Lossless Representation of Topological Spatial Data

Bart Kuijpers[1], Jan Paredaens[1] and Jan Van den Bussche[2]*

[1] University of Antwerp (UIA), Dept. Math. & Computer Sci.,
Universiteitsplein 1, B-2610 Antwerp, Belgium
Email: {kuijpers, pareda}@uia.ua.ac.be
[2] INRIA Rocquencourt, BP 105, F-78153 Le Chesnay Cedex, France
Email: jan.van_den_bussche@inria.fr

Abstract. We present a data structure used to represent planar spatial databases in the topological data model. Conceptually, such databases consist of points, lines between these points, and areas formed by these lines. The data structure has the distinctive feature that it is geared toward supporting queries involving topological properties of the database only: two databases that are topologically equivalent have the same representation. Moreover, no information is lost in this way: two databases that are not topologically equivalent never have the same representation.

1 Introduction

Spatial database applications [7] can be classified according to the particular geometrical concepts that are involved in the interpretation of the spatial information. This interpretation is apparent from the type of queries that are important for the application. For example, in queries involving directions such as "Give all cities on the west bank of the St. Lawrence river north of Québec," only differences in longitude and latitude are important. Other, metric, queries deal only with distances, such as "Is there a highway within ten miles of my house?"

A major class of queries is formed by those involving only properties of the database that are topological in nature. In this class, concepts such as adjacency, connectivity, and containment are in the focus. Queries like "Is there a highway connecting Boston to Portland?" or "Give all states of the US adjacent to the Atlantic" are typical in this respect. Characteristic of topological properties is that they do not distinguish between two databases that can be obtained from each other by a topological deformation. We will call such databases *topologically equivalent* (this notion will be made precise later).

In the present paper, we elaborate on the idea of topological property in the context of databases consisting of points, lines between these points, and areas formed by these lines [13]. A survey of application domains that can be

* On leave from the University of Antwerp. Research assistant of the Belgian National Fund for Scientific Research.

modeled in this manner was given by Laurini and Thompson [10]. In particular, applications that are topological in nature are common in this context. As an illustration of topologically equivalent databases, we refer to subway or railroad maps such as the one depicted in Fig. 1. Such maps are topological deformations of reality: the length of the lines has no correspondence to the actual length of the trajectory, and the physical track is not as straight as its drawing in the map. We say that such a map is topologically equivalent with a classical map that obeys the reality more closely.

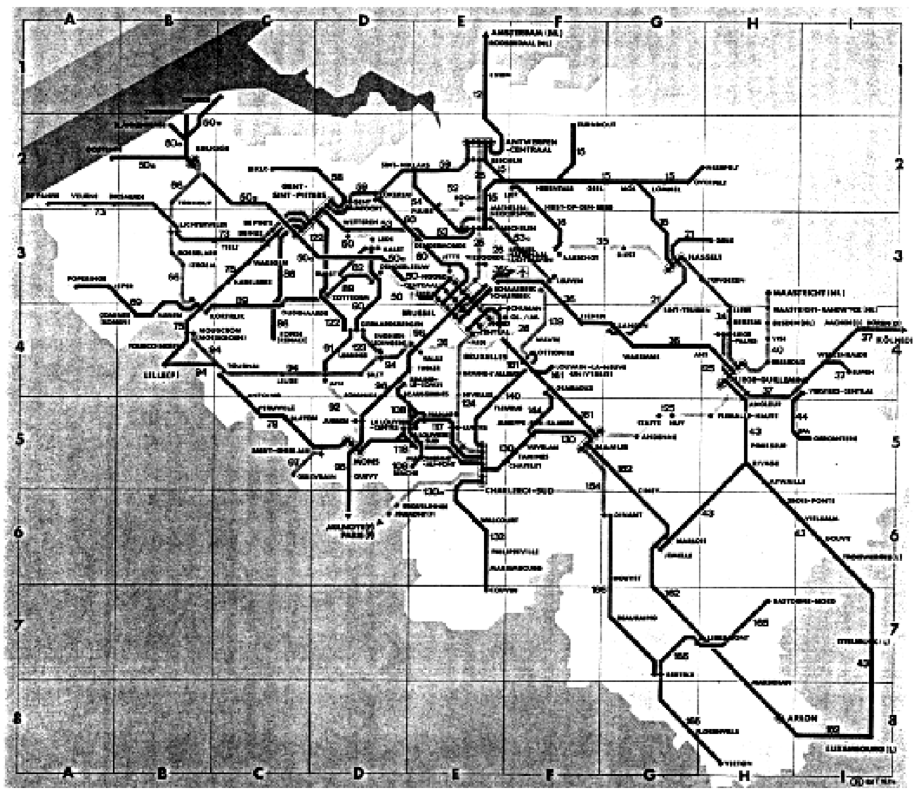

Fig. 1. A railroad map

A common representation of point-line-area spatial databases is by a data structure listing for each point its incident lines and its adjacent areas, arranged in the order in which they appear as one proceeds clockwise around the point. We call this data structure an observation-structure. Essentially this structure underlies the TIGRIS system [8], as well as the topological layer of the ARC/INFO system [12], and the original design of the cartography system of the Census Bureau of the United States [1]. It is also a common data structure for planar graph embeddings [15].

Of course, not just any structure consisting of a number of point names together with arbitrary circular listings of line and area names is the description of a database. We will show that those structures that are "sound", in this sense, can be effectively recognized by an efficient algorithm. This can be viewed as a necessary and sufficient integrity test for the topological data model (also known as "error identification" [10, Chapter 5.3]).

If only topological properties are under consideration, it may be desirable to be able to work with a representation of the database which is *topologically invariant*, meaning that two topologically equivalent databases will be represented identically. Ideally, a representation should also be *lossless*, in the sense that two databases that are *not* topologically equivalent will be represented differently. It is clear that the representation of a database by means of observation-structures, as mentioned above, is topologically invariant. The issue of its losslessness has been somewhat neglected, however. In fact, we will show that it is not lossless.

Although, as we will see, the same observation-structure can represent spatial databases of quite drastically different appearances, we will also show that this phenomenon has one single cause. Indeed, we will show that by explicitly marking one of the areas to be the unbounded one (i.e., the infinitude of space on the "outside" of the database), losslessness is achieved. The formal proof of this claim proceeds by reasoning on the nesting structure of the connected components of the database, and involves the careful composition of local isotopies.

It might seem probable to the reader that the issue of losslessness has already been addressed in the mathematical theory of planar graph embeddings or in the field of planar graph drawing. Indeed, this was also the sentiment of the authors at the initial stages of this investigation. However, this is not the case. In mathematics, the primary interest lies in embeddings on the sphere rather than in the plane. Topological equivalence on the sphere is not equal to topological equivalence in the plane. In graph drawing, one is interested in drawing *one particular* planar embedding of a given planar graph, according to certain criteria, rather than characterizing a unique such embedding in terms of a certain data structure. Moreover, both approaches do not consider the areas as first-class objects.

Our work is driven by the same motivations as those concerning the large body of work done on "spatial relationships" in spatial databases. In particular the various topological relationships that can exist between two specified spatial objects have been extensively investigated by Egenhofer and his collaborators [2, 3, 4, 5]. One way to think of our results is that we generalize these ideas to global topological properties of the entire spatial database [14]. The issues are also relevant from a user interface point of view: a topologically invariant, lossless representation of the database corresponds to an interface which allows the user to concentrate only on the topological aspects of the spatial data, and on all of them, if he so desires.

2 Spatial Databases and Observations

In this section, we define what a spatial database is in the present discussion. We will also introduce the notion of an observation.

In the following, we work in the real Euclidean plane.

Definition. A *spatial database* consists of a finite set of named points, a finite set of named lines and a finite set of named areas. Each point name is assigned to a distinct point in the plane. Each line name is assigned to a distinct non-selfintersecting continuous curve[1] in the plane that starts and ends in a named point and does not contain any other named points except these. Each area name is assigned to a distinct area formed by the named lines.

We remark that this definition allows lines to start and end in the same point, i.e., the database may contain loops. It also may contain more than one line between the same two points. Fig. 2 gives an example of a spatial database. This database has eight points, ten lines and five areas.

We apply the following notational convention throughout the remainder of the paper: Roman characters $p, q, \ldots$ denote point names, Roman capitals $A, B, \ldots$ denote line names and Greek characters $\alpha, \beta, \ldots$ are used for area names.

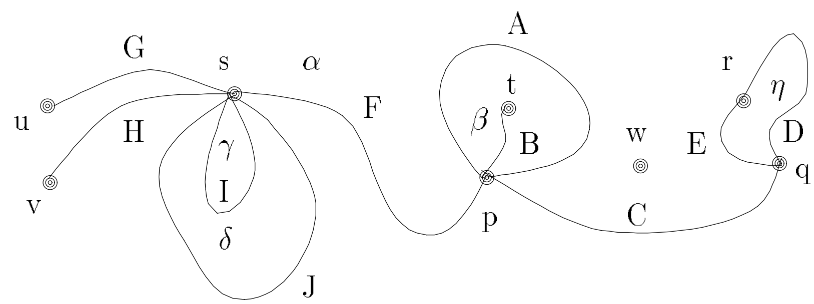

Fig. 2. An example of a spatial database

As a first step in achieving an effective, finite representation of a spatial database, we introduce the notion of *an observation of a spatial database from one of its points*.

For each named point in a spatial database, we make a circular alternating list of area names and line names corresponding respectively to the areas and lines

[1] Formally speaking, a simple Jordan Curve [11].

that an observer, placed in the named point, sees when he makes one clockwise full turn and scans the environment of the point. This is illustrated in Fig. 3. There, the alternating list for the point with name p is $(\alpha\ B\ \alpha\ A\ \beta\ C\ \beta\ A)$.

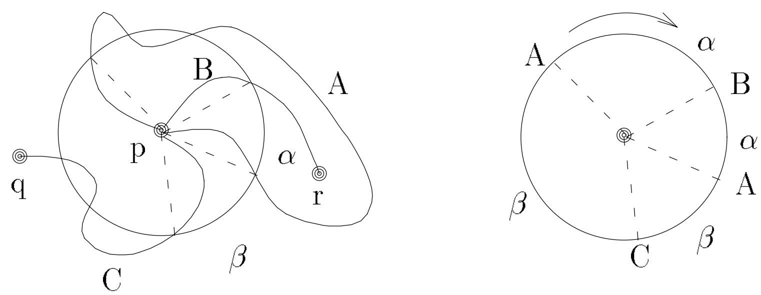

Fig. 3. An observation of a figure from one of its points

We make this more formal. A point p is the endpoint of a finite number of lines (and loops) of the spatial database. We choose a circle with center the point p and a radius such that the circle has at least two intersection points with each loop in p and at least one intersection point in each line ending in p. Such a circle always exists. In Fig. 3, two lines start from the point p and there is one loop. The circle in Fig. 3 has an appropriate radius for it cuts the loop in at least two points and the lines in at least one.

We next mark on the chosen circle the intersection points where the lines and loops "first" intersect with the circle as we follow the lines and loops starting from the point. For a line there is one such point, for a loop there are two such points (one on the right and one on the left). Knowing these markings, we make a circular alternating list of line names and area names, as is illustrated in Fig. 3.

A point which is isolated from the remainder of the database gives rise to an observation consisting of one single area name. For example, the observation of the spatial database of Fig. 2 from the point w is (α). The concept of observation is well-defined. Indeed:

The definition of observation is independent of the chosen radius. Any radius that is sufficiently small produces the same circular list. We can therefore, independently of a radius, write $O_{\mathcal{D}}(p)$ for an *observation of the spatial database $\mathcal{D}$ from one of its points named p.*

We refer to the list of observations of a spatial database from each of its named points as *the observation of the spatial database*.

3 A Lossless Representation

In this section, we use the notion of observation, as a building block of a data structure that is a topologically invariant and lossless representation of a spatial database.

In order to formally specify the notions of topological invariance and losslessness we first need to define "topological equivalence" among spatial databases. Fig. 4 depicts two topologically equivalent databases. Intuitively, two spatial databases are topologically equivalent if one can be obtained from the other by a continuous deformation. In other words, there is a "continuous motion picture" in the plane by which one is transformed into the other.

The mathematical formalization of such "motion pictures" is given by the notion of *isotopy* [11]. An isotopy h is a continuous series $(h_t \mid 0 \leq t \leq 1)$ of homeomorphisms of the plane. We thus define:

Definition. Two spatial databases $\mathcal{D}_1$ and $\mathcal{D}_2$ are called *topologically equivalent* if there exists an isotopy h such that $h_0(\mathcal{D}_1) = \mathcal{D}_1$ and $h_1(\mathcal{D}_1) = \mathcal{D}_2$, with the understanding that h respects the names of points, lines and areas.

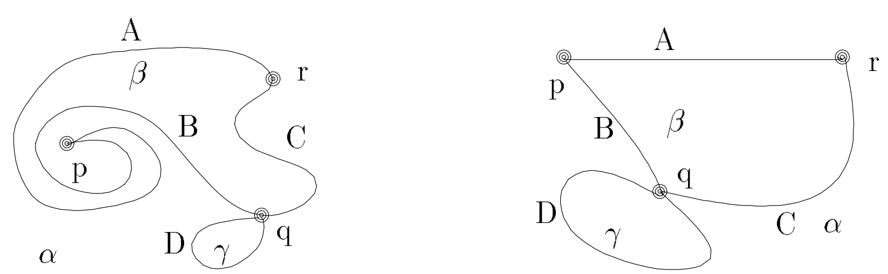

Fig. 4. Two topologically equivalent databases

Definition. A representation of a spatial database is called *topologically invariant* if any two topologically equivalent spatial databases are represented in the same way. A representation of a spatial database is called *lossless* if any two spatial databases that are not topologically equivalent are distinguished by the representation.

As motivated in the Introduction, it may be desirable to represent spatial databases by means of data structures that are topologically invariant and lossless. The data structures of Sect. 2 are topologically invariant representation of a spatial database. More precisely:

Property. Two topologically equivalent databases have the same observation.

The proof of this fact is quite straightforward. A circle with center the point p, that is located inside a topological deformation of a (possibly other) circle that was used to obtain an observation from p, gives rise to an identical observation as the original circle.

Is this representation, on the other hand, lossless? The answer is *no*. Fig. 5 contains two spatial databases that are represented by identical lists of observations. These databases are however clearly not continuously deformable into each other.

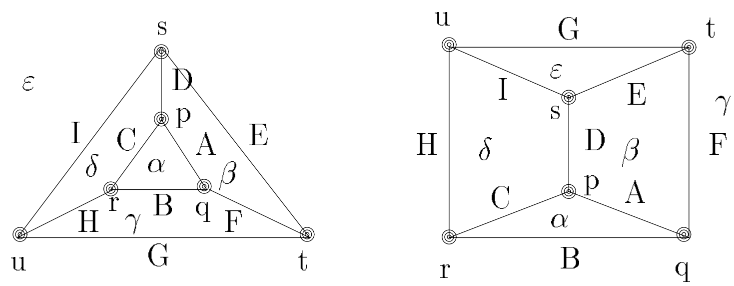

Fig. 5. Two spatial databases that are not topologically equivalent but that have the same observation list

A key observation that can be made in Fig. 5 is that ε is the unbounded region in the first database, but not in the second. Indeed, a necessary condition for two spatial databases to be topologically equivalent is that for both the unbounded area has the same label. We note that a spatial database has only one unbounded area. It is therefore justified to reserve a special label for the unbounded area: α^∞.

In Fig. 6, we have a simpler example that captures the essence of the problem. These two databases have the same list of observations. If embedded in the sphere, these two databases could be deformed into each other (by pulling the loop over the sphere). In the plane, however, this is not possible. Here however it is obvious that giving the unbounded area a special status is also sufficient to distinguish between the two databases.

Is giving, in addition to the list of the observations of a spatial database from all its named points, the information that the unbounded area has the name α^∞ not only necessary but also sufficient to obtain losslessness? We answer this question affirmative in the following theorem.

Theorem. Suppose we have two databases that give the unbounded area the

Fig. 6. Is α^∞ enough?

same name α^∞. If they are not topologically equivalent they have different observations.

We have to prove that any two databases with the same list of observations are isotopic. The proof of this statement is rather technical. It can be proved by induction on the connected components of the figure. In the case of one connected component the proof is by induction on the number of lines of the figure and involves the construction of an "local" isotopy. The basis of the latter induction is trivial. For connected spatial databases, we can always drop out one of the lines so that the resulting databases are still connected. The resulting databases also have the same observations. So, by assumption, they are isotopic. With the information of the observations, there is isotopically only one way to put the line back. What remains is gluing the local isotopies together in the proper way into one global isotopy.

We now have a data structure that is invariant and lossless. We therefore give it a name.

Definition. For a given spatial database $\mathcal{D}$, we call the data structure $(\mathcal{P}, \mathcal{L}, \mathcal{A}, \alpha^\infty, \mathrm{Obs}())$ the *PLA-structure of* $\mathcal{D}$ if $\mathcal{P}$ is the set of point names of $\mathcal{D}$, $\mathcal{L}$ is the set of line names of $\mathcal{D}$, $\mathcal{A}$ is the set of area names of $\mathcal{D}$, α^∞ is the name of the unbounded area of $\mathcal{D}$, and Obs() is a function that associates with each element p of $\mathcal{P}$, $O_{\mathcal{D}}(p)$, the observation of $\mathcal{D}$ from p.

4 An Efficient Algorithm to Recognize PLA-structures

In this section, we deal with error identification problems in PLA-structures. We will give an algorithm that can be used as an integrity test for representations of spatial databases. More precisely, not every structure consisting of a number of point names, line names, area names together with arbitrary circular listings of line and area names for each point name describes a spatial database.

For example, it is easily verified that the structure $\mathcal{S} = (\{p\}, \{A\}, \{\alpha^\infty, \beta\}, \alpha^\infty, \mathrm{Obs}())$, with $\mathrm{Obs}(p) = (\alpha^\infty A)$ is not the PLA-structure of any spatial database. On the other hand for $\mathrm{Obs}(p) = (\alpha^\infty A \beta A)$, Fig. 7 depicts a database for which it is the PLA-structure. So, there are structures that "look" like PLA-structures, but make no real sense. This broader class of structures can be characterized as follows.

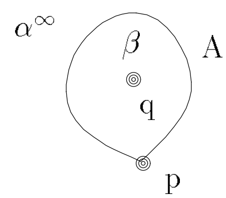

Fig. 7. A loop

Definition. We call any tuple

$$(\mathcal{P}, \mathcal{L}, \mathcal{A}, \alpha^\infty, \mathrm{Obs}())$$

a structure if $\mathcal{P}$, $\mathcal{L}$, respectively $\mathcal{A}$ are finite sets, α^∞ is an element of $\mathcal{A}$, and Obs() is a function that associates to each element of $\mathcal{P}$ or a circular list of alternatingly elements of $\mathcal{L}$ and $\mathcal{A}$ or just one element of $\mathcal{A}$.

Clearly, PLA-structures are exactly those structures that represent a spatial database.

In this section we give a positive answer to the following question.

Is it syntactically decidable whether a given structure is a PLA-structure?

We will describe an efficient decision procedure to solve this problem. Suppose we are given an arbitrary structure $\mathcal{S} = (\mathcal{P}, \mathcal{L}, \mathcal{A}, \alpha^\infty, \mathrm{Obs}())$. To decide whether or not it is a PLA-structure of a spatial database, we start by constructing what we will refer to as the *dual graph* $\mathcal{G}_\mathcal{S}$ of $\mathcal{S}$. Fig. 8 illustrates this construction for a loop and a line. $\mathcal{G}_\mathcal{S}$ is an undirected graph and consists of a set of vertices V and a set of edges E.

V contains a vertex for each point name, for each line name and each area name in $\mathcal{S}$. In addition to these, V contains a different vertex for each occurrence of a line name in an observation (called an observation vertex of the line) and a different vertex for each occurrence of an area in an observation (called an observation vertex of the area). E contains an edge between the vertex of a line name A and and each observation vertex of A, an edge between the vertex of an area name α and and each observation vertex of α. If $\mathrm{Obs}(p) = (\alpha)$, E contains an edge between the vertex of p and the corresponding observation vertex of α, and an edge from the corresponding observation vertex of α to itself. If, on the

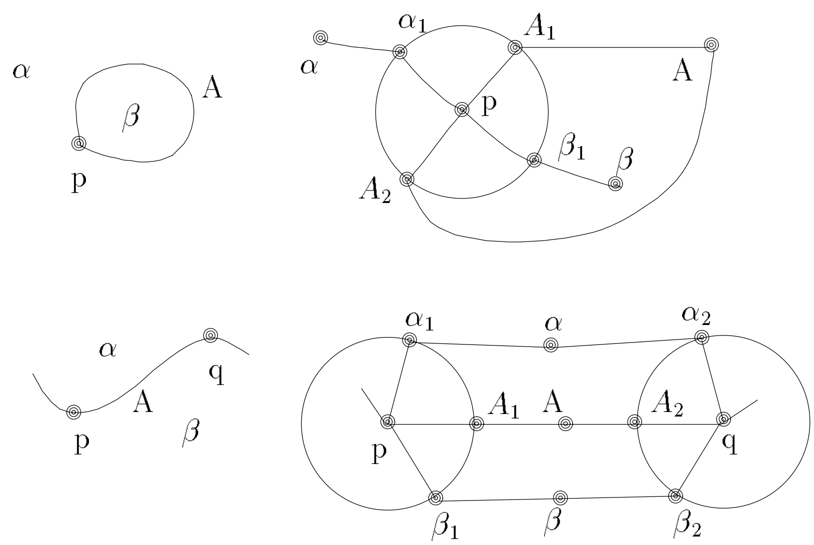

Fig. 8. The dual graph of a loop and a line

other hand, $\mathrm{Obs}(p) = (A_1\alpha_1 \ldots A_n\alpha_n)$, E contains an edge between the vertex of p and the $2n$ corresponding observation vertices, and an edge between two successive observation vertices and between the observation vertices of α_n and of A_1. We refer to the latter series of $2n$ edges as the *cycle of* p.

We now give our decision procedure in terms of decidable properties of the dual graph and of the structure $\mathcal{S}$.

Decision procedure. The following are necessary and sufficient conditions for a structure $\mathcal{S}$ to be a PLA-structure:
• If $\alpha A\beta$ occurs in the observation from p, then there is a point q (possibly equal to p) such that $\beta A\alpha$ occurs in the observation from q. Both occurrences of A are different and are the only occurrences of A in an observation.
• $\mathcal{G}_{\mathcal{S}}$ is a connected graph.
• $\mathcal{G}_{\mathcal{S}}$ is a planar graph.

We will briefly outline the (rather technical) proof of the correctness of this procedure. The conditions are clearly necessary. Indeed, if $\mathcal{S}$ is a PLA-structure then there is a spatial database $\mathcal{D}$ of which $\mathcal{S}$ is the PLA-structure. We can very easily transform the database in the dual graph $\mathcal{G}_{\mathcal{S}}$. $\mathcal{G}_{\mathcal{S}}$ is planar and connected. Furthermore, each line A has to endpoints, say p and q. This results in two occurrences of A in the way as specified the first condition.

To prove sufficiency, we start with a planar embedding of $\mathcal{G}_{\mathcal{S}}$ in the plane. This embedding exists because of the third condition. For each point that is located outside the interior of its cycle, we switch the interior and the exterior of the cycle with the result that it is inside. With a similar inversion, we can make sure that the vertex of α^{∞} is in the unbounded area of the embedding. For some of the points the cycles may have the wrong, i.e., counterclockwise orientation. It can be proved that the points with a wrong orientation are "clustered" together and can all be turned without affecting the points with a clockwise cycle. It is easy to derive the wanted spatial database from an embedding of the dual graph in which α^{∞} is located in the unbounded area and each point is located inside its correctly oriented cycle.

We end this section with the comment that this decision algorithm is efficient. The construction of the dual graph is linear in the size of the structure. Also the first condition can be checked in linear time. For both the second and the third condition efficient algorithms are known [6].

5 Further Remarks

In Sect. 3 we have defined two spatial databases to be topologically equivalent if they are isotopic. Intuitively this means that one database can be continuously deformed into the other. In classical topology, however, two figures are usually called topologically equivalent if they are *homeomorphic*. Homeomorphisms form a broader class of continuous transformations than the one that we have considered. The reflection ρ of the plane along the y-axis is a homeomorphism of the the plane but not an isotopy. ρ transforms a left hand continuously a right hand, as is illustrated in Fig. 9. This continuous transformation can not be achieved by a continuous deformation in the plane. To deform a left hand continuously into a right hand it is necessary to leave the plane and to use a third dimension.

The following well-known result classifies homeomorphisms of the real plane as either being continuous deformations or composed of a reflection followed by a continuous deformation (see e.g. [16]).

Property. A homeomorphism of the plane is either isotopic to the identity or to ρ.

This property allows us to define an invariant and lossless representation for spatial databases if toplogically equivalent corresponds to homeomorphic. The reflection ρ reverses the orientation and as a consequence also each observation of a spatial database from one of its points. For the databases in Fig. 9 this is best visible for the point p. Two isotopic spatial databases have the same clockwise, respectively counterclockwise observations. For two homeomorphic but non-isotopic spatial databases the clockwise observations of the first correspond to the counterclockwise observations of the other and vice versa.

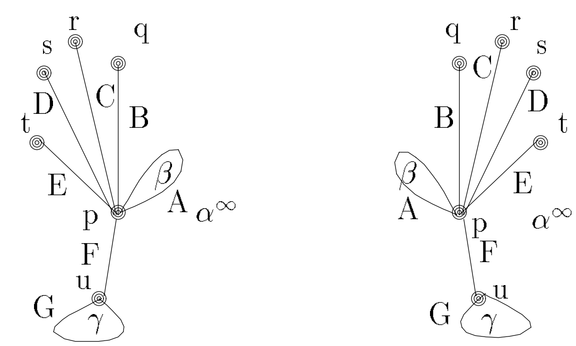

Fig. 9. Two spatial databases: a left and a right hand

Acknowledgements

The authors would like to thank Rudi Penne and Johan Van Biesen for many helpful discussions. Furthermore, we would like to thank Jan Hidders [9] and Xiao-Song Lin for useful suggestions concerning the proof of losslessness.

References

1. J.P. Corbett. Topological Principles of Cartography. Technical Paper No. 48, US Bureau of the Census, Washington, DC, USA: US Government Printing Office, 1979.
2. M. Egenhofer. Reasoning about Binary Topological Relations. *Advances in Spatial Databases, Lecture Notes in Computer Science, 525*, (eds. O. Günther and H.-J. Schek), 143–160, Springer-Verlag, 1991.
3. M. Egenhofer. Topological Relations between Regions of $\mathbf{R}^2$ and $\mathbf{Z}^2$. *Advances in Spatial Databases, Lecture Notes in Computer Science, 692*, (eds. D. Abel and B.C. Ooi), 316–336, Springer-Verlag, 1993.
4. M. Egenhofer and R. Franzosa. On the equivalence of topological relations. *International Journal of Geographical Information Systems*, 523–542, 1994.
5. M. Egenhofer and J. Herring A mathematical framework for the definition of topological relationships. *Proceedings of the Fourth International Symposium on Spatial Data Handling*, Zurich, Switzerland (eds. K. Brassel and H. Kishimoto), 803–813, 1990.
6. S. Even. *Graph Algorithms.* Computer Science Press, 1979.
7. R.H. Güting. An Introduction to Spatial Database Systems. *The VLDB Journal*, 3(4), 1994.
8. J. Herring. TIGRIS: Topologically Integrated Geographic Information Systems. *Proceedings of Auto Carto 8 Conference*, Baltimore, MD, (ed. N.R. Chrisman), 1987.

9. J. Hidders. An Isotopic Invariant for Planar Drawings of Connected Planar Graphs. Computing Science Note, 95/04, Eindhoven University of Technology, 1995.
10. R. Laurini and D. Thompson. *Fundamentals of Spatial Information Systems.* The A.P.I.C. Series, 37, Academic Press, 1992.
11. E.E. Moise. *Geometric Topology in Dimensions 2 and 3.* Graduate Texts in Mathematics, 47, Springer-Verlag, 1977.
12. S. Morehouse. The architecture of ARC/INFO. *Proceedings of the Auto Carto 9 Conference*, Baltimore, MD, American Society for Photogrammetry and Remote Sensing/American Congress for Surveying and Mapping, 266–277, 1989.
13. J. Paredaens. Spatial Databases. The Final Frontier. *Database Theory - ICDT '95, Lecture Notes in Computer Science, 893*, 14–32, Springer-Verlag, 1995.
14. J. Paredaens, J. Van den Bussche and D. Van Gucht. Towards a Theory of Spatial Database Queries. *Proceedings of the 13th ACM SIGACT-SIGMOD-SIGART Symposium on the Principles of Database Systems*, 279–288, ACM Press, 1994.
15. P. Preparata and M.I. Shamos. *Computational geometry.* Springer-Verlag, 1985.
16. J. Stillwell. *Classical Topology and Combinatorial Group Theory.* Graduate Texts in Mathematics, 72, Springer-Verlag, 1980.