Safe Constraint Queries

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

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

Abstract

We extend some of the classical characterization theorems of relational database theory — particularly those related to query safety — to the context where database elements come with fixed interpreted structure, and where formulae over elements of that structure can be used in queries. We show that the addition of common interpreted functions such as real addition and multiplication to the relational calculus preserves important *characterization theorems* of the relational calculus, and also preserves certain *combinatorial properties* of queries. Our main result of the first kind is that there is a syntactic characterization of the collection of safe queries over the relational calculus supplemented by a wide class of interpreted functions — a class that includes addition, multiplication, and exponentiation — and that this characterization gives us an interpreted analog of the concept of range-restricted query from the uninterpreted setting. Furthermore, our range-restricted queries are particularly intuitive for the relational calculus with real arithmetic, and give a natural syntax for safe queries in the presence of polynomial functions. We use these characterizations to show that safety is decidable for Boolean combinations of conjunctive queries for a large class of interpreted structures. We show a dichotomy theorem that sets a polynomial bound on the growth of the output of a query that might refer to addition, multiplication and exponentiation.

We apply the above results for finite databases to get results on constraint databases, representing potentially infinite objects. We start by getting syntactic characterizations of the queries on constraint databases that preserve geometric conditions in the constraint data model. We consider classes of convex polytopes, polyhedra, and compact semi-linear sets, the latter corresponding to many spatial applications. We show how to give an effective syntax to safe queries, and prove that for conjunctive queries the preservation properties are decidable.

1 Introduction

The power of classical query languages is linked to the fact that these queries express a restricted class of declarative programs. The class of semantic objects expressible through queries in the relational calculus, for example, is limited in a number of helpful ways: each such query is polynomial-time computable, each is local, and each has well-defined asymptotics. Although relational calculus queries may not return finite results, a natural subclass of the relational calculus does: namely the class of *range-restricted queries*. This class gives guarantees of finite output and is complete in this respect: it captures all relational calculus queries whose outputs are always finite, the *safe* queries [1, 21, 39].

*Contact author. Address: Bell Laboratories, Room 2C-407, 600 Mountain Avenue, Murray Hill, NJ 07974, USA. Phone: (908)582-7647. Fax: (908)582-5857. Email: libkin@research.bell-labs.com.

The relational theory on which these results are based deals only with *pure* relational queries: that is, those containing no interpreted predicates. Practical query languages, in contrast, contain interpreted functions such as $+$ and $*$. The resulting queries, then, make use of the domain semantics, rather than being independent of them as pure relational queries are. For example, if the underlying structure is the field of real numbers $\langle \mathbb{R}, +, *, 0, 1, < \rangle$, the extension of relational calculus is achieved by using polynomial (in)equalities. For example, the query $\varphi(x, y) \equiv \exists z. R(x, z) \wedge R(z, y) \wedge x^2 + y^2 = z$ defines a subset of the self-join with the condition that in joinable tuples (x, z) and (z, y), z must be the sum of squares of x and y. A natural question, then, is what sort of restrictions still apply to queries given with interpreted structure.

Clearly, many standard results fail in the presence of interpreted structure; for example, queries may no longer express only local properties of inputs. Complexity bounds are often dependent on fragile properties of both the functions present and the encodings of the structures in some computational model. In contrast, we show here that certain kinds of *structural* properties of relational calculus queries remain when a reasonable interpreted structure is present. These include the classical equivalence of safe and range-restricted queries, decidability of safety for restricted classes of queries, as well as *combinatorial* properties of the queries; restrictions on the growth rate of the result sets, for example. A primary example of well-behaved combinatorics of these structures is a *growth dichotomy theorem*, which says that the output of a query is either polynomial in the database or infinite. We show that the well-behavedness of a structure, together with the decidability of its first-order theory, has *algorithmic* consequences; for example, the set of range-restricted formulae can be effectively computed.

A problem related to safety is that of *state-safety*, studied in [3]: for a query *and* a database, determine if the output is finite. Unlike the safety problem, which is undecidable (cf. [1]), the state-safety problem is decidable for some domains, for example, natural numbers with the order relation, see [3, 5]. However, there are interpreted structures (even having a decidable first-order theory) for which this problem is undecidable [36]. Moreover, [36] established that for the same interpreted structure, no recursive set of queries captures the class of safe queries; that is, it is impossible to have an analog of the concept of range-restriction. In contrast, we show that for many well-behaved structures, state-safety is decidable. Furthermore, safety over all states is decidable for restricted classes of queries (namely, Boolean combinations of conjunctive queries).

The above results are for the standard relational calculus with interpreted functions on finite structures; we then apply these results to get structural restrictions on the behavior of queries in other models, particularly the constraint database model introduced by Kanellakis, Kuper and Revesz [20]. This model is motivated by new applications involving spatial and temporal data, which require storing and querying infinite collections. The constraint model generalizes the relational model by means of "generalized relations". These are possibly infinite sets defined by quantifier-free first-order formulae in the language of some underlying infinite structure $\mathcal{M} = \langle \mathcal{U}, \Omega \rangle$. Here $\mathcal{U}$ is a set (assumed to be infinite), and Ω is a signature that consists of a number of interpreted functions and predicates over $\mathcal{U}$. For example, in spatial applications, $\mathcal{M}$ is usually the real field $\langle \mathbb{R}, +, *, 0, 1, < \rangle$, and generalized relations describe sets in $\mathbb{R}^n$.

A database given by a quantifier-free formula $\alpha(x_1, \ldots, x_n)$ in the language of Ω defines a (possibly infinite) subset of $\mathcal{U}^n$ given by $D_\alpha = \{\vec{a} \in \mathcal{U}^n \mid \mathcal{M} \models \alpha(\vec{a})\}$. Such databases are called *finitely representable* [16], as the formula α provides a finitary means for representing an infinite set. For example, if $\alpha(x, y) \equiv (x^2 + y^2 \leq 1)$, then D_α is the circle of radius 1 with the center in $(0, 0)$.

Relational calculus can be straightforwardly extended to this model, by incorporating atomic formulas

which are Ω-constraints, that is, atomic Ω-formulae. For example, $\varphi(x, y) \equiv (D(x, y) \wedge y = x^2)$ is a first-order query which, on D_α defined above, returns the intersection of the circle with the graph of the function $y = x^2$.

One of the reasons why the constraint model can be used in spatial applications is that such queries admit a form of safety: the closed form evaluation over structures $\langle \mathbb{R}, +, -, 0, 1, < \rangle$ (linear constraints) and $\langle \mathbb{R}, +, *, 0, 1, < \rangle$ (polynomial constraints), most often used to represent spatial data. This sort of closure is a reformulation of the fact that the two structures above admit *effective quantifier-elimination*. To evaluate a query, one can replace each occurrence of a database symbol D by its representation as a collection of constraints, apply the quantifier-elimination procedure to the resulting formula, and obtain a quantifier-free formula giving a finite representation of the output. There has been work in extending these closure properties to other classes of constraint databases and other logics, and this work indicates that the existence of closure properties is often problematic. For example, for integer gap-order constraints $x <_n y$ (meaning $x < y + n$), restrictions guaranteeing safety were studied in [32] for relational calculus and stratified datalog, and the inherent incompleteness of those restrictions was later shown in [38]. However, in this paper we do obtain new positive results on closure properties for constraint queries— albeit of a different nature than [32] and [38] — and we do so in domains that are quite relevant to spatial applications.

For those domains, we consider the preservation of restricted geometric classes of databases within powerful constraint query languages. In particular, we consider the behavior of queries with polynomial functions over *linear* constraint databases. Linear constraints are used to represent spatial data in many applications, see [13, 17, 29] and references therein. Linear constraints have several advantages over polynomial: the quantifier-elimination procedure is less costly, and numerous algorithms have been developed to deal with figures represented by linear constraints, cf. [26]. At the same time, the extension of relational calculus with linear constraints has severely limited power as a query language, see [2, 29]. Thus, it appears to be natural to use a more powerful language, such as relational calculus with polynomial constraints, to query databases represented by linear constraints [43].

As soon as the class of constraints used in queries is more general than the class used to define databases, we encounter the safety/closure property again: the output of a query using polynomial constraints may fail to be definable with linear constraints alone! More generally, if spatial databases are required to have certain geometric properties, then the safety problem is whether those geometric properties are preserved by a given query language.

When the underlying query language is relational calculus with polynomial constraints, there is a recursively enumerable class of programs that express exactly those queries that preserve the property of being definable with linear constraints; this follows from the decidability of the latter property, shown in [13]. Corresponding to our results in the first part of the paper, we are interested in getting explicit and natural complete languages for preserving linear-constraints, and also natural effective syntax for other geometric properties. We give a general schema for coming up with such languages. As applications, we consider the properties of being a convex polytope, a convex polyhedron and a compact semi-linear set. For those classes, we provide an effective syntax for polynomial constraint queries preserving the properties. We also show that for unions of conjunctive queries with polynomial constraints, it is decidable whether the properties of being a convex polytope or a compact semi-linear set are preserved. By applying the growth bounds for the relational calculus with interpreted functions, we find restrictions of the growth in the number of vertices of polytopes, and use them to show that certain kinds of triangulations cannot be done even with very powerful constraints.

Organization In Section 2, we present the notations. Sections 3 through Section 6 all deal primarily with the standard relational calculus with interpreted functions (although generalizations to the "natural semantics" hold, and are given here, as well). Section 3 shows that the underlying interpreted structure one uses does matter: we define the concept of a *safe translation* of queries, and show that some structures admit it, and some don't. It follows that for many common structures the state-safety problem is decidable.

In Section 4, we define the concept of range-restriction, and show that the classes of safe and range-restricted queries coincide over well-behaved structures. The general concept of range-restriction is based on the notion of algebraic formulae in the underlying model. We show that for polynomial functions, these range-restricted formulae have a particularly nice characterization, namely as queries that are bounded by roots of polynomials with coefficients from the database. We then show that for underlying structures admitting quantifier-elimination, it is possible to construct, effectively, a range-restricted query that coincides with a given query Q on all databases for which Q is safe.

In Section 5 we show that over well-behaved structures, it is decidable whether a Boolean combination of conjunctive queries is safe.

Section 6 establishes the dichotomy result: for every query Q over a well-behaved structure, one can find a polynomial p such that the size of $Q(D)$ is either infinite, or at most the value of p on the size of D. We also prove a stronger trichotomy theorem for monotone queries.

Section 7 deals with finitely representable databases. We first introduce a general schema for transferring results about query safety to the finitely-representable setting. We then apply this to show bounds on the growth of vertices of polytopes in safe constraint queries, and give effective syntax for queries preserving geometric properties, such as the classes of convex polytopes, polyhedra and compact semi-linear sets (the latter only in two-dimensional case). We also show that it is decidable whether unions of conjunctive queries with polynomial constraints preserve the first and the third property. Concluding remarks are given in Section 8.

An extended abstract of this paper appeared in the Proceedings of the 17th Symposium on Principles of Database Systems [9].

2 Notations

The notations we use are fairly standard in the literature on constraint databases, cf. [6, 8, 7, 28, 29]. We study databases over infinite structures. Let $\mathcal{M} = \langle \mathcal{U}, \Omega \rangle$ be an infinite structure, where $\mathcal{U}$ is an infinite set, called a carrier (in the database literature it is often called domain), and Ω is a set of interpreted functions, constants, and predicates. For example, for the real field $\langle \mathbb{R}, +, *, 0, 1, < \rangle$, the carrier is $\mathbb{R}$ (the set of real numbers), and the signature consists of the functions $+$ and $*$, constants 0 and 1, and predicate $<$.

A (relational) *database schema* SC is a nonempty collection of relation names $\{S_1, \ldots, S_l\}$ with associated arities $p_1, \ldots, p_l > 0$. Given $\mathcal{M}$, an *instance* of SC over $\mathcal{M}$ is a family of finite sets, $\{R_1, \ldots, R_l\}$, where $R_i \subset \mathcal{U}^{p_i}$. That is, each schema symbol S_i of arity p_i is interpreted as a finite p_i-ary relation over $\mathcal{U}$. Given an instance D, $adom(D)$ denotes its *active domain*, that is, the set of all elements that occur in the relations in D. We normally assume $adom(D) \neq \emptyset$. Although often convenient in simplifying notation, this restriction is by no means necessary, as all results straightforwardly extend to empty databases.

As our basic query language, we consider relational calculus, or *first-order logic*, FO, over the underlying models and the database schema. In what follows, $L(SC, \Omega)$ stands for the language that contains all symbols of SC and Ω; by $\mathrm{FO}(SC, \Omega)$ we mean the class of all first-order formulae built up from the atomic SC and Ω-formulae by using Boolean connectives $\vee, \wedge, \neg$ and quantifiers $\forall, \exists$ and $\forall x \in adom, \exists x \in adom$. When Ω is $(+, -, 0, 1, <)$, or $(+, *, 0, 1, <)$, or $(+, *, e^x, 0, 1, <)$, we use the standard abbreviations FO + Lin, FO + Poly and FO + Exp, often omitting the schema when it is understood. Regardless of whether we are in the 'classical' setting, where these queries are applied to finite databases, or in the constraint query setting discussed later in the paper, we will refer to the syntactic query languages as *relational calculus with Ω constraints*.

The semantics is as follows. For a structure $\mathcal{M}$ and a SC-instance D, the notion of $(\mathcal{M}, D) \models \varphi$ is defined in a standard way for $\mathrm{FO}(SC, \Omega)$ formulae, where $\exists x \in adom$ is the *active-domain* quantification. That is, $(\mathcal{M}, D) \models \exists x\ \varphi(x, \cdot)$ if for some $a \in \mathcal{U}$ we have $(\mathcal{M}, D) \models \varphi(a, \cdot)$, and $(\mathcal{M}, D) \models \exists x \in adom\ \varphi(x, \cdot)$ if for some $a \in adom(D)$ we have $(\mathcal{M}, D) \models \varphi(a, \cdot)$. If $\mathcal{M}$ is understood, we write $D \models \varphi$. The output of a query $\varphi(x_1, \ldots, x_n)$ on D is $\{\vec{a} = (a_1, \ldots, a_n) \in \mathcal{U}^n \mid D \models \varphi(\vec{a})\}$, and it is denoted by $\varphi[D]$. For example, $\varphi(x, y) \equiv (S(x, y) \wedge y = x * x)$ is a FO + Poly query over the schema that contains one binary relation S; and $\varphi[D]$ is the set of pairs in (x, y) in D where $y = x^2$.

We now, following [21, 22] say that a $\mathrm{FO}(SC, \Omega)$ query $\varphi(\vec{x})$ is *safe* for a SC-database D if it has finitely many satisfiers for D; that is, $\varphi[D]$ is finite. A query is safe if it is safe for all databases.

As we explained before, we need to distinguish a class of well-behaved models. Following [6, 7, 8], we use *o-minimality* [30, 42] and quantifier-elimination [11] for this purpose. We say that $\mathcal{M}$ is o-minimal, if every definable set is a finite union of points and open intervals $\{x \mid a < x < b\}$, $\{x \mid x < a\}$, and $\{x \mid x > a\}$, (we assume that $<$ is in Ω). Definable sets are those of the form $\{x \mid \mathcal{M} \models \varphi(x)\}$, where φ is a first-order formula in the language of $\mathcal{M}$, possibly supplemented with symbols for constants from $\mathcal{M}$. We say that $\mathcal{M}$ admits *quantifier-elimination (QE)* if, for every formula $\varphi(\vec{x})$, there is an equivalent quantifier-free formula $\psi(\vec{x})$ such that $\mathcal{M} \models \forall \vec{x}.\varphi(\vec{x}) \leftrightarrow \psi(\vec{x})$. Below we list the most important examples, which correspond to classes of interpreted structures and constraints often used in applications.

Linear Constraints: $\langle \mathbb{R}, +, -, 0, 1, < \rangle$ is o-minimal, has QE, and its first-order theory is decidable, cf. [11].

Polynomial Constraints: The real field $\langle \mathbb{R}, +, *, 0, 1, < \rangle$ is o-minimal, has QE, and its first-order theory is decidable. This follows from Tarski's theorem, cf. [11, 42].

Exponential Constraints: $\langle \mathbb{R}, +, *, e^x, 0, 1, < \rangle$ is o-minimal [48] but does not have QE [41].

An example of a structure that is not o-minimal is $\langle \mathbb{N}, +, < \rangle$ as the formula $\exists y(x = y + y)$ defines the set of even numbers. Figure 1 provides examples of some often-encountered structures and their standing with respect to o-minimality and quantifier-elimination.

We will need the following result about o-minimal structures, that will be used numerous times in proofs:

Fact 1 (Uniform Bounds) (see [30]) *If $\mathcal{M}$ is o-minimal, and $\varphi(x, \vec{y})$ is a first-order formula in the language of $\mathcal{M}$, possibly supplemented with symbols for constants from $\mathcal{M}$, then there is an integer K such that, for each vector $\vec{a}$ from $\mathcal{M}$, the set $\{x \mid \mathcal{M} \models \varphi(x, \vec{a})\}$ is composed of fewer than K intervals.*

structure	o-minimal	has quantifier-elimination
$\langle \mathbb{R}, < \rangle$	Y	Y
$\langle \mathbb{R}, +, -, 0, 1, < \rangle$	Y	Y
$\langle \mathbb{R}, +, *, 0, 1, < \rangle$	Y	Y
$\langle \mathbb{R}, +, *, e^x \rangle$	Y	N
$\langle \mathbb{Q}, < \rangle$	Y	Y
$\langle \mathbb{Q}, +, *, 0, 1, < \rangle$	N	N
$\langle \mathbb{N}, < \rangle$	Y	N
$\langle \mathbb{N}, +, 0, 1, \{\equiv_k\}_{k>1}, < \rangle$	N	Y
$\langle \mathbb{N}, +, *, 0, 1, < \rangle$	N	N

Figure 1: Examples of structures

If only quantifiers $\forall x \in adom$ and $\exists x \in adom$ are used in a query, it is called an *active-semantics* query. This is the usual semantics for databases, and it will be the one used in most of the results in this paper. If quantification over the entire infinite universe is allowed, we speak of a *natural-semantics* query. Active-semantics queries admit the standard bottom-up evaluation, while for natural-semantics it is not clear a priori if they can be evaluated at all. However, in many cases one can restrict one's attention to active-semantics queries. The following result was first shown for the pure case (no interpreted structure) in [18] and for linear constraints [28], and then for a large class of structures as follows:

Fact 2 (Natural-Active Collapse) (see [7, 8]) *If $\mathcal{M}$ is o-minimal and has QE, and φ is an arbitrary* $\mathrm{FO}(SC, \Omega)$ *query, then there exists an equivalent* $\mathrm{FO}(SC, \Omega)$ *active-semantics query φ_{act}. Moreover, if the first-order theory of $\mathcal{M}$ is decidable and QE is effective, then the translation $\varphi \to \varphi_{\mathrm{act}}$ is effective.* □

We now define the classes of conjunctive queries (CQ), unions of conjunctive queries (UCQ) and Boolean combination of conjunctive queries (BCCQ) in the interpreted setting. CQs are built up from atomic SC formulae and *arbitrary* Ω-formulae by using $\wedge$ and quantifiers $\exists x$ and $\exists x \in adom$. Note that we can always assume that parameters of each SC relation are variables, as Ω-terms can be eliminated by using existential (active-domain) quantifiers. It is easy to see that each CQ can be represented in the form

$$\varphi(\vec{z}) \;\equiv\; \exists \vec{x} \exists \vec{y} \in adom\ S_1(\vec{u}_1) \wedge \ldots S_k(\vec{u}_k) \wedge \gamma(\vec{x}, \vec{y}, \vec{z})$$

where S_is are schema relations (not necessarily distinct), $\vec{u}_i$ is a vector of variables from $\vec{x}, \vec{y}, \vec{z}$ of appropriate arity, and γ is a Ω-formula. If $\Omega = \emptyset$ and $\vec{x} = \emptyset$, this is the usual notion of conjunctive queries. If γ is quantifier-free, this is the notion of conjunctive queries used in [19].

We define UCQs to be built up from atomic SC formulae and *arbitrary* Ω-formulae by using $\wedge$, $\vee$ and quantifiers $\exists x$ and $\exists x \in adom$. Again, it is easy to see that those are precisely the queries of the form $\varphi_1 \vee \ldots \vee \varphi_k$ where each φ_i is a CQ. Finally, BCCQ are arbitrary Boolean combinations of CQs.

Although we could define active domain versions of conjunctive queries, the results we state here (e.g. decidability of safety) for the more general classes above will automatically imply the corresponding results for the more restricted class of active-domain conjunctive queries.

For background on finitely representable databases, see the beginning of Section 7.

3 Safe translations

The main goal of this section is to show that what kind of interpreted structure one uses does matter, when one studies query safety. As a by-product, we show that the state-safety problem is decidable over certain structures. We study *safe translations*, that is, translations from arbitrary queries into safe ones that do not change the result of a query if it is finite. Formally:

Definition 3.1 *We say that there is a* safe translation *of (active-semantics) first-order queries over $\mathcal{M}$ if there is a function $\varphi \to \varphi_{\text{safe}}$ on (active-semantics) formulae such that for every φ,*

1. *φ_{safe} is safe, and*
2. *if φ is safe for D, then $\varphi[D] = \varphi_{\text{safe}}[D]$.*

A translation is canonical if $\varphi_{\text{safe}}[D] = \emptyset$ whenever φ is not safe on D. A translation is recursive if the function $\varphi \to \varphi_{\text{safe}}$ is recursive. □

It is known that there are domains over which safe translations do not exist, see [36]. The result of [36] uses quantification over the entire universe in an essential way. Here we restrict our attention to active-domain quantification. The following generalizes our result from [8].

Proposition 3.1 *Let $\mathcal{M}$ be o-minimal, based on a dense order, admit effective QE, and have a decidable theory. Then there exists a recursive canonical safe translation of active-semantics formulae over $\mathcal{M}$.*

Proof: Given an active-semantics formula φ, let Let $\alpha(x)$ be a formula defining the active domain of the output of φ. Let Ψ be an active-semantics sentence equivalent to

$$\neg\exists x_1, x_2\ ((x_1 < x_2) \wedge (\forall x\ x_1 < x < x_2 \to \alpha(x)))$$

(it exists by Fact 2). We then define φ_{safe} as $\varphi \wedge \Psi$. The proposition then follows from the following claim: $D \models \Psi$ iff $\varphi[D]$ is finite.

First, assume that $D \models \neg\Psi$; then clearly $\varphi[D]$ is infinite because $<$ is dense. Suppose $D \models \Psi$. We then look at all occurrences of SC predicates in α and replace each of them with a disjunction of tuples. This results in $\alpha'(x)$ in the language of Ω and constants for elements of $\mathcal{U}$; further, $D \models \alpha(a)$ iff $\mathcal{M} \models \alpha'(a)$. Let Ψ' be obtained from Ψ by substituting α' for α. We then have $\mathcal{M} \models \Psi'$. Since $\alpha'(\mathcal{M}) = \{a \mid \mathcal{M} \models \alpha'(a)\}$ is a finite union of points and intervals, and since $\mathcal{M} \models \Psi'$, it follows that no nondegenerate interval is in $\alpha'(\mathcal{M})$. Thus, from o-minimality, we get that $\alpha'(\mathcal{M})$ is a finite union of points. Hence, $\{a \mid D \models \alpha(a)\}$ is finite, thus showing finiteness of $\varphi[D]$. □

Examples of structures satisfying the conditions of Proposition 3.1 are $\langle \mathbb{R}, +, -, 0, 1, < \rangle$ and $\langle \mathbb{R}, +, *, 0, 1, < \rangle$. An immediate corollary to the proposition above is the following:

Corollary 3.1 *Let $\mathcal{M}$ be as in Proposition 3.1. Then the state-safety problem over $\mathcal{M}$ is decidable. That is, given a first-order query φ, and a database D, it is decidable whether $\varphi[D]$ is finite.*

Proof: We showed that the active-semantics sentence Ψ tests if $\varphi[D]$ is finite. □

We next show that safe translations (recursive or not!) need not exist even when one restrict one's attention to active-semantics queries, and all predicates in the signature Ω are computable.

Proposition 3.2 *There is a structure $\mathcal{M} = \langle \mathbb{N}, P \rangle$, where P is a computable predicate, such that there is no safe translation of active-semantics first-order queries over $\mathcal{M}$.*

Proof: Let P be a ternary predicate defined as: $P(i,j,k)$ iff the ith Turing machine on the input j makes at least k moves (assuming some standard encoding of machines and inputs). Consider the schema that consists of a single binary relation U. Assume to the contrary that there is a safe translation over $\mathcal{M}$. Let $\varphi(k) \equiv \exists i,j \in adom\ U(i,j) \wedge P(i,j,k)$, and let $\psi(k)$ be φ_{safe}. Note that ψ is an active-domain formula in the language of U and P. We now show how to use ψ to decide the halting problem.

Suppose we are given the ith machine M_i and the input j. We assume without loss of generality that M_i makes at least one move on j. Define D that consists of a single pair (i,j). Since we know that ψ is safe, we then compute the minimum number l such that $D \not\models \psi(l)$. It is computable since a) it exists, and b) for each k, it is decidable whether $D \models \psi(k)$.

Assume that $D \models \varphi(l)$. Then M_i does not halt on j. Indeed, if M_i halts, then $\varphi[D]$ is finite, and hence $\varphi[D] = \psi[D]$, but we have $l \in \varphi[D] - \psi[D]$. Assume that $D \not\models \varphi(l)$. Then M_i makes $k < l$ moves on j, and thus halts. Hence, $D \models \varphi(l)$ iff M_i halts on j. Since it is decidable whether $D \models \varphi(l)$, we get a contradiction. □

In the remainder of the paper, we concentrate on *well-behaved* structures; typically, o-minimal ones. For computability, we often impose QE and decidability of first-order theory.

4 Range-restriction and safety

Let us informally describe the concept of range-restriction for databases over interpreted structures. It can be seen as a generalization, to arbitrary structures, of the idea of finiteness dependencies [31]. Consider a query $\varphi(x)$ over a database which is a finite set S of real numbers:

$$\varphi(x,y) \ \equiv \ \exists z\ [S(z) \wedge (x > y) \wedge (x > 0) \wedge (x * x = z) \wedge (y + y = z)].$$

This query defines a set of pairs of reals. Clearly, it is safe, as the size of its output is at most twice the size of the input. Moreover, and this is the key observation, from the query φ and any database S, one can compute an upper bound on the output $\varphi[S]$: indeed, every element in $adom(\varphi[S])$ is either $\sqrt{a}$ or $\frac{a}{2}$ for some element $a \in S$. Equivalently, in this upper bound every element is a solution to either $x^2 = a$ or $2x = a$ when a ranges over S. That is, in this example, an upper bound on the result of a safe query can be found as the set of roots of polynomials with coefficients coming from the active domain of the database and a finite set of constants.

This is essentially the idea of range-restriction: we find, for a safe query, a set of formulae defining an upper bound on the output. A similar approach was used in [14], although the focus of [14] is different: it does not consider how the underlying structure affects safety, but instead gives a syntactically restricted class of queries with interpreted functions that can be translated into algebra expressions, along the lines of [46]. In contrast, we are interested in how the underlying structure interacts with queries. For example, we show that not only a set of range-restriction formulae exists

for any query over a well-behaved structure, but also, under additional conditions, it can be found effectively. The result that we prove is actually a bit stronger, as it shows that the upper bound works not only for safe queries, but for arbitrary queries, provided they are safe on a given database.

The first difficulty we encounter is finding an analog of the set of roots of polynomials, when we deal with arbitrary structures (e.g., $\langle \mathbb{R}, +, *, e^x \rangle$). The solution to this is provided by the model-theoretic notion of *algebraic* formulae, reviewed in subsection 4.1. The range-restriction theorem is proved in subsection 4.2. Then, in subsection 4.3, we give two examples: the pure case, where our main result trivially translates into a classical relational theory result, and a much less trivial FO + POLY case, where we confirm the original intuition that the upper bound is a set of roots of polynomials. We finish the section by giving a couple of extensions of the main result.

4.1 Algebraic formulae over o-minimal structures

In this subsection, we study formulae over $\mathcal{M}$, that is, first-order formulae in the language $L(\Omega)$. We shall consider formulae $\varphi(\vec{x}; \vec{y})$ with distinguished parameter variables $\vec{y}$; we use ";" to separate those variables. Assume that $\vec{x}$ is of length n and $\vec{y}$ is of length m. Such a formula (in the language of Ω and constants for elements of $\mathcal{U}$) is called *algebraic* if for each $\vec{b}$ in $\mathcal{U}^m$ there are only finitely many satisfiers of $\varphi(\vec{x}, \vec{b})$; that is, the set $\{\vec{a} \in \mathcal{U}^n \mid \mathcal{M} \models \varphi(\vec{a}; \vec{b})\}$ is finite. A collection of formulae is algebraic if each of its elements is algebraic.

For example, the formula $\varphi(x; y) \equiv (x^2 = y)$ is algebraic over $\langle \mathbb{R}, +, *, 0, 1, < \rangle$. It can be easily seen that if $\varphi_1(\vec{x}; \vec{y})$ and $\varphi_2(\vec{x}; \vec{y})$ are algebraic, then so are $\varphi_1 \wedge \varphi_2$, $\varphi_1 \vee \varphi_2$ and $\exists x \varphi_1$ where x is one of the variables in $\vec{x}$; however, algebraic formulae are not closed under negation.

Let us list some simple properties of algebraic formulae over o-minimal structures.

Lemma 4.1 *Let $\mathcal{M} = \langle \mathcal{U}, \Omega \rangle$ be o-minimal and based on a dense order, and let $\gamma(\vec{x}; \vec{y})$ be algebraic. Then:*

- *There exists a number K such that for any $\vec{b} \in \mathcal{U}^m$, the set $\{\vec{a} \in \mathcal{U}^n \mid \mathcal{M} \models \gamma(\vec{a}; \vec{b})\}$ has fewer than K elements.*
- *There is a recursively enumerable collection of algebraic formulae $\{\chi_i(\vec{x}; \vec{y})\}$ such that every algebraic formula $\gamma(\vec{x}; \vec{y})$ is equivalent to one of the χ_i [1]. If $\mathcal{M}$ is decidable, then the collection of algebraic formulae is recursive.*

Proof: For the first item, consider all formulae $\gamma_i(x_i; \vec{y}) \equiv \exists \vec{x}^{(i)}\ \gamma(\vec{x}; \vec{y})$, where $\vec{x}^{(i)}$ contains all components of $\vec{x}$ except x_i. By Fact 1, there is an integer K_i such that for each $\vec{b}$, $\{a \mid \mathcal{M} \models \gamma_i(a; \vec{b})\}$ is composed of fewer than K_i intervals. Since γ is algebraic and $<$ is dense, it means that $\{a \mid \mathcal{M} \models \gamma_i(a; \vec{b})\}$ has fewer than K_i elements. Hence, there are at most $K = \prod_{i=1}^{n} K_i$ vectors $\vec{a}$ such that $\mathcal{M} \models \gamma(\vec{a}; \vec{b})$.

For the second item, for each $\gamma(\vec{x}; \vec{y})$, let $\chi^k_\gamma(\vec{x})$ be a first-order formula

$$\forall \vec{y} \neg \exists \vec{x}_1 \ldots \exists \vec{x}_k . \bigwedge_{i=1}^{k} \gamma(\vec{x}_i; \vec{y}) \wedge \bigwedge_{i \neq j} (\vec{x}_i \neq \vec{x}_j)$$

[1] Provided that Ω is a recursive set.

saying that there are fewer than k vectors $\vec{x}$ that satisfy $\gamma(\vec{x};\vec{y})$, for all $\vec{y}$. Consider the recursively enumerable collection of formulae of the form $\gamma(\vec{x};\vec{y}) \wedge \chi^k_\gamma(\vec{x})$ where γ ranges over $\mathrm{FO}(\Omega)$ and k ranges over $\mathbb{N}$. It follows from the first item that it enumerates all algebraic formulae. If $\mathcal{M}$ is decidable, it is also decidable whether $\gamma(\vec{x};\vec{y})$ is algebraic. Indeed, the latter happens iff $\gamma^*(x;\vec{y}) = \bigvee_i \gamma_i(x;\vec{y})$ is algebraic (see the proof of the first item), which in turn happens iff there is no open interval contained in $\{a \mid \mathcal{M} \models \gamma^*(a;\vec{b})\}$ for any $\vec{b}$. The latter can be formulated as a first-order sentence $\forall\vec{y}\neg\exists u, v \forall z.(u < z < v \rightarrow \gamma^*(z;\vec{y}))$. □

While we do get an enumeration of algebraic formulae from Lemma 4.1, we need one more representation for algebraic formulae as a tool in proofs. This representation is provided below. We first treat the one-variable case, that is, formulae $\varphi(x;\vec{y})$.

Let $\Xi = \{\xi_1(x;\vec{y}), \ldots, \xi_k(x;\vec{y})\}$ be a collection of formulae. Let

$$\mathrm{same}_\Xi(x, x';\vec{y}) \quad \equiv \quad \bigwedge_{i=1}^{k} (\xi_i(x;\vec{y}) \leftrightarrow \xi_i(x';\vec{y})).$$

Now define

$$\beta_\Xi(x;\vec{y}) \;\equiv\; \forall x', x''.x' < x < x'' \rightarrow (\exists z.x' \leq z \leq x'' \wedge \neg\mathrm{same}_\Xi(x, z;\vec{y})).$$

Proposition 4.1 *Let $\mathcal{M}$ be an o-minimal structure based on a dense order. Then a formula $\varphi(x;\vec{y})$ is algebraic (over $\mathcal{M}$) iff there exists a collection of formulae Ξ such that φ is equivalent to β_Ξ. A formula $\varphi(\vec{x};\vec{y})$ is algebraic iff there exists a collection of formulae Ξ in variables $(x;\vec{y})$ and a formula $\psi(\vec{x};\vec{y})$ such that φ is equivalent to $\beta_\Xi(x_1;\vec{y}) \wedge \ldots \wedge \beta_\Xi(x_n;\vec{y}) \wedge \psi(\vec{x};\vec{y})$.*

Proof. Prove the one-variable case first.

Let Ξ be a collection of formulae, and assume that β_Ξ is not algebraic. That is, for some $\vec{b}$ over $\mathcal{U}$, $\beta_\Xi(\mathcal{M};\vec{b}) = \{a \mid \mathcal{M} \models \beta_\Xi(a;\vec{b})\}$ is infinite. Since $\mathcal{M}$ is o-minimal, $\beta_\Xi(\mathcal{M};\vec{b})$ is a finite union of points and intervals. Since $<$ is dense, it means that there exist $a_0 < b_0 \in \mathcal{U}$ such that $[a_0, b_0] \subseteq \beta_\Xi(\mathcal{M};\vec{b})$. We now consider the formulae $\xi'_i(x) = \xi_i(x;\vec{b})$ for all $\xi_i \in \Xi$. Since both $\xi'_i(\mathcal{M}) = \xi_i(\mathcal{M};\vec{b})$ and $\neg\xi'_i(\mathcal{M}) = \neg\xi_i(\mathcal{M};\vec{b})$ are finite unions of intervals and $<$ is dense, for every non-degenerate interval J, it is the case that either $J \cap \xi'_i(\mathcal{M})$ or $J \cap \neg\xi'_i(\mathcal{M})$ contains an infinite (closed) interval. Using this, we construct a sequence of intervals as follows: $I_0 = [a_0, b_0]$, $I_1 \subseteq I_0$ is an interval that is contained either in $I_0 \cap \xi'_1(\mathcal{M})$ or in $I_0 \cap \neg\xi'_1(\mathcal{M})$. At the jth step, $I_j \subseteq I_{j-1}$ is an interval that is contained either in $I_{j-1} \cap \xi'_j(\mathcal{M})$ or in $I_{j-1} \cap \neg\xi'_j(\mathcal{M})$. Let $I = I_k$. Then, for any $c, d \in I$, $\mathcal{M} \models \xi_i(c,\vec{b}) \leftrightarrow \xi_i(d;\vec{b})$.

Since $I = [a', b'] \subseteq [a_0, b_0]$ and $\mathcal{M} \models \beta_\Xi(c;\vec{b})$ for all $c \in I$, we obtain that, for every $c \in (a', b')$, there exists $d \in [a', b']$ such that $\mathcal{M} \models \neg\mathrm{same}_\Xi(c, d;\vec{b})$. That is, for some $\xi_i \in \Xi$, $\mathcal{M} \models \neg(\xi_i(c;\vec{b}) \leftrightarrow \xi_i(d;\vec{b}))$, which is impossible by construction of I. This proves that β_Ξ is algebraic.

For the converse, we let, for any $\varphi(x;\vec{y})$, Ξ consist of just φ. That is, $\beta_\Xi(x;\vec{y})$ is

$$\forall x', x''.x' < x < x'' \rightarrow (\exists z.x' \leq z \leq x'' \wedge \neg(\varphi(x;\vec{y}) \leftrightarrow \varphi(z;\vec{y}))).$$

We claim that φ and β_Ξ are equivalent, if φ is algebraic. Fix any $\vec{b}$ of the same length as $\vec{y}$, and assume that $\varphi(a;\vec{b})$ holds. If $\beta_\Xi(a;\vec{b})$ does not hold, then there exist $a' < a < a''$ such that for every $c \in [a', a'']$, $\varphi(c;\vec{b}) \leftrightarrow \varphi(a;\vec{b})$ holds; thus, $\varphi(c;\vec{b})$ holds for infinitely many c, contradicting algebraicity of φ. Hence, $\beta_\Xi(a;\vec{b})$ holds. Conversely, assume that $\beta_\Xi(a;\vec{b})$ holds. If $\varphi(a;\vec{b})$ does not hold, then there is an interval containing a on which $\varphi(\cdot;\vec{b})$ does not hold. Indeed, $\neg\varphi(\mathcal{M};\vec{b})$ is a finite union of

intervals, whose complement is a finite set of points, so the above observation follows from the density. We now pick $a' < a''$ such that $\varphi(\cdot;\vec{b})$ does not hold on $[a', a'']$. Since $\beta_\Xi(a;\vec{b})$ holds, we find $c \in [a', a'']$ such that $\neg(\varphi(a;\vec{b}) \leftrightarrow \varphi(c;\vec{b}))$ holds; that is, $\varphi(c;\vec{b})$ holds for $c \in [a', a'']$, which is impossible. Thus, we conclude that $\varphi(a;\vec{b})$ holds, proving that for any $\vec{b}$, $\forall x.\ (\varphi(x;\vec{b}) \leftrightarrow \beta_\Xi(x;\vec{b}))$. This finishes the one variable case.

For the multi-variable case, we note that algebraicity of β_Ξ implies that $\varphi'(\vec{x};\vec{y}) = \beta_\Xi(x_1;\vec{y}) \wedge \ldots \wedge \beta_\Xi(x_n;\vec{y}) \wedge \psi(\vec{x};\vec{y})$ is algebraic. Conversely, let $\varphi(\vec{x};\vec{y})$ be algebraic. Consider

$$\varphi_i(x_i;\vec{y}) = \exists x_1 \ldots \exists x_{i-1} \exists x_{i+1} \ldots \exists x_n . \varphi(x_1, \ldots, x_i, \ldots, x_n;\vec{y}).$$

Let $\chi(x;\vec{y})$ be $\varphi_1(x;\vec{y}) \vee \ldots \vee \varphi_n(x;\vec{y})$. Obviously each φ_i is algebraic, and thus $\chi(x;\vec{y})$ is algebraic. Hence, $\chi(x;\vec{y})$ is equivalent to $\beta_\Xi(x;\vec{y})$ for some finite collection Ξ of formulae in $(x;\vec{y})$. Note that if $\varphi(\vec{a};\vec{b})$ holds and a_i is the ith component of $\vec{a}$, then $\chi(a_i;\vec{b})$ holds and thus $\beta_\Xi(a_i;\vec{b})$ holds. This shows that φ is equivalent to $\beta_\Xi(x_1;\vec{y}) \wedge \ldots \wedge \beta_\Xi(x_n;\vec{y}) \wedge \varphi(\vec{x};\vec{y})$, thus completing the proof. □

Corollary 4.1 *If $\mathcal{M}$ is an o-minimal structure based on a dense order, and $\varphi(\vec{x};\vec{y})$ is algebraic over $\mathcal{M}$, then φ is algebraic over any $\mathcal{M}'$ elementary equivalent to $\mathcal{M}$.*

Proof: There exists a collection of formulae Ξ such that $\mathcal{M} \models \forall\vec{x}\forall\vec{y}.\varphi(\vec{x};\vec{y}) \leftrightarrow (\varphi(\vec{x};\vec{y}) \wedge \bigwedge_i \beta_\Xi(x_i;\vec{y}))$. Hence the same sentence is true in $\mathcal{M}'$. Since $\mathcal{M}'$ is also o-minimal and based on a dense order, we get from Proposition 4.1 that φ is algebraic in $\mathcal{M}'$, too. □

4.2 Main theorem

We start with a few definitions. For a $L(\Omega)$-formulae $\gamma(\vec{x};\vec{y})$ and database D, let

$$\gamma(D) \ = \ \{\vec{a} \mid \exists\vec{b} \in adom(D)^m \text{ such that } D \models \gamma(\vec{a};\vec{b})\}$$

If Γ is a collection of formulae in $\vec{x};\vec{y}$, define

$$\Gamma(D) = \bigcup_{\gamma\in\Gamma} \gamma(D).$$

Note that if Γ is algebraic and finite, then $\Gamma(D)$ is finite.

Definition 4.1 (Range-restriction) *Let $\mathcal{M}$ be an interpreted structure. A* range-restricted query *over $\mathcal{M}$ and a database schema SC is a pair $Q = (\Gamma, \varphi(\vec{x}))$, where Γ is a finite collection $\{\gamma_1(\vec{x};\vec{y}), \ldots, \gamma_m(\vec{x};\vec{y})\}$ of algebraic $L(\Omega)$ formulae, and $\varphi(\vec{x})$ is a $L(SC, \Omega)$ query.*

The semantics of Q is as follows:

$$Q[D] \quad = \quad \{\vec{a} \in \Gamma(D) \mid D \models \varphi(\vec{a})\}.$$

That is, Γ provides an upper bound on the output of a query; within this bound, a usual first-order query is evaluated. For example, let $\varphi(x)$ be the FO + Poly query $S(x) \vee (x > 5)$. Clearly, it is not safe. Let now $\gamma(x;y) \equiv (x * x = y)$ and $Q = (\{\gamma\}, \varphi)$. Then, for any database S (which a finite set of the reals), $Q[S]$ is the set of those elements a such that $a^2 \in S$ and either $a \in S$ or $a > 5$. Clearly, this is a finite set.

Observation *Every range-restricted query is safe.*

We call a range-restricted query (Γ, φ) *active-semantics* if φ is an active-semantics formula. Note that Γ does not mention the database. It turns out that range-restricted active queries characterize all the safe active-semantics queries in the following sense.

Theorem 4.1 *Let $\mathcal{M}$ be any o-minimal structure based on a dense linear order. Then there is a function* Make_Safe *that takes as input an active-domain formula $\varphi(\vec{x})$, and outputs a range-restricted active query $Q = (\Gamma, \psi)$ with the property that* Make_Safe(φ) *is equivalent to φ on all databases D for which φ is safe. Furthermore, if $\mathcal{M}$ has effective quantifier-elimination, then* Make_Safe *is recursive.*

Proof: We deal with the one-variable case first. Let

$$\varphi(z) \;\equiv\; Q_1 w_1 \in adom \ldots Q_1 w_l \in adom.\alpha(z, \vec{w})$$

where each Q_i is $\exists$ or $\forall$ and $\alpha(z, \vec{w})$ is quantifier-free, and all atomic subformulae $R(\cdots)$ contain only variables, excluding z. Any formula can be transformed into such by adding existential quantifiers, cf. [6, 8]. Let $\Xi = \{\xi_i(z, \vec{w}) \mid i = 1, \ldots, k\}$ be the collection of all Ω-atomic subformulae of α. We may assume without loss of generality that the length of $\vec{w}$ is nonzero, and that Ξ is nonempty (if this is not true for φ, take $\varphi \wedge \forall w \in adom(w = w)$ and transform it to the above form).

Define $\text{same}_\Xi(a, b, \vec{w})$, as before, to be $\bigwedge_{i=1}^{k}(\xi_i(a, \vec{w}) \leftrightarrow \xi_i(b, \vec{w}))$, and define $\gamma(x; \vec{w})$ to be $\beta_\Xi(x; \vec{w})$; that is, $\gamma(x, \vec{w}) \;\equiv\; \forall x', x''.x' < x < x'' \rightarrow \exists y.(x' \leq y \leq x'' \wedge \neg\text{same}_\Xi(x, y, \vec{w}))$. Now Γ consists just of γ, with $\vec{w}$ being distinguished parameters. We let Make_Safe(φ) output $(\{\gamma\}, \varphi)$.

Since γ is algebraic by Proposition 4.1, we must show that $\{a \mid D \models \varphi(a)\} = \{a \in \Gamma(D) \mid D \models \varphi(a)\}$ for every nonempty database for which φ is safe.

Assume otherwise; that is, for some nonempty D for which φ is safe, we have $D \models \varphi(a)$ but $a \notin \Gamma(D)$. Let $\vec{c}_1, \ldots, \vec{c}_M$ be an enumeration of all vectors of the length of $\vec{w}$ of elements of the active domain. Note that $M > 0$. Since $a \notin \Gamma(D)$, we have that for each $i = 1, \ldots, M$, there exist a_i', a_i'' such that $a_i' < a < a_i''$ and $\mathcal{M} \models \text{same}_\Xi(a, c, \vec{c}_i)$ for all $c \in [a_i', a_i'']$.

Let $b' = \max\{a_i'\}, b'' = \max\{a_i''\}$. We have $b' < a < b''$, and for each $\vec{c}$ (of length of $\vec{w}$) over the active domain, we have $\xi_i(a; \vec{c}) \leftrightarrow \xi_i(c, \vec{c})$ for every $c \in [b', b'']$. From this, by a simple induction on the structure of the formula (using the fact that z does not appear in any atomic formula $R(\cdots)$), we obtain that $D \models \alpha(a, \vec{c}) \leftrightarrow \alpha(c, \vec{c})$ for every $\vec{c}$ over $adom(D)$ and every $c \in [b', b'']$, and thus $D \models \varphi(a) \leftrightarrow \varphi(c)$, which implies that φ is not safe for D. This contradiction proves correctness of Make_Safe for the one-variable case.

This completes the proof for the one-variable case. We handle the multivariable case by reducing to the one-variable case.

Let $\mathcal{M}' = \langle \mathcal{U}, \Omega' \rangle$ be a definable extension of $\mathcal{M}$ that has quantifier-elimination. Note that $\mathcal{M}'$ is o-minimal. Let $\varphi(z_1, \ldots, z_n)$ be given, and define

$$\varphi_i(z_i) \quad\equiv\quad \exists z_1 \ldots \exists z_{i-1} \exists z_{i+1} \ldots \exists z_n.\varphi(z_1, \ldots, z_{i-1}, z_i, z_{i+1}, \ldots, z_n).$$

By [7, 8], there is a $L(SC, \Omega')$ *active* formula $\psi_i(z_i)$ such that $D \models \forall z.\psi_i(z) \leftrightarrow \varphi_i(z)$ for all D. Let $(\{\gamma_i(z_i, \vec{w}_i)\}, \psi_i(z_i))$ be the output of Make_Safe on ψ_i. Since $\mathcal{M}'$ is a definable extension, we can assume without loss that γ_i is a Ω-formula.

We now define

$$\gamma(\vec{z}; \vec{w}_1, \ldots, \vec{w}_n) \quad\equiv\quad \gamma_1(z_1; \vec{w}_1) \wedge \ldots \wedge \gamma_n(z_n; \vec{w}_n),$$

where each $\vec{w}_i$ is of the same length as the vector of distinguished parameters in the formulae γ_i. Finally, Make_Safe(φ) outputs $(\{\gamma\}, \varphi)$. To see that it works, first notice that algebraicity of all γ_is implies algebraicity of γ. Now assume that $D \models \varphi(\vec{a})$ where $\vec{a} = (a_1, \ldots, a_n)$. Then $D \models \varphi_i(a_i)$, and thus for some vector $\vec{c}_i$ of elements of the active domain, we have that $\gamma_i(a_i, \vec{c}_i)$ holds. Thus, if $\vec{c}$ is the concatenation of all $\vec{c}_i$s, then $\gamma(\vec{a}, \vec{c})$ holds, showing that $\vec{a} \in \Gamma(D)$, where $\Gamma = \{\gamma\}$. This completes the proof of the multivariable case.

We finally notice that Make_Safe for one-variable formulae is recursive, no matter what $\mathcal{M}$ is. For the multivariable case, to make it recursive, we need a procedure for converting natural-quantification formulae into active-quantification formulae. Such a procedure exists by Fact 2. □

Corollary 4.2 (Range-restricted = Safe) *For any o-minimal structure based on a dense order, the class of safe active-semantics queries is the same as the class of range-restricted queries.* □

Combining this with the natural-active collapse (Fact 2), we obtain:

Corollary 4.3 *Let $\mathcal{M}$ be any o-minimal structure based on a dense linear order that admit QE. Then there is a function* Make_Safe *(recursive if so is QE and $\mathcal{M}$ is decidable) that takes as input a natural-semantics formula $\varphi(\vec{x})$, and outputs a range-restricted active query $Q = (\Gamma, \psi)$ with the property that* Make_Safe(φ) *is equivalent to φ on all databases D for which φ is safe. In particular, over $\mathcal{M}$, the classes of safe natural-semantics queries and range-restricted queries coincide.* □

Corollary 4.4 *For any o-minimal structure based on a dense order (decidable or not), the collection of safe queries is recursively enumerable.* □

We finish this section with a proposition that follows from the special form of formulae in Γ, as established in the proof of Theorem 4.1.

Proposition 4.2 *Let $\mathcal{M}$ be o-minimal and based on a dense order. Let $\varphi(\vec{x})$ be a first-order query. Then there exists a set Γ of algebraic formulae $\gamma(x; \vec{y})$ (that can be effectively constructed in $\mathcal{M}$ has effective QE and is decidable) such that, for any database D, if $\varphi[D]$ is finite, then $adom(\varphi[D]) \subseteq \Gamma(D)$.*

Proof: Assume φ is active-semantics. Then the proof follows the proof of Theorem 4.1, but at the end we replace $\gamma(x_1, \ldots, x_n; \vec{y})$ by

$$\gamma'(x; \vec{y}) \;=\; \bigvee_{i=1}^{n} \exists x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n \; \gamma(x_1, \ldots, x_{i-1}, x, x_{i+1}, \ldots, x_n; \vec{y})$$

to get a bound on the active domain, and output $(\{\gamma'\}, \varphi)$.

For an arbitrary φ, let $\mathcal{M}'$ be a definitional expansion of $\mathcal{M}$ that has QE (such an expansion always exists, cf. [11]). Since $\mathcal{M}$ and $\mathcal{M}'$ are elementary equivalent, $\mathcal{M}'$ is o-minimal [30] and the order is dense. Thus, there exists an active-semantics FO($\mathcal{M}', SC$) query $\psi(\vec{x})$ that is equivalent to φ (see Fact 2), and, by the above, we have a formula γ_0 in the language of $\mathcal{M}'$ such that $adom(\varphi[D]) = adom(\psi[D]) \subseteq \gamma_0(D)$. Now obtain a $L(\Omega)$ formula γ from γ_0 by replacing each new predicate symbol from $\mathcal{M}'$ by its definition by a $L(\Omega)$ formula. Then $\gamma(D) = \gamma_0(D)$, which proves the proposition. □

4.3 Examples

Below we give two examples – we consider the pure case, where Theorem 4.1 translates into a well known result; and we also consider the case of polynomial constraints and show a more explicit form of range-restriction.

The pure case We assume that we have the pure relational calculus; that is, our underlying structure is $\mathcal{M}^{\emptyset} = \langle \mathcal{U}, \emptyset \rangle$. The reason we can apply Theorem 4.1 is that we can extend $\mathcal{M}$ to $\mathcal{M}^{<} = \langle \mathcal{U}, < \rangle$ by adding a dense order without endpoints on $\mathcal{U}$; although it is never mentioned in queries, $\mathcal{M}^{<}$ is o-minimal (and has QE), and hence the results are applicable. Next, we need:

Lemma 4.2 *Let $\gamma(x; \vec{y})$ be algebraic over $\mathcal{M}^{<}$. If $\mathcal{M}^{<} \models \gamma(a; \vec{b})$, then a coincides with one of $\vec{b}$'s components.*

Proof: Assume not and fix a counterexample $a, \vec{b}$ and let B be the set of components of $\vec{b}$ together with $-\infty, \infty$. Let b', b'' in B be such that $b' < a < b''$ and (b', b'') contains no other member of B. Since $\mathcal{M}^{<}$ has QE, γ is equivalent to a quantifier-free formula. For any $a' \in (b', b'')$, $(a, \vec{b})$ and $(a', \vec{b})$ satisfy the same atomic $<$-formulae, and hence $\mathcal{M}^{<} \models \gamma(a, \vec{b}) \leftrightarrow \gamma(a', \vec{b})$, which contradicts algebraicity. □

This immediately implies that for an algebraic Γ and a database D, $\Gamma(D)$ is either empty (if Γ contains no formulae) or $\Gamma(D) = adom(D)$. Thus, the classes of safe pure relational calculus queries and relational calculus queries whose output is restricted to the active domain, coincide. Of course, this is a standard result in database theory, proved here in a rather unusual way.

The real field: FO + Poly Can we find a more concrete representation of range-restricted queries over the real fieldΓ Intuitively, it should be sufficient to look for roots of polynomials $p(x, \vec{a})$ where $\vec{a}$ ranges over tuples of elements of the active domain, as was suggested by the example in the beginning of the section. However, even quantifier-free algebraic formulae do not give us directly this representation. Nevertheless, the following can be shown.

Let $p(x, \vec{y})$ be a multivariate polynomial over the real field. Define $Root(p, \vec{a})$ as $\emptyset$ if $p(x, \vec{a})$ is identically zero, and the set of roots of $p(x, \vec{a})$ otherwise. Given a collection P of polynomials $\{p_1(x, \vec{y}), \ldots, p_m(x, \vec{y})\}$ and a database D, let

$$P(D) \;=\; \bigcup_{i=1}^{m} \; \bigcup_{\vec{a} \subseteq adom(D)} Root(p_i, \vec{a})$$

where $\vec{a}$ ranges over tuples of elements of $adom(D)$, of the same length as $\vec{y}$.

Definition 4.2 *A query in* polynomial range-restricted *form is a pair (P, φ), where P is a finite collection of multivariate polynomials, and $\varphi(x_1, \ldots, x_n)$ is a* FO + Poly *query. The semantics is defined as $(P, \varphi)[D] = \varphi[D] \cap P(D)^n$.*

Proposition 4.3 *The class of safe* FO + Poly *queries (arbitrary or active-semantics) coincides with the class of queries in polynomial range-restricted form. Moreover, for every* FO + Poly *query φ,*

a collection of polynomials P can be effectively found such that φ and (P, φ) are equivalent on all databases on which φ is safe.

Proof: Given a query $\varphi(\vec{x})$, find effectively a collection of algebraic formulae $\Gamma = \{\gamma_j(x; \vec{y})\}$ such that for any D for which φ is safe, $adom(\varphi[D]) \subseteq \Gamma(D)$. For each $\vec{a}$ and each $\gamma \in \Gamma$, the set $\gamma[\vec{a}] = \{c \mid \gamma(c; \vec{a})\}$ is finite, and by o-minimality there is a uniform bound M such that $card(\gamma[\vec{a}]) < M$ for all $\vec{a}$ and $\gamma \in \Gamma$.

Now let $\gamma_j^i(x; \vec{y})$, $i < M$, be defined as follows: $\mathcal{M} \models \gamma_j^i(c; \vec{a})$ if either (1) $\gamma_j[\vec{a}]$ has at least i elements, and c is the ith element in the order $<$, or (2) $\gamma_j[\vec{a}]$ is nonempty, has fewer than i elements, and c is the largest element of $\gamma_j[\vec{a}]$, or (3) $\gamma_j[\vec{a}]$ is empty, and $c = 0$. Note that $\gamma_j^i(x; \vec{y})$'s are indeed $L(+, *, 0, 1, <)$ formulae. It is easy to see that each $\gamma_j^i(x; \vec{y})$ defines a function $f_{ij} : \mathbb{R}^m \to \mathbb{R}$, where m is the length of $\vec{y}$, by $f_{ij}(\vec{a}) = c$ iff c is the unique element such that $\gamma_j^i(c; \vec{a})$ holds. Furthermore, this function is semialgebraic and the following property holds: if φ is safe for D, then $adom(\varphi[D])$ is contained in $\bigcup_{i,j} \bigcup_{\vec{a}} f_{ij}(\vec{a})$, where $\vec{a}$ ranges over $adom(D)$.

It follows from [23] that each $f_{ij}(\vec{y})$ is algebraic, that is, there exists a polynomial $p_{ij}(x, \vec{y})$ such that $p_{ij}(x, \vec{y}) = 0$ iff $x = f_{ij}(\vec{y})$. It is easy to see that for $P = \{p_{ij} \mid i < M, \gamma_j \in \Gamma\}$, $\Gamma(D) \subseteq P(D)$ and $P(D)$ is always finite. To complete the proof, we must show effectiveness. We can effectively construct Γ, and thus find M (by writing formulae saying that each $\gamma(x; \vec{y})$ has fewer than M satisfiers for each $\vec{y}$, and checking if it is true by applying QE; since it is true for some M, the process terminates). Hence, we can effectively construct γ_j^is, and the procedure for finding p_{ij}s is effective (although not stated in [23], it follows from the analysis of the proof there). □

4.4 Extensions

Suppose we are given two elementary equivalent structures $\mathcal{M}$ and $\mathcal{M}'$, for example, $\langle \mathbb{R}, +, < \rangle$ and $\langle \mathbb{Q}, +, < \rangle$. Assuming $\mathcal{M}$ is o-minimal based on a dense order, so is $\mathcal{M}'$, and thus the characterization of a safe queries applies to $\mathcal{M}'$ as well. However, we would like to know more. Suppose φ is safe over $\mathcal{M}$. Is the same φ safe over $\mathcal{M}'\Gamma$ The positive answer is provided for o-minimal structures.

Proposition 4.4 *If $\mathcal{M}$ is an o-minimal structure based on a dense order, and φ is a safe active-semantics query, then φ is safe in any structure elementary equivalent to $\mathcal{M}$.*

Proof: Let $\varphi(\vec{z})$ be safe, and let $(\{\gamma(\vec{z}; \vec{w})\}, \varphi)$ be the output of Make_Safe (we just saw that it always can be made to have this form). Since φ is safe, we obtain that for every database D over $\mathcal{M}$ and for every $\vec{a}$ of the same length as $\vec{z}$ of elements of $\mathcal{M}$, $D \models \varphi(\vec{a}) \to \exists \vec{w} \in adom. \gamma(\vec{a}; \vec{w})$. Now putting $\neg\varphi$ in prenex form where only quantified variables appear inside the schema predicates (which can always be done), we have

$$D \models \exists \vec{w} \in adom. Q_1 y_1 \ldots Q_n y_n. (\alpha(\vec{a}, \vec{y}) \vee \gamma(\vec{a}; \vec{w}))$$

where all quantification $Q_i y_i$ is active, and α is the quantifier-free part of $\neg\varphi$. We now claim that the same is true in $\mathcal{M}'$ for every D over $\mathcal{M}'$ and every $\vec{a}$ of elements of $\mathcal{M}'$, as long as $\mathcal{M}'$ is elementary equivalent to $\mathcal{M}$. Note that this would imply that $D \models \varphi(\vec{a}) \to \exists \vec{w} \in adom. \gamma(\vec{a}; \vec{w})$ holds in $\mathcal{M}'$ as well.

To prove this claim, assume to the contrary that this fails for some D and $\vec{a}$ over $\mathcal{M}'$. Let $\{b_1, \ldots, b_m\}$ be $adom(D)$. We now define a sequence of formulae as follows: $\chi_n(\vec{a}, \vec{b}, \vec{w}, y_1, \ldots, y_{n-1}) = \bigvee_{i=1}^m \alpha(\vec{a}, y_1, \ldots, y_{n-1}, b_i) \vee \gamma(\vec{a}; \vec{w})$ if Q_n is $\exists$; if Q_n is $\forall$, we change $\bigvee$ to $\bigwedge$. Similarly,

$\chi_j(\vec{a},\vec{b},\vec{w},y_1,\ldots,y_{j-1}) = \bigvee_{i=1}^{m}\chi_{j+1}(\vec{a},\vec{b},\vec{w},y_1,\ldots,y_{j-1},b_i)$ if Q_j is $\exists$; otherwise change $\bigvee$ to $\bigwedge$. Next, define $\chi'(\vec{a},\vec{b})$ as $\bigvee_{\vec{w}\in B}\chi_1(\vec{a},\vec{b},\vec{w})$ where B is the collection of all vectors from $\{b_1,\ldots,b_m\}$ of the same length as $\vec{w}$. Notice that the only occurrence of the schema predicates is of the form $R(\cdot)$, where we list some of b_is as parameters. We finally replace those by *true* or *false*, depending on whether a particular tuple is in the database D. This results in a formula $\chi(\vec{a},\vec{b})$ that does not mention the schema predicates and has the property that $\mathcal{M}' \models \chi(\vec{a},\vec{b})$ iff $\exists\vec{w}\in adom.Q_1y_1\ldots Q_ny_n.(\alpha(\vec{a},\vec{y})\vee\gamma(\vec{a};\vec{w}))$ holds for $\vec{a}$ and the given D with the active domain $\{b_1,\ldots,b_m\}$. Now the assumption gives us that $\mathcal{M}' \models \exists\vec{a}\exists\vec{b}.(\neg\chi(\vec{a},\vec{b})\wedge\bigwedge_{i\neq j}\neg(b_i=b_j))$, thereby showing that the same sentence is true in $\mathcal{M}$. Then, by picking $\vec{a}$ and $\vec{b}$ in $\mathcal{M}$ that witness the failure of χ and defining a database D' on $\vec{b}$ in the same way as D was defined, we see, by an argument similar to the one above, that $D' \models \neg\chi'(\vec{a},\vec{b})$ and thus $\exists\vec{w}\in adom.Q_1y_1\ldots Q_ny_n.(\alpha(\vec{a},\vec{y})\vee\gamma(\vec{a};\vec{w}))$ fails in D', which is impossible. This finishes the proof of the claim.

Thus, $D \models \varphi(\vec{a}) \rightarrow \exists\vec{w}\in adom.\gamma(\vec{a};\vec{w})$ holds in $\mathcal{M}'$, and all we need to show to prove safety of φ is that γ is algebraic in $\mathcal{M}'$. From Proposition 4.1 we know that $\forall\vec{z}\forall\vec{w}.\gamma(\vec{z};\vec{w}) \leftrightarrow (\bigwedge_i \beta_\Xi(z_i;\vec{w})\wedge\gamma(\vec{z};\vec{w}))$ holds in $\mathcal{M}$ for an appropriately chosen Ξ, and thus in $\mathcal{M}'$. Since both o-minimality and density are preserved under elementary equivalence, we get that β_Ξ is algebraic in $\mathcal{M}'$, and hence γ is algebraic in $\mathcal{M}'$, too. □

5 Deciding safety of conjunctive queries and relatives

Safety of arbitrary calculus queries is undecidable even in the pure case [47], and of course it remains undecidable when interpreted functions are present. The main goal of this section is to show that safety is *decidable* for Boolean combinations of conjunctive queries in the presence of an interpreted structure such as $\langle\mathbb{R},+,*,0,1,<\rangle$. In particular, safety of FO + Poly and FO + Lin conjunctive queries is decidable.

Recall that we are using CQ, UCQ and BCCQ for conjunctive, unions of conjunctive, and Boolean combinations of conjunctive queries (see Section 2 for their definition in the presence of an interpreted structure). Conjunctive queries, having dominated early research in relational theory because of their nice properties, resurfaced recently in a number of new applications, cf. [24, 19, 40]. Our proof will be by reduction to the containment problem, which is decidable for UCQs over certain structures. (Of course, *without* an interpreted structure, this is well known [35], as is the decidability of safety for BCCQs, cf. [1]). Note that CQs and UCQs are monotone (that is, $D \subseteq D'$ implies $\varphi[D] \subseteq \varphi[D']$). Since there are nonmonotone BCCQs, the class of UCQs is strictly contained in the class of BCCQs.

The main result is:

Theorem 5.1 *Let $\mathcal{M}$ be o-minimal, based on a dense order, decidable, and admit effective QE. Then it is decidable if a given BCCQ $\varphi(\vec{x})$ over $\mathcal{M}$ is safe.*

The proof is contained in the following two lemmas, which are of independent interest, and will be used later in Section 7. Recall that by containment $\varphi \subseteq \psi$ we mean $\varphi[D] \subseteq \psi[D]$ for any D.

Lemma 5.1 *Let $\mathcal{M}$ be o-minimal and based on a dense order, and $\varphi(\vec{x})$ be a first-order query. Then there exists an active-semantics CQ $\psi(\vec{x})$ such that φ is safe iff $\varphi \subseteq \psi$.*

Proof follows from Proposition 4.2: take $\Gamma = \{\gamma_1(x;\vec{y}), \ldots, \gamma_k(x;\vec{y})\}$ given by the proposition; let $\gamma = \bigvee_i \gamma_i$ and let $\psi(x_1, \ldots, x_n)$ be

$$\exists \vec{y}_1 \in adom \ldots \exists \vec{y}_n \in adom\ \gamma(x_1;\vec{y}_1) \wedge \ldots \wedge \gamma(x_n;\vec{y}_n)$$

If $\varphi \subseteq \psi$, then φ is safe since all γ_is are algebraic. If φ is safe, then $adom(\varphi[D]) \subseteq \Gamma(D)$ for every D, which implies $\varphi \subseteq \psi$. □

Lemma 5.2 *Let $\mathcal{M}$ be as in Theorem 5.1. Then containment of a BCCQ in a UCQ is decidable; that is, for a BCCQ $\varphi(\vec{x})$ and a UCQ $\psi(\vec{x})$ it is decidable if $\varphi \subseteq \psi$. This continues to hold if both φ and ψ are active-semantics queries.*

Proof. We start with the following:

Claim 5.1 *Given φ and ψ, one can effectively find a number k such that $\varphi \subseteq \psi$ iff for every database D with at most k tuples, $\varphi[D] \subseteq \psi[D]$.*

This clearly implies the result, as the latter condition can be expressed as a $L(\Omega)$ sentence. For example, if the schema contains one relational symbol S, this sentence is $\forall \vec{x}_1 \ldots \vec{x}_k \forall \vec{x}.\ \varphi(\vec{x})[\{\vec{x}_i\}/D] \rightarrow \psi(\vec{x})[\{\vec{x}_i\}/D]$ where $\varphi(\vec{x})[\{\vec{x}_i\}/D]$ is obtained from $\varphi(\vec{x})$ by replacing each occurrence of $S(\vec{z})$ by $\bigvee_i(\vec{z} = \vec{x}_i)$, and similarly for an arbitrary schema. The decidability of $\mathcal{M}$ now implies the lemma.

The proof of the claim proceeds similarly to [19]. Note that every BCCQ α can be represented as $\bigvee_{i=1}^{n}(\chi_i(\vec{x}) \wedge \neg\xi_i(\vec{x}))$ where each χ_i is CQ and each ξ_i is UCQ; this follows if one writes α as a DNF. We take k to be the maximum length of χ_i (measured as the sum of the number of atomic formulae and the number of quantified variables).

Assume that $\varphi[D] \not\subseteq \psi[D]$ for some D; that is, we have $\vec{a} \in \varphi[D] \not\subseteq \psi[D]$. Assume $D \models \chi_i(\vec{a}) \wedge \neg\xi_i(\vec{a})$ and let $\chi_i(\vec{x}) = \exists \vec{y} \exists \vec{z} \in adom\ \bigwedge_{j=1}^{l} \alpha_j(\vec{x},\vec{y},\vec{z})$. Then, for some $\vec{b}$ over $\mathcal{U}$ and $\vec{c}$ over $adom(D)$, we get $D \models \bigwedge_{j=1}^{l} \alpha_j(\vec{a},\vec{b},\vec{c})$. Consider those α_js which are SC-formulae. For each such α_j, which is of the form $R(\cdots)$ where $R \in SC$, there is a tuple in D that satisfies it. Select one such tuple for each SC-atomic α_j, and let D' be D restricted to those tuples. Choose a set of at most $length(\vec{c})$ tuples in D contain all the components of $\vec{c}$, and add it to D'. Let the resulting database be D''. Clearly, it has at most k tuples.

Note that $D'' \models \bigwedge_{j=1}^{l} \alpha_j(\vec{a},\vec{b},\vec{c})$, and thus $D'' \models \chi_i(\vec{a})$ since $\vec{c} \subseteq adom(D'')$. On the other hand, $D'' \models \neg\xi_i(\vec{a})$ by monotonicity of ξ_i. Thus, we get that $\vec{a} \in \varphi[D''] - \psi[D'']$ where D'' has at most k tuples. This implies that each counterexample to containment is witnessed by a $\leq k$-element database, and finishes the proof Lemma 5.2. □

To complete the proof of Theorem 5.1, note that under the assumption on $\mathcal{M}$, the CQ ψ such that $\varphi \subseteq \psi$ is equivalent to safety of φ can be effectively constructed – this follows from the procedure for constructing Γ given in Proposition 4.2. The theorem now follows from Lemma 5.2. □

Corollary 5.1 *It is decidable whether any Boolean combination of* FO+Lin *or* FO+Poly *conjunctive queries is safe.* □

Note, however, that safety of CQs is not decidable over every structure. For example, for $\langle \mathbb{N}, +, *, 0, 1, < \rangle$, decidability of CQ safety would imply decidability of checking whether a Diophantine equation has finitely many solutions, which is known to be undecidable [12].

6 Dichotomy theorem and outputs of queries

The main result of this section is a simple but powerful combinatorial structure theorem, saying that over a well-behaved structure, outputs of safe queries cannot grow arbitrarily large in terms of the size of the input. In fact, we prove a *dichotomy* result: either a query φ is not safe on D, or $\varphi[D]$ is at most polynomial in the size of D, where the bounding polynomial depends only on φ. This result shows tame behavior of relational calculus queries over some important interpreted structures, in particular those giving rise to linear, polynomial and exponential constraints. It can be used to show negative results, that is, new expressivity bounds, as well as positive results: the dichotomy theorem is a key ingredient in the decidability results of the next section.

We use the notation $\mathsf{size}(D)$ for the size of a database, measured here as the number of tuples. It can equivalently be measured as the cardinality of the active domain, or the number of tuples multiplied by their arity, and all the results will hold.

Theorem 6.1 (Dichotomy Theorem) *Let $\mathcal{M}$ be o-minimal and based on a dense order. Let $\varphi(\vec{x})$ be a first-order query. Then there exists a a polynomial $p_\varphi : \mathbb{R} \to \mathbb{R}$ such that, for any database D, either $\varphi[D]$ is infinite, or $\mathsf{size}(\varphi[D]) \leq p_\varphi(\mathsf{size}(D))$.*

Proof: Consider Γ given by Proposition 4.2. Since each $\gamma \in \Gamma$ is algebraic, there exists, by Lemma 4.1, a number c_γ such that $card(\{x \mid \gamma(x;\vec{y})\}) < c_\gamma$ for every $\vec{y}$. Thus, if $adom(\varphi[D])$ is finite, then its cardinality is at most

$$\sum_{\gamma \in \Gamma} c_\gamma \cdot n^{m_\gamma}$$

where m_γ is the number of $\vec{y}$ variables in γ, and n is the size of the active domain of D. This is clearly bounded by $card(\Gamma) \cdot C_\Gamma \cdot n^{M_\Gamma}$, where $C_\Gamma = \max c_\gamma$ and $M_\Gamma = \max m_\gamma$.

Now notice that for every schema SC, there exist constants $c_0, d_0, c_1 > 0$ such that $c_1 \cdot card(adom(D)) \leq \mathsf{size}(D) \leq c_0 \cdot card(adom(D))^{d_0}$ for every D. Hence, for appropriately chosen $c_0, d_0 > 0$, we obtain that if $\varphi[D]$ is finite, then

$$\mathsf{size}(\varphi[D]) \leq c_0 \cdot card(adom(\varphi[D]))^{d_0} \leq c_0 \cdot (card(\Gamma) \cdot C_\Gamma \cdot n^{M_\Gamma})^{d_0} \leq cn^d,$$

where $n = card(adom(D))$ and c, d are constant that depend only on φ. Hence, for some $c_1 > 0$ that depends on the schema only, we have $\mathsf{size}(\varphi[D]) \leq c \cdot (\frac{n}{c_1})^d$, which proves the theorem. □

Are the assumptions on a structure important for the dichotomy resultΓ That is, can we find structures over which it failsΓ The following is a simple counterexample to the dichotomy theorem: Let $\mathcal{M} = \langle \mathbb{N}, < \rangle$. Let $\varphi(x)$ be the following active-semantics query: $\exists y \in adom. x < y$. Clearly, $\varphi(x)$ is safe, but $\mathsf{size}(\varphi[D])$ can be arbitrarily large even for a database whose active domain consists of just one element.

The dichotomy theorem can also be stated in terms of a function measuring the growth of the output size. Formally, given a query φ, define $growth_\varphi : \mathbb{N} \to \mathbb{N} \cup \{\infty\}$ as

$$growth_\varphi(n) = \max\{\mathsf{size}(\varphi[D]) \mid \mathsf{size}(D) = n\}.$$

Corollary 6.1 *Let $\varphi(\vec{x})$ be a* FO + Lin, *or* FO + Poly, *or* FO + Exp *query. Then there exists a polynomial p_φ such that, for every $n \in \mathbb{N}$, either $growth_\varphi(n) = \infty$, or $growth_\varphi(n) \leq p_\varphi(n)$.* □

Note that for the query $\varphi(x)$ over $\langle \mathbb{N}, < \rangle$, which we used as a counterexample to the dichotomy theorem, $growth_\varphi(n) = \infty$ for all $n > 0$. Thus, we have a question whether corollary 6.1 fails over some structures. The following proposition provides an example.

Proposition 6.1 *Let $\mathcal{M} = \langle \mathbb{N}, +, <, 1 \rangle$. Then there exists an active-semantics first-order query $\varphi(x)$ over $\mathcal{M}$ such that $growth_\varphi(n) = 2^n$ for every $n > 0$.*

Proof: Let SC consist of one unary relation S. We show that there exists a $\mathrm{FO}(\mathcal{M}, SC)$ sentence Ψ such that $S \models \Psi$ iff S is of the form $S_n = \{2^i \mid 1 \leq i \leq n\}$. This is done by letting Ψ be

$$\begin{array}{ll} & (\exists x \in adom. x = 1 + 1 \wedge S(x)) \\ \wedge & (\forall x \in adom. x = 1 + 1 \vee x > 1 + 1) \\ \wedge & (\forall x \in adom. x = 1 + 1 \vee \exists y \in adom. y + y = x) \\ \wedge & (\forall x \in adom. (\forall y \in adom. y < x \vee y = x) \vee (\exists y \in adom. y = x + x)) \end{array}$$

Now define $\varphi(x)$ as $\Psi \wedge \neg(x < 1) \wedge (\exists y \in adom. x < y \vee x = y)$. Then, for S not of the form S_n, we have $\varphi[S] = \emptyset$, and $\varphi[S_n] = \{1, 2, 3, \ldots, 2^n\}$. Since $card(S_n) = n$, this implies $growth_\varphi(n) = 2^n$ for $n > 0$. □

The dichotomy theorem gives easy expressivity bounds based on the growth of the output size, in fact, sometimes somewhat surprising ones: even if we use exponentiation, we still cannot express any queries with superpolynomial growth. For example, consider the following query Q: given a binary relation S containing $n+1$ distinct points $x_0, x_1, \ldots, x_n$ on a plane, return the vertices of the projection of an n-dimensional cube $[0,1]^n$, where the edges along the axes are projected onto $x_0\vec{x}_1, \ldots, x_0\vec{x}_n$. It is easy to see that for each fixed n, this query is expressible in FO + Poly. As a consequence of the dichotomy theorem, we conclude that Q cannot be expressed uniformly for all n even as a FO + Exp query.

For monotone queries, we can do better. We prove a trichotomy theorem that provides us with a lower bound as well. Recall that monotone queries are those for which $D_1 \subseteq D_2$ implies $\varphi[D_1] \subseteq \varphi[D_2]$. For example, any union of conjunctive queries is monotone.

Theorem 6.2 *Let $\mathcal{M}$ be o-minimal based on a dense order. Then, for each monotone query $\varphi(\vec{x})$, there exist two polynomials p^1_φ and p^2_φ such that either $growth_\varphi$ is bounded by a constant, or, for every n, either $p^1_\varphi(n) \leq growth_\varphi(n) \leq p^2_\varphi(n)$, or $growth_\varphi(n) = \infty$.*

Proof: In view of the previous theorem, it remains to show that if $growth_\varphi$ is not bounded by a constant, then it is bounded below by a polynomial. Assuming $growth_\varphi$ is not bounded by a constant, we get a family of databases $\{D_i\}_{i \in \mathbb{N}}$ such that $\mathsf{size}(\varphi[D_i]) > i$ for all i. Because of monotonicity, we can ensure, by adding elements to D_i, that $\mathsf{size}(D_i) \geq i$.

In the proof of Lemma 5.2 we showed that there exists a constant k that depends on φ only, such that $\vec{a} \in \varphi[D]$ iff there is $D' \subseteq D$ with at most k tuples such that $\vec{a} \in \varphi[D']$. Thus, there is $D'_i \subseteq D_i$ with at most $k(i+1)$ tuples such that $\mathsf{size}(\varphi[D'_i]) > i$.

Now, for a given n, let i be the maximal such that $k(i+1) \leq n$. Consider D'_i and extend it to contain exactly n tuples. For the resulting D''_i, we have $\mathsf{size}(\varphi[D'_i]) > i$ by monotonicity; hence $growth_\varphi(n) > i$. Since $n \leq k(i+1)$, we get from this that $growth_\varphi(n) \geq \frac{n}{k} - 1$, which completes the proof. □

It also follows from the proof that the lower polynomial bound does not require o-minimality. This gives us some new expressivity bounds. It is possible to find first-order queries with $growth_\varphi = O(f(n))$

for many non-polynomial functions f. For example, consider a schema consisting of one unary relation X and one binary relation E, and a sentence Ψ saying that E codes the powerset of X; that is, the family of sets $X_a = \{y \mid E(y,a)\}$ is exactly the family of all nonempty subsets of X, when a ranges over the second projection of E. Such a sentence Ψ can be defined in first-order logic, cf. [1]. We now let $\varphi(x) \equiv X(x) \wedge \Psi$; it then follows that $growth_\varphi = O(\log n)$. Similarly, one can find queries with $growth_\varphi = O(\sqrt[k]{n})$ for any constant k. The trichotomy theorem says that such queries cannot be defined as monotone queries (e.g., unions of conjunctive queries) over *any* interpreted structure.

7 Preserving geometric properties of constraint databases

In this section, we switch from the finite world to the infinite; that is, we deal with constraint databases that represent potentially infinite objects. The notion of safety over constraint databases is different: we are interested in identifying languages that guarantee *preservation of certain geometric properties.*

To give a very simple example, assume that spatial objects stored in a database are convex polytopes in $\mathbb{R}^n$. A simple query "return the convex hull of all the vertices x with $\| x \| < 1$" does always return a convex polytope. This query must be written in a rather expressive language: it can be expressed in FO + Poly but not FO + Lin [43]. Now, our question is: can we ensure in some way that a class of FO + Poly programs preserves a given property, like being a convex polytope? That is, can we find an effective syntax for the class of queries that preserve certain geometric properties?

For FO + Poly and the class of databases definable with linear constraints (semi-linear databases), [13] gave a solution, based on deciding semi-linearity by a FO + Poly query. The resulting language is not quite natural, and [13] posed a problem of finding natural languages that capture queries with certain preservation properties. Our first goal here is to present a general scheme, different from the decidability approach, for enumerating such queries in $\mathrm{FO}(\mathcal{M})$. Here $\mathcal{M}$ is some structure on the reals, not necessarily $\langle \mathbb{R}, +, *, 0, 1, < \rangle$. The approach is based on reduction to the finite case, and using our results about finite query safety. A similar approach was used in [37], where a coding was applied to reduce certain questions about ordered constraint databases to ones about finite databases.

As it often happens, the general case is solved rather easily, and gives us a pleasant characterization of queries preserving geometric properties, but working out the details of important motivating examples is a painful process. We do so for three properties: being a convex polytope, a convex polyhedron, and a compact semi-linear set in $\mathbb{R}^2$ (the latter are perhaps the most often encountered class of constraint databases).

We then use our characterizations together with the dichotomy theorem of the previous section to show a somewhat surprising result that for unions of conjunctive FO + Poly queries, it is *decidable* whether they preserve convex polytopes or compact semi-linear sets in $\mathbb{R}^2$.

To define a general framework for talking about queries that preserve geometric properties, we recall some basic definitions on constraint (or finitely representable) databases. As before, we have a language of some underlying structure $\mathcal{M}$ and a schema SC, but now m-relations in SC are given by quantifier-free formulae[2] $\alpha(x_1, \ldots, x_m)$ in $L(\Omega)$. If $\mathcal{M}$ is $\langle \mathbb{R}, +, -, 0, 1, < \rangle$, then sets so defined are called *semi-linear*; for $\langle \mathbb{R}, +, *, 0, 1, < \rangle$ they are called *semi-algebraic*, cf. [42]. The query languages for constraint databases are the same as those we considered for finite ones: $\mathrm{FO}(SC, \Omega)$.

[2]Without loss of generality, we do not assume relational attributes, as in [13, 27] and some other papers. They do not affect our results, but would make notation heavier.

If $\mathcal{M} = \langle \mathcal{U}, \Omega \rangle$ is an infinite structure, let $\mathsf{Obj}(\Omega)$ be the class of finitely representable databases over $\mathcal{M}$, that is, $\mathsf{Obj}(\Omega) = \bigcup_{n<\omega} \mathsf{Obj}_n(\Omega)$ and $\mathsf{Obj}_n(\Omega)$ is the collection of subsets of $\mathcal{U}^n$ of the form $\{(x_1, \ldots, x_n) \mid \mathcal{M} \models \alpha(x_1, \ldots, x_n)\}$ where α is quantifier-free first-order formula in $L(\Omega)$. We use SAlg_n for semi-algebraic sets.

Let S be an m-ary relational symbol, and let $\psi(y_1, \ldots, y_n)$ be a first-order formula in the language of S and Ω. Then this query defines a map from $\mathsf{Obj}_m(\Omega)$ to $\mathsf{Obj}_n(\Omega)$ as follows: for any $X \in \mathsf{Obj}_m(\Omega)$, $\psi[X] = \{\vec{y} \mid (\mathcal{M}, X) \models \psi(\vec{y})\}$. Clearly $\psi[X] \in \mathsf{Obj}(\Omega)$, if $\mathcal{M}$ has quantifier-elimination.

Let $\mathcal{C}$ be a class of objects in $\mathsf{Obj}(\Omega)$. We say that a first-order query ψ *preserves* $\mathcal{C}$ if for any $X \in \mathcal{C}$, $\psi[X] \in \mathcal{C}$. For example, $\mathcal{C}$ can be the class of convex polytopes in SAlg.

Thus, the safety question for constraint databases is the following. Is there a effective syntax for the class of $\mathcal{C}$-preserving queriesΓ Below, we show an approach to solution, based on the characterization theorems for the finite case.

Definition 7.1 *The class $\mathcal{C}$ has a* canonical representation *in $\mathsf{Obj}(\Omega)$ if there is a recursive injective function $g : \mathbb{N} \to \mathbb{N}$ with computable inverse, and for each n, two functions $code_n : 2^{\mathcal{U}^n} \to 2^{\mathcal{U}^m}$ and $decode_n : 2^{\mathcal{U}^m} \to 2^{\mathcal{U}^n}$, where $m = g(n)$, such that:*

1. *$decode_n \circ code_n(x) = x$ if $x \in \mathsf{Obj}_n(\Omega)$;*
2. *$|code_n(x)| < \omega$ if $x \in \mathcal{C}$; $decode_n(x) \in \mathcal{C}$ if x is finite;*
3. *$code_n$ is $\mathrm{FO}(\Omega)$-definable on $\mathsf{Obj}_n(\Omega)$;*
4. *$decode_n$ is $\mathrm{FO}(\Omega)$-definable on finite sets.*

Intuitively, the canonical representation is a finite representation of $\mathcal{C}$ within $\mathsf{Obj}(\Omega)$ that can be defined in first-order logic over $\mathcal{M}$. For example, an approach to obtaining a canonical representation of convex polytopes would be to compute their vertices. This suffices to reconstruct the polytope, and the vertices can be defined by a first-order formula. The actual representation (Proposition 7.1) is indeed based on computing the vertices.

Next, we show that canonical representations solve the safety problem. We always assume that the set Ω is recursive.

Theorem 7.1 *Let $\mathcal{M} = \langle \mathcal{U}, \Omega \rangle$ be o-minimal, based on a dense order, decidable, and have effective QE. Suppose $\mathcal{C}$ is a class that has a canonical representation in $\mathsf{Obj}(\Omega)$. Then there is an effective syntax for $\mathcal{C}$-preserving $\mathrm{FO}(\Omega)$ queries; that is, there exists a recursively enumerable set of $\mathcal{C}$-preserving $\mathrm{FO}(\Omega)$ queries such that every $\mathcal{C}$-preserving $\mathrm{FO}(\Omega)$ query is equivalent to a query in this set.*

Proof: Consider an enumeration of all safe $\mathrm{FO}(\Omega)$ queries $\langle \varphi_i \rangle$ (from Corollary 4.4, we know that it exists). Let φ use the extra relation symbol of arity m, and assume that n is such that $g(n) = m$; given the assumptions, we can compute that. Let φ_i have l parameters, and again let k be such that $g(k) = l$. If n and k are found for a given φ_i, we let ψ be:

$$decode_k \circ \varphi_i \circ code_n.$$

This produces the required enumeration. So we have to check that every query above preserves $\mathcal{C}$, and for every $\mathcal{C}$ preserving ψ, we can get φ such that $decode \circ \varphi \circ code$ coincides with ψ. The first

one is clear: if we have $X \in \mathcal{C}$, then $code_n(X)$ is finite, hence $\varphi_i[code_n(X)]$ is finite too, and applying $decode_k$ we get an object in $\mathcal{C}$.

For the converse, suppose we have a $\mathcal{C}$-preserving query $\psi : \mathsf{Obj}_n(\Omega) \to \mathsf{Obj}_k(\Omega)$. Define α as follows: $\alpha = code_k \circ \psi \circ decode_n$. That is, α is a query $\mathsf{Obj}_m(\Omega) \to \mathsf{Obj}_l(\Omega)$. Given this, notice that

$$decode_k \circ \alpha \circ code_n = decode_k \circ code_k \circ \psi \circ decode_n \circ code_n = \psi$$

on $\mathsf{Obj}_n(\Omega)$. Thus, it remains to show that α is safe, i.e. preserves finiteness. Let X be a finite set in $\mathcal{U}^m$. Then $decode_n(X) \in \mathcal{C}$, $decode_n(X) \subset \mathcal{U}^n$. Since ψ is $\mathcal{C}$-preserving, we get that $Y = \psi[decode_n(X)] \in \mathsf{Obj}_k(\Omega)$ is in $\mathcal{C}$, too, and thus $code_k(Y)$ is finite. This proves finiteness of α, and concludes the proof of the theorem. □

We now turn to examples in the case when $\Omega = (+, *, 0, 1, <)$; that is, we are looking for canonical representations in SAlg. Let $\mathcal{CPH}$ be the class of convex polyhedra (intersections of a finite number of closed halfspaces) and $\mathcal{CPT}$ be the class of convex polytopes (bounded polyhedra). For the basic facts on convex sets that will be used in the proofs of the propositions below, see [33].

Proposition 7.1 *The class $\mathcal{CPT}$ has canonical representation in* SAlg.

Proof: Given a convex polytope X in $\mathbb{R}^n$, its vertices can be found as $V(X) = \{\vec{x} \in \mathbb{R}^n \mid \vec{x} \in X, \vec{x} \notin \mathrm{conv}(X - \vec{x})\}$. Thus, vertices of convex polytopes are definable in $\mathrm{FO}(+, *, 0, 1, <)$, because the convex hull of a finite set of points is definable, and, in view of Carathéodory's theorem, we have

$$V(X) = \{\vec{x} \in \mathbb{R}^n \mid \vec{x} \in X, \forall \vec{x}_1, \ldots, \vec{x}_{n+1} \in X - \vec{x}.\ \vec{x} \notin \mathrm{conv}(\{\vec{x}_1, \ldots, \vec{x}_{n+1}\})\}.$$

We now define $code_n$. To simplify the notation, we let it produce a pair of n-ary relations, but it can be straightforwardly coded by one relation. If $X = \mathrm{conv}(V(X))$, then $code_n(X) = (V(X), \emptyset)$; otherwise, $code_n(X) = (\mathbb{R}^n, X)$. The function $decode_n : 2^{\mathbb{R}^n} \times 2^{\mathbb{R}^n} \to 2^{\mathbb{R}^n}$ is defined as follows:

$$decode_n(Y, Z) = \begin{cases} \bigcup_{(\vec{y}_1, \ldots, \vec{y}_{n+1}) \in Y} \mathrm{conv}(\{\vec{y}_1, \ldots, \vec{y}_{n+1}\}) & \text{if } Y \neq \mathbb{R}^n, \\ Z & \text{otherwise.} \end{cases}$$

Clearly, $decode_n \circ code_n$ is the identity function for any (semialgebraic) set; these functions are also first-order definable. If X is a polytope, $V(X)$ is finite, and by Carathéodory's theorem each point of X is contained in the convex hull of at most $n + 1$ vertices of X. Hence, $card(code_n(X)) \leq card(V(X))^{n+1} < \omega$. If (Y, Z) is finite, then $decode_n(Y)$ is $\mathrm{conv}(Y)$, and thus a convex polytope. This proves the proposition. □

Proposition 7.2 *The class $\mathcal{CPH}$ has canonical representation in* SAlg.

Proof: We first give a brief sketch of the coding scheme. Start by recalling a few basic facts about convex polyhedra (see [15, 33]). Let X be a convex polyhedron in $\mathbb{R}^n$. Then $X = L + (X \cap L^-)$, where L is its lineality space, defined as $\{\vec{y} \mid \vec{y} = \vec{0} \text{ or } \forall \vec{x} \in X \forall \lambda.\ \vec{y} + \lambda \cdot \vec{x} \in X\}$ (it is a subspace of $\mathbb{R}^n$) and L^- is the orthogonal subspace $\{\vec{y} \mid \forall \vec{x} \in L.\ \langle \vec{x}, \vec{y} \rangle = 0\}$. We shall use X_0 for $X \cap L^-$ in this proof. It is known that X_0 is a convex polyhedron of lineality zero, that is, it contains no line. By $A + B$ we mean $\{\vec{a} + \vec{b} \mid \vec{a} \in A, \vec{b} \in B\}$. Note the difference between the translate $X - \vec{x} = \{\vec{y} - \vec{x} \mid \vec{y} \in X\}$ and the set-theoretic difference $X - x$; we use $\vec{x}$ to distinguish between them.

For X_0, define its vertices as $x \in X_0$ such that $x \notin \mathrm{conv}(X_0 - x)$. A direction is given by a vector $\vec{y}$ and corresponds to the equivalence class of rays which are translate of each other. Note that each

direction can be canonically represented by $\vec{y}$ such that $\| \vec{y} \| = 1$. A direction $\vec{y}$ is an extreme direction of X_0 if for some vertex $\vec{x}$, the ray through $\vec{x}$ in the direction of $\vec{y}$, $l(\vec{x}, \vec{y}) = \{\vec{x} + \lambda \cdot \vec{y} \mid \lambda \geq 0\}$, is a face of X_0. Since X_0 is polyhedral of lineality zero, the set of vertices and extreme directions is finite. By (generalized) Carathéodory's theorem [15, 33], every point $\vec{z}$ of X_0 is a combination of at most $n+1$ vertices and extreme directions,

$$\lambda_1 \vec{x}_1 + \ldots + \lambda_k \vec{x}_k + \mu_1 \vec{y}_1 + \ldots + \mu_m \vec{y}_m,$$

where $\lambda_i, \mu_j \geq 0, \lambda_1 + \ldots + \lambda_k = 1, k + m \leq n + 1$.

This suggests the following coding scheme. As before, for simplicity of exposition, we assume several coding relations, but they can be combined into one easily. We also do not spell out every first-order formula, but the reader should be convinced from the mathematical definitions that all the concepts we use are first-order definable over the real field. We use relations $LINEAL_k$; each such relation contains a canonical representation (roughly, an orthogonal basis) of the lineality space of X, provided its dimension is k. That is, at most one of these relations actually contains some information. We then have the relations *Vert* and *ExtDir* for storing vertices and extreme directions of X_0. Finally, we have a relation *Points* that contains points that do not belong to $L + X_0$ (recall that the coding scheme applies to any semi-algebraic set, so there could be such points, and we need to record them for the *decode* function).

Thus, to code (an arbitrary semi-algebraic) set X, we first note that its lineality space $L(X) = \{\vec{y} \mid \forall \vec{x} \in X.\ \vec{x} + \vec{y} \in X\}$ and its orthogonal $L(X)^- = \{\vec{y} \mid \forall \vec{x} \in L(X).\ \langle \vec{x}, \vec{y} \rangle = 0\}$ are definable in FO + Poly (note that one can define the inner product in FO + Poly). Furthermore, for each $k \leq n$, there exists a FO + Poly sentence dim_k expressing the fact that $L(X)$ is a subspace of $\mathbb{R}^n$ and its dimension is k. This is true because in FO + Poly we can test linear independence; thus, we can check if there exists a system of k linearly independent vectors in L such that every vector in L is a linear combination of them.

Next, we show how to compute $LINEAL_k$ and *VertDir*. We first sketch the coding scheme for $LINEAL_k$. The set $L(X)$ is FO + Poly-definable. Assume that it is a k-dimensional linear space (which is tested by dim_k). Let Δ_n be some canonically chosen n-dimensional simplex of diameter 1 such that the origin has barycentric coordinates $(\frac{1}{n}, \ldots, \frac{1}{n})$. Consider intersection of $L(X)$ with 1-dimensional faces of Δ_n (unless $L(X)$ is a line, in which case we consider its intersection with 2-dimensional faces of Δ_n). If the intersection is a point, we record that point; if it contains the whole face, we record both endpoints of the face. From the selected points, find a linearly independent subsystem (note that it can be done canonically, for example, by listing the vertices and 1-dimensional faces of Δ_n in some order). It then serves as a basis of $L(X)$, which we use to code $L(X)$. Note that $L(X)$ can be reconstructed in FO + Poly from its basis.

Now that we have a representation for the lineal space of X, and a first-order formula defining L^-, we have a FO + Poly-formula defining X_0. Using it, we can compute vertices

$$V(X_0) = \{x \in X_0 \mid \neg \exists x_1, \ldots, x_{n+1} \in X_0 - x.\ x \in \mathrm{conv}(\{x_1, \ldots, x_{n+1}\})\}.$$

Clearly, this is a first-order definition. Next, we find the set

$$E(X_0) = \{\vec{y} \mid \langle \vec{y}, \vec{y} \rangle = 1 \text{ and } \exists \vec{x} \in V(X_0).\ l(\vec{x}, \vec{y}) \text{ is a face}\}.$$

A subset Y of X_0 is a face if every closed line segment in X_0 with a relative interior point in Y has both endpoints in Y. Clearly, this is first-order definable, and thus $E(X_0)$, the set of extreme directions of X_0, is first-order definable.

Given two sets V and E in $\mathbb{R}^n$, by $\mathrm{conv}(V,E)$ we denote their convex hull, that is, the set of elements of $\mathbb{R}^n$ definable as $\sum_{i=1}^{k} \lambda_i \cdot \vec{x}_i + \sum_{j=1}^{m} \mu_j \cdot \vec{y}_j$, where $\vec{x}_i \in V$, $\vec{y}_j \in E$, $\sum_{i=1}^{k} \lambda_i = 1$, $\lambda_i, \mu_j \geq 0$ and $k + m \leq n + 1$. Again, this can be done in FO + Poly.

We now describe $code_n$. For a semi-algebraic set X, it produces a tuple of relations

$$(\mathit{LINEAL}_0, \ldots, \mathit{LINEAL}_n, \mathit{Vert}, \mathit{ExtDir}, \mathit{Point})$$

as follows. It first determines, by computing $L(X)$, $L(X)^-$, $V(X_0)$ and $E(X_0)$ if it is the case that $L(X)$ is a linear subspace of $\mathbb{R}^n$ and

$$X \;=\; L(X) + \mathrm{conv}(V(X_0), E(X_0)).$$

If this is the case, then LINEAL_k, *Vert*, and *ExtDir* are produced as before, and *Point* is empty. Otherwise, *Point* coincides with X, and all other sets in the coding are taken to be $\mathbb{R}^n$. From the description above it follows that $code_n$ is FO + Poly-definable.

To compute $decode_n$, we first check if the first $n+2$ relations in the code coincide with $\mathbb{R}^n$, and, if this is the case, output the last relation in the code. Otherwise, we use the nonempty LINEAL_k with least k to compute a linear subspace L of $\mathbb{R}^n$ generated by the vectors in LINEAL_k (if all LINEAL_k are empty, we let this subspace be $\{\vec{0}\}$). Next, compute $Y = \mathrm{conv}(\mathit{Vert}, \mathit{ExtDir})$. Note that both are FO + Poly-definable. Finally, return $L + Y \cap L^-$; this is FO + Poly-definable also.

We now sketch the proof that this coding scheme satisfies the conditions of the definition of canonical representation. Both *code* and *decode* are FO + Poly-definable. If $X \in \mathcal{CPH}$, then $L(X)$ is a linear space, X_0 has finitely many vertices and extreme directions, and $X = L + X_0$ implies that *Point* is empty, thus showing that *code* produces a finite set. Assume that *decode* is given a finite input Y. Then none of the first $n+2$ relations is $\mathbb{R}^n$, and thus the output of *decode* is the sum of a vector space and a convex hull of a finite set of vertices and directions, and thus a convex polyhedron. To show that $\mathit{decode} \circ \mathit{code}(X) = X$ for any semi-algebraic X, consider two cases. If $X = L(X) + \mathrm{conv}(V(X_0), E(X_0))$, then *Point* is empty, and *Vert* and *ExtDir* record all vertices and extreme directions of X_0, and one of LINEAL_k codes the lineality space. Thus, *decode* applied to $\mathit{code}(X)$ will return $L + X_0 = X$. If $X \neq L(X) + \mathrm{conv}(V(X_0), E(X_0))$, then all relations but *Point* coincide with $\mathbb{R}^n$, and *Point* contains X, and thus *decode* returns X. This completes the proof. □

Let SLinComp be the class of *compact* (closed and bounded) semi-linear sets. We resolve this case for dimension 2.

Proposition 7.3 *The class* $\mathsf{SLinComp}_2$ *has canonical representation in* SAlg_2.

Proof (sketch): An object in $\mathsf{SLinComp}_2$ is a finite union of convex polytopes in $\mathbb{R}^2$ – this easily follows from cell decomposition. Any such object X admits a triangulation that does not introduce new vertices [26]. Thus, the idea of the coding is to find the set $V(X)$ of vertices and use as the code triples of vertices (not necessarily distinct) $(\vec{x}, \vec{y}, \vec{z})$ with $\mathrm{conv}(\{\vec{x}, \vec{y}, \vec{z}\}) \subseteq X$. More precisely, a triple $(\vec{x}, \vec{y}, \vec{z})$ belongs to $\mathit{code}(X)$ if either $\vec{x}, \vec{y}, \vec{z} \in V(X)$ and $\mathrm{conv}(\{\vec{x}, \vec{y}, \vec{z}\}) \subseteq X$, or $\vec{x} = \vec{y} = \vec{z}$ and there is no triple of elements of $V(X)$ whose convex hull is contained in X and contains $\vec{x}$. Thus, $\mathit{code}(X) \subseteq \mathbb{R}^6$. For *decode*, we use

$$\mathit{decode}(Y) = \bigcup_{(\vec{x},\vec{y},\vec{z}) \in Y} \mathrm{conv}(\{\vec{x}, \vec{y}, \vec{z}\}).$$

Clearly, *decode* $\circ$ *code* is the identity, *decode* is first-order definable, and *decode*(Y) is compact and semi-linear when Y is finite. Thus, it remains to show that $V(X)$ is finite and FO + Poly-definable. The former is well-known (see [25]). For the first-order definition of $V(X)$ we use the following result from [25, 10]. Let X be a finite union of polyhedra (in $\mathbb{R}^n$) and let $B_\epsilon(\vec{x})$ be the ball of radius ϵ around $\vec{x}$. Then for each $\vec{x}$, there exists $\delta > 0$ such that for any $0 < \epsilon_1, \epsilon_2 < \delta$, we have

$$\vec{x} + \bigcup_{\lambda>0} \lambda \cdot [(X \cap B_{\epsilon_1}(\vec{x})) - \vec{x}] \quad = \quad \vec{x} + \bigcup_{\lambda>0} \lambda \cdot [(X \cap B_{\epsilon_2}(\vec{x})) - \vec{x}].$$

We denote this set by $X(\vec{x})$. Define the equivalence relation $\equiv_X$ by $\vec{y} \equiv_X \vec{z}$ if $X(\vec{y}) = X(\vec{z})$. Then the vertices of X are precisely the one-element equivalence classes of $\equiv_X$. It is routine to verify that the above can be translated into a FO + Poly definition of vertices. This completes the proof. □

Note that the coding scheme used in the proof of Proposition 7.3 *cannot* be used in higher dimensions. We used the fact there is a triangulation of a 2-dimensional polygon that does not introduce new vertices. However, in 3-dimensional case, there exist (non-convex) polygons for which such a triangulation is impossible, cf. [34]. In fact, [34] shows that the problem of deciding if a 3-dimensional polygon admits such a triangulation is NP-complete.

Summing up, we have

Theorem 7.2 *There exists a recursively enumerable class of* FO + Poly *queries that captures the class of $\mathcal{CPT}$ ($\mathcal{CPH}$ and* $\mathsf{SLinComp}_2$*, respectively) preserving queries.* □

The coding technique given here gives more information, however, as shown in the next section.

7.1 Decidability results and geometric bounds

While the classes of FO+Poly queries preserving certain properties have been shown to be recursively enumerable, in general, testing nontrivial preservation properties for arbitrary first-order queries is undecidable. For example, it is shown in [44] that it is undecidable whether a FO + Poly-query preserves semi-linearity. Here, we show that for a restricted class of FO + Poly queries – unions of conjunctive queries – preserving two of the properties considered here is decidable. The proofs are based on the representation theorems of this section, and the dichotomy theorem of the previous section. We first give the following bound on the behavior of conjunctive queries on convex polytopes.

Lemma 7.1 *Let $\varphi(x_1, \ldots, x_n)$ be a union of* FO + Poly *CQs that mention one m-ary relational symbol S. Then one can effectively find two numbers k and l such that φ is $\mathcal{CPT}$-preserving iff for every convex polytope D in $\mathbb{R}^m$ with at most k vertices, the output $\varphi[D]$ is a convex polytope with at most l vertices in $\mathbb{R}^n$.*

Proof: We can assume without loss of generality that $\varphi(\vec{x})$ is of the form

$$\bigvee_j \exists \vec{z} \bigwedge_i \alpha_{ij}(\vec{x}, \vec{z})$$

where each α_{ij} is either $S(\cdots)$ or a $L(+, *, 0, 1, <)$ formula. Let k_0 be the maximal number of S-atomic formulae in a disjunct of φ. Then the argument made in the proof of Lemma 5.2 shows that for each $\vec{a} \in \varphi[D]$, there exists a subset of $D' \subseteq D$ with at most k_0 points such that $\vec{a} \in \varphi[D']$.

Now assume that D is a convex polytope in $\mathbb{R}^m$, and $V(D)$ is the set of its vertices. Then each element of D belongs to $\mathrm{conv}(V')$ where V' is a subset of V of cardinality at most $m+1$. Now let $k' = k_0(m+1)$. Then it follows from monotonicity that for every $\vec{a} \in \varphi[D]$, there is a subset of $V' \subseteq V(D)$ of cardinality at most k such that $a \in \varphi[\mathrm{conv}(V')]$; in particular,

$$\varphi[D] \;=\; \bigcup_{V' \subseteq V(D), card(V') \leq k'} \varphi[\mathrm{conv}(V')].$$

We now set $k = 2k'$.

Next, consider the following FO + Poly query φ' which uses one m-ary relational symbol R. First, φ' constructs the convex hull of points in $\mathbb{R}^m$ which are in R. Then it applies φ to the result, to get a set Y. Finally, it returns $V(Y)$, that is, the set $\{y \in Y \mid y \notin \mathrm{conv}(Y - y)\}$. Clearly, φ' is expressible in FO + Poly. From the dichotomy theorem, we know that there is a polynomial p with the following property: if R is finite and contains i points, then either $\varphi'[R]$ is infinite, or it contains at most $p(i)$ points. We now let $l = \max\{p(i) \mid i = 1, \ldots, k\}$. Note that both k and l can be effectively calculated for a given φ.

It remains to show that if φ has the property that it sends a convex polytope with $\leq k$ vertices in $\mathbb{R}^m$ into a convex polytope with $\leq l$ vertices in $\mathbb{R}^n$, then it is $\mathcal{CPT}$-preserving. First, assume that D is a convex polytope with $\leq k$ vertices in $\mathbb{R}^m$. Assume $\varphi[D]$ is a convex polytope. If we apply φ' to a relation storing vertices of D, then the result is a finite set of vertices of $\varphi[D]$. Hence, by the dichotomy theorem, it has at most l vertices. That is, it now suffices to show that if φ has the property that it sends a convex polytope with $\leq k$ vertices in $\mathbb{R}^m$ into a convex polytope, then it is $\mathcal{CPT}$-preserving.

Let D be a convex polytope. We know that $\varphi[D] = \bigcup_{V'} \varphi[\mathrm{conv}(V')]$ where V' ranges over subsets of $V(D)$ that have at most $k' \leq k$ elements. Thus, each $\varphi[\mathrm{conv}(V')]$ is a convex polytope. Assume that $\varphi[D]$ is not convex. Then we can find two sets of vertices V_1, V_2, having at most k' elements each, and two points $\vec{a} \in \varphi[\mathrm{conv}(V_1)]$, $\vec{b} \in \varphi[\mathrm{conv}(V_2)]$, and $\vec{c}$ between $\vec{a}$ and $\vec{b}$ such that $\vec{c} \notin \varphi[D]$. Let $V_0 = V_1 \cup V_2$. Then, by monotonicity of φ, $\vec{a}, \vec{b} \in \varphi[\mathrm{conv}(V_0)]$. By the assumption, $\varphi[\mathrm{conv}(V_0)]$ is convex (since $card(V_0) \leq k$) and thus $\vec{c} \in \varphi[D]$. Hence, we showed that $\varphi[D]$ is convex. Since it is convex and a finite union of convex polytopes, it is a convex polytope itself. This completes the proof of Lemma 7.1 $\square$.

We now prove a similar bound for compact semi-linear sets in $\mathbb{R}^2$. When we speak of a triangle, we mean convex hulls of three points in $\mathbb{R}^2$. In particular, a degenerate triangle can be a segment or a point. We now prove the following.

Lemma 7.2 *Let $\varphi(x, y)$ be a union of conjunctive* FO + Poly *queries that mention one binary relational symbol S. Then one can effectively find two numbers k and l such that φ is $\mathsf{SLinComp}_2$-preserving iff for every set $D \subseteq \mathbb{R}^2$ which is a union of at most k triangles in $\mathbb{R}^2$, it is the case that $\varphi[D]$ is a union of at most l triangles in $\mathbb{R}^2$.*

Proof: Assume, as in the proof of Lemma 7.1, that $\varphi(\vec{x})$ is of the form $\bigvee_j \exists \vec{z} \bigwedge_i \alpha_{ij}(\vec{x}, \vec{z})$ where each α_{ij} is either $S(\cdots)$ or a $L(+, *, 0, 1, <)$ formula, and let k be the maximal number of S-atomic formulae in a disjunct of φ. Then, by the same argument as in the proof of Lemma 7.1, we obtain that if $\vec{x} \in \varphi[D]$, then there exist k points $\vec{z}_1, \ldots, \vec{z}_k$ in D such that $\vec{x} \in \varphi[\{\vec{z}_1, \ldots, \vec{z}_k\}]$. Assume that D is compact and semi-linear; since $D \subseteq \mathbb{R}^2$, it can be triangulated using only vertices of D. Let $V(D)$ be the set of vertices of D, which can be computed by a FO + Poly query, as shown in the

proof of Proposition 7.3. Since every point of D is in the convex hull of a triangle whose vertices come from $V(D)$, we obtain, by monotonicity of φ, that if $\vec{x} \in \varphi[D]$, then there exists a set V' of triples of elements from $V(D)$ such that $card(V') \leq k$ and

$$\vec{x} \quad \in \quad \varphi[\bigcup_{(\vec{u},\vec{v},\vec{w}) \in V'} \mathrm{conv}(\{\vec{u},\vec{v},\vec{w}\})].$$

Next, consider the following FO + Poly query ψ on finite databases. It uses one 6-ary schema relation R, that can be thought of as storing triples $(\vec{u},\vec{v},\vec{w})$ of points in $\mathbb{R}^2$. First, ψ computes

$$P(R) \quad = \quad \bigcup_{(\vec{u},\vec{v},\vec{w}) \in R} \mathrm{conv}(\{\vec{u},\vec{v},\vec{w}\}),$$

which is a compact semi-linear set, and then it computes the set $V(P(R))$ of vertices of $P(R)$, using the technique of [25], that we exploited in the proof of Proposition 7.3: for each $\vec{x}$, the set $P(R)(\vec{x}) = \vec{x} + [(P(R) \cap B_\epsilon(\vec{x})) - \vec{x}]$ does not depend on a particular value ϵ below some threshold δ. We then define vertices as those $\vec{x}$ for which there is no $\vec{x}_0 \neq \vec{x}$ with $P(R)(\vec{x}) = P(R)(\vec{x}_0)$. Thus, the query computing $V(P(R))$ is definable in FO + Poly, and by the dichotomy theorem, there is a polynomial p such that, for each R, either $V(P(R))$ is infinite, or has at most $p(n)$ vertices, where n is the number of tuples in R. We now let l be $\max\{p(i) \mid i = 1, \ldots, k\}$.

To show that these k and l witness the conclusion of the lemma, assume that φ is $\mathsf{SLinComp}_2$-preserving. Then the output of every φ on every union of k triangles is a compact semi-linear set. From the construction above, it follows that such an output can have no more than l vertices. Conversely, assume that the output of every union of k or fewer triangles is a union of l or fewer triangles. Since $\varphi[D]$ is the union of $\varphi[\bigcup_{(\vec{u},\vec{v},\vec{w}) \in V} \mathrm{conv}(\{\vec{u},\vec{v},\vec{w}\})]$, where V ranges over k-element sets of triples of vertices of D, we obtain that $\varphi[D]$ is a union of a finite number of triangles, and thus compact and semi-linear. This concludes the proof of the lemma. □

The promised decidability results now follow from the bounds established in the lemmas above.

Theorem 7.3 *The following two properties of unions of conjunctive* FO + Poly *queries are decidable:*

(a) being $\mathcal{CPT}$-preserving;

(b) being $\mathsf{SLinComp}_2$-preserving.

Proof of (a): Note that for each i, there is a FO + Poly query ψ_i for each i that tests if a set D is a convex polytope with at most i vertices: it checks that the set of vertices $V(D) = \{x \in D \mid x \notin \mathrm{conv}(D - x)\}$ has at most i elements, and that $D = \mathrm{conv}(V(D))$. In order to check if a UCQ φ in FO + Poly is $\mathcal{CPT}$-preserving, one applies Lemma 7.1 to compute the numbers k and l, and then writes a sentence saying that for every $\leq k$-element set V in $\mathbb{R}^m$, applying φ to $\mathrm{conv}(V)$ yields a polytope with at most l vertices. Since conv and ψ_l are definable, this property can be expressed as a FO$(+, *, 0, 1, <)$ sentence and thus it is decidable if it is true. Hence, the property of being $\mathcal{CPT}$-preserving is decidable.

Proof of (b): As in the proof of (a), we notice that the condition of Lemma 7.2 can be written as a first-order $L(+, *, 0, 1, <)$ sentence equivalent to:

$$\forall \{\vec{x}_i^j\}_{i=1,\ldots,k}^{j=1,2,3} \; \exists \{\vec{y}_s^p\}_{s=1,\ldots,l}^{p=1,2,3} \; \forall \vec{x}. \; (\vec{x} \in \bigcup_{s=1}^{l} \mathrm{conv}(\{\vec{y}_s^p \mid p = 1,2,3\})) \leftrightarrow \varphi'(\vec{x})$$

where φ' is obtained from φ by replacing each occurrence of $S(u,v)$ with a formula expressing the fact that $(u,v) \in \bigcup_{i=1}^{k} \mathrm{conv}(\{\vec{x}_i^j \mid j = 1,2,3\})$. Since the convex hull of a finite number of points is FO + POLY-definable, the condition of the lemma is indeed definable by a $L(+,*,0,1,<)$ sentence, and thus its validity is decidable. Hence, it is decidable if a union of conjunctive FO + POLY queries is $\mathsf{SLinComp}_2$-preserving. □

7.2 New expressivity bounds

We can also obtain new expressivity bounds by combining the dichotomy theorem with the idea of canonical representation. First, as an immediate consequence of the technique of Proposition 7.1, Lemma 7.1, and the dichotomy theorem, we obtain the following.

Corollary 7.1 *Let $\varphi(\vec{x})$ be a* FO + POLY *or* FO + EXP $\mathcal{CPT}$*-preserving query. Then there exists a polynomial p_φ such that, whenever D is a convex polytope with n vertices, $\varphi[D]$ has at most $p_\varphi(n)$ vertices.* □

From the proof of Lemma 7.2, one can extract the following, by applying the dichotomy theorem to a query that works on representations of compact semi-linear sets in $\mathbb{R}^2$ as finite unions of triangles:

Corollary 7.2 *Let $\varphi(x,y)$ be a* FO + POLY *query that is $\mathsf{SLinComp}_2$-preserving. Then there exists a polynomial p_φ such that, whenever D is a compact semi-linear set with n vertices, $\varphi[D]$ has at most $p_\varphi(n)$ vertices.* □

Consider the following problem: given a polyhedron P and $\epsilon > 0$, find a triangulation of P of mesh $< \epsilon$. That is, a triangulation such that the diameter of each simplex (triangle in dimension 2) is less than ϵ. It is a well-know result that each polyhedron admits such a triangulation [4]. The output of such a query can be structured in several ways, for example, by storing the information about the face structure of the triangulation. We only impose one requirement that the vertices of the triangulation be computable.

Proposition 7.4 *There is no* FO + EXP *query that finds a triangulation of a given polygon with a given mesh. This continues to hold if we restrict to convex polytopes on a plane.*

Proof: Suppose such a query exists; now consider a new query that does the following. Its input is one binary relation containing a set X of points $\vec{x}_1, \ldots, \vec{x}_n$ on the real plane, and one unary relation containing a single real number $\epsilon > 0$. First, in FO + POLY, construct $\mathrm{conv}(X)$, and then find vertices of a triangulation with mesh $< \epsilon$. This is clearly a safe query, so by the dichotomy theorem, there exists a polynomial p such that the number of vertices of the triangulation is at most $m = p(n+1)$ ($n+1$ is the size of the input). Let d be the maximal distance between the points $\vec{x}_i, \vec{x}_j$ (and thus the diameter of $\mathrm{conv}(X)$). Since the segment $[\vec{x}_i, \vec{x}_j]$ with $d(\vec{x}_i, \vec{x}_j) = d$ must be covered by the simplexes of the triangulation, it is possible to find a number ϵ such that it cannot be covered by fewer than $m+1$ triangles of diameter ϵ, and hence the number of points in the triangulation is $> m$. This contradiction proves the proposition. □

8 Conclusion and future work

Let us summarize the main themes of the paper.

- The relational calculus with interpreted functions is a nontrivial and interesting extension of the relational calculus. Many useful properties of the relational calculus remain in place in the presence of built-in functions, but many don't. Identifying the analogs to classical results in the interpreted setting can be tricky; proving them is not necessarily a piece of cake either.
- What sort of interpreted structure one adds matters.
- By combining results on the relational calculus with interpreted functions with some simple canonical representations of constraint databases, one can get interesting bounds on the behavior of constraint queries.

We now discuss extensions of each of these themes to other settings. In the first part of the paper, we identified some helpful properties of the relational calculus that remain in the presence of well-behaved built-in functions, with the real arithmetic functions being our prototypical example. Looking at structures such as real arithmetic or rational addition was quite helpful in discovering these results, but these characterization theorems are by no means limited to functions on real or even rational domains. Results in this paper and in [8] indicate that the safety and bound results fail badly for full integer arithmetic. However, we are currently working on extensions of these results to well-behaved structures over the integers, such as linear integer constraints. Although the growth dichotomy and range-restriction theorems as stated here fail for integer linear constraints, modifications of the characterization results still hold. In addition, several of their algorithmic consequences, such as the decidability results for conjunctive queries are still valid in the integer case.

In this paper we focused mainly on the relational calculus. Many of the proofs here, such as the results on range-restriction and safety, generalize straightforwardly to higher-order logics (fixpoint, second-order). Still, the safety question for many higher-order logics – particularly fixpoint logic in its many variations – is quite intricate, and we lack a full picture of what interpreted structures and recursion constructs permit a well-behaved theory of query safety.

Our emphasis here was showing that a wide class of interpreted functions satisfying some weak structural assumptions all exhibit certain kinds of tame behavior. In contrast, papers such as [19] give more detailed algorithmic analyses for specific structures. It still remains to give a complexity-theoretic analyses of both the safe translation problem and the query safety problem for conjunctive queries in the case of polynomial and linear constraints. A related interesting question is the complexity of deciding preservation properties for conjunctive queries over finitely representable databases.

We are working on several kinds of extensions of the growth bound theorems of Section 6. Some of our current work is on broadening the class of models these results apply to, and some of it concerns getting more precise bounds on the behavior of the growth function. It is fairly clear why these bounds have never been stated for the pure case: they are completely obvious for any pure query language. However, the pure case does put some strong limits on what sort of more precise information one can obtain on the behavior of the growth function. For example, results in the pure first-order case show that the function $f(n)$ giving the minimum nonzero size of output over models of size n can be sublogarithmic. Well-known results on the spectrum problem give restrictions on the structure of the set $\{n : growth_\varphi(n) = k\}$, for k any constant or ∞. One can, however, give theorems relating the

behavior of the growth function of a polynomial constraint query to that of a pure first-order query. We are also working on characterizations of the classes of interpreted structures for which the growth bound dichotomy theorem holds, and on characterizations of structures for which the growth function is bounded not by a polynomial, but by other definable functions of the input size (e.g., exponential).

The second part of the paper deals with applying our results on finite databases to finitely-representable ones, with the main technique coming via canonical codes. The main point of our codings was to facilitate this transfer of results. We are working on refining the results here to get natural canonical codings for larger geometric classes, and on studying these codes in themselves. The codings given here often capture a significant model-theoretic observation about the geometric class (e.g., the codings based on Carathéodory's theorem and its generalizations show that membership is determined by a bounded number of elements, and that these elements are definable from the database); they also give quite a bit of intuition on how queries that preserve these classes behave. We think this approach is quite promising one for arriving at useful languages for queries that preserve geometric structure. In fact, very recently, a syntactically defined subquery language of FO + POLY for manipulating semi-linear databases was given in [45]. Their approach was to combine Theorems 4.1 and 7.1 with the coding technique of [13] to find a canonical coding for semi-linear sets. Further study of canonical codes may also shed light on decidability of preservation properties for special classes of queries.

Acknowledgements We thank Ken Clarkson, Steve Fortune and Jianwen Su for helpful discussions, and anonymous referees for numerous comments and suggestions.

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] A.K. Ailamazyan, M.M. Gilula, A.P. Stolboushkin and G.F. Shvarts. Reduction of a relational model with infinite domains to the finite-domain case. *Doklady Akademii Nauk SSSR*, 286 (1) (1986), 308–311. Translation in *Soviet Physics – Doklady*, 31 (1986), 11–13.

[4] P.S. Aleksandrov. *Combinatorial Topology.* Graylock Press, 1956.

[5] A. Avron and J. Hirshfeld. On first order database query languages. In *LICS'91*, pages 226–231.

[6] M. Benedikt, G. Dong, L. Libkin and L. Wong. Relational expressive power of constraint query languages. *J. ACM* 45 (1998), 1–34.

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

[8] M. Benedikt and L. Libkin. Languages for relational databases over interpreted structures. In *PODS'97*, pages 87–98.

[9] M. Benedikt and L. Libkin. Safe constraint queries. In *PODS'98*, pages 99–108.

[10] H. Bieri and W. Nef. Elementary set operations with d-dimensional polyhedra. In *Proc. Workshop on Computational Geometry (CG'88)*, Springer LNCS 333, pages 97–112.

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

[12] M. Davis. On the number of solutions of Diophantine equations. *Proc. AMS* 35 (1972), 552–554.

[13] F. Dumortier, M. Gyssens, L. Vandeurzen, and D. Van Gucht. On the decidability of semi-linearity of semi-algebraic sets, and its implications for spatial databases. In *PODS'97*, pages 68–77.

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

[15] M. Grötschel, L. Lovász and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization.* Springer, 1993.

[16] S. Grumbach and J. Su. Finitely representable databases, *JCSS* 55 (1997), 273–298.

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

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

[19] O.H. Ibarra and J. Su. On the containment and equivalence of database queries with linear constraints. In *PODS'97*, pages 32–43.

[20] P. Kanellakis, G. Kuper, and P. Revesz. Constraint query languages. *Journal of Computer and System Sciences*, 51 (1995), 26–52. Extended abstract in *PODS'90*, pages 299–313.

[21] M. Kifer. On safety, domain independence, and capturability of database queries. In *Proc. Data and Knowledge Base*, Jerusalem, 1988.

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

[23] D. Marker, M. Messmer and A. Pillay. *Model Theory of Fields.* Springer Verlag, 1996.

[24] A. Levy and D. Suciu. Deciding containment for queries with complex objects. In *PODS'97*, pages 20–31.

[25] W. Nef. *Beiträge zur Theorie der Polyeder.* Herbert Lang, Bern, 1978.

[26] J. O'Rourke. *Computational Geometry in C.* Cambridge Univ. Press, 1994.

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

[28] J. Paredaens, J. Van den Bussche, and D. Van Gucht. First-order queries on finite structures over the reals. *SIAM J. Comput.* 27 (1998), 1747–1763.

[29] J. Paredaens, B. Kuijpers, G. Kuper and L. Vandeurzen. Euclid, Tarski and Engeler encompassed. In *DBPL'97*, Sprinegr LNCS 1369, 1998, pages 1–24.

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

[31] R. Ramakrishnan, F. Bancilhon and A. Silberschatz. Safety of recursive Horn clauses with infinite relations. In *PODS'87*, pages 328–339.

[32] P. Revesz. Safe query languages for constraint databases. *ACM TODS* 23 (1998), 58–99.

[33] R.T. Rockafellar. *Convex Analysis.* Princeton Univ. Press, 1970.

[34] J. Ruppert and R. Seidel. On the difficulty of triangulating three-dimensional nonconvex polyhedra. *Discrete and Computational Geometry* 7 (1992), 227–254.

[35] Y. Sagiv and M. Yannakakis. Equivalence among relational expressions with the union and difference operators. *J. ACM* 27 (1980), 633–655.

[36] A. Stolboushkin and M. Tsaitlin. Finite queries do not have effective syntax. In *PODS'95*, pages 277–285. Full version to appear in *Information and Computation.*

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

[38] A. Stolboushkin and M. Tsaitlin. Safe stratified datalog with integer order does not have syntax. *ACM TODS* 23 (1998), 100–109.

[39] J. D. Ullman. *Principles of Database and Knowledge-Base Systems.* Computer Science Press, 1988.

[40] J. D. Ullman. Information integration using logical views. In *ICDT'97*, pages 19–40.

[41] L. van den Dries. Remarks on Tarski's problem concerning (R,+,*,exp). In *Logic Colloquium'82*, North Holland, 1984, pages 97–121.

[42] L. van den Dries. *Tame Topology and O-Minimal Structures.* Cambridge, 1998.

[43] L. Vandeurzen, M. Gyssens and D. Van Gucht. On the desirability and limitations of linear spatial database models. In *SSD'95*, pages 14–28.

[44] L. Vandeurzen, M. Gyssens and D. Van Gucht. On query languages for linear queries definable with polynomial constraints. In *Proc. Principles and Practice of Constraint Programming*, Springer LNCS 1118, 1996, pages 468–481.

[45] L. Vandeurzen, M. Gyssens and D. Van Gucht. An expressive language for linear spatial database queries. In *PODS'98*, pages 109–118.

[46] A. van Gelder and R. Topor. Safety and translation of relational calculus queries. *ACM TODS*, 16 (1991), 235–278.

[47] M.Y. Vardi. The decision problem for database dependencies. *Inf. Proc. Let.*, 12 (1981), 251–254.

[48] 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.