Ordering, Logic and *PTIME*

Romanos Aslanis

National and Kapodistrian University

2023 年 3 月 26 日

Introduction

On the one hand, we have seen that $FO(LFP)$ captures $PTIME$ over ordered structures.

- Why do we need an ordering?
- What if we "remove" the ordering assumption?

Introduction

On the other hand, by Fagin's Theorem, $SO(\exists)$ captures NP over all finite structures.

- What does this mean?
- Why an ordering assumption is not necessary?

Introduction

Why to insist on unordered structures?

- All the problems we deal with are iso-invariant.
- For practical reasons. (databases)
- The above mentioned theorems seem to imply a fundamental difference between P and NP.

Introduction

The question we are going to talk about is:

$$\textit{Is there a logic that captures } P\textit{, without an ordering?} \qquad (1)$$

Introduction

A "yes" answer to (1) would lead in a purely logical characterization of the well-known problem P vs NP.

Introduction

A "yes" answer to (1) would lead in a purely logical characterization of the well-known problem P vs NP.

A "no" answer to (1) would (directly) imply that $P \neq NP$.

Introduction

So far, all attempts to produce a logic that captures P (without an ordering) have failed.
In fact, it is conjectured (*Gurevich*) that no such logic exists.

Collections of structures

Boolean queries

A boolean query will be thought of as a subset of a class of structures that is closed under isomorphisms.

For example, let $\tau =< E^2 >$ (the vocabulary of graphs) and $Q_{dc} \subseteq STRUC[\tau]$ be the *disconectivity* boolean query, i.e. for all $\mathcal{A} \in STRUC[\tau]$,

$$\mathcal{A} \in Q_{dc} \texttt{ iff } \mathcal{A} \texttt{ is a disconnected graph.}$$

$\mathcal{L}$-definable boolean queries

$\mathcal{L}$-definable query

A boolean query $Q \subseteq STRUC[\tau]$ is $\mathcal{L}$-definable if there is a formula $\phi \in \mathcal{L}(\tau)$ such that for every $\mathcal{A} \in STRUC[\tau]$,

$$\mathcal{A} \in Q \texttt{ iff } \mathcal{A} \models \phi$$

Or, simpler, if $Q = Mod(\phi)$.

For example, the boolean query *disconnectivity* is $SO(\exists)$-definable, by the following formula:

$$(\exists S)\big((\exists x)S(x) \land (\exists y)\neg S(y) \land (\forall z)(\forall w)(S(z) \land \neg S(w) \rightarrow \neg E(z,w))\big)$$

Encoding structures

We have already mentioned that in order to consider a structure as an input for a TM, we have to represent it as a string and, in order to achieve that, an ordering on the universe must exist.

We can fix any "good" encoding function and understand ordered structures to be represented by their encodings.

For an unordered structure $\mathcal{A}$, we associate the set of all encodings $enc(\mathcal{A}, <^{\mathcal{A}})$ where $<^{\mathcal{A}}$ is a linear order on $|\mathcal{A}|$.

Encoding a class of structures

Let $\mathcal{Q} \subseteq STRUC[\tau_<]$ (boolean query over *ordered* τ-structures) .

$$enc(\mathcal{Q}) = \{enc(\mathcal{A}) : \mathcal{A} \in \mathcal{Q}\}$$

If $\mathcal{Q} \subseteq STRUC[\tau]$ (boolean query over *unordered* τ-structures), then we fix the class of ordered representations of structures in $\mathcal{Q}$ as:

$$enc(\mathcal{Q}) = enc(\mathcal{Q}_<)$$

Where
$\mathcal{Q}_< = \{(\mathcal{A}, <) : \mathcal{A} \in \mathcal{Q} \texttt{ and } < \texttt{ is a linear order on } |\mathcal{A}|\}$

Isomorphism Invariance

Encoding an unorder structure involves selecting an ordering on the universe.

Different orderings, may, produce different encodings.

However, we want to study properties of structures, not of their encodings and thus an algorithm that takes as input $enc(\mathcal{A}, <)$ should produce the same answer for all encodings of the same structure.

In other words the algorithm should be isomorphism-invariant, a property that, in general, is undecidable.

Order-invariant senteces

Order-invariance

A first-order sentence ϕ of vocabulary $\tau \cup \{<\}$ is order-invariant on a class $\mathcal{K}$ of τ-structures if, for any $\mathcal{A} \in \mathcal{K}$ and any pair $<_1, <_2$ of linear orderings:

$$(\mathcal{A}, <_1) \models \phi \Leftrightarrow (\mathcal{A}, <_2) \models \phi$$

Order-invariant senteces

Order-invariance

A first-order sentence ϕ of vocabulary $\tau \cup \{<\}$ is order-invariant on a class $\mathcal{K}$ of τ-structures if, for any $\mathcal{A} \in \mathcal{K}$ and any pair $<_1, <_2$ of linear orderings:

$$(\mathcal{A}, <_1) \models \phi \Leftrightarrow (\mathcal{A}, <_2) \models \phi$$

Hint for later: *It is undecidable whether a given first-order formula is order-invariant on finite structures.*

Logic captures complexity class

Let $\mathcal{L}$ be a logic, $\mathcal{C}$ a complexity class and $\mathcal{D}$ a domain of finite structures.

Definition

We say that $\mathcal{L}$ captures $\mathcal{C}$ on $\mathcal{D}$ if:
For every vocabulary τ and for every boolean query $\mathcal{Q} \subseteq \mathcal{D}(\tau)$,

$$enc(\mathcal{Q}) \in \mathcal{C} \texttt{ iff } \mathcal{Q} \texttt{ is } \mathcal{L}\texttt{-definable on } \mathcal{D}$$

Logic captures complexity class

Let $\mathcal{L}$ be a logic, $\mathcal{C}$ a complexity class and $\mathcal{D}$ a domain of finite structures.

Definition

We say that $\mathcal{L}$ captures $\mathcal{C}$ on $\mathcal{D}$ if:
For every vocabulary τ and for every boolean query $\mathcal{Q} \subseteq \mathcal{D}(\tau)$,

$$\mathit{enc}(\mathcal{Q}) \in \mathcal{C} \texttt{ iff } \mathcal{Q} \texttt{ is } \mathcal{L}\texttt{-definable on } \mathcal{D}$$

In other words, for every formula $\phi \in \mathcal{L}(\tau)$ the model-checking problem for ϕ is in $\mathcal{C}$, *and* for every query $Q \subseteq STRUC[\tau]$, whose membership problem is in $\mathcal{C}$, there exists a formula $\phi \in \mathcal{L}(\tau)$ s.t:

$$Q = \{\mathcal{A} \in STRUC[\tau] : \mathcal{A} \models \phi\}$$

First example

Let $\mathcal{L}_1$ be the logic whose τ-sentences are the order-invariant $FO(LFP)$-sentences of vocabulary $\tau \cup \{<\}$.
As for the semantics of that logic, a structure $\mathcal{U}$ is a model of ϕ iff $(\mathcal{U}, <) \models^{\mathit{LFP}} \phi$ for some, and hence all, linear orders, i.e.

$$\mathcal{U} \models^{\mathcal{L}_1} \phi \texttt{ iff } (\mathcal{U}, <) \models^{\mathit{LFP}} \phi$$

First example

Let $\mathcal{L}_1$ be the logic whose τ-sentences are the order-invariant $FO(LFP)$-sentences of vocabulary $\tau \cup \{<\}$.
As for the semantics of that logic, a structure $\mathcal{U}$ is a model of ϕ iff $(\mathcal{U}, <) \models^{LFP} \phi$ for some, and hence all, linear orders, i.e.

$$\mathcal{U} \models^{\mathcal{L}_1} \phi \texttt{ iff } (\mathcal{U}, <) \models^{LFP} \phi$$

Proposition

$\mathcal{L}_1$ captures $PTIME$ on the domain of all finite structures. But ...

First example

Let $\mathcal{L}_1$ be the logic whose τ-sentences are the order-invariant $FO(LFP)$-sentences of vocabulary $\tau \cup \{<\}$.
As for the semantics of that logic, a structure $\mathcal{U}$ is a model of ϕ iff $(\mathcal{U}, <) \models^{LFP} \phi$ for some, and hence all, linear orders, i.e.

$$\mathcal{U} \models^{\mathcal{L}_1} \phi \texttt{ iff } (\mathcal{U}, <) \models^{LFP} \phi$$

Proposition

$\mathcal{L}_1$ captures $PTIME$ on the domain of all finite structures. But ...
It is undecidable whether a given sentence $\phi \in FO(LFP)$ is order-invariant $\rightarrow$ this "logic" does not have an *effective* syntax.

Second example

Similarly, let $\mathcal{L}_2$ be the logic whose τ-sentences are the $FO(LFP)(\tau \cup \{<\})$ sentences. As for the semantics, let $\mathcal{U}$ be a τ-structure, then:

$$\mathcal{U} \models^{\mathcal{L}_2} \phi \texttt{ iff } \phi \texttt{ is order-invariant and}$$
$$\texttt{there is an ordering } <^{\mathcal{U}} \texttt{ on } |\mathcal{U}|$$
$$\texttt{such that } (\mathcal{U}, <^{\mathcal{U}}) \models^{\mathit{LFP}} \phi$$

$\mathcal{L}_2$ also captures $PTIME$ on the domain of all finite structures, but this time its semantics is not effective.

Exclude such "logics"

If we want to discuss the problem of whether $PTIME$ can be captured without the need of ordering, we need to make precise the notion of a logic. Furthermore, in order to avoid pathological examples (as the previous ones), we need to refine the notion of a logic capturing a complexity class.

Definition of logic

Definition

A logic on a domain $\mathcal{D}$ of finite structures is a pair $(\mathcal{L}, \models)$, where $\mathcal{L}$ is a function that assigns to each vocabulary τ a *recursive* set $\mathcal{L}(\tau)$ and $\models$ is a binary relation between $\mathcal{L}$-formulas and finite structures, so that for each formula $\phi \in \mathcal{L}(\tau)$, the class $\{\mathcal{U} \in \mathcal{D}(\tau) : \mathcal{U} \models \phi\}$ is closed under isomorphism.

Effectively capture

Definition

A logic $(\mathcal{L}, \models)$ **effectively captures** $\mathcal{C}$ on a domain $\mathcal{D}$ of finite structures if it captures $\mathcal{C}$ in the sence of the (already) known definition and, moreover, there exist a computable function which associates with every sentence $\phi \in \mathcal{L}(\tau)$ a TM M and a function f[1], such that M decides $\{\mathcal{U} \in \mathcal{D}(\tau) : \mathcal{U} \models \phi\}$ and f is witnessing that M is resource bounded according to $\mathcal{C}$.

[1]We code f by a number d meaning that M is *dlog*-space bounded, n^d time-bounded, etc

titlos

All the logics we have seen so far, satisfy the effectivity conditions.

(Last) Example

Let $\mathcal{L}$ be the logic where:

- $\mathcal{L}(\tau) = (M, d)$: M is a x^d time-bounded NTM for ordered represantations of τ-structures.
- For every $\phi = (M, d) \in \mathcal{L}(\tau)$ and every τ-structure $\mathcal{A}$, $\mathcal{A} \models \phi$ iff M accepts some ordered representation of $\mathcal{A}$ in $\leq ||\mathcal{A}||^d$ steps.

Exercise: *$\mathcal{L}$ effectively captures NP.*

enan titlo

We thus are led to reformulate question 1 as follows:

Is there a logic that effectively captures $PTIME$, without ordering?

Anuj Dawar's theorem

Theorem

If there is any logic that effectively captures $PTIME$, then there also exists a "natural" one.

Dual of a first-order query

A first-order query $I : STRUC[\sigma] \rightarrow STRUC[\tau]$ maps any structure $\mathcal{A} \in STRUC[\sigma]$ to a structure $I(\mathcal{A}) \in STRUC[\tau]$.
It does this by defining the relations of $I(\mathcal{A})$ via first-order formulas of $FO(\sigma)$.

Every such I, has a dual mapping $\hat{I}$, which translates any formula in $FO(\tau)$ to $FO(\sigma)$.

Dual map

Let $I : STRUC[\sigma] \to STRUC[\tau]$ be a k-ary first-order query. Then I defines a dual map $\hat{I} : FO(\tau) \to FO(\sigma)$ as follows:
For $\phi \in FO(\tau)$, define $\hat{I}(\phi)$ using a map f_I defined as follows:

- Map each variable u to a k-tuple of variables:

$$f_I(u) = u_1, \ldots, u_k$$

- Replace input relations by their corresponding formulas:

$$f_I(R_i(u_1, \ldots, u_{a_i})) = \phi_i(f_I(u_1), \ldots, f_I(u_{a_i}))$$

Dual map

- Replace constants by an existentially quantified special k-tuple:

$$\exists z_{i_1} \ldots \exists z_{i_k} \psi_i(z_{i_1}, \ldots, z_{i_k})$$

- Replace quantifiers by restricted quantifiers:

$$f_I(\exists u) = (\exists f_I(u).\phi_0(f_I(u)))$$

- Replace equality with equality over k-tuples [2]:

$$f_I(u_1 = u_2) = f_I(u_1) = f_I(u_2)$$

[2] To be precise, we have to complete query I by writing a formula defining the equality relation.

Thus ...

$\hat{I}$ is defined as follows, for $\phi \in FO(\tau)$:

$$\hat{I}(\phi) = (\exists z_{1_1} \dots z_{1_k}.\psi_1(z_{1_1}, \dots, z_{1_k})) \dots (\exists z_{s_1} \dots z_{s_k}.\psi_s(z_{s_1}), \dots, z_{s_k}))$$
$$f_I(\phi)$$

Proposition

For all sentences $\phi \in FO(\tau)$ and all structures $\mathcal{A} \in STRUC[\sigma]$,

$$\mathcal{A} \models \hat{I}(\phi) \texttt{ iff } I(\mathcal{A}) \models \phi$$

"Example"

We want to interpret the field of rationals $< \mathbb{Q}, 0, 1, +, * >$ inside of $< \mathbb{Z}, 0, 1, +, * >$. I is given as follows:

- $\phi_0(x_1, x_2) \equiv x_1 = x_1 \wedge x_2 \neq 0$

"Example"

We want to interpret the field of rationals $< \mathbb{Q}, 0, 1, +, * >$ inside of $< \mathbb{Z}, 0, 1, +, * >$. I is given as follows:

- $\phi_0(x_1, x_2) \equiv x_1 = x_1 \wedge x_2 \neq 0$
- $\epsilon(x_1, x_2, y_1, y_2) \equiv x_2 \neq 0 \wedge y_2 \neq 0 \wedge x_1 y_2 = y_1 x_2$

"Example"

We want to interpret the field of rationals $< \mathbb{Q}, 0, 1, +, * >$ inside of $< \mathbb{Z}, 0, 1, +, * >$. *I* is given as follows:

- $\phi_0(x_1, x_2) \equiv x_1 = x_1 \wedge x_2 \neq 0$
- $\epsilon(x_1, x_2, y_1, y_2) \equiv x_2 \neq 0 \wedge y_2 \neq 0 \wedge x_1 y_2 = y_1 x_2$
- $\phi_+(x_1, x_2, y_1, y_2, z_1, z_2) \equiv x_2, y_2, z_2 \neq 0 \wedge z_2 * (x_1 * y_2 + y_1 * x_2) = z_1 * x_2 * y_2$

"Example"

We want to interpret the field of rationals $< \mathbb{Q}, 0, 1, +, * >$ inside of $< \mathbb{Z}, 0, 1, +, * >$. I is given as follows:

- $\phi_0(x_1, x_2) \equiv x_1 = x_1 \wedge x_2 \neq 0$
- $\epsilon(x_1, x_2, y_1, y_2) \equiv x_2 \neq 0 \wedge y_2 \neq 0 \wedge x_1 y_2 = y_1 x_2$
- $\phi_+(x_1, x_2, y_1, y_2, z_1, z_2) \equiv x_2, y_2, z_2 \neq 0 \wedge z_2 * (x_1 * y_2 + y_1 * x_2) = z_1 * x_2 * y_2$
- $\phi_*(x_1, x_2, y_1, y_2, z_1, z_2) \equiv x_2, y_2, z_2 \neq 0 \wedge x_1 * y_1 * z_2 = x_2 * y_2 * z_1$

"Example"

We want to interpret the field of rationals $<\mathbb{Q}, 0, 1, +, *>$ inside of $<\mathbb{Z}, 0, 1, +, *>$. I is given as follows:

- $\phi_0(x_1, x_2) \equiv x_1 = x_1 \wedge x_2 \neq 0$
- $\epsilon(x_1, x_2, y_1, y_2) \equiv x_2 \neq 0 \wedge y_2 \neq 0 \wedge x_1 y_2 = y_1 x_2$
- $\phi_+(x_1, x_2, y_1, y_2, z_1, z_2) \equiv x_2, y_2, z_2 \neq 0 \wedge z_2 * (x_1 * y_2 + y_1 * x_2) = z_1 * x_2 * y_2$
- $\phi_*(x_1, x_2, y_1, y_2, z_1, z_2) \equiv x_2, y_2, z_2 \neq 0 \wedge x_1 * y_1 * z_2 = x_2 * y_2 * z_1$
- $\psi_0(x_1, x_2) \equiv x_1 = 0 \wedge x_2 \neq 0$

"Example"

We want to interpret the field of rationals $< \mathbb{Q}, 0, 1, +, * >$ inside of $< \mathbb{Z}, 0, 1, +, * >$. *I* is given as follows:

- $\phi_0(x_1, x_2) \equiv x_1 = x_1 \wedge x_2 \neq 0$
- $\epsilon(x_1, x_2, y_1, y_2) \equiv x_2 \neq 0 \wedge y_2 \neq 0 \wedge x_1 y_2 = y_1 x_2$
- $\phi_+(x_1, x_2, y_1, y_2, z_1, z_2) \equiv x_2, y_2, z_2 \neq 0 \wedge z_2 * (x_1 * y_2 + y_1 * x_2) = z_1 * x_2 * y_2$
- $\phi_*(x_1, x_2, y_1, y_2, z_1, z_2) \equiv x_2, y_2, z_2 \neq 0 \wedge x_1 * y_1 * z_2 = x_2 * y_2 * z_1$
- $\psi_0(x_1, x_2) \equiv x_1 = 0 \wedge x_2 \neq 0$
- $\psi_1(x_1, x_2) \equiv x_2 \neq 0 \wedge x_1 = x_2$

Finally. . .

Up to first-order queries, every structure can be viewed as a graph. More specifically,

Proposition

For every vocabulary τ, there are FO-queries I, J such that $I : STRUC[\sigma] \to STRUC[\tau_e], J : STRUC[\tau_e] \to STRUC[\sigma]$ and for every finite structure $\mathcal{A} \in STRUC[\sigma]$, $I(\mathcal{A})$ is a graph and $J(I(\mathcal{A})) \cong \mathcal{A}$.

Proof:

Finally. . .

Up to first-order queries, every structure can be viewed as a graph. More specifically,

Proposition

For every vocabulary τ, there are FO-queries I, J such that $I : STRUC[\sigma] \to STRUC[\tau_e], J : STRUC[\tau_e] \to STRUC[\sigma]$ and for every finite structure $\mathcal{A} \in STRUC[\sigma]$, $I(\mathcal{A})$ is a graph and $J(I(\mathcal{A})) \cong \mathcal{A}$.

Proof: Left as an exercise!

Finally. . .

Up to first-order queries, every structure can be viewed as a graph. More specifically,

Proposition

For every vocabulary τ, there are FO-queries I, J such that $I : STRUC[\sigma] \rightarrow STRUC[\tau_e], J : STRUC[\tau_e] \rightarrow STRUC[\sigma]$ and for every finite structure $\mathcal{A} \in STRUC[\sigma]$, $I(\mathcal{A})$ is a graph and $J(I(\mathcal{A})) \cong \mathcal{A}$.

Proof: Left as an exercise!

Note that we have already seen a similar proposition, namely that everything can be encoded as a string, but an ordering was necessary. This time, we do **not** need any ordering.

So what?

We can now transfer (1) (the question in the begging), on the class of finite graphs. Rather than searching for a logic that (effectively) captures P on the domain of all finite structures, we can just focus on finite graphs.

Corollaries

Proposition

If there is a logic that effectively captures $PTIME$ on the class of finite graphs, then there is a logic that effectively captures $PTIME$ on every structure.

Proposition

If there is a $PTIME$-canonization for graphs, then there is a $PTIME$-canonization and thus, there is a logic that effectively captures $PTIME$.

Conclusion

It has been shown that there are logics that (effectively) capture $PTIME$ on restrictions over the class of finite graphs. In particular, on finite trees, planar graphs, of bounded tree width and more.

But, as already mentioned, it is not known whether there exists a logic capable of expressing every (graph) property in $PTIME$.