On the Orthographic Dimension of Constraint Databases*

Stéphane Grumbach[1], Philippe Rigaux[2], and Luc Segoufin[1]

[1] INRIA, Rocquencourt BP 105, F-78153 Le Chesnay, France — {Stephane.Grumbach,Luc.Segoufin}@inria.fr
[2] Cedric/CNAM, 292 rue St Martin, F-75141 Paris Cedex 03, France — rigaux@cnam.fr

Abstract. One of the most important advantages of constraint databases is their ability to represent and to manipulate data in arbitrary dimension within a uniform framework. Although the complexity of querying such databases by standard means such as first-order queries has been shown to be tractable for reasonable constraints (e.g. polynomial), it depends badly (roughly speaking exponentially) upon the dimension of the data. A precise analysis of the trade-off between the dimension of the input data and the complexity of the queries reveals that the complexity strongly depends upon the use the input makes of its dimensions. We introduce the concept of orthographic dimension, which, for a convex object O, corresponds to the dimension of the (component) objects $O_1, ..., O_n$, such that $O = O_1 \times \cdots \times O_n$. We study properties of databases with bounded orthographic dimension in a general setting of o-minimal structures, and provide a syntactic characterization of first-order orthographic dimension preserving queries.
The main result of the paper concerns linear constraint databases. We prove that orthographic dimension preserving Boolean combination of conjunctive queries can be evaluated independently of the global dimension, with operators limited to the orthographic dimension, in parallel on the components. This results in an extremely efficient optimization mechanism, very easy to use in practical applications.

1 Introduction

The recent field of constraint databases, initiated at the beginning of the decade [KKR90], has lead to sound data models and query languages for multi-dimensional data [PVV94,GST94,KG94,KPV95,GK97]. It allows to represent infinite relations of arbitrary dimension by quantifier-free formulae over some arithmetical domain, and to manipulate these relations in a symbolic way. There have been many theoretical studies on constraint databases, mostly focused on the

* Work supported in part by the ESPRIT TMR project Chorochronos and in part by the french CNRS GDR CASSINI. First author partly supported by IASI CNR in Rome.

data model and on fundamental issues (expressive power and complexity) pertaining to the associated query languages. More recently, prototypes have started to emerge, showing the practical relevance of the constraint paradigm. Target applications are mainly spatial and temporal databases, where the expected improvement over traditional approaches follows from the formal framework which provides sound foundations for data modeling and query language design.

An important feature of constraint databases is their ability to handle in a uniform way pointsets in arbitrary dimension. The model is thus a natural candidate for applications which manipulate 3d objects (CAD), spatio-temporal data, or more generally for any scientific domain handling high dimensional pointsets. Although the complexity of querying such databases by standard means such as first-order queries has been shown to be tractable for reasonable constraints (e.g. polynomial constraints), it depends badly (roughly speaking exponentially) upon the dimension of the data. This seems to put a severe restriction on the ability of the forthcoming constraint database systems to handle efficiently high-dimensional relations.

In this paper we study techniques to overcome this problem which are based on restrictions of the geometry of the spatial objects allowed. They are expressible as conditions on the formulae representing the pointsets. We investigate how the query evaluation process can take advantage of these restrictions by manipulating the objects through their projections on lower dimensional subspaces, with a complexity which depends only linearly upon the dimension.

We first consider relations with *loose orthographic dimension* ℓ, as relations containing objects which are equal to the *join* of their projections on subspaces of dimension ℓ. They can be expressed by formulae over a first-order language with constraints, such that each atomic subformula involves at most ℓ variables. Unfortunately, this class is not closed under fundamental operations such as projection.

We thus investigate a more drastic restriction, the (strict) *orthographic dimension*, orthodim, of the data. A relation of dim d has orthodim bounded by ℓ if it can be represented by a formula with d variables, such that there is a partition of the set of variables into (say k) *components* of at most ℓ variables, such that each constraint in the formula can only involve variables from a single component. It is easy to see that a relation of orthodim bounded by ℓ is a finite collection of objects O, such that $O = \pi_{C_1} O \times \cdots \times \pi_{C_k} O$, where $\pi_{C_i} O$ is the projection of O on the ith component.

We consider the problem of deciding if a constraint relation has a given orthodim, or more generally, if it can be rotated to satisfy some given orthodim. We study the decidability of these problems in terms of the context structure, and prove that both problems are tractable for linear constraints.

We then consider first-order queries over databases of bounded orthodim. We first show that although it is undecidable in general if a query preserves the orthodim of its input, the class of preserving queries can be syntactically characterized. The characterization is rather ad hoc. We were able to define a natural class of *safe* queries that preserve the orthodim.

Finally, we focus on linear constraint databases. We consider the symbolic algebra, manipulating the generalized relations, and simulating operations of classical relational algebra over infinite relations. A detailed study of the complexity of the operations reveal that the evaluation of first-order queries depends exponentially upon the global dimension of the input data.

The main result of the paper shows that for safe Boolean Combination of Conjunctive Queries (BCCQ), the complexity of the evaluation depends exponentially upon the orthodim, and only linearly upon the global dimension.

The proof of this result is far from trivial. First note that among the operations of relational algebra, selection is the only one that does not preserve the orthodim since it might introduce a dependency between variables of different components. All other operations preserve the orthodim. We define ALG^{ℓ} as the set of queries which can be expressed with operations restricted to inputs of dimension at most the orthodim, including an approximated selection, which preserves the orthodim. We show that queries in ALG^{ℓ} capture the full expressive power of safe BCCQ.

The technique presented in this paper has been implemented in the DEDALE system [GRS98b] which can handle objects of orthodim 2 of any dimension d.

The paper is organized as follows. Section 2 introduces and studies the notion of orthographic dimension, Section 3 introduces a class of queries preserving the orthographic dimension of their inputs, and Section 4 focuses on the linear case.

2 Orthographic dimension

We consider databases in the context of well-behaved infinite structures. We assume some first-order language $\mathcal{L}$, consisting of two interpreted *predicate* symbols, equality and order, and interpreted *function* and *constant* symbols. In the sequel, we consider an arbitrary $\mathcal{L}$-structure $\mathcal{A}$ with universe A. We will also make the assumptions that the structure $\mathcal{A}$ is *o-minimal* [VMM94] and admits *quantifier elimination*. A structure $\mathcal{A}$ is *o-minimal* if every definable set, $\{x|\varphi(x)\}$, with φ a first-order formula in $\mathcal{L}$, is a finite union of isolated points and open intervals. $\mathcal{A}$ admits *quantifier elimination* if for every first-order formula $\varphi(\bar{x})$, there exists an equivalent *quantifier free* formula $\psi(\bar{x})$ (i.e. $\mathcal{A} \models \forall \bar{x} \varphi(\bar{x}) \leftrightarrow \psi(\bar{x})$). Examples of structures of interest in the present context are: the rationals with addition, $(\mathbb{Q}, \leqslant, +, 0, 1)$, and the real polynomial arithmetic, $(\mathbb{R}, \leqslant, +, \times, 0, 1)$. Both structures are o-minimal, admit quantifier elimination, and have decidable first-order theories.

A *(database) schema* s is a finite set of relation symbols such that $s \cap \mathcal{L} = \emptyset$. We always assume that the schema is disjoint from the first-order language, and we distinguish between logical predicates (such as $=, \leqslant$) in $\mathcal{L}$, and relations in s.

We next define the finitely representable databases in the context of some $\mathcal{L}$-structure $\mathcal{A}$. Kanellakis, Kuper and Revesz [KKR95] introduced the concept of a d-ary *generalized tuple*, which is a conjunction of atomic formulas in $\mathcal{L}$ with d variables. For instance in the context of the real numbers, the expression

$(x^2 + y^2 = 1) \wedge (x \leqslant 0)$ is a binary generalized tuple representing a half circle in the real plane. A d-ary generalized relation is defined by a finite set of d-ary generalized tuples. In this framework, a tuple $[a, b]$ of the classical relational model [Ull88,Mai83] is an abbreviation of the formula $(x = a \wedge y = b)$ involving only the equality symbol and constants.

Definition 1 Let $\varphi(\bar{x})$ be a formula in $\mathcal{L}$ with d distinct variables $x_1, \ldots, x_d$. A d-ary relation $S \subseteq A^d$ is *represented by* φ over the $\mathcal{L}$-structure $\mathcal{A}$ if

$$\forall \bar{a} \in A, \mathcal{A} \models \varphi(\bar{a}) \quad \text{iff} \quad \bar{a} \in S$$

S is a finitely representable relation represented by φ over $\mathcal{A}$, and φ is a *finite representation of* S *over* $\mathcal{A}$. The attributes of S are denoted by the corresponding variables of φ.

Consider a d-ary relation R represented by a quantifier free formula in disjunctive normal form φ, of the form:

$$\varphi \equiv \bigvee_{i=1}^{n} \bigwedge_{j=1}^{\ell_i} \varphi_{i,j}$$

where the $\varphi_{i,j}$'s are atomic formulas in $\mathcal{L}$. Then, we also write the representation φ as a collection of generalized tuples t_i in the set notation:

$$\left\{ t_i \;\middle|\; 1 \leqslant i \leqslant n,\ t_i = \bigwedge_{j=1}^{\ell_i} \varphi_{i,j} \right\}$$

A finitely representable database instance I over a schema s is a collection of finitely representable relations, each associated to a relation name in the schema. In the sequel, we often use the word "object" to denote finitely representable sets of points not associated to a relation name.

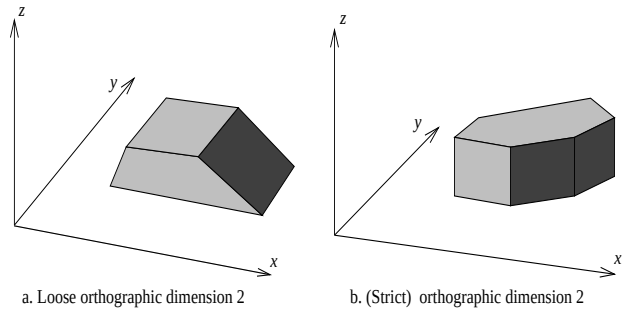
Because the structure $\mathcal{A}$ admits quantifier elimination, a relation is finitely representable iff it is finitely representable by a quantifier free formula. Therefore the restriction to quantifier free formula does not limit the definable relations. The use of quantifier free formula in DNF to represent relations, widely adopted in constraint databases, has a serious impact on the data complexity as was originally noticed in [KKR90]. In the present paper, we consider, in addition, the impact of the restrictions on the use of variables in the formulae on the complexity of query evaluation. Unless otherwise stated, we assume that all relations are finitely representable.

We introduce a first restriction based on the number of distinct variables in each atomic formula, and corresponding to the classical notion of orthographic projections.

Definition 2 A quantifier-free formula $\varphi(\bar{x})$ in $\mathcal{L}$ with d distinct variables $x_1, \ldots, x_d$ has *loose orthographic dimension* ℓ if each atom in $\varphi(\bar{x})$ involves at most ℓ distinct variables.

A relation S has *loose orthographic dimension* ℓ if there exists a representation of S with a formula of loose orthographic dimension ℓ. Note that ℓ is not unique, and if a relation has loose orthographic dimension ℓ, it has also loose orthographic dimension $(\ell + 1)$.

A convex d-dimensional object O with loose orthographic dimension ℓ can be seen as the join of all its projections on the $\binom{d}{\ell}$ ℓ-dimensional subspaces. For example, if O is a 3-dimensional object with loose orthographic dimension 2, then $O = \pi_{x,y}O \bowtie \pi_{x,z}O \bowtie \pi_{y,z}O$. That is the object O can be characterized by its orthographic projections. Figure 1.a shows an example of such an object.



a. Loose orthographic dimension 2 b. (Strict) orthographic dimension 2

Fig. 1. 3-d objects with bounded orthographic dimension

Unfortunately, the class of relations with loose orthographic dimension ℓ is not closed under fundamental operations such as for instance the projection. Consider the relation S in $\mathbb{Q}^5$ defined by the formula $x-y-z = 0 \wedge x-t-u = 0$. S has loose orthographic dimension 3. Its projection on y, z, t, u is defined by the formula $y + z - t - u = 0$ which has loose orthographic dimension 4, but is not equivalent to a formula with loose orthographic dimension 3.

We therefore propose to consider a more drastic restriction which ensures better closure properties, and tighten the conditions on the use of the variables.

We first define the notion of *dependent* variables with respect to a formula. This notion was studied in [CGK96] in the context of linear constraint databases. Let $\varphi(x_1, \ldots, x_d)$ be a formula in $\mathcal{L}$ with d distinct free variables $x_1, \ldots, x_d$. Two distinct variables which occur in the same atom in $\varphi(x_1, \ldots, x_d)$ are said to be dependent in the same atom. The dependency relation between variables in φ is the reflexive symmetric transitive closure of the dependency in the same atom relation defined above. Variables which are not dependent are said to be independent. The *orthographic partition* of the set of variables of a formula is the partition in equivalence classes of the dependency relation. We can now introduce the concept of *orthographic dimension*.

Definition 3 A quantifier-free formula $\varphi(\bar{x})$ in $\mathcal{L}$ with d distinct variables $x_1, \ldots, x_d$ has *orthographic dimension* (orthodim) ℓ if each class of its orthographic partition has cardinality at most ℓ.

It is now possible to define the orthographic dimension of a relation.

Definition 4 A relation S is of *orthographic dimension* ℓ if there exists a representation of S with a quantifier-free formula φ of orthographic dimension ℓ.

Note that, as for the loose orthographicity, the orthographic dimension of a relation is not uniquely defined. We do not consider the intrinsic unique (e.g. minimal) orthographic dimension of relations, but merely, given ℓ, the relations that are of orthographic dimension ℓ.

To a relation, we can associate a partition of its attributes as follows.

Definition 5 A relation S admits an *orthographic decomposition* $\mathcal{P}$, if there exists a representation of S with a formula φ of orthographic partition $\mathcal{P}$. The subsets of the partition are called the *components* of the decomposition.

Note that the orthographic decomposition of a relation is not unique. Indeed, a relation can be defined by different formulae with distinct orthographic partitions. It can be shown, however, that the orthographic partitions of a set of k variables form a lattice for the sub-partition relation. Nevertheless, we do not consider a unique (e.g. thiner) decomposition associated to a relation, but conversely, given a fix decomposition, the relations that admit this decomposition.

A convex d-dimensional object O with orthographic decomposition $\mathcal{P}$ is equal to the Cartesian product of all its projections on the components of $\mathcal{P}$. Figure 1.b shows the example of such an object.

Note that for $\ell = 1$, the orthographic dimension 1 coincides with the loose orthographic dimension 1. In this case, we say that a relation is *rectangular*. It admits a definition φ with exclusively constraints of the form: $x \Theta a$, where a is a constant, and Θ is a predicate.

We can now prove the following proposition which relates the orthographic decomposition to the rectangularity.

Proposition 1 A relation R admits an orthographic decomposition $\mathcal{P} = (P_1, \ldots, P_n)$, iff for each pair of variables x, y, such that $x \in P_i$, and $y \in P_j$, with $i \neq j$, and all interpretation θ of the attributes of R distinct from x, y, the set $S_\theta = \{x, y \mid R(\cdots, x, \cdots, y, \cdots)\theta\}$ is a rectangular relation.

Therefore the orthographic decomposition can be reduced to the rectangularity of the projections on planes of independent axis. In the context of o-minimal structures, the boundary of a relation is definable in $FO(<)$. A binary relation is rectangular if each point in its boundary is either (i) an isolated point, (ii) a point inside a vertical or horizontal segment, or (iii) a point on the corner of a rectangle. Each of these properties can be expressed easily by a formula of $FO(<)$. The above development together with Proposition 1, lead to the following proposition.

Proposition 2 There is a $FO(<)$ formula $\mu_{\mathcal{P}}$ such that for all instance I, $\mu_{\mathcal{P}}(I)$ is true iff I admits the orthographic decomposition $\mathcal{P}$.

The relations of bounded orthographic dimension are very sensitive to various transformations. We consider the closure of the class of relations of some fixed orthographic dimension through algebraic operations such as set operations (union, intersection, set difference), projection, selection, and transformations, such as translation, rotation, etc. The selection operator can relate by a single constraint a group of independent variables, thus possibly modifying the orthographic dimension of the input. For instance, if x and y are two independent variables in a relation R, then in $\sigma_{x \leqslant y}(R)$, x and y are in general dependent.

Note that the class of relations of orthographic dimension ℓ, in the context of the real field, is closed under set operations, projection, translation, symmetries (axial), but not under rotations.

Although rotations do not preserve the orthographic dimension, there are relations originally not of orthodim ℓ, but which can be rotated adequately to become of orthodim ℓ, that is such that there exists a vector basis in which the relation has orthodim ℓ.

We consider the general problem of deciding if a relation is of orthodim ℓ. Finitely representable relations are defined and manipulated by means of quantifier-free formulae. In general these formulae need not be of orthodim ℓ even if the intended relation is. In such a case however, they are equivalent to some formula of orthographic dimension ℓ, defining the relation of corresponding orthographic dimension. It is important to determine if this is the case. We consider the two following problems.

1. **Explicit orthographic dimension** Given a quantifier-free formula in $\mathcal{L}$ with d distinct variables, is there an equivalent formula with orthodim $\ell \leqslant d$.
2. **Implicit orthographic dimension** Given a quantifier-free formula in $\mathcal{L}$ with d distinct variables, is there a linear transformation of the system of coordinates, in which an equivalent formula with orthodim $\ell \leqslant d$ can be found.

The decidability of these two fundamental problems is characterized in the sequel. The following result can be obtained using Proposition 2.

Theorem 3 Let $\mathcal{A}$ be an $\mathcal{L}$-structure which has a decidable first-order theory. Then it is decidable if a quantifier-free formula φ in $\mathcal{L}$ is equivalent to a formula ψ with orthographic dimension ℓ.

Note that the complexity is exponential in the arity of the relation (the number of variables) and proportional to the complexity of deciding the first-order theory.

The following can be shown for the implicit orthographic dimension.

Theorem 4 Let $\mathcal{A}$ be an enumerable $\mathcal{L}$-structure which has a decidable first-order theory, and such that addition and multiplication are also decidable. Then it is decidable if a quantifier-free formula φ in $\mathcal{L}$ is equivalent, modulo a linear transformation of the system of coordinates, to a formula ψ with orthographic dimension ℓ.

More precise results can be obtained for specific choices of the language $\mathcal{L}$, and the structure $\mathcal{A}$. For instance in the case of linear constraints over rational numbers, a polynomial time bound can be obtained for both orthographic dimension problems. The following result generalizes the result of [CGK96].

Theorem 5 The problem of explicit and implicit orthographic dimension of a linear constraint relation defined by a linear formula φ can be solved in polynomial time.

3 Query languages

We consider queries which can be expressed as first-order formulae in the language $\mathcal{L}$ in the context of $\mathcal{A}$. If s is a schema, each formula φ in $\mathcal{L} \cup s$ with free variables $x_1, \ldots, x_n$ $(n \geqslant 0)$ defines a *query over* s in the context of $\mathcal{A}$, mapping instances I of s to n-ary relations defined by $\{(a_1, \ldots, a_n) \mid \mathcal{A} \sqcup I \models \varphi(a_1, \ldots, a_n)\}$.

Suppose $q = \{(x_1, \ldots, x_n) \mid \varphi\}$ is a query over s, and I is an instance of s. Since each relation R in I can be defined by a quantifier-free formula in $\mathcal{L}$, and φ is a formula in $\mathcal{L} \cup s$, we can replace in φ each occurrence of the relation symbol $R \in s$ by a formula defining R. The resulting formula φ' is a formula in $\mathcal{L}$ with no reference to relation symbols in s, which defines the *answer to the query* q *on* I, denoted by $q(I)$.

For complexity reasons, we consider as usual answers defined by a quantifier-free formula ψ in $\mathcal{L}$ such that φ' and ψ are logically equivalent in $\mathcal{A}$. We assume in the sequel that the structure $\mathcal{A}$ admits effective quantifier elimination.

As mentioned in the previous section, there are queries manipulating data in dimension d with orthodim ℓ, which do not have an output of orthodim ℓ. This is the case for instance of $\sigma_{x \leqslant y}(R)$, where x and y are variables from two different components of the orthographic decomposition. We restrict our attention to queries preserving the orthographic decomposition.

Definition 6 Let s be a database schema, and $\mathcal{P}$ be an orthographic decomposition of the relations in s. A query Q over s *preserves the orthographic decomposition* $\mathcal{P}$ if for each instance I of orthographic decomposition $\mathcal{P}$, $q(I)$ admits an orthographic decomposition which refines $\mathcal{P}$.

We consider also a generalization of the previous preservation property with no reference to a specific decomposition. A query Q over s *preserves the orthographic decomposition* if for each instance I of orthographic decomposition $\mathcal{P}$, $q(I)$ admits an orthographic decomposition which refines $\mathcal{P}$. It is clear that

if a query preserves the orthographic decomposition, it also preserves a given orthographic decomposition $\mathcal{P}$.

As for many other preservation properties of queries [VGV96,BL98], we prove that the preservation of the orthographic decomposition is undecidable in a o-minimal context structure, and for a schema with at least a binary relation symbol.

Theorem 6 Let $\mathcal{A}$ be an o-minimal context $\mathcal{L}$ structure, and s be a schema with a relation symbol of arity at least 2. Then it is undecidable if a formula in $\mathcal{L} \cup s$ expresses a query over s which preserves the orthographic decomposition.

Proof : The proof is done by a reduction from the problem of satisfaction of a Boolean query, which was shown to be undecidable under the assumptions of the theorem [GS99]. Consider a schema with a single relation R of arity n with a non trivial orthographic decomposition. The formula:

$$\bigwedge_{0<i<j<n} x_i \leqslant x_j \wedge \varphi(R)$$

defines a query which preserves the orthographic decomposition of the input iff $\varphi(R)$ is unsatisfiable. □

If we restrict our attention to the class of conjunctive queries, then preserving the orthographic decomposition becomes decidable.

Proposition 7 It is decidable whether a conjunctive query preserves the orthographic decomposition.

Proof :(sketch) Let q be a conjunctive query. It can be recursively changed into an equivalent query of the form :

$$\pi_A \ \sigma_F \ R_1 \times ... \times R_n$$

where A is a set of attributes and F a conjunction of constraints.

Then it suffices to compute the non rectangular connections between pairs of variables from distinct components introduced by F. These are all pairs of variables dependent in F, but those introducing a rectangular connection. The connection between two variables x, y is said to be rectangular if the projection over the plane defined by x, y of $\sigma_F \ R_1 \times ... \times R_n$ is a rectangular relation. Since this property can be expressed in first-order logic and that the structure admits effective quantifier elimination, it is decidable.

We then check whether all connections between components added by F are *destroyed* by the projection on A. If this is the case, the query clearly preserves the orthographic decomposition. If this is not the case it is possible to construct an instance I such that $q(I)$ has an orthographic decomposition which does not refine the one of I. □

We now provide a syntactic characterization of the set of orthographic decomposition preserving queries. Let $\mu_{\mathcal{P}}$ be a Boolean formula (cf. Proposition 2) which holds if ϕ defines a relation that admits decomposition $\mathcal{P}$.

Let s be a schema, $\mathcal{P} = (P_1, ..., P_n)$ an orthographic decomposition, and I an instance of s, represented by a formula Φ_I, which satisfies the orthographic decomposition $\mathcal{P}$. Consider a formula $\varphi(y_1, \ldots, y_k)$ in $\mathcal{L} \cup s$, and assume wlog that $\{y_1, \ldots, y_k\} \subseteq \{x_1, \ldots, x_d\}$. A formula of the form

$$\varphi(y_1, \ldots, y_k) \wedge \mu_{\mathcal{P}}(\Phi_I) \rightarrow \mu_{\mathcal{P}}(\varphi)$$

is called *decomposition $\mathcal{P}$ restricted.*

The following proposition follows easily from Proposition 2.

Theorem 8 The class of orthographic decomposition preserving queries and the class of decomposition restricted queries coincide.

The definition of decomposition restricted formulae is of little use in practice. It is not natural to write queries in this form. We therefore propose a second restriction which is intuitive and easy to check.

In order to give the next syntactic restriction, we consider an algebra, equivalent to first-order logic. The algebra consists of the following operations: Cartesian product, $\times$, projection, π, union, $\cup$, set difference, $-$. The algebra operations, whose effect is described below, are performed on sets of generalized tuples. Let R_1 and R_2 be two relations, and respectively e_1 and e_2 be sets of generalized tuples defining them.

1. $R_1 \times R_2 = \{t_1 \wedge t_2 \mid t_1 \in e_1, t_2 \in e_2\}$.
2. $\sigma_F(R_1) = \{t_1 \wedge F \mid t_1 \in e_1\}$ where F is any atom of $\mathcal{L}$.
3. $\pi_{\overline{x}}\ R_1$ is computed using the algorithm of quantifier elimination[1].
4. $R_1 \cup R_2 = e_1 \cup e_2$.
5. $R_1 - R_2 = \{t_1 \wedge t_2 \mid t_1 \in e_1, t_2 \in (e_2)^c\}$, where e^c is the set of tuples or disjuncts of a DNF formula corresponding to $\neg e$.
6. $simplify(R_1)$, eliminates redundancies and detects inconsistencies.

The symbolic operations above are well defined in the sense that their effect on the intensional definition of sets corresponds exactly to the semantics of the corresponding relational operators from the relational algebra over the possibly infinite extension of the sets. The relational intersection and join are definable with the symbolic Cartesian product. The sets of variables are different in these three operations. For the Cartesian product, the variables of the two relations are disjoint, they are similar in the case of the intersection, and with a non empty intersection in the case of the join and the selection. The *simplify* operator, given a conjunction of constraints, eliminates redundancies, and detects inconsistencies. Simplification is necessary for checking the satisfiability of formulae in $\mathcal{A}$.

All the logical operators, except selection, preserve the orthographic decomposition and can be carried out independently on each component of the input. For instance, the intersection of two objects can be processed by composing,

[1] In the linear case this is done by the Fourier-Motzkin elimination method [Sch86].

at the tuple's level, the intersection of the corresponding components. Union, set difference and simplification can also be carried out independently on each component. Cartesian product does not affect the components, and projection applies to the appropriate components.

One way to be sure that a query preserves the orthographic dimension is to forbid selections introducing a binding between components. But this would limit in a drastic way the expressive power of the language. Some queries preserving the orthographic dimension might need intermediate result which have a higher dimension. Indeed queries like $\pi_x\ \sigma_{x<y}(R)$ where x and y are independent variables would not be expressible.

We now introduce a class of restricted algebraic expressions preserving an orthographic decomposition $\mathcal{P}$ which allows intermediate higher dimension. Basically, it consists in checking that when a selection introduces a binding between independent variables, this binding is further eliminated through a projection.

Let E be an algebraic expression. We recursively define the collection of bad bindings, as the set $BB(E)$ of pairs of components which are linked at each step of the computation process.

1. If E is a relation name or variable, $BB(E) = \emptyset$.
2. If $E = \sigma_F(E_1)$, then $BB(E)$ is the union of $BB(E_1)$ with the collection of pairs $< C_i, C_j >$ such that x is a variable of C_i, y is a variable of C_j and x and y are bound in F, where C_i and C_j are distinct components of $\mathcal{P}$.
3. If $E = \pi_{x_{i_1},\ldots,x_{i_n}}(E_1)$, then $BB(E)$ is obtained by removing from $BB(E_1)$ all pairs p such that one of the components in p is only defined with variables that don't appear in $\{x_{i_1}, \ldots, x_{i_n}\}$.
4. If $E = E_1 \Theta E_2$ with Θ among $\{\cup, \cap, -, \times\}$, then $BB(E) = BB(E_1) \cup BB(E_2)$.

Now if E is an algebraic expression such that $BB(E) = \emptyset$, E is said to be bad-binding free.

Definition 7 $\mathcal{P}$-*Safe* queries are defined as the set of bad-binding free algebraic queries.

The following is immediate :

Proposition 9 Every $\mathcal{P}$-*safe* query preserves the orthographic decomposition.

The natural question is whether this restriction captures all the orthographic decomposition preserving queries. We can only give a partial answer to this question.

Theorem 10 Every conjunctive query which preserves the orthographic decomposition $\mathcal{P}$ is equivalent to a $\mathcal{P}$-*safe* query.

Proof :(sketch) Let φ be a conjunctive query preserving the orthographic dimension. It can be recursively transformed into a equivalent query ψ of the form $\pi_A \sigma_F(R)$ where R is a cross product. If $BB(\psi) = \emptyset$ we are done. If this is not

the case, it means that there are variables which are connected by F but which have a *rectangular* connection. Let x and t be such variables. In ψ we replace equivalently the selection criteria F by F' which is the formula $\exists t\ F \wedge \exists x\ F$, after quantifier elimination. This can be done because F is a conjunction of constraints and represent a convex set. By repeating this process we get an equivalent conjunctive query with an empty BB. □

It is possible to extend the above theorem to the disjunction of conjunctive queries. But it is not clear how to extend it to more general classes of queries like Boolean combination of conjunctive queries.

In the following section, we restrict our attention to a popular context structure of practical interest, and show the importance of the orthographic dimension for query evaluation.

4 Linear constraint databases

This section is limited to the case where the context structure is $\mathcal{A} = (\mathbb{Q}, \leqslant, +)$. The finitely representable relations are called linear constraint relations. We study the impact of the various parameters which characterize a linear instance (such as the number of variables, the orthographic dimension, the number of constraints, etc.) on the complexity of queries, and show that the evaluation process can take advantage of the orthographic dimension of an input. We exhibit queries on d-dimensional databases which can be solved using only manipulation of ℓ-dimensional pointsets.

We first consider the complexity of the algebraic operations. We consider an algebra, containing the operators, $\times$, π, $\cup$, $-$, σ and $simplify$ presented above.

The DNF formulae representing pointsets are structured with the following constructs: (i) a top-level disjunction, (ii) conjunctions, (iii) predicates of the atomic constraints, and finally (iv) the parameters of the constraints.

The algebraic operations apply to different levels of the formulae. $\cup$, $\times$, and σ_F apply at the level of the logical connectives and can be evaluated in a purely symbolic way independently of the dimension. The other operations have an effect on lower levels. Their complexity depends strongly upon the dimension of the input.

- Set difference has an effect at all levels till the predicates of the constraints, which can be modified (e.g. $<$ replaced by $>$). It can be computed by first computing the cell decomposition of the space induced by all the constraints that occur in both relations, and then checking which cells are in the result. Cell decomposition can be computed in time complexity $O(n^{d+1})$ [GO97].
- Projection has an effect at all levels including the parameters, and implies numerical computation. The Fourier-Motzkin algorithm [Sch86] can be used to eliminate one variable of a convex set of n facets in dimension d in time complexity $O(n^2)$. If k variables need to be eliminated, the time complexity is then $O(n^{2^k})$. A more subtle algorithm would be to *simplify* after eliminating each variable, which reduces the output to a size linear in n. The overall

complexity to eliminate k variables is then : $O((n^2+2^{2^k}n^4)+...+(n^2+2^{2^k}n^4))$ which is $O(k2^{2^k}n^4)$.

- The complexity of simplification (and therefore satisfaction) of n linear constraints in dimension d can be found in [Sch86,GO97]. It is essentially exponential in the dimension.

Figure 2 summarizes the costs of the algebraic operators in dimension 1, 2, 3 and d with respect to the following parameters: n is the total number of constraints in the relations, t is the number of tuples, and k the number of variables projected out. All complexities are upper-bounds given modulo a coefficient factor. The blow up in complexity comes from projection, set difference, and simplification.

Dimension $\rightarrow$ Operator $\downarrow$	1	2	3	d
$\times$	t^2	t^2	t^2	t^2
$\cup$	1	1	1	1
π_x	n	$n \log n$	n^4	$k2^{2^k}n^4$
σ_F	t	t	t	t
$-$	$n \log n$	n^2	n^4	n^{d+1}
$simplify$	n	$n \log n$	n^2	$(2^{2^d})n^2$

Fig. 2. Complexity of the operators in the dimension

In order to reduce the complexity, we consider input relations of orthodim bounded by a given ℓ. As mentioned in Section 3, queries over such relations can be processed independently over each component except when a selection binds two components. In this later case, it seems necessary to use operators which operate on pointsets whose dimension is higher than ℓ.

We introduce a new selection operator which preserves the orthographic decomposition, and can be used in place of the classical selection. The new symbolic selection, denoted by $\tilde{\sigma}_F$, computes, for each generalized tuple t, the projection $\pi_{C_i}(F \wedge t)$ for each component C_i and returns $\tilde{\sigma}_F(t) = \pi_{C_1}(F \wedge t) \times \ldots \times \pi_{C_n}(F \wedge t)$. Note that $\tilde{\sigma}_F(t)$ is not equal to $\sigma_F(t)$. It defines an approximation of $\sigma_F(t)$, such that the two coincide on each component. We show how this new selection can be used in place of the traditional one.

To catch the intuition, consider a simple conjunctive $\mathcal{P}$-*safe* query $q(R) = \pi_z(\sigma_F(R))$, in the context of a relation of dimension 3 and orthodim 2, with decomposition $\mathcal{P} = (\{x, y\}, \{z\})$. We focus on the evaluation of q on a single tuple, O (see Figure 3.a).

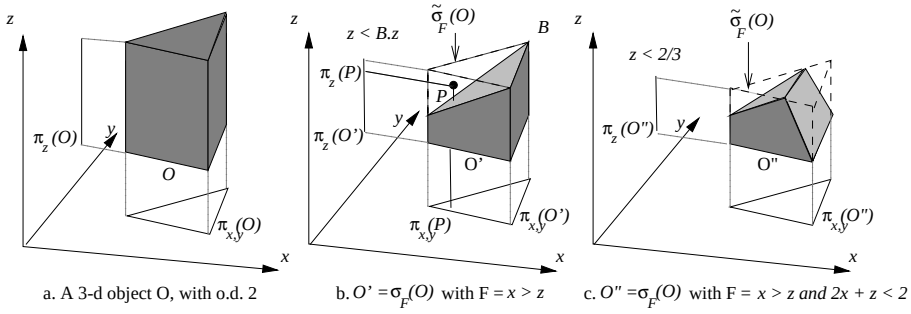


Fig. 3. Selection over a 3-d object of orthodim 2

In Figure 3.b: the selection $\sigma_F(O)$, with $F = x > z$, yields a polytope O'. The exact value of O' is not relevant for the final output of the query which is $\pi_z(O')$: O' can be approximated by $\tilde{\sigma}_F(O)$ (in dashed lines on Figure 3.b). Two comments are noteworthy. First, the result features a new constraint $z < \alpha$ where α is the highest coordinate of O', $B.z$. Note that the point B can be simply computed using F and the bounding-box of O. Second, a point in $\tilde{\sigma}_F(O)$, P for instance, might not belong to O', but its projections on $\{x, y\}$ and $\{z\}$ belong to $\pi_{x,y}(O')$ and $\pi_z(O')$ respectively.

In Figure 3.c, $F = x > z \wedge 2x + z < 2$ consists of two constraints. The same technique applies: we approximate the result of $\sigma_F(O)$ by the cross-product of its components. The new constraint in the final result is here $z < 2/3$, a constant value which results from the intersection of the half-planes $x > z$ and $2x + z < 2$. Note that it depends upon the query *and not upon the database* and can thus be computed only once.

We now introduce ALG^ℓ as the set of queries expressed with the usual algebra where σ is replaced by $\tilde{\sigma}$.

Definition 8 ALG^ℓ is the class of queries which can be expressed by the following operations:

- $\cup$, $\times$
- π^ℓ, $-^\ell$, and $simplify^\ell$ which are the usual projection, negation, and simplification but restricted to inputs of dimension at most ℓ.
- $\tilde{\sigma}_F$.

The main interest of ALG^ℓ is that the complexity of evaluating queries is linear in d and exponential in ℓ only.

Lemma 11 Let $\mathcal{I}$ be a class of instances of dimension d with orthodim ℓ. The complexity of evaluating queries in ALG^ℓ over $\mathcal{I}$ is exponential in ℓ, but only linear in d.

Proof : As for the linearity in d, note that since all the operators used to expressed queries in ALG^{ℓ} preserve the orthographic dimension, each operator can be independently applied in parallel to the component of its input. It remains to prove that the complexity of each operator is at most exponential in ℓ. Given the complexities of Fig. 2, the result is immediate for all, except $\tilde{\sigma}$. Indeed it seems that the computation of $\pi_{C_i}(F \wedge t)$ for each component C_i involves an unrestricted projection π^k with $\ell < k \leqslant d$. We develop in the sequel an evaluation which avoids the costly computation of π^k.

The first step consists in computing, for each tuple t, $mbb(t)$, the minimal bounding-box of t. This can be obtained by applying $simplify^{\ell}$ on the input of $\tilde{\sigma}$. Once $mbb(t)$ is known, $\tilde{\sigma}(t)$ is computed by:

$$E(t) \equiv t \wedge \pi_{C_1}(F \wedge mbb(t)) \wedge \ldots \wedge \pi_{C_n}(F \wedge mbb(t))$$

where the C_i's are the components. The operator π is introduced in a logical formula as an abreviation. We show that $E(t)$ is indeed equivalent to $\tilde{\sigma}(t)$:

1. ($\subseteq$) First, if $p \in E(t)$, then $p \in t$ and p satisfies $\pi_{C_1}(F \wedge mbb(t)) \wedge \ldots \wedge \pi_{C_n}(F \wedge mbb(t))$. Therefore each component C_i in p belongs to $\pi_{C_i}(F \wedge t \wedge mbb(t))$ which is $\pi_{C_i}(\sigma_F(t))$.
2. ($\supseteq$) If now $p \in \tilde{\sigma}_F(t) = \pi_{C_1}(F \wedge t) \times \ldots \times \pi_{C_n}(F \wedge t)$, then $p \in t$ since by hypothesis t is equivalent to the cross product of the projections on its components, and the projection of p on each component satisfies the projection of F. Hence p satisfies $E(t)$ which is equivalent to $\tilde{\sigma}_F(t)$

Finally, we show that $E(t)$ can be computed in constant time for each tuple as follows. Assume that $F = \{a^j\overline{x} + y \leqslant a_0^j, j = 1, \ldots, J\} \cup \{b^k\overline{x} - y \leqslant b_0^k, k = 1, \ldots, K\}$, where $\overline{x}$ ranges over $\mathbb{Q}^{d-1}$ and y ranges over $\mathbb{Q}$, and that $mbb(t)$ is defined by the couples $(x_{min}^l, x_{max}^l), l = 1, \ldots, d-1)$ and (y_{min}, y_{max}). Then the Fourier-Motzkin projection $\pi_{\overline{x}}(F \wedge mbb(t))$ is:

$$\begin{cases} b^k\overline{x} - b_0^k \leqslant a_0^j - a^j\overline{x} & \text{for } j = 1, \ldots, J, k = 1, \ldots, K \quad (1) \\ b^k\overline{x} - b_0^k \leqslant y_{max} & \text{for } k = 1, \ldots, K \quad (2) \\ a_0^j - a^i\overline{x} \geqslant y_{min} & \text{for } j = 1, \ldots, J \quad (3) \\ x_{min}^l \leqslant x^l \leqslant x_{max}^l & \text{for } l = 1, \ldots, d-1 \quad (4) \end{cases}$$

The set of constraints in (1) depends only upon F and can thus be computed only once. Constraints in (2), (3) and (4) depends upon the bounding box and must therefore be computed for each tuple, but their number is linear in the size of F. This means that, when evaluating $E(t)$, we can replace the costly computation of π by the computation of (2) and (3) in time $O(|F|)$.

The algorithm for computing $\tilde{\sigma}$ is summarized below:

1. **Step 1** Project F, using the Fourier-Motzkin algorithm, on each component. Note that this operation is done on the query and not on the database. For each component C, this gives a set of constraints similar to those of expression (1) above. See for instance the example of Figure 3.c, where $z < 2/3$ is obtained by applying Fourier-Motzkin on $F = x > z \wedge 2x + z < 2$.

2. **Step 2** For each tuple t,
   (a) **2.a** Simplify t with the $simplify^\ell$ operator. This yields as a side effect $mbb(t)$.
   (b) **2.b** Compute expressions (2) and (3) above for every component C_i. This can be done in time $O(|F|)$:
      i. For each constraint cst in F, put cst in the form $f(x_1^i, \ldots x_l^i)\,\Theta\,\Phi(x_1^j, \ldots x_m^j)$ where f and Φ are linear functions, the x^i are the variables of C_i and $\Theta \in \{\leqslant, \geqslant\}$.
      ii. If Θ is $\geqslant$ (resp. $\leqslant$), compute the local minimum (resp. maximum) L of Φ using the values of $mbb(t)$,
      iii. Output the constraint $f(x_1^i, \ldots x_l^i)\Theta L$.
   (c) **2.c** Finally, construct $E(t)$ using the conjunctions of constraints in (1, 2, 3, 4).

The operation $\tilde{\sigma}$ can be computed with $simplify^\ell$ and an $O(1)$ operation. This concludes the proof of Lemma 11. □

Intuitively, queries in ALG^ℓ can be evaluated in parallel on each component. It remains to characterize the class of queries over databases with orthodim ℓ which can be equivalently rewritten as queries in ALG^ℓ. Note first that all first-order queries can be expressed in some ALG^ℓ. Indeed, it is easy to see that FO $= \cup_\ell ALG^\ell$. On the other hand, if $\mathcal{P}$ (and thus ℓ) is fixed, it is conjectured that not all first-order queries on databases of orthographic decomposition $\mathcal{P}$ can be expressed in ALG^ℓ, even if the query is supposed to be $\mathcal{P}$-*safe*.

Consider the relation $R(x, y, z)$ with orthographic partition $(\{x, y\}, \{z\})$, the relation $S(y, z)$, and the $\mathcal{P}$-*safe* query:

$$q = \pi_y(\underbrace{\pi_{y,z}(\sigma_{x<z}(R))}_{A} - S)$$

The subquery A is computed by $\pi^3_{y,z}(R \wedge x < z)$ which is in ALG^3. This suggests that the hierarchy of ALG^ℓ is strict.

However, if we restrict our attention to $\mathcal{P}$-*safe* Boolean combinations of conjunctive queries (BCCQ) we can state the following fundamental result.

Theorem 12 Let s be a database schema of orthographic decomposition $\mathcal{P}$ of orthodim ℓ. Let q be a $\mathcal{P}$-safe BCCQ over s. Then q can be easily rewritten in an equivalent query q' in ALG^ℓ.

Proof: We first prove the result in the conjunctive case. Let q be a $\mathcal{P}$-*safe* conjunctive query, where $\mathcal{P} = \{C_1, \ldots C_n\}$ is the orthographic decomposition of the schema s. As already mentioned, (see Proposition 7), it can be put into the form $\pi_A(\sigma_F(R_1 \times \ldots \times R_\ell))$, where A admits an orthographic decomposition $\{C'_1, \ldots C'_m\}$ which is a refinement of $\mathcal{P}$.

Lemma For each instance I of s, $q(I) \equiv \pi_A(\sigma_F(R_1 \times \ldots \times R_\ell))$ is equivalent to $q'(I) \equiv \pi_A(\tilde{\sigma}_F(R_1 \times \ldots \times R_\ell))$.

Proof of Lemma: Let t be a tuple in $q(I)$, and t' be the tuple in $R_1 \times \ldots \times R_\ell$ such that $t = \pi_A(\sigma_F(t'))$. Since q is $\mathcal{P}$-*safe*,

$$t \equiv \pi_{C'_1}(t) \times \ldots \times \pi_{C'_m}(t) \equiv \pi_{C'_1}(\sigma_F(t')) \times \ldots \times \pi_{C'_m}(\sigma_F(t'))$$

For each $C'_i \in A$, there exists some $C_{j_i} \in \mathcal{P}$ such that (i) $C'_i \subseteq C_{j_i}$ and (ii) $\forall k \neq i, C'_k \cap C_{j_i} = \emptyset$. Therefore we have:

$$\pi_{C'_i}(t) \equiv \pi_{C'_i}(\pi_{C_{j_i}}(t)) \equiv \pi_A(\pi_{C_{j_i}}(t))$$

It follows that $t \equiv \pi_A(\pi_{C_{j_1}}(\sigma_F(t'))) \times \ldots \times \pi_A(\pi_{C_{j_m}}(\sigma_F(t')))$ which can be equivalently rewritten as
$\pi_A(\pi_{C_{j_1}}(t') \times \ldots \times \pi_{C_{j_m}}(t'))$ because the set $\{C_{j_k}, k \in \{1, \ldots m\}\}$ is an orthographic partition. This proves that $t \equiv \pi_A(\tilde{\sigma}_F(t'))$, and thus that $q(I) \equiv q'(I)$. This concludes the proof of the Lemma. □

Now, if q is a $\mathcal{P}$-*safe* BCCQ query, is consists of a Boolean combination of conjunctive queries $\{q_1, \ldots q_n\}$. Since q is $\mathcal{P}$-*safe*, we have $BB(q) = \emptyset = BB(q_1) \cup \ldots \cup BB(q_n)$. Hence $BB(q_i) = \emptyset$, for $i \in \{1, \ldots n\}$. Each q_i is safe, belongs to ALG^ℓ, and yields relations with orthodim ℓ. Finally q itself involves Boolean operations on relations with orthodim ℓ, and belongs to ALG^ℓ. □

The new query q' can be obtained easily. Indeed, if q is a Boolean combination of conjunctive queries of the form $\pi\sigma$, then q' is obtained from q by simply replacing all the σ by $\tilde{\sigma}$. We can therefore conclude that there exists an evaluation of $\mathcal{P}$-*safe* BCCQs such that each operator manipulates only pointsets of dimension ℓ. These results can be summarized as follows.

Corollary 13 Let s be a database schema of orthographic decomposition $\mathcal{P}$. The complexity of evaluating $\mathcal{P}$-safe BCCQs over s depends only linearily upon the global dimension.

This follows directly for Lemma 11 and Theorem 12. It is open whether the result extends to more general queries.

5 Conclusion

We have introduced restrictions on the geometry of the objects contained in a multidimensional databases, which (i) can be easily characterized by syntactic restrictions on the constraint formulae that represent the database, and (ii) ensure better performance for the evaluation of large classes of queries such as $\mathcal{P}$-*safe* BCCQs. We have indeed demonstrated that the complexity of query evaluation depends linearly upon the global dimension.

We have thus restricted both the class of databases, and the class of queries of interest. Although both classes are of great practical interest, we can try to weaken the restrictions. The poor closure properties of the class of inputs with bounded loose orthographic dimension, lead us to consider strict orthographic

dimension instead. Nevertheless, this class deserves more study in the case of relations of small dimension (e.g. $d \leqslant 3$). It can be shown that several results of the paper (Theorem 3, 4, etc.) extend to the case of such relations.

The class of queries can also be extended in various directions. For instance, the subclass of $\mathcal{P}$-*safe* queries with projection limited to projection on variables of a same component, enjoys also a similar property. They can be rewritten in ALG^{ℓ} queries.

A spatio-temporal application runs on the DEDALE system with objects of orthodim 2 [GRS98a]. The restriction is transparent to the user, and the evaluation of $\mathcal{P}$-*safe* BCCQs relies on 2d operations only. This allows the manipulation of multidimensional data at the cost of 2d data.

References

[BL98] M. Benedikt and L. Libkin. Safe constraint queries. In *Proc. ACM Symp. on Principles of Database Systems*, 1998.

[CGK96] J. Chomicki, D.Q. Goldin, and G. Kuper. Variable Independence and Aggregation Closure. In *Proc. ACM Symp. on Principles of Database Systems*, pages 40–48, 1996.

[GK97] S. Grumbach and G. Kuper. Tractable recursion over geometric data. In *International Conference on Constraint Programming*, 1997.

[GO97] Jacod E. Goodman and Joseph O'Rourke. *Handbook of Discrete and Computational Geometry*. CRC Press, 1997.

[GRS98a] S. Grumbach, P. Rigaux, and L. Segoufin. Spatio-Temporal Data Handling with Constraints. In *Proc. Intl. Symp. on Geographic Information Systems*, 1998.

[GRS98b] S. Grumbach, P. Rigaux, and L. Segoufin. The DEDALE System for Complex Spatial Queries. In *Proc. ACM SIGMOD Symp. on the Management of Data*, 1998.

[GS99] S. Grumbach and J. Su. Finitely representable databases. *Journal of Computer and System Sciences*, Vol 55(2), pages 273-298, 1997.

[GST94] S. Grumbach, J. Su, and C. Tollu. Linear constraint query languages: Expressive power and complexity. In D. Leivant, editor, *Logic and Computational Complexity*, Indianapolis, 1994. Springer Verlag. LNCS 960.

[KG94] P. Kanellakis and D. Goldin. Constraint programming and database query languages. In *Manuscript*, 1994.

[KKR90] P. Kanellakis, G Kuper, and P. Revesz. Constraint query languages. In *Proc. 9th ACM Symp. on Principles of Database Systems*, pages 299–313, Nashville, 1990.

[KKR95] P.C. Kanellakis, G.M. Kuper, and P.Z. Revesz. Constraint query languages. *Journal of Computer and System Sciences*, 51:26–52, 1995.

[KPV95] B. Kuijpers, J. Paredaens, and J. Van den Bussche. Lossless representation of topological spatial data. In M. J. Egenhofer and J. R. Herring, editors, *Advances in Spatial Databases, 4th Int. Symp., SSD'95*, pages 1–13. Springer, 1995.

[Mai83] D. Maier. *The Theory of Relational Databases*. Computer Science Press, 1983.

[PVV94] J. Paredaens, J. Van den Bussche, and D. Van Gucht. Towards a theory of spatial database queries. In *Proc. 13th ACM Symp. on Principles of Database Systems*, pages 279–288, 1994.

[Sch86] A. Schrijver. *Theory of Linear and Integer Programming.* Wiley, 1986.

[Ull88] J.D. Ullman. *Database and Knowledge Base Systems.* Computer Science Press, 1988.

[VGV96] L. Vandeurzen, M. Gyssens, and D. Van Gucht. On query languages for linear queries definable with polynomial constraints. In *Proc. Second Int. Conf. on Principles and Practice of Constraint Programming*, pages 468–481. LNCS 1118, August 1996.

[VMM94] L. Van den Dries, A. Macintyre, and D. Marker. The elementary theory of restricted analytic fields with exponentiation. *Annals of Mathematics*, 1994.