# Constraint Databases, Data Structures and Efficient Query Evaluation*

Joos Heintz[1,2,3] and Bart Kuijpers[4]

[1] Universitat de Buenos Aires
Departamento de Computación
Ciudad Universitaria, Pabellón I, 1428 Buenos Aires, Argentina
joos@dc.uba.ar
[2] Consejo Nacional de Investigaciones
Científicas y Tecnológicas (CONICET), Argentina
[3] Universidad de Cantabria
Facultad de Ciencias
Departamento de Matemáticas, Estadística y Comptación
Avda. de los Castros s/n, E–39005 Santander, Spain
[4] Limburgs Universitair Centrum
Department WNI
Universitaire Campus, 3590 Diepenbeek, Belgium
bart.kuijpers@luc.ac.be

**Abstract.** Constraint databases that can be described by boolean combinations of polynomial inequalities over the reals have received ample research attention. In particular, the expressive power of first-order logic over the reals, as a constraint database query language, has been studied extensively. The difficulty of the effective evaluation of first-order queries, usually involving some form of quantifier elimination, has been largely neglected.

The contribution of this paper is a discussion of various aspects that influence the efficiency of the evaluation of queries expressible in first-order logic over the reals. We emphasize the importance of *data structures* and their effect on the complexity of quantifier-elimination. We also propose a novel data model that supports data exploration and visualization as well as efficient query evaluation. In this context, we introduce the concept of *sample point query*. Finally, we show that a particular kind of sample point query cannot be evaluated in polynomial sequential time by means of branching-parsimonious procedures.

## 1 Introduction and summary

The framework of *constraint databases* was introduced in 1990 by Kanellakis, Kuper and Revesz [26] as an extension of the relational model that allows the

* Research partially supported by the following Argentinian, Belgian, German and Spanish grants: UBACyT X198, PIP CONICET 2461, FW/PA/02–EIII/007, ALA 01–E3/02 and DGCyT BFM 2000–0349.

use of possibly infinite, but first-order definable relations rather than just finite relations. It provides an elegant and powerful model for applications that deal with infinite sets of points in some real affine space $\mathbb{R}^n$. In the setting of the constraint model, infinite relations are finitely represented as boolean combinations of polynomial equalities and inequalities, which we interpret, in this paper, over the real and exceptionally over the complex numbers. The case of the interpretation over the real numbers has applications in spatial databases [31].

Various aspects of the constraint model are well-studied by now (for an overview of research results we refer to [28] and the textbook [33]). The relational calculus augmented with polynomial constraints, or equivalently, first-order logic over the reals augmented with relation predicates to address the database relations $R_1, \ldots, R_s$, $\mathsf{FO}(+, \times, <, 0, 1, R_1, \ldots, R_s)$ for short, is the standard first-order query language for constraint databases. The expressive power of first-order logic over the reals, as a constraint database query language, has been studied extensively. However, the difficulty of the efficient evaluation of first-order queries, usually involving some form of quantifier elimination, has been largely neglected. The existing constraint database systems are based on general purpose quantifier-elimination algorithms and are, in most cases, restricted to work with linear data, i.e., they use first-order logic over the reals without multiplication [28, Part IV].

The intrinsic inefficiency of quantifier elimination represents a bottle-neck for real-world implementations of constraint database systems. General purpose elimination algorithms (such as, e.g., [6, 12, 18, 23, 32]) and standard data structures prevent query evaluation to become efficient. The fact that the knapsack problem can be formulated in this setting indicates that geometric elimination may be intrinsically hard. Another example for this complexity phenomenon is given by the permanent of a generic $n \times n$ matrix, which appears as the output of a suitable first-order query (see [22] for details on both examples).

In the literature of constraint databases, the data model proposed to describe geometric figures in $\mathbb{R}^n$ is based on quantifier-free first-order formulas over the reals [28, 33]. The data structures needed to implement this data model are left largely unspecified. It is widely understood that these formulas should be represented by explicitly giving disjunctive normal forms using dense or sparse encoding of polynomials. However, disjunctive normal forms may be unnecessarily large and the sparse representation of elimination polynomials may be very inefficient. For example, the sparse representation of the determinant of a generic $n \times n$ matrix contains $n!$ terms. On the other hand, the determinant can be represented by an $O(n^3)$ arithmetic boolean circuit (with divisions) which describes the Gaussian elimination algorithm. This suggests the use of alternative data structures for the representation of the classical data model of constraint databases. Indeed, the use of *arithmetic boolean circuits* as alternative data structure allows the design of a new generation of elimination algorithms which produce an exponential time complexity gain compared to the most efficient algorithms using traditional data structures (see [36] for a definition of the notion of arithmetic boolean circuit and [22] for this kind of complexity results). Nevertheless, in terms of the syntactical size of the input, the worst-case sequential time complexity of the elimination of a single existential quantifier block by means of the new algorithms remains still exponential. However, when we measure the input in a semantic (i.e., geometric) way, as is achieved, e.g., by the *system degree*, elimination of a single existential quantifier block becomes polynomial in this quantity (see e.g. [3, 4, 13, 15]). Unfortunately, this does not suffice for the design of algorithms able to evaluate purely existential queries in sequential time which is polynomial in the number of bounded variables. In fact, an non-polynomial lower bound for the underlying elimination problem can be deduced from the $P_{\mathbb{R}} \neq NP_{\mathbb{R}}$ conjecture in the Blum-Shub-Smale complexity model over the real numbers [8].

Another shortcoming of the classical data model for constraint databases is that it does not support data exploration and local visualization. Indeed, a quantifier-free formula in disjunctive normal form, describing the output of a query, allows the answering of, for instance, the membership question, but it is does not allow an easy exhibition of the output, by, e.g., the production of *sample points*, or, for low dimensions, a visualization of the output. To increase the tangibility of the output, we suggest considering a new type of query that produces sample points. Furthermore, it could be desirable to support an exploration of the neighborhood of such a sample point. This could be achieved by representing the output by a cell decomposition consisting of cells which are non-empty open subsets of smooth real varieties. In this way, starting from any sample point, its neighborhood within its cell may be explored. In this sense, we propose a novel data model for constraint databases, consisting of *smooth cells accompanied by sample points*. The known most efficient elimination procedures produce naturally such output representations, a typical example being CAD [12]. Therefore, when constraint database theory invokes quantifier elimination in query evaluation, it should also incorporate these features of the existing elimination algorithms.

In this context, we extend the concept of sample point query to queries that give rationally parameterized families of polynomial functions as output. Such queries will be called *extended sample point queries*. The main outcome of the paper is a proof that extended sample point queries, associated to first-order formulas containing a *fixed* number of quantifier alternations, cannot be evaluated in polynomial sequential time by so-called "branching-parsimonious algorithms". This lower bound result suggest that further research on the complexity of query evaluation in constraint database theory should be directed towards the identification of database and query classes that have a strongly improved complexity behavior. As a pattern for the development of such a theory, we may consider a new type of elimination algorithms which are based on the notion of system degree and use non-conventional data structures (see [2–4, 13, 15, 17, 20, 21, 24, 30, 34]).

This paper introduces a number of new concepts for constraint database theory that sometimes require certain notions from algebraic complexity theory, algebraic geometry and commutative algebra. These notions can be found in standard textbooks, such as [10] (algebraic complexity theory), [1] (commutative algebra) and [35] (algebraic geometry). The reader only interested in database issues may read this paper while skipping these technical details (and in particular the rather involved proof of Theorem 1 below).

The remainder of this paper is organized as follows. In Section 2, we discuss some properties of the classical data model for constraint databases and the impact of these properties on the complexity of query evaluation. In Section 3, we introduce a novel data model and the concept of sample point query. Then, we extend this concept to queries that return rationally parameterized algebraic families of polynomial functions as output. Furthermore, we introduce for this type of extended sample point queries the notion of a branching-parsimonious query evaluation algorithm. In Section 4, we prove that branching-parsimonious algorithms for extended sample point queries, associated to first-order formulas with a fixed number of quantifier alternations, require exponential sequential execution time. In Section 5, we discuss further directions of research in efficient query evaluation for constraint databases.

## 2 A discussion of the classical data model of constraint databases and its data structures

In the current theory, a constraint database consists of a finite number of relations that are interpreted as geometric figures in some space $\mathbb{R}^n$. Quantifier-free first-order formulas over the reals are used to store these relations in the database [28, 33]. The actual data structures needed to implement this data model[5] are only partially specified: the quantifier-free formulas are supposed to be given in disjunctive normal form and the polynomials appearing in them are supposedly given in dense or sparse representation. Geometric data in $\mathbb{R}^n$ are therefore described by expressions of the form

$$\bigvee_{i=1}^{d} \bigwedge_{j=1}^{c_i} p_{ij}(x_1, \ldots, x_n) \; \theta_{ij} \; 0,$$

where the $p_{ij}(x_1, \ldots, x_n)$ are polynomials with integer coefficients in the real variables $x_1, \ldots, x_n$ and where $\theta_{ij} \in \{\leq, \geq, <, >, =, \neq\}$. The sets that can be described in this way are known as *semi-algebraic sets* [9]. For example, the spatial figure consisting of the set of points on the northern hemisphere together with the points on the equator and the south pole of the unit sphere in the three-dimensional space $\mathbb{R}^3$ can be represented by the formula $(x^2 + y^2 + z^2 = 1 \wedge z \geq 0) \vee (x = 0 \wedge y = 0 \wedge z = -1)$.

The standard query language $\mathsf{FO}(+, \times, <, 0, 1, R_1, \ldots, R_s)$ for constraint databases is first-order logic over the reals augmented with relation predicates

[5] In the sequel, we use the terms *data model* and *data structure* in the way as above: the former refers to a conceptual notion whereas the latter refers to the structures that implement this notion.

$R_1, \ldots, R_s$ to address the relations that appear in the input database. When we want to emphasize that a relation predicate $R$ has arity $k$, we write $R^{(k)}$. As an example of a first-order query, we take the $\mathsf{FO}(+, \times, <, 0, 1, R^{(3)})$-sentence $(\exists r)(\forall x)(\forall y)(\forall z)(R(x, y, z) \rightarrow x^2 + y^2 + z^2 < r^2)$ expresses that the three-dimensional spatial relation $R^{(3)}$ is bounded. Query evaluation based on quantifier elimination in elementary geometry guarantees that output relations defined by $\mathsf{FO}(+, \times, <, 0, 1, R_1, \ldots, R_s)$-queries can again be described within this data model.

### 2.1 Alternative data structures for quantifier-free formulas

Let us now make a number of straightforward comments on the choice of this data model and data structure. It is clear that quantifier-free formulas give a effective way to check *membership* (e.g., to query outputs). In fact, the main aim of quantifier elimination in logic is to produce an effective method for checking membership. From the point of view of complexity of query evaluation, insisting on disjunctive normal form and dense or sparse representation of polynomials has a number of obvious disadvantages. For example, given a relation in disjunctive normal form, its complement may require a description that contains a number of disjuncts that grows exponentially in dimension of the ambient space (see [25] for upper and lower bounds on the number of disjuncts).

As a consequence of Bézout's theorem, the elimination of a block of $n$ existential quantifiers in a formula containing polynomials of degree at most $d$, may produce output polynomials of degree $d^n$. For an illustration of this phenomenon, we refer to Example (2) in Section 2.2.

Consider now the family of queries, indexed by $n$ ($n = 1, 2, \ldots$), expressed by the formulas $\Phi_n(a_{11}, \ldots, a_{nn})$ given as

$$(\exists x_1) \cdots (\exists x_n)(R(a_{11}, \ldots, a_{nn}, x_1, \ldots, x_n) \wedge \bigvee_{i=1}^{n} x_i \neq 0)$$

in $\mathsf{FO}(+, \times, <, 0, 1, R^{(n^2+n)})$. When the query expressed by $\Phi_n(a_{11}, \ldots, a_{nn})$ is applied to the input database

$$A_n = \{(\alpha_{11}, \ldots, \alpha_{nn}, v_1, \ldots, v_n) \in \mathbb{R}^{n^2+n} \mid \bigwedge_{i=1}^{n} \sum_{j=1}^{n} \alpha_{ij} v_j = 0\},$$

we obtain, after plugging in the definition of $A_n$ in the description of $\Phi_n(a_{11}, \ldots, a_{nn})$, a first-order sentence expressing that the determinant of the matrix $(\alpha_{ij})_{1 \leq i,j \leq n}$ is zero. This example shows that elimination polynomials may become dense even when their degrees are moderate. This makes the use of dense or sparse representation of polynomials is unsuitable for query evaluation.

The problem of the exploding representations leads to the idea of changing the data structure. This suggests the use of *arithmetic boolean circuits* for the representation of quantifier-free formulas. In order to illustrate this idea, let us observe that the Gaussian elimination algorithm realizes an $O(n^3)$ size arithmetic boolean circuit (with divisions) which decides whether a given inhomogeneous linear $n \times n$ equation system has a solution.

### 2.2 The influence of the choice of data structure on quantifier elimination

We want to argue that complexity theory for geometric elimination requires *simultaneous* optimization of data structures *and* algorithms. To illustrate this point, let us consider the family of the first-order formulas

$$(\exists x_1) \cdots (\exists x_n)(x_1 = t + 1 \wedge R(x_1, x_2) \wedge \cdots \wedge R(x_{n-1}, x_n) \wedge y = x_n^2),$$

indexed by $n$ ($n = 1, 2, \ldots$). When we apply these queries to the binary relation $A = \{(v_1, v_2) \in \mathbb{R}^2 \mid v_1^2 = v_2\}$, after plugging in the description of $A$ in these formulas, we obtain the formulas $\Phi_n(t, y)$, given by

$$(\exists x_1) \cdots (\exists x_n)(x_1 = t + 1 \wedge x_1^2 = x_2 \wedge \cdots \wedge x_{n-1}^2 = x_n \wedge y = x_n^2). \quad (1)$$

The formula $\Phi_n(t, y)$ is logically equivalent to the following quantifier-free formula $\Psi_n(t, y)$:

$$y = \sum_{i=0}^{2^n} \binom{2^n}{i} t^i.$$

If we choose as data structure the dense or sparse representation of polynomials by their coefficients, then $\Phi_n(t, y)$ has length $O(n)$, whereas the length of $\Psi_n(t, y)$ is $O(2^n)$. When we use division-free arithmetic circuits (or straight-line programs) as data structure, then both $\Phi_n(t, y)$ and $\Psi_n(t, y)$ have length $O(n)$, since the polynomial $(t+1)^{2^n}$ can be evaluated using an iteration of $n$ squarings (see [10] for a definition of the notions of arithmetic circuits and straight line program).

Unfortunately, one can show that general-purpose elimination algorithms, satisfying reasonable restrictions on the quality of the output representations, are not able to produce systematically polynomial size output descriptions. This complexity result holds for any continuous data structure, including the representation of polynomials by division-free arithmetic circuits (see [11]). With respect to upper complexity bounds for elimination algorithms, the state of the art is illustrated by the following simple knapsack-like example of an elimination problem. Consider the family of formulas

$$(\exists x_1) \cdots (\exists x_n)(R(x_1) \wedge \cdots \wedge R(x_n) \wedge y = u_1 x_1 + \cdots + u_n x_n),$$

indexed by $n$ ($n = 1, 2, \ldots$). When applying these queries to the unary constraint relation $A = \{(v) \in \mathbb{R} \mid v^2 - v = 0\}$, we obtain the first-order formulas $\Phi_n(y, u_1, \ldots, u_n)$:

$$(\exists x_1) \cdots (\exists x_n)(x_1^2 - x_1 = 0 \wedge \cdots \wedge x_n^2 - x_n = 0 \wedge y = u_1 x_1 + \cdots + u_n x_n). \quad (2)$$

This elimination problem has the following canonical quantifier-free output formula $\Psi_n(y, u_1, \ldots, u_n)$:

$$\prod_{(\varepsilon_1,\ldots,\varepsilon_n)\in\{0,1\}^n} (y - (\varepsilon_1 u_1 + \cdots + \varepsilon_n u_n)) = 0. \tag{3}$$

Using the dense or sparse representation of polynomials, the formula $\Psi_n(y, u_1, \ldots, u_n)$ takes $O(2^{n^2})$ space. However, checking membership by $\Psi_n(y, u_1, \ldots, u_n)$ requires only $O(2^n)$ arithmetical operations. The standard dense representation based general-purpose algorithms, cited in the constraint database literature [28], require at least $O(2^{n^2})$ sequential time to eliminate the $n$ existential quantifiers in the input formula, whereas the use of the arithmetic boolean circuit representation leads to a sequential time complexity of $O(2^n)$ and therefore produces an exponential complexity gain (compare, e.g., [16, 22]). However, this example shows that in terms of the syntactical size of the input formula, the worst-case sequential time complexity of the elimination of a single existential quantifier block remains still exponential.

Let us make the following observations: specializing the free variables $u_1, \ldots, u_n$ and $y$ of formula (2) into arbitrary non-negative integer values $\alpha_1, \ldots, \alpha_n$ and $\beta$, the resulting formula $\Phi_n(\beta, \alpha_1, \ldots, \alpha_n)$ becomes equivalent to the statement that there exists a set $I \subseteq \{1, \ldots, n\}$ with $\sum_{i\in I} \alpha_i = \beta$. Therefore, the closed formula $\Phi_n(\beta, \alpha_1, \ldots, \alpha_n)$ defines the knapsack problem given by the instance $\alpha_1, \ldots, \alpha_n$ and $\beta$.

On the other hand, specializing the free variables $u_1, \ldots, u_n$ into the integer values $2^0, \ldots, 2^{n-1}$ the elimination polynomial contained in the formula (3) becomes the well-known Pochhammer-Wilkinson polynomial

$$\prod_{0\le j<2^n} (y - j),$$

whose arithmetic circuit complexity is has been open since the times of Euler. In this context, let us mention that the polynomial $\prod_{0\le j<2^n}(y - \sqrt{j})$ requires $2^{\Omega(n)}$ arithmetic operations for its evaluation (see [22]). However, this polynomial has algebraic coefficients and therefore it cannot be produced by an elimination problem.

Altogether, the above examples reflect adequately the main complexity issues in elimination theory over the complex and real numbers both from the point of view of upper and lower sequential time complexity bounds (compare [14, 16, 11]).

### 2.3 Where does the exponentiality of elimination algorithms come from?

The subformula $x_1 = t + 1 \wedge x_1^2 = x_2 \wedge \cdots \wedge x_{n-1}^2 = x_n$ in (1) defines for each value of $t$ just one point of the $n$-dimensional affine ambient space. On the other hand, the second system $x_1^2 - x_1 = 0 \wedge \cdots \wedge x_n^2 - x_n = 0$ in (2) has $2^n$ roots. This observations may be paraphrased as follows: the "system degree" of formula (1) is one and that of formula (2) is $2^n$.

In general, a given algebraic variety may be defined by different systems of polynomial equalities and inequalities. One may associate to each of these systems an invariant, called *system degree*, that is at least as large as the degree of the given variety and depends only on the polynomials and their order in the system but not on their representation (for formal definitions of different variants of the notion of system degree in the algebraic and the arithmetic setting see [3, 4, 13, 15, 20, 27]). The complexity issue of both of the above examples can be explained by a single general fact:

> *the sequential time complexity of elimination of an existential quantifier block is polynomial in the syntactic length of the input formula and its system degree.*

However, the second example shows that this fact does not prevent quantifier elimination to become exponential in the number of existentially quantified variables. This is due to the circumstance that the system degree may become exponential in the syntactical length of the input formula.

The system degree has shown to be a fruitful notion for the complexity analysis of quantifier elimination in elementary geometry. However, for query evaluation in constraint databases, the notion of system degree is still unsatisfactory since it is determined both by the query formula and the quantifier-free formulas describing the input database relations. It is a task for future constraint database research to develop a well-adapted complexity invariant in the spirit of the system degree in elimination theory.

On the other hand, the non-polynomial worst-case complexity character of the elimination of a single block of existential quantifiers can easily be deduced from the $P_{\mathbb{R}} \neq NP_{\mathbb{R}}$ conjecture in the Blum-Shub-Smale complexity model over the real numbers [8].

In Section 4, we show an exponential complexity lower bound, not depending on conjectures, in a slightly extended query model.

## 3 A novel data model for constraint databases

We pass now to another shortcoming of the classical data model for constraint databases, i.e., of using quantifier-free formula in disjunctive normal form (encoded by whatever data structure): a quantifier-free formula in disjunctive normal form, describing, e.g., the output of a query, allows to answer the corresponding membership question, but it is does not allow an easy visualization of the output, i.e., this data model does not directly support data exploration and local visualization.

### 3.1 The role of sample points in constraint databases

It is well known in elementary geometry that geometric figures can be given by equations and inequalities or in parametric form. In the first case, it is easy to decide membership of a given point of the ambient space to the figure whereas it is difficult to produce even a single point belonging to the figure. When the figure is given in parametric form, the exact opposite is true.

A well-known algorithmic problem in elimination theory is the implicitation of a parametrically given geometric figure. The inverse problem is not always solvable in terms of polynomial or rational mappings. For example, the plane curve given by the equation $x^2 - y^3 = 0$ cannot be rationally parameterized. A reasonable half-way solution to overcome this problem is augmenting the implicit representation of a geometric figure with *sample points*.

The existing efficient elimination algorithms produce sample points as a natural byproduct (compare e.g. [3, 4, 6, 18, 23, 32]). A famous example is CAD (see [12]), where sample points are used to determine the truth value of a given formula in a given cell. Therefore, when constraint database theory invokes one of the existing quantifier-elimination algorithms in query evaluation, it can incorporate this feature without loss of complexity.

A sample point of a geometric figure is not just any point belonging to that figure. In elimination theory, the notion of sample point has a specific meaning. Below, we give the definition of the notion of sample point in the context of semi-algebraic geometry (in the context of algebraic geometry over the complex numbers the corresponding definition is slightly different).

**Definition 1.** A *sample point* of a semi-algebraic set $A$ in $\mathbb{R}^n$ is a quantifier-free first-order formula that defines (or encodes) exactly one point $(a_1, \ldots, a_n)$ of $A$. We require that this formula allows us to determine, for any polynomial $p \in \mathbb{Z}[x_1, \ldots, x_n]$, the sign of $p(a_1, \ldots, a_n)$ in a finite number of steps using only arithmetic operations and comparations in $\mathbb{Q}$. □

We will also use the term sample point to refer to the geometric point defined by this quantifier-free formula. A *sample point query* is a computable function that returns on input a semi-algebraic set, represented by a formula, at least one sample point of this set (e.g., one for each of its connected components).

Sample points are computable from a given first-order description of the set $A$. As the following proposition shows, sample points are also first-order definable in the sense that there is a expression that depends on the dimension of the ambient space and that returns as output a point that allows a description in the sense of Definition 1.

**Proposition 1.** *For any integer $n \geq 1$, there exists a formula $\sigma_n(x_1, \ldots, x_n)$ in $\mathsf{FO}(+, \times, <, 0, 1, R^{(n)})$, such for any non-empty semi-algebraic subset $A$ of $\mathbb{R}^n$, the set $\{(a_1, \ldots, a_n) \in \mathbb{R}^n \mid (A, a_1, \ldots, a_n) \models \sigma_n(x_1, \ldots, x_n)\}$ consists of exactly one point that belongs to $A$ and that satisfies the requirements of a sample point as stated in Definition 1.*

*Proof.* We prove this property by induction on $n$. For $n = 1$, the following procedure (returning a number $x_1$) can be expressed by a formula $\sigma_1(x_1)$ in $\mathsf{FO}(+, \times, <, 0, 1, R^{(1)})$: if $R$ equals $\mathbb{R}$ (i.e., $\partial R$ is empty[6]), then return the origin

[6] We denote the topological border of a set $R$ by $\partial R$.

0; else, if $\partial R$ contains exactly one element $p$, then return the minimum of the set $\{p-1, p, p+1\} \cap R$; else, if $\partial R$ contains at least two elements, let $p$ be the smallest and $q$ be the one but smallest element of $\partial R$ and return the minimum of the set $\{p-1, p, \frac{p+q}{2}\} \cap R$.

Clearly, for any non-empty semi-algebraic subset $A$ of $\mathbb{R}$, the set $\{a \in \mathbb{R} \mid (A, a) \models \sigma_1(x)\}$ contains a single point that belongs to $A$. Since the above procedure is $\mathsf{FO}(+, \times, <, 0, 1, R^{(1)})$-expressible, the defined point is algebraic and algebraic points satisfy the requirements of a sample point as stated in Definition 1.

Let $n > 1$ and assume the property holds for all $m$, $1 \leq m < n$. The following procedure (returning the real vector $(x_1, \ldots, x_n)$) can be expressed by a formula $\sigma_n(x_1, \ldots, x_n)$ in $\mathsf{FO}(+, \times, <, 0, 1, R^{(n)})$: let $\pi_1(R)$ be the projection of the subset $R$ of $\mathbb{R}^n$ onto the first coordinate $x_1$ and let $\alpha_1$ be the only element of $\mathbb{R}$ satisfying $\sigma_1(x)$ when applied to $\pi_1(R)$; let $(\alpha_2, \ldots, \alpha_n)$ be the unique vector defined by $\sigma_{n-1}(x_2, \ldots, x_n)$, when applied to the intersection of $R$ and the hyperplane $x_1 = \alpha_1$; return $(\alpha_1, \alpha_2, \ldots, \alpha_n)$.

Clearly, this procedure, when applied to a non-empty semi-algebraic subset $A$ of $\mathbb{R}^n$, defines a single point of $A$. The same argument as above shows that this point satisfies the requirements of a sample point as stated in Definition 1. □

A well-known encoding of sample points is based on Thom's lemma, saying that any real algebraic number may be encoded by listing sign conditions on a suitable polynomial $p \in \mathbb{Z}[x]$ and all its derivatives.

*Optimization problems.* An illustration of the relevance of sample points can be found in optimization theory, where one typically asks the following three questions:

1. Given a system of linear inequalities $\sum_{j=1}^{n} a_{ij}x_j \geq b_i$ $(1 \leq i \leq n)$, determine whether it has a solution, i.e., decide whether the formula

$$(\exists x_1) \cdots (\exists x_n) \bigwedge_{i=1}^{m} \sum_{j=1}^{n} a_{ij}x_j \geq b_i$$

   is true.
2. If a system of linear inequalities $\sum_{j=1}^{n} a_{ij}x_j \geq b_i$ $(1 \leq i \leq n)$ has a non-empty solution set $V$, decide whether a given affine target function $f$ defined by $(x_1, \ldots, x_n) \mapsto \sum_{i=1}^{n} c_i x_i + d$ reaches a finite maximum on $V$.
3. If $f$ reaches such a maximum on $V$, give an example of a point in $V$ that realizes this maximum.

Clearly, given the linear equality system and the affine function in a suitable form as a relation (we refer to the encoding of a system of equations in Section 2.1 as an example of such a representation), Problems 1 and 2 can readily be seen as instances of constraint database queries. Furthermore, these queries are first-order expressible. By Proposition 1, also Problem 3 can be expressed in first-order logic. Problem 3 is an example of a sample point query. These problems show that optimization problems naturally belong to constraint database theory. They also show that enriching implicit descriptions of constraint data with sample points, facilitates the answering of optimization queries.

### 3.2 A novel data model for constraint databases with cell decompositions and sample points

In order to make constraint relations more explicit, we have argued that adding sample points to first-order formulas is important. Furthermore, it can be desirable that the data model supports an exploration of the neighborhood of such a sample point. This can be achieved by representing constraint relations by cell decompositions consisting of cells which are open subsets of smooth real algebraic varieties together with suitable sample points. In this sense, we are going to propose a novel data model for constraint databases. On the other hand, existing efficient elimination procedures produce in a natural way output representations which fit perfectly well in this data model.

In this manner, starting from any sample point, its neighborhood within its cell can be explored.

Let us now make our ideas more precise.

**Definition 2.** Let $A$ be a semi-algebraic set contained in the $n$-dimensional affine space $\mathbb{R}^n$. A *cell decomposition of* $A$ is a finite family $\mathcal{F}_1, \ldots, \mathcal{F}_m$ of data of the following kind. For each $k$, $1 \leq k \leq m$, $\mathcal{F}_k$ is a list consisting of non-zero polynomials of $\mathbb{Q}[x_1, \ldots, x_n]$, namely $f_1^{(k)}, \ldots, f_{s_k}^{(k)}$, with $0 \leq s_k \leq n$, $g_1^{(k)}, \ldots, g_{t_k}^{(k)}$, and $\rho_k$, for $r_k := n - s_k$ a rational $r_k \times n$ matrix $M_k$ of maximal rank and a finite family of sample points $x_1^{(k)}, \ldots, x_{q_k}^{(k)}$ of $A$ such that the following conditions are satisfied:

(i) the equations $f_1^{(k)} = 0, \ldots, f_{s_k}^{(k)} = 0$, intersect transversally in all its common real zeroes where the polynomial $\rho_k$ does not vanish (and such real zeroes exist);

(ii) for $(y_1, \ldots, y_{r_k})^\tau = M_k(x_1, \ldots, x_n)^\tau$ the polynomial $\rho_k$ belongs to $\mathbb{Q}[y_1, \ldots, y_{r_k}]$ and the linear forms $y_1, \ldots, y_{r_k}$ induce a finite unramified morphism from the complex variety defined by the equations $f_1^{(k)} = 0, \ldots, f_{s_k}^{(k)} = 0$ in $\mathbb{C}^n \setminus \{\rho_k \neq 0\}$ onto $\mathbb{C}^{r_k} \setminus \{\rho_k \neq 0\}$;

(iii) the semi-algebraic set defined by the conjunction $f_1^{(k)} = 0 \wedge \ldots \wedge f_{s_k}^{(k)} = 0 \wedge g_1^{(k)} > 0 \wedge \ldots \wedge g_{t_k}^{(k)} > 0 \wedge \rho_k \neq 0$ is non-empty and contained in $A$ and $\dim_x A = r_k$ holds for any point $x$ of $A_k$;

(iv) the sample points $x_1^{(k)}, \ldots, x_{q_k}^{(k)}$ belong to $A_k$ and each connected component of $A_k$ contains at least one of these sample points;

(v) the semi-algebraic set $A$ is the union of all cells $A_k$ $(1 \leq k \leq m)$. $\square$

Remark that the well-known CAD algorithm produces a cell decomposition in the sense of the above definition [12]. However, it usually produces adjacent cells that can be merged still resulting in a cell decomposition.

In view of Subsection 2.1, we think cell decompositions suitably encoded by arithmetic boolean circuits. In particular, we think the polynomials $f_1^{(k)}, \ldots, f_{s_k}^{(k)}$, $g_1^{(k)}, \ldots, g_{t_k}^{(k)}$, and $\rho_k$ represented by division-free arithmetic circuits. Using numerical procedures, it is now clear that cell decompositions allow local visualization of the semi-algebraic sets they represent. In the following, we discuss our new data model rather informally.

The cell decompositions we are proposing as data model for constraint databases should come as close as possible to stratifications. Geometrically speaking, an arbitrary cell decomposition of a semi-algebraic set does not capture adequately the phenomenon of cell degeneration, and, algorithmically speaking, an *arbitrary* arithmetic boolean circuit representation of this cell decomposition may introduce spurious branchings, mainly by the use of membership tests based on algorithms which require divisions. This leads us to the requirement that the arithmetic boolean circuit representations of cell decompositions should be branching-parsimonious (see [11, 14]). In particular, we require that our arithmetic boolean circuits do not contain branchings where traditional or alternative but still efficient membership tests do not introduce them. For instance, testing membership to a determinantal variety may be performed using an arithmetic boolean circuit with branchings implementing the traditional Gaussian elimination method. On the other hand, Berkowitz' polynomial time algorithm for the computation of the determinant allows an efficient alternative and division-free representation of the same variety [7]. In this sense, Berkowitz'algorithm is a branching-parsimonious (in fact, branching-free) substitute for Gaussian elimination.

We call an *algorithm* for query evaluation *branching-parsimonious* if on any input a branching-parsimonious representation of its output is returned.

If a database is given in traditional form (by a user), namely by polynomial constraints (e.g., encoded by arithmetic boolean circuits), one may ask for more suitable alternative representations in order to improve the complexity of query evaluation. Such an improved representation may be a branching-parsimonious cell decomposition of the constraint relations, as described before. The generation of such a cell decomposition may be considered as a pre-processing of the data before actually storing it in the database.

*The example of finite databases.* Let us explain the idea behind this observation in the case of a finite database containing just one relation $D$. For this purpose we shall make free use of the main complexity results of [13, 15, 17, 21]. Let us suppose that the domain of $D$ is described by a conjunction of the form

$$f_1 = 0 \wedge \cdots \wedge f_s = 0 \wedge g \neq 0,$$

with $f_1, \ldots, f_s, g \in \mathbb{Q}[x_1, \ldots, x_n]$, and that this conjunction defines a finite set $V$ of points of $\mathbb{C}^n$ having a non-empty intersection with $\mathbb{R}^n$. Suppose furthermore that the conjunction $f_1 = 0 \wedge \cdots \wedge f_s = 0 \wedge g \neq 0$ is encoded by a division-free arithmetic boolean circuit of size at most $L$ that contains only decision gates for equality. Let $d$ be an upper bound for the degrees of the polynomials $f_1, \ldots, f_s$

and $\delta$ be the system degree associated to the conjunction $f_1 = 0 \wedge \cdots \wedge f_s = 0 \wedge g \neq 0$ (thus we have $\#V \leq \delta \leq d^n$). Under some very weak conditions on the data $f_1, \ldots, f_s$ and $g$ (e.g., we may require $s := n$ and that $f_1, \ldots, f_n$ form a reduced regular sequence outside the locus $\{g = 0\}$) one can find in sequential time $L(nd\delta)^{O(1)}$ a rational linear form $u = \lambda_1 x_1 + \cdots \lambda_n x_n$ with $\lambda_1 \neq 0$ and univariate polynomials $q, v_2, \ldots, v_n \in \mathbb{Q}[t]$ (where $t$ is a new variable) satisfying the following conditions:

(i) $\deg q \leq \#V$, $\deg v_2 \leq \#V, \ldots, \deg v_n \leq \#V$;
(ii) $q$ has no multiple zeroes;
(iii) $V$ is definable by the conjunction $q(u) = 0 \wedge x_2 = v_2(u) \wedge \cdots \wedge x_n = v_n(u)$.

One sees immediately that the polynomials $q(u), x_2 - v_2(u), \ldots, x_n - v_n(u)$ yield a description of $D = \mathbb{R}^n \cap V$ by a single cell in the sense of Definition 2. Moreover, for $v_1 := \frac{1}{\lambda_1}(t - \lambda_2 v_2 - \cdots - \lambda_n v_n)$ we obtain the following *parametric* description of the complex and real varieties $V$ and $D$ respectively. Namely, $V := \{(v_1(\tau), \ldots, v_n(\tau)) \mid q(\tau) = 0, \tau \in \mathbb{C}\}$ and $D := \{(v_1(\tau), \ldots, v_n(\tau)) \mid q(\tau) = 0, \tau \in \mathbb{R}\}$.

Evaluation of purely existential active domain $\mathsf{FO}(+, \times, <, 0, 1)$-queries over the database $D$ may now be reduced to greatest common divisor computations between univariate polynomials over $\mathbb{Q}$ (here we suppose that all polynomials occurring in the query are given by division-free arithmetic boolean circuits). It is possible to associate to each such query a degree, depending only on the occurrences of relation predicate addressing $D$ in the query.

Evaluation of such a query becomes now linear in the syntactic query size and polynomial in its degree.

*The example of rationally parameterized families of polynomial functions.* A particular instance of interest is the case that the semi-algebraic set $A$ is contained in $\mathbb{R}^{m+n+1}$ and represents a rational family of polynomial functions from $\mathbb{R}^n$ to $\mathbb{R}$. To be more precise, let $\pi : \mathbb{R}^{m+n+1} \to \mathbb{R}^m$ be the canonical projection of any point of $\mathbb{R}^{m+n+1}$ on its first $m$ coordinates. Suppose that $A$ is non-empty and that for any $u = (u_1, \ldots, u_m) \in \pi(A)$ the semi-algebraic set $(\{u\} \times \mathbb{R}^{n+1}) \cap A$ is the graph of an $n$-variate polynomial $f_u \in \mathbb{R}[x_1, \ldots, x_n]$. It is a natural extension of our previously introduced sample-point query to ask for a procedure which enables us for each $u \in \pi(A)$ and each $x \in \mathbb{R}^n$ to compute the value of $f_u(x)$. The output of such a procedure may be a purely existential prenex first-order formula in the free variables $u_1, \ldots, u_m$ and $x_1, \ldots, x_n$ which represents for each $u \in \pi(A)$ a division-free arithmetic circuit which evaluates the polynomial $f_u$ (observe that there exists a uniform degree bound for all these polynomials). One easily verifies that all our requirements on the semi-algebraic set $A$, except that of the polynomial character of the function represented by the graph $(\{u\} \times \mathbb{R}^{n+1}) \cap A$, are first-order definable over the reals. Nevertheless, over the complex numbers, when $A$ is a constructible (i.e., a first-order definable subset of $\mathbb{C}^{m+n+1}$), all these requirements are first-order expressible. This leads us to a new type of computable queries which return on input a semi-algebraic or constructible set $A$ as above and an element $u \in \pi(A)$, a first-order formula which represents a division-free arithmetic circuit evaluating the polynomial $f_u$. Uniformity of query evaluation with respect to $u$ is expressed by the requirement that the terms contained in this formula have to depend *rationally* on $u$.

In the following, we shall refer to this type of queries as *extended sample point queries*. We shall refer to $u_1, \ldots, u_m$ as the *parameters* and to $x_1, \ldots, x_n$ as the *variables* of the query.

Suppose now that $A$ is a constructible subset of $\mathbb{C}^{m+n+1}$ with irreducible Zariski-closure. Let $V$ be the Zariski-closure of $\pi(A)$. Then $V$ is an irreducible affine subvariety of $\mathbb{C}^m$. We denote the function field of $V$ by $K$. It is not difficult to see that for generically chosen parameter points $u \in \pi(A)$, the extended sample point query associated to $A$ can be realized by a greatest common divisor computation in the polynomial ring $K[x_1, \ldots, x_n]$.

*Variables versus parameters.* The previous example is motivated by spatial data that come from physical observation and are only known with uncertainty. Another motivation comes from parametric optimization. The optimization problems described in Section 3.1 can also be studied in parametric form, i.e., in the case where the linear inequalities and the target function contain coefficients that depend on a time parameter [5]. In this case, an optimum is not an arbitrary set of sample points but an analytic (or at least continuous) function which depends on a time parameter.

We are now going to explain why we distinguished between the parameters $u_1, \ldots, u_m$ and the variables $x_1, \ldots, x_n$ in our discussion of rationally parameterized families of polynomial functions. In the example above, let $\Phi(u_1, \ldots, u_m, x_1, \ldots, x_m, y)$ be a quantifier-free formula which defines the semi-algebraic or constructible set $A$. Let us suppose that $\Phi$ contains a subformula $\Psi(u_1, \ldots, u_m)$ which expresses an internal algebraic dependency between the parameters $u_1, \ldots, u_m$. With respect to the variables $x_1, \ldots, x_n$ there is no such subformula contained in $\Phi$. For the sake of simplicity, we shall suppose that $\Psi$ defines the set $\pi(A)$ and that there exists a formula $\Omega(u_1, \ldots, u_m, x_1, \ldots, x_n, y)$ such that $\Phi(u_1, \ldots, u_m, x_1, \ldots, x_m, y)$ can be written as

$$\Psi(u_1, \ldots, u_m) \wedge \Omega(u_1, \ldots, u_m, x_1, \ldots, x_m, y).$$

In Section 4, we shall meet natural examples of parameterized algebraic families of polynomial functions where the formula $\Psi$ becomes of uncontrolled size and is of few interest, whereas the formula $\Omega$ becomes the relevant part of the output information of a suitable elimination algorithm. This situation occurs for instance when the points $u$ of $\mathbb{R}^m$ satisfying the formula $\Psi$ are given in parametric form (i.e., when they are image points of some polynomial or rational map coming from some affine source space). In this case, we are only interested in the subformula $\Omega$, since points $u \in \mathbb{R}^m$ satisfying $\Psi$ can easily be produced in sufficient quantity. In subsequent queries, $x_1, \ldots, x_n$ may appear as bounded variables, whereas the parameters $u_1, \ldots, u_m$ are not supposed to be subject to quantification. The example of the $u_1, \ldots, u_m$ expressing uncertainty in physical spatial data illustrates this. These different rôles motivate us to distinguish between $u_1, \ldots, u_m$ and $x_1, \ldots, x_n$ and to call them parameters and variables, respectively.

*The branching-parsimonious algorithmic model.* In the model, that we are going to use in the sequel, parameters and variables receive a different treatment. (Free) variables may be specialized into arbitrary real (or complex) values, whereas the specialization of parameters may be subject to certain restrictions. In the above example of a rationally parameterized family of polynomial functions, the $n$-tuple of variables $(x_1, \ldots, x_n)$ may be specialized into any point of the affine space $\mathbb{R}^n$ (or $\mathbb{C}^n$), whereas the $m$-tuple of parameters $(u_1, \ldots, u_m)$ may only be specialized into points satisfying $\Psi(u_1, \ldots, u_m)$, i.e., into points belonging to $\pi(A)$. Once the $n$-tuple of variables $(x_1, \ldots, x_n)$ is specialized into a point of the corresponding affine space, this point cannot be modified anymore. However, we allow infinitesimal modifications of a given specialization of the $m$-tuple of parameters $(u_1, \ldots, u_m)$ within the domain of definition determined by the formula $\Psi$. In the branching-parsimonious model, we require that an arithmetic boolean circuit which represents the semi-algebraic or constructible set $A$ does not contain divisions which involve the variables $x_1, \ldots, x_n$. Similarly, for a given point $u \in \pi(A)$, we require that the arithmetic circuit representing the polynomial $f_u$ is division-free. However, divisions by algebraic expressions in the parameters $u_1, \ldots, u_m$ are sometimes unavoidable (e.g., in the case of parametric greatest common divisor computations; see [11, 14]). Therefore, we allow certain limited divisions by algebraic expressions which depend only on the parameters $u_1, \ldots, u_m$. More precisely, we allow that the arithmetic boolean circuits representing the set $A$ or the output of the corresponding extended sample point query computes certain, but not arbitrary, rational functions in the parameters $u_1, \ldots, u_m$, called *scalars* of the circuit. However, we do not allow the division of a positive-degree polynomial in the variables $x_1, \ldots, x_n$ by a non-constant scalar. In the above sense, we require for our branching-parsimonious algorithmic model that arithmetic boolean circuits are *essentially division-free with respect to variables* (see [11, 14] for a precise definition).

*Branching-free output representations of extended sample point queries.* Since we allow certain infinitesimal modifications of the parameters $u_1, \ldots, u_m$ within their domain of definition, we sometimes may replace divisions (and corresponding branchings) by limit processes in the spirit of L'Hôpital's rule. It is possible to mimic algebraically this kind of limit process by places (see [29] for the notion of place and [11, 14] for motivations of this idea).

Branchings corresponding to divisions can trivially be avoided by restricting input data. Therefore a meaningful notion of branching-parsimonious (or branching-free) algorithm requires the consideration of Zariski-closures of input data sets. This may partially explain the rather technical assumptions and tools in the following ad hoc definition of a branching-free representation of the output of an extended sample point query.

Suppose now that in the example above $A$ is a constructible subset of $\mathbb{C}^{m+n+1}$ with irreducible Zariski-closure $B$. Let $V$ be the Zariski-closure of $\pi(A)$ in $\mathbb{C}^m$.

Then $V$ is an irreducible affine variety whose function field we denote by $K$. Moreover, the irreducible affine variety $B$ is birationally equivalent to $V \times \mathbb{C}^n$. Suppose furthermore that $\pi(B) = V$ holds and that $B$ represents a rationally parameterized family of polynomial functions which extends the family represented by $A$. Then we say that the extended sample point query associated with $A$ admits a *branching-free output representation* if there exists an essentially division-free, single-output arithmetic circuit $\beta$ with inputs $x_1, \ldots, x_n$ and scalars $\theta_1, \ldots, \theta_s \in K$ satisfying the following conditions:

(i) for any point $u \in \pi(A)$ where the rational functions $\theta_1, \ldots, \theta_s$ are defined, the division-free arithmetic circuit, obtained from $\beta$ by specializing the scalars $\theta_1, \ldots, \theta_s$ into the complex values $\theta_1(u), \ldots, \theta_s(u)$, evaluates the polynomial $f_u$;

(ii) for any point $u \in V$ and any place $\varphi : K \to \mathbb{C} \cup \{\infty\}$ whose valuation ring extends the local ring of the affine variety $V$ at the point $u$, the values $\varphi(\theta_1), \ldots, \varphi(\theta_s)$ are finite and uniquely determined by $u$ (therefore we shall write $\theta_1(u) := \varphi(\theta_1), \ldots, \theta_s(u) := \varphi(\theta_s)$).

Let an arithmetic circuit $\beta$ be given as above. Then we call $\beta$ a branching-free representation of the output of the extended sample point query associated to $A$.

Observe that the output of the circuit $\beta$ represents a polynomial belonging to $K[x_1, \ldots, x_n]$ whose coefficients satisfy condition (ii). Moreover, the arithmetic circuit $\beta$ constitutes a division-free representation of the extended sample point query associated to the Zariski-closure $B$ of $A$. Finally, let us remark that for any $u \in V$, $x \in \mathbb{C}^n$ and $y \in \mathbb{C}$, the point $(u, x, y)$ belongs to $B$ if and only if the circuit $\beta_u$, obtained from $\beta$ by replacing the scalars $\theta_1, \ldots, \theta_s$ by the complex numbers $\theta_1(u), \ldots, \theta_s(u)$, computes on input $x$ the output $y$.

We require that a *branching-parsimonious query evaluation algorithm* produces a branching-free output representation of the given extended sample point query if the query admits such a representation.

Let us also observe that extended sample point queries appear in a natural way if we apply the constraint database concept to data processing in the context of approximation theory and functional analysis.

## 4 A lower complexity bound for extended sample point queries

In this section, we restrict our attention to constraint databases defined in the language $\mathsf{FO}(+, \times, 0, 1, =)$ over the complex numbers. We shall consider two ternary relational predicates, namely $S(v_1, v_2, w)$ and $P(v_1, v_2, w)$. Our query language will therefore be $\mathrm{FO}(+, \times, 0, 1, =, S, P)$. Let $L, n$ be given natural numbers and let $r := (L + n + 1)^2$. For any polynomial $f \in \mathbb{C}[x_1, \ldots, x_n]$, we denote by $L(f)$ the minimal non-scalar size of all division-free arithmetic circuits with inputs $x_1, \ldots, x_n$ and scalars from $\mathbb{C}$ which evaluate the polynomial $f$. Let

$$W_{L,n} := \{f \in \mathbb{C}[x_1, \ldots, x_n] \mid L(f) \leq L\}.$$

One sees easily that all polynomials contained in $W_{L,n}$ have degree at most $2^L$ and that $W_{L,n}$ forms a $\mathbb{Q}$-definable object class which has a $\mathbb{Q}$-definable holomorphic encoding by the continuous data structure $\mathbb{C}^r$. Observe that Zariski-closure $\overline{W_{L,n}}$ of $W_{L,n}$ is a $\mathbb{Q}$-definable, absolutely irreducible algebraic variety consisting of the polynomials of $\mathbb{C}[x_1,\ldots,x_n]$ which have approximate complexity at most $L$. Moreover, the affine variety $\overline{W_{L,n}}$ forms a cone in its ambient space (i.e., for any $\lambda \in \mathbb{C}$ we have $\lambda\overline{W_{L,n}} \subseteq \overline{W_{L,n}}$). For details on complexity and data structure models we refer to [10, 11].

Let $z_1,\ldots,z_r$ and $y$ be new variables. Choose now a directed acyclic graph $\mathcal{D}_{L,n}$ representing a generic, division-free arithmetic circuit with input nodes $x_1,\ldots,x_n$, output node $y$ and scalar nodes $z_1,\ldots,z_r$ such that any polynomial of $W_{L,n}$ may be evaluated by the division-free arithmetic circuit obtained from $\mathcal{D}_{L,n}$ by a suitable specialization of the parameters $z_1,\ldots,z_r$ into complex values. Without loss of generality, we may assume that the number of internal nodes of $\mathcal{D}_{L,n}$ is of order $O((L+n)^2)$. Translating the structure of the directed acyclic graph $\mathcal{D}_{L,n}$ into first-order logic one infers easily a formula $\Psi_{L,n}(S,P,z_1,\ldots,z_r,x_1,\ldots,x_n,y)$ in the free variables $z_1,\ldots,z_r,x_1,\ldots,x_n,y$ of the query language $\mathrm{FO}(+,\times,0,1,=,S,P)$ such that $\Psi_{L,n}(S,P)$ satisfies the following conditions.

(i) $\Psi_{L,n}(S,P)$ is prenex, purely existential and of length $O((L+n)^2)$;

(ii) interpreting the predicates $S$ and $P$ in $\Psi_{L,n}$ as the graphs of the addition and the multiplication of complex numbers, and specializing the variables $z_1,\ldots,z_r$ into the complex numbers $\zeta_1,\ldots,\zeta_r$, the formula $\Psi_{L,n}(S,P,\zeta_1,\ldots,\zeta_r,x_1,\ldots,x_n,y)$ describes the graph of the polynomial of $\mathbb{C}[x_1,\ldots,x_n]$ computed by the arithmetic circuit, obtained from $\mathcal{D}_{L,n}$ by specializing the scalars $z_1,\ldots,z_r$ into $\zeta_1,\ldots,\zeta_r$.

Let $m := 4(L+n)^2+2$. From [11], Corollary 3 (see also [14, Lemma 4]) we deduce that there exists an identification sequence $\gamma_1,\ldots,\gamma_m \in \mathbb{Q}^n$ for the object class $\overline{W_{L,n}}$. Let $\Delta_{L,n}(S,P)$ be a closed $\mathrm{FO}(+,\times,0,1,=,S,P)$-formula saying that $S$ and $P$ are the graphs of two binary operations which map $\mathbb{C}^2$ into $\mathbb{C}$, that $\gamma_1,\ldots,\gamma_n$ is an identification sequence for the object class of applications from $\mathbb{C}^n$ to $\mathbb{C}$, defined by the $\mathrm{FO}(+,\times,0,1,=,S,P)$-formula $\Psi_{L,n}(S,P)$ and that this object class is not empty. Without loss of generality, we may assume that $\Delta_{L,n}(S,P)$ has length $O((L+n)^2)$ and is prenex with a fixed number of quantifier alternations (which is independent of $L$ and $n$).

We consider now the $\mathrm{FO}(+,\times,0,1,=,S,P)$-formulas $\Phi_{L,n}(S,P,u_1,\ldots,u_r,x_1,\ldots,x_n,y)$ defined by

$$(\exists z_1)\cdots(\exists z_r)(\Psi_{L,n}(S,P,z_1,\ldots,z_r,x_1,\ldots,x_n,y)\wedge \bigwedge_{1\leq k\leq m} \Psi_{L,n}(S,P,z_1,\ldots,z_r,\gamma_k,u_k))\wedge \Delta_{L,n}(S,P)$$

and $\Omega_{L,n}(S, P, u_1, \ldots, u_r)$ defined by

$$(\exists z_1) \cdots (\exists z_r)( \bigwedge_{1 \le k \le m} \Psi_{L,n}(S, P, z_1, \ldots, z_r, \gamma_k, u_k)) \wedge \Delta_{L,n}(S, P).$$

Without loss of generality, we may assume that $\Phi_{L,n}(S, P)$ and $\Omega_{L,n}(S, P)$ are prenex formulas of length $O((L + n)^2)$ having a *fixed* number of quantifier alternations and containing the free variables $u_1, \ldots, u_m, x_1, \ldots, x_n, y$ and $u_1, \ldots, u_m$, respectively. Let $\pi : \mathbb{C}^{m+n+1} \to \mathbb{C}^m$ be the canonical projection which maps each point of $\mathbb{C}^{m+n+1}$ on its first $m$ coordinates and let $D$ be a constraint database over the schema $(S, P)$ over the complex numbers. Suppose that $D$ satisfies the formula $\Delta_{L,n}(S, P)$. With respect to the database $D$, the formula $\Phi_{L,n}(S, P, u_1, \ldots, u_r, x_1, \ldots, x_n, y)$ defines a non-empty constructible subset $A_{L,n}(D)$ of $\mathbb{C}^{m+n+1}$ and the formula $\Omega_{L,n}(S, P, u_1, \ldots, u_r)$ defines the set $\pi(A_{L,n}(D))$. Moreover, for any $u \in \pi(A_{L,n}(D))$, the formula $\Phi_{L,n}(S, P, u, x_1, \ldots, x_n, y)$ describes the graph of a $n$-variate polynomial map. Therefore, it makes sense to consider, for any natural numbers $n$ and $L$, the generalized sample point query associated to the formula $\Phi_{L,n}(S, P, u_1, \ldots, u_n, x_1, \ldots, x_n, y)$. Suppose now that there is given a *branching-parsimonious* procedure $\mathcal{P}$ which evaluates this family of extended sample point queries. We are now going to analyze the complexity behaviour of $\mathcal{P}$ for this query on the particular input database $D$, where $S$ and $P$ are interpreted as the graphs of the sum and the product of complex numbers.

We are now able to state and to prove the main complexity result of this paper.

**Theorem 1.** *Let notations and assumptions be as before. Then the branching-parsimonious procedure $\mathcal{P}$ requires sequential time $2^{\Omega(n)}$ in order to evaluate on input the database $D$ the extended sample point query associated to the size $O(n^2)$ first-order formula $\Phi_{n,n}(S, P)$. In particular, extended sample point queries associated to first-order formulas with a fixed number of quantifier alternations cannot be evaluated by branching-parsimonious procedures in polynomial time.*

*Proof.* The arguments we are now going to use follow the general lines of the proofs of [14, Theorem 5] and [11, Theorem 4].

For the moment let us fix the integer parameters $L$ and $n$.

Observe that the closed formula $\Delta_{L,n}(S, P)$ is valid on the database $D$. Therefore the constructible set $A_{L,n} := A_{L,n}(D)$ is nonempty.

Let $B_{L,n}$ and $V_{L,n}$ be the Zariski-closures of $A_{L,n}$ and $\pi(A_{L,n})$ in $\mathbb{C}^{m+n+1}$ and $\mathbb{C}^m$, respectively.

Let $\lambda_{L,n} := \overline{W}_{L,n} \times \mathbb{C}^n \to \mathbb{C}^{m+n+1}$ and $\mu_{L,n} := \overline{W}_{L,n} \to \mathbb{C}^m$ be the morphisms of $\mathbb{Q}$-definable affine varieties defined for $f \in \overline{W}_{L,n}$ and $x \in \mathbb{C}^n$ by $\lambda_{L,n}(f, x) := (f(\gamma_1), \ldots, f(\gamma_m), x, f(x))$ and $\mu_{L,n}(f) := (f(\gamma_1), \ldots, f(\gamma_m))$. From the syntactic form of $\Phi_{L,n}(S, P)$ and $\Omega_{L,n}(S, P)$ one infers immediately that

$$\lambda_{L,n}(W_{L,n} \times \mathbb{C}^n) = A_{L,n}$$

and

$$\mu_{L,n}(W_{L,n}) = \pi(A_{L,n})$$

holds.

Therefore $B_{L,n}$ and $V_{L,n}$ are $\mathbb{Q}$-definable absolutely irreducible affine varieties and we may consider $\lambda_{L,n}$ and $\mu_{L,n}$ as dominant morphism mapping $\overline{W}_{L,n} \times \mathbb{C}^n$ into $B_{L,n}$ and $\overline{W}_{L,n}$ into $V_{L,n}$. Observe that $\overline{W}_{L,n}$ and $V_{L,n}$ form closed cones in their respective ambient spaces. Since $\gamma_1, \ldots, \gamma_m$ were chosen as an identification sequence for the object class $\overline{W}_{L,n}$, we may conclude that $\mu_{L,n} : \overline{W}_{L,n} \to V_{L,n}$ is an injective dominant morphism of closed affine cones, which is homogeneous of degree one.

Therefore $\mu_{L,n}$ is a finite, bijective and birational morphism of affine varieties (see, e.g., [35, I.5.3 Theorem 8 and proof of Theorem 7], [14, Lemma 4] or [11, Lemma 5]).

Let $\widetilde{\pi} : \mathbb{C}^{m+n+1} \to \mathbb{C}^{m+n}$ be the canonical projection which maps each point of $\mathbb{C}^{m+n+1}$ on its first $m+n$ coordinates. Then $\widetilde{\pi} \circ \lambda_{L,n} : \overline{W}_{L,n} \times \mathbb{C}^n \to V_{L,n} \times \mathbb{C}^n$ is a finite bijective and birational morphism of affine varieties and therefore $\lambda_{L,n} : \overline{W}_{L,n} \times \mathbb{C}^n \to B_{L,n}$ has the same property. This implies $\pi(B_{L,n}) = V_{L,n}$ and that $B_{L,n}$ represents a rationally parameterized family of polynomial functions which extends the family represented by $A_{L,n}$.

Let $K_{L,n}$ the function field over $\mathbb{C}$ of the absolutely irreducible variety $V_{L,n}$ and let $R_{L,n}$ be the $\mathbb{C}$-algebra of all rational functions $\theta$ of $K_{L,n}$ such that for any point $u \in V_{L,n}$ and any place $\varphi : K_{L,n} \to \mathbb{C} \cup \{\infty\}$ whose valuation ring extends the local ring of the affine variety $V_{L,n}$ at the point $u$, the value $\varphi(\theta)$ is finite and uniquely determined by $u$. Thus, for $u \in V_{L,n}$ and $\varphi : K_{L,n} \to \mathbb{C} \cup \{\infty\}$ as above, we may associate to any polynomial $f := \sum a_{i_1 \ldots i_n} x_1^{i_1} \ldots x_n^{i_n} \in R_{L,n}[x_1, \ldots, x_n]$ the polynomial $f(u, x_1, \ldots x_n) := \sum a_{i_1 \ldots i_n}(u) x_1^{i_1} \ldots x_n^{i_n} := \sum \varphi(a_{i_1 \ldots i_n}) x_1^{i_1} \ldots x_n^{i_n}$, which belongs to $\mathbb{C}[x_1, \ldots, x_n]$.

Since $\mu_{L,n} : \overline{W}_{L,n} \to V_{L,n}$ is a finite, bijective and birational morphism of affine varieties, we conclude that $R_{L,n}$ contains the coordinate ring of the affine variety $\overline{W}_{L,n}$ (see e.g. [29]). This implies that there exists a polynomial $f_{L,n} \in R_{L,n}[x_1, \ldots, x_n] \subset K[x_1, \ldots x_n]$ with the following property: for any $u \in V_{L,n}$, $x \in \mathbb{C}^n$ and $y \in \mathbb{C}$, the point $(u, x, y)$ belongs to $B_{L,n}$ if and only if $f_{L,n}(u, x) = y$ holds.

A branching-free output representation of the extended sample point query associated to the constructible set $A_{L,n}$ can now easily be realized by any arithmetic circuit which first computes all monomial terms of the polynomial $f_{L,n}$ and finally sums them up.

Therefore, the given branching-parsimonious query evaluation procedure $\mathcal{P}$ produces on input consisting of the database $D$ and the formula $\Phi_{L,n}(S, P)$ a branching-free representation of the extended sample point query associated to $A_{L,n}$. This branching-free representation is realized by an essentially division-free single-output arithmetic circuit $\beta_{L,n}$ with inputs $x_1, \ldots, x_n$ and scalars $\theta_1^{(L,n)}, \ldots, \theta_{s_{L,n}}^{(L,n)}$ belonging to $R_{L,n}$ such that $\beta_{L,n}$ computes at its output the polynomial $f_{L,n} \in R_{L,n}[x_1, \ldots, x_n]$.

Let $t$ and $\ell_1, \ldots, \ell_n$ be new variables and let us now consider the polynomial $g_n := t \prod_{1 \le i \le n} (\ell_i + x_i)$ defining the constructible object class

$$\Gamma_n := \{\tau \prod_{1 \le i \le n} (\lambda_i + x_i) \ \mid \ \tau, \lambda_1, \ldots, \lambda_n \in \mathbb{C}\}$$

of $n$-variate complex polynomial functions.

Observe that each element of $\Gamma_n$ has nonscalar sequential time complexity at most $n$.

Therefore the Zariski-closure $\overline{\Gamma}_n$ of the object class $\Gamma_n$ is contained in $\overline{W}_{n,n}$.

Observe that $\overline{\Gamma}_n$ is an absolutely irreducible, $\mathbb{Q}$-definable affine variety. Since $\mu_{n,n} : \overline{W}_{n,n} \to V_{n,n}$ is a finite morphism of irreducible affine varieties, we conclude that $C_n := \mu_{n,n}(\overline{\Gamma}_n)$ is an absolutely irreducible, $\mathbb{Q}$-definable closed affine subvariety of $V_{n,n}$.

Let $E_n$ and $L_n$ be the coordinate ring and the rational function field over $\mathbb{C}$ of the absolutely irreducible affine variety $C_n$.

Observe that we may identify $E_n$ with $\mathbb{C}[g_n(t, \ell_1, \ldots, \ell_n, \gamma_1), \ldots, g_n(t, \ell_1, \ldots, \ell_n, \gamma_m)]$ and $L_n$ with $\mathbb{C}(g_n(t, \ell_1, \ldots, \ell_n, \gamma_1), \ldots, g_n(t, \ell_1, \ldots, \ell_n, \gamma_m))$. Therefore we may consider $E_n$ as a $\mathbb{C}$-subdomain of the polynomial ring $\mathbb{C}[t, \ell_1, \ldots, \ell_n]$ and $L_n$ as a $\mathbb{C}$-subfield of $\mathbb{C}(t, \ell_1, \ldots, \ell_n)$.

The rational functions $\theta_1^{(n,n)}, \ldots, \theta_{s_{n,n}}^{(n,n)}$ of the affine variety $V_{n,n}$ may be not defined on the subvariety $C_n$. Nevertheless, since they belong to the $\mathbb{C}$-algebra $R_{n,n}$, one verifies easily that there exist rational functions $\sigma_1^{(n)}, \ldots, \sigma_{s_{n,n}}^{(n)}$ of the affine variety $C_n$ satisfying the following condition: for any point $u \in C_n$ and any place $\psi : L_n \to \mathbb{C} \cup \{\infty\}$ whose evaluation ring extends the local ring of $C_n$ at the point $u$, the values of $\psi$ at $\sigma_1^{(n)}, \ldots, \sigma_{s_{n,n}}^{(n)}$ are given by $\psi(\sigma_1^{(n)}) = \theta_1^{(n,n)}(u), \ldots, \psi(\sigma_{s_{n,n}}^{(n)}) = \theta_{s_{n,n}}^{(n,n)}(u)$ and therefore finite and uniquely determined by $u$.

In particular, the rational functions $\sigma_1^{(n)}, \ldots, \sigma_{s_{n,n}}^{(n)} \in L_n$ are integral over the $\mathbb{C}$-algebra $E_n$ and hence contained in the polynomial ring $\mathbb{C}[t, \ell_1, \ldots, \ell_n]$ (see, e.g., [29]). Therefore we may consider $\sigma_1^{(n)}, \ldots, \sigma_{s_{n,n}}^{(n)}$ as polynomials in the variables $t$ and $\ell_1, \ldots, \ell_n$ (i.e. as elements of $\mathbb{C}[t, \ell_1, \ldots, \ell_n]$).

¿From $g_n = t \prod_{1 \le i \le n} (\ell_i + x_i)$, we infer the identities

$$g_n(0, \ell_1, \ldots, \ell_n, \gamma_1) = 0, \ldots, g_n(0, \ell_1, \ldots, \ell_n, \gamma_m) = 0.$$

Since the polynomials $\sigma_1^{(n)}, \ldots, \sigma_{s_{n,n}}^{(n)}$ depend integrally from $g_n(t, \ell_1, \ldots, \ell_n, \gamma_m), \ldots, g_n(t, \ell_1, \ldots, \ell_n, \gamma_m)$, we deduce now easily that the polynomials $\sigma_1^{(n)}(0, \ell_1, \ldots, \ell_n), \ldots, \sigma_{s_{n,n}}^{(n)}(0, \ell_1, \ldots, \ell_n)$ do not depend on the variables $\ell_1, \ldots, \ell_n$, i.e., they belong to $\mathbb{C}$.

Let now $\widetilde{\beta}_n$ be the division-free arithmetic circuit with scalars in $\mathbb{C}[t, \ell_1, \ldots, \ell_n]$ obtained by replacing in the circuit $\beta_{n,n}$ the scalars $\theta_1^{(n,n)}, \ldots, \theta_{s_{n,n}}^{(n,n)}$ by the polynomials $\sigma_1^{(n)}, \ldots, \sigma_{s_{n,n}}^{(n)}$.

One verifies easily that the circuit $\widetilde{\beta}_n$ computes the polynomial

$$g_n = t \prod_{1 \le i \le n} (\ell_i + x_i) = \sum_{\substack{\delta_1,\ldots,\delta_n,\varepsilon_1,\ldots,\varepsilon_n \in \{0,1\} \\ \delta_1+\varepsilon_1=1,\ldots,\delta_n+\varepsilon_n=1}} t\ell_1^{\delta_1} \ldots \ell_n^{\delta_n} x_1^{\varepsilon_1} \ldots x_n^{\varepsilon_n}.$$

Let $v_1, \ldots, v_{s_{n,n}}$ be new variables. From the directed acyclic graph structure of $\widetilde{\beta}_n$ (or $\beta_{n,n}$) one deduces immediately that for each $(\delta_1, \ldots, \delta_n) \in \{0,1\}^n$ there exists a polynomial $Q^{(n)}_{(\delta_1,\ldots,\delta_n)} \in \mathbb{Q}[v_1, \ldots, v_{s_{n,n}}]$ satisfying the condition $Q^{(n)}_{(\delta_1,\ldots,\delta_n)}(\sigma_1^{(n)}, \ldots, \sigma_{s_{n,n}}^{(n)}) = t\ell_1^{\delta_1} \ldots \ell_n^{\delta_n}$. Let $Q_n : \mathbb{C}^{s_{n,n}} \to \mathbb{C}^{2^n}$ the polynomial map defined by $Q_n := (Q^{(n)}_{(\delta_1,\ldots,\delta_n)}; (\delta_1, \ldots, \delta_n) \in \{0,1\}^n)$.

Consider now an arbitrary integer $1 \le \rho \le 2^n$ and let $\lambda_{\rho,1} := \rho^{2^0}, \ldots, \lambda_{\rho,n} := \rho^{2^{n-1}}$, $\lambda_\rho := (\lambda_{\rho,1}, \ldots, \lambda_{\rho,n})$ and $\alpha_\rho^{(n)} : \mathbb{C} \to \mathbb{C}^{s_{n,n}}$ and $\beta_\rho^{(n)} : \mathbb{C} \to \mathbb{C}^{2^n}$ be the parameterized algebraic curves defined for $\tau \in \mathbb{C}$ by

$$\alpha_\rho^{(n)}(\tau) := (\sigma_1^{(n)}(\tau, \lambda_\rho), \ldots, \sigma_{s_{n,n}}^{(n)}(\tau, \lambda_\rho))$$

and

$$\beta_\rho^{(n)}(\tau) := (\tau\lambda_{\rho,1}^{\delta_1} \ldots \lambda_{\rho,n}^{\delta_n};\ (\delta_1, \ldots, \delta_n) \in \{0,1\}^n) = (\tau\rho^j;\ 0 \le j < 2^n).$$

Observe that the functional identity

$$\beta_\rho^{(n)} = Q_n \circ \alpha_\rho^{(n)} \tag{1}$$

is valid.

Since the polynomials $\sigma_1^{(n)}(0, \ell_1, \ldots, \ell_n), \ldots, \sigma_{s_{n,n}}^{(n)}(0, \ell_1, \ldots, \ell_n)$ do not depend on the variables $\ell_1, \ldots, \ell_n$, there exists a point $a_n \in \mathbb{C}^{s_{n,n}}$, independent on $\rho$, such that $\alpha_\rho^{(n)}(0) = a_n$ holds.

Let us denote the derivatives of $\alpha_\rho^{(n)}$, $\beta_\rho^{(n)}$ and $Q_n$ by $\frac{d\alpha_\rho^{(n)}}{d\tau}$, $\frac{d\beta_\rho^{(n)}}{d\tau}$ and $DQ_n$. Furthermore let $\omega_\rho^{(n)} := \frac{d\alpha_\rho^{(n)}}{d\tau}(0) \in \mathbb{C}^{s_{n,n}}$, let $\eta_n : \mathbb{C}^{s_{n,n}} \to \mathbb{C}^{2^n}$ be the $\mathbb{C}$-linear map defined by $\eta_n := (DQ_n)(a_n)$ and observe that $\frac{d\beta_\rho^{(n)}}{d\tau}(0) = (\rho^j;\ 0 \le j < 2^n)$ holds. Applying the chain rule to (1), we infer the following identities:

$$\begin{aligned}(\rho^j;\ 0 \le j < 2^n) &= \frac{d\beta_\rho^{(n)}}{d\tau}(0) = (DQ_n)(\alpha_\rho^{(n)}(0)(\frac{d\alpha_\rho^{(n)}}{d\tau}(0)) \\ &= (DQ_n)(a_n)(\omega_\rho^{(n)}) = \eta_n(\omega_\rho^{(n)}).\end{aligned}$$

Since $(\rho^j)_{1 \le \rho \le 2^n, 0 \le j < 2^n}$ is a nonsingular Vandermonde matrix, we conclude now that the image of the $\mathbb{C}$-linear map $\eta_n : \mathbb{C}^{s_{n,n}} \to \mathbb{C}^{2^n}$ contains $2^n$ linear independent points. Therefore $\eta_n$ is surjective. This implies $s_{n,n} \ge 2^n$.

Therefore the arithmetic circuit $\beta_{n,n}$, which represents the output produced by the procedure $\mathcal{P}$ on input consisting of the database $D$ and the formula $\Phi_{n,n}$, contains at least $2^n$ scalars.

This implies that the nonscalar size of the circuit $\beta_{n,n}$ is at least $2^{\frac{n}{2}} - n - 1$. In conclusion, the procedure $\mathcal{P}$ requires $2^{\Omega(n)}$ sequential time in order to produce the output $\beta_{n,n}$ on input consisting of the data base $D$ and the size $O(n^2)$ formula $\Phi_{n,n}$. $\square$

The main outcome of Theorem 1 and its proof can be paraphrased as follows: *constraint database theory applied to quite natural computation tasks, as, e.g., branching-parsimonious interpolation of low complexity polynomials, leads necessarily to non-polynomial sequential time lower bounds.*

In view of the $P_{\mathbb{R}} \neq NP_{\mathbb{R}}$ conjecture in the algorithmic model of Blum–Shub–Smale over the real and complex numbers, it seems unlikely that this worst case complexity behavior can be improved substantially if we drop some or all of our previously introduced requirements on queries and their output representations. Nevertheless we wish to stress that these requirements constitute a fundamental technical ingredient for the argumentation in the proof of Theorem 1.

## 5 Conclusion and future research on the complexity of query evaluation

In this paper, we have emphasized the importance of *data structures* and their effect on the complexity of quantifier elimination. We have also proposed a novel data model for constraint databases, consisting of *smooth cells accompanied by sample points*, as produced by the known most efficient elimination procedures.

However, the intrinsic inefficiency of quantifier-elimination procedures represents a bottle-neck for real-world implementations of constraint database systems. As we have argued, it is unlikely that constraint database systems that are based on general purpose quantifier-elimination algorithms will ever become efficient. Also, restriction to work with linear data, as in most existing constraint database systems [28, Part IV], will also not lead to more efficiency. A promising direction is the study of a concept like the *system degree*, that has shown to be a fruitful notion for the complexity analysis of quantifier elimination in elementary geometry. In the context of query evaluation in constraint databases, the notion of system degree is still unsatisfactory since it is determined both by the query formula and the quantifier-free formulas describing the input database relations. It is a task for future constraint database research to develop a well-adapted complexity invariant in the spirit of the system degree in elimination theory.

Another direction of research is the study of query evaluation for first-order languages that capture certain genericity classes. For example, the first-order logic $\mathsf{FO}(\mathsf{between})$ has point variables rather than being based on real numbers and it captures the fragment of first-order logic over the reals that expresses queries that are invariant under affine transformations of the ambient space [19]. Although a more efficient complexity of query evaluation in this language cannot be expected, it is interesting to know whether languages such as $\mathsf{FO}(\mathsf{between})$ have quantifier elimination themselves (after an augmentation with suitable predicates).

## Acknowledgments

The first author wishes to thank to Alejandro Vaisman, Pablo Solernó, Universidad de Buenos Aires, and especially to Lidia Quintero for many stimulating discussions supporting his start in the field of database theory.

The second author wishes to thank Joos Heintz and the members of his research group for their hospitality at the University of Buenos Aires during his visit supported by the project "Efficient Query Evaluation for Constraint Database Systems" of the FWO-Vlaanderen and SECyT, Argentina.

## References

1. M.F. Atiyah and I.G. Macdonald. *Introduction to commutative algebra*. Addison-Wesley, 1969.
2. B. Bank, M. Giusti, J. Heintz, and G. M. Mbakop. Polar varieties, real equation solving and data structures: The hypersurface case. *Journal of Complexity*, 13:5–27, 1997. Best Paper Award Journal of Complexity 1997.
3. B. Bank, M. Giusti, J. Heintz, and G.M. Mbakop. Polar varieties and efficient real elimination. *Mathematische Zeitschrift*, 238:115–144, 2001.
4. B. Bank, M. Giusti, J. Heintz, and L.M. Pardo. Generalized polar varieties and an efficient real elimination procedure. *Submitted to Kibernetika*, 2003.
5. B. Bank, J. Guddat, D. Klatte, B. Kummer, and K. Tammer. *Non-Linear Parametric Optimization*. Birkhauser Verlag, Basel, 1983.
6. S. Basu, Pollack R., and M.-F. Roy. On the combinatorial and algebraic complexity of quantifier elimination. *Journal of the ACM*, 43:1002–1045, 1996.
7. S. Berkowitz. On computing the determinant in small parallel time using a small number of processors. *Information Processing Letters*, 18:147–150, 1984.
8. L. Blum, F. Cucker, M. Shub, and S. Smale. *Complexity and real computation*. Springer-Verlag, New York, 1998.
9. J. Bochnak, M. Coste, and M.-F. Roy. *Real Algebraic Geometry*, volume 36 of *Ergebenisse der Mathematik und ihrer Grenzgebiete. Folge 3.* Springer-Verlag, 1998.
10. P. Bürgisser, M. Clausen, and M. A. Shokrollahi. *Algebraic Complexity Theory*. Springer-Verlag, 1997.
11. D. Castro, M. Giusti, J. Heintz, G. Matera, and L. M. Pardo. The hardness of polynomial solving. *Foundations of Computational Mathematics*, 3:347–420, 2003.
12. G.E. Collins. Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In *Automata Theory and Formal Languages*, volume 33 of *Lecture Notes in Computer Science*, pages 134–183. Springer-Verlag, 1975.
13. M. Giusti, K. Hägele, J. Heintz, J. E. Morais, J. L. Montaña, and L. M. Pardo. Lower bounds for diophantine approximation. *Journal of Pure and Applied Algebra*, 117&118:277–317, 1997.
14. M. Giusti and J. Heintz. Kronecker's smart, little black boxes. In R. DeVore, Iserles A., and E. Suli, editors, *Foundations of Computational Mathematics*, pages 69–104, Cambridge, 2001. Cambridge University Press.
15. M. Giusti, J. Heintz, J. E. Morais, J. Morgenstern, and L. M. Pardo. Straight–line programs in geometric elimination theory. *Journal of Pure and Applied Algebra*, 124:101–146, 1998.

16. M. Giusti, J. Heintz, J. E. Morais, and L. M. Pardo. Le rôle des structures des données dans les problèmes d'élimination. *Comptes Rendues de l'Académie des Sciences*, 325(Série I):1223–1228, 1997.
17. M. Giusti, G. Lecerf, and B. Salvy. A Gröbner free alternative for polynomial system solving. *Journal of Complexity*, 17(1):154–211, 2001.
18. D. Grigor'ev and N. N. (jr.) Vorobjov. Solving systems of polynomial inequalities in subexponential time. *Journal of Symbolic Computation*, 5:37–64, 1988.
19. M. Gyssens, J. Van den Bussche, and D. Van Gucht. Complete geometrical query languages. *Journal of Computer and System Sciences*, 58(3):483–511, 1999. A preliminary report appeared in the *Proceedings 16th ACM Symposium on Principles of Database Systems (PODS'97)*.
20. J. Heintz, T. Krick, S. Puddu, J. Sabia, and A. Waissbein. Deformation techniques for efficient polynomial equation solving. *Journal of Complexity*, 16:70–109, 2000.
21. J. Heintz, G. Matera, and A. Weissbein. On the time–space complexity of geometric elimination procedures. *Applicable Algebra in Engineering, Communication and Computing*, 11(4):239–296, 2001.
22. J. Heintz and J. Morgenstern. On the intrinsic complexity of elimination theory. *Journal of Complexity*, 9:471–498, 1993.
23. J. Heintz, M.-F. Roy, and P. Solernó. Sur la complexité du principe de Tarski-Seidenberg. *Bulletin de la Société Mathématique de France*, 118:101–126, 1990.
24. G. Jeronimo, T. Krick, J. Sabia, and M. Sombra. The computational complexity of the Chow form. *To appear in Foundations of Computational Mathematics.* Avalilable at arXiv: math.AG/021009v1.
25. G. Jeronimo and J. Sabia. On the number of sets definable by polynomials. *Journal Algebra*, 227(2):633–644, 2000.
26. P.C. Kanellakis, G.M. Kuper, and P.Z. Revesz. Constraint query languages. *Journal of Computer and System Science*, 51(1):26–52, 1995. A preliminary report appeared in the *Proceedings 9th ACM Symposium on Principles of Database Systems (PODS'90)*.
27. T. Krick, L.M. Pardo, and M. Sombra. Sharp estimates for the arithmetic Nullstellensatz. *Duke Mathematical Journal*, 109(3):521–598, 2001.
28. G.M. Kuper, J. Paredaens, and L. Libkin. *Constraint databases*. Springer-Verlag, 1999.
29. S. Lang. *Algebra*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1969.
30. G. Lecerf. Quadratic Newton iterations for systems with multiplicity. *Foundations of Computational Mathematics*, 2:247–293, 2002.
31. J. Paredaens, J. Van den Bussche, and D. Van Gucht. Towards a theory of spatial database queries. In *Proceedings of the 13th ACM Symposium on Principles of Database Systems (PODS'94)*, pages 279–288, New York, 1994. ACM Press.
32. J. Renegar. On the computational complexity and geometry of the first-order theory of the reals I, II, III. *Jornal of Symbolic Computation*, 13:255–352, 1992.
33. P. Revesz. *Introduction to Constraint Databases*. Springer-Verlag, 2002.
34. E. Schost. Computing parametric geometric resolutions. *Applicable Algebra in Engineering, Communication and Computing*, 13(5):349–393, 2003.
35. I. R. Shavarevich. *Basic Algebraic Geometry : Varieties in Projective Space*. Springer-Verlag, 1994.
36. J. von zur Gathen. Parallel linear algebra. In J. Reif, editor, *Synthesis of parallel algorithmns*, pages 573–617. Morgan-Kaufmann, 1993.