Region-Based Query Languages for Spatial Databases in the Topological Data Model

Luca Forlizzi[1], Bart Kuijpers[2], and Enrico Nardelli[1,3]

[1] University of L'Aquila
Dipartimento di Informatica
Via Vetoio, 67010 L'Aquila, Italy
{forlizzi,nardelli}@univaq.it
[2] University of Limburg
Dept. of Mathematics, Physics and Computer Science
3590 Diepenbeek, Belgium
bart.kuijpers@luc.ac.be
[3] IASI–CNR
Viale Manzoni 30, 00185 Roma, Italy

Abstract. We consider spatial databases in the topological data model, i.e., databases that consist of a finite number of labeled regions in the real plane. Such databases partition the plane further into *elementary regions*. We propose a first-order language, which uses elementary-region variables and label variables, to query spatial databases. All queries expressible in this first-order logic are *topological* and they can be evaluated in polynomial time. Furthermore, the proposed language is powerful enough to distinguish between any two spatial databases that are not topologically equivalent. This language does not allow the expression of all computable topological queries, however, as is illustrated by the connectivity query. We also study some more powerful extensions of this first-order language, e.g., with a while-loop. In particular, we describe an extension that is sound and computationally complete for the topological queries on spatial databases in the topological data model.

1 Introduction and Motivation

We consider planar spatial databases in the topological data model, i.e., databases that consist of a finite number of labeled regions in the real plane. Egenhofer and his collaborators, who were among the first to consider this model, have studied the possible topological relationships between regions in the plane and proposed a number of predicates (the so-called 9-intersection model) to express topological properties of pairs of regions [7, 8, 9]. Independently, in the area of spatial reasoning, these topological relations were studied by Randell, Cui and Cohn [30]. Later on, the topological data model was investigated further and given a theoretical foundation by Papadimitriou, Segoufin, Suciu, and Vianu [25, 32], who considered first-order languages, built on the predicates of

T. Hadzilacos et al. (Eds.): SSTD 2003, LNCS 2750, pp. 344–361, 2003.

the 9-intersection model, to express topological properties of spatial data (for an overview and a general discussion on topological spatial data and topological queries see also [24]). In these languages the input databases as well as the variables range over some infinite class of regions.

Recently, the advantages of region-based models over point-based models (e.g., [23, 25, 32]) or coordinate-based models (as are found in the constraint database model for spatial databases [26, 31]) have been investigated by Pratt and his collaborators [27, 28, 29]. Although Pratt *et al.* have concluded that in some non-topological contexts the power of languages in these three models coincides, they show that a region-based approach is more efficient and parsimonious in a topological setting both from a logic and an AI point of view.

Inspired by one of the languages of Papadimitriou, Suciu, and Vianu [25], namely $\mathrm{FO}(Alg, Alg)$, in which both the variables and the inputs range over the set of labeled semi-algebraic disks, we propose, in this paper, an alternative region-based first-order language, named $\mathcal{RL}$, which is less expressive but which does have a semantics that is computable. Query evaluation has polynomial time complexity (in the size of the input database). The language $\mathcal{RL}$, just like the one of Papadimitriou, Suciu, and Vianu, is a two-sorted logic. Variables of a first type range over region labels. The labeled regions of a database in the topological data model partition the plane further into a finite number of *elementary regions.* In $\mathcal{RL}$, a second sort of variables range over elementary regions. Apart from some set-theoretical predicates, the only topological predicates available in $\mathcal{RL}$ express in which order elementary regions appear around an elementary region.

First, we show that all queries, expressible in $\mathcal{RL}$, are *topological.* Furthermore, the proposed language is shown to be powerful enough to distinguish between any two spatial databases that are not topologically equivalent. Although our first-order language can express all the predicates of the 9-intersection model, it does not allow the expression of all computable topological queries, however, as is illustrated by the connectivity query. Also Papadimitriou, Suciu and Vianu [25] have shown that their logic is not powerful enough to express all computable topological queries and they study an extension with infinitary recursive disjunctions. The latter language is shown to be complete for the topological queries. The topological connectivity query can be viewed as the spatial analogue of the standard relational query of graph connectivity, which is also not expressible in the standard relational calculus [1, 35]. To be able to express queries such as graph connectivity, one typically uses a more powerful query language such as Datalog [35], an extension of the relational calculus with recursion.

Also in the constraint model for spatial data [26, 31], various people have proposed and studied extensions of first-order logic over the reals with tractable recursion mechanisms to obtain more expressive languages. For example, Datalog versions with constraints have been proposed [14, 20]; a programming language extending first-order logic over the reals with assignments and a while-loop has been shown to be a computationally complete language for constraint databases [26, Chapter 2]; extensions of first-order logic over the reals with topological predicates have been proposed and studied [2, 13]; and various extensions of first-order logic over the reals with various transitive-closure operators have been proposed [12, 14, 18]. These extensions are more expressive, in particular, they allow the expression of connectivity and reachability queries and some are even computationally complete (in general or for what concerns topological queries).

Motivated by these results, we also study an extension of the first-order language $\mathcal{RL}$, with ad-hoc predicates, with a transitive-closure operator and with *while-loop*. Of the latter languages we can show different kinds of completeness with respect to certain complexity classes. In particular, we describe an extension of $\mathcal{RL}$ with while-loop and some set-theoretic operators that is sound and computationally complete for the topological queries on spatial databases in the topological data model.

This paper is organized as follows. In the next section, we define spatial databases in the topological data model, topological equivalence of spatial databases and spatial database queries. In Section 3, we define the region-based first-order query language $\mathcal{RL}$ and investigate its expressive power. The different extensions of $\mathcal{RL}$ their completeness are discussed in Section 4. We end this paper with a discussion of the obtained results and future work.

2 Definitions and Preliminaries

In this section, we define spatial databases, topological equivalence of spatial databases and spatial database queries. We denote the set of real numbers by $\mathbf{R}$ and the real plane by $\mathbf{R}^2$.

2.1 Spatial Databases

We adopt the well-known *topological data model* for spatial data in which a spatial database consists of labeled regions in the plane [7, 9, 24, 25, 32]. We assume the existence of an infinite set **Names** of *region labels*.

Definition 1. A *spatial database* (*instance*) Δ consists of a finite subset $names_\Delta$ of **Names** and a mapping ext_Δ from $names_\Delta$ to semi-algebraic regions in $\mathbf{R}^2$ that are homeomorphic[1] to the open unit disk. □

We remark that semi-algebraic regions can be finitely described as a Boolean combination of polynomial constraint expressions of the form $p(x, y) > 0$, where $p(x, y)$ is a polynomial in the real variables x, y with integer coefficients. The upper half of the open unit disk, that can be described by the polynomial constraint formula $x^2 + y^2 < 1 \wedge y > 0$ is an example of a semi-algebraic region.

Spatial databases are therefore within the framework of constraint databases [26, 31] in which spatial data is modeled as semi-algebraic sets. Figure 1 (a) gives an example of a spatial database instance with four regions, labeled

[1] Two sets A and B in $\mathbf{R}^2$ are called *homeomorphic* if there exists an homeomorphism h of the plane, i.e., a continuous bijective mapping from $\mathbf{R}^2$ to $\mathbf{R}^2$ with a continuous inverse, such that $h(A) = B$.

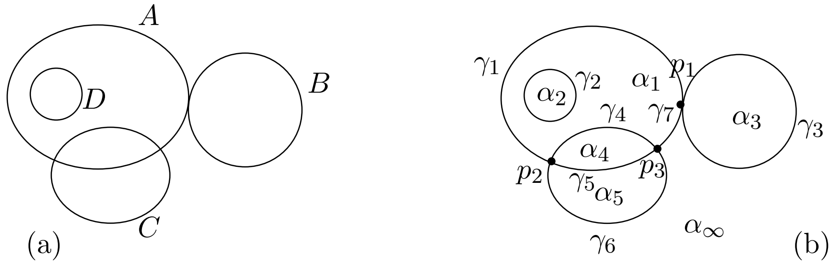

Fig. 1. In (a), an example of a spatial database with four labeled regions (remark that the regions are actually the interiors of the curves) and its elementary points, curves, and regions (b)

with A, B, C and D. All these regions are homeomorphic to the open unit disk. Remark that the region labeled D is a subset of the region labeled A.

In the remainder of this paper, we denote the topological interior, the topological border, and the topological closure of a set S respectively by S°, ∂S, and $\bar{S}$.

Definition 2. Let Δ be a spatial database instance.

- We refer to the union of the bordering curves of the labeled regions of Δ, i.e.,
$$\bigcup_{A \in names_\Delta} \partial(ext_\Delta(A)),$$
as the *frame of* Δ, and denote this set by $Frame(\Delta)$;
- We call the points of the frame where the frame is locally not homeomorphic to a straight line, the *elementary points of* Δ, and denote the set of these points by P_Δ;
- We call the connected components of $Frame(\Delta) \setminus P_\Delta$ the *elementary curves of* Δ and denote this set of curves by C_Δ;
- We call the connected components of $\mathbf{R}^2 \setminus Frame(\Delta)$ the *elementary regions of* Δ, and denote the set of elementary regions by R_Δ. □

For the spatial database instance depicted in Figure 1 (a), these sets are illustrated in Figure 1 (b). There are three elementary points: p_1, p_2 and p_3 (the frame has four branches locally around these points). There are seven elementary curves: γ_1, ..., γ_7 in Figure 1 (b). The complement of the frame has six connected components: α_1, ...,α_5, and α_∞ in Figure 1 (b).

From well-known properties of semi-algebraic sets it follows that P_Δ, C_Δ and R_Δ are always finite sets and that there is exactly one unbounded elementary region, which we denote by the constant α_∞ [3]. Throughout the remainder of this paper, we use $p_1, p_2, \ldots$ to denote elementary points, $\gamma_1, \gamma_2, \ldots$ to denote elementary curves, and $\alpha_1, \alpha_2, \ldots$ to denote elementary regions.

2.2 Topological Equivalence of Spatial Databases

It is well-known that the homeomorphisms of $\mathbf{R}^2$ are either orientation-preserving or orientation-reversing [33]. A reflection around a line is an example of an orientation-reversing homeomorphism. Orientation preserving homeomorphisms are commonly referred to as *isotopies* [33]. To increase readability, we will work with isotopies in this paper. We can think of isotopies as continuous deformations of the plane that take place completely within the plane (for a reflection around a line, we need to leave the plane for a moment). The results presented in this paper can be easily extended to homeomorphisms, however.

Definition 3. We call two spatial databases Δ_1 and Δ_2 *topologically equivalent* if $names_{\Delta_1} = names_{\Delta_2}$ and if there exists an isotopy i of $\mathbf{R}^2$ such that $i(ext_{\Delta_1}(A)) = ext_{\Delta_2}(A)$ for all A in $names_{\Delta_1}$. □

We denote the fact that Δ_1 and Δ_2 are topologically equivalent by an isotopy i, by $i(\Delta_1) = \Delta_2$. Topological equivalence of spatial databases can be decided in polynomial-time [19, 21].

2.3 Spatial Database Queries

We now turn to spatial database queries. In this paper, we are mainly interested in Boolean queries. We consider a *spatial database query* to be computable mapping on spatial database instances with a one-bit output. Furthermore, we are especially interested in topological queries.

Definition 4. We say that a spatial database query Q is *topological*, if for any topologically equivalent spatial databases Δ_1 and Δ_2, $Q(\Delta_1) = Q(\Delta_2)$. □

"Is the union of the labeled regions in Δ connected?" is an example of a (Boolean) topological query. "Are there more than four labeled regions of Δ that are above the x-axis?" is *not* topological, however.

The restriction to Boolean queries is not fundamental, however. Indeed, for instance by reserving specific labels for input database regions and others labels for output regions, we can simulate a spatial database query Q that on input Δ_1 returns output Δ_2, by a Boolean query Q' that takes as input the disjoint union $\Delta_1 \cup_d \Delta_2$ and that is such that $Q(\Delta_1) = \Delta_2$ if and only if $Q'(\Delta_1 \cup_d \Delta_2)$ is true.

3 $\mathcal{RL}$: An Elementary-Region Based First-Order Query Language

In this section, we describe the two-sorted first-order logic $\mathcal{RL}$, a spatial query language which uses label variables and elementary-region variables. We also study its expressive power as a topological query language.

3.1 Syntax and Semantics of the Language $\mathcal{RL}$

Syntax of $\mathcal{RL}$*.* The language $\mathcal{RL}$ is a two-sorted first-order logic with label variables (typically denoted by a with or without accents and subscripts) and elementary-region variables (typically denoted by r with or without accents and subscripts). The logic $\mathcal{RL}$ has α_∞ as an elementary-region constant and all A, for $A \in \mathbf{Names}$, as name constants.

A *query* in the language $\mathcal{RL}$ is expressed by a first-order formula

$$\varphi(a_1, \ldots, a_m, r_1, \ldots, r_n),$$

with free label variables $a_1, \ldots, a_m$ and free elementary region variables $r_1, \ldots,$ r_n. Such first-order formulas are built with the connectives $\wedge, \vee$, $\neg$, $\rightarrow$ and $\leftrightarrow$, quantification $(\exists r)$ and $(\forall r)$ over elementary regions, and quantification $(\exists a)$ and $(\forall a)$ over labels, from atomic formulas of the form

- $r \subseteq a$,
- $a = a'$, $a = A$, for $A \in \mathbf{Names}$,
- $r = r'$, $r = \alpha_\infty$, and
- $\mathsf{cw}^{d_1 d_2 d_3}(r, r'_1, r'_2, r'_3)$, for $d_1, d_2, d_3 \in \{0, 1\}$,

where r, r', r'_1, r'_2, and r'_3 are elementary-region variables and a and a' are label variables. □

Further on, we will also use expressions like $r \subseteq A$. These abbreviate the formulas $(\exists a)(r \subseteq a \wedge a = A)$. We now turn to the semantics of $\mathcal{RL}$ queries.

Semantics of $\mathcal{RL}$*.* The truth value of an $\mathcal{RL}$ query $\varphi(a_1, \ldots, a_m, r_1, \ldots, r_n)$, when evaluated on an input database Δ and with the instantiations $A_1, \ldots A_m$ for $a_1, \ldots, a_m$, and $\alpha_1, \ldots, \alpha_n$ for $r_1, \ldots, r_n$ is defined as follows (in terms of logic, we are going to define the meaning of what is usually denoted as $\Delta \models \varphi[A_1, \ldots,$ $A_m, \alpha_1, \ldots, \alpha_n]$). The elementary region variables appearing in $\varphi(a_1, \ldots, a_m, r_1,$ $\ldots, r_n)$ are interpreted to range over the finite set R_Δ of elementary regions of Δ and the label variables are interpreted to range over the elements of the set $names_\Delta$. The expression $r \subseteq a$ means that the elementary region r is contained in the labeled region with label a. The formula $a = a'$ expresses equality of labels, and $a = A$ express the equality of the label variable a and the constant label A. The expressions $r = r'$ and $r = \alpha_\infty$ express respectively equality of elementary regions and the equality with the unbounded elementary region in Δ. Finally, the formula $\mathsf{cw}^{d_1 d_2 d_3}(r, r'_1, r'_2, r'_3)$ means that the elementary regions r'_1, r'_2 and r'_3 (possibly, some or all of these are the same) appear consecutively in clockwise order around the bounded elementary region r such that the intersection of the closure of r and the closure of r'_i is d_i-dimensional $(i = 1, 2, 3)$. A 0-dimensional intersection is a point, and a 1-dimensional intersection is a curve segment. If r is an elementary region, surrounded by a single elementary region r', we agree that $\mathsf{cw}^{111}(r, r', r', r')$ holds. We agree that $\mathsf{cw}^{d_1 d_2 d_3}(\alpha_\infty, r', r'', r''')$ evaluates to *false* for any values of r', r'' and r'''. □

For examples of the latter expressions, we turn to the database of Figure 1 (a). Both the expressions $\mathsf{cw}^{101}(\alpha_1, \alpha_\infty, \alpha_3, \alpha_\infty)$, $\mathsf{cw}^{010}(\alpha_4, \alpha_\infty, \alpha_5, \alpha_\infty)$ and $\mathsf{cw}^{111}(\alpha_2, \alpha_1, \alpha_1, \alpha_1)$ hold but $\mathsf{cw}^{000}(\alpha_1, \alpha_\infty, \alpha_3, \alpha_\infty)$ does not hold.

When evaluated on the database shown in Figure 1 (a), the sentence $(\exists r)(\exists a)(r \subseteq a \wedge a = A)$ evaluates to *true*, since there is an elementary region within the region labeled A. The sentence $(\exists r)(\exists r')(\exists a)(\neg r = r' \wedge r \subseteq a \wedge r' \subseteq a \ \wedge a = D)$ evaluates to *false* on this database instance, however. Indeed, the region labeled D contains only one elementary region.

In the above definition, we allow an $\mathcal{RL}$ query to be expressed by a formula $\varphi(a_1, \ldots, a_m, r_1, \ldots, r_n)$ with free variables. As stated in the previous section, we are mainly interested in Boolean queries, i.e., queries expressed by formulas without free variables.

The following proposition says that $\mathcal{RL}$ queries can be efficiently computed.

Proposition 1. *$\mathcal{RL}$ queries can be evaluated in polynomial time (in the size of the constraint formulas that describe the input database).*

Proof sketch. Let $\varphi(a_1, \ldots, a_m, r_1, \ldots, r_n)$ be an $\mathcal{RL}$ formula. To evaluate this formula on a given input database Δ, we can proceed as follows. Firstly, the sets of elementary points, curves and regions of Δ are computed. The sets P_Δ, C_Δ and R_Δ have sizes that are bounded polynomially in the size of Δ (more precisely, in the size of the constraint formulas describing Δ) and they can be computed in polynomial time. The set P_Δ can be computed from the polynomial constraint formulas of the labeled regions in Δ in first-order logic over the reals (see, e.g., [22]). The computation of C_Δ and R_Δ from the given polynomial constraint formulas also requires polynomial time (in the number of polynomials used to describe Δ and their degrees) [15].

Subformulas of $\varphi(a_1, \ldots, a_m, r_1, \ldots, r_n)$ of the form $(\exists r)\psi(a_1, \ldots, a_k, r, r_1, \ldots, r_l)$ can be equivalently replaced by

$$\bigvee_{\alpha \in R_\Delta} \psi(a_1, \ldots, a_k, \alpha, r_1, \ldots, r_l),$$

and subformulas of the form $(\exists a)\psi(a, a_1, \ldots, a_k, r_1, \ldots, r_l)$ can be equivalently replaced by

$$\bigvee_{A \in names_\Delta} \psi(A, a_1, \ldots, a_k, r_1, \ldots, r_l).$$

These formulas are polynomially long in the size of Δ. (Remark that strictly speaking the latter formulas are not in $\mathcal{RL}$. But we write them in an $\mathcal{RL}$-like fashion to show how their evaluation can be performed). After these replacements, we obtain a quantifier-free expression, that equivalently expresses the original query, to be evaluated.

To compute the output set of all of $(A_1, \ldots, A_m, \alpha_1, \ldots, \alpha_n) \in (names_\Delta)^m \times (R_\Delta)^n$ for which $\Delta \models \varphi[A_1, \ldots, A_m, \alpha_1, \ldots, \alpha_n]$, we can then proceed as follows. We generate all possible candidate outputs $(A_1, \ldots, A_m, \alpha_1, \ldots, \alpha_n) \in (names_\Delta)^m \times (R_\Delta)^n$ and test each of them. Since, for the given formula

$\varphi(a_1, \ldots, a_m, r_1, \ldots, r_n)$, m and n are fixed, the number of possible outputs is again polynomial in the size of Δ. The latter test can be done because when all variables are instantiated, the atomic formulas can be evaluated. Indeed, the formulas $r \subseteq A_i$, $r = r'$ and $r = \alpha_\infty$ can be checked in first-order logic over the reals (in polynomial time again), whereas, $\mathsf{cw}^{d_1 d_2 d_3}(r, r'_1, r'_2, r'_3)$ can be verified by computing adjacency information on the elements of P_Δ, C_Δ and R_Δ. Also the adjacency information can be computed in time polynomial in the size of Δ.

In conclusion, we can say that for a fixed $\mathcal{RL}$ expression $\varphi(a_1, \ldots, a_m, r_1, \ldots, r_n)$, this expression can be evaluated on each input database Δ in time polynomial in the size of Δ. □

We remark that, even for a fixed number of labeled regions, the number of elementary regions is not bounded. So, $\mathcal{RL}$ is by no means equivalent to a propositional logic.

3.2 Some First Observations on Expressing Topological Queries in $\mathcal{RL}$

Here, we start by observing that the language $\mathcal{RL}$ is powerful enough to express the relations of the 9-intersection model. We also state that all queries expressible in $\mathcal{RL}$ are topological.

The 9-Intersection Model. So, firstly we show that $\mathcal{RL}$ is expressive enough to allow the formulation of the predicates of the 9-intersection model. Consider these spatial predicates on labeled regions that were investigated in depth by Egenhofer and his collaborators [7, 8, 9]:

- disjoint(A, B), meaning that the topological closure of A is disjoint with that of B;
- overlap(A, B), meaning that A and B have intersecting interiors;
- meet$_{Line}(A, B)$, meaning that A and B have disjoint interiors and that part of their borders have a 1-dimensional intersection;
- meet$_{Point}(A, B)$, meaning that A and B have disjoint interiors and that part of their borders have a zero-dimensional intersection;
- contain(A, B), meaning that $B \subseteq A$ and that their borders are disjoint;
- cover(A, B), meaning that $B \subset A$ and their borders touch;
- equal(A, B), meaning that $A = B$.

Proposition 2. *The predicates* disjoint*,* overlap*,* contain cover*,* equal*,* meet$_{Line}$*, and* meet$_{Point}$ *of the 9-intersection model are expressible in $\mathcal{RL}$.*

Proof. The formula

$$\psi(A, B) \equiv (\forall r)(\forall r')\Big(r \subseteq A \ \wedge r' \subseteq B \to (\forall r'')(\forall r''')\big(\bigwedge_{\delta} \neg \mathsf{cw}^{\delta}(r, r', r'', r''')\big)\Big),$$

where δ ranges over $\{0,1\}^3$, expresses that the borders of A and B are disjoint. Now, disjoint(A,B) can be equivalently expressed in $\mathcal{RL}$ by the sentence $\neg(\exists r)(r \subseteq A \wedge r \subseteq B) \wedge \psi(A,B)$. The fact meet$_{Line}(A,B)$ is expressed as

$$\neg(\exists r)(r \subseteq A \wedge r \subseteq B) \wedge \\ (\exists r)(\exists r')(\exists r'')(\exists r''')\big(r \subseteq A \wedge r' \subseteq B \wedge \bigvee_{(d_1,d_3)\in\{0,1\}^2} \mathsf{cw}^{d_1 1 d_3}(r, r'', r', r''')\big).$$

And meet$_{Point}(A,B)$ is expressed as

$$\neg(\exists r)(r \subseteq A \wedge r \subseteq B) \wedge \\ (\exists r)(\exists r')(\exists r'')(\exists r''')\big(r \subseteq A \wedge r' \subseteq B \wedge \bigvee_{(d_1,d_3)\in\{0,1\}^2} \mathsf{cw}^{d_1 0 d_3}(r, r'', r', r''')\big).$$

The formula overlap(A,B) is expressed as $(\exists r)(r \subseteq A \ \wedge r \subseteq B)$, contains$(A,B)$ is expressed as $(\forall r)(r \subseteq B \rightarrow r \subseteq A) \wedge \psi(A,B)$, covers$(A,B)$ is expressed as $(\forall r)(r \subseteq B \rightarrow r \subseteq A) \wedge \neg\psi(A,B)$, and finally equal$(A,B)$ is expressed as $(\forall r)(r \subseteq A \leftrightarrow r \subseteq B)$. □

Topological Queries in $\mathcal{RL}$. In Section 2.3 we have given the definition of a topological query (Definition 4). As already remarked, $\mathcal{RL}$ also allows the expression of queries that produce a non-Boolean output. Using the remark made at the end of Section 2.3, we can generalize the definition of a topological query to situations where queries can produce an arbitrary output as follows.

Definition 5. We say that a formula $\varphi(a_1,\ldots,a_m,r_1,\ldots,r_n)$ in $\mathcal{RL}$ is *topological* if and only if for any spatial databases Δ_1 and Δ_2 that are topologically equivalent by some isotopy i, we also have that $\{(A_1,\ldots,A_m,\alpha_1,\ldots,\alpha_n) \in (names_{\Delta_1})^m \times (R_{\Delta_1})^n \mid \Delta_1 \models \varphi[A_1,\ldots,A_m,\alpha_1,\ldots,\alpha_n]\}$ is mapped to $\{(A_1,\ldots,A_m,\alpha_1,\ldots,\alpha_n) \in (names_{\Delta_2})^m \times (R_{\Delta_2})^n \mid \Delta_2 \models \varphi[A_1,\ldots,A_m,\alpha_1,\ldots,\alpha_n]\}$ by the function $(id,\ldots,id,i,\ldots,i)$, where id is the identity mapping. □

Using this more general definition of topological query, the following proposition can be proven straightforwardly by induction on the syntactic structure of $\mathcal{RL}$ formulas.

Proposition 3. *All queries expressible in $\mathcal{RL}$ are topological.* □

3.3 Further Results on Expressing Topological Queries in $\mathcal{RL}$

Here, we discuss lower and upper bounds on the expressive power of $\mathcal{RL}$ as a language to express topological properties of spatial databases.

Lower Bound on the Expressiveness of $\mathcal{RL}$. First, we give the definition of elementarily equivalent spatial databases. The notion of elementary equivalence of a language captures the power of this language to distinguish different databases.

Definition 6. We denote the fact that two spatial databases Δ_1 and Δ_2 cannot be distinguished by any Boolean $\mathcal{RL}$ query (i.e., for every $\mathcal{RL}$ sentence ψ, $\Delta_1 \models \psi$ if and only if $\Delta_2 \models \psi$) by $\Delta_1 \equiv_{\mathcal{RL}} \Delta_2$, and we say that Δ_1 and Δ_2 are *elementarily equivalent.* □

The following result gives a lower bound for the expressive power of $\mathcal{RL}$.

Theorem 1. *We have that*

(i) if two spatial databases Δ_1 and Δ_2 are topologically equivalent then they are elementarily equivalent, i.e., then $\Delta_1 \equiv_{\mathcal{RL}} \Delta_2$;
(ii) if two spatial databases Δ_1 and Δ_2 are elementarily equivalent, i.e., if $\Delta_1 \equiv_{\mathcal{RL}} \Delta_2$, then they are topologically equivalent;
(iii) for every database instance Δ there exists a $\mathcal{RL}$ sentence χ_Δ such that for every database instance Δ', $\Delta' \models \chi_\Delta$ if and only if Δ and Δ' are topologically equivalent. □

Item (iii) states that for every spatial database there is a *characteristic formula* that exactly describes the topology of the spatial database.

Whereas (i) of this Theorem follows immediately from Proposition 3, (ii) and (iii) require more work.

We first prove the following technical lemma.

Lemma 1. *Two spatial database instances Δ_1 and Δ_2 are topologically equivalent if and only if there exists a bijection between R_{Δ_1} and R_{Δ_2} that maps the unbounded elementary region to the unbounded elementary region, that maps elementary regions within certain labeled regions to elementary regions in regions with the same region label and that preserves for any d_1, d_2, $d_3 \in \{0, 1\}$ the eight relations $\mathsf{cw}^{d_1 d_2 d_3}(r, r_1, r_2, r_3)$.*

Proof sketch. The only-if direction is obvious. For the if-direction, we first observe that two spatial database instances Δ_1 and Δ_2 are topologically equivalent if and only if their frames are isotopic[2] by an isotopy that respects the labels. So, we proceed with their frames. We first remark that the frame of a spatial database can be constructed by applying the following two operations Op_1 and Op_2 a finite number of times starting from the empty plane:

Op_1 : add a closed curve in the unbounded region;
Op_2 : add a curve in the unbounded region between two points of already existing curves such that a new region is created.

This can be proven easily by induction on the number of elementary curves in the spatial database.

[2] We call two subsets of $\mathbf{R}^2$ isotopic if there is an isotopy (i.e., an orientation-preserving homeomorphism) of $\mathbf{R}^2$ that maps one to the other.

We prove the if-direction by induction on the number of elementary curves in the frame of Δ_1. If the number of elementary curves is zero, Δ_1 only has one elementary region, namely α_∞. By assumption, also Δ_2 has only one elementary region and therefore the identity mapping is the desired isotopy.

Assume that the number n of elementary curves of Δ_1 is strictly positive. Let b be the bijective mapping between R_{Δ_1} and R_{Δ_2} that we assume to exist. Because any frame can be constructed using operations Op_1 and Op_2, it follows that Δ_1 has an elementary curve γ that is adjacent to α_∞ with contact of dimension 1 (since a frame can be constructed using Op_1 and Op_2, γ can be either an isolated curve or a curve that connects two points of some other curves). Suppose γ separates α_∞ from the elementary region α_0 in Δ_1 and let γ' correspond in Δ_2 to γ. So, γ' separates α_∞ from $b(\alpha_0)$. If we remove γ and γ' from Δ_1 and Δ_2 respectively this results in two spatial database frames F_1 and F_2 such that α_0 and $b(\alpha_0)$ are identified with α_∞. It is not difficult to show that hereby the bijection b induces a bijection between the elementary regions of F_1 and F_2 that preserves the clockwise appearance of elementary regions around each elementary region. By the induction hypothesis, there exists an isotopy i of the plane that maps F_1 to F_2, and that respects the labeling. This isotopy maps the curve γ to $i(\gamma)$ which is not necessarily equal to γ'. We remark that $i(\gamma)$ creates a new elementary region. We can make sure that the labeling is respected. A "local" isotopy can be constructed however that locally maps $i(\gamma)$ to γ' and that leaves the remainder of the frame F_1 unaltered. Since by assumption, the labels of the elementary regions are respected, the composition of this local isotopy with i gives the desired isotopy that maps Δ_1 to Δ_2. □

Proof sketch of Theorem 1. If two databases Δ_1 and Δ_2 are isotopic, they cannot be distinguished by any $\mathcal{RL}$ sentence because of Proposition 3. This proves (i). To prove (ii), it suffices to prove (iii). We show that any spatial database Δ can be characterized up to isotopy by an $\mathcal{RL}$ sentence χ_Δ. This formula is of the form

$$(\exists r_1)\cdots(\exists r_n)\Big(\big((\forall r)\bigvee_{i=1}^{n} r = r_i\big) \wedge \bigwedge_{i<j} r_i \neq r_j \wedge \bigwedge_{i} r_i \neq \alpha_\infty \wedge \\ \bigwedge_{i,j_i} r_i \subseteq A_{j_i} \wedge \bigwedge_{i,(i_1,i_2,i_3)} \mathsf{cw}^{d_{i_1} d_{i_2} d_{i_3}}(r_i, r_{i_1}, r_{i_2}, r_{i_3})\Big)$$

which expresses that there are exactly n bounded elementary regions, says to which of the labeled regions these n elementary regions belong, and completely describes the clockwise appearance of elementary regions around all elementary regions.

Suppose that another database Δ' satisfies χ_Δ. Then there exists an assignment of the variables $r_1, \ldots, r_n$ to distinct bounded elementary regions of Δ' that makes χ_Δ true. This variable assignment then determines a bijection between the elementary regions of Δ and of Δ'. Because both databases satisfy χ_Δ, the

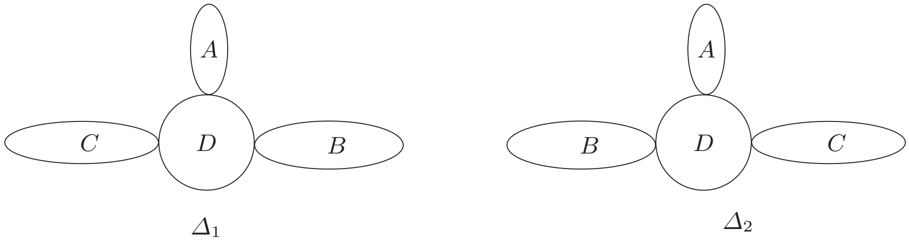

Fig. 2. Two databases Δ_1 and Δ_2 that cannot be distinguished using the less powerful predicate $\mathsf{cw}^{d_1 d_2}(r, r_1, r_2)$ alone

corresponding bounded elementary regions have the same clockwise appearance of areas around them. By Lemma 1, Δ and Δ' are therefore isotopic. This proves the theorem. □

We remark that a predicate $\mathsf{cw}^{d_1 d_2}(r, r_1, r_2)$, that expresses that r_1 and r_2 appear clockwise around r with contact of dimension d_1 and d_2 respectively is not sufficient to obtain Theorem 1. This is illustrated by the two databases in Figure 2. They cannot be distinguished using $\mathsf{cw}^{d_1 d_2}(r, r_1, r_2)$ alone. This is the reason why the more powerful $\mathsf{cw}^{d_1 d_2 d_3}(r, r_1, r_2, r_3)$ is used in $\mathcal{RL}$.

Upper Bound on the Expressiveness of $\mathcal{RL}$. The following result shows that not all topological queries can be expressed in $\mathcal{RL}$. With $\mathsf{connect}(r_1, r_2)$ we denote that the elementary regions r_1 and r_2 can be connected by a connected path that completely belongs to union of the closure of the labeled regions.

Proposition 4. *The predicate* connect *is not expressible in $\mathcal{RL}$.*

This result can be proven using a classical Ehrenfeucht-Fraïssé game argument. The proof of the above proposition would be too technical to give here in full, but the idea is outlined below.

Proof idea. An Ehrenfeucht-Fraïssé game is a game played over a certain number of rounds on two databases by two players; the first player is usually called the spoiler and the second the duplicator. (For the technical details of Ehrenfeucht-Fraïssé games we refer to theoretical database books [1] or logic texts [6].)

Assume that the predicate $\mathsf{connect}(r_1, r_2)$ *is* expressible in $\mathcal{RL}$. The sentence $\varphi_{\text{connected}}$ given as

$$(\forall r_1)(\forall r_2)(((\exists a_1)(\exists a_2)(r_1 \subseteq a_1) \wedge (r_2 \subseteq a_2)) \rightarrow \mathsf{connect}(r_1, r_2))$$

expresses that the spatial database is *topologically connected*.[3] So, if the predicate $\mathsf{connect}(r_1, r_2)$ is expressible in $\mathcal{RL}$, then also the topological connectivity

[3] More precisely, we call a spatial database here *topologically connected* if the union of the closure of the labeled regions in the spatial database is a path-connected subset of the plane.

test is expressible in $\mathcal{RL}$. The sentence $\varphi_{\text{connected}}$ has a certain quantifier rank, say k (basically, this is the number of quantifiers appearing in the quantifier-prefix of $\varphi_{\text{connected}}$ when it is transformed in prenex normal form). When using Ehrenfeucht-Fraïssé games to prove that the sentence $\varphi_{\text{connected}}$ of quantifier rank k is not expressible it suffices to give two spatial databases Δ_k and Δ'_k such that $\Delta_k \models \varphi_{\text{connected}}$ and $\Delta'_k \not\models \varphi_{\text{connected}}$ (i.e., Δ_k is connected and Δ'_k is disconnected), and such that the duplicator has a winning strategy for the k-round game on these two spatial databases. The two databases Δ_k and Δ'_k that are needed here can be found by adapting the well-known Ehrenfeucht-Fraïssé games that show that graph-connectivity is not expressible in the relational calculus (see, for instance, the proof of Proposition 17.2.3 in [1]) to this situation. Roughly, Δ_k would consist of an exponentially (in k) long chain of regions in which two neighboring regions are connected and Δ'_k would consist of two disjoint such chains. Using similar arguments as in the relational case [1], it can be shown that the duplicator has a winning strategy on these databases for the k-round game. □

We remark that there is a variety of examples of computable topological queries that are not expressible in $\mathcal{RL}$. For instance, the parity queries "Is the number of elementary regions in the database even?" and "Is the number of connected components of the database even?" are both not expressible in $\mathcal{RL}$.

4 More Powerful Query Languages: Extensions of $\mathcal{RL}$

Although many interesting properties of spatial databases in the topological data model can be expressed in $\mathcal{RL}$, an important deficiency of $\mathcal{RL}$ is that for practical applications important queries such as the connectivity test and reachability are not expressible in this first-order language, as we have seen in the previous section. In this section, we will briefly study a number of extensions of $\mathcal{RL}$: $\mathcal{RL}$ augmented with connect; $\mathcal{RL}$ augmented with a transitive closure operator and $\mathcal{RL}$ augmented with a while-loop (and some set-theoretic operators).

4.1 $\mathcal{RL}$ Augmented with Connect or Transitive Closure

An obvious approach to obtain the expressibility of the connectivity test is simply to augment $\mathcal{RL}$ with the predicate $\mathsf{connect}(r_1, r_2)$. Then connectivity of a database is expressible, as shown above, by the formula $(\forall r_1)(\forall r_2)(((\exists a_1)(\exists a_2)$ $(r_1 \subseteq a_1) \wedge (r_2 \subseteq a_2)) \rightarrow \mathsf{connect}(r_1, r_2))$. However, it is not clear if the language $\mathcal{RL} + \mathsf{connect}$ is complete in the sense that all computable topological queries are expressible in it. In the constraint model, for instance, when first-order logic over the reals is augmented with a predicate that expresses connectivity of two-dimensional sets, then parity of a set of real numbers is expressible. For $\mathcal{RL} + \mathsf{connect}$ we conjecture the opposite, however.

A transitive-closure operator can be added to $\mathcal{RL}$ in several ways. One possibility is that we add to $\mathcal{RL}$ expressions of the form

$$[\mathrm{TC}\ \varphi(r'_1, r'_2)](r_1, r_2),$$

where $\varphi(r_1', r_2')$ is a $\mathcal{RL}$ formula with two free elementary-region variables. The meaning of $[\mathrm{TC}\ \varphi(r_1', r_2')](r_1, r_2)$ is that the couple (r_1, r_2) of elementary regions belongs to the transitive closure of the binary relation defined by the $\mathcal{RL}$ formula $\varphi(r_1', r_2')$, i.e., the set $\{(r_1', r_2') \mid \varphi(r_1', r_2')\}$. Various more powerful extensions of $\mathcal{RL}$ could be thought of, but this one is strong enough to express the topological connectivity test. Indeed,

$$(\forall r_1)(\forall r_2)(((\exists a_1)(\exists a_2)((r_1 \subseteq a_1) \wedge (r_2 \subseteq a_2))) \rightarrow \\ [TC\ (\exists a_1)(\exists a_2)((r_1' \subseteq a_1) \wedge (r_2' \subseteq a_2) \wedge \mathrm{meet}(r_1', r_2'))](r_1, r_2))$$

where $\mathrm{meet}(r_1', r_2')$ abbreviates $\mathrm{meet}_{Line}(r_1', r_2') \vee \mathrm{meet}_{Point}(r_1', r_2') \vee r_1' = r_2'$ (the predicates meet_{Line} and meet_{Point} are defined as in Section 3.2), expresses that every pair of elementary regions that are in a labeled region are also in the transitive closure of the binary relation defined by meet (which contains all pairs of elementary regions which are adjacent). The computation of the transitive closure is guaranteed to terminate because the number of elementary regions is finite for any input database. This expresses that the union of the closure of the labeled regions in the spatial database is a connected subset of the plane. It is not clear what the expressive power of this extension of $\mathcal{RL}$ exactly is, however.

4.2 $\mathcal{RL}$ Augmented with a While-Loop

In the final part of this section, we introduce the language $\mathcal{RL}$ + While. This language is essentially the extension of the first-order logic $\mathcal{RL}$ with a while-loop and some set-theoretic operations. This extension of $\mathcal{RL}$ is a sound and complete language for the computable topological queries on spatial databases in the topological data model.

Definition 7. An $\mathcal{RL}$ + While-*program* is a finite sequence of *statements* and *while-loops*.

A statement is either an $\mathcal{RL}$-definition of a relation, based on previously defined relations or the result of a set-theoretic operation on previously defined relations. An $\mathcal{RL}$-definition of a relation has the form

$$R := \{(a_1, \ldots, a_m, r_1, \ldots, r_n) \mid \varphi(a_1, \ldots, a_m, r_1, \ldots, r_n)\};$$

where R is a relation variable of arity $m+n$ and φ is a formula in $\mathcal{RL}$ augmented with expressions $S(a_1, \ldots, a_k, r_1, \ldots, r_l)$ where S is some previously introduced relation variable of arity $k+l$. The other (set-theoretic) form of defining relations is one of the following: $R := S \cap S'$; $R := \neg S$; $R := S\downarrow$; $R := S\uparrow^a$; $R := S\uparrow^r$; and $R := S\sim$, where S and S' are some previously introduced relation variables.

A while-loop has the form

$$\textbf{while}\ \varphi\ \textbf{do}\ \{P\};$$

where P is a program and φ is a sentence in $\mathcal{RL}$ augmented with expressions $S(a_1, \ldots, a_k, r_1, \ldots, r_l)$ where S is some previously introduced relation variable of arity $k + l$. □

In this definition, it is assumed that there is a supply of untyped relation variables (this is important because relations in the while-language can grow arbitrarily wide). Semantically, a program in the query language $\mathcal{RL}+$While allows the creation of relations and in a loop like **while** φ **do** P , P is executed until φ becomes false. A program therefore expresses a query in the obvious way as soon as one of its relation variables has been designated as the output variable (e.g., R_{out}). The semantics of the set-theoretic operations needs some further clarification. An assignment $R := S \cap S'$ simply expresses the intersection. The assignment $R := \neg S$ expresses the complement with respect to the appropriate domains. The assignment $R := S{\downarrow}$ is projecting out the first dimension or coordinate. The assignment $R := S{\uparrow}^a$ is projecting in on the right with a extra label-dimension. And $R := S{\uparrow}^r$ is similar for a region-dimension. Finally, $R := S{\sim}$ exchanging the two right-most coordinates of S.

Obviously, the while-loops of $\mathcal{RL}+$While can be non-terminating. However, if a while-loop terminates (or a $\mathcal{RL}+$While-program for that matter), then all computed relations are $\mathcal{RL}$-definable.

As an example, we give an $\mathcal{RL}+$While-program that expresses that the input spatial database is connected (i.e., that the union of all the labeled regions in the input is a connected subset of the plane). In the following meet(r, r') abbreviates meet$_{Line}(r, r') \vee$ meet$_{Point}(r, r') \vee r = r'$, where meet$_{Line}$ and meet$_{Point}$ are in turn the abbreviations introduced in Section 3.2.

```
$R := \{(r, r') \mid (\exists a)(\exists a')(r \subseteq a \wedge r' \subseteq a' \wedge \text{meet}(r, r'))\};$
$R_1 := R;$
$R_2 := \{(r, r') \mid (\exists r'')(R_1(r, r'') \wedge R(r'', r'))\};$
 while $R_1 \neq R_2$ do {
        $R_1 := R_2;$
        $R_2 := \{(r, r') \mid (\exists r'')(R_1(r, r'') \wedge R(r'', r'))\};$
          };
$R_{\text{out}} := \{() \mid (\forall r)(\forall r')((\exists a)(\exists a')(r \subseteq a \ \wedge r' \subseteq a' \rightarrow R_2(r, r')))\};$
```

Here, an expression like $R_1 := R$ is an abbreviation of $R_1 := \{(r, r') \mid R(r, r')\}$. In this program, first a binary relation consisting of all adjacent elementary regions that are in a labeled region is computed. Next, the transitive closure of this binary relation is computed. The computation of the transitive closure is guaranteed to terminate because the number of elementary regions is finite for any input database. Finally, the relation R_{out} is defined. This relation is empty for a disconnected input and non-empty for a connected input database.

The main result of this section is the following.

Theorem 2. *The language* $\mathcal{RL}+$While *is sound and computationally complete for the topological queries on spatial databases in the topological data model.*

Proof idea. Using the results in Section 3, it is easy to show soundness. To prove completeness, we first observe that Cabibbo and Van den Bussche [36] have shown that many-sorted logics, like $\mathcal{RL}$ can be equivalently expressed in an untyped logic that is augmented with unary type-predicates (here ER and L). So, we can consider untyped variants of $\mathcal{RL}$ and $\mathcal{RL}+\text{While}$: $\mathcal{RL}^u$and $\mathcal{RL}^u + \text{While}$. The proof of the theorem is then in two steps. For what concerns the first step we observe that the relations $R_L = \{x \mid L(x)\}$, $R_{ER} = \{x \mid ER(x)\}$, $R_{\text{label}} = \{(x,y) \mid ER(x) \wedge L(y) \wedge x \subseteq y\}$, and $R_{d_1d_2d_3} = \{(x, x_1, x_2, x_3) \mid \mathsf{cw}^{d_1d_2d_3}(x, x_1, x_2, x_3) \wedge ER(x) \wedge ER(x_1) \wedge ER(x_2) \wedge ER(x_3)\}$ (with $d_1, d_2, d_3 \in \{0, 1\}$) contain the complete topological information of the input spatial database (this follows directly from Lemma 1). For a given input spatial database Δ, the relational database Δ_{fin}, consisting of these eleven relations is therefore definable in $\mathcal{RL}^u$, as just shown.

Secondly, we observe that by adding the set-theoretic operations to $\mathcal{RL}+$ While we have obtained a language powerful enough to express all generic Turing computable functions on Δ_{fin} (we can do this by showing that the query language QL of Chandra and Harel [4] can be simulated in $\mathcal{RL}^u + \text{While}$). □

Finally, we remark that if we extend $\mathcal{RL}$ with while as in Chandra and Harel [5] (also [1, Chapter 17]) and if we assume an ordering on the (elementary) regions, that then, using a well-known result, this extension of the language $\mathcal{RL}$ captures the PSPACE topological queries on spatial databases in the topological data model.

5 Conclusion and Discussion

In this paper, we have continued the search for effective, convenient and expressive languages for querying topological properties of spatial databases in the topological data model. In searching for such languages we face a number of challenges. We typically want languages to be natural in the sense that the primitives appearing in it express natural concepts such as intersection, adjacency and connectivity. We also want that queries are computable and have a complexity that belongs to a nice class such as PSPACE or PTIME. A third issue is completeness: all topological queries from preferably some suitable computational class should be captured.

To deal with these issues we propose the two-sorted logic $\mathcal{RL}$ and a number of extensions of it. In the language $\mathcal{RL}$ variables range over regions from the active domain of a spatial database instance, as opposed to an infinite universe of regions in previously discussed languages [25]. This logic, by the predicate cw, is descriptive enough to characterize the topological information of a spatial database instance. As we have shown in this paper, with $\mathcal{RL}$ and its extensions we meet some of the above set challenges. Especially on the level of naturalness improvement should be expected.

The topological data model allows a representation where a prominent role is given to the spatial containment relation [10, 11, 16, 17]. This is interesting from a practical point of view since it allows to use efficient data structures for the management of (partial) order relations [34]. Future work will focus on the translation of $\mathcal{RL}$ queries in terms of operations on suitably enriched order-based data structures.

References

[1] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995. 345, 355, 356, 359

[2] M. Benedikt, M. Grohe, L. Libkin, and L. Segoufin. Reachability and connectivity queries in constraint databases. In *Proceedings of the 19th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS'00)*, pages 104–115, 2000. 345

[3] J. Bochnak, M. Coste, and M. F. Roy. *Géométrie Algébrique Réelle*. Springer-Verlag, 1987. 347

[4] A. Chandra and D. Harel. Computable queries for relational database systems. *Journal of Computer and System Sciences*, 21(2):156–178, 1980. 359

[5] A. Chandra and D. Harel. Structure and complexity of relational queries. *Journal of Computer and System Sciences*, 25:99–128, 1982. 359

[6] H.-D. Ebbinghaus, J. Flum, and W. Thomas. *Mathematical Logic*. Undergraduate Texts in Mathematics. Springer-Verlag, 1984. 355

[7] M. Egenhofer. Reasoning about binary topological relations. In *Advances in Spatial Databases, Second International Symposium (SSD'91)*, volume 525 of *Lecture Notes in Computer Science*, pages 143–160. Springer-Verlag, 1991. 344, 346, 351

[8] M. Egenhofer. Topological relations between regions in $\mathbf{R}^2$ and $\mathbf{Z}^2$. In *Advances in Spatial Databases, Third International Symposium (SSD'93)*, volume 692 of *Lecture Notes in Computer Science*, pages 316–336. Springer-Verlag, 1993. 344, 351

[9] M. Egenhofer and R. Franzosa. On the equivalence of topological relations. *International Journal Geographical Information Systems*, pages 523–542, 1994. 344, 346, 351

[10] L. Forlizzi and E. Nardelli. Some results on the modelling of spatial data. In *Proceedings of the 25th Conference on Current Trends in Theory and Practice of Informatics (SOFSEM'98)*, pages 332–343, 1998. 359

[11] L. Forlizzi and E. Nardelli. Characterization results for the poset based representation of topological relations-I: Introduction and models. *Informatica (Slovenia)*, 23(2):332–343, 1999. 359

[12] F. Geerts and B. Kuijpers. Linear approximation of planar spatial databases using transitive-closure logic. In *Proceedings of the 19th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS'00)*, pages 126–135, 2000. 346

[13] Ch. Giannella and D. Van Gucht. Adding a path connectedness operator to FO+poly (linear). *Acta Informatica*, 38(9):621–648, 2002. 345

[14] S. Grumbach and G. Kuper. Tractable recursion over geometric data. In *Proceedings of Principles and Practice of Constraint Programming (CP'97)*, volume 1330 of *Lecture Notes in Computer Science*, pages 450–462. Springer-Verlag, 1997. 345, 346

[15] J. Heintz, M.-F. Roy, and P. Solernó. Description of the connected components of a semialgebraic set in single exponential time. *Discrete and Computational Geometry*, 6:1–20, 1993. 350

[16] W. Kainz. Spatial relationships-topology versus order. In *Proceedings of the 4th International Symposium on Spatial Data Handling*, volume 2, pages 814–819, 1990. 359

[17] W. Kainz, M. Egenhofer, and I. Greasley. Modelling spatial relations and operations with partially ordered sets. *International Journal of Geographical Information Systems*, 7(3):215–229, 1993. 359

[18] S. Kreutzer. Fixed-point query languages for linear constraint databases. In *Proceedings of the 19th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS'00)*, pages 116–125, 2000. 346

[19] B. Kuijpers. *Topological Properties of Spatial Databases in the Polynomial Constraint Model*. PhD thesis, University of Antwerp (UIA), 1998. 348

[20] B. Kuijpers, J. Paredaens, M. Smits, and J. Van den Bussche. Termination properties of spatial Datalog programs. In D. Pedreschi and C. Zaniolo, editors, *International Workshop on Logic in Databases (LID'96)*, volume 1154 of *Lecture Notes in Computer Science*, pages 101–116. Springer-Verlag, 1996. 345

[21] B. Kuijpers, J. Paredaens, and J. Van den Bussche. Lossless representation of topological spatial data. In *Proceedings of the 4th International Symposium on Spatial Databases*, volume 951 of *Lecture Notes in Computer Science*, pages 1–13. Springer-Verlag, 1995. 348

[22] B. Kuijpers, J. Paredaens, and J. Van den Bussche. Topological elementary equivalence of closed semi-algebraic sets in the real plane. *The Journal of Symbolic Logic*, 65(4):1530–1555, 2000. 350

[23] B. Kuijpers and J. Van den Bussche. On capturing first-order topological properties of planar spatial databases. In *7th International Conference on Database Theory (ICDT'99)*, volume 1540 of *Lecture Notes in Computer Science*, pages 187–198, 1999. 345

[24] B. Kuijpers and V. Vianu. Topological queries. In J. Paredaens, G. Kuper, and L. Libkin, editors, *Constraint databases*, chapter 2, pages 231–274. Springer-Verlag, 2000. 345, 346

[25] Ch. H. Papadimitriou, D. Suciu, and V. Vianu. Topological queries in spatial databases. *Journal of Computer and System Sciences*, 58(1):29–53, 1999. An extended abstract appeared in PODS'96. 344, 345, 346, 359

[26] J. Paredaens, G. Kuper, and L. Libkin, editors. *Constraint databases*. Springer-Verlag, 2000. 345, 346

[27] I. Pratt. First-order qualitative spatial representation languages with convexity. *Spatial Cognition and Computation*, 1:181–204, 1999. 345

[28] I. Pratt and O. Lemon. Ontologies for plane, polygonal mereotopology. *Notre Dame Journal of Formal Logic*, 38(2):225–245, Spring 1997. 345

[29] I. Pratt and D. Schoop. A complete axiom system for polygonal mereotopology of the real plane. *Journal of Philosophical Logic*, 27(6):621–661, 1998. 345

[30] D. A. Randell, Z. Cui, and A. G. Cohn. A spatial logic based on regions and connection. In *Principles of Knowledge Representation and Reasoning: Proceedings of the 3rd International Conference (KR'92)*, pages 165–176. Morgan Kaufmann, 1992. 344

[31] P. Revesz. *Introduction to Constraint Databases*. Springer-Verlag, 2002. 345, 346

[32] L. Segoufin and V. Vianu. Querying spatial databases via topological invariants. *Journal of Computer and System Sciences*, 61(2):270–301, 2000. An extended abstract appeared in PODS'98. 344, 345, 346

[33] J. Stillwell. *Classical Topology and Combinatorial Group Theory*, volume 72 of *Graduate Texts in Mathematics*. Springer-Verlag, 1980. 348

[34] M. Talamo and P. Vocca. A data structure for lattice representation. *Theoretical Computer Science*, 175(2):373–392, 1997. 359

[35] J. Ullman. *Principles of Database and Knowledge-Base Systems, volumes I and II*. Computer Science Press, 1989-1990. 345

[36] J. Van den Bussche and L. Cabibbo. Converting untyped formulas to typed ones. *Acta Informatica*, 35(8):637–643, 1998. 358