Relational Expressive Power of Constraint Query Languages

Michael Benedikt, Guozhu Dong, Leonid Libkin, and
Limsoon Wong

Relational Expressive Power of Constraint Query Languages*

Michael Benedikt† AT&T Bell Labs

Guozhu Dong‡ U. of Melbourne

Leonid Libkin§ AT&T Bell Labs

Limsoon Wong¶ Inst. of Systems Science

Abstract

We study the expressive power of first order query languages with several classes of equality and inequality constraints and give three main results. (1) We settle the conjecture that recursive queries such as parity test and transitive closure cannot be expressed in relational calculus with polynomial inequality constraints over the reals. (2) Noting that relational queries exhibit several forms of genericity, we establish a number of collapse results of the following form: The class of generic queries expressible in relational calculus augmented with a class of constraints coincides with the class of queries expressible in relational calculus (with or without an order relation). We prove such results for both natural and active-domain semantics. (3) As a consequence, the relational calculus augmented with polynomial inequalities expresses the same classes of generic relational queries under the natural and active-domain semantics.

In the course of proving these results for the active-domain semantics, we establish Ramsey-type theorems saying that any query involving constraints coincides with a constraint-free query on databases whose elements come from a certain infinite subset of the domain. To prove the collapse results for the natural semantics, we make use of techniques from nonstandard analysis and from the model theory of analytic functions.

1 Introduction

In new database applications involving spatial data (as in geographical databases) and temporal data, it is necessary to store in databases infinite collections of items and to evaluate queries on such infinite collections. The *constraint database model*, introduced by Kanellakis, Kuper and Revesz in their seminal paper [18], is designed to meet the requirements of such applications and is a powerful generalization of Codd's relational model. In this new paradigm, instead of tuples, queries act on "generalized tuples" expressed as quantifier-free first-order constraints. A generalized relation is a

*Part of this work was done when Limsoon Wong was visiting University of Melbourne and AT&T Bell Labs in Murray Hill.

†AT&T Bell Laboratories, 1000 E Warrenville Rd, P.O. Box 3013, Naperville, IL 60566-7013, USA, Email: benedikt@research.att.com.

‡Department of Computer Science, University of Melbourne, Parkville, Vic. 3052, Australia, Email: dong@cs.mu.oz.au.

§AT&T Bell Laboratories, 600 Mountain Avenue, Murray Hill, NJ 07974, USA, Email: libkin@research.att.com.

¶Real World Computing Partnership Novel Function Institute of Systems Science Laboratory, Heng Mui Keng Terrace, Singapore 0511, E-mail: limsoon@iss.nus.sg.

finite set of such generalized tuples. Interesting constraint query languages are then obtained by coupling the relational calculus with arithmetic constraints. Well-known example of queries that are inexpressible in the pure relational calculus but are expressible with this extension include testing whether a set of points forms a convex hull.

So the coupling of relational calculus with arithmetic constraints enhances power. A natural question, which attracted much attention recently, arises: How much more power can we gain from this coupling? In particular, several recent papers [17, 18, 16, 26, 11, 23] made and attempted to settle the following conjecture.

Conjecture. *Queries such as transitive closure, connectivity test, parity test are not definable in the relational calculus plus polynomial inequality constraints over the reals.*

These three queries are singled out because they involve two basic primitives, recursion and counting, and because it was known that they cannot be expressed by the relational calculus [4]. In [10] it was shown that useful properties for proving the inexpressibility of these queries in the relational calculus, such as locality [8] and 0/1-law [7], do not carry over to the constraint query languages. Nevertheless, with a considerable amount of effort, a number of inexpressibility results were established recently. In [12] it is shown, via an AC^0 data complexity result, that the parity query cannot be expressed if only linear constraints are added to the relational calculus. In [2] it is shown that testing whether a constraint database is contained in a line is not definable with linear constraints. Testing whether a constraint database represents a line is not definable in first-order with the order relation [3].

Parity and connectivity tests are examples of *generic* queries. That is, their answer does not change when a bijective map on the domain is applied to a database. It is therefore natural to pose the more general question below. The answer to this question can be applied to queries whose inputs are generalized relations using order constraints built from a fixed set of variables, since such generalized relations can be translated to relations [10].

Question. *Do constraints add pure relational expressive power? More specifically, when limited to relational inputs and outputs, do the extended query languages express more generic queries than the relational calculus?*

We answer this question under both the active and natural semantics. Under the active semantics, quantification variables are assumed to range over the active domain of the database. Under the natural semantics, quantification variables are assumed to range over the whole universe of reals numbers. (1) We show that the addition of constraints to the relational calculus does not add more power beyond ordering when interpreted under the active domain semantics. We establish these results by proving, mostly constructively, several Ramsey-style theorems. (2) We show similar results for the natural semantics. We establish these results using techniques from nonstandard analysis and some results in the model theory of analytic functions. (3) As a consequence, the conjecture mentioned above is affirmed, and the relational calculus plus polynomial inequality constraints is shown to express the same generic queries under the two different semantics.

The coincidence of the two semantics was established for the special cases of the relational calculus by Hull and Su [15] and of the relational calculus with linear constraints by Paredaens, Van den Bussche, and Van Gucht [24]. However, these two results of [15] and [24] are not limited to generic queries. It was also shown in [24] that linear constraints do not add expressive power beyond $<$. Our results generalize the result of [24] to apply to a wide class of signatures. Another constrast to our results is that of Grumbach and Su [11]. They showed that, with an integer test function in the signature, one can define parity of cardinality of finite relations over the reals under the natural interpretation. It is necessary, therefore to put some sort of restriction on the signature. Our result about the natural expansion restricts to signatures that are *o-minimal* [25]: signatures that can define only a certain kind of subset of the real line. We are able to use a number of results that establish o-minimality, and this turns out to be sufficient to settle the main conjecture.

In addition to the constraint model, other extensions of Codd's relational model have received much attention in the past few years. Among them are extensions to complex objects and bags. It is interesting to compare the addition of constraints with those extensions. Several conservativity results proved in [33, 20, 21] show that the addition of nested relations (as in complex objects) does not add relational expressive power. On the other hand, the addition of bags does add relational expressive power [9, 19, 22]. In particular, parity test is definable using bags on ordered domains [19].

Organization. Section 2 presents the notations that are used throughout the paper. We also explain the active and natural semantics of relational calculus and state two previous results relating them. Section 3 describes three notions of genericity of queries. We also briefly investigate their relationship.

Section 4 studies the expressibility of various classes of generic queries using the relational calculus under the active semantics as the ambient language. We prove powerful collapse theorems saying that the expressibility of these queries is *independent* of the presence of arithmetic and other operators in the ambient language. These results are proved via several Ramsey-like results. Moreover, for the special cases of real, rational, and integer arithmetic, our proofs are constructive.

Section 5 considers the expressibility of generic queries, but using the relational calculus under the natural semantics as the ambient language. We prove an analogous collapse theorem saying that the expressibility of these queries is again independent of the presence of a large class of arithmetic and other operators in this more powerful ambient language. This result is proved via an excursion through nonstandard models. The conjecture mentioned earlier follows as a corollary.

Section 6 concludes the paper by relating the 'active' and the 'natural' results of this paper. It is known that the two semantics of relational calculus coincide if $<$ is the only extra primitive. This fact and our collapse theorems extend the coincidence of the active and natural semantics, with respect to generic queries, to a larger number of additional primitives.

2 Notations

Fix a database schema $SC = \langle R_1, \ldots, R_k \rangle$ where R_i is of arity τ_i. Assume that the domain is an infinite set $\mathbb{U}$. A **database** (or database instance) $\mathbf{DB}$ is given by an interpretation of each relational symbol R_i as a finite τ_i-ary relation over $\mathbb{U}$. We denote the set of all database instances of SC with the domain $\mathbb{U}$ by $\mathit{Inst}(\mathbb{U}, SC)$. Given a database $\mathbf{DB}$, its **active domain** $\mathit{adom}(\mathbf{DB})$ is the set of all elements in $\mathbb{U}$ that appear in the database.

A **relational boolean query** (rbq) is a first-order sentence built up from atomic formulae ψ in the language containing $\langle R_1, \ldots, R_k \rangle$ and equality via the usual logical connectives and quantifiers of the form $Q_i x_i$, where Q_i is either $\forall$ or $\exists$. If we allow our atomic formulae ψ to also mention a symbol $<$, which is interpreted as a linear order on $\mathbb{U}$ then we speak of a **$\leq$-rbq**. For each of the semantics we will give for these sentences, it will be the case that an arbitrary rbq can be effectively transformed into a semantically equivalent one of the form

$$Q_1 x_1 \ldots . Q_n x_n . \Psi(x_1, \ldots, x_n) \tag{1}$$

where each Q_i is either $\forall$ or $\exists$ and $\Psi(\cdot)$ is a quantifier-free formula with free variables among $x_1, \ldots, x_n$.

Let Ω be an interpreted signature on $\mathbb{U}$; that is, a family of operations $\omega : \mathbb{U}^{arity(\omega)} \to \mathbb{U}$. For example, if $\mathbb{U} = \mathbb{R}$, then Ω may be $\langle +, -, *, \div, e^x \rangle$. A **constraint boolean query** (cbq) is a sentence built up from atomic formulae ψ by connectives and quantifiers such that ψ is in the language that consists of predicate symbols $R_1, \ldots, R_k$, equality and symbols for operations in Ω. For example, if $\mathbb{U} = \mathbb{R}$, then a cbq may ask if for two reals in a database, their sum is also in the database. Again, each cbq can be transformed into an equivalent one in the prenex normal form (1).

There are two possible interpretations of each of these classes of sentences. Under the **active-domain** semantics (or just active semantics), all quantified variables range over the active domain of a database. That is, a sentence given by (1) defines, under active semantics, the query Q such that the value of Q on a database $\mathbf{DB}$ is the value of

$$Q_1 x_1 \in adom(\mathbf{DB}) \ldots . Q_n x_n \in adom(\mathbf{DB}) . \Psi(x_1, \ldots, x_n)$$

Under the **natural** semantics, all quantified variable range over $\mathbb{U}$. That is, the sentence defines the query Q whose value on $\mathbf{DB}$ equals to

$$Q_1 x_1 \in \mathbb{U} \ldots . Q_n x_n \in \mathbb{U} . \Psi(x_1, \ldots, x_n)$$

Convention: When the semantics (active or natural) is clear, we will often identify the the sentence Q that defines an rbq or a cbq with the semantic object Q, that is, the map from $Inst(\mathbb{U}, SC)$ to $\{\mathbf{T}, \mathbf{F}\}$.

The value of Q on $\mathbf{DB}$ will be denoted by $Q(\mathbf{DB})$. If this value is true, we also write $\mathbf{DB} \models Q$. It will always be clear from the context which interpretation is used for evaluating Q.

The classes of first-order Boolean queries (that is, maps from $Inst(\mathbb{U}, SC)$ to $\{\mathbf{T}, \mathbf{F}\}$) definable under the active and natural semantics will be denoted by $\mathcal{AFO}(\cdot)$ and $\mathcal{NFO}(\cdot)$ respectively, where we list the domain $\mathbb{U}$ (if it is not understood) and the operations from Ω in parentheses. For example, $\mathcal{AFO}$ is just the relational calculus, $\mathcal{AFO}(<)$ is the relational calculus with the total order relation, $\mathcal{AFO}(\mathbb{R}, +, -, *, 0, 1, <)$ is the class of Boolean queries with polynomial inequality constraints definable under active semantics, and $\mathcal{NFO}(\mathbb{R}, +, *, 0, 1)$ is the same class of queries interpreted under natural semantics. Note that in this case we do not have to list the order relation as it is definable under natural semantics: $x < y \Leftrightarrow \exists z . (z \neq 0) \wedge (y - x = z * z)$, since z ranges over the reals.

There are a number of results showing that the natural semantics does not add expressiveness over the active. In particular, we shall use the following two.

Fact 1 (See [15]) *$\mathcal{AFO} = \mathcal{NFO}$.* $\square$

Fact 2 (See [24]) $\mathcal{AFO}(\mathbb{R}, +, -, 0, 1, <) = \mathcal{NFO}(\mathbb{R}, +, -, 0, 1, <)$. $\square$

Definition 1 Two queries Q_1 and Q_2 are **a-equivalent** iff for any database $\mathbf{DB}$, under the active-domain interpretation we have $Q_1(\mathbf{DB}) = Q_2(\mathbf{DB})$. They are **n-equivalent** iff for any database $\mathbf{DB}$, under the natural interpretation we have $Q_1(\mathbf{DB}) = Q_2(\mathbf{DB})$. When it is clear from the context how Q_1 and Q_2 are interpreted, we speak of **equivalent** queries. $\square$

Facts 1 and 2 imply that when the signature contains only $=$, or linear constraints on the reals, two queries are a-equivalent iff they are n-equivalent.

3 Notions of Genericity

Given a constraint language, one may ask queries that are specific to that language. For example, one may ask if a database contains a root of a given polynomial. However, purely relational queries must conform to the data independence principle which says that the internal structure of data has no effect on the answers to queries. This is usually captured by a notion of genericity. The most standard notion of genericity states that queries are invariant under arbitrary permutations of the domain [1]. For constraint databases, more complex notions of genericity were considered [23]. Here we use three notions of genericity. If $\varphi : \mathbb{U} \to \mathbb{U}$ is an endomap on the domain, then φ can be extended to databases over $\mathbb{U}$ in an obvious way. We use $\varphi\mathbf{DB}$ to denote this extension.

Definition 2

- A query Q is **totally generic** (TG) with respect to a domain $\mathbb{U}$ if for any database $\mathbf{DB}$ and any permutation π of $\mathbb{U}$, it is the case that $Q(\pi\mathbf{DB}) = Q(\mathbf{DB})$.
- A query Q is **monotone generic** (MG) with respect to an ordered domain $\mathbb{U}$ if for any database $\mathbf{DB}$ and any monotone injective map $\varphi : \mathbb{U} \to \mathbb{U}$, it is the case that $Q(\varphi\mathbf{DB}) = Q(\mathbf{DB})$.
- A query Q is **locally generic** (LG) with respect to an ordered domain $\mathbb{U}$ if for any database $\mathbf{DB}$ and any partial monotone function $\varphi : \mathbb{U} \mapsto \mathbb{U}$ defined on $adom(\mathbf{DB})$, it is the case that $Q(\varphi\mathbf{DB}) = Q(\mathbf{DB})$.

For any language $\mathcal{L}$, we let $\mathcal{L}^{\mathbf{TG}}$, $\mathcal{L}^{\mathbf{LG}}$, and $\mathcal{L}^{\mathbf{MG}}$ stand respectively for the class of TG-, LG-, and MG-queries expressible in $\mathcal{L}$. $\square$

For example, $\mathcal{AFO}^{\mathbf{TG}}(<)$ is the class of TG-queries definable in relational calculus with order, and $\mathcal{NFO}^{\mathbf{LG}}(\mathbb{R}, +, *, 0, 1)$ the class of LG-queries definable under the natural interpretation with polynomial constraints over the reals. Since queries definable in $\mathcal{AFO}$ and $\mathcal{AFO}(<)$ are TG and LG (MG) respectively, we have the equations $\mathcal{AFO}^{\mathbf{TG}} = \mathcal{AFO}$ and $\mathcal{AFO}^{\mathbf{LG}}(<) = \mathcal{AFO}^{\mathbf{MG}}(<) = \mathcal{AFO}(<)$.

When $\mathbb{U}$ is ordered, TG is the strongest notion because it implies both LG and MG. Also, LG implies MG. Under certain mild restrictions on the order, we also have MG implies LG. We do not need the local notion of genericity for unordered sets because it is equivalent to TG for any infinite $\mathbb{U}$.

We call a linearly ordered set $\langle S, < \rangle$ *doubly transitive* [27] if for every $a < b$ and $x < y$ there exists an automorphism $f : S \to S$ such that $f(a) = x$ and $f(b) = y$.

Proposition 1 *With respect to any infinite universe $\mathbb{U}$, it is the case that every TG-query is also a LG-query and every LG-query is also a MG-query. With respect to any infinite ordered universe $\mathbb{U}$ that is doubly transitive, it is the case that every MG-query is also a LG-query.*

Proof. To prove that TG implies LG, consider a database with $D = adom(\mathbf{DB})$. Let $\pi_D : \mathbb{U} \to \mathbb{U}$ be a partial monotone injective map defined on D. Since both $\mathbb{U} - D$ and $\mathbb{U} - \pi_D(D)$ have the same cardinality, there exists a permutation π on $\mathbb{U}$ that extends π_D. Using TG for Q we obtain $Q(\pi_D\mathbf{DB}) = Q(\pi\mathbf{DB}) = Q(\mathbf{DB})$, which implies that Q is LG. If Q is LG, consider any monotone injection $\varphi : \mathbb{U} \to \mathbb{U}$. Given a database with $D = adom(\mathbf{DB})$. Let φ_D be the restriction of φ on D. Then $Q(\varphi\mathbf{DB}) = Q(\varphi_D\mathbf{DB}) = Q(\mathbf{DB})$. Hence Q is MG. The proof that MG implies LG for doubly transitive orders is similar to that of TG implies LG, because double transitivity provides the needed extension of a partial map, see [27]. $\square$

To see that most of the examples of ordered domains used in the theory of constraint databases are doubly transitive, we state the following lemma.

Lemma 1 ([27]) *$\langle \mathbb{Q}, < \rangle$ and $\langle \mathbb{R}, < \rangle$ are doubly transitive. Also, any countable dense linear ordering without endpoints is doubly transitive.*

Thus, when we prove the collapse results for classes of generic queries over ordered databases, we shall aim to prove the result for LG-queries. Then they will automatically imply the corresponding results for TG-queries and, if $\mathbb{U}$ is doubly transitive, for MG-queries as well.

4 Relational Expressive Power: Active Semantics

In this section we prove a number of collapse results for the active-domain semantics. Thus throughout this section we assume that queries are interpreted under the active-domain semantics. We start by stating the results. Then we prove the main technical results of the section (which we call Ramsey theorems for constraint databases). These results establish the existence of an infinite subset of the domain on which a given cbq is equivalent to some query that does not use constraints. Finally, we use those theorems to conclude the proofs of the collapse theorems.

First, we show a very general collapse result for ordered databases.

Theorem 1 *Let Ω be an arbitrary interpreted signature on the linearly ordered domain $\langle \mathbb{U}, < \rangle$. Then for every LG cbq there exists an equivalent $\leq$-rbq. In other words,*

$$\mathcal{AFO}^{\mathbf{LG}}(\Omega, <) \;=\; \mathcal{AFO}(<).$$

This result has a number of corollaries. One of them was also independently proved by Van den Bussche and Otto [29], who used the Ehrenfeucht-Mostowski theorem about the existence of indiscernibles [5].

Another corollary of this theorem is that the class of MG-queries in $\mathcal{AFO}(\Omega, <)$ is exactly the class of $\mathcal{AFO}(<)$ queries when the domain is doubly transitive. While this covers domains such as $\mathbb{Q}$ and

$\mathbb{R}$, it excludes $\mathbb{Z}$ which is not doubly transitive. To cover the case of $\mathbb{Z}$, we prove a more powerful corollary. We call a linearly-ordered set $\langle S, < \rangle$ without endpoints *weakly homogeneous* if S is infinite and for every finite $X \subset S$ and infinite $Y \subseteq S$ there exists a monotone injection $f : S \to S$ such that $f(X) \subset Y$. It is not hard to see that any doubly transitive order is weakly homogeneous and, in addition, so is any linear order without endpoints in which all intervals are finite (in particular, $\langle \mathbb{Z}, < \rangle$).

Corollary 1

- *Let $\langle \mathbb{U}, < \rangle$ be weakly homogeneous and let Ω be an arbitrary signature. Then $\mathcal{AFO}^{\mathbf{MG}}(\Omega, <) = \mathcal{AFO}(<)$.*
- (See [29]) *Let $\langle \mathbb{U}, < \rangle$ be an ordered domain and let Ω be an arbitrary signature. Then $\mathcal{AFO}^{\mathbf{TG}}(\Omega, <) = \mathcal{AFO}^{\mathbf{TG}}(<)$.*
- *Transitive closure, parity test, and connectivity test are* not *first-order definable over ordered databases under the active-domain interpretation, even in the presence of an arbitrary interpreted signature on the domain.* □

There is no analog of the theorem for the unordered case, because one of the interpreted operations can define a linear order, and it is known that $\mathcal{AFO}(<) \neq \mathcal{AFO}$; see [1, page 462]. However, we can still prove the collapse result for unordered databases for a large class of signatures.

Let the domain be $\mathbb{R}$, and let $\Omega = (h_1, h_2, \ldots)$ be an at most countable signature that consists of functions $\mathbb{R}^{arity(h_i)} \to \mathbb{R}$. We call Ω **sparse** if the subtraction is in Ω and for every Ω-term $h(x_1, \ldots, x_n)$ and any constants $c_1, \ldots, c_{i-1}, c_{i+1}, \ldots, c_n \in \mathbb{R}$, the function $h_i(x) = h(c_1, \ldots, c_{i-1}, x, c_{i+1}, \ldots, c_n)$ is either identical to zero or has at most countably many zeros.

There is a simple way to obtain a large number of sparse signatures over the reals, using the fact that any analytic function has at most countably many zeros [28].

Proposition 2 *Let $(H_1, H_2, \ldots)$ be any collection of analytic functions such that the value of each H_i is in $\mathbb{R}$ if all its arguments are in $\mathbb{R}$. Let h_i be the restriction of H_i to the real arguments. If one of the H_is is subtraction, then $\Omega = (h_1, h_2, \ldots)$ is sparse.* □

For example, $(+, -, *, e^x, 0)$, $(+, -, 0, 1)$, and $(+, -, *, 0, 1)$ are sparse signatures.

Theorem 2 *Let Ω be a sparse signature. Then for every TG cbq in $\mathcal{AFO}(\mathbb{R}, \Omega)$ there exists an a-equivalent rbq. That is,*

$$\mathcal{AFO}^{\mathbf{TG}}(\mathbb{R}, \Omega) \ = \ \mathcal{AFO}.$$

Note that the statements of Theorems 1 and 2 are of a very different nature than complexity based results such as [12].

Theorem 2 does not easily generalize for $\mathbb{Q}$ or $\mathbb{Z}$ because the fact that $\mathbb{R}$ is uncountable is crucial for the proof. However, if we are only interested in the ring structure, there is a nice generalization. Recall that an **integral domain** $\mathbb{K}$ is a commutative ring without divisors of zero.

Theorem 3 *Let $\mathbb{K}$ be an infinite integral domain. Then every TG-query in $\mathcal{AFO}(\mathbb{K}, +, -, *, 0, 1)$ is a-equivalent to some rbq. That is,*

$$\mathcal{AFO}^{\mathbf{TG}}(\mathbb{K}, +, -, *, 0, 1) \;=\; \mathcal{AFO}$$

4.1 Ramsey Theorems for Constraint Databases

The following propositions are the basis for the proofs of the theorems in this section. They state that any cbq coincides with some rbq on databases whose active domain is a subset of a certain infinite subset of $\mathbb{U}$.

Given $U \subseteq \mathbb{U}$, we call two queries Q and Q' **U-equivalent** if, for any database $\mathbf{DB}$,

$$adom(\mathbf{DB}) \subseteq U \text{ implies } Q(\mathbf{DB}) = Q'(\mathbf{DB}).$$

Recall that we assume the active-domain interpretation of queries. We start with the ordered case.

Proposition 3 *Let Ω be an arbitrary interpreted signature on the linearly ordered domain $\langle \mathbb{U}, < \rangle$. Then for any cbq Q there exists an infinite subset $\mathbb{U}_Q \subseteq \mathbb{U}$ and a $\leq$-rbq Q' which is $\mathbb{U}_Q$-equivalent to Q.*

Proof. First we make an observation that will be useful later. Assume that $\Phi_i(x_1, \ldots, x_n)$, $i = 1, 2$, are formulae in n variables, and let $\Phi_1(\vec{d}) = \Phi_2(\vec{d})$ for any $\vec{d} \in D^n$ for some set D. Then for any quantifier prefix we have $Q_1 x_1 \in D \ldots Q_n x_n \in D.\Phi_1(\vec{x}) = Q_1 x_1 \in D \ldots Q_n x_n \in D.\Phi_2(\vec{x})$.

We can transform an arbitrary first-order sentence into prenex normal form. Thus, from now on we assume that Q is given by (1). Without loss of generality, we can assume that Ψ is a formula in disjunctive normal form whose literals are of form:

a. $R_i(z_1, \ldots, z_{\tau_i})$, where all z_js are variables.

b. $t_1(\vec{y})\theta t_2(\vec{z})$, where t_1 and t_2 are terms in the signature Ω, $\theta \in \{=, <\}$, and $\vec{y}$, $\vec{z}$ contain variables from $x_1, \ldots, x_n$.

Any formula Q can be rewritten to the form described above. Indeed, if the arguments of the predicate R_i are terms rather than variables, then we replace each occurrence of $R_i(t_1(y_1^1, \ldots, y_1^{l_{i1}}), \ldots, t_{\tau_i}(y_{\tau_i}^1, \ldots, y_{\tau_i}^{l_{i\tau_i}}))$ by $\exists z_1 \ldots \exists z_{\tau_i}.(z_1 = t_1(\vec{y}_1)) \wedge \ldots \wedge (z_{\tau_i} = t_{\tau_i}(\vec{y}_{\tau_i})) \wedge R_i(z_1, \ldots, z_{\tau_i})$. It is easy to see that such a replacement does not change the value of the formula for every assignment of values in the domain to the variables. Then we can convert the resulting formula into the desired prenex normal form, whose quantifier-free part is in disjunctive normal form.

We may also assume that no literal of the form b is negated. This follows because the domain is linearly ordered. That is, $\neg(t_1(\vec{y}) = t_2(\vec{z}))$ can be replaced by $(t_1(\vec{y}) < t_2(\vec{z})) \vee (t_2(\vec{z}) < t_1(\vec{y}))$, and $\neg(t_1(\vec{y}) < t_2(\vec{z}))$ can be replaced by $(t_1(\vec{y}) = t_2(\vec{z})) \vee (t_2(\vec{z}) < t_1(\vec{y}))$.

Now subdivide the disjuncts into two classes. The first one consists of those which are conjunctions of (possibly negated) literals of the form a. The second class consists of those which are conjunctions of literals, at least one of them being of the form b. Thus, $\Psi = \Psi_1 \vee \Psi_2$, where Ψ_1 is a formula in the signature $\langle R_1, \ldots, R_k, =, \leq \rangle$.

Let p_n be the number of ordered partitions of an n-element set. (An ordered partition of X is a partition X_1, ..., X_s with a linear order $X_{i_1} < \ldots < X_{i_s}$ on its elements.) Let ζ_i be the formula that specifies that values of x_1, ..., x_n form the ith ordered partition. For example, if $n = 3$, for the partition $\{\{1,2\},\{3\}\}$ with $\{1,2\} < \{3\}$, the corresponding ζ is $(x_1 = x_2) \wedge (x_1 < x_3)$. Note that $\bigvee_{i=1}^{\mathsf{p}_n} \zeta_i = \mathbf{T}$. Hence, Ψ_2 can be replaced by $\bigvee_{i=1}^{\mathsf{p}_n} (\zeta_i \wedge \Psi_2)$.

Let $c_1, \ldots, c_m$ denote all the constraints of the form b present in Ψ_2. It can be readily seen that Q is equivalent to

$$Q_1 x_1 . \ldots Q_n x_n . \Psi_1 \vee \bigvee_{i=1}^{\mathsf{p}_n} \Psi_2[(c_j \wedge \zeta_i)/c_j] \tag{2}$$

Here $[(c_j \wedge \zeta_i)/c_j]$ means that every c_j is replaced by $c_j \wedge \zeta_i$. The equivalence follows from the fact that in Ψ_2 in every disjunct at least one literal is a condition, and no condition literal is negated. Now we need the following lemma.

Lemma 2 *Suppose that a query Q is given by $(2')$, where*

$$Q_1 x_1 . \ldots Q_n x_n . \Psi_1 \vee \bigvee_{i \in I} \Psi_2[(c_j \wedge \zeta_i)/c_j] \tag{$2'$}$$

I is a nonempty subset of $\{1, \ldots, \mathsf{p}_n\}$ and Ψ_1 does not mention any symbol from Ω. Then, for any infinite $D \subseteq \mathbb{U}$, there exists an infinite $D' \subseteq D$, a formula Ψ_1' that does not mention any symbol in Ω, and an $i_0 \in I$ such that for Q' defined by

$$Q_1 x_1 . \ldots Q_n x_n . \Psi_1' \vee \bigvee_{i \in I - \{i_0\}} \Psi_2[(c_j \wedge \zeta_i)/c_j] \tag{$2''$}$$

it is the case that Q and Q' coincide for any database $\mathbf{DB}$ with $adom(\mathbf{DB}) \subseteq D'$.

Before we prove this lemma, let us observe that the proposition follows straightforwardly from it. We just apply the lemma inductively to (2) until all formulae that mention symbols in Ω disappear.

To prove the lemma, pick an element $i_0 \in I$ arbitrarily. Let ζ_{i_0} specify an ordered partition with s classes $X_1, \ldots, X_s \subseteq \{x_1, \ldots, x_n\}$ ordered by $X_1 < \ldots < X_s$. Let $y_i \in X_i$; then we may rewrite $\Psi_2[(c_j \wedge \zeta_{i_0})/c_j]$ to $\Psi_2[c_j'/c_j] \wedge \gamma$ where c_j' is a condition of form $t_1 \theta t_2$ that only uses $\{y_1, \ldots, y_s\}$ and γ states that all y_is are different and that $y_1 < \ldots < y_s$.

Since there are m conditions c_j', there are 2^m possible truth values for the system $c_1', \ldots, c_m'$. Let A_l be the set of s-tuples $(d_1, \ldots, d_s)$ of distinct elements of D ordered by $d_1 < \ldots < d_s$ such that the truth values of $c_i'(d_1, \ldots, d_s)$ form the binary representation of the number l, $0 \leq l \leq 2^m - 1$. Then $A_0, \ldots, A_{2^m-1}$ form a partition of the set of s-tuples of distinct elements of D.

By Ramsey's theorem, we can find an infinite $D' \subseteq D$ such that all s-tuples of distinct elements of D' are in the same class of the partition, say A_l. Let l correspond to the truth values of c_j's being t_js. Let

$$\Psi^{i_0} \;=\; (\Psi_2[t_1/c_1, \ldots, t_m/c_m] \wedge \gamma) \quad \text{and} \quad \Psi_1' = \Psi_1 \vee \Psi^{i_0}.$$

Then, for any n-tuples of elements of D', the values of

$$\Psi_1 \vee \bigvee_{i \in I \subseteq \{1,\ldots,\mathsf{p}_n\}} \Psi_2[(c_j \wedge \zeta_i)/c_j] \ \text{ and } \ \Psi_1' \vee \bigvee_{i \in I - \{i_0\}} \Psi_2[(c_j \wedge \zeta_i)/c_j]$$

coincide. Indeed, if $(d_1, \ldots, d_n)$ does not satisfy ζ_{i_0}, this is immediate (because the disjuncts that make the difference between the formulae evaluate to $\mathbf{F}$), and if it does satisfy ζ_{i_0}, it follows from the fact that Ψ^{i_0} and $\Psi_2[(c_j \wedge \zeta_{i_0})/c_j]$ coincide on such tuples.

Applying the observation made at the beginning of the proof, we see that for Q' defined by $(2'')$ where $\Psi_1' = \Psi_1 \vee \Psi^{i_0}$, it is the case that Q coincides with Q' on all databases with active domain in D'. Since Ψ^{i_0} does not mention any symbols in Ω, this proves the lemma and the proposition. $\Box$

Next, we prove a similar result for sparse signatures.

Proposition 4 *Let Ω be a sparse signature. Then for any cbq Q there exists an infinite subset $\mathbb{U}_Q \subseteq \mathbb{U}$ and a rbq Q' which is $\mathbb{U}_Q$-equivalent to Q.*

Proof. First, assume that Q is given by

$$Qx_1 \ldots Qx_n.\Psi(x_1, \ldots, x_n)$$

where Ψ is in DNF, and its literals are of form $R_i(z_1, \ldots, z_{T_i})$ where all z_js are variables, or they are condition literals $h_1(\vec{x})\theta h_2(\vec{x})$ where h_1 and h_2 are Ω-definable functions and $\theta \in \{=, \neq\}$; see the proof of Proposition 3. Further assume without loss of generality that no condition literal is negated.

Let B_n be the nth Bell number; that is, B_n is the number of partitions of an n-element set. Let ξ_i represent the ith partition on an n-element set of variables, $i = 1, \ldots, B_n$. For example, for the partition $\{\{1,2\},\{3,4\}\}$ of $\{1,2,3,4\}$ we would get ξ as $(x_1 = x_2) \wedge (x_3 = x_4) \wedge (x_1 \neq x_3)$. Since $\bigvee_{i=1}^{B_n} \xi_i = \mathbf{T}$, we have $\Psi = \bigvee_{i=1}^{B_n}(\xi_i \wedge \Psi)$.

Consider each $\xi_i \wedge \Psi$. Let ξ_i specify the partition $X_1, \ldots, X_s$ and let y_i be a representative of X_i. For each condition $h_1(\vec{x})\theta h_2(\vec{x})$ in Ψ, we replace h_i by h_i' in which only $y_1, \ldots, y_s$ are used. Furthermore, if h_1' is identically h_2', we replace the corresponding condition by $\mathbf{T}$ or $\mathbf{F}$ respectively. Thus, Q is equivalent to a query of form

$$Qx_1 \ldots Qx_n. \bigvee_{i=1}^{B_n} (\xi_i \wedge \Psi_i) \tag{3}$$

where Ψ_i is a formula in s variables (provided ξ_i specifies a partition with s classes) and each condition is of form $h_1'(\vec{x})\theta h_2'(\vec{x})$, where h_1' and h_2' are not identical. Moreover, h_1' and h_2' use variables for which ξ_i implies that their values are distinct. Next, we need the following lemma.

Lemma 3 *Given a system of equations $f_1(\vec{x}) = 0, \ldots, f_k(\vec{x}) = 0$ where f_is are Ω-definable functions in at most n variables $x_1, \ldots, x_n$. If no f_i is identically zero, then there exists an infinite set $R \subseteq \mathbb{R}$ such that assigning distinct values of R to variables $x_1, \ldots, x_n$ simultaneously invalidates all the equations.*

This lemma immediately implies the statement of Proposition 4. Define a system of equations $\mathcal{F}$ as follows: go through all the disjuncts in Q — assumed to be in the form (3) — and for each condition $h'_1(\vec{x})\theta h'_2(\vec{x})$ add $h'_1(\vec{x}) - h'_2(\vec{x}) = 0$ to $\mathcal{F}$. Note that $h'_1(\vec{x}) - h'_2(\vec{x})$ is not identically zero by construction. By the lemma, find an infinite $R \subseteq \mathbb{R}$ such that assigning distinct values from R to variables in equations in $\mathcal{F}$ simultaneously invalidates all of them. Then rewrite Q to Q' by replacing each $h'_1(\vec{x}) = h'_2(\vec{x})$ by $\mathbf{F}$ and each $h'_1(\vec{x}) \neq h'_2(\vec{x})$ by $\mathbf{T}$. Then, for any $\mathbf{DB}$ with $adom(\mathbf{DB}) \subseteq R$ it is the case that $Q(\mathbf{DB}) = Q'(\mathbf{DB})$, and Q' does not mention any symbols from Ω, which concludes the proof.

To prove the lemma, we call a function $\Omega_{\mathbb{R}}$-definable if it is definable in Ω extended with constants for all elements of $\mathbb{R}$. Let $f(x_1, \ldots, x_n)$ be a function in n variables and $1 \leq i \leq n$. We let $f^{i,s}$ be the function in $n-1$ variables defined as $f(x_1, \ldots, x_{i-1}, s, x_{i+1}, \ldots, x_n)$. We need the following

Claim 1 *For any at most countable collection of $\Omega_{\mathbb{R}}$-definable functions $\mathcal{G} = g_1, g_2, \ldots$ that are not identically zero and any at most countable set $S \subset \mathbb{R}$, there exists $s \in \mathbb{R} - S$ such that none of the functions $g^{i,s}$, where $g \in \mathcal{G}$, is identically zero.*

To prove this claim, consider a $\Omega_{\mathbb{R}}$-definable non-zero function $g(x_1, \ldots, x_n)$ and assume without loss of generality that $i = n$. First notice that there exists a $(n-1)$-vector $\vec{c}$ such that $g_{\vec{c}}(x) = g(\vec{c}, x)$ is not identically zero. Indeed, if $g_{\vec{c}}$ is identically zero for any $\vec{c}$, then g is identically zero. Now pick such a $\vec{c}$ and, by applying the fact that Ω is sparse, find a countable set $S_{g,i}$ of roots of $g_{\vec{c}}$. Note that for every $s \notin S_{g,i}$ we obtain that $g^{i,s}$ is not identically zero.

Now consider the countable set

$$S_{\mathcal{G}} \;=\; S \;\cup\; \bigcup_{g \in \mathcal{G}} \bigcup_{i} S_{g,i}$$

It follows from our construction that for any $s \in \mathbb{R} - S_{\mathcal{G}}$ none of the functions $g^{i,s}$ is identically zero. This finishes the proof of the claim.

To conclude the proof of the lemma, we start with our original set $\mathcal{F}_0 = \mathcal{F}$ of functions and $S = \emptyset$, and use the claim to find s_0 such that none of the functions in $\mathcal{F}_1 = \mathcal{F}_0 \cup \{g^{i,s_0} \mid g \in \mathcal{F}_0\}$ is identically zero. Continuing, at the mth step we have a finite set $S = \{s_0, \ldots, s_{m-1}\} \subset \mathbb{R}$ and a set $\mathcal{F}_{m-1}$ of functions none of which is identically zero. We find $s_m \in \mathbb{R} - S$ such that none of the functions in $\mathcal{F}_m = \mathcal{F}_{m-1} \cup \{g^{i,s_m} \mid g \in \mathcal{F}_{m-1}\}$ is identically zero. Now consider the infinite set $R = \{s_0, s_1, s_2, \ldots\}$. It follows immediately from our construction that assigning distinct s_is to distinct variables simultaneously invalidates all equations $f(\vec{x}) = 0$, $f \in \mathcal{F}_0$. This finishes the proof of the lemma and the proposition. □

A similar result can be shown for integral domains.

Proposition 5 *Let $\mathbb{K}$ be an infinite integral domain, i.e. $\Omega = (+, -, *, 0, 1)$. Then for any cbq Q there exists an infinite subset $\mathbb{U}_Q \subseteq \mathbb{K}$ and a rbq Q' which is $\mathbb{U}_Q$-equivalent to Q.*

Proof. We use an argument that is very similar to the proof of Proposition 4. Let $p \in \mathbb{K}[x_1, \ldots, x_n]$

be a polynomial in n variables over $\mathbb{K}$. We can represent it as

$$\text{(P1)} \qquad p = \sum_{i \in I} c_i * x_1^{m_1^i} * \ldots * x_n^{m_n^i}$$

where I is a finite set of indices, $M_i = (m_1^i, \ldots, m_n^i)$ is an n-tuple of natural numbers, $M_i \neq M_j$ for $i \neq j$ and $c_i \in \mathbb{K} - \{0\}$ for all i. (The zero polynomial is represented by $I = \emptyset$.) From now on assume that all polynomials are represented by (P1). Note that all conditions used in Q can be assumed to be of the form $p(\vec{x})\theta 0$, where p is a polynomial over $\mathbb{K}$. Hence, according to the proof of Proposition 4, all that is required to conclude the proof is the following:

Lemma 4 *For any system of equations $p_1(\vec{x}) = 0, \ldots, p_k(\vec{x}) = 0$ where $p_i \in \mathbb{K}[x_1, \ldots, x_n]$ are polynomials that are not identically zero, there exists an infinite set $K \subseteq \mathbb{K}$ such that assigning distinct values of K to variables $x_1, \ldots, x_n$ simultaneously invalidates all the equations.*

We first need this

Claim 2 *If $p \in \mathbb{K}[x_1, \ldots, x_n]$ is written in (P1), then p is identically zero iff either $I = \emptyset$ or $c_i = 0$ for all $i \in I$.*

To prove this claim, first notice that it is enough to consider polynomials in $\mathbb{K}[x]$. Indeed, if $p \in \mathbb{K}[x_1, \ldots, x_n]$ is identically zero, then $p'(x) = p(x, 1, \ldots, 1) \in \mathbb{K}[x]$ is identically zero. To prove the claim for $p \in \mathbb{K}[x]$, we use induction on $card(I)$. If I is empty, we are done. If I is a singleton, this follows from the fact that there are no divisors of zero. Assume $card(I) > 1$ and $p(x) = c_n x^n + \cdots + c_1 x + c_0$. Since $p(0) = 0$, we have $c_0 = 0$. Then by cancellation we have $p'(x) = c_n x^{n-1} + \cdots + c_2 x + c_1$ is identically zero. By the induction hypothesis, we obtain $c_1 = \ldots = c_n = 0$.

Let $p \in \mathbb{K}[x_1, \ldots, x_k]$ be a polynomial in k variables. For any index i and any $s \in \mathbb{K}$ we denote the polynomial in $k-1$ variables, obtained by substituting s for x_i in p, by $p^{i,s}$.

Next, we claim the following.

Claim 3 *For any finite collection of polynomials $p_1, \ldots, p_m \in \mathbb{K}[x_1, \ldots, x_p]$ that are not identically zero, and for any finite set $S \subset \mathbb{K}$, there exists $s \in \mathbb{K} - S$ such that none of the polynomials $p_j^{i,s}$, $j = 1, \ldots, m$, $i = 1, \ldots, k$ is identically zero.*

To prove this claim, assume that all p_js are represented in the form (P1). Then none of the coefficients c_is is zero by Claim 2. Fix a polynomial $p = p_j$ and assume it is given by (P1). Let $M_{i_1}, \ldots, M_{i_l}$ be those Ms whose components, except the ith one, coincide. If $p^{i,s}$ is identically zero, then

$$\text{(4)} \qquad c_{i_1} s^{k_1} + \cdots + c_{i_l} s^{k_l} = 0$$

where k_r is the ith component of M_r. Doing this operation for all p_js and all collections of Ms in the prepresentation (P1) that coincide outside of their ith component, we obtain a finite number of equations that must hold if some of the polynomials $p^{i,s}$ are identically zero. Since all the coefficients in equations (4) are nonzero, they may only have finitely many roots. Thus, if we choose s outside of S such that s does not coincide with any root of the polynomials (4), then none of $p^{i,s}$ is identically zero by Claim 2.

Now we can conclude the proof of Lemma 4 (and thus of Proposition 5) in exactly the same way as we did for Proposition 4. $\square$

Note that we did not give a constructive way of finding the infinite set $\mathbb{U}_Q$. While the mere existence of such a set is sufficient for us to prove the main theorems, we now demonstrate that for the signature of a ring the sets $\mathbb{U}_Q$ have a very simple structure. By D_r we denote the set $\{r^{r^i} \mid i \in \mathbb{N}_+\}$.

Proposition 6 *Let $\mathbb{U}$ be $\mathbb{R}$ or $\mathbb{Q}$ or $\mathbb{Z}$. Then for any cbq Q there exists a number r in $\mathbb{N}$ such that the set $\mathbb{U}_Q$ from the previous propositions can be taken to be D_r. Moreover, r and the corresponding Q' can be effectively constructed from Q.*

Proof. The proofs of all propositions in this section were based on the fact that we can simultaneously invalidate all nontrivial constraints. Since in the given signature all constraints are of the form $p(\vec{x})\{=, \neq, <, \not<\}0$, we have to find an infinite set on which a finite number of such constraints are simultaneously validated or invalidated, provided they are all nontrivial.

Let p be a polynomial in n variables given by (P1). Consider an arbitrary ordering $i_1 \succ i_2 \succ \ldots \succ i_n$ of $\{1, \ldots, n\}$. Order M_js lexicographically with respect to $\succ$, i.e. $M_j \succ M_l$ if $m^j_{i_1} > m^l_{i_1}$ or $m^j_{i_1} = m^l_{i_1}$ and $m^j_{i_2} > m^l_{i_2}$ etc. Let M_k be the maximal one of M_j, $j \in I$, wrt $\succ$. Notice that M_k is uniquely defined. The following is the key lemma.

Lemma 5 *For p and M_k (as constructed above) there exists $r_0 \in \mathbb{R}$ (which can be effectively constructed) such that for every $r \geq r_0$ and for every $\vec{x} = (x_1, \ldots, x_n) \in D_r \times \ldots \times D_r$ such that $x_{i_1} > x_{i_2} > \ldots > x_{i_n}$, it is the case that*

$$sign(p(\vec{x})) \;=\; sign(c_k)$$

It is easy to see that Lemma 5 implies the proposition. For the ordered case, we can constructively rewrite Q to the form (2). This lemma gives us the sign and the corresponding r_0 of each polynomial in the rewritten formula. Hence it allows us to replace each inequality constraint by true or by false and each equality constraint by false. We can then take r to be the greatest of these r_0. Also, at this point, we would have already constructed the desired Q' for the ordered case — the nonconstructive steps described in Lemma 2 are thus avoided.

For the unordered case, we can rewrite Q to the form (3). Then for each polynomial in the rewritten formula and for each possible order of its variables, we determine a r_0 using Lemma 5. Then r can be taken as the greatest of these r_0. Then all equations in the rewritten formula can be replaced by false and all inequations by true — the nonconstructive steps described in Lemma 3 can thus be skipped.

It remains now to prove Lemma 5. To make the notation bearable, assume without loss of generality that $1 \succ 2 \succ \ldots \succ n$. Let $c_k > 0$. We use the notation $\vec{x}^M$ for $\prod_{i=1}^n x_i^{m_i}$ for $M = (m_1, \ldots, m_n)$. Let

$(\star)$ stand for $x_1 > x_2 \ldots > x_n$. Let

$$\Delta \;=\; \max_{i \in I} \max_{j=1,\ldots,n} m^i_j$$

Claim 4 *If $r > n\Delta + 1$, then for any $\vec{x} \in D^n_r$ satisfying $(\star)$ and for any $M_i \neq M_k$, we have*

$$\vec{x}^{M_k} \;>\; \frac{\vec{x}^{M_i}}{r}$$

To prove this claim, assume that $M = M_i$ is under M in the lexicographic ordering. Then, for $M_k = (m_1, \ldots, m_n)$ and $M = (\mu_1, \ldots, \mu_n)$ we can find an i such that $m_j = \mu_j$ for $j < i$ and $m_i > \mu_i$. For now assume $i \neq n$. Let $x_i = r^{r^s}$. We obtain, since $r > 1$,

$$\vec{x}^{M_k} = \prod_{j=1}^{n} x_j^{m_j} \quad \geq (\prod_{j<i} x_j^{m_j}) * (r^{r^s})^{m_i}$$

Furthermore,

$$\begin{aligned}
\vec{x}^{M} \;&=\; (\prod_{j<i} x_j^{\mu_j}) * (r^{r^s})^{\mu_i} * (\prod_{j>i} x_j^{\mu_j}) \\
&\leq\; (\prod_{j<i} x_j^{m_j}) * (r^{r^s})^{m_i - 1} * (\prod_{j>i} (r^{r^{s-1}})^{\Delta}) \\
&\leq\; (\prod_{j<i} x_j^{m_j}) * (r^{r^s})^{m_i - 1} * (r^{r^{s-1}})^{n\Delta}
\end{aligned}$$

since x_js come from D_r and we know $r > 1$ and $x_i > x_j$ for $j > i$. Finally,

$$\frac{\vec{x}^{M_k}}{\vec{x}^{M}} \;\geq\; \frac{r^{r^s}}{(r^{r^{s-1}})^{n\Delta}} \;\geq\; r^{(r-n\Delta)r^{s-1}} \;>\; r^{r^{s-1}} \;\geq\; r$$

since $r - n\Delta > 1$ and $r > 1$. The proof for the case $i = n$ is similar (essentially the last product is not needed). The claim is thus proved.

Define $G \;= \max_i(|\, c_i \,|) * card(I)$ and then it suffices to set

$$r_0 \;=\; \max(\frac{G}{c_k}, n\Delta) + 1$$

To see this, suppose $r \geq r_0$. If $I = \{k\}$, we are done. If not, using Claim 4, we obtain for any vector $\vec{x} \in D^n_r$ satisfying $(\star)$:

$$\begin{aligned}
p(\vec{x}) \;&\geq\; c_k * \vec{x}^{M_k} \;-\; \sum_{i \neq k} |\, c_i \,| * \vec{x}^{M_i} \\
&>\; c_k * \vec{x}^{M_k} \;-\; \sum_{i \neq k} |\, c_i \,| * \frac{\vec{x}^{M_k}}{r} \\
&\geq\; c_k * \vec{x}^{M_k} \;-\; \sum_{i \neq k} \max_l(|\, c_l \,|) * \frac{\vec{x}^{M_k}}{r} \\
&>\; c_k * \vec{x}^{M_k} \;-\; \max_l(|\, c_l \,|) * card(I) * \frac{\vec{x}^{M_k}}{r} \\
&=\; \vec{x}^{M_k}(c_k - \frac{G}{r}) \;>\; 0
\end{aligned}$$

Thus, $p(\vec{x}) > 0$, which proves the case $c_k > 0$. To prove the case $c_k < 0$, just apply the above proof to $-p$. Lemma 5 is proved. $\square$

4.2 Proofs of Main Theorems

We can now present the straightforward proofs of the main results of this section.

Proof of Theorem 1 and Corollary 1 Consider a cbq Q and find a set $\mathbb{U}_Q$ and a $\leq$-rbq Q' as in Proposition 3. Now we claim that Q and Q' are equivalent. Indeed, take any database $\mathbf{DB}$ and let $D = adom(\mathbf{DB})$. Find any partial monotone injective map π defined on D such that $\pi(D) \subset \mathbb{U}_Q$. This is possible because $\mathbb{U}_Q$ is infinite. Since Q is locally generic, $Q(\mathbf{DB}) = Q(\pi\mathbf{DB})$. By Proposition 3, $Q(\pi\mathbf{DB}) = Q'(\pi\mathbf{DB})$ since $adom(\pi\mathbf{DB}) \subseteq \mathbb{U}_Q$. Since Q' is a $\leq$-rbq, it is locally generic, and hence $Q'(\pi\mathbf{DB}) = Q'(\mathbf{DB})$. Thus, $Q(\mathbf{DB}) = Q'(\mathbf{DB})$, proving that Q and Q' are equivalent as desired.

For the proof of Corollary 1, part a, note that if $\mathbb{U}$ is weakly homogeneous, then there is a monotone injective map such that $\pi(D) \subset \mathbb{U}_Q$. Furthermore, $Q(\mathbf{DB}) = Q(\pi\mathbf{DB})$ because Q is MG. Then the proof follows. $\square$

Proof of Theorems 2 and 3 Consider a cbq Q and find an infinite set $\mathbb{U}_Q$ and a rbq Q' as in Proposition 4 or 5. We claim that Q and Q' are equivalent. For a database $\mathbf{DB}$ let $D = adom(\mathbf{DB})$. Find any permutation π of $\mathbb{R}$ such that $\pi(D) \subset \mathbb{U}_Q$. Since Q is generic, $Q(\mathbf{DB}) = Q(\pi\mathbf{DB})$. By Proposition 4 or 5, $Q(\pi\mathbf{DB}) = Q'(\pi\mathbf{DB})$ since $adom(\pi\mathbf{DB}) \subseteq \mathbb{U}_Q$. Since Q' is a rbq, it is totally generic, and thus $Q'(\pi\mathbf{DB}) = Q'(\mathbf{DB})$. Therefore, $Q(\mathbf{DB}) = Q'(\mathbf{DB})$, proving that Q and Q' are equivalent as desired. $\square$

5 Relational Expressive Power: Natural Semantics

In this section we prove the collapse theorem for the natural interpretation of queries. That is, we prove that for certain signatures Ω, any LG-query in $\mathcal{NFO}(\Omega, <)$ can be defined just in $\mathcal{NFO}(<)$.

Throughout this section we assume that the domain $\mathbb{U}$ is linearly ordered by $<$, and that queries are evaluated under the natural interpretation. We say that Ω is **o-minimal** (see [14]) if every subset of $\mathbb{U}$ that is definable with parameters in the model $\langle \mathbb{U}, \Omega \rangle$ is composed of a finite union of (possibly degenerate) intervals. By intervals we mean sets of form $\{x \mid x < a\}$, $\{x \mid x > a\}$, $\{x \mid a < x < b\}$ or closed, or half-open half-closed versions of these.

If $\mathbb{U} = \mathbb{R}$, then examples of o-minimal signatures include:

- $(+, *, <, 0, 1)$ — this follows from Tarski's quantifier elimination theorem.
- $(+, *, e^x, 0)$ — this follows from the work of [32], see also [31].
- $(+, *, e^x, \Gamma(x), 0)$ — this was recently announced [30].

There are many other intersting examples of o-minimal structures, see [31].

For the rest of this section, we restrict our attention to signatures Ω that are o-minimal. Our main result is

Theorem 4 *If Ω is an o-minimal signature on $\mathbb{U}$, then for every LG cbq there exists an n-equivalent $\le$-rbq. That is*

$$\mathcal{NFO}^{\mathbf{LG}}(\Omega,<) \;=\; \mathcal{NFO}^{\mathbf{LG}}(<).$$

We choose to make use of the technique of nonstandard universes. This will be of use in simplifying some of the bookkeeping involved in Ehrenfeucht-Fraisse games. It also allows us to construct a proof that follows this basic intuition: constraint boolean queries over the real field cannot distinguish "large" instances which agree on "all" relational boolean queries.

5.1 Preliminaries

We start with some definitions of nonstandard universes. For more information, the reader should consult [5].

For any set S, the **superstructure** $V(S)$ over S is defined as $V(S) = \bigcup_{n<\omega} V_n(S)$ where $V_1(S) = S$, and $V_{n+1}(S) = V_n(S) \cup \{X \mid X \subset V_n(S)\}$.

We will work with the structure $\langle V(S), \in\rangle$ considered as a structure for the first-order language for the epsilon relation. A **bounded-quantifier formula** in this language is a formula built up from atomic formulas by the logical connectives and the quantification: $\forall X \in Y$, $\exists X \in Y$, where X and Y are variables.

A **nonstandard universe** consists of a pair of superstructures $V(S)$ and $V(Y)$ and a mapping $* : V(S) \to V(Y)$ which is the identity when restricted to S (i.e. $^*x = x$ for each x in S) and which satisfies

1. $Y = {}^*S$.
2. *(Transfer Principle)* For any bounded quantifier formula $\phi(v_1, \ldots, v_n)$ and any list $a_1, \ldots, a_n$ of elements from $V(S)$, $\phi(a_1, \ldots, a_n)$ is true in $V(S)$ if and only if $\phi(^*a_1, \ldots, {}^*a_n)$ is true in $V(Y)$.

An element of $V(Y)$ is **standard** if it is in the image of the $*$-map. An element of $V(Y)$ is **internal** if it is in the downward transitive closure of the standard sets under $\in$. Elements of $V(Y)$ that are not internal are called **external**. An internal map is a map whose graph is an internal set.

We will assume that our universe also satisfies the following

3. *(Countable Saturation Principle)* For every standard A, and every countable collection $\Sigma(x, \vec{v})$ of bounded-quantifier formulas, and for every vector $\vec{c}$ of internal sets, if every finite subset of $\Sigma(x, \vec{c}/\vec{v})$ is satisfied in $V(Y)$ by some element of A, then $\Sigma(x, \vec{c}/\vec{v})$ is satisfied by an element of A.

Finally, we work in a nonstandard universe that satisfies the following axiom:

4. *(Isomorphism Property)* For any finite first-order language L, and any two L-structures M_1 and M_2 that are internal, if M_1 and M_2 are L-elementary equivalent (i.e., agree on all standard sentences of L), then there is an (not necessarily internal) L-isomorphism between M_1 and M_2.

For example, the isomorphism property above guarantees that any two sets whose cardinality is a nonstandard integer have the same external cardinality. For basic facts about the isomorphism property, and a proof that saturated models with the isomorphism property exist, the reader is referred to [13].

We will often omit the $*$ when convenient: for example, if $<$ is an ordering on S, x_1 and x_2 are elements of $^*S$, then we will write $x_1 < x_2$ rather than $x_1 {}^*\!< x_2$. Similarly, if Q is a query on schema SC, and M is a *-database (an element of the set $^*S$, where S is the set of databases for SC), then we will refer to $q(M)$ rather than $^*q(M)$.

For the proof we take S to be the disjoint union of $\mathbb{U}$ and $\mathbb{N}$.

We first state a couple of propositions that will be useful.

Proposition 7 *Let L be a first-order language such that each symbol is represented by an internal set in the nonstandard universe (although the whole set of symbols need not be internal). Let M be an L-structure such that the domain of M is internal and the interpretation of each symbol in L is internal (such a model is called "internally presented" in the literature). Let $\Gamma(y)$ be a countable collection of L-formula. Then, if Γ is finitely satisfiable in M, then Γ is satisfied in M.*

Proof. For each formula $\phi(y)$ in Γ, let M_ϕ be the reduct of M to the (finite, hence internal) language of ϕ, and let $\phi'(y)$ be the formula (in the language of set theory) that says $\langle N_\phi, y\rangle$ satisfies $\phi(y)$. Then each $\phi'(y)$ is an internal formula satisfied in the nonstandard universe, so by countable saturation, there is a y satisfying each $\phi'(y)$. □

Proposition 8 *Let L be a language for which each symbol is internal, and let M be an L-structure such that the domain of M is internal and the interpretation of each symbol in L is internal. Let $\phi(\vec{x})$ be a formula of L that has standard finite cardinality (i.e. number of symbols). Then the internal satisfaction predicate $^*\models$ agrees with the external satifaction predicate $\models$ on ϕ. That is, if $\vec{c}$ is a finite sequence of parameters from M, then $M \,^*\!\models \phi(\vec{c})$ iff $M \models \phi(\vec{c})$.*

The proof is by straightforward induction on logical complexity, using the Tarskian definition of truth and the transfer principle.

Given the above proposition we will not distinguish the two kinds of satisfaction predicates when we are dealing with finitary ϕ's.

5.2 Proof of theorem 4

Note that it suffices to prove the theorem for Ω finite, since any counterexample to collapse would involve a single constraint boolean query, which would involve only finitely many symbols from the language of Ω. So henceforth we will asume Ω to be finite.

Let Q be a counterexample query over some schema $SC = \{R_1, \ldots, R_n\}$. That is, Q is expressible in Ω and is locally generic, but is not expressible only with order.

Proposition 9 *There are two $^*SC$-databases M_1 and M_2 in the nonstandard universe (that is, elements of the image under $*$ of the set of databases) such that M_1 and M_2 agree on every $<$-query but disagree on Q.*

Proof. Note that M_1 and M_2 are internal. Let ϕ_n enumerate the $<$-queries. Let $=_n$ be the equivalence relation on databases given by $D_1 =_n D_2$ iff D_1 and D_2 agree on the first n ϕ_i.

By saturation, it suffices to show that for every standard natural number n, there are two models that agree on ϕ_i for each $i \leq n$ but disagree on Q. If this were not the case, then the models of Q are composed of finitely many $=_n$ equivalence classes. But since each equivalence class is definable by a $<$-cbq, this would make Q definable as a $<$-cbq as well, since it would be the disjunction of the finitely many sentences defining the $=_n$ classes contained in it, contrary to the assumption on Q. $\Box$

Now, let the active domain of M_1 have size H, and the active domain of M_2 have size K. Without loss of generality, we will assume $K \leq H$.

Let $c_1, \ldots, c_H$ be an (internal) sequence of elements of $^*\mathbb{U}$ indiscernible over Ω. Such a sequence exists by transfer, since for each n there is a sequence indiscernible for the first n formulas of Ω. Let $d_1, \ldots, d_K$ be any subsequence of the c_i's of length K.

There is an (internal) order-preserving bijection from the active domain of M_1 to $c_1, \ldots, c_H$. This is true by transfer, since the nonstandard universe believes that for any two subsets of $\mathbb{U}$ with the same cardinality, there is an order-preserving map from one to the other. Let M_1' be the image of M_1 under this mapping — the active domain of M_1' is now $c_1, \ldots, c_H$. Similarly, we can get a model M_2' with active domain $d_1, \ldots, d_K$ by applying a different order-preserving mapping. Since the mappings preserve order, we still have that M_1' and M_2' also agree on every $<$-query. By the genericity of Q, M_1' and M_2' still disagree about Q.

Consider M_1' and M_2' as models for the first-order language containing the schema predicates and $<$, with $^*\mathbb{U}$ as the domain of both models. The fact that M_1' and M_2' agree on all $<$-queries says exactly that these two models are elementary equivalent.

Applying the Isomorphism Principle to these models, we get that there is a mapping f from $^*\mathbb{U}$ onto itself that is an isomorphism (in the language of $<$ plus the schema relations) of M_1' onto M_2'. Note that since f is an isomorphism of the database relations, f maps something in the active domain of M_1' to something in the active domain of M_2'. Hence f maps the $c_1, \ldots, c_H$ bijectively onto $d_1, \ldots, d_K$. Also note that f is external.

Now we consider a new language L' with symbols for the elements of Ω and $<$, plus constant symbols k_I for $I \leq H$. Let N_1 be the model for L' with domain $^*\mathbb{U}$, the elements of Ω and order interpreted in the usual way, and with k_I interpreted by c_I for $I \leq H$. Let N_2 be the same, except k_I is interpreted by $f(c_I)$. Note that the model N_2 is not internal.

We define the relation $\vec{x} \Rightarrow \vec{y}$ on vectors of the same length from ${}^*\mathbb{U}$ to mean that the model $\langle N_1, \vec{x}\rangle$ is L'-elementary equivalent to $\langle N_2, \vec{y}\rangle$.

Proposition 10 $\emptyset \Rightarrow \emptyset$

Proof. Let $\phi(k_{j_1}, \ldots, k_{j_n})$ be any sentence in L' satisfied by N_1. That is, ${}^*\Omega, c_{j_1}, \ldots, c_{j_n}$ satisfies ϕ', where ϕ' is the formula mentioning only the symbols in the language of Ω that is obtained from ϕ by replacing each k_j by a free variable. By indiscernibility, ${}^*\Omega, c_{m_1}, \ldots, c_{m_n}$ satisfies ϕ' whenever the m_i's are ordered the same as the j_i's. Since the mapping f is order-preserving, and since (as noted above) it maps each c_i to some d_j, we get that ${}^*\Omega, f(c_{j_1}), \ldots, f(c_{j_n})$ satisfies ϕ'. But this means N_2 satisfies ϕ. □

Proposition 11 *The relation $\vec{x} \Rightarrow \vec{y}$ has the back and forth property. That is, if $\vec{x} \Rightarrow \vec{y}$, then*

- *For each w in ${}^*\mathbb{U}$ there is z in ${}^*\mathbb{U}$ such that $\langle \vec{x}, w\rangle \Rightarrow \langle \vec{y}, z\rangle$.*
- *For each w in ${}^*\mathbb{U}$ there is z in ${}^*\mathbb{U}$ such that $\langle \vec{x}, z\rangle \Rightarrow \langle \vec{y}, w\rangle$.*

Proof. Let $\vec{x}$ and $\vec{y}$ with $\vec{x} \Rightarrow \vec{y}$ be given, and let w in ${}^*\mathbb{U}$ be arbitrary. Let L^n be the language with symbols for the operations in Ω, and a constant symbol k_I for each $I \leq H$, and n extra constants where n is the length of $\vec{x}$.

Suppose

Claim 5 *There is a countable set of formulas $\Gamma(z)$ in L^n with the property that 1) every formula in Γ is satisfied by w in $\langle N_1, \vec{x}\rangle$ and 2) for every formula $\beta(z)$ satisfied by w in $\langle N_1, \vec{x}\rangle$, there is a formula $\tau(z)$ in Γ such that $\langle N_1, \vec{x}\rangle$ satisfies $\forall z.\tau(z) \rightarrow \beta(z)$.*

Then the first part of the proposition can be argued as follows. First, we show that $\Gamma(z)$ is satisfied in $\langle N_2, \vec{y}\rangle$. That is, with the constants interpreted by $f(c_i)$ instead of c_i and the finitely many extra constants interpreted by $\vec{y}$ instead of $\vec{x}$. By Proposition 7, it suffices to show that $\Gamma(z)$ is finitely satisfied in $\langle N_2, \vec{y}\rangle$. But this follows from the fact that $\Gamma(z)$ is finitely satisfiable in $\langle N_1, \vec{x}\rangle$ and the fact that $\vec{x} \Rightarrow \vec{y}$.

So we have a w' that satisfies $\Gamma(z)$ in $\langle N_2, \vec{y}\rangle$. We now show that this w' satisfies all the same formulas of L^n that w does in $\langle N_1, \vec{x}\rangle$. Let $\phi(z)$ be a formula of L^n satisfied by w in $\langle N_1, \vec{x}\rangle$. By Claim 5, there is a formula $\tau(z)$ of $\Gamma(z)$ such that

$$\langle N_1, \vec{x}\rangle \models \forall z.\tau(z) \rightarrow \phi(z)$$

Since $\vec{x} \Rightarrow \vec{y}$ we have

$$\langle N_2, \vec{y}\rangle \models \forall z.\tau(z) \rightarrow \phi(z)$$

Since w' satisfies Γ in $\langle N_2, \vec{y}\rangle$, it satisfies τ in $\langle N_2, \vec{y}\rangle$. Therefore, w' satisfies ϕ in $\langle N_2, \vec{y}\rangle$ as desired.

The second part of the proposition is proved similarly by assuming the analogous claim below on $\langle N_2, \vec{y}\rangle$.

Claim 6 *There is a countable set of formulas $\Gamma(z)$ in L^n with the property that 1) every formula in Γ is satisfied by w in $\langle N_2, \vec{y}\rangle$ and 2) for every formula $\beta(z)$ satisfied by w in $\langle N_2, \vec{y}\rangle$, there is a formula $\tau(z)$ in Γ such that $\langle N_2, \vec{y}\rangle$ satisfies $\forall z.\tau(z) \rightarrow \beta(z)$.*

Next, we prove Claim 5. For any one-variable formula $\psi(y)$ in the language of Ω, possibly including parameters from $^*\mathbb{U}$, an **endpoint** of $\psi(y)$ means an endpoint of some maximal interval contained in the subset of $^*\mathbb{U}$ defined by ψ. By o-minimality, each such formula has only finitely many endpoints. Also, if we have two endpoints $x_0 < x_1$ for the formula $\psi(y)$, and there is no other endpoint for $\psi(y)$ lying between these two, then the truth value of $\psi(y)$ is constant on the open interval (x_0, x_1).

For each formula $\chi(\vec{z}, y)$ in the language of Ω with parameters from $\vec{x}$, let

$$a_{l\chi} = \max\{u \mid \text{there is } \vec{c} \text{ from } c_1, \ldots c_H \text{ such that } u \text{ is an endpoint of } \chi(\vec{c}/\vec{z}, y) \text{ and } u \leq w\}$$

$$a_{r\chi} = \min\{u \mid \text{there is } \vec{c} \text{ from } c_1, \ldots c_H \text{ such that } u \text{ is an endpoint of } \chi(\vec{c}/\vec{z}, y) \text{ and } w \leq u\}$$

Then $a_{l\chi} \leq w \leq a_{r\chi}$ and there is a vector $\vec{c}$ from $c_1, \ldots, c_H$ such that both $a_{r\chi}$ and $a_{l\chi}$ are definable from $\vec{c} \cup \vec{x}$ (since there are only finitely endpoints for $\chi(\vec{c}/\vec{z}, y)$, and we have a linear ordering to distinguish these endpoints, each endpoint is definable from the parameters in the formula χ).

Case 1. If $a_{l\chi} = w$ or $a_{r\chi} = w$ for some χ, then w is definable from parameters in $c_1, \ldots, c_H$ by a formula ψ in the language of Ω. By replacing the parameter c_i by the constant symbol from k_i, we get that w is definable in L^n by a single formula $\psi(y)$. This clearly proves the claim as we can take $\Gamma(y)$ to be $\psi(y)$.

Case 2. Suppose $a_{l\chi} < w < a_{r\chi}$ for each χ. Now let $\phi_\chi(y)$ be the formula

$$\exists z \exists z'.\phi_{1\chi}(z) \wedge \phi_{2\chi}(z') \wedge z \leq y \leq z'$$

where $\phi_{1\chi}$ and $\phi_{2\chi}$ are the formulas defining $a_{l\chi}$ and $a_{r\chi}$. Let $\psi_\chi(y)$ be the formula of L^n formed by replacing each parameter c_I by the constant symbol k_I. Then each ψ_χ is a formula of L^n.

Now let $\Gamma(y) = \langle \psi_\chi(y) \rangle$, where χ varies over the countable set of formulas in the variables $\vec{z}$ and y with parameters from $\vec{x}$. Then we show that $\Gamma(y)$ works.

If we have another element w' in $^*\mathbb{U}$ and $\chi(y)$ is a formula in L^n satisfied by w, then there is a formula $\chi'(\vec{z}, y)$ in the language of Ω with parameters from $\vec{x}$ such that χ is obtained from χ' by replacing variables in $\vec{z}$ by constant symbols from $\langle k_I \rangle_{I<H}$. But then the definitions imply that $\chi(y)$ does not change truth value between $a_{l\chi'}$ and $a_{r\chi'}$, hence χ is of constant truth value strictly between these two elements.

We claim that $\langle N_1, \vec{x}\rangle \models \psi_{\chi'}(y) \rightarrow \chi(y)$. Indeed, if w' satisfies $\psi_{\chi'}(y)$, then both w and w' lie strictly between $a_{l\chi'}$ and $a_{r\chi'}$. Therefore, $\chi(w) \leftrightarrow \chi(w')$, and hence $\chi(w')$ holds. This finishes the proof of Claim 5.

We now briefly trace out the proof of Claim 6 to complete the proof of the proposition. For each formula $\chi(\vec{z}, y)$ in the language of Ω with parameters from $\vec{y}$, let

$$a_{l\chi} = max\{u \mid \text{there is } \vec{d} \text{ from } d_1, \ldots d_K \text{ such that } u \text{ is an endpoint of } \chi(\vec{d}/\vec{z}, y) \text{ and } u \leq w\}$$

$$a_{r\chi} = min\{u \mid \text{there is } \vec{d} \text{ from } d_1, \ldots d_K \text{ such that } u \text{ is an endpoint of } \chi(\vec{d}/\vec{z}, y) \text{ and } w \leq u\}$$

Case 1. If $a_{l\chi} = w$ or $a_{r\chi} = w$ for some χ, then w is definable from parameters coming from the d_I's and $\vec{y}$ by a formula ψ in the language of Ω. By replacing the parameter d_I with the constant symbol k_J, where $f(c_J) = k_I$, we get that w is definable in L^n by a single formula $\psi'(y)$. This clearly proves the claim since we can take $\Gamma(y) = \psi'(y)$.

Case 2. Suppose $a_{l\chi} < w < a_{r\chi}$ for each χ. Now let $\phi_\chi(y)$ be the formula

$$\exists z \exists z'. \phi_{1\chi}(z) \wedge \phi_{2\chi}(z') \wedge z \leq y \leq z'$$

where $\phi_{1\chi}$ and $\phi_{2\chi}$ are the formulas defining $a_{l\chi}$ and $a_{r\chi}$. Let $\psi_\chi(y)$ be the formula of L^n formed by replacing each parameters d_I by the constant symbol k_J as before (i.e. k_J such that $f(c_J) = d_I$). Then each ψ_χ is a formula of L^n.

We now let $\Gamma(y) = \langle \psi_\chi(y) \rangle$ as before, and check that this works. This completes the proof of Proposition 11. □

Finally, let the language L^+ contain symbols for the database relations in the schema, and also the operations in Ω. Let P_1 and P_2 be the expansions of M_1' and M_2' to L^+. Then we have

Proposition 12 *P_1 is elementary equivalent in L^+ to P_2.*

Proof. We show how to win the Ehrenfeucht game for L^+ (equivalently, we show P_1 is partially isomorphic to P_2). If our opponent plays c_I in P_1, then we play $f(c_I)$ in P_2, and if our opponent plays d_I in P_2, then we play $f^{-1}(d_I)$ in P_1. If our opponent plays a w in P_1 that is not one of the c_I's, then we apply Proposition 11 to get our response in P_2, and similarly when our opponent plays in P_2. The fact that this strategy works follows from Proposition 11 and the fact that the mapping f is an isomorphism of the schema relations. □

But now we have two models P_1 and P_2 that agree on every Ω-query, but disagree about Q (since they are expansions of M_1' and M_2'). Hence Q cannot be expressible in the language of Ω, which contradicts the assumption on Q. This concludes the proof of Theorem 4.

6 Putting it all together

In this section, we first tie together our results on the active semantics and the natural semantics of generic queries. Then we provide a summary of this paper followed by some concluding remarks.

Theorem 5 *Let $\Omega_{o\text{-}min}$ be an o-minimal signature. Then $\mathcal{NFO}^{\mathbf{LG}}(\Omega_{o\text{-}min}, <) = \mathcal{AFO}(<)$.*

Proof. Let Q be in $\mathcal{NFO}^{\mathbf{LG}}(\Omega_{o\text{-}min}, <)$. By Theorem 4 it is in $\mathcal{NFO}^{\mathbf{LG}}(<)$ and hence in $\mathcal{NFO}^{\mathbf{LG}}(\mathbb{R}, +, -, 0, 1, <)$. By Fact 2, Q is in $\mathcal{AFO}^{\mathbf{LG}}(\mathbb{R}, +, -, 0, 1, <)$. By Theorem 1 it is in $\mathcal{AFO}(<)$. Note that even though we have used the real domain in the argument above, $\Omega_{o\text{-}min}$ does not need to be on the reals. □

$$
\begin{tikzcd}
\mathcal{AFO}^{\mathbf{TG}}(\Omega_{sparse}) \arrow[rr, hook] \arrow[d, equal] & & \mathcal{NFO}^{\mathbf{LG}}(\Omega_{o\text{-}min}, <) \arrow[r, equal] \arrow[d, equal] & \mathcal{AFO}^{\mathbf{LG}}(\Omega, <) \arrow[d, equal] \\
\mathcal{AFO}^{\mathbf{TG}}(+,-,*,0,1) \arrow[r, hook] \arrow[d, equal] & \mathcal{NFO}^{\mathbf{TG}}(+,*,0,1) \arrow[r, hook] \arrow[d, equal] & \mathcal{NFO}^{\mathbf{LG}}(+,*,0,1) \arrow[r, equal] \arrow[d, equal] & \mathcal{AFO}^{\mathbf{LG}}(+,-,*,0,1,<) \arrow[d, equal] \\
\mathcal{AFO}^{\mathbf{TG}}(+,-,0,1) \arrow[r, hook] \arrow[d, equal] & \mathcal{NFO}^{\mathbf{TG}}(+,-,0,1,<) \arrow[r, hook] & \mathcal{NFO}^{\mathbf{LG}}(+,-,0,1,<) \arrow[r, equal, "\text{[24]}"] \arrow[d, equal] & \mathcal{AFO}^{\mathbf{LG}}(+,-,0,1,<) \arrow[d, equal] \\
\mathcal{AFO} \arrow[r, equal, "\text{[15]}"] & \mathcal{NFO}^{\mathbf{TG}} \arrow[r, hook] \arrow[u, hook] & \mathcal{NFO}^{\mathbf{LG}}(<) \arrow[r, equal] & \mathcal{AFO}(<)
\end{tikzcd}
$$

Figure 1: Summary of the expressiveness results for databases over the reals

It follows immediately that

Corollary 2 *Let $\Omega_{o\text{-}min}$ be an o-minimal signature. Then $\mathcal{NFO}(\Omega_{o\text{-}min}, <)$ cannot express transitive closure, deterministic transitive closure, parity test, and connectivity test. In particular, none of the above is expressible under the natural interpretation of the relational calculus with constraints of the form $f(\vec{x})\theta g(\vec{x})$ where $\theta \in \{=, \leq\}$ and f, g are functions definable in the signature $(+, -, *, e^x, 0)$.* $\square$

In conclusion, we have thus settled the open problem of whether parity and connectivity can be expressed in the relational calculus with arithmetic constraints. In addition, we have shown that the addition of arithmetics does not give us more power to define generic queries, for both the active domain semantics and the natural semantics. In fact, we have proved that the two semantics often coincide when limited to generic queries. Thus, we have given a clear picture of the expressiveness of the relational calculus with arithmetic constraints, where generic queries are concerned.

The diagram shown in Figure 1 summarizes our expressiveness results for databases over the reals. By $\Omega_{o\text{-}min}$ we mean any o-minimal signature and by Ω_{sparse} we mean any sparse signature. The $=$ arrow means equality and the $\hookrightarrow$ arrow means proper embedding. That $\mathcal{AFO}^{\mathbf{TG}}(+,-,*,0,1)$ $\hookrightarrow$ $\mathcal{NFO}^{\mathbf{TG}}(+,*,0,1)$ follows from the fact that there are TG-queries that are not expressible in relational calculus without order. The embedding $\mathcal{NFO}^{\mathbf{TG}}(+,*,0,1) \hookrightarrow \mathcal{NFO}^{\mathbf{LG}}(+,*,0,1)$ follows from the following observation. Let the scheme contain two unary relation symbols R_1 and R_2 and let Q be $\forall x \forall y.(R_1(x) \wedge R_2(y)) \rightarrow x < y$. Since Q is LG but not TG, it separates $\mathcal{NFO}^{\mathbf{LG}}(+,*,0,1)$ from $\mathcal{NFO}^{\mathbf{TG}}(+,*,0,1)$. Other embeddings and equations follow from the results of this paper.

Acknowledgements. We thank Jianwen Su, Jan Van den Bussche, Martin Otto, and Dirk Van Gucht for discussions, encouragements, and references.

References

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

[2] F. Afrati, S. Cosmadakis, S. Grumbach and G. Kuper. Linear vs polynomial constraints in database query languages. In *Proceedings of Conference on Constraint Programming*, 1994.

[3] F. Afrati, T. Andronikos and T. Kavalieros. On the expressiveness of first-order query languages. To appear in *Proceedings of Conference on Constraint Programming*, 1995.

[4] A. V. Aho and J. D. Ullman. Universality of data retrieval languages. In *Proceedings of 6th Symposium on Principles of Programming Languages, Texas*, pages 110–120, January 1979.

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

[6] J. Denef and L. Van den Dries. p-adic and real subanalytics sets. *Annals of Mathematics* 128 (1988), 79-138.

[7] R. Fagin. Probabilities on finite models. *Journal of Symbolic Logic*, 41(1):50-58, 1976.

[8] H. Gaifman. On local and non-local properties. In *Proceedings of the Herbrand Symposium, Logic Colloquium '81*, pages 105–135, North Holland, 1982.

[9] S. Grumbach and T. Milo. Towards tractable algebras for bags. In *Proceedings of 12th ACM Symposium on Principles of Database Systems, Washington, D. C.*, pages 49–58, May 1993.

[10] S. Grumbach and J. Su. Finitely representable databases, In *Proceedings of 13th ACM Symposium on Principles of Database Systems, Minneapolis, Minnesota*, pages 289–300, May 1994.

[11] S. Grumbach and J. Su. First-order definability over constraint databases. To appear in *Proceedings of Conference on Constraint Programming*, 1995.

[12] S. Grumbach, J. Su, and C. Tollu. Linear constraint databases. In *Proceedings of LCC, 1994.*

[13] C.W. Henson. The isomorphism property in nonstandard analysis and its use in the theory of Banach spaces. *Journal of Symbolic Logic* 39 (1974), 717–731.

[14] W. Hodges. *Model Theory.* Cambridge University Press, 1992.

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

[16] P. Kanellakis. Constraint programming and database languages: A tutorial. In *Proceedings of 14th ACM Symposium on Principles of Database Systems, San Jose Ca*, pages 46–53, May 1995.

[17] G. Kuper. On the expressive power of the relational calculus with arithmetic constraints. In *Proceedings of International Conference on Database Theory*, pages 201-211, 1990.

[18] P. Kanellakis, G. Kuper, and P. Revesz. Constraint query languages. *Journal of Computer and System Sciences*, to appear. Extended abstract in *Proceedings of 9th ACM Symposium on Principles of Database Systems, Nashville, Tennessee*, pages 299–313, May 1990.

[19] L. Libkin and L. Wong. Some properties of query languages for bags. In *Proceedings of 4th International Workshop on Database Programming Languages, New York*, pages 97–114, August 1993.

[20] L. Libkin and L. Wong. Aggregate functions, conservative extension, and linear orders. In *Proceedings of 4th International Workshop on Database Programming Languages, New York*, pages 282-294, August 1993.

[21] L. Libkin and L. Wong. Conservativity of nested relational calculi with internal generic functions, *Information Processing Letters* 49:273–280, 1994.

[22] L. Libkin and L. Wong. New techniques for studying set languages, bag languages, and aggregate functions. In *Proceedings of 13th ACM Symposium on Principles of Database Systems, Minneapolis, Minnesota*, pages 155–166, May 1994.

[23] J. Paredaens, J. Van den Bussche, and D. Van Gucht. Towards a theory of spatial database queries. In *Proceedings of 13th ACM Symposium on Principles of Database Systems, Minneapolis, Minnesota*, pages 279–288, May 1994.

[24] J. Paredaens, J. Van den Bussche, and D. Van Gucht. First-order queries on finite structures over the reals. In *Proceedings of 10th IEEE Symposium on Logic in Computer Science, San Diego, California*, pages 79–87, 1995.

[25] A. Pillay, C. Steinhorn. Definable sets in ordered structures Bulletin of the AMS, 11,pages 159-62.

[26] P. Revesz. The computational complexity of constraint query languages. In *Working notes of the Workshop on Semantics in Databases*, Prague, 1995. Full paper to appear in the Proceedings of the Workshop.

[27] J. G. Rosenstein. *Linear Orderings.* Academic Press, New York, 1982.

[28] W. Rudin. *Real and Complex Analysis.* McGraw-Hill, New York, 1966.

[29] J. Van den Bussche. Private email. June 1995.

[30] L. Van den Dries. New o-minimal expansions of the reals. Lecture. May 1995

[31] L. Van den Dries, A. Macintyre and D. Marker. The elementary theory of restricted analytic fields with exponentiation. *Annals of Mathematics* 85 (1994), 19–56.

[32] A.J. Wilkie. Model completeness results for expansions of the ordered field of real numbers by restricted Pfaffian functions and the exponentail function. To appear in Journal of the AMS.

[33] L. Wong. Normal form and conservative properties for query languages. In *Proc. 12th ACM Symp. on Principles of Database Systems*, 1993.