[0] sentences: e.g. $E(s,s) \wedge \neg E(t,s)$. $2^{2^{2^2}}$

FO[0] atoms with 1 variable x: $E(x,x) \wedge \neg E(x,s), \ldots$. 3^2

FO[0] formulas with 1 variable x: $2^{2^{3^2}}$

FO[1] sentences: Large, but finite.

Main Takeaway for FO[k]

FO[k] is finite!

Next: let's talk about FO[k] types.

FO[k] Types

Overview

A type in logic is everything you can say about an element, or a tuple of elements.

Types in programming languages are very simple special cases.

We will define FO[k] types and use them in the proof of the EF theorem. Later, we define types of other logics.

The EF Theorem

Theorem (Ehrenfeucht-Fraisse)

$\boldsymbol{A} \equiv_k \boldsymbol{B}$ *iff* $\boldsymbol{A} \sim_k \boldsymbol{B}$.

The EF Theorem

Theorem (Ehrenfeucht-Fraisse)

$\boldsymbol{A} \equiv_k \boldsymbol{B}$ *iff* $\boldsymbol{A} \sim_k \boldsymbol{B}$.

Part 1: (we will review)
If duplicator wins k rounds then $\boldsymbol{A} \models \varphi$ iff $\boldsymbol{B} \models \varphi$.

If $\boldsymbol{A} \models \varphi$ we use the duplicator's strategy to show $\boldsymbol{B} \models \varphi$.

The EF Theorem

Theorem (Ehrenfeucht-Fraisse)

$\boldsymbol{A} \equiv_k \boldsymbol{B}$ *iff* $\boldsymbol{A} \sim_k \boldsymbol{B}$.

Part 1: (we will review)
If duplicator wins k rounds then $\boldsymbol{A} \models \varphi$ iff $\boldsymbol{B} \models \varphi$.

If $\boldsymbol{A} \models \varphi$ we use the duplicator's strategy to show $\boldsymbol{B} \models \varphi$.

Part 2: (we will prove)
If $\boldsymbol{A}, \boldsymbol{B}$ satisfy the same FO[k] sentences, then duplicator wins k rounds

Here we use "FO-types" to inform the duplicator how to play.

Review: Proof of the Ehrenfeucht-Fraisse Theorem Part 1

If duplicator wins k rounds, then $\boldsymbol{A} \models \varphi$ iff $\boldsymbol{B} \models \varphi$.

[1] Formally: $(\boldsymbol{A}, a) \sim_{k-1} (\boldsymbol{B}, b)$; by induction $(\boldsymbol{A}, a) \equiv_{k-1} (\boldsymbol{B}, b)$, thus $\boldsymbol{B} \models \psi(b)$.

Review: Proof of the Ehrenfeucht-Fraisse Theorem Part 1

If duplicator wins k rounds, then $\boldsymbol{A} \models \varphi$ iff $\boldsymbol{B} \models \varphi$.

Base Case $k = 0$: duplicator wins in 0 rounds.
Then the constants in $\boldsymbol{A}, \boldsymbol{B}$ form a partial isomorphism, thus $\boldsymbol{A} \equiv_0 \boldsymbol{B}$.

[1] Formally: $(\boldsymbol{A}, a) \sim_{k-1} (\boldsymbol{B}, b)$; by induction $(\boldsymbol{A}, a) \equiv_{k-1} (\boldsymbol{B}, b)$, thus $\boldsymbol{B} \models \psi(b)$.

Review: Proof of the Ehrenfeucht-Fraisse Theorem Part 1

If duplicator wins k rounds, then $\boldsymbol{A} \models \varphi$ iff $\boldsymbol{B} \models \varphi$.

Induction $k > 0$: The interesting cases are $\exists x\psi$ and $\forall x\psi$.

If $\boxed{\boldsymbol{A} \models \exists x\psi(x)}$, we prove that[1] $\boxed{\boldsymbol{B} \models \exists x\psi(x)}$

[1] Formally: $(\boldsymbol{A}, a) \sim_{k-1} (\boldsymbol{B}, b)$; by induction $(\boldsymbol{A}, a) \equiv_{k-1} (\boldsymbol{B}, b)$, thus $\boldsymbol{B} \models \psi(b)$.

Review: Proof of the Ehrenfeucht-Fraisse Theorem Part 1

If duplicator wins k rounds, then $\boldsymbol{A} \models \varphi$ iff $\boldsymbol{B} \models \varphi$.

Induction $k > 0$: The interesting cases are $\exists x\psi$ and $\forall x\psi$.

If $\boxed{\boldsymbol{A} \models \exists x\psi(x)}$, we prove that[1] $\boxed{\boldsymbol{B} \models \exists x\psi(x)}$

There exists $a \in A$ such that $\boldsymbol{A} \models \psi(a)$.
Use the duplicator's strategy to find $b \in B$ s.t. $\boldsymbol{B} \models \psi(b)$.

[1] Formally: $(\boldsymbol{A}, a) \sim_{k-1} (\boldsymbol{B}, b)$; by induction $(\boldsymbol{A}, a) \equiv_{k-1} (\boldsymbol{B}, b)$, thus $\boldsymbol{B} \models \psi(b)$.

Review: Proof of the Ehrenfeucht-Fraisse Theorem Part 1

If duplicator wins k rounds, then $\boldsymbol{A} \models \varphi$ iff $\boldsymbol{B} \models \varphi$.

Induction $k > 0$: The interesting cases are $\exists x\psi$ and $\forall x\psi$.

If $\boxed{\boldsymbol{A} \models \exists x\psi(x)}$, we prove that[1] $\boxed{\boldsymbol{B} \models \exists x\psi(x)}$

There exists $a \in A$ such that $\boldsymbol{A} \models \psi(a)$.
Use the duplicator's strategy to find $b \in B$ s.t. $\boldsymbol{B} \models \psi(b)$.

If $\boxed{\boldsymbol{A} \models \forall x\psi(x)}$ then $\boxed{\boldsymbol{B} \models \forall x\psi(x)}$. Proof by contradiction:

If there exists $b \in B$ s.t. $\boldsymbol{B} \models \neg\psi(b)$,
Use duplicator to find $a \in \boldsymbol{A}$ s.t. $\boldsymbol{A} \models \neg\psi(a)$. Contradiction.

[1] Formally: $(\boldsymbol{A}, a) \sim_{k-1} (\boldsymbol{B}, b)$; by induction $(\boldsymbol{A}, a) \equiv_{k-1} (\boldsymbol{B}, b)$, thus $\boldsymbol{B} \models \psi(b)$.

Discussion

- Existential quantifier: spoiler plays on $\boldsymbol{A}$, duplicator plays on $\boldsymbol{B}$.

- Universal quantifier: spoiler plays on $\boldsymbol{B}$, duplicator plays on $\boldsymbol{A}$.

Proof of the Ehrenfeucht-Fraisse Theorem Part 2

If $\boldsymbol{A} \models \varphi$ iff $\boldsymbol{B} \models \varphi$, then duplicator wins k rounds.

Proof of the Ehrenfeucht-Fraisse Theorem Part 2

If $\boldsymbol{A} \models \varphi$ iff $\boldsymbol{B} \models \varphi$, then duplicator wins k rounds.

Suppose spoiler placed the third pebble a_3:

$$a_1, a_2, a_3 \in A$$
$$b_1, b_2 \in B$$

Proof of the Ehrenfeucht-Fraisse Theorem Part 2

If $\boldsymbol{A} \models \varphi$ iff $\boldsymbol{B} \models \varphi$, then duplicator wins k rounds.

Suppose spoiler placed the third pebble a_3:

$$a_1, a_2, a_3 \in A$$
$$b_1, b_2 \in B$$

a_1, a_2 and b_1, b_2 form a partial isomorphism.

Where should the duplicator place $b_3 \in B$?

Proof of the Ehrenfeucht-Fraisse Theorem Part 2

If $\boldsymbol{A} \models \varphi$ iff $\boldsymbol{B} \models \varphi$, then duplicator wins k rounds.

Suppose spoiler placed the third pebble a_3:

$$a_1, a_2, a_3 \in A$$
$$b_1, b_2 \in B$$

a_1, a_2 and b_1, b_2 form a partial isomorphism.

Where should the duplicator place $b_3 \in B$?

The partial isomorphism is not sufficient for us to help duplicator find b_3. We need to strengthen the requirement on the duplicator to:

Proof of the Ehrenfeucht-Fraisse Theorem Part 2

If $\boldsymbol{A} \models \varphi$ iff $\boldsymbol{B} \models \varphi$, then duplicator wins k rounds.

Suppose spoiler placed the third pebble a_3:

$$a_1, a_2, a_3 \in A$$
$$b_1, b_2 \in B$$

a_1, a_2 and b_1, b_2 form a partial isomorphism.

Where should the duplicator place $b_3 \in B$?

The partial isomorphism is not sufficient for us to help duplicator find b_3. We need to strengthen the requirement on the duplicator to:

$$\boldsymbol{a}, \boldsymbol{b} \text{ have the same FO}[k]\text{-Types}$$

FO[k]-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:

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

Discussion

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

- $\mathrm{tp}_{k,m}(\boldsymbol{A}, \boldsymbol{a})$ is complete:
  For all $\varphi \in FO[k]$, either $\varphi \in \mathrm{tp}_{k,m}(\boldsymbol{A}, \boldsymbol{a})$ or $\neg\varphi \in \mathrm{tp}_{k,m}(\boldsymbol{A}, \boldsymbol{a})$

Discussion

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

- $\mathrm{tp}_{k,m}(\boldsymbol{A}, \boldsymbol{a})$ is complete:
  For all $\varphi \in FO[k]$, either $\varphi \in \mathrm{tp}_{k,m}(\boldsymbol{A}, \boldsymbol{a})$ or $\neg\varphi \in \mathrm{tp}_{k,m}(\boldsymbol{A}, \boldsymbol{a})$

- Equivalent definition: an k, m-type is a complete, consistent set of formulas with $qr \leq k$ and m free variables $x_1, \ldots, x_m$.
  Note: may differ for finite or infinite: e.g. end-of-the-line.

Discussion

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

- $\mathrm{tp}_{k,m}(\boldsymbol{A}, \boldsymbol{a})$ is complete:
  For all $\varphi \in FO[k]$, either $\varphi \in \mathrm{tp}_{k,m}(\boldsymbol{A}, \boldsymbol{a})$ or $\neg\varphi \in \mathrm{tp}_{k,m}(\boldsymbol{A}, \boldsymbol{a})$

- Equivalent definition: an k, m-type is a complete, consistent set of formulas with $qr \leq k$ and m free variables $x_1, \ldots, x_m$.
  Note: may differ for finite or infinite: e.g. end-of-the-line.

- What is $\mathrm{tp}_{k,0}$?

Discussion

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

- $\mathrm{tp}_{k,m}(\boldsymbol{A}, \boldsymbol{a})$ is complete:
  For all $\varphi \in FO[k]$, either $\varphi \in \mathrm{tp}_{k,m}(\boldsymbol{A}, \boldsymbol{a})$ or $\neg\varphi \in \mathrm{tp}_{k,m}(\boldsymbol{A}, \boldsymbol{a})$

- Equivalent definition: an k, m-type is a complete, consistent set of formulas with $qr \leq k$ and m free variables $x_1, \ldots, x_m$.
  Note: may differ for finite or infinite: e.g. end-of-the-line.

- What is $\mathrm{tp}_{k,0}$? Set of FO[k] sentences true in $\boldsymbol{A}$.

Discussion

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

- $\mathrm{tp}_{k,m}(\boldsymbol{A}, \boldsymbol{a})$ is complete:
  For all $\varphi \in FO[k]$, either $\varphi \in \mathrm{tp}_{k,m}(\boldsymbol{A}, \boldsymbol{a})$ or $\neg\varphi \in \mathrm{tp}_{k,m}(\boldsymbol{A}, \boldsymbol{a})$
- Equivalent definition: an k, m-type is a complete, consistent set of formulas with $qr \leq k$ and m free variables $x_1, \ldots, x_m$.
  Note: may differ for finite or infinite: e.g. end-of-the-line.
- What is $\mathrm{tp}_{k,0}$? Set of FO[k] sentences true in $\boldsymbol{A}$.
- What is $\mathrm{tp}_{0,m}$??

Discussion

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

- $\mathrm{tp}_{k,m}(\boldsymbol{A}, \boldsymbol{a})$ is complete:
  For all $\varphi \in FO[k]$, either $\varphi \in \mathrm{tp}_{k,m}(\boldsymbol{A}, \boldsymbol{a})$ or $\neg\varphi \in \mathrm{tp}_{k,m}(\boldsymbol{A}, \boldsymbol{a})$
- Equivalent definition: an k, m-type is a complete, consistent set of formulas with $qr \leq k$ and m free variables $x_1, \ldots, x_m$.
  Note: may differ for finite or infinite: e.g. end-of-the-line.
- What is $\mathrm{tp}_{k,0}$? Set of FO[k] sentences true in $\boldsymbol{A}$.
- What is $\mathrm{tp}_{0,m}$?? Boolean expressions on atoms with constants.

Proof of the Ehrenfeucht-Fraisse Theorem Part 2

Will prove that $\boldsymbol{A} \equiv_k \boldsymbol{B}$ implies $\boldsymbol{A} \sim_k \boldsymbol{B}$.

We design a strategy for the duplicator that ensures, for $i = 0, 1, \ldots, k$:

$$\boxed{\mathrm{tp}_{k-i,i}(\boldsymbol{A}, \boldsymbol{a}) = \mathrm{tp}_{k-i,i}(\boldsymbol{B}, \boldsymbol{b})}$$

Proof of the Ehrenfeucht-Fraisse Theorem Part 2

Will prove that $\boldsymbol{A} \equiv_k \boldsymbol{B}$ implies $\boldsymbol{A} \sim_k \boldsymbol{B}$.

We design a strategy for the duplicator that ensures, for $i = 0, 1, \ldots, k$:

$$\boxed{\mathrm{tp}_{k-i,i}(\boldsymbol{A}, \boldsymbol{a}) = \mathrm{tp}_{k-i,i}(\boldsymbol{B}, \boldsymbol{b})}$$

Why is this sufficient?

Proof of the Ehrenfeucht-Fraisse Theorem Part 2

Will prove that $\boldsymbol{A} \equiv_k \boldsymbol{B}$ implies $\boldsymbol{A} \sim_k \boldsymbol{B}$.

We design a strategy for the duplicator that ensures, for $i = 0, 1, \ldots, k$:

$$\boxed{\mathrm{tp}_{k-i,i}(\boldsymbol{A}, \boldsymbol{a}) = \mathrm{tp}_{k-i,i}(\boldsymbol{B}, \boldsymbol{b})}$$

Why is this sufficient?

After k rounds, $\mathrm{tp}_{0,k}(\boldsymbol{A}, \boldsymbol{a}) = \mathrm{tp}_{0,k}(\boldsymbol{B}, \boldsymbol{b})$, thus $\boldsymbol{a}, \boldsymbol{b}$ partial isomorphism.

Proof of the Ehrenfeucht-Fraisse Theorem Part 2

Will prove that $\boldsymbol{A} \equiv_k \boldsymbol{B}$ implies $\boldsymbol{A} \sim_k \boldsymbol{B}$.

We design a strategy for the duplicator that ensures, for $i = 0, 1, \ldots, k$:

$$\boxed{\mathrm{tp}_{k-i,i}(\boldsymbol{A}, \boldsymbol{a}) = \mathrm{tp}_{k-i,i}(\boldsymbol{B}, \boldsymbol{b})}$$

Proof by induction on i.

When $i = 0$: $\boldsymbol{A} \equiv_k \boldsymbol{B}$ is same as $\mathrm{tp}_{k,0}(\boldsymbol{A}) = \mathrm{tp}_{k,0}(\boldsymbol{B})$.

Proof of the Ehrenfeucht-Fraisse Theorem Part 2

Will prove that $\boldsymbol{A} \equiv_k \boldsymbol{B}$ implies $\boldsymbol{A} \sim_k \boldsymbol{B}$.

We design a strategy for the duplicator that ensures, for $i = 0, 1, \ldots, k$:

$$\boxed{\mathrm{tp}_{k-i,i}(\boldsymbol{A}, \boldsymbol{a}) = \mathrm{tp}_{k-i,i}(\boldsymbol{B}, \boldsymbol{b})}$$

Induction step Assume $\mathrm{tp}_{k-i,i}(\boldsymbol{A}, \boldsymbol{a}) = \mathrm{tp}_{k-i,i}(\boldsymbol{B}, \boldsymbol{b})$; spoiler picks $a_{i+1} \in A$.

Proof of the Ehrenfeucht-Fraisse Theorem Part 2

Will prove that $\boldsymbol{A} \equiv_k \boldsymbol{B}$ implies $\boldsymbol{A} \sim_k \boldsymbol{B}$.

We design a strategy for the duplicator that ensures, for $i = 0, 1, \ldots, k$:

$$\boxed{\mathrm{tp}_{k-i,i}(\boldsymbol{A}, \boldsymbol{a}) = \mathrm{tp}_{k-i,i}(\boldsymbol{B}, \boldsymbol{b})}$$

Induction step Assume $\mathrm{tp}_{k-i,i}(\boldsymbol{A}, \boldsymbol{a}) = \mathrm{tp}_{k-i,i}(\boldsymbol{B}, \boldsymbol{b})$; spoiler picks $a_{i+1} \in A$.

We need to help duplicator find "the right" $b \in B$.

Proof of the Ehrenfeucht-Fraisse Theorem Part 2

Will prove that $\boldsymbol{A} \equiv_k \boldsymbol{B}$ implies $\boldsymbol{A} \sim_k \boldsymbol{B}$.

We design a strategy for the duplicator that ensures, for $i = 0, 1, \ldots, k$:

$$\boxed{\mathrm{tp}_{k-i,i}(\boldsymbol{A}, \boldsymbol{a}) = \mathrm{tp}_{k-i,i}(\boldsymbol{B}, \boldsymbol{b})}$$

Induction step Assume $\mathrm{tp}_{k-i,i}(\boldsymbol{A}, \boldsymbol{a}) = \mathrm{tp}_{k-i,i}(\boldsymbol{B}, \boldsymbol{b})$; spoiler picks $a_{i+1} \in A$.

We need to help duplicator find "the right" $b \in B$.

$\boldsymbol{B}$ is finite, $B = \{u_1, u_2, \ldots, u_n\}$, and assume u_j is "wrong":

$$\boxed{\mathrm{tp}_{k-i-1,i+1}(\boldsymbol{A}, (\boldsymbol{a}, a_{i+1})) \neq \mathrm{tp}_{k-i-1,i+1}(\boldsymbol{B}, (\boldsymbol{b}, u_j))}$$

Proof of the Ehrenfeucht-Fraisse Theorem Part 2

Will prove that $\boldsymbol{A} \equiv_k \boldsymbol{B}$ implies $\boldsymbol{A} \sim_k \boldsymbol{B}$.

We design a strategy for the duplicator that ensures, for $i = 0, 1, \ldots, k$:

$$\boxed{\mathrm{tp}_{k-i,i}(\boldsymbol{A}, \boldsymbol{a}) = \mathrm{tp}_{k-i,i}(\boldsymbol{B}, \boldsymbol{b})}$$

Induction step Assume $\mathrm{tp}_{k-i,i}(\boldsymbol{A}, \boldsymbol{a}) = \mathrm{tp}_{k-i,i}(\boldsymbol{B}, \boldsymbol{b})$; spoiler picks $a_{i+1} \in A$.

We need to help duplicator find "the right" $b \in B$.

$\boldsymbol{B}$ is finite, $B = \{u_1, u_2, \ldots, u_n\}$, and assume u_j is "wrong":

$$\boxed{\mathrm{tp}_{k-i-1,i+1}(\boldsymbol{A}, (\boldsymbol{a}, a_{i+1})) \neq \mathrm{tp}_{k-i-1,i+1}(\boldsymbol{B}, (\boldsymbol{b}, u_j))}$$

There exists $\varphi_j \in \mathrm{FO}[k-i-1]$ s.t.:

$$\boldsymbol{A} \models \varphi_j(\boldsymbol{a}, a_{i+1}) \text{ and } \boldsymbol{B} \models \neg\varphi_j(\boldsymbol{b}, u_j)$$

Proof of the Ehrenfeucht-Fraisse Theorem Part 2

Will prove that $\boldsymbol{A} \equiv_k \boldsymbol{B}$ implies $\boldsymbol{A} \sim_k \boldsymbol{B}$.

We design a strategy for the duplicator that ensures, for $i = 0, 1, \ldots, k$:

$$\boxed{\mathrm{tp}_{k-i,i}(\boldsymbol{A}, \boldsymbol{a}) = \mathrm{tp}_{k-i,i}(\boldsymbol{B}, \boldsymbol{b})}$$

Induction step Assume $\mathrm{tp}_{k-i,i}(\boldsymbol{A}, \boldsymbol{a}) = \mathrm{tp}_{k-i,i}(\boldsymbol{B}, \boldsymbol{b})$; spoiler picks $a_{i+1} \in A$.

We need to help duplicator find "the right" $b \in B$.

$\boldsymbol{B}$ is finite, $B = \{u_1, u_2, \ldots, u_n\}$, and assume u_j is "wrong":

$$\boxed{\mathrm{tp}_{k-i-1,i+1}(\boldsymbol{A}, (\boldsymbol{a}, a_{i+1})) \neq \mathrm{tp}_{k-i-1,i+1}(\boldsymbol{B}, (\boldsymbol{b}, u_j))}$$

There exists $\varphi_j \in \mathrm{FO}[k-i-1]$ s.t.:

$$\boldsymbol{A} \models \varphi_j(\boldsymbol{a}, a_{i+1}) \text{ and } \boldsymbol{B} \models \neg\varphi_j(\boldsymbol{b}, u_j)$$

Let $\varphi(x_1, \ldots, x_i, x_{i+1}) = \varphi_1 \wedge \cdots \wedge \varphi_n$. Then:

$$\boldsymbol{A} \models \exists x \varphi(a, x) \text{ and } \boldsymbol{B} \models \neg\exists x \varphi(b, x)$$

Proof of the Ehrenfeucht-Fraisse Theorem Part 2

Will prove that $\boldsymbol{A} \equiv_k \boldsymbol{B}$ implies $\boldsymbol{A} \sim_k \boldsymbol{B}$.

We design a strategy for the duplicator that ensures, for $i = 0, 1, \ldots, k$:

$$\boxed{\mathrm{tp}_{k-i,i}(\boldsymbol{A}, \boldsymbol{a}) = \mathrm{tp}_{k-i,i}(\boldsymbol{B}, \boldsymbol{b})}$$

Induction step Assume $\mathrm{tp}_{k-i,i}(\boldsymbol{A}, \boldsymbol{a}) = \mathrm{tp}_{k-i,i}(\boldsymbol{B}, \boldsymbol{b})$; spoiler picks $a_{i+1} \in A$.

We need to help duplicator find "the right" $b \in B$.

$\boldsymbol{B}$ is finite, $B = \{u_1, u_2, \ldots, u_n\}$, and assume u_j is "wrong":

$$\boxed{\mathrm{tp}_{k-i-1,i+1}(\boldsymbol{A}, (\boldsymbol{a}, a_{i+1})) \neq \mathrm{tp}_{k-i-1,i+1}(\boldsymbol{B}, (\boldsymbol{b}, u_j))}$$

There exists $\varphi_j \in \mathrm{FO}[k-i-1]$ s.t.:

$$\boldsymbol{A} \models \varphi_j(\boldsymbol{a}, a_{i+1}) \text{ and } \boldsymbol{B} \models \neg\varphi_j(\boldsymbol{b}, u_j)$$

Let $\varphi(x_1, \ldots, x_i, x_{i+1}) = \varphi_1 \wedge \cdots \wedge \varphi_n$. Then:

$$\boldsymbol{A} \models \exists x \varphi(a, x) \text{ and } \boldsymbol{B} \models \neg\exists x \varphi(b, x)$$

Contradiction, since $\varphi \in \mathrm{tp}_{k-i,i}(\boldsymbol{A}, \boldsymbol{a}) = \mathrm{tp}_{k-i,i}(\boldsymbol{B}, \boldsymbol{b})$

Discussion

- Types are defined for various logics; they are useful to reason about both formulas and elements in structures.

- Since $\mathrm{tp}_{k,m}$ is finite, there is a single formula that captures the type: $\varphi(x_1, \ldots, x_m) \stackrel{\text{def}}{=} \bigwedge_{\psi \in \mathrm{tp}_{k,m}(\boldsymbol{A}, \boldsymbol{a})} \psi(x_1, \ldots, x_m)$

- A thought experiment: replace a huge database $\boldsymbol{D}$, with the finite collection of FO[k] types that hold in $\boldsymbol{D}$. This is sufficient to answer all queries with $\leq k$ quantifiers over $\boldsymbol{D}$!

Hanf's Lemma

Describing Winning Strategies

- Recall: to prove inexpressibility, we need to show that, for all k, the duplicator has a winning strategy for k rounds: $\boldsymbol{A}_k \sim_k \boldsymbol{B}_k$.
- But difficult to prove that the duplicator has a winning strategy.
- Hanf's lemma: sufficient condition for a winning strategy.

Neighborhoods

$\sigma = (R_1, \ldots, R_m, c_1, \ldots, c_s)$.
The Gaifman graph[2] of a $\boldsymbol{A}$: edges (a, b), a, b co-occurring in a relation.

d-neighborhood of $a \in A$: $N(a, d) \stackrel{\text{def}}{=} \{b \in A \mid d(a, b) \leq d\} \cup \{c_1^A, \ldots, c_s^A\}$.

Definition

The d-type of a is the isomorphism type of the substructure induced by $N(a, d)$, plus the constant a.

Definition

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

[2] A.k.a. primal graph.

Hanf's Lemma

Theorem

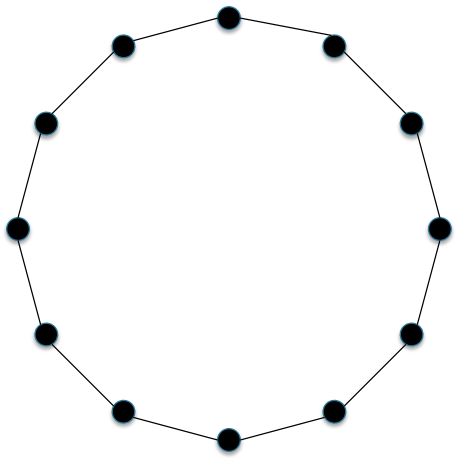
Let $d \geq 3^{k-1} - 1$. If $\boldsymbol{A}, \boldsymbol{B}$ are d-equivalent, then $\boldsymbol{A} \sim_k \boldsymbol{B}$.

The proof exhibits a winning strategy for the duplicator.

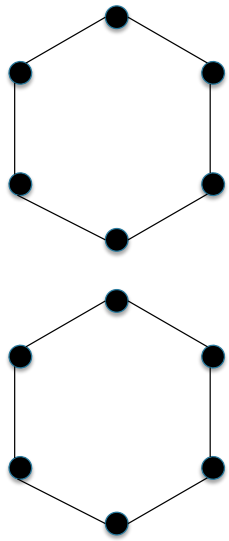
We will omit the proof.

Example: Connectivity

Prove that duplicator has winning strategy with $k = 3$ pebbles.



C_{12}

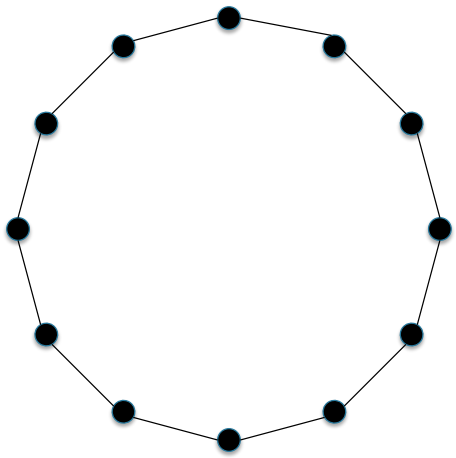


$C_6 \cup C_6$

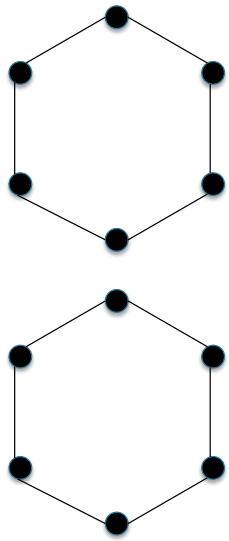
Example: Connectivity

Prove that duplicator has winning strategy with $k = 3$ pebbles.

Let's use Hanf's lemma: $k = 2$ and $d = 2 (= 3^{k-1} - 1)$



C_{12}

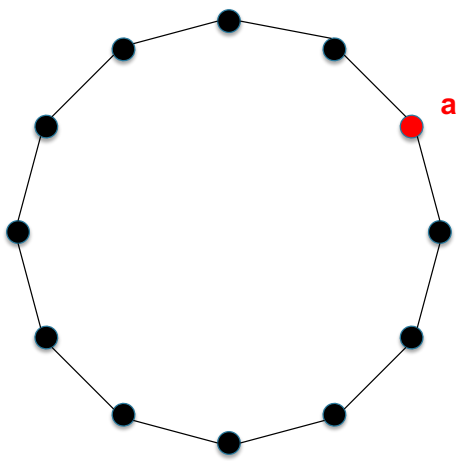


$C_6 \cup C_6$

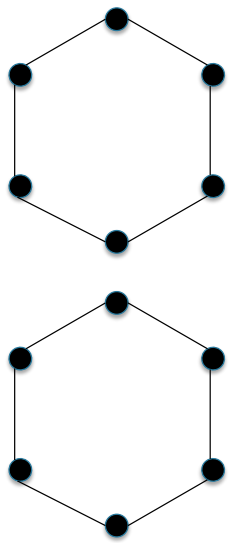
Example: Connectivity

Prove that duplicator has winning strategy with $k = 3$ pebbles.

Let's use Hanf's lemma: $k = 2$ and $d = 2 (= 3^{k-1} - 1)$



C_{12}



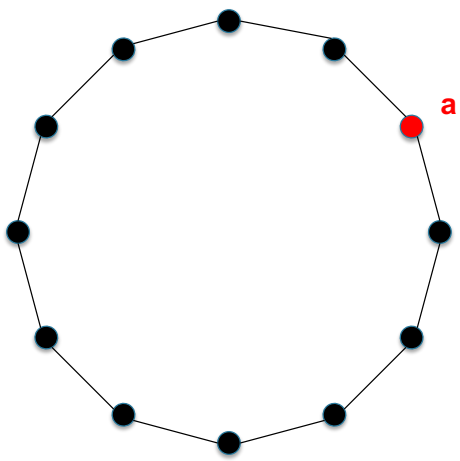
$C_6 \cup C_6$

Example: Connectivity

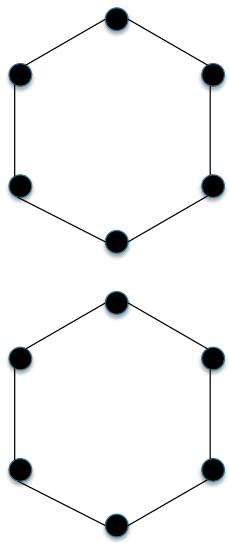
Prove that duplicator has winning strategy with $k = 3$ pebbles.

Let's use Hanf's lemma: $k = 2$ and $d = 2 (= 3^{k-1} - 1)$

What is $N(a, d)$?



C_{12}



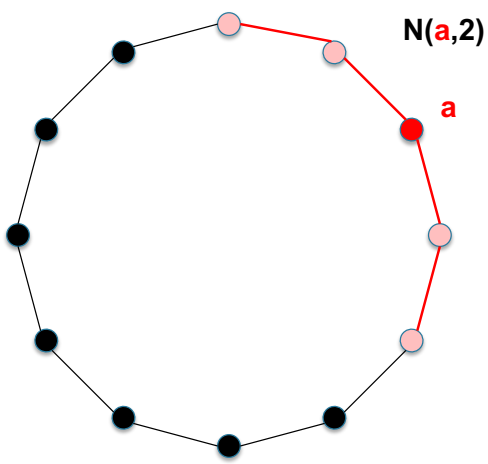
$C_6 \cup C_6$

Example: Connectivity

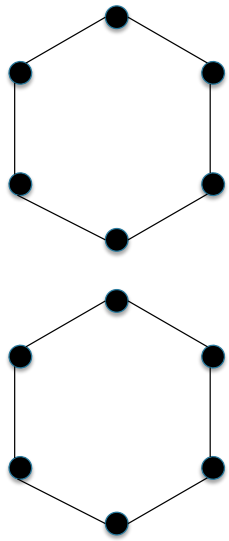
Prove that duplicator has winning strategy with $k = 3$ pebbles.

Let's use Hanf's lemma: $k = 2$ and $d = 2 (= 3^{k-1} - 1)$

What is $N(a, d)$?



C_{12}



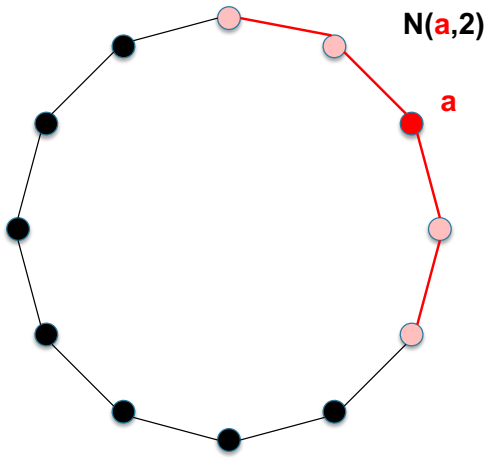
$C_6 \cup C_6$

Example: Connectivity

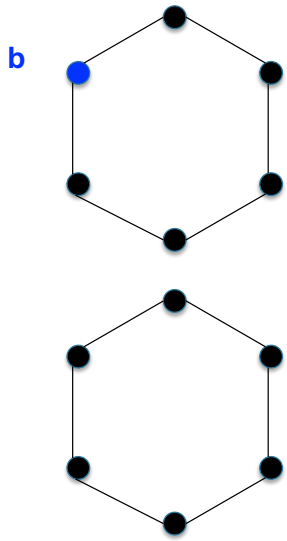
Prove that duplicator has winning strategy with $k = 3$ pebbles.

Let's use Hanf's lemma: $k = 2$ and $d = 2 (= 3^{k-1} - 1)$

What is $N(a, d)$?



C_{12}



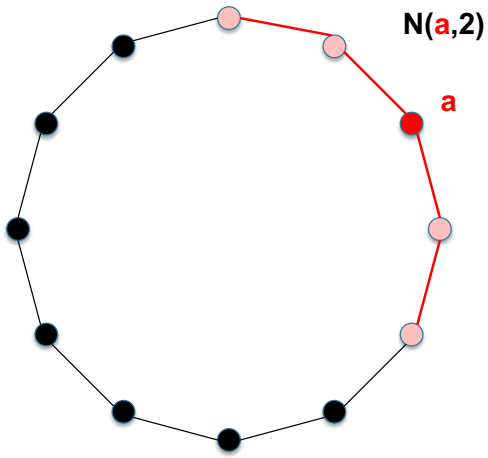
$C_6 \cup C_6$

Example: Connectivity

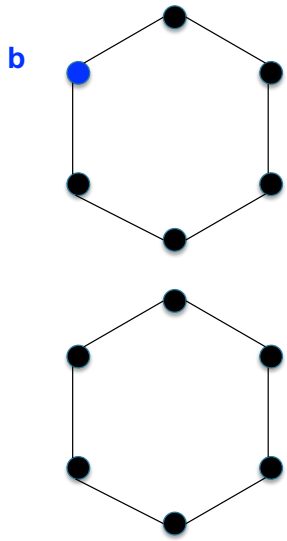
Prove that duplicator has winning strategy with $k = 3$ pebbles.

Let's use Hanf's lemma: $k = 2$ and $d = 2 (= 3^{k-1} - 1)$

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



C_{12}



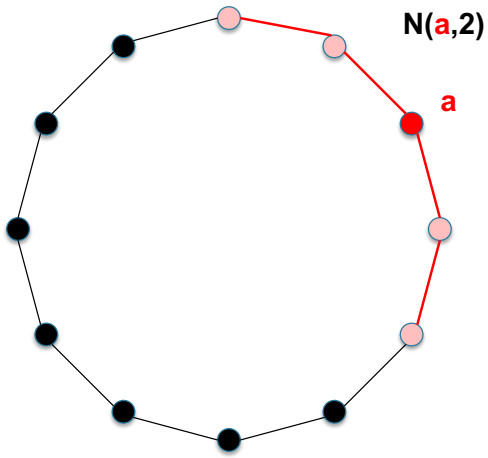
$C_6 \cup C_6$

Example: Connectivity

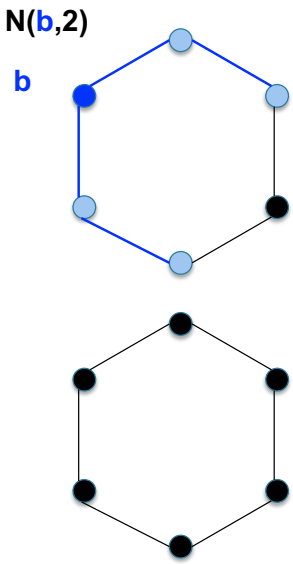
Prove that duplicator has winning strategy with $k = 3$ pebbles.

Let's use Hanf's lemma: $k = 2$ and $d = 2(= 3^{k-1} - 1)$

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



C_{12}



$C_6 \cup C_6$

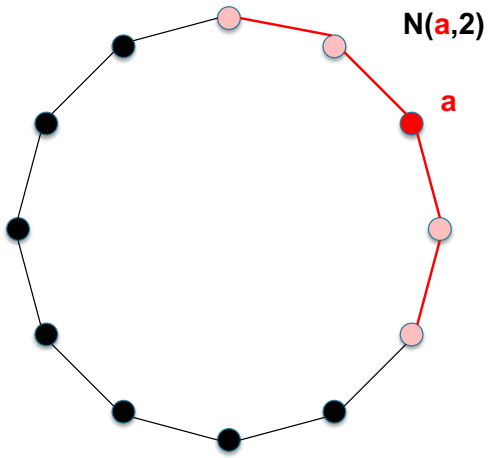
Example: Connectivity

Prove that duplicator has winning strategy with $k = 3$ pebbles.

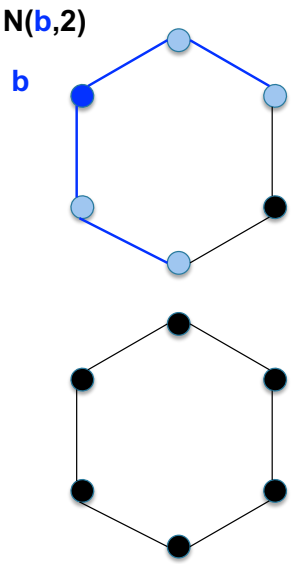
Let's use Hanf's lemma: $k = 2$ and $d = 2 (= 3^{k-1} - 1)$

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

Their types are structures $x - x - * - x - x$; there are 12 copies in each structure. Therefore, duplicator wins with $k = 2$ pebbles.



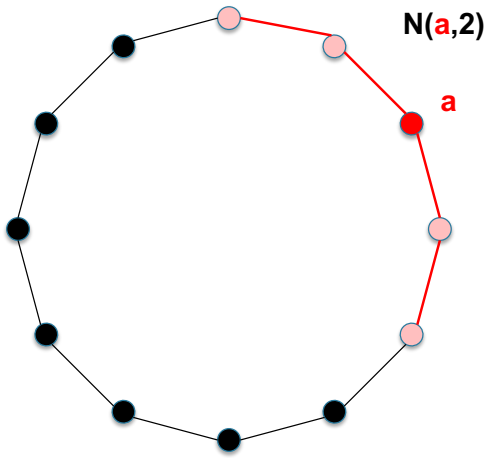
C_{12}



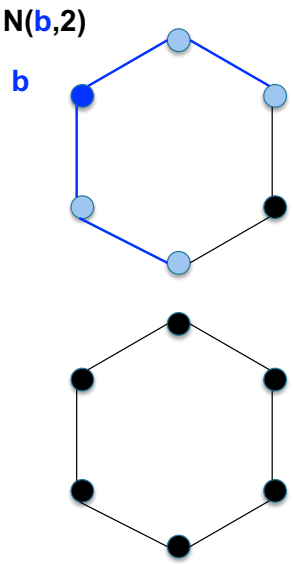
$C_6 \cup C_6$

Example: Connectivity

Prove that duplicator has winning strategy with $k = 3$ pebbles.
Let's use Hanf's lemma: $k = 2$ and $d = 2(= 3^{k-1} - 1)$
What is $N(a, d)$? What is $N(b, d)$?
Their types are structures $x - x - * - x - x$; there are 12 copies in each structure. Therefore, duplicator wins with $k = 2$ pebbles.



C_{12}



$C_6 \cup C_6$

At home: prove that spoiler wins in 3 rounds

Discussion

- Ehrenfeucht-Fraisse games can be applied to infinite structures as well! Try at home: update Part 2 of the proof of the EF theorem to remove the need for $\boldsymbol{A}, \boldsymbol{B}$ to be finite.

- EF games generalize to other games to prove inexpressibility results. We will discuss two:

  - Inexpressibility for $\exists$MSO

  - Inexpressibility for logics with recursion.