Studies in Logic, Vol. 13, No. 3 (2020): 01–18
PII: 1674-3202(2020)-03-00001-18

A Logic that Captures βP on Ordered Structures*

Kexu Wang Xishun Zhao

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 $\exists^{\log^{\omega}}$ IFP captures the limited nondeterminism class βP. 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.

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. —*Martin Grohe* ([8])

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

Received 2019-12-05

Kexu Wang Institute of Logic and Cognition, Sun Yat-sen University
Department of Philosophy, Sun Yat-sen University
wangkexuphy@163.com

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

*We would like to thank Dr. Shiguang Feng, who greatly helped the progress of our research.

This research was partially supported by the project of "National Key Research Institutes for the Humanities and Social Sciences" under grant number 19JJD720002.

This article was reported with a title "Logarithmic-Bounded Second-Order Quantifiers and Limited Nondeterminism" in National Conference on Modern Logic 2019, on November 9 in Beijing.

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. in [13]. Then in [3] Cai et al. discussed a more general case, i.e., the "Guess-then-Check" model.

Definition 1.1 ([3])

Let $s : \mathbb{N} \mapsto \mathbb{N}$ and $\mathcal{C}$ be a complexity class. A language L is in the class $GC(s, \mathcal{C})$ if there is a language $L' \in \mathcal{C}$ 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 $u\#v \in L'$.

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

$$\beta_f = GC(f, \mathrm{P})$$

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

$$\beta\mathrm{P} = \bigcup_{k\in\mathbb{N}} \beta_k$$

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 ϕ, any relation variable X and any structure $\mathscr{A}$,

$$\mathscr{A} \vDash \exists^f X\phi \Longleftrightarrow \text{there is a subset } S \subseteq A^{arity(X)} \text{ with } |S| \leq f(|A|),$$
$$\text{such that } \mathscr{A} \vDash \phi[\frac{X}{S}]$$

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 $\exists^{\log^\omega}$IFP is obtained by extending the *inflationary fixed-point logic* IFP with all the log-quantifiers. The main theorem will show that $\exists^{\log^\omega}$IFP captures βP 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 ([5]) showed that NP is captured by the existential second-order logic Σ^1_1, which consists of formulas in the form

$$\exists X_1 \ldots \exists X_m \phi$$

where ϕ 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, PH, is captured by a layer of the

second-order logic. ([4]) The fundamental result of capturing P is Immerman-Vardi's theorem. ([11, 16]) 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 ([14]), planar graphs ([14]), graphs of bounded treewidth ([2]), graphs of bounded degree ([1]). Researchers are also interested in using logics to define a canonization. There are IFP-definable canonizations on cycles ([4]), grids ([4]) or 3-connected planar graphs ([6]). 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 ([12]), planar graphs ([6]), graphs of bounded treewidth ([9]), graphs of bounded rank width ([10]).

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-Fraïssé game and prove $\exists^{\log^\omega}$IFP fails to capture βP 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* τ is a finite class of relation symbols. For convenience we do not talk about constant symbols and function symbols. $\mathscr{L}[\tau]$ is the formulas of logic $\mathscr{L}$ formed with symbols in τ. A τ-*structure* (or structure over τ) $\mathscr{B}$ explains the symbols in τ on a domain B. In this paper we only consider *finite* structures, i.e., whose domain is a finite set. STRUC$[\tau]$ is the class of all τ-structures. A *graph* is a structure over signature $\{E\}$ whose domain V is a set of vertices. STRUC$[\tau]_<$ is the class of all ordered τ-structures (there is a built-in linear order of whose domain). STRING is the class of all strings. Let $\tau_{\text{str}} = \{<, P_0, P_1, P_\#, P_\langle, P_\rangle\}$. A string u is a structure over τ_{str}, i.e.,

$$u = (U, <, P_0^u, P_1^u, P_\#^u, P_\langle^u, P_\rangle^u)$$

where

- $U = \{0, 1, \ldots, |u| - 1\}$
- $<$ is the natural linear order of U
- $P_0^u i \iff$ the i-th bit of u is 0

- $P_1^u i \Longleftrightarrow$ the i-th bit of u is 1
- $P_{\#}^u i \Longleftrightarrow$ the i-th bit of u is #
- $P_{\langle}^u i \Longleftrightarrow$ the i-th bit of u is $\langle$
- $P_{\rangle}^u i \Longleftrightarrow$ the i-th bit of u is $\rangle$

"#" is used to separate two concatenated strings, for instance, "$u\#v$". "$\langle$" and "$\rangle$" 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 $\mathcal{Q}$ on τ is a class of structures over the same signature τ, and closed under isomorphism, i.e., for any $\mathscr{A}, \mathscr{B} \in \text{STRUC}[\tau]$, if $\mathscr{A} \simeq \mathscr{B}$, then,

$$\mathscr{A} \in \mathcal{Q} \Longleftrightarrow \mathscr{B} \in \mathcal{Q}$$

For example, *languages* (classes of strings) are Boolean queries on τ_{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 $\phi(x, X)$, "$\phi[\frac{x}{a}, \frac{X}{R}]$" means the value a (resp. R) is substituted into x (resp. X) if x (resp. X) is free. We abuse the notation $|\cdot|$. If u is a string, $|u|$ is its length. If A is a set, $|A|$ is its cardinal. 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 *arity*$(R_i) = r_i$ $(1 \leq i \leq m)$, any $\mathscr{A} \in \text{STRUC}[\tau]_<$ with domain $A = \{a_0, \ldots, a_{|A|-1}\}$

1. $\mathit{enc}(\mathscr{A}) = \langle \mathit{enc}(A)\mathit{enc}(R_1^{\mathscr{A}}) \ldots \mathit{enc}(R_m^{\mathscr{A}}) \rangle$
2. $\mathit{enc}(A) = \langle \mathit{enc}(a_0) \ldots \mathit{enc}(a_{|A|-1}) \rangle$
3. For any $i \in \{1, \ldots, m\}$, suppose $\vec{a}_1, \ldots, \vec{a}_{|R_i^{\mathscr{A}}|}$ are all r_i-tuples in $R_i^{\mathscr{A}}$,

$$\mathit{enc}(R_i^{\mathscr{A}}) = \langle \mathit{enc}(\vec{a}_1), \ldots, \mathit{enc}(\vec{a}_{|R_i^{\mathscr{A}}|}) \rangle$$

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

$$\mathit{enc}(\vec{t}) = \langle \mathit{enc}(t_1) \ldots \mathit{enc}(t_s) \rangle$$

5. Suppose a is the j-th element in A, $0 \leq j < |A|$,

$$\mathit{enc}(a) = \langle \text{"the } \log|A| \text{-long binary expression of } j\text{"} \rangle$$

Note that

$$|enc(\mathscr{A})| = \log|A| \cdot O(\sum_{1\leq i\leq m}(|R_i^{\mathscr{A}}| \cdot r_i))$$

and for $i \in \{1, \ldots, m\}$,

$$|enc(R_i^{\mathscr{A}})| = \log|A| \cdot O(|R_i^{\mathscr{A}}| \cdot r_i)$$

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

2.2 Logic characterization of complexity

Definition 2.2 ([7])

A logic $\mathscr{L}$ *captures* a complexity class $\mathcal{C}$ on a class $\mathcal{K}$ of structures, if the following conditions are satisfied,

1. $\mathscr{L}[\tau]$ is decidable, for any signature τ.
2. There is an effective procedure to associate with each $\mathscr{L}$-sentence ϕ a $\mathcal{C}$-bounded Turing machine M, such that, for any $\mathscr{A} \in \mathcal{K}$, M can decide whether
$$\mathscr{A} \vDash \phi$$
3. For any Boolean query $\mathcal{Q}$ in $\mathcal{C}$, there is an $\mathscr{L}$-sentence ϕ such that for any $\mathscr{A} \in \mathcal{K}$,
$$\mathscr{A} \vDash \phi \text{ iff } \mathscr{A} \in \mathcal{Q}$$

(We assume that $\mathcal{K}$ is closed under isomorphism.)

If $\mathcal{K}$ is the class of all structures, we simply say $\mathscr{L}$ captures $\mathcal{C}$.

There are two most classical theorems in descriptive complexity theory.

Theorem 2.3 (Fagin's Theorem, [5]) Σ_1^1 captures NP.

Theorem 2.4 (Immerman-Vardi Theorem, [11, 16]) IFP captures P on ordered structures.

IFP is gotten by extending the first-order logic FO with the inflationary fixed-point operator. IFP inherits the formation rules of FO besides

- If ψ is a formula, then so is $[\text{IFP}_{\vec{y}\,Y}\psi(\vec{y}, Y)]\vec{t}$, where Y is a relation variable and $|\vec{y}\,| = |\vec{t}\,| = arity(Y)$

$[\text{IFP}_{\vec{y}\,Y}\psi(\vec{y},Y)]$ is the fixed point of the function $f^{Y\vee\psi}$ defined by the fomula $Y\vec{y}\vee\psi(\vec{y},Y)$. This semantics will not be used in this paper, so readers can turn to [4] and [15] for details.

In logic we needn't even study structures over all different signatures. We particularly care about STRING and graphs, which the structures over other signatures can be interpreted to.

Definition 2.5 Let $\mathscr{L}$ be a logic. Let τ,σ be two signatures. $\sigma=\{R_1,R_2,\ldots,R_m\}$, where *arity*$(R_i)=r_i$ $(1\leq i\leq m)$. An *k-ary $\mathscr{L}$-interpretation* from τ to σ is a sieres of $\mathscr{L}[\tau]$-formulas

$$I=(\phi_{\text{uni}}(\vec{x}),\phi_{R_1}(\vec{x}_1,\ldots,\vec{x}_{r_1}),\ldots\phi_{R_m}(\vec{x}_1,\ldots,\vec{x}_{r_m}))$$

where the variables $\vec{x}$, $\vec{x}_1$, ... are k-tuples. For any $\mathscr{A}$ on τ,

$$I(\mathscr{A})=(\phi^{\mathscr{A}}_{\text{uni}}(\_),\phi^{\mathscr{A}}_{R_1}(\_,\ldots,\_),...,\phi^{\mathscr{A}}_{R_m}(\_,\ldots,\_))$$

is a σ-structure, if we consider the k-tuples satisfying $\phi^{\mathscr{A}}_{\text{uni}}(\vec{x})$ as individual elements. (Note that: $\phi^{\mathscr{A}}(\_):=\{\vec{a}\mid\mathscr{A}\vDash\phi[\frac{\vec{x}}{\vec{a}}]\}$)

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 $\mathscr{A}\in\text{STRUC}[\tau]$,

$$\mathscr{A}\in S_1\Longleftrightarrow I(\mathscr{A})\in S_2$$

we say I is an *$\mathscr{L}$-reduction* from S_1 to S_2.

It is not hard to prove for any $\mathscr{A},\mathscr{B}\in\text{STRUC}[\tau]$

$$\mathscr{A}\simeq\mathscr{B}\Longrightarrow I(\mathscr{A})\simeq I(\mathscr{B})$$

Lemma 2.6 For any signature τ, there is an FO-reduction I from $\text{STRUC}[\tau]_<$ to STRING and for any $\mathscr{A},\mathscr{B}$ in $\text{STRUC}[\tau]_<$,

$$\mathscr{A}\simeq\mathscr{B}\Longleftrightarrow I(\mathscr{A})=I(\mathscr{B})$$

Lemma 2.7 Let ϕ be a formula of $\text{IFP}[\sigma]$,

$$I=\langle\phi_{\text{uni}},\phi_{R_1},...,\phi_{R_m}\rangle$$

is an k-ary reduction from $\text{STRUC}[\tau]$ to $\text{STRUC}[\sigma]$. ϕ^I is obtained by

- replacing every variable x occuring in ϕ by a new k-tuple $\vec{x}$ (which consists of all new variables, let's denote it by x^I),
- replacing every relation R_i in ϕ by ϕ_{R_i},
- changing the subformula $\forall x\ldots$ in ϕ to $\forall x^I(\phi_{\text{uni}}(x^I)\rightarrow\ldots)$,

- changing the subformula $\exists x \dots$ in ϕ to $\exists x^I(\phi_{\text{uni}}(x^I) \wedge \dots)$,
- for $[\text{IFP}_{\vec{y}\,Y}\psi(\vec{y}, Y)]$ is in ϕ, where $\vec{y} = y_1 y_2 \dots y_l$ and $\mathit{arity}(Y) = l$, then replacing $[\text{IFP}_{\vec{y}\,Y}\psi(\vec{y}, Y)]$ by

$$[\text{IFP}_{y_1^I y_2^I \dots y_l^I\ Y^I} \bigwedge_{1 \le i \le l} \phi_{\text{uni}}(y_i^I) \wedge \psi^I(y_1^I y_2^I \dots y_l^I, Y^I)]$$

where Y^I is an $l \cdot k$-ary new relation variable.

Then for $\mathscr{A} \in \text{STRUC}[\tau]$,

$$\mathscr{A} \vDash \phi^I \iff I(\mathscr{A}) \vDash \phi$$

These two lemmas tell us STRING and ordered structures are deeply related. $\mathscr{L}$ captures $\mathcal{C}$ on STRING if and only if $\mathscr{L}$ captures $\mathcal{C}$ on ordered structures. 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 βP prepared for our later proofs:

Definition 3.1 A language L is in the class β_k if there is a language $L' \in \text{P}$ together with an integer $c > 0$ such that for any string u, $u \in L$ if and only if $\exists v \in \{0,1\}^{\le c \cdot \log^k(|u|)}$, $u\#v \in L'$. (where $\{0,1\}^{\le c \cdot \log^k(|u|)}$ is all the 0-1 strings of length at most $c \cdot \log^k(|u|)$.)

$\beta\text{P} = \bigcup_{k \in \mathbb{N}} \beta_k$.

Since $\beta_1 = GC(\log, \text{P})$, in fact the "guess" part can be computed in time $2^{c \cdot \log}$, which is a polynomial. Thus we have

$$\text{P} = \beta_1 \subseteq \beta_2 \subseteq \dots \subseteq \beta\text{P} \subseteq \text{NP}$$

3.1 Logarithmic-bounded quantifiers

The log-quantifier $\exists^{\log^k}$ is the second-order quantifier with a bound $\log^k$. As we mentioned,

$$\mathscr{A} \vDash \exists^{\log^k} X\phi \iff \text{there is a subset } S \subseteq A^{\mathit{arity}(X)} \text{ with } |S| \le \log^k(|A|),$$
$$\text{such that } \mathscr{A} \vDash \phi[\frac{X}{S}]$$

It doesn't matter how large $\mathit{arity}(X)$ is. As long as $\mathit{arity}(X)$ is a nonzero natural number, $\exists^{\log^k}$ can be applied. Naturally

$$\forall^{\log^k} X := \neg\exists^{\log^k} X\neg$$

Let $\log^\omega = \{\log^k \mid k > 0\}$. Then $\exists^{\log^\omega} = \{\exists^{\log^k} \mid k > 0\}$

Definition 3.2 An formula of $\exists^{\log^\omega}$IFP is in the form,

$$\exists^{\log^{k_1}} X_1 \exists^{\log^{k_2}} X_2 \ldots \exists^{\log^{k_m}} X_m \psi$$

where $m \geq 0$; $k_1, k_2, \ldots k_m > 0$; ψ is an IFP-formula.

Those formulas without any occurrences of log-quantifiers are *log-quantifier-free*.

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

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

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

3.2 Main theorem

Theorem 3.3 $\exists^{\log^\omega}$IFP captures βP on STRING.

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

But here is a problem: for any v in "$\exists v \in \{0,1\}^{\leq c\cdot\log^k(|u|)}$" 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 *surjection* from $\{S \mid S \subseteq U^2$ and $|S| \leq \log^k(|U|)\}$ to $\{0,1\}^{\leq \log^k(|U|)\cdot(\log(|U|)-1)}$.

And let $\tau_r = \tau_{\text{str}} \cup \{R_1, R_2, \ldots R_r\}$, where $R_1, \ldots R_r$ are binary relation symbols. There is an IFP-reduction I from STRUC$[\tau_r]$ to STRING such that for any $u \in$

STRING and binary relations $R_1^u, \dots R_r^u \in \{S \mid S \subseteq U^2 \text{ and } |S| \leq \log^k(|U|)\}$,

$$I((u, R_1^u, \dots R_r^u)) = u\#J^U(R_1^u)J^U(R_2^u)\dots J^U(R_r^u)$$

Proof (of lemma 3.4)

For any $S \in \{S \mid S \subseteq U^2 \text{ and } |S| \leq \log^k(|U|)\}$, $J^U(S)$ is gotten by doing as follows

1. getting *enc*(S);
2. removing the first element of each tuple of S from *enc*(S);
3. removing the $\log(|U|)$-th bit of each consecutive binary substrings in the encoding;
4. removing the symbols "$\langle$" and "$\rangle$".

For example suppose $\log(|U|) = 3$, and $S = \{(1,3),(1,0),(2,0)\}$, then

$$\textit{enc}(S) = \langle\langle\langle 100\rangle\langle 110\rangle\rangle\langle\langle 100\rangle\langle 000\rangle\rangle\langle\langle 010\rangle\langle 000\rangle\rangle\rangle$$

Then we do

$$\begin{array}{c}
\langle\langle\langle 100\rangle\langle 110\rangle\rangle\langle\langle 100\rangle\langle 000\rangle\rangle\langle\langle 010\rangle\langle 000\rangle\rangle\rangle \\
\Downarrow \\
\langle\langle\langle \cancel{100}\rangle\langle 110\rangle\rangle\langle\langle \cancel{100}\rangle\langle 000\rangle\rangle\langle\langle \cancel{010}\rangle\langle 000\rangle\rangle\rangle \\
\Downarrow \\
\langle\langle\langle 11\emptyset\rangle\rangle\langle\langle 00\emptyset\rangle\rangle\langle\langle 00\emptyset\rangle\rangle\rangle \\
\Downarrow \\
110000
\end{array}$$

So $J^U(S) = 110000$ in this example.

It is easy to verify that J^U is a surjection.

Now we construct the IFP-reduction I. With the help of the linear order $<^u$, we can construct IFP-formula 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 turn to [15, p. 96].)

Let $\vec{x} = x_1x_2x_3x_4x_5yz_1\dots z_{\log(r)}$. It's an $(\log(r)+6)$-ary tuple of variables. Now we define:

$$\begin{aligned}
&\phi_< \text{ is the lexicographic order of } (\log(r)+6)\text{-ary tuples} \\
&\phi_{P_0}(\vec{x}) :=(x_1 = x_2 = 0 \wedge P_0(x_4)) \vee (x_1 = x_2 = 1 \wedge x_4 = 0) \\
&\phi_{P_1}(\vec{x}) :=(x_1 = x_2 = 0 \wedge P_1(x_4)) \vee (x_1 = x_2 = 1 \wedge x_4 = 1) \\
&\phi_{P_\#}(\vec{x}) :=(x_1 = x_2 = 0 \wedge P_\#(x_4)) \vee (x_1 = 0 \wedge x_2 = 1) \\
&\phi_{P_\langle}(\vec{x}) :=(x_1 = x_2 = 0 \wedge P_\langle(x_4)) \\
&\phi_{P_\rangle}(\vec{x}) :=(x_1 = x_2 = 0 \wedge P_\rangle(x_4))
\end{aligned}$$

$$\begin{aligned}\phi_{\text{uni}}(\vec{x}) := \quad & (x_1 = x_2 = 0 \\ & \wedge y = z_1 = \ldots = z_{\log(r)} = x_3 = x_5 = 0) \\ & \vee (x_1 = 0 \wedge x_2 = 1 \\ & \wedge y = z_1 = \ldots = z_{\log(r)} = x_3 = x_4 = x_5 = 0) \\ & \vee (x_1 = x_2 = 1 \\ & \wedge (\bigvee_{1 \le i \le r} (R_i x_5 y \\ & \qquad \wedge \text{“}z_1 z_2 \ldots z_{\log(r)} \text{ is the binary expression of } i\text{”} \\ & \qquad \wedge x_3 < \log(|U|) - 1 \\ & \qquad \wedge x_4 = 1 \leftrightarrow \text{BIT}(y, x_3) \\ & \qquad \wedge x_4 = 0 \leftrightarrow \neg\text{BIT}(y, x_3))))\end{aligned}$$

So $I = (\phi_{\text{uni}}, \phi_<, \phi_{P_0}, \phi_{P_1}, \phi_{P_\#}, \phi_{P_\langle}, \phi_{P_\rangle})$ is an IFP-reduction that we want. □

Proof (of theorem 3.3)

By definition 2.2, our proof consists of three parts. Let $k > 0$.

Firstly. $\exists^{\log^\omega}\text{IFP}[\tau]$ is decidable, for any signature τ.

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

$$\begin{aligned}u \vDash \phi \Longleftrightarrow & \text{there are } S_1 \subseteq U^{arity(X_1)}, \ \ldots\ , S_m \subseteq U^{arity(X_m)} \\ & \text{and } |S_1| \le \log^{k_1}|u|, \ \ldots\ , |S_m| \le \log^{k_m}|u| \\ & \text{such that } u \vDash \psi[\frac{X_1}{S_1}, \ldots, \frac{X_m}{S_m}]\end{aligned}$$

By theorem 2.4, there is a P-bounded Turing machine M_ψ that can verify whether

$$\mathscr{A} \vDash \psi[\frac{X_1}{R_1}, \ldots, \frac{X_m}{R_m}]$$

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

In order to guess and store the values of $X_1, \ldots, X_m$, by definition 2.1, M_ϕ will need

$$O(\log^{k_1+1}|u| \cdot arity(X_1) + \ldots + \log^{k_m+1}|u| \cdot arity(X_m))$$

nondeterministic bits, or simply, $O(\log^{k+1}|u|)$ nondeterministic bits in total.

then M_ϕ returns TRUE if there are $S_1, \ldots, S_m$ with $|S_1| \le \log^{k_1}|u|, \ \ldots\ , |S_m| \le \log^{k_m}|u|$, such that M_ψ accepts $\langle u, S_1, \ldots, S_m\rangle$. Otherwise M_ϕ returns FALSE.

So M_ϕ is a β_{k+1}-bounded machine that we want.

Thirdly. Suppose L is a language in β_{k+1}. By definition 3.1, there is a function $f(n) = O(\log^{k+1}(n))$ and a P-bounded Turing machine M, such that for any $u \in$ STRING,

$$u \in L \Longleftrightarrow \exists v \in \{0,1\}^{\leq f(|u|)} \text{ M accepts } u\#v$$

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

$$\text{M}' \text{ accepts } u\#z \Longleftrightarrow \text{M accepts } u\#v \text{ and } v \text{ is the leftmost } f(|u|) \text{ bits of } z.$$

(M$'$ need not compute the function f, so it does not matter whether f is computable or not.) By theorem 2.4, there is an IFP[τ_{str}]-sentence $\phi_{\text{M}'}$ such that for any $v \in$ STRING,

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

By lemma 3.4, there is a $(\log(r)+6)$-ary IFP reduction from STRUC[$\tau_{\text{str}} \cup \{R_1, \ldots R_r\}$] to STRING, $I = \langle \phi_{\text{uni}}, \phi_<, \phi_{P_0}, \phi_{P_1}, \phi_{P_\#}, \phi_{P_\langle}, \phi_{P_\rangle} \rangle$. With the help of lemma 2.7, let

$$\psi := \psi^I_{\text{M}'}$$

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

$$\phi = \exists^{\log^k} R_1, \ldots \exists^{\log^k} R_r \psi$$

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

$$u \in L \Longleftrightarrow u \vDash \phi$$ □

In the above proof, we can see only binary relation symbols $R_1, \ldots R_r$ are bounded by the log-quantifiers. So we obtain

Corollary 3.5 On ordered structures, every formula of $\exists^{\log^k}$IFP is equivalent to a formula of $\exists^{\log^k}$IFP whose bounded relation variables are binary.

4 The Expressive Power

IFP fails on a very important P-decidable Boolean query, EVEN. ([4]) For any graph $\mathcal{G}$, $\mathcal{G} \in$ EVEN if and only if domain $|V|$ is even. There is *no* sentence ϕ of IFP[$\{E\}$] such that

$$\mathcal{G} \in \text{EVEN} \Longleftrightarrow \mathcal{G} \vDash \phi$$

(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 $\exists^{\log^\omega}$IFP.

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

$$\mathscr{L}^\omega_{\infty\omega} = \bigcup_{s\in\mathbb{N}} \mathscr{L}^s_{\infty\omega}$$

For the details readers can turn to [4, ch. 3]. For every single IFP-formula, we can always construct an equivalent $\mathscr{L}^s_{\infty\omega}$-formula for some s. So IFP is a sublogic of $\mathscr{L}^\omega_{\infty\omega}$. Now we define a new logic $\mathcal{L}$ (*Beware*! It is not $\mathscr{L}$!) as follows: for any formula ϕ

- $\phi \in \mathcal{L}$ if $\phi \in \mathscr{L}^\omega_{\infty\omega}$
- $\exists^{\log^k} X\phi \in \mathcal{L}$ if $\phi \in \mathcal{L}$, where $k > 0$ and X is some relation variable.
- $\forall^{\log^k} X\phi \in \mathcal{L}$ if $\phi \in \mathcal{L}$, where $k > 0$ and X is some relation variable.
- $\psi \wedge \chi \in \mathcal{L}$ if $\psi \in \mathcal{L}$ and $\chi \in \mathcal{L}$
- $\psi \vee \chi \in \mathcal{L}$ if $\psi \in \mathcal{L}$ and $\chi \in \mathcal{L}$

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

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

Definition 4.2 $\mathscr{L}$ 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-Fraïssé game for $\mathscr{L}$, if for any τ, any $\mathscr{A}$ and $\mathscr{B} \in$ STRUCT$[\tau]$, the following are equivalent,

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

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

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 $\vec{a}$ consists of elements in $\bar{a}$, we simply say $\vec{a}$ is *from* $\bar{a}$. We say $\bar{a} \mapsto \bar{b} \in$ Part($\mathscr{A}, \bar{P}, \mathscr{B}, \bar{Q}$), i.e., $\bar{a} \mapsto \bar{b}$ is a *partial isomorphism* from $\langle \mathscr{A}, \bar{R}\rangle$ to $\langle \mathscr{B}, \bar{S}\rangle$, where $\bar{R} = \{R_1, \dots R_l\}$ and $\bar{S} = \{S_1, \dots S_l\}$, 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 $\vec{t}$ from $\bar{a}$,
$$\vec{t} \in P^{\mathscr{A}} \iff f(\vec{t}) \in P^{\mathscr{B}}$$
3. for $1 \leq i \leq l$, and any tuple $\vec{t}$ from $\bar{a}$,
$$\vec{t} \in R_i \iff f(\vec{t}) \in S_i$$

In the expansions, actually $\bar{R}$, $\bar{S}$ act as new relations.

The Ehrenfeucht-Fraïssé game for $\mathscr{L}^s_{\infty\omega}$ is the pebble game with s pairs of pebbles, denoted by PG^s. In a play of PG$^s(\mathscr{A}, \mathscr{B})$, there are s (or less) vertices in each of $\mathscr{A}$ and $\mathscr{B}$ covered by pebbles. In each move, each player can do nothing, move one pebble or add 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$^s(\mathscr{A}, \mathscr{B})$. For the details readers can turn to [4, ch. 3].

Now let $\mathcal{L}^{m,r,k,s}$ be the sublogic of $\mathcal{L}$ such that for any ϕ in it,

- $lqr(\phi) \leq m$,
- $mva(\phi) \leq r$,
- $height(\phi) \leq k$,
- at most s element variables occur in ϕ.

Let's design a game G^{m,r,k,s} for $\mathcal{L}^{m,r,k,s}$. As $\mathcal{L}^{m,r,k,s}$ is extended from $\mathscr{L}^s_{\infty\omega}$ 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 $\mathscr{A}$ or $\mathscr{B}$. (W.l.o.g. we assume the spoiler chooses $\mathscr{A}$. Otherwise $\mathscr{A}$ and $\mathscr{B}$ 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$^{m,r,k,s}(\mathscr{A}, \mathscr{B})$, the spoiler first chooses an arbitrary $m' \leq m$ and they play m' relation moves and then the two structures are expanded as $\langle \mathscr{A}, R_1, \ldots, R_{m'} \rangle$ and $\langle \mathscr{B}, S_1, \ldots S_{m'} \rangle$. Then they play PG$^s(\langle \mathscr{A}, R_1, \ldots R_{m'} \rangle, \langle \mathscr{B}, S_1, \ldots S_{m'} \rangle)$. Once this pebble game begins, no more relation moves are allowed. If the duplicator wins PG$^s(\langle \mathscr{A}, R_1, \ldots R_{m'} \rangle, \langle \mathscr{B}, S_1, \ldots S_{m'} \rangle)$, she wins the play.

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

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

Proof Let $\mathscr{A}$ and $\mathscr{B}$ be two structures over a given signature τ.

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

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

then inductively

$$\phi^{m+1,r,k,s}_{\mathscr{A},\bar{R}}(\bar{X}) = \bigwedge_{i\leq r}\bigwedge_{j\leq k}[(\bigwedge_{R\subseteq A^i, |R|\leq log^j(|A|)} \exists^{\log^j} X \phi^{m,r,k,s}_{\mathscr{A},\bar{R}R}(\bar{X}X)) \wedge (\forall^{\log^j} X \bigvee_{R\subseteq A^i, |R|\leq log^j(|A|)} \phi^{m,r,k,s}_{\mathscr{A},\bar{R}R}(\bar{X}X))]$$

When $\bar{R} = \varnothing$, we simply write $\phi_{\mathscr{A}}^{m,r,k,s}$, which is a sentence of $\mathcal{L}^{m,r,k,s}$.

Suppose $\mathscr{A} \equiv^{\mathcal{L}^{m,r,k,s}} \mathscr{B}$, then $\mathscr{B} \vDash \phi_{\mathscr{A}}^{m,r,k,s}$. The isotype indicates a winning strategy for the duplicator. After m moves if the two structures are expanded as $\langle \mathscr{A}, \bar{R} \rangle$ and $\langle \mathscr{B}, \bar{S} \rangle$,

$$\mathscr{B} \vDash \phi_{\mathscr{A},\bar{R}}^{0,r,k,s}[\frac{\bar{X}}{\bar{S}}]$$

This means $\langle \mathscr{A}, \bar{R} \rangle$ and $\langle \mathscr{B}, \bar{S} \rangle$ satisfy the same $\mathscr{L}_{\infty\omega}^{s}$-formulas. Therefore the duplicator can win $\mathrm{PG}^s(\langle \mathscr{A}, \bar{R} \rangle, \langle \mathscr{B}, \bar{S} \rangle)$. Then she wins $\mathrm{G}^{m,r,k,s}(\mathscr{A}, \mathscr{B})$.

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

$$\phi = Q_1^{\log^{k_1}} X_1 \dots Q_m^{\log^{k_m}} X_m \psi$$

where ψ is an $\mathscr{L}_{\infty\omega}^{s}$-sentence and $k_1, \dots k_m \leq k$ and $Q_1, \dots Q_m \in \{\exists, \forall\}$. Then

- $\mathscr{A} \vDash Q_1^{\log^{k_1}} X_1 \dots Q_m^{\log^{k_m}} X_m \psi$
- $\mathscr{B} \vDash \hat{Q}_1^{\log^{k_1}} X_1 \dots \hat{Q}_m^{\log^{k_m}} X_m \neg\psi$

(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 $\mathscr{A}$ and the relation $R_i \subseteq A^{\mathit{arity}(X_i)}$; otherwise she should choose $\mathscr{B}$ and the relation $S_i \subseteq B^{\mathit{arity}(X_i)}$. After m relation moves, the structures have been expanded as $\langle \mathscr{A}, R_1, \dots R_m \rangle$ and $\langle \mathscr{B}, S_1, \dots S_m \rangle$.

- $\langle \mathscr{A}, R_1, \dots R_m \rangle \vDash \psi[\frac{X_1}{R_1}, \dots \frac{X_m}{R_m}]$
- $\langle \mathscr{B}, S_1, \dots S_m \rangle \vDash \neg\psi[\frac{X_1}{S_1}, \dots \frac{X_m}{S_m}]$

The duplicator cannot win $\mathrm{PG}^s(\langle \mathscr{A}, \bar{R} \rangle, \langle \mathscr{B}, \bar{S} \rangle)$. So she cannot win $\mathrm{G}^{m,r,k,s}(\mathscr{A}, \mathscr{B})$. □

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

$$|\mathit{ment}(R)| \leq \mathit{arity}(R) \cdot \log^k(|A|)$$

and we denote $\mathit{ment}(\bar{R}) = \bigcup_{R \in \bar{R}} \mathit{ment}(R)$

Theorem 4.4 EVEN is not definable in $\mathcal{L}$.

Proof If EVEN is defined by a sentence ϕ of $\mathcal{L}[\{E\}]$, ϕ should also work on empty graphs, namely on the graphs that have no edges. Now we assume $E = \varnothing$ in order to get a contradiction. There are $m \geq 0$ and $r, k, s > 0$ such that $\phi \in \mathcal{L}^{m,r,k,s}[\{E\}]$.

Let $\mathscr{A}$ and $\mathscr{B}$ 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 $\mathscr{A} \vDash \phi$ and $\mathscr{B} \nvDash \phi$. The duplicator can play $\mathrm{G}^{m,r,k,s}(\mathscr{A}, \mathscr{B})$ as follows:

Before this play begins, vacuously $\varnothing \mapsto \varnothing \in \mathrm{Part}(\mathscr{A}, \mathscr{B})$. Let $f : \varnothing \mapsto \varnothing$. Suppose after i moves ($0 \leq i < m$), the players have $\langle \mathscr{A}, \bar{R}\rangle$ and $\langle \mathscr{B}, \bar{S}\rangle$ and f has been extended as $\mathit{ment}(\bar{R}) \mapsto \mathit{ment}(\bar{S})$. In the $(i+1)$-th move, w.l.o.g. the spoiler chooses $R \subseteq A^{r_{i+1}}$ and $|R| \leq \log^{k_{i+1}}(|A|)$, where $r_{i+1} \leq r$ and $k_{i+1} \leq k$. For $a \in \mathit{ment}(R) - \mathit{ment}(\bar{R})$, the duplicator can casually choose $b \notin \mathit{ment}(\bar{S})$ and extend f with $f(a) = b$. Since

$$|\mathit{ment}(\bar{S})| \leq m\cdot r\cdot \log^k(|B|)$$

which is much smaller than $|B|$, there are enough "unmentioned" b's to choose to make f a partial isomorphism. Let

$$S = f(R) = \{(f(t_1), f(t_2), \ldots f(t_{r_{i+1}})) \mid (t_1, t_2, \ldots t_{r_{i+1}}) \in R\}$$

So the duplicator chooses S. the structures are expanded as $\langle \mathscr{A}, \bar{R}R\rangle$ and $\langle \mathscr{B}, \bar{S}S\rangle$

After m moves, $\mathscr{A}$ and $\mathscr{B}$ are expanded as $\langle \mathscr{A}, R_1, \ldots R_m\rangle$ and $\langle \mathscr{B}, S_1, \ldots S_m\rangle$ which we still denote by $\langle \mathscr{A}, \bar{R}\rangle$ and $\langle \mathscr{B}, \bar{S}\rangle$ for short. Consider the substructures

$$\langle \mathit{ment}(\bar{R}), \bar{R}\rangle \simeq \langle \mathit{ment}(\bar{S}), \bar{S}\rangle$$

The other elements which aren't in the substructures are all isolated nodes. One can easily check that the duplicator wins $\mathrm{PG}^s(\mathscr{A}, \bar{R}, \mathscr{B}, \bar{S})$.

So the duplicator wins $\mathrm{G}^{m,r,k,s}(\mathscr{A}, \mathscr{B})$. By proposition 4.3, $\mathscr{A} \equiv^{\mathcal{L}^{m,r,k,s}} \mathscr{B}$. That is a contradiction.

So EVEN is not definable in $\mathcal{L}$. □

Since $\exists^{\mathrm{log}^\omega}$IFP is a sublogic of $\mathcal{L}$, EVEN is not definable in $\exists^{\mathrm{log}^\omega}$IFP, either. Hence $\exists^{\mathrm{log}^\omega}$IFP does not capture βP (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 "$\exists^f$IFP" can capture $\beta_{(f\cdot\log)}$, i.e., the complexity

class $GC(f(n) \cdot \log(n), \mathrm{P})$, 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, $\exists^{\log^\omega}$IFP can capture βP while IFP cannot capture P.
- whether there is a problem in P which $\exists^{\log^\omega}$IFP can define while IFP cannot.

These questions could be interesting.

References

[1] L. Babai and E. M. Luks, 1983, "Canonical labeling of graphs", *15th Annual ACM Symposium on Theory of Computing*, pp. 171–183, New York: ACM Press.

[2] H. L. Bodlaender, 1990, "Polynomial algorithms for graph isomorphism and chromatic index on partial k-trees", *Journal of Algorithms*, **11(4)**: 631–643.

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

[4] H. D. Ebbinghaus and J. Flum, 2005, *Finite Model Theory*, Berlin: Springer.

[5] R. Fagin, 1974, "Generalized first-order spectra and polynomial-time recognizable sets", *Complexity of Computation*, **vol. 7**, pp. 43–73, Providence, RI: American Mathematical Society.

[6] M. Grohe, 1998, "Fixed-point logics on planar graphs", *13th Annual IEEE Symposium on Logic in Computer Science*, pp. 6–15, Washington: IEEE Computer Society.

[7] M. Grohe, 2008, "The quest for a logic capturing ptime", *23rd Annual IEEE Symposium on Logic in Computer Science*, pp. 267–271, Washington: IEEE Computer Society.

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

[9] M. Grohe and J. Mariño, 1999, "Definability and descriptive complexity on databases of bounded tree-width", *International Conference on Database Theory*, pp. 70–82, Berlin: Springer.

[10] M. Grohe and D. Neuen, 2019, "Canonisation and definability for graphs of bounded rank width", *34th Annual ACM/IEEE Symposium on Logic in Computer Science*, pp. 1–13, Washington: IEEE Computer Society.

[11] N. Immerman, 1982, "Relational queries computable in polynomial time", *14th Annual ACM Symposium on Theory of Computing*, pp. 147–152, New York: ACM Press.

[12] N. Immerman and E. Lander, 1990, "Describing graphs: A first-order approach to graph canonization", pp. 59–81, New York: Springer.

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

[14] J. Köbler, 2006, "On graph isomorphism for restricted graph classes", *Conference on Computability in Europe*, pp. 241–256, Berlin: Springer.

[15] L. Libkin, 2013, *Elements of finite model theory*, Berlin: Springer.

[16] M. Y. Vardi, 1982, "The complexity of relational query languages", *14th Annual ACM Symposium on Theory of Computing*, pp. 137–146, New York: ACM Press.

在有序结构上刻画 βP 的逻辑

王克诩　　赵希顺

摘　要

我们在膨胀不动点逻辑 IFP 的基础上，加入一种带有（多重）对数上界的新二阶量词，并且证明了，在有序结构上我们的新逻辑 $\exists^{\log^{\omega}}$IFP 刻画受限非确定性复杂类 βP。为了研究该逻辑的表达力，我们也设计了一种新的 Ehrenfeucht-Fraïssé 博弈，并说明在最一般的情况下，也就是在全体有穷模型之上，该逻辑对 βP 的刻画并不成立。

王克诩　中山大学逻辑与认知研究所
中山大学哲学系
wangkexuphy@163.com
赵希顺　中山大学逻辑与认知研究所
中山大学哲学系
hsszxs@mail.sysu.edu.cn