Normalizable Linear Orders and Generic Computations in Finite Models

Alexei P. Stolboushkin*
UCLA Department of Mathematics
and
Fourth Dimension Software
555 Twin Dolphin Dr.
Redwood City, CA 94065-2102
aps@4ds.com, aps@math.ucla.edu

Michael A. Taitslin†
Department of Computer Science
Tver State University
Tver, Russia 170000
mat@tversu.ac.ru

Abstract

Numerous results about capturing complexity classes of queries by means of logical languages work for ordered structures only, and deal with non-generic, or order-dependent, queries. Recent attempts to improve the situation by characterizing wide classes of finite models where linear order is definable by certain simple means have not been very promising, as certain commonly believed conjectures were recently refuted (Dawar's Conjecture).

We take on another approach that has to do with normalization of a given order (rather than with defining a linear order from scratch). To this end, we show that normalizability of linear order is a strictly weaker condition than definability (say, in the least fixpoint logic), and still allows for extending Immerman-Vardi-style results to generic queries. It seems to be the weakest such condition. We then conjecture that linear order is normalizable in the least fixpoint logic for any finitely axiomatizable class of rigid structures. Truth of this conjecture, which is a strengthened version of Stolboushkin's conjecture, would have the same practical implications as Dawar's Conjecture.

Finally, we suggest a series of reductions of the two conjectures to specialized classes of graphs, which we believe should simplify further work.

1 Introduction

An extensive study of expressibility by different logical languages in finite models has been motivated by applications to query languages for relational databases.

*This work has been partially supported by NSF Grant CCR 9403809.

†A part of this research was carried out while the author was visiting UCLA and supported in part by NSF Grant CCR 9403809.

The language *FO* of first order logic is equivalent to the so called relational calculus—the query language suggested by Codd [Cod70, Cod72]. Aho and Ullman [AU79] observed that the natural query of transitive closure is not expressible in the language, and the study of more powerful logical languages in the context of finite models became imminent.

Chandra and Harel [CH80, CH82] considered a computationally universal query language, and noticed that, however, only languages within reasonable complexity classes are of practical interest.

Immerman [Imm86, Imm87] and independently Vardi [Var82] established that the language *LFP* of least fixpoint logic captures exactly the *PTime*-computable queries in the class of all *ordered* finite models. Similar characterizations have been established for many other classes of "reasonable" complexity, including *LogSpace* and *NLogSpace* (see [Imm86, Imm87]).

However, for the general case of all finite models, similar characterizations are not known for the classes below *NP*—for *NP* a universal characterization was discovered by Fagin [Fag74]. Meanwhile, limiting our consideration to the case of ordered models only has severe disadvantages. To mention only one conceptual problem, the answers to the same query obtained under two different orderings of elements do not have to be the same, thus, queries in ordered models do not exclusively rely on the information stored in the database.

Of course, should the ordering be *deducible* from the information stored in database, rather than arbitrarily imposed, things would go differently.

This gave rise to the following approach: to try to extend the results for ordered models to some wider classes of models where an ordering is definable. The first paper in this direction was [Sto92] where Stolboushkin put in this context the following conjecture due to Zilber: *for any finitely axiomatizable (in FO) class of rigid finite models there exists an FO query expressing a linear order in every model in the class.* A model is called rigid iff it does not have nontrivial automorphisms. It is well known that almost all finite models are rigid (see [Wri80, Bol85]).

The conjecture sounded quite natural. Indeed, if a linear order is definable in a finite model by means of a *logical* language, the model is rigid. On the other hand, if the fixed *FO* sentence—our axiomatization—is powerful enough to disable all automorphisms, it looks as if there should be a way to transform this sentence into a formula that would define a linear order.

As for databases, all the query languages that have been considered extend *FO,* thus, if Zilber's conjecture were true, it would give natural generalizations of the aforementioned results on capturing complexity classes to finitely axiomatizable classes of rigid models.

However, Stolboushkin [Sto92] disproved Zilber's conjecture by constructing a counter-example—a finitely axiomatizable class $\mathcal{K}$ of rigid finite models where a linear order is not definable by *FO* formulas. Second however, Stolboushkin showed that in the class $\mathcal{K}$, a linear order is definable in first order logic implicitly.

It had generally been known (see [Gur88]) that implicit definability in finite models is not equivalent to explicit definability. Kolaitis [Kol90] suggested the logic *IMP* of generalized implicit definability and showed that it is at least as expressive as *LFP,* and captures, in ordered models, exactly the complexity class $UP \bigcap co\text{-}UP$ (UP being the class of unambiguous NP languages).

Stolboushkin [Sto92] then formulated the following generalization of Zilber's conjecture whose status remains open:

Conjecture 1.1 (Stolboushkin) *For any finitely axiomatizable (in FO) class of rigid finite models there exists an IMP query expressing a linear order in every model in the class.*

A stronger version of Conjecture 1.1, for *LFP* instead of *IMP*, was formulated by Dawar [Daw93]. At that moment, there was a substantial evidence in favor of Dawar's conjecture, as follows:

- The only then-known counter-example to Zilber's conjecture, the class $\mathcal{K}$, was known to admit ordering by *LFP*, and even by the logic of transitive closure—see [Sto92]
- Abiteboul and Vianu [AV91], and Dawar, Lindell, and Weinstein [DLW95] showed that the so-called k-types of elements of finite models can be uniformly ordered in *LFP*; [DLW95] conjectured then that if a k-equivalence class of finite models contains only rigid models, then all k-types are one-element

Besides, truth of Dawar's conjecture would have had a believably greater practical impact than of Stolboushkin's conjecture. Indeed, if Dawar's conjecture were true, it would give a characterization of *PTime*-computable queries for axiomatizable classes of rigid finite models, while truth of Stolboushkin's conjecture would only imply a characterization of $UP \bigcap co\text{-}UP$, which may or may not be equivalent to *PTime.*

However, Gurevich and Shelah [GS96] refuted Dawar's conjecture by constructing a finitely axiomatizable class $\mathcal{M}$ of rigid finite models called "odd multipedes", in which linear order is not *LFP*-definable.

In this paper we take on another approach which has to do with normalization of a given order (rather than with defining a linear order from scratch). Indeed, it is not a problem to impose a foreign linear order on the domain of a database state, actually, any representation of a database state inside a computer implies an ordering of the domain. The problem is, this arbitrarily imposed linear order may influence computations, and it is impossible to tell whether it does, for a specific query. We are trying to capture the cases when every such arbitrarily imposed linear order can be *normalized* to the same resulting linear order, which is therefore intrinsically related to the database state. This way we will capture *generic,* or order-independent, queries.

Of course, if we can normalize linear order, it implies rigidity of the considered structures by the definition of isomorphism. So we continue to consider rigid structures only. Then, if structures are rigid, we can in principle normalize linear orders, for example, by their first order types. However, what we are interested in are *simple* normalization procedures.

With this in mind, we are suggesting the following strengthened version of Conjecture 1.1. Observe that the practical implications of truth of this conjecture would be the same as those of Dawar's Conjecture.

Conjecture 1.2 *In any finitely axiomatizable (in FO) class of rigid finite models linear order is LFP-normalizable.*

Surprisingly, it turns out that normalizability of linear order is a strictly weaker requirement than definability. We show that Conjecture 1.2 survives all known examples of axiomatizable classes of rigid models, including the odd multipedes. We investigate relations between these two conjectures (1.1 and 1.2), and other open problems in Finite Models and Complexity Theory, and in particular show that even if Conjecture 1.2 is refutable, its refutation can not be easy.

We further show that in the orderings on classes of rigid finite models where linear order is *LFP*-normalizable, certain subsets of *LFP* capture exactly *PTime*-computable *generic* queries. This way, we further extend applicability of Immerman-Vardi's result on capturing *PTime* to wider classes of finite models (including the odd multipedes). On the other hand, our Theorem 3.8 shows that further weakening of the normalizability requirement—that by itself is a weakening of definability—will not allow to preserve Immerman-Vardi-style results.

Finally, in the positive direction, we suggest a series of reductions of the two conjectures to specialized classes of graphs, which we believe should simplify further work.

2 Some Definitions

Definition 2.1 (Ordering) *Let Ω be a signature, and let M be a model of the signature. Let N be a model of the signature $\Omega \bigcup \{<\}$, where $<$ is a new binary predicate not in Ω, such that*

- *M is Ω-reduct of N*
- *$<^N$ satisfies the axioms of linear order*

We will call N an ordering of M.

If $\mathcal{C}$ is a class of models, the class $\mathcal{C}^<$ of all possible orderings of all models in $\mathcal{C}$ will be referred to as the ordering of $\mathcal{C}$.

Definition 2.2 (Global predicate) *Let $\mathcal{C}$ be a class of finite models. A k-ary $\mathcal{C}$-global predicate is a mapping that maps each $M \in \mathcal{C}$ into a specific k-ary predicate in the domain of M. Formulas of logical languages naturally define global predicates.*

Let $\mathcal{C}^<$ be the ordering of $\mathcal{C}$. A $\mathcal{C}^<$-global predicate is called generic iff it correctly defines a $\mathcal{C}$-global predicate. In other words, the global predicate is actually order-independent.

A database state is essentially a finite model. Any representation of a finite model, for instance in the form of the input tape for a Turing machine, imposes a certain ordering on the model. Then nothing can prevent the Turing machine from using this imposed linear order in computations. Our ultimate goal is to characterize complexity classes of generic global predicates.

Definition 2.3 *LFP is the least fixpoint logic, and we will assume that its definition is known. Let ψ be an LFP formula of a signature Ω with k free variables, and let $P \in \Omega$ be a k-ary predicate symbol.*

By LFP^P_ψ we denote all the LFP-formulas of the signature Ω with ψ substituted for P. Thus, LFP^P_ψ is a simply defined subset of LFP-formulas of the signature Ω.

The following generalization of implicit definability is from [Kol90].

Definition 2.4 ((generalized) Implicit Definability) *Let Ω be a signature, and $R_1, R_2, \ldots, R_k$ be a sequence of predicate symbols of certain arities not in Ω. Let $\phi(R_1, R_2, \ldots, R_k)$ be a sentence of the signature Ω extended by $R_1, R_2, \ldots, R_k$.*

$\phi(R_1, R_2, \ldots, R_k)$ is said to implicitly define *$R_1, R_2, \ldots, R_k$ iff for any model M there exist unique predicates $R_1^M, R_2^M, \ldots, R_k^M$ of appropriate arities that make the sentence $\phi(R_1, R_2, \ldots, R_k)$ true.*

A global predicate R is said to be implicitly definable iff there exist non-signature predicate symbols $R, R_1, R_2, \ldots, R_k$ and a sentence $\phi(R, R_1, R_2, \ldots, R_k)$ such that $\phi(R, R_1, R_2, \ldots, R_k)$ implicitly defines $R, R_1, R_2, \ldots, R_k$, and the value of R^M assigned to R to make the sentence $\phi(R, R_1, R_2, \ldots, R_k)$ true in M coincides to the value of the global predicate in the model M.

The class of implicitly definable predicates will be referred to as IMP.

As usual, we will also consider class-parameterized notions of implicit definability. E.g., in most cases we consider implicit definability for the class off all finite models.

We will also need the following definition from [GS96].

Definition 2.5 (Odd multipedes) *A multipede is a finite model of the relational signature $\langle <^{(2)}, H^{(3)}, \epsilon^{(2)}, F^{(2)} \rangle$, whose domain is partitioned into three disjoint subsets, $\mathcal{F}, \mathcal{S}, \mathcal{P}$, called, respectively, feet, segments, and sets, and such that:*

- *$<$ is defined on segments only and is a linear order there*
- *F connects exactly two feet to every segment, and each foot is connected by F to exactly one segment*
- *$\mathcal{P}$ is thought of as the set of all possible subsets of segments, and ϵ is the membership relation, i.e., $s \epsilon p$ iff the segment s belongs to the set p.*
- *H is a hyper-graph relation separately defined in segments, and feet, such that for any foot-hyperedge there exists the corresponding segment-hyperedge, and for any segment-hyperedge, four out of eight possible foot-hyperedges belong to the foot-hyper-graph, and*

- *for any foot-hyperedge, if we rotate exactly two pairs of feet, the result would be another foot-hyperedge ("rotate" stands for replacing a foot with the other foot of the same segment)*

A multipede is called odd iff for any non-empty set of segments there exists a segment-hyperedge that intersects the set in either exactly one, or exactly three, segments.

The class of all odd multipedes will be denoted $\mathcal{M}$.

The following two theorems are from Gurevich and Shelah [GS96].

Theorem 2.1 *Odd multipedes are finitely first order axiomatizable and rigid.*

Theorem 2.2 *No $\mathcal{M}$-global LFP-definable predicate satisfies the axioms of linear order.*

3 Normalizable Linear Orders

Definition 3.1 *Let Ω be a signature, and let $\mathcal{C}$ be a class of rigid finite models of the signature. Let $\mathcal{C}^<$ be the ordering of $\mathcal{C}$, i.e., the class of all possible orderings of all models in $\mathcal{C}$. We say that linear order is LFP-normalizable in the class $\mathcal{C}$ iff there exists an LFP-formula ψ in the signature $\Omega \bigcup \{<\}$ that defines a linear order in any model $M \in \mathcal{C}^<$, and this linear order does not depend on $<^M$. In other words, the $\mathcal{C}^<$-global predicate defined by ψ is generic. We will call such predicates* **normalizations** *of linear order in $\mathcal{C}$.*

Similarly, we define FO-normalizable linear orders.

Essentially, we are trying to formalize the situation when, on a class $\mathcal{C}$, a certain intrinsic linear order can be obtained from any given one by means of an *LFP*-transformation (or *FO*-transformation). An example of such an intrinsic order is the *minimal* linear order defined in [HKL94]. However, we do not want to stick to any specific linear order.

Of course the question to address first is, whether normalizability of linear order implies definability, in *LFP* if not in *FO*. Surprisingly, it turns out to be not the case, even for finitely axiomatizable classes.

Theorem 3.1 *There exists a finitely axiomatizable class of rigid finite models with FO-normalizable linear order, such that no linear order is LFP-definable in the class.*

Idea of the Proof: Consider the class $\mathcal{M}$ of odd multipedes (see Definition 2.5). No linear order is *LFP*-definable in the class—Theorem 2.2, and odd multipedes are finitely axiomatizable and rigid—Theorem 2.1.

Given an odd multipede, let us restrict our attention to the class of linear orders that agree with the existing linear order on the segments, lexicographically extend it to the sets of segments, and relate feet of different segments according to the order on the segments. Such "regular" orders differ from one another only on the feet of individual segments.

Now suppose that a linear order O, whether regular or not, is given. There exists a first order formula that uses O and does not depend on the model, and that defines the regular linear order which agrees with O in the feet of every segment. Let us call this new regular order R.

Further, we can express in first order logic (since we have sets of segments) that there exists a regular modification R' of R such that any other regular modification of R is bigger, in the sense of the hyper-graph on the feet. Following [DHK95], we compare two orders R and R' as follows. Let $a_1, a_2, \ldots, a_r$ and $b_1, b_2, \ldots, b_r$ be the feet ordered according R and R', respectively. Let (i,j,k) is the lexicographically minimal triplet of natural numbers such that one of (a_i, a_j, a_k) and (b_i, b_j, b_k) is a hyperedge, and the other isn't. Then if (a_i, a_j, a_k) is a hyperedge, we say that R' *is bigger than* R.

This linear order R' is the normalization that we have been looking for.

The technical details of the proof are left to the reader. So, in the class $\mathcal{M}$, linear order is FO-normalizable. *Q.E.D.*

However, FO-normalization is by itself too weak, as is demonstrated by the following:

Theorem 3.2 *There exists a finitely axiomatizable class of rigid finite models where linear order is LFP-definable (and hence LFP-normalizable), but not FO-normalizable.*

PROOF (HINT): The proof can be carried out for the class $\mathcal{K}$ from [Sto92] along the lines of the proof of first order undefinability of linear order in $\mathcal{K}$.

Let

$$\{(i,0),(j,1) \mid i = 1,2,\ldots,k, j = 1,2,\ldots k+1\}$$

be the two-pile model considered in [Sto92]. In this model, we have the successor relation within each pile ($(i,l) \Leftrightarrow (i+1,l)$, idempotent at the last element of a pile), and a "level ordering" ((i,l) is less than (j,r), whenever $i < j$, for any $l, r \in \{0,1\}$). In addition to the signature of $\mathcal{K}$, throw in the linear order defined as the transitive closure of the following successor relation:

$$(r,0)' = (r,1)$$

$$(r,1)' = \begin{cases} (r+1,0) & \text{if it exists} \\ (r+1,1) & \text{otherwise} \end{cases}$$

If k is big enough (for the formula that allegedly normalizes linear order in $\mathcal{K}$), then in the middle of the piles—for r's around $k/2$—this normalized linear order should be defined the same way, that is, either $(r,0) < (r,1)$, or the other way around, uniformly for all such r's. This can be demonstrated by a standard Ehrenfeucht-Fraïssé game.

Because linear order is normalizable, the same normalization should result from any initial linear order. Specifically, consider now a "zigzagging" l.o. that makes one step up in agreement with the pile successor, and then goes to the equivalent element in the other pile, etc. Still, as we observed, in the middle the "left" elements are all smaller than the "right" elements, or the other way around, uniformly.

To get a contradiction, employ an Ehrenfeucht-Fraïssé game to show that, under the zigzagging order, $(r, 0) < (r, 1)$ implies $(r+1, 0) > (r+1, 1)$. *Q.E.D.*

As for usefulness of the conjecture, we offer the following results.

Theorem 3.3 *Let $\mathcal{S}$ be a finitely axiomatizable class of rigid finite models with LFP-normalizable linear order. Then there exists an IMP definition that defines a linear order in any model in $\mathcal{S}$.*

HINT: *LFP* is translatable into *IMP* (see [Kol90]). We can guess a linear order at first, and then make sure that it coincides with its normalization (which makes the order unique). *Q.E.D.*

Corollary 3.4 (Dawar, Hella, Kolaitis) *In the class of odd multipedes, linear order is IMP-definable.*

This result by itself is due to Dawar, Hella, and Kolaitis [DHK95].

Theorem 3.5 *Let $\mathcal{S}$ be a class of rigid finite models with LFP-normalizable linear order. Let ψ be an LFP formula that witnesses normalization.*

Then in the class $\mathcal{S}^<$ of all orderings of $\mathcal{S}$, $LFP_\psi^<$ captures exactly PTime-computable generic queries.

HINT: Combine the fact that *LFP* captures *PTime* in ordered structures with the fact that ψ yields a normalized linear order. *Q.E.D.*

In terms of the evidence supporting Conjecture 1.2, we can say that it holds for all finitely axiomatizable classes of rigid structures known at this moment. Further, we offer the following argument that essentially says that, even if the conjecture is refutable, its refutation can not be simple.

Theorem 3.6 *If PTime = NPTime then linear order is LFP-normalizable in the class of all rigid finite models.*

IDEA OF PROOF: For a given linear order on a structure, checking whether it is minimal for the structure obviously is in the complement of *NPTime*. Hence, for rigid structures, the minimal linear order is definable in *NPTime* with oracles in *co-NPTime*. This observation was first made in [HKL94].

If *PTime* = *NPTime*, the minimal order would be *PTime*-definable, hence *LFP*-normalizable, by virtue of *LFP* being equivalent to *PTime* on ordered models. *Q.E.D.*

It shows that even a strengthened version of Conjecture 1.2, with the finite axiomatizability constraint dropped, is hard to refute. As we will see in the next section, this strengthened version is essentially equivalent to existence of a *PTime*-normalization procedure for rigid graphs, by itself, an open problem.

While refutation of Conjecture 1.2 (or Conjecture 1.1) would have such tremendous consequences in Complexity Theory, a proof of either of the conjectures does not seem to have consequences of this nature, as the following fact holds.

Fact 3.7 *The class of all rigid models is not finitely axiomatizable.*

HINT: Consider a binary "successor" relation. A standard application of Ehrenfeucht-Fraïssé games shows that if a sentence holds on all the finite structures where this relation behaves as a true successor (essentially, orders all the elements of the structure linearly), it also holds on any large enough true successor with a large enough "loop" on the side. The loop, however, makes the model not rigid. *Q.E.D.*

Finally, refutation of Conjecture 1.2 would make any further pursuit of Immerman-Vardi-style results for the classes of rigid structures impossible, as the following result shows:

Theorem 3.8 *Let $\mathcal{C}$ be a class of rigid finite models, and Q be a normalization of linear order in $\mathcal{C}$. Then Q is PTime-computable iff Q is LFP-normalizable.*

SKETCH OF PROOF: Everything definable in *LFP* is *PTime*-computable, hence **if**. For the **only if** part, recall that since we start off with some linear order, everything *PTime*-computable is *LFP*-definable. *Q.E.D.*

4 Reduction to the Case of Fully Separated Digraphs

The goal of this section is to reduce the general case of the problem of implicit definability, and *LFP*-normalizability, of linear order to special families of graphs. We consider here three reductions that preserve finite axiomatizability, rigidity, and implicit definability and *LFP*-normalizability of linear order, and the size of the models under the reductions grows only polynomially. The reductions are basically of the same flavor as those discussed in [ELTT65]. It is important to mention that, however, not all the reductions from [ELTT65] work for our purpose, because some of them do not preserve the rigidity.

We start from describing the three reductions.

4.1 Transitive Digraphs With no Loops or Cycles

Our goal here is to reduce the general case of an arbitrary signature and arbitrary structures to the case of structures of one transitive binary relation (essentially, to transitive directed graphs).

The general method of reducing one class to another is to throw in additional elements, and then to encode the old relation(s) by means of the new relation. In all the cases that we consider, the single relation in the encoding class of structures is going to be binary.

We assume here that the signature does not contain any function symbols, and therefore consists of relational symbols only.

So let $P_1, P_2, \ldots, P_k$ be the list of all predicates in the signature of the original class of structures. We first separately encode each individual predicate.

We denote the (only) binary relation in the encoding structures by $<$ (as always, we may use equality though).

Consider the reduction in the case of a ternary predicate P. For each triplet a, b, c of elements such that $P(a, b, c)$, we throw in a new element $[abc]^P$, and connect it to the elements a, b, c (i.e., $[abc]^P < a$, $[abc]^P < b$, and $[abc]^P < c$).

This by itself does not distinguish a from b and from c. To distinguish them from one another, we throw in 3 additional elements $[abc]^P_{b,1}$, $[abc]^P_{c,1}$, and $[abc]^P_{c,2}$.

We now extend the relation $<$ as follows:

$$
\begin{aligned}
[abc]^P &< [abc]^P_{x,1} < x, \\
[abc]^P_{c,1} &< [abc]^P_{c,2} < c, \text{ and} \\
[abc]^P &< [abc]^P_{c,2}.
\end{aligned}
$$

for x in $\{b, c\}$.

This would end the encoding if we only had one ternary predicate. However, we may have more than one predicate, and to distinguish tuples that belonged to different predicates, we will attach a sequence (in the sense of $<$) of r additional elements just before any the element $[\ldots]^{P_r}$.

The argument above assumes that a, b, c are always different. But since the equality always is available, we can consider only such "good" models, and if our models are not good, we can first encode them by good models in an extended signature. This first pre-reduction does not use additional elements and is trivial.

This finishes the first reduction. The result obviously is a binary transitive anti-reflexive relation with no isolated points, as desired.

4.2 Graphs

At this point we consider only digraphs with no loops (as provided by the first reduction). Transitivity of such graphs will not be used. Our goal now is to encode those models by models of one binary *symmetrical* anti-reflective relation.

So let $<$ be the only binary relation of the original class, and let us denote the only new binary relation by $\equiv$.

Let a pair a, b of elements be in the relation $<$, that is $a < b$. We throw in six new elements $[ab]_1$, $[ab]_2$, $[ab]_3$, $[ab]_4$, $[ab]_5$, $[ab]_6$, and connect them symmetrically as follows (we only give one direction, but mean both):

$$
\begin{aligned}
[ab]_1 &\equiv a \\
[ab]_1 &\equiv [ab]_2
\end{aligned}
$$

$$
\begin{array}{rcl}
[ab]_1 & \equiv & [ab]_4 \\
a & \equiv & [ab]_4 \\
[ab]_2 & \equiv & [ab]_4 \\
[ab]_3 & \equiv & [ab]_4 \\
b & \equiv & [ab]_4 \\
[ab]_2 & \equiv & [ab]_3 \\
[ab]_3 & \equiv & b \\
[ab]_3 & \equiv & [ab]_5 \\
[ab]_3 & \equiv & [ab]_6
\end{array}
$$

$[ab]_5$ and $[ab]_6$ are used to make a distinction between $a < b$, and $b < a$. Indeed, given a pair, a, the first element of the encoded pair, can be identified as the element connected to exactly one element of degree 5 (while b is connected to two of them).

The second reduction is now fully defined. It is obvious that the resulting relation $\equiv$ is with no isolated points, anti-reflective, and symmetrical. Notice also that all the new elements in the graph are of odd degrees, while all the old elements are of even degrees.

4.3 One fully separated binary relation with strictly two successors

This is the most trivial of our reductions. By "fully separated" we mean that there is a partition of all the elements into two disjoint classes A and B such that if two elements are in this binary relation, then the first is in the class A, and the second in the class B. "Two successors" means that for each element of the first class A there are exactly 2 elements of the class B that are in this relation with the element.

The structures can be visualized as bipartite digraphs of out-degree 2.

So let's encode a structure of one anti-reflective, symmetrical predicate with no isolated points by a structure of one fully separated binary relation with strictly two successors.

Let $\equiv$ be the "old" relation, and $\triangleleft$ be the "new" encoding relation.

For each pair a, b of elements such that $a \equiv b$ we throw in one additional element $[ab]$ and define the relation $\triangleleft$ as follows:

$$[ab] \triangleleft a \text{ and } [ab] \triangleleft b.$$

This does the job. Note that for each pair we add at most one element.

Fact 4.1 *All the three reductions described above increase the size of encoding models polynomially.*

In the case of the first reduction (to digraphs), the polynomial depends on the original signature.

Fact 4.2 *All the three reductions described above preserve finite axiomatizability.*

Fact 4.3 *For each of the three reductions described above, the original structure is rigid iff the result of the reduction is rigid.*

PROOF: can be done by instant verification in all the three cases. *Q.E.D.*

Theorem 4.4 *For each of the three reductions described above, if a linear order is uniformly implicitly definable in every reduct model of a certain class, a linear order is uniformly implicitly definable in every model of the class of originals of these structures as well.*

PROOF: The idea behind the proof is clear, although its implementation, especially in the case of the first reduction, may be messy.

We describe the idea only for the case of the last reduction, but it should be clear how to apply the same technique in two other cases.

So in the case of the last reduction, for any two elements a, b of an original model, we either don't introduce new elements at all, or otherwise introduce exactly 1 new element. This makes the total number of new elements less than or equal to n^2, where n is the cardinality of the original model. Besides, for any pair (a, a), we don't introduce a new element.

Thus, the cardinality of the reduct model is at most n^2.

So let $\psi(R_1, R_2, \ldots, R_k)$ be the formula that defines a linear order on the reduct models implicitly, i.e., one of the predicates in the list $R_1, R_2, \ldots, R_k$, say, the first one, is a binary relation that is a linear order on any reduct.

We are going to effectively rewrite this formula into a formula of the original signature with all the predicates $R_1, R_2, \ldots, R_k$, whose arities are duplicated, plus one additional binary relation. The latter binary relation that we will denote by $<$ will produce the linear order on the original model. Of course, the resulting formula is going to implicitly define all the non-signature relations on all originals.

We start from writing the axioms of linear order for $<$ in the original model.

In the original model, pairs of elements are going to encode elements of the reduct, as follows.

$$(a, a) \text{ encodes } a\text{—elements shared by both the models;}$$
$$(a, b), \text{ for } a < b, \text{ encodes } [ab].$$

These pairs are called "alive".

There are also "dead" pairs (a, b), such that either $a > b$, or $a < b$ but there is no element for the pair (a, b) in the reduct. Notice that "dead" and "alive" pairs are first-order definable in the original model. Say, $\alpha(x, y)$ holds for alive pairs only. Now we are fully prepared to rewrite ψ.

All variables are now represented by pairs of variables. Say, for any variable x in ϕ, we introduce an additional variable x'. The translation of the formula itself is done according to the following rules:

- Any quantifier $(\exists x)(\Phi)$ is replaced with $(\exists x, x')(\alpha(x,x') \wedge (\Phi))$;
- $x = y$ is replaced with $x = y \wedge x' = y'$;
- $x \lhd y$ is replaced with $\alpha(x,x') \wedge y = y' \wedge (x = y \vee x' = y) \wedge x \equiv x'$;
- $R(x_1, x_2, \ldots, x_l)$ for a new predicate R is replaced with $(\wedge_{i=1}^{l} \alpha(x_i, x_i') \wedge R(x_1, x_2, \ldots, x_l, x_1', x_2', \ldots, x_l'))$.

This leaves undefined the behavior of the new predicates $R_1, R_2, \ldots, R_k$ when one of the pairs is "dead". To make the definitions unique for this case, for any R_i we add a sentence saying that it does not hold when at least one of the pairs is dead.

Finally, we add the following formula:

$$x < y \text{ iff } R_1(x, y, x, y)$$

that essentially says that our linear order $<$ agrees with the linear order R_1.

It is now clear that the transformed formula defines all the predicates implicitly in any original model. *Q.E.D.*

Similarly, we have:

Theorem 4.5 *Let $\mathcal{S}$ be a class of rigid finite models.*

For each of the three reductions described above, linear order is LFP-normalizable in $\mathcal{S}$ iff linear order is LFP-normalizable in the class of $\mathcal{S}$-results of the reduction.

This shows that Conjectures 1.1 and 1.2 can be specialized to either one of the three considered classes of finite models, without loss of generality.

5 Acknowledgments

We wish to sincerely thank the anonymous referee for his insightful remarks and valuable suggestions, and Bruce Rothschild for advising us on Graph Theory.

References

[AU79] A.V. Aho and J.D. Ullman. Universality of data retrieval languages. In *Proc. 6th ACM Symp. on Principles of Programming Languages*, pages 110–117, 1979.

[AV91] S. Abiteboul and V. Vianu. Generic computation and its complexity. In *Proc. 23rd ACM Symp. on Theory of Computing*, pages 209–219, 1991.

[Bol85] B. Bollobas. *Random Graphs*. Academic Press, 1985.

[CH80] A. Chandra and D. Harel. Computable queries for relational databases. *Journal of Computer and System Sciences*, 21:156–178, 1980.

[CH82] A. Chandra and D. Harel. Structure and complexity of relational queries. *Journal of Computer and System Sciences*, 25:99–128, 1982.

[Cod70] E.F. Codd. A relational model for large shared data banks. *Communications of the ACM*, 13:377–387, 1970.

[Cod72] E.F. Codd. Relational completeness of data base sublanguages. In R. Rustin, editor, *Database Systems*, pages 33–64. Prentice-Hall, 1972.

[Daw93] A. Dawar. *Feasible computations through model theory.* PhD thesis, University of Pennsylvania, 1993.

[DHK95] A. Dawar, L. Hella, and Ph.G. Kolaitis. Implicit definability and infinitary logic in finite model theory. In Z. Fulop and F. Gecseg, editors, *Proc. 22nd International Colloquium on Automata, Languages, and Programming. ICALP 95*, pages 624–635. Springer-Verlag, 1995.

[DLW95] A. Dawar, S. Lindell, and S. Weinstein. Infinitary logic and inductive definability over finite structures. *Information and Computation*, 119(2):129–145, 1995.

[ELTT65] Yu.L. Ershov, I.A. Lavrov, A.D. Taimanov, and M.A. Taitslin. Elementary theories. *Russian Mathematical Surveys*, 20(4):35–105, 1965.

[Fag74] R. Fagin. Generalized first–order spectra and polynomial–time recognizable sets. In R.M. Karp, editor, *Complexity of Computation, SIAM-AMS Proceedings, Vol. 7*, pages 43–73, 1974.

[GS96] Y. Gurevich and S. Shelah. On finite rigid structures. *Journal of Symbolic Logic*, 61(2):549–562, 1996.

[Gur88] Y. Gurevich. Logic and the challenge of computer science. In E. Börger, editor, *Current trends in theoretical computer science*, pages 1–57. Computer Science Press, 1988.

[HKL94] L. Hella, Ph.G. Kolaitis, and K. Luosto. How to define a linear order on finite models. In *Proc. 9th IEEE Symp. on Logic in Computer Science*, pages 40–49. IEEE Computer Society Press, Los Alamitos, CA, 1994.

[Imm86] N. Immerman. Relational queries computable in polynomial time. *Information and Control*, 68:86–104, 1986.

[Imm87] N. Immerman. Languages that capture complexity classes. *SIAM Journal of Computing*, 16:760–778, 1987.

[Kol90] Ph.G. Kolaitis. Implicit definability on finite structures and unambiguous computations. In *Proc. 5th IEEE Symp. on Logic in Computer Science*, pages 168–180, 1990.

[Sto92] A.P. Stolboushkin. Axiomatizable classes of finite models and definability of linear order. In *Proc. 7th IEEE Symp. on Logic in Computer Science*, pages 64–70, Los Alamitos, CA, 1992. IEEE Computer Society Press.

[Var82] M.Y. Vardi. The complexity of relational query languages. In *Proc. 14th ACM Symp. on Theory of Computing*, pages 137–146, 1982.

[Wri80] E.M. Wright. Graphs on unlabelled nodes with a given number of edges. *Acta Mathematicae*, 126:1–9, 1980.