

# Measuring Consistency with respect to Topological Dependency Constraints\*

M. Andrea Rodríguez  
Universidad de Concepción  
4070409 Concepción, Chile  
andrea@udec.cl

Nieves Brisaboa  
Universidade da Coruña  
15071 A Coruña, Spain  
brisaboa@udc.es

Jazna Meza  
Universidad de Concepción  
4070409 Concepción, Chile  
jaznameza@udec.cl

Miguel R. Luaces  
Universidade da Coruña  
15071 A Coruña, Spain  
luaces@udc.es

## ABSTRACT

In contrast to the enormous development of database management systems to support spatial databases, very little work has been done in evaluating the quality of spatial data in terms of how much they satisfy a set of topo-semantic integrity constraints, in particular, a set of topological dependency constraints. In the same way, mechanisms for enforcing the satisfaction of those constraints are not necessarily available or even feasible. In this paper we propose measures to evaluate the degree of violation of a topological dependency constraint by geometries stored in a spatial database instance. We also propose how these measures can be aggregated to globally evaluate the data quality of a database instance such that they enable to compare database instances in terms of their constraint satisfaction. We provide an experimental evaluation of those measures using synthetic and real data. We validate our measures by i) analyzing their correlation with the semantic distance of topological relations and ii) checking that the more we randomly modify geometries to make database instances inconsistent, the more our global data quality measure decreases, showing its sensibility to the introduced constraint violations.

## Categories and Subject Descriptors

H.2.8 [Database Management]: Database Applications—*Spatial databases and GIS*

## General Terms

Theory

---

\*Partially funded by Fondecyt 1080138, CONICYT-CHILE, and MICINN grant TIN2009-14560-C03-02

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ACM GIS '10, November 2-5, 2010, San Jose, CA, USA  
Copyright 2010 ACM ISBN 978-1-4503-0428-3/10/11 ...\$10.00.

## Keywords

Consistency, consistency measures, database consistency

## 1. INTRODUCTION AND MOTIVATION

Inconsistency is an undesirable state of database systems. It rises when data in the database violate a set of integrity constraints (ICs). Although many would argue that systems should be completely free of inconsistency, inconsistency is a reality in many real-world cases and, as such, it should be formalized and used, rather than rejected [10].

Different reasons may lead to an inconsistent state of a database instance with respect to a set of integrity constraints. There are cases where enforcing integrity constraints as the database is updated is impossible or impractical. In such cases, spatial inconsistency may arise easily since spatial data is inherently vague. The intrinsic vagueness of spatial features produces different observations of the same spatial phenomenon and, therefore, generates conflicting representations [2]. In addition, spatial databases may need to treat different levels of detail in the spatial representation. These different levels of detail can be handled by using more than one geometric representation of the same object, which leads us to potential inconsistency problems. Moreover, in the case of integrated systems that consolidate data from different sources, the global database instance may become inconsistent with respect to global ICs, even if the sources are locally consistent.

Inconsistency in spatial database occurs most likely when integrity constraints are not embedded in the database management systems, and therefore, they are left to the expertise of the database designer. This is the case of topo-semantic integrity constraints, which are equivalent to the semantic constraints in the relational context. A topo-semantic integrity constraint imposes topological considerations onto the semantics of geographic objects. The following sentences are typical examples of this type of ICs: two parcels must not *overlap*, a house must be *inside* a parcel, two roads must not be *equal*, a river and a road cannot *cross* except in the geographic object bridge, and so on.

Assuming that inconsistency may occur, data quality of a database instance becomes relevant, which in turn calls for determining the satisfaction degree of the database instance with respect to its set of ICs. This enables not only to decide whether or not a database is useful for an application, but

also to compare database instances from where one wishes to extract information. This comparison could lead us to choose the most reliable source of information.

This paper presents first results in quantifying the inconsistency of a database instance with respect to a particular type of topo-semantic integrity constraints, namely, topological dependency constraints [4], which establish a dependency of entities with respect to topological relations. For example, two land parcels must be internally disconnected. The work proposes quantitative measures about the degree of satisfaction of two geometries with respect to an expected topological relation as expressed by a topological dependency constraint. We define measures for each expected topological relation and apply them to synthetic and real data. We compare the results to two other different measures: (1) the semantic distance in a conceptual neighborhood graph of topological relations [6][8], and (2) the distance between boundary points of geometries in an inconsistent versus consistent database.

The organization of the paper is as follows. After presenting some related work and preliminaries in Section 2, in Section 3 we provide the considerations and strategies we followed to evaluate the satisfaction of each type of topological relation. In Section 4, we present the method to evaluate the data quality of a database instance. Finally, Section 5 is devoted to a preliminary empirical evaluation of our proposal and Section 6 provides our conclusions.

## 2. PRELIMINARIES

This section introduces related work concerning spatial integrity constraints and treatment of consistency in spatial databases. It also presents an abstract model of spatial databases and the set of integrity constraints treated in this work.

### 2.1 Related Work

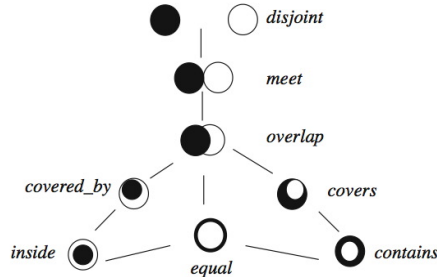
A classification of integrity constraints in relational database systems distinguishes three types: domain constraints, key or relational constraints, and general semantic constraints [9]. Spatial integrity constraints relate to two of these types of constraints: (a) domain constraints (also known as *topological constraints*) that specify admissible values of the geometric representation, and (b) semantic constraints (also known as *topo-semantic integrity constraints*) that associate the semantics of the modeled entities with spatial properties, in particular, with topological relations between spatial objects [5]. A more general categorization of semantic integrity constraints distinguishes thematic, temporal, spatial, complex semantics, and change constraints [15]. Then, spatial constraints can be subclassified by the spatial aspects they consider, namely, topology, orientation, shape, and distance.

In terms of the specification of spatial integrity constraints, related work addresses the specification of topological constraints [3] and spatial semantic constraints [14, 12]. Later in [24], there is a proposal to specify constraints with spatial semantics, which introduces explicitly four types of cardinalities: forbidden, at least  $n$  times, at most  $n$  times, and exactly  $n$  times. Within this proposal, one could express, for example, that a sluice joins a waterpipe exactly two times. More recently, types of semantic integrity constraints and an analysis of the database consistency problem with these constraints were presented in [4]. These types of integrity constraints combine classical functional and referential integrity

constraints with topological relations and check constraints (e.g., the numeric area of a geometric).

In the context of spatial databases, some studies also deal with detection and cleaning of inconsistent databases. A methodology for the consistency improvement of geographic databases is presented in [24]. This methodology proposes alternatives of improvements under different types of integrity constraints. However, it does not analyze data complexity nor the interaction of different inconsistencies and, therefore, interaction under modification of the data.

Similarity measures between topological relations are in theory applicable to define the degree of inconsistency in terms of the difference between an existing topological relation and the expected topological relation as expressed by topological dependency constraints. We distinguish qualitative from quantitative approaches to compare topological relations. A qualitative representation of topological relations uses a symbolic representation of spatial relations, such as the topological relations defined by Egenhofer and Franzosa [7] or by Randell *et al.* [21]. Under this representation, a similarity measure compares topological relations by the semantic distance between relations defined in a conceptual neighborhood graph [19, 6] (Figure 1). The disadvantage of comparing topological relations from a qualitative perspective is that it does not make distinction between particular geometries. For example, it does not distinguish between two pairs of geometries, both disjoint, but where in one case two close geometries are disjoint and in the other case two geometries are far apart. Even more, all edges in the conceptual graph will usually have the same weight in the determination of the semantic distance.



**Figure 1: Conceptual neighborhood of topological relations**

A quantitative representation of topological relations is given by the distance and angle between centroid of objects [1]. Using this representation, similarity between topological relations is defined as the inverse of the difference between representations. A more recent work uses the Minimum Bounding Rectangle (MBR) of objects for their quantitative representation [11]. Using this representation, a topological relation is characterized by quantitative measures that combine overlapping areas with distance between objects. The problem of using the previous measures for evaluating the degree of inconsistency is that although database instances handle geometries of objects, topological dependency constraints are expressed by qualitative topological relations, and therefore, only a symbolic representation of the expected topological relations exists.

A repair semantics was defined and used as an instrumental concept to define consistent query answers to range

queries over inconsistent spatial databases with respect to constraints expressed by denial logical formulas [23]. In this work, a repair is any database instance that satisfies the set of constraints and that results from admissible transformations that shrink, or even, make empty geometries. The work defines a distance measure to determine repairs that minimally differ with the original database instance (i.e., minimal repairs). In principle, one could use the concept of repair semantics and the inverse of the distance between an inconsistent database and its minimal repairs to define a measure of the data quality of an inconsistent database instance. This work, however, does not use the repair semantics as defined in [23] for the following reasons: (1) the repair semantics in [23] solves conflicts with respect to denials, that is, it solves conflicts when geometries must not hold a particular topological relation. In this work, in contrast, we want to make geometries to hold a specific topological relation. (2) There is a potentially exponential number of repairs and the decision problem of deciding if a database instance is a minimum repair is intractable. (3) We do not limit the type of transformations to shrinking geometries, and we quantify data quality at both an individual conflict level and a global database instance level.

To the best of our knowledge, related work addressing inconsistency measures in databases exists only in the relational context. The work in [13] presents an approach to measuring inconsistency through the minimal set of inconsistent objects of a knowledgebase. This work addresses the inconsistency treatment of a set of logic formulas. It also describes the use of Shapley inconsistency values to define a weight of each formula within a general measure applied to the knowledgebase. A strategy to measuring consistency with respect to referential integrity constraints in distributed databases is in [18]. It defines both local and global measures to consistency and completeness of data, from consistency of tables to consistency of the database.

## 2.2 Abstract Model and Integrity Constraints

We specify integrity constraints on an extended relational database model. A spatio-relational database schema is of the form  $\Sigma = (\mathcal{U}, \mathcal{A}, \mathcal{S}, \mathcal{R}, \mathcal{T}, \mathcal{O})$ , where: (a)  $\mathcal{U}$  is the possibly infinite database domain of atomic thematic values that includes  $\mathbb{R}$ . (b)  $\mathcal{A} = \{A_1, \dots, A_n\}$  where each  $A_i$  is a thematic attribute which takes values in  $\mathcal{U}$ . (c)  $\mathcal{S} = \{S_1, \dots, S_n\}$  where each  $S_i$  takes admissible values in  $\mathcal{P}(\mathbb{R}^2)$ , the power set of  $\mathbb{R}^2$ . (d)  $\mathcal{R}$  is a finite set of spatio-relational predicates each of them with a finite and ordered set of attributes belonging to  $\mathcal{A}$  or  $\mathcal{S}$ . (e)  $\mathcal{T}$  is a fixed set of binary topological predicates. (f)  $\mathcal{O}$  is a fixed set of geometric operators that take spatial and thematic arguments and return a geometry or a value in  $\mathcal{U}$ .

A database instance  $D$  of a spatio-relational schema  $\Sigma$  is a finite collection of ground atoms (or *tuples*) of the form  $R(c_1, \dots, c_i, \dots, c_n)$ , where (a)  $R(A_1, \dots, A_i, \dots, A_n) \in \mathcal{R}$ , (b) if  $A_i \in \mathcal{A}$ , then  $c_i \in \mathcal{U}$  (c) if  $A_i \in \mathcal{S}$ , then  $c_i \in \mathcal{Ad} \subseteq \mathcal{P}(\mathbb{R}^2)$  where  $\mathcal{Ad}$  is the class of geometries as specified by the standard of the OGC [17]. For this work, we will concentrate on geometries that are lines or regions on a plane.

The elements of  $\mathcal{T}$  are binary topological relations with a fixed semantics. There are eight pairwise disjoint topological relations that were formalized between regions [20, 7], thirty three between lines, and twenty between a region

and a line [16]. Current database management systems represent geometries in terms of spatial data types (i.e., polygons, polylines, points) and distinguish a subset of topological relations in terms of the type of intersection between geometries. Table 1 provides the definitions of topological relations, which were extracted from the OpenGIS Simple Feature Specification [17] and used in the subsequent specification of topological dependency constraints, where  $I(x)$  indicates the interior,  $E(x)$  the exterior and  $dim$  the dimension of a geometry  $x$ .

Relation	Definition
Disjoint( $x, y$ )	True if $x \cap y = \emptyset$ .
Touches( $x, y$ )	True if $I(x) \cap I(y) = \emptyset$ and $x \cap y \neq \emptyset$ .
Equal( $x, y$ )	True if $x \subseteq y$ and $y \subseteq x$ .
Within( $x, y$ )	True if $x \cap y = x$ and $I(x) \cap E(y) = \emptyset$ .
Contains( $x, y$ )	True if Within( $y, x$ ).
Overlaps( $x, y$ )	True if $dim(I(x)) = dim(I(y)) = dim(I(x) \cap I(y))$ and $x \cap y \neq x$ , $x \cap y \neq y$ .
Crosses( $x, y$ )	True if $I(x) \cap I(y) \neq \emptyset$ and $x \cap y \neq x$ and $x \cap y \neq y$ , where $x$ and $y$ are both lines or one a line and the other one a region.

**Table 1: Definition of topological relations ( $x$  and  $y$  are lines or regions)**

A schema  $\Sigma$  determines a first-order (FO) language  $\mathcal{L}(\Sigma)$  of predicate logic. It can be used to syntactically characterize and express topo-semantic constraints (SICs). In this paper we concentrate on *Topological Dependency* (TD) constraints [4], which are sentences of the form:

$$\forall \bar{u} \bar{y}_1 \bar{y}_2 \bar{g}_1 \bar{g}_2 (P(\bar{u}, \bar{y}_1, g_1) \wedge R(\bar{u}, \bar{y}_2, g_2) \bigwedge_{i=0}^{T(g_1, g_2)} x_i \neq z_i \rightarrow T(g_1, g_2)), \quad (1)$$

where  $\bar{y}_1 \cap \bar{y}_2 = \emptyset$  and for every  $i$ , variable  $x_i \in \bar{u} \cup \bar{y}_1$  and variable  $z_i \in \bar{u} \cup \bar{y}_2$ .

For all purposes, we will consider that a set  $\Psi$  of TDs of a database schema is consistent, i.e., there exists a non-empty database instance that can satisfy  $\Psi$ . A database instance  $D$  violates a topological dependency constraints of the form (1) when there are data values  $\bar{u} \bar{y}_1 \bar{y}_2 \bar{g}_1 \bar{g}_2$  for the variables in the constraint for which  $((P(\bar{u}, \bar{y}_1, g_1) \wedge R(\bar{u}, \bar{y}_2, g_2) \wedge x \neq z) \rightarrow T(g_1, g_2))$  becomes false in the database under those values. When this is the case, we consider that  $T(g_1, g_2)$  is false.

## 3. DEGREE OF CONSTRAINT SATISFACTION

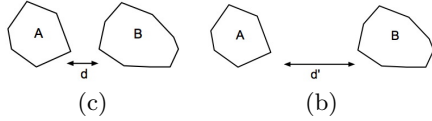
Let  $td$  be a TD with topological relation  $T \in \mathcal{T}$ , and  $g_1$  and  $g_2$  be geometries stored in tuples of a database instance. The satisfaction degree of  $g_1$  and  $g_2$  with respect to  $td$  is defined by the complement of the violation degree of the topological relation between  $g_1$  and  $g_2$  with respect to  $T$ . Consequently, we define measures that compare the topological relation between geometries and an expected topological relation. We call these measures *constraint-violation measures*. Before giving definitions of these measures, we will introduce their basic components.

### 3.1 Components of constraint-violation measures

To define the degree of violation of topological dependency constraints, we systematically consider two aspects: *A)* the degree of the violation itself and *B)* the weight or relevance of objects in the database instance.

**A) Violation degree.** We consider that a constraint violation has a degree that depends on how different is the

actual relationship between objects from the expected relation. For example, if two objects must *Touches* and they are *Disjoint* the violation is larger as they are farther from each other (Figure 2). If they must not *Overlaps*, the violation is larger as larger the overlapping region is. If an object must be *Within* another object, but they are *Disjoint*, the violation gets larger as they are farther apart.



**Figure 2: Distance consideration in the constraint-violation measure with respect to relation Touches: (a) smaller violation (b) larger violation**

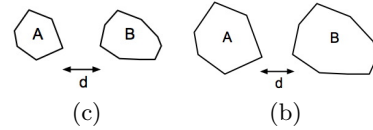
We argue that this difference will be proportional to the effort or cost of making consistent a pair of geometries in conflict. Consequently, for each expected topological relation  $T$  we define modifications on geometries that hold  $T' \neq T$  so that they will hold  $T$  after transformation. This resembles the concept of semantic repairs for spatial databases defined in [23], however, we do not limit ourselves to transformations that shrink geometries and we are not expected to repair the database instance. In Table 2 we summarize the basic transformations that change the topological relation between geometries. For space limitations, we describe only the transformations for topological relations between regions.

<b>From</b>	<b>To: Disjoint</b>
Touches	Insert a minimum separation
Overlaps	Eliminate overlapping area and insert a minimum separation
Within	
Equal	
<b>From</b>	<b>To: Touches</b>
Disjoint	Eliminate separation
Overlaps	Eliminate overlapping area
Within	
Equal	
<b>From</b>	<b>To: Overlaps</b>
Disjoint	Eliminate separation and create a minimum overlapping area
Touches	Create a minimum overlapping area
Within	Eliminate internal distance and create a minimum overlapping area
Equal	Create a minimum overlapping area
<b>From</b>	<b>To: Within</b>
Disjoint	Eliminate separation, make sizes of geometries compatible, and move one into the other one
Touches	Make sizes of geometries compatible, and move one into the other one
Overlaps	Make sizes of geometries compatible, and move one into the other one
Contains	Make sizes of geometries compatible
<b>From</b>	<b>To: Equal</b>
Disjoint	Eliminate separation, make geometries of the same size and move one over the other one
Touches	Make geometries of the same size and move one over the other one
Overlaps	Make geometries of the same size and move one over the other one

**Table 2: Basic transformations that change topological relations**

Then, the constraint-violation measures will quantify the cost or effort to carry out the transformation for each particular case.




**B) Relevance of objects.** Since we do not have in advance any information about the relevance of objects, we study the use of their size to define a relative weight of objects in conflict. Thus, the larger the objects that participate in a constraint violation, the larger the degree of violation; that is, if two objects must *Touches* and they *Overlaps* 25% of their sizes, their conflict degree is smaller than the conflict degree of other two larger objects that also overlap 25% of their sizes (Figure 3). Notice, however, that there may be other options of relevance that could be application dependent. To have a measure to compare all objects, we decide to use the largest geometry in the database to establish a relative weight of a geometry in conflict with respect to the other geometries in the database.



**Figure 3: Size consideration in the conflict measure with respect to relation Touches: (a) smaller conflict (b) larger conflict**

### 3.2 Basic spatial concepts and notation

A basic characterization of a geometry is its size. This helps to establish the relevance of a geometry with respect to any other geometry in the database, and in particular, to the largest geometry in the database. In order to use a single magnitude to characterize the size of geometries of different dimensions, we use the perimeter for areas and the length for lines as basic concepts for describing the size of geometries. Table 3 shows the first concepts we will use in the definition of conflict measures.

Symbol	Description	Example
$c_i$	Perimeter of a region $g_i$	 $c_i = 4p$
$l_i$	Length of a line $g_i$	 $l_i = p$
$l_c$	Length of the shortest segment of crossing lines	 $l_c = p/4$

**Table 3: Basic measures for the size of geometries**

There is a basic conceptual difference between topological relations. On the one hand, for all topological relations other than *Disjoint*, geometries intersect. The intersection between geometries results in a geometry of the same or lower dimension than the highest dimension of geometries that participate in the conflict. In particular, the intersection of two polygons can be polygons, lines, or points. The intersection of two lines results in lines or points. On the other hand, for geometries that are *Disjoint* do not intersect, but there exists a distance between them.

Our approach will distinguish degrees of conflicts by measuring the intersection of geometries or the distance between geometries. Due to the fact that there are cases where both the intersection and distance must be combined in a single measure, and that we want to avoid to deal with different types of magnitudes, we again characterize polygons and

lines by their perimeters and lengths, respectively, which is in agreement with the linear magnitude of the distance between geometries. Table 4 shows the concepts that distinguish overlapping areas, crossing segments, and distances between geometries.

	Description	Example	
$c_s$	Perimeter of $g_1 \cap g_2$		$c_s = 1.5p$
$c_{g_1}$	Perimeter of $g_1 \setminus g_2$		$c_{g_1} = 4p$
$c_{mg}$	$Minimum(c_{g_1}, c_{g_2})$		$c_{mg} = 1.5p$
$d_{me}$	Minimum external distance between disjoint geometries. It applies to regions and lines		$d_{me} = 1/2p$
$d_{mi}$	Minimum internal distance between a geometry within another region		$d_{mi} = 1/4p$
$l_s$	Length of $g_1 \cap g_2$		$l_s = 1/2p$
$l_{g_1}$	Length of $g_1 \setminus g_2$		$l_{g_1} = 1/2p$
$l_{mg}$	$Minimum(l_{g_1}, l_{g_2})$		$l_{mg} = 0$

**Table 4: Basic measures for the degree of overlap and distance between geometries**

In some cases, a constraint may establish a type of relation between objects that implies that they must be separated by a distance or must overlap, but the separation or the overlapping area is unknown. In those cases, we assume a minimum value of the area (perimeter) or distance that is unknown, assuming that the minimum distance or area is a value that depends of the size of the objects in conflict or the size of all objects in the database. Table 5 gives the different parameters used for the definition of constraint-violation measures.

Finally, table 6 introduces some concepts that simplify the notation in the definition of measures.

### 3.3 Constraint-violation Measures

Constraint-violation measures are defined by the multiplication of (1) the conflict degree that compares the actual versus the expected topological relations and (2) the relative size (weight) of objects that participate in the conflict. Both components of the measure are normalized so that their values range in  $[0..1]$ .

The normalization of the degree of violation is done by dividing the cost of changing a topological relation between two geometries by the size of the geometries plus the cost of transformations. Consequently, the denominator is always the numerator plus the total area of both geometries. Other alternatives for normalization could be explored in order to make the value of the measure larger. We use this type of normalization so that we could guarantee that conflict

Symbol	Description	Example	
$C_{mo}$	Perimeter representing the minimum overlapping area— equivalent to 10% of the smallest perimeter of regions under consideration		$C_{mo} = 0.2p$
$L_{mc}$	Crossing-segment minimum length—equivalent to 10% of the shortest of the two lines under consideration		$L_{mc} = 0.1p$
$L_{mo}$	Length of the minimum overlapping line— equivalent to 10% of the shortest of the two lines under consideration		$L_{mo} = 0.1p$
$D_m$	Minimum distance between disjoint geometries		

**Table 5: Basic parameters for the minimum length of lines, perimeter of areas, and distance between geometries.**

Symbol	Description	Example	
$c_m$	$Minimum(c_1, c_2)$		$c_m = 2p$
$c_d$	$c_1 - c_2$		$c_d = 2p$
$c_x$	$Maximum(c_d, 0)$		$c_x = 0$
$l_m$	$Minimum(l_1, l_2)$		$l_m = p/2$
$l_d$	$l_1 - l_2$		$l_d = p/2$
$l_x$	$Maximum(l_d, 0)$		$l_x = 0$
$P$	Maximum perimeters of regions in the database		
$D$	Maximum length of lines in the database		
$R$	$R = \frac{c_1 + c_2}{2P}$		
$L$	$L = \frac{l_1 + l_2}{2D}$		

**Table 6: Predefined parameters and functions.**

measures will not be higher than 1 and that they can be used in the same for all topological relations under evaluation. The normalization of the relative size of objects divides the total size of both objects by the double of the size of the largest object in the database. This ensures that this factor ranges between  $(0..1]$ , even when the objects in conflict are of the largest in the database.

As a result, the measure of any constraint violation will always be a very small value, but it gives us a measure to compare and rank different conflicts. In Table 7 we show the different measures we propose to evaluate the degree of vi-

olation for each expected relations and possible topological relation between geometries. In this work we concentrate on defining conflict measures with respect to topological relations between geometries of the same dimension, that is, between lines or between regions. We have left for future work relations between lines and regions.

	Region $\times$ Region	Line $\times$ Line
<b>Disjoint</b> (expected topological relation)		
Touches	$\frac{D_m}{c_1+c_2+D_m} \times R$	$\frac{D_m}{l_1+l_2+D_m} \times L$
Overlaps	$\frac{c_s+D_m}{c_1+c_2+c_s+D_m} \times R$	$\frac{l_s+D_m}{l_1+l_2+l_s+D_m} \times L$
Crosses	$\frac{c}{2c_1+c_2+D_m} \times R$	$\frac{l_c+D_m}{2l_1+l_2+l_c+D_m} \times L$
Within	$\frac{c_1+d_{mi}+D_m}{2c_1+c_2+d_{mi}+D_m} \times R$	$\frac{l_1+d_{mi}+D_m}{2l_1+l_2+d_{mi}+D_m} \times L$
Equal	$\frac{c_1+D_m}{3c_1+D_m} \times R$	$\frac{l_1+D_m}{3l_1+D_m} \times L$
<b>Touch</b>		
Disjoint	$\frac{d_{me}}{c_1+c_2+d_{me}} \times R$	$\frac{d_{me}}{l_1+l_2+d_{me}} \times L$
Overlaps	$\frac{c_s}{c_1+c_2+c_s} \times R$	$\frac{l_s}{l_1+l_2+l_s} \times L$
Crosses	$\frac{c}{2c_1+c_2+D_m} \times R$	$\frac{l_c}{l_1+l_2+l_c} \times L$
Within	$\frac{c_1+d_{mi}}{2c_1+c_2+d_{mi}} \times R$	$\frac{l_1+d_{mi}}{2l_1+l_2+d_{mi}} \times L$
Equal	$\frac{c_1}{3c_1} \times R$	$\frac{l_1}{3l_1} \times L$
<b>Overlap</b>		
Disjoint	$\frac{d_{me}+C_{mo}}{c_1+c_2+d_{me}+C_{mo}} \times R$	$\frac{d_{me}+L_{mo}}{l_1+l_2+d_{me}+L_{mo}} \times L$
Touches	$\frac{C_{mo}}{c_1+c_2+C_{mo}} \times R$	$\frac{L_{mo}}{l_1+l_2+L_{mo}} \times L$
Crosses	$\frac{c}{2c_1+c_2+D_m} \times R$	$\frac{l_c}{l_1+l_2+l_c} \times L$
Within	$\frac{d_{mi}+C_{mo}}{c_1+c_2+d_{mi}+C_{mo}} \times R$	$\frac{d_{mi}+L_{mo}}{l_1+l_2+d_{mi}+L_{mo}} \times L$
Equal	$\frac{C_{mo}}{2c_1+2C_{mo}} \times R$	$\frac{L_{mo}}{2l_1+2L_{mo}} \times L$
<b>Equal</b>		
Disjoint	$\frac{d_{me}+c_d+c_m}{c_1+c_2+d_{me}+c_d+c_m} \times R$	$\frac{d_{me}+l_d+l_m}{l_1+l_2+d_{me}+l_d+l_m} \times L$
Touches	$\frac{c_d+c_m}{c_1+c_2+c_d+c_m} \times R$	$\frac{l_d+l_m}{l_1+l_2+l_d+l_m} \times L$
Overlaps	$\frac{c_d+c_m+g}{c_1+c_2+c_d+c_m+g} \times R$	$\frac{l_d+l_m+g}{l_1+l_2+l_d+l_m+g} \times L$
Crosses	$\frac{c}{2c_1+c_2+D_m} \times R$	$\frac{l_c}{c_1+c_2+l_c} \times R$
<b>Within</b>		
Disjoint	$\frac{d_{me}+c_x+c_1}{2c_1+c_2+d_{me}+c_x} \times R$	$\frac{d_{me}+l_x+l_1}{2l_1+l_2+d_{me}+l_x} \times L$
Touches	$\frac{c_x+c_1}{2c_1+c_2+c_x} \times R$	$\frac{l_x+l_1}{2l_1+l_2+l_x} \times L$
Overlaps	$\frac{c_x+c_1+g}{c_1+c_2+c_x+c_1+g} \times R$	$\frac{l_x+l_1+g}{l_1+l_2+l_x+l_1+g} \times L$
Contains	$\frac{c_x}{c_1+c_2+c_x} \times R$	$\frac{l_x}{l_1+l_2+l_x} \times L$
Crosses	$\frac{c}{2c_1+c_2+D_m} \times R$	$\frac{l_c}{2l_1+l_2+l_c} \times L$
<b>Cross</b>		
Disjoint	$\frac{d_{me}+L_{mc}}{l_1+l_2+d_{me}+L_{mc}} \times L$	$\frac{L_{mc}}{l_1+l_2+L_{mc}} \times L$
Touches	$\frac{L_{mc}}{l_1+l_2+L_{mc}} \times L$	$\frac{L_{mc}+l_s}{l_1+l_2+L_{mc}+l_s} \times L$
Overlaps	$\frac{L_{mc}+l_s}{l_1+l_2+L_{mc}+l_s} \times L$	$\frac{L_{mc}+l_s}{l_1+l_2+L_{mc}+l_s} \times L$
Within	$\frac{L_{mc}}{2l_1+L_{mc}} \times L$	$\frac{L_{mc}}{2l_1+L_{mc}} \times L$
Equal	$\frac{L_{mc}}{2l_1+L_{mc}} \times L$	$\frac{L_{mc}}{2l_1+L_{mc}} \times L$

Table 7: Constraint-violation measures for each topological relation

## 4. GLOBAL DATA QUALITY

The first question that arises when defining a global evaluation of the data quality of a spatial database instance is *How many tuples are necessary to check?* In classical relational integrity constraints this question may not make any sense, but in the geographic domain it is very relevant because, in theory, between any two objects, there always exists a spatial relation. For example, if we want to check the constraint “land-parcels must not overlap”, *Does it make sense to check the  $n^2$  relationships among the  $n$  land-parcels in the database instance?* Probably not. It may only be necessary to check for each land-parcel its topological relation

with its immediate neighbors. Therefore, to check the quality of the database we need the concept of *checked topological relationships* (CTR). That is, we assume that for each topological dependency constraint there is an algorithm to decide the subset of topological relationships that is necessary to check. The different strategies that can be used to define this algorithm are out of the scope of this paper. We will assume that, in checking for constraint violations, it is clear that a strategy to select the pairs of objects to be analyzed can be implemented. This strategy will produce a number of comparisons, some of them will show that the constraint is satisfied and others will reveal a constraint violation. In any case, the number of comparisons is the *CTR* that we will use to create a global measure of the data quality of a database instance.

A second question is *What kind of measure characterizes the global quality of a database instance?* We consider that this question has two different aspects. On the one hand, it is interesting to know how many geometries violate a topological dependency constraint over the total number of CTR, that is, we are interested in knowing the spread of the inconsistencies. But, on the other hand, we also need to know how bad the inconsistencies are, that is, the global violation degree of the different constrains.

We call the first measure *violations spread* (*VS*), which is computed as the proportion of violations over the total number of *checked topological relationships*. That is:

$$VS = \frac{\#violations}{CTR},$$

where  $CTR = \#non\ violations + \#violations$ .

To capture the global quality of the database we consider not only the number of violations but also their importance. We call this second measure *Global Fulfillment* (*GF*). To evaluate each violation, we use the constraint-violation measures of the previous section. Obviously, each checked topological relation that does not violate any constraint has a value of 0. If we checked *CTR* topological relationships, the quality of the database will be

$$GF = \frac{\sum_1^{CTR} 1 - measured\ violation\ value}{CTR}.$$

This measure lies in the range  $[0 \dots 1]$ , where 0 indicates that the inconsistency is maximum and 1 means that the database instance is totally consistent.

## 5. EXPERIMENTAL EVALUATION

### 5.1 Data sets

To validate empirically the proposed measures, we created synthetic database instances and we used a real data set with political administrative boundaries.

For the synthetic database instances, we created a single table  $R$  with a geometric attribute whose value is taken from a random distribution of geometries over a grid of  $n \times n$  cells, where each cell is of dimension  $100 \times 100$ . Each cell contains a rectangular geometry, but the size and position of the geometry in the cells was given at random. We started by creating a consistent database with respect to a topological dependency constraints of the form<sup>1</sup>:

$$\forall (R(x_1, y, g_1) \wedge R(x_2, y, g_2) \wedge x_1 \neq x_2 \rightarrow T(g_1, g_2)), \quad (2)$$

where  $T \in \mathcal{T}$ . We consider tables with geometries of type lines and of type regions. Notice that this constraint includes

<sup>1</sup> $\forall$  means the universal quantification of all variables in the formula

a variable  $y$  that is the same in both predicates  $R$ . This limits the evaluation of topological relations to a subset of tuples. In these experiments we consider proximity as a strategy to define the potential geometries that may be in conflict (CTR), which means that we check each geometry with the corresponding geometries in neighboring cells.

Once consistent instances were created, we applied transformation to a percentage of geometries to produce inconsistencies. For example, for a database instance whose topological dependency constraint is defined with respect to a Disjoint relation, we created a table with all geometries being disjoint. Then we selected at random a percentage of geometries and one of their neighboring geometries, and we applied transformations to the neighboring geometries such that geometries now touch, overlap, and so on. When possible, we translated geometries without changing their actual size. This was possible when the expected topological relation was in  $\{\text{Disjoint, Touches, Overlaps, Crosses}\}$ . For the other expected topological relations (Equal or Within), it was necessary to change the size of geometries to make them of compatible size. At the end, we have a set of inconsistent database instances with respect to a topological dependency constraint. Recall that Within is the converse of Contains so that we just need one set of inconsistent database instances to evaluate both relations. We use different percentages of changes (from 5% to 30%) and sizes of the database (5000 and 10000 tuples). Based on the dimension of the space, we estimate the minimum distance that separates any two geometries ( $D_m$ ) equal to 10.

For the real database instance, we have used the 2009 TIGER/Line Shapefiles from the U.S. Census Bureau [25], because it is known that there are no inconsistencies in the data. We created a spatial database using the shapefiles for the states (tL2009\_us\_state) and the counties of the state of New York (tL2009\_36\_county). Then, we applied a simplification algorithm to geographic objects in the layer of counties of New York. The algorithm was applied to three different percentages of the total objects in the layer (5%, 25% and 50%), and three different tolerance values were used in the simplification (0.0001, 0.001 and 0.01 degrees). At the end, we created nine different simplified versions of the original consistent data. For these datasets, we checked two topological dependency constraints. The first one is of the form (5.1) with  $T = \text{Touches}$  and where  $R$  is replaced by *county*. We define the tuples to check (CTR) by tuples whose geometries intersect. The second topological dependency constraint is of the form:

$$\forall(\text{county}(x_1, x_2, g_1) \wedge \text{state}(x_2, g_2) \rightarrow \text{Within}(g_1, g_2)). \quad (3)$$

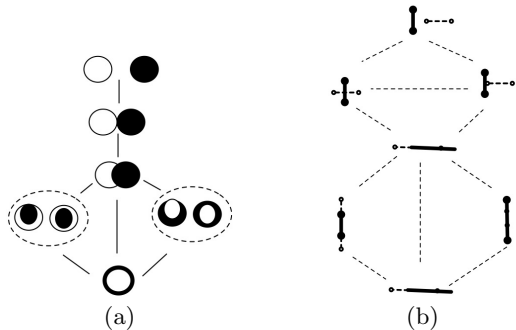
In this case, the checked topological relationships were determined using a foreign key in each county referring to the state it belongs.

## 5.2 Evaluation approach

We evaluate our proposal at two level: at the individual constraint-violation level and at the global fulfillment level (GF). When using synthetic data, we compare the proposed measures to two other approaches that quantify the difference between the expected and the actual topological relation between geometries in a database instance.

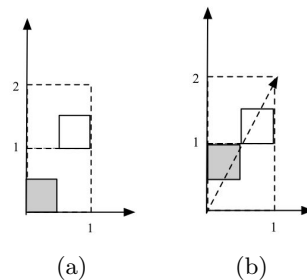
- **Semantic-distance approach.** This approach quantifies the conflict of a couple of geometries as the normalized semantic distance in a conceptual graph of topological relations [19]. We apply this semantic distance in the

same way that our individual constraint-conflict measures. Then, we define  $GF_{sd} = \frac{\sum_{i=1}^{CTR} (1-SD_i)}{CTR}$ , where  $SD_i$  is the normalized semantic distance between two topological relations. Figure 4 shows the two neighboring graphs for regions and lines, which were extracted by graphs found in the literature [6, 22] and making the aggregation needed to fit the set of topological relations used in this work.



**Figure 4: Neighboring graph for topological relations in  $\mathcal{T}$  between: (a) regions and (b) lines**

- **Boundary-point distance approach.** This approach quantifies the conflict of a couple of geometries as the sum of distances between corresponding boundary points of geometries when geometries violate versus when they satisfy a topological relation. This distance is normalized by the distance of the maximum possible translation of the boundary points, which in our case is defined by the maximum distance between adjacent cells. As an example consider the distribution of geometries in Figure 5(a). Assume that these geometries must touch, so that a translation of geometries that solves this conflict is shown in Figure 5(b). Then, a degree of violation in this case is equivalent to the normalized distance of the five boundary points that define the closed polygons in Figures 5(a) and (b), where only one geometry has changed. In this case, this is equivalent to 0.11.



**Figure 5: Example of translation-based measure: (a) Original instance and (b) Instance after modification**

We define  $GF_{pd} = \frac{\sum_{i=1}^{CTR} (1-PD_i)}{CTR}$  as the complement of the normalized distance ( $PD_i$ ) of each checked pair of tuples. Notice that this alternative measure is only used as a reference, since it requires to have a corresponding consistent database instance to compare with, which, is most likely not available or is costly to compute. Even more, different corresponding con-



sistent databases exist with respect to an inconsistent database, which leads us to the problem of finding a consistent instance that differs minimally from the inconsistent database instance.

### 5.3 Comparison at the constraint level

A comparison at the constraint level illustrates the main differences among the three basic conflict measures. In the following we give representative cases taken from the synthetic database instance to illustrate main differences between the conflict measures. For each expected topological relation, we selected three different cases with the same topological relations but different geometries and provide with the different conflict measures ( $CD$ : proposed constraint-violation measure<sup>2</sup>,  $SD$ : normalized semantic distance,  $PD$ : measure based on boundary-point distance). For space limitations, we only show partial results in Figures 6– 8.

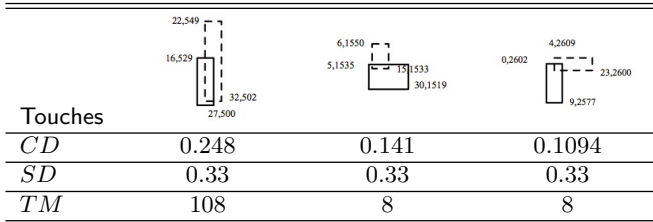


Figure 6: Comparison between measures with respect to expected relation Disjoint between regions

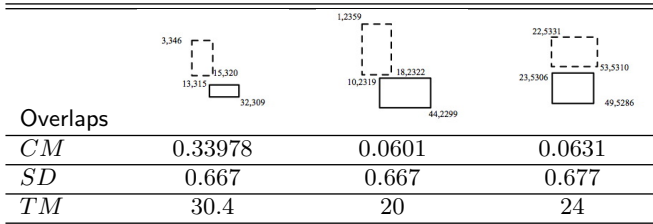


Figure 7: Comparison between conflict measures with respect to expected relation Overlaps between regions

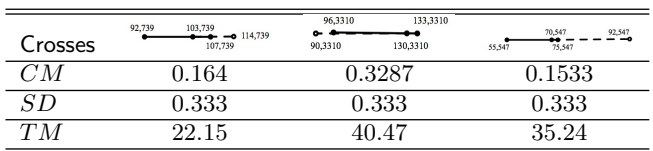


Figure 8: Comparison between conflict measures with respect to expected relation Crosses between lines

### 5.4 Comparison at the database level

In the following we show a subset of results obtained for the whole experimental evaluation, which reflect differences in the definition of the global fulfillment measure. Figure 9 shows the results obtained with the synthetic database instances for the three different measures (i.e, Metrics:  $GF$ ,  $GF_{sd}$ , and  $GF_{pd}$ ) when the expected topological relation

<sup>2</sup> $CD$  represents the degree of violation without considering the relative relevance of objects

was Disjoint and geometries hold another topological relation. Recall that for this relation, our constraint-violation measures include a minimum expected distance between geometries. In this graph, x-axis represents the percentage of tuples that were modified in the original consistent database instance and the y-axis represents the degree of consistency of different database instances. Notice that we only show results for the Within and not its converse relation Contains, which gives equivalent results. Similar results were obtained for larger sizes of database instances.

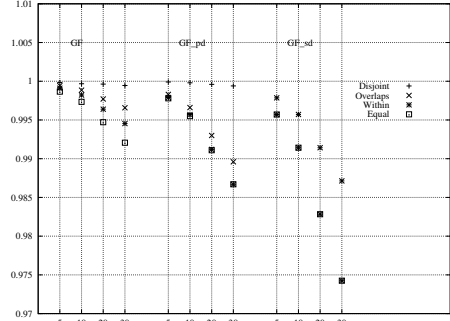


Figure 9: Global data quality with respect to relation Touches between regions

In similar way to the relation Touches, Figure 10 shows the results when the expected relation between regions is Overlaps. In this case, we estimated the overlapping area between geometries as the 10% of the perimeter of the smallest region under consideration.

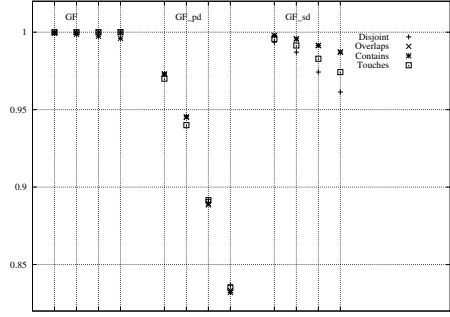


Figure 10: Global data quality with respect to relation Within between regions

Figure 11 shows the results when the expected relation between lines is Crosses. For this expected relation and a conflicting relation Touches, the constraint-violation measure adds the minimum length that must cross a line. In this case, we also consider the 10% of the length of the shortest line under consideration.

For the real data set, Table 8 shows the result of the test with respect to the first TD and for the nine modified datasets grouped in three rows of values. The top row shows the results when the tolerance used for the simplification is small, the middle row shows the results when the tolerance is medium and the bottom row shows the results when the tolerance is high. It must be noticed that the value of CTR can change in the different datasets because the modifications of the geometries may cause that counties that intersect in a test do not intersect in another one,



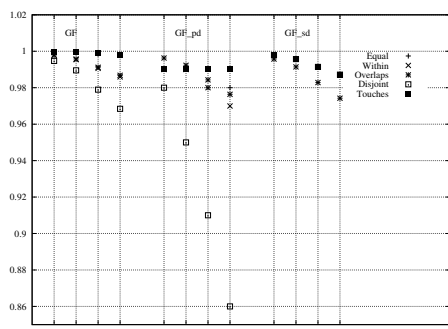


Figure 11: Global data quality with respect to relation Crosses between lines

and therefore they are no longer part of the CTR. Moreover, using different tolerances for the simplification of the geometries causes that the number and the importance of the violations between the geometries may vary. For each of the rows it can be seen that both measures (VS and GF) correlate with the common sense, because VS (spread of violation) increases and GF (global fulfillment) decreases as the number of modifications to the original data is larger. If the Table is read vertically, that is, if we consider a single column and therefore we fix the number of geometries modified, the values of VS and GF look less sensible because they do not always increase and decrease, respectively, when the tolerance used for the simplification is larger. This is caused by the fact that the number of violations does not always increase either. Even more, the degree of violation may differ in different conflicts, affecting in different ways GF, but different conflicts weight the same in VS.

Tolerance	Measure	Geometries modified		
		5%	25%	50%
Small	CTR	294	294	294
	Violation	28	100	200
	VS	0,09524	0,34014	0,68027
	GF	0,99582	0,98640	0,97437
Medium	CTR	294	294	290
	Violation	22	122	186
	VS	0,07483	0,41497	0,64138
	GF	0,99723	0,97932	0,97546
High	CTR	294	286	286
	Violation	18	116	196
	VS	0,06122	0,40559	0,68531
	GF	0,99626	0,98144	0,97131

Table 8: Results of the test *county Touches county*

Table 9 shows the result with respect to the second TD for the nine modified datasets. In this test, the value of CTR does not change because the algorithm uses a foreign key that does not change when the geometry is modified. The results of this test do not show a clear tendency as the previous test. The cause is that the geometries that are involved in the topological relationship are larger (a state and a county instead of two counties), and therefore the values of GF are closer to 1 than in the previous test. However, the measures can be used as indicators of the quality of the data because the larger the number and degree of the violations, the larger the values of VS and the smaller the values of GF.

Tolerance	Measure	Geometries modified		
		5%	25%	50%
Small	CTR	62	62	62
	Violation	2	9	18
	VS	0,03226	0,14516	0,29032
	GF	0,99998	0,99971	0,99972
Medium	CTR	62	62	62
	Violation	2	8	19
	VS	0,03226	0,12903	0,30645
	GF	0,99998	0,99983	0,99962
High	CTR	62	62	62
	Violation	2	6	16
	VS	0,03226	0,09677	0,25806
	GF	0,99987	0,99990	0,99957

Table 9: Results of the test *county Within state*

## 6. CONCLUSIONS AND FUTURE WORK

This work has presented different measures that compare topological relations between geometric attributes stored in a database with respect to expected topological relations. This is used to give a global evaluation of the database data quality with respect to topological dependency constraints. We evaluate the proposed measures with respect to two other two independent approaches: semantic distance and boundary-point distance. The results indicate that while our measures are correlated to other measures, it provides metric details about the degree in which two geometries are in conflict. The proposed measures refine the comparison of topological relations done by semantic distance approaches since it enables to distinguish among cases with the same topological relation but different spatial realization.

As an extension of the current work, we will explore measures that combine geometries of different dimensions, in particular, between lines and regions. We have here concentrated on 2D geometries, but we would like to extend our work to deal with geometries of higher dimension. We will also analyze the global evaluation of an inconsistent database under other types of topo-semantic integrity constraints, such as for example, referential constraints of the type “a building must be inside of a land parcel”. Another interesting analysis left for future work is to consider the interaction of integrity constraints, where the global fulfillment measure may not be simply the sum of isolated and independent constraint-violation measures.

## 7. REFERENCES

- [1] S. Berreti, A. D. Bimbo, and E. Vicario. The computational aspect of retrieval by spatial arrangement. In *International Conference on Pattern Recognition*, 2000.
- [2] K. Borges, C. Davis, and A. Laender. Integrity constraints in spatial databases. In *Database Integrity: Challenges and Solutions*. Ideas Group, 2002.
- [3] K. Borges, A. Laender, and C. Davis. Spatial Integrity Constraints in Object Oriented Geographic Data Modeling. In *ACM-GIS*, pages 1–6, 1999.
- [4] L. Bravo and M. A. Rodríguez. Semantic integrity constraints for spatial databases. In *Proceedings of the 3rd Alberto Mendelzon International Workshop on Foundations of Data Management, Arequipa, Peru*, volume 450. CEUR-WS.org, 2009.

- [5] S. Cockcroft. A Taxonomy of Spatial Integrity Constraints. *GeoInformatica*, 1(4):327–343, 1997.
- [6] M. Egenhofer and K. Al-Taha. Reasoning about gradual change of topological relationships. In A. Frank, I. Campari, and U. Formentini, editors, *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space, LNCS 636*, pages 196–219. Springer-Verlag, 1992.
- [7] M. Egenhofer and R. Franzosa. Point Set Topological Relations. *IJGIS*, 5:161–174, 1991.
- [8] M. Egenhofer and D. Mark. Modeling conceptual neighborhoods of topological line-region relations. *International Journal of Geographic Information Science*, 9(5):555–565, 1995.
- [9] R. Elmasri and S. B. Navathe. *Fundamentals of Database Systems, 2nd Edition*. Benjamin/Cummings, 1994.
- [10] D. M. Gabbay and A. Hunter. Making inconsistency respectable: a logical framework for inconsistency in reasoning. In *Fundamentals of Artificial Intelligence Research, International Workshop FAIR*, volume 535 of *Lecture Notes in Computer Science*, pages 19–32. Springer, 1991.
- [11] F. A. Godoy and M. A. Rodríguez. Defining and comparing content measures of topological relations. *GeoInformatica*, 8(4):347–371, 2004.
- [12] T. Hadzilacos and N. Tryfona. A Model for Expressing Topological Integrity Constraints in Geographic Databases. In *Spatio-Temporal Reasoning*, Springer LNCS 639, pages 252–268, 1992.
- [13] A. Hunter and S. Konieczny. Approaches to measuring inconsistent information. In *Inconsistency Tolerance*, volume 3300 of *Lecture Notes in Computer Science*, pages 191–236. Springer, 2005.
- [14] S. Mäs. Reasoning on Spatial Semantic Integrity Constraints. In *COSIT*, Springer LNCS 4736, pages 285–302, 2007.
- [15] S. Mas and W. Reinhardt. Categories of geospatial and temporal integrity constraints. In *Advanced Geographic Information Systems and Web Services, 2009. GEOWS 09. International Conference*, pages 146–151, 2009.
- [16] Max J. Egenhofer. Categorizing topological relations between regions, lines, and points in geographic databases. Technical Report 94-1, NCGIA, 2004.
- [17] OpenGis. Opengis Simple Features Specification for SQL. Technical report, Open GIS Consortium, 1999.
- [18] C. Ordonez, J. García-García, and Z. C. 0002. Measuring referential integrity in distributed databases. In *Proceedings of the First Workshop on CyberInfrastructure: Information Management in eScience, CIMS 2007, Sixteenth ACM Conference on Information and Knowledge Management, CIKM 2007, Lisbon, Portugal, November 9, 2007*. ACM, 2007.
- [19] D. Papadias, N. Mamoulis, and V. Delis. Algorithms for querying spatial structure. In *VLDB Conference*, pages 546–557, 1998.
- [20] D. Randell, Z. Cui, and A. Cohn. A Spatial Logic based on Regions and Connection. In *Proceedings of the 3rd International Conference on Knowledge Representation and Reasoning*, pages 165–176. Morgan Kaufmann, 1992.
- [21] D. Randell, Z. Cui, and A. Cohn. A spatial logic based on regions and connection. In B. Nebel, C. Rich, and W. Swarthout, editors, *Principles of Knowledge Representation and Reasoning*, pages 165–176. Morgan Kaufmann, 1992.
- [22] R. M. P. Reis, M. J. Egenhofer, and J. L. G. Matos. Conceptual neighborhoods of topological relations between lines. In *13th International Symposium on Spatial Data Handling, Lecture Notes in Geoinformation and Cartography*, pages 557–574, 2008.
- [23] M. A. Rodríguez, L. E. Bertossi, and M. C. Marileo. An inconsistency tolerant approach to querying spatial databases. In *16th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems, ACM-GIS 2008, November 5-7, 2008, Irvine, California, USA, Proceedings*. ACM, 2008.
- [24] S. Servigne, T. Ubeda, A. Puricelli, and R. Laurini. A Methodology for Spatial Consistency Improvement of Geographic Databases. *GeoInformatica*, 4(1):7–34, 2000.
- [25] U.S. Census Bureau. 2009 tiger/line shapefiles. Retrieved June 2010 from <http://www.census.gov/geo/www/tiger/>.