



University of Pennsylvania  
**ScholarlyCommons**

---

Technical Reports (CIS)

Department of Computer & Information Science

---

November 1991

## Infinitary Logic and Inductive Definability Over Finite Structures

Anuj Dawar  
*University of Pennsylvania*

Steven Lindell  
*University of Pennsylvania*

Scott Weinstein  
*University of Pennsylvania*

Follow this and additional works at: [https://repository.upenn.edu/cis\\_reports](https://repository.upenn.edu/cis_reports)

---

### Recommended Citation

Anuj Dawar, Steven Lindell, and Scott Weinstein, "Infinitary Logic and Inductive Definability Over Finite Structures", . November 1991.

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-91-97.

This paper is posted at ScholarlyCommons. [https://repository.upenn.edu/cis\\_reports/365](https://repository.upenn.edu/cis_reports/365)  
For more information, please contact [repository@pobox.upenn.edu](mailto:repository@pobox.upenn.edu).

---

## Infinitary Logic and Inductive Definability Over Finite Structures

### Abstract

The extensions of first-order logic with a least fixed point operators (FO + LFP) and with a partial fixed point operator (FO + PFP) are known to capture the complexity classes P and PSPACE respectively in the presence of an ordering relation over finite structures. Recently, Abiteboul and Vianu [AV91b] investigated the relation of these two logics in the absence of an ordering, using a machine model of generic computation. In particular, they showed that the two languages have equivalent expressive power if and only if  $P = PSPACE$ . These languages can also be seen as fragments of an infinitary logic where each formula has a bounded number of variables,  $L^{\omega}_{\infty\omega}$  (see, for instance, [KV90]). We present a treatment of the results in [AV91b] from this point of view. In particular, we show that we can write a formula of FO + LFP and P from ordered structures to classes of structures where every element is definable. We also settle a conjecture mentioned in [AV91b] by showing that FO + LFP is properly contained in the polynomial time computable fragment of  $L^{\omega}_{\infty\omega}$ , raising the question of whether the latter fragment is a recursively enumerable class.

### Comments

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-91-97.

**Infinitary Logic and Inductive Definability  
Over Finite Structures**

**MS-CIS-91-97  
LOGIC & COMPUTATION 44**

**Anuj Dawar  
Steven Lindell  
Scott Weinstein**

**Department of Computer and Information Science  
School of Engineering and Applied Science  
University of Pennsylvania  
Philadelphia, PA 19104-6389**

**December 1991**

# Infinitary Logic and Inductive Definability over Finite Structures

Anuj Dawar<sup>1</sup>

Department of Computer and Information Science  
University of Pennsylvania  
Philadelphia, PA 19104.

Steven Lindell<sup>2</sup>

Department of Mathematics  
Haverford College  
Haverford, PA 19041.

Scott Weinstein<sup>1</sup>

Department of Philosophy  
University of Pennsylvania  
Philadelphia, PA 19104.

November 27, 1991

<sup>1</sup>Supported in part by ONR grant N00014-89-J-1725.

<sup>2</sup>Supported in part by NSF grant CCR-9003356.

## Abstract

The extensions of first-order logic with a least fixed point operator (FO + LFP) and with a partial fixed point operator (FO + PFP) are known to capture the complexity classes P and PSPACE respectively in the presence of an ordering relation over finite structures. Recently, Abiteboul and Vianu [AV91b] investigated the relationship of these two logics in the absence of an ordering, using a machine model of generic computation. In particular, they showed that the two languages have equivalent expressive power if and only if  $P = PSPACE$ . These languages can also be seen as fragments of an infinitary logic where each formula has a bounded number of variables,  $L_{\infty\omega}^{\omega}$  (see, for instance, [KV90]). We present a treatment of the results in [AV91b] from this point of view. In particular, we show that we can write a formula of FO + LFP that defines an ordering of the  $L_{\infty\omega}^k$  types uniformly over all finite structures. One consequence of this is a generalization of the equivalence of FO + LFP and P from ordered structures to classes of structures where every element is definable. We also settle a conjecture mentioned in [AV91b] by showing that FO + LFP is properly contained in the polynomial time computable fragment of  $L_{\infty\omega}^{\omega}$ , raising the question of whether the latter fragment is a recursively enumerable class.

# 1 Introduction

In applications of finite model theory in computer science, extensions of first-order logic by various induction operations have received particular attention. Many database query languages are based on such extensions (see, for instance, [AV91a]) and in the area of descriptive complexity, they have been shown to naturally characterize certain complexity classes. In particular, the extensions of first-order logic with a least fixed point operator (FO + LFP) and with a partial fixed point operator (FO + PFP) are known to capture the complexity classes P and PSPACE respectively in the presence of an ordering relation. Recently, Abiteboul and Vianu [AV91b] investigated the relationship of these two logics in the absence of an ordering, using a machine model of generic computation. In particular, they showed that the two languages have equivalent expressive power if and only if  $P = PSPACE$ .

The languages FO + LFP and FO + PFP can also be seen as fragments of an infinitary logic where each formula has a bounded number of variables,  $L_{\infty\omega}^\omega$ . Kolaitis and Vardi [KV90] took this view and proved a generalization of the 0-1 law for FO + LFP. Following their lead, we present a treatment of the results in [AV91b] under this view. In particular, we show that we can write a formula of FO + LFP that defines an ordering of the  $L_{\infty\omega}^k$  types in any structure. This is a refinement of the technique in [AV91b], where a distinct ordering was used for every query. The proofs we present make no reference to a particular model of computation and, it is hoped, shed some light on these results.

We also settle a conjecture mentioned in [AV91b] by showing that FO + LFP is properly contained in the polynomial time computable fragment of  $L_{\infty\omega}^\omega$ . This raises the question of whether the latter fragment is a recursively enumerable class. We give some indication of how this question might be addressed in Section 9.

This paper is organized as follows. In Section 2, we define the logics FO + LFP and FO + PFP. In Sections 3 and 4, we introduce infinitary logic and some related technical tools. Sections 5 and 6 establish that types in  $L_{\infty\omega}^k$  can be uniformly defined and ordered in FO + LFP and some consequences of this fact. This construction is then used in Sections 7 and 8 to investigate the relationship of FO + LFP and FO + PFP, including the proofs of the results of [AV91b].

## Definitions and Notation

A *signature* (also sometimes called a *language* or a *vocabulary*)  $\sigma$  is a finite sequence of relation and constant symbols  $\langle R_1, \dots, R_m, c_1, \dots, c_n \rangle$ . Associated with each relation symbol,  $R_i$  is an arity  $a_i$ .

A structure over the the signature  $\sigma$ ,  $\mathfrak{A} = \langle A, R_1^{\mathfrak{A}}, \dots, R_m^{\mathfrak{A}}, c_1^{\mathfrak{A}}, \dots, c_n^{\mathfrak{A}} \rangle$  consists of a set  $A$ , the universe of the structure, relations  $R_i^{\mathfrak{A}} \subseteq A^{a_i}$  interpreting the relation symbols in  $\sigma$  and distinguished elements  $c_1^{\mathfrak{A}}, \dots, c_n^{\mathfrak{A}}$  of  $A$  interpreting the constant symbols. Unless otherwise mentioned, all structures we will be dealing with are assumed to have finite universe. For convenience, we will assume that the universe  $A$  is an initial segment of the natural numbers. We will also write  $|\mathfrak{A}|$  for the universe of the structure  $\mathfrak{A}$ .

A *query* is a collection of structures,  $K$ , over some fixed signature  $\sigma$  that is closed under isomorphism, *i.e.* if  $\mathfrak{A} \in K$  and  $\mathfrak{A} \cong \mathfrak{B}$  then  $\mathfrak{B} \in K$ .

We will write FO, FO + LFP, *etc.* both to denote logics (*i.e.* sets of formulas) and the classes of queries that are expressible in the respective logics. It will be clear from the context which usage is intended.

## 2 Inductive Logic

In the context of finite models, the expressive power of first-order logic is known to be extremely limited. Various extensions of first-order logic have been studied that correspond to independently defined complexity classes. One way of increasing the expressive power of first-order logic is by adding some kind of induction operation.

Let  $\phi(R, x_1, \dots, x_k)$  be a first-order formula over the signature  $\sigma \cup \{R\}$  with free variables  $x_1, \dots, x_k$ , where  $k$  is the arity of  $R$ . For any structure  $\mathfrak{A}$  over the signature  $\sigma$ ,  $\phi$  defines a mapping,  $\Phi$  on relations of arity  $k$  in the following sense — given a relation  $R^{\mathfrak{A}} \subseteq |\mathfrak{A}|^k$ , let  $\langle \mathfrak{A}, R^{\mathfrak{A}} \rangle$  be the expansion of  $\mathfrak{A}$  interpreting  $R$  as  $R^{\mathfrak{A}}$ . Then,  $\Phi(R^{\mathfrak{A}}) = \{ \langle a_1, \dots, a_k \rangle \mid \langle \mathfrak{A}, R^{\mathfrak{A}} \rangle \models \phi[a_1, \dots, a_k] \}$

This map  $\Phi$  is called monotone if for any relations  $R$  and  $S$  such that  $R \subseteq S$ ,  $\Phi(R) \subseteq \Phi(S)$ . A map that is monotone has a least fixed point, *i.e.* a smallest relation  $R$  such that  $\Phi(R) = R$ . Moreover, this least fixed point can be obtained by the following iterative construction: Let  $\Phi^0 = \emptyset$  and  $\Phi^{m+1} = \Phi(\Phi^m)$ . Then for some  $m$  (depending on the

structure  $\mathfrak{A}$ ),  $\Phi^{m+1} = \Phi^m =$  the least fixed point of  $\Phi$ .  $m$  is called the *closure ordinal* of  $\Phi$  on the structure  $\mathfrak{A}$ . If  $n$  is the size of  $\mathfrak{A}$ , then there are  $n^k$   $k$ -tuples in  $\mathfrak{A}$  and since  $\Phi$  is monotone,  $m \leq n^k$ .

A sufficient syntactic condition for the formula  $\phi$  to define a monotone map on all structures is that  $\phi$  be *positive* in  $R$ , that is to say that all occurrences of  $R$  in  $\phi$  be within the scope of an even number of negations. We can now define the logic FO + LFP over signature  $\sigma$  as the smallest set of formulas satisfying:

- if  $\phi$  is a first-order formula over  $\sigma$ , then  $\phi \in \text{FO} + \text{LFP}(\sigma)$ ,
- if  $\phi$  is formed from formulas in FO + LFP( $\sigma$ ) by conjunction, disjunction, negation and first-order quantification, then  $\phi \in \text{FO} + \text{LFP}(\sigma)$ , and
- if  $\phi \in \text{FO} + \text{LFP}(\sigma \cup \{R\})$ ,  $\phi$  is positive in  $R$  and  $x_1, \dots, x_k$  are distinct variables, where  $k$  is the arity of  $R$ , then  $\text{lfp}(R, x_1 \dots x_k)\phi(t_1 \dots t_k) \in \text{FO} + \text{LFP}(\sigma)$  for any terms  $t_1, \dots, t_k$ .

The way to read the last clause above is that the operator **lfp** binds the second order variable  $R$  and the first-order variables  $x_1, \dots, x_k$  in  $\phi$  to form a new predicate. This predicate is to be interpreted as the  $k$ -ary relation that is the least fixed point of the monotone operator defined by  $\phi$ . This predicate is then evaluated at the elements denoted by the terms  $t_1, \dots, t_k$ .

The following normal form result was established in [Imm86] for formulas of FO + LFP.

**Theorem 1** *In any vocabulary containing constant symbols, every formula in FO + LFP is equivalent to a formula  $\text{lfp}(R, \bar{x})\phi(\bar{t})$ , where  $\phi$  is first-order.*

For examples of the use of the **lfp** operator, see Axioms 4–6 in Section 8.

Alternatively, we can define the language FO + IFP which has an operation **ifp** (inflationary fixed point) in place of **lfp**. In  $\text{ifp}(R, x_1 \dots x_k)\phi(t_1 \dots t_k)$ ,  $\phi$  is not required to be positive in  $R$ . It denotes the least fixed point of the operator  $\Phi'$  given by  $\Phi'(R^{\mathfrak{A}}) = \{\langle a_1, \dots, a_k \rangle \mid \langle a_1, \dots, a_k \rangle \in R^{\mathfrak{A}} \text{ or } \langle \mathfrak{A}, R^{\mathfrak{A}} \rangle \models \phi[a_1, \dots, a_k]\}$ . This language is equivalent in expressive power to FO + LFP:

**Theorem 2** ([GS86]) *A query is expressible in FO + IFP if and only if it is expressible in FO + LFP.*



Immerman [Imm86] and Vardi [Var82] independently showed that when we include a total ordering on the domain as part of the logical vocabulary, the language FO + LFP expresses exactly the class of polynomial time computable queries.

**Theorem 3** ([Imm86],[Var82]) FO + LFP *with ordering* = P.

We saw above how a formula with one free predicate variable defined an operator on relations. This, of course, is true even when the formula is not positive in the predicate variable and the operator, in turn, may or may not be monotone. Moreover, the iterative stages of the operator can still be defined, though they are not guaranteed to converge to a fixed point in the case of non-monotone operators. Let  $\phi(R, \bar{x})$  be a formula that defines a (possibly non-monotone) operator  $\Phi$ . Define the *partial fixed point* of  $\phi$  to be  $\Phi^m$  if there is an  $m$  such that  $\Phi^{m+1} = \Phi^m$ , and empty otherwise. Because there are only  $2^{n^k}$  sets of  $k$ -tuples over a structure of size  $n$ , if such an  $m$  exists  $m \leq 2^{n^k}$ . We can then define another extension of first-order logic called FO + PFP with a syntax similar to that of FO + LFP except that the **lfp** operation is replaced by **pfp**, which can operate on arbitrary formulas, not just positive ones.  $\mathbf{pfp}(R, \bar{x})\phi$  denotes the partial fixed point of  $\phi$ .

It has been shown in [AV91a] that the language FO + PFP is equivalent to the query language *while* – an extension of first-order logic with an iterative operation. Putting this together with a result of Vardi [Var82], we get the following:

**Theorem 4** ([Var82],[AV91a]) FO + PFP *with ordering* = PSPACE.

### 3 Infinitary Logic

We first define the syntax of full infinitary logic. This language is denoted  $L_{\infty\omega}$ , the first subscript indicating that conjunctions and disjunctions can be taken over arbitrary sets of formulas and the second subscript that only finite quantifier blocks are allowed<sup>1</sup>. In this notation, first-order logic would be  $L_{\omega\omega}$ . The formulas of  $L_{\infty\omega}$  are defined as for first-order logic, except that conjunction and disjunction are no longer binary operations. Rather, for any set of infinitary formulas  $\Phi$ ,  $\bigvee \Phi$  and  $\bigwedge \Phi$  are both formulas of  $L_{\infty\omega}$ .

$L_{\infty\omega}$  is complete in expressive power in the following sense. Consider any class of finite structures  $\mathcal{C}$  such that  $\mathcal{C}$  is closed under isomorphism. Since any finite structure  $\mathfrak{A}$

---

<sup>1</sup>The notation for  $L_{\infty\omega}$ ,  $L_{\infty\omega}^k$  and  $L_{\infty\omega}^\omega$  is borrowed from [Bar77].

is completely characterized up to isomorphism by a first-order sentence,  $\phi_{\mathfrak{A}}$ ,  $\mathcal{C}$  is expressed by the  $L_{\infty\omega}$  sentence  $\bigvee\{\phi_{\mathfrak{A}} \mid \mathfrak{A} \in \mathcal{C}\}$ . Clearly, this language is too strong. One restriction of this language that has been studied is obtained by allowing only finitely many variables in any single formula.

**Definition 1**  $L_{\infty\omega}^k$  is the collection of formulas of  $L_{\infty\omega}$  that have at most  $k$  distinct variables (free or bound).  $L_{\infty\omega}^{\omega}$  is the collection of formulas of  $L_{\infty\omega}$  that have a finite number of distinct variables.

$$L_{\infty\omega}^{\omega} = \bigcup_{k=1}^{\infty} L_{\infty\omega}^k.$$

In what follows, we will assume that any formula in  $L_{\infty\omega}^k$  is written so as to use only the variables  $x_1, \dots, x_k$ .

The language  $L_{\infty\omega}^{\omega}$  is restricted in its expressive power when compared with  $L_{\infty\omega}$ , yet it is still powerful enough to express properties that are not recursive (see, for instance, [KV90]). To show that the restriction is real, we need to exhibit some property that cannot be expressed in the former language. To this end, we now present a version of the Ehrenfeucht-Fraïssé games.

We state and prove the following result in its full generality. In particular, the theorem, as stated, is true for *all* structures, not just finite ones. We will then consider the special cases that are of interest. We begin with some notation.  $dom(f)$  denotes the domain of the function  $f$ ,  $rng(f)$  its range and  $|f|$  its cardinality.  $qr(\phi)$  denotes the quantifier rank of a formula, defined as:

**Definition 2** The quantifier rank of a formula  $\phi$ , written  $qr(\phi)$  is defined inductively as follows:

1. if  $\phi$  is atomic then  $qr(\phi) = 0$ ,
2. if  $\phi = \neg\psi$  then  $qr(\phi) = qr(\psi)$ ,
3. if  $\phi = \bigvee \Phi$  or  $\phi = \bigwedge \Phi$  then  $qr(\phi) = \sup\{qr(\psi) \mid \psi \in \Phi\}$ , and
4. if  $\phi = \exists x\psi$  or  $\phi = \forall x\psi$  then  $qr(\phi) = qr(\psi) + 1$ .

**Definition 3** A function  $f$  is a partial isomorphism from  $\mathfrak{A}$  to  $\mathfrak{B}$  if the domain of  $f$  is a subset of  $\mathfrak{A}$  that includes the interpretations of all constants in the language of  $\mathfrak{A}$  and if  $f$  is an isomorphic map over this domain, i.e.  $f(c^{\mathfrak{A}}) = c^{\mathfrak{B}}$  for all constants  $c$  and for all relation symbols  $R$  and  $a_1, \dots, a_m$  in the domain of  $f$ ,  $\mathfrak{A} \models R^{\mathfrak{A}}(a_1, \dots, a_m)$  if and only if  $\mathfrak{B} \models R^{\mathfrak{B}}(f(a_1), \dots, f(a_m))$ .

**Definition 4** For any two structures  $\mathfrak{A} = \langle A, \dots \rangle$  and  $\mathfrak{B} = \langle B, \dots \rangle$  and any ordinal  $\alpha$ , a collection of sets of partial isomorphisms  $\{I_\beta \mid \beta \leq \alpha\}$  is said to have the  $k$  back and forth property if and only if:

1. Each  $I_\beta$  is non-empty,
2.  $I_0 \supseteq I_1 \supseteq \dots \supseteq I_\beta \supseteq \dots \supseteq I_\alpha$ ,
3. If  $f \in I_\beta$  ( $0 \leq \beta \leq \alpha$ ) and  $g \subseteq f$  then  $g \in I_\beta$ , and
4. For every  $f \in I_{\beta+1}$  ( $0 \leq \beta < \alpha$ ) such that  $|f| < k$  and every  $a \in A$  (resp.  $b \in B$ ), there is a  $g \in I_\beta$  with  $f \subseteq g$  and  $a \in \text{dom}(g)$  (resp.  $b \in \text{rng}(g)$ ).

**Theorem 5** For any two structures,  $\mathfrak{A} = \langle A, \dots \rangle$ ,  $\mathfrak{B} = \langle B, \dots \rangle$  in a purely relational language, the following statements are equivalent:

1. For all sentences  $\phi \in L_{\infty\omega}^k$  with  $qr(\phi) \leq \alpha$ ,

$$\mathfrak{A} \models \phi \text{ iff } \mathfrak{B} \models \phi$$

2. There is a collection  $\{I_\beta \mid \beta \leq \alpha\}$  of non-empty sets of partial isomorphisms from  $\mathfrak{A}$  to  $\mathfrak{B}$  with the  $k$  back and forth property.

**Proof:**

(2  $\Rightarrow$  1) We show by induction on  $\beta$  that for formulas  $\phi(y_0 \dots y_m) \in L_{\infty\omega}^k$ , with  $qr(\phi) \leq \beta$ , if  $f \in I_\beta$  and  $a_0, \dots, a_m \in \text{dom}(f)$ , then  $\mathfrak{A} \models \phi[a_0 \dots a_m]$  iff  $\mathfrak{B} \models \phi[f(a_0) \dots f(a_m)]$ .

*Basis:*

If  $qr(\phi) = 0$  then  $\phi$  is a boolean combination of atomic formulas and since  $f$  is a partial isomorphism, the result follows.

*Induction Step:*

We now proceed by induction on the structure of the formula  $\phi$ . The cases  $\phi = \neg\psi$  and  $\phi = \bigwedge_{j \in J} \psi_j$  are trivial. So, we only need to consider the case where  $\phi = \exists y_0 \psi[y_0 \dots y_m]$ . Note that  $qr(\phi) = \delta + 1$  where  $qr(\psi) = \delta$ .

Suppose  $\mathfrak{A} \models \phi[a_1 \dots a_m]$  for some  $a_1, \dots, a_m \in A$  and that  $a_1, \dots, a_m \in \text{dom}(f)$  for some  $f \in I_{\delta+1}$ . Then, there is an  $a_0 \in A$  such that  $\mathfrak{A} \models \psi[a_0 a_1 \dots a_m]$ . By clause 3 of Definition 4, there is an  $f' \in I_{\delta+1}$  with  $\text{dom}(f') = \{a_1, \dots, a_m\}$  and  $F' \subseteq f$ . Since  $|f'| < k$ , by clause 4 there is a  $g \in I_\delta$  extending  $f'$  such that  $a_0 \in \text{dom}(g)$ . But then, by the induction hypothesis,  $\mathfrak{B} \models \psi[g(a_0)g(a_1) \dots g(a_m)]$ , i.e.  $\mathfrak{B} \models \phi[g(a_1) \dots g(a_m)]$  and therefore  $\mathfrak{B} \models \phi[f(a_1) \dots f(a_m)]$ , since  $f$  and  $g$  agree on  $a_1, \dots, a_m$ .

Similarly, if  $\mathfrak{B} \models \phi[b_1 \dots b_m]$  and  $b_1, \dots, b_m \in \text{rng}(f)$ , then  $\mathfrak{A} \models \phi[f^{-1}(b_1) \dots f^{-1}(b_m)]$ .

(1  $\Rightarrow$  2) Define the  $I_\beta$  as follows:  $f \in I_\beta$  if and only if  $f$  is a partial isomorphism from  $\mathfrak{A}$  to  $\mathfrak{B}$  and for all formulas  $\phi \in L_{\infty\omega}^k$  with  $qr(\phi) \leq \beta$  and all  $a_0 \dots a_m \in \text{dom}(f)$ ,  $\mathfrak{A} \models \phi[a_0 \dots a_m]$  iff  $\mathfrak{B} \models \phi[f(a_0) \dots f(a_m)]$ .

By definition,  $I_\delta \supseteq I_\beta$  for  $\delta \leq \beta$ . Also, since  $\mathfrak{A}$  and  $\mathfrak{B}$  agree on all sentences of quantifier rank up to  $\alpha$ , the empty partial isomorphism is in  $I_\alpha$  and therefore all the  $I_\beta$  are non-empty. It is also clear that if  $f \in I_\beta$  and  $g \subseteq f$  then  $g \in I_\beta$ . Thus, we only need to show that clause 4 holds.

For contradiction, suppose that there is an  $f \in I_{\beta+1}$  with  $|f| < k$  and an  $a \in A$  such that for all  $g \in I_\beta$  with  $g \supseteq f$ ,  $a \notin \text{dom}(g)$ . Then, for every  $b \in B$ , there must be a formula  $\psi_b[y_0 y_1 \dots y_m]$  with  $qr(\psi_b) \leq \beta$  such that  $\mathfrak{A} \models \psi_b[aa_1 \dots a_m]$  and  $\mathfrak{B} \models \neg\psi_b[bf(a_1) \dots f(a_m)]$  (where  $a_1, \dots, a_m \in \text{dom}(f)$ ). Let  $\phi = \exists y \bigwedge_{b \in B} \psi_b[y y_1 \dots y_m]$ . But then,  $qr(\phi) = \beta + 1$ ,  $\mathfrak{A} \models \phi[a_1 \dots a_m]$  and  $\mathfrak{B} \models \neg\phi[f(a_1) \dots f(a_m)]$  contradicting the assumption that  $f \in I_{\beta+1}$ .  $\blacksquare$

Note that two structures  $\mathfrak{A}$  and  $\mathfrak{B}$  agree on all sentences of  $L_{\infty\omega}$  of quantifier rank less than  $n$  if they agree on all first-order sentences of quantifier rank less than  $n$ . This can be shown by a simple induction argument on the structure of the infinitary formulas<sup>2</sup>.

<sup>2</sup>This works only in finite relational languages – it is not true when function symbols are present or there are infinitely many relation symbols. This is because the use of function symbols involves a “hidden” increase in quantifier rank, as can be seen by the process of re-writing formulas with functions into equivalent relational formulas.

Moreover, if the two structures are finite, then any chain of sets of partial isomorphisms as above of length  $\omega$  can be extended to any ordinal length. To see this, note that there are only finitely many maps from subsets of  $A$  into  $B$ . Thus, one of the sets in the chain must be repeated, and hence, can be repeated indefinitely. Writing  $L_k$  for the fragment of first-order logic with at most  $k$  variables, we have the following corollary:

**Corollary 1** *For finite structures  $\mathfrak{A}$  and  $\mathfrak{B}$ , the following are equivalent:*

- *For every sentence  $\phi \in L_{\infty\omega}^k$ ,  $\mathfrak{A} \models \phi$  iff  $\mathfrak{B} \models \phi$*
- *For every sentence  $\phi \in L_k$ ,  $\mathfrak{A} \models \phi$  iff  $\mathfrak{B} \models \phi$*

We write  $\mathfrak{A} \equiv_k \mathfrak{B}$  to denote that  $\mathfrak{A}$  and  $\mathfrak{B}$  satisfy the same sentences of  $L_k$ .

When the sequence of sets of partial isomorphisms is finite, we can view it in terms of the following two-player pebble game. We have a board consisting of one copy of each of the structures  $\mathfrak{A}$  and  $\mathfrak{B}$ . There is also a supply of pairs of pebbles  $\{\langle a_1, b_1 \rangle, \dots, \langle a_k, b_k \rangle\}$ . At each move of the game, Player I picks up one of the pebbles (either an unused pebble, or one that is already on the board) and places it on an element of the corresponding structure (*i.e.* she places  $a_i$  on an element of  $A$  or  $b_i$  on an element of  $B$ ). Player II then responds by placing the unused pebble in the pair on an element of the other structure. Player II loses if the resulting map,  $f$ , from  $\mathfrak{A}$  to  $\mathfrak{B}$ , given by  $f(a_j) = b_j, 1 \leq j \leq k$ , is not a partial isomorphism. Player II wins the  $n$ -move game if she has a strategy to avoid losing in the first  $n$  moves, regardless of what moves are made by Player I. We then have the following characterization:

**Corollary 2** *If Player II has a winning strategy for  $n$  moves of the  $k$  pebble game on structures  $\mathfrak{A}$  and  $\mathfrak{B}$ , then  $\mathfrak{A}$  and  $\mathfrak{B}$  agree on all first-order sentences of quantifier rank up to  $n$  with at most  $k$  distinct variables.*

The languages FO + LFP and FO + PFP (see Section 2) can be viewed as fragments of  $L_{\infty\omega}^\omega$ . Consider any formula  $\phi \equiv \mathbf{lfp}(S, x, \dots, x_n)\psi(S)$ , where  $\psi$  is a first-order formula positive in  $S$ . We can define the  $m^{\text{th}}$  iterative stage of  $\phi$  by a first-order formula  $\psi^m$  defined inductively, as follows:

$$\begin{aligned} \psi^0 &\equiv \neg(x = x) \\ \psi^{m+1} &\equiv \psi(\psi^m) \text{ obtained from } \psi \text{ by replacing every occurrence of } S \text{ by } \psi^m \end{aligned}$$

Then,  $\phi$  is equivalent to the formula  $\bigvee_{m=0}^{\infty} \psi^m$ . Similarly,  $\mathbf{pfp}(S, x, \dots, x_n)\psi(S)$  is equivalent to  $\bigvee_{m=0}^{\infty} (\psi^m(x_1 \dots x_n) \wedge \forall x_1 \dots \forall x_n (\psi^m(x_1 \dots x_n) \leftrightarrow \psi^{m+1}(x_1 \dots x_n)))$ . Note that each of these has at most  $n$  variables more than  $\psi$  and is, therefore, in  $L_{\infty\omega}^{\omega}$ . Hence, we have:

**Corollary 3** *If two (finite) structures agree on all sentences of  $L_k$ , then they agree on all sentences of  $FO + LFP$  and  $FO + PFP$  with at most  $k$  distinct variables.*

## 4 Characterizing Structures up to $L_k$ -equivalence

It is clear that for every finite structure  $\mathfrak{A}$ , we can write a first-order sentence  $\phi_{\mathfrak{A}}$  such that any structure that satisfies  $\phi_{\mathfrak{A}}$  is isomorphic to  $\mathfrak{A}$ . A simple application of Theorem 5 shows that not all such sentences are in  $L_k$  for any given  $k$ . This raises the question of whether there is a sentence  $\phi_{\mathfrak{A}}^k$  of  $L_k$  associated with  $\mathfrak{A}$  such that any structure satisfying it is  $L_k$ -equivalent to  $\mathfrak{A}$ . In this section, we answer this question in the affirmative. The proof is adapted from the proof of Scott's theorem in [Bar73]. For the purpose of this section, we will assume that there are no constants in the language being considered. The results can be easily generalized to the case where constants are present.

Let  $A$  be the universe of  $\mathfrak{A}$  and let  $S = A^{\leq k}$  be the set of sequences of elements of  $A$  of length less than or equal to  $k$ . For  $s \in S$  and  $a \in A$ , let  $s \cdot \langle a \rangle$  denote the sequence obtained by extending  $s$  by the single element  $a$ .

We define a formula  $\phi_s^m$  for each  $s \in S$  and each  $m \in \mathbb{N}$ . The formula has free variables  $x_1, \dots, x_l$ , where  $l$  is the length of  $s$ . We want it to be the case that  $\mathfrak{A} \models \phi_s^m[s]$  and that this formula characterizes  $s$  completely up to equivalence on formulas with  $k$  variables and quantifier rank  $m$ . The  $\phi_s^m$  are defined by induction as follows:

for all  $s = \langle a_1 \dots a_l \rangle$ ,

$\phi_s^0(x_1 \dots x_l)$  is the conjunction of all atomic and negated atomic formulas  $\theta(x_1 \dots x_l)$   
such that  $\mathfrak{A} \models \theta[a_1 \dots a_l]$

if  $\text{length}(s) < k$  then,

$$\phi_s^{m+1}(x_1 \dots x_l) = \phi_s^m(x_1 \dots x_l) \wedge \quad (1)$$

$$\bigwedge_{a \in A} \exists x_{l+1} \phi_{s \cdot \langle a \rangle}^m(x_1 \dots x_{l+1}) \wedge \quad (2)$$

$$\forall x_{l+1} \bigvee_{a \in A} \phi_{s \cdot \langle a \rangle}^m(x_1 \dots x_{l+1}) \quad (3)$$

if  $\text{length}(s) = k$  then,

$$\phi_s^{m+1}(x_1 \dots x_k) = \bigwedge_{i=1 \dots k} \phi_{s_i}^{m+1}(x_1 \dots x_{i-1} x_{i+1} \dots x_k) \quad (4)$$

where  $s_i$  is the sequence obtained from  $s$  by deleting the  $i^{\text{th}}$  element

**Lemma 1** *Let  $s = \langle a_1 \dots a_l \rangle \in S$  be a sequence of elements from  $\mathfrak{A}$  with  $l \leq k$ . For any finite structure  $\mathfrak{B} = \langle B, \dots \rangle$  and  $b_1, \dots, b_l \in B$ ,  $\mathfrak{B} \models \phi_s^m[b_1 \dots b_l]$  if and only if there is a sequence of sets of partial isomorphisms  $I_0 \supseteq \dots \supseteq I_m$  with the  $k$  back and forth property and  $f = \{\langle a_1, b_1 \rangle \dots \langle a_l, b_l \rangle\} \in I_m$ .<sup>3</sup>*

**Proof:**

$\Leftarrow$  This follows immediately from the proof of Theorem 5 since the existence of such a sequence implies that for any  $\phi$  of quantifier rank  $m$ ,  $\mathfrak{B} \models \phi[b_1 \dots b_l]$  if  $\mathfrak{A} \models \phi[a_1 \dots a_l]$ . Clearly,  $qr(\phi_s^m) = m$  and  $\mathfrak{A} \models \phi_s^m[s]$ .

$\Rightarrow$  The proof is by induction on  $m$ .

**Basis** Let  $I_0 = \{g \mid g \subseteq f\}$ .  $f$  is a partial isomorphism, because  $\mathfrak{B} \models \phi_s^0[f(s)]$ . Even if  $s = \langle \rangle$  and  $f$  is the empty map,  $I_0$  is non-empty.

**Induction Step** There are two cases to be considered:

**Case:**  $l < k$

Let  $I_{m+1} = \{g \mid g \subseteq f\}$ .

By induction hypothesis and (1), there is a sequence  $I_0^s \dots I_m^s$  with the  $k$  back and forth property and  $f \in I_m^s$ .

Furthermore, by (2) and the induction hypothesis, for every  $a \in A$ , there is a  $b \in B$  and a sequence  $I_0^{s \cdot \langle a \rangle} \dots I_m^{s \cdot \langle a \rangle}$  with the  $k$  back and forth property such that  $\{\langle a_1, b_1 \rangle \dots \langle a_l, b_l \rangle, \langle a, b \rangle\} \in I_m^{s \cdot \langle a \rangle}$ .

---

<sup>3</sup>We have stated this lemma and Theorem 6 only for the case of finite structures, since that is the case that is of interest here. However, similar results can be derived for the case where the structures may be infinite. In the latter case, the conjunction in (2) and the disjunction in (3) could be infinitary. Thus, the formulas constructed are no longer first-order, but they are in  $L_{\infty\omega}^k$ .

Let  $I_j = I_j^s \cup \bigcup_{a \in A} I_j^{s \cdot \langle a \rangle}$  (for  $0 \leq j \leq m$ ). Note that, in general, the  $k$  back and forth property is preserved under this kind of element-wise union. Thus, we need to verify that  $I_{m+1} \subseteq I_m$  and that every element of  $I_{m+1}$  is extensible in  $I_m$  to arbitrary elements of  $A$  and  $B$ . The former follows from the fact that  $I_{m+1} \subseteq I_m^s$  and the latter follows from (2) and (3) respectively.

**Case:**  $l = k$

By the argument for the case above, there are sequences  $I_0^i \dots I_{m+1}^i$  corresponding to each of the partial isomorphisms,  $f_i$ , obtained by dropping the pair  $\langle a_i, b_i \rangle$  from  $f$ .

Let  $I_j = \{f\} \cup \bigcup_{i=1 \dots k} I_j^i$  for  $0 \leq j \leq m+1$ . Each of the  $I_j$  is still closed under restrictions, because if  $g \subseteq f$ , then either  $g = f$  or  $g \subseteq f_i$  for some  $i$ . Since  $|f| = k$ , extensibility of  $f$  is not required, and we are done.  $\blacksquare$

For a given sequence  $s$  of length  $l$ , let  $X_s^m = \{s' \in S \mid \mathfrak{A} \models \phi_s^m[s']\}$ . Each  $X_s^m$  is a set of  $l$ -tuples of  $A$  and  $X_s^m \supseteq X_s^{m+1}$ . Since  $A$  is finite, there must be an  $m_s$  such that  $X_s^{m_s} = X_s^m$  for all  $m > m_s$ . Let  $m^* = \max(m_s \mid s \in S)$ . Now, define the sentence  $\phi$  as follows:

$$\phi \equiv \phi_{\langle \rangle}^{m^*} \wedge \bigwedge_{s \in S} \forall x_1 \dots \forall x_k (\phi_s^{m^*} \rightarrow \phi_s^{m^*+1})$$

Note that  $\phi \in L_k$  and that  $\mathfrak{A} \models \phi$ . We now show that this sentence characterizes the structure  $\mathfrak{A}$  up to  $L_k$  equivalence.

**Theorem 6** *For every finite structure  $\mathfrak{A}$  and any  $k$ , there is a sentence,  $\phi \in L_k$  such that for any structure  $\mathfrak{B}$ ,  $\mathfrak{B} \models \phi$  if and only if  $\mathfrak{A} \equiv_k \mathfrak{B}$ .*

**Proof:**

Let  $\phi$  be as defined above. We only need to show that if  $\mathfrak{B} \models \phi$ , then  $\mathfrak{A} \equiv_k \mathfrak{B}$ . Let  $F$  be the set of maps  $\{\langle a_1, b_1 \rangle, \dots, \langle a_l, b_l \rangle\}$  such that  $\mathfrak{B} \models \phi_{\langle a_1 \dots a_l \rangle}^{m^*+1}[b_1 \dots b_l]$ . The set  $F$  is non-empty since  $\mathfrak{B} \models \phi_{\langle \rangle}^{m^*+1}$ . By Lemma 1, for each  $f \in F$ , there is a sequence  $I_0^f \dots I_{m^*+1}^f$  with the  $k$  back and forth property. Let  $I_i = \bigcup_{f \in F} I_i^f$  and let  $I_m = I_{m^*+1}^f$  for all  $m > m^* + 1$ . We claim the infinite sequence  $I_0 \supseteq \dots \supseteq I_m \dots$  has the  $k$  back and forth property. We will establish the extensibility of every element of  $I_{m^*+2}$ . The rest then follows.

Consider any  $f \in I_{m^*+2}$  with  $|f| < k$  and any  $a \in A$ . By definition,  $f \in I_{m^*+1}$ . Since we know that the sequence through  $I_{m^*+1}$  has the  $k$  back and forth property, there is a



$g \in I_{m^*}$  such that  $f \subseteq g$  and  $a \in \text{dom}(g)$ . But then, by the other direction of Lemma 1,  $\mathfrak{B} \models \phi_{\langle \text{dom}(g) \rangle}^{m^*}[\langle \text{rng}(g) \rangle]$  and therefore, by the implication in  $\phi$ ,  $g \in I_{m^*+1}$  and we are done. ■

There are some points about the above construction that are noteworthy. First of all, we could have, alternatively, defined  $m^*$  as the smallest  $m$  such that  $X_s^m = X_s^{m+1}$  for all  $s$ . To see this, just observe that this is the only property of  $m^*$  used in the above proof. Given that  $k$  is the maximum length of any sequence in  $S$ , and that there are  $n^k$   $k$ -tuples in a structure of size  $n$ , we can derive the bound  $m^* \leq n^k$ . This gives us the following:

**Corollary 4** *If  $\mathfrak{A}$  is a structure of size  $n$  and  $\mathfrak{B}$  a structure such that  $\mathfrak{A}$  and  $\mathfrak{B}$  agree on all sentences of  $L_k$  of quantifier rank up to  $n^k + k + 1$ , then  $\mathfrak{A} \equiv_k \mathfrak{B}$ .*

The following corollary is also immediate:

**Corollary 5** *If  $K$  is a query closed under  $L_k$  equivalence (that is, if  $\mathfrak{A} \in K$  and  $\mathfrak{A} \equiv_k \mathfrak{B}$  then  $\mathfrak{B} \in K$ ), then  $K$  is definable in  $L_{\infty\omega}^k$ .*

**Proof:**

If we write  $\phi_{\mathfrak{A}}$  for the sentence of  $L_k$  that characterizes a structure  $\mathfrak{A}$  up to  $L_k$  equivalence, then  $K$  is defined by the sentence  $\bigvee \{\phi_{\mathfrak{A}} \mid \mathfrak{A} \in K\}$ . ■

Finally, it is not only structures that are characterized up to  $L_k$  equivalence in the above proof, but also sequences of elements.

**Definition 5** *For any sequence  $s = \langle a_1 \dots a_l \rangle$  of elements in a structure  $\mathfrak{A}$ , with  $l \leq k$ , define the  $L_k$ -type of  $s$ , denoted  $\text{Type}_k(s)$ , to be the set of formulas,  $\phi \in L_k$  with free variables among  $x_1, \dots, x_l$ , such that  $\mathfrak{A} \models \phi[a_1 \dots a_l]$ .*

Then, we get the following:

**Corollary 6** *For every structure  $\mathfrak{A}$ , for every  $l \leq k$  and sequence  $a_1, \dots, a_l$  of elements from  $\mathfrak{A}$ , there is a formula,  $\phi \in \text{Type}_k(\langle a_1, \dots, a_l \rangle)$  such that for any structure  $\mathfrak{B}$ , and elements  $b_1 \dots b_l \in B$ , if  $\mathfrak{B} \models \phi[b_1 \dots b_l]$ , then  $\text{Type}_k(\langle a_1, \dots, a_l \rangle) = \text{Type}_k(\langle b_1, \dots, b_l \rangle)$*

## 5 Inductively Ordering the Types

Having seen how, for a particular  $L_k$  type, we can write a formula that characterizes it, we now turn to writing a formula that will define a total ordering of these types. We will show

that this can be done uniformly in FO + LFP, *i.e.* a single formula will define the ordering on all structures. From this result we will derive Abiteboul and Vianu's result [AV91b] that PSPACE = P if and only if FO + LFP = FO + PFP. The technique used for defining the ordering is inspired by a color-refinement algorithm in [IL90].

Looking again at the definitions of the  $\phi_s^m$  in the last section, we can see that these formulas were defined by a simultaneous induction on first-order formulas – simultaneous in all the sequences  $s$ . This, along with the observation that the basis of this induction is finite, in the sense that there are, up to equivalence, only finitely many quantifier free formulas in a finite relational language, suggests that we could accomplish the entire process with a single formula of FO + LFP. We formalize this intuition below.

We first construct a formula of FO + LFP which defines, on any structure  $\mathfrak{A}$ , an equivalence relation on  $k$ -tuples of elements such that two tuples are equivalent if and only if they have the same  $L_k$ -type. In the following, for ease of reading, we will use the notation  $x_1 \dots x_k$  to indicate a sequence of variables in which  $x$  has been substituted for  $x_i$ , when the particular  $i$  is clear from the context.

**Definition 6** *For any structure  $\mathfrak{A}$  and elements  $a_1 \dots a_l \in |\mathfrak{A}|$ , the basic  $L_k$ -type of  $a_1 \dots a_l$  is the set of quantifier free formulas,  $\phi$ , of  $L_k$  in  $l$  free variables such that  $\mathfrak{A} \models \phi[a_1 \dots a_l]$ .*

Note that for a given finite signature,  $\sigma$ , there are only finitely many distinct basic types. Furthermore, each basic type is characterized, up to equivalence, by a single quantifier free formula of  $L_k$ .

Let  $\alpha_1(x_1 \dots x_k), \dots, \alpha_q(x_1 \dots x_k)$  be a fixed enumeration of quantifier free formulas of  $L_k$  in  $k$  variables characterizing all the basic types in some signature  $\sigma$ . Then, define  $\phi_0$  as follows:

$$\phi_0(x_1 \dots x_k y_1 \dots y_k) \equiv \bigvee_{1 \leq i \neq j \leq q} (\alpha_i(\bar{x}) \wedge \alpha_j(\bar{y}))$$

where  $\alpha_i(\bar{y})$  is obtained from  $\alpha_i(\bar{x})$  by replacing every  $x_j$  by  $y_j$ . It should be clear that for any tuples  $\bar{a}, \bar{b} \in |\mathfrak{A}|^k$ ,  $\mathfrak{A} \models \phi_0[\bar{a}\bar{b}]$  if and only if the basic types of  $\bar{a}$  and  $\bar{b}$  are different.

Now, define  $\phi$  and  $\psi$  as follows:

$$\begin{aligned} \phi(R, x_1 \dots x_k y_1 \dots y_k) \equiv \phi_0(\bar{x}\bar{y}) \vee \bigvee_{1 \leq i \leq k} \exists x \forall y R(x_1 \dots x_k y_1 \dots y_k) \\ \vee \bigvee_{1 \leq i \leq k} \exists y \forall x R(x_1 \dots x_k y_1 \dots y_k) \end{aligned}$$

As we shall see below, the least fixed point of  $\phi$  expresses the *inequivalence* of  $L_k$ -types.

$$\psi(z_1 \dots z_{2k}) \equiv \neg \text{lfp}(R, \bar{x}, \bar{y})\phi(z_1 \dots z_{2k})$$

**Claim 1** *For any structure  $\mathfrak{A}$  on signature  $\sigma$ ,  $\mathfrak{A} \models \psi[a_1 \dots a_k a'_1 \dots a'_k]$  if and only if  $\bar{a}$  and  $\bar{a}'$  have the same  $L_k$ -type.*

**Proof:**

To establish this claim, we need to show that  $\text{lfp}(R, \bar{x}, \bar{y})\phi[\bar{a}\bar{a}']$  expresses the *inequivalence* of the two  $k$ -tuples. Picture the  $k$ -pebble game being played on two isomorphic copies of  $\mathfrak{A}$ , and at some stage the pebbles are placed on  $\langle a_1, a'_1 \rangle, \dots, \langle a_k, a'_k \rangle$ . By Lemma 1, if the two tuples have the same  $L_k$ -type, then Player II can play indefinitely from this point on without losing. We claim that if  $\phi^r[\bar{a}\bar{a}']$  (the  $r^{\text{th}}$  iterative stage of  $\phi$ ), then Player I can win in  $r$  moves or less. Clearly, if  $r = 0$ , by the definition of  $\phi_0$ ,  $\bar{a}$  and  $\bar{a}'$  differ on a quantifier free formula and hence the map from one to the other is not a partial isomorphism. If  $r = m + 1$ , then the definition of  $\phi$  tells us that we can, in one move, get to a configuration that is in  $\phi^m$ . ■

We will henceforth use the symbol  $\sim_k$  in infix notation to denote the relation defined by  $\psi$ . We will now give an inductive definition of an ordering relation on the equivalence classes defined by this relation. In other words, we will define a  $2k$ -ary relation that is a pre-order on  $k$ -tuples such that two tuples are not ordered by this pre-order just in case they have the same  $L_k$ -type. This relation is defined by an induction that can be seen to parallel the induction defining the equivalence relation  $\sim_k$ . Initially, the basic types are ordered, and at each inductive stage we refine this to an ordering of the equivalence classes under the equivalence relation obtained through that stage. At any given stage, the symmetric closure of the ordering relation is the same as the inequivalence relation at that stage.<sup>4</sup>

Define the following formulas for each  $1 \leq i \leq k$ :

$$\begin{aligned} \beta_i(\bar{x}\bar{y}) &\equiv \forall x \exists y ((x_1 \dots x_k) \sim_k (y_1 \dots y_k)) \wedge \\ &\quad \forall y \exists x ((x_1 \dots x_k) \sim_k (y_1 \dots y_k)); \\ \delta_i(\bar{x}\bar{y}) &\equiv \bigwedge_{j < i} \beta_j \wedge \neg \beta_i. \end{aligned}$$

---

<sup>4</sup>This is not completely true in the construction we give, but we will assume it for expositional purposes. The formula constructed could be made to accord with this assumption by replacing the relation  $\sim_k$ , in the definition of the formulas  $\delta_i$ , with its inductive stages.

$\beta_i$  is true of a pair of tuples if we cannot distinguish their  $L_k$ -type on the basis of their  $i^{\text{th}}$  elements.  $\delta_i$  holds of a pair of tuples if  $i$  is the first position that distinguishes the  $L_k$ -types of the two tuples.

Let

$$\theta_0(x_1 \dots x_k y_1 \dots y_k) \equiv \bigvee_{1 \leq i < j \leq q} (\alpha_i(\bar{x}) \wedge \alpha_j(\bar{y})).$$

That is,  $\theta_0$  defines a total ordering on the basic  $L_k$ -types. To refine this ordering by induction, let  $R$  be a  $2k$ -ary relation symbol. Define for each  $i$  ( $1 \leq i \leq k$ ) the following pair of formulas:

$$\begin{aligned} \sigma_1^i(x, \bar{x}\bar{y}) &\equiv \forall y (R(x_1 \dots x \dots x_k y_1 \dots y \dots y_k) \vee R(y_1 \dots y \dots y_k x_1 \dots x \dots x_k)); \\ \sigma_2^i(y, \bar{x}\bar{y}) &\equiv \forall x (R(x_1 \dots x \dots x_k y_1 \dots y \dots y_k) \vee R(y_1 \dots y \dots y_k x_1 \dots x \dots x_k)). \end{aligned}$$

Define the set of tuples  $move_i((a_1 \dots a_k)) = \{(a_1 \dots a \dots a_k) \mid a \in A\}$ , *i.e.* the tuples obtained by replacing the  $i^{\text{th}}$  element. If two tuples  $\bar{a}$  and  $\bar{b}$  are inequivalent at stage  $r + 1$  in the induction of  $\phi$ , then, for some  $i$ , there is a tuple in  $move_i(\bar{a})$  which is inequivalent to every tuple in  $move_i(\bar{b})$  (or *vice versa*) at stage  $r$ . The formula  $\sigma_1^i$  (parametrized by the tuples  $\bar{a}$  and  $\bar{b}$ ) picks out the elements  $a$  such that  $a_1 \dots a \dots a_k$  is such a tuple (similarly for  $\sigma_2^i$ ).

Consider the set,  $S$ , of all tuples in  $move_i(\bar{a})$  that are not equivalent to any tuple in  $move_i(\bar{b})$  along with those in  $move_i(\bar{b})$  that are not equivalent to any tuple in  $move_i(\bar{a})$ . There must be a tuple in  $S$  that is not greater (under the ordering defined so far) than any other tuple in this set. Assume, without loss of generality, that this tuple is in  $move_i(\bar{a})$ . It must be strictly smaller (again, under the ordering so far) than all tuples in  $S \cap move_i(\bar{b})$  or it would be equivalent to some tuple in  $move_i(\bar{b})$ . The following formula would then order the tuples  $\bar{a}$  and  $\bar{b}$ , with  $\bar{a}$  being smaller, unless they had already been ordered otherwise.

$$\begin{aligned} \theta(R, \bar{x}\bar{y}) &\equiv \theta_0(\bar{x}\bar{y}) \vee \\ &(\neg R(\bar{y}\bar{x}) \wedge \bigvee_{1 \leq i \leq k} (\delta_i \wedge \exists x (\sigma_1^i(x, \bar{x}\bar{y}) \wedge \forall y (\sigma_2^i(y, \bar{x}\bar{y}) \rightarrow \\ &R(x_1 \dots x \dots x_k y_1 \dots y \dots y_k))))). \end{aligned}$$

We cannot define the least fixed point of the above formula, since it is not positive in  $R$ . However, the inflationary fixed point gives us the required ordering. We know, by Theorem 2, that there is a formula of FO + LFP equivalent to  $\psi$  below:

$$\psi(z_1 \dots z_{2k}) \equiv \mathbf{ifp}(R, \bar{x}\bar{y})\theta(z_1 \dots z_{2k}).$$

**Claim 2**

1. On any structure,  $\mathfrak{A}$ ,  $\psi$  defines a pre-order on  $k$ -tuples. We will write  $\bar{x} <_k \bar{y}$  for  $\psi(\bar{x}\bar{y})$ .
2.  $\bar{a}$  and  $\bar{a}'$  have the same  $L_k$ -type if and only if neither  $\bar{a} <_k \bar{a}'$  nor  $\bar{a}' <_k \bar{a}$ .

Using the formulas just defined, it is possible to define the  $L_k$  equivalence and the corresponding pre-order relation on tuples shorter than  $k$ .

## 6 Rigid Structures

Consider the pre-order  $<_k^1$  on single elements, *i.e.* tuples of length 1. Clearly, if there is at most one element of any  $L_k$ -type in a structure, then  $<_k^1$  defines a total ordering on the universe of the structure. Since this ordering is definable in FO + LFP, and since FO + LFP expresses all of P in the presence of ordering, this implies that FO + LFP expresses all of P on these structures. We formalize this below:

**Definition 7** *A structure  $\mathfrak{A}$  is called rigid if the only automorphism on  $\mathfrak{A}$  is the identity.*

**Definition 8** *Call a structure  $\mathfrak{A}$   $k$ -rigid if no two elements of  $\mathfrak{A}$  have the same  $L_k$ -type.*

Clearly, every  $k$ -rigid structure is rigid. Conversely,

**Theorem 7** *Every rigid structure  $\mathfrak{A}$  is  $k$ -rigid for some  $k$ .*

**Proof:**

For contradiction, assume that  $\mathfrak{A}$  is a rigid structure that is not  $k$ -rigid for any  $k$ . Then for each  $k$  there are distinct elements  $a_1^k, a_2^k$  in  $\mathfrak{A}$  which have the same  $L_k$ -type. Since  $\mathfrak{A}$  is finite, this implies that there are distinct  $a_1, a_2$  such that for infinitely many  $k$ ,  $a_1$  and  $a_2$  have the same  $L_k$ -type. But, two elements that share their  $L_k$ -type share their  $L_l$ -type for all  $l < k$ . Hence,  $a_1$  and  $a_2$  have the same first-order type. Now, expand the vocabulary by a constant symbol  $c$ , and consider the expanded structures  $\langle \mathfrak{A}, a_1 \rangle$  and  $\langle \mathfrak{A}, a_2 \rangle$ . These structures are elementarily equivalent, since  $a_1$  and  $a_2$  have the same first-order type over  $\mathfrak{A}$ . But any two finite structures that are elementarily equivalent are isomorphic. Hence

there is an automorphism of  $\mathfrak{A}$  mapping  $a_1$  to  $a_2$  which contradicts the hypothesis that  $\mathfrak{A}$  is rigid.  $\blacksquare$

The argument we gave above on the expressiveness of FO + LFP can now be formally stated as:

**Theorem 8** *Let  $K$  be a query computable in polynomial time such that there is a  $k$  such that every structure in  $K$  is  $k$ -rigid. Then  $K$  is expressible by a sentence of FO + LFP.*

Observe that any structure with a linear ordering,  $<$ , is 2-rigid. There is a formula  $\alpha_i(x) \in L_2$  which defines the  $i^{\text{th}}$  element in the ordering uniquely. For instance,

$$\alpha_3(x) \equiv \exists y(y < x \wedge \exists x(x < y \wedge \forall y(\neg y < x))).$$

Hence Theorem 7 generalizes Theorem 3.

## 7 Reduction to an Ordered Structure

In general, on a structure,  $\mathfrak{A}$ , that is not rigid,  $<_k$  defines a pre-order, or alternatively a total ordering on the  $L_k$  equivalence classes. We can look at this as the basis for a reduction of the structure  $\mathfrak{A}$  onto a totally ordered structure in which each of the equivalence classes is collapsed to a point. This translation is interesting from the following point of view – consider any formula  $\phi \equiv \text{lfp}(R, \bar{x})\psi$  (or  $\text{pfp}(R, \bar{x})\psi$ ) with only  $k$  variables. Then, not only is the relation defined by  $\phi$  on  $\mathfrak{A}$  closed under  $L_k$  equivalence, but so is every iterative stage of  $\phi$ . This raises the possibility that we can describe  $\phi$  as an induction on the  $L_k$  equivalence classes of tuples.

More formally, for any structure  $\mathfrak{A} = \langle A, R_1, \dots, R_m \rangle$ , let

$$E_k(\mathfrak{A}) = \langle A^k / \sim_k, <_k, =', R'_1, \dots, R'_m, X_i, P_s \rangle$$

be the structure defined as follows:

- The universe of  $E_k(\mathfrak{A})$  is  $A^k / \sim_k$ , *i.e.* the equivalence classes of tuples from  $A$  of length  $k$  under the equivalence relation  $\sim_k$ . We will write  $[\bar{a}]$  to denote the equivalence class that includes the tuple  $\bar{a}$ .
- $<_k$  is the total ordering on the universe of  $E_k(\mathfrak{A})$  defined in Section 5.

- $='$  is a unary relation such that  $='$  ( $[\bar{a}]$ ) holds if and only if  $\bar{a} = \langle a_1, a_2 \dots a_k \rangle$  and  $a_1 = a_2$ . This relation is well-defined, since a tuple in which the first two elements are distinct cannot be equivalent to one in which they are identical, since they differ on a basic type.
- For each relation  $R_i$  in  $\mathfrak{A}$ , of arity  $m$ , we have a unary relation  $R'_i$  in  $E_k(\mathfrak{A})$  such that  $[\langle a_1 \dots a_k \rangle] \in R'_i$  holds if and only if  $\langle a_1 \dots a_m \rangle \in R_i$ .<sup>5</sup> Again, these relations are clearly well-defined.
- $X_i$  – the substitution relation – is a binary relation such that  $X_i([\bar{a}], [\bar{a}'])$  holds if the tuples  $\bar{a}$  and  $[\bar{a}']$  differ *at most* on their  $i^{\text{th}}$  element. To see that this relation is well-defined, observe that if two tuples,  $\bar{a}_1$  and  $\bar{a}_2$ , have the same  $L_k$ -type then Player II can indefinitely play the  $k$ -pebble game on two copies of the structure  $\mathfrak{A}$  with the pebbles initially on these tuples. But then, if we can get to  $\bar{a}'_1$  in one move from  $\bar{a}_1$  there must be a  $\bar{a}'_2$  equivalent to  $\bar{a}'_1$  one move away from  $\bar{a}_2$ .
- $P_s$  is a binary relation for every sequence  $s = \langle i_1 \dots i_k \rangle$  of integers from  $\{1, \dots, k\}$ . For any tuple  $\langle a_1 \dots a_k \rangle$ ,  $([\langle a_1 \dots a_k \rangle], [\langle a_{i_1} \dots a_{i_k} \rangle]) \in P_s$ . This relation is well-defined since, if  $\phi$  is a formula in  $\text{Type}_k(\langle a_1 \dots a_k \rangle)$ , then the formula  $\phi_s$  obtained by replacing every free occurrence of every  $x_j$  by  $x_{i_j}$  is in  $\text{Type}_k(\langle a_{i_1} \dots a_{i_k} \rangle)$ . Hence, if  $\langle a_1 \dots a_k \rangle$  and  $\langle a'_1 \dots a'_k \rangle$  have the same  $L_k$ -type, then so do  $\langle a_{i_1} \dots a_{i_k} \rangle$  and  $\langle a'_{i_1} \dots a'_{i_k} \rangle$ .

We will also write  $E_k(\sigma)$  to denote the signature of  $E_k(\mathfrak{A})$ , when  $\sigma$  is the signature of  $\mathfrak{A}$ .

**Lemma 2** *For every first-order formula  $\phi$  with  $m$  free variables in the language  $E_k(\sigma)$ , there is an FO + LFP formula  $\phi'$  with  $km$  free variables in the language  $\sigma$  such that for any structure  $\mathfrak{A}$ ,  $\mathfrak{A} \models \phi'[\bar{a}_1 \dots \bar{a}_m]$  if and only if  $E_k(\mathfrak{A}) \models \phi[[\bar{a}_1] \dots [\bar{a}_m]]$ .*

**Proof:**

All the relations on  $E_k(\mathfrak{A})$ , including equality, are definable in FO + LFP on  $\mathfrak{A}$ . Moreover, these definitions are uniform, *i.e.* for each  $R \in E_k(\sigma)$  there is a single FO + LFP formula defining it for all  $\mathfrak{A}$ . So, we obtain  $\phi'$  by substituting this definition for each occurrence of

---

<sup>5</sup>As defined, this works only if the arity,  $m$ , of  $R_i$  is at most  $k$ . If this is not the case and  $m > k$ , we first replace  $R_i$  by a collection of relations of arity  $k$  by taking all the ways that we can form an  $m$ -tuple from at most  $k$  elements. This does not affect the results, since we are only considering formulas with at most  $k$  variables.

the relation symbol in  $\phi$ . This *includes* substituting the definition of  $\sim_k$  for each occurrence of the identity symbol. For each variable in  $\phi$ , we substitute  $k$  new variables and for each quantifier, a block of  $k$  quantifiers. ■

Let  $\phi(R)$  be a first-order formula in the language  $E_k(\sigma) \cup \{R\}$  where  $R$  is a relation symbol of arity  $m$  and let  $\langle E_k(\mathfrak{A}), \mathcal{R} \rangle$  be a structure for this language. Let  $\mathfrak{A}' = \langle \mathfrak{A}, S^{\mathfrak{A}} \rangle$  be a structure interpreting the language  $\sigma \cup S$  with  $S^{\mathfrak{A}} = \{\langle a_1 \dots a_{km} \rangle | \mathcal{R}([\bar{a}_1] \dots [\bar{a}_m])\}$ . By the proof of Lemma 2, there is an FO + LFP formula,  $\phi'$ , in  $\sigma \cup S$  such that  $\mathfrak{A}' \models \phi'[\bar{a}_1 \dots \bar{a}_m]$  if and only if  $\langle E_k(\mathfrak{A}), \mathcal{R} \rangle \models \phi[[\bar{a}_1] \dots [\bar{a}_m]]$ . This gives us the following result:

**Lemma 3** *For every FO + LFP (respectively FO + PFP) formula  $\phi$  in the language  $E_k(\sigma)$ , there is an FO + LFP (respectively FO + PFP) formula  $\phi'$  in the language  $\sigma$  such that for any structure  $\mathfrak{A}$ ,  $\mathfrak{A} \models \phi'[\bar{a}_1 \dots \bar{a}_m]$  if and only if  $E_k(\mathfrak{A}) \models \phi[[\bar{a}_1] \dots [\bar{a}_m]]$ .*

Note that every relation that is defined by a formula obtained in this way by translating back from a formula in one free variable in the language  $E_k(\sigma)$  is closed under the  $L_k$  equivalence relation.

We now establish a translation of formulas in the other direction. Let  $\phi$  be a formula of  $L_k$  in the language  $\sigma$ . We will define, by induction on the structure of  $\phi$ , a first-order formula  $\phi^*$  in the language  $E_k(\sigma)$ . In the translation we define, every sub-formula of  $\phi$  with free variables among  $x_1 \dots x_k$  is translated into a sub-formula of  $\phi^*$  with exactly one free variable with the property that  $E_k(\mathfrak{A}) \models \phi^*[[\bar{a}]]$  if and only if  $\mathfrak{A} \models \phi[\bar{a}]$ . That is to say, we will treat  $\phi$  as defining a  $k$ -ary relation over  $\mathfrak{A}$  even if  $\phi$  has fewer than  $k$  free variables. The relation  $\{\langle a_1 \dots a_k \rangle | \mathfrak{A} \models \phi[a_1 \dots a_k]\}$  is closed under  $L_k$  equivalence, since  $\phi \in L_k$ .

The translation is defined as follows:

- If  $\phi \equiv x_i = x_j$ , then  $\phi^*(x) \equiv \exists y P_s(y, x) \wedge \neg(y)$   
where  $s$  is a sequence chosen so that  $s = \langle i, j \dots \rangle$ .
- If  $\phi \equiv R_j(x_{i_1}, \dots, x_{i_m})$ , then  $\phi^*(x) \equiv \exists y P_s(y, x) \wedge R'_j(y)$   
where  $s$  is a sequence chosen so that  $s = \langle i_1, \dots, i_m, \dots \rangle$ .
- If  $\phi \equiv \neg\psi(\bar{x})$ , then  $\phi^*(x) \equiv \neg\psi^*(x)$
- If  $\phi(\bar{x}) \equiv \psi_1(\bar{x}) \wedge \psi_2(\bar{x})$ , then,  $\phi^*(x) \equiv \psi_1^*(x) \wedge \psi_2^*(x)$
- If  $\phi(\bar{x}) \equiv \exists x_i \psi(\bar{x})$  then,  $\phi^*(x) \equiv \exists y (X_i(x, y) \wedge \psi^*(y))$



It should be clear from the construction of  $\phi^*$  that  $\mathfrak{A} \models \phi[\bar{a}]$  just in case  $E_k(\mathfrak{A}) \models \phi^*[[\bar{a}]]$ . Moreover, this is true even if  $\phi$  is in an expanded language  $\sigma \cup \{R\}$  as long as the interpretation of  $R$  on  $\mathfrak{A}$  is closed under  $L_k$  equivalence and  $E_k(\mathfrak{A})$  is expanded to interpret  $R'$  in the obvious way. This gives us the following result:

**Lemma 4** *For every FO + LFP (respectively FO + PFP) formula  $\phi$  in the language  $\sigma$  such that  $\phi$  has at most  $k$  distinct variables, there is an FO + LFP (respectively FO + PFP) formula  $\phi^*$  in the language  $E_k(\sigma)$  such that for any structure  $\mathfrak{A}$ ,  $\mathfrak{A} \models \phi[\bar{a}]$  if and only if  $E_k(\mathfrak{A}) \models \phi^*[[\bar{a}]]$ .*

**Proof:**

Since  $\phi$  has at most  $k$  distinct variables, every iteration of every induction operator in  $\phi$  defines a relation closed under  $L_k$  equivalence. ■

We are now in a position to prove the following result from [AV91b]:

**Theorem 9** *FO + LFP = FO + PFP if and only if  $P = PSPACE$ .*

**Proof:**

$\Rightarrow$  This follows immediately from the fact that FO + LFP = P and FO + PFP = PSPACE on ordered structures. (Theorems 3 and 4 respectively).

$\Leftarrow$  Suppose  $P = PSPACE$ . Let  $\phi$  be a sentence in FO + PFP over signature  $\sigma$  and let the number of distinct variables in  $\phi$  be  $k$ . Take  $\phi^*$  to be the corresponding sentence of FO + PFP in the language  $E_k(\sigma)$  obtained as in Lemma 4. Since  $\phi^*$  is in FO + PFP, it is computable in PSPACE and hence in P, by hypothesis. Since the structures  $E_k(\mathfrak{A})$  have a total ordering on their elements, there is a sentence of FO + LFP,  $\psi$  equivalent to  $\phi^*$ . Then, by Lemma 3 there is a  $\psi'$  in FO + LFP over  $\sigma$  that is equivalent to  $\psi$ . If  $\phi$  has fewer than  $k$  free variables, we might need to take a projection of  $\psi'$ . ■

## 8 Complete Binary Trees

It is easy to see that the size of the structures  $E_k(\mathfrak{A})$  is bounded by a polynomial over all structures  $\mathfrak{A}$  (see the proof of Theorem 13 in Section 9). Over some classes of structures, it can be considerably smaller. For instance, if we consider all structures over the language

of identity, there is a bound on the size of the structures  $E_k(\mathfrak{A})$  which depends only on  $k$ . Another class of structures for which the size of  $E_k(\mathfrak{A})$  is much smaller than that of  $\mathfrak{A}$  is the class of complete binary trees. This yields some interesting results concerning logical expressibility.

Complete binary trees are graphs, *i.e.* structures  $\langle V, E \rangle$  with one binary relation  $E$  satisfying the following axioms:

$$1. \forall x(\forall y(\neg Exy) \vee \exists y\exists z(y \neq z \wedge Exy \wedge Exz \wedge \forall w(Exw \rightarrow w = y \vee w = z)))$$

This says that every vertex has exactly 0 or 2 children.

$$2. \forall x(\forall y(\neg Eyx) \vee \exists y(Eyx \wedge \forall z(Ezx \rightarrow z = y)))$$

This says that every vertex has exactly 0 or 1 parent.

$$3. \exists x(\forall y(\neg Eyx) \wedge \forall z(\forall y(\neg Eyx) \rightarrow x = z))$$

This says that there is exactly one vertex (the root) that has no parent.

$$4. \forall x\forall y\mathbf{lfp}(R, x, y)(x = y \vee \exists z(Rxz \wedge Ezy) \vee \exists z(Ryz \wedge Ezx))(x, y)$$

This says that the graph is connected, *i.e.* every pair of vertices is in the reflexive, transitive and symmetric closure of the edge relation.

$$5. \forall x\neg(\mathbf{lfp}(R, x, y)(Exy \vee \exists z(Rxz \wedge Ezy))(x, x))$$

This says that there are no cycles.

$$6. \forall x\forall y((\forall z(\neg Ezx) \wedge \forall z(\neg Eyz)) \rightarrow \delta(x, y))$$

where,

$$\delta \equiv \mathbf{lfp}(R, x, y)((\forall z(\neg Ezx) \wedge \forall z(\neg Eyz)) \vee \exists w\exists z(Rwz \wedge Ewx \wedge Ezy))(x, y)$$

This says that all leaves are at the same distance from the root ( $\delta$  defines an equivalence relation that relates vertices at the same depth).

If we let  $CBT = \{G = \langle V, E \rangle | G \text{ is a binary tree}\}$ , then by the above definition  $CBT \in \text{FO} + \text{LFP}$ . Moreover, since we used only four distinct variables,  $CBT \in L_{\infty\omega}^4$ .

Define the formulas  $\alpha_n$  recursively as follows:

$$\alpha_0(x) \equiv \forall y\neg Eyx$$

$$\alpha_{n+1}(x) \equiv \exists yEyx \wedge \exists x(x = y \wedge \alpha_n(x))$$

Then, for  $T \in CBT$ ,  $T \models \alpha_d[v]$  just in case  $v$  is a vertex of depth  $d$  in  $T$ . So, if  $T_d$  is a complete binary tree, it has depth  $d$  if and only if  $T_d \models \exists x(\alpha_d) \wedge \neg \exists x(\alpha_{d+1})$ . Note that each  $\alpha_n$  contains only two distinct variables. Since any two complete binary trees of the same depth are isomorphic, we can conclude the following:

**Lemma 5** *If  $T_1$  and  $T_2$  are two complete binary trees such that  $T_1 \equiv_2 T_2$ , then  $T_1 \cong T_2$ .*

Combining this with the axiomatization above, we get the following result:

**Lemma 6** *If  $q$  is any query in the language of graphs consisting only of complete binary trees, then  $q$  is definable in  $L_{\infty\omega}^4$ .*

Define the class,  $\mathcal{T}$ , of labeled binary trees as the class of structures over the vocabulary  $\{E, U\}$  which satisfy, in addition to the above six axioms, the following one:

$$7. \forall x \forall y (\delta(x, y) \rightarrow ((Ux \wedge Uy) \vee (\neg Ux \wedge \neg Uy)))$$

That is, all vertices at the same depth are either labeled or unlabeled.

Observe that the propositions shown above for complete binary trees apply equally well to labeled binary trees.

We also define the class,  $\mathcal{B}$ , of binary strings as structures over the same vocabulary  $\{E, U\}$  that make true Axioms 2 through 5 above, as well as:

$$1'. \forall x (\forall y (\neg Exy) \vee \exists y (Exy \wedge \forall z (Exz \rightarrow z = y)))$$

That is every vertex has exactly 0 or 1 children.

There is a natural correspondence between labeled binary trees and binary strings. In some sense, they encode the same information, with the  $i^{\text{th}}$  bit of the binary string corresponding to the  $i^{\text{th}}$  level of the tree. While we give formal definitions below, it will be instructive to keep this intuitive picture in mind and we will make appeal to it to simplify the presentation.

**Definition 9** *If  $B \in \mathcal{B}$  and  $T \in \mathcal{T}$ , then  $B \triangleright T$  if and only if, for all  $d$ :*

- $B \models \exists x \alpha_d$  if and only if  $T \models \exists x \alpha_d$ , and
- $B \models \forall x (\alpha_d \rightarrow Ux)$  if and only if  $T \models \forall x (\alpha_d \rightarrow Ux)$ .

Note that if  $B \triangleright T$  and the size of  $B$  is  $n$ , then the size of  $T$  is  $2^n - 1$

**Definition 10** For any queries  $q_B \subseteq \mathcal{B}$  and  $q_T \subseteq \mathcal{T}$ , define:

$$\begin{aligned} h(q_B) &= \{T \mid B \triangleright T \text{ for some } B \in q_B\} \\ h^{-1}(q_T) &= \{B \mid B \triangleright T \text{ for some } T \in q_T\} \end{aligned}$$

It should be clear that  $h^{-1}(h(q_B)) = q_B$ .

Lindell[Lin91] used this correspondence between binary strings and labeled binary trees to show that  $\text{FO} + \text{LFP}$  does not express all the polynomial-time queries on binary trees.

**Lemma 7** If  $q_B \in \text{DTIME}[2^{O(n)}]$  then  $h(q_B) \in P$ .

**Proof:**

Given an input  $T$ , we can verify that it is a labeled binary tree in polynomial time, since  $T \in \text{FO} + \text{LFP}$ . We can also extract from it a  $B$  such that  $B \triangleright T$  in  $\text{DSpace}[\log(n)]$ . We then pass  $B$  as the input to the acceptor for  $q_B$  which runs in time  $2^{O(d)}$ , where  $d$  is the size of  $B$ , but this is only polynomial in the size of  $T$  which is  $2^d - 1$ . ■

**Lemma 8** If  $q_T \in \text{FO} + \text{LFP}$ , then  $h^{-1}(q_T) \in \text{FO} + \text{LFP}$ .

The proof of this lemma is based on a syntactic translation similar to the one given in Section 7. The key element of Lindell's construction is that  $k$ -tuples of vertices from the tree can be encoded as fixed length tuples in the corresponding binary string. This is because a complete set of invariants (up to automorphism) for a tuple on a complete binary tree is the sequence of depths of the least common ancestors of pairs of elements in the tuple. We refer to [Lin91] for details of the translation.

Given that there are queries on strings in  $\text{DTIME}[2^{O(n)}]$  that are not in  $P$  [HS65], we conclude the following:

**Theorem 10** ([Lin91]) *There is a  $q_T \subseteq \mathcal{T}$  such that  $q_T \in P$ , but  $q_T \notin \text{FO} + \text{LFP}$ .*

Since we observed above that for every  $q$  such that  $q \subseteq \mathcal{T}$ ,  $q \in L_{\infty\omega}^4$ , we conclude that:

**Corollary 7**  $\text{FO} + \text{LFP} \subset L_{\infty\omega}^\omega \cap P$ .

Define the class  $\text{FO} + \text{PFP|P}$  of queries expressed by a formula of  $\text{FO} + \text{PFP}$  with the property that every occurrence of the **pfp** operation closes in polynomially many steps in any structure. Any query in  $\text{FO} + \text{PFP|P}$  is clearly computable in polynomial time. Also, since the operator **lfp** can be seen as an instance of **pfp** that always closes in polynomially many steps, we get

$$\text{FO} + \text{LFP} \subseteq \text{FO} + \text{PFP|P} \subseteq L_{\infty\omega}^{\omega} \cap \text{P}$$

It had been conjectured that these three classes are, in fact, equal. We have shown above that the first and the third can be separated. Abiteboul and Vianu[AV91b] have recently shown that the first and the second are equal if and only if  $\text{P} = \text{PSPACE}$ . They prove this result using a padding technique similar to the one above. We encoded binary strings of size  $n$  as trees of size  $2^n$ . For the purpose of the next result, we will need to encode them into trees of size  $2^{n^k}$ . To this end, we introduce, for every  $k$  the class of structures  $\mathcal{T}_k$  over the signature  $\{E, U, L\}$ . The trees in  $\mathcal{T}_k$  have depth  $n^k$  with the first  $n$  levels labeled by the unary relation  $L$ . Formally,  $\mathcal{T}_k$  is the class of structures which in addition to the Axioms 1 through 7, satisfy:

$$8. \forall x \forall y (\delta(x, y) \rightarrow ((Lx \wedge Ly) \vee (\neg Lx \wedge \neg Ly)))$$

That is, all vertices at the same depth are either in  $L$  or not.

$$9. \forall x \forall y ((Lx \wedge Eyx) \rightarrow Ly)$$

If a vertex is in  $L$ , then so is its parent.

10. The depth of the tree is  $n^k$ , where  $n$  is the number of levels labeled by  $L$ . This can be stated in  $\text{FO} + \text{LFP}$  by defining a  $k$ -ary induction on the levels in  $L$  that is an ordering of length  $n^k$  on  $k$ -tuples.

The binary string encoded by a tree in  $\mathcal{T}_k$  of depth  $n^k$  can be extracted by looking at the topmost  $n$  levels (the levels labeled by  $L$ ) and looking at the string defined by the relation  $U$  on these levels. We can formalize this as before with a map  $h_k$  from queries on binary strings to queries on  $\mathcal{T}_k$ .

**Definition 11** *If  $B \in \mathcal{B}$  and  $T \in \mathcal{T}_k$ , then  $B \triangleright T$  if and only if, for all  $d$ :*

- $B \models \exists x \alpha_d$  if and only if  $T \models \exists x (Lx \wedge \alpha_d)$ , and

- $B \models \forall x(\alpha_d \rightarrow Ux)$  if and only if  $T \models \forall x(Lx \rightarrow (\alpha_d \rightarrow Ux))$ .

**Definition 12** For any queries  $q_B \subseteq \mathcal{B}$  and  $q_T \subseteq \mathcal{T}_k$ , define:

$$h_k(q_B) = \{T \mid B \triangleright T \text{ for some } B \in q_B\}$$

$$h_k^{-1}(q_T) = \{B \mid B \triangleright T \text{ for some } T \in q_T\}$$

We can define a syntactic translation of formulas that corresponds to the map  $h_k$ :

**Definition 13** Given a formula  $\phi$  in the language of binary strings, let  $\phi'$  be defined inductively as follows:

- if  $\phi$  is  $x = y$  then  $\phi'$  is  $\delta(x, y)$  where  $\delta$  is as defined in Axiom 6.
- if  $\phi$  is  $\neg\psi$  or  $\psi_1 \wedge \psi_2$  then  $\phi'$  is  $\neg\psi'$  or  $\psi'_1 \wedge \psi'_2$  respectively.
- if  $\phi$  is  $\exists x\psi$  then  $\phi'$  is  $\exists x(Lx \wedge \psi')$ .

Suppose for some formula  $\phi$  and some  $B \in \mathcal{B}$ ,  $B \models \phi[b_1 \dots b_n]$  and for some sequence of integers  $d_1 \dots d_n$ ,  $B \models \alpha_{d_i}[b_i]$ , i.e. the depths of the points  $b_i$  are given by the  $d_i$ . Also, let  $T \in \mathcal{T}_k$  be such that  $B \triangleright T$ . Then, if  $t_1 \dots t_n$  are any points in  $T$  such that  $T \models \alpha_{d_i}[t_i]$ , then  $T \models \phi'[b_1 \dots b_n]$ . This can be verified by an easy induction on the structure of the formula. One consequence of this is the following result.

**Lemma 9** If  $q \subseteq \mathcal{B}$  is a query in  $FO + LFP$  (respectively  $FO + PFP$ ), then  $h_k(q)$  is in  $FO + LFP$  (respectively  $FO + PFP$ ).

**Proof:**

Let  $\phi$  be the sentence that expresses  $q$  and let  $\chi$  be the conjunction of Axioms 1 through 10. Then,  $\phi' \wedge \chi$  expresses  $h_k(q)$ . ■

Another consequence is that if  $B \triangleright T$  then the closure ordinal of any occurrence of **fpf** (or **lfp**) in  $\phi$  over  $B$  is the same as the closure ordinal of the corresponding occurrence in  $\phi'$  over  $T$ . There may be additional inductions in  $\phi'$  which were introduced when we substituted the formula  $\delta$  for the identity, but all these are defined in **lfp**. Thus all inductions in  $\phi'$  close in a number of steps polynomial in the size of  $T$ . Moreover,  $\chi$  is defined in  $FO + LFP$ , so all inductions that occur there are also polynomial. We can now prove the following.

**Theorem 11 ([AV91b])**  $FO + PFP \upharpoonright P = FO + LFP$  if and only if  $PSPACE = P$ .

**Proof:**

One direction follows immediately from Theorem 9. In the other direction, suppose  $FO + PFP \upharpoonright P = FO + LFP$ . Let  $S$  be a language in  $PSPACE$  and hence in  $DTIME[2^{n^k}]$  for some  $k$ . Let  $q_B \subseteq \mathcal{B}$  be the collection of structures corresponding to strings in  $S$ . Since an ordering is easily (in  $FO + LFP$ ) definable on structures in  $\mathcal{B}$ ,  $q_B \in FO + PFP$ . Hence  $h_k(q_B) \in FO + PFP$ , by Lemma 9 and as we argued above, all inductions in the sentence expressing  $h_k(q_B)$  are polynomial in the size of  $T$ . Thus,  $h_k(q_B) \in FO + PFP \upharpoonright P$ . By hypothesis, then,  $h_k(q_B) \in FO + LFP$  and by an application of Lemma 8,  $q_B \in P$ . ■

This result is remarkable in that it reduces the separation of  $P$  and  $PSPACE$  to the separation of two classes that are properly contained in  $P$ .

## 9 $L_k$ Canonical Structures

In this section, we examine the question of whether the properties in the class  $L_{\infty\omega}^\omega \cap P$  are recursively indexable. Can we enumerate a set of Turing machines, for instance, each of which accepts a property in this class and such that every property in the class is accepted by some machine in the set. We know that the class  $FO + LFP$  is recursively indexable, since there is an effective way to construct, from a sentence of  $FO + LFP$ , a machine that accepts all models of the sentence. On the other hand, it is not known if the class  $P$ , of polynomial time computable queries, is recursively indexable.

Suppose we have a Turing machine  $C_k$ , for every  $k \in \mathbb{N}$ , which computes a function  $F_k$  of the input with the property that  $F_k(\mathfrak{A}) \equiv_k \mathfrak{A}$  and if  $\mathfrak{A} \equiv_k \mathfrak{B}$  then  $F_k(\mathfrak{A}) = F_k(\mathfrak{B})$ . We say that  $C_k$  computes an  $L_k$  canonical structure or an  $L_k$ -canon of its input. Suppose further that each of the  $C_k$  computes in polynomial time. If this is indeed the case, then the class  $L_{\infty\omega}^\omega \cap P$  is recursively indexable. To see this, consider an enumeration of all polynomial time Turing machines  $M_1, \dots, M_i, \dots$ . We can then enumerate all machines of the form  $C_k \rightarrow M_i$  which accepts input  $\mathfrak{A}$  if and only if  $M_i$  accepts  $F_k(\mathfrak{A})$ . This is an indexing of the class  $L_{\infty\omega}^\omega \cap P$ .

The situation is similar in the case of the class  $P$ . If we could canonically label a structure in polynomial time, then the class  $P$  would be recursively indexable. However, in this case,

even the problem of testing equivalence (*i.e.* the isomorphism problem) is not known to be in P. We can, however, test the equivalence of two structures under the relation  $\equiv_k$  in polynomial time. We can do this by computing the map  $E_k$  on the two structures and comparing the result. We show below that  $E_k(\mathfrak{A})$  and  $E_k(\mathfrak{B})$  are isomorphic just in case  $\mathfrak{A} \equiv_k \mathfrak{B}$ . Because  $E_k(\mathfrak{A})$  and  $E_k(\mathfrak{B})$  are ordered structures, if they are isomorphic, they are represented by identical bit-strings.

We now give the proof that the map  $E_k$  does indeed compute an  $L_k$ -invariant structure and that this computation can be done in polynomial time.

**Theorem 12** *For any two structures  $\mathfrak{A}$  and  $\mathfrak{B}$ ,  $\mathfrak{A} \equiv_k \mathfrak{B}$  if and only if  $E_k(\mathfrak{A}) \cong E_k(\mathfrak{B})$*

**Proof:**

$\Rightarrow$  If  $\mathfrak{A} \equiv_k \mathfrak{B}$  then every  $L_k$ -type that is realized in  $\mathfrak{A}$  is realized in  $\mathfrak{B}$  and vice versa.

To see this, let  $\bar{a}$  be a  $k$ -tuple from  $\mathfrak{A}$ . Recall from Corollary 6 that there is a formula  $\phi(x_1 \dots x_k)$  in  $L_k$  with  $k$  free variables that expresses this type. But then,  $\mathfrak{A} \models \exists x_1 \dots x_k \phi$  and therefore  $\mathfrak{B} \models \exists x_1 \dots x_k \phi$ . This tells us that the structures  $E_k(\mathfrak{A})$  and  $E_k(\mathfrak{B})$  have the same size.

Let  $f$  be the order-preserving map from  $E_k(\mathfrak{A})$  to  $E_k(\mathfrak{B})$ . If  $f([\bar{a}]) = [\bar{b}]$ , then  $\bar{a}$  and  $\bar{b}$  have the same  $L_k$ -type. This is because the definition of the ordering relation  $<_k$  is uniform, that is to say that the same types in different structures are ordered in the same way. As a result, the relations  $='$  and  $R_i$  are clearly preserved by  $f$ . Consider the case  $X_i([\bar{a}], [\bar{a}'])$ . Let  $\phi(x_1 \dots x_k)$  be the  $L_k$  formula expressing the  $L_k$ -type of  $\bar{a}'$ . Then,  $\exists x_i \phi$  is in the  $L_k$ -type of  $\bar{a}$  and hence of any element of  $f([\bar{a}])$ . It follows that  $X_i(f([\bar{a}]), f([\bar{a}']))$ . Similarly,  $f$  preserves the relation  $P_s$  for  $s = \langle i_1 \dots i_k \rangle$  because if  $\phi$  is a formula in  $\text{Type}_k(\langle a_1 \dots a_k \rangle)$ , then the formula  $\phi_s$  obtained by substituting every free occurrence of every  $x_j$  with  $x_{i_j}$ , with the appropriate renaming of bound variables, is in  $\text{Type}_k(\langle a_{i_1} \dots a_{i_k} \rangle)$ . Thus,  $f$  is an isomorphism.

$\Leftarrow$  Let  $f$  be an isomorphism from  $E_k(\mathfrak{A})$  to  $E_k(\mathfrak{B})$ . We show that Player II has a strategy for playing the  $k$ -pebble game on  $\mathfrak{A}$  and  $\mathfrak{B}$  indefinitely. Suppose that at some stage of the game, the pebbles are on the elements  $\bar{a}$  and  $\bar{b}$ . (We assume that all  $k$  pairs of pebbles are on the board. If not, then just consider any extension of these tuples.) Further suppose, without loss of generality, that Player I moves on  $\mathfrak{A}$  resulting in the



configuration  $\bar{a}'$ . Player II finds a tuple  $\bar{b}' \in f([\bar{a}'])$  such that  $\bar{b}'$  is one move away from  $\bar{b}$  and then plays that move. We need to show that such a  $\bar{b}'$  can always be found. Note that we can assume, as an inductive hypothesis that  $f([\bar{a}]) = [\bar{b}]$ . Suppose that Player I moves the pebble from  $a_i$  to a new element, then  $X_i([\bar{a}], [\bar{a}'])$  holds. Because  $f$  is an isomorphism,  $X_i(f([\bar{a}]), f([\bar{a}']))$  holds and we can get from  $\bar{b}$  to some  $\bar{b}' \in f([\bar{a}'])$  by moving the pebble on  $b_i$  ■

**Theorem 13** *The map  $E_k$  is computable on all structures  $\mathfrak{A}$  in time polynomial in the size of the structure  $\mathfrak{A}$ .*

**Proof:**

The number of tuples in  $A^k$  is  $n^k$  where  $n = |A|$ . The equivalence relation  $\sim_k$  is defined by an FO + LFP formula and hence computable in polynomial time as is the ordering  $<_k$ . We can get, therefore, in polynomial time, a representation of the universe of  $E_k(\mathfrak{A})$ . All the other relations are easily defined on  $\mathfrak{A}$  (in FO). ■

The most direct approach to constructing an  $L_k$ -canon, given a polynomial time algorithm for the translation  $E_k$ , would be to try and invert  $E_k$ , *i.e.* given an input structure  $\mathcal{K}$ , to find an  $\mathfrak{A}$  such that  $\mathcal{K} = E_k(\mathfrak{A})$ . However, this cannot be done in time polynomial in the size of  $\mathcal{K}$ . To see this, suppose for contradiction that we have a polynomial time computable  $E_k^{-1}$  which acts as a translation from the range of  $E_k$  into its domain. Since the range of  $E_k$  consists of totally ordered structures,  $E_k^{-1}$  is definable in FO + LFP. Composing this with the FO + LFP definition of  $E_k$ , we get an FO + LFP translation that yields an  $L_k$  canon, and therefore that  $L_{\infty\omega}^\omega \cap P \subseteq \text{FO} + \text{LFP}$ , which we know to be false. It is still conceivable that the computation of  $E_k^{-1}$ , while not polynomial in the size of the input  $E_k(\mathfrak{A})$  is polynomial in the size of  $\mathfrak{A}$ , since the former could be much smaller. In fact, it is exactly the case where  $E_k(\mathfrak{A})$  is much smaller than  $\mathfrak{A}$  that demonstrated that  $\text{FO} + \text{LFP} \neq L_{\infty\omega}^\omega \cap P$ .

## References

- [AV91a] S. Abiteboul and V. Vianu. Datalog extensions for database queries and updates. *Journal of Computer and System Sciences*, 43:62–124, 1991.
- [AV91b] S. Abiteboul and V. Vianu. Generic computation and its complexity. In *Proceedings of the 23rd ACM Symposium on the Theory of Computing*, 1991.

- [Bar73] J. Barwise. Back and forth through infinitary logic. In M. D. Morley, editor, *Studies in Model Theory*, Mathematical Association of America, 1973.
- [Bar77] J. Barwise. On Moschovakis closure ordinals. *Journal of Symbolic Logic*, 42:292–296, 1977.
- [GS86] Y. Gurevich and S. Shelah. Fixed-point extensions of first-order logic. *Annals of Pure and Applied Logic*, 32:265–180, 1986.
- [HS65] J. Hartmanis and R. E. Stearns. On the computational complexity of algorithms. *Transactions of the AMS*, 117:285–306, 1965.
- [IL90] N. Immerman and E. S. Lander. Describing graphs: a first-order approach to graph canonization. In A. Selman, editor, *Complexity Theory Retrospective*, Springer-Verlag, 1990.
- [Imm86] N. Immerman. Relational queries computable in polynomial time. *Information and Control*, 68:86–104, 1986.
- [KV90] Ph. G. Kolaitis and M. Y. Vardi. 0-1 laws for infinitary logics. In *Proceedings of the 5th IEEE Symposium on Logic in Computer Science*, pages 156–167, 1990.
- [Lin91] S. Lindell. An analysis of fixed-point queries on binary trees. *Theoretical Computer Science*, 85(1):75–95, 1991.
- [Var82] M. Y. Vardi. The complexity of relational query languages. In *Proceedings of the 14th ACM Symposium on the Theory of Computing*, pages 137–146, 1982.