On expressing topological connectivity in spatial Datalog

Bart Kuijpers and Marc Smits

University of Antwerp[*]

Abstract. We consider two-dimensional spatial databases defined in terms of polynomial inequalities and investigate the expressibility of the topological connectivity query for these databases in spatial Datalog. In [**?**], a spatial Datalog program for piecewise linear connectivity was given and proved to correctly test the connectivity of linear spatial databases. In particular, the program was proved to terminate on these inputs. Here, we generalize this result and give a program that correctly tests connectivity of spatial databases definable by a quantifier-free formula in which at most quadratic polynomials appear. We also show that a further generalization of our approach to spatial databases that are only definable in terms of polynomials of higher degree is impossible. The class of spatial databases that can be defined by a quantifier-free formula in which at most quadratic polynomials appear is shown to be decidable. Finally, we give a number of possible other approaches to attack the problem of expressing the connectivity query for arbitrary two-dimensional spatial databases in spatial Datalog.

1   Introduction

Kanellakis, Kuper and Revesz introduced in [**?**] the framework of *constraint databases* which provides a rather general model for spatial databases [**?**]. In this context, a spatial database, although conceptually viewed as a possibly infinite set of points in the real space, is represented as a finite union of systems of polynomial equations and inequalities. In this paper, we are interested in two-dimensional spatial databases. The set of points in the real plane that are in the open unit square, but outside the open unit disk together with the point with coordinates (1,0) (see Figure 1) is an example of such a database and it could be represented as

$$\{(x,y) \mid (0 < x < 1 \wedge 0 < y < 1 \wedge \neg(x^2 + y^2 < 1)) \vee (x = 1 \wedge y = 0)\}.$$

Several languages to query spatial databases have been proposed and studied. A simple spatial query language is obtained by extending the relational calculus

---

[*] Address: UIA, Informatica, Universiteitsplein 1, B-2610 Antwerpen, Belgium. Email: {kuijpers, msmits}@uia.ua.ac.be.
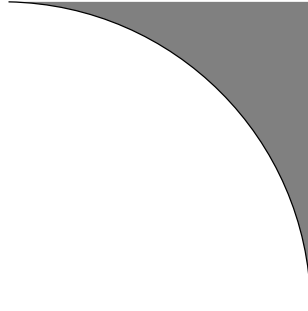
Fig. 1. A spatial database that can be represented by the formula $(0 < x < 1 \land 0 < y < 1 \land \neg(x^2 + y^2 < 1)) \lor (x = 1 \land y = 0)$.

with polynomial inequalities [**?**]. The query whether the database S is a straight line, for instance, can be expressed in this language by the sentence

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

Although variables in such expressions range over the real numbers, queries expressed in this calculus can still be effectively computed [**?**, **?**].

Recent results of Benedikt et al. [**?**] show, however, that the expressive power of this query language is limited. A combination of these results and results of Grumbach and Su [**?**] implies that one cannot express, for instance, in the calculus that the database is topologically connected. Topological connectivity is, however, a decidable property of spatial databases [**?**] and is of great importance in many spatial database applications.

In analogy with the classical graph connectivity query, which cannot be expressed in the standard relational calculus but which can be expressed in languages that typically contain a recursion mechanism (such as Datalog), *spatial Datalog* [**?**, **?**] can be considered as a suitable candidate language to express the topological connectivity query. Spatial Datalog is essentially Datalog augmented with polynomial inequalities in the bodies of rules.

In [**?**], we have studied termination properties of spatial Datalog programs and have made a first attempt to express the topological connectivity test in this language. Our implementation of this test consists of first computing a relation which contains all pairs of points of the spatial database which can be connected by a straight line segment that is completely contained in the database and by then computing the transitive closure of this relation. If the computation of the transitive closure ends and consists of all possible pairs of points of the database, the program returns *true*. In fact, this program tests for what is usually referred to as *piecewise linear connectivity*, which is a stronger condition than connectivity. The program, however, cannot be guaranteed to terminate on all spatial database inputs, not even on bounded ones. In [**?**], we

have characterized the class of spatial databases for which the program *correctly* tests piecewise linear connectivity. We have proved, in particular, that for this class of databases the program terminates. Among the databases for which the program works correctly are the *linear spatial databases*, i.e., the databases that can be defined using only linear polynomial inequalities. Furthermore, for linear databases connectivity and piecewise linear connectivity coincide. One reason why this program does not work correctly on all spatial databases is obviously due to the fact that one cannot, in general, connect two points on a curved line in the database by straight line segments. Another reason is due to the presence of "cusp-like" points on the border of databases. This is illustrated by the database in Figure 1. Indeed, since the tangent to the circle $x^2 + y^2 = 1$ in the right bottom point coincides with the vertical border line of the database, any straight line segment from an interior point of the database to the right bottom point will leave the database. Therefore, no finite number of line segments will suffice to connect the right bottom point with an arbitrary interior point.

In this paper, we generalize the ideas of the piecewise linear connectivity program. The program we propose first computes a relation in which not only all pairs of points of the database are contained that can be connected by a straight line segment that is completely contained in the database, but also those pairs that can be connected by arbitrary connected segments of conic sections. This computation is possible in Datalog augmented with polynomial inequalities since connected segments of conic sections can be parameterized by rational functions. The corresponding computations in the program for piecewise linear connectivity [?] also depends on the the fact that straight line segments can be parameterized by rational (even linear) functions.

Then, the transitive closure of this relation is computed and the program returns *true* if all pairs of points of the database are in the computed relation.

In fact, this new program tests for what we could call *piecewise quadratic connectivity*, which is again a stronger condition than connectivity but weaker than piecewise linear connectivity. We show that this new program correctly tests piecewise quadratic connectivity for databases that can be defined by a quantifier-free formula in which at most quadratic polynomials occur. For such databases piecewise quadratic connectivity and connectivity coincide. The database of Figure 1 is an example of such a database. And indeed, for this database, any two points can be connected by two conic section segments, since the right bottom point can be connected with any interior point of the database by following the lower borderline of the database untill the point vertically under the interior point is reached and by then vertically connecting to the interior point with a straight line segment. We also show that the program for piecewise quadratic connectivity does not terminate on arbitrary spatial database inputs.

We show as well that the class of spatial databases that can be defined by quantifier-free formulas of degree at most two can be effectively characterized. We characterize this class of databases by a Boolean query expressed in the relational calculus with polynomial inequalities. This calculus is effective [?, ?].

Curves defined by a polynomial of degree three or higher cannot, in general,

be parameterized by rational functions. For degree three, the so-called *elliptic curves* illustrate this phenomena. Therefore, the methods used in our programs cannot be generalized in a straightforward manner so as to obtain a correct connectivity test for spatial databases described by polynomials of degree three or higher. The use of *parametric* representations for curve segments is crucial in the sense that it guarantees that one works with connected curve segments. Other algebraic ways to represent curves (like explicit polynomial definitions) do not guarantee that the given curve is connected and are therefore unreliable in this context.

The following sections are organized as follows. The definition of a spatial database and of spatial Datalog are given in Section 2. The piecewise quadratic connectivity program is given in Section 3 where also its correctness is proved. In Section 4, we show that the class of spatial databases that can be defined by a quantifier-free formula in which at most quadratic polynomials appear is a decidable class. In Section 5, we discuss a number of promising and possible approaches to attack the problem of expressing the topological connectivity query in spatial Datalog.

2 Definitions

Here, we consider a *spatial database* to be a geometrical figure in the real plane $\mathbf{R}^2$ ($\mathbf{R}$ stands for the set of real numbers) that can be effectively represented by means of polynomial inequalities of the form $p(x, y) \ \theta \ 0$, where $p(x, y)$ is a polynomial in the variables x and y with real algebraic coefficients and $\theta \in \{=, \neq, <, >, \leq, \geq\}$.

One such inequality defines the figure $\{(x, y) \mid p(x, y) \ \theta \ 0\}$. By using Boolean combinations (union, intersection, and complement) of such figures we can describe a rather general class of figures, which we will refer to as *spatial databases*.

Spatial databases in the plane are possibly infinite sets of points. Equivalently, they are binary relations over the infinite domain $\mathbf{R}$. The spatial database shown in Figure 1 in the Introduction is an example.

We say that a spatial database S *can be defined in terms of at most quadratic polynomials* if there exist polynomials $p_{ij}(x, y)$ of degree two or lower such that

$$S = \{(x, y) \mid \bigvee_{i=1}^{n} \bigwedge_{j=1}^{m_i} p_{ij}(x, y) \ \theta_{ij} \ 0\},$$

where $\theta_{ij} \in \{=, \neq, <, >, \leq, \geq\}$.

We assume familiarity with the language Datalog. In [**?**], we have turned Datalog into a spatial query language as follows:

- The underlying domain is the set of real numbers.
- The only EDB predicate is S, which is interpreted as the set of points in the spatial database, or equivalently, as a binary relation over $\mathbf{R}$.
- Relations can be infinite.

– Polynomial inequalities are allowed in rule bodies.

Under the bottom-up semantics, the following fundamental closure property is satisfied by any spatial Datalog program P [**?**]:

> *if the input relation S is a spatial database, then every derived relation R obtained by a finite number of iterations of P is also a spatial database; moreover, a finite representation of R can be effectively computed.*

In the following section, we will write programs in this kind of Datalog with polynomial inequalities and stratified negation. We will refer to this language simply as *spatial Datalog*.

As a simple example, the following program derives in R the set of all points that either lie in the database or below a point in the database:

$$R(x, y) \longleftarrow S(x, y)$$
$$R(x, y) \longleftarrow R(x, y'), \ y < y'.$$

This program always terminates after one iteration and hence, by the above, when applied to a spatial database S, will produce a set R that is also a spatial database. For example, if S is the set of Figure 1, then R will be the set

$$\{(x, y) \mid S(x, y) \vee (\exists y')(S(x, y') \wedge y < y')\},$$

which can be defined by

$$(0 < x < 1 \wedge y < 1) \vee (x = 1 \wedge y \leq 0).$$

The above equations also illustrate that the effective computation of a representation of the resulting spatial database involves the elimination of quantifiers; a classical theorem by Tarski [**?**] says that every logical description of a spatial database with quantifiers can also be described without quantifiers. Efficient algorithms for quantifier elimination over the reals are known since the work of Collins [**?**, **?**]. We also refer to a series of papers by Renegar [**?**] for an overview of recent progresses in this field.

3 Topological connectivity

A set S of points in the plane is defined to be *topologically connected* if it cannot be partitioned by two disjoint open sets. If S is a spatial database, S is topologically connected if and only if any pair of points in S can be linked by a curve that is lying entirely in S (and that can also be described by polynomial equations) [**?**].

It is not obvious how to implement the condition "two points are linked by an arbitrary curve" in spatial Datalog. If, however, we replace this condition by the stronger one "two points are linked by a piecewise linear curve" an implementation in spatial Datalog becomes straightforward (see [**?**]). The implementation for piecewise linear connectivity is depicted in Figure 2.

$$Obstructed(x, y, x', y') \longleftarrow \neg S(\bar{x}, \bar{y}), \ \bar{x} = a_1 t + b_1, \ \bar{y} = a_2 t + b_2,$$
$$0 \le t, t \le 1, b_1 = x, b_2 = y,$$
$$a_1 + b_1 = x', a_2 + b_2 = y'$$
$$Path(x, y, x', y') \longleftarrow \neg Obstructed(x, y, x', y')$$
$$Path(x, y, x', y') \longleftarrow Path(x, y, x'', y''), \ Path(x'', y'', x', y')$$
$$Disconnected \longleftarrow S(x, y), \ S(x', y'), \ \neg Path(x, y, x', y')$$
$$Connected \longleftarrow \neg Disconnected.$$

Fig. 2. A spatial Datalog program for piecewise linear connectivity.

The first pairs of points in S that are derived by the program in Figure 2 in the relation *Path* are those that are connected by one straight line segment lying entirely in S. Then the transitive closure of *Path* is computed. After n iterations of the transitive closure, *Path* contains all pairs of points in S that can be connected by a piecewise linear curve consisting of n line segments lying entirely in S.

To prove the correctness, on a class $\mathcal{SD}$ of spatial database inputs, of a program that computes the transitive closure of a relation in which to start with all pairs of points that can be connected by one curve segment out of a set of parameterized curves segments $\mathcal{C}$ are contained and that then checks if the obtained relation contains all possible pairs of points from the database (such as the program of Figure 2), we must establish two facts for every spatial database S in $\mathcal{SD}$:

- *Soundness*: Two points in S are in the same connected component of S if and only if they can be connected by a finite chain of curves segments from $\mathcal{C}$ lying entirely in S;
- *Termination*: The number of curve segments needed to connect any such pair of points in S is bounded.

Termination guarantees that the transitive closure will terminate. Soundness then establishes the correctness of the test for connectivity performed by the program after the transitive closure is completed.

Linear spatial databases always satisfy soundness and termination for the program of Figure 2, but arbitrary spatial databases do not [**?**]. For the spatial database of Figure 1 both the soundness and the termination condition are not satisfied. Soundness is not satisfied because it requires an infinite number of straight line segments to connect an arbitrary interior point of the database with its right bottom point. Therefore the program also loops forever on this input. The unbounded spatial database defined by $(y - x^2)(x^2 - y + 1/2) > 0$, shown in Figure 3, only violates the termination property. There is no upper bound on the number of line segments needed to connect any pair of points in this spatial database: for instance, the larger the y-coordinate of a point in the database, the more line segments are needed to connect it with the point $(0, 1/4)$.

Fig. 3. A database on which termination condition for the program of Figure 2 is not satisfied. The database consists of the points lying strictly between the parabola $y = x^2$ and the translated one $y = x^2 + 1/2$.

The spatial databases shown in Figures 1 and 3 are defined by polynomials of degree two. We now give, in Figure 4, a spatial Datalog program that correctly tests connectivity for databases defined by polynomials of degree at most two.

In addition to straight line segments, this program makes use of connected segments of conic sections to construct paths (i.e., to construct the *Path* relation). It is known [**?**] that conic sections can be parameterized by rational functions in t of the following form

$$\begin{cases} x(t) = (a_1 t^2 + b_1 t + c_1)/(at^2 + bt + c) \\ y(t) = (a_2 t^2 + b_2 t + c_2)/(at^2 + bt + c). \end{cases}$$

Connected segments of conic sections are therefore parameterized by rational functions of this form where t ranges over a closed interval in which $at^2 + bt + c$ does not become zero. A straight line segment is a special case of the above parameterization: take for instance $a_1 = a_2 = a = b = 0$.

In analogy with the program for piecewise linear connectivity the program of Figure 4 first derives in the *Path* relation those pairs of points that can be connected by one straight line segment or one connected segment of a conic section. Then the transitive closure of *Path* is computed. After n iterations the *Path* relation contains those pairs of points that can be connected by n or less line segments and segments of conic sections.

We are now ready for

Theorem 1. *The spatial Datalog program of Figure 4 correctly tests piecewise quadratic connectivity, and hence connectivity, of spatial databases that can be defined in terms of at most quadratic polynomials.*

Proof. As mentioned before, to prove correctness we have to prove soundness and termination.

$$Zero(a,b,c) \longleftarrow 0 \le t, t \le 1, at^2 + bt + c = 0$$
$$Obstructed(x,y,x',y',$$
$$a_1,b_1,c_1,a_2,b_2,c_2,a,b,c) \longleftarrow \neg S(\bar{x},\bar{y}),$$
$$(at^2 + bt + c)\bar{x} = a_1t^2 + b_1t + c_1,$$
$$(at^2 + bt + c)\bar{y} = a_2t^2 + b_2t + c_2,$$
$$0 \le t, t \le 1, cx = c_1, cy = c_2,$$
$$(a+b+c)x' = a_1 + b_1 + c_1,$$
$$(a+b+c)y' = a_2 + b_2 + c_2$$
$$Path(x,y,x',y') \longleftarrow \neg Zero(a,b,c),$$
$$\neg Obstructed(x,y,x',y',$$
$$a_1,b_1,c_1,a_2,b_2,c_2,a,b,c)$$
$$Path(x,y,x',y') \longleftarrow Path(x,y,x'',y''),\ Path(x'',y'',x',y')$$
$$Disconnected \longleftarrow S(x,y),\ S(x',y'),\ \neg Path(x,y,x',y')$$
$$Connected \longleftarrow \neg Disconnected.$$

Fig. 4. A spatial Datalog program for piecewise quadratic connectivity.

A. Soundness. The if-implication is trivial. So, we concentrate on the only-if implication. Let S be a spatial database that can be defined in terms of at most quadratic polynomials.

To establish the only-if implication we use Collins's Cylindrical Algebraic Decomposition (CAD) [?, ?]. Collins's CAD, when applied to a description of the spatial database S, shows the existence of a decomposition of S in a finite number of cells, where each cell is either a *point*, a 1-dimensional *curve* (without endpoints) or a 2-dimensional (open) *region*. Each cell in the decomposition is either entirely part of S or is entirely part of the complement of S. We remark that the curves in the decomposition are part of the solution set, i.e., the zero set, of at least one of the polynomials that appear in the description of S. Figure 5 gives some examples of possible cell shapes. In Figure 5 (a) the upper and lower borders of the region have no endpoints in common. In (b) and (c) they have one, resp. two endpoints in common. Figure 5 (d) shows an unbounded region.

In order to prove that any two points in the same connected component of S can be connected by a finite number of straight line segments or segments of conic sections, it clearly suffices to show that this is true for

1. any two points of S that are in one cell, in particular,
   a. two points of S that are in a same region,
   b. two points of S that are on a same curve, and
2. any two points of S that are in adjacent cells, in particular,
   a. a point that is in a region and a point that is on an adjacent curve,
   b. a point that is in a region and an adjacent point,
   c. a point that is on a curve and an adjacent point.

1.a. A region in a cell decomposition of S that is part of S is part of the interior of S. It is known that two interior points that belong to the same region can
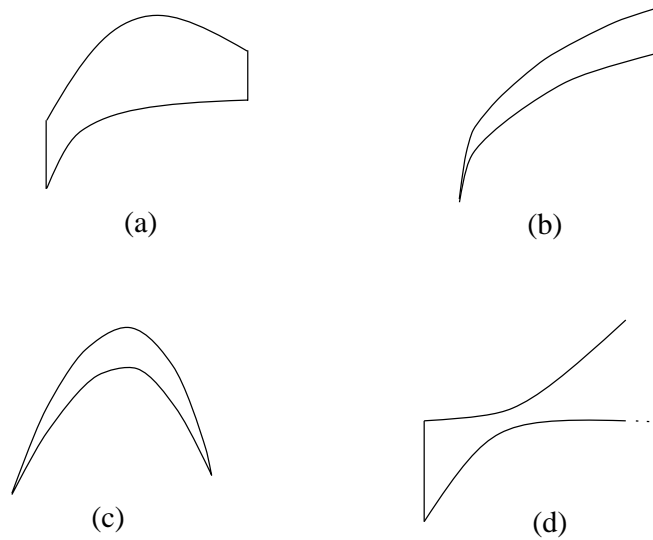
Fig. 5. Example regions and their bordering curves in Collins's CAD.

be connected by a curve, also defined in terms of polynomial inequalities, lying entirely in the the interior of S [**?**, **?**].

It is also known that such a uniformly continuous curve can be arbitrarily closely approximated by a piecewise linear curve with the same endpoints [**?**].

1.b. Let p and q be two points on a curve c in the cell decomposition of S. We have remarked that the curves in the decomposition are part of the solution set of at least one of the polynomials that appear in the description of S. Therefore c is part of an algebraic curve of degree at most two. So, c is a straight line segment or a segment of a conic section. Also the segment of c between p and q is therefore a line segment or a segment of a conic section.

More work is required to prove 2.

2.a. A point on a vertical border line of a region can be connected by one single horizontal line segment with a point in the adjacent region. This point can be connected by a finite number of line segments with an arbitrary point in the region (see 1.a). Also a point on an upper or lower border can be connected by one single vertical line segment in the adjacent region and therefore by a finite number of line segments with an arbitrary point in the region (again see 1.a).

2.b. This is the most difficult part of the proof. Here we have to distinguish between a number of cases (depending on the shape of the lower and upper border of regions), of which we will only consider two. All others are similar. These two cases are depicted in Figure 6. In both cases, the right bottom point p has to be connected with the point q in the interior of the region. From 1.a

it follows that it is sufficient to show that p can be connected with an arbitrary point r in the interior of the region. First we remark that (a) in Figure 6 is a special case of case (b). Indeed, if we reflect (a) along the diagonal $y = x$, the situation of (a) close to p is the same as the situation of (b) close to p if we take the curve c in (b) to be linear and horizontal.
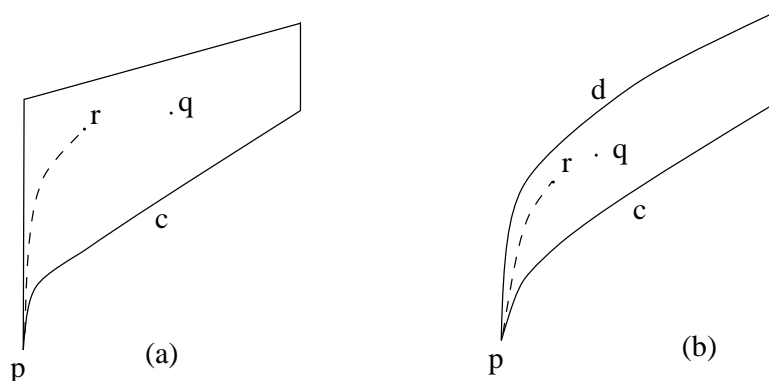
Fig. 6. The two cases in 2.b.

So, we consider the case of Figure 6 (b). As remarked before, the curves c and d are part of the solution sets of polynomials that appear in the description of the spatial database S. Suppose these polynomials are $P_c(x, y)$ and $P_d(x, y)$. As a first case, suppose that $P_c(x, y)$ and $P_d(x, y)$ are the same polynomial. Since c and d are different curves, this can only occur if the polynomial $P_c(x, y)$ (or $P_d(x, y)$) is the product of two linear polynomials (with coefficients in $\mathbf{R}$). This means that c and d are straight line segments that are part of a degenerate conic section. Any line through p with a slope between that of c and that of d will then enter the interior of the region.

Suppose on the other hand that $P_c(x, y)$ and $P_d(x, y)$ are different polynomials. Furthermore, we assume that the value of P_c is strictly positive in the region and on d. If it is not, we continue the discussion with $-P_c$. We also assume that the value of P_d is strictly positive in the region and on c.

Let I be the projection of the region on the x-axis. Let f_c and f_d be the functions from I to the real numbers whose graph is c, resp. d. We will show that there is a function f from I to $\mathbf{R}$ whose graph lies between c and d and are part of the solution set of the polynomial $P(x, y) = P_c(x, y) - P_d(x, y)$. Since $P(x, y)$ has degree at most two, the graph of f can be parameterized as a segment of a conic section or as a straight line segment.

We define the function f pointwise on the interval I. Hereto, consider the

functions

$$P_x : \mathbf{R} \to \mathbf{R}$$
$$y \mapsto P(x, y),$$

with $x \in I$. For a fixed $x_0 \in I$ we have that $P_{x_0}(f_c(x_0)) = -P_d(x_0, f_c(x_0)) < 0$ and $P_{x_0}(f_d(x_0)) = P_c(x_0, f_d(x_0)) > 0$. This means that the function P_{x_0} changes its sign in the open interval $(f_c(x_0), f_d(x_0))$. Since P_{x_0} is a continuous function it must therefore have at least one zero in the interval $(f_c(x_0), f_d(x_0))$. On the other hand, it can have at most two zeros since P_{x_0} is defined in terms of a polynomial of degree at most two. P_{x_0} cannot have two zeros in this interval since it changes sign. Therefore P_{x_0} has exactly one zero in the interval $(f_c(x_0), f_d(x_0))$. We set the value of $f(x_0)$ to the value of this unique zero. It is obvious that the graph of f is part of the solution set of $P(x, y)$.

We remark that the proof also works for arbitrary regions. For regions where the upper and lower border have a point in common it implies that there exists an at most quadratic curve from such a common endpoint into the interior of the region. It does not depend on the fact that the upper and lower border of the region have an endpoint in common.

Finally, we note that for an unbounded region the above argumentation is also valid.

2.c. A point on a curve and an adjacent point can be connected by the curve segment between them. This is again a segment of a straight line or of a conic section.

B. Termination. Since Collins's CAD produces a decomposition with a finite number of cells, it suffices to show for each cell c that there exists an upper bound $\alpha(c)$ on the number of line segments or conic section segments needed in a piecewise path between any two points in c. This is trivial for those cells that are vertical line segments or single points. 1.b of the soundness proof shows that it is also trivial for curves. So, if c is a point or a curve, we can take $\alpha(c)$ to be one. This leaves us with the regions. Let c be a region. In 2.b of the soundness proof we have shown that there is an at most quadratic curve γ_c that transverses the complete region. Therefore we can connect any two points p and q of c by at most three straight line segments or segments of conic sections by first connecting p vertically with a point on γ_c that has the same x-coordinate as p, by then following γ_c till the x-coordinate of q is reached and by reaching q with a vertical straight line segment. This is illustrated for a bounded region in Figure 7. So, if c is a region, we can take $\alpha(c)$ to be three. $\qquad \square$

The program for piecewise quadratic connectivity of Figure 4 does not correctly test connectivity on arbitrary inputs. Consider the spatial database $S = \{(x, y) \mid x^3 + y^3 = 1\}$. This database is actually an algebraic curve. Furthermore, it is an elliptic curve and has genus one [**?**]. It is known that such curves cannot be parameterized by rational functions. Although S is connected the program for piecewise quadratic connectivity will return *false*. To start it will put all tuples (x, y, x, y) with $(x, y) \in S$ in the relation *Path* and will then leave the transitive
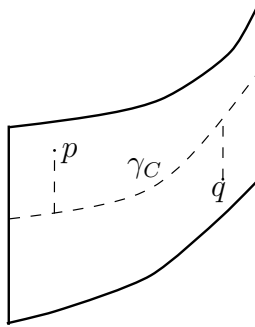
Fig. 7. Connecting the points p and q with three curve segments.

closure calculation after one iteration and conclude that not all pairs of points of S have been found.

The fact that not all algebraic curves of degree three or higher (like $x^3 + y^3 = 1$) can be parameterized makes it impossible to further generalize the approach of the piecewise linear and quadratic connectivity programs to higher degrees. The use of other algebraic representations of curves like implicit definitions are unreliable to test connectivity. First of all, not all implicitly defined curves are connected. For instance, the hyperbola $xy - 1 = 0$ is not connected. Secondly, the presence of a parameter in parameterized curve segments allows one to speak of "a point p being *between* points q and r on a curve". With implicit definitions this notion of betweenness is not obvious.

4 Decidability results

In this section we show that it is decidable whether a spatial database is definable in terms of at most quadratic polynomials. We prove this by expressing "definability in terms of at most quadratic polynomials" by a Boolean query in the relational calculus with polynomial inequalities. From [**?**] it then follows that this property is decidable.

We remark that this result also holds if we replace "at most quadratic" by "of degree at most k" for any natural number k. Since, for our purposes, this decidability result is only of interest for the spatial databases mentioned in Theorem 1 and since we don't want to complicate the sketch of the proof, we concentrate in the following on degree two. The remainder of this section is devoted to proving

Theorem 2. *It is decidable whether a spatial database can be defined in terms of at most quadratic polynomials.*

Proof. Let S be a spatial database. We define $Q(S)$ to be the set of points with coordinates (x, y) that satisfy the following expression in the relational calculus augmented with polynomial inequalities:

$$(\exists\varepsilon)(\exists a_{00})(\exists a_{01})(\exists a_{02})(\exists a_{10})(\exists a_{11})(\exists a_{20})(\varepsilon > 0 \wedge$$
$$((((\forall u)(\forall v)((x-u)^2 + (y-v)^2 < \varepsilon^2 \rightarrow (S(u,v) \leftrightarrow \textstyle\sum_{0 \leq i+j \leq 2} a_{ij} u^i v^j = 0)))$$
$$\vee((\forall u)(\forall v)((x-u)^2 + (y-v)^2 < \varepsilon^2 \rightarrow (S(u,v) \leftrightarrow \textstyle\sum_{0 \leq i+j \leq 2} a_{ij} u^i v^j \neq 0)))$$
$$\vee((\forall u)(\forall v)((x-u)^2 + (y-v)^2 < \varepsilon^2 \rightarrow (S(u,v) \leftrightarrow \textstyle\sum_{0 \leq i+j \leq 2} a_{ij} u^i v^j \leq 0)))$$
$$\vee((\forall u)(\forall v)((x-u)^2 + (y-v)^2 < \varepsilon^2 \rightarrow (S(u,v) \leftrightarrow \textstyle\sum_{0 \leq i+j \leq 2} a_{ij} u^i v^j > 0))))).$$

$Q(S)$ is therefore also a spatial database. $Q(S)$ is the set of all points in $\mathbf{R}^2$ where locally containment in S can be described by one single at most quadratic polynomial.

We will show that the following are equivalent for any spatial database S:

(1) S can be defined in terms of at most quadratic polynomials;
(2) $\mathbf{R}^2 \setminus Q(S)$ is finite.

This will be sufficient to conclude the proof since the finiteness of a spatial database can be expressed by a sentence in the relational calculus with polynomial inequalities(see [**?**]) and since this calculus is effective.

(1) implies (2): Let

$$S = \{(x, y) \mid \bigvee_{i=1}^{n} \bigwedge_{j=1}^{m_i} p_{ij}(x, y) \; \theta_{ij} \; 0\},$$

with $\theta_{ij} \in \{=, \neq, <, >, \leq, \geq\}$ and $p_{ij}(x, y)$ polynomials of degree at most two. We can assume that the polynomials $p_{ij}(x, y)$ are irreducible over $\mathbf{R}$, i.e., not the product of two non-constant polynomials with coefficients in $\mathbf{R}$. We also assume them to be non-zero. Irreducibility can be assumed since we can replace $p(x, y) \cdot q(x, y) = 0$ by $p(x, y) = 0 \vee q(x, y) = 0$, $p(x, y) \cdot q(x, y) > 0$ by $(p(x, y) > 0 \wedge q(x, y) > 0) \vee (p(x, y) < 0 \wedge q(x, y) < 0)$, etc.

We will show that the set of points with coordinates (x, y) for which the cardinality of the set $\{p_{ij} | p_{ij}(x, y) = 0\}$ is zero or one, is a subset of $Q(S)$.[2]

Suppose that (x_0, y_0) are the coordinates of a point where no p_{ij} becomes zero. Since the polynomials p_{ij} are continuous functions from $\mathbf{R}^2$ to $\mathbf{R}$ there exists an open disk D_ε with center (x_0, y_0) and radius ε on which all the polynomials p_{ij} do *not* change sign, i.e., remain strictly positive or strictly negative. Therefore, the formula $\bigvee_{i=1}^{n} \bigwedge_{j=1}^{m_i} p_{ij}(x, y) \; \theta_{ij} \; 0$ will evaluate to the same truth value for all points in D_ε. Therefore, D_ε is part of S or of $\mathbf{R}^2 \setminus S$. In the former case, there exists a polynomial of degree at most two (namely the one with $a_{00} = a_{01} = a_{02} = a_{10} = a_{11} = a_{20} = 0$) such that all points

---

[2] If a polynomial occurs more than once in the description of S, i.e., if some $p_{ij}(x, y) = p_{i'j'}(x, y)$ for $(i, j) \neq (i', j')$, than it is counted only once in the computation of the cardinality.

$(u, v) \in D_\varepsilon$ satisfy the equation $\sum_{0 \leq i+j \leq 2} a_{ij} u^i v^j = 0$. Therefore, (x_0, y_0) belongs to $Q(S)$. In the latter case, the equation $\sum_{0 \leq i+j \leq 2} a_{ij} u^i v^j \neq 0$ defines S in D_ε if $a_{00} = a_{01} = a_{02} = a_{10} = a_{11} = a_{20} = 0$.

Suppose that (x_0, y_0) are the coordinates of a point where exactly one of the polynomials, say p_{ij}, becomes zero. Again, there exists a small enough ε-environment D_ε of (x_0, y_0) where all the other polynomials do not change sign. The above formula that defines S therefore evaluates on D_ε to a formula which is only depending on the sign of p_{ij}. Therefore, for $(u, v) \in D_\varepsilon$, S is determined by $p_{ij}(u, v) \; \theta \; 0$ for some θ. We take $a_{00}, a_{01}, a_{02}, a_{10}, a_{11}$ and a_{20} such that $p_{ij}(u, v)$ equals $\sum_{0 \leq i+j \leq 2} a_{ij} u^i v^j$ if $\theta \in \{=, \neq, \leq, >\}$ and such that $-p_{ij}(u, v)$ equals $\sum_{0 \leq i+j \leq 2} a_{ij} u^i v^j$ if $\theta \in \{\geq, <\}$. This shows that $(x_0, y_0) \in Q(S)$.

So, we have shown that $\mathbf{R}^2 \setminus Q(S)$ is a subset of the set of points with coordinates (x, y) for which the cardinality of the set $\{p_{ij} | p_{ij}(x, y) = 0\}$ is greater or equal to two. We will now show that the latter set is finite. The finiteness of $\mathbf{R}^2 \setminus Q(S)$ then follows immediately.

We claim that two different polynomials p_{ij} and $p_{i'j'}$ have at most four common solutions. This claim indeed implies that the set of points with coordinates (x, y) for which the cardinality of the set $\{p_{ij} | p_{ij}(x, y) = 0\}$ is greater or equal to two is finite.

We now prove the claim. If at least one of them is irreducible over $\mathbf{C}$ ($\mathbf{C}$ stands for the set of the complex numbers) or if they are relative prime over $\mathbf{C}$, the claim follows immediately from Bezout's theorem (see for instance [**?**]). So, suppose they are reducible (over $\mathbf{C}$) and they have a complex linear factor in common, i.e.,

$$
\begin{aligned}
p_{ij}(x, y) &= q(x, y) \cdot q_1(x, y) \\
p_{i'j'}(x, y) &= q(x, y) \cdot q_2(x, y).
\end{aligned}
$$

Suppose p_{ij} and $p_{i'j'}$ have more than four real solutions in common. This implies that or q or q_1 has at least three real solutions. It can be easily verified that a linear polynomial with complex coefficients that has at least two real solutions is the product of a complex number and a linear polynomial with real coefficients. So, suppose q has at least three real zeros. Then $q = \alpha q'$ with α a complex number and q' a real linear polynomial. Therefore, $p_{ij} = q' \cdot (\alpha q_1)$, with q' and thus αq_1 real linear polynomials. This contradicts the assumption that p_{ij} is irreducible over $\mathbf{R}$. If we suppose that q_1 has at least three real zeros, we arrive at a contradiction in a similar manner.

This concludes the proof of the first implication.

(2) implies (1): Since S is a spatial database there exist polynomials p_{ij} such that S is defined by the formula $\bigvee_{i=1}^{n} \bigwedge_{j=1}^{m_i} p_{ij}(x, y) \; \theta_{ij} \; 0$. For the moment we have no information on the degrees of these polynomials. As in the proof of the other implication, we can assume that the polynomials $p_{ij}(x, y)$ are irreducible over $\mathbf{R}$ and non-zero.

Let $\mathcal{C}$ be a Cylindrical Algebraic Decomposition of $\mathbf{R}^2$ consisting of cells c on which each of the polynomials p_{ij} is sign-invariant. The existence of such

a CAD is guaranteed in [**?**]. The cells of $\mathcal{C}$ form *stacks* on a CAD $\mathcal{I}$ of $\mathbf{R}$. $\mathcal{I}$ consists of a finite number of points and open and connected intervals of $\mathbf{R}$. This is illustrated in Figure 8 for the spatial database defined by $(x^2 + (y-2)^2 \leq 1 \wedge y \neq 2) \vee (-1 \leq x \leq 1 \wedge y < 2)$, where we have $\mathcal{C} = \{c_1, \ldots, c_{15}\}$ and $\mathcal{I} = \{(-\infty, -1), \{-1\}, (-1, 1), \{1\}, (1, +\infty)\}$.

We now compress the CAD $\mathcal{C}$ to a partition $\mathcal{C}'$ of $\mathbf{R}^2$ by merging for each element of $\mathcal{I}$ the cells in its stack that are adjacent and are part of S, resp. of $\mathbf{R}^2 \setminus S$. In the example of Figure 8 this gives rise to the partition $\mathcal{C}' = \{c_1, c_2 \cup c_3, c_4, c_5, c_6 \cup c_7, c_8, c_9 \cup c_{10} \cup c_{11}, c_{12} \cup c_{13}, c_{14}, c_{15}\}$.
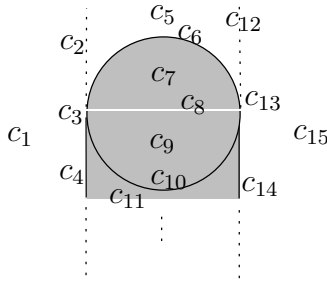
**Fig. 8.** A CAD of $(x^2 + (y-2)^2 \leq 1 \wedge y \neq 2) \vee (-1 \leq x \leq 1 \wedge y < 2)$.

We will show that every element of $\mathcal{C}'$ can be described in terms of at most quadratic polynomials. Since S is the finite union of elements of $\mathcal{C}'$, this will yield a description of S in terms of polynomials of degree at most two.

The elements of $\mathcal{C}'$ that belong to a stack that is build on a singleton element of $\mathcal{I}$ can be described by linear polynomials since they are points or line segments and since any singleton element of $\mathcal{I}$ has algebraic coordinates. The elements of $\mathcal{C}'$ that belong to a stack that is build on a open and connected interval $I \in \mathcal{I}$ can have different shapes. Some of these are shown, for bounded I, in Figure 9. In (a), the element of $\mathcal{C}'$ is the complete strip. In (b) and (c), the elements are downwards, resp. upward unbounded. In example (b) the top-border is part of the element of $\mathcal{C}'$, in (c) the bottom-border does not. In (d), the element of $\mathcal{C}'$ is bounded. (e) shows an element that is a curve. Other possible shapes of elements of $\mathcal{C}'$ are variations of (b),(c),(d) and (e) where the bordering curves have an asymptotic behavior with respect to the lines parallel to the y-axis and through the endpoints of I.

Clearly, the elements of $\mathcal{C}'$ of the form (a) in Figure 9, i.e., elements of the form $I \times \mathbf{R}$, can be described in terms of linear polynomials since the end points of I are guaranteed to be real algebraic (see [**?**]). We will prove that an element c' of $\mathcal{C}'$ of the form shown in Figure 9 (b) can be described in terms of polynomials of degree at most two. The other cases are proven analogously.
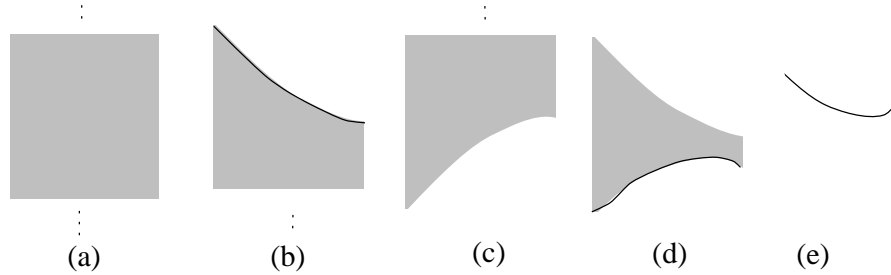
Fig. 9. Some shapes of elements of $\mathcal{C}'$ on a bounded open and connected interval.

Let the top-border of c' be called c. Remark that $c \in \mathcal{C}$. Since $\mathbf{R}^2 \setminus Q(S)$ is assumed to be finite, there must exist at least one point on c that belongs to $Q(S)$. Let (a, b) be the coordinates of such a point. There exists an ε-environment D_ε of (a, b) and an at most quadratic polynomial $p(x, y)$ such that S is described by $p(x, y) \leq 0$ in D_ε. This implies that $p(x, y)$ is zero on $D_\varepsilon \cap c$. Since c is part of the border of S, at least one of the polynomials $p_{ij}(x, y)$ is zero on c. So, $D_\varepsilon \cap c$ is an infinite set of common zeros of $p(x, y)$ and $p_{ij}(x, y)$. A reasoning similar to the one used in the proof of the first implication shows that the assumption that $p(x, y)$ and $p_{ij}(x, y)$ are different polynomials leads to a contradiction with the fact that $p_{ij}(x, y)$ is irreducible. Therefore complete c is a subset of the zero set of $p(x, y)$.

The algorithm **DEFINE**, that is part of Collins's CAD-algorithm (see page 154 of [**?**]) returns on input a polynomial p and a open and connected interval I, a series of formula's $\varphi_1, \ldots, \varphi_r$ build from p and its partial derivative $\partial p / \partial y$ and the logical connectives $\wedge, \vee, \neg$ that describe the cells in the stack on I formed by p. Since the element c' of $\mathcal{C}'$ is a finite union of cells on this stack, this shows that c' can be described in terms of at most quadratic polynomials.

This concludes the proof of the second implication and the proof. □

5 Discussion

In this section, we list a number of possible approaches to attack the problem of expressing the connectivity query for arbitrary spatial database in spatial Datalog.

Complete language: The topological connectivity is known to be an algorithmically decidable property of spatial databases [**?**]. In this respect it should be noted that extending the relational calculus with polynomial constraints and while-loops yields a computationally complete query language for spatial databases [**?**], and therefore topological connectivity is expressible in that language. We think it is possible to simulate while-loops using an inflationary version of

spatial Datalog. The implementation in this language of the topological connectivity query along the lines of [?] then includes the implementation of Collins's CAD. This seems a rather unnatural approach. So, the question remains if there is a natural way to express the topological connectivity query in spatial Datalog.

Blowing up curves in databases: In [?], the program of Figure 2 for piecewise linear connectivity was also used on non-linear databases. The obvious problem that straight line segments cannot be used to construct paths through non-linear curves was avoided by first "blowing-up" the curves of the database and by then using the piecewise linear connectivity program. Blowing up curves consists of placing disks that are small enough on every point of a curve. This is done in such a way that the connectivity is not affected. The resulting spatial database then no longer contains curves. Blowing up curves can be implemented in spatial Datalog (even without the use of recursion).

Even in the absence of curves the program for linear connectivity can not be guaranteed to work correctly. The problems that then still remain are caused by the presence of cusp-like points. These problems were mentioned in Section 3 and illustrated with the databases of Figures 1 and 3. We think that connected curve segments parameterized by functions of the form

$$\begin{cases} x(t) = \sum_{i=-N}^{N} a_i(t - \alpha_i)^i \\ y(t) = \sum_{i=-N}^{N} b_i(t - \beta_i)^i \end{cases}$$

can be used here to correctly test connectivity of blown-up databases that can be described by polynomials of degree at most $\sqrt{N}$.

Acknowledgment