Descriptive Complexity: Ehrenfeucht-Fraïssé Games

Konstantakatos Vangelis

National Technical University of Athens

Outline

Motivation

- In Descriptive Complexity we study the connections between the expressibility of logics and computational complexity.

Motivation

- In Descriptive Complexity we study the connections between the expressibility of logics and computational complexity.
- Naturally studying the expressive power of logics is part of this study!

Definition 1

Let L be a logic and C a class of σ-structures. A boolean query Q on C is **L-definable** on C if there is a sentence ψ such that for every $\mathbf{A} \in C$

$$Q(\mathbf{A}) = 1 \iff A \models \psi$$

Examples

Ex1.

On the class of all graphs the boolean query "the graph G has an isolated node is definable by the FO sentence:

$$(\exists x)(\forall y)(\neg E(x,y) \wedge \neg E(y,x))$$

Ex2.

On the class of binary strings the boolean query "the string has at least two 0's" is definable by the FO sentence:

$$(\exists x)(\exists y)(\neg(x = y) \wedge \neg P(x) \wedge \neg P(y))$$

Remarks

- The expressive power of a logic L depends on the class C of structures on which it is studied.

Remarks

- The expressive power of a logic L depends on the class C of structures on which it is studied.
- The same sentence must define the query an all structures of the class.

Remarks

- The expressive power of a logic L depends on the class C of structures on which it is studied.
- The same sentence must define the query an all structures of the class.
- Proving that a query is definable in a logic can be very straightforward. Write down the sentence that defines it!

Remarks

- The expressive power of a logic L depends on the class C of structures on which it is studied.
- The same sentence must define the query an all structures of the class.
- Proving that a query is definable in a logic can be very straightforward. Write down the sentence that defines it!
- Proving that a query is not definable seems more challenging since we have to show that no sentence of L defines it.

Remarks

- We see some analogies with complexity theory.

Remarks

- We see some analogies with complexity theory.
- Constructing and algorithm provided an upper bound on the complexity of a problem.
- Finding lower bounds is generally much harder.

Investigating the expressive power of First Order Logic

- In this chapter we will focus on First Order Logic.

Investigating the expressive power of First Order Logic

- In this chapter we will focus on First Order Logic.
- Do we have tools at our disposal to prove that a query is not definable in *FO*?

Investigating the expressive power of First Order Logic

- In this chapter we will focus on First Order Logic.
- Do we have tools at our disposal to prove that a query is not definable in FO?
- Yes we have some very powerful tools from Model Theory, such as the compactness theorem!

Compactness

Theorem 2

A set of FO sentences T (Theory) has a model iff every finite subset of T has a model.

Compactness

We will use the compactness theorem to prove that Connectivity is not definable in *FO* over the class of arbitrary graphs(finite or infinite).

Proof

Let ψ be a *FO* sentence such that for every $G = (V, E)$, $G \models \psi$ iff G is connected.

Compactness

We will use the compactness theorem to prove that Connectivity is not definable in *FO* over the class of arbitrary graphs(finite or infinite).

Proof

Let ψ be a *FO* sentence such that for every $G = (V, E)$, $G \models \psi$ iff G is connected. Let s, e be constants and for every $n \geq 1$ let ϕ_n be the sentence

$$\neg \exists x_1 \ldots \exists x_n (E(s, x_1) \wedge E(x_1, x_2) \wedge \cdots \wedge E(x_n, e))$$

Compactness

Proof

Now let T be the theory

$$T = \{\phi_n : n \geq 1\} \cup \{\psi\} \cup \{\neg(s = e) \wedge E(s, e)\}$$

Compactness

Proof

Now let T be the theory

$$T = \{\phi_n : n \geq 1\} \cup \{\psi\} \cup \{\neg(s = e) \wedge E(s, e)\}$$

Every finite subset of T obviously has a model. So by the compactness theorem T has a model, which is a contradiction.

□

Compactness

However in descriptive complexity we are more interested in finite models, and compactness fails over finite models!

Proof

Let σ be a vocabulary with no relation symbols and define

$$\lambda_n \equiv \exists x_1 \ldots \exists x_n \bigwedge_{i \neq j} (x_i \neq x_j)$$

and let T be the theory

$$T = \{\lambda_n : n \geq 0\}$$

Compactness

However in descriptive complexity we are more interested in finite models, and compactness fails over finite models!

Proof

Let σ be a vocabulary with no relation symbols and define

$$\lambda_n \equiv \exists x_1 \ldots \exists x_n \bigwedge_{i \neq j} (x_i \neq x_j)$$

and let T be the theory

$$T = \{\lambda_n : n \geq 0\}$$

Every finite subset of T has a model, but T does not have a finite model! □

Ehrenfeucht-Fraïssé Games

- We need a tool better tailored for finite models.

Ehrenfeucht-Fraïssé Games

- We need a tool better tailored for finite models.
- Answer: 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 σ.

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 σ.
- 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 structure. 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 structure. 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 structure. 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

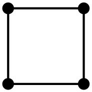
- If D has a winning strategy to win the k-move Ehrenfeucht-Fraïssé Game on **A** and **B**, we write $\mathbf{A} \equiv_k \mathbf{B}$.

Examples

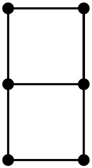
Let **A B** be sets with $|A|, |B| \geq k$ elements. D has a winning strategy for this game.

Examples

A

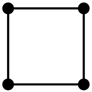


B

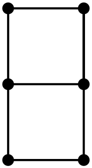


Examples

A

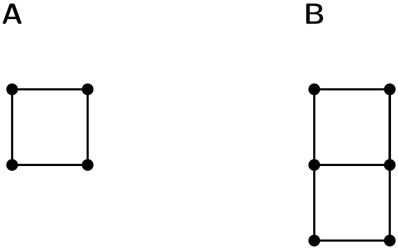


B



- 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?

Quantifier Rank

Definition 3

The Quantifier Rank of a formula $qr(\phi)$ is its depth of quantifier nesting.
We use the notation *FO* [k] for al *FO* formulae of quantifier rank up to k.

Examples

- The sentences from the previous example both had $qr = 3$.
- $(\exists x E(x, x)) \vee (\exists y \forall z \neg E(y, z))$ has $qr = 2$.

Quantifier Rank

Definition 4

Let $k \in \mathbb{N}$ and $\mathbf{A}$,$\mathbf{B}$ σ-structures. We say that $\mathbf{A} \sim_k \mathbf{B}$ agree on $FO[k]$ iff $\mathbf{A}$,$\mathbf{B}$ satisfy the same sentences of $FO[k]$.

The Ehrenfeucht-Fraïssé Theorem

Theorem 5

The following are equivalent:

1. $\boldsymbol{A}$ *and* $\boldsymbol{B}$ *agree on* $FO[k]$
2. $\boldsymbol{A} \equiv_k \boldsymbol{B}$

The Ehrenfeucht-Fraïssé Theorem

Theorem 5

The following are equivalent:

1. $\boldsymbol{A}$ *and* $\boldsymbol{B}$ *agree on* $FO[k]$

2. $\boldsymbol{A} \equiv_k \boldsymbol{B}$

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

Method

Corollary

A query Q is not definable in FO if for every $k \in \mathbb{N}$, there exists two finite σ-structures $\mathbf{A}_k$,$\mathbf{B}_k$ such that:

- $\mathbf{A}_k \equiv_k \mathbf{B}_k$
- $Q(\mathbf{A}) \neq Q(\mathbf{B})$

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. □

Examples

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

Examples

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

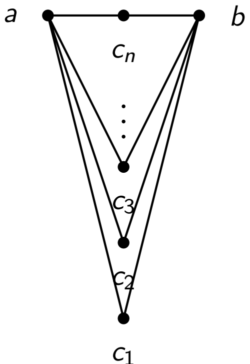
Reminder: A graph is EULERIAN if there is a cycle that traverses each edge only once. Equivalently(for a connected graph) each node has an even degree.

Examples

The EULERIAN query is not *FO* definable on the class of all finite graphs.
Reminder: A graph is EULERIAN if there is a cycle that traverses each edge only once. Equivalently(for a connected graph) each node has an even degree.

Proof.

The duplicator wins on $\mathbf{A}_{2k}$, $\mathbf{A}_{2k+1}$.



Examples

- $L_6 : 1 \leq 2 \leq \ 3 \leq 4 \leq 5 \leq 6$
- $L_7 : 1 \leq 2 \leq \ 3 \leq 4 \leq 5 \leq 6 \leq 7$
- $L_8 : 1 \leq 2 \leq \ 3 \leq 4 \leq 5 \leq 6 \leq 7 \leq 8$

Examples

- $L_6 : 1 \leq 2 \leq \; 3 \leq 4 \leq 5 \leq 6$
- $L_7 : 1 \leq 2 \leq \; 3 \leq 4 \leq 5 \leq 6 \leq 7$
- $L_8 : 1 \leq 2 \leq \; 3 \leq 4 \leq 5 \leq 6 \leq 7 \leq 8$

S wins the 3-move game on L_6 and L_7
D wins the 3-move game on L_7 and L_8

Examples

- $L_6 : 1 \leq 2 \leq \; 3 \leq 4 \leq 5 \leq 6$
- $L_7 : 1 \leq 2 \leq \; 3 \leq 4 \leq 5 \leq 6 \leq 7$
- $L_8 : 1 \leq 2 \leq \; 3 \leq 4 \leq 5 \leq 6 \leq 7 \leq 8$

S wins the 3-move game on L_6 and L_7
D wins the 3-move game on L_7 and L_8
If the linear order are sufficiently larger than k, then the duplicator wins the k-round game.

Examples

- $L_6 : 1 \leq 2 \leq\ 3 \leq 4 \leq 5 \leq 6$
- $L_7 : 1 \leq 2 \leq\ 3 \leq 4 \leq 5 \leq 6 \leq 7$
- $L_8 : 1 \leq 2 \leq\ 3 \leq 4 \leq 5 \leq 6 \leq 7 \leq 8$

S wins the 3-move game on L_6 and L_7
D wins the 3-move game on L_7 and L_8
If the linear order are sufficiently larger than k, then the duplicator wins the k-round game.
In fact it can be proved that if $m, n \geq 2^k - 1$ then $L_m \equiv_k L_n$

Examples

The EVEN CARDINALITY query is not *FO* definable on the class of all linear orders.

Examples

The EVEN CARDINALITY query is not *FO* definable on the class of all linear orders.

Proof.

k-round game on L_{2m} and L_{2m+1} where $m \geq 2^k - 1$ □

Examples

The CONNECTIVITY query is not *FO* definable on the class of finite graphs.

Examples

The CONNECTIVITY query is not *FO* definable on the class of finite graphs.

Proof.

We can use an *FO* reduction from EVEN CARDINALITY on linear orders! □

Finitness

Corollary

If σ is finite, then up to logical equivalence, $FO[k]$ over σ contains only finitely many formulae with m free variables.

Finitness

Corollary

If σ is finite, then up to logical equivalence, $FO[k]$ over σ contains only finitely many formulae with m free variables.
Proof(idea): Induction on k. An F[k+1] formula $\phi(x_1, \ldots, x_m)$ can be seen as a boolean combination of $\exists x_{m+1} \psi(x_1, \ldots, x_{m+1})$ where $\psi \in FO[k]$.

Types

Definition 6

The rank-k-m type of $x = (x_1, \ldots, x_m)$ over **A** is defined as

$$tp_k(\mathbf{A}, x) = \{\phi \in FO[k] | \mathbf{A} \models \phi(x)\}$$

Remark

Since *FO* [k] is finite, the rank-k type is determined by a unique formula for each structure. In fact the defining formula is also of $qr = k$.
Furthermore every *FO* [k] formula can be written as a_i, where a_i is a disjunction of rank-k type determining formulae.

Completeness

Corollary

The equivalence relation $\equiv_k$ is of finite index.

Completeness

Corollary

The equivalence relation $\equiv_k$ is of finite index.

Proof: From the Ehrenfeucht-Fraïssé theorem and the above remark.

Completeness

Corollary

The equivalence relation $\equiv_k$ is of finite index.
Proof: From the Ehrenfeucht-Fraïssé theorem and the above remark.

Corollary

A query Q is not expressible in *FO* iff for every k, one can find two structures $\mathbf{A}_k \equiv_k \mathbf{B}_k$ such that $Q(\mathbf{A}) \neq Q(\mathbf{B})$.

Completeness

Corollary

The equivalence relation $\equiv_k$ is of finite index.
Proof: From the Ehrenfeucht-Fraïssé theorem and the above remark.

Corollary

A query Q is not expressible in *FO* iff for every k, one can find two structures $\mathbf{A}_k \equiv_k \mathbf{B}_k$ such that $Q(\mathbf{A}) \neq Q(\mathbf{B})$. Proof(hint): Show that its definable by a disjunction of some of the a_i's.

Proof of the Ehrenfeucht-Fraïssé Theorem

Before we finally prove the Ehrenfeucht-Fraïssé Theorem we will define back-and-forth equivalence.

- $\mathbf{A} \sim_0 \mathbf{B}$ iff $\mathbf{A} \equiv_0 \mathbf{B}$
- $\mathbf{A} \sim_{k+1} \mathbf{B}$ iff the following conditions hold:
  forth: for every $a \in \mathbf{A}$, there exists $b \in \mathbf{B}$ such that $(\mathbf{A}, a) \sim_k (\mathbf{B}, b)$
  back: for every $b \in \mathbf{B}$, there exists $a \in \mathbf{A}$ such that $(\mathbf{A}, a) \sim_k (\mathbf{B}, b)$

Proof of the Ehrenfeucht-Fraïssé Theorem

Theorem 7

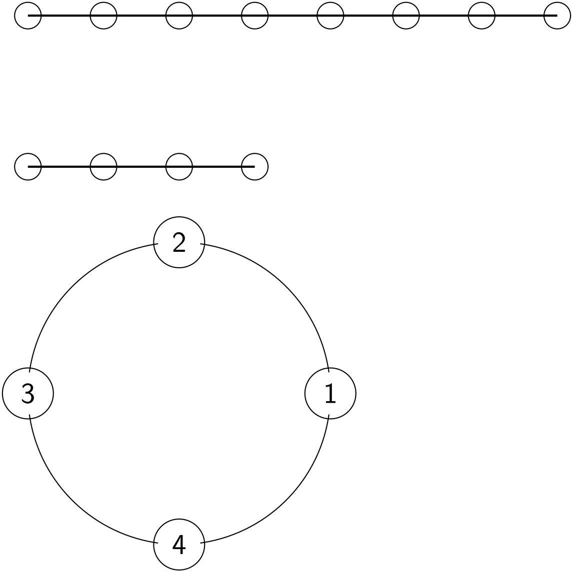
The following are equivalent:

1. ***A*** *and* ***B*** *agree on FO [k]*
2. $\boldsymbol{A} \equiv_k \boldsymbol{B}$
3. $\boldsymbol{A} \sim_k \boldsymbol{B}$

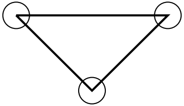
Proof Sketch

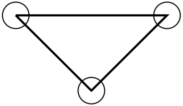
2 $\iff$ 3 By Induction on k.
1 $\iff$ 3 Use the fact that F[k+1] formula $\phi(x_1, \ldots, x_m)$ can be seen as a boolean combination of $\exists x_{m+1} \psi(x_1, \ldots, x_{m+1})$ where $\psi \in FO[k]$.

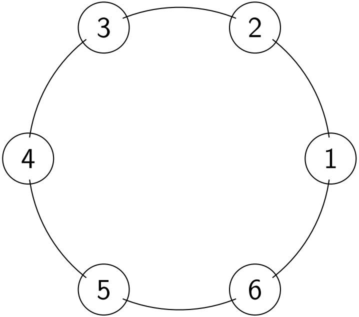
ACYCLICITY



2-COLORABILITY







Extensions

- Locality and Winning Games
- Extensions of Ehrenfeucht-Fraïssé games for other Logics

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.