Fixed Point vs. First-Order Logic on Finite Ordered Structures with Unary Relations

A. J. Kfoury*   M. Wymann-Böni†

August 11, 1993

Abstract

We prove that first order logic is strictly weaker than fixed point logic over *every* infinite classes of finite ordered structures with unary relations: Over these classes there is always an inductive unary relation which cannot be defined by a first-order formula, even when every inductive sentence (i.e., closed formula) can be expressed in first-order over this particular class.

Our proof first establishes a property valid for every unary relation definable by first-order logic over these classes which is peculiar to classes of ordered structures with unary relations. In a second step we show that this property itself can be expressed in fixed point logic and can be used to construct a non-elementary unary relation.

1 Introduction

In this paper we are concerned with a questions about finite structures for a signature $\Sigma = \{\leq, R_1, \ldots, R_l\}$, where $\leq$ has to be realized as a total order and the predicate symbols R_j are all unary.

---

*Partly supported by NSF grant CCR-9113196.

†Supported by a grant from the Swiss National Science Foundation.

Finite Model Theory has become of interest partly because it is a mathematical model for databases. It covers those aspects of database theory dealing with the expressive power of query languages. Nowadays, the expressiveness of first-order logic using the usual connectives and quantifiers seems to be the absolute minimum a query language has to offer. Yet, on finite structures, first-order logic is not very expressive in many cases. For this reason, one has started to add constructs intended to reflect more closely the computable queries on a database. (See Aho and Ullman[1], Chandra and Harel [2], Immerman [9], Ioannides [11], Kanellakis and Abiteboul [12], Naughton [16], Vardi [17] and many more.)

Another aspect of finite models is that various logics over finite (typically ordered) structures can describe complexity classes; the first result of this nature was Fagin's "$NP = \Sigma_1^1$" in his celebrated paper [5]. Since then it has become possible to find an appropriate logic for every imaginable complexity class. (See Gurevich [7], Immerman [9, 10].)

Most often, the question of whether one logic is more expressive than another is studied by considering only sentences (i.e., closed formulae) of the respective logics. This can be done by considering the model classes $\mathrm{Mod}(\varphi)$ of all structures for the signature in which the sentence φ is valid, and finding a sentence in one logic which defines a model class which is not definable by any sentence of the other logic. This approach, however, is not sufficient if we consider *arbitrary* classes of finite structures for a signature Σ. Here it can happen that the logics are very different, yet all sentences of one logic have equivalents in the other. Usually the reason for pathological behavior like this stems from the choice of a particular class. In fact, given any logic with a countable set of sentences, it is possible to find a subclass in every infinite class that allows every property of this logic to be expressed by a first-order sentence. Furthermore, the mentioned subclass can be chosen to be still infinite. For details see McColm [14, Thm 6.5]. For all these reasons, we consider the question of expressiveness of different logics with respect to arbitrary formulae not only sentences.

In this article we compare fixed point logic (or induction) and first-order logic on finite ordered structures with additional unary relations. Our goal is to prove:

Theorem 1.1 *Given a signature $\Sigma = \{\leq, R_1, \ldots, R_l\}$ consisting of order and* unary *relations $R_1, \ldots, R_l$, and an arbitrary infinite class $\mathcal{C}$ of finite* *structures for this language, there are inductive relations on $\mathcal{C}$ which are not elementary over $\mathcal{C}$.*

We understand here and everywhere in this article that an infinite class is a class with infinitely many mutually non-isomorphic structures.

Notation 1.2 We use short-hand conventions in writing formulae: E.g., "$x = 0$" stands for the clumsier $(\forall y)(x \leq y)$; similarly "$R_j(6)$" stands for $(\exists x)(R_j x \wedge x = 6)$, where the "$x = 6$" is a short form for a formula expressing that x is the seventh element with respect to the order $\leq$. Also, once we have established that a relation is definable, we immediately address it by its name instead of its definition. In the presence of a total order every element of the structure is first-order definable; thus, the symbols 0, 1, 2,... have meaning even in the absence of constants, as is the case here. Also, we freely use the notation "$x = y + 1$" to describe that y is the immediate successor (with respect to the given order) of x, again a relation which is easily expressed as a first-order formula.

2 Fixed Point Logic

We assume that the reader is already familiar with the basic concepts of first-order logic as the definition of formulae and how the notion of truth is defined. Our standard reference is "Model Theory" by Chang and Keisler [3]. We follow their notations as closely as possible.

For considerations about fixed point logic, or equivalently, inductive relations, the standard is given by Moschovakis' book "Elementary Induction on Abstract Structures" [15], or for its application to classes of finite structures rather than a single infinite structure the two articles by McColm [13, 14]. However, the following should be sufficient for the scope of this article.

Fixed point logic is an extension of first-order logic designed to reflect the power of induction. There are several formalizations which are not in general equivalent, but the differences are of no concern to us. This is also justified by the results of Gurevich and Shelah [8] stating that many different definitions of fixed point logic coincide for finite structures. We use "fixed point logic" and "induction" as synonyms.

We choose to follow here the definition of McColm [13] inspired by Moschovakis [15], using what they call *positive elementary induction.*

To start with, we need a system of positive elementary (i.e, first-order) formulae

$$\varphi_0(x_1,\ldots,x_{k_1};S_0,\ldots,S_n),\ldots,\varphi_n(x_1,\ldots,x_{k_n};S_0,\ldots,S_n).$$

Here the S_i are new "reserved" relation symbols that do not occur in the first order language of the structure and occur only in positive parts of the formulae φ_j. If the arity of S_i is k_i, for $i=0,\ldots,n$ we can associate an operator Φ_i on relations with each of the formulae φ_i:

$$\Phi_i(X_0,\ldots,X_n)=\{(a_1,\ldots,a_{k_i}):\mathfrak{A}\models\varphi_i[a_1,\ldots,a_{k_i};X_1,\ldots X_n]\}$$

The simultaneous fixed point of the system is then given by:

$$\varphi_i^0 \quad = \quad \Phi_i(\emptyset,\ldots,\emptyset)$$

and

$$\varphi_i^{m+1} \quad = \quad \Phi_i(\varphi_0^m,\ldots,\varphi_n^m).$$

Since we assume that the relation symbols S_i occur only positively in the formulae φ_i, the operators Φ_i are monotone, and the process eventually reaches the least fixed point of the system. We say that this system defines the fixed point, or the inductive relation

$$\varphi_0^\infty \quad = \quad \bigcup_{m\in\omega}\varphi_0^m.$$

Fixed point logic now contains along with all first order formulae new formulae denoting these fixed points. Because of the clumsiness of available notation we abstain from formally introducing new language constructs for these fixed points and address them always via the corresponding system of positive elementary formulae.

Fact 2.1 *Fixed point logic on finite structures as defined above is closed under all first order operations; that is, if there are fixed point relations* φ^∞ *and* ψ^∞*, then all of* $\varphi^\infty\vee\psi^\infty$*,* $\neg\varphi^\infty$*, and* $(\exists x)\varphi^\infty$ *are again expressible as fixed points of a system of positive elementary formulae.*

These facts are proved by Moschovakis [15] and McColm [13] for closure under positive operations (for the one infinite structure and for a class of finite structures respectively), and by Immerman [9] for negation. (Note, however, that for infinite models the closure under negation is not given in general. It is only true for quite pathological structures.)

We will freely apply these facts whenever convenient.

The expressiveness of fixed point logic on classes of finite structures was extensively investigated by McColm [13, 14]. It is there that he states the conjecture which was the motivation for our work.

Conjecture 2.2 (McColm) *If a class of finite structures $\mathcal{C}$ allows an unbounded induction, (i.e., there is an inductive relation φ^∞, such that for every m, we can find a structure $\mathfrak{A} \in \mathcal{C}$ with $\mathfrak{A} \models \varphi^\infty \neq \varphi^m$), then there exists an inductive but non-elementary relation on this class.*

He also proved this under some restrictions:

Theorem 2.3 (McColm) *If an infinite class $\mathcal{C}$ of finite structures is recursively enumerable and has an almost-complete*[1] *first order theory, i.e. every first order sentence is true on either finitely many, or co-finitely many structures of $\mathcal{C}$, then $\mathcal{C}$ admits an inductive, but non-elementary relation iff it admits an unbounded induction.*

To prove this theorem, McColm employs parametrization techniques for inductive relations that are definable with a fixed number of variables. This bound is necessary because of some peculiarities of finite structures: Roughly speaking, by admitting an unbounded induction the whole class has a potential of computation similar to that of induction on the integers, but it is not easily accessible, because every single structure has only finitely many elements and coding is difficult. To make the necessary room for a computation using a form of codes, one has to introduce more variables, but this has to be a fixed number that cannot be adapted after the inductive property is defined. (This also corresponds to the fact that all inductive definitions on finite structures can be computed in *PTIME*, see Immerman [9].) For details of this approach see McColm [14].

We prove a similar theorem under different conditions. Our method of proof is partly inspired by the parametrization idea of McColm. However,

[1]This is a modification of the usual notion of completeness of a theory for use with classes of finite structures which is necessary because the original notion of completeness is to restrictive and rules out infinite classes of finite structures.

in our situation a bound on the number of variables used would be inappropriate, and would necessitate additional undesired premises for our main theorem. To compensate for this, we have to generate an extra bit of knowledge using simple pebble games, which is peculiar to the situation where we only have order plus additional *unary* relations in the language.

In our setting, thanks to the presence of order, the unboundedness of fixed point logic is equivalent to saying that the class is infinite. From this we get the formulation of our theorem as stated in the Introduction.

3 Logical Games

One of the most successful techniques for the treatment of classes of finite models is the use of games to describe logical properties of structures. We use only the simplest of all the variants, the so-called Ehrenfeucht-Fraïssé games (see Ehrenfeucht [4], or Fraïssé [6]).

Definition 3.1 *Given the signature $\Sigma = \{R_1, \ldots, R_l; c_1, \ldots, c_m\}$ with only finitely many relation and constant symbols, and no function symbols, define the relations $\equiv_k$ between structures for Σ:*

- $\mathfrak{A} \equiv_0 \mathfrak{B}$, *iff the submodels of $\mathfrak{A}$ and $\mathfrak{B}$ generated by the constants are isomorphic, or else Σ has no constant symbols at all.*
- $\mathfrak{A} \equiv_{k+1} \mathfrak{B}$, *iff for every element $a \in A$ there is an element $b \in B$ such that $(\mathfrak{A}, a) \equiv_k (\mathfrak{B}, b)$, and for every $b \in B$ there is an $a \in A$ such that $(\mathfrak{A}, a) \equiv_k (\mathfrak{B}, b)$.*

Before we proceed to describe some properties of the relation $\equiv_k$, we give a natural way to describe it in terms of two-person games as they were introduced by Fraïssé and Ehrenfeucht.

We describe the notion of an k-move game with perfect information between two players on two structures $\mathfrak{A}$ and $\mathfrak{B}$ for a finite signature Σ without function symbols:

The game is played by two players, the *spoiler* and the *duplicator*[2]. In every round of the game the spoiler first chooses one of the structures $\mathfrak{A}$

[2]Often the player are just referred to as *player I* and *player II*. The new names are meant to reflect their goals in the game in a more intuitive fashion, and, to our knowledge, were introduced by Joel Spencer.

or $\mathfrak{B}$ and marks an element in the structure of his choice. The duplicator responds by choosing an element in the other structure. The k-move game finishes after the k-th response by the duplicator.

At the end, we determine who has won the game as follows: In every move a pair of elements (a_i, b_i) from $\mathfrak{A}$ and $\mathfrak{B}$ has been produced by the choices of the two players. We check whether the map given by $a_i \mapsto b_i$ (and sending every distinguished constant of $\mathfrak{A}$ to the corresponding distinguished element in $\mathfrak{B}$) is an isomorphism between the submodels of $\mathfrak{A}$ and $\mathfrak{B}$ generated by the constants and all the played elements a_i and b_i. If so, the duplicator wins, otherwise, in case this is not a partial isomorphism, the spoiler wins the game.

We say that a player has a winning strategy for the k-move game on $\mathfrak{A}$ and $\mathfrak{B}$ iff he has a method of choosing elements to ensure non-isomorphism after k moves in the case of the spoiler, or a way of responding to maintain isomorphism until the end of the game in the case of the duplicator, no matter what the moves of the adversary player are.

It is not difficult to see that the previously defined relation $\equiv_k$ describes exactly the existence of a winning strategy for the duplicator for the k-move game.

We state some basic properties of the relation $\equiv_k$ that we use later. (For details and proofs, see Ehrenfeucht [4].)

Theorem 3.2 (Ehrenfeucht)

1. *For all k, the relation $\equiv_k$ is an equivalence relation.*

2. *All relations $\equiv_k$ have finite index, i.e there are finitely many sentences $\chi_1, \ldots, \chi_{n(k)}$ with the following property: Every model satisfies exactly one of these sentences, and if two models $\mathfrak{A}$ and $\mathfrak{B}$ satisfy the same sentence then $\mathfrak{A} \equiv_k \mathfrak{B}$.*

3. *If $\mathfrak{A} \equiv_k \mathfrak{B}$, then $\mathfrak{A}$ and $\mathfrak{B}$ satisfy the same sentences of quantifier-depth at most k.*

4. *Two finite models $\mathfrak{A}$ and $\mathfrak{B}$ are isomorphic iff for all $k \in \omega$: $\mathfrak{A} \equiv_k \mathfrak{B}$.*

4 Non-Elementary Relations

Before we start our work to construct an example of a relation which must lie outside first order, but is inductive, we have to establish a criterion by which we can recognize the non-definability of a relation by a first-order formula.

Definition 4.1 *Two elements a, $a' \in A$ are k-*indiscernible *in a structure $\mathfrak{A}$ iff $(\mathfrak{A}, a) \equiv_k (\mathfrak{A}, a')$. Similarly, two sequences $(a_1, \ldots, a_m)$ and $(a'_1, \ldots, a'_m)$ of elements from A are k-indiscernible iff $(\mathfrak{A}, a_1, \ldots, a_m) \equiv_k (\mathfrak{A}, a'_1, \ldots, a'_m)$.*

Consequence 4.2 *Two elements a, a' are k-*indiscernible *in a structure $\mathfrak{A}$ iff no formula $\varphi(x)$ of quantifier-depth at most k can distinguish between the two elements. More formally: For all formulae φ with quantifier-depth at most k, we have*

$$\mathfrak{A} \models \varphi[a] \iff \mathfrak{A} \models \varphi[a']$$

Proof: This is an immediate consequence of Theorem 3.2.
□

Since every elementary subset, or equivalently, every unary first-order formula has a certain bound associated with it — its quantifier depth —, we can easily see that no first order relation can distinguish between k-indiscernible elements of a structure for arbitrary k. The discerning power of every formula is limited to its own quantifier-depth.

In the case of order plus unary relations the following key fact is available. We write $[0, n] \subseteq \mathfrak{A}$ for the uniquely determined submodel of $\mathfrak{A}$ that contains the least $n + 1$ elements (w.r.t the order $\leq$) of $\mathfrak{A}$. Note that this also implies that the additional unary relation on $[0, n]$ are the restrictions of the corresponding relations in $\mathfrak{A}$.

Lemma 4.3 (Key Lemma) *If two elements a, a' are k-indiscernible in an initial segment $[0, n]$ of a model $\mathfrak{A}$, then a and a' are also k-indiscernible in the whole model $\mathfrak{A}$. I.e.,*

$$\textit{If } a \simeq_k a' \textit{ in } [0, n] \subseteq \mathfrak{A}, \textit{ then } a \simeq_k a' \textit{ in } \mathfrak{A}.$$

Proof: We have to describe a winning strategy for the duplicator in the k-move game on the structures $(\mathfrak{A}, a_0)$ and $(\mathfrak{A}, a'_0)$ based on the knowledge that the duplicator has such a strategy for games on $[0, n] \subseteq \mathfrak{A}$. We choose the new names a_0 and a'_0 to denote a and a' for easier indexing.

In this strategy the duplicator responds to a move of the spoiler either by mimicking the choice in case the latter has chosen an element outside $[0, n]$, or by playing the response he would have played in a game on the interval $[0, n]$ if the spoiler chooses an element within this range (here the duplicator can ignore all elements played outside $[0, n]$ for his decision). More precisely: Let the elements played so far in the game form the pairs (a_i, a'_i) from $[0, n]$, and (b_j, b'_j) outside $[0, n]$. The duplicator responds with the identity function to the next element outside $[0, n]$ played by the spoiler, and to respond to a move of the spoiler inside $[0, n]$ he uses his strategy for the corresponding game on $[0, n]$, ignoring the pairs (b_j, b'_j).

We have to verify that this yields indeed a winning strategy for the duplicator in games on the models $(\mathfrak{A}, a_0)$ and $(\mathfrak{A}, a'_0)$.

Recall that the duplicator is "alive" after a certain number l of moves if the mapping given by $a_i \mapsto a'_i$ and $b_j \mapsto b'_j$ is a partial isomorphism. To verify this we have to check that all the atomic sentences which can be formed using the elements played so far preserve their validity under the above map.

This is the point where the restriction to unary relations comes into play: The only atomic sentences we need to consider are of the forms $a_{i_1} \leq a_{i_2}$, $b_{j_1} \leq b_{j_2}$, $a_{i_1} \leq b_{j_2}$, $b_{j_1} \leq a_{i_2}$, or Ra_i, Rb_j for every additional unary relation symbol. Note also, that the elements b_j are always greater than the elements a_i.

Obviously, the statement of the lemma is trivially true for games of length 0. Assuming its validity for games of length k, we show that it holds for $(k+1)$-games.

From the assumption we know that the duplicator has winning strategies for $(k+1)$ moves on $[0, n]$ and for k moves on $\mathfrak{A}$.

If the spoiler plays an element b outside $[0, n]$ in his $(k+1)$-th move it is obvious that the duplicator wins by playing the same element $b' = b$ on the other structure, since none of the newly created atomic formulae distinguish between b and b'.

On the other hand if the spoiler chooses an element a_i from $[0, n]$, then the response a'_i of the duplicator is good enough if we consider only pairs of elements played in $[0, n]$. But then all the atomic sentences preserve truth under the mapping induced by $a_i \mapsto a'_i$ and $b_j \mapsto b'_j$.
□

Intuitively, this means that once we have established indiscernibility of two elements considering an initial segment $[0, n]$ as the universe, this judgement stays correct also for the original larger structure.

This lemma is the key to all that follows. It is however not valid in general without the restriction to *unary* relations in the signature.

Now, let us find a relation that cannot be elementary. To achieve this, we observe that by virtue of Theorem 3.2 every elementary unary relation respects the equivalence classes formed by k-indiscernible elements starting from some k and for all bigger $k' > k$. We would like to stress that this behavior is uniform on every structure $\mathfrak{A}$ of the class $\mathcal{C}$. That is, for all structures $\mathfrak{A} \in \mathcal{C}$: $a \simeq_k a'$ in $\mathfrak{A}$ implies $\mathfrak{A} \models \varphi[a] \leftrightarrow \mathfrak{A} \models \varphi[a']$.

We want to construct a unary relation that splits pairs of k-indiscernible elements of every degree k in some model $\mathfrak{A}$. That is, for every k there should be a structure $\mathfrak{A}$ in our class and two k-indiscernibles a and a' such that one of these elements is in the relation and the other is not. From this we would conclude that the latter relation cannot have a first-order definition of quantifier depth less or equal to k, for every k. Thus, it would not be definable at all in first-order.

Assume for the moment that we have access to a relation $Ind(k, n, x, y)$ which computes k-indiscernibles (for some values k and n which depend on the size of $\mathfrak{A}$) and also, that we know a relation $Arith(k, n)$ that guarantees that the model $\mathfrak{A}$ is large enough to accommodate the values for k and n.

$$\mathfrak{A} \models Arith[k, n] \rightarrow (Ind[k, n, x, y] \leftrightarrow x \simeq_k y \text{ in } [0, n])$$

After having defined these relations it will be easy to finish our programme:

Theorem 4.4 *Over every infinite class $\mathcal{C}$ of finite ordered structures with finitely many unary predicates, there is a unary relation that is non-elementary yet inductive.*

Proof: We proceed under the assumption that $Ind(k, n, x, y)$ and $Arith(k, n)$ have already been proved to be inductive.

Now, we can easily give a definition for a set that does not respect the equivalence classes of k-indiscernible elements for any k, provided that we have arbitrarily large models in the class $\mathcal{C}$:

$$
\begin{aligned}
&\psi(x) \equiv \\
&(\exists k)(\exists n)\Big(\; Arith(k,n) \\
&\qquad \wedge (\exists y)\Big[Ind(k,n,x,y) \wedge x \neq y \wedge (\forall z)(Ind(k,n,x,z) \to x \leq z)\Big] \\
&\qquad \wedge (\forall k' > k)(\forall n')\Big[Arith(k',n') \to (\forall x)(\forall y)(Ind(k',n',x,y) \to x = y)\Big]\Big)
\end{aligned}
$$

In English this reads: "Take the maximal number k and an appropriate n such that

- there is a non-trivial class of k-indiscernibles on $[0,n] \subseteq \mathfrak{A}$,
- the values of k and n are appropriately chosen in relation to $\|\mathfrak{A}\|$. (This will be realized as: All codes used in the evaluation of $Ind(n,k,x,y)$ in fixed point logic are available inside $\mathfrak{A}$.)

Then extract the minimal elements of the computed indiscernibility classes for this k and n into the set defined by $\psi(x)$."

It is easy to see that ψ does not respect k-indiscernibility classes for any k if there are arbitrarily large models in the class $\mathcal{C}$.
$\square$

5 Coding the Indiscernibility Relation

The goal of this section is to develop an inductive definition for the relations Ind and $Arith$ postulated earlier. The main task is to define Ind such that

$$\mathfrak{A} \models Ind(n,k,a,b) \quad \Longleftrightarrow \quad \mathfrak{A} \models a \simeq_k b \text{ in } [0,n] \subseteq \mathfrak{A}$$

provided that the size $\|\mathfrak{A}\|$ is large in comparison with k and n in a sense still to be made precise. From the definition we will see how to define the predicate $Arith$ in order to guarantee correctness.

Our model for this definition is the inductive definition of the winning situation for the duplicator in the Ehrenfeucht-Fraïssé game with k moves. Recall the definition of k-indiscernible elements (now we use sequences of element rather than single elements).

$$
\begin{aligned}
&(a_1,\ldots,a_m) \simeq_0 (b_1,\ldots,b_m) \quad \Longleftrightarrow \quad a_i \mapsto b_i \text{ induces a partial isomorphism} \\
&(a_1,\ldots,a_m) \simeq_{k+1} (b_1,\ldots,b_m) \Longleftrightarrow \\
&\qquad\qquad (\forall a_{m+1})(\exists b_{m+1})((a_1,\ldots,a_{m+1}) \simeq_{k+1} (b_1,\ldots,b_{m+1})) \\
&\qquad\qquad \wedge (\forall b_{m+1})(\exists a_{m+1})((a_1,\ldots,a_{m+1}) \simeq_{k+1} (b_1,\ldots,b_{m+1}))
\end{aligned}
$$

To implement this scheme we first define relations that define formation of pairs and sequences of elements.

Lemma 5.1 *There are inductive pairing relations. (For definiteness we choose the pairing $\langle x, y\rangle = (x+y)(x+y+1)/2+y$.) More precise: There is a relation "Pair" with the property:*

$$Pair(e, x, y) \iff e = \langle x, y\rangle.$$

Proof: Define:

$$\begin{aligned}\varphi(e, x, y; S) \equiv\ & (e = 0 \land x = 0 \land y = 0) \\ & \lor(e > 0 \land y > 0 \land S(e-1, x+1, y-1)) \\ & \lor(e > 0 \land y = 0 \land x > 0 \land S(e-1, 0, x-1))\end{aligned}$$

The fixed point φ^∞ satisfies $\varphi^\infty(e, x, y) \iff e = \langle x, y\rangle$.
□

The corresponding projections are simply given by the formulae $\langle e\rangle_1 = x \equiv (\exists y)\, Pair(e, x, y)$ and $\langle e\rangle_2 = y \equiv (\exists x)\, Pair(e, x, y)$. For the next step, define m-tuples of elements, by $\langle x_1, \ldots, x_m\rangle = \langle x_1, \langle x_2, \ldots, x_m\rangle\rangle$. We need decomposition of m-tuples.

Lemma 5.2 *There is an inductive relation "Tail" with the property:*

$$Tail(i, e, f) \iff e = \langle x_1, \ldots, x_m\rangle \land i \leq m \land f = \langle x_i, \ldots, x_m\rangle,$$

Proof: Define:

$$\varphi(i, e, f; S) \equiv (i = 1 \land e = f) \lor (i > 1 \land S(i-1, \langle e\rangle_2, f))$$

Take *Tail* to be $\psi^\infty(i, e, f)$
□

We need still another construct for tuples which allows us to access the i-th element rather than the i-th tail of a tuple and which gives us the possibility to recognize the length of the tuple.

$$(x_1, \ldots, x_m) = \langle m, \langle x_1, \ldots, x_m\rangle\rangle$$

It is clear that the corresponding projections $(x_1, \ldots, x_m)_i = x_i$, as well as the length $lh((x_1, \ldots, x_m)) = m$ are inductive.

The next step consists in defining a relation that checks for partial isomorphism between sequences of elements.

Lemma 5.3 *There is an inductive relation "Iso(e, f)" such that*

$$\mathfrak{A} \models \mathit{Iso}(e,f) \iff \begin{array}{l} e \text{ codes } (a_1,\ldots,a_m),\ f \text{ codes } (b_1,\ldots,b_m), \text{ and} \\ a_i \mapsto b_i \text{ induces a partial isomorphism.} \end{array}$$

Proof: Assume the signature to be $\Sigma = \{\leq, R_1, \ldots, R_l\}$. To check a partial isomorphism between two sequences of elements we have basically to make sure that all the atomic sentences which can be formed using the elements of the sequences as constants preserve their value under the map $a_i \mapsto b_i$. This is reflected by the following definition:

$$\begin{aligned} \mathit{Iso}(e,f) \equiv (\exists m)\Big(\ & lh(e) = lh(f) = m \\ & \wedge (\forall i)(\forall j)(1 \leq i < j \leq m \rightarrow ((e)_i < (e)_j \leftrightarrow (f)_i < (f)_j)) \\ & \wedge (\forall i)(\forall j)(1 \leq i < j \leq m \rightarrow ((e)_i = (e)_j \leftrightarrow (f)_i = (f)_j)) \\ & \wedge (\forall i)(1 \leq i \leq m \rightarrow (R_1(e)_i \leftrightarrow R_1(f)_i)) \\ & \wedge \ldots \\ & \wedge (\forall i)(1 \leq i \leq m \rightarrow (R_l(e)_i \leftrightarrow R_l(f)_i))\Big) \end{aligned}$$

□

Note, that this relation yields correct results for all codes which are representable in a particular model $\mathfrak{A}$, since all computation is done by looking at parts of already given codes and no new codes are generated.

Now, we can proceed to the inductive step in the definition of indiscernible sequences.

Proposition 5.4 *There is an inductive relation "Ind(n, k, e, f)" such that*

$$\mathfrak{A} \models \mathit{Ind}(n,k,e,f) \iff \begin{array}{l} e \text{ codes } (a_1,\ldots,a_m),\ f \text{ codes } (b_1,\ldots,b_m), \text{ all the} \\ \text{elements } a_i,\ b_i \text{ are in } [0,n], \text{ and } (a_1,\ldots,a_m) \simeq_k \\ (b_1,\ldots,b_m) \text{ in } [0,n] \subseteq \mathfrak{A} \end{array}$$

whenever n, k, e, and f are small in comparison to $\|\mathfrak{A}\|$. (For the exact meaning of the latter see below.)

Proof: We use the following inductive definition: Set

$$\begin{aligned} \varphi(n,k,e,f;S) \equiv\ & (k = 0 \wedge \mathit{Iso}(e,f)) \\ & \vee\Big(k > 0 \wedge (\forall a \leq n)(\exists b \leq n) S(n, k-1, e \star a, f \star b) \\ & \qquad \wedge (\forall b \leq n)(\exists a \leq n) S(n, k-1, e \star a, f \star b) \end{aligned}$$

Here, we omitted the check that all elements of e and f are in $[0, n]$ for better readability. Also, $e \star a$ denotes the sequence e with the element a appended to its end; this process is clearly definable in fixed point logic. Now take $Ind \equiv \varphi^\infty$.
$\square$

A note concerning the necessary size of the model $\mathfrak{A}$ to make this work is in order. If e and f are sequences of elements of length m then for a particular k we need sequences up to length $k+m$. Since all elements occurring in the sequences are in $[0, n]$ the model is certainly large enough for this relation to work if the sequence $(n, \ldots, n)$ (with $k+m$ repetitions of the element n) can be represented. From this we get the definition of the earlier mentioned predicate $Arith$.

$$Arith(n, k) \equiv (\exists e)(lh(e) = k + 1 \wedge (\forall i)(1 \leq i \leq lh(e) \rightarrow (e)_i = n))$$

It is clear from the above, that $Arith$ and Ind work together as promised earlier.

Proposition 5.5 *It is true in all models $\mathfrak{A}$ that*

$$\mathfrak{A} \models Arith(n, k) \rightarrow (Ind(n, k, (a), (b)) \leftrightarrow a \simeq_k b \textit{ in } [0, n])$$

This concludes the proof that there is a non-elementary fixed point relation on every infinite class of ordered structures with monadic relations.

References

[1] A. V. Aho and J. D. Ullman, *Universality of data retrieval languages*, Proc. 6th ACM Symp. on Principles of Programming Languages, 1979, pp. 110–117.

[2] A. Chandra and D. Harel, *Structure and complexity of relational queries*, Journal of Computer and System Sciences **25** (1982), 99–128.

[3] C. C. Chang and Keisler H. J., *Model theory*, 3rd ed., Studies in Logic, vol. 73, North-Holland, 1990.

[4] A. Ehrenfeucht, *An application of games to the completeness problem for formalized theories*, Fundamentæ Mathematicæ **49** (1961), 129–141.

[5] R. Fagin, *Monadic generalized spectra*, Zeitschrift für mathematische Logik und Grundlagen der Mathematik **21** (1975), 89–96.

[6] R. Fraïssé, *Sur quelques classifications des systèmes de relations*, Publ. Sci. Univ. Alger Ser. A **1** (1954), 35–185.

[7] Y. Gurevich, *Towards a logic tailored for computational complexity*, Computation and Proof Theory, LNCS, vol. 1104, Springer, 1984, pp. 175–216.

[8] Y. Gurevich and S. Shelah, *Fixed-point extensions of first-order logic*, Annals of Pure and Applied Logic **32** (1986), 265–280.

[9] N. Immerman, *Relational queries computable in polynomial time*, Information and Control **68** (1986), 76–98.

[10] ———, *Languages that capture complexity classes*, SIAM J. Comput. **16** (1987), 760–778.

[11] Y. E. Ioannides, *Bounded recursion in deductive databases*, Algorithmica **1** (1986), 86–104.

[12] P. Kanellakis and S. Abiteboul, *Deciding bounded recursion in database logic programs*, SIGAct News **20** (1989), no. 4.

[13] G. L. McColm, *Parametrization over inductive relations of abounded number of variables*, Ann. of Pure and Applied Logic **48** (1990), 103–134.

[14] ———, *When is arithmetic possible?*, Ann. of Pure and Applied Logic **50** (1990), 29–51.

[15] Y. N. Moschovakis, *Elementary induction on abstract structures*, North Holland, 1974.

[16] J. F. Naughton, *Data independent recursion in deductive databases*, Journal of Computer and System Sciences **38** (1989), 259–289.

[17] M. Y. Vardi, *The complexity of relational query languages*, Proc. 14th ACM Symp. on Theory of Computing, 1982, pp. 137–146.