Constraint databases, geometric elimination and geographic information systems

Bernd Bank, Humboldt University
Max Egenhofer, University of Maine
Joos Heintz, University of Buenos Aires and Cantabria
Bart Kuijpers, Hasselt University
Peter Revesz, University of Nebraska

Dagstuhl Seminar 07212

1 Description and goals of the seminar

During the past 15 years the topic of constraint databases (CDB) [2, 3] has evolved into a mature area of computer science with sound mathematical foundations and with a profound theoretical understanding of the expressive power of a variety of query languages. Constraint databases are especially suited for applications in which possibly infinite sets of continuous data, which have a geometric interpretation, have to be stored in a computer. Today, the most important application domains of constraint databases are geographic information systems (GIS), spatial databases and spatio-temporal databases [2, 3, 4]. In these applications infinite geometrical sets of continuous data are finitely represented by means of finite combinations of polynomial equality and inequality constraints that describe these data sets (in mathematical terms these geometrical data sets are known as semi-algebraic sets and they have been extensively studied in real algebraic geometry). On the other hand, constraint databases provide us with a new view of classic (linear and nonlinear) optimization theory.

A variety of languages, mostly extensions of first-order logic over the reals, has been proposed and studied for querying constraint databases in various applications. The expressive power of these query languages has been analyzed in many aspects, especially with applications in GIS and spatial databases in mind. On the other hand, beyond of general complexity results of real algebraic geometry, very few is known about the specific complexity of query evaluation in constraint database systems. Consequently the propagation of theoretical research results into database practice is hindered by the inefficiency of general purpose algorithms from real algebraic geometry used up to now for the implementation of query evaluation. These implementations are mostly based on quantifier-elimination and only query languages for linear constraint databases have been implemented in practice. The need for efficient algorithms is most visible for the basic query language FO, first-order logic over the reals. Also extensions of FO with for instance the "sum (of a finite set)", "topological connectivity","path connectivity" or other topological operators have received much attention in recent years and are considered to be of importance for practical applications, specifically in GIS. Both for FO and for these extensions, query evaluation was implemented in the past through the standard general purpose algorithms from real algebraic geometry. The sequential time complexity of

Dagstuhl Seminar Proceedings 07212
Constraint Databases, Geometric Elimination and Geographic Information Systems
http://drops.dagstuhl.de/opus/volltexte/2007/1282

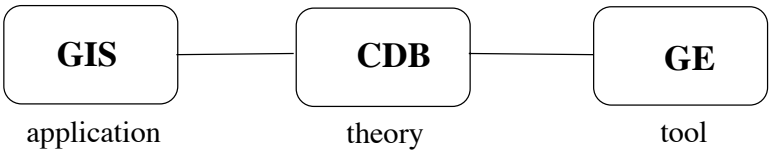
these algorithms depends intrinsically (and in worst case exponentially) on the arrangement size of the data and (superexponentially) on the number of quantifier alternations of the query under consideration. On the other hand this complexity is polynomial for fixed arrangement size of the data and fixed number of quantifier alternations of the query.

From the above it should be clear that researchers from the areas of constraint databases, geometric elimination (GE) algorithms and geographic information systems should work together to address the feasibility of the constraint database approach to deal with application needs in geographic information systems.

GIS researchers find in the constraint database model a powerful and elegant tool for application in spatial databases and GIS. Its clean mathematical formulation allows the study of the expressive power of query languages in a much more rigorous way than by the ad-hoc approaches which are frequently followed in GIS. From the users side, GIS researchers can describe the requirements of applications and specify which fragments of the constraint database query languages are useful and needed in GIS practice.

Researchers in geometric elimination theory find a practical application par excellence of their algorithms in constraint databases. Efficient elimination algorithms form a bottleneck for the development of practical development of constraint database systems that could be commercially usable in GIS.

The goal of this seminar is to bring together researchers from the Areas of constraint databases, geometric elimination algorithms and geographic information systems to address the feasibility of the constraint database in the area of geographic information systems. This seminar also has the explicit purpose of setting up a joint conference (or series of conferences) on this topic.



In the following sections, the relationship between GIS and CDB on the one hand and between CDB and GE are discussed. Topics of research that are relevant to the goals of the seminar are listed.

2 The relationship between geographic information systems and constraint databases

Geographic Information Systems (GISs) capture geometric, temporal, and semantic information about spatial phenomena and offer methods for querying and analyzing spatial entities and their relationships, typically presenting results visually. The dichotomy of spatial phenomena—discrete spatial entities described as objects and distributed phenomena modeled as fields—has led to a large variety of spatial data models in GISs. Objects are typically expressed in the form of vector models, which approximate an object's shape as a point, line, or polygon, and capture explicitly the objects' topology. Field models reflect the sampling resolution—either as measures at selected points or as estimations over regular or irregular sampling areas.

The studies of constraint-based models for representing such spatial information have revealed some interesting alternative opportunities as constraints offer not only a method for

describing the objects' shapes, but also offer the seamless integration of temporal aspects to be expressed by the same model. As such, constraints provide a unified treatment of geometry and temporally varying information.

2.1 Spatial Constraints for Objects

Earliest efforts to represent spatial objects resorted to linear constraints to describe explicitly lines (e.g., boundaries of polygons or edges in networks), as well as polygons through intersections of half planes, and points through equalities. This conceptually clean approach does not require a need for specific spatial data types to address any special cases, such as polygons with holes or separations. Likewise, it models all geometric items-0-dimensional, 1-dimensional, 2-dimensional (and potentially 3-dimensional) objects-in a uniform way so that networks and polygons can be analyzed computationally through the same methods. Below an example is shown in which a polygon is expressed compactly by four constraints (i.e., four inequalities of half planes in a 2-dimensional Cartesian coordinate space).

$y \leq x + 10$
$y \geq 0$
$x \geq 0$
$y \leq 20$

The analytical geometric operations include set operations to determine intersections, differences, unions, and complements over linear constraints. These operations are then embedded into query languages and used in the implementation of SQL and Datalog. Compared to the traditional approaches to spatial modeling in GIS this unified approach has the potential of reducing the implementations of implementations of such analytical operations since only one implementation per operation is needed, rather than more complex and more involved implementations that address each special case of representation explicitly.

2.2 Spatio-Temporal Constraints for Objects

An elegant extension of the constraint-based approach to geometric representations covers time-varying spatial objects by adding temporal parameters into the constraints. For example, a habitat area of a species may change over time, either due to migration or because the animals are endangered by natural or human causes. Such change in the habitat's location can be expressed through spatio-temporal constraints modeled by the intersection of moving half planes.

$y \leq x + 10 + 2t$
$y \geq 0$
$x \geq 0$
$y \leq 20 + t$

In this example, two boundaries of the polygon are moving-the top boundary at twice the speed of the right boundary.

Such an extension allows for the modeling of moving objects as well as such deformations as expansions, contractions, and their combinations. Other changes modeled by spatio-temporal constraints include the splitting or merging of regions. Operation on spatio-temporal constraints include then query operators for overlapping moving areas (for instance the to analyze the co-location of prey and predator species) or areas moving over static backgrounds (e.g.,

the habitat's movement through a national park). Such unified spatio-temporal modeling has its greatest potential in concisely simulating complex scenarios and calibrating and verifying them iteratively as new space-time data become available. Realistic, domain-specific simulation models may require further specialized spatio-temporal operators, which in turn can be captured by constraints again:

- Models of forest-fires, for example, need a block operator to capture that firefighting, lakes, rivers, or roads may prevent the spread of the fire.
- Airport congestion models need such aggregate operators as maximum number of moving objects at any instance of time during a given time interval.
- Frequently movement in geographic space occurs within networks (roads, railways, rivers) or other constrained spaces. The combination of embedding spaces modeled as constraints with spatio-temporal phenomena, also modeled as constraints, could yield a further integration under the same unifying approach.

2.3 Spatio-Temporal Constraints for Fields

The same principles that apply to the constraint-based modeling of spatial objects hold for modeling distributed phenomena as spatio-temporal fields. For instance, the ozone level across an area can be expressed in terms of constraints. In this case, the 3-dimensional representation (x, y, and time) of a spatio-temporal phenomenon is broken into a tetrahedral mesh, within which any point's value can be interpolated with linear constraints. This yields the ozone concentration value at any x, y, t combination in the field. Operations on such fields include the search for those areas that have a value (e.g., ozone concentration) above a certain threshold, or the intersection of multiple fields, such as areas of high ozone concentration and high elevation. The uniform modeling of objects and fields in terms of constraints blurs some of the implementation differences inherent in traditional GISs. For instance, the intersection of a field with static of moving background polygons is straightforward as both spatial representations are expressed in the same uniform framework. The domain of spatio-temporal fields is particularly promising for the modeling and analysis in the emerging domain of geo-sensor networks.

2.4 Future Opportunities for Constraints and GIS

As with any database operations, efforts must be made to speed up query processing, in particular for large data sets. For this goal it is necessary to further develop indexes of constraints, in particular for moving objects.

Another extension relates to the occasional need for non-linear constraints to capture complex spatio-temporal analyses. The alibi query based on lifeline beads is such a setting. It tests whether two individuals, whose space-time information about their departures and arrivals, together with their maximum travel speeds, are known, could have met, and if so, for how long and in what area. While spatio-temporal constraints (in the form of four intersecting half-cones) are a natural way to express this model, the resulting constraints are non-linear. Other interesting scenarios with non-linear constraints include the modeling of satellite trajectories including forecasting about potential collisions, as well as the modeling of curved lines for roads such that more accurate speed assessments can be made in simulations.

Constraints may offer another bridge to GIS as computational methods for qualitative spatio-temporal information are gaining increasing attention. Such qualitative information does not rely on the detailed quantitative spatial and temporal representation in terms of detailed geometries, both rather focuses on the essence of the relationships between spatial objects and how these relationships change over time. Most popular within the GIS community has been the modeling of qualitative topological and direction relations and their analysis for consistency and inferences through constraint networks. Such qualitative reasoning often reflects more closely people's own inferences, and the underlying qualitative values come close to the semantics of the expressions people use when they interact through natural language. The constraint database approach may provide a useful framework to the querying and analysis of such qualitative spatial information, and with the integration of quantitative constraints, may offer new directions, for instance to augment qualitative with quantitative spatial reasoning, or to extend interaction modalities to address both verbal (qualitative) as well as visual (quantitative) spatial information.

In the same vein, hybrid systems are of interest, as they would allow the exploitation of static traditional geometric models with constraint-based temporal variations.

3 The relationship between constraint databases and elimination theory

In this section, we elaborate the relationship between constraint databases and elimination theory. As already observed in the Introduction, the development of a practical constraint database system, usable in applications such as GIS, requires efficient query evaluation of, at least, first-order queries through quantifier elimination. However, this does not mean that efficient evaluation of *any* first-order query on *any* constraint database is necessary for practical purposes. We shall turn back to this question later on.

3.1 Arbitrary sets versus fixed-degree sets

The difference between classical elimination theory and the constraint database approach can be highlighted by the fact that a given database schema may be interpreted by algebraic or semi-algebraic sets of arbitrary "degree" (whatever this mathematically means). A consequence of this circumstance is the appearing contradiction that connectivity is not first-order expressible (in the database sense), while the connected components of semi-algebraic sets are a finite number of "computable" semi-algebraic sets. The explanation of this contradictory phenomenon consists in the observation that the algorithm to compute connected components is only "uniform" for semi-algebraic sets of bounded degree.

Therefore, constraint database theory introduces a really new viewpoint and can not be simply reduced to elimination theory. In fact, constraint database theory can be used in order to specify certain algorithmic tasks of elimination theory.

Different from the case of standard bit-complexity theory, in continuous and scientific computing there does not exist a commonly accepted universal computation model. One possibility is to interpret, in the continuous context, the notion of algorithm as a sequence of constraint database queries. This leads to the question to which extent continuous computation can be subdivided into smaller queries and to the problem of modeling mathematically the

intuitive meaning of uniformity. With respect to the first point, linear algebra subroutines should be expressible as constraint database queries.

3.2 Relational versus more general signatures

Observe that in a given constraint database schema the arity of relations is fixed. Dynamic vectors of variables or relations are out of the scope of the theory. It is easy to exhibit examples which show that this limitation is unsatisfactory for the application of constraint database theory to other fields (e.g., quadratic optimization theory). Moreover, even standard matrix algebra and algorithms cannot be adequately represented in the constraint database formalism.

This shows the need for the inclusion of the concept of "dynamic vector" (of variables, relations and functions) into the query languages and schemes of constraint databases. If this task could be achieved we would be able to specify a large spectrum of elimination tasks which are closely related to practical applications and certainly we would also be able to give a mathematical meaning to the concept of a correctness proof of a (specified) algorithm. In other words, we would obtain a mathematical model for the intuitive meaning of asserted program in the continuous context.

3.3 Exact versus approximate modeling

Until now we used the expression elimination theory or algorithm in a standard way referring only to algebraic and semi-algebraic geometry (polynomial or linear equations and inequalities). For concrete applications of the intuitive meaning of constraint databases, this interpretation is far too restrictive.

In practical problem solving tasks, linear and polynomial equation and inequality systems which *exactly* model real-world situations are relatively rare. They may occur as mixed integer and continuous programming problems in chemical engineering, in applications of combinatorial optimization to, e.g., scheduling, etc. Unfortunately, in almost all of these cases, the number of (often spurious) solutions is extremely high and this implies also an extremely high complexity for even the most efficient presently-known elimination algorithms. This complexity aspect motivates generally a remodeling of the underlying real-world problem.

Much more frequently, polynomials occur as truncations of analytic functions, which themselves are defined as solutions of algebro-differential equations (ADE). However, in this case, exact solutions of the corresponding polynomial equation system need not to be related to approximative solutions of the underlying differential equation system. In other words, truncations of series may introduce a spurious qualitative behavior of the new model.

This circumstance produces the effect that some queries appears as "legitimated" and others not. Let us take as an example the connectivity query for semi-algebraic varieties, given by equations. Let us imagine two circles that touch in just one point. Such a curve has just one singularity and can be defined by the product of two circular equations. When we perturb this equation by an infinitesimal additive term, the corresponding curve become disconnected and the singularity disappears. Therefore, we have to take care when we apply the connectivity query to approximative models, it may be "legitimated" or not.

We draw two conclusions from these considerations. Firstly, we need a theory of legitimate databases and queries. Suppose now that we have achieved this goal. Then it is thinkable that the actual complexity problems which appear as a consequence of blind application of algorithmic elimination theory, may vanish since these problems occur only in pathological situations which are not represented by legitimate databases and queries.

The other conclusion is that we have to develop a broader understanding of the intuitive concept of constraint databases. From the point of view of practical applications, the limitation to polynomial equation and inequality systems is unrealistic and should be replaced by a larger definition of constraint databases, which includes, algebro-differential and difference equations as constraints. In cases where as we are inhibited to use approximations, we have to rely on (differential) elimination theory in order to treat (in an exact way) the functions which arise as solutions of the new constraints. In this sense, a broader meaning should be given to the words elimination theory and elimination algorithm.

3.4 Data structures

The complexity of elimination algorithms is intimately related to data structures and data type issues. A careful choice of data structures may dramatically improve the complexity of elimination algorithms based on traditional data structures and types. For example, the circuit encoding of polynomials leads to an exponential improvement of the worst-case complexity of traditional elimination algorithms based on sparse or dense representation of polynomials. Moreover, the circuit encoding of polynomials allowed for the first time the design of effective incremental elimination algorithms which distinguish between "well-posed" and "ill-posed" polynomial equation systems. These new elimination algorithms, which are not based on rewriting (Gröbner bases) or linear algebra techniques, are implemented in the software package "Kronecker" written by G. Lecerf, (CNRS, Versailles University, Paris) [1].

The optimal representation issue was never considered in the traditional literature of constraint databases. Generally, it is supposed that polynomials are given by the list of their coefficients and that constraints (i.e., semi-algebraic sets) are defined by formulas given in some suitable disjunctive form. This entails an unnecessary blowup of complexity. Therefore, it is unavoidable that the question of more efficient data structures and types for the representation of constraints and constraint databases should be addressed in the future.

References

[1] G. Lecerf. *Kronecker, Polynomial Equation System Solver.* http://www.math.uvsq.fr/~lecerf/software/kronecker/

[2] J. Paredaens, G. Kuper, and L. Libkin, editors. *Constraint databases.* Springer-Verlag, 2000.

[3] P. Revesz, *Introduction to Constraint Databases*, Springer 2002.

[4] P. Rigaux, M. Scholl, and A. Voisard. *Spatial Databases.* Morgan Kaufmann, 2002.