

Linear versus Polynomial Constraint Databases

Peer-reviewed author version

KUIJPERS, Bart (2015) Linear versus Polynomial Constraint Databases. In:
Shekhar, Shashi; Xiong, Hui (Ed.). Encyclopedia of GIS, p. 612-615.

Handle: <http://hdl.handle.net/1942/18591>

Linear versus Polynomial Constraint Databases

Bart Kuijpers

Databases and Theoretical Computer Science Research Group

Hasselt University and Transnational University of Limburg

`bart.kuijpers@uhasselt.be`

SYNONYMS

Constraint Database Systems; Constraint Query Languages; Infinite Relational Databases; Linear Constraint Databases; First Order Logic with Constraints Queries; Moving Object Constraint Databases; Spatiotemporal Constraint Databases

DEFINITION

The framework of *constraint databases* provides a rather general model for spatial databases [14]. In the constraint model, a spatial database contains a finite number of relations, that, although conceptually viewed as possibly infinite sets of points in the real space, are represented as a finite union of systems of polynomial equations and inequalities.

More specifically, in the *polynomial constraint database model*, a relation is defined as a boolean combination (union, intersection, complement) of subsets of some real space \mathbf{R}^n (in applications, typically $n = 2$ or 3) that are definable by polynomial constraints of the form $p(x_1, \dots, x_n) \geq 0$, where p is a polynomial in the real variables x_1, \dots, x_n with integer coefficients. For example, the spatial relation consisting of the set of points on the upper half of the unit disk in \mathbf{R}^2 can be represented by the formula $x^2 + y^2 \leq 1 \wedge y \geq 0$. In mathematical terminology, these spatial relations are known as *semi-algebraic* sets [4].

A basic query language for constraint databases is obtained by extending the relational calculus with polynomial inequalities [14]. This language is usually referred to as **FO + poly**. The query deciding whether the two-dimensional spatial relation S is a straight line, for instance, can be expressed in this language by the sentence

$$\exists a \exists b \exists c (\neg(a = 0 \wedge b = 0) \wedge \forall x \forall y (S(x, y) \leftrightarrow ax + by + c = 0)).$$

Formulas in the logic **FO + poly** can also be used to define output sets. For instance, the formula

$$\exists \varepsilon (\varepsilon \neq 0 \wedge \forall x' \forall y' ((x - x')^2 + (y - y')^2 < \varepsilon^2 \rightarrow S(x', y'))) \quad (1)$$

defines the topological interior of a two-dimensional set S .

Although variables in expressions such as (1) range over the real numbers, queries expressed in this calculus can still be computed effectively. In particular, the closure property holds: *any FO + poly query, when evaluated on a spatial database in the constraint model, yields a spatial database in the constraint model*. This follows immediately from Tarski's quantifier-elimination procedure for the first-order theory of real closed fields [17]. For instance, the evaluation of the query expressed by (1), on the input relation containing the upper half disk given by $x^2 + y^2 \leq 1 \wedge y \geq 0$, can be obtained by plugging in this description of the input relation in the query formula where S appears, resulting in the formula

$$\exists \varepsilon (\varepsilon \neq 0 \wedge \forall x' \forall y' ((x - x')^2 + (y - y')^2 < \varepsilon^2 \rightarrow x'^2 + y'^2 \leq 1 \wedge y' \geq 0))$$

and then eliminating the quantifiers. This would result in a formula like $(x^2 + y^2 = 1 \wedge y \geq 0) \vee (y = 0 \wedge -1 \leq x \leq 1)$, describing the output relation.

In the *linear constraint database model*, in the description of relations, only linear polynomial constraints are allowed and in the formulation of queries the relational calculus with linear inequalities is used. The latter

language is usually referred to as $\text{FO} + \text{lin}$. Spatial relations definable by linear polynomial constraints are known as *semi-linear* sets. Also in the context of linear constraints there is a closure property: *any $\text{FO} + \text{lin}$ query, when evaluated on a spatial database in the linear constraint model, yields a spatial database in the linear constraint model.*

HISTORICAL BACKGROUND

The polynomial and linear constraint database models were introduced by Kanellakis, Kuper, and Revesz [11] in 1990. This model was well-studied in the 1990s and has become mainstream database research with practical and mathematical motivations. The research on the constraint database model has been published in all important database conferences and computer science journals. A state of the art book “Constraint databases,” edited by G. Kuper, L. Libkin, J. Paredaens appeared in 2000 [12], and the textbook “Introduction to Constraint Databases” by P. Revesz was published in 2002 [15].

The PhD thesis of Luc Vandeurenzen was devoted to the study of the linear constraint database model [18].

SCIENTIFIC FUNDAMENTALS

Here, we discuss some differences in expressive power between $\text{FO} + \text{poly}$ and $\text{FO} + \text{lin}$. In particular, we will illustrate that the expressive power of $\text{FO} + \text{lin}$ is less than that of $\text{FO} + \text{poly}$.

As mentioned before, the topological interior of a two-dimensional set S can be expressed in $\text{FO} + \text{poly}$ by the formula (1). Since the topology of \mathbf{R}^2 based on open discs is equivalent to the one based on open rectangles, we can equivalently express the topological interior of a semi-algebraic subset of \mathbf{R}^2 in $\text{FO} + \text{lin}$ by the formula

$$\exists \varepsilon (\varepsilon > 0 \wedge \forall x' \forall y' ((|x - x'| < \varepsilon \wedge |y - y'| < \varepsilon) \rightarrow S(x', y'))).$$

But there are other queries for which the multiplication seems to be really necessary to express them. If we want to express that a two-dimensional semi-linear set is *convex*, for instance, then we can do this in $\text{FO} + \text{poly}$ with the formula

$$\forall \vec{x} \forall \vec{y} (S(\vec{x}) \wedge S(\vec{y}) \rightarrow \forall \lambda (0 \leq \lambda \leq 1 \rightarrow S(\lambda \vec{x} + (1 - \lambda) \vec{y}))).$$

Clearly, the subexpression $\lambda \vec{x} + (1 - \lambda) \vec{y}$ uses quadratic polynomials and it may seem difficult to imagine that convexity of semi-linear sets might be expressible without multiplication. But it turns out that a semi-linear set of \mathbf{R}^n is convex if and only if it is closed under taking midpoints. We can therefore express convexity of semi-linear sets by the $\text{FO} + \text{lin}$ -formula

$$\forall \vec{x} \forall \vec{y} (S(\vec{x}) \wedge S(\vec{y}) \rightarrow \exists \vec{z} (2\vec{z} = \vec{x} + \vec{y} \wedge S(\vec{z}))).$$

We can wonder whether all $\text{FO} + \text{poly}$ expressible queries and properties on semi-linear sets are also expressible in $\text{FO} + \text{lin}$. Clearly, $\text{FO} + \text{poly}$ is more expressive than $\text{FO} + \text{lin}$ for what concerns queries that return some n -dimensional result (the constant query that returns on any input the n -dimensional unit sphere, for instance, is not expressible in $\text{FO} + \text{lin}$). But also for what concerns properties of sets, it turns out that $\text{FO} + \text{poly}$ is more expressive than $\text{FO} + \text{lin}$, as is illustrated by the following result: *the boolean query deciding whether a semi-linear subset S of \mathbf{R}^2 contains a line is expressible in $\text{FO} + \text{poly}$, but not in $\text{FO} + \text{lin}$ [3].*

As another example, we give the boolean query deciding whether a semi-linear set S contains real numbers u and v satisfying $u^2 + v^2 = 1$. This query is expressible in $\text{FO} + \text{poly}$, but not in $\text{FO} + \text{lin}$ [1].

Several researchers have addressed the expressive power of $\text{FO} + \text{lin}$ as a query language (see e.g., [9, 18]). For $\text{FO} + \text{lin}$, also a number of more technical expressiveness results were obtained that concern the expressibility of on finite semi-linear sets. In particular, *generic* queries, i.e., queries that are invariant under certain transformations of the ambient space in which the semi-linear sets reside, are considered [2]. There are also a number of collapse results between different semantics of $\text{FO} + \text{lin}$ (notably from natural semantics in which variables range over all real numbers, to active semantics, in which variables range over the active domain of the database). It was shown by Paredaens, Van den Bussche and Van Gucht that these semantics coincide [13].

KEY APPLICATIONS

Polynomial and linear constraint databases have their main applications in spatial databases and geographic information systems (GIS) [16]. In the vector model for GIS, data is often modeled by points, polylines and polygons. Points may represent objects, like buildings, trees or even cities (depending on the level of detail used). Polylines are often used to model streets and rivers and polygons to model regions like cities, provinces and states. Clearly points, polylines and polygons are covered by the linear constraint model and this model therefore offers a lot of potential for applications in GIS. The constraint model has the advantage over other GIS database systems that it has well-defined logical query languages whose expressive power can be studied more precisely. It has the disadvantage that query evaluation depends on quantifier elimination, which has a too high complexity to be practically applicable.

The most important attempt to implement a GIS by means of linear constraints was undertaken at INRIA in Paris and resulted in the system DEDALE [6, 7, 8].

Linear constraint databases can also model applications that go beyond traditional GIS systems, in the sense that it can model time as an additional dimension. These applications typically require the representation of moving objects or spatiotemporal data. The PhD thesis of Sofie Haesevoets was devoted to the study of various constraint database models for spatio-temporal data [10].

FUTURE DIRECTIONS

Spatial constraint database systems that implement linear constraints have been developed. A prominent example of this is the system DEDALE system [6, 7, 8] that combines the use of a quantifier-elimination algorithm for the reals with linear constraints with techniques from computational geometry [5]. To achieve systems that are usable in the GIS practice, query evaluation algorithms will have to be improved. The high complexity of the quantifier-elimination algorithms, on which query evaluation relies, proves to be the bottleneck in developing usable linear constraint database systems.

RECOMMENDED READING

RECOMMENDED READING

- [1] F. Afrati, S. Cosmadakis, S. Grumbach, and G. Kuper. Linear versus polynomial constraints in database query languages. In A. Borning, editor, *Proceedings of the 2nd Workshop on Principles and Practice of Constraint Programming*, volume 874 of *Lecture Notes in Computer Science*, pages 181–192. Springer-Verlag, 1994.
- [2] M. Benedikt, G. Dong, L. Libkin, and L. Wong. Relational expressive power of constraint query languages. *Journal of the ACM*, 45(1):1–34, 1998.
- [3] M. Benedikt and H. J. Keisler. Definability over linear constraints. In P. Clote and H. Schwichtenberg, editors, *Proceedings of Computer Science Logic, 14th Annual Conference of the EACSL*, volume 1862 of *Lecture Notes in Computer Science*, pages 217–231. Springer-Verlag, 2000.
- [4] J. Bochnak, M. Coste, and M.-F. Roy. *Géométrie algébrique réelle*. Springer-Verlag, 1987.
- [5] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, 2000.
- [6] S. Grumbach. Implementing linear constraint databases. In V. Gaede, A. Brodsky, O. Günther, D. Srivastava, V. Vianu, and M. Wallace, editors, *Proceedings of the 2nd Workshop on Constraint Databases and Applications*.
- [7] S. Grumbach, P. Rigaux, M. Scholl, and L. Segoufin. DEDALE, a spatial constraint database. In S. Cluet and R. Hull, editors, *Proceedings of the 6th International Workshop on Database Programming Languages (DBPL)*, volume 1369 of *Lecture Notes in Computer Science*, pages 38–59, 1998.
- [8] S. Grumbach, P. Rigaux, and L. Segoufin. The DEDALE system for complex spatial queries. In *Proceedings of the 23th ACM International Conference on Management of Data (SIGMOD)*, pages 213–224. ACM Press, 1998.
- [9] S. Grumbach, J. Su, and C. Tollu. Linear constraint query languages: expressive power and complexity. In D. Leivant, editor, *Logic and Computational Complexity*, volume 960 of *Lecture Notes in Computer Science*, pages 426–446. Springer-Verlag, 1995.
- [10] S. Haesevoets. *Modelling and Querying Spatio-temporal Data*. 2005. PhD thesis, Limburgs Universitair Centrum.
- [11] P. C. Kanellakis, G. Kuper, and P. Z. Revesz. Constraint query languages. *Journal of Computer and System Sciences*, 51:26–52, 1995.

- [12] G. M. Kuper, L. Libkin, and J. Paredaens, editors. *Constraint Databases*. Springer-Verlag, 2000.
- [13] J. Paredaens, J. Van den Bussche, and D. Van Gucht. First-order queries on finite structures over the reals. *SIAM Journal on Computing*, 27(6):1747–1763, 1998.
- [14] 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*, pages 279–288, 1994.
- [15] R. Z. Revesz. *Introduction to Constraint Databases*. Springer-Verlag, 2002.
- [16] Ph. Rigaux, M. Scholl, and A. Voisard. *Introduction to Spatial Databases: Applications to GIS*. Morgan Kaufmann, 2000.
- [17] A. Tarski. *A Decision Method for Elementary Algebra and Geometry*. University of California Press, 1948.
- [18] L. Vandeurzen. *Logic-Based Query Languages for the Linear Constraint Database Model*. 1999. PhD thesis, Limburgs Universitair Centrum.