# Ehrenfeucht-Fraïssé Games

## Automata and Computational Models

National Technical University of Athens

# Outline

# Definition of the language

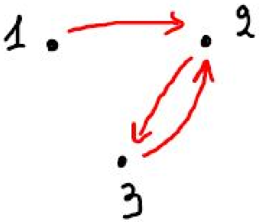
- A **vocabulary** $\tau = \langle R_1^{a_1}, ..., R_k^{a_k}, c_1, ..., c_s \rangle$ is a tuple of relational symbols and constant symbols.
- A **structure with vocabulary** $\boldsymbol{\tau}$* is a tuple $\mathcal{A} = \langle A, R_1^{\mathcal{A}}, ..., R_k^{\mathcal{A}}, c_1^{\mathcal{A}}, ..., c_s^{\mathcal{A}} \rangle$, where $A$ is the universe, a nonempty set.

*It is also called a $\tau$-structure.

# Example 1

The vocabulary of graphs is $\tau_g = \langle E^2, s, t \rangle$.

A specific "graph structure" is the structure
$\mathcal{G} = \langle \{1,2,3\}, E^{\mathcal{G}} = \{(1,2),(2,3),(3,2)\}, 1^{\mathcal{G}}, 3^{\mathcal{G}} \rangle$




*We will write $E(1,2)$ or even $1E2$.

# Example 2

The vocabulary of strings is $\tau_S = \langle \leqslant^2, S_a^1, S_b^1 \rangle$

A specific "string structure" is the structure $\mathcal{S} = \langle \{1, 2, 3, 4, 5\},$
$\leqslant^{\mathcal{S}} =$
$\{(1,1), (1,2), (1,3), ..., (1,5), (2,2), (2,3), .., (2,5), ..., (5,5)\},$
$S_a^{\mathcal{S}} = \{1, 4\},$
$S_b^{\mathcal{S}} = \{2, 3, 5\} \rangle$

1 2 3 4 5
a b b a b

*We will write $1 \leqslant 3$, $S_a(1)$, $S_b(5)$ etc.

# First order language

For any vocabulary $\tau$, define the **first order language** $\mathcal{L}(\tau)$ to be the set of formulas built up from:

- the relation and constant symbols of $\tau$,
- the logical relation symbol $=$,
- the boolean connectives $\neg, \wedge, \vee, \rightarrow$,
- variables $\{x, y, z, ...\}$ and
- quantifiers $\forall, \exists$.

- The formula $\forall x \exists y \big(x \leqslant y \wedge S_a(y)\big)$ is a first order formula in the language of strings: For every position, there is a following position that has an a.
- The formula $\exists x \forall y (\neg E(x,y) \wedge \neg E(y,x))$ is a first order formula in the language of graphs. There is a vertex such that it is not connected to any other vertex by neither an incoming nor an outgoing edge.
- What about the following formulas?
  1. $\exists x \exists y (\neg(x = y) \wedge S_a(x) \wedge S_a(y))$
  2. $\forall x (E(x,x) \vee \forall y (\neg(x = y) \rightarrow E(x,y))$

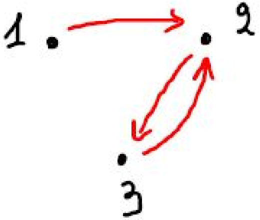
**Exercise:** Write the following properties using first order formulas in the vocabulary $\tau_S = \langle \leqslant^2, S_a^1, S_b^1 \rangle$ of strings:

1. $\mathit{first}_{\leqslant}(x)$: $x$ is the first of all elements in the universe
2. $\mathit{last}_{\leqslant}(x)$: $x$ is the last of all elements in the universe
3. $\mathit{succ}_{\leqslant}(x, y)$: $y$ is the successor of $x$

# Truth in a structure

For a structure $\mathcal{A}$ and a formula $\phi$, we write $\mathcal{A} \models \phi$ iff "$\phi$ is true in $\mathcal{A}$" or "$\mathcal{A}$ satisfies $\phi$".

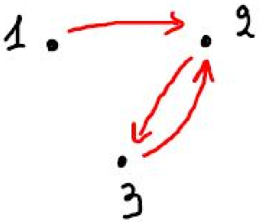
**Example 1:** Can we say that $\mathcal{G} \models \exists x \forall y (\neg E(x, y) \wedge \neg E(y, x))$?



# Truth in a structure

For a structure $\mathcal{A}$ and a formula $\phi$, we write $\mathcal{A} \models \phi$ iff "$\phi$ is true in $\mathcal{A}$" or "$\mathcal{A}$ satisfies $\phi$".

**Example 1:** Can we say that $\mathcal{G} \models \exists x \forall y (\neg E(x,y) \wedge \neg E(y,x))$?




No! So we write $\mathcal{G} \not\models \exists x \forall y (\neg E(x,y) \wedge \neg E(y,x))$.

$\mathcal{S}$ is the structure that corresponds to string abbab.

**Example 2:** Can we say that $\mathcal{S} \models \forall x \exists y \big(x \leqslant y \wedge S_a(y)\big)$ ?

$\mathcal{S}$ is the structure that corresponds to string abbab.

**Example 2:** Can we say that $\mathcal{S} \models \forall x \exists y \big( x \leqslant y \wedge S_a(y) \big)$ ?

**Example 3:** Can we say that $\mathcal{S} \models \forall x \exists y \big( x \leqslant y \wedge S_b(y) \big)$ ?

$\mathcal{S}$ is the structure that corresponds to string abbab.

**Example 2:** Can we say that $\mathcal{S} \models \forall x \exists y \big(x \leqslant y \wedge S_a(y)\big)$ ?

**Example 3:** Can we say that $\mathcal{S} \models \forall x \exists y \big(x \leqslant y \wedge S_b(y)\big)$ ?

**Example 4:** Can we say that

$$\mathcal{S} \models \exists x \exists y (\neg(x = y) \wedge S_a(x) \wedge S_a(y))\,?$$

# A step forward...

- A formula can define a language!
- For example, the formula $\forall x \left[ last_{\leqslant}(x) \rightarrow S_a(x) \right]$ defines the language of strings that end with an $a$, which is a regular language.

## Definition 1

Let $\mathcal{L}$ be a logic and $C$ a class of $\tau$-structures. A property $P$ is **$\mathcal{L}$-definable** on $C$ if there is a sentence $\psi$ such that for every structure $\mathcal{A} \in C$

$$\mathcal{A} \models \psi \text{ iff } \mathcal{A} \text{ has property } P.$$

# Motivation

- What is the expressive power of first order logic?
- Can we define all regular languages in first order logic?
- If not, which logic has the expressive power to define all regular languages?

- Ehrenfeucht-Fraïssé games help us to prove that the property EVEN is not first-order definable.
- The regular language that contains the strings in $\Sigma = \{a\}$ with even length is not first-order definable.

**Theorem 2 (Büchi)**

*A language is definable in Monadic Second Order Logic (MSO)*

*iff*

*it is regular*

- **Monadic Second Order Logic** is an extension of First Order Logic.
- **Second order:** We also have second-order variables ranging over sets and relations on the universe and quantification over such variables.
- **Monadic:** The second-order variables have arity one. In other words, the second order variables correspond to sets.

# The property EVEN is definable in MSO on strings

A structure $\mathcal{S}$ corresponds to a string with even length if it satisfies the following formula $\phi_{EVEN}$:

$$\exists X \left( \begin{array}{l} \quad \forall x \ (\mathrm{first}_{\leqslant}(x) \rightarrow X(x)) \\ \wedge \ \forall x \ (\mathrm{last}_{\leqslant}(x) \rightarrow \neg X(x)) \\ \wedge \ \forall x \forall y \ \mathrm{succ}_{\leqslant}(x, y) \rightarrow (X(x) \leftrightarrow \neg X(y)) \end{array} \right)$$

**Example:**
$\mathcal{S}_2 = \langle \{1, 2, 3, 4, 5, 6\}, \leqslant, S_a = \{1, 2\}, S_b = \{3, 4, 5, 6\} \rangle$, which corresponds to aabbbb, satisfies $\phi_{EVEN}$.

$X = \{1, 3, 5\}$ is the set described.

# Investigating the expressive power of First Order Logic

- **Goal:** Prove that a property of finite structures is not definable in FO.

# Investigating the expressive power of First Order Logic

- **Goal:** Prove that a property of finite structures is not definable in FO.
- **Tool:** Ehrenfeucht-Fraïssé games.

# Rules of the Game

- The game is played by two players called S(or spoiler) and D(or duplicator).

# Rules of the Game

- The game is played by two players called S(or spoiler) and D(or duplicator).
- The game is played on two structures **A** and **B** over the same vocabulary $\tau$.

# Rules of the Game

- The game is played by two players called S(or spoiler) and D(or duplicator).
- The game is played on two structures **A** and **B** over the same vocabulary $\tau$.
- The game is played for a predetermined positive integer k number of rounds.

# Rules of the Game

- In each round i, S picks an element of one of the two structures. Then D picks an element of the other structure.

# Rules of the Game

- In each round i, S picks an element of one of the two structures. Then D picks an element of the other structure.
- Each round produces a pair $(a_i, b_i)$ where $a_i \in \mathbf{A}$, $b_i \in \mathbf{B}$

# Rules of the Game

- In each round i, S picks an element of one of the two structures. Then D picks an element of the other structure.
- Each round produces a pair $(a_i, b_i)$ where $a_i \in \mathbf{A}, b_i \in \mathbf{B}$
- D wins the run if the mapping

$$a_i \mapsto b_i, 1 \leq i \leq k \quad \text{and} \quad c_j^A \mapsto c_j^B, 1 \leq j \leq s$$

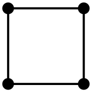
  is a partial isomorphism form **A** to **B**.
- Otherwise S wins the run.

# Rules of the Game

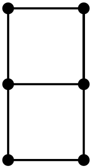
- D has a $k$-round winning strategy on $\mathbf{A}$ and $\mathbf{B}$ if he can play in a way that guarantees a winning position after $k$ rounds no matter how S plays.
- Then we write $\mathbf{A} \equiv_k \mathbf{B}$.

# Examples

**A**

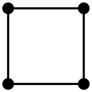


**B**

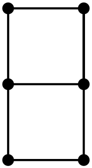


# Examples

**A**

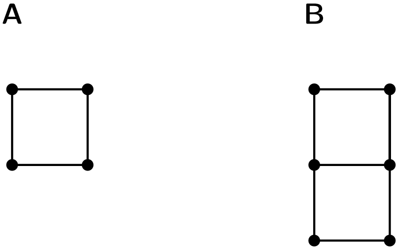


**B**



- D has a winning strategy for the 2-move game.

# Examples




- D has a winning strategy for the 2-move game.
- S has a winning strategy for the 3-move game.

# Examples

- Why does S have a winning strategy for the 3-move game?

# Examples

- Why does S have a winning strategy for the 3-move game?
- We can find a sentence that is true for **B** and false for **A**

$$\exists x \exists y \exists z ((x \neq y) \wedge (x \neq z) \wedge (y \neq z) \wedge \neg E(x,y) \wedge \neg E(x,z) \wedge \neg E(y,z))$$

# Examples

- Why does S have a winning strategy for the 3-move game?
- We can find a sentence that is true for **B** and false for **A**

$$\exists x \exists y \exists z((x \neq y) \wedge (x \neq z) \wedge (y \neq z) \wedge \neg E(x, y) \wedge \neg E(x, z) \wedge \neg E(y, z))$$

- Or a sentence that is true for **A** and false for **B**

$$\forall x \forall y \exists z((x \neq y \wedge (E(x, y) \vee E(y, z)))$$

# Examples

- Why does S have a winning strategy for the 3-move game?
- We can find a sentence that is true for **B** and false for **A**

$$\exists x \exists y \exists z ((x \neq y) \wedge (x \neq z) \wedge (y \neq z) \wedge \neg E(x,y) \wedge \neg E(x,z) \wedge \neg E(y,z))$$

- Or a sentence that is true for **A** and false for **B**

$$\forall x \forall y \exists z ((x \neq y \wedge (E(x,y) \vee E(y,z)))$$

- What do these sentences have in common?
  They have quantifier rank 3.

# The Ehrenfeucht-Fraïssé Theorem

**Theorem 3**

*The following are equivalent:*

1. ***A*** *and* ***B*** *agree on all first-order formulas of quantifier rank k*
2. $\boldsymbol{A} \equiv_k \boldsymbol{B}$ *(the duplicator has a k-round winning strategy)*

# The Ehrenfeucht-Fraïssé Theorem

## Theorem 3

*The following are equivalent:*

*1. **A** and **B** agree on all first-order formulas of quantifier rank k*

*2. **A** $\equiv_k$ **B** (the duplicator has a k-round winning strategy)*

How can we use this theorem to prove that a property is not definable in *FO*?

# Method

- Suppose we have property P.
- For $k \in \mathbb{N}$, find two structures $\mathbf{A}_k$ and $\mathbf{B}_k$ such that:
  1. $\mathbf{A} \equiv_k \mathbf{B}$
  2. $\mathbf{A}_k$ has property P and $\mathbf{B}_k$ does not have property P
- Then $\mathbf{A}_k$ and $\mathbf{B}_k$ agree on all first-order formulas with quantifier rank $k$, but they don't agree on P.
- So P cannot be defined by a first order formula with quantifier rank $k$.
- Do this for every $k$.

# Examples

The EVEN CARDINALITY query is not *FO* definable on the class of all finite graphs.

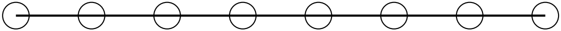
# Examples

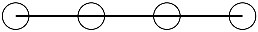
The EVEN CARDINALITY query is not *FO* definable on the class of all finite graphs.

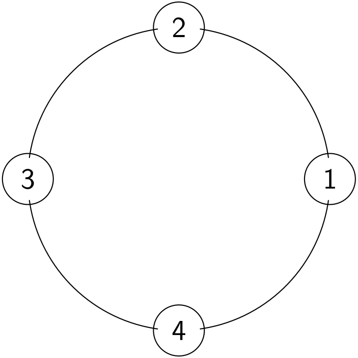
### Proof.

For every $k \in \mathbb{N}$ let $\mathbf{A}_k$ be the totally disconnected graph with $2k$ nodes, and $\mathbf{B}_k$ be the totally disconnected graph with $2k+1$ nodes. For every k, D wins the game trivially, but one graph has even nodes and the other one odd. $\square$

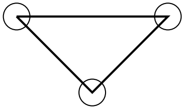
# Exercises: ACYCLICITY

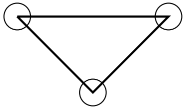


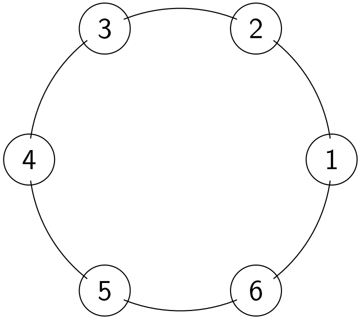




# Exercises: 2-COLORABILITY







# References


- Libkin, Leonid. Elements of finite model theory.
- Kolaitis, Phokion. On the expressive power of Logics on Finite Models.
- Immerman, Neil. Descriptive complexity.
- Van Benthem, Johan. Logic in Games.