Photogrammetric Engineering & Remote Sensing 60 (6): 783-790, 1994.

Pre-Processing Queries with Spatial Constraints*

Max J. Egenhofer
National Center for Geographic Information and Analysis
and
Department of Surveying Engineering
Department of Computer Science
Boardman Hall
University of Maine
Orono, ME 04469-5711, U.S.A.
max@mecan1.maine.edu

Abstract

This paper discusses strategies for pre-processing spatial queries before executing them in a geographic database. A computational method has been devised, which is capable of assessing whether or not a spatial query is consistent, i.e., that its constraints are free of any self-contradictions. If the query is inconsistent, it can be rejected immediately without a need to process it in the database. The query evaluation is based on algebraic properties of spatial relations such as symmetry, converseness, transitivity, and composition and employs constraint propagation in a constraint network to detect conflicts. Examples are given for how this method works for spatial queries with constraints about topological relations, for which a comprehensive algebraic formalization exists.

1 Introduction

Geographic databases are typically very large (Smith and Frank, 1990). In order to obtain acceptable response times, appropriate measures must be employed when processing requests to retrieve spatial information. This is the tasks of *spatial query processing*, which is concerned with reducing the time necessary to report the result of a spatial query in a spatial database. Traditionally, attempts to improve the performance of geographic databases have focused on the development of access structures for fast retrieval of spatial data from secondary storage media. The methods designed are numerous and well documented. Comprehensive surveys and comparisons can be found in Samet (1989) and Kriegel *et al.* (1989). While such spatial access methods are necessary for the fast retrieval of spatial data from large databases (Frank, 1989), additional strategies considering the semantics of the spatial operations must be pursued to ensure the fast processing of spatial queries.

A particular aspect of processing spatial queries that has not been addressed in the past is the assessment of the *consistency* of a spatial query with respect to the semantics of spatial constraints. Constraints in queries are Boolean combinations (AND, OR, NOT) of predicates that must hold true among a set of objects. For example, a spatial query to retrieve specific rivers may include constraints like, “the river must be crossed by a highway and be the boundary between two states.” Consistency of a query means that there is no logical contradiction among the individual constraints. For example, a query that asks for all cities that are both larger than 50,000 and smaller than 10,000 inhabitants is inconsistent as it has a contradicting constraint. Consistency must not be confused with *correctness*, which describes the relationship between the data stored in a database and the real world. Consistency of a query matters for query processing, because any query whose formulation has contradicting constraints can be rejected immediately

* This work was partially funded by grants from Intergraph Corporation. Additional support from NSF for the NCGIA under grant number SES 88-10917 is gratefully acknowledged.

and need not be processed against the database. Processing queries that ask for inconsistent configurations would only make sense as a mechanism to check a database for its integrity.

The detection of inconsistencies in a query will be referred to as *query pre-processing* as it can be applied *before* starting the actual query processing. Pre-processing may considerably speed up queries that cannot have a valid answer. It will never produce a result, independent of the data stored in the database[1]. In lieu of searching for all large cities (>50,000) and then testing the cities found for the second constraint (<10,000), it is more reasonable—and more economical—to assess the conditions, independently of the data, in order to identify potential contradictions.

In the past, most efforts to improve the processing of spatial queries have focused on identifying the best sequence in which spatial constraints should be executed (Menon and Smith, 1989; Lu and Han, 1990; 1992) and on developing strategies to retrieve information from distributed spatial and non-spatial databases (Ooi and Sacks-Davis, 1989; Aref and Samet, 1991). A rule-based approach to spatial query processing attempts to find query execution plans that optimize for spatial joins using meta information about spatial index structures (Becker and Güting, 1992). Specialized spatial query processors optimize multiple map-overlay operations (Yost and Skelton, 1990; Dohrenbeck and Egenhofer, 1991) and queries in line-segment databases such as road networks (Hoel and Samet, 1991). An approach to processing spatial queries iteratively uses a time constraint, within which an approximate result must be provided, or an accuracy requirement with which a query result must comply (Barrera *et al*., 1992).

Fast query evaluation strategies have been built into traditional relational databases over such simple data types as integers and character strings (Kim *et al.*, 1985), however, current geographic information systems lack such tests for spatial relations. The assessment of the consistency of spatial queries is distinct due to the specific properties of the spatial relations involved (Egenhofer and Sharma, 1992). For example, if a spatial query contains the constraint that *A* overlaps with *B* and *B* is completely inside of *A*, based on the semantics of the spatial relations overlap and inside, the query will never produce a result and can be rejected even before examining the actual data stored in the database. Sometimes, the evaluation of the consistency of a query may be straightforward if all spatial relations are related by such standard properties as transitivity. For example, the description *A* contains *B*, and *B* contains *C*, and *A* disjoint *C* is obviously inconsistent because, by the transitive property of contains, *A* must contain *C*, which contradicts with the statement that *A* is disjoint form *C* . More difficult are such evaluations if a larger set of objects is involved or multiple relations occur. For example, the query in Figure 1, formulated in a spatial SQL dialect (Egenhofer, 1991b), contains several topological constraints, whose consistency evaluation is not obvious. A possible strategy for evaluating the consistency of the constraints would be to draw a figure of the objects that satisfies the constraints, and to conclude from a successful drawing that the constraints do not contradict. Two major disadvantages are associated with this approach: first, it relies heavily on subjective decision-making and, therefore, it is difficult to implement as a computational process. Second, it cannot deal easily with incomplete information as each drawing represents a single configuration.

[1]This assumes that the database itself is consistent, i.e., that it has no internal logical contradictions.

```
SELECT   lake.name
FROM     state, county, lake
WHERE    state.geometry CONTAINS county.geometry and
         county.geometry CONTAINS lake.geometry and
         (state.geometry DISJOINT lake.geometry or
         state.geometry MEET lake.geometry)
```

Figure 1: A query in a spatial SQL dialect that contains several topological constraints.

This paper proposes a computational mechanism to assess some spatial queries for the absence of logical contradictions. It focuses on the evaluation of topological constraints as disjunctions (OR-conditions) of binary spatial predicates over the same pair of spatial objects. Having applied such an evaluation before processing a spatial query will allow a query processor to find out whether the particular query is worthwhile to process against the database or whether it cannot produce any result at all. The assessment of disjunctions of spatial predicates exploits algebraic properties of the spatial relations such as symmetry, transitivity, converseness, and composition as defined in a high-level spatial data model (Egenhofer and Herring, 1991). It is important to note that all evaluations will be based on the descriptions of these properties of spatial data, not their actual values.

The remainder of this paper is structured as follows: Section 2 reviews the data model we use for topological relationships, including a discussion of their properties. Section 3 describes how constraints in spatial queries can be represented as a constraint network. Section 4 compiles the consistency constraints that must hold in a constraint network and Section 5 applies them to evaluate the consistency of spatial queries. Our conclusions in Section 6 summarize the major results of the paper and identify directions for future research.

2 Spatial Relations

Conditions among spatial data are commonly expressed in terms of *spatial relations* (Frank and Mark, 1991) or *spatial prepositions* (Herskovits, 1986). Some examples are *inside*, *north*, and *far* (Freeman, 1975; Peuquet, 1986). Most categorizations of spatial relations distinguish between topological relations, direction, and distance (Pullar and Egenhofer, 1988; Worboys and Deen, 1991). This study focuses on topological relations between spatial regions in continuous 2-dimensional space IR^2 (Egenhofer and Herring, 1990; Egenhofer and Franzosa, 1991). *Topological relations* are preserved under groups of transformations, such as scaling, rotation, and translation, and describe concepts of adjacency, containment, and intersection. A rigorous computational method has been designed, which allows for reasoning about binary topological relations between spatial regions (Egenhofer, 1991a) and to infer the consistency of complete and incomplete topological information (Egenhofer and Sharma, 1992). The model for binary topological relations is based on the usual concepts of point-set topology with open and closed sets (Alexandroff, 1961) distinguishing the interior of a set *A*, denoted by $A°$; and the boundary of *A*, denoted by ∂A. The definition of binary topological relations between two regions, *A* and *B*, is based on the four intersections of *A*'s boundary and interior with the boundary and interior of *B* (Egenhofer and Franzosa, 1991). A 2×2 matrix, *M*, called the 4-intersection, concisely represents these criteria (Equation 1).

$$M = \begin{pmatrix} \partial A \cap \partial B & \partial A \cap B° \\ A° \cap \partial B & A° \cap B° \end{pmatrix} \quad (1)$$

By considering the values empty (ø) and non-empty (¬ ø), one can distinguish sixteen binary topological relations, eight of which can be realized for two regions with connected boundaries if the objects are embedded in IR^2 (Egenhofer and Herring, 1990). They are *disjoint*, *meet*, *equal*, *inside*, *contains*, *covers*, *coveredBy*, and *overlap* (Figure 2). This set provides a complete coverage and is mutually exclusive so that exactly one of these topological relations holds true between any two regions (Egenhofer and Franzosa, 1991). The formalization is used extensively, e.g., to describe more complex spatial relations (Herring, 1991; Pigot, 1991; Clementini *et al.*, 1992; Cui *et al.*, 1992; Hadzilakos and Tryfona, 1992), in spatial query languages (Svensson and Zhexue, 1991; de Hoop and van Oosterom, 1992), and as a basis for cognitive-linguistic studies (Mark and Egenhofer, 1992).

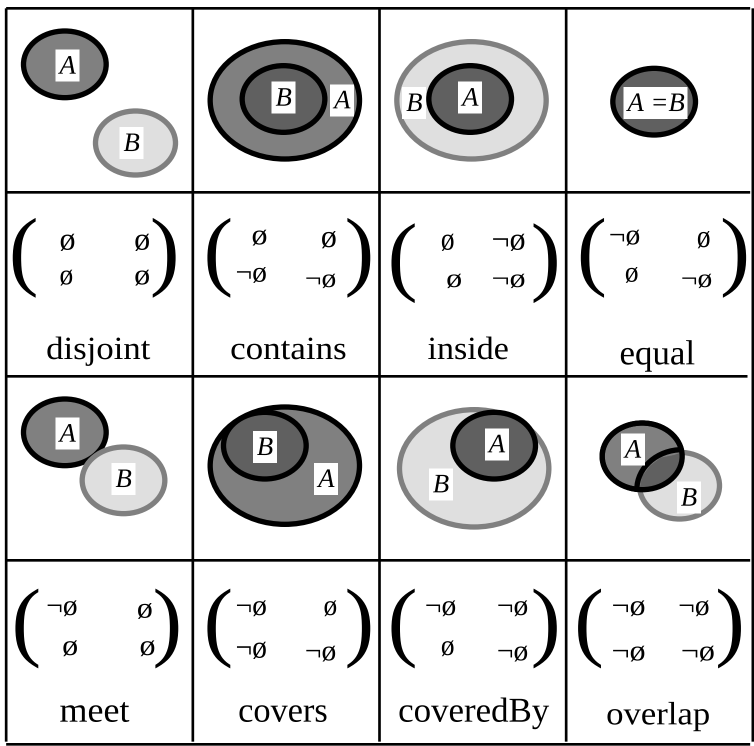

Figure 2: The eight topological relations between two regions with connected boundaries for the 4-intersection $\begin{pmatrix} \partial A \cap \partial B & \partial A \cap B^\circ \\ A^\circ \cap \partial B & A^\circ \cap B^\circ \end{pmatrix}$.

Incomplete topological information can be expressed as the disjunction ( ) of several topological relations. For example, from Figure 2 we can derive that the constraint, "*A* and *B* should not have common interiors" holds for two configurations (disjoint and meet) and, therefore, can be expressed in terms of the disjunction disjoint (*A*, *B*) meet (*A*, *B*). Likewise, the negation (¬) of a relation can be represented as the complement with respect to the universal relation. For example,

¬ *disjoint (A, B)* = *meet (A, B)* *equal (A, B)* *covers (A, B)* *contains (A, B)* *coveredBy (A, B)* *inside (A, B)* *overlap (A, B)*

There are certain properties and constraints of topological relations that determine whether or not a topological description is consistent (Egenhofer and Sharma, 1992). These properties are ingredients of a *relation algebra*[2] (Tarski, 1941). They are associated with the set of topological relations, not the spatial data themselves.

- The topological relation between every object and itself is *equal*. It is the *identity relation*, as it is reflexive, symmetric, and transitive.

- Each of the eight possible topological relations *t (A, B)* between two spatial objects in $\mathbb{R}^2$ has a *converse* relation $\breve{t}$ *(B, A)*. They are:

  disjoint (A, B) = disjoint (B, A)
  meet (A, B) = meet (B, A)
  equal (A, B) = equal (B, A)
  overlap (A, B) = overlap (B, A)
  inside (A, B) = contains (B, A)
  contains (A, B) = inside (B, A)
  covers (A, B) = coveredBy (B, A)
  coveredBy (A, B) = covers (B, A)

- The *composition* of two binary topological relations t_i *(A, B)* and t_j *(B, C)* over a common object *B*, denoted by $t_i ; t_j$, allows for the derivation of the relation t_k between *A* and *C*. For example, if *A* meets *B* and *B* contains *C* then *A* disjoint *C*. The composition table (Table 1), formally derived elsewhere (Egenhofer, 1991a), depicts the outcome of all 64 possible compositions among all eight topological relations. It shows that all compositions are valid, however, not all of them are unique. Furthermore, it verifies that *equal* is the identity relation, because all compositions with equal result in the initial relation.

[2] Not to be confused with the *relational algebra* (Codd, 1972).

	disjoint	meet	equal	inside	coveredBy	contains	covers	overlap
disjoint	*U*	disjoint, meet, inside, coveredBy, overlap	disjoint	disjoint, meet, inside, coveredBy, overlap	disjoint, meet, inside, coveredBy, overlap	disjoint	disjoint	disjoint, meet, inside, coveredBy, overlap
meet	disjoint, meet, contains, covers, overlap	disjoint, meet, equal, coveredBy, covers, overlap	meet	inside, coveredBy, overlap	meet, inside, coveredBy, overlap	disjoint	disjoint, meet	disjoint, meet, inside, coveredBy, overlap
equal	disjoint	meet	equal	inside	coveredBy	contains	covers	overlap
inside	disjoint	disjoint	inside	inside	inside	*U*	disjoint, meet, inside, coveredBy, overlap	disjoint, meet, inside, coveredBy, overlap
coveredBy	disjoint	disjoint, meet	coveredBy	inside	inside, coveredBy	disjoint, meet, contains, covers, overlap	disjoint, meet, equal, coveredBy, covers, overlap	disjoint, meet, inside, coveredBy, overlap
contains	disjoint, meet, contains, covers, overlap	contains, covers, overlap	contains	equal, inside, coveredBy, contains, covers, overlap	contains, covers, overlap	contains	contains	contains, covers, overlap
covers	disjoint, meet, contains, covers, overlap	meet, contains, covers, overlap	covers	inside, coveredBy, overlap	equal, coveredBy, covers, overlap	contains	contains, covers	contains, covers, overlap
overlap	disjoint, meet, contains, covers, overlap	disjoint, meet, contains, covers, overlap	overlap	inside, coveredBy, overlap	inside, coveredBy, overlap	disjoint, meet, contains, covers, overlap	disjoint, meet, contains, covers, overlap	*U*

Table 1: The composition table for the eight binary topological relations between two regions (Egenhofer, 1991a).

We introduce two other notions: (1) The *universal relation*, *U*, is the union of all possible topological relations. It is valid between any pair of spatial objects. (2) The empty topological relation, ø, describes a non-existing topological relation. It will be used to denote an inconsistent topological description. The converse relations of *U* and ø are *U* and ø, respectively. Likewise, all compositions with the empty relation are empty as well. Except for the composition with the empty relation, all compositions with the universal relations result in the universal relation. Egenhofer and Sharma (1992) have derived these properties from the composition table (Table 1).

3 Query Representation

The properties of a set of spatial relations must be guaranteed for any configuration of spatial objects, otherwise, those objects would violate some fundamental geometric concepts and the geographic database would be in an inconsistent state[3]. This consistency requirement applies also to the constraints of a spatial query. A spatial query is inconsistent if the properties of the spatial relations are not fulfilled.

While the three algebraic properties of identity, converseness, and composition address some of the consistency constraints among the objects involved—they are sufficient for up to three objects—these properties *per se* are not powerful enough to assure the consistency of any arbitrary set of objects. A more generic method is necessary to evaluate a complex set of spatial relations, independent of the number of objects involved. For this goal, we translate the spatial predicates and the spatial objects among which they hold, into a representation for which such consistency evaluations are well known. In essence, the spatial predicates are mapped from a domain in which one attempts to get a result, into a domain in which appropriate means are available to solve the problem at hand. Since the two representations and the mappings between them form a category (Herring *et al.*, 1990), the result of the consistency evaluation in the new domain implies the consistency or inconsistency of the set of spatial predicates in the initial domain (Figure 3).

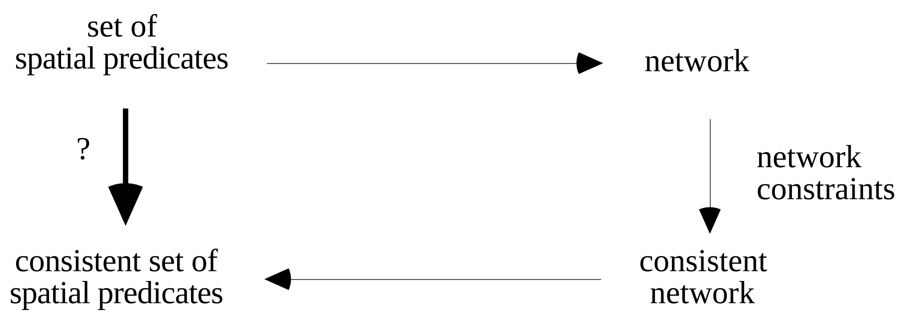

Figure 3: Solving the consistency evaluation of spatial predicates in a constraint network.

To evaluate spatial consistency, we choose to represent the constraints of a spatial query as a *constraint network*. Constraint networks have been extensively investigated in mathematics and computer science (Montanari, 1974; Mackworth, 1977; Freuder, 1978; Maddux, 1990). They provide a means to evaluate formally whether a complex set of interrelated constraints can be satisfied. These abstract concepts are valuable to spatial query pre-processing as they can be applied to solve the concrete problem of whether a complex spatial query is consistent.

A constraint network consists of (1) a set of *nodes*, (2) *directed labeled edges* linking always two nodes, and (3) some *rules* about the semantics of the labels and their combinations. Subsequently, t_{ij} will refer to the label of a directed edge between the nodes i and j, and T will be used for a constraint network. In such a network, a path is a connection between two nodes over edges with the same direction. The path length is then defined as the number of edges along a

[3] This statement must be qualified for temporal GISs (Barrera *et al.*, 1991; Langran, 1992), in which spatial relations may change over time due to deformations of the objects (Egenhofer and Al-Taha, 1992). The properties of the spatial relationships in a consistent spatio-temporal database must be fulfilled at any single state in world time.

path. If the relations are mutually exclusive, the constraint network over *n* objects contains *n* nodes and n^2 directed edges. It forms a complete graph, i.e., between any two nodes, there is exactly one directed edge.

The mapping from the problem domain, which represents multiple spatial constraints among spatial objects, onto a constraint network is straightforward: Each spatial object corresponds to a *node*, and each binary spatial relation is a directed, labeled *edge*. Figure 4 shows two configurations for the same configuration of topological relations. One is a geometric interpretation in IR^2, the other is a constraint network. An alternative representation of such a network is a table with the Cartesian product of all objects and their corresponding binary relations.

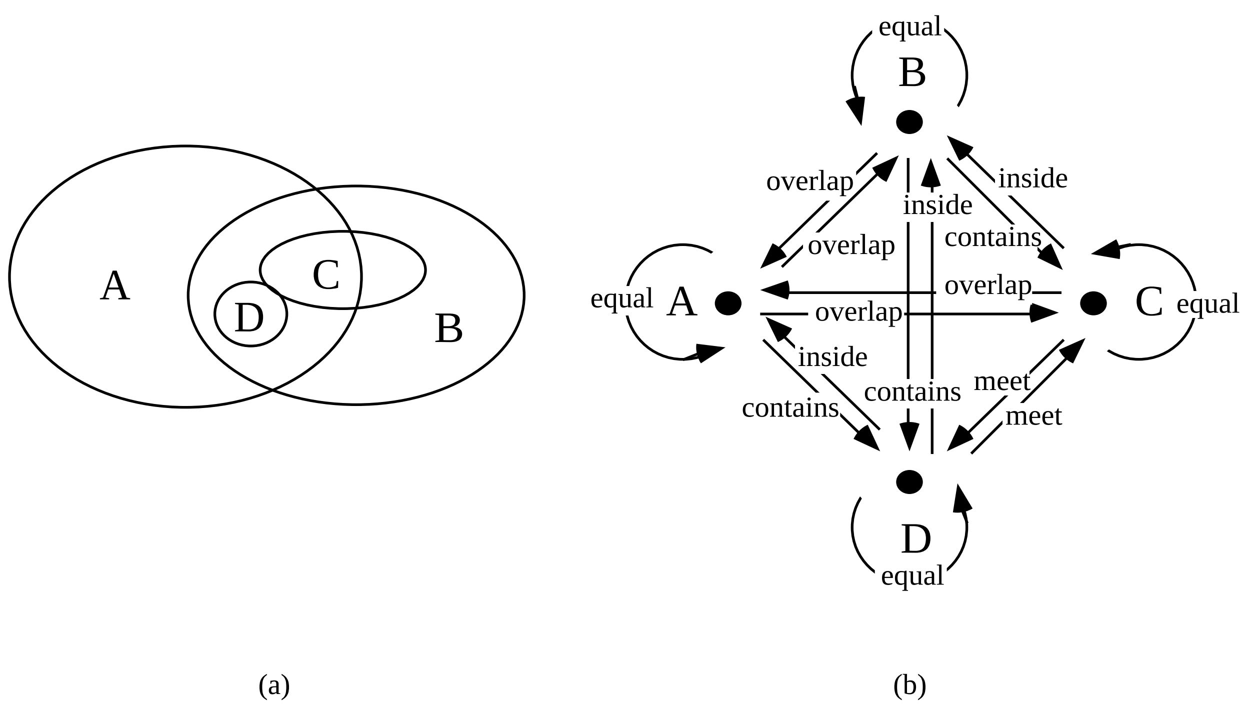

Figure 4: A geometric interpretations of the topological relations between four objects (a) and their representation as a network (b).

4 Consistency Constraints

If the set of all relations is considered a network with the objects being the nodes and the relations between them forming a directed graph, a set of three constraints applies for a consistent constraint network (Mackworth, 1977):

- *Node consistency* implies that between each node and itself the identity relation holds.
- *Arc consistency* guarantees the converseness of the relations between each pair of nodes.
- *Path consistency* implies that the relations that can be derived through all possible compositions do not contradict.

A theorem by Montanari (1974) provides the basis for reducing the problem of guaranteeing path-consistency to the intersection of all binary compositions. It states that a network is path-consistent if the compositions of all paths of length 2 are consistent and no other combinations need to be considered.

In terms of a constraint network over *n* nodes, the three levels of consistency mean that:

- Each node must have a self-loop for the identity relation denoting the node-consistency:

$$t_{AA} = equal \tag{2}$$

- For each edge between two nodes A and B, labeled t_{AB}, there must be an edge from *B* to *A* labeled with the relation that is converse to t_{AB}:

$$t_{AB} = \breve{t}_{BA} \tag{3}$$

- The label of each edge, t_{AB}, must match with the *induced relation,* which results from the intersection (∩) of all *n* compositions of path length 2:

$$t_{AB} = (t_{A1} ; t_{1B}) \cap (t_{A2} ; t_{2B}) \cap \ldots \cap (t_{An} ; t_{nB}) \tag{4}$$

The following examples, which refer to the network in Figure 4b, show how these constraints apply to a set of spatial objects and their topological relations.

Example 1: All nodes are node-consistent, because each of them has a directed edge pointing to itself that is labeled equal.

Example 2: The edge t_{AD} is arc-consistent, because the two relations *A* contains *D* and *D* inside *A* are converse.

Example 3a: The edge t_{AD}, labeled *contains*, is path-consistent, because

$$\begin{aligned} t_{AD} &= (t_{AA} ; t_{AD}) \cap (t_{AB} ; t_{BD}) \cap (t_{AC} ; t_{CD}) \cap (t_{AD} ; t_{DD}) \\ &= (equal ; contains) \cap (overlap ; contains) \cap (overlap ; meet) \cap (contains ; equal) \\ &= \{contains\} \cap \{disjoint, meet, contains, overs, verlap\} \cap \{disjoint, meet, contains, covers, overlap\} \cap \{contains\} \\ &= \{contains\} \end{aligned}$$

Example 3b: If the edges t_{AD} and t_{DA} were labeled *covers* and *coveredBy*, respectively, the network was path-consistent, because

$$\begin{aligned} t_{AD} &= (t_{AA} ; t_{AD}) \cap (t_{AB} ; t_{BD}) \cap (t_{AC} ; t_{CD}) \cap (t_{AD} ; t_{DD}) \\ &= (equal ; covers) \cap (overlap ; contains) \cap (overlap ; meet) \cap (coveredBy ; equal) \\ &= \{covers\} \cap \{disjoint, meet, ontains, overs, overlap\} \cap \{disjoint, meet, contains, covers, overlap\} \cap \{coveredBy\} \\ &= \{\o\} \end{aligned}$$

Example 3b demonstrates that a network that is node consistent and arc consistent need not be necessarily path consistent.

5 Consistency Evaluation

The computational evaluation of the consistency of a constraint network consists of the following steps:

Step 1: Apply to all relations the node-consistency constraint:

$$t'_{ii} := t_{ii} \cap \mathit{equal} \quad (5a)$$
$$t'_{ij} := t_{ij} \text{ if } i \neq j \quad (5b)$$

This leads to a node-consistent network, T'.

Step 2: Apply to all relations of the node-consistent network the arc-consistency constraint:

$$t''_{ij} := t'_{ij} \cap \breve{t}'_{ji} \quad (6)$$

This provides an arc-consistent network, T''.

Step 3: Apply to all relations of the arc-consistent network the path-consistency constraint:

$$t'''_{ij} := \bigcap_{k=A}^{N} (t''_{ik} \,;\, t''_{kj}) \quad (7)$$

This step provides a path-consistent network, T'''.

The constraint network is *inconsistent* if any of the inferred relations of T''' is empty.

To evaluate the consistency of a constraint network it is generally necessary to iterate over Steps 2 and 3, using the inferred relations of T''' as input in Step 2, until the path-consistent network stabilizes, i.e., no new relations can be inferred.

Example 4: Given a spatial query with the following constraints among the four objects *A*, *B*, C, and *D*: contains *(A, D)* and disjoint *(C, A)* and meet *(A, B)* and disjoint *(D, C)* and overlap *(B, C)* and contains *(B, D)*.

The initial constraint network, *T*, is shown in Table 2.

	A	*B*	*C*	*D*
A	*U*	meet	*U*	contains
B	*U*	*U*	overlap	contains
C	disjoint	*U*	*U*	*U*
D	*U*	*U*	disjoint	*U*

Table 2: Tabular representation of the constraint network for the query “A meet B and A contains D and B overlap C and B contains D and C disjoint A and D disjoint C.

Unknown information is denoted by the universal relation. The node-consistency constraint (Equation 5) checks that the identity relation holds for the relation between each object and itself, transforming *T* into T'. Table 3 shows that the network is node-consistent as there is no empty relation.

	A	*B*	*C*	*D*
A	equal	meet	*U*	contains
B	*U*	equal	overlap	contains
C	disjoint	*U*	equal	*U*
D	*U*	*U*	disjoint	equal

Table 3: Constraint network after applying the node consistency.

By applying the arc-consistency to *T´* (Equation 6), the converse relations get synchronized, creating a new network, *T´´*. Table 4 depicts *T´´*, in which all relations are non-empty, which verifies that *T´´* is arc-consistent.

	A	*B*	*C*	*D*
A	equal	meet	disjoint	contains
B	meet	equal	overlap	contains
C	disjoint	overlap	equal	disjoint
D	inside	inside	disjoint	equal

Table 4: Constraint network after applying the arc-consistency constraint.

Next, the path-consistency constraint (Equation 7) is applied to all relations in *T´´*, creating a new network, *T´´´*. Table 5 depicts the relations of *T´´´* and identifies that the network is inconsistent, because several inferred relations are empty.

	A	*B*	*C*	*D*
A	equal	ø	disjoint	ø
B	ø	equal	overlap	ø
C	disjoint	overlap	equal	disjoint
D	ø	ø	disjoint	equal

Table 5: After applying the path-consistency constraint, it becomes clear that the query constraints were inconsistent.

The consistency evaluation may be shortcut by stopping the process whenever a consistency constraint generates a network with at least one empty relation. Since an empty relation will remain empty under any further manipulations, it is safe to interrupt the evaluation once an inconsistency has been found. However, in order to guarantee the reverse—that the network is consistent—it is necessary to process the constraints exhaustively.

We have implemented such a consistency tester for the binary topological relations, using the visual programming language Prograph™. It demonstrated that the formalism can be immediately translated into programming code. The pre-processing requires only a small portion of the total time necessary to process a consistent spatial query. Note that no disk accesses to data in the database are necessary. Though the complexity of the problem is known to be NP complete (Maddux, 1990), i.e., there is no algorithm that guarantees to complete it in polynomial time, the consistency tester with the proposed method is feasible even for complex spatial queries. Based on our experience with complex spatial queries (Egenhofer, 1991b), the number of objects among which the constraints must hold is usually small, and therefore the performance of the consistency tester is fast.

6 Conclusions

A formal method has been introduced to evaluate whether the constraints in a spatial query are consistent. With this method, inconsistent queries can be detected early during query processing, without ever looking at the data stored, and hence avoid considerable time delays in giving an empty answer. Actually, the method allows users to distinguish between an inconsistent query that will never produce a result and a query that is consistent but, based on the data available, does not produce a result.

The computational method uses a representation of a constraint network to model spatial objects and their spatial relations. The constraint network employs algebraic properties of a set of spatial relations and analyzes whether a particular configuration fulfills these properties. The application of the computational method has been demonstrated for a set of binary topological relations between regions (2-dimensional objects without holes), for which a comprehensive algebra had been developed. Queries over other spatial relations can be assessed accordingly. The necessary properties are: (1) the identity relation; (2) a compilation of which pairs of relations are converse; and (3) the composition table over all relations in the set.

Several interesting tasks remain for future research:

- Assist the user in identifying which relations in an inconsistent network are most likely incorrect. This would help the user to re-formulate a consistent query.
- Develop algebras for other spatial relations so that these relations can be assessed the same way we assessed topological relations. This is particularly important for cardinal directions between extended (non-point-like) objects and approximate distances.
- Integrate the algebras, i.e., derive the compositions among different kinds of spatial relations. This is necessary to assess whether it is a consistent query to ask for “all objects *A* that are north of *B* such that *B* contains *C* and *C* is south of *A*.”
- Extend the query pre-processor to account for the semantics of the spatial relations *and* the spatial objects. For example, while it is valid to ask for all islands that meet another island, it does not make sense based on the semantics of islands, because any two islands must be disjoint.

7 Acknowledgments

A number of people helped clarify the ideas presented. Particularly, Jayant Sharma’s and Kathleen Hornsby's comments on an earlier version of this paper were very helpful. Thanks also to Todd Rowell who implemented the prototype of the topological consistency checker.

8 References

Alexandroff, P., 1961. *Elementary Concepts of Topology,* Dover Publications, Inc., New York, NY.

Aref, W., and H. Samet, 1991, Optimization Strategies for Spatial Query Processing, in: G. Lohman, A. Sernadas, and R. Camps, editors, *17th International Conference on Very Large Data Bases*, Barcelona, Spain, pp. 81-90.

Barrera, R., A. Frank, and K. Al-Taha, 1991. Temporal Relations in Geographic Information Systems, *SIGMOD RECORD*, Vol. 20, No. 3, pp. 85-91.

Barrera R., M. Egenhofer, and A. Frank, 1992. Robust Evaluation of Spatial Queries, in: D. Cowen, editor, *Fifth International Symposium on Spatial Data Handling*, Charleston, SC, pp. 241-248.

Becker, L., and H. Güting, 1992. Rule-Based Optimization and Query Processing in an Extensible Geometric Database System, *ACM Transactions on Database Systems*, Vol. 17, No. 2, pp. 247-303.

Clementini, E., P. Di Felice, and P. van Oosterom, 1992. A Small Set of Formal Topological Relationships Suitable for End-User Interaction, Technical Report, University of L'Aquilla, Italy.

Cui, Z., A. Cohn, and D. Randell, 1992. Qualitative and Topological Relationships in Spatial Databases, Technical Report, Leeds University, England.

Codd, E., 1970. A Relational Model for Large Shared Data Banks, *ACM Transactions on Database Systems*, Vol. 13, No. 6, pp. 377-387.

de Hoop, S., and P. van Oosterom, 1992. Storage and Manipulation of Topology in Postgres. *EGIS '92*, Munich, Germany.

Dohrenbeck, C., and M. Egenhofer, 1991. Algebraic Optimization of Combined Overlap Operations, in: D. Mark and D. White, editors, *Autocarto 10*, Baltimore, MD, pp. 296-312.

Egenhofer, M., 1991a. Reasoning about Binary Topological Relations, in: O. Günther and H.-J. Schek, editors, *Advances in Spatial Databases—Second Symposium, SSD '91, Lecture Notes in Computer Science*, Vol. 525, Springer-Verlag, New York, NY, pp. 143-160.

Egenhofer, M., 1991b. Extending SQL for Cartographic Display. *Cartography and Geographic Information Systems*, Vol. 18, No. 4, pp. 230-245.

Egenhofer, M., and K. Al-Taha, 1992. Reasoning about Gradual Changes of Topological Relationships, in: A. Frank, I. Campari, and U. Formentini, editors, *GIS, From Space to Territory, Theories and Methods of Spatio-Temporal Reasoning*, Lecture Notes in Computer Science, Vol. 639, Springer-Verlag, New York, NY, pp. 196-219.

Egenhofer, M., and R. Franzosa, 1991. Point-Set Topological Spatial Relations. *International Journal of Geographical Information Systems*, Vol. 5, No. 2, pp. 161-174.

Egenhofer, M., and J. Herring, 1990. A Mathematical Framework for the Definition of Topological Relationships, in: K. Brassel, editor, *Fourth International Symposium on Spatial Data Handling*, Zurich, Switzerland, pp. 803-813.

Egenhofer, M., and J. Sharma, 1992. Topological Consistency, in: D. Cowen, editor, *Fifth International Symposium on Spatial Data Handling*, Charleston, SC, pp. 335-343.

Frank, A., 1991. Qualitative Spatial Reasoning about Cardinal Directions, in: D. Mark and D. White, editors, *Autocarto 10*, Baltimore, MD, pp. 148-167.

Frank, A. and D. Mark, 1991. Language Issues for GIS, in: D. Maguire, M. Goodchild, and D. Rhind, editors, *Geographical Information Systems: Principles and Applications*, Longman, London, Vol. 1, pp. 147-163.

Freeman, J. 1975. The Modeling of Spatial Relations, *Computer Graphics and Image Processing*, Vol. 4, pp. 156-171.

Freuder, E.,1978. Synthesizing Constraint Expressions, *Communications of the ACM*, Vol. 21, No. 11, pp. 958-966.

Hadzilacos, T. and N. Tryfona, 1992, A Model for Expressing Topological Integrity Constraints in Geographic Databases, in: A. Frank, I. Campari, U. Formentini, editors, *Theories and Models of Spatio-Temporal Reasoning in Geographic Space*, Lecture Notes in Computer Science, Vol. 639, Springer-Verlag, New York, NY, pp. 252-268.

Herring, J., 1991. The Mathematical Modeling of Spatial and Non-Spatial Information in Geographic Information Systems, in: D. Mark and A. Frank, editors, *Cognitive and Linguistic Aspects of Geographic Space*, Kluwer Academic Publishers, Dordrecht, pp. 313-350.

Herring, J., M. Egenhofer, and A. Frank, 1990. Using Category Theory to Model GIS Applications, in: K. Brassel, editor, *Fourth International Symposium on Spatial Data Handling*, Zurich, Switzerland, pp. 820-829.

Herskovits, A.,1986. *Language and Spatial Cognition*, Cambridge University Press, Cambridge, MA.

Hoel, E., and H. Samet, 1991. Efficient Processing of Spatial Queries in Line Segment Databases, in: O. Günther and H.-J. Schek, editors, *Advances in Spatial Databases—Second Symposium, SSD '91*, *Lecture Notes in Computer Science,* Vol. 525, Springer-Verlag, New York, NY, pp. 237-256.

Kim, W., D. Reiner, and D. Batory, 1985. *Query Processing in Database Systems*, Springer-Verlag, New York, NY.

Kriegel, H.-P., M. Schiwietz, R. Schneider, and B. Seeger, 1989. Performance Comparison of Point and Spatial Access Methods, in: A. Buchmann, O. Günther, T. Smith, and Y. Wang, editors, *Symposium on the Design and Implementation of Large Spatial Databases, Lecture Notes in Computer Science,* Vol. 409, Springer-Verlag, New York, NY, pp. 89-114.

Langran, G., 1992. *Time in Geographic Information Systems,* Taylor and Francis, London.

Lu, W., and J. Han, 1990. Decomposition of Spatial Database Queries by Deduction and Compilation, in: K. Brassel, editor, *Fourth International Symposium on Spatial Data Handling,* Zurich, Switzerland, pp. 579-588.

Lu, W., and J. Han, 1992. Deductive Spatial Query Optimization by Dynamic Connection Graph Transformation, in: D. Cowen, editor, *Fifth International Symposium on Spatial Data Handling*, Charleston, SC, pp. 324-334.

Mackworth, A., 1977. Consistency in Networks of Relations, *Artificial Intelligence*, Vol. 8, pp. 99-118.

Maddux, R., 1990. *Some Algebras and Algorithms for Reasoning about Time and Space*. Technical Report, Department of Mathematics, Iowa State University, Ames, IO.

Mark, D., and M. Egenhofer, 1992. An Evaluation of the 9-Intersection for Region-Line Relations, *GIS/LIS '92*, San Jose, CA.

Menon, S., and T. Smith, 1989. A Declarative Spatial Query Processor for Geographic Information Systems, *Photogrammetric Engineering & Remote Sensing*, Vol., 55, No. 11, pp. 1593-1600.

Montanari, U., 1974. Networks of Constraints: Fundamental Properties and Applications to Picture Processing, *Information Sciences*, Vol. 7, pp. 95-132.

Ooi, B., and R. Sacks-Davis, 1989. Query Optimization in an Extended DBMS, in: W. Litwin and H.-J. Schek, editors, *Foundations of Data Organization and Algorithms*, Paris, France, *Lecture Notes in Computer Science*, Vol. 367, Springer-Verlag, New York, NY, pp. 48-63.

Peuquet, D., 1986. The Use of Spatial Relationships to Aid Spatial Database Retrieval, in: D. Marble, editor, *Second International Symposium on Spatial Data Handling*, Seattle, WA, pp. 459-471.

Pigot, S., 1991. Topological Models for 3D Spatial Information Systems, in: D. Mark and D. White, editors, *Autocarto 10*, Baltimore, MD, pp. 368-392.

Pullar, D., and M. Egenhofer, 1988. Towards Formal Definitions of Topological Relations Among Spatial Objects, *Third International Symposium on Spatial Data Handling*, Sydney, Australia, pp. 225-243.

Samet, H., 1989. *The Design and Analysis of Spatial Data Structures*, Addison-Wesley Publishing Company, Reading, MA.

Smith, T., and A. Frank, 1990. Very Large Spatial Databases: Report from the Specialist Meeting. *Journal of Visual Languages and Computing*, Vol. 1, No. 3, pp. 291-309.

Svensson, P. and H. Zhexue, 1991. A Query Language for Spatial Data Analysis, in: O. Günther and H.-J. Schek, editors, *Advances in Spatial Databases—Second Symposium, SSD '91*, *Lecture Notes in Computer Science,* Vol. 525, Springer-Verlag, New York, NY, pp. 119-140.

Tarski, A., 1941. On the Calculus of Relations, *The Journal of Symbolic Logic*, Vol. 6: No. 3, pp. 73-89.

Worboys, M., and S. Deen, 1991. Semantic Heterogeneity in Geographic Databases, *SIGMOD RECORD*, Vol. 20, No. 4, pp. 30-34.

Yost, M., and B. Skelton, 1990. Programming Language Technology for Raster GIS Modeling, *GIS/LIS '90*, Anaheim, CA, pp. 319-327.