Topological Queries in Spatial Databases

C.H. Papadimitriou* **D. Suciu†** **V. Vianu‡**

Abstract

We study *topological queries* over two-dimensional spatial databases. First, we show that the topological properties of semi-algebraic spatial regions can be completely specified using a classical finite structure, essentially the embedded planar graph of the region boundaries. This provides an *invariant* characterizing semi-algebraic regions up to homeomorphism. All topological queries on semi-algebraic regions can be answered by queries on the invariant whose complexity is polynomially related to the original. Also, we show that for the purpose of answering topological queries, semi-algebraic regions can always be represented simply as polygonal regions.

We then study query languages for topological properties of two-dimensional spatial databases, starting from the topological relationships between pairs of planar regions introduced by Egenhofer and Franzosa. We show that the closure of these relationships under appropriate logical operators yields languages which are *complete* for topological properties. This provides a theoretical *a posteriori* justification for the choice of these particular relationships. Unlike the point-based languages studied in previous work on constraint databases, our languages are region based – quantifiers range over regions in the plane. This yields a family of languages, whose complexity ranges from NC to undecidable. Another type of completeness result shows that the region-based language of complexity NC expresses precisely the same topological properties as well-known point-based languages.

1 Introduction

The manipulation of spatial data is an increasingly important part of database systems. Spatial data is involved in a wide range of applications: geographic information systems, video databases, medical imaging, CAD-CAM, VLSI, robotics, etc. While numerous models and languages for spatial data have been proposed, the field has only recently begun to acquire formal foundations. This paper is a contribution to the formal study of query languages for spatial databases; in particular, we focus on *topological* query languages.

Different applications of spatial databases pose different requirements on query languages. In many cases the precise size of the regions is important, while in other applications we may only be interested in the *topological* relationships between regions —intuitively, those that pertain to connectivity properties of the regions, and are therefore invariant under continuous mappings. Such differences in scope and emphasis are crucial, as they affect the data model, the query language, and performance. We can formalize the intuitive notion of "relevant information" with respect to

* U.C. Berkeley, christos@cs.berkeley.edu.
† AT&T Research, suciu@research.att.com: this work was done while the author was visiting U.C. San Diego.
‡ U.C. San Diego, vianu@cs.ucsd.edu. V.Vianu supported in part by the NSF under grant number IRI-9221268.

a certain class of queries by specifying a group G of permutations (bijections) of the space which can be applied without changing the answers to the queries of interest. Once G has been defined, then the queries of interest are simply those that are G-*generic* [Par+94] (i.e., whose answer does not change if the database undergoes any transformation in G; this concept extends the notion of genericity of Chandra and Harel for classical queries). For example, the topological queries are those that are generic with respect to *homeomorphisms* (continuous mappings with continuous inverses). A first group of results of this paper provides a key technical tool: we show that the topological properties of semi-algebraic spatial regions can be completely specified using a classical finite structure —intuitively, the embedded planar graph of the region boundaries. Such a structure acts as an *invariant* characterizing an equivalence class of sets of spatial regions with respect to homeomorphism. It can be viewed as an abstraction capturing exactly the topological properties of a set of regions. While abstractions of topological properties of this flavor have been considered before, this is the first time, to our knowledge, that such an invariant is shown to completely characterize inputs up to homeomorphism in the setting we consider (see also discussion below on related work).

We show that for inputs which are semi-algebraic regions, the invariant can be computed in polynomial time (and NC). Moreover, once this structure is computed, topological queries can be answered by classical database queries posed against that structure, of complexity polynomially related to the original query. This provides a bridge between the spatial and classical database domains. Furthermore, the invariants are used to show that each equivalence class of sets of semi-algebraic spatial regions with respect to homeomorphism has a representative where the regions are polygonal. This shows that, for the purpose of answering topological queries, semi-algebraic regions can always be represented simply as polygonal regions.

Alternatively, the invariants can be used as the basis for a spatial model capturing precisely the topological properties of regions. Indeed, the structures we produce contain information similar to the PLA model proposed by the U.S. Census Bureau, which contains topological properties on points, lines, and areas [Cor79, Par95]. Our invariants can be viewed as an augmentation of the PLA model.

The invariant problem is related, more broadly, to multimedia databases. Such databases have to manage a mix of classical database information and information of some special type (spatial, video, sound, etc). The relation between the two types of information is a fundamental problem in such systems. Some queries are best answered by processing the special information by specific means, while for others it might be sufficient to keep annotations about the spatial data in classical database form. The topological invariant can be viewed as an annotation to spatial data —a simple way of associating with each spatial database a relational "thematic" database—- that happens to be sufficient for answering all topological queries.

The second part of the paper focuses on query languages for spatial databases, and particularly on languages for expressing topological queries. For such query languages, one would like to have intuitive, natural primitives that are geared towards the information of interest: it is natural to talk about topological relationships among regions, but not about the distance between points. Given a query language geared towards topological queries, several questions come up: (i) is there a sense in which the language is complete with respect to this target class of queries? If not, does it represent some significant fragment? (ii) what are the appropriate representations of spatial information so that topological queries can be readily answered?

We consider such questions starting from a well-known set of natural language constructs proposed by Egenhofer and Franzosa for use in topological queries for geographic information systems. The constructs specify eight topological relationships among pairs of regions, based on the intersections of their topological interiors, boundaries, and exteriors. These mutually exclusive relations are: *overlaps*, *disjoint*, *equal*, *meets* (overlaps only at the boundary), *contains*, *covers* (contains, and also shares a boundary), and the inverses of the last two. For example, $overlaps(A, B)$ indicates that the interiors of the two regions have a nonempty intersection, and that the boundary of each region has a nonempty intersection with the other region's exterior. The eight relationships are complete in a fairly weak sense, namely that any two regions are in exactly one of those relationships to each other, and furthermore no finer relationships can be defined based only on the emptyness of the intersection of the interiors, exteriors, or boundaries of two regions.

The expressive power of the Egenhofer-Franzosa predicates has not been formally investigated. We point out that these relationships between pairs of regions are not sufficient to determine topological properties of a set of regions. For example, they cannot express the property of nonempty intersection of three regions. Even when we only have two regions, the Egenhofer-Franzosa predicates cannot express certain important topological properties of the regions, such as having a connected intersection. However, we show that under generous assumptions about the spatial regions, *the closure of the Egenhofer-Franzosa predicates under appropriate logical operators provides a complete language for all topological queries.* Of course, this language is noneffective, since it expresses noneffective topological queries. However, effective topological queries can be expressed in an effective way in the complete language.

A second kind of completeness result takes as a point of reference a natural first-order spatial logic and shows that a certain first-order closure of the Egenhofer-Franzosa predicates can express all topological queries definable in that language. This is related in spirit to what is done in temporal databases, where languages with temporal predicates are measured against temporal first-order logic that explicitly manipulates temporal variables. The first-order closure we use is effective on inputs which are semi-algebraic regions, and has data complexity NC. This stands in contrast to the complete language.

Previous formal work on query languages for spatial data considers logic languages in which the database is a collection of regions, and in which quantifiers range over real numbers and/or over points. By contrast, in our languages the quantifiers range over regions. They come in several variations, depending on the nature of the regions handled by the database, and on that of the quantified variables. We consider the question of decidability/complexity of these various languages, as well as their relative expressiveness.

Related work. Work in spatial databases has focused on developing models and query languages targeted to various application domains, as well as appropriate data structures and efficient evaluation techniques. We refer to [Par95] for a survey of the field emphasizing geographic information systems. Of particular interest are the topological relationships among regions proposed by Egenhofer and Franzosa in [EgFra]. Their model of topological relationships appears to have been widely adopted in geographic information systems. The satisfiability problem for Egenhofer-Franzosa relationships (essentially, the existential fragment of our language, applied on the empty database) is investigated in [GPP95]. The expressiveness of these relationships has not been investigated beyond the observation that they cover all possibilities that are expressible in the language that includes disjointness of two sets, interior, exterior, boundary, and Boolean connectives, and the argument that they are natural and cognitively plausible [EgFra].

The PLA model was proposed by the U.S. Census Bureau in [Cor79] (see also [Par95]). Its ability to capture topological information has not been formally studied. The complexity of computing topological information similar to the PLA model and to our invariants has been studied in computational geometry [BKR84, KY85]. Our complexity results make extensive use of these results. Closest to our topological invariants is a representation of topological information recently proposed in [KPV95], which is lossless with respect to isotopy-generic[1] information and applies to a spatial model different from ours. Query languages are not considered in [KPV95].

Various notions of G-genericity for different groups G of permutations are discussed in [Par+94]. They propose a spatial database model that includes spatial and classical database information, and propose a calculus and an equivalent algebra.

Much of the formal work related to spatial databases focuses on "constraint databases", consisting of relations whose tuples represent semi-algebraic regions, specified by polynomial inequalities. Such databases and corresponding query languages were first considered in [KKR90]. In particular, they investigate the question of when the answer of a query on a constraint database is representable as a constraint database. Their results are based on quantifier elimination in the first-order theory of the reals [Tar51]. Further work on expressiveness of query languages on constraint databases includes [GS94, GST94, Par+95, BDLW95].

Region-based logical formalisms date back to [Cla85], and have been intensively used in reasoning about spatial knowledge in AI [RC89, CRC94]. They use first-order logic to express topological relationships between regions starting from a single primitive *connect*. The main focus is on the adequacy of such formalisms to model domain specific knowledge. This is typically discussed using case studies. The expressive power of the languages is not formally investigated.

Logics over topological spaces are investigated in *topological model theory* (e.e., see [Zie85]). The underlying structure is a topological space and quantification is over open sets. Research in this area typically considers classical questions such as compactness, the Lowenheim-Skolem property, recursive axiomatizability, preservation and definability.

The paper is organized as follows. Section 2 introduces a simple spatial database model, several useful groups of permutations of $\mathbb{R}^2$, and a review of the Egenhofer-Franzosa relationships. The topological invariants are presented in Section 3. In Section 4 we define several first-order region-based languages starting from the Egenhofer-Franzosa relationships, and establish some facts on their relative expressive power. Completeness-style results on languages based on the Egenhofer-Franzosa relationships are provided in Section 5. Decidability and complexity results on the region based languages are provided in Section 6. We conclude in Section 7.

2 Basics

Practical spatial databases (such as geographic systems) mix spatial information with classical database information (sometimes referred to as *thematic*). Answers to queries can also be multi-sorted. Since our focus is on the spatial aspect, we adopt a simplified model where the only thematic information consists of region names. Also, for the sake of simplicity and uniformity, we only consider boolean queries, defining *properties* of sets of regions. We consider only regions in the two-dimensional space.

We will use the following model for spatial databases. We assume given an infinite set **Names**

[1]Intuitively, isotopies result from continuous deformations of the plane.

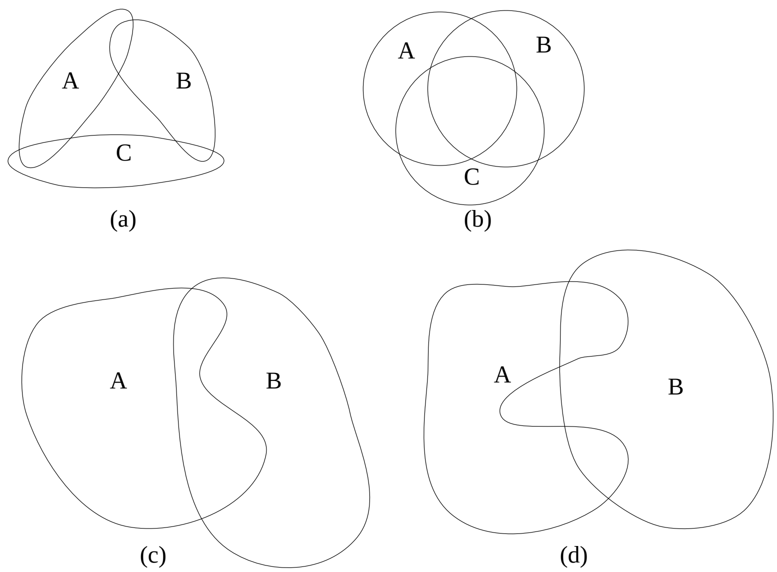


Figure 1: Four examples of spatial database instances.

(consisting of names of regions). All spatial databases use the same schema consisting of a unary relation *db*. An *instance* I consists of a finite subset of **Names** denoted $names(I)$, together with a mapping $ext(I, -)$ from $names(I)$ to subsets of $\mathbb{R}^2$. For each $r \in names(I)$, $ext(I,r)$ provides a set of points called the *extent of* r. We generally refer to a set of points in the plane as a *region*. In practice, each $ext(I,r)$ is finitely specified, although this may be transparent to the user. We use the notation $ext(r)$ whenever I is understood. Figure 1 gives four examples of database instances.

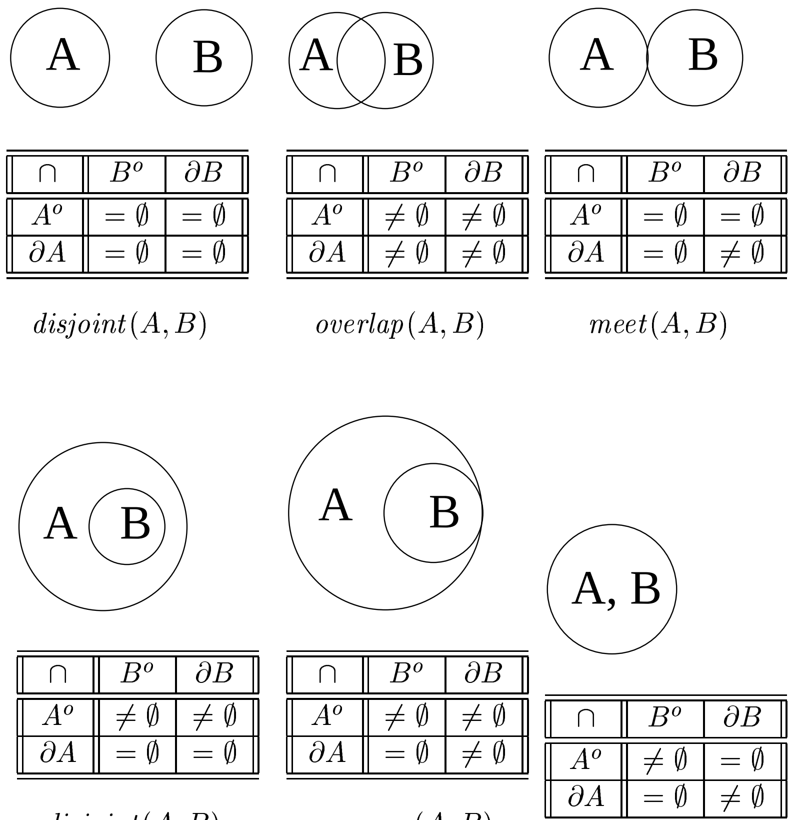
As discussed earlier, the kind of spatial information relevant to a particular domain can be formalized by specifying a group G of permutations (bijections) $\lambda : \mathbb{R}^2 \to \mathbb{R}^2$ that preserves that information. Our main interest will be in queries generic with respect to the group $\mathcal{H}$ of *homeomorphisms*, i.e. bijections $\lambda : \mathbb{R}^2 \to \mathbb{R}^2$ for which both λ and λ^{-1} are continuous. We will consider however other groups as well.

For some group G of permutations of $\mathbb{R}^2$ we say that two database instances I and J are G-equivalent iff $names(I) = names(J)$ and for some $\varphi \in G$, $\varphi(ext(I,r)) = ext(J,r)$ for each $r \in names(I)$. A property of instances is G-generic if it is closed under G-equivalence. When instances range over some restricted set of regions, the definition of G-genericity is relativized to that set of regions; that is, a G-generic property of such instances must be closed under G-equivalence among instances in that set of regions.

Note that G-equivalent instances have the same set of names. This factors out permutations of the names, which are not an essential aspect here. It also simplifies dealing with queries which mention region names explicitly, as their G-genericity is not affected.

Example 2.1 *Consider the property "$A \cap B$ has one connected component"; this property is $\mathcal{H}$-generic . The database instances of Figure 1 (a), (b), and (c) satisfy this property; (d) does not.*

The $\mathcal{H}$-generic properties are called *topological properties*. They are of particular importance in many domains. Egenhofer and Franzosa proposed eight binary topological relations among spatial regions as the basis for query languages for such application domains, including geographic



$\cap$	B^o	∂B
A^o	$= \emptyset$	$= \emptyset$
∂A	$= \emptyset$	$= \emptyset$

disjoint(A, B)

$\cap$	B^o	∂B
A^o	$\neq \emptyset$	$\neq \emptyset$
∂A	$\neq \emptyset$	$\neq \emptyset$

overlap(A, B)

$\cap$	B^o	∂B
A^o	$= \emptyset$	$= \emptyset$
∂A	$= \emptyset$	$\neq \emptyset$

meet(A, B)

$\cap$	B^o	∂B
A^o	$\neq \emptyset$	$\neq \emptyset$
∂A	$= \emptyset$	$= \emptyset$

disjoint(A, B)
inside(B, A)

$\cap$	B^o	∂B
A^o	$\neq \emptyset$	$\neq \emptyset$
∂A	$= \emptyset$	$\neq \emptyset$

cover(A, B)
covered-by(B, A)

$\cap$	B^o	∂B
A^o	$\neq \emptyset$	$= \emptyset$
∂A	$= \emptyset$	$\neq \emptyset$

equal(A, B)

Figure 2: The Egenhofer-Franzosa topological relationships.

information systems. These relations are obtained by classifying the intersection of the interior, boundary, and exterior of two spatial regions A and B: see Figure 2 for an illustration.

The Egenhofer-Franzosa relationships do not determine an instance up to $\mathcal{H}$-equivalence, and therefore do not provide sufficient information for checking all topological properties. More precisely, let us call two spatial instances I, I' *E-F-equivalent* iff $names(I) = names(I')$ and for every $p, q \in names(I)$, $ext(p)$ and $ext(q)$ stand in the same Egenhofer-Franzosa relationship in I as in I'. It is easy to see that there exist spatial instances I, I' which are E-F-equivalent but not topologically equivalent. Figure 1 contains two such examples: the instances in (a) and (b) are E-F-equivalent, but not $\mathcal{H}$-equivalent, and similarly for those in (c) and (d).

In addition to $\mathcal{H}$, we will consider two other permutation groups: *symmetries* and *piece-wise linear functions*. These are defined next. Call a function $\rho : \mathbb{R} \to \mathbb{R}$ *increasing* iff $x < x' \Longrightarrow \rho(x) < \rho(x')$, and *decreasing* iff $-\rho(x)$ is increasing; a function $\lambda : \mathbb{R}^2 \to \mathbb{R}^2$ is *linear* iff $\lambda(\langle x, y \rangle) = \langle ax + by + c, dx + ey + f \rangle$, for some $a, b, c, d, e, f \in \mathbb{Q}$.

- Symmetries: $\mathcal{S} \stackrel{\text{def}}{=} \{\lambda \mid \lambda(\langle x,y\rangle) = \langle \rho_1(x), \rho_2(y)\rangle\} \cup \{\lambda \mid \lambda(\langle x,y\rangle) = \langle \rho_1(y), \rho_2(x)\rangle\}$, with $\rho_1, \rho_2 : \mathbb{R} \to \mathbb{R}$ monotone bijections (i.e. each is either increasing or decreasing). Each such permutation maps horizontal lines to either horizontal or vertical lines, and similarly for vertical lines, but may map other lines into arbitrary curves.

- Piece-wise linear: $\mathcal{L}$ is the group generated by continuous 2-piece linear permutations of the form $\lambda(\langle x,y\rangle) = \mathit{if}\ x \leq x_1 \mathit{then}\ \lambda_1(\langle x,y\rangle)\ \mathit{else}\ \lambda_2(\langle x,y\rangle)$, where λ_1, λ_2 are linear mappings, and $x_1 \in \mathbb{Q}$. Note that λ is required to be continuous, and this implies that $\lambda_1(\langle x_1,y\rangle) = \lambda_2(\langle x_1,y\rangle)$, $\forall y \in \mathbb{R}$. Equivalently, λ is piece-wise linear if there exists a triangulation of $\mathbb{R}^2$ s.t. λ is linear on each triangle [SW94]. Note that such functions map any line into some polygonal line with infinite end segments.

Observe that $\mathcal{S}, \mathcal{L} \subset \mathcal{H}$ but $\mathcal{S}$ and $\mathcal{L}$ are incomparable.

We will consider throughout the paper the following types of regions. In all cases, a region will be an open, simply-connected, nonempty subset of $\mathbb{R}^2$, with a connected boundary.

- *Disc* consists of homeomorphic images of $D^2 \stackrel{\text{def}}{=} \{\langle x,y\rangle \mid x^2 + y^2 < 1\}$.

- *Alg* consists of all discs of the form $\{\langle x,y\rangle \mid \bigvee_i \bigwedge_j C_{ij}(x,y)\}$, where each condition $C_{ij}(x,y)$ is of the form $P(x,y) > 0$, for some polynomial P with integer coefficients. This definition is adapted from [KKR90]: equivalently, *Alg* consists of all discs whose boundaries are piece-wise algebraic curves.

- *Poly* are all *simple polygons* (i.e., polygons with non-intersecting boundary, specified by linear inequalities with integer coefficients).

- *Rect* $\stackrel{\text{def}}{=} \{\{\langle x,y\rangle \mid x_1 < x < x_2 \wedge y_1 < y < y_2\} \mid x_1 < x_2 \wedge y_1 < y_2\}$, where $x_1, x_2, y_1, y_2 \in \mathbb{R}$. We call these regions *rectangles*.

- *Rect** is the set of *discs* which are finite unions of *rectangles*.

See Figure 3 for some simple examples of regions in each class.

Unless otherwise stated, we will implicitly assume in the sequel that every class of regions *Region* is one of the above. Notice that *Rect* $\subset$ *Rect** $\subset$ *Disc* and *Poly* $\subset$ *Alg* $\subset$ *Disc*. Importantly, instances over *Poly* and *Alg* are finitely specifiable; so are instances over *Rect* and *Rect** when *Rect* consists of rectangles with corners with rational coordinates. Among the finitely specifiable types of regions *Alg* is the most general we consider, and is therefore of special importance in our investigation.

We say that a family of regions *Region* is *invariant* under some group of permutations G if $\forall r \in \mathit{Region}, \forall \lambda \in G \Longrightarrow \lambda(r) \in \mathit{Region}$. Figure 4 summarizes which class of regions is invariant under which of the groups described above.

3 The topological invariant

In this section we address the *invariant problem:* Can we extract from a spatial instance a finite structure which captures *exactly* the topological properties of the spatial instance? This is related to three important, overlapping problems:

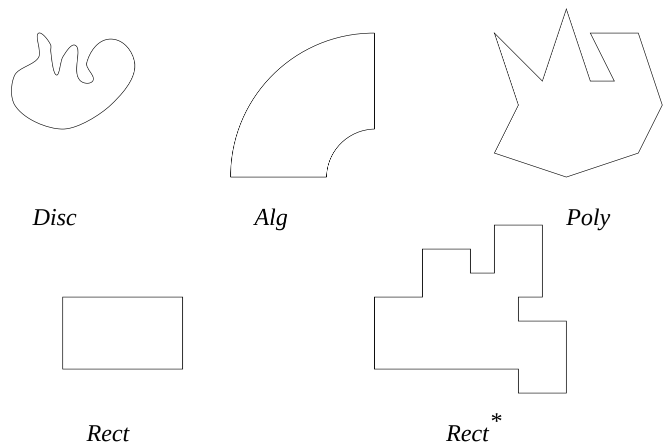


Figure 3: Examples of regions in $Disc, Alg, Poly, Rect$, and $Rect^*$.

Invariant ?	$Rect$	$Rect^*$	$Poly$	Alg	$Disc$
$\mathcal{S}$	Y	Y	N	N	Y
$\mathcal{L}$	N	N	Y	Y	Y
$\mathcal{H}$	N	N	N	N	Y

Figure 4: Which $Region$ is invariant under which group G.

- the *thematic problem:* When can domain-specific queries be answered by precomputed annotations stored in classical (relational, say) database form?
- the *spatial representation* problem: can spatial instances over some class of regions always be represented by topologically equivalent spatial instances over some simpler class of regions?
- the *topological model* problem: What is an appropriate data model for storing and retrieving topological information?

We address the invariant question for topological queries on two-dimensional semi-algebraic regions, i.e. database instances in Alg. For any semi-algebraic input, we construct in polynomial time (and in NC) a finite structure (relational instance) T_I, called topological invariant of I, over a fixed relational schema. Relational instance T_I summarizes *exactly* the spatial information needed to answer all topological queries. To our knowledge, this is the first time such explicit invariant is obtained for topological queries on semi-algebraic regions (the proof uses classical results from topology). A similar invariant is described in [KPV95], with two main differences: the invariant captures isotopy-generic rather than $\mathcal{H}$-generic information, and the spatial model is different from ours.

Topological queries on I can be answered by classical database queries against the invariant T_I, of complexity polynomially related to that of the original query; this answers the thematic problem. Furthermore, T_I can be used to construct, for each spatial instance I in Alg, a topologically equivalent spatial instance in $Poly$. This provides an elegant answer to the spatial representation problem: for the purpose of answering topological queries, semi-algebraic spatial regions can always be represented simply by polygonal regions. Lastly, the invariant can provide the basis for a topological model for spatial databases. In this scenario, only the information in the invariant would be kept, without an underlying spatial instance. In particular, direct updates to the invariant would have to be allowed, which brings up the question of checking whether the result of an update is in fact an invariant. We characterize structures which are invariants and show that this can be checked in NC.

Our construction of the invariant relies on results on *cell complexes* obtained in [KY85]. We start by recalling briefly the terminology and results of [KY85]. Then we return to our framework and exhibit the topological invariant.

Cell complexes [KY85] Given a set Σ of polynomials with 2 variables (x, y) and with rational coefficients, a *sign assignment* is a mapping $\sigma : \Sigma \to \{-1, 0, +1\}$, and the *sign class* of σ is $I^\sigma \stackrel{\text{def}}{=} \{\langle x, y\rangle \mid sign(p(x, y)) = \sigma(p), \forall p \in \Sigma\}$. The purpose in [KY85] is to describe, for a given Σ, the connected components of (the non-empty) I^σ, and their adjacency relationships. The starting point is the notion of a cell complex. A *cell complex* for Σ is a partition of $\mathbb{R}^2$ into finitely many, non-empty, pairwise disjoint regions, $\{R_v\}_{v\in V}$, called *cells*, such that:

1. each cell R_v is homeomorphic to $\mathbb{R}^{\delta(v)}$, for some $\delta(v) = 0, 1$, or 2; for each $v \in V$, $\delta(v)$ is called the *dimension* of v.
2. the closure of each cell R_v is the union of other cells;
3. each R_v is included in some sign class I^σ; we denote $\sigma = l(v)$.

To each cell complex we associate the graph $G = (V, E, \delta, l)$, where $(u, v) \in E$ iff R_u is contained in the closure of R_v.

It turns out that there are generally several possible cell complexes for a given set Σ of polynomials. To see this, note that in general the cells in the complex are not the maximal connected components of the I^σ's (as one might be tempted to believe): these can be obtained by collapsing adjacent cells, but the resulting sets are not necessarily a cell complex (they may violate (1)). To see such an example, consider $\Sigma = \{p\}$, $p(x, y) = xy$. Here $I^{+1} = \{\langle x, y\rangle \mid xy > 0\}, I^{-1} = \{\langle x, y\rangle \mid xy < 0\}, I^0 = \{\langle x, y\rangle \mid xy = 0\}$. There are five maximal connected components (two in I^{+1}, two in I^{-1}, and I^0 itself), but they do not form a cell complex, because I^0 violates condition (1) above (it is not homeomorphic to R^1). A cell complex for Σ would further partition I^0, e.g. into $\{\langle x, 0\rangle \mid x \in \mathbb{R}\}, \{\langle 0, y\rangle \mid y > 0\}, \{\langle 0, y\rangle \mid y < 0\}$. This is in some sense a maximal cell complex for Σ, but it is not unique, because $\{\langle 0, y\rangle \mid y \in \mathbb{R}\}, \{\langle x, 0\rangle \mid x > 0\}, \{\langle x, 0\rangle \mid x < 0\}$ is maximal too.

Kozen and Yap [KY85] describe an NC algorithm for finding a graph G representing *some* cell complex for Σ: Importantly, it follows from their result that the number of connected components of the non-empty I^σ's is bounded by a polynomial in the size of Σ.

The topological invariant Returning to our setting, suppose we are given an instance I in Alg. We call a *labeling* of I a function $\sigma : names(I) \to \{o, \partial, -\}$, and denote with 3^I the set of all labelings. By abuse of terminology, we call the *sign class* of σ the set $I^\sigma \stackrel{\text{def}}{=} \bigcap_{r\in names(I)} r^{\sigma(r)}$, where r^o is the interior of r ($r^o = r$, since all regions are open sets), r^∂ is the boundary, and r^- is the exterior. We define a *cell complex* for I as above, but replace condition (1) with: (1') each cell

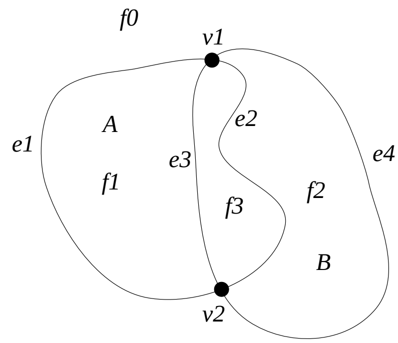


Figure 5: The graph G_I associated to an instance I.

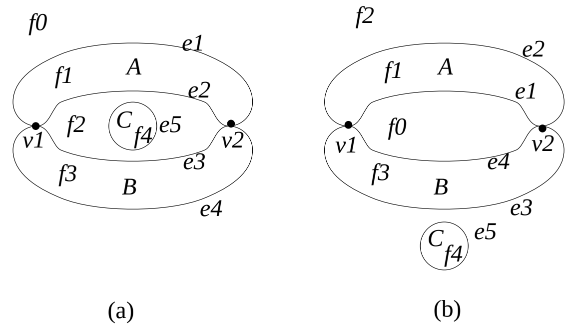


Figure 6: Except for the exterior cell, G_I and $G_{I'}$ are isomorphic.

R_v is homeomorphic to $\mathbb{R}^{\delta(v)}$, except for a single cell R_{f_0} which is homeomorphic to $\mathbb{R}^2 - \{\langle 0, 0\rangle\}$. We call f_0 the *exterior cell*. We can prove that, for any algebraic instance I, there exists a maximal cell complex $\{R_v\}_{v \in V}$ for I, and denote with $G_I \stackrel{\text{def}}{=} (V, E, \delta, f_0, l)$ its associated graph, where $l : V \to 3^I$. We call cells of dimension 0 *vertices*, cells of dimension 1 *edges*, and cells of dimension 2 *faces*.

Example 3.1 *Consider the instance I in Figure 1 (c). Then G_I consists of 2 vertices v_1, v_2, four edges e_1, e_2, e_3, e_4, and 4 faces, f_0, f_1, f_2, f_3, Figure 5. The adjacency relation E contains the edges*

$$\begin{array}{c}(v_1, e_1), (v_1, e_2), (v_1, e_3), (v_1, e_4),\\ (v_2, e_1), (v_2, e_2), (v_2, e_3), (v_2, e_4),\\ (e_1, f_0), (e_1, f_1), (e_2, f_2), (e_2, f_3),\\ (e_3, f_3), (e_3, f_1), (e_4, f_0), (e_4, f_2)\end{array}$$

The labeling is:

$$l(v_1) = l(v_2) = (A^\partial, B^\partial), l(e_1) = (A^\partial, B^-), l(e_2) = (A^\partial, B^o), l(f_3) = (A^o, B^o), etc.$$

To see the importance of specifying the exterior cell, consider the instances I, I' in Figure 6 (a) and (b). Both G_I and $G_{I'}$ have 2 vertices, 5 edges, and 5 faces. They coincide in their adjacency relation E and their labeling l, except that G_I's exterior cell is f_0, while $G_{I'}$'s exterior cell is f_2.

In summary, G_I provides in a concise manner the following information: for each region name in *names(I)*, the cells it contains (vertices, edges, faces); for each face, the edges on its boundary; for each edge, its endpoint(s); and lastly the unbounded face, f_0.

It is worth noting that all the information provided by G_I is useful and essential. For example, the dimension $\delta(v)$ of a cell $v \in V$ cannot be determined from the its labeling: when, say, $l(v) = (A^\partial, B^\partial)$, the dimension of v may be either 0 or 1. Also, the external face is not determined by the other information in G_I. This is shown by Example 3.1. Note in particular that the external face is not determined by its sign. Indeed, consider $\sigma = l(f_0)$. Clearly, $\sigma(r) = -, \forall r \in names(I)$. However, f_0 is not necessarily the unique such cell.

Before we proceed, we need the following terminology. Call a database instance I *connected* if $\bigcup_{r\in I} r^\partial$ is topologically connected. Alternatively, I is connected if the subgraph of G_I consisting only of vertices and edges (i.e. no faces) is connected. For example both database instances in Figure 7 (a) are non-connected. Recall that a closed curve in the plane is *simple*, if it is non self-intersecting[2]. Call a database instance I *simple* if the boundary of each face in G_I is a simple curve. For example, all four instances in Figure 1 are simple, while the four instances in Figure 7 are not. A simple instance is also connected, because else it is easy to show that the boundary of the exterior cell is not a closed curve: the instances in Figure 7 (a) illustrate this. The converse is not true: the two regions in Figure 7 (b) illustrate two connected instances which are not simple (the boundary of the external face is not simple).

G_I is almost what we need for an invariant. Specifically, we have:

Lemma 3.2 *Let I, I' be simple spatial database instances over Alg with $names(I) = names(I')$. Then I and I' are topologically equivalent iff G_I and $G_{I'}$ are isomorphic via an isomorphism which is the identity on $names(I)$. Moreover, any isomorphism between G_I and $G_{I'}$ can be lifted to a homeomorphism mapping I to I'.*

Proof Suppose first that I and I' are spatial instances over Alg such that $names(I) = names(I')$. It is clear that, if I and I' are topologically equivalent, then G_I and $G_{I'}$ are isomorphic by an isomorphism which is the identity on $names(I)$. Consider the converse. Let I, I' be simple spatial instances over Alg such that $names(I) = names(I')$, and suppose G_I and $G_{I'}$ are isomorphic via an isomorphism which is the identity on $names(I)$. We show that I and I' are topologically equivalent.

Let S be the union of all vertices and edges in the cell complex associated to I. We call S the *skeleton* of I: I is connected iff S is connected. Define S' similarly for I'. S consists of a finite set of points and edges, which are simple Jordan curves connecting two points. Since G_I and $G_{I'}$ are isomorphic, we can construct a homeomorphism $\lambda_0 : S \to S'$ (where both S and S' are topological subspaces of $\mathbb{R}^2$), which extends the isomorphism from G_I to $G_{I'}$, by patching the homeomorphisms between corresponding Jordan curves. In order to extend λ_0 to a homeomorphism $\mathbb{R}^2 \to \mathbb{R}^2$, we use Schönflies' Theorem ([Moi], pp. 72), stating that for any closed Jordan curve J, any homeomorphism $\lambda_0 : J \to \mathbb{R}^2$ can be extended to a homeomorphism $\lambda : \mathbb{R}^2 \to \mathbb{R}^2$. Namely, let R_f be a 2-dimensional cell in the cell decomposition associated to I, different from the exterior cell. Its boundary is a closed Jordan curve, and we use Schönflies' Theorem to extend λ_0 to R_f (we discard the exterior part of the extension). For the exterior cell R_{f_0} we proceed similarly, but keep

[2]Formally, a closed Jordan curve is a continuous function $\varphi : [0,1] \to \mathbb{R}^2$ such that $\varphi(0) = \varphi(1)$. It is *simple* iff $\varphi(x) = \varphi(y)$ with $x \neq y$ implies $\{x, y\} = \{0, 1\}$.

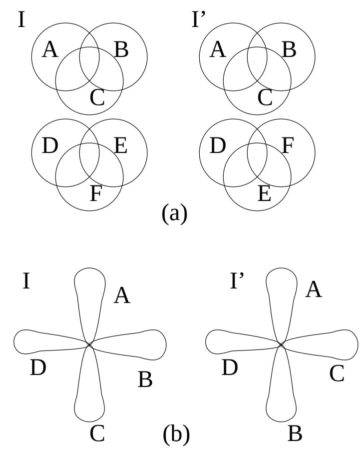


Figure 7: Two examples when $G_I, G_{I'}$ are isomorphic, but I, I' are not topologically equivalent.

only the exterior part and drop the interior. Patching together all λ's corresponding to different cells yields the desired homeomorphism $\lambda : \mathbb{R}^2 \to \mathbb{R}^2$. $\square$

Lemma 3.2 cannot be extended to non-simple instances. We illustrate this with the two examples in Figure 7 (a) and (b). In (a) we have two database instances I, I' which are not connected. Here G_I and $G_{I'}$ are isomorphic, but I, I' are not topologically equivalent (there is no homeomorphism mapping I to I'). Similarly for (b), where the two instances are connected, but not simple, because the boundary of the exterior cell is a non-simple curve, both in I and in I'.

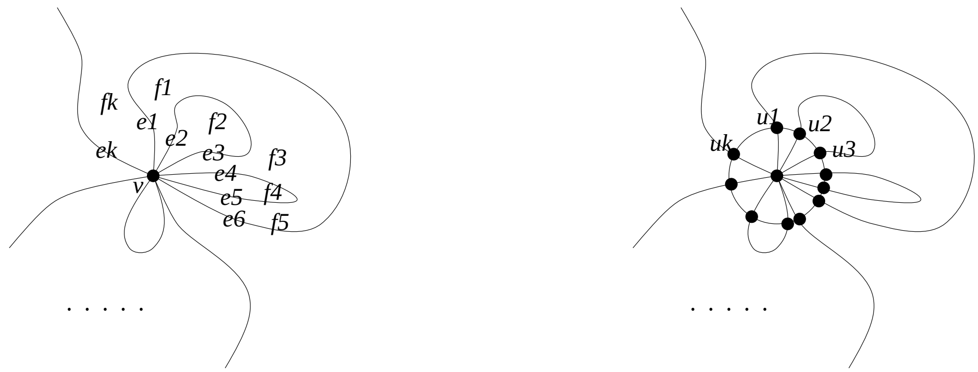
It turns out that the only additional information needed to capture all topological information about non-simple instances I is the *order* whereby all edges incident to each vertex are arranged, clockwise say, around the vertex. This is done by introducing a new relation $O \subseteq \{\hookleftarrow, \hookrightarrow\} \times V^3$, with the following meaning: $(\hookleftarrow, v, e_1, e_2) \in O$ iff v is a vertex and e_1, e_2 are clockwise consecutive[3] edges incident to v, and $(\hookrightarrow, v, e_1, e_2) \in O$ iff v is a vertex and e_1, e_2 are counterclockwise consecutive edges incident to v. But note that in the presence of loops at v, like in Figure 7 (b), we may have $(\hookleftarrow, v, e, e), (\hookrightarrow, v, e, e) \in O$. We define the *topological invariant* associated with an instance I to be the following finite structure: $T_I = (V, E, \delta, f_0, l, O)$ where $G_I = (V, E, \delta, f_0, l)$ is the structure defined earlier, and O is defined as above. Note that an isomorphism from T_I to $T_{I'}$ maps the set $\{\hookleftarrow, \hookrightarrow\}$ to itself, *possibly reversing* the orientation.

Example 3.3 *Continuing with the instance I in Figure 1 (recall also Figure 5), the invariant T_I would consist of the adjacency and labeling information (the graph G_I) as defined in the previous example, together with the following orientation information:*

$$\begin{aligned} O = \; & \{(\hookleftarrow, v_1, e_1, e_4), (\hookleftarrow, v_1, e_4, e_2), (\hookleftarrow, v_1, e_2, e_3), (\hookleftarrow, v_1, e_3, e_1), (\hookleftarrow, v_2, e_1, e_3), \\ & (\hookleftarrow, v_2, e_3, e_2), (\hookleftarrow, v_2, e_2, e_4), (\hookleftarrow, v_2, e_4, e_1), (\hookrightarrow, v_1, e_1, e_3), (\hookrightarrow, v_1, e_3, e_2), \\ & (\hookrightarrow, v_1, e_2, e_4), (\hookrightarrow, v_1, e_4, e_1), (\hookrightarrow, v_2, e_1, e_4), (\hookrightarrow, v_2, e_4, e_2), (\hookrightarrow, v_2, e_2, e_3), (\hookrightarrow, v_2, e_3, e_1), \} \end{aligned}$$

We are now ready to show:

[3] Two edges e_1, e_2 are *consecutive* if they share a face f and can be connected with a curve lying entirely in f, and which does not separate any other two edges bordering f.



(a) (b)

Figure 8: Illustration of G_I (a) and G_J (b). In this example $e_1 = e_6$, $f_1 = f_3 = f_5$, etc.

Theorem 3.4 *Let I, I' be spatial database instances over Alg with $names(I) = names(I')$. Then I and I' are topologically equivalent iff T_I and $T_{I'}$ are isomorphic via an isomorphism which is the identity on $names(I)$. Moreover, any isomorphism between T_I and $T_{I'}$ can be lifted to a homeomorphism mapping I to I'.*

Proof Assume first that I is connected: this implies that I' is also connected. For each vertex v in G_I, let $e_1, \ldots, e_n$ be the sequence of consecutive edges adjacent to v, in clockwise orientation (note that an edge may occur repeatedly – once or twice – among the e_i). Since all regions in I are semi-algebraic, we can find a circle centered in v with a radius $\varepsilon_v > 0$ small enough so that it intersects each edge e exactly as many times as e occurs among the e_i. Let $D_v = \{p \mid p \in \mathbb{R}^2, dist(p, v) < \varepsilon_v\}$, and furthermore, assume that the ε_v's are small enough s.t. $v \neq w \Longrightarrow D_v \cap D_w = \emptyset$. Construct similar discs $D'_{v'}$ for I'. Let $J \stackrel{\text{def}}{=} I \cup \{D_v \mid v \in V\}$, $J' \stackrel{\text{def}}{=} I' \cup \{D'_{v'} \mid v' \in V'\}$: then both J and J' are simple, by the following argument. The faces in J correspond either to old faces in I with their corners chopped out, or are new triangular faces. The first kind of faces will now have a simple boundary even if the original face had not, because we chopped out the corners. For the second kind, it is easy to see that each face with a triangle as a boundary (that is with three edges and three vertices) is simple.

Moreover, using the orientation information captured by O we will show that G_J and $G_{J'}$ are isomorphic. Hence J and J' are topologically equivalent by Lemma 3.2. It easily follows that I and I' are also topologically equivalent.

So it remains to show that $G_J, G_{J'}$ are isomorphic. Let $\varphi : V \to V'$ be an isomorphism between T_I and $T_{I'}$. We will extend it to an isomorphism between G_J and $G_{J'}$. Take a closer look at how G_J is derived from G_I. For some vertex v in G_I, let us denote with L the circular list $L = (e_1, f_1, e_2, f_2, \ldots, e_{k-1}, f_{k-1}, e_k, f_k)$ consisting of all outgoing edges from v and adjacent faces, in clock-wise fashion, as illustrated in Figure 8 (a). Note that we may have $e_i = e_j$, for $i < j$, when there are loops in I: each edge e may occur once or twice in L. Moreover, a face may occur arbitrarily many times in L. Let $v' = \varphi(v)$, and denote $L' = (e'_1, f'_1, \ldots, e'_k, f'_k)$ be the circular list of edges and faces adjacent to v' in $G_{I'}$, taken in clockwise order if $\varphi(\hookleftarrow) = \hookleftarrow, \varphi(\hookrightarrow) = \hookrightarrow$, and counterclockwise otherwise. The notation does not imply that $\varphi(e_1) = e'_1$, etc[4]: all we know is that $\{\varphi(e_1), \ldots, \varphi(e_k)\} = \{e'_1, \ldots, e'_k\}$, and $\{\varphi(f_1), \ldots, \varphi(f_k)\} = \{f'_1, \ldots, f'_k\}$. The key observation is that the new vertices, edges, and faces in G_J "around" v are uniquely determined by the circular list L, see Figure 8 (b). Namely there are k new vertices $u_1, \ldots, u_k$, $2k$ new edges, and k new faces. To prove that G_J and $G_{J'}$ are isomorphic, it suffices to show that φ maps the circular list L into L', i.e. that there exists $i \geq 1$ such that:

$$(\varphi(e_1), \varphi(f_1), \ldots, \varphi(e_k), \varphi(f_k)) = (e'_i, f'_i, e'_{i+1}, f'_{i+1}, \ldots, e'_{i-1}, f'_{i-1}) \tag{1}$$

Before proving that, we make an observation about L (and L'): any consecutive edge-cell or cell-edge pair is unique in L. That is, for $i < j$, we have $(e_i, f_i) \neq (e_j, f_j)$ and $(f_{i-1}, e_i) \neq (f_{j-1}, e_j)$. Back to proving (1), consider $\varphi(e_1)$. There can be at most two i's for which $e'_i = \varphi(e_1)$: if there are two such, then we will pick that i for which $\varphi(f_1) = f_i$. When there is a single i s.t. $e'_i = \varphi(e_1)$, we show that $\varphi(f_1) = f'_i$. Indeed, then we have $(\hookleftarrow, v, e_1, e_2) \in O$, hence $(\varphi(\hookleftarrow), v', \varphi(e_1), \varphi(e_2)) \in O'$. But we also have $(\varphi(\hookleftarrow), v', e'_i, e'_{i+1}) \in O'$, hence $e'_{i+1} = \varphi(e_2)$. Now f_1 is a common face for e_1 and e_2, so both $\varphi(f_1)$ and f'_i are common faces for e'_i, e'_{i+1}. It is easy to show, in general, that in any cell complex, two edges sharing a common vertex v' can have at most one common face. Hence $\varphi(f_1) = f'_i$. So we have chosen i such that (1) holds on the first two positions. We will prove that it holds on all positions. Comparing from left to right the two sequences in (1), consider the first position where they differ. If this is a face, then we have $\varphi(f_{j-1}) = f'_{i+j-2}, \varphi(e_j) = e'_{i+j-1}, \varphi(f_j) \neq f'_{i+j-1}$. This is a contradiction, because f_{j-1}, f_j are the unique two faces adjacent to e_j in G_I, and therefore are mapped by φ into f'_{i+j-2}, f'_{i+j-1}, which are the unique two faces adjacent to e'_{i+j-1} in $G_{I'}$. So suppose L, L' differ first at an edge, i.e. $\varphi(e_{j-1}) = e'_{i+j-2}, \varphi(f_{j-1}) = f'_{i+j-2}, \varphi(e'_j) \neq e'_{i+j-1}$. Let $e' = \varphi(e'_j)$. Here we use the orientation information. We have $(\hookleftarrow, v, e_{j-1}, e_j) \in O$, hence $(\varphi(\hookleftarrow), v', e'_{i+j-2}, e') \in O'$. It implies that e'_{i+j-2} must occur a second time in L', followed by e'. Moreover, f'_{i+j-2} is the only common face to e'_{i+j-2}, e', so there exists $p \neq j$ such that $e'_{i+p-2} = e'_{i+j-2}, f'_{i+p-2} = f'_{i+j-2}, e'_{i+p-1} = e'$. But this is a contradiction because, as we said earlier, any edge-face pair (e'_{i+j-2}, f'_{i+j-2}) occurs at most once in the circular list L'.

Now assume that I is not connected (recall: this means that its skeleton is not connected). For clarity assume that it consists of two connected components $I = I_1 \cup I_2$ and, hence, so does I', $I' = I'_1 \cup I'_2$. Let us call $\varphi : T_I \to T_{I'}$, the isomorphism between the first order structures T_I and $T_{I'}$: this splits into two isomorphisms φ_1, φ_2 between T_{I_1} and $T_{I'_1}$ and T_{I_2} and $T_{I'_2}$ respectively, which, in turn, extend to homeomorphisms λ_1, λ_2, mapping I_1 to I'_1 and I_2 to I'_2. For the sake of argument we can assume that both I_1 and I_2 are simple (if not, we apply the construction described above). It is easy to see that all regions in I_2 lie entirely inside one of the 2-dimensional cells R_v of T_{I_1} (not necessarily the exterior cell). By [St93] every homeomorphism $\lambda : \mathbb{R}^2 \to \mathbb{R}^2$ is isotopic to either the identity or to a reflection: in short, there are two possible orientations for a homeomorphism. It is possible to show that we can join λ_1 and λ_2 together into a homeomorphism mapping I to I', provided that λ_1, λ_2 have the same "orientation", i.e. are either both isotopic to the identity, or both to a reflection. Assume the contrary, i.e. that they have different orientations. Here we distinguish two cases:

[4] L and L' have the same length, because the length of L is uniquely determined by the number of edges incident to v, and by the number of "loops" at v, i.e. edges having v as the unique endpoint. Since G_I and $G_{I'}$ are isomorphic, it follows that L, L' have the same length.

1. I_1 has no vertices, or all its vertices have at most two outgoing edges. It is easy to see that in this case I_1 consists of only one region, or of two regions whose boundaries touch in exactly one point. Then φ_1 can be extended both to a homeomorphism λ_1 isotopic to the identity, or to one isotopic to a reflection, so we can arrange for λ_1 and λ_2 to have the same orientation.

2. There exists a vertex v_1 in I_1 with at least three distinct outgoing edges, say e_{11}, e_{12}, e_{13} in clockwise order, and similarly $v_2, e_{21}, e_{22}, e_{23}$ in I_2. This fact is recorded in O. Now λ_1 and λ_2 have different orientations, say λ_1 is isotopic to the identity and λ_2 to a reflection. It follows that λ_1 maps $v_1, e_{11}, e_{12}, e_{13}$ to $v_1', e_{11}', e_{12}', e_{13}'$ where e_{11}', e_{12}', e_{13} have the same orientation as e_{11}, e_{12}, e_{13}, so $\varphi(\hookleftarrow) = \hookleftarrow$ and $\varphi(\hookrightarrow) = \hookrightarrow$. On the other hand, λ_2 maps $v_2, e_{21}, e_{22}, e_{23}$ to $v_2', e_{21}', e_{22}', e_{23}'$ where $e_{21}', e_{22}', e_{23}'$ are of opposite orientation than e_{21}, e_{22}, e_{23}, so $\varphi(\hookleftarrow) = \hookrightarrow$ and $\varphi(\hookrightarrow) = \hookleftarrow$. This is a contradiction.

□

Theorem 3.4 says that each $\mathcal{H}$-equivalence class of spatial instances over *Alg* is characterized by its topological invariant. Furthermore, the invariant, *as well as a representative instance in Poly*, can be constructed efficiently.

Theorem 3.5 *For each instance I in Alg, we can compute in polynomial time (and in NC) the invariant T_I, and an instance I' in Poly such that $T_I = T_{I'}$.*

Proof For I in *Alg*, T_I can be computed in *NC* using the cell decomposition algorithm of [KY85]. The topological invariant also allows to construct, for each $\mathcal{H}$-equivalence class, a representative instance over *Poly*, as follows.

A classical result in graph theory known as *Fáry's Theorem* [Fá48, Wag36, St51] states that any planar graph can be embedded in the plane so that *all of its edges are straight lines* (except of its loops, of course, which can be triangles), and so that its unbounded face forms a convex polygon. Furthermore, this can be accomplished in linear time. It is interesting to ask whether it can be done in *NC*; we sketch below an argument that it can.

It follows from the standard parallel planarity algorithms [JS85] that this can be done in *NC*if the graph is triply connected. For general graphs, once we have found and embedded the triply connected components in *NC*, each component represents the adjacent components as edges. Starting from any component, we identify the edges that stand for other components, and embed these components by straight lines in the space provided by the edge (we may have to lose the convex perimeter property in the process, because an edge may be standing for many triply connected components). Once these components have been embedded, we embed in them their adjacent components, and so on (and this seemingly sequential process can be carried out in *NC*by path doubling).

Finally, once a straight line embedding has been obtained, region A is defined to be the following open polygonal disc: The interior of the union of all cells that are labeled A^o or A^∂. □

The thematic mapping

Theorem 3.4 shows that from each instance I over *Alg* one can build in polynomial time a "summary" T_I which contains *exactly* enough information needed to answer all topological queries. The invariant is easily represented in classical relational database, i.e. *thematic*, form. Rather than doing this, we exhibit a variation which has a more intuitive presentation and suggests the elements of a topological data model, in the spirit of the PLA model. Let **Th** be the relational database schema consisting of the following relations (their intuitive meaning in relation to the invariant is also described):

1. *Regions*, *Vertices*, *Edges*, *Faces*, and *Exterior-face* are unary relations providing the region names, the cells of dimension 0, 1, 2, and the exterior face.
2. *Endpoints* is a ternary relation providing endpoint(s) for edges.
3. *Face-Edges* is a binary relation providing, for each face (including the exterior cell), the edges on its boundary.
4. *Region-faces* is a binary relation providing, for each region name, the set of faces it contains.
5. *Orientation* is a 4-ary relation coinciding with relation O in the invariant.

For each spatial instance I in Alg, it is now easy to construct a relational instance $thematic(I)$ over schema **Th**. This is illustrated next.

Example 3.6 *Consider the spatial instance I in Figure 1 (c). The invariant T_I is described in Examples 3.1 and 3.3. The instance $thematic(I)$ is depicted in Figure 9.*

We can now show:

Corollary 3.7 *(i) The mapping* thematic *from spatial database instances over Alg to relational instances over* **Th** *is computable in polynomial time (and NC).*
(ii) For all spatial instances I, J over Alg, such that $names(I) = names(J)$, I and J are topologically equivalent iff $thematic(I)$ and $thematic(J)$ are isomorphic by an isomorphism which is the identity on $names(I) = names(J)$.
(iii) For each recursive topological property τ of spatial instances over Alg there exists a recursive property $thematic(\tau)$ of instances over **Th**, *of complexity polynomially related to that of τ, such that for each spatial instance I over Alg,*

$$I \models \tau \textit{ iff } thematic(I) \models thematic(\tau).$$

Proof For instances I, J in Alg such that $names(I)$=$names(J)$, it is clear that T_I is isomorphic to T_J iff $thematic(I)$ is isomorphic to $thematic(J)$ (both by isomorphisms which are the identity on *names(I)*). This together with Theorem 3.4 show (i) and (ii). Consider (iii). The proof makes use of the fact that one can check in *NC*whether an instance over **Th** is in the image of *thematic* (Theorem 3.8). For some topological property τ of algebraic instances, let $thematic(\tau)$ be defined as follows. An instance T over **Th** satisfies $thematic(\tau)$ if (a) it is in the image of *thematic*, and (b) the spatial instance I in $Poly$ such that $thematic(I) = T$, constructed in *NC*from T as described in Theorem 3.5, satisfies τ. Since τ is a topological property, $thematic(\tau)$ does not depend on the particular choice of I and so is well defined. □

In summary, it should be clear from our discussion that for a spatial instance I in Alg, T_I and $thematic(I)$ are basically cosmetic variants of each other. Also, it is easy to obtain $thematic(I)$ given T_I, and conversely. The invariant T_I has the advantage of being closer to existing formalism developed in the context of computational geometry (e.g. [KY85]) while $thematic(I)$ is closer to

Regions
A
B

Vertices
v_1
v_2

Edges
e_1
e_2
e_3
e_4

Faces
f_0
f_1
f_2
f_3

Exterior-face
f_0

Orientation			
$\hookleftarrow$	v_1	e_1	e_4
$\hookleftarrow$	v_1	e_4	e_2
$\hookleftarrow$	v_1	e_2	e_3
$\hookleftarrow$	v_1	e_3	e_1
$\hookleftarrow$	v_2	e_1	e_3
$\hookleftarrow$	v_2	e_3	e_2
$\hookleftarrow$	v_2	e_2	e_4
$\hookleftarrow$	v_2	e_4	e_1
$\hookrightarrow$	v_1	e_1	e_3
$\hookrightarrow$	v_1	e_3	e_2
$\hookrightarrow$	v_1	e_2	e_4
$\hookrightarrow$	v_1	e_4	e_1
$\hookrightarrow$	v_2	e_1	e_4
$\hookrightarrow$	v_2	e_4	e_2
$\hookrightarrow$	v_2	e_2	e_3
$\hookrightarrow$	v_2	e_3	e_1

Endpoints	
e_1	v_1
e_1	v_2
e_2	v_1
e_2	v_2

Face-edges	
f_0	e_1
f_0	e_4
f_1	e_1
f_1	e_3
f_2	e_2
f_2	e_4
f_3	e_2
f_3	e_3

Region-faces	
A	f_1
A	f_2
B	f_2
B	f_3

Figure 9: Thematic instance for the spatial instance in Figure 1 (c)

topological data models. We note that, as a model for topological spatial information, $thematic(I)$ can be viewed as an augmentation of the PLA model of [Cor79] (see also descriptions in [LT92, Par95]).

If topological invariants are to be used as a model for topological spatial databases, it becomes important to check whether a given instance over **Th** is in fact in the image of the *thematic* mapping. Indeed, this functions as an integrity constraint when updates are performed. Clearly, not every instance over **Th** is a valid specification of a topological invariant. There are obvious integrity constraints satisfied by every instance in the image of *thematic*. For example, the instance must represent a graph, so vertices and edges are disjoint sets and each edge has one or two endpoints (which must be vertices). But there are also more subtle constraints, such as the fact that Euler's formula must hold, that is $|Faces| = |Edges| - |Vertices| + 2$. To fully characterize the instances over **Th** which describe valid topological information, we characterize instances which are invariants as *labeled planar graphs*. This will allow us to show the following:

Theorem 3.8 *It can be checked in NC whether an instance over* **Th** *is the image of a spatial instance over Alg via the* thematic *mapping.*

We next present the characterization of topological invariants as labeled planar graphs. This provides a purely combinatorial (that is, with no recourse to geometry) characterization of the invariants. Although the definition of labeled planar graph is common knowledge among researchers in algorithmic graph theory, we have been unable to find a rigorous standard exposition.

The topological invariant as labeled planar graph

Consider an instance over **Th**. We use the terms *vertex, edge, face* and *exterior face* for elements of relations *Vertices, Edges, Faces* and *Exterior-face*, respectively. We begin by identifying some very basic requirements of every instance which is in the image of *thematic*, which essentially ensure that the instance represents a graph:

(1) $Vertices, Edges, Faces$ and $Regions$ are pairwise disjoint and $Exterior\text{-}face$ consists of a single element in $Faces$. $\pi_1(Orientation)$ has two elements.

(2) $\pi_1(Endpoints) \subseteq Edges$, $\pi_2(Endpoints) \subseteq Vertices, \pi_1(Face\text{-}edges) \subseteq Faces, \pi_2(Face\text{-}edges) \subseteq Edges, \pi_2(Region\text{-}faces) \subseteq Faces$, $\pi_1(Orientation)$ has two elements, $\pi_2(Orientation) = Vertices$, $\pi_3(Orientation) \cup \pi_4(Orientation) \subseteq Edges$.

(3) every edge has one or two vertices as endpoints (as specified by relation *Endpoints*).

Call an instance satisfying (1)-(3) a *candidate graph.* A candidate graph is *connected* if it is connected in the usual sense.

Notice that nothing is said so far about the meaning of faces, and of the orientation given in a candidate graph. Recall that the orientation is supposed to provide, for each vertex, the clockwise and counterclockwise orientation of its incident edges. This yields the following additional requirement:

(4) for each $\xi \in \pi_1(Orientation)$ and $v \in Vertices$, $\pi_{3,4}(\sigma_{1=\xi,2=v}(Orientation))$ is a cyclic permutation of edges.

For a graph satisfying (4), let us fix $\xi \in \pi_1(Orientation)$ and denote by π_v the cyclic permutation $\pi_{3,4}(\sigma_{1=\xi,2=v}(Orientation))$, for each vertex v. Intuitively, π_v is a clockwise (or counterclockwise) arrangement of the edges incident upon v.

There is an important connection between faces and the orientation of edges. If e, f are edges that share an endpoint v and are on the boundary of a face, then e and f are consecutive with respect to π_v. This yields the next requirement on the candidate graph:

(5) for each face f and edge e of f, there is a unique edge e' of f such that e and e' share an endpoint v, and e, e' are consecutive edges (in this order) in π_v: and a unique edge e'' of f such that e'' and e share an endpoint u, and e'', e are consecutive edges in π_u. That is to say, faces are sets of closed paths.

Let us call a candidate graph satisfying (4)-(5) an *embedded graph.* Unfortunately, we are not quite done: there are embedded graphs which are not invariants, *because they cannot be drawn in a planar fashion.* For example, each such embedded graph, provided it is connected, must satisfy Euler's formula:

(6) $|Faces| \ = \ |Edges| \ - \ |Vertices| \ + \ 2$.

A connected embedded graph satisfying (6) is called a *connected planar graph.*

To generalize this to the case when the embedded graph is not connected is not very hard, and we sketch one way below. Intuitively, the connected components of the graph must be embedded in some face of one another; this "embedded-in" relation is a tree. The faces of the overall graph now contain the edges of the unbounded faces of the graphs embedded in them. The unbounded face of the whole graph is that of the root of the tree. Such an embedded graph is called a *planar graph.*

The last point to be dealt with is the assignment of faces to region names. We have to make sure that each region can be embedded in the plane as a disc. This is done using the notion of dual graph. The *dual graph* of a planar graph is a graph with *Faces* as nodes, and with two faces connected iff they share an edge. For example, the dual graph of the planar graph in Figure 5 is the graph consisting of the cycle $[f_0, f_1, f_3, f_2]$. The condition desired is now the following:

(7) For each $X \in Regions$, let $faces(X)$ be the set of faces in X (provided by *Region-faces*). Then for each region name X, (i) the restrictions of the dual graph to $faces(X)$ and to $Faces \ - \ faces(X)$ are connected graphs, and (ii) $f_0 \notin faces(X)$ where f_0 is the exterior face.

A planar graph satisfying (7) is called a *labeled planar graph.*

For example, it is easily seen that the structure over **Th** in Figure 9 is a labeled planar graph. The following can now be shown:

Lemma 3.9 *An instance over* **Th** *is an invariant iff it is a labeled planar graph.*

Checking that a graph (an instance over **Th**) is a labeled planar graph can be done in *NC*, since it only involves only arithmetic and variants of graph connectivity.

This observation together with Lemma 3.9 yields Theorem 3.8.

4 Region-Based Languages

Recent work on constraint query languages [KKR90, GS94, GST94, Par+95, BDLW95] has focused on languages having finitely specified *regions* as inputs, and whose variables range over *reals* and/or *points*. Here we investigate query languages for which both data and variables range over *regions*. These languages can be viewed as natural closures of the Egenhofer-Franzosa relationships under first-order operators. The various languages we consider have the same syntax, but vary in the set of regions over which their quantifiers range; this yields in effect a family of region-based languages.

The syntax uses region variables $p, r, \ldots$, name variables $a, b, c, \ldots$, and name constants $A, B, \ldots$ from **Names**. A name expression is either a (a name variable) or A from **Names**. Region expressions are either a region variable or $ext(a)$ for a a name expression. Atoms are expressions $a = b$ with a, b name expressions, or $relationship(p, q)$ where *relationship* is one of the Egenhofer-Franzosa relationships, and p, q are region expressions. The boolean connectives and quantifiers are standard. We will drop the $ext(\_)$ notation whenever it is clear from the context. E.g. we will write $\exists p.inside(p, A) \wedge inside(p, B) \wedge inside(p, C)$ for the query testing whether $A \cap B \cap C \neq \emptyset$, instead of the official $\exists p.inside(p, ext(A)) \wedge inside(p, ext(B)) \wedge inside(p, ext(C))$. For a given database instance I, the name variables range over $names(I)$ (which is a finite set). The region variables range over an infinite set of regions, independent of I, consisting of all regions of a certain type. Each of the languages we consider is parameterized by both the type of regions over which region variables range, and the type of the input regions as follows: $FO(Region, Region')$ denotes the language where region variables range over *Region* and inputs are of type *Region'*.

We shall denote with $FO_G(Region, Region')$ the subset of queries expressible in $FO(Region, Region')$ which are G-generic relative to *Region'*.

We note that the languages $FO(Region, Region')$ can be assumed to only use the Egenhofer-Franzosa relationship *disjoint*, or, equivalently, its negation $connect(r, r') \stackrel{\text{def}}{=} \neg disjoint(r, r')$, which is topologically equivalent to $\bar{r} \cap \bar{r}' \neq \emptyset$ ($\bar{r}$ is the topological closure of r). Indeed, first observe that $r \subseteq r'$ is expressed as $\forall r''.(connect(r, r'') \Rightarrow connect(r', r''))$. We have: $overlap(r, r') = \exists r''.(r'' \subseteq r \wedge r'' \subseteq r') \wedge \neg(r \subseteq r') \wedge \neg(r' \subseteq r)$, $meet(r, r') = connect(r, r') \wedge \neg overlap(r, r') \wedge \neg(r \subseteq r') \wedge \neg(r' \subseteq r)$, etc. Note that $r \subseteq \overline{r' \cup r''}$ can be expressed as $\forall q.connect(r, q) \Longrightarrow connect(r', q) \vee connect(r'', q)$.

Example 4.1 *We illustrate $FO(Region, Region')$, where Region can be any of Rect, $Rect^*$, Poly, Alg, Disc. Consider the two database instances I, I' in Figure 1 (a) and (b). The following query φ separates them, in the sense that $I \models \varphi$ but $I' \not\models \varphi$: $\varphi = \exists r.(r \subseteq A \cap B \cap C)$.*

Example 4.2 *We can test in $FO(Region, Region')$ whether a set is topologically connected, provided that we restrict Region to $Rect^*$, Poly, Alg, Disc (i.e. it cannot be Rect). Consider the two instances in Figure 1 (c) and (d): they are separated by $\forall r.\forall r'.(r \cup r' \subseteq A \cap B \Longrightarrow \exists r''.(r'' \subseteq A \cap B \wedge connect(r'', r) \wedge connect(r'', r')))$. Next consider first instances I, I' in Figure 7 (b): they are separated by the query $\exists r.\exists r'.path(A, r, B) \wedge path(C, r', D) \wedge r \cap r' = \emptyset$. Here and in the sequel we will use the ambiguous notation $path(A, r, B)$ to mean that "r is a path from A to B without touching the other regions", in our case: $connect(A, r) \wedge connect(B, r) \wedge \neg connect(C, r) \wedge \neg connect(D, r)$. Finally, consider the instances I, I' in Figure 7 (a). They are separated by the following query:*

$$\neg(\exists r.\exists r'.\exists r''.path(A, r, D) \wedge path(B, r', E) \wedge path(C, r'', F) \wedge disjoint(r, r', r''))$$

Consider a group G of permutations of the space. Whenever both *Region* and *Region'* are invariant under G, all queries expressed in $FO(Region, Region')$ are G-generic. It follows that the languages $FO(Region, Disc)$ are generic with respect to the groups shown in Figure 10. Interestingly, even if *Region* and *Region'* are not G-invariant, queries in $FO(Region, Region')$ may be G-generic (relative to Region'). Indeed, using a straightforward variant of the Ehrenfeucht-Fraissé games for FO and Theorem 3.4 we prove:

Proposition 4.3 *All queries expressible in $FO(Alg, Alg)$ and $FO(Poly, Poly)$ are $\mathcal{H}$-generic.*

Proof We sketch the proof for $FO(Alg, Alg)$; the one for $FO(Poly, Poly)$ is similar. Given two homeomorphic instances I, J in Alg, we have to show that:

$$I \models \varphi \iff J \models \varphi \tag{2}$$

for any formula φ in $FO(Alg, Alg)$.

We define an Ehrenfeucht-Fraissé game with unbounded number of moves, which is played on two homeomorphic instances I, J in Alg. As usual, the game is played by two players, Spoiler and Duplicator. In the first move, Spoiler augments one of the instances, say I, with an additional region I_1 in Alg, and Duplicator responds by augmenting J with a new region J_1 in Alg. This is repeated, with Spoiler choosing again one of the instances and Duplicator responding in the opposite instance. Let $I_1, \ldots, I_k$ be the regions added to I after k moves, and $J_1, \ldots, J_k$ be those added to J. A *round* of the game consists of a sequence of pairs $\{\langle I_i, J_i\rangle\}_{1\leq i\leq k}$ as above. The Duplicator wins the round $\{\langle I_i, J_i\rangle\}_{1\leq i\leq k}$ if I augmented with $I_1, \ldots, I_k$ and J augmented with $J_1, \ldots, J_k$ are isomorphic as finite, first-order structures: this amounts to $connect(I_m, I_n) \iff connect(J_m, J_n)$, for $m, n = 1, k$. The Duplicator has a winning strategy if he can win any round k of any game, no matter how Spoiler plays. Let us denote by $I \approx J$ the fact that Duplicator has a winning strategy on I and J. The following can now be shown for I and J in Alg:

(†) if $I \approx J$ then $I \equiv_{FO(Alg,Alg)} J$;

(‡) if I and J are topologically equivalent then $I \approx J$.

Note that the proposition follows from (†) and (‡). The proof of (†) is as for classical games. We sketch a proof for (‡). We show that the duplicator has a winning strategy in a harder game, namely in which he is required to make $I_1, \ldots, I_k$ homeomorphi to $J_1, \ldots, J_k$, for every $k \geq 0$. We have to show that Duplicator can maintain indefinitely the homeomorphism between the constructed instances, no matter how Spoiler plays. Suppose I', J' extending I, J have been constructed and are homeomorphic by some homeomorphism λ, and suppose Spoiler augments I' with I_k (in Alg), yielding I''. Clearly, J' augmented with $\lambda(I_k)$ is homeomorphic to I''. Unfortunately, Duplicator cannot simply choose $\varphi(I_k)$, since this region is not necessarily in Alg (we only know it is a disc). However, it is easily seen that there exists a J_k in Alg (and even in $Poly$) such that J' augmented with J_k is homeomorphic to I''. Intuitively, J_k is obtained by approximating the boundary of $\varphi(I_k)$ with a polygon closely enough that it generates the same topological invariant as $\varphi(I_k)$ (we omit the details). $\square$

Relative expressiveness

With some exceptions, the languages $FO(Region, Region')$ are incomparable. The following theorem summarizes their relationships when inputs range over Alg and $Disc$. This is mainly due to the different non topological queries they express: by restricting to topological queries we obtain a nice hierarchy, which justifies in part their choice in this paper.

$FO(Rect, Disc)$	$FO(Rect^*, Disc)$	$FO(Poly, Disc)$	$FO(Alg, Disc)$	$FO(Disc, Disc)$
$\mathcal{S}$	$\mathcal{S}$	$\mathcal{L}$	$\mathcal{L}$	$\mathcal{H}$

Figure 10: Groups with respect to which various languages $FO(Region, Disc)$ are generic

	$FO(Rect^*, Region')$	$FO(Poly, Region')$	$FO(Alg, Region')$	$FO(Disc, Region')$
$FO(Rect, Region')$	$\subset$	#	#	#
$FO(Rect^*, Region')$		#	#	#
$FO(Poly, Region')$			#	#
$FO(Alg, Region')$				$\subset$ and #

Figure 11: Relationships between various languages $FO(Region, Region')$, for $Region' \in \{Alg, Disc\}$: # denotes "incomparable". The last entry is $\subset$ for Alg, and # for $Disc$.

Theorem 4.4 *The relationships of Figure 11 hold between the various languages $FO(Region, Region')$, for $Region' \in \{Alg, Disc\}$. Moreover:*

$$\begin{aligned} FO_{\mathcal{H}}(Rect, Alg) \subset FO_{\mathcal{H}}(Rect^*, Alg) &= FO_{\mathcal{H}}(Poly, Alg) \\ &= FO_{\mathcal{H}}(Alg, Alg) \subset FO_{\mathcal{H}}(Disc, Alg) \end{aligned}$$

Proof (sketch) To prove most of the incomparability results, consider, for some language $FO(Region, Disc)$, the query $Q_{Region} = (\exists r. r = A)$, stating that input region A is in $Region$. To prove that $FO(Region, Disc) \not\subseteq FO(Region', Disc)$, it suffices to check in Figure 4 that $Region$ is not invariant under the group G associated to $FO(Region', Disc)$ in Figure 10. This proves all # entries except the third entry in the third column and those in the last column, and proves the $\not\subseteq$ relationships for the last column. We use the complexity-theoretic arguments of Theorem 6.1 to prove the $\not\supseteq$ relationships of the first and last column. The third # entry in the third column is proven by a standard game-theoretic argument. Finally $FO(Rect, Disc) \subseteq FO(Rect^*, Disc)$ follows from the following fact:

(†) $FO(Rect^*, Rect^*)$ can express the query "is r is rectangle ?".

To prove (†), let $edge(r, r') \stackrel{\text{def}}{=} meet(r, r') \wedge \exists r''.(overlap(r, r'') \wedge overlap(r', r''))$ be the predicate testing whether r, r' meet and have at least a non-zero length portion of an edge in common ($meet(r, r')$ only ensures that r and r' meet at one or several corners). Next, let $corner(r, r') \stackrel{\text{def}}{=} meet(r, r') \wedge \neg edge(r, r')$. Then some $r \in Rect^*$ is a rectangle when "it has exactly 4 corners", i.e. when there exists 4 pairwise disjoint regions r_1, r_2, r_3, r_4 cornering r, but there do not exists 5 such regions. $\square$

As an aside, the relationship between $FO(Rect, Region')$ and $FO(Rect^*, Region')$ is further illuminated by the following result. We define the second order query language $SO(Rect, Region')$ by extending $FO(Rect, Region')$ as follows. The language uses second-order variables $X, Y, Z, \ldots$, and atoms $r \in X$, where r is a region variable. Both $\exists\, X$ and $\forall\, X$ quantifiers over the second-order variables are allowed. The meaning is that the second-order variables range over *finite* sets of regions in *Rect*. A common query in $SO(Rect, Region')$ is $chain(X)$, testing whether $X = \{r_1, r_2, \ldots, r_n\}$ such that $connect(r_i, r_{i+1})$, $i = 1, n-1$ and $\neg(connect(r_i, r_j))$ for $|i-j| > 1$. Namely we express $chain(X)$ as $\exists r_1.\exists r_n.(r_1 \in X \wedge r_n \in X \wedge \psi)$, where ψ assers that any other $r \in X$ different from r_1, r_n is connected to exactly two regions in X, while r_1, r_n are each connected to exactly one region in X.

Proposition 4.5 *The following holds:*

$$SO(Rect, Region') = FO(Rect^*, Region')$$

Proof The above result is quite intuitive, since quantifying over regions in $Rect^*$ is, in some sense, quantifying over sets of regions in *Rect*. But there are subtle technical differences between the two languages, since an arbitrary finite set X of rectangles does not correspond immediately to a region in $Rect^*$ ($\bigcup X$ may be disconnected, or may have holes). We sketch the proof below.

For $FO(Rect^*, Region') \subseteq SO(Rect, Region')$, let φ be a formula in $FO(Rect^*, Region')$. We replace each quantifier $\exists r$ in φ (where r ranges over $Rect^*$) with: $\exists X.isDisc(X)$, and replace every subformula $connect(r, \ldots)$ with $\exists r' \in X.connect(r', \ldots)$. Here $isDisc(X)$ tests whether $\bigcup X$ is in *Disc*, i.e. is topologically connected and has no holes. We can test connectedness, because we can quantify over *chains*, see above. Testing whether $\bigcup X$ has no holes amounts to testing whether its complement is connected.

For the converse, $SO(Rect, Region') \subseteq FO(Rect^*, Region')$, we have to replace first-order quantifiers $\exists r_0$ over *Rect* with quantifiers over $Rect^*$, and second-order quantifiers $\exists X$ over finite sets of regions in *Rect* with quantifiers over $Rect^*$. The first part is simple, because we can express $isRect(r)$ in $FO(Rect^*, Region')$, see the proof of Theorem 4.4. To see the difficulties for the second order quantifier, note that even when $\bigcup X$ is a disc, $r = \bigcup X$ does not capture accurately the information held by X, because there may be several ways to decompose r into a union of rectangles. The first idea is to represent X as the intersection of two regions $r_1, r_2 \in Rect^*$: this works if all rectangles in X are disjoint, as illustrated in Figure 12 (a). So assuming all rectangles in X to be disjoint, we can replace the quantifier $\exists X$ with $\exists r_1\, \exists r_2$, and the atomic formulas $r \in X$ with $isRect(r) \wedge r \subseteq r_1 \wedge r \subseteq r_2 \wedge$ "*r is maximal such*". So it suffices to observe that an arbitrary set X of rectangles can be encoded in terms of two sets X_1, X_2 of disjoint rectangles, and a set C of *correspondences*, see Figure 12 (b). Here X, X_1, X_2 all have the same number of rectangles: n. The set C defines a bijection between the rectangles in X_1 and those in X_2: it consists of n disjoint chains, each connecting some $r_1 \in X_1$ with some $r_2 \in X_2$. We do not have to express C in $SO(Rect, Region')$, but rather in $FO(Rect^*, Region')$, as the intersection of two regions in $Rect^*$. The idea is that each rectangle $r \in X$ can be uniquely identified in $FO(Rect^*, Region')$ from its two corresponding rectangles $r_1 \in X_1, r_2 \in X_2$, as having r_1, r_2 as "cartesian coordinates", see Figure 12 (b). There is a single remaining twist here: r has a mirror image having the same cartesian coordinates r_1, r_2, so it is not uniquely identifiable. To distinguish it from its mirror image, we define bounding boxes q, q_1, q_2 for X, X_1, X_2, such that q_1, q_2 are the cartesian coordinates of q, require that the mirror image of q be disjoint from q, and finally require for all rectangles $r \in X$, to satisfy $r \subseteq q$. We leave the details to the reader. $\square$

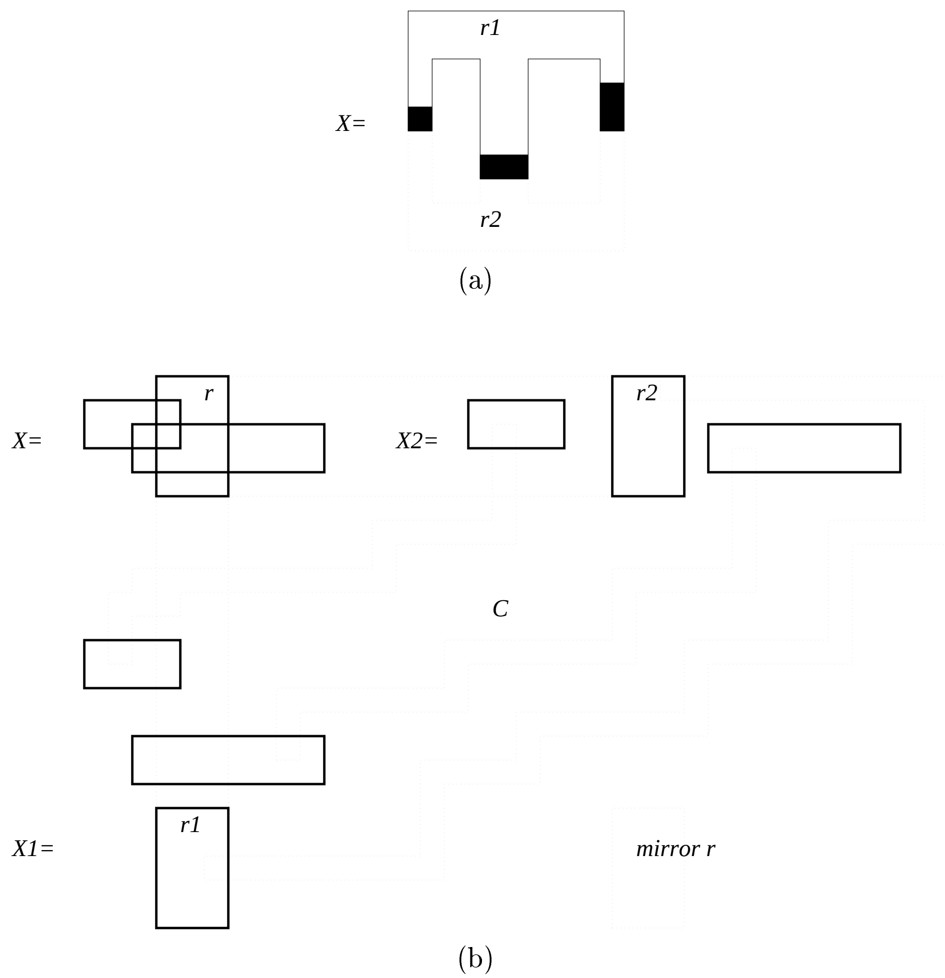


Figure 12: Figure (a) shows how to encode a set X of three disjoint rectangles (black) as an intersection $r_1 \cap r_2$, with $r_1, r_2 \in Rect^*$. Figure (b) shows how to encode a set X of three non-disjoint rectangles as two sets X_1, X_2 of disjoint rectangles and a "correspondence" C: here C is represented as the intersection of two regions in $Rect^*$ (not shown). Each r is uniquely determined by two rectangles $r_1 \in X_1, r_2 \in X_2$ in "correspondence". The choice between r and its mirror is made by specifying three rectangles B_1, B_2, B which are bounding boxes for X_1, X_2, and X respectively (not shown), and requiring that $r \subseteq B$.

5 Completeness

As promised, we will show that the closure of the Egenhofer-Franzosa relationships under appropriate logical operators is, in some sense, complete. We present two types of results: first, we aim for "absolute" completeness by showing how *all* topological properties of inputs over Alg can be expressed in languages based on the Egenhofer-Franzosa relationships. Second, we look at "relative" completeness results that take as reference a point-based spatial language: here we show that $FO(Rect, Disc)$ expresses precisely the same topological properties as those definable in the point-based language.

Absolute completeness

We have seen that the pairwise Egenhofer-Franzosa relationships between pairs of regions in an instance are not sufficient to determine the input up to homeomorphism (see Figure 1). However, the information provided by their first-order closure, in the style of our region based languages, is sufficient. The proofs use the *topological invariants* discussed at length in Section 3. This works for inputs in Alg, but not for general inputs in $Disc$. The restriction to semi-algebraic regions is needed because, as discussed in Section 3, every database instance in Alg has a topological invariant expressible as a finite structure. For database instances in $Disc$ the corresponding structure may be infinite.

Let $\equiv_{FO(Region,Alg)}$ be the equivalence relation on spatial instances over Alg defined by: $I \equiv_{FO(Region,Alg)} J$ iff I and J cannot be distinguished by sentences in $FO(Region, Alg)$. Also, let $I \equiv_{\mathcal{H}} J$ denote the fact that I, J are $\mathcal{H}$-equivalent. We will make use of the following key definability result:

Proposition 5.1 *Each $\mathcal{H}$-equivalence class of spatial instances over Alg is definable by a sentence in any of the languages $FO(Rect^* - Disc, Alg)$.*

Proof For each instance I over Alg we construct a sentence φ_I that tests whether the topological invariant of an instance I' is isomorphic to T_I. Then φ_I defines the $\mathcal{H}$-equivalence class of I. First, φ_I checks if $names(I') = names(I) = \{R_1, \ldots, R_n\}$: $\varphi_I = \exists a_1, \ldots, a_n.(\bigwedge_{1\leq i\leq n} a_i = R_i) \wedge \forall a.(\bigvee_{1\leq i\leq n} a = R_i) \wedge \psi$. Next, ψ checks whether $R_1, \ldots, R_n$ satisfy precisely the same topological relationships as dictated by T_I. For this, we start by picking a region for each cell in V: $\psi = \exists r_1, \ldots, r_{|V|}.\omega$. Here ω "says" that the regions r_v are disjoint, that each "covers" the corresponding cell in T_I, and that their topological relationships are those of T_I. By "cover" we mean different things depending on the dimension of the cell R_v: for 2-dimensional cells R_v, "covers" means $r_v \subseteq R_v$; for 1-dimensional cells "covers" means that r_v has a nonempty intersection with the edge R_v, while for points, it simply means $R_v \in r_v$. Finally, ω checks, one by one, all relationships in T_I. First, check that f_0 is the exterior face: $\exists r.R_1 \subseteq r \wedge \ldots \wedge R_n \subseteq r \wedge \neg(connect(r, r_{f_0}))$. Next check the adjacency relationships in E. For a edge-face pair $(e, f) \in E$, check $\exists r.path(r_e, r, r_f)$, where *path* is in the spirit of Example 4.1: here it checks that r is connected to both r_e and r_f, that it lies entirely in the regions R_i for which the label of f is "interior", and that it does not touch any other regions. For a vertex-edge pair $(v, e) \in E$ and vertex-face pair $(v, f) \in E$ we proceed similarly, with minor variations. Finally, we check the orientations enforced by T_I, using the idea in Example 4.1 to separate the instances in Figure 7 (a). Note that we use in an essential way the fact that each of the languages $FO(Region, Alg)$ considered can express the fact that a region is topologically connected. □

We can now show the following central result:

Theorem 5.2 *For every pair of instances I, J over Alg and $Region \in \{Alg, Disc\}$,*

$$I \equiv_{\mathcal{H}} J \text{ iff } I \equiv_{FO(Region, Alg)} J.$$

The "if" part of the theorem follows from Proposition 5.1, and the "only if" part from Proposition 4.3.

For $Region \in \{Rect, Rect^*, Poly\}$ only the "if" part of the equivalence holds, but not the "only if" part, because these languages can express properties which are not $\mathcal{H}$-generic. Thus, we have:

Proposition 5.3 *For $Region \in \{Rect, Rect^*, Poly\}$, and inputs I, J in Alg,*

$$I \equiv_{FO(Region, Alg)} J \Longrightarrow I \equiv_{\mathcal{H}} J.$$

Proof The statement for $Rect^*, Poly$ follows again from Proposition 5.1. For $Rect$, a direct proof is used: for every pair of instances I, J over Alg which are not $\mathcal{H}$-equivalent, we construct a sentence $\sigma_{I,J} \in FO(Rect, Alg)$ that separates them. This is done as in he proof of Proposition 5.1, up to the existential quantifiers for r_v, $v \in V$. The only part we cannot express in $FO(Rect, Alg)$ is $\exists r.path(r_e, r, r_f)$ that asserts that some region r connects r_e with r_f, and lies entirely within certain of the input regions $R_1, \ldots, R_n$. But for a particular input I, we can replace each such r with a fixed number of rectangles: $\exists r_1 \ldots \exists r_p.path(r_e, r_1, r_2) \wedge path(r_2, r_3, r_4) \wedge \ldots \wedge path(r_{p-1}, r_p, r_f)$, where p depends on the instance I. □

The question of finding a language expressing precisely the topological properties of instances over Alg has been open [Par95]. Theorem 5.2 suggests both $FO(Disc, Alg)$ and $FO(Alg, Alg)$ as likely candidates. However, they clearly cannot suffice, since there are uncountably many topological properties of inputs in Alg, but only countably many sentences in these languages. The required topological language is obtained by augmenting $FO(Disc, Alg)$ or $FO(Alg, Alg)$ with *infinitary (countable) disjunction.* Of course, this language is noneffective, as it must express noneffective topological properties (as a matter of fact, even $FO(Disc, Alg)$ and $FO(Alg, Alg)$ alone are noneffective, see Theorem 6.1).

If $\mathcal{F}$ is a countable set of sentences, $\bigvee \mathcal{F}$ denotes the infinitary disjunction of sentences in $\mathcal{F}$. Let $\bigvee FO(Region, Alg)$ be the set of infinitary formulas $\bigvee \mathcal{F}$, where $\mathcal{F}$ is a set of $FO(Region, Alg)$ sentences.

Theorem 5.4 *A property of spatial instances over Alg is a topological property iff it is expressible in $\bigvee FO(Region, Alg)$, for $Region \in \{Alg, Disc\}$.*

Proof All queries $\bigvee FO(Region, Alg)$ are topologically invariant, by Proposition 4.3. For the converse, let τ be a topological property of instances, i.e. a union of classes of topologically equivalent instances. Recall that, by Proposition 5.1, each class e of topologically equivalent instances is definable by a sentence σ_e in $FO(Region, Alg)$. Let $\mathcal{F}$ consist of all sentences σ_e for which $e \subseteq \tau$. Clearly, $\bigvee \mathcal{F}$ defines τ. □

Similarly to Proposition 5.3, we have:

Proposition 5.5 *Every topological property of spatial instances over Alg is expressible in*

$$\bigvee FO(Region, Alg),$$

where Region $\in \{Rect^*, Poly\}$.

In Proposition 5.5 only one direction of the implication holds, because the languages can express non-topological properties. For *Rect*, all topological properties can be expressed in $FO(Rect, Alg)$ augmented with both infinitary disjunctions and conjunctions.

Note that there is an appealing analogy with classical infinitary logic (first-order logic closed under infinitary disjunction and conjunction [BF85]), which can define all properties of finite structures which are invariant under isomorphisms.

Now suppose we are interested in *computable* topological properties. The following refines Theorem 5.4 and Proposition 5.5. *Region* can be any of $\{Disc, Alg, Poly, Rect^*\}$.

Theorem 5.6 *For each recursive topological property* τ *of instances over Alg there exists a recursive set* $\mathcal{F}$ *of* $FO(Region, Alg)$ *sentences such that* $\bigvee \mathcal{F}$ *defines* τ*; moreover, there exists a mapping* f *from instances over Alg to* $FO(Region, Alg)$ *computable in polynomial time, such that: for each instance* I *over Alg,* $I \models f(I)$*, and* $I \models \tau$ *iff* $f(I) \in \mathcal{F}$.

Proof By Theorem 3.5 and Proposition 5.1, for each I in *Alg* one can construct in polynomial time a sentence σ_{T_I} in $FO(Region, Alg)$, defining the equivalence class of I. Let the mapping f be defined by $f(I) = \sigma_{T_I}$; clearly, f is computable in polynomial time and $I \models f(I)$. Let τ be a recursive topological property and $\mathcal{F}$ consist of the sentences σ_{T_I} where $I \models \tau$. By definition, for every I in *Alg*, $I \models \tau$ iff $f(I) \in \mathcal{F}$. Lastly, since τ is recursive, F is recursive: to check if σ is in $\mathcal{F}$, first verify that $\sigma = \sigma_{T_I}$ for some invariant T_I, and if so reconstruct T_I (an examination of the the construction in Proposition 5.1 shows that this reverse engineering can indeed be achieved). Next, use Theorem 3.5 to construct from T_I an instance I' in *Poly* such that $T_I = T_{I'}$, and check that $I' \models \tau$. □

This provides an interesting normal form for computable topological queries. It says that the purely spatial information needed to answer the query can be extracted from the input at some bounded complexity cost (polynomial time); beyond this, the complexity is due to the logical complexity of the query (the complexity of deciding membership in $\mathcal{F}$), independent of the input. Once again, there is an analogy with results within the realm of finite structures. A normal form similar in flavor to the above is shown in [AVV95]. The question of checking a computable property of finite structures expressible in $L^k_{\infty\omega}$ (infinitary logic with k variables [Ba77]) is reduced in polynomial time to checking membership of a sentence in a particular recursive set of FO^k sentences[5].

Relative completeness

Most of the languages previously proposed for spatial databases, including constraint query languages [KKR90, GS94, GST94, Par+95, BDLW95], use variables ranging over *numbers* (reals or rationals), or over *points*. These languages may express non-topological properties. Perhaps surprisingly, we show here that one such point-based language, with data complexity *NC*, expresses

[5] FO^k is first-order with k variables.

precisely the same topological properties as $FO(Rect, Alg)$. This result, proving essentially the equivalence of a point-based language with a region-based language, is somewhat similar to what is done in temporal databases, where languages with temporal predicates are measured against temporal first-order logic that explicitly manipulates temporal variables.

There seem to be two alternative candidates as analogs of first-order temporal logic to the spatial domain: (i) a logic talking about real values and about points as pairs of real values, or (ii) a logic talking explicitly about points in $\mathbb{R}^2$ using point variables. We consider both alternatives, and prove an elegant semantic connection between them, before proceeding with the comparison with region-based languages.

We begin with alternative (i). As before, an input is a spatial instance. Since we must talk about real values as well as regions in the input, we need a two-sorted logic. There are region variables $a, b, c, \ldots$ ranging over names of regions in the input, and variables $x, y, z, \ldots$ ranging over real values. The atoms are of the form $a(x, y), a = A, x < y$, where x, y are real variables, a is a region variable, and $A \in$ **Names**. Boolean operators and quantifiers are standard, with real variables ranging over $\mathbb{R}$ and name variables ranging over the finite set $names(I)$. This language is denoted $FO(\mathbb{R}, <, Region)$, if the inputs are restricted to regions of type *Region*. It is similar to the dense linear-order constraint languages of [GS94, GS95, KG94] but works with $\mathbb{R}$ instead of $\mathbb{Q}$.

For (ii), we use explicit point variables. Instead of the order $<$, we use two order relations $<_x$ and $<_y$ with the following meaning: $p <_x q$ iff the x-coordinate of p $<$ the x-coordinate of q, and similarly for $<_y$. The atoms are $a(p), a = A, p <_x q, p <_y q$, where p, q are point variables, a is a region variable, and $A \in$ **Names**. The boolean connectives and quantifiers are standard, with point variables ranging over the set $\mathbb{P}$ of points in $\mathbb{R}^2$, and region variables ranging over the names of regions in the input. This language is denoted by $FO(\mathbb{P}, <_x, <_y, Region)$, where *Region* designates the type of regions to which inputs are restricted.

It is known that both of the above logics define computable queries with data complexity NC, when inputs are restricted to Alg. This follows from results of [KKR90].

What is the connection between $FO(\mathbb{R}, <, Disc)$ and $FO(\mathbb{P}, <_x, <_y, Disc)$? Clearly, $FO(\mathbb{R}, <, Disc)$ subsumes $FO(\mathbb{P}, <_x, <_y, Disc)$, and can express queries which $FO(\mathbb{P}, <_x, <_y, Disc)$ cannot, such as: "does region A intersect the diagonal ?" as $\exists x.A(x, x)$. To pinpoint the difference between them, we will consider the groups of isomorphisms with respect to which each language is generic: $FO(\mathbb{P}, <_x, <_y, Disc)$ is generic with respect to the group of *monotone isomorphisms*, $\mathcal{M} = \{\lambda \mid \lambda(\langle x, y\rangle) = \langle \rho_1(x), \rho_2(y)\rangle\}$ with $\rho_1, \rho_2 : \mathbb{R} \to \mathbb{R}$ bijective and increasing, while $FO(\mathbb{R}, <, Disc)$ is generic only with respect to the subgroup $\mathcal{M}_0 = \{\lambda \mid \lambda(\langle x, y\rangle) = \langle \rho(x), \rho(y)\rangle\}$, with ρ bijective and increasing. It turns out that $FO(\mathbb{P}, <_x, <_y, Disc)$ expresses *precisely* those queries in $FO(\mathbb{R}, <, Disc)$ that are $\mathcal{M}$-generic. Note that in many applications the origin is chosen arbitrarily, in which case one would only be interested in $\mathcal{M}$-generic queries. We have:

Proposition 5.7

$$FO_{\mathcal{M}}(\mathbb{R}, <, Disc) = FO(\mathbb{P}, <_x, <_y, Disc).$$

Proof The inclusion $FO(\mathbb{P}, <_x, <_y, Disc) \subseteq FO_{\mathcal{M}}(\mathbb{R}, <, Disc)$ is straight forward: every query in $FO(\mathbb{P}, <_x, <_y, Disc)$ can be translated into one in $FO(\mathbb{R}, <, Disc)$, and is obviously $\mathcal{M}$-generic. For the converse inclusion, $FO_{\mathcal{M}}(\mathbb{R}, <, Disc) \subseteq FO(\mathbb{P}, <_x, <_y, Disc)$, let φ be an $\mathcal{M}$-generic query in $FO(\mathbb{R}, <, Disc)$, and suppose φ is given in prenex normal form, $Q_1 z_1.Q_2 z_2 \ldots Q_n z_n.\varphi_0$, where $Q_1, \ldots, Q_n \in \{\forall, \exists\}$ and $\varphi_0(z_1, \ldots, z_n)$ is quantifier free. We will construct a query ψ in $FO(\mathbb{P}, <_x$

$, <_y, Disc)$ that is equivalent to φ. The fact that φ is $\mathcal{M}$-generic plays is used in an essential way in the proof of equivalence. The equivalence does not hold for arbitrary φ in $FO_{\mathcal{M}}(\mathbb{R}, <, Disc)$.

The main idea of the construction of ψ from φ is to "simulate" each real variable z_i used in φ with two point variables, p_i, q_i. Intuitively, p_i plays the role of the point $\langle z_i, 0\rangle$ and q_i that of the point $\langle 0, z_i\rangle$. However, this cannot be literally enforced, since $FO(\mathbb{P}, <_x, <_y, Disc)$ cannot state that the x-coordinate of a point (p_i) equals the y-coordinate of another point (q_i). Getting around this obstacle will require some ingenuity; this is where the $\mathcal{M}$-genericity of φ will be used.

The translation of φ into ψ proceeds as follows. First, quantifiers over a real variable are replaced with quantifiers over pairs of points. This is done by structural induction; the ok_i are conditions to be specified shortly:

- $\exists z_i \xi$ is translated into $\exists p_i . \exists q_i . ok_i \wedge \xi$; and,
- $\forall z_i \xi$ is translated into $\forall p_i . \forall q_i . ok_i \Rightarrow \xi$.

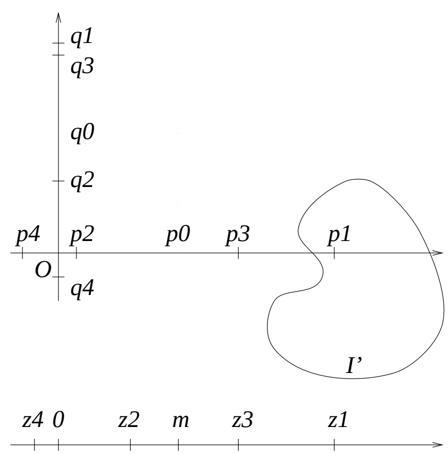
In addition, formula φ starts by specifying a coordinate system: a point O serving as origin, and two points p_0 and q_0 determining the x-axis and y-axis with origin O. For technical reasons, it is also required that the input instance is situated to the right of p_0 and below q_0. In summary, the query ψ is of the form:

$$\exists O . \exists p_0 . \exists q_0 . ok_0 \wedge (Q_1 p_1 . Q_1 q_1 . ok_1 op_1 (Q_2 p_2 . Q_2 q_2 . ok_2 op_2 (\ldots (Q_n p_n . Q_n q_n . ok_n op_n \psi_0) \ldots))) \qquad (3)$$

where op_i is $\wedge$ when Q_i is $\exists$, and op_i is $\Rightarrow$, when Q_i is $\forall$. Condition ok_0 checks that O, p_0, q_0 define a coordinate system with O, p_0 in the same order on Ox as O, q_0 on Oy ($O \neq p_0 \wedge O \neq q_0 \wedge O =_y p_0 \wedge O =_x q_0 \wedge (O <_x p_0 \Leftrightarrow O <_y q_0)$) and that the entire instance lies to the right of p_0 and below q_0 ($\bigwedge_{a \in names(I)} (\forall r . a(r) \Rightarrow (p_0 <_x r \wedge r <_y q_0))$). After each quantifier Q_i, condition ok_i checks that (1) p_i is on the Ox axis and q_i on the Oy axis, and (2) the order relation of $O, p_0, \ldots, p_i$ on the Ox axis is the same as that of $O, q_0, \ldots, q_i$ on the Oy axis. Finally ψ_0 contains a translation of φ_0, in which each atomic predicate $a(z_i, z_j)$ is replaced with $\exists r . (r =_x p_i \wedge r =_y q_j \wedge a(r))$, and each atomic predicate $z_i < z_j$ with $p_i <_x p_j$ (there is no need to check $q_i <_y q_j$, because p_i, p_j and q_i, q_j are in the same order).

We show next that φ and ψ are equivalent. First we show that $I \models \varphi$ implies $I \models \psi$. The idea is that the variable p_i corresponds to $\langle z_i, 0\rangle$, and q_i to $\langle 0, z_i\rangle$. However, as indicated earlier, we cannot enforce simultaneously $p_i = \langle z_i, 0\rangle$ and $q_i = \langle 0, z_i\rangle$, because in $FO(\mathbb{P}, <_x, <_y, Disc)$ we cannot impose x coordinates to be equal with y coordinates. Here we use the fact that φ is $\mathcal{M}$-generic. Let $m \in \mathbb{R}$ be some number, and $\lambda : \mathbb{P} \to \mathbb{P}$ be a translation (which is a monotone isomorphism) such that $\lambda(I)$ lies inside $(m, \infty) \times (-\infty, m)$. Denoting $I' = \lambda(I)$, we have $I' \models \varphi$. Intuitively, we will match those z_i's which are $< m$ with q_i and those $\geq m$ with p_i. Formally, let $z_{i_1}, \ldots, z_{i_k}$ be the universally quantified variables in φ. Since $I' \models \varphi$, there exists Skolem functions $z_i = f_i(z_{i_1}, z_{i_2}, \ldots)$[6] for all existentially quantified variables z_i, such that $I' \models \varphi^*(z_{i_1}, \ldots, z_{i_k})$ for all $z_{i_1}, \ldots, z_{i_k}$. Here φ^* is the *functional closure* of φ_0, i.e. in which each existential variable is substituted with its Skolem function. From the f_i's, we will derive Skolem functions for the existential variables p_i, q_i in ψ: $p_i = g_i(p_{i_1}, q_{i_1}, p_{i_2}, \ldots)$ and $q_i = h_i(p_{i_1}, q_{i_1}, p_{i_2}, \ldots)$, such that $I' \models \psi^*(p_{i_1}, q_{i_1}, \ldots, p_{i_k}, q_{i_k})$ for all points $p_{i_1}, \ldots, q_{i_k}$. We illustrate in Figure 13. First we pick

[6] The Skolem function $f_i(z_{i_1}, \ldots, z_{i_j})$ depends on those universal variables whose index is $< i$, i.e. $i_1 < i_2 < \ldots < i_j < i < i_{j+1}$.



$$
\begin{array}{ccccccccccc}
p_4 & <_x & O & <_x & p_2 & <_x & p_0 & <_x & p_3 & <_x & p_1 \\
q_4 & <_y & O & <_y & q_2 & <_y & q_0 & <_y & q_3 & <_y & q_1 \\
z_4 & < & 0 & < & z_2 & < & m & < & z_3 & < & z_1
\end{array}
$$

$$
\begin{array}{rcl}
q_4 & = & \langle 0, z_4 \rangle \\
O & = & \langle 0, 0 \rangle \\
q_2 & = & \langle 0, z_2 \rangle \\
q_0 & = & \langle 0, m \rangle \\
p_0 & = & \langle 0, m \rangle \\
p_3 & = & \langle 0, z_4 \rangle \\
p_1 & = & \langle 0, z_1 \rangle
\end{array}
$$

Figure 13: Illustration for the proof of Proposition 5.7.

$O = \langle 0, 0\rangle, p_0 = \langle m, 0\rangle, q_0 = \langle 0, m\rangle$: obviously ok_0 holds. Next assume $p_{i_1}, q_{i_1}, p_{i_2}, q_{i_2}, \ldots$ be given, such that (1) all the p-points are on the Ox axis, and all the q-points on the Oy axis, and (2) their order relations on the two axis, together with O, p_0, q_0, agree: if these two conditions do not hold, then $I' \models \psi^*(p_{i_1}, q_{i_1}, \ldots, p_{i_k}, q_{i_k})$ holds vacuously, because of the ok_i predicates after the universal quantifiers (see 3). First define $z_{i_1}, \ldots, z_{i_k}$ as follows: if $p_{i_j} \geq_x p_0$ then z_{i_j} is the x-coordinate of p_{i_j}, else it is the y-coordinate of q_{i_j}. Compute now $z_i = f_i(z_{i_1}, z_{i_2}, \ldots)$. Case 1: $z_i < m$. Then define $q_i = h_i(p_{i_1}, q_{i_1}, p_{i_2}, \ldots) \stackrel{\text{def}}{=} \langle 0, z_i\rangle$, and define $p_i = g_i(p_{i_1}, q_{i_1}, p_{i_2}, \ldots)$ to be some point on the Ox axis such that the order of $O, p_0, p_1, \ldots, p_i$ coincides with that of $O, q_0, q_1, \ldots, q_i$. Case 2: $z_i \geq m$. Then define $p_i = g_i(p_{i_1}, q_{i_1}, p_{i_2}, \ldots) \stackrel{\text{def}}{=} \langle z_i, 0\rangle$, and define q_i to be any point on the Oy axis preserving the order. This concludes the construction of the Skolem functions for ψ. We will show now that $I' \models \psi^*$. Let $O, p_0, q_0, p_1, q_1, \ldots, p_n, q_n$ be the universal variables, plus the existential ones computed by the new Skolem functions. Let z_i be the x-coordinate of p_i, when $p_i \geq_x p_0$, and the y-coordinate of q_i otherwise. Then the order of $O, p_0, p_1, \ldots, p_n$ on the Ox axis is the same as that of $O, q_0, q_1, \ldots, q_n$ on the Oy axis, and the same as that of $0, m, z_1, \ldots, z_n$. Moreover, for any existential variable z_i, we have $z_i = f_i(z_{i_1}, z_{i_2}, \ldots)$. We will show that all atomic predicates in φ_0 have the same truth value as their translations in ψ_0. Obviously $z_i < z_j$ iff $p_i <_x p_j$. Consider $a(z_i, z_j)$. Case 1: $z_i \geq m$ and $z_j < m$. Then $p_i = \langle z_i, 0\rangle, q_j = \langle 0, z_j\rangle$, and $a(z_i, z_j)$ is equivalent to its translation $\exists r.(r =_x p_i \wedge r =_y q_j \wedge a(r))$. Case 2: $z_i < m$. Here we don't have $p_i = \langle z_i, 0\rangle$, but both $a(z_i, z_j)$ and $\exists r.(r =_x p_i \wedge r =_y q_j \wedge a(r))$ are *false* in I'. Case 3: $z_j \geq m$ is similar. In conclusion, since $I' \models \varphi^*(z_{i_1}, \ldots, z_{i_k})$, we have $I' \models \psi^*(p_{i_1}, q_{i_1}, \ldots, p_{i_k}, q_{i_k})$. This proves $I' \models \psi$, which implies $I \models \psi$, because ψ is $\mathcal{M}$-invariant.

Conversely, suppose $I \models \psi$. Recall that ψ is of the form $\exists O.\exists p_0.\exists q_0.ok_0 \wedge \psi_1(O, p_0, q_0)$, see (3). Then there exists three points O, p_0, q_0 such that ok_0 and $I \models \psi_1(O, p_0, q_0)$. Let $m \in \mathbb{R}$ be some arbitrary number such that $m > 0$ iff $p_0 >_x O$ (and iff $q_0 >_y O$). Then there exists a monotone isomorphism λ mapping the points O, p_0, q_0 to $O' = \langle 0, 0\rangle, p_0' = \langle m, 0\rangle, q_0' = \langle 0, m\rangle$. Let $I' = \lambda(I)$: we have $I' \models \psi_1(O', p_0', q_0')$. This follows from a slight generalization of $\mathcal{M}$-genericity of $FO(\mathbb{P}, <_x, <_y, Disc)$ queries. Moreover, as before, I' lies inside $(m, \infty) \times (-\infty, m)$. Now we can repeat the above argument in reverse, constructing the Skolem functions f_i from the Skolem functions g_i, h_i: as before, we compute p_i, q_i first, and define z_i to be the x-coordinate of p_i, if $p_i \geq_x p_0$, and to be the y-coordinate of q_i, if $p_i <_x p_0$. We conclude that $I' \models \varphi$ and, hence, that $I \models \varphi$. □

Note that, as a consequence of Proposition 5.7, $FO(\mathbb{R}, <, Disc)$ and $FO(\mathbb{P}, <_x, <_y, Disc)$ express the same set of topological queries and the same set of $\mathcal{S}$-generic queries.

We next show the connection between $FO(\mathbb{P}, <_x, <_y, Disc)$ and $FO(Rect, Disc)$. It is easy to see that $FO(Rect, Disc) \subset FO(\mathbb{P}, <_x, <_y, Disc)$. We can show the following, which provides the promised relative completeness result. Recall that $FO(Rect, Disc)$ queries are $\mathcal{S}$-generic, but not necessarily $\mathcal{H}$-generic, and let $FO_{\mathcal{S}}(\mathbb{P}, <_x, <_y, Disc)$ be the set of $\mathcal{S}$-generic queries expressible in $FO(\mathbb{P}, <_x, <_y, Disc)$.

Theorem 5.8 *$FO(Rect, Disc) = FO_{\mathcal{S}}(\mathbb{P}, <_x, <_y, Disc)$. In particular,*

$$FO(Rect, Disc) \text{ and } FO(\mathbb{P}, <_x, <_y, Disc)$$

express precisely the same topological properties.

Proof (sketch) One direction is easy: every query in $FO(Rect, Disc)$ can be expressed in $FO(\mathbb{P}, <_x, <_y, Disc)$ by simply replacing every rectangle variable r with two point variables, representing two opposite corners of r. For the other direction we have to talk about points in a language in which we only have rectangles. First recall from the proof of the Theorem 4.4 that $FO(Rect, Disc)$ can express the predicates $edge(r, r')$, meaning that the rectangles r, r' meet at an edge, and $corner(r, r')$, meaning that r, r' meet at a corner only, see Figure 14. Next, define $one\_edge(r, r')$ to mean that r, r' meet and share a complete edge (including two corners). Then every query in $FO_{\mathcal{S}}(\mathbb{P}, <_x, <_y, Disc)$ is translated into a query which starts as follows: $\exists r_1, r_2, r_3, r_4 . one\_edge(r_1, r_2) \wedge one\_edge(r_2, r_3) \wedge one\_edge(r_3, r_4) \wedge one\_edge(r_4, r_1) \wedge \ldots$ Here r_1, r_2, r_3, r_4 serves as a system of coordinates. We leave out the remaining lengthly but straightforward details of the translation. Finally, note that we have no control over how the rectangles r_1, r_2, r_3, r_4 are chosen, but the fact that the query is $\mathcal{S}$-generic guarantees that this does not matter. $\square$

6 Decidability and Complexity

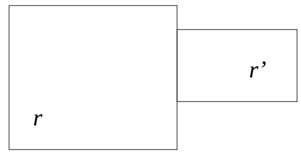
Effective evaluation of queries in the region-based languages only makes sense when inputs are finitely specified. Among the classes of regions we considered, $Rect, Rect^*, Poly$ and Alg can be finitely specified[7]. These are the input classes we shall consider in this section (quantifiers can again range over any class, finitely specifiable or not). Once we have finitely specified inputs, the notion of decidable query, and that of the data and query complexity of a query are standard. In particular, data complexity is with respect to the size of a standard encoding of an input I, which combines two factors: the size of $names(I)$ and the size of the specifications of the regions in $names(I)$. These components are to some extent orthogonal, and it is sometimes useful to consider them separately (this is done, for example, in Theorem 6.1 below).

We start with undecidability results. To succinctly state our results we shall use expressions such as $FO(Rect^* - Disc, Rect^* - Alg)$ to denote, in this example, all classes of first order queries whose quantification range is between $Rect^*$ and $Disc$ (that is, $Rect^*$, $Poly$, Alg, or $Disc$) and whose input domain is between $Rect^*$ and Alg (that is, $Rect^*$, $Poly$, or Alg). Recall from [Rog] that the *arithmetical hierarchy* is the set of properties over $\mathbb{N}$ which can be expressed in the *elementary arithmetic*, i.e. in first order logic over the symbols $+, \times, 0, 1, 2, 3, \ldots$ This includes all recursive properties[8], but many non-recursive properties as well. To see that, recall that every property $P(x)$ in the arithmetical hierarchy can be expressed either as $\exists x_1 . \forall x_2 . \exists x_3 . \ldots Q_n x_n . R(x, x_1, \ldots, x_n)$, or as $\forall x_1 . \exists x_2 . \forall x_3 . \ldots Q_n x_n . R(x, x_1, \ldots, x_n)$, where R is a recursive predicate: the class Σ_n consists of all properties of the first form, and Π_n of all those of the second form. Then $\Sigma_0 = \Pi_0$ is the class of recursive, Σ_1 that of recursively enumerable, and Π_1 of co-recursively enumerable properties. Moreover $\Sigma_0 \subset \Sigma_1 \subset \Sigma_2 \subset \ldots$ and $\Pi_0 \subset \Pi_1 \subset \Pi_2 \subset \ldots$ form two strict hierarchies, whose limit is the arithmetical hierarchy, denoted AH.

Also, recall [Rog] that the *analytical hierarchy*, denoted here AnH, is the set of properties over $\mathbb{N}$ expressible in second order logic over the symbols $+, \times, 0, 1, 2, 3, \ldots$, in which the second order variables range over functions $f : \mathbb{N} \to \mathbb{N}$. The analytical hierarchy contains even more properties

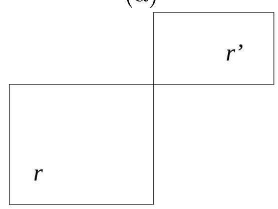
[7] For $Rect, Rect^*$ we restrict inputs to rational coordinates.

[8] By a theorem due to Gödel, every recursive relation $R(x)$ can be expressed as $Q_1 y_1 . Q_2 y_2 . \ldots Q_n y_n . P(x, y_1, \ldots, y_n) = 0$, where $Q_i \in \{\forall, \exists\}$, for $i = 1, \ldots, n$, and P is a polynomial with integer coefficients.



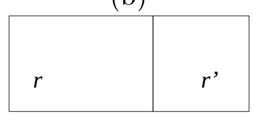
$$
\begin{aligned}
edge(r,r') \quad &\equiv \quad meet(r,r') \wedge \\
&\qquad \exists r''.(overlap(r,r'') \wedge overlap(r',r''))
\end{aligned}
$$

(a)



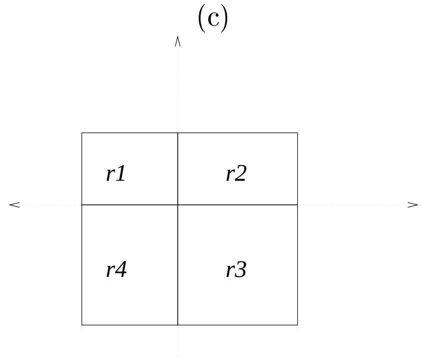
$$corner(r,r') \equiv meet(r,r') \wedge \neg(edge(r,r'))$$

(b)



$$
\begin{aligned}
one\_edge(r,r') \quad &\equiv \quad edge(r,r') \wedge \\
&\qquad (\exists q.\exists q'.(edge(r,q) \wedge edge(r,q') \\
&\qquad \wedge edge(r',q) \wedge edge(r',q') \\
&\qquad \wedge \neg(connect(q,q'))))
\end{aligned}
$$

(c)



Four rectangles define a coordinate system.
(d)

Figure 14: Illustrations for the proof of Theorem 5.8.

than the arithmetical one. An example of a property in AnH but not in AH is:

$$\{x \mid \text{“}x \text{ is the encoding of a true formula in elementary arithmetic”}\}$$

As usual, we extend the notations AH, AnH to instances in *Region*. Namely some query Q of instances $I \in Alg$ is in AH (AnH) iff the set of encodings $\{encode(I) \mid I \in Q\}$ is in AH (or AnH respectively). Of course, both definitions depend on *Region*, which also has to be finitely representable. For some group of permutations G, we define AH_G, AnH_G to be the restrictions of AH, AnH to G-generic queries.

Theorem 6.1 (Undecidability) *The query classes $FO(Rect^* - Disc, Rect - Alg)$ are undecidable. Moreover, assuming that we restrict our database instances to have fixed set of names, $names(I) = \{A_1, \ldots, A_k\}$, we have the following characterization of their expressive power:*

- $FO(Alg, Rect - Alg) = AH_{\mathcal{H}}$, $FO(Poly, Rect - Poly) = AH_{\mathcal{H}}$.
- $AH_{\mathcal{H}} \subseteq FO(Rect^*, Poly - Alg) \subseteq AH$, $AH_{\mathcal{H}} \subseteq FO(Poly, Alg) \subseteq AH$.
- $FO(Rect^*, Rect - Rect^*) = AH_{\mathcal{S}}$.
- $AnH_{\mathcal{H}} \subseteq FO(Disc, Rect - Alg)$.

Proof We start by showing $FO(Alg, Alg) \subseteq AH$. Any region variable $r \in Alg$ can be encoded as a finite set of finite sets of finite sets of integers[9] and, hence, as a single number. Thus, quantifiers over Alg become quantifiers over numbers. Also, it is easy to see that $connect(r, r')$ is expressed as a recursive property on the encodings of r, r'. Next we show $AH_{\mathcal{H}} \subseteq FO(Alg, Alg)$. Here the crucial observation is that we can encode numbers, and the operations $+, \times, =$ in $FO(Alg, Alg)$. Indeed, we express a natural number x as two regions $r, q \in Alg$ s.t. $r \cap q$ has x connected components. We express equality of two numbers x, x' represented by r, q, r', q' informally as follows: “there exists a finite set of regions $S \subseteq Alg$ which is a *one-to-one relation* from the connected components of $r \cap q$ to those of $r' \cap q'$, i.e.: (1) for every connected component t of $r \cap q$ there exists a unique connected component t' of $r' \cap q'$ such that both t and t' are contained in some set $s \in S$, and (2) for every connected component t' of $r' \cap q'$ there exists a unique connected component t of $r \cap q$ such that t, t' are contained in some set $s \in S$”. A finite set of regions S is encoded as the intersection of two regions in Alg. The addition predicate $x'' = x + x'$ is expressed similarly: that is, assuming the connected components of $r \cap q$ and $r' \cap q'$ to be pairwise disjoint, we search for a one-to-one relation S from $r'' \cap s''$ and the union of the connected components of $r \cap s$ and $r' \cap s'$. For multiplication, $x'' = x \times x'$, we search for many-to-one relations S, S' from x to x'' and x' to x'' respectively, and require the relation $\{((c, c'), c'') \mid (c, c'') \in S, (c', c'') \in S'\}$ to be one-to-one, where c, c', c'' are connected components of $r \cap s, r' \cap s', r'' \cap s''$ respectively. Hence, it follows that we can express in $FO(Alg, Alg)$ all number-theoretic properties in AH. Back to the proof of $AH_{\mathcal{H}} \subseteq FO(Alg, Alg)$, suppose we are given a formula $\varphi \in AH$ expressing a topological property over the number encoding of some input $I \in Alg$. We have to construct some formula $\psi \in FO(Alg, Alg)$, having I as input. We do this in two steps. First, given I we compute some encoding of the invariant T_I. Namely, following the proof of Proposition 5.1, the invariant T_I can be represented by $|V|$ regions, $r_1, \ldots, r_{|V|}$, where V is the set of cells in T_I, and a number of additional

[9] Recall that a region in Alg is expressed as $\bigvee_i \bigwedge_j P_{ij}(x, y) > 0$, where P_{ij} is a polynomial with integer coefficients.

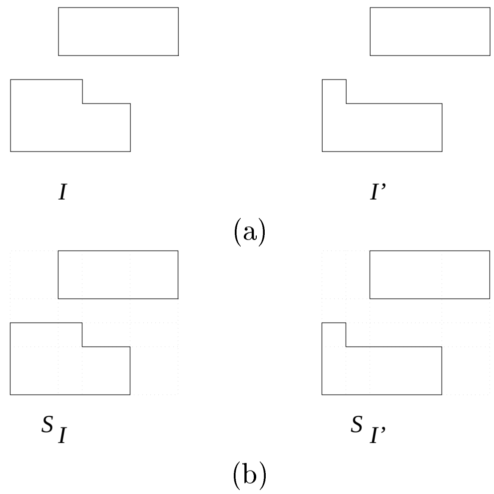


Figure 15: Illustrations of S_I. Two instances I, I' which are not $\mathcal{S}$-equivalent, (a), and their invariants $S_I, S_{I'}$, (b).

regions corresponding to the the other components of T_I, such as the labeling relation l and the orientation O. Although the number $|V|$ and other components of this representation depend on I, we can encode a set of regions as the connected components of the intersection of two regions, and it follows that the entire invariant T_I can be encoded by a fixed number of regions $r_1, \ldots, r_k$ (the number k depends only on the size of $names(I)$, not on I). In short, we can extract a number encoding of T_I, and then search for a number x representing an encoding of a algebraic instance I' for which T_I and $T_{I'}$ are isomorphic (recall that $T_{I'}$ can be computed from I' in NC). Finally we check $\varphi(x)$. In summary, we have so far proven that $FO(Alg, Alg) = AH$. Of course, this implies that $FO(Alg, Rect - Alg) = AH_{\mathcal{H}}$.

$FO(Poly, Poly) = AH_{\mathcal{H}}$ and the relationships $AH_{\mathcal{H}} \subseteq FO(Rect^*, Poly - Alg) \subseteq AH$ and $AH_{\mathcal{H}} \subseteq FO(Poly, Alg) \subseteq AH$ are proven similarly. Note that $FO(Rect^*, Poly - Alg)$ and $FO(Poly, Alg)$ can express non-topological queries so we do not have equality with $AH_{\mathcal{H}}$.

For $FO(Rect^*, Rect - Rect^*) = AH_{\mathcal{S}}$, we proceed as for $FO(Alg, Alg) = AH_{\mathcal{H}}$. We need however to generalize Theorem 3.4 as follows. For any instance I in $Rect^*$, we construct (in NC) an invariant S_I such that I is $\mathcal{S}$-equivalent to I' iff S_I is isomorphic to $S_{I'}$. Here S_I has to capture in addition to the information in T_I, information about other vertices, edges, and faces generated by all points obtained as intersections of vertical and horizontal lines, see Figure 15. Then the proof proceeds as for $FO(Alg, Alg) = AH_{\mathcal{H}}$.

Finally, we sketch how to show $AnH_{\mathcal{H}} \subseteq FO(Disc, Rect - Alg)$. First we notice that here $r \cap q$ may have, in general, infinitely many connected components, hence not every pair r, q is a correct encoding of a number. However we can test in $FO(Disc, Rect - Alg)$ whether $r \cap q$ has finitely many connected components, by checking that there is no one-to-one relation from the set of connected components of $r \cap q$ to a proper subset of it. So we proceed as in the proof of $AH_{\mathcal{H}} \subseteq FO(Alg, Alg)$: it suffices to observe that $FO(Disc, Alg)$ can encode functions $f : \mathbb{N} \to \mathbb{N}$: see Figure 16. $\square$

Keeping fixed the set of region names has not as much to do with the complexity of the queries, as with the generic nature of the fragment of our languages dealing with region names. E.g. the

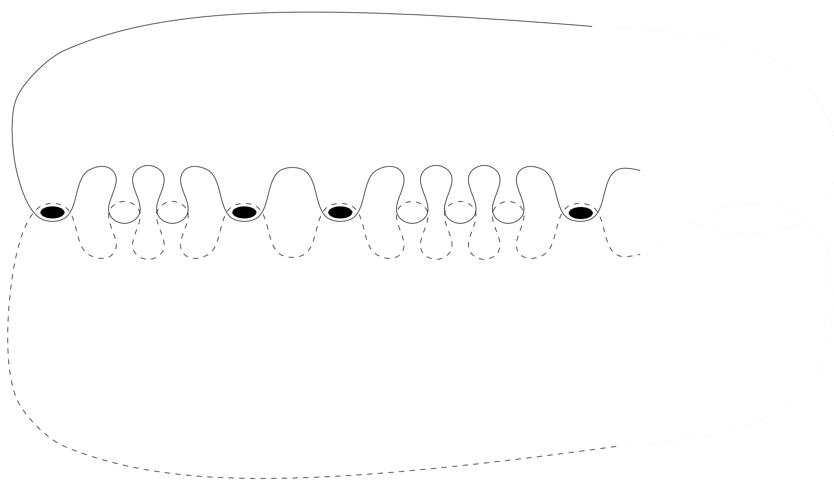


Figure 16: Encoding of the function $f : \mathbb{N} \to \mathbb{N}$, $f(0) = 2, f(1) = 0, f(2) = 3, \ldots$ The two regions r, q yield infinitely many connected components in $r \cap q$. The black regions are the connected components of the intersection of two other regions, $r' \cap s'$, not shown.

following property of instances I is not expressible in $FO(Disc, Alg)$: "the cardinality of $names(I)$ is even". On the other hand, $FO(Disc, Alg)$ can express the property "$\{ext(a) \mid a \in names(I)\}$ consists of an even number of disjoint regions". Thus, the characterizations do not work if the set of names is allowed to vary.

In view of the very strong undecidability results in this section, we are motivated to consider decidable fragments of topological query languages. Unfortunately, we next note that even extremely restricted fragments of these languages can be undecidable. Let us consider the query class $\Sigma_1(Rect^*, \emptyset)$: existential sentences with quantifiers over finite unions of rectangles, and not involving any input regions. As it turns out, whether this theory is decidable is equivalent to a fairly well-known and -looked at problem in graph theory, the *string graph problem* [EET76, Kra91, KM94], which can be restated as follows: *Given a graph G, is it the intersection graph of a set of curves on the plane?* That is, is there a set of curves (non-self-intersecting paths on the infinite two-dimensional grid, say), one corresponding to each node, such that two curves intersect if and only if the corresponding vertices are connected in G? It is a fairly well-known open problem whether string graphs are decidable.

Proposition 6.2 $\Sigma_1(Rect^*, \emptyset)$ *is decidable if and only if string graphs are decidable.*

Proof To translate between curves and finite unions of rectangles, in one direction we replace a region by a curve that "fills it", in the other we approximate a curve as closely as desired by a finite union of "very thin" rectangles. Thus the string graph problem is the special case of the $\Sigma_1(Rect^*, \emptyset)$ sentences with matrix which is the conjunction of literals of the form $connect(A, B)$ or $\neg connect(A, B)$, one for each pair of quantified variables. Conversely, with exponentially many calls to an algorithm for this fragment (one for each satisfying truth assignment of the matrix of the $\Sigma_1(Rect^*, \emptyset)$ sentence) we can answer any query. $\square$

This result, together with the negative results in [Kra91, KM94] (NP-hardness of string graph recognition, and an exponential lower bound on the size of the smallest string model of a graph) has the following negative consequences:

Corollary 6.3 *$\Sigma_1(Rect^*, \emptyset)$ is NP-hard in terms of query complexity. Furthermore, for every n there is a true $\Sigma_1(Rect^*, \emptyset)$ query with n symbols such that the smallest set of regions that can instantiate the existential quantifiers has total size that is* doubly exponential.

We now turn to the decidable sublanguages, and their data complexity. Since queries in $FO(Region, Alg)$ involve quantification over the infinite sets in *Region*, computing such queries must involve results along the lines of the well-known quantifier elimination theorems *à la* [Tar51]. Indeed, the following is a direct consequence of results in [KKR90].

Theorem 6.4 (Data Complexity) *The queries in $FO(Rect, Rect - Alg)$ are effectively computable, and have data complexity NC. Similarly for queries in $FO(\mathbb{P}, <_x, <_y, Rect - Alg)$.*

Finally, in terms of query complexity we have the following (we omit the straightforward proofs).

Theorem 6.5 (Query Complexity) *The query complexity of $FO(Rect, Rect-Alg)$ and $FO(\mathbb{P}, <_x, <_y, Rect - Alg)$ is PSPACE. The query complexity of the* conjunctive queries *(existential queries with conjunctive matrix) in $FO(Rect, Rect^* - Alg)$ and $FO(\mathbb{P}, <_x, <_y, Rect^* - Alg)$ is NP, and the query complexity of conjunctive queries in $FO(Rect, Rect)$ and $FO(\mathbb{P}, <_x, <_y, Rect)$ is NC.*

7 Conclusions and Further Work

We have made progress towards the theoretical underpinnings of the topological fragment of spatial databases. We have identified the essential part of a spatial data base vis-á-vis topological queries: It is the planar graph structure of the region boundaries. This structure can be represented as an instance of a simple and fixed relational scheme, and can be extracted easily from any spatial instance. The resulting relational instance can then be used to answer any topological query on the original spatial instance. It would be very interesting to find other domains in which this methodology is useful; this line of research can result in database systems for extremely complex data which, however, also contain simple relational "summaries" that suffice for large ranges of interesting queries.

We have also introduced natural topological languages, first-order extensions of the well-known but insufficient Egenhofer-Franzosa relations, and discussed their expressibility, completeness, decidability, and complexity. We have mostly restricted ourselves to the domain *ALG*; similar results can be obtained for the $\mathcal{S}$-generic properties of inputs over $Rect^*$ (as outlined in the proof of Theorem 6.1), and for $\mathcal{L}$-generic properties of inputs over *Poly*. This includes the existence and efficient construction of the respective invariants, and the definability of $\mathcal{S}$-equivalence classes and $\mathcal{L}$-equivalence classes by sentences in $FO(Rect^*, Rect^*)$ and $FO(Poly, Poly)$, respectively. The latter can also be used to obtain completeness results in the spirit of those of Section 5, only this time for the $\mathcal{S}$-generic and $\mathcal{L}$-generic properties of inputs over $Rect^*$ and *Poly*.

One important question has been left open in this paper: Are there interesting tractable topological query languages[10] that are more expressive than the Boolean closure of the Egenhofer-Franzosa relations? In work currently in progress [PSV97], we have identified an interesting candidate: The first order closure of Egenhofer-Franzosa relations where the quantifiers are restricted to two kinds of regions: (a) One weak class of quantifiers range over all *cells* of the plane as divided by the

[10] Among the languages we considered, $FO(Rect, Alg)$ is tractable, but is not a purely topological language.

boundaries of the instance; and (b) A stronger quantifier ranges over all possible unions of cells that are disc homeomorphs (that is to say, legitimate regions); a related language quantifies over points and paths. The first order closure of Egenhofer-Franzosa relations under these two kinds of quantifiers seems to be very expressive —for example, the queries exemplified in Figure 1 (a) vs. (b) and (c) vs. (d), which are known not to be expressible by Boolean combinations of the Egenhofer-Franzosa relations, are expressible when these quantifiers are allowed. The data complexity of this language is NC, and the query complexity PSPACE. Finally, we hope to discover clean characterizations of the expressive power of this language, and perhaps obtain a purely algebraic counterpart in the spirit of Codd's Theorem.

References

[AVV95] S. Abiteboul, M. Vardi and V. Vianu. Computing with infinitary logic. *Theoretical Computer Science*, to appear. Extended abstract in *Proc. Int. Conf. on Database Theory*, 1992.

[BDLW95] M. Benedikt, G. Dong, L. Libkin and L. Wong. Relational expressive power of constraint query languages. Manuscript, 1995.

[Ba77] J. Barwise. On Moschovakis closure ordinals. J. Symbolic Logic, 42 (1977), pp. 292-296.

[BF85] Barwise, J., S. Feferman (eds.). Model-Theoretic Logics. Springer-Verlag, 1985.

[BKR84] M. Ben-Or,D. Kozen, and J. Reif. The complexity of elementary algebra and geometry. In *Proc. 16th ACM Symp. on Theory of Computing*, 1984, 457–464.

[Cla85] B. Clark. A calculus of individuals based on 'connection'. *Notre Dame Journal of Formal Logic*, 22(3), 1985.

[CRC94] A.G. Cohn, D.A. Randell, and Z. Cui. Taxonomies of Logically Defined Qualitative Spatial Relations. N. Guarino, and R. Poli (eds), *Formal Ontology in Conceptual Analysis and Knowledge Representation*, Kluwer 1994.

[Cor79] J.P. Corbett. Topological principles in cartography. Technical Paper 48, US Bureau of the Census, Washington D.C., 1979.

[EgFra] M.J. Egenhofer and R. Franzosa. Point-set topological spatial relations. *Int. J. Geo. Info. Sys.* **5(2)**, 1991, 161–174.

[EET76] G. Ehrlich, S. Even, R. E. Tarjan Intersection graphs of curves on the plane *J. Combinatorial Theory Ser. B*, 21 (1976), pp. 8–20.

[Fá48] I. Fáry On straight line representations of planar graphs *Acta Scien. Math. Szeged., 11*, pp. 229–233, 1948.

[GPP95] M. Grigni, D. Papadias and C.H. Papadimitriou. Topological Inference *IJCAI*, 1995.

[GS94] S. Grumbach and J. Su. Finitely representable databases. In Proc. ACM Symp. on Principles of Database Systems, 1994, 289–300.

[GS95] S. Grumbach and J. Su. Dense-order Constraint Databases. In Proc. ACM Symp. on Principles of Database Systems, 1995, 66-77.

[GST94] S. Grumbach, J. Su and C. Tollu. Linear constraint databases. In *Proc.Logic and Complexity Workshop*, D. Leivant ed., Indiana, 1994, to appear.

[JS85] J. Já Já, J. Simon Parallel algorithms for graph theory: planarity testing. In *SIAM J. Comp., 11*, pp. 314–328, 1982.

[KG94] P.C. Kanellakis and D. Q. Goldin. Constraint programming and database query languages. In Proc. 2nd Conf. on Theoretical Aspects of Computer Software (TACS), 1994, Sendai, Japan, LNCS 789, 96-120.

[KKR90] P.C. Kanellakis, G.M. Kuper and P.Z. Revesz. Constraint query languages. In Proc. ACM Symp. on Principles of Database Systems, 1990, 299–313.

[KY85] D. Kozen and C-K. Yap. Algebraic Cell Decomposition in NC. In *Proc. IEEE Symp. on Foundations of Computer Science*, 1985, 515-521.

[Kra91] J. Kratochvíl. String graphs II: Recognizing string graphs is *NP*-hard. *J. Combinatorial Theory Ser. B*, 52 (1991), pp. 67–78.

[KM94] J. Kratochvíl, Jiři Matoušek. Intersection graphs of segments. *J. Combinatorial Theory Ser. B*, 62 (1994), pp. 289–315.

[KPV95] B. Kuijpers, J. Paredaens and J. Van den Bussche. Lossless Representation of Topological Spatial Data. In *Proc. Fourth Int'l. Symp. on Large Spatial Databases*, 1995.

[LT92] R. Laurini and D. Thompson. Fundamentals of Spatial Information Systems. Academic Press, APIC Series 37, 1992.

[Moi] E. Moise. Geometric topology in dimensions 2 and 3. Springer Verlag, 1977.

[Par+94] J. Paredaens, J. Van Den Bussche and D. Van Gucht. Towards a theory of spatial database queries. In Proc. ACM Symp. on Principles of Database Systems, 1994, 279–288.

[Par95] J. Paredaens. Spatial Databases, the Final Frontier. In *Proc. 5th Int'l. Conf. on Database Theory*, 1995, 14–32.

[Par+95] J. Paredaens, J. Van Den Bussche and D. Van Gucht. First-order queries on finite structures over the reals. In *Proc. Conf. on Logic in Computer Science*, 1995.

[PSV97] C.H. Papadimitriou D. Suciu, V. Vianu Tractable topological query languages. Work in progress.

[RC89] D.A. Randell, and A.G. Cohn. Modelling topological and metrical properties of physical processes. R. Brachman, H. Levesque, and R. Reiter (eds), *Proceedings 1st International Conference on the Principles of Knowledge Representation and Reasoning*, Morgan Kaufmann, 1989.

[Rog] H. Rogers, Jr. Theory of Recursive Functions and Effective Computability. MIT Press, 1987.

[SW94] D.L. Souvaine, R. Wenger. Constructing Piecewise Linear Homeomorphisms. DIMACS Technical Report 94-52, December 1994.

[St51] S. K. Stein Convex maps. in *Proc. Amer. Math. Soc., 2*, pp. 464–466, 1951.

[St93] J. Stillwell. Classical topology and combinatorial group theory. Springer Verlag, 1993.

[Tar51] A. Tarski. A Decision Method for Elementary Algebra and Geometry. University of California Press, 1951.

[Zie85] Topological Model Theory. In *Model-Theoretic Logics*, ed. J.Barwise and S.Feferman, Springer Verlag, 1985.

[Wag36] K. Wagner Bemerkungen zum Vierarbenproblem. *Jber. Deutsch. Math. verein., 46*, pp. 26–32, 1936.