Checking the Integrity of Spatial Semantic Integrity Constraints*

Stephan Mäs

AGIS - Arbeitsgemeinschaft GIS,
Universität der Bundeswehr München,
Werner Heisenberg Weg 39,
85577 Neubiberg, Germany

{Stephan.Maes}@unibw.de

Abstract. Integrity constraints play a major role when the quality of spatial data is checked by automatic procedures. Nevertheless the possibilities of checking the internal consistency of the integrity constraints themselves are hardly researched yet. This work analyses the applicability of reasoning techniques like the composition of spatial relations and constraint satisfaction in networks of relations to find conflicts and redundancies in sets of spatial semantic integrity constraints. These integrity rules specify relations among entity classes. Such relations must hold to ensure that the data complies with the semantics intended by the data model. For spatial data, many semantic integrity constraints are based on spatial properties described for example through qualitative topological or metric relations. Since integrity constraints are defined at the class level, the reasoning properties of these spatial relations can not directly be applied at that level. Therefore a set of class relations has been defined which, combined with the instance relations, enables for the specification of integrity constraints and logical reasoning on them.

Keywords: Semantic Integrity Constraints, Spatial Relations, Class Level Relations, Reasoning, Consistency of Constraints, Constraint Networks

1 Introduction

Semantic integrity constraints specify relations which refer to the semantics of the concepts represented by the data model. Unlike domain or key and relationship constraints, which are usually inherently or implicitly specified through the data model constructs, semantic integrity constraints have to be explicitly specified. This is mostly done by programming checking procedures for database transactions or through the declaration via an assertion specification language. [3]

Semantic integrity constraints play a major role when the logical consistency of a data set has to be evaluated. For spatial data in particular constraints which comprehend the spatial peculiarities are of interest. Therefore spatial semantic

* A more extensive study on this will be published in [6]

Dagstuhl Seminar Proceedings 07212
Constraint Databases, Geometric Elimination and Geographic Information Systems
http://drops.dagstuhl.de/opus/volltexte/2007/1285

integrity constraints restrict the spatial properties of the modelled concepts. These restrictions are mostly defined through qualitative constraints. While quantitative constraints rely on quantitative measurements or calculations, qualitative relations are non-numerical descriptions of a situation. Thereby continuous properties are represented by discrete sets of relations. Qualitative representations are characterized by making only as many distinctions in the domain as necessary in a given context [5]. Thus qualitative descriptions are less precise than the quantitative. A typical example of qualitative spatial relations used for the definition of integrity constraints are the topological relations between areal entities shown in figure 1.

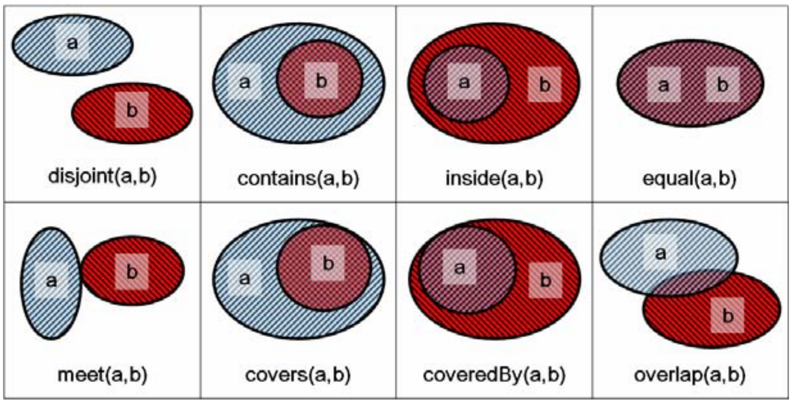

Fig. 1. Set of topological relations between areal entities

For the formalisation of integrity constraints this abstraction is sufficient, since they mostly restrict an infinite set of possible situations through the definition of a single abstract rule. Verbal descriptions of semantic integrity constraints are usually also not numerically precise and for users the abstract qualitative descriptions are easier to understand than numerical information. Examples of such natural language descriptions of spatial semantic integrity constraints are for topological restrictions: “contour lines are not allowed to intersect with a lake”, directional restrictions: “the backyard should always be in the back of a house” and metric restrictions: “a petrol station must be at least 300 meters away from a school”.

For the quality assurance of a geodata set many of such spatial integrity constraints can be defined. At present, the formalisation of contents and restrictions of semantic integrity constraints are not sufficiently researched. Thus their definition and management is mostly reserved to specialists since inexperienced users are likely to define redundant or conflicting constraints. This paper aims to contribute to a solution of these problems. It investigates reasoning algorithms which can be used to check the internal consistency of a set of spatial semantic integrity constraints. Focus is on semantic integrity constraints on binary spatial relations between entity classes.

Integrity constraints are defined at the level of the entity classes, but must be linked to the instance relations which are proved for all instances during quality assurance. As extensively argued in [1] and [6] these instance relations do not correctly represent the semantics of class relations. Therefore the formalisation of semantic integrity constraints requires specific class level relations. In the following chapter, the main properties of class relations are analysed and based on these properties a set of 17 abstract class relations is defined. Chapter 3 reviews logical properties of those properties and shows how they can be used to find conflicts and redundancies in sets of semantic integrity constraints.

2 Class Relations for Integrity Constraint Definition

2.1 Properties of Class Relations

As demonstrated in [1] and [6] the logical properties of instance relations do not hold, if they are directly applied at the class level. To overcome this problem a set of class level relations has to be defined. For the definition of integrity constraints the class relations should represent existential and universal quantifiers for the corresponding instance relations. In the following four basic properties of class relations are defined and later on combined for the definition of a set of class relations. For the validity of these definitions the involved classes must conform to the following two requirements. First, they must have at least one instance, i.e. empty classes are not valid. As stated before class relations are linked to instance relations. Thus the second condition specifies that if a class relation is defined, there must be at least one corresponding instance relation existent among the instances of the involved classes.

x,y,z	Denote variables for individuals / instances. Every instance must be associated with a class.
A, B, C	Denote variables for classes. Every class must have at least one instance.
Inst(x,A)	Means individual x is an instance of class A.
r(x,y)	Means individual x has the relation r to individual y; x and y are said to participate in the relationship instance r. The meta-variable r can stand for any relation of individuals (e.g. topological relations). Every relationship instance r can be associated to a class relation R.
R(A,B)	Denotes that R relates the classes A and B. The meta-variable R can stand for any class relationship. Every R is related to an instance relation r. If a class relation R(A,B) is defined at least one r must exist between the instances of A and B.

$$\mathrm{LT}(A,B,r) := \forall x(\mathrm{Inst}(x,A) \rightarrow \exists y(\mathrm{Inst}(y,B) \cap r(x,y))). \quad (1)$$

$$\mathrm{RT}(A,B,r) := \forall y(\mathrm{Inst}(y,B) \rightarrow \exists x(\mathrm{Inst}(x,A) \cap r(x,y))). \quad (2)$$

$$\text{LD}(A,B,r) := \forall x,y,z(\text{Inst}(x,A) \cap \text{Inst}(y,B) \cap \text{Inst}(z,A) \cap r(x,y) \cap r(z,y) \to x = z) \cap \text{ex}(A,B,r). \quad (3)$$

$$\text{RD}(A,B,r) := \forall x,y,z(\text{Inst}(x,A) \cap \text{Inst}(y,B) \cap \text{Inst}(z,B) \cap r(x,y) \cap r(x,z) \to y = z) \cap \text{ex}(A,B,r). \quad (4)$$

$$\text{ex}(A,B,r) := \exists x \exists y(\text{Inst}(x,A) \cap \text{Inst}(y,B) \cap r(x,y)). \quad (5)$$

Equations (1) and (2) define left totality and right totality of a class relationship, respectively. (1) holds if every instance of A has a relation r to some instance of B. (2) holds if for each instance of B there is some instance of A which stands in relation r to it. This means that every instance of B has the inverse relation of r to some instance of A. These constraints define a total participation for the classes A or B, respectively.

Equations (3) and (4) restrict the cardinality ratio of the class relation. (3) specifies that there is no instance of B that has more than one instance of A which stands in relation r to it. This relation restricts the number of R relations an instance of B can participate in; the instances of A are not restricted. Class relations, for which (3) holds, are left-definite. (4) holds for a class relation if no instance of A participates in a relationship instance of R to more than one instance of B. When this relation is defined, all instances of A are restricted while the instances of B are not affected. Equation (5) is used in (3) and (4) to ensure that at least one instance relation r does exist between the instances of A and B.

Such properties of class relations are well established in data modelling, for example when total participation and cardinality ratio constraints are described using the Entity-Relationship notation. In such models a total participation is represented by a double line for the relation and cardinality ratio for example by a N:1 next to the relation signature (see figure 2). In this example all buildings are restricted to be contained by exactly one parcel, while the parcels are allowed to contain an undefined number of buildings.

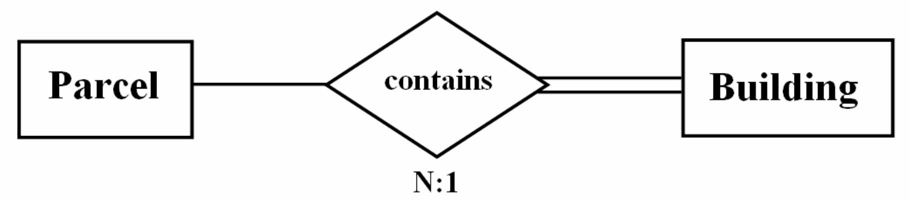

Fig. 2. Constraints in an Entity-Relationship Model

2.2 Definition of Class Relations

As the example in figure 2 illustrates, the properties defined in the previous subsection can be combined for the definition of a class relation. The four properties are independent, which means none of the properties implies or precludes one of the others. The class relation in the example is based on the topological instance relation *contains* and the properties left definite and right total. The other two properties are not valid. The corresponding class relation *CONTAINS*$_{LD.RT}$*(Parcel, Building)* is based on the abstract class relation defined in equation (6).

$$R_{RD.LT}(A,B) := LD(A,B,r) \cap RT(A,B,r) \cap \neg RD(A,B,r) \cap \neg LT(A,B,r). \qquad \textbf{(6)}$$

In analogy to equation (6) the four properties can be used to define a set of 15 abstract class relations, where for each relation at least one of the four properties holds and the others are excluded, respectively.

For a better description of integrity constraints and to allow for reasoning two additional cases are considered in this approach. First an abstract class relation is defined for the situation that none of the four properties is valid and nevertheless some instances of A stand in relation r to some instances of B (equation (7)). An example is shown in figure 3a. Here the arrows link instances, which are related by the same instance relation. The class relation is neither total nor definite for the classes A or B. This case must be considered to achieve a jointly exhaustive set of relations. Second, an additional abstract class relation is defined for the case that all instances of A have a relationship instance of R to all instances of B (Equation (8)). As illustrated in figures 3b to 3d this is a special case of a left total and right total relation ($R_{LT.RT}(A,B)$). The corresponding integrity constraints are very restrictive, which is for example useful, when all instances of two classes are not allowed to intersect: *DISJOINT* $_{LT.RT\text{-}all}$ *(Streets, Lakes)*.

$$R_{some}(A,B) := ex(A,B,r) \cap \neg LD(A,B,r) \cap \neg RD(A,B,r) \cap \neg LT(A,B,r) \cap \neg RT(A,B,r). \qquad \textbf{(7)}$$

$$R_{LT.RT-all}(A,B) := \forall x \forall y (Inst(x,A) \cap Inst(y,B) \rightarrow r(x,y)). \qquad \textbf{(8)}$$

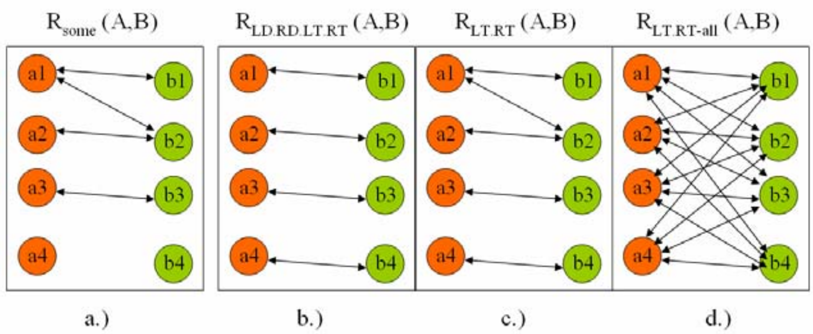

Fig. 3. Examples of class relations

The set of 17 class relations enables the definition of integrity constraints based on any kind of instance relation. The detailed definitions of the abstract class relations can be found in [6]. It is possible to define more than one class relation between two classes, even when the applied instance relations are part of the same jointly exhaustive and pair wise disjoint (JEPD) set of instance relations. Nevertheless it can be proven that for a JEPD set of instance relations the corresponding class relations are also JEPD.

3 Reasoning on Integrity Constraints

The application of reasoning algorithms for checking consistency and discovering redundancies in networks of instance relations has for example been demonstrated in [2] and [7]. This approach applies these reasoning concepts to the class level relations. The proof of consistency of a network of binary relations is a constraint satisfaction problem. In a consistent network of JEPD relations the following three constraints are fulfilled: node consistency, arc consistency and path consistency.

Node consistency is satisfied if every node has an identity relation. For the class relation networks this means that every class must have a relation to itself. This requirement is fulfilled if a corresponding identity instance relation is available, for example for the topological relations in figure 1 the identity class relation is $EQUAL_{LD.RD.LT.RT}(A, A)$.

A network of relations is arc consistent if every edge of the network has an edge in the reverse direction, i.e. every relation has an inverse relation. For class relations this is only possible, if the corresponding instance relations have an inverse relation or are symmetric. Most spatial relations fulfil this requirement. The symmetry properties of the class relations can be derived from the symmetry of the applied instance relations and the definitions of their properties (equations (1)-(5), (7) and (8)). The inverse of a class relation is also based on the inverse of the applied instance relation. If a class relation is left total / left definite the inverse relation is right total / right definite and vice versa. The following two examples demonstrate the derivation of inverse class relations. Here the class relations are based on the symmetric instance relation *disjoint* and the inverse relations *contains* and *inside*:

$$(DISJOINT_{RD.LT}(A, B))^i = DISJOINT_{LD.RT}(B,A).$$
$$(CONTAINS_{LD.RD.LT}(A, B))^i = INSIDE_{LD.RD.RT}(B,A).$$

Therewith it can be proven that if an instance relation is symmetric or has an inverse relation, there is also an inverse relation for each of the corresponding class relations. If there are more than one integrity constraints defined between two classes the consistency of these constraints has to be additionally proven for the arc consistency.

For the proof of path consistency the compositions of all possible node triples must be checked. The composition of binary relations enables for the derivation of implicit knowledge about a triple of entities. If two binary relations are known the corresponding third one can be inferred or some of the potential relations can be excluded. This knowledge can also be used to find conflicts if all of the three relations are known. The compositions of a set of relations are usually stored in a composition table like it has been done for the topological relations between areal entities in [4].

In general, the composition of class relations is not independent of the composition of the applied instance relations. The composition is only possible if the applied instance relations belong to the same set of JEPD relations and this set allows for compositions at the instance level. Using for example the 17 class relations together with the 8 topological relations between regions (see figure 1) results in 136 topological class relations and almost 18500 compositions. Since such an amount of compositions is hardly manageable, a two level reasoning formalism is proposed here. This approach separates the compositions of the abstract class relations from those of the instance relations (see figure 4). Therewith the composition of class relations can be defined independently of a certain set of instance relations.

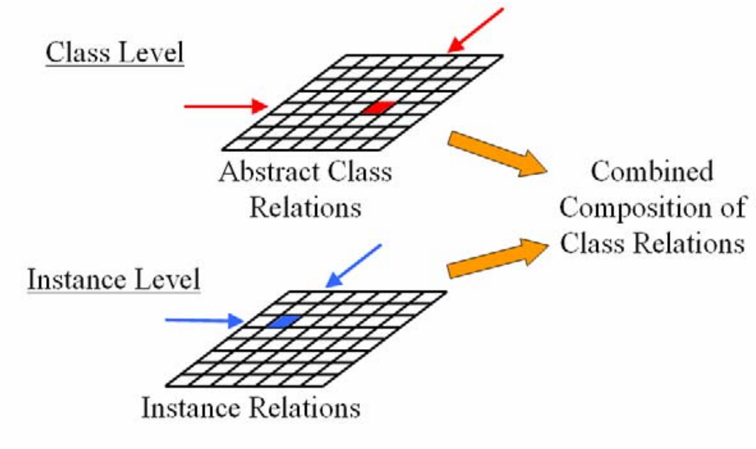

Fig. 4. Two levels composition of class relations

The following example illustrates the two levels composition. With the class relations *MEET*$_{LT.RT\text{-}all}$*(A,B)* and *COVERS*$_{LT.RT\text{-}all}$*(B,C)* given, the relation between the classes A and C can be derived. Such a situation is schematically represented in figure 5. All instances of A have a *meet* relation to all instances of B and all instances of B have a *covers* relation instances of C. Therewith the composition of the abstract class relations can be derived: since all instances of A have the same kind of relation to all instances of B and all instances of B participate in same kind of relation to all instances of C, it is obvious that all instances of A must have the same relation to all instances of C. In other words, every possible triple of instances of A, B and C is related by the same relations. Thus the composition of the abstract class relations must be:

$$R1_{LT.RT-all}(A,B) \cap R2_{LT.RT-all}(B,C) \Rightarrow R3_{LT.RT-all}(A,C).$$

The composition at the instance level is (taken from the composition table in [4]):

$$\text{meet}(a,b) \cap \text{covers}(b,c) \Rightarrow \text{disjoint}(a,c) \cup \text{meet}(a,c).$$

The combination of the compositions of the two levels results in:

$$\text{MEET}_{\text{LT.RT-all}}(\text{A},\text{B}) \cap \text{COVERS}_{\text{LT.RT-all}}(\text{B},\text{C}) \Rightarrow [\text{DISJOINT} \cup \text{MEET}]_{\text{LT.RT-all}}(\text{A},\text{C}).$$

This conclusion is obvious, because the given class relations constrain all relations among the instances of the three classes. Nevertheless this example shows that the composition of the defined class relations is possible and the compositions of other class relations can be derived in a similar way.

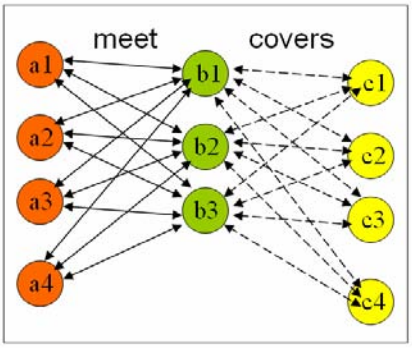

Fig. 5. Scene defined by the class relations $MEET_{LT.RT\text{-}all}(A,B)$ and $COVERS_{LT.RT\text{-}all}(B,C)$

With the described logics it is possible to find conflicts and redundancies in networks of class relations. This can for example be applied to prove consistency of sets of integrity constraints.

4 Conclusion

In this paper, 17 abstract class level relations are defined, which enable a formalised specification of semantic integrity constraints. The investigated reasoning concepts can be used to find conflicts and redundancies in sets of spatial semantic integrity constraints. The introduced two levels composition of class relations allows for a separate analysis of instance relations and abstract class relations. Therewith the overall reasoning formalism can be used with any spatial or non-spatial JEPD set of instance relations.

There are many different kinds of semantic integrity constraints, but not all of them can be covered by this approach. The framework presented in this paper covers constraints based on binary instance relations. Semantic integrity constraints, which constrain three or more classes or attributes, are not considered. Further on, only total participation and a cardinality ratio of 0..1 are included. Nevertheless this framework provides a basis, which can be extended for other possibly more complex types of semantic integrity constraints.

References

1. Donnelly, M., Bittner, T.: Spatial Relations Between Classes of Individuals. In: Cohn, A.G., Mark, D.M. (eds.) COSIT 2005. Lecture Notes in Computer Science, vol. 3693, pp. 182–199. Springer, Heidelberg (2005)

2. Egenhofer, M., Sharma, J.: Assessing the Consistency of Complete and Incomplete Topological Information. Geographical Systems 1(1), 47–68 (1993)

3. Elmasri, R., Navathe, S.B.: Fundamentals of Database Systems, 2nd edn. (Addison-Wesley), The Benjamin/Cummings Publishing Company Inc. (1994)

4. Grigni, M., Papadias, D., Papadimitriou, C.: Topological inference. In: Proceedings of the International Joint Conference of Artificial Intelligence (IJCAI), pp. 901–906 (1995)

5. Hernandez, D.; Clementini, E.; Felice, P.D.: Qualitative Distances. In: Spatial Information Theory: A Theoretical Basis for GIS International Conference COSIT, Semmering, Austria, 1995, A.U. Frank and W. Kuhn (Eds.), Lecture Notes in Computer Science, 988, pp. 45 - 57. (1995)

6. Mäs, Stephan: Reasoning on Spatial Semantic Integrity Constraints. In: Conference on Spatial Information Theory (COSIT 2007), Melbourne, Australia; Lecture Notes in Computer Science 4736, S. Winter et al. (Eds.), pp. 285–302. Springer, Heidelberg (2007)

7. Rodríguez, A., Van de Weghe, N., De Maeyer, P.: Simplifying Sets of Events by Selecting Temporal Relations. In: Egenhofer, M.J., Freksa, C., Miller, H.J. (eds.) GIScience 2004. Lecture Notes in Computer Science, vol. 3234, pp. 269–284. Springer, Heidelberg (2004)