Aggregate Operators in Constraint Query Languages

Michael Benedikt
Bell Laboratories
263 Shuman Blvd
Naperville, IL 60566
E-mail: benedikt@research.bell-labs.com

Leonid Libkin*
Department of Computer Science
University of Toronto
Toronto, Ontario M5S 3H5, Canada
E-mail: libkin@cs.toronto.edu

Abstract

We investigate the problem of how to extend constraint query languages with aggregate operators. We deal with standard relational aggregation, and also with aggregates specific to spatial data, such as volume. We study several approaches, including the addition of a new class of *approximate aggregate operators* which allow an error tolerance in the computation. We show how techniques of [23, 25] based on VC-dimension can be used to give languages with approximation operators, but also show that these languages have a number of shortcomings. We then give a set of results showing that it is impossible to get constraint-based languages that admit definable aggregation operators, both for exact operators and for approximate ones. These results are quite robust, in that they show that closure under aggregation is problematic even when the class of functions permitted in constraints is expanded.

This motivates a different approach to the aggregation problem. We introduce a language FO + Poly + Sum, which permits standard discrete aggregation operators to be applied to the outputs of range-restricted constraint queries. We show that this language has a number of attractive closure and expressivity properties, and that it can compute volumes of linear-constraint databases.

1 Introduction

New applications of database technology, such as Geographical Information Systems, have spurred a considerable amount of research into generalizations of the standard relational model to deal with the manipulation of geometric or spatial data. One common approach to modeling spatial databases is to consider input databases as given by a set of well-behaved relations in euclidean space – for example, by a set of semi-linear or semi-algebraic sets. There are a number of proposed query languages that extend classical relational algebra to this setting, languages that allow the use of various geometric operations in manipulating spatial databases. One of the most well-developed models for spatial queries is the *constraint database model* [22, 27]. In this model, spatial databases are represented as sets of linear or polynomial constraints. Databases are queried using standard relational calculus with linear (resp. polynomial) inequalities as selection criteria, see [3, 4, 5, 19, 20, 31, 36]. These languages, denoted by FO + Lin and FO + Poly, have become the dominant ones in the constraint database literature. They have a very important *closure property*: the application of a FO + Lin query to a linear constraint set yields a new set of linear constraints; similarly FO + Poly queries on sets definable with polynomial constraints produce sets that can still be defined with polynomial constraints.

*Contact author. Research affiliation: Bell Laboratories.

Constraint Query Languages, then, give a natural analog of relational calculus in the geometric context. A crucial question, though, concerns how to extend standard aggregation constructs from the relational model to the geometric setting. This question has two components. First, we would like our languages to be able to apply standard SQL operators such as TOTAL and AVG to spatial queries, whenever these operators make sense. Since the output of queries in constraint query languages (and in other spatial query languages) may be merely finitely representable (that is, representable by some finite means, e.g., a finite set of constraints) and not finite, the aggregation operators cannot be allowed to be applied to any constraint query output. One problem then, is to design a language that allows the safe application of classical aggregates.

The second component of the 'aggregation question' concerns aggregation notions that are specific to the spatial databases. Most commonly, given a database and the output of a query over it, one wishes to form new queries about the *volume* of this output. One may also extend standard aggregates such as AVG, and ask for the *average* value of a polynomial over a spatial object. Such aggregates arise both from practical concerns of GIS, and also as the natural continuous analogs of classical aggregation queries. Thus, we would like to extend constraint query languages to incorporate the ability to calculate volumes and other aggregates arising in the spatial setting.

In attempting to add aggregation to constraint query languages, one immediately encounters some daunting obstacles. While standard constraint databases are closed under first-order operations such as join and projection, they are clearly *not* closed under taking of volumes. This fact is well-known in the literature [22, 26, 11], and stems from the fact that neither the semi-linear nor semi-algebraic sets are closed under integrals. To take an example from the semi-algebraic setting, a query asking for the volume of initial slices of the epigraph of $1/x$ outputs the graph of the ln function, while iterating volume queries in this fashion would give as output transcendental functions that are not even expressible using field operations, logarithms and exponents. Thus, one cannot hope to add a general volume operator to existing first-order constraint query languages such as $\mathrm{FO} + \mathrm{POLY}$ and get a *closed* language while still remaining within the domain of polynomial constraint databases.

There are several approaches to the volume problem mentioned above. First, one could weaken the requirement that volumes be computed *exactly* and instead aim only to compute *approximate volumes*. Thus a query might have a tolerance associated with each instance of a volume operator, with output required only to be correct within the given tolerance. There are a number of practical and theoretical motivations for this approach. While it is known that computing volumes of even simple geometric objects (convex polytopes) is a hard problem (#P-hard, see [13]), approximation of volumes, at least of convex sets, can be done in polynomial time by a randomized algorithm [14]. Moreover, in contrast to the well-known fact that semi-algebraic and semi-linear sets are not closed under volume operators, the papers [23, 24, 25] show that volumes of sets definable with polynomial constraints can be approximated, for any given $\epsilon > 0$, by a first-order formula with polynomial constraints. By giving up exact volume and settling for an approximation, one might hope to retain desirable closure properties.

A second approach to the aggregation problem would be to expand out of the domain of polynomial constraints, and add new functions to the signature of both the constraints and the query language. This would give the possibility of retaining a constraint-based representation of databases, while gaining closure under volume operators. Of course, in this approach one should expand the constraint set so that it still defines only topologically well-behaved objects.

A third approach to the volume problem is to search for languages which can compute or approximate the volumes of important classes of sets, but which may not be closed under iterative application of

volume operators. For example, one could allow volume and other aggregation operators to be applied only to a subclass of the input queries. Restrictions on the nesting of volume operators would then have to be imposed.

An example of this last approach in the existing literature is [10], where it is shown that polynomial constraint query languages can express the (exact) volume for any set that admits a special condition called 'variable independence'. This condition means, informally, that in the constraint specification of sets in, say, $\mathbb{R}^2$, there is no interaction between x and y. Unfortunately, this condition is too restrictive: it excludes many of the sets that arise most often in spatial applications. As for practical applications of aggregation in constraint databases, implemented systems normally do not have aggregate operations (the DEDALE system, for example). One spatial extension of SQL, proposed in [28], does include aggregates, but a careful examination of the language shows that they are severely restricted: the only allowed aggregate operations are traditional relational ones applied to finite relations, and spatial union and intersection, which are first-order definable.

In this paper, we analyze the feasibility of each of the above approaches in detail. For the first approach, we show that techniques based on VC dimension, coming out of the work of [23, 24, 25] give us approximate volume operators that give semi-algebraic output on semi-algebraic input. However, we show a number of shortcomings of such an approach. Not only are the approximate volume operators obtained according to the technique of [23, 24, 25] sensitive to the input representation, but the blow-up in the size of the constraint databases produced in query evaluation precludes any possible use of these operators in practice.

Turning to the second approach, we show that it is completely infeasible. No first-order constraint language based on any reasonably well-behaved class of functions can express, or even approximate, volume. In the process of showing this, we develop a new set of techniques for proving inexpressibility results, techniques not based on the usual method of reduction to generic queries.

We then consider solutions that give up full closure under volume, and give a number of positive results. We present a higher-order language that allows one to calculate the volume of arbitrary semi-linear sets. Specifically, we give a language, called FO + POLY + SUM, that has attractive closure properties, remains within the domain of polynomial constraint databases, and allows the exact calculation of volumes for linear-constraint input databases. This language also has the pleasant feature that it is closed under the classical aggregation operators SUM and AVG. Since FO+POLY+SUM includes SQL aggregation, contains FO + POLY, and also allows one to make use of standard aggregation evaluation techniques in calculating volumes, it seems to be a good candidate for the constraint analog of classical aggregation languages.

We remark that another approach to the aggregation problem was considered in [11], which gave a new aggregate operator μ, under which FO + LIN is closed. However, $\mu(X) = 0$ for any bounded set X; thus, this operator cannot be used to deal with volumes.

Organization Section 2 introduces the notation. Approximability is studied in Section 3. The method of defining approximate volumes of [23, 24, 25] is analyzed, and the main difficulties in applying the approximation operators coming from this work are outlined. Section 4 shows that approximate volume operators cannot be defined in first-order constraint languages, even when the signature is expanded. Section 5 defines an extension of FO + POLY with SQL-like aggregation (summation over finite sets) and shows that this extension can express volumes of semi-linear databases.

The extended abstract of this paper appeared in the Proceedings of the 18th ACM Symposium on

Principles of Database Systems [7].

2 Notation

Structures, instances, queries Most notations are fairly standard in the literature on constraint databases, cf. [4, 5, 27, 31, 19]. Let $\mathcal{M} = \langle \mathcal{U}, \Omega \rangle$ be an infinite structure, where $\mathcal{U}$ is an infinite set, called a universe (in the database literature often called the domain), and Ω is a set of interpreted functions, constants, and predicates. In the field of constraint databases, most examples have $\mathcal{U} = \mathbb{R}$, the set of real numbers. Examples of signatures (and corresponding classes of constraints) that have been considered are:

Dense Order Constraints:	$\langle \mathbb{R}, < \rangle$;
Linear Constraints:	$\mathbf{R}_{\text{lin}} = \langle \mathbb{R}, +, -, 0, 1, < \rangle$;
Polynomial Constraints:	$\mathbf{R} = \langle \mathbb{R}, +, *, 0, 1, < \rangle$;
Exponential Constraints:	$\mathbf{R}_{\text{exp}} = \langle \mathbb{R}, +, *, e^x, < \rangle$.

A (relational) *database schema* SC is a nonempty collection of relation names $\{S_1, \ldots, S_l\}$ with associated arities $p_1, \ldots, p_l > 0$. We shall consider both finite and finitely representable instances. Given $\mathcal{M}$, an *finite 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 a finite instance D, $adom(D)$ denotes its *active domain*, that is, the set of all elements that occur in the relations in D.

A *finitely-representable (f.r.) instance* of SC over $\mathcal{M}$ is a family of sets $\{X_1, \ldots, X_l\}$, with $X_i \subset \mathcal{U}^{p_i}$, such that for each X_i there exists a quantifier-free formula $\alpha_i(x_1, \ldots, x_{p_i})$ in the language of $\mathcal{M}$ with $X_i = \{\vec{a} \in \mathcal{U}^{p_i} \mid \mathcal{M} \models \alpha_i(\vec{a})\}$. Most applications of constraint databases consider f.r. instances defined over $\mathbf{R}_{\text{lin}}$ (these are called *semi-linear sets*) or over $\mathbf{R}$ (called *semi-algebraic sets*). For example, in the spatial setting, a f.r. instance interprets the schema predicates as semi-linear or semi-algebraic sets.

As our basic query language, we consider relational calculus, or *first-order logic*, FO, over the underlying structure and the database schema. In what follows, $L(SC, \Omega)$ stands for the language that contains all symbols of SC and Ω; $\text{FO}(SC, \Omega)$ is 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 $\forall, \exists$.

Regardless of whether we are in the 'classical' setting, where these queries are applied to finite databases, or in the constraint query setting, we will refer to the above syntactic query languages as *relational calculus with* Ω *constraints*. This will be denoted by $\text{FO} + \Omega$. When Ω is $(+, -, 0, 1, <)$, or $(+, *, 0, 1, <)$, or $(+, *, e^x, <)$, we use the standard abbreviations FO + LIN, FO + POLY and FO + EXP.

In the case of finite databases, we shall also use the *active-domain quantifiers*: for a formula $\varphi(x, \vec{y})$, one can form formulae $\exists x \in adom. \varphi(x, \vec{y})$ and $\forall x \in adom. \varphi(x, \vec{y})$. 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 $\text{FO}(SC, \Omega)$ formulae, where $(\mathcal{M}, D) \models \exists x\, \varphi(x, \cdot)$ means for some $a \in \mathcal{U}$ we have $(\mathcal{M}, D) \models \varphi(a, \cdot)$, and $(\mathcal{M}, D) \models \exists x \in adom\, \varphi(x, \cdot)$ means that for some $a \in adom(D)$ we have $(\mathcal{M}, D) \models \varphi(a, \cdot)$. If $\mathcal{M}$ is understood, we write $D \models \varphi$.

Given $\varphi(\vec{x}, \vec{y})$ and $\vec{a}$, we write $\varphi(\vec{a}, D)$ for $\{\vec{b} \mid D \models \varphi(\vec{a}, \vec{b})\}$; in the absence of $\vec{x}$ we just write $\varphi(D)$ for the output of φ on D.

The class of subformulae of FO that only use the active-domain quantification is denoted by FO_{act}.

Adding aggregate operators We shall use $\text{VOL}(X)$ to denote the volume of a set $X \subseteq \mathbb{R}^n$. More precisely, $\text{VOL}(X)$ is the measure of any Lebesgue-measurable set $X \subseteq \mathbb{R}^n$. We shall not worry about dealing with non-measurable sets, as all bounded sets defined with constraints relevant for spatial applications (those listed above, plus some extensions) are measurable.

We shall consider adding volume to a query language as follows. If $\varphi(\vec{x}, \vec{y})$ is a formula, then the following is a formula with free variables $\vec{x}, z$:

$$[\text{VOL } \vec{y}.\varphi(\vec{x}, \vec{y})](\vec{x}, z)$$

Assume that a structure $\mathcal{M} = \langle \mathbb{R}, \Omega \rangle$ is fixed. Let an instance (finite or f.r.) D be given. Then

$$D \models [\text{VOL } \vec{y}.\varphi(\vec{x}, \vec{y})](\vec{a}, v) \text{ iff } v = \text{VOL}(\varphi(\vec{a}, D)).$$

Recall that $\varphi(\vec{a}, D) = \{\vec{b} \mid D \models \varphi(\vec{a}, \vec{b})\}$.

The extension of any query language $\mathcal{L}$ with VOL will be denoted by $\mathcal{L} + \text{VOL}$; for example, one can speak of $\text{FO} + \text{LIN} + \text{VOL}$ or $\text{FO} + \text{POLY} + \text{VOL}$. Of course we know that due to the nonclosure results mentioned in the introduction, $\text{FO} + \text{LIN} \subsetneqq \text{FO} + \text{LIN} + \text{VOL}$ and $\text{FO} + \text{POLY} \subsetneqq \text{FO} + \text{POLY} + \text{VOL}$.

As the next step, we restrict our attention to bounded sets. Without any loss of generality, we shall deal with subsets of $I^n \subset \mathbb{R}^n$, where I throughout this paper denotes the interval $[0, 1]$. We define $\text{VOL}_I \ \vec{y}.\varphi(\vec{x}, \vec{y})$ just as above, except that now we require that $v = \text{VOL}(\varphi(\vec{a}, D) \cap I^n)$. In particular, $0 \leq v \leq 1$. We similarly define languages $\mathcal{L} + \text{VOL}_I$. As with VOL, languages like $\text{FO} + \text{LIN}$ and $\text{FO} + \text{POLY}$ are not closed under VOL_I: for example, $\arctan(x) = \int_0^x \frac{dy}{y^2+1} = \text{VOL}_I(\{(y, z) \mid (0 \leq y \leq x) \wedge (0 \leq z \leq 1/(y^2 + 1))\})$, for $0 \leq x \leq 1$.

As standard languages are not closed under taking volume, we address the question of whether one can obtain closure by lowering one's demands. In particular, we would like to see if *approximating* the volume, rather than computing it directly, gives us a closed language. The hope that closure might be obtained in this way is motivated by the fact that for every formula $\varphi(\vec{x}, \vec{y})$ in $\mathbf{R}$ and for every $\epsilon > 0$, one can find a formula $\psi_\epsilon(\vec{x}, z)$ that gives ϵ-approximation of volumes of sets $\varphi(\vec{a}, \mathbb{R}) = \text{VOL}_I(\{\vec{b} \mid \models \varphi(\vec{a}, \vec{b})\})$, see [23, 24, 25].

We have to explain what we mean by approximating volume in this context. Clearly, we cannot hope to find $\psi_\epsilon(\vec{x}, z)$ with z defining an ϵ-interval around the real value of the volume – then the volume itself would be definable as the center of the interval! Thus, we settle for less. Similar to [23, 24, 25], we say for every $\epsilon > 0$, that an operator VOL^ϵ is an *ϵ-approximation operator* if for every f.r., over $\mathcal{M}$, set $A \in \mathbb{R}^n \times \mathbb{R}^m$, given by a formula $\varphi(\vec{x}, \vec{y})$, VOL^ϵ returns a f.r. set in $\mathbb{R}^n \times \mathbb{R}$, given by $\psi_\epsilon(\vec{x}, z)$ such that :

1. For every $\vec{a} \in \mathbb{R}^n$, $\psi_\epsilon(\vec{a}, \cdot)$ must be satisfiable (that is, $\mathcal{M} \models \exists z.\psi_\epsilon(\vec{a}, z)$);
2. If $\mathcal{M} \models \psi_\epsilon(\vec{a}, v)$, then $v \geq 0$ and $|v - \text{VOL}(\varphi(\vec{a}, \mathbb{R}))| < \epsilon$.

Thus, VOL^ϵ must return a ψ_ϵ that is guaranteed to find an (absolute) ϵ-approximation of the volume. We next say that a query language $\mathcal{L}$ *defines* VOL^ϵ, if there is a query in L that defines such an operator. That is, for each query $\varphi(\vec{x}, \vec{y})$ in $\mathcal{L}$ and $\epsilon > 0$ there is a $\mathcal{L}$-query $\psi_\epsilon(\vec{x}, z)$ such that for any database D, and any $\vec{a}$, we have

- $D \models \exists z.\psi_\epsilon(\vec{a}, z)$, and

- $D \models \psi_\epsilon(\vec{a}, v)$ implies $v \geq 0$ and $|v - \text{VOL}(\varphi(\vec{a}, D))| < \epsilon$.

Notice that in the last definition ψ_ϵ is independent of D. Also notice that to show definability of approximate operators in standard query languages, it suffices to show that there is a query in the language returning the ϵ-approximate volume on every base relation of some fixed arity.

We also define ϵ-approximation operators to volume in the case where we restrict to bounded sets. As before, we use, w.l.o.g., I^n as bounding set. An ϵ-approximation operator in the bounded setting is denoted by VOL_I^ϵ. Such an operator must satisfy the variant of condition 2) above: $|v - \text{VOL}(\varphi(\vec{a}, D) \cap I^n)| < \epsilon$ and $0 \leq v \leq 1$.

These operators, and their definability in query languages, are studied in Sections 3 and 4.

O-minimality, VC dimension Many results that we prove extend beyond linear and polynomial constraints. To state them in greater generality, we shall use *o-minimality* [35], which plays an important role in the study of constraint query languages (cf. [4, 5, 6]).

A structure $\mathcal{M} = \langle \mathcal{U}, \Omega \rangle$ is o-minimal, if every definable set is a finite union of points and open intervals $(a, b) = \{x \mid a < x < b\}$, $(-\infty, a) = \{x \mid x < a\}$, and $(a, \infty) = \{x \mid x > a\}$ (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 $\mathcal{M}$. All the structures on the reals we mentioned so far – $\mathbf{R}_{\text{lin}}$, $\mathbf{R}$, $\mathbf{R}_{\text{exp}}$ – are o-minimal (the first two by Tarski's quantifier-elimination, the last one by [37]).

If $\mathcal{M} = \langle \mathbb{R}, \Omega \rangle$, we define $\mathcal{M}_{+,*}$ to be $\langle \mathbb{R}, \Omega, +, * \rangle$. We often require that not just $\mathcal{M}$ but also $\mathcal{M}_{+,*}$ be o-minimal.

We also consider structures having finite VC dimension of definable families [2, 29] (also known as structures without the independence property [34]). VC dimension, introduced in statistics to study uniform convergence of stochastic processes, has become central to computational learning theory [2, 9], and found application in other areas, e.g., complexity [30].

Suppose X is an infinite set, and $\mathcal{C} \subseteq 2^X$. Let $F \subset X$ be finite; we say that $\mathcal{C}$ *shatters* F if the collection $\{F \cap C \mid C \in \mathcal{C}\}$ is 2^F. The *Vapnik-Chervonenkis (VC) dimension* of $\mathcal{C}$, $\text{VCdim}(\mathcal{C})$, is the maximal cardinality of a finite set shattered by $\mathcal{C}$. If arbitrarily large finite sets are shattered by $\mathcal{C}$, we let $\text{VCdim}(\mathcal{C}) = \infty$.

Let $\mathcal{M} = \langle \mathcal{U}, \Omega \rangle$, and let $\varphi(\vec{x}, \vec{y})$ be a formula in the language of $\mathcal{M}$ with $|\vec{x}| = n, |\vec{y}| = m$. For each $\vec{a} \in \mathcal{U}^n$, define $\varphi(\vec{a}, \mathcal{M}) = \{\vec{b} \in \mathcal{U}^m \mid \mathcal{M} \models \varphi(\vec{a}, \vec{b})\}$, and let $F_\varphi(\mathcal{M})$ be $\{\varphi(\vec{a}, \mathcal{M}) \mid \vec{a} \in \mathcal{U}^n\}$. Families of sets arising in such a way are called *definable families*. We say that $\mathcal{M}$ is a *structure with finite VC dimension* if the VC dimension of each definable family is finite. Every o-minimal structure is a structure with finite VC dimension [29], and the latter class is in fact much larger than the class of o-minimal structures.

3 Approximating aggregates in constraint query languages

3.1 The VC dimension-based implementation of approximate volume operators

We now start our investigation of the expressibility of approximate volume operators. The results of [23, 24, 25] do immediately give a closed language for computing approximate volumes. From [23, 24, 25] we can easily derive:

Theorem 1 *Let $\epsilon > 0$, and let $\varphi(\vec{x}, \vec{y})$ be a* FO + Poly *query. Then for every semi-algebraic (resp. semi-linear) database instance D there exists a formula $\varphi_D^\epsilon(\vec{x}, z)$ over the real ordered field $\mathbf{R}$ (resp. group $\mathbf{R}_{\rm lin}$) such that $\varphi_D^\epsilon(\vec{a}, \cdot)$ is satisfiable for all $\vec{a}$, and $\models \varphi_D^\epsilon(\vec{a}, v)$ implies $|\, v - \textsc{Vol}_I(\varphi(\vec{a}, D))\,| < \epsilon$ and $0 \leq v \leq 1$. Hence, there is a collection of ϵ-approximation operators $\textsc{Vol}_I^\epsilon$, $\epsilon > 0$, for $\mathbf{R}$ and $\mathbf{R}_{\rm lin}$.*

Since we want to examine those operators with regard to their efficiency, we now review the ideas of [23, 24, 25] that lead to this theorem.

Pre-requisites (see [2, 9, 23, 25]) The idea of the approximation technique can be traced back to the simplest randomized method for computing volumes. For a set $S \subseteq I^n \subset \mathbb{R}^n$, take k points $x_1, \ldots, x_k$ from the uniform distribution on I^n. Then $\textsc{Vol}(S)$ can be approximated as $v_S = \sum_{i=1}^k \chi_S(x_i)/k$, where χ_S is the characteristic function of S: $\chi_S(x) = 1$ if $x \in S$ and $\chi_S(x) = 0$ if $x \notin S$. Then for $\epsilon > 0$,

$$P(|\, v_S - \textsc{Vol}(S)\,| \geq \epsilon)) \;<\; 2e^{-2k\epsilon^2}\;;$$

this follows from Hoeffding's inequality. There are two reasons why this is not sufficient for getting ϵ-approximations to volume. First, the volume operators, as we defined them, may depend on parameters. Indeed, $\textsc{Vol}\ \vec{y}.\varphi(\vec{x}, \vec{y})$ requires computing the volume for *every* instantiation of parameters $\vec{x}$. Secondly, the randomized method above only tells us that $|\, v_S - \textsc{Vol}(S)\,| < \epsilon$ with high probability, and thus the procedure must be derandomized to ensure a certain answer.

To overcome the first problem, we use techniques from statistics and machine learning to ensure that one sample will suffice to test multiple volumes. Let $\varphi(\vec{x}, \vec{y})$ be a first-order formula over the real field $\mathbf{R}$, with $|\, \vec{x}\,| = n$ and $|\, \vec{y}\,| = m$, and let $\epsilon, \delta > 0$. Define $\varphi(\vec{a}, \mathbf{R}) = \{\vec{b} \in \mathbb{R}^m \mid \mathbf{R} \models \varphi(\vec{a}, \vec{b})\}$. Let $M > 0$ be given, and assume that an M-point sample $C = \{\vec{c}_1, \ldots, \vec{c}_M\}$ is randomly chosen in I^m. For each $\vec{a}$, let $v(\vec{a}, C)$ be the fraction of C that falls into $\varphi(\vec{a}, \mathbf{R}) \cap I^m$. Then one wants to achieve $|\, v(\vec{a}, X) - \textsc{Vol}_I(\varphi(\vec{a}, \mathbf{R}))\,| < \epsilon$ for all $\vec{a} \in \mathbb{R}^n$, with probability at least $1 - \delta$.

The classical results of learning theory [2, 9] say that this is possible when the VC dimension of the family $F_\varphi(\mathbf{R}) = \{\varphi(\vec{a}, \mathbf{R}) \mid \vec{a} \in \mathbb{R}^n\} \subseteq 2^{\mathbb{R}^m}$ is finite, and the size of the sample of M is then proportional to the VC dimension. In the construction of approximating formulae, we shall use the following corollary of this result, that states the existence of so-called ϵ-*nets*:

Fact 1 (ϵ-nets) *Let $\varphi(\vec{x}, \vec{y})$ be a first-order formula over the real field $\mathbf{R}$, with $|\, \vec{y}\,| = m$, and let $\epsilon > 0$. Let $d = \mathrm{VCdim}(F_\varphi(\mathbf{R}))$. If $M \geq \frac{8d}{\epsilon} \log \frac{13}{\epsilon}$, then there exists an M-element set $C = \{\vec{c}_1, \ldots, \vec{c}_M\} \subset I^m$ such that for every $\vec{a}$ with $\textsc{Vol}(\varphi(\vec{a}, \mathbf{R}) \cap I^m) \geq \epsilon$ it is the case that $\varphi(\vec{a}, \mathbf{R}) \cap I^m \cap C \neq \emptyset$.* □

Approximation method We now combine the existence of ϵ-nets with a derandomization procedure.

Assume that we are given a FO+Poly query $\psi(\vec{x},\vec{y})$ and a semi-algebraic database D. Put the definition of D into ψ, to obtain a new formula $\varphi(\vec{x},\vec{y})$ in the language of the real field, such that $\mathbf{R} \models \varphi(\vec{a},\vec{b})$ iff $D \models \psi(\vec{a},\vec{b})$. For example, if $\psi(x,y) \equiv \exists u\ (S(x,y,u) \wedge x < 0)$ and S is defined as $p(x,y,u) > 0$, where p is a polynomial, then $\psi(x,y)$ is $\exists u\ (p(x,y,u) > 0 \wedge x < 0)$.

Thus, we have to define approximating formulae for a formula $\varphi(\vec{x},\vec{y})$ over $\mathbf{R}$. To simplify notations, write $\Phi(\vec{a})$ for $\varphi(\vec{a},\mathbf{R}) \cap I^m$. Fix a number $k \in \mathbb{N}$ and $\nu, \delta \in I$. Define, for each $\vec{a} \in \mathbb{R}^n$,

$$S(\vec{a},\nu,\delta) \quad = \quad \left\{ (\vec{c}_1,\ldots,\vec{c}_k) \in (I^m)^k \quad \middle| \quad |\frac{1}{k} \cdot \sum_{i=1}^{k} \chi_{\Phi(\vec{a})}(\vec{c}_i) \ - \ \nu| \leq \delta \right\}.$$

That is, $S(\vec{a},\nu,\delta)$ is the set of k-samples that produce an approximate volume of $\Phi(\vec{a})$ within δ of ν. Note that for every fixed k, this set is definable with parameters $\vec{a},\nu,\delta$.

Next, define an operation $\oplus : I^n \times I^n \to I^n$ by $x \oplus y = (x+y) \bmod 1$, with the mod 1 operation applied component-wise. Let $\ominus$ be the inverse: $x \ominus y = z$ iff $x = y \oplus z$. These operations can then be extended to sets: $x \ominus S = \{x \ominus y \mid y \in S\}$.

Let $\mathbf{c} \in (I^m)^k$ be a k-element sample of points in I^m. Define $T(\mathbf{c},\vec{a},\nu,\delta) = \mathbf{c} \ominus S(\vec{a},\nu,\delta)$. For a fixed k, this is definable if FO over the real field.

Fix now $\vec{c} \in I^m$, and define the family $\vec{c} \ominus \Phi(\vec{a})$. As this is a definable family over $\mathbf{R}$, it has finite VC dimension [15, 29], which we denote by d. Then [23, 24, 25] calculate an upper bound on the VC dimension of the family $\mathcal{T}$ of all sets $T(\mathbf{c},\vec{a},\nu,\delta)$ as $4d \cdot k \log k$, for each fixed ν and δ.

Applying Fact 1, we obtain that for $M \geq \frac{32dk\log k}{\epsilon} \log \frac{13}{\epsilon}$, there is an ϵ-net $\{t_1,\ldots,t_M\}$ for $\mathcal{T}$. As translation by $\ominus$ does not change the volume, we see that all elements of $\mathcal{T}$ have the same volume; thus, if this volume is $\geq \epsilon$, then every member of $\mathcal{T}$ contains one of the t_is. From this one derives that the sets $t_i \oplus S(\vec{a},\nu,\delta)$ cover the entire $(I^m)^k$, if the volume of $S(\vec{a},\nu,\delta)$ is at least ϵ.

By calculations based on Hoeffding's inequality, [23, 24, 25] show that the inequality $\textsc{Vol}(S(\vec{a},\nu,\epsilon/2)) > 2e^{-k\epsilon^2/2}$ implies $|\,\nu - \textsc{Vol}(\Phi(\vec{a}))\,| \leq \epsilon$, and that $|\,\nu - \textsc{Vol}(\Phi(\vec{a}))\,| \leq \epsilon/4$ implies $\textsc{Vol}(S(\vec{a},\nu,\epsilon/2)) > 1 - 2e^{-k\epsilon^2/8}$. Using this, one arrives at the following.

Proposition 1 (Karpinski, Macintyre, Koiran) *Let*

$$0 < \epsilon \leq 1/2, \quad k \geq \frac{8 \cdot \ln 4}{\epsilon^2}, \quad M \geq \max\left(\frac{1}{2e^{k\epsilon^2/2}},\ \frac{32dk\log k}{\epsilon}\log\frac{13}{\epsilon}\right)$$

where d is the VC dimension of the family $\vec{c} \ominus \Phi(\vec{a})$, $\vec{c} \in I^m$. Then the formula saying that M translates (by $\ominus$) of $S(\vec{a},\nu,\epsilon/2)$ cover $(I^m)^k$ defines ν as an ϵ-approximation of $\textsc{Vol}(\Phi(\vec{a}))$. □

As d is finite and depends only on φ, the statement of the proposition can be converted into a FO-definition, which serves as an approximating definition of volumes $\varphi(\vec{x},\mathbf{R})$. Note that the resulting approximating formulae satisfy a rather strong condition: every ν within $\epsilon/4$ of the real volume is returned by the approximating formulae. Also note that the approximating formulae have entirely semi-linear character – multiplication is never used except in the formula φ itself. We thus obtain Theorem 1 as a corollary of the above results.

3.2 Shortcomings of the approximation technique

We note here some shortcomings of the technique of Lemma 1 in the context of constraint databases. In the technique, one has to put the definition of a constraint database D into a query φ, and then apply the method of [23, 24, 25] to the result. That method produces an output formula whose size is a polynomial in the input formula and $\frac{1}{\epsilon}$: theoretically, a nice bound. In attempting to apply this technique in practice, however, we find that the bounds obtained are rather unpleasant, even for modest values of ϵ, as the size of the quantifier prefix is quite large. In the constraint database setting, those will have to be eliminated, via a quantifier-elimination procedure, which will be very costly. Let us illustrate this by a simple example.

Example: Let the schema contain one unary predicate U interpreted as a subset of $[0, 1]$. The query $\varphi(x_1, x_2; y_1, y_2)$ is given by

$$U(x_1) \wedge U(x_2) \wedge x_1 < y_1 \wedge y_1 < x_2 \wedge 0 \leq y_2 \wedge y_2 \leq y_1$$

For $a, b \in U, a < b$, we have $\textsc{Vol}(\varphi(a, b, \mathbf{R})) = (b^2 - a^2)/2$.

Let $\epsilon = 1/10$. We want to evaluate the query

$$[\textsc{Vol}_I^\epsilon \vec{y}\, \varphi(x_1, x_2; y_1, y_2)](x_1, x_2, z)$$

saying that z is an ϵ-approximation to the volume of $\varphi(x_1, x_2, U) = \{(y_1, y_2) \mid U \models \varphi(x_1, x_2; y_1, y_2)\}$, where $\textsc{Vol}_I^\epsilon$ is the operator obtained through the method above. Note that $\textsc{Vol}_I(\varphi(a, b, U)) = (b^2 - a^2)/2$, for $a < b$ in U, and 0 otherwise. To evaluate this query on a database where U consists of N elements, by applying Theorem 1, we would first plug U in φ to obtain a formula with $> 2N$ atomic subformulae that does not mention U.

We then use bounds of Proposition 1 and obtain, by simple calculations:

$$k > 1,109 \qquad M > 25,206,250\ .$$

The formula saying that there exist M translates starts with a prefix $\exists \vec{t}_1 \cdots \exists \vec{t}_M$ where each t_i ranges over $(I^2)^k$; that is, this existential prefix binds $2kM > 5.5 \cdot 10^{10}$ variables.

The formula bound by these quantifiers must say that every element of $(I^m)^k$ is one of the translates, which requires at least $2Mk$ atomic formulae, and that ν is indeed the average value of the characteristic function, which requires at least $2kN$ atomic formulae. Thus, a crude lower bound for the length of the quantifier-free part of the formula is $5.5 \cdot 10^{10} + 2N \cdot 10^3$.

As eliminating $> 10^{10}$ quantifiers from a formula of length at least 10^{11} is completely infeasible, the approximation method has no chance of being applicable in practice. Still, the result that one can achieve closure by using approximate operators is very interesting, and contrasts sharply with the situation with the exact volume operators, where closure cannot be guaranteed.

Thus, applying the method of [23, 24, 25] 'as is' appears to be infeasible in the context of constraint databases.

The technique of Lemma 1 also tells us nothing about the definability of the operators $\textsc{Vol}_I^\epsilon$, nor the power of the language that results from adding them to a standard language, like $\mathrm{FO} + \textsc{Poly}$, since the approximating formula φ_D^ϵ varies with the input database.

4 Uniformly definable volume operators and expansion of the signature

We saw in the last section that the main shortcoming of all known examples of approximate volume operators was the blow-up in the size of the representation. It was also left open whether some volume approximation operators can be defined in standard languages, like FO + POLY, uniformly for all database instances. We now investigate whether we can find other approximation methods that can be expressed in nicely-behaved languages and that admit low complexity evaluation techniques. The main result is that one *cannot* capture approximate volume operators in a nice constraint language such as FO+POLY. That is,

Inexpressibility of Approximate Operators FO + LIN, FO + POLY *and* FO + EXP *cannot express* VOL_I^ϵ *for any* $\epsilon < 1/2$. □

In fact, we prove a stronger result. Theorem 3 shows that even if one extends the constraint signature to include functions beyond FO + EXP, as long as we stay within a well-behaved structure, we cannot capture approximate volume. Furthermore, we show that in languages like FO + POLY, only *trivial* approximations are possible. An example of a trivial approximation is returning 1/2 for every subset of I^n – in this case we know that the difference between the real volume and its approximation is $\leq 1/2$.

Proving expressivity bounds such as Theorem 3 and Corollary 1 is not very simple. Almost all, if not all, existing expressivity bounds for constraint query languages either involve generic queries (e.g., the parity test, see [4, 5, 31, 3]) or are proved by reduction to generic queries (e.g., [20]). However, queries involving approximation defined as in Section 2 are extremely nongeneric. We introduce the main ideas for the proof in several steps. We first consider an easier case of the aggregate AVG for finite instances and prove that it can be neither defined nor approximated in languages like FO + POLY. The proof introduces the idea of reduction to what we call a (c_1, c_2)-separating sentence, with c_1, c_2 being constant real numbers. We then show how the same reduction easily proves that FO + POLY and the likes cannot produce *relative* approximations of VOL. For the absolute approximation VOL_I^ϵ, the reduction only works under very special assumptions on the input, and to conclude the proof we need to use results from circuit complexity.

This section gives further evidence that if one wants to stay within a reasonable (for spatial applications) class of constraints, one must give up uniform closure under any nontrivial approximation to the volume.

Prerequisites: Collapse results We shall need the following two results proved in [4, 5], stated here in a form most convenient for the proofs below.

Fact 2 *a) Given any ordered structure* $\mathcal{M} = \langle \mathcal{U}, \Omega \rangle$, *an infinite set* $X \subseteq \mathcal{U}$, *and an active-semantics query* $\varphi(\vec{x})$ *in* $\text{FO}_{\text{act}}(SC, \Omega)$, *there exists an infinite set* $Y \subseteq X$ *and a* $\text{FO}_{\text{act}}(SC, <)$ *query* $\psi(\vec{x})$ *such that* $D \models \varphi(\vec{a})$ *iff* $D \models \psi(\vec{a})$ *whenever* $adom(D) \cup \vec{a} \subset Y$.

b) Let $\mathcal{M} = \langle \mathcal{U}, \Omega \rangle$ *be o-minimal, and* $\varphi(\vec{x})$ *and arbitrary natural-semantics* $\text{FO}(SC, \Omega)$ *query. Then there exists an expansion* $\mathcal{M}' = \langle \mathcal{U}, \Omega' \rangle$ *and an active-semantics query* $\psi(\vec{x})$ *in* $\text{FO}_{\text{act}}(SC, \Omega')$ *such that for every SC-database* D *over* $\mathcal{U}$ *and for every* $\vec{a}$, $D \models \varphi(\vec{a})$ *iff* $D \models \psi(\vec{a})$. *Furthermore, each relation in* $\Omega' - \Omega$ *is interpreted as a set definable over* $\mathcal{M}$; *thus, if* $\mathcal{M}$ *admits quantifier-elimination,*

one can take $\mathcal{M}'$ to be $\mathcal{M}$. □

Separating sentences We shall consider a relational database schema SC that consists of two unary relations, U_1 and U_2. Let $c_1, c_2 > 1$ be two real numbers. We say that Φ is a (c_1, c_2)-*separating sentence* if for any finite instance D of SC, it is the case that $card(U_1) > c_1 \cdot card(U_2)$ implies $D \models \Phi$ and $card(U_2) > c_2 \cdot card(U_1)$ implies $D \models \neg\Phi$. Note that this definition says nothing about the case when $\frac{1}{c_2} \cdot card(U_2) \leq card(U_1) \leq c_1 \cdot card(U_2)$, and thus direct application of bounds on expressiveness of generic queries is impossible. Still, we can show:

Proposition 2 *Let $\mathcal{M} = \langle \mathcal{U}, \Omega \rangle$ be o-minimal, $c_1, c_2 > 1$, and SC as above. Then no (c_1, c_2)-separating sentence is definable in* $\mathrm{FO}(SC, \Omega)$.

Proof. Assume that there is a (c_1, c_2)-separating sentence Φ. From Fact 2, b), we conclude that there is a $\mathrm{FO}_{\mathrm{act}}(SC, \Omega')$ (c_1, c_2)-separating sentence Φ' for some extension $\Omega' \supseteq \Omega$. From Fact 2, a), we obtain that there is an infinite set $Y \subseteq \mathcal{U}$ and a $\mathrm{FO}_{\mathrm{act}}(SC, <)$-sentence Ψ such that for every instance D with $adom(D) \subset Y$ it holds: $D \models \Phi'$ iff $D \models \Psi$. Thus, it remains to show that $\mathrm{FO}_{\mathrm{act}}(SC, <)$ cannot express a (c_1, c_2)-separating sentence Ψ, on instances over an infinite set.

Assume it can; and let q be the quantifier rank of Ψ. We now consider two instances over Y. In both instances D_1 and D_2 all elements of U_1 precede U_2 in the linear order $<$. In D_1, $card(U_1) = \lceil c_1(2^q+1) \rceil$ and $card(U_2) = 2^q + 1$; in D_2, $card(U_1) = 2^q + 1$ and $card(U_2) = \lceil c_2(2^q + 1) \rceil$. Since Ψ is a (c_1, c_2)-separating sentence, we must have $D_1 \models \Psi$ and $D_2 \models \neg\Psi$. We shall obtain contradiction by showing that $D_1 \models \Psi$ iff $D_2 \models \Psi$.

To show the latter, we must prove that the duplicator can win in a q-move Ehrenfeucht-Fraïssé game on D_1 and D_2. This follows from the fact for every $n, m > 2^q$, the duplicator can win a q-move game on two ordered sets of cardinalities n and m [21]. Thus, for D_1 and D_2, the duplicator picks a separate strategy for U_1 and U_2, and whenever the spoiler plays in U_1, the duplicator forgets about the moves in U_2 and responds in U_1 using the strategy for U_1, and likewise in the case when the spoiler plays in U_2. Let $(a_1, b_1), \ldots, (a_l, b_l)$ be moves made in the U_1 part of D_1 and D_2, with a_is played on D_1 and b_is played in D_2. Similarly, let $(c_1, d_1), \ldots, (c_k, d_k)$ be moves made in the U_2 part of D_1 and D_2, $k + l = q$. Then both $a_i \mapsto b_i$, $i = 1, \ldots, l$ and $c_i \mapsto d_i$, $i = 1, \ldots, k$, are partial isomorphisms; since all elements of U_1 precede all elements of U_2, putting them together we get a partial isomorphism between D_1 and D_2. This shows that $D_1 \models \Psi$ iff $D_2 \models \Psi$, and thus concludes the proof. □

4.1 Dealing with AVG

We assume that instances store elements of a numerical domain, for example $\mathbb{R}$. Given a query $\varphi(\vec{x}, z)$, we define $\textsc{Avg}_\varphi(\vec{x}, y)$ by letting $D \models \textsc{Avg}_\varphi(\vec{a}, v)$ iff $card(\varphi(\vec{a}, D)) < \infty$ and $v = \textsc{Avg}(\varphi(\vec{a}, D))$, where $\textsc{Avg}(C) = (\sum_{c \in C} c)/card(C)$. Note that the aggregate Avg is typically defined using the bag semantics; however, as we show *in*expressibility results, dealing with this simplified version will suffice[1].

It can be easily shown (by reduction to equal cardinality) that $\textsc{Avg}_\varphi$ is not definable in $\mathrm{FO} + \textsc{Poly}$, even if $D \models \varphi(\vec{a}, c)$ implies $0 \leq c \leq 1$. We now define ϵ-approximation of Avg just as we did it for Vol. Assume a query $\varphi(\vec{x}, z)$ is given, and $|\vec{x}| = n$. An operator $\textsc{Avg}^\epsilon_I$, when applied to φ, produces

[1] We shall come back to the multiset semantics later.

a query $\psi_\epsilon(\vec{x}, z)$ such that, for any instance D and any $\vec{a}$, $D \models \exists z.\varphi(\vec{a}, z)$, and if $D \models \varphi(\vec{a}, v)$, then $|\, v - \textsc{Avg}(\varphi(\vec{a}, D) \cap I)| < \epsilon$ and $0 \leq v \leq 1$. For convenience, we let $\textsc{Avg}(C) = 0$ for C infinite.

For $\epsilon \geq 1/2$, $\textsc{Avg}_I^\epsilon$ is definable in $\mathrm{FO}(SC, \Omega)$ if the input is finite or f.r. over Ω, as long as the constants $0, 1/2$ and 1 are definable. However,

Theorem 2 *Let $M = \langle \mathbb{R}, \Omega \rangle$, and let $\mathcal{M}_{+,*}$ be o-minimal. Let $\epsilon < 1/2$. Then $\textsc{Avg}_I^\epsilon$ is not definable in $\mathrm{FO} + \Omega$, even over finite instances. In particular, $\textsc{Avg}_I^\epsilon$ is not definable in $\mathrm{FO} + \textsc{Poly}$.*

Proof. Assume $\textsc{Avg}_I^\epsilon$ is definable. Let the schema SC consists of two unary predicates, U_1 and U_2. Let $\Delta = (1 - 2\epsilon)/16$. Given two finite sets U_1 and U_2, containing at least two elements each, we translate them into intervals $[0, \Delta]$ and $[1 - \Delta, 1]$. By translating a finite set X with $\min X = c, \max C = d > c$ into an interval $[a, b]$ we mean the set X' containing exactly the numbers of the form $a + \frac{(x-c)(b-a)}{d-c}$ where $x \in X$; clearly $X' \subset [a, b]$. As the next step, we define $U_1^0 = U_1' \cup \{4\Delta - x \mid x \in U_1'\}$ and $U_2^0 = U_2' \cup \{2 - 4\Delta - x \mid x \in U_2'\}$. One observes $U_1^0 \subseteq [0, 4\Delta]$ and $U_2^0 \subseteq [1 - 4\Delta, 1]$.

The preceding shows that U_1^0 and U_2^0 are $\mathrm{FO} + \textsc{Poly}$-definable. Thus, the set $C = U_1^0 \cup U_2^0 \subset [0, 1]$ is definable in $\mathrm{FO} + \textsc{Poly}$. Now easy calculations show that

$$\textsc{Avg}(C) \quad = \quad \frac{1}{8} - \frac{\epsilon}{4} + \frac{m}{n+m} \cdot \frac{3 + 2\epsilon}{4}$$

where n is the cardinality of U_1 and m is the cardinality of U_2.

We now define a Boolean query Φ by letting $D \models \Phi$ iff $\textsc{Avg}^\epsilon(C) = \textsc{Avg}_I^\epsilon(C) > 1/2$. More precisely, C is defined by a $\mathrm{FO} + \textsc{Poly}$ query $\alpha(x)$, and thus under the assumption that $\textsc{Avg}_I^\epsilon$ is definable, we have a satisfiable formula $\beta_\epsilon(x)$ such that $D \models \beta_\epsilon(a)$ implies that $|\, a - \textsc{Avg}(C)| < \epsilon$. We now let Φ be $\exists x.\beta_\epsilon(x) \wedge (x > 1/2)$. Under the assumption that $\textsc{Avg}_I^\epsilon$ is definable in $\mathrm{FO} + \Omega$, we would obtain that Φ is in $\mathrm{FO} + \Omega$ as well.

Let $c_0 = 1 + \frac{16\epsilon}{3 - 6\epsilon} > 1$. Assume $m > c_0 \cdot n$. Plugging this into the equation for $\textsc{Avg}(C)$, we derive $\textsc{Avg}(C) > 1/2 + \epsilon$; thus, in this case $\textsc{Avg}^\epsilon(C) > 1/2$ no matter which ϵ-approximation of the average is picked, and thus $D \models \Phi$. Similarlly, if we assume $n > c_0 \cdot m$, we derive $\textsc{Avg}(C) < 1/2 - \epsilon$, and thus $\textsc{Avg}^\epsilon(C) < 1/2$ and $D \models \neg\Phi$. Hence, Φ is a (c_0, c_0)-separating sentence, which is definable in $\mathrm{FO} + \Omega$. This contradiction proves the theorem. □

4.2 Dealing with volume

We start with two easy results. First, for unbounded measures (no restriction to I^n) volume cannot be approximated in languages like $\mathrm{FO} + \textsc{Poly}$.

Proposition 3 *Let $\mathcal{M} = \langle \mathbb{R}, \Omega \rangle$, and let $\langle \mathbb{R}, \Omega, +, * \rangle$ be o-minimal. Then no ϵ-approximation operator $\textsc{Vol}^\epsilon$ is definable in $\mathrm{FO} + \Omega$.*

Proof. Let $m > 2\epsilon + 1$ be an integer. Consider a scheme with two unary symbols U_1 and U_2 and let $\psi^1(x) \equiv \exists y\ U_1(y) \wedge |\, y - x| < m$ and $\psi^2(x) \equiv \exists y\ U_2(y) \wedge |\, y - x| < m$. Assume that $\textsc{Vol}^\epsilon$ is definable; we then have queries $\beta_\epsilon^1(x)$ and $\beta_\epsilon^2(x)$ which give ϵ-approximation for the measure of outputs of $\psi^1(x)$ and $\psi^2(x)$. Now let

$$\Psi \quad \equiv \quad \exists x_1 x_2\ (\beta_\epsilon^1(x_1) \wedge \beta_\epsilon^2(x_2) \wedge |\, x_1 - x_2| < 2\epsilon)$$

Let $(*)$ be the following condition on $U = U_1 \cup U_2$: for every $a, b \in U$, if $|\, a - b \,| \leq 2M$ then $a = b$. Then under $(*)$ it holds: $D \models \Psi$ iff $card(U_1) = card(U_2)$. However, this is impossible: Fact 2, a), implies that any generic query definable in $\mathrm{FO} + \Omega$ on databases over an infinite set must be definable in $\mathrm{FO}_{\mathrm{act}}(<)$, but it is well known that equal cardinality is not $\mathrm{FO}_{\mathrm{act}}(<)$-definable (cf. [21]). □

Thus, FO + LIN and FO + POLY cannot define ϵ-approximations of volumes. Note that the proof above is by reduction to equal cardinality, for sparse finite sets. It relies on the fact that there is no a priori bound on the outputs of queries. Thus, a different approach is needed to show inexpressibility of VOL_I^ϵ.

For a query $\varphi(\vec{x}, \vec{y})$ and two constants $0 < c_1 < c_2$, we say that $\psi(\vec{x}, z)$ gives a (c_1, c_2)-*relative approximation* of the volume if for any $\vec{a}$, $\psi(\vec{a}, \cdot)$ is satisfiable, and

$$D \models \psi(\vec{a}, v) \quad \Rightarrow \quad c_1 \ < \ (v/\mathrm{VOL}(\varphi(\vec{a}, D))) \ < \ c_2$$

By a reduction to separating sentences, we will now show:

Proposition 4 *Assume that $\langle \mathbb{R}, \Omega \rangle$ is such that $\langle \mathbb{R}, \Omega, +, * \rangle$ is o-minimal. Then for any $0 < c_1 < c_2$, the (c_1, c_2)-relative approximation of the volume is not definable in $\mathrm{FO} + \Omega$, for any dimension $k > 0$, even for queries restricted to $[0, 1]^k$.*

Proof. Let $k = 1$ (extension to $k > 1$ is trivial by taking a product with $[0, 1]^{k-1}$) and let the schema contain two unary relations U_1 and U_2. We shall assume that their interpretations are subsets of $[0, 1]$. Let $n = card(U_1), m = card(U_2)$. Let $c' = \frac{c_1}{2c_2} < \frac{1}{2}$ and $c'' \in (\frac{1}{2}, \frac{c_2}{2c_1})$. We claim that with a (c_1, c_2)-relative approximation of the volume we can define a sentence Φ such that $n < c' \cdot m$ implies $D \models \Phi$ and $n > c'' \cdot m$ implies $D \models \neg\Phi$. This will suffice, as such a sentence Φ would be a $(c'', 1/c')$-separating sentence, which cannot be defined in $\mathrm{FO} + \Omega$.

Given a finite set $X = \{x_1 < \ldots < x_p\} \subseteq [0, 1]$ and $\delta > 0$, define

$$X(\delta) \quad = \quad \bigcup_{i=1}^{p-1} [x_i, x_i + \delta] \quad \cup \quad [x_p - \delta, x_p]$$

Note that for a given δ and X, this is FO + LIN-definable. We now let

$$\delta = \frac{1}{3} \cdot \min_{a_1, a_2 \in (U_1 \cup U_2), a_1 \neq a_2} |\, a_1 - a_2 \,|$$

Then $U_1(\delta), U_2(\delta) \subseteq [0, 1]$ and $\mathrm{VOL}(U_1(\delta)) = n\delta, \mathrm{VOL}(U_2(\delta)) = m\delta$. Let $\Omega' = \Omega \cup \{+, *\}$. We now have two queries in $\mathrm{FO} + \Omega'$, $\varphi_1(y)$ and $\varphi_2(y)$ defining $U_1(\delta)$ and $U_2(\delta)$; assuming that (c_1, c_2)-relative approximation of the volume is definable, we have two queries $\psi_1(z)$ and $\psi_2(z)$ that produce such an approximation for $U_1(\delta)$ and $U_2(\delta)$. We next define

$$\Phi \quad \equiv \quad \exists z_1 \exists z_2 . \psi_1(z_1) \wedge \psi_2(z_2) \wedge (z_1/z_2 < 1/2).$$

Suppose $\psi_1(v_1) \wedge \psi_2(v_2)$ holds. It follows then that

$$\frac{c_1 n}{c_2 m} \ < \ \frac{v_1}{v_2} \ < \ \frac{c_2 n}{c_1 m}$$

Thus $n < c' \cdot m$ implies $v_1/v_2 < 1/2$ for any v_1 and v_2 that satisfy ψ_1 and ψ_2, and hence $D \models \Phi$. Conversely, $n > c'' \cdot m$ implies $v_1/v_2 > 1/2$ for any v_1 and v_2 that satisfy ψ_1 and ψ_2, and thus in this case $D \models \neg\Phi$. This completes the proof. □

4.3 Absolute approximation

We shall now prove the strongest of the inexpressibility results: that VOL_I^ϵ, for $\epsilon < 1/2$, cannot be defined in languages like FO + LIN and FO + POLY. First note:

Proposition 5 FO + LIN *defines* VOL_I^ϵ *for* $\epsilon \geq 1/2$.

Proof sketch. If the volume is not 0 or 1, then 1/2 is the ϵ-approximation. □

It turns out that this trivial approximation is the best one can hope for in languages like FO + LIN and FO + POLY.

Theorem 3 *Let* $\mathcal{M} = \langle \mathbb{R}, \Omega \rangle$, *and let* $\langle \mathbb{R}, \Omega, +, * \rangle$ *be o-minimal. Assume that* $\epsilon < 1/2$. *Then* VOL_I^ϵ *is not definable in* FO + Ω.

Proof. Let SC consist of two unary relations A and B. Call a finite instance good if two properties are satisfied: A is an initial fragment of natural numbers (that is, $\{0, 1, 2, \ldots, k\}$) and B is a nonempty proper subset of A. Let

$$c_1 = \frac{1 - 2\epsilon}{3} \quad \text{and} \quad c_2 = \frac{2 + 2\epsilon}{3}$$

We have $0 < c_1 < c_2 < 1$ and $c_1 + c_2 = 1$.

Consider a sentence Φ in the language of SC and Ω. We call it a (c_1, c_2)-good sentence is the following two conditions hold, whenever (A, B) is a good instance:

1. If $card(B) < c_1 \cdot card(A)$, then $D \models \neg\Phi$;
2. If $card(B) > c_2 \cdot card(A)$, then $D \models \Phi$.

Note that this is the same as having a separating sentence for B and $A - B$; however, here we only require that the above conditions hold for a good instance. The result now follows two lemmas.

Lemma 1 *Assume* VOL_I^ϵ *is definable in* FO + Ω. *Then for* c_1, c_2 *as above there exists a signature* Ω' *extending* Ω *and a* (c_1, c_2)*-good sentence in* $\text{FO}_{\text{act}}(SC, \Omega')$.

Proof of Lemma 1. Assume that an instance (A, B) with $B \subset A$ is given. Let $n = card(B)$ and $m = card(A - B)$; $n, m > 0$. We now construct A' and B' by translating A and B into $[0, 1]$. That is, each element x of A is replaced by x/x_M where x_M is the maximal element of A. Note that A', B' are FO + POLY-definable.

$$P = \bigcup_{b \in B', a \in A', (a,b) \cap A' = \emptyset} [b, a] \quad \text{and} \quad R = \bigcup_{b \in A' - B', a \in A', (a,b) \cap A' = \emptyset} [b, a]$$

Note that both P and R are definable in FO + POLY. We now have the following: if the instance (A, B) is good, then

$$\frac{n-1}{n+m-1} \leq \text{VOL}(P) \leq \frac{n}{n+m-1} \qquad \frac{m-1}{n+m-1} \leq \text{VOL}(R) \leq \frac{m}{n+m-1}$$

If $\textsc{Vol}_I^\epsilon$ is definable in FO + Ω, we have a FO($SC, \Omega \cup \{+, *\}$) queries $\psi_P(z)$ and $\psi_R(z)$ such that $D \models \psi_P(v)$ implies $|v - \textsc{Vol}(P)| < \epsilon$, and likewise for R. We now define Ψ as

$$\exists z_1 \exists z_2.\ \psi_P(z_1) \wedge \psi_R(z_2) \wedge z_1 > z_2$$

Let $c' = \frac{2+2\epsilon}{1-2\epsilon}$. Assume $card(B) > c_2 \cdot card(A)$; then $n > c'm$. Then simple calculations show that $\frac{n-1}{n+m-1} > 1/2 + \epsilon$ and $\frac{m}{n+m-1} < 1/2 - \epsilon$ which implies that no matter which ϵ-approximations v_1 and v_2 for $\textsc{Vol}^\epsilon(P)$ and $\textsc{Vol}^\epsilon(R)$ we have, it is the case that $v_1 - v_2 > 0$. Since ψ_P and ψ_R are satisfiable, we conclude that, under the assumption that the instance is good and $card(B) > c_2 \cdot card(A)$, $D \models \Psi$.

Next we assume that $card(B) < c_1 \cdot card(A)$. Then we get $m > c'n$. Again, with simple calculations we obtain $\frac{m-1}{n+m-1} > 1/2 + \epsilon$ and $\frac{n}{n+m-1} < 1/2 - \epsilon$; hence, for every ϵ-approximations v_1 and v_2 for $\textsc{Vol}^\epsilon(P)$ and $\textsc{Vol}^\epsilon(R)$, it is the case that $v_1 - v_2 < 0$, and thus $D \models \neg\Psi$.

Now the lemma follows from Fact 2, b). □

Lemma 2 *Let Θ be an arbitrary signature on $\mathbb{R}$. Then $\mathrm{FO}_{\mathrm{act}}(SC, \Theta)$ cannot define a (c_1, c_2)-good sentence.*

Proof of Lemma 2. Suppose for $0 < c_1 < c_2 < 1$ and for some signature Θ, there is a $\mathrm{FO}_{\mathrm{act}}(SC, \Theta)$ sentence Φ that is (c_1, c_2)-good according to the definition above. We may assume without loss of generality (just by adding existential quantifiers over the active domain) that all atomic formulae are either $A(x)$ or $B(x)$, where x is a variable, or Θ-atomic formulae. Next, make a signature Θ_Φ by putting a k_γ-ary symbol P_γ for each Θ-atomic subformula $\gamma(x_1, \ldots, x_{k_\gamma})$ of Φ into it. We then define a sentence Ψ in the language of Θ_Φ and B by replacing, in Φ, each atomic Θ-formula γ by the corresponding symbol P_γ, and $A(\cdot)$ by true.

Next, with each $n > 1$ and each $B \subseteq \{0, \ldots, n-1\}$ associate a $\Theta_\Phi \cup \{U\}$ structure $S(B, n)$ whose universe is $\{0, \ldots, n-1\}$, the unary symbol U is interpreted as B, and Θ_Φ predicates inherit their interpretation from $\langle \mathbb{R}, \Theta \rangle$ (this is possible since Θ_Φ does not contain any function symbols). We then have, by a straightforward induction on the structure of a formula

$$S(B, n) \models \Psi \quad \text{iff} \quad (\{0, \ldots, n-1\}, B) \models \Phi$$

where $(\{0, \ldots, n-1\}, B)$ is the good instance with A interpreted as $\{0, \ldots, n-1\}$. Thus, for $card(B) < c_1 n$ we have $S(B, n) \models \neg\Psi$ and for $card(B) > c_2 n$ we have $S(B, n) \models \Psi$.

It follows from [12] that Ψ is definable by a family of non-uniform AC^0 circuits, with size bounded by some polynomial $p(n)$, and depth d. This is because Ψ can be transformed into a Boolean formula by replacing each $\exists x \in adom$ by a disjunction over $\{0, 1, \ldots, n-1\}$ and each $\forall x \in adom$ by a conjunction over $\{0, 1, \ldots, n-1\}$. Once quantifiers are replaced, each occurrence of a Θ_Φ predicate only mentions constants and is replaced by its truth value (this is why the circuit may be non-uniform). It now follows from [12] that such a family of formulae is definable by a polynomial-size constant depth family of AC^0 circuits.

According to Lemma 5 from [12], for large enough inputs, constant-depth circuits cannot distinguish cardinalities in $[\sqrt{n}, n - \sqrt{n}]$. Thus, there is a number $N_1 \in \mathbb{N}$ such that for all $n > N_1$ it is the case that for any $p, q \in [\sqrt{n}, n - \sqrt{n}]$, $p \neq q$, there exists sets B_1 and B_2 of cardinalities p and q respectively such that $S(B_1, n)$ and $S(B_2, n)$ agree on Ψ. We now let N be an integer that exceeds both N_1 and $\frac{4}{c_1^2}$. Let n be an arbitrary integer bigger than N. Then there are integers n_1, n_2 such

that $n_1, n_2 \in [\sqrt{n}, n - \sqrt{n}]$ and $n_1 < c_1 n$, $n_2 > c_2 n$. In particular, for any two B_1 and B_2 such that $n_1 = card(B_1)$ and $n_2 = card(B_2)$, we have $S(B_1, n) \models \neg\Psi$ and $S(B_2, n) \models \Psi$ (since Ψ is equivalent to Φ, which is a (c_1, c_2)-good sentence). However, this contradicts the above observation that for some B_1 and B_2 as above, $S(B_1, n)$ and $S(B_2, n)$ must agree on Ψ. This contradiction concludes the proof of the lemma and the theorem. □

Corollary 1 FO + LIN, FO + POLY *and* FO + EXP *cannot express* VOL_I^ϵ *for any* $\epsilon < 1/2$. □

Theorem 3 shows that one cannot possibly adjust the method of [23, 24, 25] to get the approximation operators uniformly definable. This is somewhat surprising, for the following reasons. It is possible that there exists an o-minimal structure which is closed under taking integrals. That is, for every $\varphi(\vec{x}, \vec{y})$ in the language of the structure, there is a formula $\psi(\vec{x}, z)$ such that $\models \psi(\vec{a}, v)$ iff $v = \int \ldots \int \chi_{\varphi(\vec{a}, \mathbb{R}^n) \cap I^n} d\vec{y} = \text{VOL}(\varphi(\vec{a}, \mathbb{R}^n) \cap I^n)$. The existence of such a structure is conjectured in [24]. By Theorem 3, even if such a structure $\mathcal{M} = \langle \mathbb{R}, \Omega \rangle$ existed, the volume of outputs of very simple queries on finite instances could not be approximated in FO + Ω!

Is it possible that one can express the approximate volume computation over outputs of some particularly simple queries? We now show that for two very simple classes, this remains impossible in FO + POLY and similar languages.

Corollary 2 *In languages* FO + LIN, FO + POLY, FO + EXP, *it is impossible to express* VOL_I^ϵ *even restricted to a) outputs of conjunctive* $<$*-queries over finite instances, or b) schema predicates, interpreted as f.r. instances definable with dense-order constraints.*

Proof. Let the schema consist of three unary symbols A, B, C, and one binary symbol E. A finite instance D is called good if B, C form a partition of A, the distance between any two consecutive elements of A is the same, and E is the successor relation on $A \subset [0, 1]$. With this, we follow the proof of Theorem 3. We define P and R as before, and note that with C and E in the signature, they can be defined by conjunctive queries. For example, for P: $\psi_P(z) \equiv \exists b \in adom \exists c \in adom.\ B(b) \wedge C(c) \wedge E(b, c) \wedge b < z \wedge z < c$. Now, assuming VOL_I^ϵ is definable in $\text{FO}(SC, \Omega)$, we obtain, as in Lemma 1, that a (c_1, c_2)-good sentence is definable in $\text{FO}(SC, \Omega)$, for a good instance as defined above. This easily leads to contradiction: if a (c_1, c_2)-good sentence is definable in $\text{FO}(SC, \Omega)$ for instances with $A \subset [0, 1]$, it is definable in $\text{FO}(SC, \Omega, +, *)$ for instances with A being an initial fragment of natural numbers. Then the proof of Lemma 2 applies, as in the translation into a family of Boolean formulae the symbols C and E can be eliminated: $C(x)$ is replaced by $\neg B(x)$, and $E(x, y)$ by $y - x = 1$. This completes the proof. □

Remarks One may ask where the procedure of [23, 24, 25] fails if we try to apply it, in a uniform way, to, say, FO + POLY queries. Note that the method of [23, 24, 25] produces a formula whose quantifier prefix is proportional to the VC dimension of the family of sets defined by the input formula. However, for relational calculus queries, this may depend on the size of the database, thus making it impossible to quantify uniformly over random samples. For a query $\varphi(\vec{x}, \vec{y})$ with and a database D, the *definable family given by* φ *and* D is $F_\varphi(D) = \{\varphi(\vec{a}, D) \mid \vec{a} \in \mathcal{U}^n\}$ where $\varphi(\vec{a}, D) = \{\vec{b} \mid D \models \varphi(\vec{a}, \vec{b})\}$. The size of a finite database D, $|D|$, is defined to be $card(adom(D))$.

Proposition 6 *There exists a (quantifier-free) relational calculus query* $\varphi(x, y)$*, and a sequence of databases* $D_1, D_2, \ldots$ *of increasing size such that* $\text{VCdim}(F_\varphi(D_n)) \geq \log |D_n|$.

Proof. Let SC contain a single binary symbol P. Let D_n be an instance with the second projection being an n-element set A_n, and the first projection coding the powerset of A_n (as in [1, page 462]). That is, for each $B \subseteq A_n$ there is a_B such that $(a_B, b) \in P$ iff $b \in B$. Let $\varphi(x, y) \equiv P(x, y)$. We now consider the family $\mathcal{F}_n = \{\varphi(a, D) \mid a \in \mathcal{U}\}$. It follows immediately from the construction that $\mathcal{F}_n$ shatters A_n; thus, $\mathrm{VCdim}(F_\varphi(D_n)) \geq n$. Since one needs 2^n elements to code the powerset of A_n, one can choose D_n to have the active domain of 2^n elements. This proves the proposition. □

We also remark that under some special assumptions on the outputs of the queries, their volumes can be approximated. One can show, using Löwner-John ellipsoids [16], that for a FO + Poly query $\varphi(\vec{x}, \vec{y})$ with $|\vec{y}| = k$, under the assumption that $\varphi(\vec{a}, D)$ is convex, a relative (c_1, c_2) approximation of its volume can be found with $c_1 = \frac{k^k+1}{2 \cdot k^k} - \epsilon$ and $c_2 = \frac{k^k+1}{2} + \epsilon$ for an arbitrarily small $\epsilon > 0$.

5 FO + Poly + Sum: An aggregate language for constraint databases

We now introduce a language for extending FO + Poly with a summation operator. The main difficulty is to make sure that when summation is done over all elements in some query output, we are guaranteed that the query output is *finite*. To do this, we use techniques from [6] for guaranteeing that a query is *safe* (that is, that a query yields finite output).

Let Q be a non-boolean query over a database schema SC. We say that Q is a *semi-algebraic query* if it gives semi-algebraic output on semi-algebraic inputs. We say Q is *semi-algebraic-to-finite* and write $Q \in \mathrm{SAF}$ if Q produces finite output on semi-algebraic input databases. If Q is expressed as $\varphi(y, \vec{x})$, we say that Q is *$\vec{x}$-semi-algebraic-to-finite* if for every $\vec{a}$ the query $\varphi(y, \vec{a})$, with one free variable y, is in SAF. In the language FO + Poly + Sum, *all* queries are semi-algebraic queries, but in the construction we will have to ensure that certain subqueries are in the smaller class SAF.

A first-order formula $\gamma(x, \vec{w})$ with distinguished variable x in the language of the real field is said to be *deterministic* if it produces at most one output x for every vector of real numbers $\vec{w}$. Deterministic formulae are the building blocks from which safe queries can be formed. Given a deterministic formula $\gamma(x, \vec{w})$ and a finite set of tuples of reals A (having the same length as $\vec{w}$), we let $\gamma(A)$ refer to the *bag* $\uplus_{\vec{a} \in A} f_\gamma(\vec{a})$, where f_γ is the corresponding partial function taking $\vec{w}$ to the unique x such that $\gamma(x, \vec{w})$ holds. Note that it is decidable if a formula is deterministic.

Definition of FO + Poly + Sum The query language FO + Poly + Sum is defined inductively as follows. Atomic queries are the same as for FO + Poly. The formulae of FO + Poly + Sum are closed under boolean connectives and quantification $\forall$ and $\exists$ (over the reals).

Next, we define the *summation term-former*. Given any FO + Poly + Sum formula $\varphi(y, \vec{z})$, we let $\textsc{End}[y, \varphi(y, \vec{z})](u, \vec{z})$ be the query that holds for a tuple $(b, \vec{a})$ on an input database D iff b is an endpoint of the intervals that compose $\varphi(D, \vec{a}) = \{c \in \mathbb{R} \mid D \models \varphi(c, \vec{a})\}$. Note that if φ is a semi-algebraic query (which is guaranteed by Theorem 4 below), then $\textsc{End}[y, \varphi(y, \vec{z})]$ is $\vec{z}$-SAF.

A *range-restricted* FO + Poly + Sum expression is an expression of the form $\rho(\vec{w}, \vec{z}) \equiv (\varphi_1(\vec{w}, \vec{z}) | \textsc{End}[y, \varphi_2(y, \vec{z})])$ where $\varphi_1(\vec{w}, \vec{z})$ and $\varphi_2(y, \vec{z})$ are FO + Poly + Sum queries. It binds y, that is, the free variables are $\vec{z}, \vec{w}$. We have $D \models \rho(\vec{a}, \vec{b})$ for $\vec{a} = (a_1, \ldots, a_n)$ iff $D \models \varphi_1(\vec{a}, \vec{b})$ and

$$D \models (\textsc{End}[y, \varphi_2(y, \vec{z})])(a_i, \vec{b}), \ i = 1, \ldots, n.$$

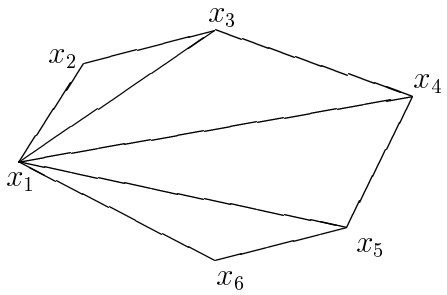

Figure 1: Area of convex polygon in FO+POLY+SUM

It then follows from the closure property (Theorem 4) that for any D and any $\vec{b}$, the set $\rho(D, \vec{b}) = \{\vec{a} \mid D \models \rho(\vec{a}, \vec{b})\}$ is finite.

For any deterministic formula $\gamma(x, \vec{w})$ and any range-restricted expression $\rho(\vec{w}, \vec{z})$ as above we now define a term $t(\vec{z})$ by

$$[\sum_{\rho(\vec{w},\vec{z})} \gamma](\vec{z})$$

Given D and $\vec{b}$, the value of $t(\vec{b})$ in D is the sum of all the members of the finite bag $\gamma(A)$, where $A = \rho(D, \vec{b})$.

Finally, new terms in FO + POLY + SUM can be built by applying composition with the real functions $+, *$. If t_is are terms and φ is a formula, then $t_1 = t_2, t_1 < t_2$ and $\varphi(t_1, \ldots, t_k)$ are FO + POLY + SUM formulae.

Examples of FO + POLY + SUM queries Let $\varphi(w)$ be an FO + POLY query. Let $\gamma(x, w) \equiv (x = w)$ and $\rho(w) = (w = w)|\text{END}[w, \varphi(w)]$. Then the FO + POLY + SUM term (without free variables) $\sum_{\rho(w)} \gamma$ gives the sum of all the endpoints of the intervals that compose $\varphi(D)$.

The area of a convex polygon in $\mathbb{R}^2$ can be defined in FO + POLY + SUM. The idea of the query is illustrated in Figure 1. Suppose we triangulate the polygon as shown. Then the area of the polygon is the sum of the areas of triangles. We thus have to define the triangulation and then apply the summation term of FO+POLY+SUM to calculate the area.

This is done as follows. Assume that the polygon is given by a predicate $P(x, y)$ (it could be an input relation or the output of a query). There is a FO + POLY query $\varphi_P(x, y)$ that computes all the vertices of P – this is because $\vec{a}$ is vertex iff $\vec{a} \notin \text{conv}(P - \{\vec{a}\})$. Since one can compute the boundary of P by a FO + POLY query, it follows that there is a FO + POLY query $\nu_P(\vec{x}, \vec{y})$ that tests if $\vec{x}, \vec{y}$ are two adjacent vertices of P.

We now form two FO + POLY queries. The query $\psi_2(u)$ tests if u is a coordinate of a vertex of P. The query $\psi_1(\vec{x}, \vec{y}, \vec{z})$ tests the following conditions: (1) $\varphi_P(\vec{x}) \wedge \varphi_P(\vec{y}) \wedge \varphi_P(\vec{z})$ holds; (2) $\vec{x}$ is a lexicographically minimal vertex of P; (3) either $\nu_P(\vec{y}, \vec{z})$ holds and $\vec{y}$ is lexicographically less than $\vec{z}$ and $\neg\nu_P(\vec{x}, \vec{y}) \wedge \neg\nu_P(\vec{x}, \vec{z})$, or $\nu_P(\vec{x}, \vec{y}) \wedge \nu_P(\vec{y}, \vec{z}) \wedge \neg\nu_P(\vec{x}, \vec{z})$.

We now let $\rho(\vec{x}, \vec{y}, \vec{z})$ be the range-restricted expression $(\psi_1(\vec{x}, \vec{y}, \vec{z})|\text{END}[u, \psi_2(u)])$. It can be easily seen that for P convex, the output of ρ is finite and produces a triangulation of P. That is, $\rho(\vec{a}, \vec{b}, \vec{c})$

holds iff $\vec{a}, \vec{b}, \vec{c}$ are the vertices of one of the triangles such as those shown in Figure 1.

Since for each triangle with vertices (a_1, a_2), (b_1, b_2), (c_1, c_2), its area is computable as $| (a_1b_2 - a_2b_1 + a_2c_1 - a_1c_2 + b_1c_2 - c_1b_2)/2 |$, we obtain a deterministic formula $\gamma(v, \vec{x}, \vec{y}, \vec{z})$ saying that v is the area of the triangle with vertices $\vec{x}, \vec{y}, \vec{z}$. We then conclude that the term $\sum_{\rho(\vec{x},\vec{y},\vec{z})} \gamma$ defines the area of P. □

Note that the above method codes a standard computation of area used in computational geometry [33] which generalizes to nonconvex polygons, and is in fact used in GISs for area computation [38].

Properties of FO+Poly+Sum The language FO+Poly+Sum has a number of attractive features. It extends both FO+Poly and the relational calculus with summation and other standard aggregates. It is also related to aggregate languages for statistical databases studied recently in [17]. Furthermore, we have the following property.

Theorem 4 FO+Poly+Sum *is closed. That is, every* FO+Poly+Sum *query returns semi-algebraic output on a semi-algebraic input.*

Proof. We show this by structural induction on the construction of the query. Suppose we know inductively that $\varphi(w, \vec{z})$ is a semi-algebraic query, and fix a semi-algebraic database D. There is an integer n such that for any $\vec{a}$, $D \models \textsc{End}[y, \varphi(y, \vec{z})](c, \vec{a})$ for at most n distinct values of c (by o-minimality of the real field and the uniform bounds result of [32]). Moreover, this integer can be effectively computed given φ and D. Hence, for every $\vec{z}$,

$$\rho(\vec{w}, \vec{z}) \quad \equiv \quad \varphi_1(\vec{w}, \vec{z}) \mid \textsc{End}[y, \varphi(y, \vec{z})]$$

holds for at most n^m tuples $\vec{w}$, where m is the length of $\vec{w}$. We then see that the set $\{(v, \vec{a}) \mid D \models v = \sum_{\rho(\vec{w},\vec{a})} \gamma(x, \vec{w})\}$ is semi-algebraic, since it is definable by the disjunction of $v = 0 \wedge \forall \vec{w}\ \neg\rho(\vec{w}, \vec{a})$ with

$$\bigvee_{1 \le k \le n^m} \exists \vec{w}_1 \cdots \exists \vec{w}_k \Big(\bigwedge_{i=1}^{k} \rho(\vec{w}_i, \vec{a}) \wedge (\forall \vec{w}\ \rho(\vec{w}, \vec{a}) \rightarrow \bigvee_i (\vec{w} = \vec{w}_i)) \wedge \bigwedge_{i \neq j} (\vec{w}_i \neq \vec{w}_j)$$
$$\wedge \exists u_1 \cdots \exists u_k\ ((\bigwedge_{i=1}^{k} \gamma(u_i, \vec{w}_i) \vee ((\forall z \neg\gamma(z, \vec{w}_i)) \wedge u_i = 0))\ \wedge\ (v = u_1 + \cdots + u_k)) \Big) .$$

The language is also closed under the standard relational aggregation.

Proposition 7

- *For any* SAF FO+Poly+Sum *query* $\varphi(\vec{z})$*, we can express in* FO+Poly+Sum *the cardinality of the output of* φ*.*
- *For any* SAF *query* FO + Poly $\varphi(\vec{z})$ *and any deterministic formula* $\chi(x, \vec{w})$ *we can express in* FO + Poly + Sum *the sum of the* x *values of* χ *for* $\vec{w}$ *ranging over the output of* φ *and the average of the* x *values of* χ *over the output of* φ*.*

Proof. To see the first item, consider an arbitrary SAF FO + Poly + Sum query $\varphi(\vec{w})$. Let $\varphi'(w)$ be the query returning the active domain of the output of φ. Then φ' is clearly SAF as well, and

$\textsc{End}[w, \varphi'(w)]$ is the same as $\varphi'(w)$. Let $\rho(\vec{w}) = \varphi(\vec{w})|\textsc{End}[w, \varphi'(w)]$ and $\gamma(x, \vec{w})$ be $x = 1$. Then $\sum_{\rho(\vec{w})} \gamma$ is an FO + Poly + Sum query returning the number of items in the output of φ.

To see the second item, let ρ be as in the previous paragraph. For any deterministic formula $\chi(x, \vec{w})$ we have that $\sum_{\rho(\vec{w})} \chi$ is an FO + Poly + Sum query returning the sum of the x-values of χ over the output of φ. The average of φ is simply the quotient of the sum of φ and the cardinality of φ. Since the FO + Poly definable functions are closed under division, we can define average. □

6 Computing the volume of Semi-linear sets in FO + Poly + Sum

In this section we show how to use the aggregate language FO + Poly + Sum for volume computation and approximation. Our goal is to prove that FO + Poly + Sum can compute the volume of semi-linear sets. We start by noting that taking volumes of semi-linear sets does not take us out of the semi-algebraic setting. This fact is easily derived from known results in the literature (and may have been published before, see, for example, [8] for a closely related result).

Lemma 3 *For any formula $\varphi(\vec{x}, \vec{y})$ over the real ordered group $\mathbf{R}_{\text{lin}}$, the volume of φ is semi-algebraic. That is, $\{\vec{r}, s \mid [\textsc{Vol}\ \vec{y}.\varphi(\vec{x}, \vec{y})](\vec{r}, s)\}$ is a semi-algebraic set.*

Proof. By Fubini's Theorem, $[\textsc{Vol}\ \vec{y}.\varphi(\vec{x}, \vec{y})](\vec{x}, z)$ holds exactly when $z = \int \int \ldots \int \chi_\varphi(\vec{x}, \vec{y}) dy_n \ldots dy_1$, where χ_φ is the characteristic function of the set defined by φ.

Let $F_1(y_1 \ldots y_{n-1}, \vec{x})$ be the innermost integral $\int \chi_\varphi(\vec{y}, \vec{x}) dy_n$. We first show that $F_1(y_1 \ldots y_{n-1}, \vec{x})$ is semi-algebraic. Let $l_i(y_1 \ldots y_{n-1}, \vec{x})$ and $u_i(y_1 \ldots y_{n-1}, \vec{x})$ be the ith lower and upper endpoint of the set $\varphi_{\vec{x}, y_1 \ldots y_{n-1}} = \{y_n \mid \varphi(\vec{x}, y_1 \ldots y_n)\}$. We know that u_i and l_i are semi-linear definable partial *functions*. We now note that any such function is piecewise linear with the coefficients in the linear polynomial being rational, cf. [35]. That is, for each function, its domain can be partitioned into finitely many semi-linear sets on which it is linear. To see this, note that on its domain U_i, $u_i(y_1 \ldots y_n, \vec{x})$ is the unique solution to a disjunction of conjunctions of linear inequalities in $y_1, \ldots, y_n, \vec{x}$. Each disjunct must then have at most one solution. Let a disjunct be a conjunction $\bigwedge_{l \in T_1} C_l(y_1 \ldots y_n, \vec{x}) \theta 0$, where $\theta \in \{<, >, \leq, \geq\}$. We know that this must have at most one solution r_n for each $r_1 \ldots r_{n-1}, \vec{s} \in U_i$. But this solution must then be the solution to the conjunction of some subset of the corresponding equalities $C_l(y_1 \ldots y_n, \vec{x}) = 0$ where $l \in T_2 \subset T_1$. (Otherwise fix a counterexample $r_1 \ldots r_{n-1}, \vec{s}$ and let T_2 be the set of $l \in T_1$ such that the solution r_n satisfies $C_l(r_1 \ldots r_n, \vec{s}) = 0$. If the corresponding solution space is not 0-dimensional, then the set of proper inequalities of the form $C_l(y_1 \ldots y_n, \vec{x})\{<, >\}0$ with $l \in T_1 - T_2$ satisfied by $\vec{r}, \vec{s}$ defines an open subset of this space, which would then have to be infinite or empty, giving a contradiction.) But by linear algebra, we know that when a set of linear equalities $C_l(y_1 \ldots y_n, \vec{x})$ has a unique solution y_n, this solution is given by a linear function with coefficients in the field generated by $y_1 \ldots y_{n-1}, \vec{x}$. Hence piecewise u_i is linear, and similarly for l_i.

Hence we can find a decomposition of $\mathbb{R}^{m+n-1}$ into semilinear sets $A_1 \ldots A_k$, and find a function $b : k \to N$ and linear functions $f_{ij}(y_1 \ldots y_{n-1}, \vec{x}) : i \leq k, j \leq b(i)$ such that

$$\forall r_1 \ldots r_{n-1} s_1 \ldots s_m \in A_i \quad F_1(r_1 \ldots r_{n-1}, s_1 \ldots s_m) = \Sigma_{k \leq b(i)} f_{ij}(r_1 \ldots r_{n-1}, s_1 \ldots s_m).$$

But now we have that $[\textsc{Vol}\ \vec{y}.\varphi(\vec{x}, \vec{y})](\vec{x}, z)$ holds when $z = \int \int \ldots \int F_1(x_1 \ldots x_{n-1}, \vec{y}) dx_1 \ldots dx_{n-1}$, so we can partition $\mathbb{R}^m$ into finitely many pieces, on each one of which $[\textsc{Vol}\ \vec{y}.\varphi(\vec{x}, \vec{y})](\vec{x}, z)$ is given by the graph of a polynomial in $\vec{x}$. Hence $\textsc{Vol}\ \vec{y}.\varphi(\vec{x}, \vec{y})$ is semi-algebraic. □

We now prove that the language FO + POLY + SUM can express volumes of semi-linear sets.

Theorem 5
- *For every schema predicate $S \in SC$ there is an* FO + POLY + SUM *term τ which, for any semi-linear database D, computes the volume of S in D.*
- *For every* FO + LIN *query φ there is an* FO + POLY + SUM *term τ_φ such that for any semi-linear database D, $\tau_\varphi(D)$ returns the volume of $\varphi(D)$.*

Proof: Note that the first item clearly implies the second, because, given such a term τ we can compose it with the query φ to get the necessary term in the second item. Hence we only prove the first item here.

For any semi-linear S we have $\text{VOL}(S) = \int\int\ldots\int \chi_S(\vec{x})dx_n\ldots dx_1$, where χ_S is the characteristic function. The innermost integral is $[\sum_{\rho_1(w,x_1\ldots x_{n-1})}\gamma](x_1\ldots x_{n-1})$, where $\rho_1(w, x_1\ldots x_{n-1})$ is the query saying w is the sum of difference of consecutive endpoints of the set $\{x_n \mid S(x_1\ldots x_{n-1}, x_n)\}$, and $\gamma(w) \equiv (w = w)$. Note that by o-minimality, ρ_1 is an FO + POLY + SUM query mapping semi-algebraic sets to finite sets. The proof of Proposition 7 shows that any such query can be written as a range-restricted expression in FO + POLY + SUM.

Let $f^1_{x_1\ldots x_{n-1}} = [\sum_{\rho_1(w,x_1\ldots x_{n-1})}\gamma](x_1\ldots x_{n-1})$. We know from the proof of Lemma 3 that for each fixed $r_1,\ldots,r_{n-2}$, the function $g^1_{r_1,\ldots,r_{n-2}}(x_{n-1}) = f^1(r_1,\ldots,r_{n-2},x_{n-1})$ is piecewise a linear function of x_{n-1}. Since f^1 is an FO + POLY + SUM definable function, we can also define in FO + POLY + SUM the set of points $\{r_1,\ldots,r_{n-2},r_{n-1}$: the function $g^1_{r_1,\ldots,r_{n-2}}$ is not smooth at $r_{n-1}\}$ We can do this because a piecewise linear function is smooth whenever it is differentiable, and the latter property can be tested by an FO + POLY query.

Let $f^2(x_1,\ldots,x_{n-2})$ be the sum of all values of the function $(mu^2 - ml^2)/2 + b(u-l)$, where the quadruples (u,l,m,b) vary over all quadruples of points such that (l,u) are consecutive points of nonsmoothness of $g^1_{x_1,\ldots,x_{n-2}}$, and $g^1_{x_1,\ldots,x_{n-2}} = mx + b$ on the interval (l,u).

Note that since $g^1_{x_1,\ldots,x_{n-2}}$ is piecewise linear, there are only finitely many points where f^1 is not smooth, hence only finitely many pairs of consecutive points of nonsmoothness. Therefore there are only finitely many quadruples (u,l,m,b) as above. Also note that the formula $\gamma(w,l,u,m,b)$ given by $w = (mu^2 - ml^2)/2 + b(u-l)$ is a deterministic formula. Hence, by Proposition 7, there is an FO + POLY + SUM query returning the sum of all γ output values w as (l,u,m,b) vary. Hence $f^2(x_1,\ldots,x_{n-2})$ is an FO + POLY + SUM definable function.

Claim 1 *$f^2(x_1\ldots x_{n-2})$ is exactly the volume of the fiber of S based on $x_1,\ldots,x_{n-2}$. That is, $f^2(x_1\ldots x_{n-2}) = \text{VOL}(\{(x_{n-1},x_n) \mid (x_1,\ldots,x_{n-2},x_{n-1},x_n) \in S\})$.*

Proof:. By Fubini's theorem, the volume is the integral of the one variable function $g^1_{x_1,\ldots,x_{n-2}}(x_{n-1})$. Since this function is piecewise linear, for each fixed $r_1,\ldots,r_{n-2}$ there are finitely many $a_1,\ldots,a_k \in \mathbb{R}\cup\{\infty,-\infty\}$ with $a_1 < \ldots < a_k$ such that g^1 is linear on (a_j,a_{j+1}). Hence the integral of $g^1_{r_1\ldots r_{n-2}}$ is just the sum of the integral of g^1 over the intervals (a_j,a_{j+1}). But the integral of a linear function $h(x) = mx + b$ over an interval l,u is just $mx^2/2 + bx|^u_l$, and hence the result follows. □

Continuing this inductively, we have the function $f^{k-1}(x_1,\ldots,x_{n-k+1})$ giving the volume of the fiber of S defined by $x_1,\ldots,x_{n-k+1}$. If we fix the first $n-k$ parameters in this function, we obtain a function $g^{k-1}_{x_1,\ldots,x_{n-k}}(x_{n-k+1})$ which is piecewise polynomial of degree at most $k-1$. That is, $\mathbb{R}$ is

partitioned into finitely many intervals, and on each of them $g^{k-1}_{x_1,\ldots,x_{n-k}}(y)$ is give by $b_{k-1}y^k + \ldots + b_0$. One can now determine all the points of nonsmoothness (since this is the same as not being $k-1$-times differentiable) of $g^{k-1}_{x_1,\ldots,x_{n-k}}$ by a FO + POLY + SUM query. Furthermore, one can write a query, using polynomial constraints, that on every point in every interval between the points of nonsmoothness finds the coefficients of the polynomial of degree $k-1$ that gives $g^{k-1}_{x_1,\ldots,x_{n-k}}$ on every such interval (e.g., by computing the derivatives and applying Taylor's theorem). Thus, we have a range-restricted FO + POLY + SUM query $\rho_k(b_{k-1}, b_{k-2}, \ldots, b_0, u, l, x_1, \ldots, x_{n-k})$ that for every $x_1, \ldots, x_{n-k}$ produces the tuples $(b_{k-1}, b_{k-2}, \ldots, b_0, u, l)$ such that on (u, l), $g^{k-1}_{x_1,\ldots,x_{n-k}}$ is given by the polynomial $b_{k-1}y^{k-1} + \ldots + b_0$, and furthermore (u, l) list all such intervals, which cover all $\mathbb{R}$ except finitely many points of nonsmoothness.

Now let $\gamma_k(b_{k-1}, b_{k-2}, \ldots, b_0, u, l)$ be defined by

$$\frac{b_{k-1}(u^k - l^k)}{k} + \frac{b_{k-2}(u^{k-1} - l^{k-1})}{k-1} + \ldots + b_0(u - l)$$

Hence, $f^k(x_1 \ldots x_{n-k})$ given by

$$[\sum_{\rho_k(b_{k-1}, b_{k-2}, \ldots, b_0, u, l, x_1, \ldots, x_{n-k})} \gamma_k(b_{k-1}, b_{k-2}, \ldots, b_0, u, l)] \quad (x_1, \ldots, x_{n-k})$$

defines, for each $(x_1, \ldots, x_{n-k})$,

$$\int g^{k-1}_{x_1 \ldots x_{n-k}}(x_{n-k+1}) dx_{n-k+1},$$

and thus by Fubini's theorem it is the volume of the fiber of S over $x_1 \ldots x_{n-k}$.

Now it immediately follows that f^n is a FO + POLY + SUM function giving the volume of S. Theorem 5 is proved. □

7 Conclusions

This paper has dealt with the key question of how to add aggregation to constraint query languages. The first fundamental question is whether there can be a language that is closed under the natural spatial aggregation operators, and which also retains the basic closure property that is fundamental to a constraint-based approach: namely, that every query output can be again represented as a constraint solution set. Our results give indication that this is impossible: these two closure properties are fundamentally incompatible. Perhaps more surprisingly, we show that the problem is not particular to the polynomial or linear constraint model; even going to a larger well-behaved constraint set does not remedy the problem.

The results above motivated us to look for languages that are not closed under volume operators, but which are closed under natural discrete aggregations and which permit the computation of volumes for semi-linear sets. The language FO + POLY + SUM defined here gives a natural approach to the addition of discrete aggregation operators to a constraint language. The key idea is the notion of range-restricted querying: allowing aggregation to be formed only on sets that are guaranteed to be finite. We show not only that FO + POLY + SUM has some attractive closure properties analogous to classical aggregate languages, but it allows one to do a significant amount of spatial aggregation — e.g. volumes of semi-linear sets, averages over semi-linear sets — as well.

The approach given here based on classical summation over range-restricted sets is natural, and allows one to re-use many of the evaluation strategies for classical aggregation operators; it is clear, however, that the syntax given here for FO + POLY + SUM is quite awkward. We hope to find more streamlined and natural syntax for FO + POLY + SUM, and we are looking at subsets of FO + POLY + SUM that can be more efficiently evaluated than the full language. It remains to discover how one could best provide support for directly expressing volumes in some language built 'on top of' FO + POLY + SUM, and how to add grouping constructs to the language.

A challenging issue on the theoretical side is how to prove expressive bounds on aggregate constraint database languages like FO + POLY + SUM. For example, the results of this paper give strong evidence that FO + POLY + SUM does *not* suffice to calculate volumes of semi-algebraic sets, but this is at this point only a conjecture.

Acknowledgements Part of this work was done while the second author was visiting INRIA. We thank Serge Abiteboul, Stéphane Grumbach, Michel Scholl and Luc Segoufin for helpful discussions. Libkin thanks all the members of the Verso team for their hospitality.

References

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

[2] M. Anthony and N. Biggs. *Computational Learning Theory.* Cambridge Univ. Press, 1992.

[3] O. Belagradek, A. Stolboushkin, M.Tsaitlin. Extended order-generic queries. *APAL*, to appear.

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

[5] M. Benedikt and L. Libkin. Relational queries over interpreted structures. *Journal of the ACM*, to appear. Extended abstract in *PODS'97*, pages 87–98.

[6] M. Benedikt and L. Libkin. Safe constraint queries. *SIAM J. Comput.*, to appear. Extended abstract in *PODS'98*, pages 99–108.

[7] M. Benedikt and L. Libkin. Exact and approximate aggregation in constraint query languages. In *PODS'99*, pages 102–113.

[8] H. Bieri and W. Nef. A sweep-plane algorithm for computing the volume of polyhedra represented in Boolean form. *Linear Algebra and Its Applications* 52/53 (1983), 69–97.

[9] A. Blumer, A. Ehrenfeucht, D. Haussler, M. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM* 36 (1989), 929–965.

[10] J. Chomicki, D. Goldin and G. Kuper. Variable independence and aggregation closure. In *PODS'96*, pages 40–48.

[11] J. Chomicki and G. Kuper. Measuring infinite relations. In *PODS'95*, pages 78–85.

[12] L. Denenberg, Y. Gurevich and S. Shelah. Definability by constant-depth polynomial-size circuits. *Information and Control* 70 (1986), 216–240.

[13] M. Dyer and A. Frieze. On the complexity of computing the volume of a polytope. *SIAM J. Comput.*, 17 (1988), 967–974.

[14] M. Dyer, A. Frieze and R. Kannan. A random polynomial-time algorithm for approximating the volume of convex bodies. *Journal of the ACM*, 38 (1991), 1–17.

[15] P. Goldberg and M. Jerrum. Bounding the Vapnik Chervonenkis dimension of concept classes parameterized by real numbers. *Machine Learning* 18 (1995), 131–148.

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

[17] S. Grumbach, M. Rafanelli and L. Tininini. Querying aggregate data. In *PODS'99.*

[18] S. Grumbach, P. Rigaux, L. Segoufin. The DEDALE system for complex spatial queries. In *SIGMOD'98*, pages 213–224.

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

[20] S. Grumbach and J. Su. Queries with arithmetical constraints. *Theoretical Computer Science* 173 (1997), 151–181.

[21] Y. Gurevich. Toward logic tailored for computational complexity. In *Computation and Proof Theory*, Springer, 1984, pages 175–216.

[22] P. Kanellakis, G. Kuper, and P. Revesz. Constraint query languages. *JCSS*, 51 (1995), 26–52. Extended abstract in *PODS'90*, pages 299–313.

[23] M. Karpinski and A. Macintyre. Approximating the volume of general Pfaffian bodies. In *Structures in Logic and Computer Science: A Selection of Essays in Honor of A. Ehrenfeucht*, Springer LNCS 1261, 1997, pages 162–173.

[24] M. Karpinski and A. Macintyre. Approximating volume and integrals in o-minimal and p-minimal theories. Technical Report, University of Bonn, 1997.

[25] P. Koiran. Approximating the volume of definable sets. In *FOCS'95*, pages 134–141.

[26] G. Kuper. Aggregation in constraint databases. In *PPCP'93*, 166–173.

[27] G. Kuper, L. Libkin and J. Paredaens, eds. *Constraint Databases.* Springer Verlag, 1999.

[28] G. Kuper, S. Ramaswamy, K. Shim, J. Su. A constrint-based spatial extension to SQL. In *Proceedings of ACM-GIS*, ACM Press, 1998, pages 112–117.

[29] M. C. Laskowski. Vapnik-Chervonenkis classes of definable sets. *J. London Math. Soc.*, 45:377–384, 1992.

[30] C. Papadimitriou and M. Yannakakis. On limited nondeterminism and the complexity of the V-C dimension. *JCSS* 53 (1996), 161–170.

[31] J. Paredaens, J. Van den Bussche, and D. Van Gucht. First-order queries on finite structures over the reals. *SIAM J. Computing* 27 (1998), ???-???.

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

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

[34] S. Shelah. Stability, the f.c.p., and superstability. *Ann. of Math. Logic* 3 (1971), 271–362.

[35] L. van den Dries. *Tame Topology and o-minimal Structures.* Cambridge Univ. Press, 1998.

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

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

[38] M. Worboys. *GIS: A Computing Perspective.* Taylor & Francis, 1995.