Descriptive Complexity: Finite Variable Logics

Christina Spiliopoulou

ALMA
INTER-INSTITUTIONAL GRADUATE PROGRAM "ALGORITHMS, LOGIC AND DISCRETE MATHEMATICS"

Overview

Overview

Preliminaries (1/2)

Definition ($\mathcal{L}_{\infty\omega}$)

The logic $\mathcal{L}_{\infty\omega}$ is defined as an extension of *FO* with infinitary connectives $\bigvee$ and $\bigwedge$:
if φ_i's are formulae, for $i \in I$, where I is not necessarily finite, and the free variables of all the φ_i's are among $\vec{x}$, then

$$\bigvee_{i \in I} \varphi_i \text{ and } \bigwedge_{i \in I} \varphi_i$$

are formulae.
Their free variables are those variables in $\vec{x}$ that occur freely in one of the φ_i's.
The semantics is as expected.

Preliminaries (2/2)

Proposition

Let $\mathcal{C}$ be a class of finite structures closed under isomorphism. Then there is an $\mathcal{L}_{\infty\omega}$ sentence $\Phi_{\mathcal{C}}$ such that $\mathfrak{A} \in \mathcal{C}$ iff $\mathfrak{A} \models \Phi_{\mathcal{C}}$.

Preliminaries (2/2)

Proposition

Let $\mathcal{C}$ be a class of finite structures closed under isomorphism. Then there is an $\mathcal{L}_{\infty\omega}$ sentence $\Phi_{\mathcal{C}}$ such that $\mathfrak{A} \in \mathcal{C}$ iff $\mathfrak{A} \models \Phi_{\mathcal{C}}$.

Proof.

We know that for every finite structure $\mathfrak{B}$ there is an *FO* sentence $\Phi_{\mathfrak{B}}$ such that $\mathfrak{A} \models \Phi_{\mathfrak{B}}$ iff $\mathfrak{A} \cong \mathfrak{B}$.Hence we take $\Phi_{\mathcal{C}}$ to be $\bigvee_{\mathfrak{B}\in\mathcal{C}} \Phi_{\mathfrak{B}}$.

□

Motivation

So, since it defines every property of finite structures, $\mathcal{L}_{\infty\omega}$ is too powerful to be of interest in finite model theory...

Motivation

So, since it defines every property of finite structures, $\mathcal{L}_{\infty\omega}$ is too powerful to be of interest in finite model theory...

(Keep in mind that, from the construction of the above sentence ($\Phi_{\mathcal{C}}$), to define arbitrary classes of finite structures in $\mathcal{L}_{\infty\omega}$, one needs, in general, infinitely many variables.)

From infinite to finite (1/4)

Let $\varphi_n(x, y)$, $n \geq 1$, be *FO* formulae stating that there is a path from x to y of length n.

From infinite to finite (1/4)

Let $\varphi_n(x, y)$, $n \geq 1$, be *FO* formulae stating that there is a path from x to y of length n.

Then, we could express the transitive closure query in $\mathcal{L}_{\infty\omega}$ by

$$\bigvee_{n \geq 1} \varphi_n(x, y).$$

From infinite to finite (2/4)

Definition of φ_n's (1st idea)

$\varphi_n(x, y) \equiv \exists x_1 ... \exists x_{n-1}(E(x, x_1) \wedge ... \wedge E(x_{n-1}, y))$, $n > 1$
$\varphi_1(x, y) \equiv E(x, y)$

From infinite to finite (2/4)

Definition of φ_n's (1st idea)

$$\varphi_n(x,y) \equiv \exists x_1 ... \exists x_{n-1}(E(x,x_1) \wedge ... \wedge E(x_{n-1},y)) \ , \ n > 1$$
$$\varphi_1(x,y) \equiv E(x,y)$$

Definition of φ_n's (2nd idea: Inductively)

$$\varphi_1(x,y) \equiv E(x,y)$$
$$\varphi_{n+1}(x,y) \equiv \exists z_n(E(x,z_n) \wedge \varphi_n(z_n,y))$$

From infinite to finite (2/4)

Definition of φ_n's (1st idea)

$\varphi_n(x, y) \equiv \exists x_1 ... \exists x_{n-1}(E(x, x_1) \wedge ... \wedge E(x_{n-1}, y))$, $n > 1$
$\varphi_1(x, y) \equiv E(x, y)$

Definition of φ_n's (2nd idea: Inductively)

$\varphi_1(x, y) \equiv E(x, y)$
$\varphi_{n+1}(x, y) \equiv \exists z_n(E(x, z_n) \wedge \varphi_n(z_n, y))$

Either definition together with $\bigvee_{n\geq 1} \varphi_n(x, y)$ use infinitely many variables and we saw that the logic $\mathcal{L}_{\infty\omega}$ is useless in the context of finite model theory.

From infinite to finite (3/4)

Definition of φ_n's (3nd idea: Three variables are enough!)

$\varphi_1(x, y) \equiv E(x, y)$

...

$\varphi_{n+1}(x, y) \equiv \exists z(E(x, z) \wedge \exists x(z = x \wedge \varphi_n(x, y)))$

From infinite to finite (3/4)

Definition of φ_n's (3nd idea: Three variables are enough!)

$\varphi_1(x, y) \equiv E(x, y)$

...

$\varphi_{n+1}(x, y) \equiv \exists z(E(x, z) \land \exists x(z = x \land \varphi_n(x, y)))$

Now, each formula φ_n uses only three variables!

From infinite to finite (3/4)

Definition of φ_n's (3nd idea: Three variables are enough!)

$\varphi_1(x, y) \equiv E(x, y)$

...

$\varphi_{n+1}(x, y) \equiv \exists z(E(x, z) \wedge \exists x(z = x \wedge \varphi_n(x, y)))$

Now, each formula φ_n uses only three variables!

To define $\varphi_{n+1}(x, y)$, we need to say that there is a z such that $E(x, z)$ holds and $\varphi_n(z, y)$ holds. But with three variables we only know how to say that $\varphi_n(x, y)$ holds.

From infinite to finite (3/4)

Definition of φ_n's (3nd idea: Three variables are enough!)

$\varphi_1(x, y) \equiv E(x, y)$

...

$\varphi_{n+1}(x, y) \equiv \exists z(E(x, z) \wedge \exists x(z = x \wedge \varphi_n(x, y)))$

Now, each formula φ_n uses only three variables!

To define $\varphi_{n+1}(x, y)$, we need to say that there is a z such that $E(x, z)$ holds and $\varphi_n(z, y)$ holds. But with three variables we only know how to say that $\varphi_n(x, y)$ holds.

Problem solved with careful reuse of x, y, z!

From infinite to finite (4/4)

- With the above formulae, the transitive closure can still be defined by

$$\bigvee_{n \geq 1} \varphi_n(x, y).$$

From infinite to finite (4/4)

- With the above formulae, the transitive closure can still be defined by

$$\bigvee_{n \geq 1} \varphi_n(x, y).$$

BUT: Now, the resulting formula only uses three variables!

From infinite to finite (4/4)

- With the above formulae, the transitive closure can still be defined by

$$\bigvee_{n\geq 1} \varphi_n(x, y).$$

  BUT: Now, the resulting formula only uses three variables!
- Recall that in the proof of the proposition we saw, we needed (in general) infinitely many variables. We will see that an infinitary logic in which the number of variables is finite is useful in finite model theory.

Definition of finite variable logics

Definition (Finite variable logics)

The class of *FO* formulae that use at most k distinct variables will be denoted by FO^k. The class of $\mathcal{L}_{\infty\omega}$ formulae that use at most k variables will be denoted by $\mathcal{L}^k_{\infty\omega}$. We define the finite variable infinitary logic by

$$\mathcal{L}^{\omega}_{\infty\omega} = \bigcup_{k\in\mathbb{N}} \mathcal{L}^k_{\infty\omega}.$$

That is, $\mathcal{L}^{\omega}_{\infty\omega}$ has formulae of $\mathcal{L}_{\infty\omega}$ that only use finitely many variables.

Quantifier rank

Definition (Quantifier rank of $\mathcal{L}^{\omega}_{\infty\omega}$ formulae)

The quantifier rank $qr(\cdot)$ of $\mathcal{L}^{\omega}_{\infty\omega}$ formulae is defined as for *FO* for Boolean connectives and quantifiers; for infinitary connectives, we define

$$qr(\bigvee_i \varphi_i) = qr(\bigwedge_i \varphi_i) = \sup_i qr(\varphi_i).$$

Overview

Cardinalities (1/2)

We consider linear orderings (the vocabulary contains only binary relation $<$). We define the formulae:

$$\psi_1(x) \equiv (x = x)$$

...

$$\psi_{n+1}(x) \equiv \exists y((x > y) \land \exists x(y = x \land \psi_n(x)))$$

Cardinalities (1/2)

We consider linear orderings (the vocabulary contains only binary relation $<$). We define the formulae:

$\psi_1(x) \equiv (x = x)$

...

$\psi_{n+1}(x) \equiv \exists y((x > y) \wedge \exists x(y = x \wedge \psi_n(x)))$

- The above formulae are in $\mathcal{L}^{2}_{\infty\omega}$.

Cardinalities (1/2)

We consider linear orderings (the vocabulary contains only binary relation $<$). We define the formulae:

$\psi_1(x) \equiv (x = x)$

...

$\psi_{n+1}(x) \equiv \exists y((x > y) \wedge \exists x(y = x \wedge \psi_n(x)))$

- The above formulae are in $\mathcal{L}^{2}_{\infty\omega}$.

When is the formula $\psi_n(a)$ true in a linear order L?

Cardinalities (2/2)

Thus, for each n we have a sentence $\Psi_n \equiv \exists x \psi_n(x)$ that is true in L iff $|L| \geq n$.

Cardinalities (2/2)

Thus, for each n we have a sentence $\Psi_n \equiv \exists x \psi_n(x)$ that is true in L iff $|L| \geq n$.

Arbitrary cardinalities of linear orerings can be tested in $\mathcal{L}^2_{\infty\omega}$

For an arbitrary subset C of $\mathbb{N}$, the sentence

$$\bigvee_{n \in C} (\Psi_n \wedge \neg\Psi_{n+1})$$

is true in L iff $|L| \in C$.
(Notice that the above is an $\mathcal{L}^2_{\infty\omega}$ sentence.)

Queries over ordered finite σ-strustures

Proposition

Every query over ordered finite σ-structures is expressible in $\mathcal{L}^{\omega}_{\infty\omega}$.

Queries over ordered finite σ-strustures

Proposition

Every query over ordered finite σ-structures is expressible in $\mathcal{L}^{\omega}_{\infty\omega}$.

Proof's keys:

Queries over ordered finite σ-strustures

Proposition

Every query over ordered finite σ-structures is expressible in $\mathcal{L}^{\omega}_{\infty\omega}$.

Proof's keys:

- We use the $\mathcal{L}^{2}_{\infty\omega}$ formulae ψ_i's that we defined in the previous example.

Queries over ordered finite σ-strustures

Proposition

Every query over ordered finite σ-structures is expressible in $\mathcal{L}^{\omega}_{\infty\omega}$.

Proof's keys:

- We use the $\mathcal{L}^{2}_{\infty\omega}$ formulae ψ_i's that we defined in the previous example.
- Since ψ_i's are in $\mathcal{L}^{2}_{\infty\omega}$, we know that for each n we have an $\mathcal{L}^{2}_{\infty\omega}$ formula $\psi_{=n}(x)$ which holds iff x is the nth element in the ordering.

Queries over ordered finite σ-strustures

Proposition

Every query over ordered finite σ-structures is expressible in $\mathcal{L}^{\omega}_{\infty\omega}$.

Proof's keys:

- We use the $\mathcal{L}^{2}_{\infty\omega}$ formulae ψ_i's that we defined in the previous example.
- Since ψ_i's are in $\mathcal{L}^{2}_{\infty\omega}$, we know that for each n we have an $\mathcal{L}^{2}_{\infty\omega}$ formula $\psi_{=n}(x)$ which holds iff x is the nth element in the ordering.
- For simplicity, we consider ordered graphs. The basic idea is that for each graph we define an $\mathcal{L}^{3}_{\infty\omega}$ formula that characterizes it. By infinitary disjunctions of these formulae we are able to characterize any class of ordered graphs.

Overview

Introduction

- We 've seen that the formula φ_{tc} of transitive closure induces an operator $F_{\varphi_{tc}}$. This can be generalized:

Introduction

- We 've seen that the formula φ_{tc} of transitive closure induces an operator $F_{\varphi_{tc}}$. This can be generalized:

  Every *FO* formula $\varphi(R, \vec{x})$ gives rise to an operator F_{φ}, in the same way that we 've seen in the transitive closure example.

Introduction

- We 've seen that the formula φ_{tc} of transitive closure induces an operator $F_{\varphi_{tc}}$. This can be generalized:

  Every *FO* formula $\varphi(R, \vec{x})$ gives rise to an operator F_{φ}, in the same way that we 've seen in the transitive closure example.

  (The idea is that the operator applied to a set X gives us the elements that satisfie the formula when R is interpreted as X.)

Introduction

- We 've seen that the formula φ_{tc} of transitive closure induces an operator $F_{\varphi_{tc}}$. This can be generalized:

  Every *FO* formula $\varphi(R, \vec{x})$ gives rise to an operator F_{φ}, in the same way that we 've seen in the transitive closure example.

  (The idea is that the operator applied to a set X gives us the elements that satisfie the formula when R is interpreted as X.)
- We computed the least fixed point of the monotone operator $F_{\varphi_{tc}}$ in stages, where we computed $F^{r}_{\varphi_{tc}}(\emptyset)$ for $r = 1, 2, \ldots$.

Introduction

- We 've seen that the formula φ_{tc} of transitive closure induces an operator $F_{\varphi_{tc}}$. This can be generalized:
  Every FO formula $\varphi(R, \vec{x})$ gives rise to an operator F_{φ}, in the same way that we 've seen in the transitive closure example.
  (The idea is that the operator applied to a set X gives us the elements that satisfie the formula when R is interpreted as X.)
- We computed the least fixed point of the monotone operator $F_{\varphi_{tc}}$ in stages, where we computed $F^{r}_{\varphi_{tc}}(\emptyset)$ for $r = 1, 2, \ldots$.
  In general, for such an operator, we can define $F^{0}_{\varphi}(\emptyset) \equiv \emptyset$
  and formulae φ^{i}'s
  so that $\forall n$, $F^{n}_{\varphi}(\emptyset)$ gives us the elements that satisfie φ^{n}.
- The new thing is that we will define these formulae with finitely many variables!

Defining φ^i's

Suppose that an *FO* formula $\varphi(R, \vec{x})$ defines a monotone operator.

Defining φ^i's

Suppose that an *FO* formula $\varphi(R, \vec{x})$ defines a monotone operator. Assume that φ in addition to $\vec{x} = (x_1, ..., x_k)$, uses variables $z_1, ..., z_l$.

Defining φ^i's

Suppose that an *FO* formula $\varphi(R, \vec{x})$ defines a monotone operator.

Assume that φ in addition to $\vec{x} = (x_1, ..., x_k)$, uses variables $z_1, ..., z_l$.

We introduce additional variables $\vec{y} = (y_1, ..., y_k)$ and define $\varphi^0(\vec{x}) \equiv \neg(x_1 = x_1)$, i.e. *false*, and then inductively $\varphi^{n+1}(\vec{x})$ as $\varphi(R, \vec{x})$ in which every occurence of $R(u_1, ..., u_k)$, where $u_1, ..., u_k$ are variables among $\vec{x}$ and $\vec{z}$, is replaced by

$$\exists \vec{y}((\vec{y} = \vec{u}) \wedge (\exists \vec{x}((\vec{x} = \vec{y}) \wedge \varphi^n(\vec{x})))).$$

Note: $\vec{x} = \vec{y}$ is an abbreviation for $(x_1 = y_1) \wedge ... \wedge (x_k = y_k)$.
Important: We at most doubled the variables of the *FO* formula φ!

Example (1/2)

Let's see the above in the transitive closure example:

- $\varphi_{tc}(R, x_1, x_2) = E(x_1, x_2) \vee \exists z_1(E(x_1, z_1) \wedge R(z_1, x_2))$,
  so
  $\vec{x} = (x_1, x_2)$
  $\vec{z} = z_1$
  $\vec{y} = (y_1, y_2)$ and
  $\vec{u} = (u_1, u_2) = (z_1, x_2)$.
- $\varphi^0_{tc}(x_1, x_2) \equiv \neg(x_1 = x_1) \equiv$ *false*
- $\varphi^1_{tc}(x_1, x_2) \equiv E(x_1, x_2) \vee \exists z_1(E(x_1, z_1) \wedge \exists \vec{y}((y_1 = z_1) \wedge (y_2 = x_2) \wedge (\exists x((\vec{x} = \vec{y}) \wedge \varphi^0_{tc}(x_1, x_2)))))$
  which gives us
  $\varphi^1_{tc}(x_1, x_2) \equiv E(x_1, x_2)$.

Example (2/2)

One can test that:

- $\varphi^2_{tc}(x_1, x_2) \equiv E(x_1, x_2) \vee \exists z_1(E(x_1, z_1) \wedge E(z_1, x_2))$
- etc

The trick is the same: We carefully reused variables and achieved to only use (x_1, x_2) as input in the definition of the φ^i_{tc}'s.

Conclusion

We achieved defining φ^i's so that for any structure $\mathfrak{A}$:

- $F^i_\varphi(\emptyset) = \{\vec{x} \mid \mathfrak{A} \models \varphi^i(\vec{x})\}$
- We at most doubled the variables of φ in order to define every φ^i.

From a theorem that we've seen in Inductive Definitions, if F_φ is a monotone operator, then for any (finite) structure $\mathfrak{A}$ the least fixed point exists and it is equal to $F^r_\varphi(\emptyset)$ for some $r \in \mathbb{N}$. Therefore:

- For some $r \in \mathbb{N}$, $\varphi^r(\vec{x})$ tests the least fixed point of the operator F_φ and for this r, it holds that

$$\varphi^r(\vec{x}) = \bigvee_n \varphi^n(\vec{x}).$$

Finally, we are there!

Theorem

$\mathrm{LFP} \subseteq \mathcal{L}^{\omega}_{\infty\omega}$

Proof.

We proved that if φ is an *FO* sentence that uses m variables, then $\mathbf{lfp}_{R,\vec{x}}\varphi$ is expressible in $\mathcal{L}^{2m}_{\infty\omega}$.
If we have a complex fixed point formula (e.g., involving nested fixed points), we can then apply the construction inductively, using the same substitution, since φ^n need not be an *FO* formula, and we can have infinitary connectives. Again, we at most double the number of variables, which completes the proof of the theorem. □

Overview

Introduction

Ehrenfeucht-Fraïssé-style games:

Introduction

Ehrenfeucht-Fraïssé-style games:

- $\mathfrak{A}$,$\mathfrak{B}$$\in$ STRUCT[σ] (i.e. finite)

Introduction

Ehrenfeucht-Fraïssé-style games:

- $\mathfrak{A}$,$\mathfrak{B} \in$ STRUCT[σ] (i.e. finite)
- fixed set of pairs of pebbles: $\{(p^1_{\mathfrak{A}}, p^1_{\mathfrak{B}}), ..., (p^k_{\mathfrak{A}}, p^k_{\mathfrak{B}})\}$

Introduction

Ehrenfeucht-Fraïssé-style games:

- $\mathfrak{A}, \mathfrak{B} \in$ STRUCT$[\sigma]$ (i.e. finite)
- fixed set of pairs of pebbles: $\{(p^1_{\mathfrak{A}}, p^1_{\mathfrak{B}}), \ldots, (p^k_{\mathfrak{A}}, p^k_{\mathfrak{B}})\}$
- the number of rounds is not necessarily finite (but we can determine who wins)

Description of one round

- Spoiler chooses structure (w.l.o.g. $\mathfrak{A}$) and pebble-pair, assume i.

Description of one round

- Spoiler chooses structure (w.l.o.g. $\mathfrak{A}$) and pebble-pair, assume i.
- Spoiler places $p^i_{\mathfrak{A}}$ on an element of $\mathfrak{A}$. (If it is already placed on an element of $\mathfrak{A}$, he decides either to leave it there or to place it on another element of $\mathfrak{A}$.)

Description of one round

- Spoiler chooses structure (w.l.o.g. $\mathfrak{A}$) and pebble-pair, assume i.
- Spoiler places $p^i_{\mathfrak{A}}$ on an element of $\mathfrak{A}$. (If it is already placed on an element of $\mathfrak{A}$, he decides either to leave it there or to place it on another element of $\mathfrak{A}$.)
- Dublicator responds by placing $p^i_{\mathfrak{B}}$ on an element of $\mathfrak{B}$.

Description of one round

- Spoiler chooses structure (w.l.o.g. $\mathfrak{A}$) and pebble-pair, assume i.
- Spoiler places $p^i_{\mathfrak{A}}$ on an element of $\mathfrak{A}$. (If it is already placed on an element of $\mathfrak{A}$, he decides either to leave it there or to place it on another element of $\mathfrak{A}$.)
- Dublicator responds by placing $p^i_{\mathfrak{B}}$ on an element of $\mathfrak{B}$.
- After each round, $F \subseteq A \times B$ contains exactly the pebble-pairs that have been placed until that moment.

Winning strategy

Notation:

Winning strategy

Notation:

- $PG_k^n(\mathfrak{A}, \mathfrak{B})$ is the k-pebble game that continues for n rounds.

Winning strategy

Notation:

- $PG_k^n(\mathfrak{A}, \mathfrak{B})$ is the k-pebble game that continues for n rounds.
- $PG_k^\infty(\mathfrak{A}, \mathfrak{B})$ is the k-pebble game that continues forever.

Winning strategy

Notation:

- $PG_k^n(\mathfrak{A}, \mathfrak{B})$ is the k-pebble game that continues for n rounds.
- $PG_k^\infty(\mathfrak{A}, \mathfrak{B})$ is the k-pebble game that continues forever.

Winning strategy

- Dublicator has a winning strategy in $PG_k^n(\mathfrak{A}, \mathfrak{B})$ iff he can ensure that after each round $j \leq n$, F is a graph of a partial isomorphism. In this case, we write

$$\mathfrak{A} \equiv_{k,n}^{\infty\omega} \mathfrak{B}.$$

Winning strategy

Notation:

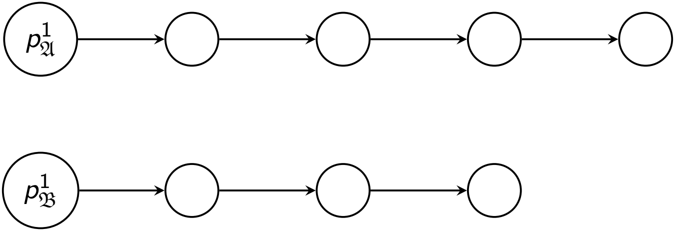
- $PG_k^n(\mathfrak{A}, \mathfrak{B})$ is the k-pebble game that continues for n rounds.
- $PG_k^\infty(\mathfrak{A}, \mathfrak{B})$ is the k-pebble game that continues forever.

Winning strategy

- Dublicator has a winning strategy in $PG_k^n(\mathfrak{A}, \mathfrak{B})$ iff he can ensure that after each round $j \leq n$, F is a graph of a partial isomorphism. In this case, we write
$$\mathfrak{A} \equiv_{k,n}^{\infty\omega} \mathfrak{B}.$$
- Dublicator has a winning strategy in $PG_k^\infty(\mathfrak{A}, \mathfrak{B})$ iff he can ensure that after every round F is a graph of a partial isomorphism. In this case we write
$$\mathfrak{A} \equiv_{k}^{\infty\omega} \mathfrak{B}.$$

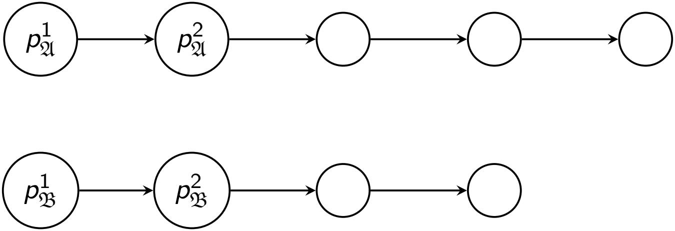
2-pebble game example (1/5)

1st round (spoiler chooses $\mathfrak{A}$ and 1):



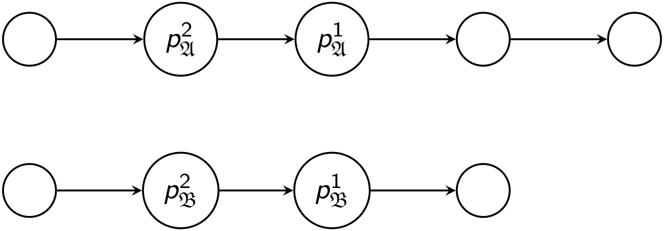
2-pebble game example (2/5)

2nd round (spoiler chooses $\mathfrak{A}$ and 2):



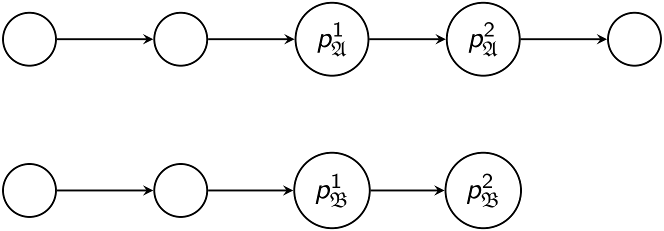
2-pebble game example (3/5)

3rd round (spoiler chooses $\mathfrak{A}$ and 1):



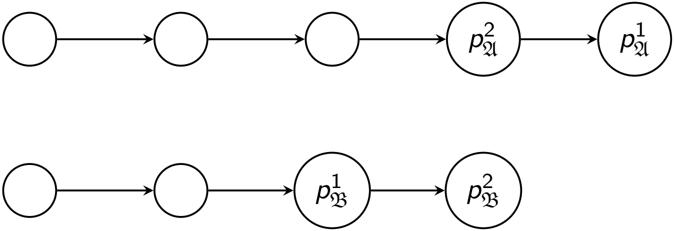
2-pebble game example (4/5)

4th round (spoiler chooses $\mathfrak{A}$ and 2):



2-pebble game example (5/5)

5th round (spoiler chooses $\mathfrak{A}$ and 1 and wins the game!):



Characterization

Theorem

1. *Two structures* $\mathfrak{A}, \mathfrak{B} \in \text{STRUCT}[\sigma]$ *agree on all sentences of* $\mathcal{L}^{k}_{\infty\omega}$ *of quantifier rank up to* n *iff*

$$\mathfrak{A} \equiv^{\infty\omega}_{k,n} \mathfrak{B}.$$

Characterization

Theorem

1. *Two structures* $\mathfrak{A}, \mathfrak{B} \in \text{STRUCT}[\sigma]$ *agree on all sentences of* $\mathcal{L}^k_{\infty\omega}$ *of quantifier rank up to n iff*

$$\mathfrak{A} \equiv^{\infty\omega}_{k,n} \mathfrak{B}.$$

2. *Two structures* $\mathfrak{A}, \mathfrak{B} \in \text{STRUCT}[\sigma]$ *agree on all sentences of* $\mathcal{L}^k_{\infty\omega}$ *iff*

$$\mathfrak{A} \equiv^{\infty\omega}_{k} \mathfrak{B}.$$

Characterization

Theorem

1. *Two structures* $\mathfrak{A}, \mathfrak{B} \in \mathrm{STRUCT}[\sigma]$ *agree on all sentences of* $\mathcal{L}^{k}_{\infty\omega}$ *of quantifier rank up to n iff*

$$\mathfrak{A} \equiv^{\infty\omega}_{k,n} \mathfrak{B}.$$

2. *Two structures* $\mathfrak{A}, \mathfrak{B} \in \mathrm{STRUCT}[\sigma]$ *agree on all sentences of* $\mathcal{L}^{k}_{\infty\omega}$ *iff*

$$\mathfrak{A} \equiv^{\infty\omega}_{k} \mathfrak{B}.$$

(The proof is very similar to the proof of the Ehrenfeucht-Fraïssé theorem.)

An application example

The query *EVEN* is not expressible in $\mathcal{L}^{\omega}_{\infty\omega}$.

An application example

The query *EVEN* is not expressible in $\mathcal{L}^{\omega}_{\infty\omega}$.

- Assume, to the contrary, that *EVEN* is expressed by a sentence $\Phi \in \mathcal{L}^{k}_{\infty\omega}$ and choose two structures $\mathfrak{A}$ and $\mathfrak{B}$ of cardinallities k and $k+1$, respectively, that are only sets. It's easy to see that $\mathfrak{A} \equiv^{\infty\omega}_{k} \mathfrak{B}$ and hence, from the previous theorem, we get $\mathfrak{A} \models \Phi$ iff $\mathfrak{B} \models \Phi$, which leads us to contradiction.

Not for now...

Theorem (Abiteboul-Vianu)

$\text{PTIME} = \text{PSPACE}$ *iff* $\text{LFP} = \text{PFP}$

The End!

> The more I think about language, the more it amazes me that people ever understand each other
>
> *Kurt Gödel*

The End!

> The more I think about language, the more it amazes me that people ever understand each other
>
> *Kurt Gödel*

Bibliography

- Libkin, Leonid. Elements of finite model theory. Springer Science & Business Media, 2013.