# Languages for Relational Databases over Interpreted Structures

Michael Benedikt and Leonid Libkin

# Languages for Relational Databases over Interpreted Structures

**Michael Benedikt**
Bell Laboratories
1000 E. Warrenville Rd
Naperville IL 60566, USA
Email: `benedikt@bell-labs.com`

**Leonid Libkin**
Bell Laboratories
600 Mountain Avenue
Murray Hill NJ 07974, USA
Email: `libkin@bell-labs.com`

**Abstract**

We rework parts of the classical relational theory when the underlying domain is a structure with some interpreted operations that can be used in queries. We identify parts of the classical theory that go through 'as before' when interpreted structure is present, parts that go through only for classes of nicely-behaved structures, and parts that only arise in the interpreted case. The first category includes a number of results on equivalence of query languages, as well as expressive power characterizations for the active-domain semantics for a variety of logics. The second category includes most of our results on the natural semantics, including results on cases where the natural semantics collapses to the active semantics. While these collapse results have been proved by nonconstructive means for first-order logic in previous work, we here give a set of *algorithms* for eliminating unbounded quantifications in favor of bounded ones. Furthermore, we show these results for a new class of higher-order logics that mix unbounded and bounded quantification. We give a set of normal forms for these logics, under special conditions on the interpreted structures.

As a by-product, we obtain an elementary proof of the fact that parity test is not definable in the relational calculus with polynomial inequality constraints.

## 1 Introduction

We would like to a start with an example that can be found in most database textbooks. When relational algebra is introduced, the conditions in selection operators are defined to be boolean combinations of $x = y$ and $x < y$, where $x$ and $y$ are variables or constants. Typically, a few examples of programming in relational algebra are given before a relational language, such as SQL or QUEL, is introduced. Soon after that we will probably see an example of query like "Select employees who make at least 90% of their manager's salary". Such an example is likely to be followed by a remark that this query, strictly speaking, is not definable in relational algebra because it involves arithmetic operations, but it is definable in SQL or QUEL which allow arithmetic and comparisons of the form $x > 0.9 \cdot y$.

Does it mean that the relational algebra is in some sense inadequate as a basic relational query language? It is certainly so if one has to deal with arithmetic, but database textbooks counter this criticism by saying that an extension to arithmetic is straightforward. For example, in his textbook [36], Ullman writes:

> Often, atoms in calculus expressions or selections in algebraic expressions can involve arithmetic computation as well as comparisons, e.g., $A < B + 3$. Note that $+$ and other arithmetic operators appear in neither relational algebra nor calculus, but the extension of those notations to include arithmetic should be obvious.

It is obvious how to extend these notations, but it isn't completely obvious how to extend the fundamental results and techniques of relational theory to include arithmetic. For example, if arithmetic constraints are allowed in selection predicates, how would one prove the classical result that parity test and transitive closure are not definable in relational algebra? It appears that the standard techniques either are not directly applicable (such as 0-1 laws), or require considerably more complex arguments (such as games). This situation extends beyond arithmetic to other interpreted operations over the items that could be stored in a database.

As noted in [18], the proposal to combine finite databases with an infinite interpreted structure, made in the seminal paper of Chandra and Harel [11], has not been fully explored in the database community, despite the fact that the rest of that paper made enormous impact on the development of database theory. An exception, in addition to [18], has been the attention to order relation. This includes well-known connections between query languages and complexity classes, cf. [1, 15]; also, order constraints have been studied in the context of Datalog queries (see, e.g., [28]).

Let us give a sample of the results we obtain in this paper. We first look at queries that may use the information about the underlying model (for example, they may use arithmetic operations), but only ask questions about elements that occur in a database. These are called active-semantics queries, and are the ones intended in the 'classical database' context of the quotation above. For those queries, we will review and extend a technique that has arisen recently [6, 31] for analyzing interpreted structure: *generic collapse* results. These state that generic queries definable by various logics can be defined without additional interpreted operations. That is, they do not add extra relational power. The collapse results show that for active queries many (but not all!) traditional relational theory results generalize smoothly to the interpreted setting. We then turn to queries which may ask questions about elements outside of the database (these are called natural-semantics queries), and prove a number of *natural-active collapse* results showing that over a certain class of interpreted structures, this extension does not add any expressive power. We prove these results for several logics. We concentrate not only on proving those results, but we also continue to refine and generalize a collection of techniques that are likely to be used as the theory further develops.

As mentioned above, there are issues concerning interpreted structure that are not straightforward, even in the setting of the relational model. Much recent interest in databases over interpreted structures, however, stems from the constraint database model introduced by Kanellakis, Kuper and Revesz [24]. They were motivated by new applications involving spatial and temporal data, which require storing and querying infinite collections. The constraint model generalizes Codd's relational model by means of "generalized relations". These are possibly infinite sets defined by quantifier-free first-order formulae in the language of some underlying infinite model $\mathcal{M} = \langle \mathcal{U}, \Omega \rangle$. Here $\mathcal{U}$ is a set (assumed to be infinite), and $\Omega$ is a signature that consists of a number of interpreted functions and predicates over $\mathcal{U}$. For example, in spatial applications, $\mathcal{M}$ is usually the real field $\langle \mathbb{R}, +, *, 0, 1, < \rangle$, and generalized relations describe sets in $\mathbb{R}^n$.

One of the main contributions of [24] is an extension of relational calculus and DATALOG to generalized relations. The development of constraint query languages made clear some of the ways in which the interpreted structure adds new issues to the analysis of query languages. For example, these extensions do add expressive power: one may ask if there is a pair $(x, y)$ with $xy = 2y + 3$. In [6] it was shown that this extension, with the real field as the underlying model, does not add *relational* expressive power: one cannot define parity or transitive closure, and, more generally, queries that are *generic* in the sense of [1, 12], other than those already definable with order. Since it is known that such expressive bounds cannot be obtained for arbitrary structures [20], these results give indication that relational calculi over some interpreted structures are much more similar to pure relational calculus than others are.

In addition to questions of expressive power, the constraint model also shows how issues concerning quantification that don't arise in the classical theory come to the fore when interpreted structure is present. Relational calculus expressions are first-order formulae that may use quantifiers. What does $\exists x.\varphi(x)$ mean? It could mean that $\varphi$ is satisfied by some element of a database (this is called *active interpretation*), or by some element of the underlying universe $\mathcal{U}$ (this is called *natural interpretation*). The issue of two semantics arises in the context of pure relational calculus, but Hull and Su [23] proved that the expressive power of relational calculus is the same under both interpretations. This is what we call the natural-active collapse. (In fact, an earlier result [3] showed that infinite domains are unnecessary for pure relational calculus.) This collapse is known not to hold over arbitrary structures, hence questions concerning the two semantics become much richer in the interpreted setting. Since constraint database formalisms aim to give finite means for describing infinite objects, manipulating natural quantification plays a fundamental role, forcing us to give much more attention to it than in the pure case.

It might appear that we have to develop separate theories for the two interpretations of quantifiers, unless we show that they coincide. It then constituted significant progress when [32] proved the analog of the Hull-Su theorem for *linear* constraints. These results were extended to polynomial constraints in [7].

Although the above results show how to reduce questions about natural semantics to ones concerning active semantics for first-order queries over many structures, there are a number of issues left open. In particular, the proofs in [6] and [7] are quite involved: they use techniques of nonstandard universes [13] and o-minimal structures [33]. In addition, the proofs in those papers are nonconstructive. In [23] (that deals with the pure case) and [32] (that deals with linear constraints), an *algorithm* is given that converts a natural query into an active one. However, in [7] (which deals with polynomial constraints and other o-minimal structures), a mere existence of such a query is proved, but no algorithm is given. One of the main results of the current paper is an algorithm for doing this conversion. It is hoped that this will lay the groundwork for transforming a number of optimization algorithms for pure relational calculi to the interpreted setting.

Going from first-order logic to more expressive logics (fixpoint, second-order), we observe a new interesting phenomenon. Even when we identify structures over which first-order queries have tame behavior under the natural interpretation (for example, $\langle \mathbb{R}, +, *, 0, 1, < \rangle$), fixpoint logics and second-order logic may still be too expressive. More precisely, the set of natural numbers is definable in those logics over the real field, and then an encoding with natural numbers shows that any recursive query can be expressed in these languages. In addition, second-order constructs when interpreted naively over the natural domain lead to undefined queries as well as undecidable ones: counting quantifiers and monotone fixpoints need no longer be well-defined on infinite sets.

For higher-order logics, however, there are numerous subclasses that one can restrict to that do not arise naturally in the pure case, and which have the same closure properties as active queries, while also admitting interesting expressive bounds. In particular, we consider a new class of logics that we call *hybrid*. The idea is that first-order quantification can still be interpreted naturally (that is, $\exists x.\varphi(x)$ means $\varphi(a)$ for some $a \in \mathcal{U}$), but the higher-order features are interpreted actively. For instance, for the hybrid second-order logic, second-order quantifiers range over subsets of the active domain, while first-order quantifiers range over $\mathcal{U}$. Similarly, fixpoints are only taken within the active domain. Such logics are of interest in the constraint database context, since the natural versions of the higher-order constructs are not available. We show that the behavior of such hybrid logics is close to the behavior of first-order logic.

For logics that use a consistent semantics (active or natural) for all quantifiers, it is possible to adapt the normal forms known in the pure case. An interesting aspect of hybrid logics is that standard normal forms that hold straightforwardly for both natural and active interpretation logics become highly nontrivial in the hybrid case. We will show that for reasonably-behaved models we can obtain standard normal forms for both fixpoint and second-order logic in the hybrid case. For arbitrary interpreted structure we conjecture that these normal forms fail.

**Goals of the paper** The main goal of this paper is to study a number of issues of the classical relational theory for databases and languages over interpreted structures. These include equivalence of traditional query languages and various logics, expressive power of query languages, the relationship between the active and the natural interpretation and data complexity.

Included in this examination will be an elementary proof that parity can not be defined using relational calculus with polynomial inequality constraints. This new proof completely avoids techniques of nonstandard universes, and gives a constructive 'parameterized quantifier-elimination' procedure that can be seen as an extension of the work in [32, 9, 35, 34].

In addition to proving a number of results, this paper also provides a set of techniques that can be used for other logics and query languages.

**Organization** Notations are introduced in Section 2. In Section 3, we deal with the active interpretation. We present the Ramsey technique (which is abstracted from the proofs in [6] and [32]), and show that it can be uniformly applied to a variety of logics (defined in Section 2), to obtain generic collapse results. This leads to a number of expressivity results for these logics over interpreted structures. We show that natural extensions of relational algebra, DATALOG$^{\neg}$ and WHILE, are still equivalent with first-order, least- and partial-fixpoint logic over arbitrary models. We look at infinitary logic with finitely many variables (which subsumes all fixpoint logics), and show how to extend the proof techniques to deal with it. Finally, we give a flavor of the kind of result about the active semantics which *does* depend nontrivially on the particular interpreted structure: the ability to guard against unsafe queries.

In Section 4, we deal with the natural interpretation. Our main goal is to give a *constructive* proof of the natural-active collapse for databases over the reals. We prove a more general result, showing that the natural and active interpretations are equally expressive over any structure that is o-minimal [33] and has quantifier elimination. We prove this for the hybrid second-order logic (second-order quantifiers range over the subsets of active domain). This result was proved for first-order logic by us earlier [7] in a nonconstructive way; now we give a recursive translation from the natural formulae into active formulae, provided the model is recursive and quantifier elimination is effective. We derive the natural-active collapse for first-order logic, hybrid fixpoint logics, and also give normal forms for formulae in these hybrid logics. We also give a much simplified proof of the Hull-Su theorem in the pure case, and show a tighter result: the Hull-Su theorem holds for variable-bounded logics.

Concluding remarks are given in section 5. Proofs can be found in the full version [8].

## 2 Notations

**Databases over infinite models** In this paper, we study databases over infinite models. Let $\mathcal{M} = \langle \mathcal{U}, \Omega \rangle$ be an infinite model, where $\mathcal{U}$ is an infinite set, called a carrier (in the database literature it is often called domain), and $\Omega$ is a set of interpreted functions, constants, and predicates. For example, the real field $\langle \mathbb{R}, +, *, 0, 1, < \rangle$ has carrier $\mathbb{R}$ (the set of real numbers), functions $+$ and $*$, constants 0 and 1, and predicate $\leq$.

A (relational) **database schema** $SC$ is a nonempty set of relation names $\{S_1, \ldots, S_l\}$ with associated arities $p_1, \ldots, p_l > 0$. Given a model $\mathcal{M}$ and $X \subseteq \mathcal{U}$, an **instance** of $SC$ over $X$ is a family of finite sets, $\{R_1, \ldots, R_l\}$, where $R_i \subset X^{p_i}$. That is, each schema symbol $S_i$ of arity $p_i$ is interpreted as a finite $p_i$-ary relation over $X$. We use $Inst(SC, X)$ or $Inst(SC, \mathcal{M})$ to denote the set of all instances of $SC$ over $X$ or $\mathcal{M}$.

Given an instance $D$, $adom(D)$ denotes its **active domain**, that is, the set of all elements that occur in the relations in $D$. If $S$ is a new $n$-ary symbol not in $SC$ and $R$ is a finite subset of $\mathcal{U}^n$, then $D_R$ denotes the instance of $SC \cup \{S\}$ where $S$ is interpreted as $R$.

In Section 4, we will use **o-minimal** models [33]. Recall that an ordered model $\mathcal{M}$ is o-minimal, if every definable set is a finite union of points and open intervals $\{x \mid a < x < b\}$. Definable sets are those of the form $\{x \mid \mathcal{M} \models \varphi(x)\}$, where $\varphi$ is a first-order formula in the language of $\mathcal{M}$, possibly supplemented with symbols for constants from $\mathcal{M}$.

A model admits quantifier elimination if, for every formula $\varphi(\vec{x})$, there is an equivalent quantifier-free formula $\psi(\vec{x})$ such that $\mathcal{M} \models \forall \vec{x}.\varphi(\vec{x}) \leftrightarrow \psi(\vec{x})$. A model is recursive if its language is recursive and validity of atomic sentences is decidable. An example of o-minimal, recursive model having quantifier elimination is the real field $\langle \mathbb{R}, +, *, 0, 1, < \rangle$.

**Logics** Since our goal is to develop a theory that can be used beyond the first-order case, we consider a variety of logics here. Fix a model $\mathcal{M} = \langle \mathcal{U}, \Omega \rangle$. By $L(SC, \Omega)$ we denote the language that consists of the schema predicates and the symbols in $\Omega$. By $\mathcal{FO}(SC, \mathcal{M})$ (or just $\mathcal{FO}(\mathcal{M})$ if the schema is understood) we denote the first-order logic over the language $L(SC, \Omega)$; we use $\mathcal{FO}$ for first-order logic in the language of the schema. We also define the semantic notion $D \models \varphi(\vec{a})$, where $\varphi(\vec{x})$ is a formula and $\vec{a}$ a vector over $\mathcal{U}$. Note that $D \models \exists x.\varphi(x, \vec{a})$ means that for some $a_0 \in \mathcal{U}$, $D \models \varphi(a_0, \vec{a})$. This corresponds to the **natural** interpretation of queries, cf. [1, 23].

Under the **active** interpretation of first-order logic [1, 23], $\exists x$ means that $x$ can be found within the active domain. To deal with this in the same framework as the natural interpretation (that is, to avoid introduction of a different notion of satisfaction), we introduce the **active quantification** of the form $\exists x \in adom.\varphi(x)$ and $\forall x \in adom.\varphi(x)$. The semantics is as follows: $D \models \exists x \in adom.\varphi(x, \vec{a})$ if for some $a_0 \in adom(D)$, $D \models \varphi(a_0, \vec{a})$, and likewise for $\forall x \in adom$. Note that this restricted quantification can be expressed in first-order logic. We shall call a formula **active** if all quantifiers in it are active. For every language $\mathbb{L}$, its subset consisting of the active formulae is denoted by $\mathbb{L}_{\text{act}}$.

Next, we consider fixpoint extensions of first-order logic. The presentation here differs slightly from the standard definitions for the finite [15, 21, 2] or infinite [30] case because we have a mix of these cases: finite structures over an infinite universe. (This corresponds to what we called **hybrid** logics in the introduction.) The **least-fixpoint** logic $\mathcal{FO}+\mathbf{lfp}$ adds the following construction rule: if $S$ is an $n$-ary relation symbol not in the schema, $\varphi(x_1, \ldots, x_n, \vec{y}, S)$ is a $L(SC \cup \{S\}, \Omega)$ formula where $S$ occurs positively, and $\vec{t}$ is an $n$-vector of variables or terms, then

$$[\mathrm{LFP}_{\vec{x},S}\varphi(\vec{x}, \vec{y}, S)](\vec{t})$$

is a formula, whose free variables are $\vec{y}$ and free variables from $\vec{t}$. The semantics is as follows. Given $D \in \mathit{Inst}(SC, \mathcal{U})$ and $\vec{a}$ of the same length as $\vec{y}$, define the sequence $R_0^{\vec{a}} = \emptyset$,

$$R_{i+1}^{\vec{a}} = \{(b_1, \ldots, b_n) \in adom(D)^n \mid D_{R_i^{\vec{a}}} \models \varphi(\vec{b}, \vec{a})\}.$$

It is known that this sequence increases, and thus reaches a fixpoint, denoted by $R_\infty^{\vec{a}}$. Now $D \models [\mathrm{LFP}_{\vec{x},S}\varphi(\vec{x}, \vec{a}, S)](\vec{c})$ iff $\vec{c} \in R_\infty^{\vec{a}}$. The least-fixpoint logic over the logical language $L(SC, \Omega)$ will be denoted by $\mathcal{FO}+\mathbf{lfp}(\mathcal{M})$.

Note that we may have instances of first-order quantification in least-fixpoint formulae, which are interpreted naturally, but the fixpoint itself is always taken within the active domain, which makes the logic hybrid: it combines both active and natural interpretations. In the pure finite case, it is unnecessary to add parameters $\vec{y}$ to the iterated formula, because it does not add power [15]. The same is true in the pure infinite case, that is, when $R_{i+1}^{\vec{a}}$ is constructed as the set of *all* vectors $\vec{b}$ over $\mathcal{U}$ that satisfy $\varphi(\vec{b}, \vec{a})$. But it is not clear whether the extra parameters $\vec{y}$ can be dropped in hybrid fixpoint logics without losing expressiveness. We shall say that a fixpoint formula is in **normal form** if no application of LFP uses these extra parameters $\vec{y}$.

The **inflationary-fixpoint** logic $\mathcal{FO}+\mathbf{ifp}$ and the **partial-fixpoint** logic $\mathcal{FO}+\mathbf{pfp}$ are defined similarly to the least-fixpoint logic. They have constructors

$$[\mathrm{IFP}_{\vec{x},S}\varphi(\vec{x}, \vec{y}, S)](\vec{t}) \quad \text{and} \quad [\mathrm{PFP}_{\vec{x},S}\varphi(\vec{x}, \vec{y}, S)](\vec{t})$$

respectively, with no restriction on $S$. The semantics is as above, where $R_\infty^{\vec{a}}$ is constructed as follows. For $\mathcal{FO}+\mathbf{ifp}$, $R_{i+1}^{\vec{a}} = R_i^{\vec{a}} \cup \{(b_1, \ldots, b_n) \in adom(D)^n \mid D_{R_i^{\vec{a}}} \models \varphi(\vec{b}, \vec{a})\}$ (this sequence is increasing, and hence reaches a fixpoint). For $\mathcal{FO}+\mathbf{pfp}$, the sequence $R_i$ is constructed as for the least-fixpoint, and $R_\infty$ is taken to be its fixpoint, if it exists, and $\emptyset$ otherwise.

The concept of a normal form is defined for inflationary- and partial-fixpoint logic in the same way it was defined for $\mathcal{FO}+\mathbf{lfp}$. When formulae of a fixpoint logic are restricted to normal form formulae only, we call such a fixpoint logic **closed**. In the case of finite domain, there is no loss of expressiveness due to restriction to the closed version of a fixpoint logic, cf. [1, 15].

We also consider logic with counting, $\mathcal{FO}+\mathbf{count}(\mathcal{M})$, following the presentation in [17]. This logic is two-sorted. One sort has $\mathcal{U}$ as its domain, the other sort has the domain $\mathbb{N}$. The logic adds second-sort quantifiers that range over the initial segment of $\mathbb{N}$ of the size of a finite instance, constants 0 and max (interpreted as the size of a finite instance minus 1), the BIT predicate, where $\mathrm{BIT}(i, j)$ is true if the $i$th bit of the binary representation of $j$ is 1, the order $<$ on the second sort, and finally counting quantifiers of the form $\exists ix.\varphi(x)$ which binds $x$ but not $i$, and means that there are at least $i$ satisfiers of $\varphi$ in the active domain. For example, a database $D$ satisfies $\exists i.(\mathrm{BIT}(i,0) \wedge \exists ix.\varphi(x) \wedge \forall j.(\exists jx.\varphi(x) \rightarrow j \leq i))$ iff the cardinality of $\{a \in adom(D) \mid D \models \varphi(a)\}$ is odd.

The **hybrid second-order logic** $\mathcal{HSO}$ permits second-order quantifiers $\exists S$ and $\forall S$ which are interpreted as follows: $D \models \exists S.\varphi(S)$, where $S$ is $k$-ary, iff there exists a set $R \subseteq adom(D)^k$ such that $D_R \models \varphi$. Note that this notion is even weaker than weak second-order, where second-order quantifiers range over finite sets. We use $\mathcal{SO}$ for full second-order logic ($k$-ary second-order quantifiers range over subsets of $\mathcal{U}^k$). Formulae of $\mathcal{SO}$ or $\mathcal{SO}_{\mathrm{act}}$ can be converted into normal form $Q_1 S^1 \ldots Q_k S^k.\psi$ where $Q_i S^i$ are second-order quantifiers, and $\psi$ is first-order. In the case of $\mathcal{HSO}$, it is not immediately clear if the same is true.

While all these logics are relevant to database query languages (as we shall see shortly), we shall also consider **infinitary** logic, which is of interest in finite-model theory, as logic which subsumes fixpoint logics and possesses nice properties, such as 0-1 law [27]. It is defined exactly as first-order logic, except that arbitrary disjunctions and conjunctions are allowed. That is, if $\{\varphi_i(\vec{x})\}$ is an arbitrary collection of formulae, then $\bigvee_i \varphi_i(\vec{x})$ and $\bigwedge_i \varphi_i(\vec{x})$ are formulae. We use $\mathcal{L}_{\infty\omega}$ to denote infinitary logic.

Suppose $\mathbb{L}$ is one of the logics introduced above, but the formation rules are modified so that only finitely many variables, $x_1, \ldots, x_k$, can be used in formulae. The restriction thus obtained is denoted by $\mathbb{L}^k$. For example, $\mathcal{L}^k_{\infty\omega}$ is infinitary logic with $k$ variables. We use $\mathcal{L}^\omega_{\infty\omega}$ for $\bigcup_{k\in\mathbb{N}} \mathcal{L}^k_{\infty\omega}$.

In the absence of interpreted symbols in $\Omega$, we speak of a **pure** logic (over a schema $SC$).

**Queries definable by logics** A **query** is a mapping from $Inst(SC_1)$ to $Inst(SC_2)$, where $SC_1$ and $SC_2$ are two schemas. For simplicity of exposition, assume that $SC_2$ consists of a single $p$-ary relation. Given a logic $\mathbb{L}$ and a model $\mathcal{M}$, we say that a query $Q$ is $\mathbb{L}$-definable over $\mathcal{M}$ (or $\mathbb{L}(\mathcal{M})$-definable) if there exists an $\mathbb{L}$ formula $\varphi(x_1, \ldots, x_p)$ in the language $L(SC_1, \Omega)$ such that $Q(D) = \{\vec{a} \mid D \models \varphi(\vec{a})\}$. We denote this query by $Q_\varphi$.

As in the case of relational calculus and algebra, we often consider queries that do not extend the active domain. Thus, we define the query $Q^{\mathrm{act}}_\varphi$ by $Q^{\mathrm{act}}_\varphi(D) = \{\vec{a} \in adom(D)^p \mid D \models \varphi(\vec{a})\}$. Note that any query $Q$ obtained in such a way is *domain-preserving*: $adom(Q(D)) \subseteq adom(D)$.

**Query languages** Relational calculus is just first-order logic over the database schema: its expressions are of the form $e = \{\vec{x} \mid \varphi(\vec{x})\}$ where $\varphi(\vec{x})$ is a $\mathcal{FO}$ formula in the language of the schema relations. By the Hull-Su theorem, we can use $\mathcal{FO}_{\mathrm{act}}$ expressions. We use CALC to denote the family of all calculus queries under the active interpretation (that is, $Q^{\mathrm{act}}_\varphi$), and ALG to denote relational algebra. It is a classical result of relational theory that CALC = ALG.

We consider DATALOG$^\neg$, which is datalog with negation allowed in bodies of rules. That is, a rule is of the form $H \coloneq B_1, \ldots, B_n$, $n \geq 0$, where each $B_i$ is an atom or a negated atom, and $H$ is an atom. Following [1, 2], we give it a simple inflationary semantics. That is, each iteration infers new facts and adds them to the facts already inferred; thus, a fixpoint is always reached.

We also consider the WHILE language. It extends ALG by allowing the statement *while change begin e end* where $e$ is an expression [2, 1, 12]. It iterates $e$ as long as it changes at least one relation. A WHILE statement is either an assignment of the form $V{:=}E$, where $V$ is a variable and $E$ is an expression, or a *while* expression above. A WHILE program is a sequence of *while* statements. See [1, 2] for more details.

**Equivalences for the finite domain** Query languages introduced here have been studied in depth in the classical relational theory. Many equivalence results are known in the pure finite case. By the pure finite case we mean this: the only free nonlogical symbols are the schema relations, and the universe is finite and coincides with the active domain. That is, a query does not extend the active domain, and all quantification is active. Below we list some of the most important equivalences.

First, CALC = ALG = $\mathcal{FO}$ [1]. Similar equivalences have been obtained in the case of interpreted operations (with some restrictions) given by abstract datatypes, see [5]. For fixpoint logics, $\mathcal{FO}$+**lfp** = $\mathcal{FO}$+**ifp** = DATALOG$^\neg$ [2, 21] and WHILE = $\mathcal{FO}$+**pfp** [2].

In the presence of an order relation, these equivalences continue to hold and, in addition, $\mathcal{FO}$+**lfp** captures all PTIME queries, and $\mathcal{FO}$+**pfp** captures all PSPACE queries [1, 15]. Also, in the presence of order, $\mathcal{FO}$+**count** captures uniform $\mathrm{TC}^0$ [4, 17].

It appears that $\mathcal{FO}$+**count** is close to relational languages with aggregates [29], but the precise connection is not fully understood. Second-order logic was shown to be relevant to the study of languages for complex objects, see [22]. $\mathcal{L}^\omega_{\infty\omega}$ is of interest in finite-model theory because it subsumes fixpoint logics. Until recently, variable-bounded logics were studied primarily in (finite) model theory, but [37] demonstrated nice connections with expression and combined complexity.

**Genericity** A query $Q$ (that is, a mapping from $Inst(SC_1, X)$ to $Inst(SC_2, X)$) is **totally generic** if, for any $D \in Inst(SC_1)$ and any injective map $\pi : adom(D) \cup adom(Q(D)) \rightarrow X$, it is the case that $Q(\pi(D)) = \pi(Q(D))$. A query is **locally generic** if $X$ is ordered, and the above holds for any injective monotone $\pi$. It is known that total genericity implies local [6]. Examples of generic (locally or totally) queries are any

ALG, DATALOG$^\neg$ or WHILE query, when no interpreted operations are present. Example of a locally but not totally generic query is $Q(S_1, S_2) \equiv \forall x \forall y. S_1(x) \wedge S_2(y) \rightarrow x < y$. Also note that totally and locally generic queries are domain-preserving: $adom(Q(D)) \subseteq adom(D)$, see [6]. For more of genericity, see [1] and [6].

## 3 Behind the iron curtain — active interpretation

The active interpretation does not allow a query to ask any question about what is outside of the finite database. That is, we live behind the iron curtain: the outside world – our infinite model – is there, we can even see a small part of it, but cannot ask much about it. So, how much more does the mere presence of the outside world add to the expressiveness of query languages? The answer is (and this is what one would expect) – practically nothing. We start by proving this result for a variety of logics. Then we show that many of the equivalences among languages continue to hold, when languages are appropriately modified. We derive some complexity corollaries, and also consider infinitary logic as a separate case. Finally, we mention one property of active semantics queries that holds, but not 'as before': the ability to test for query safety.

**Note:** in this section we only deal with active quantification, so we omit the $\in adom$ part of restricted quantifiers. Also, we write $\mathbb{L}$ instead of $\mathbb{L}_{\text{act}}$, since the natural semantics case is not considered in this section. Similarly, when we write $Q_\varphi$ for a query definable by the formula, we mean $Q_\varphi^{\text{act}}$, and omit the superscript.

### 3.1 Ramsey property and expressivity bounds

The main goal of this section is to prove generic collapse results for a number of logics. We say that a logic $\mathbb{L}$ has a **locally generic collapse** over an ordered model $\mathcal{M} = \langle \mathcal{U}, \Omega \rangle$ if, for every schema $SC$, every $\mathbb{L}(\mathcal{M})$-definable locally generic query on $SC$-databases, is already $\mathbb{L}(\langle \mathcal{U}, < \rangle)$-definable. That is, $\mathcal{M}$ is as expressive as just the order relation, with respect to locally generic queries. A logic $\mathbb{L}$ has a **generic collapse** over a model $\mathcal{M}$ if every $\mathbb{L}(\mathcal{M})$-definable totally generic query on $SC$-databases is definable in pure $\mathbb{L}$.

This problem of collapsing signatures for the active quantification was considered for first-order logic in [6] and independently in [31]. However, the techniques in [6] relied heavily on translation into prenex form, and the extension to second-order logic [7] was ad hoc. In [31], an elementary extension is used that possesses a set of indiscernibles, and it is unclear whether this technique works beyond the first-order case.

However, we show here that the technique of [6, 7] can be modified so that it can be applied to a variety of other logics. In particular, we show that a proof based on Ramsey's theorem [19], can proceed inductively on the structure of a formula, thus making it unnecessary to impose syntactic restrictions. Consequently, we get a series of results that give us expressivity bounds for logics under the active interpretation.

**Definition 1** *Let $\mathbb{L}$ be a logic. We say that it has a* **Ramsey property** *over an ordered model $\mathcal{M} = \langle \mathcal{U}, \Omega \rangle$ if, for any $SC$, the following is true:*

> *Let $\varphi(\vec{x})$ be an $\mathbb{L}$-formula in the language $L(SC, \Omega)$, and $X$ an infinite subset of $\mathcal{U}$. Then there exists an infinite set $Y \subseteq X$ and a $L(SC, <)$ formula $\psi(\vec{x})$ such that for any $D \in Inst(SC, Y)$ and any $\vec{a}$ over $Y$, it is the case that $D \models \varphi(\vec{a}) \leftrightarrow \psi(\vec{a})$.*

*We also speak of a formula $\varphi$ having the Ramsey property if the above is true. We speak of* **total** *Ramsey property if $\psi$ is a $L(SC)$ formula.*

As was shown previously [6, 7], the Ramsey property implies the following collapse for generic queries:

**Lemma 1 (Generic Collapse Lemma)**

1. *If $\mathbb{L}$ has the Ramsey property over $\mathcal{M} = \langle \mathcal{U}, \Omega \rangle$, and every $\mathbb{L}(\langle \mathcal{U}, < \rangle)$-query is locally generic, then $\mathbb{L}$ has the locally generic collapse over $\mathcal{M}$.*
2. *If $\mathbb{L}$ has the total Ramsey property over $\mathcal{M}$, and every $\mathbb{L}$-query is totally generic, then $\mathbb{L}$ has the generic collapse over $\mathcal{M}$.* □

The condition that every $\mathbb{L}(\langle \mathcal{U}, < \rangle)$-query is locally generic, and every $\mathbb{L}$-query is totally generic, holds for all the logics we introduced. Thus, to limit their expressiveness over infinite models, we have to prove the Ramsey property. First, we state a simple lemma that is often used as a first step in such proofs.

**Lemma 2 (Separation of atomic subformulae)** *Let $\varphi(\vec{x})$ be an $\mathbb{L}$ formula in the language $L(SC, \Omega)$, where $\mathbb{L}$ is one of the logics introduced in the previous section. Then there exists an equivalent formula $\psi(\vec{x})$ such that every atomic subformula of $\psi$ is either an $L(SC)$ formula, or a $L(\Omega)$ formula. Furthermore, for any set $\vec{z} \subseteq \vec{x}$ of free variables of $\varphi$, there is an equivalent formula $\psi(\vec{x})$ such that none of $\vec{z}$-variables occurs in an $L(SC)$-atomic formula.* □

The key in the inductive proofs of the Ramsey property is the case of $\Omega$-atomic subformulae. This was the key idea of the proof for the first-order case in [6], though the lemma below was not stated explicitly.

**Lemma 3** *Let $\mathcal{M} = \langle \mathcal{U}, \Omega \rangle$ be an infinite ordered model, and $\varphi(\vec{x})$ an atomic formula. Then $\varphi$ has the Ramsey property.* □

Now an inductive argument proves:

**Proposition 1** *The following have the Ramsey property: $\mathcal{FO}$; $\mathcal{FO}$+**lfp**; $\mathcal{FO}$+**ifp**; $\mathcal{FO}$+**pfp**; $\mathcal{FO}$+**count**; $\mathcal{SO}$.*

**Proof sketch.** Proof is by induction on the formula. By Lemma 2, assume that every atomic subformula is a $L(SC)$ formula or a $L(\Omega)$ formula. We start with the $\mathcal{FO}$ case. The basis follows from Lemma 3. If $\varphi(\vec{x}) = \varphi_1(\vec{x}) \wedge \varphi_2(\vec{x})$, and $X \subseteq \mathcal{U}$ is infinite, find $\psi_1$, $Y_1 \subseteq X$ such that for any $D$ and $\vec{a}$ over $Y_1$, $D \models \varphi_1(\vec{a}) \leftrightarrow \psi_1(\vec{a})$ and, by the hypothesis for $\varphi_2$ and $Y_1$, find $Y_2 \subseteq Y_1$ such that for any $D$ and $\vec{a}$ over $Y_2$, $D \models \varphi_2(\vec{a}) \leftrightarrow \psi_2(\vec{a})$. Now $\psi = \psi_1 \wedge \psi_2$ and $Y = Y_2$. The $\varphi = \neg\varphi'$ and $\varphi(\vec{x}) = \exists y.\varphi'(y, \vec{x})$ cases are similar.

Let us sketch the proof for $\mathcal{FO}$+**lfp**. In addition to doing the inductive step for LFP, we also note that all transformations preserve positiveness. Here, without loss of generality, we give the proof for a closed LFP formula applied to a vector of variables (see [8]). Let $S$ be a new $n$-ary relation symbol that occurs positively in a $L(SC \cup \{S\}, \Omega)$ formula $\alpha(\vec{x})$, and let $\varphi(\vec{y}) = [\mathrm{LFP}_{\vec{x},S}\alpha(\vec{x}, S)](\vec{z})$. Given infinite $X \subseteq \mathcal{U}$, we use the hypothesis to find an infinite $Y \subseteq X$ and a $L(SC \cup \{S\}, <)$ formula $\beta(\vec{x})$ such that for any $SC \cup \{S\}$-database $D_R \in \mathit{Inst}(SC \cup \{S\}, Y)$ it is the case that $D_R \models \alpha(\vec{a}) \leftrightarrow \beta(\vec{a})$ for all $\vec{a} \in Y^n$. Fix $D \in \mathit{Inst}(SC, Y)$ and $\vec{a} \in Y^n$, and define $R_0(\alpha) = \emptyset$, and $R_i(\alpha) = \{\vec{b} \mid \vec{b} \subseteq \mathit{adom}(D), D_{R_{i-1}(\alpha)} \models \alpha(\vec{b})\}$, and similarly define $R_i(\beta)$. It can be easily shown by induction on $i$ that $R_i(\alpha) = R_i(\beta)$ for all $i$. From this we derive $D \models [\mathrm{LFP}_{\vec{x},S}\alpha(\vec{x}, S)](\vec{a}) \leftrightarrow [\mathrm{LFP}_{\vec{x},S}\beta(\vec{x}, S)](\vec{a})$, as required. Proofs for other logics are similar. $\Box$

The main technique of the proof can easily be extended to other logics, (e.g., transitive closure logics [15]).

It is clear from the proof that only the case of atomic $L(\Omega)$ formulae requires the introduction of the order relation. Thus, if atomic $L(\Omega)$ formulae had the total Ramsey property over $\mathcal{M}$, so would all of the logics in the statement of Proposition 1. Following [7], we call a signature $\Omega$ **analytic** on $\mathbb{R}$ if it consists of real-analytic functions. For example, $(+, *)$ is an analytic signature. It follows from the results in [7] that any $L(\Omega)$ atomic formula has the total Ramsey property if $\Omega$ is analytic.

**Corollary 1** *If $\mathcal{M} = \langle \mathbb{R}, \Omega \rangle$, where $\Omega$ is analytic, and $\mathbb{L}$ is $\mathcal{FO}$, or $\mathcal{FO}$+**lfp**, or $\mathcal{FO}$+**pfp**, or $\mathcal{FO}$+**ifp**, or $\mathcal{FO}$+**count**, or $\mathcal{SO}$, then $\mathbb{L}$ has the total Ramsey property over $\mathcal{M}$.* $\Box$

From Proposition 1 and corollary 1 we obtain the main result of this subsection.

**Theorem 1** *Let $\mathbb{L}$ be $\mathcal{FO}$, or $\mathcal{FO}$+**lfp**, or $\mathcal{FO}$+**pfp**, or $\mathcal{FO}$+**ifp**, or $\mathcal{FO}$+**count**, or $\mathcal{SO}$. Let $\mathcal{M} = \langle \mathcal{U}, \Omega \rangle$ be an arbitrary ordered model. Then $\mathbb{L}$ has locally generic collapse over $\mathcal{M}$. If $\mathcal{M} = \langle \mathbb{R}, \Omega \rangle$, where $\Omega$ is analytic, then $\mathbb{L}$ has generic collapse over $\mathcal{M}$.* $\Box$

This result can be combined with classical results in finite-model theory and descriptive complexity (cf. [1, 15]) to obtain a large number of corollaries that give expressivity bounds for the languages of the form $\mathbb{L}(\mathcal{M})$ under the active interpretation. For example,

**Corollary 2** *Let $\mathcal{M} = \langle \mathcal{U}, \Omega \rangle$ be an ordered model. Then the class of locally generic queries definable in $\mathbb{L}(\mathcal{M})$, where $\mathbb{L}$ is $\mathcal{FO}$, $\mathcal{FO}$+**count**, $\mathcal{FO}$+**lfp**, $\mathcal{FO}$+**ifp**, $\mathcal{FO}$+**pfp** and $\mathcal{SO}$, is the class of all uniform $\mathrm{AC}^0$, uniform $\mathrm{TC}^0$, PTIME, PTIME, PSPACE and PH queries, respectively.* $\Box$

From the 0-1 law for $\mathcal{FO}$+**pfp** [26], one can get

**Corollary 3** *Parity test is not definable in $\mathcal{FO}$+**pfp**$(\langle \mathbb{R}+, * \rangle)$.* $\Box$

### 3.2 Equivalence results

In this section we show that a number of well-known results on equivalence between logics and relational query languages generalize straightforwardly in the presence of interpreted structures.

We define the calculus over $\mathcal{M}$, denoted by CALC($\mathcal{M}$) simply as $\mathcal{FO}(\mathcal{M})$. More precisely, its expressions are of the form $e = \{\vec{x} \mid \varphi(\vec{x})\}$, where $\varphi$ is a $\mathcal{FO}_{\mathrm{act}}(\mathcal{M})$ formula. An algebra over $\mathcal{M}$, denoted by ALG($\mathcal{M}$), contains all the same operations as relational algebra ALG; the only difference is the selection predicates. Define *selection terms* by the grammar $st := \#i \mid f(st, \ldots, st)$ where $f$ ranges over the function symbols in $\Omega$. Then, *selection conditions* are given by $sc := C(st, \ldots, st) \mid st = st \mid \neg sc \mid sc \vee sc$, where $C$ ranges over the predicates in $\Omega$. For example, $\sigma_{\#1*\#2>\#1+\#3}(R)$ is an algebra expression that selects triples $(x, y, z)$ from $R$ such that $x * y > x + z$. Similar extensions exist in the literature, see, for example, [16]. A simple extension of the classical equivalence CALC = ALG yields

**Proposition 2** *For any $\mathcal{M}$, CALC($\mathcal{M}$) = ALG($\mathcal{M}$).* $\Box$

Next, we consider DATALOG$^\neg$($\mathcal{M}$), which extends DATALOG$^\neg$ by allowing $L(\Omega)$-atomic formulae in the bodies of rules. For example, if $\mathcal{U} = \mathbb{R}$ and $\Omega$ contains addition, then the following DATALOG$^\neg$($\mathcal{M}$) program

$$\begin{array}{lcl} R(x,y) & :- & E(x,y), x > y + y \\ R(x,y) & :- & E(x,z), x > z + z, R(z,y) \end{array}$$

defines the transitive closure of a subgraph that consists of the edges $(x, y)$ with $x > 2y$.

**Proposition 3** *For any* $\mathcal{M}$, $\text{DATALOG}^{\neg}(\mathcal{M}) = \mathcal{FO}+\mathbf{ifp}(\mathcal{M}) = \mathcal{FO}+\mathbf{lfp}(\mathcal{M})$. □

$\text{WHILE}(\mathcal{M})$ extends WHILE by allowing $\text{ALG}(\mathcal{M})$ expressions in place of ALG expressions.

**Proposition 4** *For any* $\mathcal{M}$, $\text{WHILE}(\mathcal{M}) = \mathcal{FO}+\mathbf{pfp}(\mathcal{M})$. □

We now combine these results with the Ramsey technique to get expressivity and complexity bounds on the active queries over interpreted structures.

**Theorem 2**

1. *Let $\mathcal{M}$ be an arbitrary ordered infinite model. Then, for locally generic queries,* $\text{ALG}(\mathcal{M}) = \mathcal{FO}_{\text{act}}(<)$, $\text{DATALOG}^{\neg}(\mathcal{M}) = \textit{PTIME}$ *and* $\text{WHILE}(\mathcal{M}) = \textit{PSPACE}$.
2. *If $\mathcal{M} = \langle \mathbb{R}, \Omega \rangle$ where $\Omega$ consists of real-analytic functions, then every totally generic query in* $\text{ALG}(\mathcal{M})$ *is* $\mathcal{FO}_{\text{act}}$*-definable, and every totally generic query in* $\text{DATALOG}^{\neg}(\mathcal{M})$ *and* $\text{WHILE}(\mathcal{M})$ *has PTIME (resp., PSPACE) data complexity. Furthermore, the parity test is not definable in any of these languages.* □

### 3.3 Infinitary logic

Here we extend our results to infinitary logic. We are not interested in the full infinitary logic $\mathcal{L}_{\infty\omega}$, nor $\mathcal{L}^{\omega}_{\infty\omega}$ over ordered structures, because they express every property of finite structures [14]. Thus, we concentrate on $\mathcal{L}^{\omega}_{\infty\omega}$ over unordered models.

We cannot use the inductive argument of Proposition 1 anymore, because it does not work for infinitary formulae. Indeed, for infinitary disjunction $\bigvee \varphi_i$, one would construct a decreasing family of infinite sets $X_1 \supseteq X_2 \supseteq \ldots$, but its intersection $\bigcap_i X_i$ is not guaranteed to be infinite. Thus, we use the approach that is closer to the proof of the collapse of generic queries for $\mathcal{FO}$ in [31]. We modify the argument in [7] to show that every uncountable subset of $\mathbb{R}$ has a set of total indiscernibles [13] with respect to a countable collection of formulae, if the signature is analytic. This gives us:

**Proposition 5** *Let $\mathcal{M} = \langle \mathbb{R}, \Omega \rangle$ where $\Omega$ is analytic, and has countably many symbols. Then $\mathcal{L}^{\omega}_{\infty\omega}$ has generic collapse over $\mathcal{M}$.* □

Using the 0-1 law for infinitary logic [27], we obtain:

**Corollary 4** *The parity test is not definable as a $\mathcal{L}^{\omega}_{\infty\omega}(\langle \mathbb{R}, +, * \rangle)$ query.* □

### 3.4 When interpreted structure matters: query safety

We now want to give the reader first indication that the kind of an interpreted structure one adds, can make a difference, even in the active case. Instead of $Q^{\text{act}}_{\varphi}(D) = \{\vec{a} \in \mathit{adom}(D)^n \mid D \models \varphi(\vec{a})\}$, one could consider $Q_{\varphi}(D) = \{\vec{a} \in \mathcal{U}^n \mid D \models \varphi(\vec{a})\}$. Unfortunately, this may fail to define a query, because $Q_{\varphi}(D)$ may be infinite. This is the classical problem of query safety. Following [25], we say that a formula $\varphi(\vec{x})$ in $L(SC, \Omega)$ is **safe** for an instance $D$ if $Q_{\varphi}(D)$ is finite.

**Proposition 6** *Let $\mathcal{M} = \langle \mathbb{R}, +, *, 0, 1, < \rangle$. Then there is a recursive function that takes an active $\mathcal{FO}(\mathcal{M}, SC)$ formula $\varphi(\vec{z})$, and outputs another active formula $\varphi_{\text{safe}}(\vec{z})$ such that*

$$Q_{\varphi_{\text{safe}}}(D) = \begin{cases} Q_{\varphi}(D) & \text{if } \varphi \text{ is safe for } D \\ \emptyset & \text{otherwise} \end{cases}$$

We will prove this in the next section, via a detour into the natural semantics. In the full paper, we will also show that this result does *not* hold for an arbitrary interpreted structure.

## 4 The infinite world – natural interpretation

So far we have produced a set of techniques that can be applied to analyze expressiveness and complexity of active-semantics queries in a variety of languages. The next question is: how does one approach the case of natural queries? It appears that this case is "infinitely" harder than the active case, because now we can ask questions about *any* element in the universe: the iron curtain of the finite active domain, that limited our visibility, no longer applies.

**Note**: 1) The proviso of the previous section that only active formulae are considered and that $\mathbb{L}$ is used in place of $\mathbb{L}_{\text{act}}$ is **not** in force in this section.
2) We say that a logic $\mathbb{L}$ admits the **natural-active collapse** over $\mathcal{M}$ if $\mathbb{L}(\mathcal{M}) = \mathbb{L}_{\text{act}}(\mathcal{M})$; that is, for every schema and every $\mathbb{L}$-formula $\varphi(\vec{x})$ in the language $L(SC, \Omega)$, there exists an equivalent active formula $\psi(\vec{x})$ in the same language.

The Hull-Su theorem states that pure $\mathcal{FO}$ admits the natural-active collapse. But this result is not robust: if $\mathcal{N} = \langle \mathbb{N}, +, *, 0, 1 \rangle$, then *every* recursive query is definable in $\mathcal{FO}(\mathcal{N})$ [20], but $\mathcal{FO}_{\text{act}}(\mathcal{N})$ cannot express parity; thus $\mathcal{FO}_{\text{act}}(\mathcal{N}) \neq \mathcal{FO}(\mathcal{N})$.

It is known that $\mathcal{FO}$ admits the natural-active collapse over $\langle \mathbb{R}, +, -, 0, 1, < \rangle$ [32] and over the real field [7]. One might ask whether a similar result holds for higher-order logics. It was shown in [7] that this is not true for the full second-order logic $\mathcal{SO}$. The reason is the same as above: the set of natural numbers can be defined, together with arithmetic operations, and thus coding can be done. Similarly, full fixpoint logics will not exhibit a natural-active collapse. However, the question whether *hybrid* logics admit such a collapse is open.

Thus, our goal is to see to what extent we can recover the natural-active collapse for relational languages for databases over infinite interpreted structures. In addition to proving such results, we are interested in finding *algorithms* that convert natural queries into active queries. We start by giving a new proof of the Hull-Su theorem, that can then be extended to show that $\mathcal{FO}^k$ collapses to $\mathcal{FO}^k_{\text{act}}$. We extend this result to infinitary logic. We then present our main result, which is a *constructive* proof that $\mathcal{HSO}(\mathcal{M}) = \mathcal{HSO}_{\text{act}}(\mathcal{M})$ when $\mathcal{M}$ is o-minimal and admits quantifier elimination. (The proof is constructive if the quantifier elimination can be done effectively.) This implies that $\mathcal{FO}$, $\mathcal{FO}+\mathbf{lfp}$ and $\mathcal{FO}+\mathbf{ifp}$ admit the collapse over such models, and also leads to normal forms for hybrid logics.

### 4.1 Natural-active collapse in the pure case

Our goal is to have a set of general algorithms for collapsing natural queries to active over interpreted structures. We start with the pure case, and give new algorithms for several logics. We also give a simple constructive proof of the Hull-Su theorem. Theorem 4 refines it to work with variable-bounded logics. These ideas will be expanded upon to deal with interpreted structures. The original proof in [23] is algorithmic but quite complex. In a recent unpublished manuscript [10], a simpler proof is given that uses many-sorted logic. Below we sketch a simple direct proof.

**Theorem 3 (Hull-Su)** $\mathcal{FO} = \mathcal{FO}_{\text{act}}$.

**Proof sketch.** Proof is by induction on the structure of the formula. The cases of atomic formulae and Boolean connectives are obvious. For the existential case, we define transformation $[\gamma]^x$ that eliminates all free occurrences of variable $x$:

If $\gamma$ is $(x = x)$, then $[\gamma]^x = \mathbf{T}$;
If $\gamma$ is $(x = y)$ or $R(\ldots, x, \ldots)$, then $[\gamma]^x = \mathbf{F}$;
If $\gamma$ is any other atomic formula, then $[\gamma]^x = \gamma$;
If $\gamma = \gamma_1 \vee \gamma_2$, then $[\gamma]^x = [\gamma_1]^x \vee [\gamma_2]^x$;
If $\gamma = \neg\gamma'$, then $[\gamma]^x = [\neg\gamma']^x$;
If $\gamma = \exists y \in adom.\gamma'$, then $[\gamma]^x = \exists y \in adom.[\gamma']^x$.

Let $\varphi(\vec{z}) = \exists x.\alpha(x, \vec{z})$ where $z = (z_1, \ldots, z_n)$. By the hypothesis, $\alpha$ is equivalent to an active formula $\alpha'(x, \vec{z})$. Assume that $\alpha'$ is in prenex form; in particular, $x$ cannot be a bound variable in any subformula of $\alpha'$.

Define $\varphi_0(\vec{z}) = \exists x \in adom.\alpha'(x, \vec{z})$, $\varphi_i(\vec{z}) = \alpha'(z_i, \vec{z})$ and $\varphi_\infty(\vec{z}) = [\alpha'(x, \vec{z})]^x$. Let $\varphi'(\vec{z}) = \varphi_0 \vee (\bigvee_{i=1}^n \varphi_i) \vee \varphi_\infty$. Then a simple induction argument shows that $D \models \varphi(\vec{a}) \leftrightarrow \varphi'(\vec{a})$ for every instance $D$ and $\vec{a} \in \mathcal{U}^n$. $\Box$

The idea behind this proof can be implemented a bit more carefully to yield the following stronger result:

**Theorem 4** *$\mathcal{FO}^k$ and $\mathcal{L}^k_{\infty\omega}$ admit the natural-active collapse: $\mathcal{FO}^k = \mathcal{FO}^k_{\text{act}}$ and $\mathcal{L}^k_{\infty\omega} = (\mathcal{L}^k_{\infty\omega})_{\text{act}}$.* $\Box$

**Corollary 5** *Pure $\mathcal{L}^\omega_{\infty\omega}$ admits the natural-active collapse.* $\Box$

### 4.2 Natural-active collapse over interpreted structures: The algorithm

Recall that an ordered model $\mathcal{M} = \langle \mathcal{U}, \Omega \rangle$ is **o-minimal** if every definable set is a finite union of points and open intervals. Definable sets are those of the form $\{x \in \mathcal{U} \mid \mathcal{M} \models \varphi(x)\}$ where $\varphi$ is a first-order formula in the language of $\Omega$ and constants for elements of $\mathcal{U}$.

Our goal is to give a *constructive* proof of the following result from which the natural-active collapse for first-order and some other logics will follow:

**Theorem 5 (Natural-active collapse)** *Let $\mathcal{M} = \langle \mathcal{U}, \Omega \rangle$ be an o-minimal model that admits quantifier elimination. Then $\mathcal{HSO}(\mathcal{M})$ admits the natural-active collapse over $\mathcal{M}$. That is, for every schema $SC$, and for every $\mathcal{HSO}$-formula $\varphi(\vec{x})$ in the language $L(SC, \Omega)$ without free second-order variables, there exists an equivalent active $\mathcal{HSO}$-formula $\varphi_{\text{act}}(\vec{x})$ in the same language. Moreover, if $\mathcal{M}$ is recursive and the quantifier elimination procedure is effective, then the transformation from $\varphi$ to $\varphi_{\text{act}}$ is effective.*

To present an algorithm, we need the following

**Fact 1** (see [33]) *If $\mathcal{M}$ is o-minimal, and $\varphi(x, \vec{y})$ is a first-order formula in the language of $\mathcal{M}$, possibly supplemented with symbols for constants from $\mathcal{M}$, then there is an integer $K$ such that, for each vector $\vec{a}$ from $\mathcal{M}$, the set $\{x \mid \mathcal{M} \models \varphi(x, \vec{a})\}$ is composed of fewer than $K$ intervals.*

**Algorithm for transforming $\varphi$ into $\varphi_{\text{act}}$.** The algorithm is recursive on the structure of the formula. Fix a schema $SC$. If $\varphi$ is atomic, then $\varphi_{\text{act}} = \varphi$. If $\varphi = \neg\alpha$, then $\varphi_{\text{act}} = \neg\alpha_{\text{act}}$. If $\varphi = \alpha \vee \beta$, then $\varphi_{\text{act}} = \alpha_{\text{act}} \vee \beta_{\text{act}}$. If $\varphi = \exists S.\psi$, then $\varphi_{\text{act}} = \exists S.\psi_{\text{act}}$.

Let $\varphi(\vec{x}) = \exists z.\alpha(z, \vec{x})$. First, we recursively apply the transformation to get an active formula $\alpha_{\text{act}}(z, \vec{x})$. Next, using Lemma 2, we transform $\alpha_{\text{act}}$ into an equivalent formula $\beta(z, \vec{x})$ such that that each atomic subformula of $\beta$ is either a $L(SC)$ formula or a $L(\Omega)$ formula, and $z$ occurs only in $L(\Omega)$ atomic formulae.

Let $\Psi$ be the collection of all $L(\Omega)$ atomic subformulae of $\beta$. Let $\vec{y}$ be the collection of $n$ bound variables used in $\beta$. Then for each $\rho(\vec{x},\vec{y},z)$ in $\Psi$, find the number $K_\rho$ such that $S_\rho(\vec{b},\vec{c}) = \{a \mid \mathcal{M} \models \rho(\vec{b},\vec{c},a)\}$ is composed of fewer than $K_\rho$ intervals for every vectors $\vec{b}$ and $\vec{c}$ of elements of $\mathcal{U}$. Note that this number is computable if $\mathcal{M}$ is recursive and the quantifier elimination is effective. For every $k$, the statement that $\{a \mid \mathcal{M} \models \rho(\vec{b},\vec{c},a)\}$ is composed of fewer than $k$ intervals for all $\vec{b},\vec{c}$ can be written as a first-order sentence $\rho_k$ in the language of $\mathcal{M}$. Applying quantifier elimination to $\rho_k$ one can see if it is true or not. Since $\rho_k$ is true for some $k$ (by fact 1), to find $K_\rho$ one should keep checking $\rho_1,\rho_2,\rho_3,\ldots$ until the one evaluating to true is found, which gives us an algorithm for finding $K_\rho$. Now let $K = \max_{\rho\in\Psi}\{K_\rho, K_{\neg\rho}\}$.

Given two vectors $\vec{b},\vec{c}$ of the same arity as $\vec{x},\vec{y}$, we say that an interval $S$ is a $(\vec{b},\vec{c})$-*interval* if either $S \subset S_\rho(\vec{b},\vec{c})$, or $S \subset \mathcal{U} - S_\rho(\vec{b},\vec{c})$ for every $\rho \in \Psi$, and $S$ is a maximal interval with this property. Then there exists an integer $J$ such that there are at most $J$ $(\vec{b},\vec{c})$-intervals for each pair $(\vec{b},\vec{c})$. In fact, $J$ can be taken to be $(4K+1)^{|\Psi|}$; this follows from the uniform bound $K$ on the number of intervals in the sets $S_\rho$.

Thus, for each $j < J$, there is a first-order formula $\gamma_j(\vec{x},\vec{y},z)$ saying that there are at least $j$ $(\vec{x},\vec{y})$-intervals, and $z$ is inside the $j$th one. Next, we let $\chi_{ij}(\vec{s},\vec{t},\vec{x})$ be a quantifier-free formula equivalent to

$$\exists u.\gamma_i(\vec{x},\vec{s},u) \wedge \gamma_j(\vec{x},\vec{t},u)$$

and $\rho_{ij}(\vec{s},\vec{t},\vec{x},\vec{y})$ a quantifier-free formula equivalent to

$$\forall u.(\gamma_i(\vec{x},\vec{s},u) \wedge \gamma_j(\vec{x},\vec{t},u)) \to \rho(\vec{x},\vec{y},u),$$

where $\vec{s}$ and $\vec{t}$ are $n$-element vectors. Let $\beta_{ij}$ be the formula obtained from $\beta$ by replacing each $\rho(\vec{x},\vec{y},z)$ in $\Psi$ with $\rho_{ij}(\vec{s},\vec{t},\vec{x},\vec{y})$.

Let us use $\delta_{ij}(\vec{x},\vec{s},\vec{t},u)$ as an abbreviation for $\gamma_i(\vec{x},\vec{s},u) \wedge \gamma_j(\vec{x},\vec{t},u)$. Let $\eta_{ij}(\vec{s},\vec{t},\vec{r},\vec{x})$ be the quantifier-free formula equivalent to

$$\begin{array}{ll}\forall u\forall u'. & [(\delta_{ij}(\vec{x},\vec{s},\vec{t},u) \wedge \delta_{ij}(\vec{x},\vec{s},\vec{t},u')) \\ \to & \bigwedge_{\rho\in\Psi} (\rho(\vec{x},\vec{r},u) \leftrightarrow \rho(\vec{x},\vec{r},u'))]\end{array}$$

Let $\pi_{ij}(\vec{s},\vec{t},\vec{x})$ be $\forall\vec{r}\in adom.\eta_{ij}(\vec{s},\vec{t},\vec{r},\vec{x})$. It says that any $u$ and $u'$ in the intersection of the $i$th $(\vec{x},\vec{s})$-interval and the $j$th $(\vec{x},\vec{t})$-interval have the property that for any other vector $\vec{r}$ of elements of the active domain, and any atomic formula $\rho$ in $\Psi$, it is the case that $\rho(\vec{x},\vec{r},u)$ and $\rho(\vec{x},\vec{r},u')$ have the same truth value.

Finally, we output, as $\varphi_{\mathrm{act}}$, the following formula:

$$\begin{array}{l}\exists s_1\in adom\ldots\exists s_n\in adom\exists t_1\in adom\ldots\exists t_n\in adom. \\ \bigvee_{i,j<J} \left(\ \chi_{ij}(\vec{s},\vec{t},\vec{x}) \wedge \pi_{ij}(\vec{s},\vec{t},\vec{x}) \wedge \beta_{ij}(\vec{s},\vec{t},\vec{x})\ \right)\end{array}$$

where $\vec{s} = (s_1,\ldots,s_n)$ and $\vec{t} = (t_1,\ldots,t_n)$.

Theorem 5 now follows from

**Lemma 4 (Correctness of the algorithm)** *For every $\mathcal{HSO}(\mathcal{M})$ formula $\varphi(\vec{x})$ without free second-order variables, and every nonempty database instance, $D \models \forall\vec{x}.\varphi(\vec{x}) \leftrightarrow \varphi_{\mathrm{act}}(\vec{x})$.* □

It is well known that for any $\mathcal{SO}$ or $\mathcal{SO}_{\mathrm{act}}$ formula, there exists an equivalent one in normal form (second-order quantifiers are in front of first-order quantifiers). Theorem 5 gives us the normal form for $\mathcal{HSO}$.

**Corollary 6 (Normal form for $\mathcal{HSO}$)** *Let $\mathcal{M}$ be o-minimal. Then every $\mathcal{HSO}(\mathcal{M})$-formula is equivalent to a $\mathcal{HSO}(\mathcal{M})$-formula in normal form.* □

### 4.3 The natural-active collapse for first-order queries

The following is an immediate corollary to Theorem 5.

**Theorem 6** *Let $\mathcal{M}$ be an o-minimal model that admits quantifier elimination. Then $\mathcal{FO}$ admits the natural-active collapse over $\mathcal{M}$. Moreover, the transformation from a natural formula to an active formula is effective if $\mathcal{M}$ is recursive and the quantifier elimination procedure is effective.* □

From Tarski's quantifier-elimination, we get

**Corollary 7** *There is an algorithm that converts any $\mathcal{FO}(\langle\mathbb{R},+,*,0,1,<\rangle)$-query into an equivalent $\mathcal{FO}_{\mathrm{act}}(\langle\mathbb{R},+,*,0,1,<\rangle)$-query.* □

Finally, we have an elementary proof of

**Corollary 8** (see [6]) *Parity test cannot be defined as a relational calculus query with polynomial inequality constraints over the reals.* □

Now we can complete the proof of proposition 6. Note that if the order $<$ is dense, then for a given formula $\varphi(\vec{z})$ it is possible to write a (natural semantics) sentence $\Phi$ that tests if the number of satisfiers of $\varphi$ is finite. This is done by testing if each projection of the set of satisfiers of $\varphi$ is bounded and discrete (also, endpoints must be considered with care). By Theorem 6, $\Phi$ can be converted into an equivalent active sentence $\Phi_{\mathrm{act}}$, and then $\varphi(\vec{z}) \wedge \Phi_{\mathrm{act}}$ can be taken as $\varphi_{\mathrm{safe}}(\vec{z})$. □

### 4.4 Natural-active collapse for fixpoint queries

From Theorem 5, we derive

**Theorem 7** *Let $\mathcal{M}$ be an o-minimal model that admits quantifier elimination. Then $\mathcal{FO}+\mathbf{ifp}$ and $\mathcal{FO}+\mathbf{lfp}$ admit the natural-active collapse over $\mathcal{M}$. Moreover, the transformation from a natural formula to an active formula is effective if $\mathcal{M}$ is recursive and the quantifier elimination procedure is effective.* □

This theorem cannot be extended to partial-fixpoint logic, because it is proved by embedding inflationary (or least) fixpoint into second-order logic. However, we can prove the following for closed partial- and inflationary-fixpoints. Recall that by *closedness* of a fixpoint formula we mean that no application of fixpoint involves extra free variables, that is, it is of the form $[\mathrm{LFP}_{\vec{x},S}\varphi(\vec{x},S)](\vec{t})$, or similarly for IFP and PFP.

**Theorem 8** *Let $\mathcal{M}$ be a model such that $\mathcal{FO}$ admits the natural-active collapse over $\mathcal{M}$. Then the closed $\mathcal{FO}+\mathbf{ifp}(\mathcal{M})$ and the closed $\mathcal{FO}+\mathbf{pfp}(\mathcal{M})$ admit the natural-active collapse.* □

Finally, combining the results of this section, we see that for some models $\mathcal{M}$, there is no difference between closed and unrestricted hybrid fixpoint logics, which gives us the desired normal form result.

**Corollary 9 (Normal form for fixpoint)** *If $\mathcal{M}$ is o-minimal and has quantifier elimination, then*

$$\begin{aligned}\mathcal{FO}+\mathbf{lfp}(\mathcal{M}) &= \text{closed } \mathcal{FO}+\mathbf{lfp}(\mathcal{M})\\ \mathcal{FO}+\mathbf{ifp}(\mathcal{M}) &= \text{closed } \mathcal{FO}+\mathbf{ifp}(\mathcal{M}).\end{aligned}$$

Using a simple induction argument, we show

**Proposition 7** *Let $\mathcal{M}$ be a model such that $\mathcal{FO}$ admits the natural-active collapse over $\mathcal{M}$. Then $\mathcal{FO}+\mathbf{count}(\mathcal{M})$ admits the natural-active collapse.* □

### 4.5 Summing up

Figure 1 puts together results about both active and natural semantics. Roughly, each line in Figure 1 shows the equivalence of a logic $\mathbb{L}$, its active-semantics version $\mathbb{L}_{\mathrm{act}}$, a procedural language (if one exists), and a complexity class. Here we use classical descriptive complexity results (cf. [1, 15]). Note that by $\mathrm{AC}^0$ and $\mathrm{TC}^0$ we mean their uniform versions, cf. [4].

Techniques developed here make it easy to prove results for other logics. For example, one can show that for the hybrid version of the transitive closure logic [15], the natural and the active interpretations coincide over the real field, and by the Ramsey property, the class of generic queries definable in it under the active interpretation does not depend on the set of operations $\Omega$, and thus is precisely the class of NLOGSPACE queries.

## 5 Conclusion

Our main goal was to delineate the extent to which standard results from the pure relational case 'go through as before' when interpreted structures are present. We have distinguished several aspects of the standard theory that extend routinely, as well as several whose extension requires special techniques and/or special assumptions on the interpreted structure. That is, we have three categories of results.

First, extensions of results from pure relational theory to any interpreted structure. We show that most of the standard language equivalences, expressive bounds and complexity results continue to hold with interpreted structures, assuming the active semantics. Second, extensions of results from the pure case that hold with interpreted structure under special assumptions on the structure. We show that most statements about the natural semantics continue to hold for reasonably-behaved interpreted structures. Third, results that arise in the interpreted case that either do not arise or are not of interest in the pure case. We introduced a class of hybrid logics, that arise naturally in the interpreted case: those with 'mixed' quantification. These logics admit the same normal forms as their unmixed counterparts, if the structures are reasonably behaved.

This is only the tip of the iceberg for each of these classes. The understanding of appropriate algebras for queries with interpreted structures is still incomplete. We are working on algebras and range-restricted calculi for safe queries, other normal forms for nonboolean queries and operations that preserve safety. Although we have shown that for nicely-behaved structures we get very satisfying extensions of results about the natural semantics, we do not *characterize* the class of structures for which this holds. It is also open whether the normal forms results for hybrid logics hold in general.

The algorithm for converting natural quantification to active is of interest in its own right. It extends work done in the constraint community (e.g., in [32] for linear constraints), and can also be seen as extending quantifier elimination algorithms, that originate with Tarski [35], to handle large parameter sets. We would like to have a unified picture of the relation between the algorithm presented here and those in, for example, [9, 32]. We plan to study the integration of such algorithms into optimization systems for constraint queries.

We are interested in understanding the connection between our results and metafinite model theory [18]. Some of the motivations for [18] are very close to those for our work, but it does not appear that we can use any of the results in [18] to derive any of our results.

**Acknowledgements** We thank Victor Vianu and Scott Weinstein for their prompt answers to our questions, Rick Hull and anonymous reviewers for their comments, and Jan Van den Bussche for a copy of [10].

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | $\mathcal{FO}(\mathcal{M})$ | $=$ | $\mathcal{FO}_{\mathrm{act}}(\mathcal{M})$ | $_{\mathrm{dp}}=$ | $\textsc{Alg}(\mathcal{M})$ | $_{\mathrm{lg}}=$ | $\mathrm{AC}^0$ |
| | | $\mathcal{FO}+\mathbf{count}(\mathcal{M})$ | $=$ | $\mathcal{FO}+\mathbf{count}_{\mathrm{act}}(\mathcal{M})$ | | | $_{\mathrm{lg}}=$ | $\mathrm{TC}^0$ |
| $\mathcal{FO}+\mathbf{lfp}(\mathcal{M})$ | $=$ | closed $\mathcal{FO}+\mathbf{lfp}(\mathcal{M})$ | $=$ | $\mathcal{FO}+\mathbf{lfp}_{\mathrm{act}}(\mathcal{M})$ | $_{\mathrm{dp}}=$ | $\textsc{Datalog}^{\neg}(\mathcal{M})$ | $_{\mathrm{lg}}=$ | PTIME |
| $\mathcal{FO}+\mathbf{ifp}(\mathcal{M})$ | $=$ | closed $\mathcal{FO}+\mathbf{ifp}(\mathcal{M})$ | $=$ | $\mathcal{FO}+\mathbf{ifp}_{\mathrm{act}}(\mathcal{M})$ | $_{\mathrm{dp}}=$ | $\textsc{Datalog}^{\neg}(\mathcal{M})$ | $_{\mathrm{lg}}=$ | PTIME |
| | | $\mathcal{HSO}(\mathcal{M})$ | $=$ | $\mathcal{HSO}_{\mathrm{act}}(\mathcal{M})$ | | | $_{\mathrm{lg}}=$ | PH |
| | | closed $\mathcal{FO}+\mathbf{pfp}(\mathcal{M})$ | $=$ | $\mathcal{FO}+\mathbf{pfp}_{\mathrm{act}}(\mathcal{M})$ | $_{\mathrm{dp}}=$ | $\textsc{While}(\mathcal{M})$ | $_{\mathrm{lg}}=$ | PSPACE |

Note: $\mathcal{C}\ _{\mathrm{lg}}=\ \mathcal{C}'$ means the class of locally generic queries in $\mathcal{C}$ is $\mathcal{C}'$.
$\mathcal{C}\ _{\mathrm{dp}}=\ \mathcal{C}'$ means the class of domain preserving $(adom(Q(D)) \subseteq adom(D))$ queries in $\mathcal{C}$ is $\mathcal{C}'$.

Figure 1: Summary of the expressiveness results for $\mathcal{M} = \langle \mathbb{R}, +, *, 0, 1, < \rangle$ or any other o-minimal model that admits quantifier elimination

## References


[1] S. Abiteboul, R. Hull and V. Vianu. *Foundations of Databases.* Addison-Wesley, 1995.

[2] S. Abiteboul and V. Vianu. Datalog extensions for database queries and updates. *JCSS* 43 (1991), 62–124.

[3] A.K. Ailamazyan, M.M. Gilula, A.P. Stolboushkin and G.F. Shvarts. Reduction of a relational model with infinite domains to the finite-domain case. *Soviet Physics – Doklady*, 31 (1986), 11–13.

[4] D.A. Barrington, N. Immerman, H. Straubing. On uniformity within $NC^1$. *JCSS*, 41:274–306,1990.

[5] C. Beeri and T. Milo. Functional and predicate programming in OODBs. In *PODS'92*, pages 176–190.

[6] M. Benedikt, G. Dong, L. Libkin and L. Wong. Relational expressive power of constraint query languages. In *PODS'96*, pages 5–16.

[7] M. Benedikt and L. Libkin. On the structure of queries in constraint query languages. *LICS'96*, pages 25–34.

[8] M. Benedikt and L. Libkin. Languages for relational databases over interpreted structures. Technical Memo, Bell Labs, 1996.

[9] M. Ben-Or, D. Kozen, J. Reif. The complexity of elementary algebra and geometry. *JCSS* 32:251–264, 1986.

[10] L. Cabibbo and J. Van den Bussche. On the expressive power of many-sorted logic. Manuscript, 1996.

[11] A. Chandra and D. Harel. Computable queries for relational databases. *JCSS* 21(2):156–178, 1980.

[12] A. Chandra and D. Harel. Structure and complexity of relational queries. *JCSS* 25 (1982), 99–128.

[13] C.C. Chang and H.J. Keisler. *Model Theory.* North Holland, 1990.

[14] A. Dawar, S. Lindell, S. Weinstein. First order logic, fixed point logic, and linear order. In *Computer Science Logic '95*, LNCS vol. 1092, 1996, pages 161–177.

[15] H.-D. Ebbinghaus and J. Flum. *Finite Model Theory.* Springer Verlag, 1995.

[16] M. Escobar-Molano, R. Hull and D. Jacobs. Safety and translation of calculus queries with scalar functions. In *PODS'93*, pages 253–264.

[17] K. Etessami. Counting quantifiers, successor relations, and logarithmic space. *JCSS*, to appear.

[18] E. Grädel and Y. Gurevich. Metafinite model theory. In *Proc. LCC*, LNCS vol. 960, 1994, pages 313–366.

[19] R.L. Graham, B.L. Rothschild and J.H. Spencer. *Ramsey Theory.* John Wiley & Sons, 1990.

[20] S. Grumbach and J. Su. First-order definability over constraint databases. In *PCCP'95.*

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

[22] R. Hull, J. Su. On the expressive power of databases with intermediate types. *JCSS* 43 (1991), 219–267.

[23] R. Hull and J. Su. Domain independence and the relational calculus. *Acta Informatica* 31:513–524, 1994.

[24] P. Kanellakis, G. Kuper, and P. Revesz. Constraint query languages. *JCSS* 51 (1995), 26–52.

[25] M. Kifer, R. Ramakrishnan and A. Silberschatz. An axiomatic approach to deciding query safety in deductive databases. In *PODS'88*, pages 52–60.

[26] Ph. Kolaitis and M. Vardi. 0-1 laws and decision problems for fragments of second-order logic. *Information and Computation*, 87 (1990), 302–338.

[27] Ph. Kolaitis, M. Vardi. Infinitary logic and 0-1 laws. *Information and Computation*, 98 (1992), 258–294.

[28] A. Levy, I. Mumick, Y. Sagiv, O. Shmueli. Equivalence, query reachability and satisfiability in datalog extensions. In *PODS'93*, pages 109–122.

[29] L. Libkin and L. Wong. Query languages for bags and aggregate functions. *JCSS*, to appear. Extended abstract in *PODS'94.*

[30] Y. Moschovakis. *Elementary Induction on Abstract Structures.* North Holland, 1974.

[31] M. Otto and J. Van den Bussche. First-order queries on databases embedded in an infinite structure. *Information Processing Letters*, to appear.

[32] J. Paredaens, J. Van den Bussche, and D. Van Gucht. First-order queries on finite structures over the reals. In *LICS'95*, pages 79–87.

[33] A. Pillay, C. Steinhorn. Definable sets in ordered structures. III. *Trans. of the AMS* 309 (1988), 469–476.

[34] A.P. Stolboushkin and M.A. Taitslin. Linear vs. order constraint queries over rational databases. In *PODS'96*, pages 17–27.

[35] A. Tarski. *A Decision Method for Elementary Algebra and Geometry.* 2nd ed., Univ. California Press, 1951.

[36] J. D. Ullman. *Principles of Database and Knowledgebase Systems*, Volume I, Computer Science Press, 1989.

[37] M. Vardi. On the complexity of bounded-variable queries. In *PODS'95*, pages 266–276.