Topological Relationships

Eliseo Clementini

January 2025

Abstract

Topological relationships constitute a category of binary relationships between spatial objects. They are based on specific connection properties between geometries, mostly related to having common parts, and disregarding other spatial properties such as the distance and the direction between geometries. There exist many variations of models for representing topological relationships, due to the underlying adopted spatial model and the context of applications. In spatial databases, adopting the vector data model of the Open GeoSpatial Consortium, topological relationships are evaluated either on the base of the Dimension-Extended 9-Intersection Model (DE9-IM) or the so-called Named Topological Relationships.

1 Keywords

Spatial Relationships; Spatial Data Model; Vector Data Model; Simple features; OGC standard.

2 Definitions

Point-set topology deals with basic set-theoretic definitions related to points of the line, the plane or 3D space. A crucial component of point-set topology is the concept of open set. In fact, a topology is defined as a family of open sets with union and intersection operations. For a formal treatment of topology, we refer to (Munkres, 2014), while in the rest of this chapter we will try to limit the formal definitions to the essentials and accompany them with practical explanations. Informally, an open set is a point-set that do not contain its boundary, and a closed set is a point-set that contains its boundary. An open set is connected if it is possible to draw a line between any two points of the set and all the points in the path belong to the set.

To understand the meaning of the word "topological", one fundamental notion is the homeomorphism, which is a mathematical transformation that maps near points in one space to near points in another space. Informally, a homeomorphism is a transformation that changes the shape of an object as it were

"elastic", without provoking ruptures or gluing its parts. Therefore, a topological property remains invariant under a homeomorphism. Speaking of relationships between two objects, a topological relationship is unaltered if the scene made up by the two objects undergoes a homeomorphism. In essence, topological relationships capture connection properties of two objects that are invariant even if the objects would change their shape. Formally, the definition of homeomorphism is the following:

Definition 2.1. Let X and Y be two topological spaces. A mapping $f\colon X \to Y$ is a *homeomorphism* if both f and f^{-1} are continuous functions.

Spatial objects, that is, various kinds of points, lines, and regions, need to be precisely defined in geometric terms, in such a way that topological relationships can be calculated without ambiguities. Since various models for topological relationships make use of the fundamental notions of objects' boundaries and interiors, the latter must be defined as well. In some cases, the adopted GIS definitions deviate from mathematical rigorousness to accommodate practical situations. For example, in topology, a simple line embedded in $\mathbb{R}^2$, being a one-dimensional set, has an empty interior. Following normal practice in both GIS (Egenhofer and Herring, 1990) and in OGC standards (OGC Open Geospatial Consortium Inc., 1999), a line boundary is the set of its end-points and a line interior corresponds to all its points except the end-points. Hence, in this case the GIS definition deviates from mathematics. Let us recap the main definitions, adapted from (Clementini and Di Felice, 1996). Let x be a two-dimensional point-set embedded in $\mathbb{R}^2$.

Definition 2.2. The *interior* x° of x is the union of all open sets contained in x.

Definition 2.3. The *closure* $\overline{x}$ of x is the intersection of all closed sets containing x.

Definition 2.4. The *boundary* ∂x of x is the set difference between its closure and interior, i.e., $\overline{x} - x^\circ$.

Definition 2.5. The *exterior* x^- of x is the set difference $\mathbb{R}^2 - \overline{x}$.

Definition 2.6. x is *regular closed* if $x = \overline{x^\circ}$.

Definition 2.7. A *simple region* is a regular closed non-empty two-dimensional point-set x with a connected interior and connected exterior.

Definition 2.7 implies that a simple region is homeomorphic to the closed unit disk. A simple region does not have holes and is connected. If we omit the constraint of connected exterior from the definition, we obtain regions with holes (Egenhofer et al., 1994):

Definition 2.8. A *region with holes* is a regular closed non-empty two-dimensional point-set x with a connected interior.

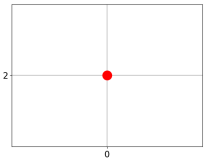
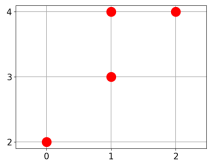
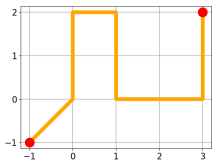
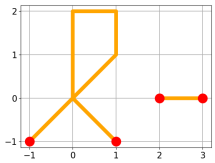
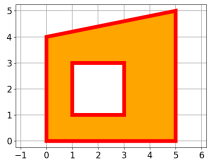
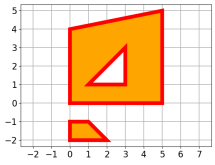
Spatial Type	geometry	WKT representation
POINT		POINT (0 2)
MULTIPOINT		MULTIPOINT (0 2, 1 3, 2 4, 1 4)
LINESTRING		LINESTRING (-1 -1, 0 0, 0 2, 1 2, 1 0, 3 0, 3 2)
MULTILINESTRING		MULTILINESTRING ((-1 -1, 0 0, 0 2, 1 2, 1 1, 0 0, 1 -1), (2 0, 3 0))
POLYGON		POLYGON ((0 0, 5 0, 5 5, 0 4, 0 0), (1 1, 3 1, 3 3, 1 3, 1 1))
MULTIPOLYGON		MULTIPOLYGON (((0 0, 5 0, 5 5, 0 4, 0 0), (1 1, 3 1, 3 3, 1 1)), ((0 -1, 0 -2, 2 -2, 1 -1, 0 -1)))

Table 1: OGC spatial types and their WKT representation.

Region with holes are implemented with the Polygon spatial data type in the OGC simple feature specifications. For the details about the vector data model adopted by the OGC, we refer to the OGC documentation. In this chapter, we limit ourselves to illustrate the OGC spatial data types together with their Well-Known Text (WKT) representation in Table 1. Omitting the constraint of a connected interior results in complex regions, i.e., regions with holes and separations:

Definition 2.9. A *complex region* is a regular closed non-empty two-dimensional point-set x.

The MultiPolygon spatial data type is used in the OGC feature models to implement complex regions.

Definition 2.10. A *simple line* is a closed non-empty one-dimensional point-set x defined as the image of a continuous mapping $f : [0, 1] \to \mathbb{R}^2$, such that $\forall t_i, t_j \in [0, 1], t_i \neq t_j, f(t_i) \neq f(t_j)$.

Thus a simple line is defined as the mapping of the unit interval in the plane with no self-intersections. Intuitively, a simple line can be constructed by taking a pen and tracing a line on a piece of paper, never passing twice through the same position and never removing the pen until the line is finished. The initial and final points of a simple line, otherwise called the endpoints of the line, are denoted as $f(0)$ and $f(1)$. The boundary ∂x of a line x is defined to be the set of its two endpoints, whilst the interior of the line is defined as $x - \partial x$. Simple lines are implemented with the Linestring spatial data type in the OGC feature model. If the constraint of no self-intersections is removed from Definition 2.10, then we obtain lines with self-intersections, a particular case of which is the closed ring, where $f(0) = f(1)$:

Definition 2.11. A *line with self-intersections* is a closed non-empty one-dimensional point-set x defined as the image of a continuous mapping $f : [0, 1] \to \mathbb{R}^2$.

A complex line is composed of several components that may be disjoint or not, i.e., the union of several mappings from the unit interval to the plane:

Definition 2.12. A *complex line* is a closed non-empty one-dimensional point-set x defined as the union of n images of continuous mappings $f_i : [0, 1] \to \mathbb{R}^2, \forall i \in 1..n$.

Complex lines are implemented with the MultiLinestring spatial data type in the OGC feature model.

Definition 2.13. A *simple point* is a zero-dimensional element of $\mathbb{R}^2$.

Definition 2.14. A *complex point* is a finite set whose elements are simple points.

Following the OGC convention, point features have an empty boundary and they coincide with their interior. The Point and Multipoint spatial data types are used to implement simple and complex point features in OGC standards, respectively.

type	R/R	L/R	P/R	L/L	P/L	P/P
DE-9IM	12	31	3	48	3	2

Table 2: The number of different topological relationships expressible with the DE-9IM for simple spatial objects, distinguished by type: region=R, line=L, point=P.

3 Description

Various models for representing topological relationships have been published starting from the year 1990. Three of them had major impact: the 9-intersection model (9IM) (Egenhofer and Franzosa, 1991) with its dimension extension DE-9IM (Clementini et al., 1993), the Region Connection Calculus (RCC-8) (Cui et al., 1993), and the Calculus Based Method (CBM) (Clementini et al., 1993). We will concentrate on the DE-9IM and the CBM that have been recommended in the OGC specifications. The DE-9IM corresponds to the *Relate* OGC operator and the CBM corresponds to the OGC named topological relationships (*Equals*, *Disjoint*, *Intersects*, *Touches*, *Crosses*, *Within*, *Contains*, *Overlaps*).

The DE-9IM categorizes topological relationships on the basis of the intersections between interiors, boundaries, and exteriors of two spatial objects A and B, which can be of any type (point, line, or region). The dimension of such intersections are indicated with values -1, 0, 1, 2, corresponding to an empty intersection, a 0-dimensional, 1-dimensional, and 2-dimensional intersection, respectively. The 9 intersections are structured in a 3×3 matrix, leading to the following definition:

Definition 3.1. Given two spatial objects A and B, the *DE-9IM* is the following matrix:

$$\begin{pmatrix} dim(A^\circ \cap B^\circ) & dim(A^\circ \cap \partial B) & dim(A^\circ \cap B^-) \\ dim(\partial A \cap B^\circ) & dim(\partial A \cap \partial B) & dim(\partial A \cap B^-) \\ dim(A^- \cap B^\circ) & dim(A^- \cap \partial B) & dim(A^- \cap B^-) \end{pmatrix} \quad (1)$$

While the matrix can have 4^9=262144 different values from a combinatorial point of view, only a limited number of values corresponds to actual spatial configurations with 2D geometries. Those configurations can be obtained with a lengthy process by imposing geometric constraints and eliminating the corresponding matrices. The number of possible configurations for simple objects is given in Table 2, distinguishing the type of involved objects.

If we consider complex geometries instead, such as regions with holes and separate components, the number of possible relationships is higher. Nobody up to now has calculated the number of possible relationships for complex features that is expressible with the 9I+DEM matrix. In the following figures, we illustrate the relationships for simple features. Figure 1 illustrates the 12 prototypes of topological relationships between simple regions. Figure 2 represents the 31 DE-9IM different relationships between a simple line and a simple region.

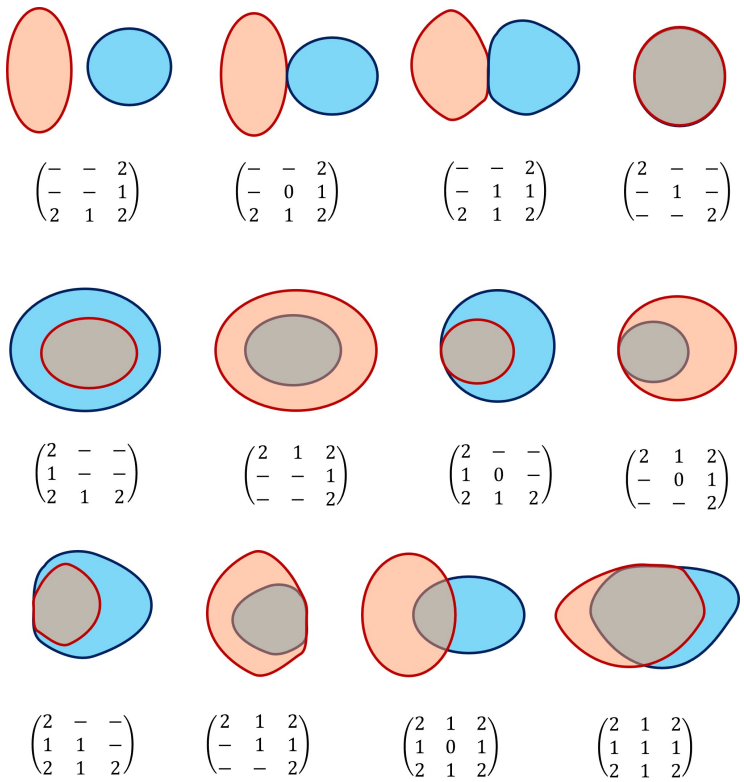


Figure 1: The 12 topological relationships between simple regions. Region A is red and region B is blue. In the matrices, the value -1 has been substituted by the symbol $-$ to make the matrices more compact.

Figure 3 represents the 48 DE-9IM different relationships between two simple lines.

The CBM provides five names of topological relationships that apply to categories of configurations involving two spatial objects, plus three boundary operators that in combination with the relationship names allow the formulation of logical expressions able to capture a variety of topological aspects. It has been proved equivalent to the DE-9IM in the sense that every DE-9IM matrix can be expressed by a CBM logical expression and, vice versa, every CBM relationship can be expressed by a set of DE-9IM matrices (Clementini and Di Felice, 1995). Given two spatial objects A and B, the CBM definitions are the following:

Definition 3.2. The *touch* relationship is defined as:

$$touch(A, B) \iff (A^\circ \cap B^\circ = \emptyset) \wedge (A \cap B \neq \emptyset) \tag{2}$$

Definition 3.3. The *in* relationship is defined as:

$$in(A, B) \iff (A^\circ \cap B^\circ \neq \emptyset) \wedge (A \cap B = A) \tag{3}$$

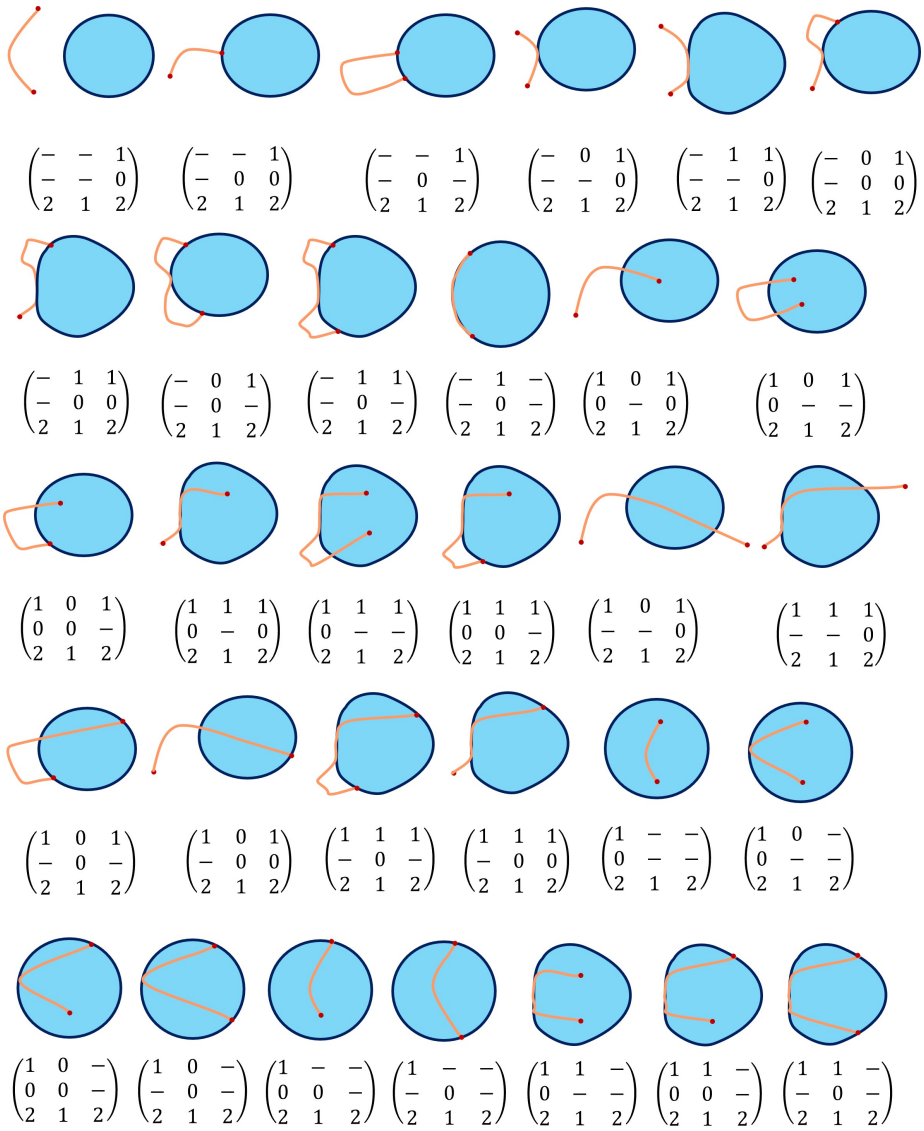


Figure 2: The 31 topological relationships between a simple line and a simple region.

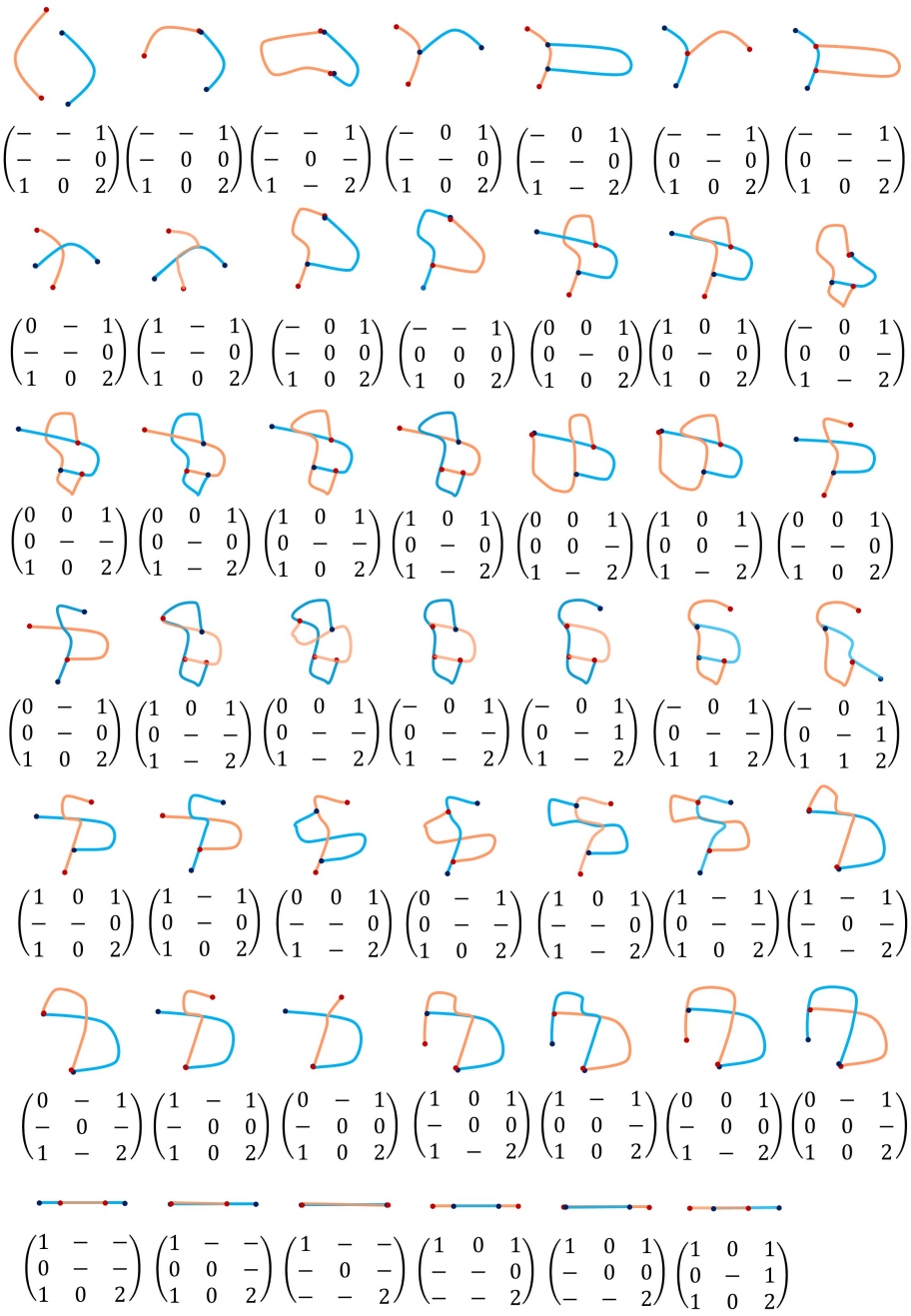


Figure 3: The 48 topological relationships between two simple lines.

Definition 3.4. The *cross* relationship is defined as:

$$cross(A,B) \iff (dim(A^\circ \cap B^\circ) < max(dim(A^\circ), dim(B^\circ))) \wedge \\ \wedge (A \cap B \neq A) \wedge (A \cap B \neq B) \tag{4}$$

Definition 3.5. The *overlap* relationship is defined as:

$$overlap(A,B) \iff (dim(A^\circ) = dim(B^\circ) = dim(A^\circ \cap B^\circ)) \wedge \\ \wedge (A \cap B \neq A) \wedge (A \cap B \neq B) \tag{5}$$

Definition 3.6. The *disjoint* relationship is defined as:

$$disjoint(A,B) \iff A \cap B = \emptyset \tag{6}$$

Definition 3.7. Given a region A and a line L, the *boundary* operators are defined as:

$$b(A) \iff \partial A \tag{7}$$

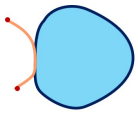
$$from(L), to(L) \iff \partial L \tag{8}$$

The OGC named topological relationships (*Equals*, *Disjoint*, *Intersects*, *Touches*, *Crosses*, *Within*, *Contains*, *Overlaps*) correspond to the CBM as follows:

$$\begin{aligned} Touches(A,B) &\iff touch(A,B) \\ Within(A,B) &\iff in(A,B) \\ Crosses(A,B) &\iff cross(A,B) \\ Overlaps(A,B) &\iff overlap(A,B) \\ Disjoint(A,B) &\iff disjoint(A,B) \\ Contains(A,B) &\iff in(B,A) \\ Equals(A,B) &\iff in(A,B) \wedge in(B,A) \\ Intersects(A,B) &\iff \neg disjoint(A,B) \end{aligned}$$

Let us make an example to illustrate how the CBM works. Consider the fifth case on the upper row of Figure 2. To describe such a topological configuration with a CBM expression, we say that the line is touching the region and that at the same time the line is overlapping the region's boundary and the line's endpoints are disjoint from the region. Compare the DE-9IM matrix and the CBM logical expression in Figure 4.

To bridge from the definitions into practice, it is useful to include some instances of sample code in OGC standard. In the following, the first program finds the US states bordering on New Mexico by using the *Touches* relationship (see graphical output in Figure 5a). The Python code uses the OGR GDAL library for the management of geographic data and the implementation of topological relationships (see https://gdal.org). The dataset is taken from Natural Earth (see https://www.naturalearthdata.com).



$$\begin{pmatrix} - & 1 & 1 \\ - & - & 0 \\ 2 & 1 & 2 \end{pmatrix}$$

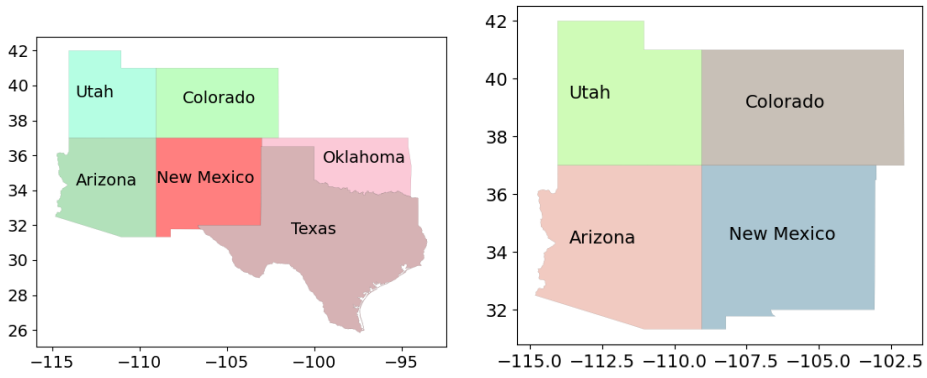
$$touch(L, A) \wedge$$
$$overlap\left(L, b(A)\right) \wedge$$
$$disjoint(from(L), A) \wedge$$
$$disjoint(to(L), A)$$

Figure 4: An example of correspondence between a DE-9IM matrix and a CBM expression.

```
from osgeo import ogr
def test_bordering(states,statename):
    driver=ogr.GetDriverByName("ESRI Shapefile")
    vector=driver.Open(states+'.shp',0)
    layer=vector.GetLayer(0)
    layer.SetAttributeFilter("name='{}'".format(statename))
    state = layer.GetNextFeature()
    state_geom = state.GetGeometryRef()
    layer.SetAttributeFilter("admin='United States of America'")
    bordering_states=[]
    for f in layer:
        geom = f.GetGeometryRef()
        if geom.Touches(state_geom):
            print(f.name)
            bordering_states.append(f)
    return bordering_states,state
if __name__=="__main__":
    stateshapes='data/ne_10m_admin_1_states_provinces'
    statename="New Mexico"
    bordering_states,state=test_bordering(stateshapes,statename)
```

The *Relate* operator implements the DE-9IM. To illustrate its use, let us find the US states that are bordering on a single point. The DE-9IM relationship that holds when two polygons touch in a point is evaluated by the *Relate* operator with the string "FF2F01212" (see graphical output in Figure 5b). Since the OGR GDAL library does not implement the *Relate* operator, it is necessary to include the 'shapely' library as well (see https://shapely.readthedocs.io/en/2.0.6/manual.html), as follows.

```
from osgeo import ogr
from shapely import wkt
def relate(geom1,geom2):
    st1=geom1.ExportToWkt()
    st2=geom2.ExportToWkt()
    obj1=wkt.loads(st1)
    obj2=wkt.loads(st2)
    pattern=obj1.relate(obj2)
```



(a) (b)

Figure 5: (a) US states bordering New Mexico, as an application of the *Touches* topological relationship; (b) States bordering on a single point, found out with the *Relate* value equal to "FF2F01212".

```
    return pattern
def test_relation(shpfname,relation):
    driver=ogr.GetDriverByName("ESRI Shapefile")
    vector=driver.Open(shpfname+'.shp',0)
    layer=vector.GetLayer(0)
    layer.SetAttributeFilter("admin='United States of America'")
    features=[]  #array for features
    for f in layer: #we bring the features to main memory
        features.append(f)
    statelist = []
    for f in features:
        geom=f.GetGeometryRef()
        for f1 in features:
            geom1=f1.GetGeometryRef()
            dem=relate(geom,geom1)
            if dem==relation:
                print(dem,f.name,f1.name)
                statelist.append(f)
    return statelist #it returns states satisfying the relation
if __name__=="__main__":
    shpfname='data/ne_10m_admin_1_states_provinces'
    relation="FF2F01212" #between two polygons touching in a point
    statelist=test_relation(shpfname,relation)
```

A third sample code is about topological relationships between polygons and polylines. We show the use of the *Crosses* relationship to find out the US states crossed by the river Mississippi. In the Natural Earth dataset, as it is quite common with linear shapes, the river Mississippi is represented by more than a single feature. Therefore, as a first step it is necessary to make the geometric

union of all the features with name Mississippi to obtain a single MultiPolyline representing the river. After that, with the the *Crosses* relationship, we can check the states that are crossed by the river. See the following program and the graphical output in Figure 6.

```
from osgeo import ogr
def test_crossing(states,rivers):
    driver=ogr.GetDriverByName("ESRI Shapefile")
    vectorstates=driver.Open(states+'.shp',0)
    layerstates=vectorstates.GetLayer(0)
    layerstates.SetAttributeFilter("admin='United States of America'")
    vectorrivers=driver.Open(rivers+'.shp',0)
    layerrivers=vectorrivers.GetLayer(0)
    layerrivers.SetAttributeFilter("name='Mississippi'")
    n = layerrivers.GetFeatureCount()
    #n number of features named "Mississippi"
    feat0 = layerrivers.GetNextFeature()
    mississippi = feat0.GetGeometryRef()
    for i in range(n - 1): #it makes the union of features
        feat1 = layerrivers.GetNextFeature()
        geom1 = feat1.GetGeometryRef()
        mississippi = mississippi.Union(geom1)
    crossing_states=[]
    for f in layerstates:
        geom = f.GetGeometryRef()
        if mississippi.Crosses(geom):
            crossing_states.append(f)
            print(f.name)
    return crossing_states,mississippi
if __name__=="__main__":
    stateshapes='data/ne_10m_admin_1_states_provinces'
    rivershapes='data/ne_10m_rivers_lake_centerlines'
    crossing_states,mississippi=test_crossing(stateshapes,rivershapes)
```

The study of topological relationships is a fascinating research topic that has produced many different results, broken down into different application fields. Despite efforts, the potential of the topic has not been fully exploited yet, especially regarding the extensions to 3D space and to objects with uncertain boundaries.

References

Clementini, E. and Di Felice, P. (1995). A comparison of methods for representing topological relationships. *Information Sciences*, 3(3):149–178.

Clementini, E. and Di Felice, P. (1996). A model for representing topological relationships between complex geometric features in spatial databases. *Information Sciences*, 90(1-4):121–136.

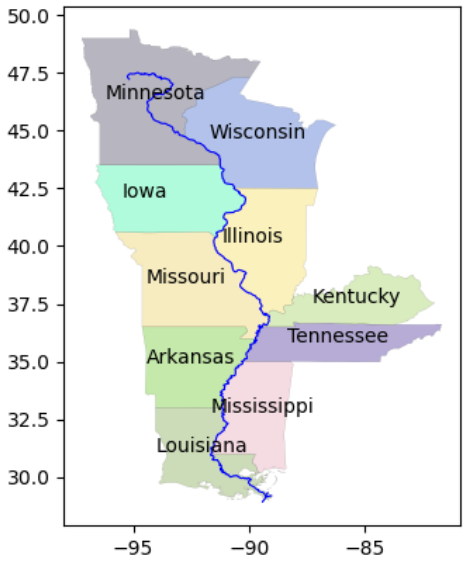


Figure 6: An application of the *Crosses* relationship to find out the US states crossed by the river Mississippi.

Clementini, E., Di Felice, P., and van Oosterom, P. (1993). A small set of formal topological relationships suitable for end-user interaction. In Abel, D. and Ooi, B. C., editors, *Advances in Spatial Databases - Third International Symposium, SSD '93*, volume 692 of *LNCS*, pages 277–295, Berlin. Springer-Verlag.

Cui, Z., Cohn, A. G., and Randell, D. A. (1993). Qualitative and topological relationships in spatial databases. In Abel, D. J. and Ooi, B. C., editors, *Advances in Spatial Databases - Third International Symposium, SSD '93*, volume 692 of *LNCS*, pages 296–315, Berlin. Springer-Verlag.

Egenhofer, M. J., Clementini, E., and Di Felice, P. (1994). Topological relations between regions with holes. *Int. J. Geogr. Inf. Syst.*, 8(2):129–142.

Egenhofer, M. J. and Franzosa, R. D. (1991). Point-set topological spatial relations. *Int. J. Geogr. Inf. Syst.*, 5(2):161–174.

Egenhofer, M. J. and Herring, J. R. (1990). Categorizing binary topological relationships between regions, lines, and points in geographic databases. Report Technical Report, Department of Surveying Engineering, University of Maine.

Munkres, J. (2014). *Topology, Second Edition*. Pearson Education Limited, Edinburgh Gate, Harlow.

OGC Open Geospatial Consortium Inc. (1999). Opengis simple features implementation specification for sql. Report.

Learning Objectives

- Define the most widespread models for describing topological relationships.
- Identify the correct topological relationship in a geometric scene of two spatial objects.
- Describe a topological relationship in either the DE-9IM or the CBM models.