Relational Expressive Power of Constraint Query Languages*

Michael Benedikt
Bell Laboratories
1000 East Warrenville Rd
Naperville, IL 60566-7013, USA
Email: benedikt@bell-labs.com

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

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

Limsoon Wong †
BioInformatics Center &
Institute of Systems Science
Heng Mui Keng Terrace, Singapore 119597
E-mail: limsoon@iss.nus.sg

Dedicated to the memory of Paris C. Kanellakis

Abstract

The expressive power of first-order query languages with several classes of equality and inequality constraints is studied in this paper. We settle the conjecture that recursive queries such as parity test and transitive closure cannot be expressed in the relational calculus augmented with polynomial inequality constraints over the reals. Furthermore, noting that relational queries exhibit several forms of genericity, we establish a number of collapse results of the following form: The class of generic boolean queries expressible in the relational calculus augmented with a given class of constraints coincides with the class of queries expressible in the relational calculus (with or without an order relation). We prove such results for both the natural and active-domain semantics. As a consequence, the relational calculus augmented with polynomial inequalities expresses the same classes of generic boolean queries under both 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 certain kinds of 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 ordered structures.

*Submitted to *Journal of the ACM*. Extended abstract in *Proceedings of 15th ACM Symposium on Principles of Database Systems*, Montreal, Canada, 5–16, June 1996

†Part of this work was done when Limsoon Wong was visiting University of Melbourne and Bell Labs in Murray Hill.

1 Introduction

Much of the work in the foundation of relational databases revolves around using techniques from logic to formalize the data model and to analyze the expressive power of query languages. A database relation is formalized as a finite collection of tuples, and a database is modeled as a finite structure, which is a collection of relations. Database queries can then be modeled as formulae on these structures. The first fundamental result is that classical query languages, such as relational algebra and calculus, have precisely the power of first-order logic. From there, we can use logical techniques to derive important bounds on the expressiveness of these relational languages, such as the inexpressibility of parity and graph connectivity [4, 9].

In new database applications involving spatial data (as in geographical databases) and temporal data, it is necessary to move beyond the relational model of data, and 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 [20], 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. For example, a generalized tuple $x + y > 5$ represents the infinite set of tuples (x, y) such that the constraint $x + y > 5$ holds.

A generalized relation is a finite set of generalized tuples. Interesting constraint query languages are then obtained by coupling traditional relational query languages, such as the relational calculus, with various classes of arithmetic constraints. Examples of queries that are inexpressible in the pure relational calculus but are expressible with such an extension include the test of whether all points in a binary relation R lie on some common circle or whether R contains four vertices of some diamond.

Thus, the coupling of relational calculus with arithmetic constraints enhances power. A natural question arises, attracting much attention recently: How much more power can we gain from this coupling? The following conjecture, discussed extensively in the literature [22, 20, 19, 30, 14, 26], has been open for several years.

Conjecture. *Queries such as transitive closure, connectivity test, and 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 is known that they cannot be expressed by the relational calculus. It was noted in [13] that useful properties for proving the inexpressibility of these queries in the relational calculus, such as locality [12] and 0/1-law [11], do not carry over to constraint query languages. Nevertheless, a number of inexpressibility results were established recently. In [15] 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. In [3] it is shown that testing whether a constraint database represents a line is not definable in first-order logic with order.

Transitive closure, parity test, and connectivity test are examples of *generic* queries [9, 18]. Generic queries cannot distinguish between "isomorphic" databases. Formally, 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.

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 two different semantics of the relational calculus. Under the *active semantics*, quantification variables are assumed to range over the active domain of the database, that is, the set of all elements that occur in the database. Under the *natural semantics*, quantification variables are assumed to range over the whole universe (for example, the real line in the case of polynomial constraints over the reals).

We prove the following main results.

1. 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 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 ordered structures.
3. As a consequence, the conjecture mentioned above is confirmed. It also follows that the relational calculus plus polynomial inequality constraints expresses the same generic boolean 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 [17] and of the relational calculus with linear constraints by Paredaens, Van den Bussche, and Van Gucht [27]. These two results, [17] and [27], are not limited to generic queries. Thus we have generalized these two results to polynomial constraints, when queries are restricted to generic ones. Similar techniques can be used to show the coincidence of the two semantics for arbitrary polynomial constraints, as is shown in [8]. It was also shown in [27] that linear constraints do not add pure expressive power beyond $<$. Our results generalize this to a wider class of signatures, including polynomial constraints and exponentiation. Another generalization of this kind that is similar to ours but uses a slightly different setting was found independently by Otto and Van den Bussche [25].

In contrast to our results, Grumbach and Su [14] 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 to get a collapse result for the natural semantics. Our most general natural-semantics result uses signatures that are *o-minimal* [28] — these signatures can define only a certain kind of subsets of the real line. This restriction is sufficiently general to confirm the main conjecture for polynomial constraints.

In this paper we concentrate on expressiveness of constraint query languages on ordinary relational databases. It is possible to use such results to find some expressivity bounds for constraint databases, that is, sets of generalized tuples. General techniques for such extensions are discussed in [14, 33].

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. We prove powerful collapse theorems saying that the expressibility of these queries is *independent* of the presence of arithmetic and other operators. 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. We prove an analogous collapse theorem saying that the expressibility of these queries is independent of the presence of a large class of arithmetic and other operators in this more powerful framework. This result is proved via an excursion through nonstandard models. The main conjecture follows as a corollary.

Section 6 extends the results of sections 4 and 5 to arbitrary (nonboolean) queries. We prove that every collapse result for boolean queries implies the corresponding collapse result for nonboolean queries.

Section 7 concludes the paper by relating the "active" and the "natural" results of this paper. For some signatures, it is known that the two semantics of relational calculus coincide. Using this and our collapse theorems, we extend the coincidence of the active and natural semantics, with respect to generic queries, to a larger number of additional primitives.

An extended abstract of this paper appeared in the Proceedings of the 15th Symposium on Principles of Database Systems [7].

2 Notations

A database **schema** is a nonempty collection of relation names, $\langle R_1, \ldots, R_k \rangle$, where each name R_i is assigned an arity τ_i. We fix a database schema SC for now, and we also fix a database domain, which is an infinite set $\mathbb{U}$. All values that occur in databases are drawn from this domain.

A **database** (or database instance) 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 DB, its **active domain** $\mathit{adom}(DB)$ 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 in the language containing $\langle R_1, \ldots, R_k \rangle$ and equality via the usual logical connectives and quantifiers of the form $\forall x$ and $\exists x$. If we allow our atomic formulae to also mention a symbol $<$, interpreted as a linear order on $\mathbb{U}$, then we speak of a **$<$-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 DB is the value of

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

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

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

The notion of satisfaction is thus defined straightforwardly for both semantics. Since it will always be clear from the context which semantics is being used, we shall write $DB \models Q$ if Q evaluates to $\mathbf{T}$, true, on DB.

We shall also write $Q(DB)$ for the value of Q on DB. That is, $Q(DB)$ is either $\mathbf{T}$ or $\mathbf{F}$. Thus, each query defines a semantic object, that is, a map from $Inst(\mathbb{U}, SC)$ to $\{\mathbf{T}, \mathbf{F}\}$.

The classes of first-order Boolean queries (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, $\mathcal{AFO}(\mathbb{R}, +, -, *, 0, 1, <)$ 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 the last 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. We also do not need minus, since it is definable using $+$ and natural quantification.

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

Fact 1 [17] $\mathcal{AFO} = \mathcal{NFO}$

Fact 2 [27] $\mathcal{AFO}(\mathbb{R},+,-,0,1,<) = \mathcal{NFO}(\mathbb{R},+,-,0,1,<)$

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

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. Intuitively, a query is generic if it returns the same answer for "isomorphic" databases. Typically, what is meant by "isomorphic" databases DB_1 and DB_2, is that applying some permutation π on $\mathbb{U}$ to DB_1 yields DB_2. In other words, a generic query is then a query that is invariant under arbitrary permutations of the domain [1].

Sometimes this notion must be relaxed. Suppose that database elements are ordered, and a query may refer to the ordering. In this case the right notion of genericity is invariance under maps that preserve the order relation. For constraint databases, even more complex notions of genericity have been considered [26].

In this paper we use three notions of genericity for Boolean queries. Let $\varphi : \mathbb{U} \to \mathbb{U}$ be a mapping on the domain. Then φ can be extended to databases over $\mathbb{U}$: φDB denotes a database obtained from DB by replacing each occurrence of $x \in adom(DB)$ by $\varphi(x)$.

Definition 2

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

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 $\mathcal{L}$. □

For example, $\mathcal{AFO}^{\mathbf{TG}}(<)$ is the class of TG-queries definable in relational calculus with order, while $\mathcal{NFO}^{\mathbf{LG}}(\mathbb{R},+,*,0,1)$ is 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* [31] 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 ordered 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 DB with $D = adom(DB)$. 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 $Q(\pi_D DB) = Q(\pi DB) = Q(DB)$, which implies that Q is LG. If Q is LG, consider any monotone injection $\varphi : \mathbb{U} \to \mathbb{U}$. Given a database with $D = adom(DB)$. Let φ_D be the restriction of φ on D. Then $Q(\varphi DB) = Q(\varphi_D DB) = Q(DB)$. 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 [31]. $\square$

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 ([31]) *$\langle \mathbb{Q}, < \rangle$ and $\langle \mathbb{R}, < \rangle$ are doubly transitive. Also, any 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.

Finally, we note that the same query Q may be generic under the active interpretation and nongeneric under the natural, and vice versa. For example, consider $Q \equiv \exists x \forall y. R(x) \wedge ((x < y) \vee (x = y))$, where R is a unary schema predicate and $<$ interprets the usual order on the natural numbers. This is equivalent to true (for nonempty databases) under the active domain interpretation (hence generic), but it is nongeneric under natural semantics. Let $Q \equiv (\exists x \exists y. R(x) \wedge R(y) \wedge R(x+y)) \wedge (\exists x \forall y. x \leq y)$, where $+$ and $\leq$ interpret the addition and the order over the reals. This Q is equivalent to false under the natural interpretation, but it is non-generic under the active interpretation.

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 this section by stating the main results below. We then introduce the main technical tool that 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. In subsection 4.2, we state these Ramsey Theorems and show how the main collapse results follow from them. In subsection 4.3 we prove the Ramsey Theorems. The final subsection 4.4 steps through an example.

4.1 Statement of Main Results

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 $<$-rbq. In other words,*

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

We will state a number of corollaries of this result. Several of these were independently proved by Van den Bussche and Otto [25], who used the Ehrenfeucht-Mostowski theorem about the existence of indiscernibles [10].

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 so is any discrete linear order without endpoints (in particular, $\langle \mathbb{Z}, < \rangle$).

Corollary 1

- *Let $\langle \mathbb{U}, < \rangle$ be weakly homogeneous and let Ω be an arbitrary signature. Then $\mathcal{AFO}^{\mathbf{MG}}(\Omega, <) \;=\; \mathcal{AFO}(<)$.*
- ([25]) *For an ordered domain $\langle \mathbb{U}, < \rangle$, and any signature Ω, $\mathcal{AFO}^{\mathbf{TG}}(\Omega, <) = \mathcal{AFO}^{\mathbf{TG}}(<)$.*
- *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]. Therefore, collapse results in the unordered case can only exist for a restricted class of signatures. Although we are far from having a good characterization of the class of signatures for which this holds, we will show a few collapse results for particular classes of signatures below.

Let the domain be $\mathbb{R}$, and $\Omega = (h_1, h_2, \ldots)$ be a signature that consists of functions $\mathbb{R}^{arity(h_i)} \to \mathbb{R}$. We call Ω **sparse** if the subtraction function 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 is either identical to zero or has at most countably many zeros, and composition of analytic functions is analytic again [24].

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.* □

Some examples of sparse signatures are $(+, -, *, e^x, 0)$, $(+, -, 0, 1)$, and $(+, -, *, 0, 1)$.

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 Theorems 1 and 2 are of a very different nature from complexity-based results such as [15].

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. But if we are only interested in the ring structure, there is a 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,*

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

4.2 Ramsey Theorems for Constraint Databases and Proofs of Main Theorems

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 DB,

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

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 $<$-rbq Q' which is $\mathbb{U}_Q$-equivalent to Q.* $\square$

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.* $\square$

A similar result to Proposition 4 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.* $\square$

Now, using these propositions, we can present the straightforward proofs of the main results of this section.

Proof of Theorem 1. Consider a locally generic cbq Q and find a set $\mathbb{U}_Q$ and a $<$-rbq Q' as in Proposition 3. Now we claim that Q and Q' are equivalent. Indeed, take any database DB and let $D = adom(DB)$. 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, $Q(DB) = Q(\pi DB)$. By Proposition 3, $Q(\pi DB) = Q'(\pi DB)$ since $adom(\pi DB) \subseteq \mathbb{U}_Q$. Since Q' is a $<$-rbq, it is locally generic, and hence $Q'(\pi DB) = Q'(DB)$. Thus, $Q(DB) = Q'(DB)$, proving that Q and Q' are equivalent as desired.

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

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

4.3 Proofs of Ramsey Theorems for Constraint Databases

We begin with the ordered case.

Proof of Proposition 3.

Since an arbitrary first-order sentence can be converted into prenex normal form, from now on we assume that Q is given by (1). For convenience, we extend the language with the symbols $\neq$ and $\not<$ with the obvious interpretations. 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, $\theta \in \{=, <, \neq, \not<\}$, 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 $R_i(t_1(\vec{y}_1), \ldots, t_{\tau_i}(\vec{y}_{\tau_i}))$ 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 truth 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, since the symbols for $\neq$ and $\not<$ are added to the language.

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). Let Ψ_1 and Ψ_2 be the disjunctions of the conjuncts from the first and the second class respectively. Then $\Psi = \Psi_1 \vee \Psi_2$ and Ψ_1 is a formula in the language $\langle R_1, \ldots, R_k, =, \leq \rangle$.

Let P 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 indicies 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 $\bigvee_{i=1}^{\mathsf{P}} \zeta_i = \mathbf{T}$. Hence, Ψ_2 can be replaced by $\bigvee_{i=1}^{\mathsf{P}}(\zeta_i \wedge \Psi_2)$.

Let $c_1, \ldots, c_m$ denote all the constraints of the form (b) present in Ψ_2. By $\Psi_2[(c_j \wedge \zeta_i)/c_j]_{j \leq m}$ we denote the formula obtained from Ψ_2 as follows. For every disjunct in Ψ_2 (recall that Ψ_2 is in DNF) find all conjuncts in it which are of the form c_j, and replace them by $(c_j \wedge \zeta_i)$. Now it can be readily seen that Q is equivalent to

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

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*

$$(2') \qquad Q_1x_1.\ldots Q_nx_n.\Psi_1 \vee \bigvee_{i\in I} \Psi_2[(c_j \wedge \zeta_i)/c_j]_{j\leq m}$$

I is a nonempty subset of $\{1,\ldots,\mathsf{P}\}$ 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

$$(2'') \qquad Q_1x_1.\ldots Q_nx_n.\Psi_1' \vee \bigvee_{i\in I-\{i_0\}} \Psi_2[(c_j \wedge \zeta_i)/c_j]_{j\leq m}$$

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

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 $\Psi_2[(c_j \wedge \zeta_{i_0})/c_j]_{j\leq m}$ to $\Psi_2[c_j'/c_j]_{j\leq m} \wedge \gamma$ 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 at most 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 the family of nonempty sets among $A_0,\ldots,A_{2^m-1}$ forms 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, where $A_l \neq \emptyset$. Let l correspond to the truth values of c_j's being t_js. Let

$$\Psi^{i_0} \;=\; (\Psi_2[t_j/c_j]_{j\leq m} \;\wedge\; \gamma) \quad\text{and}\quad \Psi_1' = \Psi_1 \vee \Psi^{i_0}.$$

That is, Ψ^{i_0} is obtained from Ψ_2 by replacing each condition literal c_j by the corresponding truth value t_j. Then, for any n-tuples of elements of D', the values of

$$\Psi_1 \vee \bigvee_{i\in I} \Psi_2[(c_j \wedge \zeta_i)/c_j]_{j\leq m} \quad\text{and}\quad \Psi_1' \vee \bigvee_{i\in I-\{i_0\}} \Psi_2[(c_j \wedge \zeta_i)/c_j]_{j\leq m}$$

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 $\Psi_2[(c_j \wedge \zeta_{i_0})/c_j]_{j\leq m}$ coincide on such tuples.

Now observe the following. Assume that $\Phi_i(x_1,\ldots,x_n)$, $i = 1,2$, are formulae in n variables, and let $\Phi_1(\vec{d})$ and $\Phi_2(\vec{d})$ be equivalent for any $\vec{d} \in D^n$ for some set D. Then, for any quantifier prefix, the sentences $Q_1x_1 \in D \ldots Q_nx_n \in D.\Phi_1(\vec{x})$ and $Q_1x_1 \in D \ldots Q_nx_n \in D.\Phi_2(\vec{x})$ are equivalent.

Applying this observation, 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. $\square$

We now turn to the Ramsey result for sparse sets.

Proof of Proposition 4. We start by getting a normal form analogous to (2) for the unordered case. First, assume that Q is given by

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

where Ψ is in DNF, and its literals are of form $R_i(z_1, \dots, z_{\tau_i})$ 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-terms and $\theta \in \{=, \neq\}$; see the proof of Proposition 3. Further assume without loss of generality that no condition literal is negated.

Let B be the nth Bell number; that is, B is the number of partitions of an n-element set. Let ξ_i represent the ith partition on an n-element set of variables, $i = 1, \dots, B$. 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 $\bigvee_{i=1}^{B} \xi_i = \mathbf{T}$, we have $\Psi = \bigvee_{i=1}^{B}(\xi_i \wedge \Psi)$.

Consider each $\xi_i \wedge \Psi$. Let ξ_i specify the partition $X_1, \dots, X_s$ 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 $y_1, \dots, y_s$ 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

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

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 $f_1(\vec{x}) = 0, \dots, f_k(\vec{x}) = 0$ where the functions f_is are Ω-terms in at most n variables $x_1, \dots, x_n$. 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 $x_1, \dots, x_n$ simultaneously invalidates all the equations.*

This lemma immediately implies 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 DB with $adom(DB) \subseteq R$ it is the case that $Q(DB) = Q'(DB)$, and Q' does not mention any symbols from Ω, which concludes the proof.

To prove the lemma, we define $\Omega_{\mathbb{R}}$-functions as those given by a term in the signature Ω extended with constants for all elements of $\mathbb{R}$. Let $f(x_1, \dots, x_n)$ be a function in n variables and $1 \leq i \leq n$.

For $s \in \mathbb{R}$, 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 $\Omega_{\mathbb{R}}$-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 $\Omega_{\mathbb{R}}$-function $g(x_1, \ldots, x_n)$ which is not identically zero 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 every $\vec{c}$, 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 $\mathcal{F}_0 = \{f_1, \ldots, f_k\}$ 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. □

We now turn to the integral domain case.

Proof of Proposition 5. 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

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

where I is a finite set of indices, $M_i = (m_1^i, \ldots, m_n^i)$ is a multiindex (i.e., 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 $I = \emptyset$.*

We prove this claim by induction on n. For the base case $n = 1$ (i.e. $p \in \mathbb{K}[x]$) we use induction on $card(I)$. If I is empty, we are done. If I is a singleton, p cannot be identically zero because there are no divisors of zero. Assume $card(I) > 1$ and $p(x) = \sum_{i \in I} c_i x^{m_i}$. If $m_i = 0$ for some $i \in I$, then from $p(0) = 0$ we obtain $c_i = 0$, contradiction. Otherwise, let $m = \min m_i$, and apply the argument above to $p'(x)$ where $p(x) = x^m \cdot p'(x)$.

For the induction case $n > 1$, consider a polynomial p represented by (P1). Let $I \neq \emptyset$. Two cases arise. Case 1: for every $i \in I$ and every j between 1 and n it is the case that $m^i_j \neq 0$. Case 2: one can find $i \in I$ and $j \in 1, \ldots, n$ such that $m^i_j = 0$. In Case 1, let $\mu_j = \min_{i \in I} m^i_j$. Then $\mu_j > 0$ and we obtain $p = x_1^{\mu_1} \cdot \ldots \cdot x_n^{\mu_n} \cdot p'$ where p' is a polynomial which satisfies the condition of Case 2. By cancellation p' is identically zero. Hence, it is enough to prove Case 2 only. Assume that p is given by (P1) and assume without loss of generality that $m^1_1 = 0$. Define $p_1(x_2, \ldots, x_n)$ as $p(0, x_2, \ldots, x_n)$. Represent p_1 in the form (P1). Then c_1 remains one of the coefficients in this representation. But p_1 is a polynomial in $n-1$ variables, and is identically zero. Hence, it cannot be represented in form (P1) with nonempty set of coefficients. This contradiction finishes the proof of Claim 2.

Let $p \in \mathbb{K}[x_1, \ldots, x_n]$ be a polynomial in n variables. For any index i and any $s \in \mathbb{K}$ we denote the polynomial in $n-1$ 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 $p_1, \ldots, p_m \in \mathbb{K}[x_1, \ldots, x_n]$ 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 $p_j^{i,s}$, $j = 1, \ldots, m$, $i = 1, \ldots, n$, is identically zero.*

To prove this claim, assume that all p_js are represented in the form (P1) with $I \neq \emptyset$. Fix a polynomial p given by (P1), and define the equivalence relation $\approx_i$ on multiindices by $M_t \approx_i M_r$ iff $\forall j \neq i : m^t_j = m^r_j$. By Claim 2, $p^{i,s}$ is identically zero iff for every equivalence class $\{M_{i_1}, \ldots, M_{i_l}\}$ of $\approx_i$ we have

$$c_{i_1} s^{d_1} + \cdots + c_{i_l} s^{d_l} = 0 \tag{4}$$

where d_k is the ith component of M_{i_k}. Doing this operation for all p_js and all equivalence classes of $\approx_i$, 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. □

4.4 Example

It is difficult from the previous proofs to give concrete constructions for finding the Ramsey sets $\mathbb{U}_Q$. Although we make no claim to have deeply considered the algorithmic aspects of such transformations, we will now show how such transformations can be done for some of the standard arithmetic structures (see also [33]).

We start with an example. Consider a schema with one binary predicate S, and a query saying that for any pair in S, none of its components is the square of the other. That is, the query

$$Q \;\equiv\; \forall x \forall y. S(x,y) \rightarrow (\neg(x = y^2) \wedge \neg(y = x^2))$$

which is expressible in any language that contains multiplication. The underlying domain can be $\mathbb{R}$, or $\mathbb{Q}$, or $\mathbb{Z}$.

We claim that the Ramsey set $\mathbb{U}_Q$ can be chosen to be $\{3^{3^i} \mid i > 0\}$, and the equivalent query is just $\mathbf{T}$. Indeed, the constraint $x = y^2$ cannot be satisfied if $x, y \in \mathbb{U}_Q$: if we assume $3^{3^i} = (3^{3^j})^2$, then $3^i = 2 \cdot 3^j$ which is impossible for any $i, j > 0$. Since on such $\mathbb{U}_Q$ the constraint part of the query Q is always true, so is the whole query.

The above example generalizes straightforwardly to show that sparsely distributed sets necessarily give Ramsey sets for the active-semantics queries that use polynomial constraints.

Let D_r denote the set $\{r^{r^i} \mid i \in \mathbb{N}_+\}$.

Proposition 6 *Assume that $\mathbb{U}$ is either $\mathbb{R}$ or $\mathbb{Q}$ or $\mathbb{Z}$, and let Ω be $(+,*,0,1,<)$ or $(+,*,0,1)$. Then for any cbq Q there exists a number r in $\mathbb{N}$, and a $\leq$-rbq Q' (or rbq Q' if Ω does not contain order) that is D_r-equivalent to Q. Moreover, r and the corresponding Q' can be effectively constructed from Q.*

Proof. The proofs of all Ramsey 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 the polynomials 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, $j \in I$, with respect to $\succ$. 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 $\vec{x} = (x_1,\ldots,x_n) \in D_r \times \ldots \times D_r$ satisfying $x_{i_1} > x_{i_2} > \ldots > x_{i_n}$, 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 any natural number above all of these r_0.

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 any number in $\mathbb{N}$ above 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

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

Then a simple chain of calculations shows the following.

Claim 4 *If $r > n\Delta + 1$, then for any $\vec{x} \in D_r^n$ satisfying $x_1 > x_2 \ldots > x_n$ and for any two multiindices $M_i \neq M_k$, we have*

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

Define $G = \max_i(|\,c_i\,|) \cdot card(I)$ 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 $\vec{x} \in D_r^n$ satisfying $x_1 > x_2 \ldots > x_n$:

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

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, and Proposition 6 are proved. □

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 also be defined 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 [28]) 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 the form $\{x \mid xRa\}$ or $\{x \mid aRx\}$ or $\{x \mid aRxR'b\}$, where each binary relation R or R' is either $<$ or $\leq$.

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

- $(+, *, <, 0, 1)$ — this follows from Tarski's quantifier elimination theorem [34].
- $(+, *, e^x, 0)$ — this follows from [37, 36].
- $(+, *, e^x, \Gamma(x), 0)$ — this was proved in [32].

There are many other interesting examples of o-minimal structures, see [23, 35, 36].

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 $<$-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 on Nonstandard Universes

We start with some definitions of nonstandard universes. For more information, consult [10]. An overview of using techniques of nonstandard universes for proving expressivity bounds in given in [6].

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 $V_{n+1}(S) = V_n(S) \cup \{X \mid X \subseteq V_n(S)\}$. The set S is called the **base set** of the superstructure.

We will work with the structure $\langle V(S), \in \rangle$ considered as a structure for the first-order language for the membership 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 $V(Y)$ over infinite sets S and 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. $S \subset Y = {}^*S$.
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 contained in a standard set. Elements of $V(Y)$ that are not internal are called **external**. An internal map is a map whose graph is an internal set. We now explain the significance of a set being internal. Intuitively, properties of all sets in the smaller universe $V(S)$ will hold of all internal sets in the larger universe. Specifically, if we have a bounded-quantifier property that holds for all elements of some set A in $V(S)$ (for example, a property that holds of the set A of all sequences of graphs), we know by transfer that it applies to all elements of ${}^*A$ (that is, to all internal sequences of $*$-graphs). Hence this gives us that if a bounded-quantifier property P holds for all sequences from B, then the property P holds for all internal sequences from the set ${}^*B$. Similarly, a property that holds for all subsets of natural numbers, will hold for all internal subsets of ${}^*\mathbb{N}$.

We want our universe to be sufficiently "rich": to contain many nonstandard integers, for example. We will therefore assume that our universe also satisfies the following axiom (also standard in the literature):

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.

We will work with a superstructure whose base set S includes both the domain $\mathbb{U}$ of our databases and the integers $\mathbb{N}$. Now all objects such as pairs, tuples, predicates and functions from Ω "live" in the superstructure $V(S)$. For example, a pair (a, b), where $a, b \in V_n(S)$, can be encoded as $\{\{a\}, \{a, b\}\} \in V_{n+2}(S)$. We similarly encode tuples. Then relations are in the superstructure as sets of tuples, and so are functions since they can be associated with their graphs. Since $\mathbb{N} \subset S$, we define nonstandard integers as elements of ${}^*\mathbb{N}$. Then a **hyperfinite** set (a set whose cardinality is a nonstandard integer) is a set A for which there exists $H \in {}^*\mathbb{N}$ and an internal bijection from $\{K \mid K \in {}^*\mathbb{N}, K^* < H\}$ onto A. We can then talk about hyperfinite databases, hyperfinite sequences, etc.

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 $x_1 < x_2 + 3$ rather than ${x_1}^* {<}\, x_2^* +^* 3$.

5.2 Logic in Nonstandard Universes

We will consider the logical symbols as being coded by integers, and assume a countable set of variable symbols $x_1, \ldots$ and relational symbols $R_1, \ldots$ as being coded by their integer indices, so that all relational schemas SC and all strings of formulae built from these schemas are now objects in $V(S)$.

For what follows, we fix a schema SC. The set $Inst(SC, S)$ belongs to $V(S)$. Hence, we can talk about $^*SC$-databases, or, for short ***-databases**, namely the elements of the set $^*Inst(S, SC)$ for our fixed schema. Given a *-query Q and *-database D, we can (by transfer) apply $^*Q$ to D. We will often refer to $Q(DB)$ rather than $^*Q(DB)$.

Although it is not necessary to formalize logic within the nonstandard universe, we will do so here for completeness of exposition. Uninterpreted logical formulae over SC can be coded by Gödel numbers. Occasionally, we will have a language that is indexed by a set of elements in the nonstandard universe. That is, we will have some internal set I and we will want to talk about a language with constant or relation symbols b_i for each $i \in I$. In order to talk about this within the model, we assume a standard convention in $V(S)$ for making an element of $V(S)$ into a symbol: a formula $\phi(b_i)$ that uses the symbol will be coded as, say, the pair consisting of a code for $\phi(b)$ and the element i. With this convention, for any sets A and I in $V(S)$, the satisfaction predicate $\models$ for formulae built over $\langle b_i, i \in I\rangle$, and for structures with interpretation functions from the b_i into A, lies inside of $V(S)$. Hence $D \ {}^*\!\models \phi$ is well defined whenever A and I are internal sets, ϕ is a (valid code for) a formula over $b_i, i \in I$, and D is a structure with an internal interpretation function from b_i into A. If the interpretation function for D mapping each b_i to a subset of A is *external* but maps each b_i to an internal set, we can still evaluate $D \ {}^*\!\models \phi$, since ϕ only makes use of finitely many b_i, and the restriction of the interpretation to this set is internal (since it is a finite sequence of internal pairings). If the interpretation function for D maps each b_i to an internal set, then we say that D is **internally presented**. If the interpretation function is itself an internal mapping, then we say that D is **internal**.

The preceding details of coding may give the impression that $A \ {}^*\!\models \phi$ is a very difficult notion. However, we will show soon that one can make sense of satisfaction in the nonstandard model without referring to coding at all.

For the rest of this section, all languages L will always be assumed to be built on an internal set I in the nonstandard universe. If a coding for symbols in L is not given explicitly, we assume the nth symbol is coded by the integer n. An L-formulae will always mean a finitary first-order formulae.

We now have two satisfaction relations for a *-structure M in the nonstandard universe: either by using the standard definition of satisfaction (since M is a structure in the usual sense) or by looking at the satisfaction predicate as an element of the superstructure $V(S)$ and considering its *-image. The proposition below establishes the equivalence of these two notions. It can be proved by straightforward induction on logical complexity, using the Tarskian definition of truth and the transfer principle.

Proposition 7 *Let L be a finite language, and let M be an internal L-structure. 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 satisfaction predicate $\models$ on ϕ. That is, if $\vec{c}$ is a finite sequence* *of parameters from M, then $M\ {}^*\!\models \phi(\vec{c})$ iff $M \models \phi(\vec{c})$.*

Thus, we will not distinguish the two kinds of satisfaction predicates when we are dealing with first-order ϕ's.

We now require that our nonstandard universe satisfy the following additional axiom:

4. *(Isomorphism Property)* For any 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 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 hyperfinite sets have the same external cardinality (it is easy to show that for any hyperintegers K and H the structures $[1, K]$ and $[1, H]$ are elementary equivalent in the language of equality). For basic facts about the isomorphism property, and a proof that saturated models with the isomorphism property exist, the reader is referred to [16].

We state another proposition that will be useful.

Proposition 8 *Let L be a first-order language. Let M be an internally presented L-structure, and let $\Gamma(y)$ be a countable collection of L-formulae, possibly with parameters from M. 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 $\langle M_\phi, y\rangle$ satisfies $\phi(y)$. Then each $\phi'(y)$ is a bounded-quantifier formula satisfied in the nonstandard universe, so by countable saturation, there is a y satisfying each $\phi'(y)$. □

The starting point for the use of nonstandard methods is the following proposition. Recall that by a *-database we mean an element of the image under $*$ of the set of databases.

Proposition 9 *Let SC be our schema, and Θ be a finite signature, and Q be any query:*

a) In a nonstandard universe that does not necessarily satisfy the Isomorphism Property, the following is true. Q is expressible over Θ if and only if, every two *-databases M_1 and M_2 that agree on all standard queries over Θ, also agree on Q.

b) In a nonstandard universe satisfying Isomorphism Property, Q is expressible over Θ if and only if, every two *-databases M_1 and M_2 that are isomorphic in the language of Θ, agree on Q.

Proof. a) The *if* direction is trivial. We now prove the contrapositive of the *only if* direction: that is, we show that if Q is *not* expressible over Θ, then there are two *-databases that agree on all Θ queries but disagree on Q.

Let $\phi_1, \phi_2, \ldots$ enumerate the Θ-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_is.

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. Therefore, fix a natural number n, and assume there are no two models that agree on each ϕ_i for each $i \leq n$ but disagree on Q. Then the models of Q are composed of finitely many $=_n$ equivalence classes. But since each equivalence class is definable by a Θ-cbq, this would make Q definable as a Θ-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.

Part b) follows easily from a) and the definition of Isomorphism Property. $\square$

5.3 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 assume Ω to be finite.

Let Q be a counterexample query over our 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.

We now apply Proposition 9 to our counterexample Q, with Θ being $<$. This gives us *-databases M_1 and M_2 that agree on each $<$-query but disagree on Q.

Now consider the active domains of M_1 and M_2. Since these are hyperfinite sets, they have cardinalities H and K respectively, where $H, K \in {}^*\mathbb{N}$. 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. That is, $c_1 < \ldots < c_H$ and the models $(^*\mathbb{U}, {}^*\Omega, s_1)$ and $(^*\mathbb{U}, {}^*\Omega, s_2)$ are elementary equivalent for any two finite sequences of c_is that have the same length and whose elements are ordered similarly. To see that such a sequence exists, we note that for any $n, m \in \mathbb{N}$, there exists a sequence $a_1 < \ldots < a_m \in \mathbb{U}$ indiscernible over the first n formulae $\phi_1, \ldots \phi_n$ of the first-order language of Ω. Indeed, this condition can be expressed by a first order sentence ϕ. By [10], there is an elementary extension of $(\mathbb{U}, \Omega)$ that has an infinite set of indiscernibles. Since this elementary extension satisfies ϕ, it must hold in (U, Ω) as well.

Now the sequence $c_1, \ldots, c_H$ exists by applying saturation to the family of formulae $\Gamma_i(C)$, where $\Gamma_i(C)$ is the formula saying that C is a sequence of length $> i$ such that all elements in C are indiscernible for $\phi_1, \ldots \phi_n$.

Let $d_1, \ldots, d_K$ be an internal 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 *-database M_2' with active domain $d_1, \ldots, d_K$ by applying a different order-preserving mapping to M_2. Since the mappings preserve order, we still have that M_1' and M_2' also agree on every $<$-query whose quantifiers are restricted to the active domains. By the genericity of Q, and transfer, 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 the respective active domains as the domains of both models (that is, the union of the c_i's for M_1', and the union of the d_i's for M_2'). The fact that M_1' and M_2' agree on all $<$-queries with restricted quantification says exactly that these two models are elementary equivalent.

Applying the Isomorphism Property to these models, we get that there is an injective mapping f from $c_1, \ldots, c_H$ onto $d_1, \ldots, d_K$ that is an isomorphism (in the language of $<$ plus the schema relations) of M_1' onto M_2'. Note that f is not necessarily internal.

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, $(^*\mathbb{U}, {}^*\Omega) \models \phi'(c_{j_1}, \ldots, c_{j_n})$, 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, $(^*\mathbb{U}, {}^*\Omega) \models \phi'(c_{m_1}, \ldots, c_{m_n})$ whenever the m_i's are ordered the same as the j_i's. Since the mapping f is order-preserving, and since it maps each c_i to some d_j, we get that $(^*\mathbb{U}, {}^*\Omega) \models \phi'(f(c_{j_1}), \ldots, f(c_{j_n}))$. But this means N_2 satisfies ϕ. $\square$

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

Before we prove Proposition 11, let us show how the theorem follows from it. 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 $(^*\mathbb{U}, {}^*\Omega)$ to L^+ obtained by interpreting the schema relations as in M_1' for P_1, and as in M_2' for P_2. Then we have

Proposition 12 *P_1 is elementary equivalent 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 , 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, as shown below.

It is clear from Proposition 11 that at any point in the game if the two structures P_1 and P_2 are pebbled as $\langle e_1, \ldots, e_n\rangle$ and $\langle g_1, \ldots, g_n\rangle$, respectively, then $\langle e_1, \ldots, e_n\rangle \Rightarrow \langle g_1, \ldots, g_n\rangle$. This implies immediately that the operations in Ω are duplicated correctly at each stage of the game, since L' contains Ω. If one of the e_j's for $j \leq n$ is one of the c_I's, then the definition of the arrow relation ensures that the corresponding g_i is $f(c_I)$. Conversely, if g_j is d_I then e_j must be $f^{-1}(d_I)$. Since f preservers the schema relations, we have that the schema relations are preserved at each stage of the game. $\square$

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 the theorem.

Proof of Proposition 11. 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 $\vec{x}$.

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

Assuming Claim 5, the first part of Proposition 11 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 8, 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 formulae 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 Proposition 11 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 $\Gamma(z)$ 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 $\Gamma(z)$ 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. Since the model $({}^*\mathbb{U}, {}^*\Omega)$ is elementary equivalent to $(\mathbb{U}, \Omega)$ [10], and o-minimality is preserved under elementary equivalence [21, 29], we see that the nonstandard structure is also o-minimal. By o-minimality of $({}^*\mathbb{U}, {}^*\Omega)$, each such formula $\psi(y)$ 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

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

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

Note that max and min above exist by transfer, since these are both hyperfinite sets, and transfer tells us that every hyperfinite linear order has a maximal element.

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. We can then concatenate the parameters needed to define these endpoints together in order to get $\vec{c'}$.

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

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

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

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

We now briefly trace out the proof of Claim 6 to complete the proof of proposition 11. For each formula $\chi(\vec{z}, y)$ in the language of Ω with parameters from $\vec{y}$, let

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

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

Note that the max and min above exist by transfer, since these are both hyperfinite sets.

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. □

5.4 Some Corollaries of Theorem 4

There are three corollaries of Theorem 4 that we would like to mention here. First, looking at the proof of Theorem 4, we can observe the following sufficient condition for verifying when two hyperfinite databases satisfy the same natural semantics queries over the nonstandard universe.

Corollary 2 *Suppose that M is an o-minimal structure, SC a schema, and D_1, D_2 two *database instances of the schema SC. Assume that there exists a mapping f from $adom(D_1)$ onto $adom(D_2)$ such that f preserves the SC relations and for each formula $\psi(\vec{x})$ in the language of M and each vector $\vec{c}$ of elements of $adom(D)$, ${}^*M \models \psi(\vec{c})$ if and only if ${}^*M \models \psi(f(\vec{c}))$. Then D_1 and D_2 agree on all natural semantics queries over ${}^*M$.* □

This observation is made use of in [8] where the coincidence of the active and natural semantics is established in certain o-minimal structures.

Also, theorem 4 implies collapse results for domains other than the real numbers. For example, since $(\mathbb{Q}, +, <)$ is elementary equivalent to $(\mathbb{R}, +, <)$, we obtain the following result of [33] as a corollary.

Corollary 3 $\mathcal{NFO}^{\mathbf{LG}}(\mathbb{Q}, +, <) = \mathcal{NFO}^{\mathbf{LG}}(\mathbb{Q}, <)$. □

The proof technique given here is not restricted to continuous domains, or to o-minimal structures. In fact, we can notice that the result applies to any domain such that Claim 5 holds for any internally presented structure. This was used in [5] to give the following result about the integers.

Corollary 4 **(**see [5]**)** $\mathcal{NFO}^{\mathbf{LG}}(\mathbb{N}, +, <) = \mathcal{NFO}^{\mathbf{LG}}(\mathbb{N}, <)$. □

This result is in contrast to [14] which proved that any computable query is definable in $\mathcal{NFO}(\mathbb{N}, +, *, <, 0, 1)$.

6 Expressiveness of Nonboolean Queries

So far we have only considered boolean queries given by first-order sentences. This was enough to prove some of the desired inexpressibility results. For example, inexpressibility of transitive closure follows from inexpressibility of connectivity test. But how far can we go using our results for the boolean case? In this section we present a simple technique that lifts the results about boolean queries to cover arbitrary queries as well.

To speak about nonboolean queries, we need two schemas: the input schema $SC_{in} = \langle R_1, \ldots, R_k \rangle$, $k > 0$, with relation names for the input database, and the output schema $SC_{out} = \langle T_1, \ldots, T_l \rangle$, $l > 0$, with the names of the output relations. Now nonboolean queries are maps from instances of SC_{in} to SC_{out}. Boolean queries can be viewed as queries with the output schema that consists of a single 0-ary relation.

A **relational query** is given by a first-order formula $\phi(x_1, \ldots, x_n)$ with n free variables for each n-ary relation symbol in SC_{out}; this formula is in the language of SC_{in} and equality. Again, we speak of $<$-relational queries if ϕ is in the language of SC_{in} and the order relation $<$. For a signature Ω, **constraint queries** are given by first-order formulae $\phi(x_1, \ldots, x_n)$ in the language that contains SC_{in} and all the symbols in Ω.

Similarly to the boolean case, queries have both active-domain and natural interpretation. A query (relational or constraint) $(\phi_1, \ldots, \phi_l)$, applied to a SC_{in}-database DB, results in SC_{out}-database DB' whose ith relation, of arity n, consists of all tuples $(c_1, \ldots, c_n) \in \mathbb{U}^n$ such that $DB \models \phi_i(c_1, \ldots, c_n)$[1].

[1] We only consider queries that produce finite results. For the natural semantics case our results generalize easily for infinite outputs.

When we use the active-domain (natural) interpretation for the sentence $\phi_i(c_1, \ldots, c_n)$, we obtain the class of queries definable under the active-domain (natural) interpretation. These classes will be denoted by $\mathcal{AFO}[SC_{in} \to SC_{out}](\mathbb{U}, \Omega)$ and $\mathcal{NFO}[SC_{in} \to SC_{out}](\mathbb{U}, \Omega)$ respectively.

The notions of genericity are generalized for nonboolean queries straightforwardly. Instead of saying $Q(\varphi DB) = Q(DB)$, as we did for boolean queries, we define generic queries as those satisfying $Q(\varphi DB) = \varphi(Q(DB))$ for a given class of maps. More precisely, for a class $\mathcal{F}$ of partial injective endofunctions on $\mathbb{U}$, we say that Q is **$\mathcal{F}$-generic** if, for any database DB and any $\varphi \in \mathcal{F}$ defined on $adom(DB) \cup adom(Q(DB))$, it is the case that $Q(\varphi DB) = \varphi(Q(DB))$. Total genericity and local genericity are examples of $\mathcal{F}$-genericity for the classes of injective partial maps and monotone partial maps. We denote classes of $\mathcal{F}$-generic queries in $\mathcal{AFO}[SC_{in} \to SC_{out}](\mathbb{U}, \Omega)$ and $\mathcal{NFO}[SC_{in} \to SC_{out}](\mathbb{U}, \Omega)$ by $\mathcal{AFO}^{\mathcal{F}}[SC_{in} \to SC_{out}](\mathbb{U}, \Omega)$ and $\mathcal{NFO}^{\mathcal{F}}[SC_{in} \to SC_{out}](\mathbb{U}, \Omega)$ respectively.

Now we are ready to prove the main result of this section saying that a collapse result for boolean queries implies a similar result for arbitrary queries. We say that $\mathcal{F}$ is **adom-preserving** if for any $\mathcal{F}$-generic query Q and any database DB, $adom(Q(DB)) \subseteq adom(DB)$. We say that $\mathcal{F}$ is **extensible** if for any $\mathcal{F}$-generic Q, any database DB and any function φ in $\mathcal{F}$ whose domain is $adom(DB)$, we can find an extension of φ to $\varphi' \in \mathcal{F}$ whose domain is $adom(DB) \cup adom(Q(DB))$. These two notions impose very mild restrictions on the classes of generic queries. The class of all partial maps is extensible on any infinite set, and the class of all monotone partial maps is extensible on any ordered set without endpoints. Also, any adom-preserving $\mathcal{F}$ is extensible. Furthermore,

Proposition 13 *Totally generic and locally generic queries do not extend active domains of their inputs.*

Proof. We prove the local genericity case; the simpler proof for TG queries is omitted. We use the notation $adom(DB, Q)$ for $adom(DB) \cup adom(Q(DB))$. First, observe the following. For any finite set $Y \subset \mathbb{U}$ and any $x \notin Y$, and any number n, we can find maps $\pi_1, \ldots, \pi_n$ in $\mathcal{F}$ such that for all i, j: $\pi_i(Y) = \pi_j(Y)$, but all $\pi_1(x), \ldots, \pi_n(x)$ are distinct. This is true since $\mathbb{U}$ is infinite, and hence it either has a dense subset or does not have either right or left endpoint; in both situations we easily construct the π_is.

Now fix a counterexample: $x \in adom(Q(DB)) - adom(DB)$ for a LG query Q. Let $X = adom(Q(DB))$. Let $Y = adom(DB)$ and let $n = card(X) + 1$. Construct $\pi_1, \ldots, \pi_n$ as above. Now for any i, j: $\pi_i(Q(DB)) = Q(\pi_i(DB)) = Q(\pi_j(DB)) = \pi_j(Q(DB))$; hence $\pi_1(X) = \ldots = \pi_n(X)$. In particular, every $\pi_i(x) \in \pi_1(X)$, whence $card(\pi_1(X)) > card(X)$. This contradiction proves the case. $\square$

If Ω and Θ are two signatures on $\mathbb{U}$, we say that there is a (Ω, Θ) *Boolean collapse of $\mathcal{F}$-generic natural queries* if $\mathcal{NFO}^{\mathcal{F}}(\mathbb{U}, \Omega) = \mathcal{NFO}^{\mathcal{F}}(\mathbb{U}, \Theta)$ holds for any schema SC. We say that there is a (Ω, Θ) *complete collapse of $\mathcal{F}$-generic natural queries* if $\mathcal{NFO}^{\mathcal{F}}[SC_{in} \to SC_{out}](\mathbb{U}, \Omega) = \mathcal{NFO}^{\mathcal{F}}[SC_{in} \to SC_{out}](\mathbb{U}, \Theta)$ holds for any input schema SC_{in} and output schema SC_{out}. We similarly define the collapse of the active queries, replacing $\mathcal{NFO}$ with $\mathcal{AFO}$.

Theorem 5 *Let $\mathcal{F}$ be an extensible class of partial endomaps on $\mathbb{U}$, and Ω and Θ two signatures on*

$\mathbb{U}$. *Then:*

1. *If there is a* (Ω, Θ) *Boolean collapse of* $\mathcal{F}$*-generic natural queries, then there is a* (Ω, Θ) *complete collapse of* $\mathcal{F}$*-generic natural queries.*

2. *If* $\mathcal{F}$ *is adom-preserving and there is a* (Ω, Θ) *Boolean collapse of* $\mathcal{F}$*-generic active queries, then there is a* (Ω, Θ) *complete collapse of* $\mathcal{F}$*-generic active queries.*

Proof. It is enough to prove the theorem for just one n-ary relation in the output schema. Assume that a $\mathcal{F}$-generic query Q is given by a formula $\phi(x_1, \ldots, x_n)$; that is, $(c_1, \ldots, c_n) \in Q(DB)$ iff $DB \models \phi(c_1, \ldots, c_n)$. Now we extend SC_{in} to a new schema SC by adding n unary relational symbols $S_1, \ldots, S_n$ that are not present in $SC_{in} \cup SC_{out}$. Define a boolean constraint query Q_ϕ as follows:

$$\bigwedge_{i=1}^{n}((\exists x.S_i(x)) \wedge (\forall x \forall y.(S_i(x) \wedge S_i(y) \rightarrow x = y))) \wedge (\forall x_1 \ldots \forall x_n.(S_1(x_1) \wedge \ldots \wedge S_n(x_n)) \rightarrow \phi(x_1, \ldots, x_n))$$

That is, Q_ϕ says that in a SC database all S_is are singletons and ϕ is satisfied on their elements.

Next, we claim that Q_ϕ is $\mathcal{F}$-generic. The proof given below works for both active and natural cases. Let DB be a SC-database and φ a map in $\mathcal{F}$ defined on $adom(DB)$. We must show that $DB \models Q_\phi$ iff $\varphi DB \models Q_\phi$. It follows from the definition of Q_ϕ that $\varphi DB \models Q_\phi$ iff the following conditions hold:

(1) For every $i = 1, \ldots, n$, φS_i is a singleton $\{b_i\}$.
(2) $\varphi DB \models \phi(b_1, \ldots, b_n)$.

Using extensibility, we find a map $\varphi' \in \mathcal{F}$ that extends φ to $adom(DB) \cup adom(Q(DB))$. Now every b_i is in the image of φ (and φ'); we let a_i be the element of $adom(DB)$ that is mapped to b_i. Such an element is unique by injectivity. Now (2) is equivalent to $(b_1, \ldots, b_n) \in Q(\varphi DB) = Q(\varphi' DB) = \varphi'(Q(DB))$, which in turn is equivalent to $(a_1, \ldots, a_n) \in Q(DB)$ or $DB \models \phi(a_1, \ldots, a_n)$. We also see that (1) is equivalent to

(1$'$) In DB, for every $i = 1, \ldots, n$, S_i is a singleton $\{a_i\}$.

Since (1$'$) and $DB \models \phi(a_1, \ldots, a_n)$ hold iff $DB \models Q_\phi$, we conclude the proof of $\mathcal{F}$-genericity of Q_ϕ. Note that this proof works for both active and natural interpretations. (In fact, all quantification in Q_ϕ other than that inside ϕ is bounded to the active domain.)

Now we prove 1. Since Q_ϕ is in $\mathcal{AFO}^{\mathcal{F}}(\mathbb{U}, \Omega)$, we can find a query Q' in $\mathcal{AFO}^{\mathcal{F}}(\mathbb{U}, \Theta)$ equivalent to Q_ϕ. That is, Q' is a sentence in the language of SC and Θ such that $Q_\phi(DB) = Q'(DB)$ for any $DB \in \mathit{Inst}(\mathbb{U}, SC)$. Assume that $y_1, \ldots, y_n$ are variables not mentioned by Q'. We then define a formula $\psi(y_1, \ldots, y_n)$ in the language of SC_{in} and Θ by replacing each subformula of Q' of the form $S_i(z)$ by $z = y_i$.

We now claim that for every $\vec{c} = (c_1, \ldots, c_n) \in \mathbb{U}^n$ and any SC_{in} database DB, $DB \models \phi(\vec{c})$ iff $DB \models \psi(\vec{c})$ (under the natural interpretation); this clearly implies 1. To see this, we define $DB[\vec{c}]$ as a SC database in which all SC_{in} relations are interpreted as in DB, and each S_i is interpreted as $\{c_i\}$. Then we obtain the following equivalences, since ϕ and ψ do not mention any symbols S_i:

$$\begin{array}{rl} & DB \models \psi(\vec{c}) \\ \Leftrightarrow & DB[\vec{c}] \models \psi(\vec{c}) \\ \Leftrightarrow & DB[\vec{c}] \models Q' \\ \Leftrightarrow & DB[\vec{c}] \models Q_\phi \\ \Leftrightarrow & DB[\vec{c}] \models \phi(\vec{c}) \\ \Leftrightarrow & DB \models \phi(\vec{c}) \end{array}$$

To prove 2, we define ψ as before and observe that the proof above shows that $DB \models \phi(\vec{c})$ iff $DB \models \psi(\vec{c})$ for any $\vec{c} \in adom(DB)^n$. Indeed, to prove the first and the last equivalences, observe that for formulae not mentioning any of the S_is, the only difference in terms of satisfaction on DB and $DB[\vec{c}]$ is the range of quantification. For $\vec{c} \in adom(DB)^n$, $adom(DB[\vec{c}]) = adom(DB)$, so all quantified variables range over the same set.

Let $\mathsf{adom}[SC_{in}](x)$ be a formula in the language of SC_{in} such that $DB \models \mathsf{adom}[SC_{in}](c)$ iff $c \in adom(DB)$; such a formula exists, see [1]. Then, using the fact that $\mathcal{F}$ is adom-preserving, we finish the proof of 2 as follows. For any $\vec{c} \in \mathbb{U}^n$,

$$\begin{array}{rlll} & \vec{c} \in Q(DB) & & \\ \Leftrightarrow & \vec{c} \in adom(DB)^n & \text{and} & \vec{c} \in Q(DB) \\ \Leftrightarrow & \vec{c} \in adom(DB)^n & \text{and} & DB \models \phi(\vec{c}) \\ \Leftrightarrow & \vec{c} \in adom(DB)^n & \text{and} & DB \models \psi(\vec{c}) \\ \Leftrightarrow & \multicolumn{3}{l}{DB \models \psi(\vec{c}) \wedge \mathsf{adom}[SC_{in}](c_1) \wedge \ldots \wedge \mathsf{adom}[SC_{in}](c_n).} \end{array}$$

Thus, the query given by ϕ can be expressed by a formula in the language of SC_{in}, Θ and equality. The proof is complete. □

Now we immediately obtain the following collapse results for nonboolean queries.

Corollary 5 *1) Under the active semantics, for any signature Ω on ordered $\mathbb{U}$, every locally generic constraint query is equivalent to a $<$-relational query.*
2) Under the active semantics, for any sparse signature Ω on $\mathbb{R}$, every totally generic constraint query is equivalent to a relational query.
3) Under the natural semantics, for any o-minimal signature Ω on ordered $\mathbb{U}$, every locally generic constraint query is equivalent to a $<$-relational query. □

7 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.

$$
\begin{tikzcd}
\mathcal{AFO}^{\mathbf{TG}}(\Omega_{sparse}) \arrow[rr, hook] \arrow[d, equal] & & \mathcal{NFO}^{\mathbf{LG}}(\Omega_{o\text{-}min}, <) \arrow[r, equal] \arrow[d, equal] & \mathcal{AFO}^{\mathbf{LG}}(\Omega, <) \arrow[d, equal] \\
\mathcal{AFO}^{\mathbf{TG}}(+,-,*,0,1) \arrow[r, hook] \arrow[d, equal] & \mathcal{NFO}^{\mathbf{TG}}(+,*,0,1) \arrow[r, hook] \arrow[d, equal] & \mathcal{NFO}^{\mathbf{LG}}(+,*,0,1) \arrow[r, equal] \arrow[d, equal] & \mathcal{AFO}^{\mathbf{LG}}(+,-,*,0,1,<) \arrow[d, equal] \\
\mathcal{AFO}^{\mathbf{TG}}(+,-,0,1) \arrow[r, hook] \arrow[d, equal] & \mathcal{NFO}^{\mathbf{TG}}(+,-,0,1,<) \arrow[r, hook] & \mathcal{NFO}^{\mathbf{LG}}(+,-,0,1,<) \arrow[r, equal, "{[27]}"] \arrow[d, equal] & \mathcal{AFO}^{\mathbf{LG}}(+,-,0,1,<) \arrow[d, equal] \\
\mathcal{AFO} \arrow[r, equal, "{[17]}"] & \mathcal{NFO}^{\mathbf{TG}} \arrow[r, hook, "{[1]}"] \arrow[u, hook] & \mathcal{NFO}^{\mathbf{LG}}(<) \arrow[r, equal] & \mathcal{AFO}(<)
\end{tikzcd}
$$

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

Theorem 6 *Let $\Omega_{o\text{-}min}$ be an o-minimal signature on $\mathbb{U}$. Then $\mathcal{NFO}^{\mathbf{LG}}(\Omega_{o\text{-}min}, <) = \mathcal{AFO}(<)$.*

Proof. Note that the $\mathcal{AFO}(<) \subseteq \mathcal{NFO}^{\mathbf{LG}}(\Omega_{o\text{-}min}, <)$ inclusion is trivial. For the reverse inclusion, we first give a simple proof for the case when $\mathbb{U} = \mathbb{R}$, and then outline the general proof. We have:

$$
\begin{array}{lll}
 & \mathcal{NFO}^{\mathbf{LG}}(\mathbb{R}, \Omega_{o\text{-}min}, <) & \\
= & \mathcal{NFO}^{\mathbf{LG}}(\mathbb{R}, <) & \text{by Theorem 4} \\
\subseteq & \mathcal{AFO}^{\mathbf{LG}}(\mathbb{R}, +, -, 0, 1, <) & \text{by Fact 2} \\
= & \mathcal{AFO}(\mathbb{R}, <) & \text{by Theorem 1}
\end{array}
$$

When $\mathbb{U}$ is arbitrary, let Q be in $\mathcal{NFO}^{\mathbf{LG}}(\mathbb{U}, \Omega_{o\text{-}min}, <)$. By Theorem 4 it is in $\mathcal{NFO}^{\mathbf{LG}}(\mathbb{U}, <)$. Since $(\mathbb{U}, <)$ is o-minimal, we can find a definitional expansion to $(\mathbb{U}, <, \Theta)$ that admits quantifier elimination. By [8], we obtain $Q \in \mathcal{AFO}^{\mathbf{LG}}(\mathbb{U}, <, \Theta)$, and then Q is in $\mathcal{AFO}(\mathbb{U}, <)$ by Theorem 1. $\Box$

Corollary 6 *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 $(+, -, *, e^x, 0)$.* $\Box$

In conclusion, we have settled the open problem of whether parity and connectivity can be expressed in the relational calculus with arithmetic constraints. 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 $=$ edge 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 schema contain two unary relation symbols R_1 and R_2 and let Q be $\forall x \forall y.(R_1(x) \wedge R_2(y)) \to x < y$. 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 Martin Otto, Alex Stolboushkin, Jianwen Su, Michael Taitslin, Jan Van den Bussche and Dirk Van Gucht for discussions, encouragements, and references, and anonymous referees for suggesting numerous improvements.

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 Principles and Practice of Constraint Programming*, Springer Verlag, 1994.

[3] F. Afrati, T. Andronikos and T. Kavalieros. On the expressiveness of first-order constraint languages. *Proceedings of the First Workshop on Constraint Databases and their Applications*, Springer Verlag, 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] O.V. Belegradek, A.P. Stolboushkin and M.A. Taitslin. On order-generic queries. DIMACS Technical Report 95-56, December 1995.

[6] M. Benedikt and H.J. Keisler. On the expressive power of unary counters. In *Proc. Internat. Conf. on Database Theory*, Springer LNCS vol. 1186, 1997, pages 291–305.

[7] M. Benedikt, G. Dong, L. Libkin and L. Wong. Relational expressive power of constraint query languages. In *Proceedings of 15th ACM Symposium on Principles of Database Systems*, 1996, pages 5–16.

[8] M. Benedikt and L. Libkin. On the structure of queries in constraint query languages. In *Proceedings of IEEE Symposium on Logic in Computer Science*, 1996, pages 25–34.

[9] A. Chandra and D. Harel. Computable queries for relational databases. *Journal of Computer and System Sciences*, 21(2):156–178, 1980.

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

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

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

[13] 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.

[14] S. Grumbach and J. Su. First-order definability over constraint databases. *Proc. of Principles and Practice of Constraint Programming*, pages 121–136, Springer Verlag, 1995.

[15] S. Grumbach, J. Su, and C. Tollu. Linear constraint databases. In *Proceedings of Logic and Computational Complexity*, pages 426–446, Springer Verlag, 1994.

[16] 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.

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

[18] R. Hull and C.K. Yap. The format model, a theory of database organization. *Journal of the ACM* 31(1984), 518–537.

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

[20] P. Kanellakis, G. Kuper, and P. Revesz. Constraint query languages. *Journal of Computer and System Sciences*, 51 (1995), 26–52. Extended abstract in *Proceedings of 9th ACM Symposium on Principles of Database Systems, Nashville, Tennessee*, pages 299–313, May 1990.

[21] J.F. Knight, A. Pillay and C.I. Steinhorn. Definable sets in ordered structures II. *Trans. Amer. Math. Soc.* 295 (1986), 593–605.

[22] 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.

[23] D. Marker. Model theory and exponentiation. *Notices of the AMS*, 43 (1996), 753–759.

[24] R. Narasimhan. *Several Complex Variables*, University of Chicago Press, 1971.

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

[26] 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.

[27] 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. Full paper to appear in *SIAM J. Comput.*

[28] A. Pillay, C. Steinhorn. Definable sets in ordered structures. *Bulletin of the AMS* 11 (1984), 159–162.

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

[30] 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.

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

[32] P. Speissinger. Order minimality of the Gamma function. Personal communication. 1996.

[33] A.P. Stolboushkin and M.A. Taitslin. Linear vs. order constraint queries over rational databases. In *Proceedings of 15th ACM Symposium on Principles of Database Systems*, 1996, pages 17–27.

[34] A. Tarski. *A Decision Method for Elementary Algebra and Geometry.* University of California Press, 1951.

[35] L. Van den Dries. Alfred Tarski's elimination theory for real closed fields. *Journal of Symbolic Logic* 53 (1988), 7–19.

[36] 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.

[37] A.J. Wilkie. Model completeness results for expansions of the ordered field of real numbers by restricted Pfaffian functions and the exponential function. *J. Amer. Math. Soc.* 9 (1996), 1051–1094.