On the Structure of Queries in Constraint Query Languages

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

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

Abstract

We study the structure of first-order and second-order queries over constraint databases. Constraint databases are formally modeled as finite relational structures embedded in some fixed infinite structure. We concentrate on problems of elimination of constraints, reducing quantification range to the active domain of the database and obtaining new complexity bounds. We show that for a large class of signatures, including real arithmetic constraints, unbounded quantification can be eliminated. That is, one can transform a sentence containing unrestricted quantification over the infinite universe to get an equivalent sentence in which quantifiers range over the finite relational structure. We use this result to get a new complexity upper bound on the evaluation of real arithmetic constraints. We also expand upon techniques in [21] and [4] for getting upper bounds on the expressiveness of constraint query languages, and apply it to a number of first-order and second-order query languages.

1. Introduction

Techniques of finite model theory have found applications in a number of areas such as database theory [1] and descriptive complexity [18]. Database applications of finite model theory stem from one of the basic results of relational database theory: Classical query languages, such as relational algebra, have precisely the power of first-order logic. Since relational databases can be viewed as finite models in the language of the relational schema, this basic observation allows us to apply the tools of finite model theory to study expressibility of relational query languages. Early work in that direction includes [2, 8, 11]; for a survey see [1].

In recent years, various extensions of the basic relational model have been studied. Two most notable ones are extensions to complex objects, or nested relations [27], which underlie most object-oriented datamodels, and extensions to *constraint databases* [19, 20], which are used as the basis for geographical and temporal datamodels. For nested relations most basic questions about expressive power and the structure of queries have been answered (see [27] and references therein), but only very recently has some progress been made for constraint databases.

The framework of constraint databases assumes some underlying model $\mathcal{M} = \langle \mathbb{U}, \Omega \rangle$ where $\mathbb{U}$ is a set (always assumed to be infinite in this paper), and Ω is a signature that consists of a number of interpreted functions and predicates over $\mathbb{U}$. For instance, the domain very often considered for geographical databases is $\langle \mathbb{R}, +, *, 0, 1, < \rangle$, with the intention that databases represent some regions on the real plane. In the classical framework of [20], (generalized) databases over $\mathcal{M}$ are given by quantifier-free formulae $\varphi(x_1, \ldots, x_n)$ in the language of Ω; such a database represents the set

$$\mathcal{M}_\varphi \;=\; \{\vec{a} = (a_1, \ldots, a_n) \mid \vec{a} \in \mathbb{U}^n,\ \mathcal{M} \models \varphi(\vec{a})\}.$$

For example, a convex polygon with known set of vertices can easily be represented in such a way. The model $\mathcal{M}$ is typically chosen to admit quantifier elimination. Then the query evaluation process reduces to application of the quantifier elimination procedure [20].

Recently, attention has shifted from finitely represented models (that is, those that arise as $\mathcal{M}_\varphi$) to finite ones, cf. [4, 21, 23, 24]. Since queries arising in geographical applications often involve regions that are determined by a fixed finite number of points (i.e. a convex polygon can be given by its vertices), we can convert most interesting questions involving finitely-represented models to questions involving finite models [22]. For example, it was conjectured that first-order logic with polynomial inequality constraints cannot express topological connectivity. A result of [23] reduced that problem to connectivity of finite graphs whose

nodes come from $\mathbb{R}$, and the problem for graph connectivity was recently solved in [4]. In addition, the setting of finite databases embedded in a fixed infinite structure enables one to study constraint databases via the tools of finite (and infinite) model theory, and allows for helpful characterizations of the expressive power of classes of queries even in settings where quantifier elimination does not hold in the underlying structure.

We will work in the setting of finite databases embedded in infinite fixed structures. We start with the underlying model $\mathcal{M} = \langle \mathbb{U}, \Omega \rangle$ and add a number of predicate symbols $R_1, \ldots, R_k$, R_i being of arity τ_i, for finite database relations. These will be interpreted as τ_i-ary relations over $\mathbb{U}$. Following the database tradition, we will call $R_1, \ldots, R_k$ a *schema*, and denote it by SC. As our main language we take $\mathcal{FO}(\mathcal{M}, SC)$, the first-order logic over the language $L(SC, \Omega)$ that contains Ω and SC. If SC is understood, we often omit it. We will mostly deal with sentences, since most of the results for sentences can be extended (as we'll show) for arbitrary formulae.

Let $Inst(\mathbb{U}, SC)$ be the set of k-tuples of finite relations over $\mathbb{U}$, the ith one being of arity τ_i (that is, the set of possible finite instances of SC over $\mathbb{U}$). For $D \in Inst(\mathbb{U}, SC)$ and φ a $L(\mathcal{M}, SC)$ sentence, we define $D \models \varphi$ in the usual way.

Although the syntax of our formulas can be straightforwardly adapted from classical predicate logic, the properties of queries can depend in subtle ways on the domain of quantification. For $D \in Inst(\mathbb{U}, SC)$, the minimum possible range of quantification is the *active domain* of D: $adom(D)$ is the set of all elements of $\mathbb{U}$ that occur in relations in D, cf. [1, 17]. For $A \subseteq \mathbb{U}$, we write $D \models_A \varphi$ if $D \models \varphi'$ where φ' is obtained from φ by replacing each quantifier $\mathrm{Q}x$ by its bounded version $\mathrm{Q}x \in adom(D) \cup A$.

There are two cases of the $\models_A$ relations that are of special interest: $\models_{\mathbb{U}}$ is the usual relation $\models$ (which is sometimes called the *natural interpretation* of queries), and $\models_\emptyset$ restricts quantification to the active domain of the finite D (this is sometimes called the *active*, or *active-domain interpretation* of queries). To see the difference between the two, assume that $\mathcal{M} = \langle \mathbb{R}, < \rangle$, and φ is $\exists x \forall y.(x < y) \vee (x = y)$. Then, for any nonempty $D \in Inst(\mathbb{R}, SC)$, $D \models \neg\varphi$ but $D \models_\emptyset \varphi$.

Much of what has been done in constraint databases addresses the problem of evaluating constraint queries; that is, queries in $\mathcal{FO}(\mathcal{M}, SC)$, or in a language based on another logic (e.g. fixpoint logic). To find satisfactory query evaluation algorithms for a query language, we have to address the following issues.

$\angle$ *Expressive Power.* Classical relational query languages have been studied in great depth, and their expressive power is well known. Prior results allow one to infer what sorts of recursion constructs are necessary to express properties such as parity, connectivity and others that arise in database applications. We also know much about the impact of adding these programming constructs on query optimization. For constraint query languages, an understanding of many of the fundamental expressivity questions is still lacking. In particular, there is a need for tools to assist in getting upper bounds for the expressivity of constraint languages. In this paper we continue the work of [4, 21, 24] in getting techniques to bound the expressivity of constraint query languages. We show how to extend the results of [4] to show equivalence in expressive power for many first-order and second-order constraint query languages. We prove several kinds of *collapse results*, which say that adding new predicates or functions to the signature Ω does not significantly increase expressive power.

$\angle$ *Range of Quantification.* While databases themselves are finite, the natural range of quantification for constraint queries is the whole universe $\mathbb{U}$. Thus, we need tools to reduce the problem of query evaluation to a finitary process. By choosing $\mathcal{M}$ to be decidable, we guarantee the ability to evaluate constraint queries for a fixed database, since given a query φ, we can replace each occurrence of $R_i(\vec{x})$ in φ by $\vec{x} = t_1 \vee \ldots \vee \vec{x} = t_m$ where $R_i = \{t_1, \ldots, t_m\}$ and apply the decision procedure to the resulting formula. However, this still forbids us from doing important compile-time query optimizations that are possible in the classical database setting. In particular, we lack the ability to reorder quantifiers based on the range of quantification, or to evaluate expressions 'bottom-up' by retrieving stored values of subexpressions. One possible solution seems to be this: try to show that unbounded quantification can be eliminated. This is equivalent to showing that every query φ has an equivalent one under the active interpretation. That is, there is a ψ such that $D \models \varphi$ iff $D \models_\emptyset \psi$. We will prove that this is possible for many constraint query languages of interest, and that it holds for the real ordered field.

$\angle$ *Complexity of Constraints.* There are many fundamental questions to be answered about the complexity of query evaluation for constraint query languages. Since the relational algebra and calculus are equivalent to pure first-order logic, they have AC^0 complexity [1]. Adding constraints increases this complexity. For instance, if multiplication is in the signature, the AC^0

complexity bound is lost, cf. [7]. As an upper bound for complexity, it is known that if $\mathcal{M} = \langle \mathbb{R}, +, *, 0, 1, < \rangle$, then data complexity of first-order queries is NC, see [20, 6]. Since $AC^0 \subset NC$, one could hope for more precise information about the complexity of constraint queries over the real field. We would also like to know something about the effect of adding other interpreted structure on these complexity bounds, both for first-order and higher-order logics. In this paper we will use results on equivalence of signatures and on bounding quantification to get tighter bounds on query evaluation for the real field, and to get complexity bounds for a variety of other first-order and second-order languages. One tool for doing this will be *partial collapse results*: results that show that a certain set of operations in the signature can be eliminated, assuming that our databases have all their elements coming from a certain infinite set. Using these results, we will be able to get tighter complexity and expressive bounds for queries that are *generic* [1] (those invariant under certain endomaps on $\mathbb{U}$), since their behavior over any infinite set determines their behavior globally.

Organization and quick summary In this paper we offer a detailed study of the structure of constraint queries, addressing the three issues described above.

In section 2 we introduce notation and a new notion of the 'approximate collapse' arrow relation $\mathcal{FO}(\mathcal{M}) \Rightarrow \mathcal{FO}(\mathcal{M}')$, for two models $\mathcal{M}$ and $\mathcal{M}'$ on the same set $\mathbb{U}$, meaning (informally) this: for every query φ in $\mathcal{FO}(\mathcal{M})$, one can find an infinite set $X \subseteq \mathbb{U}$ and a query ψ in $\mathcal{FO}(\mathcal{M}')$ such that for every $D \in \mathit{Inst}(\mathbb{U}, SC)$ with $\mathit{adom}(D) \subset X$, $D \models_X \varphi$ iff $D \models_X \psi$. We are interested in the case when $\mathcal{M}'$ is a reduct of $\mathcal{M}$. The arrow relation shows that we have a means for reducing expressivity and complexity questions about $\mathcal{M}$ to ones concerning $\mathcal{M}'$, a tool we will use later on in the paper.

In section 3 we study the ability to eliminate unbounded quantification in queries in favor of quantification bounded by the active domain. Our main result is Theorem 1, which shows that unbounded quantification can be removed for all models that admit quantifier elimination and satisfy the condition of *o-minimality* [25]. This class includes both cases for which the elimination result is known [17, 24], and also the important case of $\langle \mathbb{R}, +, *, 0, 1, < \rangle$, thus solving the open problem from [24].

In section 4 we state approximate collapse results for query languages, along the lines of [4, 21, 24]: we show that one can get an infinite set on which all constraints in a query can be reduced to constraints in smaller languages. We prove such results for first-order logic and for fragments of second-order logic, and show how they can be used to get expressivity bounds for first- and second-order queries.

In section 5, we apply the results of section 3 to prove a TC^0 complexity bound for first-order logic with polynomial constraints, thus improving the NC bound of [20]. We also establish some complexity bounds for active-domain second-order constraint queries.

In section 6 we show that collapse results and bounded quantification results can always be extended from boolean queries to nonboolean queries (that is, from sentences to arbitrary formulae). We apply these results to get expressivity and complexity bounds for nonboolean queries.

Section 7 contains concluding remarks.

All proofs can be found in [5].

2. Notations

Assume that the domain is an infinite set $\mathbb{U}$. A schema is a nonempty collection $SC = \langle R_1, \ldots, R_k \rangle$ of relation names, R_i being of arity τ_i. A *database instance* D of schema SC is given by an interpretation of each relational symbol R_i as a finite τ_i-ary relation over $\mathbb{U}$. The set of all instances is denoted by $\mathit{Inst}(\mathbb{U}, SC)$. The *active domain* of D, $\mathit{adom}(D)$ is the set of all elements in $\mathbb{U}$ that are in relations in D.

Let Ω be a signature, that is, a collection of interpreted functions and predicates on $\mathbb{U}$. The language that contains the schema predicates, equality and the symbols in Ω is denoted by $L(SC, \Omega)$[1]. A *boolean query* is a first-order sentence in $L(SC, \Omega)$. That is, it is built up from atomic formulae via the usual logical connectives and quantifiers of the form $\forall x$ and $\exists x$.

Let A be a subset of $\mathbb{U}$. Under the *A-interpretation* of queries, we assume that for every D the quantifiers range over $A \cup \mathit{adom}(D)$. That is, the $\emptyset$-interpretation is the active domain interpretation, and the $\mathbb{U}$-interpretation is the natural interpretation. We write $D \models_A \varphi$ to mean that φ is satisfied by D under the A-interpretation.

The class of Boolean queries (maps from instances of schema SC to $\{\mathbf{T}, \mathbf{F}\}$) under the A-interpretation is denoted by $\mathcal{FO}^A(\mathbb{U}, \Omega, SC)$. If $A = \mathbb{U}$ we omit it, i.e. we use $\mathcal{FO}(\mathbb{U}, \Omega, SC)$ for $\mathcal{FO}^{\mathbb{U}}(\mathbb{U}, \Omega, SC)$ and $\models$ for $\models_{\mathbb{U}}$. We write $\mathcal{FO}^A(\mathbb{U}, \Omega)\{=, \subseteq\}\mathcal{FO}^A(\mathbb{U}, \Theta)$ to mean that $\mathcal{FO}^A(\mathbb{U}, \Omega, SC)\{=, \subseteq\}\mathcal{FO}^A(\mathbb{U}, \Theta, SC)$ holds for any schema SC.

The kind of unbounded-quantifier elimination result we are interested in can be written as $\mathcal{FO}(\mathbb{U}, \Omega) = \mathcal{FO}^{\emptyset}(\mathbb{U}, \Omega)$. Two such results are known:

[1] All languages we consider are assumed to have equality, so we will not mention this explicitly any more.

Fact 1 (see [17, 24]) $\mathcal{FO}(\mathbb{U},\emptyset) = \mathcal{FO}^{\emptyset}(\mathbb{U},\emptyset)$; $\mathcal{FO}(\mathbb{R},+,-,0,1,<) = \mathcal{FO}^{\emptyset}(\mathbb{R},+,-,0,1,<)$. □

In section 3 we will extend these results.

We are also interested in equivalence of constraint languages, which in our notation can be written as $\mathcal{FO}(\mathbb{U},\Omega) = \mathcal{FO}(\mathbb{U},\Theta)$ where generally Θ is "simpler" than Ω.These kinds of results are hard to achieve, and often we can only find a certain approximation to equality. For this, we need the following notation.

Definition 1 *We write*

$$\mathcal{FO}^{A}(\mathbb{U},\Omega,SC) \Rightarrow \mathcal{FO}^{B}(\mathbb{U},\Theta,SC)$$

if, for every $L(SC,\Omega)$ sentence φ, we can find an infinite set $X \subseteq \mathbb{U}$ and a $L(SC,\Theta)$ sentence ψ such that for any D with $adom(D) \subseteq X$,

$$D \models_{A\cap X} \varphi \quad \textit{iff} \quad D \models_{B\cap X} \psi.$$

We write $\mathcal{FO}^{A}(\mathbb{U},\Omega) \Rightarrow \mathcal{FO}^{B}(\mathbb{U},\Theta)$ if $\mathcal{FO}^{A}(\mathbb{U},\Omega,SC) \Rightarrow \mathcal{FO}^{B}(\mathbb{U},\Theta,SC)$ holds for any schema SC.

The relation $\Rightarrow$ is an approximation to inclusion; it is an approximation in the sense that it is the inclusion restricted to models from a certain infinite set. We also further restrict quantification to that set X. For instance, if $A = B = \mathbb{U}$ (and this is the situation we encounter most often), then $\mathcal{FO}(\mathbb{U},\Omega) \Rightarrow \mathcal{FO}(\mathbb{U},\Theta)$ means that there is an infinite set X such that $\mathcal{FO}(X,\Omega) \subseteq \mathcal{FO}(X,\Theta)$, i.e. we have the inclusion for the natural interpretation over an infinite X.

Thus, the desired collapse results would be $\mathcal{FO}(\mathbb{U},\Omega) \Rightarrow \mathcal{FO}^{\emptyset}(\mathbb{U},\Omega)$, or $\mathcal{FO}(\mathbb{U},\Omega) \Rightarrow \mathcal{FO}(\mathbb{U},\Theta)$, when X happens to be $\mathbb{U}$.

We can analogously define similar arrow notation for nonboolean queries. The framework of nonboolean queries assumes the *output schema* $SC' = \{T_1,\ldots,T_l\}$, $l > 0$. Then a first-order query is given by a formula $\varphi(x_1,\ldots,x_n)$, for each n-ary output relation. Such a formula defines the relation (under A-interpretation) given by $\{\vec{a} \in \mathbb{U}^n \mid D \models_A \varphi(\vec{a})\}$ for each input $D \in Inst(\mathbb{U},SC)$. We denote the class of nonboolean queries with input schema SC and output schema SC' by $\mathcal{FFO}_{SC'}(\mathbb{U},\Omega,SC)$. (Here $\mathcal{FFO}$ stands for "full first-order", as opposed to sentences only.)

The definition of the arrow relation generalizes straightforwardly. That is, we must say $D \models_{A\cap X} \varphi(\vec{a})$ iff $D \models_{B\cap X} \psi(\vec{a})$ for any $\vec{a} \in (adom(D)\cup X)^n$. Finally, we write

$$\mathcal{FFO}(\mathbb{U},\Omega)\ \{\Rightarrow,=\}\ \mathcal{FFO}(\mathbb{U},\Theta)$$

if $\mathcal{FFO}_{SC'}(\mathbb{U},\Omega,SC) \Rightarrow \mathcal{FFO}_{SC'}(\mathbb{U},\Theta,SC)$ (respectively for $=$) holds for any pair of input and output schemas SC and SC'.

3. Eliminating unbounded quantification

In this section we prove that unbounded quantification can be eliminated in favor of quantification bounded by the active domain for a large class of structures. This is equivalent to saying that the active-domain interpretation and the natural interpretation coincide for a large class of structures.

Recall that a structure $\langle\mathbb{U},\Omega\rangle$, where the order relation $<$ is in Ω, is called *o-minimal* [25] if every definable set $\{c \in \mathbb{U} \mid \langle\mathbb{U},\Omega\rangle \models \varphi(c)\}$ is composed of a finite union of intervals. Here φ is a formula in the language that includes all symbols of Ω and constants for elements of $\mathbb{U}$. Examples of o-minimal structures are $\langle\mathbb{R},<\rangle$, $\langle\mathbb{R},+,*,0,1,<\rangle$ (this follows from quantifier elimination [9]) and $\langle\mathbb{R},+,*,e^x\rangle$ [28].

Our main result is as follows:

Theorem 1 *Let Ω be a signature on $\mathbb{U}$ such that $\langle\mathbb{U},\Omega\rangle$ is o-minimal and admits quantifier elimination. Then*

$$\mathcal{FO}(\mathbb{U},\Omega) = \mathcal{FO}^{\emptyset}(\mathbb{U},\Omega)$$

That is, every first-order query using symbols from Ω and the schema relations is equivalent to a formula where the quantifiers are bounded by the active domain.

Proof sketch: As in [4], we give here a nonconstructive proof using the technique of nonstandard universes. For all the definitions, see [4].

Lemma 1 *If we have two hyperfinite instances A and A' that agree on every standard active-semantics query, then they agree on every natural-semantics query.*

Proof of Lemma. As in [4], ${}^*\mathcal{M}$ is the nonstandard extension of $\mathcal{M}$ in a nonstandard universe satisfying the Isomorphism Property of [16]. Fix a counterexample, that is, fix A and A', and a natural-semantics query φ on which they disagree. Let ${}^*\mathcal{M}(A)$ be the expansion of ${}^*\mathcal{M}$ to $L(SC,\Omega)$ given by interpreting the schema relation symbols as in A, and let ${}^*\mathcal{M}(A')$ be likewise.

Let Ω' be the language containing (only) predicate symbols for each atomic formula of Ω. Let $\mathcal{M}'$ be the model for Ω' with domain equal to $\mathbb{U}$, and with the predicates of Ω' interpreted in the obvious way. Then $\mathcal{M}'$ also admits elimination of quantifiers. Consider $\Omega'(A)$ and $\Omega'(A')$ as structures for $L(SC,\Omega')$ in which the domains are the active domains of A and A', respectively, the schema relations are unchanged, and each predicate of Ω' is interpreted as the * of the corresponding definable subset of $\mathcal{M}$. Using the assumption that A and A' agree on standard active-semantics queries, we can show:

Claim 1 *$\Omega'(A)$ and $\Omega'(A')$ are elementary equivalent in $L(SC,\Omega')$*

We apply the Isomorphism Property to $\Omega'(A)$ and $\Omega'(A')$, to get a mapping f from the active domain of A onto active domain of A' that preserves schema relations and the predicates of Ω'. Since $\mathcal{M}'$ has elimination of quantifiers, for each $\varphi(\vec{x})$ in Ω' and $\vec{c}$ in the active domain of A, we have ${}^*\mathcal{M}' \models \varphi(\vec{c})$ if and only if ${}^*\mathcal{M}' \models \varphi(f(\vec{c}))$, since $\varphi(\vec{c})$ is equivalent to a boolean combination of atomic formulae, each of which will be preserved by f.

Now it follows from the techniques developed in [4] that ${}^*\mathcal{M}'(A)$ and ${}^*\mathcal{M}'(A')$ (which are defined analogously to ${}^*\mathcal{M}(A)$ and ${}^*\mathcal{M}(A')$ but for the language Ω' instead of Ω) satisfy all the same sentences of $L(SC, \Omega')$. Then it can be shown that ${}^*\mathcal{M}(A)$ and ${}^*\mathcal{M}(A')$ satisfy all the same sentences of $L(SC, \Omega)$. This gives us a contradiction, which proves the lemma.

To show that lemma 1 implies the theorem, suppose there were a counterexample q to the theorem (that is, q is definable as a natural-semantics query, but not as an active-semantics query). We first note that for every finite collection F of active-semantics queries, there must be two finite instances A_F and ${A'}_F$ that agree on all queries in F but disagree on q . By applying saturation, we would get two hyperfinite instances A and A' in the nonstandard universe that agree on every standard active-semantics query, but disagree on q, contradicting lemma 1. This completes the proof. □

The proof merely establishes the existence of an active-domain query that is equivalent to a query using unbounded quantification. However, the process of transforming an unbounded-quantifier sentence into a bounded-quantifier sentence can be done effectively assuming that the quantifier elimination procedure for the underlying model is effective. We shall present such a procedure in a subsequent paper.

Now, using the o-minimality of $\langle \mathbb{R}, +, *, 0, 1, < \rangle$ and theorem 1, we settle the open problem from [24].

Corollary 1 *Every first-order query in the language of the schema relations and $+, *, 0, 1, <$ can be expressed by a formula in the same language with all quantifiers bounded by the active domain. That is,* $\mathcal{FO}(\mathbb{R}, +, *, 0, 1, <) = \mathcal{FO}^{\emptyset}(\mathbb{R}, +, *, 0, 1, <)$. □

For example, consider $\langle \mathbb{R}, +, *, 0, 1, < \rangle$, and let our schema have a binary predicate $S(x, y)$. The sentence φ states that all elements of S lie on some line:

$$\varphi \;=\; \exists a \exists b \forall x \forall y. (S(x, y) \rightarrow y = a * x + b)$$

This gets converted to the equivalent active-domain sentence $\psi = (card(S) < 3) \vee ((card(S) \geq 3) \wedge \psi')$ where the conditions on cardinality of S are written as first-order sentences in the language of S, and ψ' is

$\exists x_1 \exists y_1 \forall x_2 \forall y_2 \; \forall x_3 \forall y_3 . S(x_1, y_1) \;\wedge\; (S(x_2, y_2) \;\wedge\; S(x_3, y_3) \rightarrow (x_2 - x_1)(y_3 - y_1) = (y_2 - y_1)(x_3 - x_1))$

Then $D \models \varphi$ iff $D \models_{\emptyset} \psi$ for any $D \in Inst(\mathbb{R}, \{S\})$.

The analog of corollary 1 for linear constraints was proved in [24]. These result stand in sharp contrast to the results of [14], who showed that elimination of unbounded quantifiers fails for integer arithmetic constraints.

The ability to convert natural-semantics queries to bounded-quantifier queries is important for achieving efficient query evaluation. In addition, corollary 1 gives us an alternative proof of the conjecture that parity test cannot be defined by first-order queries that use polynomial inequality constraints (this conjecture was recently confirmed in [4]). Indeed, assume that parity is definable in such a way; then it is definable under the active semantics, and we know (see [4] and next section) that this is not the case.

Using the fact that each model has a definitional expansion to a model that admits quantifier elimination, we obtain:

Corollary 2 *Let Ω be an o-minimal signature on $\mathbb{U}$. Then we can find a (definitional) expansion of Ω to Ω' such that*

$$\mathcal{FO}(\mathbb{U}, \Omega) = \mathcal{FO}(\mathbb{U}, \Omega') = \mathcal{FO}^{\emptyset}(\mathbb{U}, \Omega').$$

4. Collapsing signatures

4.1 First-order logic

The goal of this section is to investigate the approximate collapse relation. We are interested in results collapsing queries over signature Ω to queries over Θ, where Θ is much simpler than Ω. We start by reviewing the first-order case. We state a generalization of the result from [4] and [21], which can be used to get expressivity bounds on first-order constraint languages (this technique is already implicit in [24]). We will then make use of the techniques developed in these proofs to extend the arrow relation results to second-order logic and to existential second-order.

Our first approximate collapse result shows that any signature approximately collapses to the order relation. The proof of the theorem below follows the basic idea of [4]: first, rewrite a query, and then use Ramsey theorem [12] repeatedly to eliminate all constraints other than order comparisons.

Theorem 2 *Let $\mathbb{U}$ be ordered by $<$. Then for any $L(SC, \Omega)$ sentence φ, we can find an infinite set $X \subseteq \mathbb{U}$*

and a $L(SC,<)$ sentence ψ such that for any $D \in Inst(SC,X)$, and any $Y \subseteq X$, it is the case that

$$D \models_Y \varphi \quad \textit{iff} \quad D \models_Y \psi.$$

Corollary 3 *Let $\mathbb{U}$ be ordered by $<$. Then, for any signature Ω,*

$$\mathcal{FO}(\mathbb{U},\Omega) \Rightarrow \mathcal{FO}(\mathbb{U},<).$$

Theorem 2 was proved for the case $Y = \emptyset$ in [4]. Results of this kind are particularly useful for studying expressibility under the active-domain semantics, as demonstrated in [4] and [21]. For example, many queries of interest are generic, that is, independent under permutations of the underlying domain. For such queries, their behavior on an infinite subset of $\mathbb{U}$ fully determines their behavior on $\mathbb{U}$. For example, it can be immediately derived from theorem 2 that for any Ω, there is no $L(SC,\Omega)$ sentence φ such that $D \models_\emptyset \varphi$ iff $adom(D)$ has even cardinality.

It is generally impossible to eliminate the order relation from the right hand side of the arrow relation. However, it was shown in [4] that for signatures over the reals satisfying certain smoothness conditions, collapse results to pure relational algebra are available.

Definition 2 *A signature Ω on $\mathbb{R}$ is called* analytic *if it consists of restrictions of analytic functions to real arguments.*

In other words, $\Omega = (f_i)_{i\in I}$ is analytic if there is a set of analytic functions $(F_i)_{i\in I}$ such that each f_i is the restriction of F_i to the real arguments. For example, $(+,*,e^x)$ is an analytic signature.

Theorem 3 *Let Ω be an analytic signature on the reals. Then for any $L(SC,\Omega)$ sentence φ, we can find an infinite (in fact, uncountable) set $X \subseteq \mathbb{U}$ and a $L(SC)$ sentence ψ such that for any $D \in Inst(SC,X)$, and any $Y \subseteq X$, it is the case that*

$$D \models_Y \varphi \quad \textit{iff} \quad D \models_Y \psi.$$

A roughly analogous result was proved in [4], although the results there contained extra hypotheses and provided no cardinality information. The main difference between the proof of theorem 3 and the proof in [4] is that here we demonstrate the existence of an uncountable set X. This is done by showing that a family of nontrivial equations $f_i(\vec{x}) = 0$, where f_is are terms in $\Omega \cup \{r \mid r \in \mathbb{R}\}$, can be simultaneously invalidated by assigning distinct values from some uncountable set to distinct variables x_js.

As a corollary, we obtain the following fact about the arrow relation.

Corollary 4 *Let Ω be an analytic signature on the reals. Then*

$$\mathcal{FO}(\mathbb{R},\Omega) \Rightarrow \mathcal{FO}(\mathbb{R},\emptyset).$$

Combining the results of this section with the collapse result of section 3, we obtain:

Corollary 5 *Assume that $(\mathbb{U},<)$ is a dense order without endpoints. Then for an arbitrary signature Ω we have $\mathcal{FO}(\mathbb{U},\Omega) \Rightarrow \mathcal{FO}^\emptyset(\mathbb{U},<)$. Also, $\mathcal{FO}(\mathbb{R},\Omega) \Rightarrow \mathcal{FO}^\emptyset(\mathbb{R},\emptyset)$ for any analytic signature Ω on the reals.* □

We note that the hypothesis that the underlying order is dense in corollary 5 cannot be removed:

Proposition 1 *Let $(\mathbb{U},<)$ be a scattered linear ordering. Then $\mathcal{FO}(\mathbb{U},<) \not\Rightarrow \mathcal{FO}^\emptyset(\mathbb{U},<)$.* □

4.2 Second-order logic

The goal of this section is to generalize approximate collapse results to second-order logic and its fragments. When we deal with sentences, we assume that they are converted into normal form. That is, sentences

$$\mathrm{Q}'_1 P_1 \ldots \mathrm{Q}'_m P_m \mathrm{Q}_1 x_1 \ldots \mathrm{Q}_n x_n . \varphi(x_1,\ldots,x_n)$$

where $\mathrm{Q}'_i P_i$ are second-order quantifiers, and φ is a first-order formula in the language that contains $P_1,\ldots,P_m$. Now, suppose that $D \in Inst(SC,\mathbb{U})$. For Φ a second-order sentence, we define $D \models \Phi$ in the usual way. Furthermore, for $X \subseteq \mathbb{U}$ we define $D \models_X \Phi$ by letting all first-order quantifiers range over $adom(D) \cup X$ and letting each second-order quantifier $\mathrm{Q}'_i P_i$ range over $2^{(adom(D)\cup X)^k}$, if P_i is of arity k.

For a schema SC, a signature Ω on $\mathbb{U}$, the class of second-order constraint queries, under the X interpretation, will be denoted by $\mathcal{SO}^X(\mathbb{U},\Omega,SC)$. That is, $\mathcal{SO}^X(\mathbb{U},\Omega,SC)$ is the class of queries Q over some schema SC for which there exists a second-order sentence Φ such that $Q(D) = \mathbf{T}$ iff $D \models_X \Phi$. As before, we write $\mathcal{SO}$ instead of $\mathcal{SO}^\mathbb{U}$. Similarly to the first order case, omitting the schema in a statement about equation or arrow relation means "for all schemas".

We shall also consider fragments of second-order logic given by the quantifier prefixes of second-order quantifiers. Formally, a prefix is a finite sequence of pairs $(i,\forall)$ or $(i,\exists)$ where $i > 0$. Then a (normal form) second-order sentence conforms to the prefix $(i_1,\mathrm{Q}^1),\ldots,(i_s,\mathrm{Q}^s)$ if it has s second-order quantifiers, the jth one is Q^j and it binds predicates of arity i_j.

A fragment F is the set of sentences that conform to some collection of prefixes. Examples of fragments are full second-order logic, existential second-order logic

and monadic Σ_1^1. The fragment associated with a set of prefixes F (under the X-interpretation) will be denoted by $\mathcal{SO}_F^X(\mathbb{U},\Omega,SC)$, or $\mathcal{SO}_F(\mathbb{U},\Omega,SC)$ if $X=\mathbb{U}$.

We call a fragment given by a collection of prefixes F *orderable* if $F \neq \emptyset$ and for every $f \in F$, $(m,\exists)$ followed by f is in F for some $m \geq 2$. Examples are full second-order, and existential second-order $\mathcal{ESO}$. A non-orderable fragment is monadic Σ_1^1. Now we can prove a result that extends approximate collapse theorems to second-order constraint queries, and further generalizes them for orderable fragments of second-order logic.

Theorem 4 *1) Let F be an arbitrary fragment. Then $\mathcal{SO}_F(\mathbb{U},\Omega) \Rightarrow \mathcal{SO}_F(\mathbb{U},<)$. Furthermore, $\mathcal{SO}_F(\mathbb{R},\Omega) \Rightarrow \mathcal{SO}_F(\mathbb{R},\emptyset)$ if Ω is analytic.*

2) Let F be an orderable fragment of second-order logic. Then

$$\mathcal{SO}_F(\mathbb{U},\Omega) \Rightarrow \mathcal{SO}_F(\mathbb{U},\emptyset)$$

In particular, $\mathcal{SO}(\mathbb{U},\Omega) \Rightarrow \mathcal{SO}(\mathbb{U},\emptyset)$ and $\mathcal{ESO}(\mathbb{U},\Omega) \Rightarrow \mathcal{ESO}(\mathbb{U},\emptyset)$. These results are true for the active-domain interpretation as well; that is, $\mathcal{SO}_F^\emptyset(\mathbb{U},\Omega) \Rightarrow \mathcal{SO}_F^\emptyset(\mathbb{U},\emptyset)$.

Proof sketch: The proof proceeds by converting a second-order formula into a normal form, and then applying the techniques in the proofs of theorem 2 and 3 to the first-order part. For any orderable fragment we can also get rid of the order relation, because it is definable by one extra second-order quantifier over m-ary relations for any $m \geq 2$. □

It is easy to show that part 2) of theorem 4 fails for monadic Σ_1^1.

As we saw earlier, any approximate collapse result completely describes the behavior of generic queries (those invariant under permutations of the domain, such as parity test or transitive closure). Thus, we obtain

Corollary 6 *If F is an orderable fragment, and Ω is an arbitrary signature, then every generic query in $\mathcal{SO}_F^\emptyset(\mathbb{U},\Omega)$ is expressible in $\mathcal{SO}_F^\emptyset(\mathbb{U},\emptyset)$.* □

From this we get some expressivity bounds. For example, connectivity of directed graphs is not definable under the active interpretation as a monadic Σ_1^1 constraint query, no matter what operations are in the signature. Similarly, any query that is complete for exponential space cannot be defined as a second-order constraint query under the active interpretation.

Note also that the coincidence of the active and natural interpretations proved for the first-order logic with polynomial constraints does not extend to the second-order case.

Proposition 2 *For any fragment F that allows existential quantifiers over unary predicates (e.g., full second-order, existential second-order, monadic Σ_1^1), $\mathcal{SO}_F(\mathbb{R},+,*,0,1,<) \neq \mathcal{SO}_F^\emptyset(\mathbb{R},+,*,0,1,<)$.*

Proof sketch: The set of natural numbers can be defined by a second-order formula with one unary existential second order quantifier in the language of $+,0,<$. Then it follows from [14] that any total recursive query on databases whose active domain consists only of natural numbers is in $\mathcal{SO}_F(\mathbb{R},+,*,0,1,<)$. On the other hand, every query in $\mathcal{SO}^\emptyset(\mathbb{R},+,*,0,1,<)$ (and thus in $\mathcal{SO}_F^\emptyset(\mathbb{R},+,*,0,1,<)$) has *PSPACE* data complexity, which proves the proposition. □

5. Complexity of constraint queries

As was mentioned in the introduction, the results on elimination of unbounded quantification allow us to prove new low complexity bounds. In this section we use theorem 1 to give a new complexity bound for $\mathcal{FO}(\mathbb{R},+,*,0,1,<)$. We also use corollary 6 to establish complexity bound on generic second-order queries.

We are dealing with data complexity, that is, the complexity of evaluating a given query for instances that vary. We only look at boolean queries here, but all results generalize easily for nonboolean queries. Assume some encoding of instances, for example, the encoding of [1]. Given $D \in \mathit{Inst}(SC,\mathbb{U})$, we denote its encoding by $\mathit{enc}(D)$. Then for each boolean query φ we define the language $L_\varphi = \{\mathit{enc}(D) \mid D \in \mathit{Inst}(SC,\mathbb{U}), D \models \varphi\}$. The data complexity of φ is the conventional complexity of L_φ. In particular, for any complexity class $\mathcal{C}$ we say that φ has $\mathcal{C}$ data complexity if the language L_φ is in $\mathcal{C}$.

It was previously known [20] that every query from $\mathcal{FO}(\mathbb{R},+,*,0,1,<)$ has NC data complexity. In fact, this follows from the NC complexity bound for the first-order theory of real closed fields with a fixed number of variables [6]. However, pure first-order logic queries, as well as first-order queries with linear constraints, have AC^0 data complexity [1, 15] and we know that $AC^0 \subset NC$ [7]. So the question arises if we can improve the data complexity bound for $\mathcal{FO}(\mathbb{R},+,*,0,1,<)$ using the elimination of unbounded quantification result proved in section 3. This is indeed possible. We prove below a TC^0 complexity bound. Recall that AC^0 is the class of problems definable with unbounded fan-in constant depth circuits that use *and*, *or* and *not* gates, and the number of gates is polynomial in the size of input. The class TC^0 extends AC^0 by allowing threshold gates, or equivalently majority gates [3]. It is known that

$AC^0 \subset TC^0 \subseteq NC^1 \subseteq L \subseteq NL \subseteq NC$ and all $\subseteq$ inclusions are conjectured to be strict [3].

Theorem 5 *Every query in $\mathcal{FO}(\mathbb{R}, +, *, 0, 1, <)$ has TC^0 data complexity.*

Proof sketch: Let RA_{poly} be relational algebra in which polynomial inequality constraints are allowed as selection predicates. For instance, $\sigma_{\#1>\#2^3+4}(R)$ selects pairs (x, y) for which $x > y^3 + 4$. Using theorem 1 and the standard technique for equivalence of relational algebra and calculus, we show that, for each $L(SC, +, *, 0, 1, <)$ sentence φ, there is a RA_{poly} expression e_φ such that $D \models \varphi$ if $e_\varphi(D) = \{()\}$ (empty tuple) and $D \models \neg\varphi$ if $e_\varphi(D) = \{\}$.

Next, we show that every RA_{poly} query has TC^0 data complexity. The proof proceeds exactly as the proof of AC^0 data complexity for relational algebra (see [1]) with one exception: every time the σ_p operator is encountered, we have to compute the condition p. If p is of form $t_1(\vec{y})\{=, <, \neq, \not<\}t_2(\vec{z})$ where t_1, t_2 are terms (that is, polynomials), we construct circuits that compute t_1 and t_2 first and then make the comparison. Since addition and multiplication are in TC^0 [3], we can insert a threshold circuit that computes σ_p. The theorem is proved. □

Since the behavior of generic (invariant under permutations) queries is fully determined by their behavior on an infinite set, we obtain from corollary 1 (this can also be derived from combining the results of [4] and [15]):

Corollary 7 *Every generic query in $\mathcal{FO}(\mathbb{R}, +, *, 0, 1, <)$ has AC^0 data complexity.* □

From corollary 6 and classical descriptive complexity results (cf. [18]) we obtain

Corollary 8 *Every generic query in $\mathcal{SO}^\emptyset(\mathbb{U}, \Omega)$ has PH data complexity, and every generic query in $\mathcal{ESO}^\emptyset(\mathbb{U}, \Omega)$ has NP data complexity.* □

6. Extension to nonboolean first-order queries

In this section we show the following: *all results concerning boolean queries extend to arbitrary nonboolean queries.* In other words, all results we proved for sentences can be also proved for formulae with free variables. We show this by proving "transfer" results that extend a $\Rightarrow$ or an equality result from boolean queries to arbitrary ones. Note that transfer results for *generic* queries (active or natural) were proved earlier in [4].

Since we are now interested in arbitrary queries, we need two schemas: the *input schema* SC_1 and the *output schema* SC_2. Given the underlying model $\langle \mathcal{M}, \Omega \rangle$, a query is given by a formula $\varphi(x_1, \ldots, x_n)$, in the language $L(SC_1, \Omega)$, for each n-ary predicate symbol in SC_2. For each input $D \in Inst(SC, \mathbb{U})$ and each set X, under the X-interpretation such a query defines the n-ary relation $\varphi_X[D] = \{\vec{a} \mid \vec{a} \in \mathbb{U}^n, D \models_X \varphi(\vec{a})\}$. The class of first-order queries between the schemas SC_1 and SC_2, under the X-interpretation, is denoted by $\mathcal{FFO}^X_{SC_2}(\mathbb{U}, \Omega, SC_1)$. Recall that this is a set of *semantic objects.*

Note that $\varphi_X[D]$ need not be finite. That is, $\mathcal{FFO}^X_{SC_2}(\mathbb{U}, \Omega, SC_1)$ is actually a set of maps from $Inst(SC_1, \mathbb{U})$ to $Inst_\infty(SC_2, \mathbb{U})$, where $Inst_\infty(\cdot)$ is the class of finite and infinite instances. Since we are often interested in the class of maps from $Inst(SC_1, \mathbb{U})$ to $Inst(SC_2, \mathbb{U})$, we define a restriction on queries that guarantees finiteness.

We call a query Q *domain-preserving* if for any input D, $adom(Q(D)) \subseteq adom(D)$. That is, no element of $\mathbb{U}$ can be present in $Q(D)$ unless it is present in D itself. (Every query expressed in relational algebra or safe relational calculus is such.) For a class of queries $\mathcal{C}$, we denote the subclass of domain-preserving queries in $\mathcal{C}$ by $\mathsf{dp}\_\mathcal{C}$.

The arrow notation extends to nonboolean queries and to domain preserving queries in the natural way. As usual, omitting SC_1 and SC_2 in the equality or arrow relation means that the equality or arrow relation holds for all SC_1 and SC_2. Now we prove the first transfer theorem that allows us to extend the arrow results to arbitrary queries.

Theorem 6 *For any signatures Ω and Θ, $\mathcal{FO}$ $(\mathbb{U}, \Omega)$ $\Rightarrow$ $\mathcal{FO}(\mathbb{U}, \Theta)$ implies $\mathcal{FFO}(\mathbb{U}, \Omega) \Rightarrow \mathcal{FFO}(\mathbb{U}, \Theta)$.*

Furthermore, $\mathcal{FO}^A(\mathbb{U}, \Omega) \Rightarrow \mathcal{FO}^B(\mathbb{U}, \Theta)$ implies $\mathsf{dp}\_\mathcal{FFO}^A(\mathbb{U}, \Omega) \Rightarrow \mathsf{dp}\_\mathcal{FFO}^B(\mathbb{U}, \Theta)$.

Proof sketch: We follow the idea of [4], where a similar transfer result was proved for the *equality* of classes of *generic* queries. Let $\varphi(x_1, \ldots, x_n)$ be a $L(SC_1, \Omega)$ formula defining a nonboolean query for some n-ary relational symbol in SC_2. We extend SC_1 to SC' by n unary predicate symbols $S_1, \ldots, S_n$ which are not present in $SC_1 \cup SC_2$. Define the following $L(SC', \Omega)$ sentence Φ:

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

Using the assumption, we get a $L(SC', \Theta)$ sentence Ψ and an infinite set X such that for every $D \in Inst(SC', X)$, $D \models_X \Phi$ iff $D \models_X \Psi$. Let $\psi(z_1, \ldots, z_n)$ be a $L(SC_1, \Theta)$ formula obtained from Ψ be replacing

each $S_i(z)$ with $z = z_i$, where z_is are not used in Ψ. Now $\mathcal{FFO}_{SC_2}(\mathbb{U}, SC_1, \Omega) \Rightarrow \mathcal{FFO}_{SC_2}(\mathbb{U}, SC_1, \Theta)$ is proved by showing that $D \models_X \varphi(\vec{a})$ iff $D \models_X \psi(\vec{a})$ for every $D \in Inst(SC_1, X)$ and every $\vec{a} \in (adom(D) \cup X)^n$. Part 2 is proved similarly. □

From this we immediately obtain:

Corollary 9 **1)** $\mathcal{FFO}(\mathbb{U}, \Omega) \Rightarrow \mathcal{FFO}(\mathbb{U}, <)$.
2) $\mathcal{FFO}(\mathbb{R}, \Omega) \Rightarrow \mathcal{FFO}(\mathbb{R}, \emptyset)$, *if* Ω *is analytic.*
3) *If* $\mathbb{U}$ *is a dense order without endpoints, then* $\mathsf{dp}\_\mathcal{FFO}(\mathbb{U}, \Omega) \Rightarrow \mathsf{dp}\_\mathcal{FFO}^{\emptyset}(\mathbb{U}, <)$.
4) *If* Ω *is analytic, then* $\mathsf{dp}\_\mathcal{FFO}(\mathbb{R}, \Omega) \Rightarrow \mathsf{dp}\_\mathcal{FFO}^{\emptyset}(\mathbb{R}, \emptyset)$.

Note that the right hand sides in 3) and 4) are the classes of queries well known in the classical relational theory. Indeed, $\mathsf{dp}\_\mathcal{FFO}^{\emptyset}(\mathbb{U}, SC)$ is precisely the class of queries that can be defined by the relational algebra, and $\mathsf{dp}\_\mathcal{FFO}^{\emptyset}(\mathbb{U}, <, SC)$ is the class of queries definable by the relational algebra with $<$ comparisons allowed in selections.

Next, we prove a transfer theorem that allows us to extend elimination of unbounded quantification to nonboolean queries.

Theorem 7 *Suppose that for some signature* Ω *it is the case that* $\mathcal{FO}(\mathbb{U}, \Omega) = \mathcal{FO}^{\emptyset}(\mathbb{U}, \Omega)$. *Then,*

$$\mathcal{FFO}(\mathbb{U}, \Omega) = \mathcal{FFO}^{\emptyset}(\mathbb{U}, \Omega).$$

Proof sketch: The proof is similar to the proof of theorem 6, but we need a different translation from Ψ to $\psi(\cdot)$ since we are dealing with bounded quantification. We define this translation as before except for the case of existential quantification: $\exists x.\chi(x, \vec{y})$ is translated into $\exists x.\chi^\circ(x, \vec{y}) \vee \chi^\circ(z_1, \vec{y}) \vee \ldots \vee \chi^\circ(z_n, \vec{y})$ where χ° is the translation of χ and $z_1, \ldots, z_n$ are the free variables of φ. Now, define $\psi(z_1, \ldots, z_n)$ as Ψ°. It can be shown that for any $D \in Inst(SC_1, \mathbb{U})$ and any $\vec{a} \in \mathbb{U}^n$, $D \models \varphi(\vec{a})$ iff $D \models_\emptyset \psi(\vec{a})$, which proves the theorem. □

Corollary 10 *Let* Ω *be o-minimal and admit quantifier elimination. Then* $\mathcal{FFO}(\mathbb{U}, \Omega) = \mathcal{FFO}^{\emptyset}(\mathbb{U}, \Omega)$. *In particular, the classes of arbitrary nonboolean queries that are first-order expressible with polynomial inequality constraints over the reals, are the same under both natural and active-domain interpretations.* □

We can now use the above results to get some expressivity bounds.

Corollary 11 *The following cannot be expressed by any first-order constraint query under the active interpretation, nor by any first-order query with polynomial inequality constraints under the natural interpretation: transitive, or deterministic transitive closure of a graph; maximal matching in a bipartite graph; Eulerian cycle.* □

7. Conclusions

Through the results of this paper, along with recent works such as [4, 21, 24], we have a much better feel for the expressive capabilities of constraint languages. Although the implications of these results for the design of query languages are dependent on many parameters of the application domain, we can draw a few general conclusions for language design.

Our results indicate that it is particularly promising to focus on constraint query languages over *o*-minimal structures. In particular, it seems that these languages inherit most of the pleasant formal properties of the pure relational calculus, along with uniform versions of the formal properties of first-order logic over the real ordered field. We can use the techniques developed here to get finer information about formal properties of definable sets that are possessed by these languages,

Although many of our main results yield constructive proofs, we have not fully explored all the algorithmic consequences of the theorems. In particular, we are interested in investigating semantic query optimization strategies enabled by theorems 2 and 1 in detail. As mentioned in section 3, there is an effective version of theorem 1, which can be seen as generalizations of the classical Tarski-Seidenberg algorithm for quantifier elimination. We are interested in seeing if algorithms based on these results can be useful in geometric theorem-proving applications that involving large numbers of rational parameters.

The results here can be seen from a mathematical view as yielding interesting information concerning the structure of sets definable from formulae with free second-order variables. As such, they can be seen as extending works such as [28], in showing the 'tame' behavior of important fragments of analytic geometry. In particular, results such as theorem 1 can yield interesting expressivity limits even in cases where there is no known effective procedure.

Many of the results within this paper point to connections between model-theoretic properties of a structure $\mathcal{M}$ and expressibility properties of the constraint query language based on $\mathcal{M}$. In future work, we plan to give more detailed information on this relationship, including results for constraint query languages based on classes appearing in model-theoretic stability theory.

We are interested in extending the (partial) collapse results to other logics (infinitary, fixpoint) in order to establish new expressivity bounds for generic queries.

We are also interested in the interplay between

Ramsey-like theorems and collapse results. We can, for instance, get additional information about collapsing sets by making use of results in Ramsey theory and set theory, cf. [10]. For instance: we extend the arrow notation by $\mathcal{FO}(\mathbb{U},\Omega) \Rightarrow_\kappa \mathcal{FO}(\mathbb{U},\Theta)$ if, for every $L(SC,\Omega)$ sentence φ, we can find an infinite set $X \subseteq \mathbb{U}$ of cardinality κ and a $L(SC,\Theta)$ sentence ψ such that for any $D \in Inst(SC,X)$, $D \models_X \varphi$ iff $D \models_X \psi$. Furthermore, we use the notation $\Rightarrow_\kappa^\lambda$ to mean that the cardinality of $\mathbb{U} - X$ is λ. We can now show that:

Theorem 8 *1) There exists a signature Ω on $\mathbb{R}$ such that $\mathcal{FO}(\mathbb{R},\Omega) \Rightarrow_\kappa \mathcal{FO}(\mathbb{R},<)$ implies $\kappa = \aleph_0$.*

2) If Ω is analytic, then $\mathcal{FO}(\mathbb{R},\Omega) \Rightarrow_c \mathcal{FO}(\mathbb{R},\emptyset)$. However, $\mathcal{FO}(\mathbb{R},+,*,\Omega) \not\Rightarrow_c^{\aleph_0} \mathcal{FO}(\mathbb{R},\emptyset)$. □

Results such as these may be useful in analyzing the finer structure of constraint queries (for instance, in the consideration of queries satisfying weaker notions of genericity), and in analyzing constraint queries with cardinality quantifiers. We plan to make a more detailed study of cardinality collapse results in subsequent work.

Acknowledgement: We thank Dan Suciu for helpful discussions of the material presented in section 5, and the referees for their comments.

References

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

[2] A. V. Aho and J. D. Ullman. Universality of data retrieval languages. In *POPL'79*, pages 110–120.

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

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

[5] M. Benedikt and L. Libkin. On the structure of queries in constraint query languages. Bell Labs Technical Memo, 1995.

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

[7] R.B. Boppana and M. Sipser. The Complexity of Finite Functions. In *Handbook of Theoretical Computer Science*, Vol. A, chapter 14, (J. van Leeuwen editor), North-Holland, 1990.

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

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

[10] P. Erdős, A. Hajnal, A. Máté and R. Rado. *Combinatorial Set Theory: Partition Relations for Cardinals*. North Holland, 1984.

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

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

[13] S. Grumbach and J. Su. Finitely representable databases, In *PODS'94*, pages 289–300.

[14] S. Grumbach and J. Su. First-order definability over constraint databases. *Proc. Conf. on Constr. Progr.*, 1995.

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

[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] N. Immerman. Descriptive complexity: A logician's approach to computation. *Notices of the AMS* 42 (1995), 1127–1133.

[19] P. Kanellakis. Constraint programming and database languages: A tutorial. In *PODS'95*, pages 46–53.

[20] P. Kanellakis, G. Kuper, and P. Revesz. Constraint query languages. *JCSS* 51 (1995), 26–52. Extended abstract in *PODS'90*.

[21] M. Otto and J. Van den Bussche. First-order queries on databases embedded in an infinite structure. Technical Report, University of Antwerp, October 1995.

[22] C. Papadimitriou, D. Suciu and V. Vianu. Topological queries in spatial databases. In *PODS'96*, pages 81–92.

[23] J. Paredaens, J. Van den Bussche, and D. Van Gucht. Towards a theory of spatial database queries. In *PODS'94*, pages 279–288.

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

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

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

[27] V. Tannen. Languages for collection types: A tutorial. in *PODS'94*, pages 150–154.

[28] 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.