Topological Queries in Spatial Databases

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

Abstract

We study query language 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. Finally we show that each set of semi-algebraic regions is characterized up to homeomorphism by an invariant representable as a finite structure, computable in NC. This allows to answer all topological queries on semi-algebraic regions 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.

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 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).

In terms of query languages, one would like to have intuitive, natural primitives that are geared towards the information of interest. For example, if the topological aspect is the one 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 a particular set of G-generic queries, several questions come up: (i) is there a sense in which the language is complete with respect to the target class of queries? If not, does it represent some significant fragment? (ii) what are the appropriate representations of spatial information so that G-generic queries can be readily answered?

We consider such questions starting from a well-known set of natural language constructs proposed by

* 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.

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 both that the interiors 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 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 consider 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 these various languages, as well as their relative expressiveness.

Spatial databases (and more generally multimedia databases) have to manage a mix of classical database information (referred to sometimes as *thematic*) 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. We investigate this question for topological queries, and show that the topological properties of semi-algebraic spatial regions can be completely specified using a classical finite structure. 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. Closest to our 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.

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]. They can be viewed as an augmentation of the PLA model.

Related work. Work in spatial databases has focused on developing models and query languages targeted to various application domains, as well as appropriate data

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

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. As discussed earlier, the invariants proposed in [KPV95] come closest to the topological invariants we consider, but use a different spatial model and capture isotopy-generic information. 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 thematic 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.

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. In Section 3 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 4. Decidability and complexity results on the region based languages are provided in Section 5. Topological invariants and the results on spatial versus thematic information are presented in Section 6.

2 Basics

Practical spatial databases (such as geographic systems) mix thematic and spatial information. 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** (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. When present, we refer to a finite specification of $ext(I,r)$ as $spec(I,r)$. We use the notation $ext(r)$ and $spec(r)$ 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

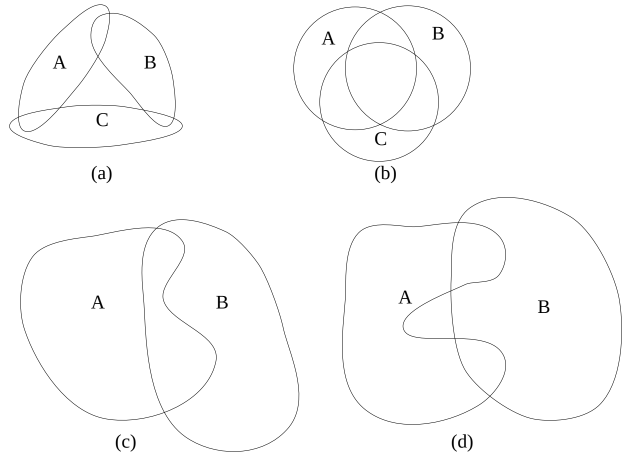

Figure 1: Four examples of spatial database instances.

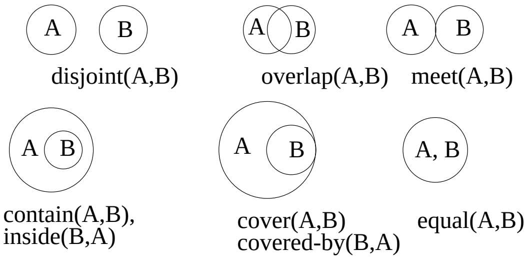

Figure 2: The Egenhofer-Franzosa topological relationships.

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 information systems. These are illustrated in Figure 2.

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).

3 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*$'$. Typically (but not necessarily) $Region \subseteq Region'$.

We will consider 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*.

Unless otherwise stated, we will implicitly assume in the sequel that every class of regions *Region* is one of the above. Notice that $\mathit{Rect} \subset \mathit{Rect}^* \subset \mathit{Disc}$ and $\mathit{Poly} \subset \mathit{Alg} \subset \mathit{Disc}$. 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 shall denote with $FO_G(\mathit{Region}, \mathit{Region}')$ the subset of queries expressible in $FO(\mathit{Region}, \mathit{Region}')$ which are G-generic relative to Region'.

We note that the languages $FO(\mathit{Region}, \mathit{Region}')$ can be assumed to only use the Egenhofer-Franzosa relationship *disjoint*, or, equivalently, its negation $\mathit{connect}(r,r') \stackrel{\text{def}}{=} \neg\mathit{disjoint}(r,r')$, 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 $\forall r''.(\mathit{connect}(r,r'') \Rightarrow \mathit{connect}(r',r''))$. We have: $\mathit{overlap}(r,r') = \exists r''.(r'' \subseteq r \wedge r'' \subseteq r') \wedge \neg(r \subseteq r') \wedge \neg(r' \subseteq r)$, $\mathit{meet}(r,r') = \mathit{connect}(r,r') \wedge \neg \mathit{overlap}(r,r') \wedge \neg(r \subseteq r') \wedge \neg(r' \subseteq r)$, etc. Note that $r \subseteq \overline{r' \cup r''}$ can be expressed as $\forall q.\mathit{connect}(r,q) \Longrightarrow \mathit{connect}(r',q) \vee \mathit{connect}(r'',q)$.

Incidentally, notice that in this language we can differentiate between the database instances (a) and (b), respectively (c) and (d), in Figure 1 by the following two queries: $\exists r.(r \subseteq A \cap B \cap C)$ and $\forall r,r'.(r \cup r' \subseteq A \cap B \Longrightarrow \exists r''.(r'' \subseteq A \cap B \wedge \mathit{connect}(r'',r) \wedge \mathit{connect}(r'',r')))$ respectively[2].

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 3 summarizes which class of regions is invariant under which of the groups described below. Call a function $\rho : \mathbb{R} \to \mathbb{R}$ *increasing* iff $x < x' \Longrightarrow \rho(x) < \rho(x')$; 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 $x_1 \in \mathbb{Q}$. 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.

- Homeomorphisms: $\mathcal{H}$ is the set of *homeomorphisms*, i.e. bijections $\lambda : \mathbb{R}^2 \to \mathbb{R}^2$ for which both λ and λ^{-1} are continuous.

[2] $r \subseteq r' \cap r''$ and $r \cup r' \subseteq r''$ are abbreviations for $r \subseteq r' \wedge r \subseteq r''$ and $r \subseteq r'' \wedge r' \subseteq r''$ respectively.

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 3: Which *Region* is invariant under which group G.

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

Whenever both *Region* and *Region*$'$ are invariant under G, all queries expressed in $FO(\mathit{Region}, \mathit{Region}')$ are G-generic. It follows that the languages $FO(\mathit{Region}, \mathit{Disc})$ are generic with respect to the groups shown in Figure 4. Interestingly, even if *Region* and *Region*$'$ are not G-invariant, queries in $FO(\mathit{Region}, \mathit{Region}')$ may be G-generic (relative to Region$'$). Indeed, using a straightforward variant of the Ehrenfeucht-Fraissé games for FO and Theorem 6.1 we prove:

Proposition 3.1 *All queries expressible in* $FO(\mathit{Alg}, \mathit{Alg})$ *and* $FO(\mathit{Poly}, \mathit{Poly})$ *are* $\mathcal{H}$*-generic.*

Relative expressiveness

With some exceptions, the languages $FO(\mathit{Region}, \mathit{Region}')$ 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.

Theorem 3.2 *The relationships of Figure 5 hold between the various languages* $FO(\mathit{Region}, \mathit{Region}')$*, for* $\mathit{Region}' \in \{\mathit{Alg}, \mathit{Disc}\}$*. Moreover:*

$$FO_{\mathcal{H}}(\mathit{Rect}, \mathit{Alg}) \subset FO_{\mathcal{H}}(\mathit{Rect}^*, \mathit{Alg}) = FO_{\mathcal{H}}(\mathit{Poly}, \mathit{Alg}) = FO_{\mathcal{H}}(\mathit{Alg}, \mathit{Alg}) \subset FO_{\mathcal{H}}(\mathit{Disc}, \mathit{Alg})$$

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

Figure 4: 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 5: Relationships between various languages $FO(Region, Region')$, for $Region' \in \{Alg, Disc\}$: # denotes "incomparable". The last entry is $\subset$ for *Alg*, and # for *Disc*.

As an aside, the relationship between $FO(Rect, Region')$ and $FO(Rect^*, Region')$ is further illuminated by the following result. We define the seqcond 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*.

Proposition 3.3 *The following holds:*

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

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). The details of the proof are quite involved, and are omitted from this abstract.

4 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 6. 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 6, 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 have the following central result:

Theorem 4.1 *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 6.3, and the "only if" part from Proposition 3.1.

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

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

The statement for $Rect^*, Poly$ follows again from Proposition 6.3. 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.

The question of finding a language expressing precisely the topological properties of instances over *Alg* has been open [Par95]. Theorem 4.1 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 5.1).

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

Similarly to Proposition 4.2, we have:

Proposition 4.4 *Every topological property of spatial instances over Alg is expressible in* $\bigvee FO(Region, Alg)$*, where* $Region \in \{Rect^*, Poly\}$.

In Proposition 4.4 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 generic properties of finite structures.

Now suppose we are interested in *computable* topological properties. The following refines Theorem 4.3 and Proposition 4.4. *Region* can be any of $\{Disc, Alg, Poly, Rect^*\}$.

Theorem 4.5 *For each recursive topological property* σ *of instances over Alg there exists a recursive set* $\mathcal{F}$ *of* $FO(Region, Alg)$ *sentences such that* $\bigvee \mathcal{F}$ *defines* σ*; 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* $I \models \sigma$ *iff* $f(I) \in \mathcal{F}$.

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[3].

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 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

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

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, 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, 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 4.6

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

Note that, in particular, $FO(\mathbb{R}, <, Disc)$ 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 4.7 $FO(Rect, Disc) = FO_{\mathcal{S}}(\mathbb{P}, <_x, <_y, Disc)$. *In particular,* $FO(Rect, Disc)$ *and* $FO(\mathbb{P}, <_x, <_y, Disc)$ *express precisely the same topological properties.*

5 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[4]. 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 5.1 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*). Let AH stand for the arithmetical hierarchy, AnH stand for the analytical hierarchy [Rog], and AH_G, AnH_G be their restrictions to G-generic queries, for some group of permutations G.

Theorem 5.1 (Undecidability) *The query classes* $FO(Rect^* - Disc, Rect - Alg)$ *are undecidable. Moreover, assuming that we restrict our database instances having a fixed set of names,* $names(I) = \{A_1, \ldots, A_k\}$, *we have the following characterization of their expressive power:*

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

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 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.

To better understand the problem of decidability of region-based languages, let us consider the more restricted 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

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

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

This result, together with the negative results in [Kra91, KM94], has the following negative consequences:

Corollary 5.3 *$\Sigma_1(Rect^*, \emptyset)$ 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 satisfies it 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 5.4 (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:

Theorem 5.5 (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.*

6 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 spatial instances ? This is related to three important, overlapping problems:

- the *thematic problem:* When can domain-specific queries be answered by precomputed annotations stored in classical database form?
- the *spatial representation* problem: can spatial instances over some class *Region* always be represented by topologically equivalent spatial instances over some simpler class of regions *Region'*?
- the *topological model* problem: What is an appropriate model for 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, which 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 input 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.

We now outline the main steps in the construction of the invariant. Let I be a non-empty spatial database instance over *Alg*. We call a *labeling* of I a function $\sigma : names(I) \rightarrow \{o, \partial, -\}$, and denote with 3^I the set of all labelings. Each labeling determines a set, namely σ determines $I^\sigma \stackrel{\text{def}}{=} \bigcap_{r \in names(I)} r^{\sigma(r)}$, 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. Because I is semi-algebraic, each nonempty I^σ is a finite union of maximal connected components (and there are only polynomially many such connected components, by [KY85]). Following established terminology [KY85], we call these connected components *cells*. Let $\{R_v \mid v \in V\}$ be the collection of all nonempty cells of I^σ, for $\sigma \in 3^I$. The fact that cell v is in I^σ is recorded by a *labeling function* $l : V \rightarrow 3^I$ associating σ to v. Each cell has dimension $0, 1$ or 2; this is recorded by a mapping $\delta : V \rightarrow \{0, 1, 2\}$. We call cells of dimension 0 vertices and cells of dimension 1 edges. There is one unbounded 2-dimensional cell called the *exterior cell*

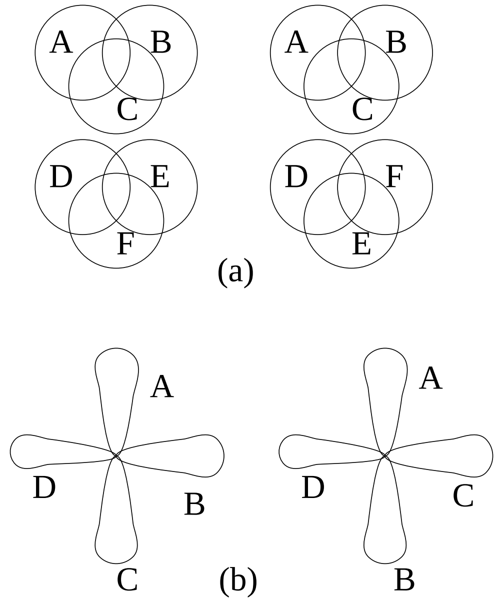

Figure 6: Two examples when $G_I, G_{I'}$ are isomorphic, but I, I' are not topologically equivalent.

and denoted v_0. Lastly, we record certain topological relationships among cells by $E = \{(u,v) \mid u,v \in V, R_u \subseteq \bar{R}_v\}$. Let $G_I \stackrel{\text{def}}{=} (V, E, \delta, v_0, l)$. Note that once the cells are computed, G_I provides in a concise manner the following information: for each region name in *names(I)*, the cells it contains (vertices, edges, and 2-dimensional cells); for each 2-dimensional cell, the edges on its boundary; for each edge, its endpoint(s); and lastly the 2-dimensional unbounded cell. G_I is almost what we need, but not quite – as illustrated in Figure 6.

It turns out that the only additional information needed to capture all topological information about I is some knowledge about the orientation of the edges incident to each vertex. This is done by specifying, for each vertex, the consecutive edges adjacent to it, both clockwise and counterclockwise. Specifically, let $\varepsilon_1, \varepsilon_{-1}$ be two symbols denoting the clockwise, respectively counterclockwise orientations. Let $O \subseteq \{\varepsilon_1, \varepsilon_{-1}\} \times V^3$ be a relation with the following meaning: $(\varepsilon_1, v, e, f) \in O$ iff v is a vertex and e, f are clockwise consecutive[5] edges incident to v, and $(\varepsilon_{-1}, v, e, f) \in O$ iff v is a vertex and e, f are counterclockwise consecutive edges incident to v. It follows[6] that $(\varepsilon_1, v, e, f) \in O$ iff $(\varepsilon_{-1}, v, f, e) \in O$. We define the *topological invariant* associated with an instance I to be the following finite structure: $T_I = (V, E, \delta, v_0, l, O)$ where $G_I = (V, E, \delta, v_0, l)$, and O is defined as above. Note that an isomorphism from T_I to $T_{I'}$ maps $\{\varepsilon_1, \varepsilon_{-1}\}$ to itself, *possibly reversing* the orientation. We can now show that T_I is the required invariant:

Theorem 6.1 *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 $names(I)$.*

Furthermore, the invariant, *as well as a representative instance in Poly*, can be constructed efficiently.

Theorem 6.2 *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 $T_I = T_{I'}$.*

Theorem 6.1 says that each $\mathcal{H}$-equivalence class of spatial instances over *Alg* is characterized by its topological invariant. Moreover, one can show the following useful definability result:

Proposition 6.3 *Each $\mathcal{H}$-equivalence class of spatial instances over Alg is definable by a sentence in any of the languages $FO(Rect^* - Disc, Alg)$.*

Proposition 6.3 proves the "if" part of Theorem 4.1, the "only if" part of Theorem 4.3, and Propositions 4.2 (except for *Rect*) and 4.4. It also offers a key step in the proof of Theorem 5.1: e.g. given some $\mathcal{H}$-generic property σ in AH, we construct a formula in $FO(Rect^*, Alg)$ deciding σ as follows. First we note that $FO(Rect^*, Alg)$ is powerful enough to encode the Peano arithmetic[7]. Then σ will be expressed as: $\exists n.\psi$, where ψ says that the number n is the encoding of some semi-algebraic database instance I, next it computes T_I using the cell decomposition algorithm in [KY85], checks whether it is isomorphic to the current instance, and, finally, checks that n satisfies the number-theoretic property in AH defining σ. Finally, Proposition 6.3 suggests a normal form for the computations in $FO(Region, Alg)$, in two steps: (1) compute the topological invariant T_I of some database instance I, and (2) perform a computation on the first-order structure T_I: this step can be as complex as the arithmetical, or analytical hierarchy, see Theorem 5.1.

The thematic mapping

Theorem 6.1 shows that from each instance over *Alg* one can build in polynomial time a "summary" which contains *exactly* enough information needed to answer all topological queries. The invariant is easily represented in classical relational database, i.e. thematic, form. This is summarized in the following:

[5] Consecutive edges share a 2-dimensional cell c and can be connected in c without separating any other two edges bordering c.

[6] But note that in the presence of blobs, Figure 6, we may have $e = f$, or both (ε_1, v, e, f) and (ε_1, v, f, e) in O.

[7] A number n is encoded as two regions $r, r' \in Rect^*$ s.t. $r \cap r'$ has n connected components.

Corollary 6.4 *There exists a relational database schema* **R** *and a mapping* **thematic** *from spatial database instances over Alg to relational instances over* **R***, computable in polynomial time (and NC), such that the following hold.*

(i) For all spatial instances I, J over Alg, I and J are topologically equivalent iff $thematic(I) = thematic(J)$.

(ii) For each recursive topological property τ of spatial instances over Alg there exists a recursive property $thematic(\tau)$ of instances over **R***, of complexity polynomially related to that of τ, such that for each spatial instance I over Alg,*

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

If topological invariants are to be used as a model for topological spatial databases, it is important to check whether a given instance over **R** is in fact in the image of the *thematic* mapping. Indeed, this functions as an integrity constraint when updates are performed. We can show (using the notation of Corollary 6.4):

Theorem 6.5 *It can be checked in* NC *whether an instance over* **R** *is the image of a spatial instance over Alg via the* thematic *mapping.*

The proof uses a combinatorial characterization of structures which are topological invariants, as labeled planar graphs.

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

Remark Results similar to the ones obtained here for topological properties of inputs over *Alg* can be obtained for the $\mathcal{S}$-generic properties of inputs over *Rect** 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 4, only this time for the $\mathcal{S}$-generic and $\mathcal{L}$-generic properties of inputs over *Rect** and *Poly*. We omit further details in this abstract.

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.

[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.

[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.

[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.

[St] 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.