A Logic that Captures $\beta$P on Ordered Structures

Kexu WANG,
email: wangkexuphy@163.com

Xishun ZHAO
email: hsszxs@mail.sysu.edu.cn
Institute of Logic and Cognition,
Department of Philosophy, Sun Yat-sen University

Abstract

We extend the inflationary fixed-point logic, IFP, with a new kind of second-order quantifiers which have (poly-)logarithmic bounds. We prove that on ordered structures the new logic $\exists^{\log^*}$IFP captures the limited nondeterminism class $\beta$P. In order to study its expressive power, we also design a new version of Ehrenfeucht-Fraïssé game for this logic and show that our capturing result will not hold in the general case, i.e., on all the finite structures.

Keywords: limited nondeterminism, $\beta$P, descriptive complexity, second-order quantifiers with bounds.

1 Introduction

In descriptive complexity theory, it is the most interesting task to find a logical characterization of a complexity class. But why do we need logics to characterize (or capture) complexity classes?

"Logics speak directly about graphs and structures, whereas most other formalisms operate on encodings of structures by strings or terms. Hence a logical characterization of a complexity class is representation-independent."

by Martin Grohe [Gro11]

We know in graph theory or database theory, more essentially we care about graph properties (or Boolean queries), i.e. the properties which do not depend on encoding. A graph property is always closed under isomorphism. This coincides with that the model class of a logic sentence is closed under isomorphism. Descriptive complexity theory intends to consider every logic sentence as a machine and vice versa. Thus every model of a sentence could be associated with an input of a corresponding machine and the logic (actually a class of sentences) would be related to a complexity class (actually a class of Turing machines). The precise definition will be given in 2.2.

In this paper, let’s turn to some limited (or bounded) nondeterminism classes, which are included in NP while including P. The idea of limited nondeterminism was first defined...
by Kintala et al [KF84]. Then in [CC97] Cai et al discussed a more general case, i.e. the “Guess-then-Check” model.

**Definition 1.1.** [CC97]

Let \( s : \mathbb{N} \to \mathbb{N} \) and \( C \) be a complexity class. A language \( L \) is in the class \( GC(s,C) \) if there is a language \( L' \in C \) together with an integer \( c > 0 \) such that for any string \( u, u \in L \) if and only if \( \exists v \in \{0, 1\}^*, |v| \leq c \cdot s(|u|) \), and \( u \neq v \in L' \).

Naturally \( NP = \bigcup_{i \in \mathbb{N}} GC(n^i, P) \). For any sublinear function \( f \), let’s define

\[
\beta_f = GC(f, P)
\]

Specially for \( k \in \mathbb{N} \) we denote \( \beta_k = GC(\log k, P) \) instead of \( \beta_{\log^*} \). Let

\[
\beta P = \bigcup_{k \in \mathbb{N}} \beta_k
\]

Correspondingly we introduce \( \exists^f \), the second-order quantifier bounded by \( f \). (We call this the \( f \)-bounded quantifier.) The semantics is straightforward. For any formula \( \phi \), any relation variable \( X \) and any structure \( \mathcal{A} \),

\[
\mathcal{A} \models \exists^f X \phi \iff \text{there is a subset } S \subseteq A^{arity}(X) \text{ with } |S| \leq f(|A|), \text{ such that } \mathcal{A} \models \phi[S]
\]

We care more about the second-order quantifiers with a logarithmic bound, written as \( \exists^{\log k} \). We call these log-quantifiers. The new logic \( \exists^{\log_1} \)IFP is obtained by extending the inflationary fixed-point logic IFP with all the log-quantifiers. The main theorem will show that \( \exists^{\log_1} \)IFP captures \( \beta P \) on ordered structures. An ordered structure is a structure whose domain has a built-in linear order. One can notice that the log-quantifiers will act as the part “\( \exists v \in \{0, 1\}^*, |v| \leq c \cdot s(|u|) \)” in definition 1.1. The log-quantifiers “guess” and then the IFP formula will “check”.

Our characterization is a natural extension of the famous Fagin’s theorem and Immerman-Vardi’s theorem. R. Fagin [Fag74] showed that NP is captured by the existential second-order logic \( \Sigma_1 \), which consists of formulas in the form

\[
\exists X_1 \ldots \exists X_m \phi
\]

where \( \phi \) is first order and \( X_1 \ldots X_m \) are relation variables. As a corollary of Fagin’s theorem, every layer of the polynomial time hierarchy, \( \text{PH} \), is captured by a layer of the second-order logic \( \Sigma^1 \). The fundamental result of capturing P is Immerman-Vardi’s theorem [Imm82, Var82]. It shows that IFP captures P on ordered structures.

The restriction on ordered structures is vital. Actually so far we do not know what logic can capture P without a built-in order. Logics are free from encoding, but when we intend to simulate a Turing machine with a logic sentence, it cannot be helped using a linear order to encode graphs or structures. This is related to a more fundamental and sophisticated problem, canonization (or canonical labeling) of graphs (or structures). A canonization is an algorithm which returns the unique labeling of a graph no matter how we label the vertices of the graph initially. The P-computable canonizations do exist on some certain classes of graphs, for instance, trees [Köb06], planar graphs [Köb06], graphs of bounded treewidth
Researchers are also interested in using logics to define a canonization. There are IFP-definable canonizations on cycles [EF05], grids [EF05] or 3-connected planar graphs [Gro98]. That means on these classes IFP can provide a canonical linear order and captures P. An important approach is to extend IFP to capture P on some more general classes. For example, IFP with counting, denoted by IFP+#, on trees [IL90], planar graphs [Gro98], graphs of bounded treewidth [GM99], graphs of bounded rank width [GN19].

Neither IFP nor IFP+# can capture P in the most general case, i.e., on all the finite structures. They were originally proved via the game method. Alongside this notion we will design a new Ehrenfeucht-Fraîssé game and prove $\exists \log^\omega$IFP fails to capture $\beta$P in the most general case, too.

## 2 Preliminaries

We assume that the readers are familiar with the basic concepts of computational complexity theory and mathematical logic. A signature $\tau$ is a finite class of relation symbols. For convenience, we do not talk about constant symbols and function symbols. $L[\tau]$ is the formulas of logic $L$ formed with symbols in $\tau$. A $\tau$-structure (or structure over $\tau$) $B$ explains the symbols in $\tau$ on a domain $B$. In this paper we only consider finite structures, i.e. whose domain is a finite set. $\text{STRUC}[\tau]$ is the class of all $\tau$-structures. A graph is a structure over signature $\{E\}$ whose domain $V$ is a set of vertices. $\text{STRUC}[\tau]_<$ is the class of all ordered $\tau$-structures (there is a built-in linear order of whose domain). STRING is the class of all strings. Let $\tau_{\text{str}} = \{<, P^0_0, P^1_1, P^#_#, P^{\langle}_\langle, P^{\rangle}_\rangle\}$. A string $u$ is a structure over $\tau_{\text{str}}$, i.e.

$$u = (U, <, P^u_0, P^u_1, P^u_#, P^u_{\langle}, P^u_{\rangle})$$

where

- $U = \{0, 1, \ldots, |u| - 1\}$
- $<$ is the natural linear order of $U$
- $P^u_0 i \iff$ the i-th bit of $u$ is 0
- $P^u_1 i \iff$ the i-th bit of $u$ is 1
- $P^u_# i \iff$ the i-th bit of $u$ is $\#$
- $P^u_{\langle} i \iff$ the i-th bit of $u$ is $\langle$
- $P^u_{\rangle} i \iff$ the i-th bit of $u$ is $\rangle$

“$#$” is used to separate two concatenated strings, for instance, “$u#v$”. “$<$” and “$>$” are used for encoding in definition 2.1. None of the three auxiliary symbols are theoretically necessary and all strings can be represented binarily, i.e. just with 0 and 1. However their attendance makes our proofs much easier.

A Boolean query $Q$ on $\tau$ is a class of structures over the same signature $\tau$, and closed under isomorphism, i.e. for any $\mathcal{A}, \mathcal{B} \in \text{STRUC}[\tau]$, if $\mathcal{A} \simeq \mathcal{B}$, then,

$$\mathcal{A} \in Q \iff \mathcal{B} \in Q$$
For example, languages (classes of strings) are Boolean queries on \(\tau_{str}\).

In the following context, we often use the logarithmic function \(\log(n)\), whose value is expected to be an integer, so we let \(\log(n) = \lceil \log_2(n) \rceil\). Let \([n] = \{0, 1, \ldots, n-1\}\). Note that \(\log(n + 1)\) is the minimal length of \(n\)’s binary expression. In this paper, for any formula \(\phi(x, X)\), “\(\phi[x, X]\)” means the value \(a\) (resp. \(R\)) is substituted into \(x\) (resp. \(X\)) if \(x\) (resp. \(X\)) is free. We abuse the notation \(|·|\). If \(u\) is a string, \(|u|\) is its length. If \(A\) is a set, \(|A|\) is its cardinality. If \(\vec{x}\) is a \(k\)-tuple, then \(|\vec{x}| = k\).

2.1 Encoding structures

In order to represent the structures in a Turing machine, we need to encode structures as strings. W.l.o.g., we take the following way of encoding:

**Definition 2.1 (Enumerating encoding).** For any signature \(\tau = \{R_1, \ldots, R_m\}\), where \(\text{arity}(R_i) = r_i\) (1 \(\leq i \leq m\)), any \(\mathcal{A} \in \text{STRUC}[\tau]<\) with domain \(A = \{a_0, \ldots, a_{|A|-1}\}\)

1. \(\text{enc}(\mathcal{A}) = (\text{enc}(A)\text{enc}(R_{1\mathcal{A}}) \ldots \text{enc}(R_{m\mathcal{A}})))\)

2. \(\text{enc}(A) = (\text{enc}(a_0) \ldots \text{enc}(a_{|A|-1}))\)

3. For any \(i \in \{1, \ldots, m\}\), suppose \(\vec{a}_1, \ldots, \vec{a}_{|R_{i\mathcal{A}}|}\) are all \(r_i\)-tuples in \(R_{i\mathcal{A}}\),
\[
\text{enc}(R_{i\mathcal{A}}) = (\text{enc}(\vec{a}_1), \ldots, \text{enc}(\vec{a}_{|R_{i\mathcal{A}}|}))
\]

4. Suppose \(\vec{t} = (t_1, \ldots, t_s)\) is a tuple with \(t_1, \ldots, t_s \in A\),
\[
\text{enc}(\vec{t}) = (\text{enc}(t_1) \ldots \text{enc}(t_s))
\]

5. Suppose \(a\) is the \(j\)-th element in \(A\), 0 \(\leq j < |A|\),
\[
\text{enc}(a) = (\text{“the log }|A|\text{-long binary expression of }j\text{”})
\]

Note that
\[
|\text{enc}(\mathcal{A})| = \log |A| \cdot O\left(\sum_{1 \leq i \leq m}(|R_{i\mathcal{A}}| \cdot r_i)\right)
\]
and for \(i \in \{1, \ldots, m\}\),
\[
|\text{enc}(R_{i\mathcal{A}})| = \log |A| \cdot O(|R_{i\mathcal{A}}| \cdot r_i)
\]

The length \(|\text{enc}(\mathcal{A})|\) is related to every size \(|R_{i\mathcal{A}}|\). The machine needs the auxiliary symbols to parse \(\text{enc}(\mathcal{A})\) because it cannot know ahead of time how long \(\text{enc}(R_{i\mathcal{A}})\) is. The extra length of auxiliary symbols can be ignored in a big-Oh notation.

2.2 Logic characterization of complexity

**Definition 2.2.** A logic \(\mathcal{L}\) captures a complexity class \(\mathcal{C}\) on a class \(\mathcal{K}\) of structures, if the following conditions are satisfied,

1. \(\mathcal{L}[\tau]\) is decidable, for any signature \(\tau\).
2. There is an effective procedure to associate with each \( L \)-sentence \( \phi \) a \( C \)-bounded Turing machine \( M \), such that, for any \( A \in K \), \( M \) can decide whether
\[
A \models \phi
\]

3. For any Boolean query \( Q \) in \( C \), there is an \( L \)-sentence \( \phi \) such that for any \( A \in K \),
\[
A \models \phi \iff A \in Q
\]
(We assume that \( K \) is closed under isomorphism.)

If \( K \) is the class of all structures, we simply say \( L \) captures \( C \).

There are two most classical theorems in descriptive complexity theory.

**Theorem 2.3** (Fagin’s Theorem). \([\text{Fag74}]\)
\( \Sigma_1^1 \) captures \( \text{NP} \).

**Theorem 2.4** (Immerman-Vardi Theorem). \([\text{Imm82, Var82}]\)
IFP captures \( \text{P} \) on ordered structures.

IFP is gotten by extending the first-order logic \( \text{FO} \) with the inflationary fixed-point operator. IFP inherits the formation rules of \( \text{FO} \) besides

- If \( \psi \) is a formula, then so is \( \text{IFP} \ y Y \psi(\bar{y},Y) \bar{t} \), where \( Y \) is a relation variable and \( |\bar{y}| = |\bar{t}| = \text{arity}(Y) \)

[IFP \ y Y \psi(\bar{y},Y) \bar{t} \] is the fixed point of the function \( f_{Y \psi} \) defined by the formula \( Y \bar{y} \vee \psi(\bar{y},Y) \).

This semantics will not be used in this paper, so readers can turn to \([\text{EF05}] \) and \([\text{Lib13}] \) for details.

In logic we needn’t even study structures over all different signatures. We particularly care about \( \text{STRING} \) and graphs, which the structures over other signatures can be interpreted to.

**Definition 2.5.** Let \( L \) be a logic. Let \( \tau, \sigma \) be two signatures. \( \sigma = \{ R_1, R_2, \ldots, R_m \} \), where \( \text{arity}(R_i) = r_i \) (1 \( \leq \) i \( \leq \) m). A k-ary \( L \)-interpretation from \( \tau \) to \( \sigma \) is a series of \( L[\tau] \)-formulas
\[
I = (\phi_{\text{uni}}(\bar{x}), \phi_{R_1}(\bar{x}_1, \ldots, \bar{x}_{r_1}), \ldots, \phi_{R_m}(\bar{x}_1, \ldots, \bar{x}_{r_m}))
\]
where the variables \( \bar{x}, \bar{x}_1, \ldots \) are k-tuples. For any \( A \) on \( \tau \),
\[
I(A) = (\phi^A_{\text{uni}}(\bar{a}), \phi^A_{R_1}(\bar{a}_1, \ldots), \ldots, \phi^A_{R_m}(\bar{a}_1, \ldots))
\]
is a \( \sigma \)-structure, if we consider the k-tuples satisfying \( \phi^A_{\text{uni}}(\bar{x}) \) as individual elements. (Note that: \( \phi^A \models (\bar{a}) := \{ \bar{a} \mid A \models \phi(\bar{a}) \})

Suppose \( S_1 \subseteq \text{STRUC}[\tau] \) and \( S_2 \subseteq \text{STRUC}[\sigma] \) are two Boolean queries. If \( I \) also makes sure for any \( A \in \text{STRUC}[\tau] \),
\[
A \in S_1 \iff I(A) \in S_2
\]
we say \( I \) is an \( L \)-reduction from \( S_1 \) to \( S_2 \).

It is not hard to prove for any \( \mathcal{A}, \mathcal{B} \in \text{STRUC}[\tau] \)
\[
\mathcal{A} \simeq \mathcal{B} \implies I(\mathcal{A}) \simeq I(\mathcal{B})
\]
Lemma 2.6. For any signature \( \tau \), there is an FO-reduction \( I \) from \( \text{STRUC}[\tau]_< \) to STRING and for any \( \mathcal{A}, \mathcal{B} \) in \( \text{STRUC}[\tau]_< \),
\[
\mathcal{A} \simeq \mathcal{B} \iff I(\mathcal{A}) = I(\mathcal{B})
\]

Lemma 2.7. \([EF05]\) Let \( \phi \) be a formula of \( \text{IFP}[\sigma] \)
\[
I = \langle \phi_{\text{uni}}, \phi_{R_1}, ..., \phi_{R_m} \rangle
\]
is a \( k \)-ary reduction from \( \text{STRUC}[\tau] \) to \( \text{STRUC}[\sigma] \). There exists an IFP[\( \tau \)]-sentence \( \phi^I \) such that for any \( \mathcal{A} \in \text{STRUC}[\tau] \),
\[
\mathcal{A} \models \phi^I \iff I(\mathcal{A}) \models \phi
\]

These two lemmas tell us STRING and ordered structures are deeply related. In the following context, we will first prove our theorem on STRING, and naturally it holds on ordered structures.

3 Capturing Results

Here is an alternative definition of \( \beta P \) prepared for our later proofs

Definition 3.1. A language \( L \) is in the class \( \beta_k \) if there is a language \( L' \in P \) together with an integer \( c > 0 \) such that for any string \( u, u \in L \) if and only if \( \exists v \in \{0,1\}^{\leq c \cdot \log k(|u|)} \), \( u \neq v \in L' \). (where \( \{0,1\}^{\leq c \cdot \log k(|u|)} \) is all the 0-1 strings of length at most \( c \cdot \log k(|u|) \).)
\[
\beta P = \bigcup_{k \in \mathbb{N}} \beta_k.
\]

Since \( \beta_1 = GC(\log, P) \), in fact the “guess” part can be computed in time \( 2^{c \log} \), which is a polynomial. Thus we have
\[
P = \beta_1 \subseteq \beta_2 \subseteq \cdots \subseteq \beta P \subseteq \text{NP}
\]

3.1 Logarithmic-bounded quantifiers

The log-quantifier \( \exists^{\log k} \) is the second-order quantifier with a bound \( \log k \). As we mentioned,
\[
\mathcal{A} \models \exists^{\log k} X \phi \iff \text{there is a subset } S \subseteq A^{\text{arity}(X)} \text{ with } |S| \leq \log k(|A|),
\]
such that \( \mathcal{A} \models \phi[\frac{X}{S}] \)

It doesn’t matter how large \( \text{arity}(X) \) is. As long as \( \text{arity}(X) \) is a nonzero natural number, \( \exists^{\log k} \) can be applied. Naturally
\[
\forall^{\log k} X := \neg \exists^{\log k} X
\]
Let \( \log^\omega = \{\log k \mid k > 0\} \). Then \( \exists^{\log^\omega} = \{\exists^{\log k} \mid k > 0\} \)

Definition 3.2. An formula of \( \exists^{\log^\omega} \text{IFP} \) is in the form,
\[
\exists^{\log^{k_1}} X_1 \exists^{\log^{k_2}} X_2 \cdots \exists^{\log^{k_m}} X_m \psi
\]
where \( m \geq 0; k_1, k_2, \ldots, k_m > 0; \psi \) is an IFP-formula.
Those formulas without any occurrences of log-quantifiers are \textit{log-quantifier-free}.

Here are three parameters we will use. The \textit{maximal variable arity} of a formula, \(mva(\phi) = \max\{\text{arity}(X) \mid X \text{ is a relation variable, free or bounded by a log-quantifier, in } \phi\}\). The \textit{height} of a formula, \(\text{height}(\phi) = \max\{k \mid \exists^{\log^k} \text{ or } \forall^{\log^k} \text{ occurs in } \phi\}\). The \textit{log-quantifier rank} of a formula, 

- \(lqr(\phi) = 0\), if \(\phi\) is atomic
- \(lqr(\phi) = lqr(\psi)\), if \(\phi = \neg \psi\)
- \(lqr(\phi) = \max(lqr(\psi_1), lqr(\psi_2))\), if \(\phi = \psi_1 \rightarrow \psi_2\)
- \(lqr(\phi) = lqr(\psi)\), if \(\phi = \exists x \psi\)
- \(lqr(\phi) = lqr(\psi) + 1\), if \(\phi = \exists^{\log^k} X \psi\) for \(k > 0\).

For \(k > 0\), \(\exists^{\log^k} \text{IFP}\) is the sublogic of \(\exists^{\log^\omega} \text{IFP}\), the heights of whose formulas are no larger than \(k\).

\section{Main theorem}

**Theorem 3.3.** \(\exists^{\log^k} \text{IFP}\) captures \(\beta P\) on STRING.

Actually we will prove for \(k \geq 1\), \(\exists^{\log^k} \text{IFP}\) captures \(\beta_{k+1}\) on STRING. Note that an \(\exists^{\log^k} \text{IFP}[\tau_{st}]-\text{sentence}\) corresponds to a \(\beta_{k+1}\)-bounded Turing machine, not a \(\beta_k\)-bounded one. It is because for any \(u \in \text{STRING}\) and any relation variable \(X\), when we encode the value of \(X\), as we did in definition 2.1. \(|\text{enc}(X)| = O(|\log^{k+1}[U]|)\). According to definition 2.2, our proof consists of three parts. The main idea is simple: we use \(\exists^{\log^k} X\) to simulate \(\exists v \in \{0, 1\}^{\leq \log^k(|u|)}\) in definition 3.1 and vice versa; then we apply Immerman-Vardi’s theorem.

But here is a problem: for any \(v\) in \(\exists v \in \{0, 1\}^{\leq \log^k(|u|)}\) in definition 3.1 can we have an IFP-reduction \(I\) such that there exists \(X\) in \(\exists^{\log^k} X\) and \(I(X) = v\)?

**Lemma 3.4.** Let \(k \in \mathbb{N} - \{0\}\)

There is an encoding \(J\) such that for any string \(u\) with domain \(U\), \(J^U\) is a \textit{surjection} from \(\{S \mid S \subseteq U^3 \text{ and } |S| \leq \log^k(|U|)\}\) to \(\{0, 1\}^{\leq \log^k(|U|)}\).

And let \(\tau_r = \tau_{st} \cup \{R_1, R_2, \ldots, R_t\}\), where \(R_1, \ldots, R_t\) are ternary relation symbols. There is an IFP-reduction \(I\) from \(\text{STRUC}[\tau_r]\) to STRING such that for any \(u \in \text{STRING}\) and binary relations \(R_1^u, \ldots, R_t^u \in \{S \mid S \subseteq U^3 \text{ and } |S| \leq \log^k(|U|)\}\),

\[ I((u, R_1^u, \ldots, R_t^u)) = u \# J^U(R_1^u) J^U(R_2^u) \ldots J^U(R_t^u) \]

**Proof.** (of lemma 3.4)

With the help of the linear order \(<^u\), we can construct IFP-formula \(\text{BIT}(y, x)\), which means “the \(x\)-th bit of the binary expression of \(y\) is 1\). (But here we do not provide the details of BIT. The readers can check Page 96 of [Lib13].)
Let \( \vec{x} = x_1 x_2 x_3 x_4 y_1 \ldots x_r x_5 y_6 \). It’s an \((r + 7)\)-ary tuple of variables. Now we define:

\[
\phi_{<} \text{ is the lexicographic order of these tuples}
\]

\[
\phi_{R_0}(\vec{x}) := (x_1 = x_2 = 0 \land P_0(x_4)) \lor (x_1 = x_2 = 1 \land x_4 = 0)
\]

\[
\phi_{R_1}(\vec{x}) := (x_1 = x_2 = 0 \land P_1(x_4)) \lor (x_1 = x_2 = 1 \land x_4 = 1)
\]

\[
\phi_{R_2}(\vec{x}) := (x_1 = x_2 = 0 \land P_2(x_4)) \lor (x_1 = 0 \land x_2 = 1)
\]

\[
\phi_{P_i}(\vec{x}) := (x_1 = x_2 = 0 \land P_i(x_4))
\]

\[
\phi_{P_i}(\vec{x}) := (x_1 = x_2 = 0 \land P_i(x_4))
\]

\[
\phi_{\text{uni}}(\vec{x}) := (x_1 = x_2 = 0
\]

\[
\land y = z_1 = \ldots = z_r = x_3 = x_5 = x_6 = 0)
\]

\[
\lor (x_1 = 0 \land x_2 = 1
\]

\[
\land y = z_1 = \ldots = z_r = x_3 = x_4 = x_5 = x_6 = 0
\]

\[
\lor (x_1 = x_2 = 1
\]

\[
\land (\bigvee_{1 \leq i \leq r} (R_i x_5 x_6 y
\]

\[
\land z_i = 0 \land z_1 = \ldots = z_{i-1} = z_{i+1} = \ldots = z_r = 1
\]

\[
\land x_6 < \log(|U|) - 1
\]

\[
\land x_3 \leq x_6
\]

\[
\land x_4 = 1 \leftrightarrow \text{BIT}(y, x_3)
\]

\[
\land x_4 = 0 \leftrightarrow \neg \text{BIT}(y, x_3))\)
\]

Let \( I = (\phi_{\text{uni}}, \phi_{<}, \phi_{R_0}, \phi_{R_1}, \phi_{R_2}, \phi_{P_i}, \phi_{P_i}) \). In each \((r + 7)\)-ary tuple, \( x_1 \) and \( x_2 \) indicates the tuple to be in \( u \), \# or \( \text{IFP}^U(R_i^n) \cdot \text{IFP}^U(R_2^n) \ldots \text{IFP}^U(R_2^n) \). For any 01 string \( v \) with \(|v| \leq \log(|u|) - 1 \), there exists some \( R_i^n \) in which there is a tuple \( x_5 x_6 y \) such that

- \( x_6 + 1 = |v| \) and
- the first \((x_6 + 1)\) bits of the binary expression of \( y \) is exactly \( v \).

(Note that when we change the bound \( \log(|u|) - 1 \) to \( \log(|u|) \), this will no more hold.) The variable \( x_5 \) makes sure that \( v \) can repeat in \( \text{IFP}^U(R_i^n) \) at most \( \log(|u|) \) times. This means for any 01 string of length \( \leq \log(|u|) \cdot (\log(|u|) - 1) \), we choose a \( R_i^n \) to make \( \text{IFP}^U(R_i^n) \) equal to it. So \( \text{IFP}^U \) is a surjection. \( \square \)

**Proof. (of theorem 3.3)**

By definition 2.22 our proof consists of three parts. Let \( k > 0 \).

Firstly. \( \exists^{\log^k} \text{IFP}^{[\tau]} \) is decidable, for any signature \( \tau \).

Secondly. For any \( \exists^{\log^k} \text{IFP}^{[\text{str}]} \)-sentence \( \phi = \exists^{\log^k} X_1 \ldots \exists^{\log^k} X_m \psi \), where \( \psi \) is log-quantifier-free and all its relation variables are among \( X_1 \ldots X_m \) and \( k_1, \ldots k_m \leq k \). We construct a \( \beta_{k+1} \)-bounded Turing machine \( M_\phi \) as follows: for any \( u \in \text{STRING} \),

\[
u \models \phi \iff \text{there are } S_1 \subseteq U^{\text{arity}}(X_1), \ldots, S_m \subseteq U^{\text{arity}}(X_m)
\]

\[
\text{and } |S_1| \leq \log^{k_1} |u|, \ldots, |S_m| \leq \log^{k_m} |u|
\]

\[
\text{such that } u \models \psi[X_1 \in S_1, \ldots, X_m \in S_m]
\]
By theorem 2.4 there is a P-bounded Turing machine $M_\phi$ that can verify whether

$$M_\phi \upharpoonright \psi_{X_1, ..., X_m}$$

for $\mathcal{A}$ on $\tau_{\text{str}} \cup \{X_1, ..., X_m\}$ and $\mathcal{A}$’s explanation $R_1, ..., R_m$ of $X_1, ..., X_m$.

In order to guess and store the values of $X_1, ..., X_m$, by definition 2.4, $M_\phi$ will need

$$O(\log^{k_1+1}|u| \cdot \text{arity}(X_1) + ... + \log^{k_m+1}|u| \cdot \text{arity}(X_m))$$
	nondeterministic bits, or simply, $O(\log^{k+1}|u|)$ nondeterministic bits in total.

then $M_\phi$ returns TRUE if there are $S_1, ..., S_m$ with $|S_1| \leq \log^k |u|$, ..., $|S_m| \leq \log^k |u|$, such that $M_\phi$ accepts $\langle u, S_1, ..., S_m \rangle$. Otherwise $M_\phi$ returns FALSE.

So $M_\phi$ is a $\beta_{k+1}$-bounded machine that we want.

Thirdly. Suppose $L$ is a language in $\beta_{k+1}$. By definition 3.1 there is a natural number $c$ and a P-bounded Turing machine $M$, such that for any $u \in \text{STRING}$,

$$u \in L \iff \exists v \in \{0, 1\}^{\leq c \log^{k+1}(|u|)} M \text{ accepts } u \# v$$

There exists $r \in \mathbb{N} - \{0\}$ such that for any $n \in \mathbb{N} - \{0\}$, $c \cdot \log^{k+1}(|u|) \leq r \cdot \log^k(n) \cdot (\log(n) - 1)$. Let $R_1, ..., R_r$ be new ternary relation symbols. We can construct a P-bounded Turing machine $M'$ such that for any strings $u, v, z$ with $v \in \{0, 1\}^{\leq c \log^{k+1}(|u|)}$ and $z \in \{0, 1\}^{\leq r \cdot \log^k(|u|)/(\log(|u|)-1)}$

$$M' \text{ accepts } u \# z \iff M \text{ accepts } u \# v$$

and $v$ is the leftmost $c \cdot \log^{k+1}(|u|)$ bits of $z$.

By theorem 2.4 there is an IFP[$\tau_{\text{str}}$]-sentence $\phi_{M'}$ such that for any $v \in \text{STRING}$,

$$v \vDash \phi_{M'} \iff M' \text{ accepts } v$$

By lemma 3.3 there is a $(r + 7)$-ary IFP reduction from STRUC[$\tau_{\text{str}} \cup \{R_1, ..., R_r\}$] to STRING, $I = \langle \phi_{\text{uni}}, \phi_c, \phi_{P_0}, \phi_{P_1}, \phi_{P_0w}, \phi_{P_1w}, \phi_{P_0}, \phi_{P_1} \rangle$. With the help of lemma 2.7 let

$$\psi := \phi_{M'}^I$$

$\psi$ is an IFP-sentence on $\tau_{\text{str}} \cup \{R_1, ..., R_r\}$. Let

$$\phi = \exists \log^k R_1, ..., \exists \log^k R_r \psi$$

which is an $\exists \log^k$ IFP[$\tau_{\text{str}}$]-sentence. And for any $u \in \text{STRING}$,

$$u \in L \iff u \vDash \phi$$

\[ \square \]

4 The Expressive Power

IFP fails on a very important P-decidable Boolean query, EVEN [EF05]. For any graph $G$, $G \in \text{EVEN}$ if and only if domain $|V|$ is even. There is no sentence $\phi$ of IFP[|{E}|] such that

$$G \in \text{EVEN} \iff G \vDash \phi$$

(EVEN is not definable in IFP.) So IFP fails to capture P (on all finite structures). Unfortunately, our strengthened version $\exists \log^\omega$ IFP fails, too.
Theorem 4.1. EVEN is not definable in $\exists^\text{log}\omega \text{IFP}$.

IFP’s failure was proven via the failure of the infinitary logic $\mathcal{L}_{\infty\omega}$. The logic $\mathcal{L}_{\infty\omega}^s$ is similar to FO, but every formula in $\mathcal{L}_{\infty\omega}^s$ can have infinite length or infinite quantifier depth but contains at most $s$ variables (free or bounded). Then

$$\mathcal{L}_{\infty\omega} = \bigcup_{s \in \mathbb{N}} \mathcal{L}_{\infty\omega}^s$$

Readers can turn to Chapter 3 of [EF05] for the details. For every single IFP-formula, we can always construct an equivalent $\mathcal{L}_{\infty\omega}^s$-formula for some $s$. So IFP is a sublogic of $\mathcal{L}_{\infty\omega}$.

Now we define a new logic $\mathcal{L}$ (Beware! It is not $\mathcal{L}$!) as follows: for any formula $\phi$, $\psi$ and $\chi$,

- $\phi \in \mathcal{L}$ if $\phi \in \mathcal{L}_{\infty\omega}^\omega$
- $\exists^{\text{log}}k X\phi \in \mathcal{L}$ if $\phi \in \mathcal{L}$, where $k > 0$ and $X$ is some relation variable.
- $\forall^{\text{log}}k X\phi \in \mathcal{L}$ if $\phi \in \mathcal{L}$, where $k > 0$ and $X$ is some relation variable.
- $\psi \land \chi \in \mathcal{L}$ if $\psi \in \mathcal{L}$ and $\chi \in \mathcal{L}$
- $\psi \lor \chi \in \mathcal{L}$ if $\psi \in \mathcal{L}$ and $\chi \in \mathcal{L}$

Obviously $\exists^\text{log}\omega$ IFP is a sublogic of $\mathcal{L}$

In order to prove theorem 4.1, we turn to the game method

Definition 4.2. $\mathcal{L}$ is any logic. $G$ is a game played by two players, the spoiler and the duplicator, on two structures. we say $G$ is an Ehrenfeucht-Fraissé game for $\mathcal{L}$, if for any $\tau$, any $A$ and $B \in \text{STRUC}[\tau]$, the following are equivalent,

1. $A \equiv^\mathcal{L} B$
2. the duplicator wins $G(A, B)$

where “$\mathcal{A} \equiv^\mathcal{L} \mathcal{B}$” means for any $\mathcal{L}[\tau]$-sentence $\phi$, $\mathcal{A} \vDash \phi$ if and only if $\mathcal{B} \vDash \phi$.

For convenience, we use the notation “$\bar{a}$”, a lowercase letter with a bar to represent a ordered set of elements and “$\bar{R}$”, a capital letter with a bar to represent a ordered set of relations. Please note that $\bar{a}$ is not tuple $\vec{a}$. In the following context we will denote $\bar{a}a = \bar{a} \cup \{a\}$, $\bar{R}R = \bar{R} \cup \{R\}$. If $\bar{a}$ consists of elements in $\bar{a}$, we simply say $\bar{a}$ is from $\bar{a}$. We say $\bar{a} \mapsto \bar{b} \in \text{Part}(\mathcal{A}, \bar{P}, \bar{R}, \bar{Q})$, i.e. $\bar{a} \mapsto \bar{b}$ is a partial isomorphism from $\langle \mathcal{A}, \bar{P}, \bar{R}, \bar{Q} \rangle$ to $\langle \mathcal{B}, \bar{S}, \bar{R} \rangle$, where $\bar{R} = \{R_1, \ldots, R_l\}$ and $\bar{S} = \{S_1, \ldots, S_l\}$, that is, there is a bijection $f$ from $\bar{a}$ to $\bar{b}$,

1. $f(a_i) = b_i$, $a_i \in \bar{a}$, $b_i \in \bar{b}$,
2. for any relation $P \in \tau$, and any tuple $\bar{t}$ from $\bar{a}$,
   $$\bar{t} \in P^\mathcal{A} \iff f(\bar{t}) \in P^\mathcal{B}$$
3. for $1 \leq i \leq l$, and any tuple $\bar{t}$ from $\bar{a}$,
   $$\bar{t} \in R_i \iff f(\bar{t}) \in S_i$$
In the expansions, actually $\bar{R}, \bar{S}$ act as new relations.

The Ehrenfeucht-Fraïssé game for $L^s_{\infty\omega}$ is the pebble game with $s$ pairs of pebbles, denoted by PG$^s$. In a play of PG$^s(\mathcal{A}, \mathcal{B})$, there are $s$ or less vertices in each of $\mathcal{A}$ and $\mathcal{B}$ covered by pebbles. In each move, each player either does nothing, moves one pebble or adds a new pebble (but on each structures there can be at most $s$ pebbles). If the duplicator can make sure the two covered substructures are always isomorphic, then she wins PG$^s(\mathcal{A}, \mathcal{B})$.

For the details readers can turn to Chapter 3 of [EF05].

Let’s design a game $G^{m,r,k,s}$ for $L^{m,r,k,s}$. As $L^{m,r,k,s}$ is extended from $L^s_{\infty\omega}$ with log-quantifiers in the “outer layers”, $G^{m,r,k,s}$ consists of at most $m$ relation moves and a game PG$^s$. The players plays a relation move as follows. The spoiler chooses $r' \leq r$ and $k' \leq k$. Then she chooses either $\mathcal{A}$ or $\mathcal{B}$. (W.l.o.g. we assume the spoiler chooses $\mathcal{A}$. Otherwise $\mathcal{A}$ and $\mathcal{B}$ are exchanged.) Then she chooses $R \subseteq A^{r'}$ with $|R| \leq \log^k(|A|)$. At last the duplicator chooses $S \subseteq B^{r'}$ with $|S| \leq \log^k(|B|)$.

In a play of $G^{m,r,k,s}(\mathcal{A}, \mathcal{B})$, the spoiler first chooses an arbitrary $m' \leq m$ and they play $m'$ relation moves and then the two structures are expanded as $\langle \mathcal{A}, R_1, \ldots, R_{m'} \rangle$ and $\langle \mathcal{B}, S_1, \ldots, S_{m'} \rangle$. Then they play PG$^s(\langle \mathcal{A}, R_1, \ldots, R_{m'} \rangle, \langle \mathcal{B}, S_1, \ldots, S_{m'} \rangle)$. Once this pebble game begins, no more relation moves are allowed. If the duplicator wins PG$^s(\langle \mathcal{A}, R_1, \ldots, R_{m'} \rangle, \langle \mathcal{B}, S_1, \ldots, S_{m'} \rangle)$, she wins the play.

If she can always win every play, we say she wins (or she has a winning strategy of) $G^{m,r,k,s}(\mathcal{A}, \mathcal{B})$.

**Proposition 4.3.** For $m \geq 0$, $r, k, s > 0$, $G^{m,r,k,s}$ is an Ehrenfeucht-Fraïssé game for $L^{m,r,k,s}$.

**Proof.** Let $\mathcal{A}$ and $\mathcal{B}$ be two structures over a given signature $\tau$.

We construct the isotype of $\mathcal{A}$. Let $\bar{R}$ be a set of new relations such that for any $R \in \bar{R}$, $\text{arity}(R) \leq r$ (and $|R| \leq \log^k(|A|)$).

$$\phi^{0,r,k,s}_{\mathcal{A}, \bar{R}}(X) = \bigwedge \{ \phi(\bar{X}) \mid \phi \text{ is an sentence of } L^s_{\infty\omega}[\tau \cup \bar{X}] \text{ such that } \mathcal{A} \models \phi(\bar{X}) \}$$

then inductively,

$$\phi^{m+1,r,k,s}_{\mathcal{A}, \bar{R}}(X) = \bigwedge \bigwedge_{i \leq r \leq k} \left( \bigwedge_{R \subseteq A^i, |R| \leq \log^j(|A|)} \exists \log^j X \phi^{m,r,k,s}_{\mathcal{A}, \bar{R}}(X) \right)$$

$$\wedge (\forall \log^j X \bigvee_{R \subseteq A^i, |R| \leq \log^j(|A|)} \phi^{m,r,k,s}_{\mathcal{A}, \bar{R}}(X) \right)$$

When $\bar{R} = \emptyset$, we simply write $\phi^{m,r,k,s}_{\mathcal{A}}$, which is a sentence of $L^{m,r,k,s}$.
Suppose \( \mathcal{A} \equiv L^{m,r,k,s} \mathcal{R} \), then \( \mathcal{B} \models \phi^{m,r,k,s}_{\mathcal{R}} \). The isotype indicates a winning strategy for the duplicator. After \( m \) moves if the two structures are expanded as \( \langle \mathcal{A}, \hat{R} \rangle \) and \( \langle \mathcal{B}, \hat{S} \rangle \),

\[
\mathcal{B} \models \phi^{0,r,k,s}_{\mathcal{A}, \hat{R}} \left[ \frac{\hat{X}}{\hat{S}} \right]
\]

This means \( \langle \mathcal{A}, \hat{R} \rangle \) and \( \langle \mathcal{B}, \hat{S} \rangle \) satisfy the same \( L_{\omega_{\omega}}^{s} \)-formulas. Therefore the duplicator can win \( PG^{*}(\langle \mathcal{A}, \hat{R} \rangle, \langle \mathcal{B}, \hat{S} \rangle) \). Then she wins \( G^{m,r,k,s}(\mathcal{A}, \mathcal{B}) \).

Suppose \( \mathcal{A} \not\equiv L^{m,r,k,s} \mathcal{B} \). There is a sentence \( \phi \) of \( L^{m,r,k,s} \) which \( \mathcal{A} \) and \( \mathcal{B} \) disagree on. W.l.o.g., we assume that \( \mathcal{A} \models \phi \) and \( \mathcal{B} \not\models \phi \) and

\[
\phi = Q_{1}^{\log k_{1}} X_{1} \ldots Q_{m}^{\log k_{m}} X_{m} \psi
\]

where \( \psi \) is an \( L_{\omega_{\omega}}^{s} \)-sentence and \( k_{1}, \ldots, k_{m} \leq k \) and \( Q_{1}, \ldots, Q_{m} \in \{ \exists, \forall \} \). Then

- \( \mathcal{A} \models Q_{1}^{\log k_{1}} X_{1} \ldots Q_{m}^{\log k_{m}} X_{m} \psi \)
- \( \mathcal{B} \models \hat{Q}_{1}^{\log k_{1}} X_{1} \ldots \hat{Q}_{m}^{\log k_{m}} X_{m} \neg \psi \)

(if \( Q_{i} = \exists \), then \( \hat{Q}_{i} = \forall \); if \( Q_{i} = \forall \), then \( \hat{Q}_{i} = \exists \), \( 1 \leq i \leq m \)). This provides a winning strategy for the spoiler. In the \( i \)-th relation move if \( Q_{i} = \exists \) then the spoiler should choose \( \mathcal{A} \) and the relation \( R_{i} \subseteq A^{arity(X_{i})} \); otherwise she should choose \( \mathcal{B} \) and the relation \( S_{i} \subseteq B^{arity(X_{i})} \). After \( m \) relation moves, the structures have been expanded as \( \langle \mathcal{A}, R_{1}, \ldots, R_{m} \rangle \) and \( \langle \mathcal{B}, S_{1}, \ldots, S_{m} \rangle \).

- \( \langle \mathcal{A}, R_{1}, \ldots, R_{m} \rangle \models \psi[\frac{X_{1}}{R_{1}}, \ldots, \frac{X_{m}}{R_{m}}] \)
- \( \langle \mathcal{B}, S_{1}, \ldots, S_{m} \rangle \models \neg \psi[\frac{X_{1}}{S_{1}}, \ldots, \frac{X_{m}}{S_{m}}] \)

The duplicator cannot win \( PG^{*}(\langle \mathcal{A}, \hat{R} \rangle, \langle \mathcal{B}, \hat{S} \rangle) \). So she cannot win \( G^{m,r,k,s}(\mathcal{A}, \mathcal{B}) \). \( \square \)

For any \( \mathcal{A} \in \text{STRUC}[\varpi] \), \( R \subseteq A^{arity(R)} \) and \( a \in A \), we say \( R \) mentions \( a \) (or \( a \) is mentioned by \( R \)), if \( a \) is a component of some tuple \( \bar{t} \in R \). Let \( \text{ment}(R) = \{ a \in A \mid a \) is mentioned by \( R \} \). Observe that if \( R \) is bounded by log-quantifier \( \exists^{\log k} \), then

\[
|\text{ment}(R)| \leq \text{arity}(R) \cdot \log^{k}(|A|)
\]

and we denote \( \text{ment}(\hat{R}) = \bigcup_{R \in R \text{ment}(R)} \)

**Theorem 4.4.** EVEN is not definable in \( \mathcal{L} \).

**Proof.** If EVEN is defined by a sentence \( \phi \) of \( L[\{ E \}] \), \( \phi \) should also work on empty graphs, namely on the graphs that have no edges. Now we assume \( E = \emptyset \) in order to get a contradiction. There are \( m \geq 0 \) and \( r, k, s > 0 \) such that \( \phi \in L^{m,r,k,s}[\{ E \}] \). Let \( \mathcal{A} \) and \( \mathcal{B} \) be two empty graphs such that \( |A| \) is a sufficiently large even number satisfying

\[
(m + 1) \cdot r \cdot s \cdot \log^{k}(|A|) < |A|
\]

and \( |B| = |A| + 1 \) and \( \log(|A|) = \log(|B|) \). So \( \mathcal{A} \models \phi \) and \( \mathcal{B} \not\models \phi \). The duplicator can play \( G^{m,r,k,s}(\mathcal{A}, \mathcal{B}) \) as follows:

Before this play begins, vacuously \( \emptyset \Rightarrow \emptyset \in \text{Part}(\mathcal{A}, \mathcal{B}) \). Let \( f : \emptyset \Rightarrow \emptyset \). Suppose after \( i \) moves \( (0 \leq i < m) \), the players have \( \langle \mathcal{A}, \hat{R} \rangle \) and \( \langle \mathcal{B}, \hat{S} \rangle \) and \( f \) has been extended as \( \text{ment}(\hat{R}) \Rightarrow \text{ment}(\hat{S}) \). In the \((i + 1)\)-th move, w.l.o.g. the spoiler chooses \( R \subseteq A^{r+1} \).
and \(|R| \leq \log^{k+1}(|A|)\), where \(r_{i+1} \leq r\) and \(k_{i+1} \leq k\). For \(a \in \text{ment}(R) - \text{ment}(\bar{R})\), the duplicator can casually choose \(b \notin \text{ment}(\bar{S})\) and extend \(f\) with \(f(a) = b\). Since
\[
|\text{ment}(\bar{S})| \leq m \cdot r \cdot \log^{k}(|B|)
\]
which is much smaller than \(|B|\), there are enough “unmentioned” \(b\)'s to choose to make \(f\) a partial isomorphism. Let
\[S = f(R) = \{(f(t_1), f(t_2), \ldots, f(t_{r+1})) \mid (t_1, t_2, \ldots, t_{r+1}) \in R\}\]
So the duplicator chooses \(S\). The structures are expanded as \(\langle \mathcal{A}, \bar{R} \rangle\) and \(\langle \mathcal{B}, \bar{S} \rangle\)
After \(m\) moves, \(\mathcal{A}\) and \(\mathcal{B}\) are expanded as \(\langle \mathcal{A}, R_1, \ldots, R_m \rangle\) and \(\langle \mathcal{B}, S_1, \ldots, S_m \rangle\) which we still denote by \(\langle \mathcal{A}, \bar{R} \rangle\) and \(\langle \mathcal{B}, \bar{S} \rangle\) for short. Consider the substructures
\[
\langle \text{ment}(\bar{R}), R \rangle \simeq \langle \text{ment}(\bar{S}), S \rangle
\]
The other elements which aren’t in the substructures are all isolated nodes. One can easily check that the duplicator wins \(\text{PG}^m(\mathcal{A}, \bar{R}, R, \mathcal{B}, \bar{S})\).
So the duplicator wins \(\text{G}^{m,r,k,s}(\mathcal{A}, \mathcal{B})\). By proposition 4.3, \(\mathcal{A} \equiv \mathcal{L}^{m,r,k,s} \mathcal{B} \). That is a contradiction.
So \(\text{EVEN}\) is not definable in \(\mathcal{L}\).

Since \(\exists \log \omega \text{IFP}\) is a sublogic of \(\mathcal{L}\), \(\text{EVEN}\) is not definable in \(\exists \log \omega \text{IFP}\), either. Hence \(\exists \log \omega \text{IFP}\) does not capture \(\beta P\) (on all finite structures).

5 Furthur Discussion

Readers might have noticed that the results can be extended onto other complexity classes. For example the existential and universal log-quantifiers can alternate several times in the formula so as to capture a corresponding limited alternation class. Furthermore, not only log-quantifiers, we can also consider other second-order quantifier with a bound of cardinality. Let \(f\) be a sublinear function on \(\mathbb{N}\). One can easily prove on ordered structures a logic \(\exists^f \text{IFP}\) can capture \(\beta (f(n) \cdot \log(n), P)\), where the parameter “\(\log(n)\)” seems unavoidable. However none of the above can capture the corresponding complexity classes without a linear order. The proofs could be analogous to our theorem 4.4.

We are not sure
- on what natural class of graphs, \(\exists \log \omega \text{IFP}\) can capture \(\beta P\) while IFP cannot capture \(P\);
- whether there is a problem in \(P\) which \(\exists \log \omega \text{IFP}\) can define while IFP cannot.

These questions could be interesting.

References

[BL83] László Babai and Eugene M Luks. Canonical labeling of graphs. In Proceedings of the fifteenth annual ACM symposium on Theory of computing, pages 171–183. ACM, 1983.
[Bod90] Hans L Bodlaender. Polynomial algorithms for graph isomorphism and chromatic index on partial k-trees. *Journal of Algorithms*, 11(4):631–643, 1990.

[CC97] Liming Cai and Jianer Chen. On the amount of nondeterminism and the power of verifying. *SIAM Journal on Computing*, 26(3):733–750, 1997.

[EF05] Heinz-Dieter Ebbinghaus and Jörg Flum. *Finite model theory*. Springer Science & Business Media, 2005.

[Fag74] Ronald Fagin. Generalized first-order spectra and polynomial-time recognizable sets. 1974.

[GM99] Martin Grohe and Julian Mariño. Definability and descriptive complexity on databases of bounded tree-width. In *International Conference on Database Theory*, pages 70–82. Springer, 1999.

[GN19] Martin Grohe and Daniel Neuen. Canonisation and definability for graphs of bounded rank width. *arXiv preprint arXiv:1901.10330*, 2019.

[Gro98] Martin Grohe. Fixed-point logics on planar graphs. In *Logic in Computer Science, 1998. Proceedings. Thirteenth Annual IEEE Symposium on*, pages 6–15. IEEE, 1998.

[Gro08] Martin Grohe. The quest for a logic capturing ptime. In *Logic in Computer Science, 2008. LICS’08. 23rd Annual IEEE Symposium on*, pages 267–271. IEEE, 2008.

[Gro11] Martin Grohe. From polynomial time queries to graph structure theory. *Communications of the ACM*, 54(6):104–112, 2011.

[IL90] Neil Immerman and Eric Lander. Describing graphs: A first-order approach to graph canonization. *Complexity theory retrospective*, 1, 1990.

[Imm82] Neil Immerman. Relational queries computable in polynomial time. In *Proceedings of the fourteenth annual ACM symposium on Theory of computing*, pages 147–152. ACM, 1982.

[KF84] Chandra Kintala and P Fischer. Refining nondeterminism in relativized complexity classes. *SIAM Journal on Computing*, 13(329-337):4, 1984.

[Köb06] Johannes Köbler. On graph isomorphism for restricted graph classes. In *Conference on Computability in Europe*, pages 241–256. Springer, 2006.

[Lib13] Leonid Libkin. *Elements of finite model theory*. Springer Science & Business Media, 2013.

[Var82] Moshe Y Vardi. The complexity of relational query languages. In *Proceedings of the fourteenth annual ACM symposium on Theory of computing*, pages 137–146. ACM, 1982.