Fall 12-20-2019

A Qualitative Representation of Spatial Scenes in R2 with Regions and Lines

Joshua Lewis
University of Maine, joshua.a.lewis@maine.edu

A QUALITATIVE REPRESENTATION OF SPATIAL SCENES IN $\mathbb{R}^2$

WITH REGIONS AND LINES

By

Joshua A. Lewis

B.A. The University of Maine, 2009

M.S. The University of Maine, 2012

A Dissertation

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

(in Spatial Information Science and Engineering)

The Graduate School

The University of Maine

December 2019

Advisory Committee:

Max J. Egenhofer, Professor in Spatial Information Science and Engineering, Advisor

Kate Beard-Tisdale, Professor in Spatial Information Science and Engineering

Robert D. Franzosa, Professor in Mathematics and Statistics

Torsten Hahmann, Associate Professor in Spatial Information Science and Engineering

John R. Herring, System Architect, Spatial Products, Oracle USA, Nashua, New Hampshire

A QUALITATIVE REPRESENTATION OF SPATIAL SCENES IN $\mathbb{R}^2$

WITH REGIONS AND LINES

By Joshua A. Lewis

Thesis Advisor: Dr. Max J. Egenhofer

An Abstract of the Dissertation Presented
in Partial Fulfillment of the Requirements for the Degree
of Doctor of Philosophy
(in Spatial Information Science and Engineering)
December 2019

Regions and lines are common geographic abstractions for geographic objects. Collections of regions, lines, and other representations of spatial objects, along with their relations, form a *spatial scene*. For instance, the states of Maine and New Hampshire can be represented by a pair of regions and related based on their topological properties. These two states are adjacent (i.e., they *meet* along their shared boundary), whereas Maine and Florida are not adjacent (i.e., they are *disjoint*).

A detailed model for qualitatively describing spatial scenes should capture the essential properties of a configuration such that a description of the represented objects and their relations can be generated. Such a description should then be able to reproduce a scene in a way that preserves all topological relationships, but without regards to metric details.

Coarse approaches to qualitative spatial reasoning may underspecify certain relations. For example, if two objects *meet*, it is unclear if they *meet* along an edge, at a single point, or multiple times along their boundaries. Where the boundaries of spatial objects converge, this is called a *spatial intersection*. This thesis develops a model for *spatial scene descriptions* primarily through sequences of detailed *spatial intersections* and object *containment*, capturing how complex spatial objects relate.

With a theory of complex spatial scenes developed, a tool that will automatically generate a formal description of a spatial scene is prototyped, enabling the described objects to be analyzed. The strengths and weaknesses of the provided model will be discussed relative to other models of spatial scene description, along with further refinements.

ACKNOWLEDGMENTS

In recognition of everyone who has supported me during the writing of this thesis and those who have supported my research directly: thank you. I wish to thank my advisor Dr. Max Egenhofer in particular for all time and energy he has invested in guiding my research and the countless ideas and contributions he has shared along the way. I cannot imagine being at this point today without his expert guidance and willingness to help. I would also like to thank the other members of my thesis committee, Dr. Kate Beard-Tisdale, Dr. Robert Franzosa, Dr. Torsten Hahmann, and Dr. John Herring, for the feedback they provided on my work, helping me to evolve my ideas from an initial conference publication into a fully developed thesis.

I would also like to thank the School of Computing and Information Science in general, but I would be remiss if I did not thank several people specifically. I wish to thank my adviser during my MSIS degree, Dr. Harlan Onsrud, for supporting my earliest graduate development. I also wish to thank Carol Roberts for her continued support of my teaching efforts through encouragement, advice, and materials. I also owe my gratitude to Dr. Matthew Dube who was always willing to provide input and collaboration during the initial research that would eventually lead me to this thesis years later.

Finally, I would like to acknowledge my friends and family for their continued understanding and support during this process. To Debra, Dwight, Marcia, and Jonathan: you have my boundless thanks.

This dissertation research was partially supported by NSF grants IIS-1016740 and IIS-1527504. I am also grateful for Teaching Assistantships in the School of Computing and Information Science, which funded me during several semesters.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

CHAPTER 1

INTRODUCTION

Physical reality, the domain within which people live, is what one typically aims to capture in a geographic information system. Yet, this physical reality is not the only perspective people have about their surroundings. The real world serves as the basis from which people construct generalizations, which yield a hierarchy of spatial representation—from sensory observations of the physical world to people's perceptions of reality based on their own deductions and decision making (Frank 2001). These abstractions include the worlds of countries and borders, sketches and maps, pictorial or other verbal and written descriptions.

Sitting in the middle of this hierarchy are the objects people form through observations. Certain objects are naturally segmented from the rest of reality based on their physical cohesion, such as a rock or a car. Other objects, however, may nonetheless maintain some structure, such as a cloud or a stream, even if they lack a similar solidity. Collectively, these are *bona fide* objects, which stand in contrast to *fiat* objects, such as countries or individual land parcels—these are given form by human agency when their boundaries are constructed (Smith 1995).

People not only observe the world and its myriad objects; they also communicate and think about reality and their experiences of it. This communication could take the form of a depiction—maps and sketches may be used *in lieu* of shared experience, allowing one person to visualize what another has seen. Such depictions, particularly maps, rely heavily on quantitative (metric) information (e.g., the distance between objects or dependence on a coordinate system), or a reliance on the shape, size, or location of the objects depicted in relation to one another within the space.

Complementary approaches, however, focus on the qualitative properties of objects and space: distilling continuous phenomena into relevant exemplars—in particular those properties that speak to whichever questions are being asked and nothing more (Cohn *et al.* 2001). To illustrate a separation of salient properties from irrelevant details (which depends on context), consider the map of a subway system, which faithfully depicts which lines connect to each station, abstracting away the distances between locations and the specific curvatures of the tracks (Avelar and Hurni 2000; Hahmann 2013). The subway map exists to answer the question, “how do I get from here to there?” The distance traveled and the shape of the route are less important to a traveler who is taking the train: therefore, connectivity is valued more highly than the distances, shapes, and directions represented in a depiction that more closely matches all aspects of reality. Representations such as these that do not rely on metric details are called *qualitative,* modeling variables based on a small set of values, rather than utilizing the full range of real values (De Kleer and Brown 1984; Egenhofer and Mark 1995a).

Often, a qualitative representation is preferable due to the improbability of a person being fully aware of every metric detail: a qualitative representation enables reasoning despite incomplete information (Sharma *et al.* 1994). To that end, the quantitative tends toward interval and ratio measures, while the qualitative tends toward nominal and ordinal values, to use the classification of Stevens (1946). A person may relate the sequence in which landmarks are encountered along a road, perhaps describing the extent between each as *near* or *far*, but the exact kilometer distance from a point of interest would likely not be provided, especially if more than one road must be traversed.

Traditionally, formal qualitative representations of space have focused on the relations between pairs of objects (Chen *et al.* 2015; Cohn *et al.* 2001; Cohn *et al.* 2008; Galton 2009); that two objects *meet* or one *contains* the other, for instance. A *spatial scene*, on the other hand, as an abstract, non-graphical representation of a space, comprises myriad objects and their spatial relations (Bruns and Egenhofer 1996). Such a representation allows more complex relations to be identified, for instance, when an ensemble of objects *surrounds* another object (Lewis *et al.* 2013). This thesis introduces the *Scene Notation*, a formal model for comprehensively describing spatial scenes, consisting of an arbitrary number of lines and regions—abstractions of objects that could be part of some real-word observation.

While regions often act as stand-ins for real-world objects, lines are also often abstractions of the objects that they represent (Lewis and Egenhofer 2014). The measurable width of the road may be abstracted away due to a larger potential focus on what a road connects to or is near rather than what it *overlaps*, for instance. Each type of object provides a different representation of real-world entities, and both regions and lines may appear together in the same depiction (Mackworth 1977) (e.g., a lake represented as a region draining into a river represented as a line), therefore they are both modeled by the *Scene Notation*.

1.1 Scene Representation

Regardless of how a spatial scene is modeled, the description of a scene should have a correspondence—a mapping—between itself and the scene it purports to describe (Figure 1.1). Modeling a spatial scene requires several considerations, including what particular spatial features to include and whether a scene should be represented qualitatively with specific spatial properties, or with the inclusion of metric refinements. Additional considerations central to describing the utility and limitations for a theory of spatial scenes include the choice of an embedding space and the types of objects supported. Different spatial relations can be prioritized for a given problem. Once the context of the problem is established, questions, such as "what is inside of the object?" or "what is adjacent?" may take a central role or be discarded altogether.

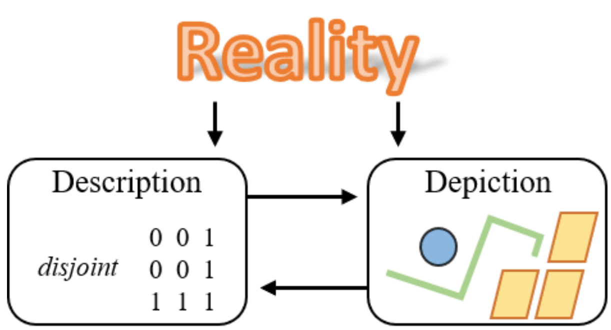


Figure 1.1 The interrelation between a qualitative scene description and a depiction of a scene, both based on some geographic reality.

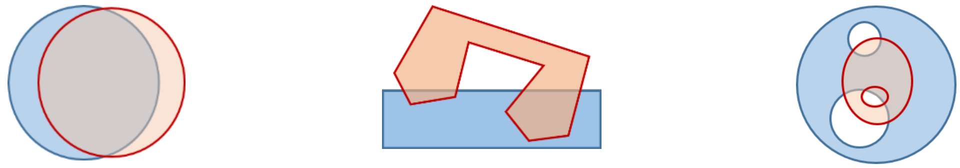
1.1.1 Coarse Models of Binary Topological Relations

While there are myriad qualitative spatial properties, topological properties are preserved under various continuous deformations, such as stretching and twisting—angles and distances are not. Since certain questions of place (e.g. 'is this object inside of another') do not require every metric property to resolve (such as angle or distance in this particular example), a topological approach is desirable (Adams and Franzosa 2008). Other nontopological relations such as orientation (Kurata and Egenhofer 2007; Moratz *et al.* 2011; Lewis *et al*. 2014), shape (Barkowsky *et al*. 2000; Brauer *et al.* 2001), direction (Peuquet and Zhan 1987; Papadias and Sellis 1994; Frank 1995; Goyal 2000), and proximity (Clementini *et al.* 1997; Worboys *et al.* 2004; Moratz and Ragni 2008) might also be necessary when answering certain questions, but are not considered here. Two foundational models for representing the topological relations between spatial regions—the 4-intersection (Egenhofer and Franzosa 1991a) and RCC-8 (Randall *et al.* 1992)—each produce a set of eight binary topological relations in $\mathbb{R}^2$ (Figure 1.2). The 4-intersection utilizes a 2x2 matrix, capturing the interplay between two objects' interiors and boundaries, recorded as empty or non-empty intersections for each cell of the matrix.

(a) (b) (c) (d) (e) (f) (g) (h)

Figure 1.2 The eight region-region relations in $\mathbb{R}^2$, described by the 4-intersection. These include (a) *disjoint*, (b) *meet*, (c) *overlap*, (d) *equal*, (e) *inside*, (f) *coveredBy*, (g) *contains*, and (h) *covers* (Egenhofer and Franzosa 1991a).

Alternatively, the topological relations developed by RCC-8 are based on connectivity, not intersection, but achieve the same general result for simple regions embedded in $\mathbb{R}^2$, yet RCC-8 allows for more complex regions, such as those with holes or separations (Cohn *et al.* 1997).



(a) (b) (c)

Figure 1.3 Three configurations that map onto the same binary relation using the 4-intersection or RCC-8. (a) An *overlap* between two simple regions, (b) two regions *overlapping* to form a gap, and (c) two holed regions that *overlap*.

These coarse qualitative models alone, however, may be insufficient to handle the complexities that may be present within a scene (Lewis *et al.* 2013), such as separations of the exterior (Figure 1.3).

1.1.2 Detailed Models of Binary Topological Relations

While the 4-intersection and RCC-8 are most commonly used to represent the relations between simple regions, these approaches have been extended, enabling more detailed relations to be modeled. The 9-intersection, a modification of the 4-intersections which also incorporates the objects' exteriors (Egenhofer and Herring 1991b), represents the topological relations between both regions and lines. Through the 9-intersection, relations between complex objects are also described, including relations between regions and lines with disconnected interiors (Schneider and Behr 2006), and holed regions (Egenhofer and Vasardani 2007; Vasardani and Egenhofer 2009, Dube *et al.* 2015).

The 9^+-intersection (Kurata and Egenhofer 2007; Kurata 2008a) allows the interior, boundary, and exterior components of the 9-intersection to be split, enabling more refined objects to be modeled, such as those with separations of interiors, boundaries, or exteriors (e.g., separated regions with several disconnected interiors and boundaries, holed regions with disconnected exteriors and boundaries, or directed lines), thereby capturing more details than the coarse models. The Dimensionally Extended 9-intersection Model (DE-9IM) extends the 9-intersection by capturing the dimension of the intersections (Clementini *et al*. 1993). Additional models have also described dimension using the 9-intersection (McKenny et al. 2005) and the 4-intersectrion (Egenhofer 1993). The compound object model (Egenhofer 2009) allows for the construction of arbitrarily complex objects and yields their topological relations, for instance for regions with cuts via set difference of basic objects, as well as separations and regions with spikes through the union of basic objects. Other approaches focus on particular domains of relations, such as various types of *overlap* (Galton 1998) and *surrounds* (Dube and Egenhofer 2014).

While these detailed models may capture essential properties of a spatial scene such that a topologically correct depiction can be reconstructed from the symbolic qualitative representation between pairs of objects, they alone are insufficient when modeling the interplay of multiple objects, such that only a single depiction can be generated for each scene. If a collection of three regions share the *overlap* and *meet* relations, for example, any number of interpretations may arise from such a coarse description (Figure 1.4).

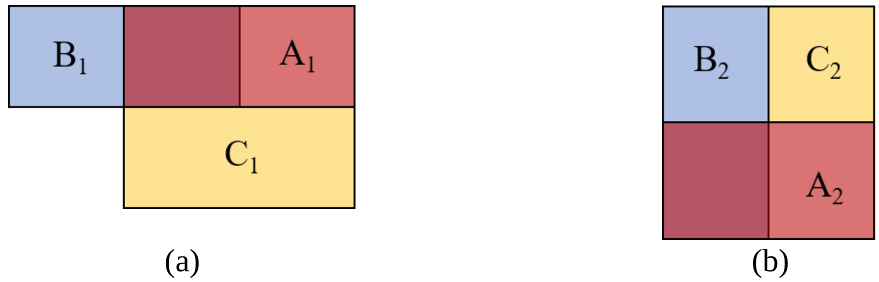


Figure 1.4 Two distinct configurations that map onto the same binary relations using 4-intersection or RCC-8. First, (a) A and B *overlap*, C *meet* A and B, and (b) A and B *overlap*, C *meet* A and B.

Extending the models of binary relations further to address each configuration of n>2 objects is infeasible, as they would amount to an infinity of relations between all pairs of objects. Alternatively, logic-based theories such as RCC (Randall *et al.* 1992) model the interplay between all objects within a scene. Of these there exists a set of approaches that have the additional property of being able to relate objects independent of their dimension—multidimensional mereotopologies—which stand in contrast to approaches that apply only to objects of specific dimensions (Gotts 1996; Galton 2004; Hahmann 2013; Hahmann and Gruinger 2011a).

The theory presented by Galton (2004) is defined for regions of varying dimensions, but has the consequence that lower-dimensional regions form the extent of higher dimensional regions; if the regions of dimension n are the regular open sets in $\mathbb{R}^n$, the regions of dimension n-1 are the regular open subsets of their boundaries.

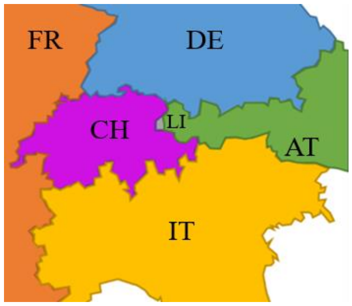
The INCH calculus (Gotts 1996), another multidimensional approach, is based on the predicate INCH(x, y), is interpreted as '*x* **IN**cludes a **CH**unk of *y*'. This is proposed as an alternative to the relation C(x, y) of Clark's connection calculus (Clarke 1981) and RCC (Randall *et al.* 1992)(Chapter 2.3.1).

A third approach, CODI (Hahmann 2013), captures detailed properties, such as betweenness, containment, dimension, and whether objects are comprised on a single component or contain additional pieces such as holes.

1.1.3 Concerning Intersection

To elaborate on the shortcomings of existing intersection-based models, consider a collection of European countries in an abstract, map-like configuration (Figure 1.5a). The topological relations between each pair of these countries modeled as simple regions can be represented through an application of the 4-intersection, listed in tabular form (Figure 1.5b). The original scene is also reproducible from the relations as listed in the table, but with complications.

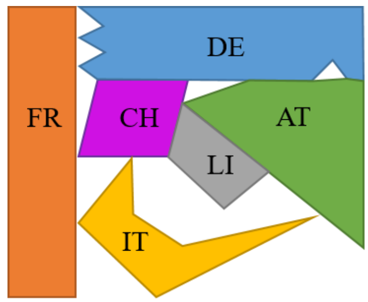
It is also conceivable to create several additional configurations from the set of valid relations in Figure 1.5b that do not match the scene that is being modeled (Figures 1.5c). Holes are absent in the original scene; however, their inclusion does invalidate the coarse spatial relations represented in the table of binary relations (Figure 1.5b).



(a)

	CH	FR	DE	AT	IT	LI
CH	*eq*	*m*	*m*	*m*	*m*	*m*
FR	*m*	*eq*	*m*	*d*	*m*	*d*
DE	*m*	*m*	*eq*	*m*	*d*	*d*
AT	*m*	*d*	*m*	*eq*	*m*	*m*
IT	*m*	*m*	*d*	*m*	*eq*	*d*
LI	*m*	*d*	*d*	*m*	*d*	*eq*

(b)

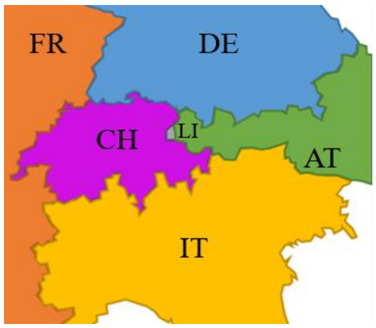


(c)

Figure 1.5 A selection of European countries. Initially (a) as they appear on a map, and (b) their binary relations under the 4-intersection (*eq* = *equal*, *m* = *meet*, *d* = *disjoint*). A third configuration (c) shows these same relations can be used to create a depiction that is altogether different from the original configuration with the inclusion of additional holes between regions, for instance.

At least as drawn, the example map of France shares a single edge with Italy, a single edge with Switzerland, and a single edge with Germany (Figure 1.5a). When the intersection with Italy ends, at that point, the intersection with Switzerland begins. When the intersection with Switzerland ends, the intersection with Germany begins. There are no gaps, and France and Germany do not *meet* at three different points with unincorporated territory between them, as indicated by the interpretation (Figure 1.5c).

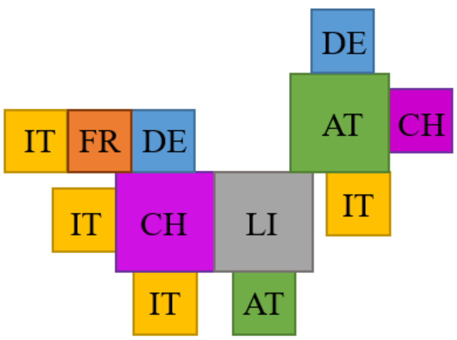
Similarly, representing complex spatial scenes with RCC-8 may be ambiguous. The holes introduced in the 4-intersection example between spatial regions (Figure 1.5c), are not a unique product of the 4-intersection. RCC-8 allows holes within regions (as well as separations), so a new set of possible interpretations of the scene appear (Figures 1.6c-d) from the original scene and its RCC-8 description (Figures 1.6a-b).



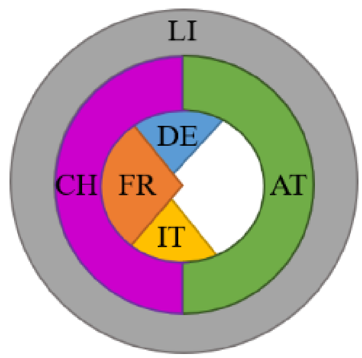
(a)

	CH	FR	DE	AT	IT	LI
CH	*eq*	*ec*	*ec*	*ec*	*ec*	*ec*
FR	*ec*	*eq*	*ec*	*dc*	*ec*	*dc*
DE	*ec*	*ec*	*eq*	*ec*	*dc*	*dc*
AT	*ec*	*dc*	*ec*	*eq*	*ec*	*ec*
IT	*ec*	*ec*	*dc*	*ec*	*eq*	*dc*
LI	*ec*	*dc*	*dc*	*ec*	*dc*	*eq*

(b)



(c)



(d)

Figure 1.6 The selection of European countries again. Initially (a) as they appear on a map, and (b) their relations under RCC-8 (*EQ* = *equal*, *EC* = *externally connected*, *DC* = *disconnected*). These same relations can be used to create depictions that are altogether different from the original configuration with additional separations (c) or additional holes between objects (d), for instance.

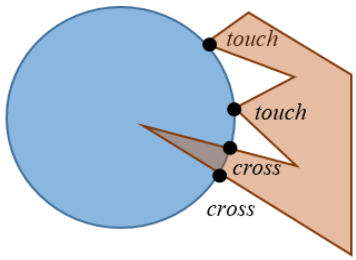
While the coarse models and their extensions are sufficient to reason about pairs of objects, ambiguities may arise when modeling scenes with more than two objects. Valid coarse descriptions of a spatial scene may result in the creation of holes where none exist, or misrepresent the sequence in which objects intersect.

1.1.4 Concerning Intersection Sequence

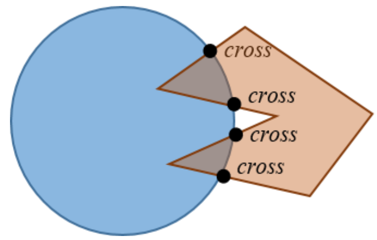
When several objects share a boundary point or an edge, the order of their intersections may be preserved to limit this ambiguity. For complex configurations of objects, modeling the sequence of intersections serves to limit the creation of potentially ambiguous constructions. For binary relations, types and sequences of boundary-boundary intersections have been addressed (Egenhofer and Franzosa 1995a), but these aspects have not been fully explored to capture the potential complexities of scenes with arbitrary numbers of complexly structured spatial objects.

While such constraints for line-like boundaries have been applied to line-line relations for complex scenes comprised of line segments (Clementini and Felice 1998), the approach is not immediately extensible to the boundaries of areal objects in a manner that allows specific region-region relations to be derived. Further refinements to coarse relations might involve recording the sequence of intersections between regions or lines, whether each intersection forms a *crossing* or a *touching* configuration (Herring 1991) or the dimension of each intersection and the relation between the objects' complements (indicating whether the exterior is partitioned, for instance) (Egenhofer 1993, Egenhofer and Franzosa 1995a).

A scene of overlapping regions with four intersections described by a *touch-touch-cross-cross* sequence (Herring 1991) (Figure 1.7a), for example, is distinct from a scene described by a *cross-cross-cross-cross* sequence (Figure 1.7b). Without capturing this sequence, the relations between numerous pairs of objects might be described coarsely as *overlap*, despite having distinct sequences of *crossing* and *touching*.



(a)



(b)

Figure 1.7 Two simple scenes with different forms of *overlap.* First (a) a *touch-touch-cross-cross* sequence and then (b) a *cross-cross-cross-cross* sequence (Herring 1991).

Thus, models that provide additional detail, such as the dimension and sequence of fine-grained spatial relations like *cross* and *touch* (Herring 1991), can describe a spatial scene with more specificity than a coarser model like the 4-intersection.

Specifying the dimension of an intersection reduces inaccuracy and ambiguity in the scene representation. For example, the states of Utah and New Mexico share a 0-dimensional boundary intersection, while Utah shares a 1-dimensional boundary intersection with Nevada (Figure 1.8a). Without such distinctions, it would be impossible to construct an accurate depiction of these states from an underspecified scene description.

The benefit of maintaining a sequence of boundary intersections also applies to a familiar example involving the United States: the correct representation of the *Four-Corners* border feature, where the boundaries of Utah, Colorado, New Mexico, and Arizona intersect at a single point. Such a sequence is recorded in counter-clockwise order around the point (Herring 1991) (Figure 1.8b). This ordering is circular: the start and end point of the sequence do not matter, but the sequence itself must be maintained. Without an associated ordering, additional scene specifications will still result in ambiguity.

(a) (b)

Figure 1.8 The US state of Utah and its boundary intersections. The (a) 0-dimensional intersection with New Mexico and a 1-dimensional intersection with Nevada, and (b) a single boundary point shared with exactly three other US states, captured through a specific sequence.

The Four-Corners feature includes four intersecting edges along state boundaries. Each edge is shared exclusively by two adjacent states, while a 0-dimensional intersection is shared by all four states. Each intersecting edge also includes the 0-dimensional intersection as an endpoint. By representing the sequence in which the edges are oriented around the 0-dimensional intersection the states are properly oriented around that common point as well.

Each state also has a boundary segment that does not intersect with the other parts of the feature. The boundaries of each depicted state are describable through a unique sequence of three edges: the unshared edge and two shared edges. No other state but Colorado shares both an edge between Colorado and Utah and an edge between Colorado and New Mexico, for example. This boundary sequence (the sequence of edges that form the boundary), along with the sequence of edges around each 0-dimentionsal intersection, serves to distinguish a detailed scene description from a coarse representation of the relations between objects.

A detailed representation of an object's boundary also enables scenes consisting of lines to be represented more completely. As a line's extent contains both its boundary and interior, boundary sequence enables containment relations to be more fully modeled (Figure 1.9a and 1.9b).

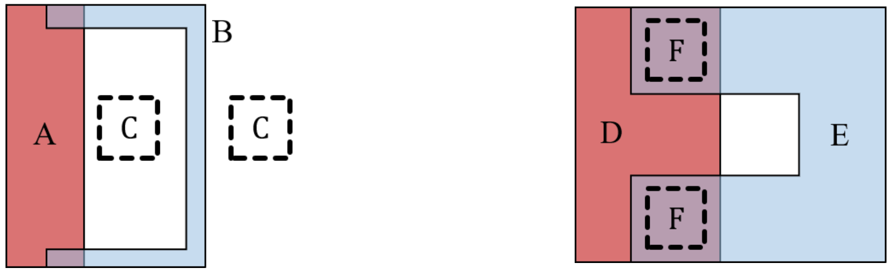
While interval relations (Allen 1983) and line-line relations (Egenhofer and Herring 1991b) may be modeled through intersection sequences for lines (Figure 1.9), containment between regions requires a different approach. Related to intersection sequence, however, especially between connected linear features, is the alternative notion of *betweenness* (Hahmann and Gruinger 2011a). The betweenness relation Btw(r, a, b, c) is defined such that an object b is between objects a and c, all embedded in a space r, only if every object connecting a and c intersects b. This notion of betweenness exists independently of cardinal direction or other properties specific to a reference object.

Figure 1.9 Two scenes built from multiple lines. (a) Line A contains line B and line C, in that sequence, and (b) line D contains line E, which itself contains line F.

Regardless of whether containment is captured between objects, capturing the sequence in which intersections occur for a line and the sequence in which boundary intersections occur for a region enables a detailed representation that is impossible with coarse models such as the 4-intersection and RCC-8. Even with two regions, for example, detailed boundary sequence information enables an unambiguous representation of how objects intersect (Figure 1.7). Reasoning about the sequence in which objects are arranged around an intersection point allows the representation to be further refined when there are more than two objects intersecting (Figure 1.8b).

1.1.5 Concerning Containment

Modeling topological spatial relations with intersections (or sequences of intersections) provides information for common spatial reasoning tasks, up to a point. For instance, when the exterior of a scene embedded in the plane, $\mathbb{R}^2$, is divided into multiple components, potentially significant problems begin to appear. As an example, a model may be insufficient to model the exact placement of a region within a split exterior (Figure 1.10a). Similar issues arise when placing an object within the separated interior of a region (Figure 1.10b).



(a) (b)

Figure 1.10 Two spatial scenes with potentially ambiguous constructions. (a) An object C sits in the exterior which is disconnected, and (b) an object F sits in the separated intersection of two objects.

An alternative version of this problem arises in the plane $\mathbb{R}^2$, when the exterior is separated by an ensemble of regions joined through *overlap* or *meet*—do the objects surround an additional object (Figure 1.11a) or is that object external to them (Figure 1.11b)? Designed to solve similar problems, the *o-notation* (Lewis *et al.* 2013) and *i-notation* (Lewis and Egenhofer 2014) can model an arbitrary number of regions, regions with holes and separations, and situations where an ensemble of regions comes together to *surround* other regions, using an operator known as the *topological hull* in order to identify separations of the exterior, including holes, enabling both the disk-like region and the exterior partition to be reasoned with independently.

Figure 1.11 Two scenes built from the union of many regions. (a) A region is *surrounded* by an ensemble and (b) a region is outside of an ensemble.

These two models, however, cannot fully represent the boundary intersection sequence for objects that all *meet* at a single point. One problem—detailed containment—is partially solved, while another problem—intersection sequence—is reopened. The *o-notation* (Lewis *et al.* 2013) and *i-notation* (Lewis and Egenhofer 2014) also do not handle scenes where lines are modeled.

While the containment relations between pairs of objects is well addressed by the coarse models, they do not uniquely describe cases where the interior of an object is partitioned or cases where the exterior is partitioned. These problems require a robust model for describing a spatial scene that also includes detailed containment relations between objects—identifying the specific partition of space in which an object is contained. The detailed models presented do enable such a representation, but also lack additional descriptive power.

1.1.6 Balancing Simplicity and Detail

To varying degrees of specificity, the models represented thus far have attempted to represent the complexity of spatial scenes between two objects or sometimes an arbitrary number of objects by capturing generally distinct sets of spatial properties.

As a first step in devising a more detailed representation of spatial scenes, it is necessary to expand and develop the relations between objects—simply adding additional regions into a scene is not always enough. For instance, modeling the relation *overlap* through an enumeration of connected components (under union and set difference) to represent the relation, along with additional complexities, yields more detail than can be accommodated through a coarse relation. This approach allows the number of partitions the exterior is divided into to be captured in addition to describing the relation simply as *overlap* (Galton 1998).

Each of the models described thus far captures a set of properties for two or more spatial objects, enabling those objects to be reasoned about (Table 1.1). These properties have so far been shown to be insufficient to describe a spatial scene up to homeomorphism in Sections 1.1.3-1.1.5. A detailed spatial scene representation based on existing intersection-based models should overcome their individually limited expressivity. Many theories, including those based on logical approaches (Cohn *et al.* 1992; Cohn *et al.* 1997; Cohn *et al.* 1997; Gotts 1996; Galton 2004; Hahmann 2013; Hahmann and Gruninger 2011) also capture many of the properties discussed this far, such as dimension, complex containment, sequence, and betweenness (Hahmann 2013), but not necessarily togeather.

Developing a new approach that produces a single description of a spatial scene, rather than producing numerous ambiguous descriptions of the same objects, is the objective of this thesis. Such an approach will still produce a more abstract representation than representing the geometry of a scene explicitly but should also be more expressive than the coarse models, sitting between the two extremes within the spectrum of representation.

Table 1.1 Various intersection-based models and their domains. Includes (points, lines, regions, volumes), the number of objects, and the properties captured. Models supporting more than two objects are highlighted.

Name / Source	Relations	# Obj.	Dimension	Sequence	Detailed Containment	Homeomorphism
(Egenhifer 1989)	Simple: R	2	No			No
4-Intersection (Egenhofer and Franzosa 1991a)	Simple: R	2				No
9-Intersection (Egenhofer and Herring 1991b)	Simple: PLRV	2				No
DE-9IM (Clementini *et al*. 1993)	Simple: PLR	2				No
(Egenhofer 1993)	Simple: PLR	2	Yes			No
(Egenhofer and Franzosa 1995)	Simple: R	2	Yes	Partial (boundary)		No
(Clementini and Felice 1998)	Simple: L	2	Yes	Partial (boundary)		No
(McKenney *et al.* 2005)	Complex: PLR	2	Yes			No
(Schneider and Behr 2006)	Complex: PLR	2				No
(Egenhofer and Vasardani 2007; 2009)	Complex: R	2			Partial (holes)	No
Spatial-Query-By-Sketch (Egenhofer 1997)	Simple: R	2+				No
Compound Object Model (Egenhofer 2009)	Complex: PLR	2+				No
o-Notation (Lewis *et al.* 2013)	Complex: R	2+	Yes	Partial (boundary)	Yes	No
(Dube and Egenhofer 2014)	Complex: R	2+			Partial (holes)	No

In the course of this thesis, a set of detailed region-region, region-line, line-region, and line-line relations are produced, as well as a bridge to connect them to the more familiar coarse relations (Figure 1.2). The opposite should also be true, representing a detailed scene as something less complex, and easier to understand (Figure 1.12).

Coarse Representation:

Generalized relations
Natural language

Detailed Representation:

Less ambiguity
Richer representation

Figure 1.12 A balance may be struck between coarse qualitative representations and detailed qualitative representations.

By improving on these detailed spatial scene representations, expanding on their utility, and accounting for their limitations, this thesis draws closer to a theory that can capture the topological detail of a spatial scene—including sequence of intersection, dimension of intersection, and complex containment relations—for any number of complex lines and regions in concert. This thesis aims to produce a new model of spatial scene representation, motivated by the *o-notation* (Lewis *et al.* 2013) and the *9-intersection* (Egenhofer and Herring 1991b).

1.2 Hypothesis

Producing a qualitative representation of an arbitrarily complex spatial scene consisting of regions and lines requires more than the descriptive power provided individually by all the coarse intersection-based theories detailed thus far. A less ambiguous representation may be possible through the development of a new set of detailed spatial relations, and a descriptive notation for capturing the details of a spatial scene. Such details include holes, separations, the dimension and sequence of intersections, and the integration of potentially many regions and lines. The components of such objects (their segmented interiors, boundaries, and exteriors) must also be uniquely identifiable. By accommodating both coarse and refined interpretations of space, a comprehensive model that expands on the benefits of either approach is formed, called *Scene Notation*, producing a result that is both strongly representative and scalable to scenes with complex compositions. Therefore, the hypothesis of this thesis is as follows:

> *When modeling an input scene [of lines and regions embedded in $\mathbb{R}^2$] by (1) decomposing the scene into a set of areas, edges, and nodes, and (2) recording the sequences of edges connecting each node and the area that contains each object, a detailed description of the scene is produced. The description enables three established topological invariants to be derived: (1) the dimension of the intersections between objects; (2) the containment relations between specific objects, holes and gaps; and (3) the relative ordering of intersecting objects around the boundary of a region and along the extent of a line. A detailed description requires all of these three properties in tandem—any omission may lead to ambiguity.*

Conjecture 1.1 The Scene Notation describes a scene uniquely, up to homeomorphism. Consequently, any two scenes produced from a given scene description are topologically identical.

An automated tool is developed that uses the *Scene Notation* model to reason about a spatial scene from such a formal description, from both a detailed structural perspective, as well as the more familiar descriptions of the coarse binary relations, establishing that the model is implementable.

1.3 Approach and Scope

This thesis is based on previous works, such as the *o-notation* (Lewis *et al.* 2013) and the *9-intersection* (Egenhofer and Herring 1991b). To develop a new model for representing spatial scenes, the basic elements of boundary intersection and containment are preserved from the *o-notation*, but a more detailed sequence of *touching* and *crossing* relations is developed. The *o-notation* accommodates the sequence of intersections along a given boundary but does not account for the sequence of objects positioned around a specific intersection (Herring 1991). This approach eschews metric refinements, direction, and points-as-objects, and it also forgoes modeling dimension directly, deriving that property instead from intersection sequences. In addition to dimension, the sequence of intersections is also derived, along with complex containment relations. These properties stem from a process of reducing the input scene into its cellular components—areas, edges, and nodes—and reasoning about those parts.

Accounting for this additional level of detail allows for the construction of 72 detailed relations between regions and lines at an intersection point. These relations are abstracted to their 9-intersection analogues, as well as used in sequence to detail the construction of a region's boundary or the extent of a line. This thesis focuses solely on line and region objects embedded in $\mathbb{R}^2$. Any application of the thesis to other embedding spaces, such as $\mathbb{R}^3$ or $\mathbb{S}^2$ may be the subject of future work.

1.4 Intended Audience

This thesis is intended for researchers concerned with qualitative spatial reasoning. It is of interest to those involved in modeling complex topological spatial relations. Due to the possibility of exchanging highly detailed representations for coarse representations of space, it may also be of interest to those studying human cognition, especially human-centric depictions of space. As the work also incorporates a means of automatically generating visual scenes from a notation and querying against that notation, it may also be of interest to GIS development.

1.5 Organization of Thesis

This thesis is organized into six chapters, including this introduction. The second chapter considers related work pertaining to the modeling of spatial scenes and compares the benefits and limitations of quantitative and qualitative descriptions of space. Topics, such as the construction of objects, the composition of spatial relations, similarity, and the application of spatial theories to problems, which are all elements of designing a theory of spatial scenes are considered in turn.

The third chapter introduces the basis for a spatial scene description. This foundation includes a discussion of the objects represented and their construction, as well as operations that may be performed on the objects or the entire scene. The properties modeled are shown to be necessary in order to faithfully describe a spatial scene uniquely.

The fourth chapter presents a set of spatial relations between combinations of lines and regions. These relations are mapped onto their 9-intersection analogues as well as a set of *surrounds* relations and are used to present detailed structural information about the boundary of a region or the extent of a line.

The fifth chapter introduces a computational solution for automatically generating formal descriptions of sketched scenes. The various methods used are discussed, and the use of the interface and its motivation are detailed, along with examples showing the sketching and analysis of objects.

The final chapter summarizes the thesis and lays out the conclusions developed in the previous chapters. The contributions of scene description are discussed, and conclusions are drawn regarding the satisfaction of the hypothesis. Opportunities for further development or benefits for future research are also be presented. This section considers situations involving embedding spaces other than the Euclidean plane, $\mathbb{R}^2$, as well as the inclusion of additional objects.

CHAPTER 2

REPRESENTING SPATIAL SCENES

The modeling of spatial scenes (Bruns and Egenhofer 1996) is a familiar topic in qualitative spatial reasoning. Whether capturing a geographic reality or some hypothetical spatial construction, certain elements and attributes are represented while others are discarded. The decision of how to model a scene has no singular solution, however the objects to be represented and the relations between those objects need to be formalized (Herring 1991). The models resulting in the most detailed depictions of space are generally those of a quantitative nature—models that capture such attributes as position, distance, and angle explicitly. Representations that do not rely on metric details are called *qualitative*, modeling variables based on a small set of values, rather than utilizing the full range of real values (De Kleer and Brown 1984; Egenhofer and Mark 1995a). Capturing qualitative properties does not produce representations that are as detailed, but often facilitate ease of communication and reasoning.

2.1 Modes of Reasoning

Specific applications benefit from quantitative modes of reasoning over qualitative reasoning, and vice versa. There are also models that employ aspects of both representations. Quantitative spatial reasoning is generally used when precise measures are required, such as calculating a viewshed using elevation data and a specific viewing angle where qualitative representations, such as A is *above* B would be significantly disadvantaged. Additional examples of quantitative measures being employed in conjunction with spatial scenes include various USGS datasets, which can have attributes, such as a timestamp, depth, and discharge rate for hydrology data (USGS 2016) or OpenStreetMap, which constructs such entities as roads through a network of geographic coordinates using the WGS 84 reference system (OpenStreetMap 2016a; OpenStreetMap 2016b).

Qualitative spatial reasoning, on the other hand, allows a scene to be described using a much more limited vocabulary of qualitative properties (Hernández 1994; De Kleer and Brown 1984; Cohn *et al.* 2001; Cohn and Renz 2008). The objects within a scene can then be related to one another using this restricted vocabulary, with the additional understanding that the precision provided by quantitative modes of reasoning may in fact be more difficult to reason with than an intuitive qualitative representation (Hernández 1991; Hernández 1994).

Qualitative models also have added flexibility—they do not require a complete representation of the geometric specifications of a scene (or other metric specifications) (Sharma *et al.* 1994). Qualitative theories can capture different properties, such as those dealing with dimension or orientation, distance, size, or mereo-topology, with many detailed surveys concerning the various facets of qualitative topological representation (Freksa 1993; Chen *et al.* 2015; Cohn *et al.* 2001; 2008; Galton 2009).

2.1.1 Cognitive Models

When considering how to represent a space within a system the framework of user experience and perception must be considered. Naive Geography, for instance, promotes the design of theories and GISs that align with human reasoning about space (Egenhofer and Mark 1995). Naive Geography is based in part on Naïve Physics (Hayes 1978; 1985), which is similarly concerned with the modeling of the physical word from a common-sense perspective, instead of focusing on smaller trivialities that do not add up to a greater whole.

Classifying space based on perception has led to several differing models. It has been theorized that there is a difference between spaces that can be manipulated and spaces that exist on a geographic scale, and that the interactive nature of a GIS has ramifications on such distinctions as they relate to how users have learned to interact with the world (Mark 1993; Mark and Freundschuh 1995; Montello 1993). Flat geographic representations, such as maps, for instance, can provide a wider awareness of a space than experiencing that space first-hand. One need only wander a maze on foot to experience this phenomenon—navigating the same space on paper with a pencil trivializes the experience.

Zubin (1989) additionally developed four types of space distinctions: *A-spaces*, which are objects that can be manipulated by hand; *B-spaces*, which are larger objects that cannot be entirely viewed from one single perspective, such as a vehicle; *C-spaces* are large scenes that can still be viewed from one vantage point, such as the vista from atop a building; and finally, *D-spaces* require some form of travel to fully conceptualize. In this manner the scale of a space directly impacts how the space is perceived, from the amount of detail available at once (having to move around an object) to the experience of perception (being able to manipulate an object directly versus seeing it pass outside of a window), for instance.

A typology for varying conceptualizations of space has been proposed by Freundschuh and Egenhofer (1997). Restricting the representation of spatial objects to a specific level of representation or abstraction may allow for more meaningful reasoning. Due to the increasing ability to collect and store information detailing a spatial scene and the imprecise nature of human reasoning over such entities it becomes necessary to consider how a detailed representation can be generalized into a specific model (Ruas and Lagrange 1995; Morehouse 1995).

Whether a user of a GIS requires fine detail or a coarse result, it is often desirable to support multiple representations (Bruegger and Kuhn 1991). Significantly, the same data can be used to generate multiple representations without affecting the underlying facts. Furthermore, in geographic space the use of a specific object type, such as a point, line, or region, over another may facilitate different levels of abstraction, such as depicting a town as a point or a region, or a road as a line segment or a region, with certain elements preserved or removed, depending on scale and interest (Timpf *et al*. 1992, Goyal and Egenhofer 2000) (Figure 2.1).

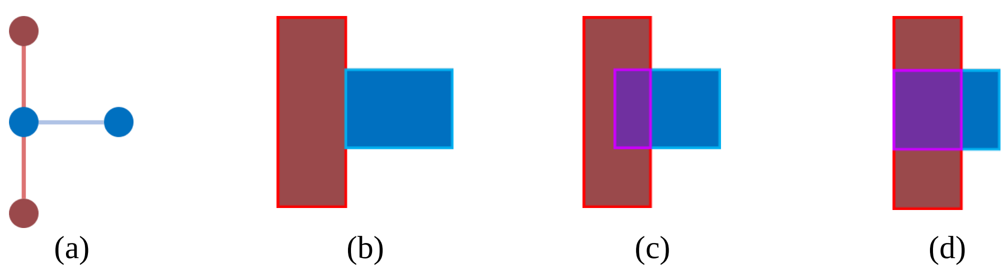


Figure 2.1 Various interpretations of a scene. (a) A line-line relation, and (b-d) three different versions (non-exhaustive) using regions to depict a similar relation.

Such abstractions may change based on what needs to be represented for a given purpose. Different views may necessitate interpreting a traditionally linear representation using regions, for example (Lewis *et al.* 2014) or a user may require a specific degree of abstraction, preserving points of interest (Barkowsky *et al.* 2000).

2.1.2 Spatial Language

A natural companion to how people think about spatial concepts is how people talk about spatial concepts. Unlike the set of limited symbols that make up a formal spatial model, natural language can lead to descriptions that are either under or over-specified, affecting the robustness of models that consider natural language (Hernández 1991; Bateman *et al.* 2010). Spatial language, however, typically is qualitative in nature and relies on similarities from the observed phenomena to a preexisting, prototypical understanding of various spatial relations (Haward and Tarr 1995). Spatial language is also relatable to qualitative properties—information that can then be used with metric refinements to more precisely identify spatial relations (Egenhofer and Shariff 1998).

The actual spatial language used in a description can be extracted from a natural language description if specific prepositions and other language elements are present within an appropriate context (Dahlgren 1988; Kordjamshidi *et al.* 2011). Spatial language can then be separated into triples consisting of reference objects, the object to be found, and the relations between them, such as *between* or *across*. The relations can then be used to connect the reference objects and the object to be found within a graph, providing the framework for reconstructing the spatial scene from a natural language description of space (Vasardani *et al.* 2013; Kim *et al.* 2016).

2.1.3 Sketching Scenes and Automation

How to represent spatial concepts visually, either by depicting them through sketch from a description **(**Vasardani *et al.* 2013; Kim *et al.* 2016**)** or querying a GIS through sketch (Egenhofer 1996), requires an understanding of cognition and an understanding of how people treat spatial language. The languages used in spatial queries are not as immediately familiar as people's everyday cognitive and visual perceptions of spatial relations (Egenhofer 1996), although work has been done to develop spatially aware query languages (Egenhofer 1994b; Calcinelli and Mainguenaud 1994; Di Loreto *et al.* 1996; Haarslev 1997). Taking the sketch approach, Wuersch (2003) developed a model that allows spatial features to be extracted from a digitized drawing where boundary lines are aggregated into areal objects. When interpreting sketches drawn by a user, distinction such as coarse or dashed lines may inform how the sketch should be interpreted (Mackworth 1977; Reiter and Mackworth 1989; Bertin 1983; Blaser 1998).

When representing a scene through a sketch, it has been shown that verbal descriptions are still necessary and provide additional information not conveyed by the drawing (Schlaisich and Egenhofer 2001). Spatial-query-by-sketch utilizes both sketch and additional attributes (Egenhofer 1996; Egenhofer 1997; Blaser and Egenhofer 2000). First a user depicts the desired spatial query as a sketch, using a touch-enabled screen, pad, or mobile device (Gross 1996; Caduff and Egenhofer 2005), then the user adds attributes to the sketch to provide specificity. These steps can be repeated as needed.

The interface the user is working with needs to be designed to aid in the depiction of the spatial elements along with their lexical counterparts (Egenhofer and Frank 1988). The sketch and attributes are then translated into a topological data model, ambiguities are resolved, and a query plan is made by the system. When these steps are completed the matching scenes are retrieved based on the spatial query.

2.2 Qualitative Spatial Relations

A common model for representing the relations between spatial objects, the Dimensionally Extended 9-intersection Model (DE-9IM) (Clementini *et al.* 1993), is a modification of the 9-intersection (which itself expands on the 4-intersection). Models such as DE-9IM and the 9-intersection are based on the intersection of objects' interiors, boundaries, and potentially exteriors, while certain other models, such as RCC-8 are based on connectivity (specifically between regions) (Randall *et al.* 1992). Both representations can be expanded to handle additional complexities, such as the addition of holes or to accommodate distinctions such as the dimension of spatial intersections.

2.2.1 Coarse Binary Relations

Deriving the topological relations between a pair of spatial objects based on intersection is the foundation of models such as the 4-intersection and the 9-intersection. In these models the content of intersections is recorded as either empty or non-empty. This property is topologically invariant. The resulting matrix for each relation defines a unique relation between two objects out of the set of eight under intersection. By considering the pairwise intersections between two objects' interiors, boundaries, and exteriors a set of base relations is generated (Figure 2.2, Equation 2.1). The basic framework for this approach is called the 9-intersection and expands upon the 4-intersection (Egenhofer and Franzosa 1991a), which omits the five exterior components.

$$R(A,B) = \begin{pmatrix} A^o \cap B^o & A^o \cap \partial B & A^o \cap B^- \\ \partial A \cap B^o & \partial A \cap \partial B & \partial A \cap B^- \\ A^- \cap B^o & A^- \cap \partial B & A^- \cap B^- \end{pmatrix} \tag{2.1}$$

Each intersection is recorded as either empty (∅ or 0) or nonempty (¬∅ or 1) based on the configuration of the objects being described. Though there are 512 (2^9) matrices of such binary values, only eight correspond to the base relations between two regions in $\mathbb{R}^2$. The DE-9IM model (Clementini *et al*. 1993) expands on this further with non-empty intersections being represented by the dimension of the intersection.

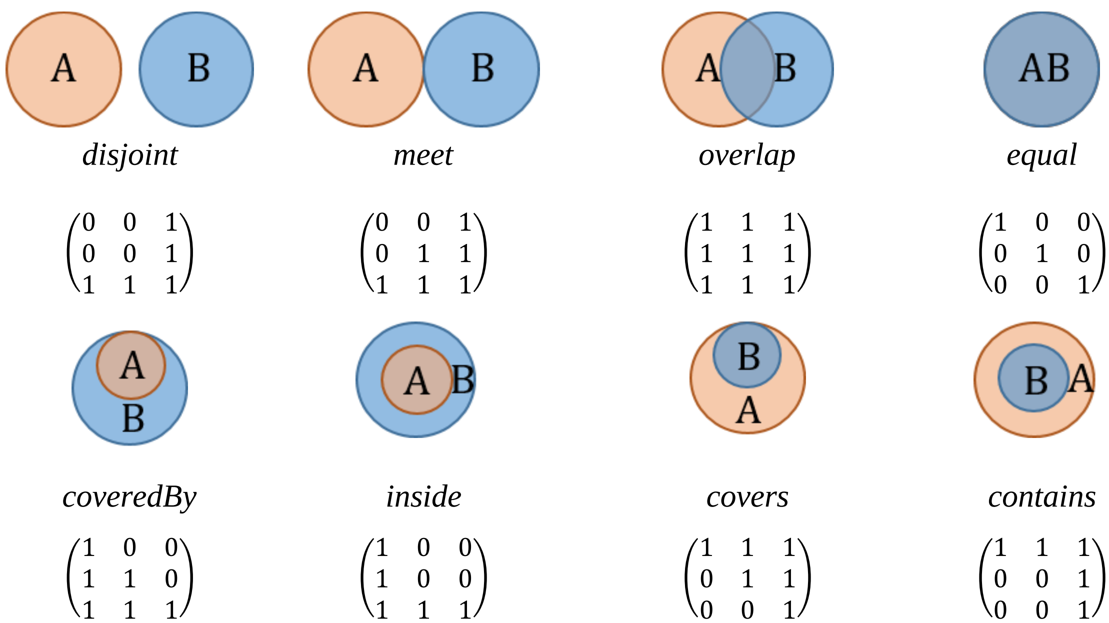


Figure 2.2 The eight region-region relations and their matrices as described by the 9-intersection (Egenhofer and Herring 1991b)

In addition to the eight region-region relations, 33 relations have been identified between two simple lines and 19 relations between a region and a line (Egenhofer and Herring 1991b). The Region Connection Calculus (RCC) is an alternative to point-based constructions, considering regions as objects themselves, instead of derived objects (Randell *et al.* 1992a). A pair of regions is considered connected if they share a common point. This model allows the representation of regions with holes or separations—such information is not explicitly captured by the 4-intersection.

This framework is based on Clarke's connection calculus, which introduces the relation C(x,y) to denote the connection between x and y (Clarke 1981). Using axioms to restrict how regions can be connected, RCC defines the same eight base relations between regions (albeit with a different naming convention) (Bennett 1998) but does not capture points or lines.

2.2.2 Detailed Binary Topological Relations

Objects within a scene can often be related to each other through a set of binary relations. This representation is the most common, being integral to both the 9-intersection and RCC. The eight region-region relations are examples of this approach.

While these theories can handle the representation of complex objects of differing construction, sets of relations that are designed to handle specific features and complexities may fare better in specific cases. The 9^+-intersection is such an approach (Kurata and Egenhofer 2007; Kurata 2008a). While the 9-intersection utilizes a 3x3 matrix, the 9^+ method allows multiple separations for the boundary, interior, or exterior of the spatial object—each cell of the matrix can be further subdivided.

$$R(D,R) = \begin{pmatrix} D^o \cap R^o & D^o \cap \partial R & D^o \cap R^- \\ \begin{bmatrix} \partial_1 D \cap R^o \\ \partial_2 D \cap R^o \end{bmatrix} & \begin{bmatrix} \partial_1 D \cap \partial R \\ \partial_2 D \cap \partial R \end{bmatrix} & \begin{bmatrix} \partial_1 D \cap R^- \\ \partial_2 D \cap R^- \end{bmatrix} \\ D^- \cap R^o & D^- \cap RB & D^- \cap R^- \end{pmatrix} \tag{2.2}$$

For example, the relation between a directed line and a simple region divides the boundary of a directed line (D) into two components, a head (∂_1) and a tail (∂_2) (Eqn. 2.2). Using this method, Kurata expanded the existing framework of the 9-intersection and represents the relations for DLine-Region relations in $\mathbb{R}^3$, as well as DLine-Line and Region to HoledRegions in numerous embedding spaces, displaying the descriptive power of this extension (Kurata and Egenhofer 2008b; Kurata 2010).

Another fine-grained binary approach allows for an advanced expression of an *overlap* relation between two non-holed objects (Eqn. 2.3), where x is the number of connected components of $A \cap B$, a is the number of connected components of $A \backslash B$, b is the number of connected components of $B \backslash A$, and o is the number of connected components of $(A \cup B)^o$, (Galton 1998).

$$[A, B] = \begin{pmatrix} x & a \\ b & o \end{pmatrix} \tag{2.3}$$

This specialized *overlap matrix* (Eqn. 2.3) distinguishes a single topological relation (*overlap*, loosely) as 23 variations based on the connectedness of objects under union, intersection, and set difference. This approach also allows the similarity between different *overlap* configurations to be determined. The expression of similarity in the topological setting leads to the distinction between the coarse topological relations that have been presented, such as the eight region-region relations, and detailed topological relations, which also may consider sequence, dimension, type of intersection, crossing direction, boundedness, and the compliment relationship (Egenhofer 1997; Egenhofer and Mark

1995a). Specifying the dimension of an intersection, for instance, can bring the representation of a scene closer to the reality that it purports to represent. Consider two overlap scenes that need intersection dimension to distinguish them (Figure 2.3).

(a) (b)

Figure 2.3 Two *overlapping* objects. (a) A simple *overlap* occuring at two boundary points and (b) an *overlap* along a boundary edge and a single boundary point.

Both scenes have been described as *overlap*, but they clearly have additional distinctions, such as a 1-dimensional boundary *cross* versus a 0-dimensional boundary *cross*. The 4-intersection has been modified to more fully represents the relations between two objects with additional topological invariants (Egenhofer and Franzosa 1995). The resulting theory requires dimension and intersection sequence, intersection type (boundaries *touching* or *crossing*), and the relationship with the complement, which determines whether a boundary component is bounded by a partition of the exterior.

The sequence of boundary intersections is also of interest; in any setting that records more than a coarse representation of a scene, allowing a pair of spatial objects to exhibit multiple intersections, it is possible to place them in sequence. The sequences are cyclic; regardless of start position the elements occur in a set order (Herring 1991). Without an associated ordering, additional scene specification will still result in ambiguity.

2.2.3 Topological Relations with Holed Regions

Of the relations described, those between regions with holes are potentially the most complex and diverse. A hole may represent any number of unique spatial phenomena, such as an independent territory carving out a space inside another country (Vatican City *inside* Italy), or a more technical scenario, like the concept of a hole in a sensor network. While the 9-intersection distinguishes eight topological relations between two simple regions, there are 23 topological relations between a simple region and a region with a hole (Vasardani and Egenhofer 2008), and 152 topological relations between two holed regions (Vasardani and Egenhofer 2009). Holes may exist either completely contained within the host object, be in contact with the boundary of an object, or split an object (Dube *et al*. 2015; Hahmann and Gruninger 2009). Gaps may also exist within the union of multiple objects, having no specific host (Casati and Varzi 1994; Hahmann and Brodaric 2012; Lewis *et al.* 2013).

Conceptually similar, a discussion on holed regions naturally leads to the need for a *surrounds* relation. A holed object *surrounds* any objects contained within its cavity in $\mathbb{R}^2$. The surrounds relation, however, is more complex as multiple objects in concert can form a gap that surrounds another object; independently these objects might be subject to the relation disjoint with the surrounded object, but together they form a ring. There are seven *surrounds* relations: *surroundsEmpty*, *surroundsAttach*, *surroundsAttachHole*, *surroundsDisjoint*, *surroundsDisjointHole*, *surroundsMeet*, and *surroundsSplitPocket* (Figure 2.4) (Dube and Egenhofer 2014).

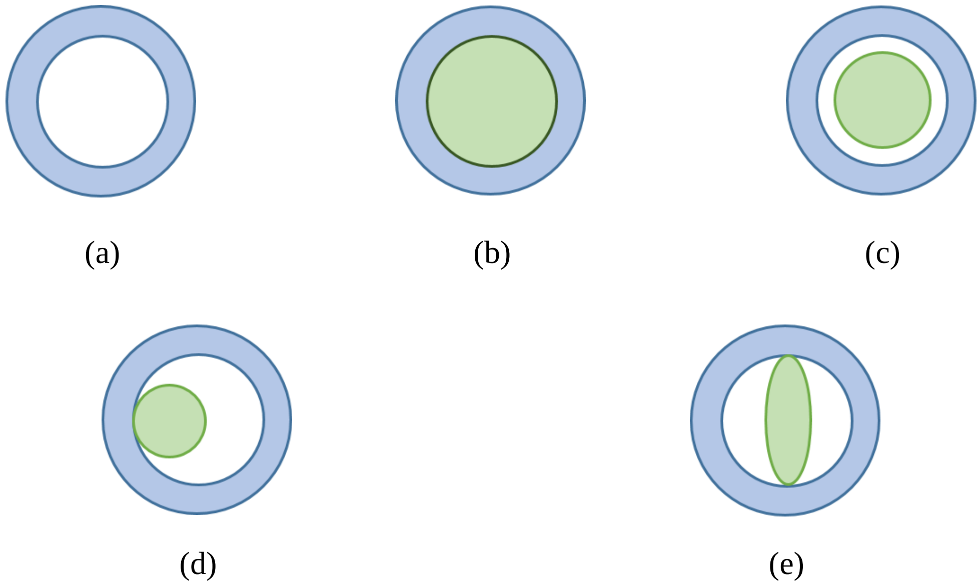


Figure 2.4 Five *surrounds* relations with a holed region. These relations include: (a) *surroundsEmpty*, (b) *surroundsAttach*, (c) *surroundsDisjoint*, (d) *surroundsMeet*, and (e) *surroundsSplitPocket* (Dube and Egenhofer 2014).

Such a construction is necessary when representing certain *fiat* objects, such as land-locked political subdivisions (Dube *et al.* 2015). The boundaries of such objects contrast with those of *bona fide* objects—those that exist naturally like the shore of a lake (Smith 1995)—but both require special attention since the shifting boundary of a lake has every possibility of being as complex as a shifting geopolitical boundary.

2.2.4 Direction and Distance

Beyond topological models, which rely on specific object constructions and specially defined spaces, exist other means of relating objects within a scene, such as through direction and distance. Additionally, topological models and direction-based models can be utilized in concert (Li 2007; Frank 1995; Goyal 2000; Cohn *et al.* 2014; Kritzman and Hahmann 2018; Freksa 1992).

One of the most common means of representing direction utilizes a familiar set of cardinal direction relations; those that relate objects through their bounding rectangles (Papadias and Sellis 1994), projection-based frames of reference (Frank 1995; Goyal 2000), or conical frames of reference (Peuquet and Zhan 1987).

By combining directional relations and proximity-based relations, models that are even closer to generalized human perception have been developed (Worboys *et al.* 2004; Moratz and Ragni 2008; Clementini *et al.* 1997). Qualitative direction and distance relations are closely related to spatial cognition (Section 2.1.1) and language-based descriptions of space (Section 2.1.2).

2.3 Spatial Objects

When a qualitative representation suffices—when metric details are abstracted away—one still needs to determine what properties are to be included—there is no one-size-fits-all solution. The specific properties—whether based on connectivity, containment, direction, or some other aspect of qualitative representation—allow the construction of specific spatial relations. Sets of spatial relations allow entities to be related against one another and reasoned about.

A discussion of spatial objects, their construction, and their relations requires appropriate motivation. To start, point-set topology is considered (Alexandroff 1961; Munkres 2000; Adams and Franzosa 2008), with an assumption that the reader possesses a basic understanding. Most depictions of geographic reality can be projected on the Euclidian Plane $\mathbb{R}^2$, retaining the local topological structure. The choice of embedding space can affect the types of objects that are represented therein—a region or a volume cannot be described in a 1-dimensional embedding, but lines can take on additional configurations when represented in two or more dimensions, for instance. Even when a model supports multiple embedding spaces there are often consequences. As an example, the 9-intersection describes eight spatial relations between simple regions in $\mathbb{R}^2$ (Egenhofer and Herring 1991b), but $\mathbb{S}^2$ allows three additional relations when the sphere is considered as the embedding space (Egenhofer 2005). These relations require the entire embedding space to be filled, which is not possible between simple regions in $\mathbb{R}^2$ (although other objects may suffice); no matter what topological transformation a pair of simple regions undergo, they cannot be scaled and positioned to mutually fill the entire space of $\mathbb{R}^2$.

2.3.1 Constructing Simple Objects

Spatial entities such as points and regions may be described in terms of sets under general (point-set) topology (Adams and Franzosa 2008). In this setting, Egenhofer and Franzosa (1992) describe a spatial region through the following definitions involving the concepts of interior (A^{o}), boundary ($\partial\mathrm{A}$), and closure ($\overline{\mathrm{A}}$), for some object A:

Definition 2.1 Let X be a connected topological space. A spatial region in X is a non-empty proper subset A of X satisfying (1) A^{o} is connected and (2) $\mathrm{A} = \overline{\mathrm{A}^{\mathrm{o}}}$.

Proposition 2.2 If A is a spatial region in X then $\partial\mathrm{A} \neq \emptyset$.

Under this specification, a region is a set of points defined by the closure of a connected interior. Later approaches would also incorporate A's exterior (A^-), (Egenhofer *et al.* 1991b; Egenhofer *et al.* 1993). As an addition to general topology, algebraic topology (Alexandrof 1961; Spanier 1966) allows for the creation of objects by gluing together cells of varying dimension, allowing more complex constructions. Egenhofer and Herring (1991b) describe the construction of points, lines, regions, and more complex objects in $\mathbb{R}^2$ using 0-cells (vertices), 1-cells (a segment connecting two 0-cells), and 2-cells (an area, represented by closed, non-intersecting 1-cells). A cell complex is taken to be an aggregate of cells. In such a manner, a point is described simply as a 0-cell, a line as a connected sequence of 1-complexes that neither cross nor loop with two disconnected boundaries, and a region is represented as a 2-complex with a connected interior, boundary, and exterior.

2.3.2 Compound Spatial Objects

One can also produce objects of mixed type, such as instances where a single object is constructed from a line and a region, for example. These compound objects expand on the previously defined objects, and the result is a significant number of additional configurations, for instance, using a point-set methodology to generate a set of spiked regions created by the union of a region and a simple line (Egenhofer 2009). Alternatively, Clementini and Di Felice expand beyond the point-set method to include additional features, such as lines with self-intersections, separated objects, and objects with holes (Clementini *et al.* 1995; Clementini and Di Felice 1996). Li is able to use the 9-intersection to represent 43 relations between regions realizable in $\mathbb{R}^2$, but does not consider the internal relations between an object and its parts, such as holes (Li 2006). Moving toward a localized representation of complex spatial relations, capturing the relation between two spatial regions and their subparts independently yields separate relations for each component which allows for more detail to be captured than by considering the relations between spatial regions in aggregate only (McKenny *et al.* 2007).

Schneider and Behr (2006) provide an extensive accounting of relations that exist between complex objects when using the 9-intersection, which may contain separations, holes, and cycles. There are, for instance, 33 relations between such complex regions, 82 relations between complex lines, and 43 relations between a complex region and a complex line. Relations between groups of points are also considered. These complex objects are specialized, including lines with bifurcations, regions with handles and spikes, cyclic lines, disconnected points, and other configurations. Separations of the exterior and interior, however, cannot be distinguished.

2.4 Composition

When two relations are known, and those relations share an object in common, an additional relation is inferable. For instance, if A *meet* B and B *contains* C, one can infer that A also *disjoint* C. The systematic reasoning behind this is known as *composition*. Properties such as composition and converseness are derived from a relation algebra over a set of relations (Tarski 1941; Maddux 1990). For 9-intersection relations, *composition* can be expressed in terms of inference rules about point sets (Egenhofer and Sharma 1992; Egenhofer 1994; Renz and Ligozat 2005).

A composition table represents the product of all pairs of relations (i.e., each as a row and as a column), representing the possibilities between a pair of relations. The composition table for region-region relations (Egenhofer 1991) contains 64 entries (8 by 8 relations), and through this composition two region-region relations can yield a unique result (27 entries), an ambiguous result (34 entries), or the universal relation (3 entries). In the case of the universal relation no information is gained through the composition, but in all other instances composition allows some degree of information to be derived for additional relations without the explicit representation of those relations, allowing additional relations to be produced from incomplete information and reducing the need for explicit storage of relation information in specific cases. Composition tables for logical approaches, such as RCC-8 have also been derived (Cohn *et al.* 1997).

Composition is also useful when considering the relation between a specific subpart of a compound object, such as the hole in a holed region, and another object in the scene (Egenhofer *et al.* 2007; Egenhofer and Sharma 1993) and are used as a check on the consistency of the relations (Montanari 1974). Complex areal objects are also able to be represented with a labeled tree graph to model the relation between objects and their subparts, with containment being explicitly represented at each level of the tree (Worboys and Bofakos 1993)

2.1 Similarity

When reasoning with complex spatial information, several problems may arise, such as the volume of information being too large—to the extent that reasoning becomes difficult—or the provided information may be incomplete. By applying constraints on spatial reasoning, the consistency of relations between objects within a scene is demonstrable (Egenhofer and Sharma 1992; Egenhofer and Sharma 1993).

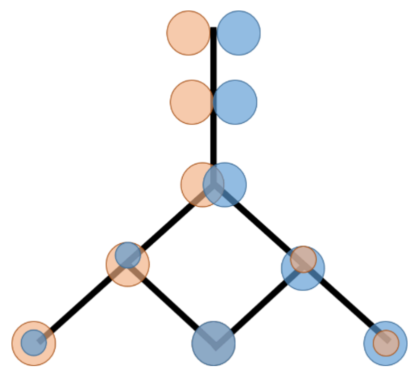
The relation between a pair of objects can be deformed by gradually changing one of the objects through translation, rotation, isotropic scaling, anisotropic scaling, or other transformations. The need for similarity assessment when handling spatial data also arises from the complexity and quantity of relations being stored (Nedas and Egenhofer 2003). Regarding each type of deformation, a conceptual neighborhood graph is formed by representing each relation by a node in a graph, with edges connecting closest neighbors. A traversal of one edge, from one relation to another, indicates those relations are separated by a single topological deformation, while less similar relations require more than a single transformation to produce (Egenhofer and Sharma 1992). By comparing the matrices for each relation in a 9-intersection setting the conceptual distance between them can be determined (Figure 2.5).

$$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix} \textit{disjoint}$$

(a)

$$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \textit{meet}$$

(b)



(c)

Figure 2.5 A conceptual neighborhood graph for region relations. (a) The matrices for *disjoint* and (b) *meet*, distinguished by a single difference in the content of the boundary-boundary intersection and a conceptual neighborhood graph (A-neighborhood) for 9-intersection showing the topological distance between relations.

The matrices for *meet and disjoint*, for instance, only vary in the content of their boundary-boundary intersection, so they are conceptually close, while *disjoint* and *inside* are significantly farther apart, the exact degree dependent on the transformation being considered.

Other work has generated additional graphs for different sets of objects beyond regions related through the 9-intersection, such as the relations between regions and lines, the relations between regions in different models such as RCC-8

(Randell *et al.* 1992), and the temporal domain (Bruns and Egenhofer 1996; Cohn *et al.* 1997; Egenhofer and Al-Taha 1992; Egenhofer and Mark 1995b; Egenhofer *et al.* 1993; Freska 1991; Klippel *et al.* 2008; Reis *et al.* 2008; Egenhofer 2010).

2.5 *N*-Object Spatial Scenes

Moving beyond coarse relations between pairs of spatial objects allows for the modeling of scenes that capture a greater degree of complexity between objects. These complex scenes may make use of simple or complex objects within some predefined embedding space.

2.1.1 The o-notation and i-notation

Contemporary work involving dimension, *touching* and *crossing* relations, and boundary intersection sequence includes the *o-notation* and its extension, the *i-notation*. Both approaches were specifically designed to accommodate an arbitrary number of regions and intersections (Lewis *et al.* 2013; Lewis *et al.* 2014). A spatial scene modeled with *o-notation* is described in terms of the individual intersections each object participates in. Each intersection is represented by a string of symbols, and strings are recorded in sequence by walking around each object in a clockwise traversal (Eqn. 2.4).

$$\partial \mathrm{A}_{comp}: o_s(dim, T, C) \tag{2.4}$$

For an *o-notation* string, ∂A_{comp} represents the boundary component of a region *A*, *S* is the collection of regions the boundary component is currently outside of, *dim* is the dimension of the intersection (0 or 1), *T* is the collection of region boundaries subject to a touch relation in the specified intersection, and *C* is the collection of region boundaries subject to a cross relation.

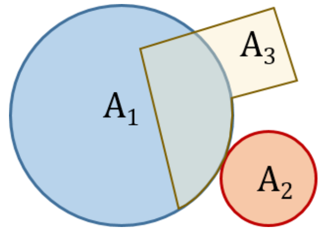


Figure 2.6 An example scene featuring 3 regions, $A_1, A_2,$ and A_3.

The notation for Figure 2.6 results in three *o-notation* strings (Eqs. 2.5-7) to completely represent the depicted scene.

$$\partial A_1 \colon o_{\{A_2,A_3\}}(0,\emptyset,A_3)o_{\{A_2\}}(1,\emptyset,A_3)o_{\{A_2\}}(0,A_2,A_3)o_{\{A_2\}}(1,\emptyset,A_3) \tag{2.5}$$

$$\partial A_2 \colon o_{\{A_1,A_3\}}(0,\{A_1,A_3\},\emptyset) \tag{2.6}$$

$$\partial A_3 \colon o_{\{A_1,A_2\}}(1,\emptyset,A_1)o_{\{A_1,A_2\}}(0,A_2,A_1)o_{\{A_1,A_2\}}(1,\emptyset,A_1)o_{\{A_2\}}(0,\emptyset,A_1) \tag{2.7}$$

The *o-notation* and *i-notation* are further empowered by their ability to discern holes and separations within an object, and holes (gaps) created by an ensemble of objects (Figure 2.7).

(a) (b)

Figure 2.7 Two scenes with an exterior separation. (a) A region is *surrounded* by an ensemble of regions and (b) where a region is outside of an ensemble of regions.

While the *o-notation* can represent many complex spatial scenes, there are also certain configurations for which the notation alone is insufficient to produce a unique representation (Figure 2.8a). Ambiguity arises when multiple objects share a single

0-dimensional intersection (Figure 2.8b). In this instance it would be possible to tell that object *B* touches *A* separately from its intersection with *C*, *D*, and *E*, but there is no basis for determining the sequence in which objects *C*, *D* and *E* are oriented around that intersection—if four or more objects intersect at a specific point there are multiple permutations of that sequence.

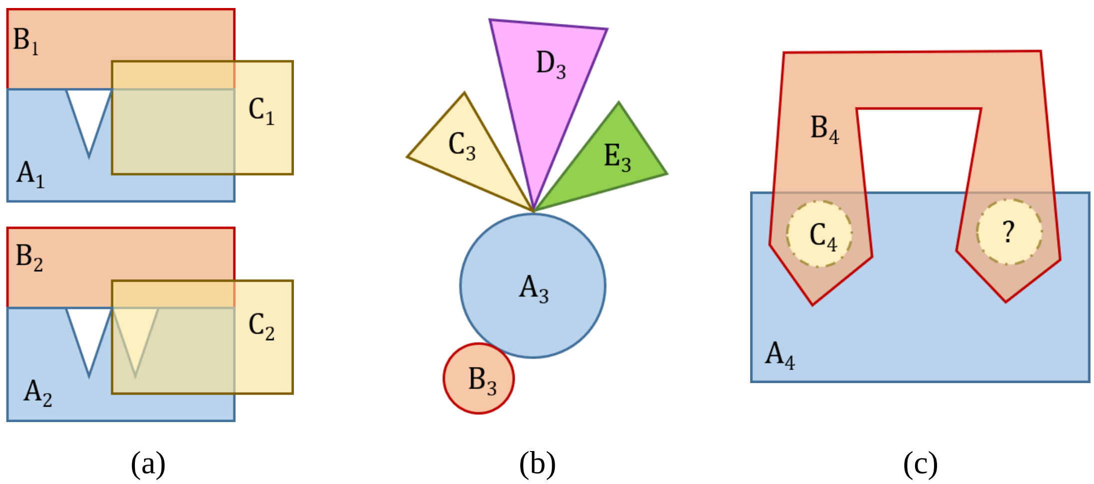


Figure 2.8 Four distinct problems arising from scenes of regions. (a) The identification of gaps between regions is needed to differentiate between two scenes where the o-notation is identical, (b) the order in which C, D, and E appear is unknown in *o-notation*, and (c) region C has an indeterminate location.

This problem arises because the intersection sequence around the boundary of an object is captured, but the sequence of objects around an individual intersection is not. This discrepancy occurs because the *o-notation* captures the set of *touched* objects and the set of *crossed* objects for a given intersection, but a set does not maintain sequence. Furthermore, it is sometimes impossible to tell where exactly a region is situated when it is fully contained within another object or the union of multiple objects when the containing space has multiple similar partitions (Figure 2.8c).

2.1.2 Maptree

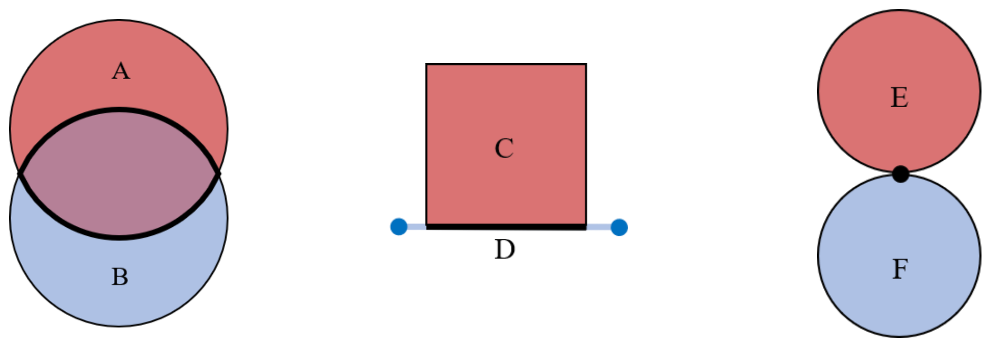
Another theory, employing a graph structure is MapTree (Worboys 2012). MapTree utilizes combinatorial maps to build a model of space based on nodes and edges in order to partition space and develop a containment hierarchy (Figure 2.9).

Figure 2.9 A complex scene modeled with MapTree.

MapTree represents complex objects, such as those with separations, as well as holed objects, and scenes containing an arbitrary number of objects, but objects are individually indistinguishable.

2.1.3 CODI

A comprehensive approach, *CODI*, is a family of mereotopological theories that are more representative than the models described thus far, capturing the relations between compositions of manifolds (Hahmann 2013). *CODI* combines relative dimension and containment in order to define three types of contact between objects (Figure 2.10): *partial overlap*, where objects of equal dimension share a part of equal dimension to the objects (Hahmann and Grüninger 2011b; Hahmann 2013); *incidence*, where the shared part is of equal dimension to one of the objects; and *superficial contact*, where the shared part is of lower dimension than the objects. CODI also captures properties such as whether an object or its boundary are single piece regions or if there are holes present through various unary predicates. Certain predicates and functions are similar to those present in related theories, such as the function *ch* which returns the convex hull of a region (the relevant function in RCC is *conv*, for instance), and may present a gradient of representation not present elsewhere, such as Con, ICon and UCon, which capture varying strengths of the notion of connectedness within an object. The approach is also extended to capture the sequence in which objects intersect, as well as *betweenness* relations which follow from the notion of sequence (Hahmann and Grüninger 2011a; Hahmann 2013).



(a) (b) (c)

Figure 2.10 Three contact relations between objects. (a) Two 2-dimensional objects share a 2-dimensional part (*partial overlap*), (b) a 2-dimensional and a 1-dimensional object share a 1-dimensional part (*incidence*), and (c) two 2-dimensional objects share a 0-dimensional part (*superficial contact*).

The models described in this section go beyond the traditional approach of representing a scene through an arbitrary number of binary relations, allowing a representation potentially much closer to the true form of the objects being described by increasing both the quantity and complexity of objects represented.

2.6 Summary

In modeling a geographic reality, one identifies the participating objects and captures the relations that exist between them. This process results in the creation of a spatial scene. Spatial scenes can be expressed in many ways, many of which are informed by human perceptions of space. How people think and talk about spatial concepts and how they choose to depict them influences the models that are developed, as human perception, formal theories, and implementations all (optimally) work in concert.

To represent a spatial scene qualitatively with topological relations there are still many considerations to be made; the chosen embedding space has implications on which types of objects one represents and the set of possible relations between them, a model may accommodate only simple regions or additional complexities such as separations and holes. Furthermore, the types of relations captured are often the centerpiece of any qualitative depiction: does one care only about intersections, does sequence matter, or dimension?

Concerning the theories discussed in this chapter, the 9-intersection (Egenhofer and Herring 1991b) captures the spatial relations between pairs of objects, such as regions or lines, coarsely. While relations can be inferred in scenes with more than two objects through composition, the result is not always conclusive. Varying extensions capture additional properties, such as dimension (Clementini *et al*. 1993; Egenhofer and Franzosa 1995), direction (Kurata and Egenhofer 2008b; Kurata 2010), or the containment relations with holes (Vasardani and Egenhofer 2008; Vasardani and Egenhofer 2009; Dube *et al.* 2015), as well including objects of additional complexity, such as bifurcated lines and other complex configurations (Li 2006; Schneider and Behr 2006). However, these approaches each branch in separate directions--they are not designed for interoperability.

The o-notation (Lewis *et al.* 2013) and i-notation (Lewis *et al.* 2014) are an attempt to incorporate some of these properties into a single theory by representing the container of each object as well as the *crossing* and *touching* interactions each object has with intersecting scene objects, and the dimension of those intersections. However, while both approaches model the sequence of intersections around the boundary of an object, they do not model the sequence of objects around an intersection, which results in ambiguity. Lines are also not explicitly represented.

Maptree (Worboys 2012) captures structural details with its graph-based approach that previously described approaches cannot model, but there is no body of work relating Maptree to any spatial relations--like the o-notation and the i-notation it is purely structural, limiting the ability to reason about a scene without further development.

While each of the theories discussed thus far represent individual elements of the desired theory, there is no cohesive base that would enable them to be meaningfully combined. The CODI (Hahmann 2013) approach achieves this--drawing from a wide body of existing work in order to capture a diverse range of spatial properties--with the exception that the family of logical theories that it derives from utilize a different methodology (in some respects) than the theories that form the basis of this work. In order to resolve this lack of representation a fine-grained model of spatial scenes is developed in Chapter 3.

CHAPTER 3

PRODUCING THE SCENE NOTATION

A comprehensive model for representing the detailed relations between regions and lines needs to capture any number of boundary intersections between combinations of objects, as well as the sequence and dimension of those intersections (Egenhofer and Herring 1991b).

3.1 Modeling Objects

In the approach developed, named *Scene Notation*, the regions and lines that make up a spatial scene are represented through algebraic topology—they are comprised of *cell complexes* as the unions of n-cells, where *n* represents the dimension of the cell (Egenhofer *et al.* 1989). A 0-cell is a singular point, a 1-cell is an edge defined between two points, and a 2-cell is an area defined by edges connected endpoint-to-endpoint in sequence, forming a cycle. To accommodate spatial objects that are more complex than these, the notion of a *cell complex* is needed—the union of a multiplicity of cells. Cell complexes allow for the creation of increasingly representative objects—beyond simple edges and triangular regions.

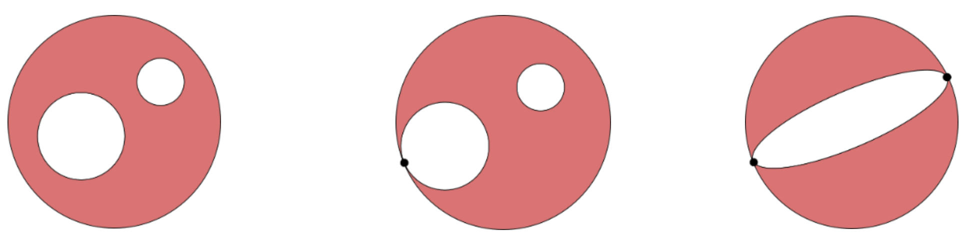
While the traditional definitions for spatial objects derived from cell complexes mostly suffice (as well as the associated definitions for object interiors, boundaries, and exteriors), a specific modification for the definition of holed regions is introduced. The original definition of a holed region given by Egenhofer and Herring (1991b) is as follows:

Definition 3.1 A *region with holes* is a region with a disconnected exterior and a disconnected boundary.

This strict definition defines holes that exist fully *inside* of the host region (Figure 3.1a) but does not accommodate holes that *touch* the boundary of the host region (Figures 3.1b and 3.1c). To accommodate such holes, additional definitions are introduced for the sake of this work:

Definition 3.2 A *point-connected* intersection occurs when an object intersects with the boundary of an areal object at a single point, or when an object intersects with the extent of a line at a single point. Objects may share multiple point-connections as long as they do not intersect along an edge.

Definition 3.3 A *region with holes* is a region with a connected interior, with a disconnected or point-connected boundary, and with a disconnected or point-connected connected exterior.



(a) (b) (c)

Figure 3.1 Three scenarios with a disconnected exterior. (a) A region with holes strictly *inside* of it, (b) a region where a hole is *coveredBy* the host region, and (c) an example lacking a hole, where two regions with disconnected interiors split the exterior to form a gap instead.

This updated definition allows a wider range of holes (Figure 3.1b) where the hole touches the region's boundary but does not split the interior. Finally, an additional type of object is included, beyond those defined by Egenhofer and Herring (1991b):

Definition 3.4 A *gap* is a bounded exterior induced by the union of distinct spatial objects with disconnected interiors, independent of the holes within any individual object.

A gap is like a hole, but is bounded by a collection of spatial objects, rather than existing within any individual object (Casati and Varzi 1994; Hahmann and Brodaric 2012). The bounded exterior in Figure 3.1c is an example of a gap. Regions, holes, gaps, lines, and points form the major elements of the developed approach.

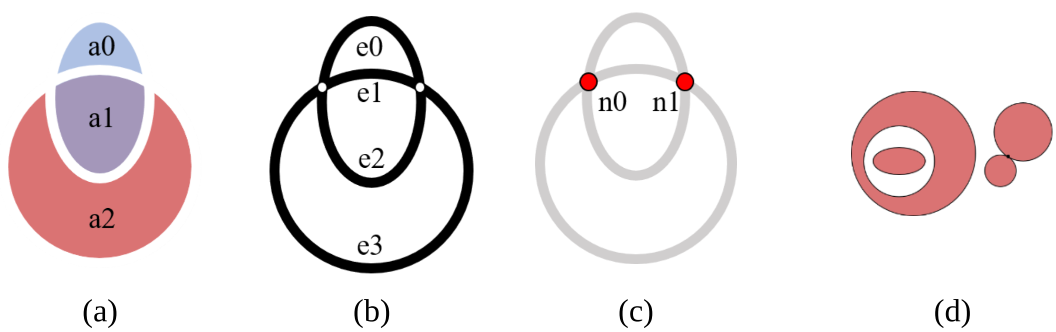


Figure 3.2 An object is constructed from (a) areas, (b) edges, and (c) nodes. More complex constructions (c) can be made by taking the union of simple objects to form a collection of similar objects.

For clarity, the components of these objects are also given a consistent naming herein—an *area* refers to any of the 2-cell faces that partition the extent of a region (Figure 3.2a); an *edge* refers to any of the 1-cell faces that partition the extent of a line, or form the boundary of an area (Figure 3.2b); a *node* is the 0-cell where edges intersect or an endpoint of a line (Figure 3.2c); and finally a *collection* is a grouping of either lines or regions (with or without holes), which enables simple objects to be combined under union in order to form complex constructions with disconnected interiors (Figure 3.2d). Together these components allow various objects of different types and constructions to be reasoned with through their individual parts.

3.2 Validating Collections of Objects

While collections are sets of objects with homogeneous dimension, adding elements to such a set is restricted to objects that are *disjoint*, or do not *meet* along an edge. Objects that *meet* at a node but do not share an edge are valid collection members, for instance.

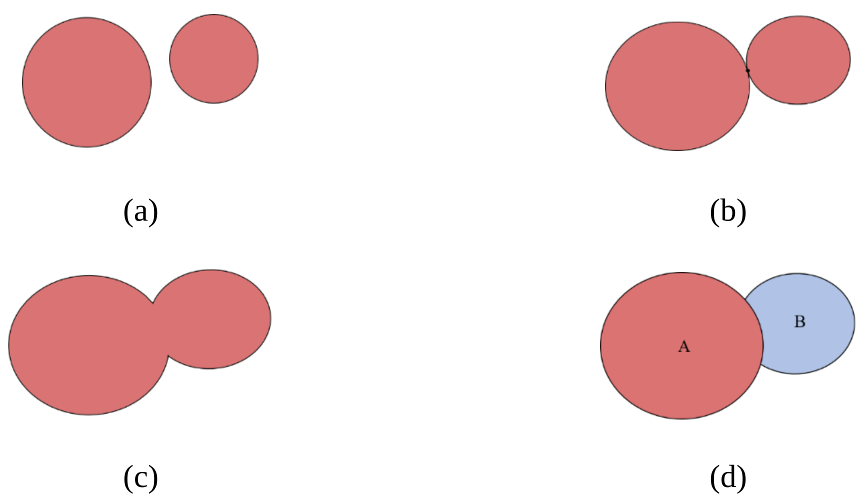


Figure 3.3 The landmass in a lake is revealed, displaying different relations. (a) The landmass is disconnected, (b) the landmass converges at a single point, but is interior disconnected, and (c) the landmasses have merged into a single region. Alternatively: (d) if the objects modeled have distinct identities (regions 'A' and 'B', opposed to 'land') they *meet* (for example), instead of merging (Coan 1996) and cannot be part of the same collection.

Two islands within the same lake, for instance, could be represented by a collection of regions whose union forms a single complex region characterizing land within the lake above the water level (Figure 3.3a). As the water level decreases the pair of islands would begin to converge, but still maintain their distinctiveness—they cannot yet be represented by a single simple region as their interiors are still disconnected (Figure 3.3b). Finally, as more water evaporates the two islands become one, sharing a single connected interior (Figure 3.3c).

When the modeled regions do not share an identity, their interiors (and boundaries) remain distinct, even when both regions converge (Figure 3.3d). The relations A *meet* B, A *equal* B, or A *overlap* B convey meaning, while the relations A *meet* A, A *equal* A, or A *overlap* A are at best a tautology and at worst meaningless. The regions or lines that constitute a collection, therefore, are related through common identity, but restricted in the relations that they share.

This example illustrates a key component of such collections: the interiors of complex objects must not intersect; two components of the same object should not exist in the same location concurrently while maintaining separate identities (Coan 1996). Specific constraints are given below for adding elements to collections of regions, holed regions, lines, and points:

Definition 3.4 A *collection* of regions or lines is a dimensionally homogeneous set of objects such that:

- The intersection between all object interiors is empty.
- The intersection between all object boundaries is a set of nodes or empty.

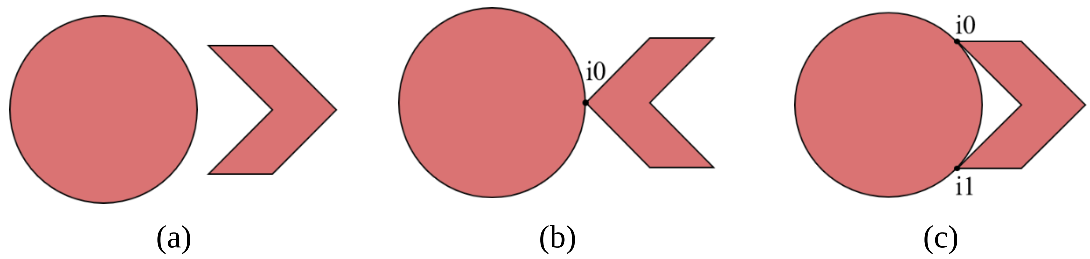


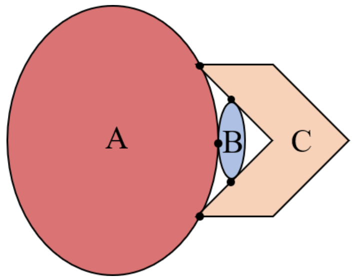
Figure 3.4 Collections of simple objects take on various configurations, assuming all interiors are disconnected. (a) The objects are *disjoint*, (b) the objects *meet* at a single node, and (c) the objects *meet* at multiple nodes.

Together these restrictions ensure that objects within a collection are *disjoint* (Figure 3.4a) or *meet* without sharing an edge (Figures 3.4b and 3.4c). Objects that share an extended boundary or interior cannot be members of the same collection (Figure 3.3d).

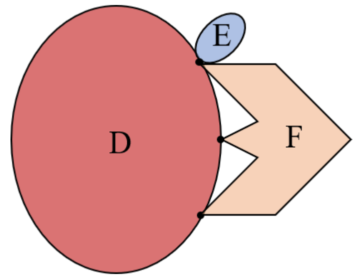
3.3 Properties of Spatial Scenes

Developing a set of primitive objects is insufficient to describe a scene up to homeomorphism. To ensure a consistent mapping, additional elements are needed, such as the boundary sequence in which intersections occur as an object is traversed and considerations such as identifying the parts into which an object is divided. These elements and more are motivated below.

A starting point to establish an accurate representation of a scene is to represent all scene intersections uniquely. Knowing that three regions *meet* each other, for instance, is insufficient to uniquely describe the scene (Figures 3.5a and 3.5b). By representing each boundary intersection explicitly, ambiguous configurations are limited. While each region *meets* the other two regions within each example scene, the number and sequence of *meet* relations differs.



(a)



(b)

Figure 3.5 Two examples, each of three objects *meet* the others, but their configurations are not equivalent. (a) The oval B exists between the circle A and the chevron C with five intersections between A, B, and C, and (b) the oval E *meets* the circle D and the chevron F from the outside, with three intersections between D, E, and F.

By representing the sequence of intersections (nodes) around the boundary of a region (or along the extent of a line), fine-grained distinctions can be made, such as how many times a pair of objects *meet* or in what order the intersections occur. For example, in Figure 3.5a region B *meets* A once between a pair of nodes (where C and A *meet*). In Figure 3.5b regionD *meets* F three times, while D *meets* E only at one of those intersection nodes.

Representing the sequence of objects that intersect at a given node also produces a more consistent representation of the scene. Rather than being limited to the knowledge that three regions *meet* at an intersection point (Figures 3.6a and 3.6b), a consistent traversal of the node allows the objects to be placed in the order encountered (Figures 3.6c and 3.6d).

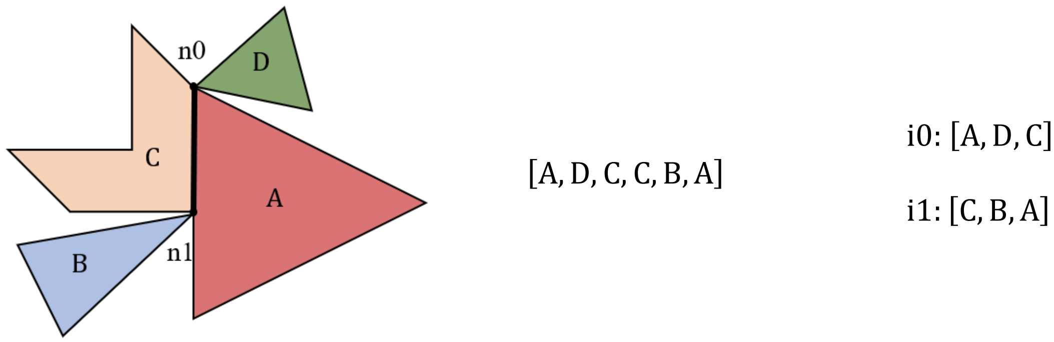
(a) (b)

$[A_0, D_0, C_0, B_0]$ $[B_1, D_1, C_1, A_1]$

(c) (d)

Figure 3.6 Four region encircle a node in two different configurations. (a) The first configuration and (c) the counter-clockwise sequence of objects around its node, and (b) the second configuration and (d) its different counter-clockwise sequence of objects around its node. The start and end of the sequence is irrelevant since it is cyclic.

However, not all boundary intersections are 0-dimensional (Figure 3.7a). Taking the sequence of objects that meet along an edge (Figure 3.7b) is less elegant than taking the same sequence for a node (Figure 3.7c). Given that an edge is defined between two nodes, the 1-dimensional intersection is instead able to be defined by a pair of nodes, each possessing its own sequence.



(a) (b) (c)

Figure 3.7 A spatial scene between several objects. (a) Objects can also intersect along edges but taking the sequence of objects oriented around an edge (b) is ambiguous compared to taking the same sequence around the endpoint nodes (c) of the edge-intersection.

Beyond boundary intersections, the components of an object must also be represented. Consider a complex region with multiple holes (Figure 3.8). Each hole may in turn intersect with or contain additional objects. Being unable to uniquely identify the specific host a hole belongs to allows the hole to be misplaced and also the objects it contains. In the depicted scene the complex region contains four holes. The square region is within a hole that is itself within an area that is disconnected from the rest of the complex region (middle-left annulus). The triangular region, however, is within a hole where the host has a weak connection to the rest of the complex region. An additional hole contains nothing at all. Placing either the square region or the triangular region in a different hole would result in an alternate configuration that captures a different topology.

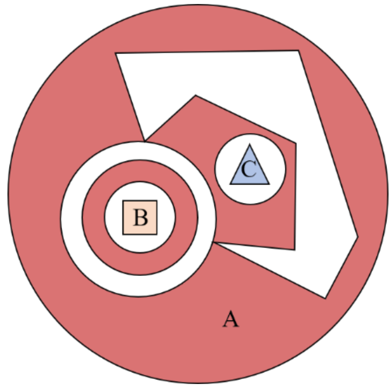


Figure 3.8 A complex object A contains four holes. Different holes contain additional objects (B and C), as well as disconnected segments A. The interior components of A need to be distinguished if the complex region is to be described accurately.

Similarly, the areas that define regions and region-like objects (holes and gaps) must also be uniquely identifiable (as with the specific edges that make up lines). If two regions intersect in two distinct areas, for instance, a third object may reside in one of the intersecting areas or the other (Figure 3.9a), or within a specific gap (Figure 3.9b). Just as the areas that make up a region (or the edges that make up a line) must be uniquely identifiable, so must the gap areas that partition the exterior.

Such refinements benefit more than scenes with regions; gaps can also be formed within scenes containing lines. For specific instances defining a gap adds refinement to the coarse line-line relations (Figure 3.10). In both configurations, the endpoints of one line are contained within the interior of the second line. However, in Figure 3.10a a gap is formed that is bounded in part by the endpoints of one line but not the other (and only two of the four edges). In figure 3.10b the gap is bounded by both sets of endpoints (and all four edges).

(a) (b)

Figure 3.9 Two similar scenes contain an ambiguously placed object. (a) Object C could be *inside* either intersecting area of A and B and (b) object F could exist within the gap between D and E, or within another partition of the exterior.

(a) (b)

Figure 3.10 Two examples of the 9-intersection relation LL3. (a) One line's endpoints are separate from the gap and (b) the same line's endpoints are within the gap.

These examples additionally demonstrate how relations between lines can benefit from the inclusion gap objects in $\mathbb{R}^2$, representing how a pair of objects with the same coarse relation might partition the exterior in different ways. By uniquely defining the components of a scene, such as holes and gaps, each element can be described unambiguously.

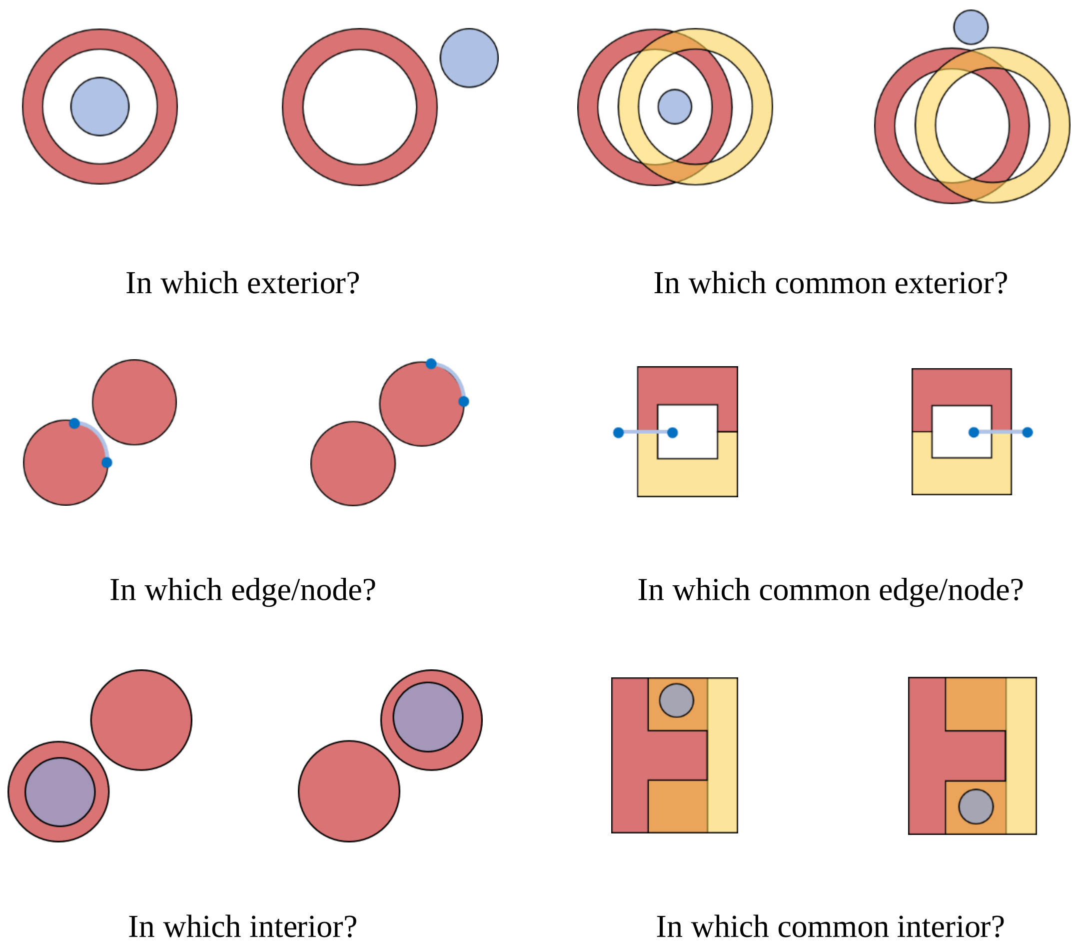


Figure 3.11 The properties to be captured by the *Scene Notation* enable complex objects to be placed. The notation should describe their explicit containment (questions of interior and exterior placement) and their edge intersections (questions of edge/node placement).

Together these properties enable the location of objects to be captured for a scene (Figure 3.11). The intersection sequence allows objects to be correctly placed along the edges of each other and explicit containment allows the correct placement of objects when their edges do not intersect.

3.4 Describing a Spatial Scene

Each scene is comprised of an arbitrary number of region-like objects (regions, holes, and gaps) that are themselves built from a set of areas, as well as an arbitrary number of lines built from a set of edges. Furthermore, the mutual components shared between various objects are represented by a set of nodes.

Section 3.3 described how the notions of intersection, sequence, and containment can be leveraged in order to specify a spatial scene uniquely. Therefore, the objects that constitute a scene will be described with respect to those criteria. To that end, a specification is provided for each of the objects discussed, starting with the areas, edges, and nodes that more complex objects are built from:

Definition 3.5 An *area* is a 2-cell partition of space specified by a tuple $(instance,\ edges)$.

Instance is a unique identifier given to an area (an integer id, for instance). *Edges* refers to the set of edges that bound the area.

Definition 3.6 An *edge* is a 1-cell partition of space specified by the tuple $\big(instance,\ (node, node)\big)$.

The definition of *instance* remains the same, however the two *nodes* refer to the pair of 0-cells that serve as the endpoints of the edge. These may be intersection points, the endpoints of a line, or both.

Definition 3.7 A *node* is a 0-cell specified by a tuple $(instance,\ edge\_sequence)$.

Again, the definition of *instance* remains the same, but *edge_sequence* refers to the ordered sequence of edges that connect to the node, obtained by a counter-clockwise traversal around it.

Definition 3.8 A region-like object (*region*, *hole*, or *gap*) is specified with a tuple $(collection,\ instance,\ type,\ \ parent,\ \ areas,\ \ container)$.

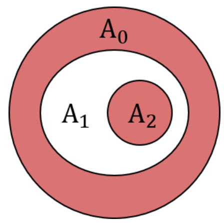


Figure 3.12 A region A_2 exists within its parent A_1. In turn, A_1 exists within its parent A_0, which has no parent within the collection A. Both regions and the holes they contain are considered independent objects in this setting; holes are more than boundary rings within a region.

Collection refers to the name of the collection the object belongs to (an object is at least a member of a collection consisting of itself). The *instance* distinguishes the object from other objects in its collection and is an integer count. Together the collection and instance can be used to identify an object, such as A0 being the 0^{th} member of collection A. *Type* is an indicator of the object's type (region, hole, gap). *Parent* in this context refers to the element within a collection that hosts the object, such as a hole being hosted with a specific region (Figure 3.12). In the example scene, region A2 is hosted within hole A1, which is hosted within region A0. Region A0 has no host within collection A. *Areas* is the set of areas contained within the boundary of a region-like object. *Container* is the specific area (if any) that the object is within (i.e., the object cannot *equal, overlap, or contain* its container). This attribute allows an object to be correctly placed within a specific partition of space when there are no intersection nodes connecting it.

Lines are represented in a similar fashion, however a line in this representation is unable to contain explicit gaps within itself, so the notion of parenthood is absent.

Definition 3.9 A line object is specified with a tuple of the form $(collection,\ instance,\ edges,\ container)$.

While *collection*, *instance*, and *container* share similar definitions to the similarly-named attributes used in the specification of a region-like object, *edges* refers to the set of edges that form the extent of the line. Together these five definitions describe the information necessary to represent a spatial scene within the provided context. They can also be used with a set of operations to gain further insight into the scene.

3.5 Operations on Scene Objects

Lines, regions, holes, gaps, areas, edges, and nodes can be manipulated by a basic set of operations in order to derive additional information (such as the boundary and interior of an object) and to construct additional objects through set operations.

3.5.1 Operations on Regions

Each region is defined primarily by the set of areas it is partitioned into. Each area, in turn, is defined by the set of edges that bound it. As each region is divided into areas, and each area is bound by a sequence of edges, each edge that bounds an area is either shared with a single adjacent area within the region or it participates in the boundary of the region. First the set of all edges within a region is calculated, including those that do not participate in the boundary (Algorithm 3.1), then the boundary of a region can be obtained by representing only those edges that occur once (Algorithm 3.2), and the set of edges partitioning the interior of a region can be derived simply by taking the difference between the first two sets (Algorithm 3.3). The set of nodes along the boundary of a region are calculated as the endpoints of the edges that bound the region (Algorithm 3.4).

Algorithm 3.1 Deriving the complete set of edges for all areas that partition a region R.

Input: The set A of areas that partition the region.

Let all_edges be an empty set
For each $area \in A$
    For each $edge \in area.edges$
        $all\_edges.add(edge)$
    End For
End For
Return all_edges

Algorithm 3.2 Deriving the set of edges that bound a region R.

Input: The set of edges for all areas that partition a region R, *all_edges*.

Let $region\_boundary\_edges$ be an empty set
For each $edge \in all\_edges$
    If $edge \notin region\_boundary\_edges$
        $region\_boundary\_edges.add(edge)$
    Else
        // An edge is at most shared between two (adjacent) areas so it only
        // needs to be removed once and will not be re-added
        $region\_boundary\_edges.remove(edge)$
    End If
End For
Return $region\_boundary\_edges$

Algorithm 3.3 Deriving the set of edges internal to a region R.

Input: The set of edges for all areas that partition a region R, *all_edges*, and the set of edges that bound a region R, region*_boundary_edges*.

1. Let $interior\_edges$ be an empty set
2. $interior\_edges = all\_edges \setminus region\_boundary\_edges$
3. **Return** $region\_interior\_edges$

Algorithm 3.4 Deriving the set of nodes along the boundary of a region R.

Input: The set of edges that bound a region R, $region\_boundary\_edges$.

1. Let $region\_boundary\_nodes$ be an empty set.
2. **For** each $edge \in region\_boundary\_edges$
3. // Remembering that each edge is a set of two of nodes:
4. $region\_boundary\_nodes.add(edge.node[0])$
5. $iregion\_boundary\_nodes.add(edge.node[1])$
6. **Return** $region\_boundary\_nodes$

The process of determining the *region_boundary_nodes* can also be applied to *region_interior_edges*. By taking the endpoints of these edges and then removing any *region_boundary_nodes* the set *region_interior_nodes* is created.

With the set of nodes and boundary edges derived for a given region, the set of edges around the intersection (assuming a counter-clockwise orientation) is used to order the boundary edges, and by extension order the nodes as well. Each boundary intersection with a region consists of a sequence of edges, two of which belong to the intersecting region. As the region's boundary will be recorded in a counter-clockwise orientation, one edge enters the intersection node, and the other edge exits the intersection node.

For a counter-clockwise boundary orientation the interior of the region is kept to the left-hand side during a traversal, and the exterior is kept to the right-hand side. Given that the boundary intersects with another object, the boundary sequence necessarily consists of additional edges (at least one per intersecting object). Those edges are either to the left of the boundary (interior) or to the right of the boundary (exterior). The boundary edges are elements of the set *boundary_edges* (Algorithm 3.2), and the edges of additional objects are either members of *interior_edges* (Algorithm 3.3) or not members of interior edges (in the exterior). Therefore, by identifying which edges belong in the interior or exterior, the orientation of edges entering and exiting an intersection can be set (Algorithm 3.5).

Algorithm 3.5 Determining the edges of a region R entering/exiting a specific boundary point.

Input: The sets $region\_boundary\_edges$, and $region\_interior\_edges$ for a region R, and the $edge\_sequence$ of a $node$.

Let in_edge and out_edge be empty strings
Let b_pos be an empty ordered list
For each $edge \in edge\_sequence$
    If $edge \in region\_boundary\_edges$
        // Record the position of the boundary edges in the sequence
        $b\_pos.add(edge\_sequence.loc(edge))$
    End if
End For
// If the boundaries are consecutive at the end of the sequence, next element is at index 0
If $(b\_pos[0] + 1 \equiv b\_pos[1]) \wedge b\_pos[1] \ \equiv edge\_sequence[-1]$
    // If the next element is in the interior, then the first boundary edge enters the node,
    // and the second boundary edge exits the node (followed by the interior edge)
    If $edge\_sequence[0] \in region\_interior\_edges$
        $in\_edge = edge\_sequence\big[b\_pos[0]\big]$
        $out_{edge} = edge\_sequence\big[b\_pos[1]\big]$

Else

$in\_edge = edge\_sequence\big[b\_pos[1]\big]$

$out\_edge = edge\_sequence\big[b\_pos[0]\big]$

End If

// If the boundary edges are consecutive (but not at the end of the sequence), the next

// element follows the second boundary edge

Elif $boundary\_pos[0] + 1 \equiv boundary\_pos[1]$

// If the next element is in the interior, then the first boundary edge enters the node,
// and the second boundary edge exits the node (followed by the interior edge)

If $edge\_sequence[b\_pos[1] + 1] \in region\_interior\_edges$

$in\_edge = edge\_sequence\big[b\_pos[0]\big]$

$out\_edge = edge\_sequence\big[b\_pos[1]\big]$

Else

$in\_edge = edge\_sequence\big[b\_pos[1]\big]$

$out\_edge = edge\_sequence\big[b\_pos[0]\big]$

End If

// If the boundary edges are nonconsecutive, check if the edge following the first boundary
// is in the interior

Elif $edge\_sequence[b\_pos[1] + 1] \in region\_interior\_edges$

$in\_edge = edge\_sequence\big[b\_pos[0]\big]$

$out\_edge = edge\_sequence\big[b\_pos[1]\big]$

Else

$in\_edge = edge\_sequence\big[b\_pos[1]\big]$

$out\_edge = edge\_sequence\big[b\_pos[0]\big]$

End If

Return in_edge, out_edge

By knowing the order in which edges enter and exit a node (Algorithm 3.5), it is possible to obtain a consistent ordering of edges around the boundary of a region and the ordering of nodes around the boundary of a region (Algorithm 3.6).

Algorithm 3.6 Ordering the boundary edges and nodes around a region R.

Input: The sets $region\_boundary\_edges$, $region\_interior\_edges$, and $region\_boundary\_nodes$ for a region R.

1. Let $edge\_sequence$ be an empty list
2. Let $node\_sequence$ be an empty list
3. Let $unprocessed\_nodes$ be an empty set
4. Let in_edge, out_edge be empty strings

   // Prime the sequences with an initial boundary edge/node

5. $node\_sequence.add(region\_boundary\_nodes[-1])$
6. $in\_edge, out\_edge\ = Algorithm3.5(region\_boundary\_nodes.pop())$
7. $edge\_sequence.add(out\_edge)$
8. $unprocessed\_bodes = region\_boundary\_nodes$
9. **While** $unprocessed\_nodes$
10.   **For** each $intersection \in region\_boundary\_nodes$
11.   Let $temp\_in$, $temp\_out$ be empty strings
12. $temp\_in, temp\_out\ = Algorithm3.5(intersection)$
            // The edge exiting the previous intersection is the same as the edge
            // entering the next intersection in the sequence
13.   **If** $temp\_in \equiv edge\_sequence[-1]$
14. $edge\_sequence.add(temp\_out)$

                // Following steps could be combined through enumeration, depending on
                // the implementation of the for loop, by popping a specific index

15. $unprocessed\_nodes.remove(intersection)$
16. $node\_sequence.add(intersection)$
17.   **End If**
18.   **End For**
19. **End While**
20. **Return** $edge\_sequence$, $node\_sequence$

The next set of areas, boundaries, and nodes of interest are those that relate to holes. First the areas comprising the holes within a region are found (Algorithm 3.7). While the boundary and intersection sequence for each hole can be found in the same manner as a region, the interior of a region is the difference between its areas and the areas of the holes it hosts (Algorithm 3.8).

Algorithm 3.7 Determining the areas for holes within a region R.

Input: The set of holes in the scene H, a region R.

1. Let $region\_holes$ be an empty set
2. **For** each $hole \in H$
3. **If** $hole.parent \equiv R.id$
4. $temp\_holes = hole.areas$
5. **For** each $area \in temp\_holes$
6. $region\_holes.add(area)$
7. **End For**
8. **End If**
9. **End For**
10. **Return** $hole\_areas$

Algorithm 3.8 Determining the interior of a region R.

Input: A region R, and the set of areas comprising holes within the region, $hole\_areas$.

1. Let $region\_interior$ be an empty set
2. $region\_interior = region.areas \setminus hole\_areas$
3. **Return** $region\_interior\_areas$

3.5.2 Operations on Lines

While the boundary of a region is described by a sequence of edges (and nodes), a line is described by a sequence of interior edges (and nodes), with the first and last nodes in the sequence representing the boundary of the line. Unlike a region, a line does not possess a consistent orientation (left-most point to right-most point, for instance), it simply exists from one end to the other (in this setting). Therefore, a boundary point for a line is a node only shared by a single edge in the sequence, of which there are two (Algorithm 3.9). Similarly, a terminal edge is an edge that connects to at most one other edge in sequence, and all subsequent edges can be placed based on their endpoint nodes (Algorithm 3.10).

Algorithm 3.9 Obtaining the boundary nodes and interior nodes of a line L.

Input: A line L.

Let $line\_boundary\_nodes$ be an empty set
Let *line_interior_nodes* be an empty set
Let $unordered\_nodes$ be an empty list
For each $edge \in L_{edges}$
    $unordered\_nodes.add(edge.node[0])$
    $unordered\_nodes.add(edge.node[1])$
End For
For each $node \in unordered\_nodes$
    $count = unordered\_nodes.count(node)$
    If $count \equiv 1$
        $line\_boundary\_nodes.add(node)$
    Else
        line_interior_nodes.add(*node*)
    End If
End For
Return $line\_boundary\_nodes$, *line_interior_nodes*

Algorithm 3.10 Obtaining the sequence of edges and nodes for a line L.

Input: A line L, the set of *nodes* for L.

```
Let node_sequence be an empty list
Let edge_sequence be an empty list
// Prime the node_sequence with the first boundary point
node_sequence.add(boundary_points[0])
While len(edge_sequence) < len(L.edges)
    For each edge ∈ L.edges
        // If one of the endpoints for an edge is the previous node, the other
        // endpoint is the next node, presuming sequenced edges have been
        // removed.
        If edge.intersections[0] ≡ node_sequence[−1] ∧ edge ∉ edge_sequence
            node_sequence.add(edge.intersections[1])
            edge_sequence.add(edge)
        Elif edge.intersections[1] ≡ node_sequence[−1] ∧ edge ∉ edge_sequence
            node_sequence.add(edge.intersections[0])
            edge_sequence.add(edge)
        End If
    End For
End While
Return node_sequence, edge_sequence
```

For consistency with the sets representing the components of lines, let the set *line_interior_edges* be equal to the set of edges that describe a line.

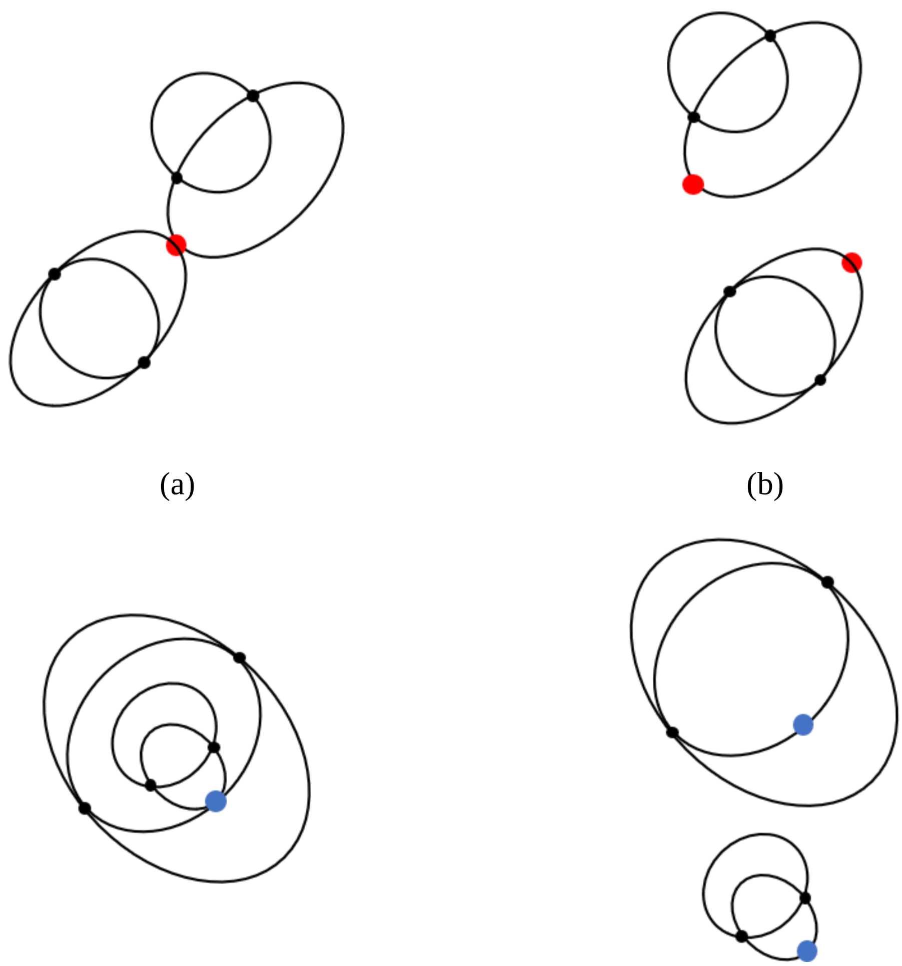
3.5.3 Operations on a Scene

In addition to operations on regions and lines (as well as their constituent components), performing operations on the scene also yields meaningful information. In particular, for a set of input objects, Algorithm 3.11 yields a set of strongly connected areas, that is, it separates the scene components at nodes that are articulation points in a graph representation of the scene (Figures 3.13a and 3.13b).

Algorithm 3.11 Splitting a region-based scene at articulation points.

Input: The set A of all areas within a scene.

Let $adjacent\_areas$ be an empty set
Let $remaining\_areas$ be an empty set
Let $temp\_areas$ be an empty set
$remaining\_areas = areas$
While $remaining\_areas$
    // Prime the first connected set of areas
    $temp\_areas = remaining\_areas.pop()$
    For each $adjacent\_area \in temp\_areas$
        For each $test\_area \in remaining\_areas$
            If $test\_area.edges \cap adjacent\_area.edges \neq \emptyset$
                $temp\_areas.add(test\_area)$
                $remaining\_areas.remove(test\_area)$
            End if
        End For
    End For
    // Once all areas edge-adjacent to the initial area have been added to the temp areas
    // set, add that set as an element to the set $adjacent\_areas$ (a set of sets of adjacent
    // areas). In this fashion, each cluster of adjacent areas is its own element.
    $adjacent\_areas.add(temp\_areas)$
End While
Return $adjacent\_areas$



(a) (b) (c) (d)

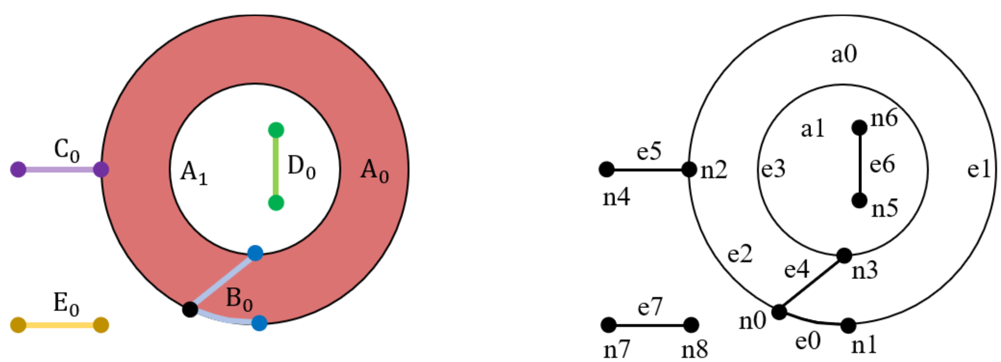
Figure 3.13 Scenes with articulation points split at the node. (a) An articulation point between left and right subgraphs is then split (b) into separate components, and (c) an articulation point between an inner subgraph and an outer subgraph is split (d) into separate components.

Such an operation preserves the boundaries of individual regions. A simple region's boundary will never be split from itself at an articulation point. It also allows nested components within a scene to be treated independently. For a depiction of a scene, for instance, placing scene objects such that there is only a single embedding is difficult.

Ensuring that all elements are visible and appealing is an additional set of considerations. By representing a scene as a graph and splitting its components it may become easier to determine the placement of scene elements, potentially creating a simpler view.

3.6 Satisfying the Hypothesis

It was hypothesized in Section 1.2 that a spatial scene description derived from the sets of areas, edges, and nodes that comprise the scene captures a set of three invariants (intersection dimension, intersection sequence, and complex containment) and that each of those invariants is necessary in tandem to unambiguously describe the scene.. An example scene, decomposed into areas, edges, and nodes, is presented (Figure 3.14) and will be used in the examples going forward.



(a) (b)

Figure 3.14 An input scene decomposed into its components. (a) The objects in the scene and (b) the areas, edges, and nodes in the scene.

Table 3.1 The elements comprising the scene are described using *Scene Notation*.

Regions/Holes/Gaps:

$(\text{collection}, \text{instance}, \text{type}, \text{parent}, \text{areas}, \text{container})$

$A_0: (A, 0, \text{region}, \emptyset, (A_0, A_1), \emptyset)$

$A_1: (A, 1, \text{hole}, A_0, A_1, A_0)$

Lines:

$(\text{collection}, \text{instance}, \text{type}, \text{edges}, \text{container})$

$B_0: (B, 0, \text{line}, (e0, \text{e}4), \emptyset)$

$C_0: (C, 0, \text{line}, \text{e}5, \emptyset)$

$D_0: (D, 0, \text{line}, \text{e}6, A_1)$

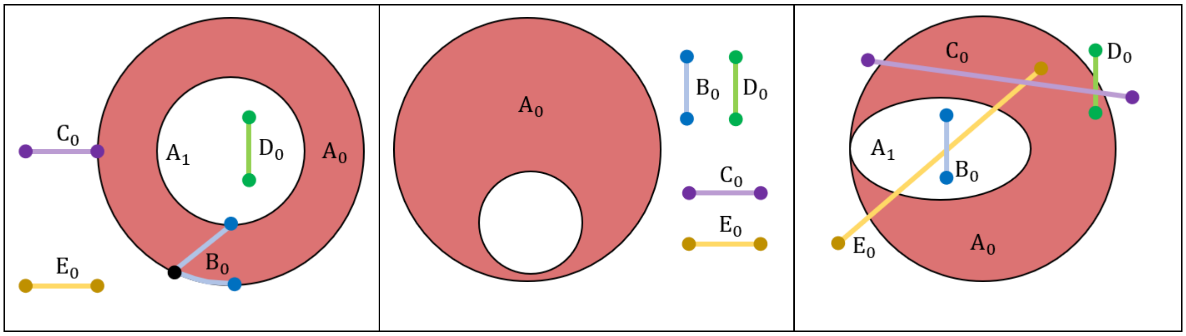
$E_0: (E, 0, \text{line}, \text{e}7, \emptyset)$

Areas:	**Edges:**	**Nodes:**
$(\text{instance}, \text{edges})$	$(\text{instance}, (\text{node}, \text{node}))$	$(\text{instance}, \text{edge\_sequence})$
$\text{a}0: (0, (\text{e}0, \text{e}1, \text{e}2))$	$\text{e}0: (0, (\text{i}0, \text{i}1))$	$\text{n}0: (0, (\text{e}0, \text{e}4, \text{e}2))$
$\text{a}1: (1, \text{e}3)$	$\text{e}1: (1, (\text{i}1, \text{i}2))$	$\text{n}1: (1, (\text{e}1, \text{e}0))$
	$\text{e}2: (2, (\text{i}2, \text{i}0))$	$\text{n}2: (2, (\text{e}1, \text{e}5, \text{e}2))$
	$\text{e}3: (3, (\text{i}3, \text{i}3))$	$\text{n}3: (3, (\text{e}3, \text{e}3, e4))$
	$\text{e}4: (4, (\text{i}0, \text{i}3))$	$\text{n}4: (4, \text{e}5)$
	$\text{e}5: (5, (\text{i}2, \text{i}4))$	$\text{n}5: (5, \text{e}6)$
	$\text{e}6: (6, (\text{i}5, \text{i}6))$	$\text{n}6: (6, \text{e}6)$
	$\text{e}7: (7, (\text{i}7, \text{i}8))$	$\text{n}7: (7, \text{e}7)$
		$\text{n}8: (8, \text{e}7)$

The example scene (Figure 3.14) is described with the *Scene Notation*. The scene consists of six objects: a region (A_0), a hole within that region (A_1), and four lines ($\mathrm{B}_0, \mathrm{C}_0, \mathrm{D}_0, \mathrm{E}_0$). Each object's identity (collection and instance) is captured, as well its elements (areas for regions and holes, edges for lines), and the explicit containment relations of each object (where each object is contained). The components that form each object are also described in detail (areas are constructed from edges, edges are constructed from pairs of nodes, and nodes are surrounded by a sequence of edges).

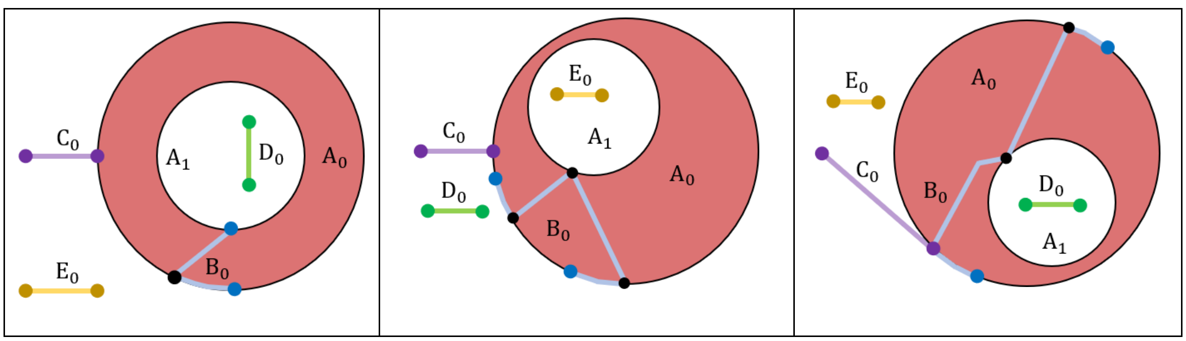
Together the components captured by the Scene Notation describe the construction of the example scene (Figure 3.14)—but the question remains as to whether the invariants of dimension, containment, and intersection produce an unambiguous scene. To answer this question, consider four sets of scenes (Figures 3.15, 3.16, 3.17, and 3.18), each missing one or more components of the *scene notation*. By generating multiple incomplete representations, it is shown that each scene representation requires properties captured by the other representations in order to capture the original scene, without introducing ambiguity.

The first modified scenes (Figure 3.15) lack not only intersection sequence, but any notion of intersection or containment—they are simply a collection of objects of varying types. It does not matter where each object is placed because nothing other than object identity is captured. By incorporating coarse intersection (e.g., the 9-intersection), the set of modified scenes (Figure 3.16) are closer to the original (Figure 3.14) but there are discrepancies: without explicit containment it is impossible to determine which object belong in the hole. Additionally, the boundary of A_0 is segmented differently in each example, with different numbers of intersection present. A coarse approach is insufficient to model the original scene.



(a) (b) (c)

Figure 3.15 Three scenes that are equivalent only when intersection dimension, intersection sequence, and explicit containment are removed from the notation.



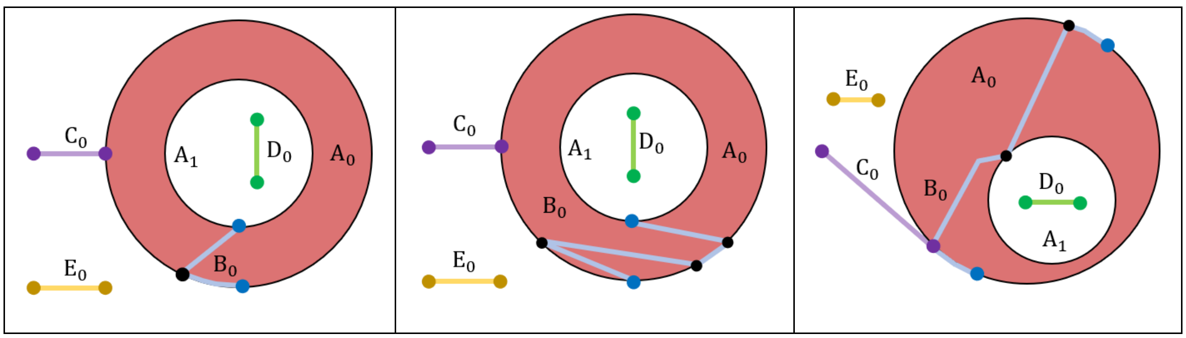
(a) (b) (c)

Figure 3.16 Three scenes that are equivalent when only intersection dimension is modeled.

The third modified scene (Figure 3.17) includes the explicit containment information, but still lacks intersection sequence details. Although he lines D_0 and E_0 are properly placed in the hole A_1 and outside of A_0 (respectively), the boundary of A_1 is still partitioned into varying numbers of edges even if it properly has intersections with both A_0

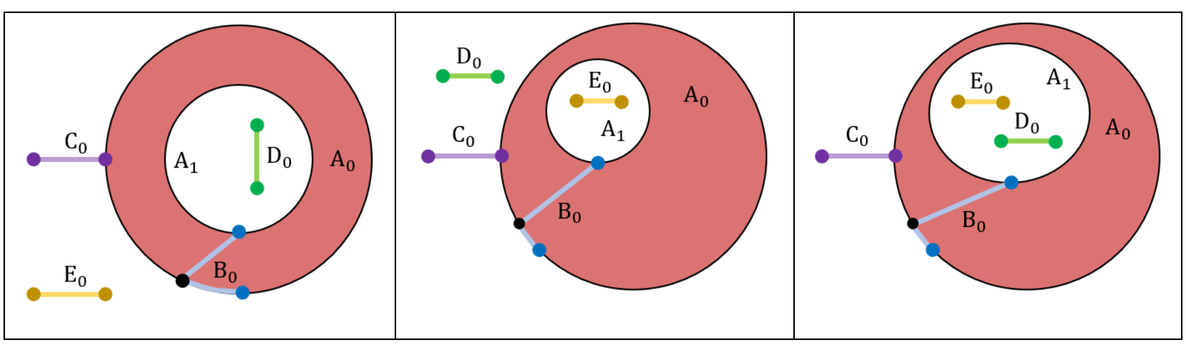
and the newly identifiable A_1 since the number and dimension of intersections is still a missing property captured through Scene Notation's intersection sequence.

The final modified scene swaps an exclusive representation of explicit containment for an exclusive representation of intersection sequence (Figure 3.15). In this example the boundary of A_1 is properly represented as consisting of three edges, but while E_0 and D_0 are contained in the exterior, E_0 is again incorrectly placed within a hole in two of the cases.



(a) (b) (c)

Figure 3.17 Three scenes that are equivalent when only intersection dimension and explicit containment are captured, but not sequence.



(a) (b) (c)

Figure 3.18 Three scenes that are equivalent when only intersection dimension and intersection sequence are captured, but not explicit containment.

With only coarse intersections, a model would likely be as descriptive as the unmodified 9-intersection—and unable to represent intersection sequence or dimension, to capture the placement of objects within specific partitions of space, or even to represent how many partitions exist between objects (Figure 3.16). By only adding explicit containment to the description of a scene, the partitions into which objects are divided are captured, but containment is unable to account for how many times objects intersect or the order in which those intersections occur, or the dimensions of those intersections (Figure 3.17). Only by adding the intersection sequence to intersection dimension does the structure of a scene of objects begins to emerge, but there is still no accounting for subdivisions of space within a given object (Figure 3.18).

Therefore, in satisfaction of the hypothesis, neither intersection sequence, intersection dimension, nor explicit containment alone are enough to unambiguously model a scene. Similarly, if any combination of properties lacks a single one of the three, the description of a scene may have multiple interpretations, leading to ambiguity.

3.7 Summary

The Scene Notation captures the boundary intersections between collections of points, lines, and regions within a spatial scene, including both the ordering of the intersections around a specific object, and the ordering of objects around a specific intersection point. It also contains information regarding the components of complex objects, such as the location of holes and separations, and where within a scene each object is embedded.

These properties are used to derive information about the structure of a scene, as well as enabling the construction of new objects through set operations. The resulting notation describes a scene related to the input based on the properties of intersection sequence, intersection dimension, and complex containment.

CHAPTER 4

SCENE RELATIONS

Scene Notation (Chapter 3) captures (1) the explicit contain relations between objects, (2) the sequence of intersections around the boundaries of objects and the extent of lines, and (3) the sequence of edges entering and exiting each intersection node. While these properties explicitly represent the structure of scene objects, they are now leveraged to relate objects to one another, providing the foundation for analyzing the relations between the objects of a scene.

Each boundary intersection node captures the sequence of edges (in a counterclockwise orientation) that connect to the node. By capturing which edges belong to a region's boundary it is possible to derive which edges are within a region's closure, and which exist outside of the region (Section 3.5.1). By maintaining a consistent representation of what is inside, outside, or coincident to a region's boundary at a given node, primitive *touching* and *crossing* relations are derived for all possible configurations of edges. The case of relating lines is slightly different. While it is possible to differentiate either side of a directed line, the interior is the space defined between the two endpoint nodes, not an area to one side of the line. All valid sequences of edges at a given intersection node for two regions (Section 4.1), two lines (Section 4.2), a region with a line, and a line with a region (Section 4.3) are detailed; these sequences are the set of *local relations* (as they only apply to individual intersection points, rather than entire objects).

The local relations are mapped onto individual 9-intersection matrices, which when joined under union result in a 9-intersection matrix that describes the objects being related.

Additionally, scene notation can be directly mapped on a 9-intersection matrix (Section 4.4) and the relations *contains*, *inside*, *equal*, and *disjoint* that do not involve boundary intersection are considered using various set operations, and a set of *surrounds* relations between an object and a holed region (or the union of objects that form a gap) (Section 4.5).

The local relations can also be sequenced, resulting in a description that captures how an object relates to other scene objects through its boundary intersection nodes in the case of a region (i.e., it *meets* one object twice, a second object *crosses* into it, it *covers* a third object, the second object *crosses* back out, it *meets* a fourth object, and the sequence repeats) or by describing the sequence of nodes from one endpoint to the other in the case of a line (Section 4.6).

4.1 Local Region-Region Relations

As the relations between lines and regions are explored, each will be represented diagrammatically. In the figures that follow, an intersection point between two objects is represented by the sequence of edges that share that node. If one of the objects is a region, that region has a consistently oriented boundary (counter-clockwise) (Section 3.5.1).

In Figure 4.1 the boundary edges of a region A are denoted by a double red line, with a large arrowhead. The arrow points to the left to indicate the counter-clockwise orientation of the region's boundary. Assuming the other object (thin blue line) is also a region, there are eight possible sequences of edges between them, before accounting for the orientation of the second region.

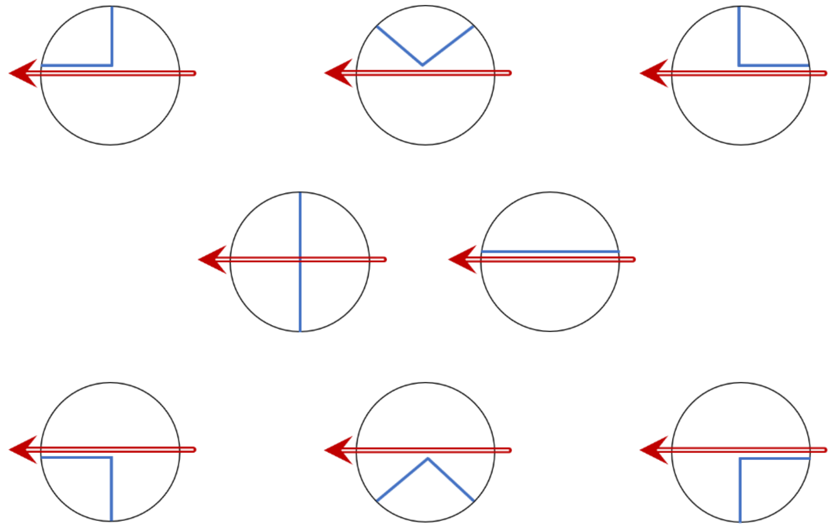
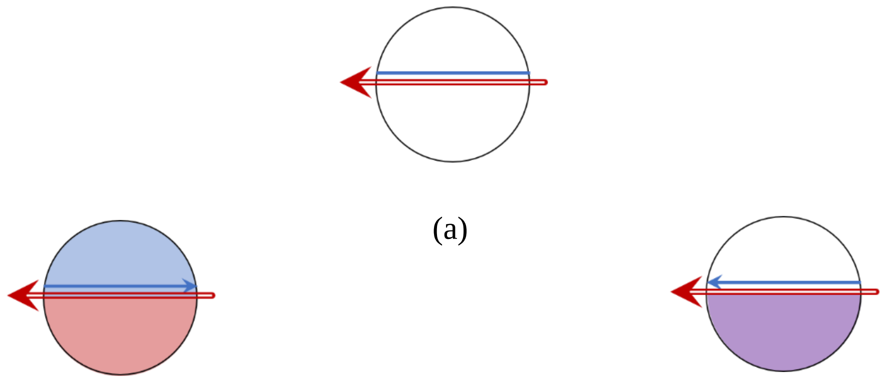


Figure 4.1 The eight prototypical edge sequences between two regions that intersect at a node.

Of importance is that the area to the left of a region's boundary (with a counter-clockwise orientation) comprises the interior of the region. The area to the right of the boundary comprises the exterior of the region. In this manner a sequence of edges (Figure 4.2a) has multiple possible interpretations, depending on how the second object is oriented (Figures 4.2b and 4.2c).

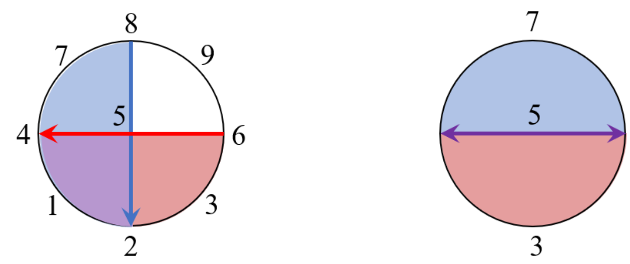


(a)

(b) (c)

Figure 4.2 A sequence of edges (a) and two possible interpretations. (b) The blue edge has an opposing orientation than the red edge (their interiors are on opposite sides of the edge) and (b) the blue edge has a similar orientation to the red edge (their interiors are coincident and their exteriors are coincident).

Depending on where the interior and exterior of a pair of objects lie, the intersections between their interiors, exteriors, and boundaries may be empty or non-empty. This property is commonly represented through the 9-intersection matrix (or similar approach), and such a representation is possible for each sequence of edges meeting at a node (Figure 4.3).



(a) (b)

$$\begin{pmatrix} \neg\emptyset & \neg\emptyset & \neg\emptyset \\ \neg\emptyset & \neg\emptyset & \neg\emptyset \\ \neg\emptyset & \neg\emptyset & \neg\emptyset \end{pmatrix} \qquad \begin{pmatrix} \emptyset & \emptyset & \neg\emptyset \\ \emptyset & \neg\emptyset & \emptyset \\ \neg\emptyset & \emptyset & \emptyset \end{pmatrix} \qquad \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

(c) (d) (e)

Figure 4.3 Two scenarios with intersecting regions. (a) One region *overlaps* another at an intersection point and (b) one region *meet-edge* another at a different intersection point. Both scenarios map onto a 9-intersection matrix: (c) the *overlap* matrix and (d) an *attach* matrix are produced (e) through a matrix that relates intersection components to matrix cells.

By taking the union of each matrix for all intersections between a pair of objects the coarse 9-intersection relation can be derived. Some matrices may immediately represent a valid coarse relation (Figure 4.3c), while others only resolve to the correct matrix when multiple intersections are considered (Figure 4.3d).

By combining all possible orientations between boundary edges, 16 *local relations* between two regions are derived, along with their 9-intersection strings:

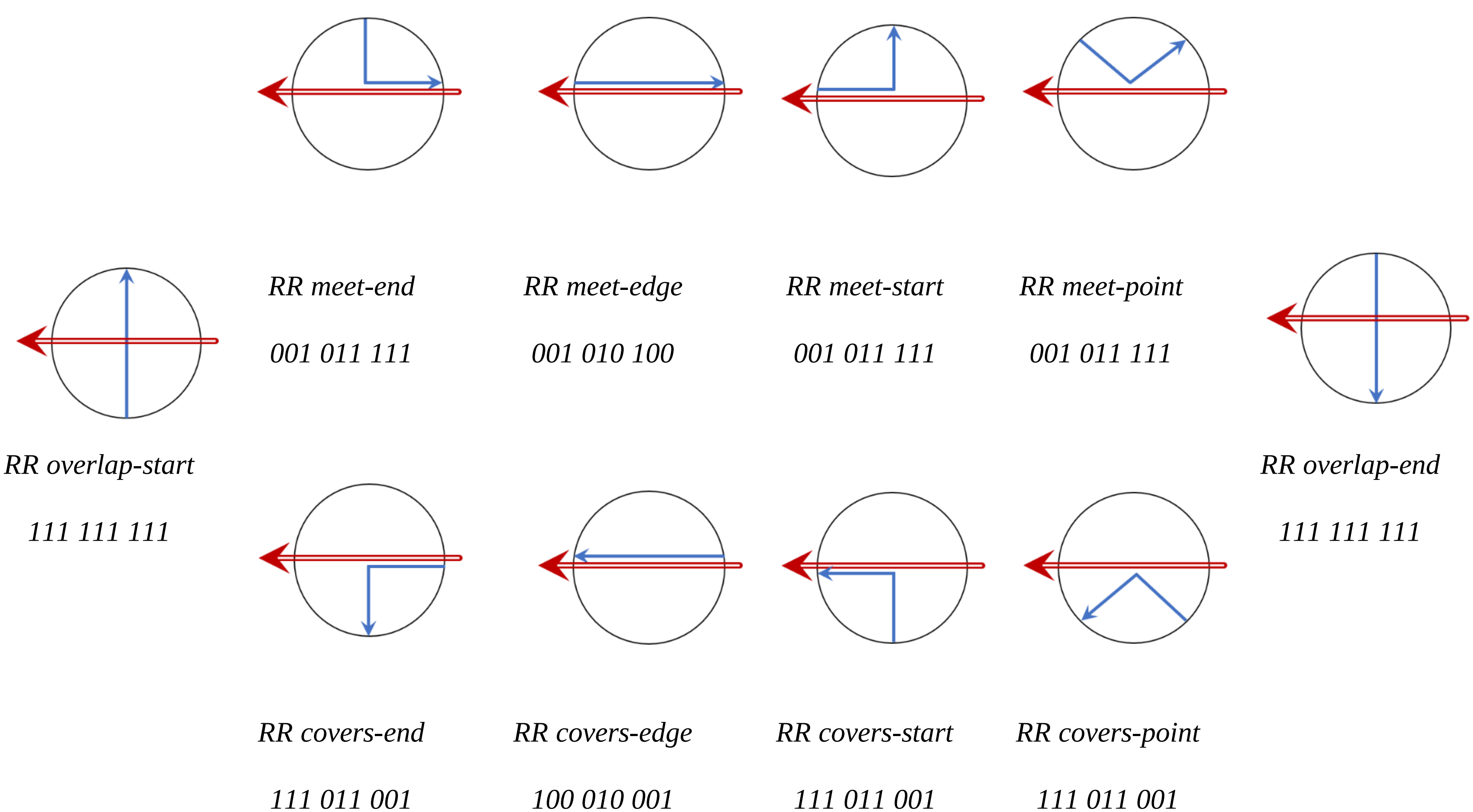


Figure 4.4 Ten of sixteen local region-region relations, representing different *meet*, *covers*, and *overlap* scenarios.

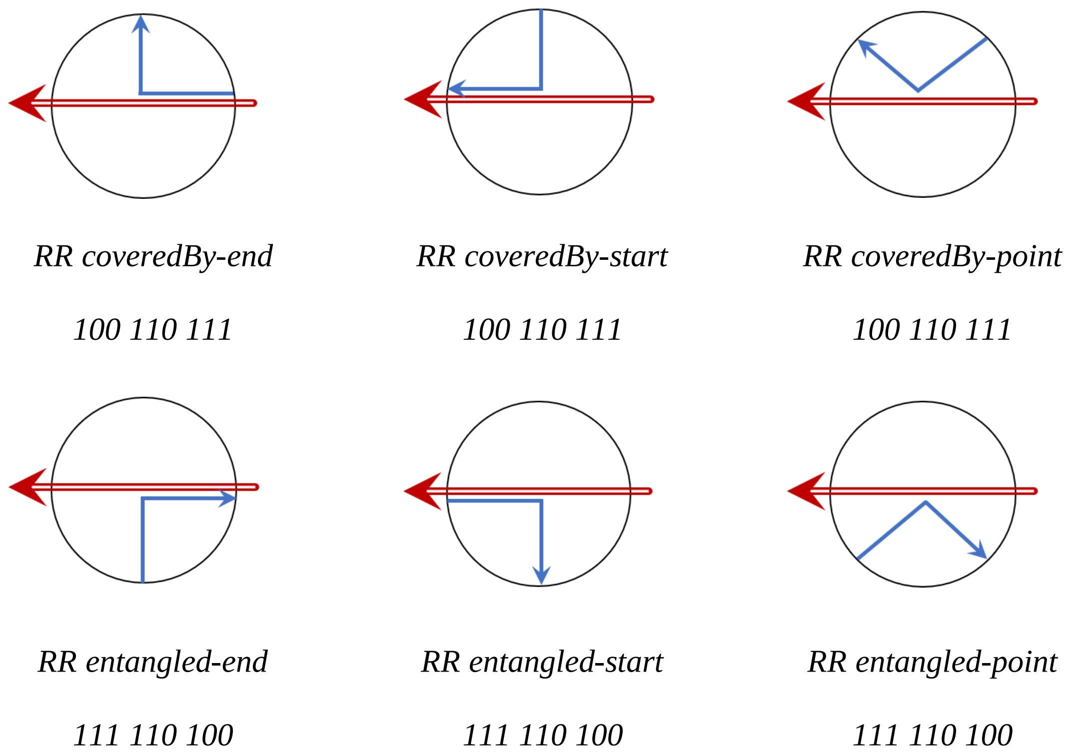


Figure 4.5 Six additional local region-region relations, representing *coveredBy* and *entangled* scenarios. When combined with more than local relation, *entwined* resolves to overlap in this setting.

While the previous local relations exist between two objects, a single node sequence can accommodate any number of additional edges (Figure 4.6a). By considering only the edges belonging to a pair of objects (Figures 4.6b, 4.6c, and 4.6d), its local relation can be derived.

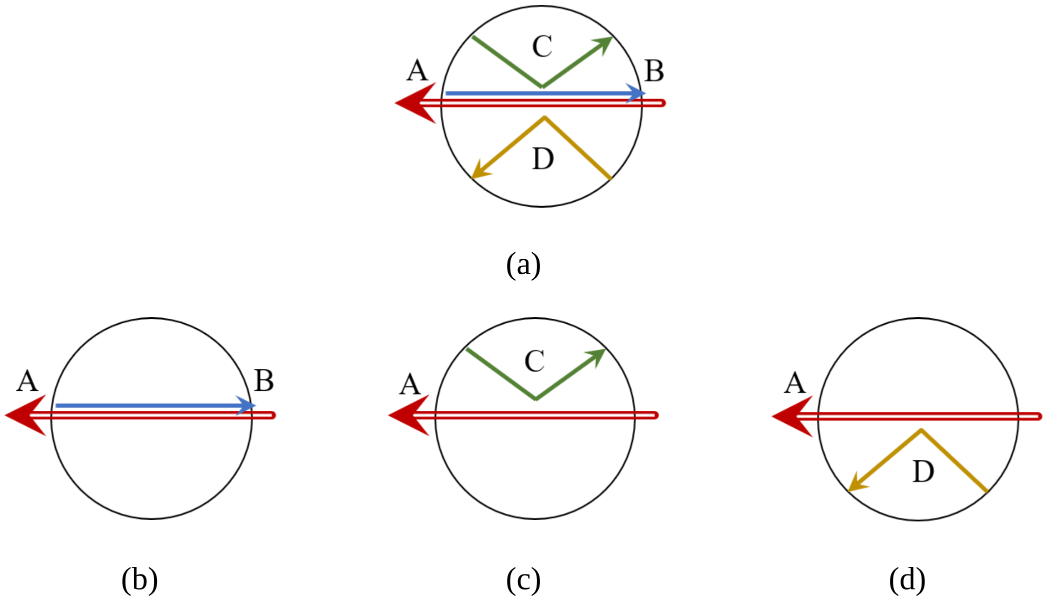


Figure 4.6 Four representations of the same boundary node. (a) Multiple edges come together in sequence, corresponding to the boundaries of four regions, (b) A *meet-edge* B, (c) A *meet-point* C, and (c) A *covers-point* D.

A sequence of edges surrounding an intersection node is also a concept that can easily be applied to lines, with an addition set of line-line local relations being developed next.

4.2 Local Line-Line Relations

For the set of region-region relations, the orientation of the boundary for each object determined the location of its interior and exterior. However, for a line the boundary is a pair of endpoint nodes, and the interior is every edge between them. For a given line the choice of start node and end node is immaterial, but when traversing the extent of a line a choice must be made and adhered to.

In this sense, a line is directed from one endpoint to the other, even if the choice of its initial endpoint is insignificant. This approach allows for intersecting edges to be placed on one side of the line or the other. One line can also begin or end while intersecting with another line (Figures 4.7a and 4.7b).

(a) (b)

Figure 4.7 Two distinct edge sequences. (a) One line begins within the interior of a second line and (b) one line touches the interior of another line and ends.

This directedness results in a set of 20 line-line *local relations*, four more than those between pairs of regions. Region-region distinctions, such as *covers* versus *coveredBy*, *meet* versus *equal*, and the beginning and ending of an *overlap* relation are all abstracted away, being replaced with a set of line-specific endpoint relations.

Unlike regions, lines have exactly two boundary nodes. When abstracting a collection of line-line *local relations* to their 9-intersection analogs, some (or all) of those boundary points may not be captured (e.g., the lines only intersect in their interiors). In such cases the boundary-exterior or exterior-boundary intersections of the 9-intersection matrix would be nonempty, depending on which line (or lines) contains boundary nodes that are not part of a *local relation* intersection.

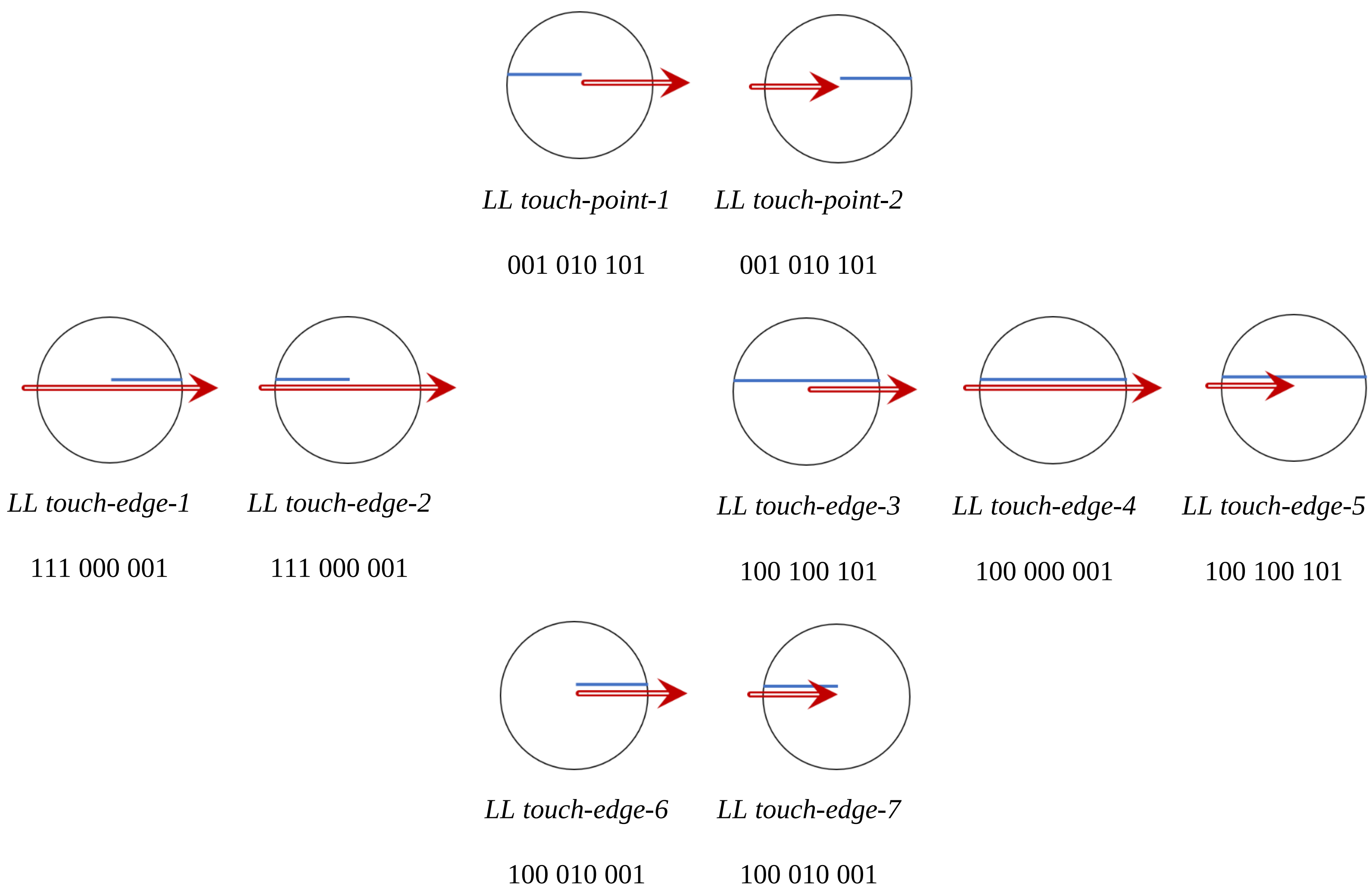


Figure 4.8 Nine of the twenty local line-line relations.

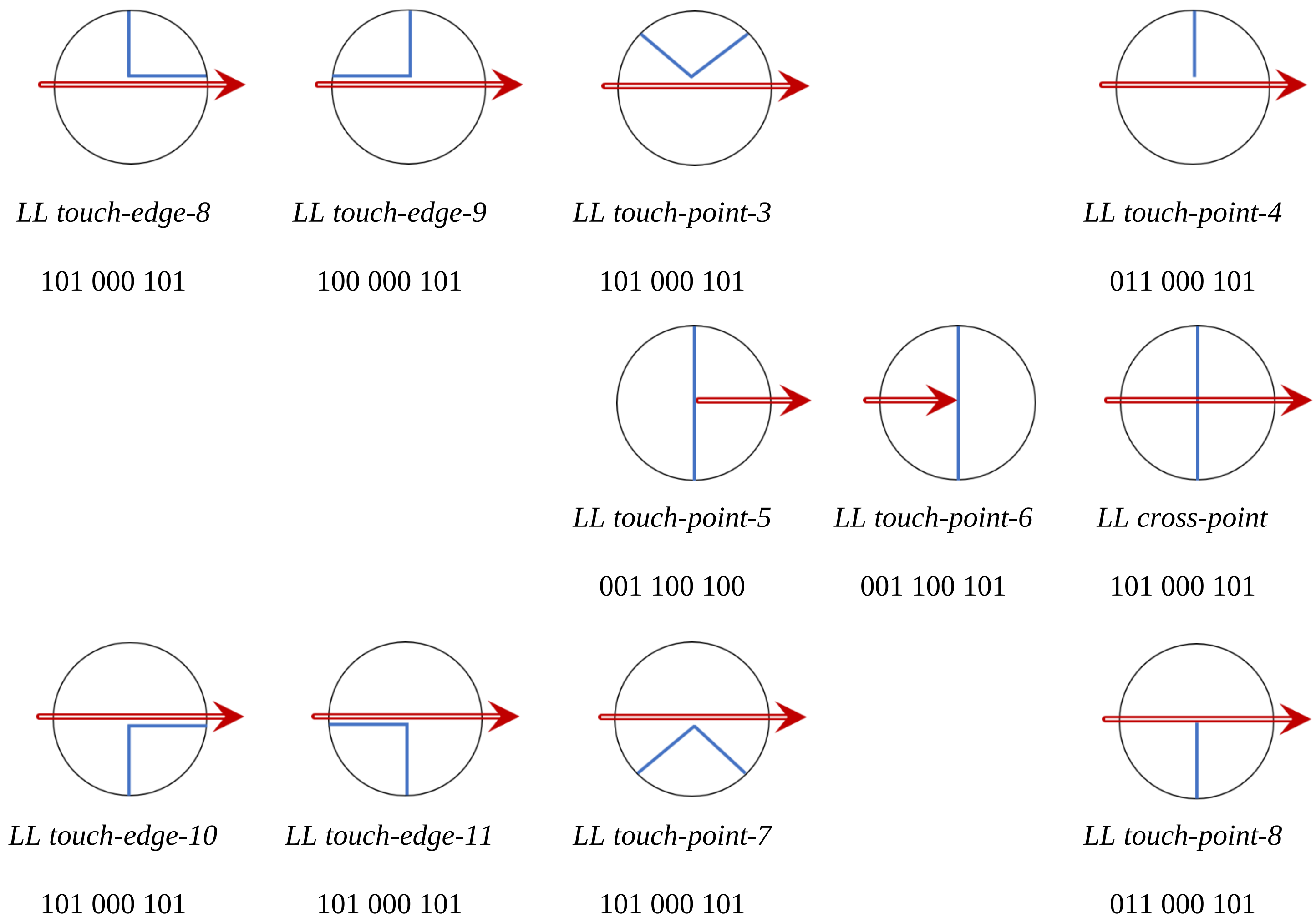


Figure 4.9 Eleven of the twenty local line-line relations.

4.3 Local Region-Line and Line-Region Relations

The local relations between lines and regions feature both the interior-exterior orientation of region objects, and the start-to-end directedness of the line relations. These sets of relations contain differing numbers of elements; there are 12 relations between a region and a line, but 24 relations between a line and a region. In totality, there are 12 region-line relations, 16 region-region relations, 20 line-line relations, and 24 line-region relations.

These *local relations* represent how a target object B impacts the boundary of a source object A. They are derived by traversing the boundary of the source object until an intersection with the target object is found. Therefore, the source object gains a bias compared to the target object. For example, the smallest set of relations, region-line, incorporates the properties of a single region (its boundary is traversed in a counter-clockwise orientation with interior on the left and exterior on the right), but directedness of the intersecting line is unrepresented. Since the approach is biased toward the source object, whether region A's boundary intersects with an endpoint of a line B can be determined, but the choice of endpoint is trivial. Asserting the start or end of a line is only useful when traversing that line consistently; if the source object is a region, the directedness of the line is indeterminant because it is nonexistent.

On the other hand, with the line-region relations the directedness of the line source object must be accommodated (including distinguishing its start and end, as well as distinguishing one side from the other—a property inherited from this directedness). The orientation of the region must also be preserved (making this the most detailed set of relations), else whether the line is within or outside of the region is unclear.

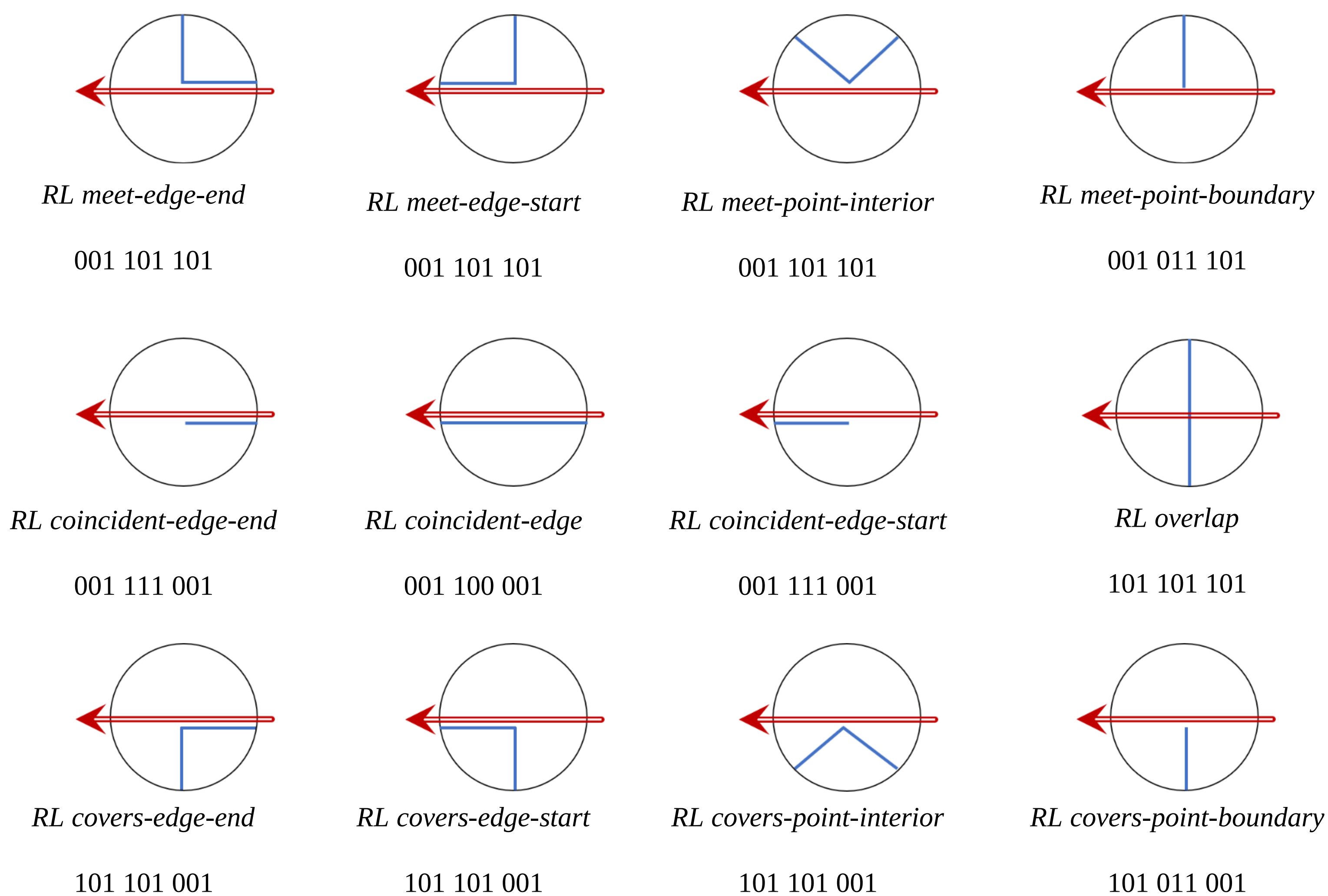


Figure 4.10 The complete set of twelve local region-line relations.

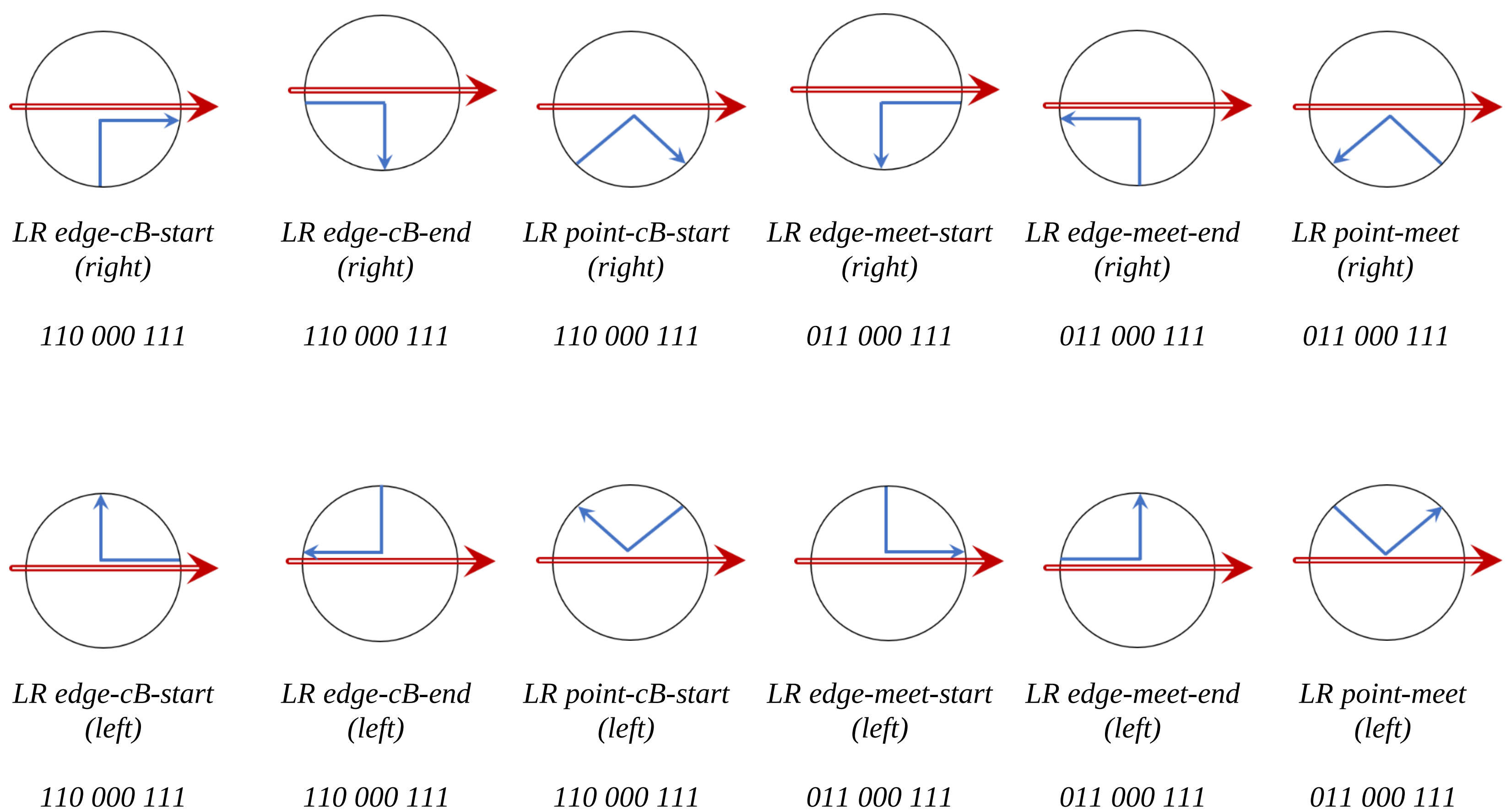


Figure 4.11 Twelve of the twenty-four local line-region relations.

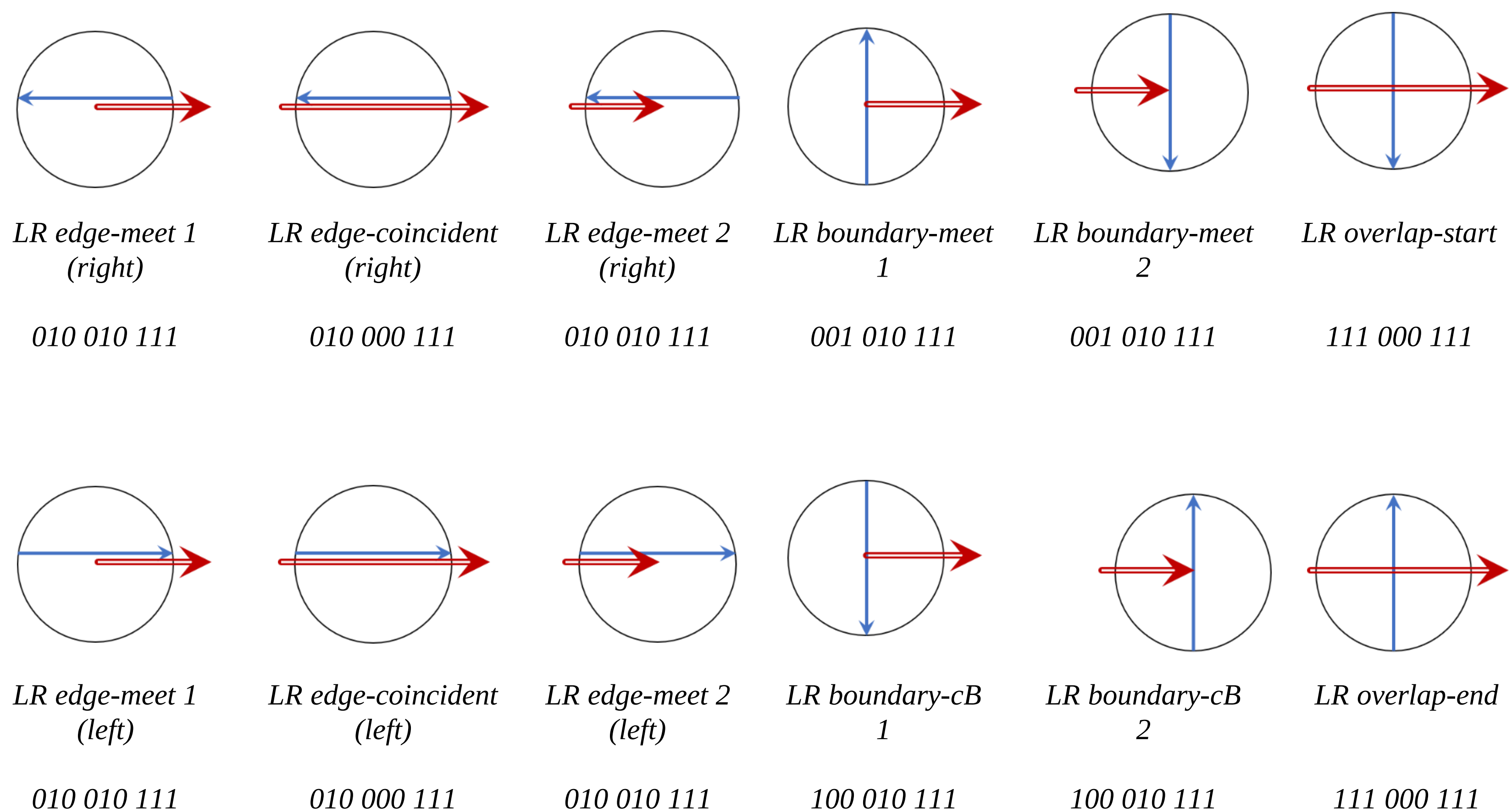


Figure 4.12 The final twelve of the twenty-four local line-region relations

4.4 9-Intersection Matrices

While the 72 local relations address cases with boundary-boundary intersections, they are insufficient to handle relations between objects that do not involve explicit intersections between region boundaries or lines, such as when one region *contains* another region, or a line is fully *inside* of a region. They also do not handle cases where the boundary of a line does not participate in an intersection. Using the *Scene Notation*, a 9-intersection matrix may be derived between pairs of objects regardless of these limitations. This holds for both region-like objects (regions, holes, and gaps) as well as lines, in combination.

Constraints for these pairings are detailed in this section and depend on whether the resulting sets are empty or nonempty between an object A and an object B (Definitions 4.1-4.3). These constraints are based upon the following sets introduced in Section 3.4-3.5), reintroduced together for clarity but without their individual derivations:

Table 4.1 The properties and derived components that describe a line:

container:	The area that contains an object
line_interior_edges:	The set of edges that partition a line.
line_interior_nodes:	The set of nodes that represent the intersection of adjacent elements of *line_interior_edges*.
line_boundary_nodes:	The set of nodes that do not represent the intersection of adjacent elements of *line_interior_edges*.

Table 4.2 The properties and derived components that describe a region:

container:	The area that contains an object
region_interior_areas:	The set of areas that partition a region.
region_interior_edges:	The set of edges that represent the intersection of adjacent elements of *region_interior_areas*.
region_interior_nodes:	The set of nodes that bound elements of *region_interior_edges* that are not a subset of *region_boundary_nodes*.
region_boundary_edges:	The set of edges that do not represent the intersection of adjacent elements of *region_interior_areas*.
region_boundary_nodes:	The set of nodes that bound elements of *region_boundary_edges*.

The set operations described in Definitions 4.1-4.3 between a pair of objects' (regions, lines, or mixed) interiors, boundaries, and container property provide a mapping onto the relations of the 9-intersection matrix (Equation 2.1). Exploring this, let A and B be a pair of objects within a spatial scene, consisting of any combination of regions or lines. Table 4.1 describes the sets the scene notation captures for a single line, derived from the edges and nodes that partition the object (Section 3.5.2). Table 4.2 describes the sets the scene notation captures for a single region, derived from the areas, edges and nodes that partition the object (Section 3.5.1).

The 9-intersection matrix (Equation 2.1) captures the content invariant property between the interiors, boundaries, and exteriors of two objects. The Scene Notation separates region interiors into areas, edges, and nodes; separates region boundaries into edges and nodes; separates line interiors into edges and nodes; represents line boundaries as nodes; and does not capture the exterior explicitly for either object type. These sets fully partition the extent of each object, through decomposition.

By taking the intersections between the interior sets and boundary sets of object A and object B the *InteriorInterior*, *InteriorBoundary*, *BoundaryInterior*, and *BoundaryBoundary* intersections are captured directly. The *InteriorExterior*, *BoundaryExterior*, *ExteriorExterior*, *ExteriorInterior*, and *ExteriorBoundary* sets are not capturable through intersection because the exterior is not captured as a set of areas, edges, and nodes.

These exterior intersections, however, may be described through set difference. Let the interior of A be represented with a collection of areas (if a region), edges, and nodes. Similarly let the interior of B be represented with a collection of areas (if a region), edges, and nodes. Let the boundary of A be represented with a collection of edges (if a region), and nodes. Similarly let the interior of B be represented with a collection of edges (if a region), and nodes.

If the union of A’s various interior sets minus the union of B’s interior sets and boundary sets is nonempty then there is a component of A’s interior that is not in B’s interior or boundary—therefore the component of A’s interior is external to B.

Similarly, if the union of A’s various boundary sets minus the union of B’s interior sets and boundary sets is nonempty then there is a component of A’s boundary that is not in B’s interior or boundary—therefore the component of A’s boundary is external to B.

These operations may be reversed for A and B to show that the converse it true—that the interior of B is external to A and that the boundary of B is external to A, if those sets are nonempty. Finally, as neither A or B can fill the embedding space fully, nor their union (Chapter 3.1), the simple existence of both object A and B ensure that their exterior-exterior intersection is nonempty, allowing the representation of each of the 9-intersections.

Definition 4.1 Constructing the region-region matrix:

$IntInt(A,B)$:	$A_{region\_interior\_areas} \cap B_{region\_interior\_areas} = \neg\emptyset$
$IntBound(A,B)$:	$A_{region\_interior\_edges} \cap B_{region\_boundary\_edges} = \neg\emptyset$
$IntExt(A,B)$:	$A_{region\_interior\_areas} \setminus B_{region\_interior\_areas} = \neg\emptyset$
$BoundInt(A,B)$:	$A_{region\_boundary\_edges} \cap B_{region\_interior\_edges} = \neg\emptyset$
$BoundBound(A,B)$:	$\left(A_{region\_boundary\_nodes} \cap B_{region\_boundary\_nodes} = \neg\emptyset\right)$ $\vee \left(A_{region\_boundary\_edges} = B_{region\_boundary\_edges} = \neg\emptyset\right)$
$BoundExt(A,B)$:	$A_{region\_boundary\_edges}$ $\setminus (B_{region\_boundary\_edges} \cup B_{region\_interior\_edges}) = \neg\emptyset$
$ExtInt(A,B)$:	$B_{region\_interior\_areas} \setminus A_{region\_interior\_areas} = \neg\emptyset$
$ExtBound(A,B)$:	$B_{region\_boundary\_edges}$ $\setminus \left(A_{region\_boundary\_edges} \cup A_{region\_interior\_edges}\right) = \neg\emptyset$
$ExtExt(A,B)$:	$(A \ \cap \ B) = \neg\emptyset$

Definition 4.2 Constructing the line-line matrix:

$IntInt(A,B)$:	$\left(A_{line\_interior\_edges} \cap B_{line\_interior\_edges} = \neg\emptyset\right)$ $\vee \left(A_{line\_interior\_nodes} \cap B_{line\_interior\_nodes} = \neg\emptyset\right)$
$IntBound(A,B)$:	$A_{line\_interior\_nodes} \cap B_{line\_boundary\_nodes} = \neg\emptyset$
$IntExt(A,B)$:	$\left(A_{line\_interior\_edges} \setminus B_{line\_interior\_edges} = \neg\emptyset\right)$ $\wedge \left(A_{line\_interior\_nodes} \setminus B_{line\_interior\_nodes} = \neg\emptyset\right)$
$BoundInt(A,B)$:	$A_{line\_boundary\_nodes} \cap B_{line\_interior\_nodes} = \neg\emptyset$
$BoundBound(A,B)$:	$A_{line\_boundary\_nodes} \cap B_{line\_boundary\_nodes} = \neg\emptyset$
$BoundExt(A,B)$:	$A_{line\_boundary\_nodes}$ $\setminus \left(B_{line\_boundary\_nodes} \cup B_{line\_interior\_nodes}\right) = \neg\emptyset$
$ExtInt(A,B)$:	$\left(B_{line\_interior\_edges} \setminus A_{line\_interior\_edges} = \neg\emptyset\right)$ $\wedge \left(B_{line\_interior\_nodes} \setminus A_{line\_interior\_nodes} = \neg\emptyset\right)$
$ExtBound(A,B)$:	$B_{line\_boundary\_nodes}$ $\setminus \left(A_{line\_boundary\_nodes} \cup A_{line\_interior\_nodes}\right) = \neg\emptyset$
$ExtExt(A,B)$:	$(A \cap B) = \neg\emptyset$

Definition 4.3 Constructing the line-region matrix:

$IntInt(A,B)$:	$\left(A_{line\_interior\_edges} \cap B_{region\_interior\_edges} = \neg\emptyset\right)$ $\vee \left(A_{line\_interior\_nodes} \cap B_{region\_interior\_nodes} = \neg\emptyset\right)$ $\vee \left(A_{container} \cap B_{region\_interior\_areas} = \neg\emptyset\right)$
$IntBound(A,B)$:	$\left(A_{line\_interior\_nodes} \cap B_{region\_boundary\_nodes} = \neg\emptyset\right)$ $\vee \left(A_{line\_interior\_edges} \cap B_{region\_boundary\_edges} = \neg\emptyset\right)$
$IntExt(A,B)$:	$\left(A_{line\_interior\_edges} \setminus B_{region\_interior\_edges} = \neg\emptyset\right)$ $\wedge \left(A_{line\_interior\_nodes} \setminus B_{region\_interior\_nodes} = \neg\emptyset\right)$ $\wedge \left(A_{container} \setminus B_{region\_interior\_areas} = \neg\emptyset\right)$
$BoundInt(A,B)$:	$\left(A_{line\_boundary\_nodes} \cap B_{region\_interior\_nodes} = \neg\emptyset\right)$ $\vee \left(A_{container} \cap B_{region\_interior\_areas} = \neg\emptyset\right)$
$BoundBound(A,B)$:	$A_{line\_boundary\_nodes} \cap B_{region\_boundary\_nodes} = \neg\emptyset$
$BoundExt(A,B)$:	$\left(A_{line\_boundary\_nodes} \setminus \left(B_{reg\_boundary\_nodes} \cup B_{reg\_interior\_nodes}\right) = \neg\emptyset\right)$ $\wedge \left(A_{container} \setminus B_{region\_interior\_areas} = \neg\emptyset\right)$
$ExtInt(A,B)$:	$(A \cap B) = \neg\emptyset$
$ExtBound(A,B)$:	$(A \cap B) = \neg\emptyset$
$ExtExt(A,B)$:	$(A \cap B) = \neg\emptyset$

The location relations describe the 9-intersection relations for specific configurations of lines and regions. For example, the local relations *LL touch-edge-6* (Figure 4.8), *LL touch-point-3* (Figure 4.9), and *LL touch-point-2* (Figure 4.8) have the following 9-intersection matrices, in that order: 100 010 001, 101 000 101, 001 010 101—with True and False values represented by 1 and 0, respectively. By taking the union of these matrices (Sections 4.1-4.2), the matrix 101 010 101 is returned. In this manner the coarse 9-intersection relation is derived from the set of local relations, rather than the raw scene notation. This approach also works between pairs of lines and pairs of regions, but not for all circumstances.

An example of a configuration that is not supported by *local relations* alone, when a line lacks any intersections its *container* property is used to determine if it has *InteriorInterior* and *BoundaryInterior* intersections with a region (Definition 4.4). A similar process enables the content of these intersections to be determined between pairs of regions that do not share boundary intersections (Definition 4.1) or pairs of lines that do not intersect at their boundaries (Definition 4.2). In this manner objects in a scene may be relation to the coarse relations of the 9-intersection, even when the local relations are insufficient.

4.5 Surrounds Relations

Beyond *local relations* and 9-intersection relations, the relations between an object and a region with a hole are considered. If an object exists within a hole, the holed region *surrounds* the object. Four basic *surrounds* relations are developed here, inspired by the originally derived *surrounds* relations (Dube and Egenhofer 2014), with a converse relation added for each. However, these relations exclude the *surroundsSplitPocket* relation (and its potential converse) which is a subset of *surroundsMeet*.

The scene notation requires the use of both the intersection-based relations (the 9-interesection relations, for instance (Section 4.4)) and the *container* property that all objects share (Chapter 3.4). The container is the area (partition) that an object sits within, and as such each object can have at most a single container regardless of how various objects are nested. An object may also lack a container if the object completely fills a hole or gap (for instance)—the boundary of the object equals the boundary of the area and cannot be contained within it. This property is utilized to determine relation *surroundsAttach*.

Theorem 4.1 Let X be a hole or a gap, per Definitions 3.3 and 3.4, respectively. Let Y be the host of X, either a region or a union of objects that bound X. Let A be a simple region or a simple line. The 9-intersection relation between A and X (i.e., *meet*) is the *surrounds* relation between Y and A (i.e., *surroundsMeet*) if and only if the *container* (Definition 3.8) of A is X or empty.

Proof. The Scene Notation decomposes lines, region, gaps, and holes into interior and boundary components (Section 3.4-3.5). Using Scene Notation, the relation between these objects is describable through the 9-intersection, resolving to a specific matrix (Definitions 4.1-4.3).

As a hole or gap is within its host, any object within the hole or gap is also within the host; containment is transitive. Therefore, it is possible to distinguish between an object that has an intersection with the host *via* a hole *vis a vis* an intersection from an object within the host, overlapping the host, or external to the host.

The boundary of a hole or gap is also a boundary of the host (Section 3.1). Any intersection between the boundary of A and the boundary of X also indicates an intersection between the boundary of A and the boundary of Y.

If the relation between A and X is *meet* or *attach*—relations with boundary intersections—the relation between Y and A is *SurroundsMeet* or *SurroundsAttach*, respectively, since the boundary of X is the boundary of Y and the exterior of X is the exterior of Y.

When an object lacks boundary intersections it may still be placed within the scene using its container property, which indicates which area serves as its container (Section 3.4). X is a region-like object, defined as a set of areas. If there are no boundary intersections between A and X, but the container of A is a subset of X then A is inside the space bounded by X. However, since X represents a bounded exterior and not a bounded interior (like a region) the converse of the relation is taken, swapping interior for exterior in the matrix. This produces the *disjoint* relation. If A is thusly *disjoint* X, Y *surroundsDisjoint* A.

If Y is not *surroundsMeet*, *SurroundsAttach*, or *SurroundsDisjoint* A then none of the substantive *surrounds* relations hold—Y is *surroundsEmpty* A. ■

The most general (nonempty) case, *surroundsDisjoint*, occurs when an object sits within a hole (or gap) but does not share a boundary intersection with the hole—the host of the hole has the *surroundsDisjoint* relation with the object within the hole (Figure 4.13a). The converse of this relation is *surroundedByDisjoint*, which is the relation the object within the hole has with the host of the hole.

Given that the scene description developed in Section 3 treats holes and gaps as a special case of a region, whenever an object is *inside* a hole, it has the *surroundedByDisjoint* relation with the host of the hole. No complex reasoning is needed; all holes have a host as part of their definition.

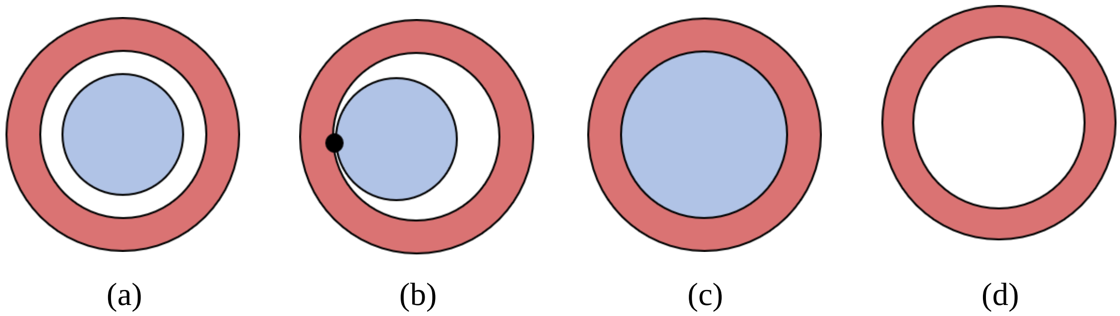


Figure 4.13 Four surrounds configurations. (a) A holed region *surroundsDisjoint* the region *inside* the hole, (b) a holed region *surroundsMeet* the region in the hole, (c) a holed region *surroundsAttach* the region filling the hole, and (d) a holed region *surroundsEmpty* a nonexistent region in the hole (Dube and Egenhofer 2014).

If an object has the *surroundedByMeet* relation with a holed region, the object *meets* the boundary of the hole as well (Figure 4.13b). Conversely, the holed region has the *surroundsMeet* relation with the object. Finally, if a region fills a hole it is has the *surroundedByAttach* relation with the holed region (Figure 4.13c). Conversely, the holed region has the *surroundsAttach* relation with the region.

The *surrounds* relations involving a gap are similar, but there is additional nuance. When a region is *inside* of a gap, it is not *surrounded* by a single region, for instance. As a gap is formed by the union of objects, the object *inside* of the gap is *surrounded* by the union of objects that form the gap. The same holds for the other relations. As a gap is defined with all of the same attributes as a region or a hole, including a set of *boundary_edges* (Section 3.5.1). The objects that form the gap are the other objects in the scene that share those boundary edges.

4.6 Boundary Description

While set operations are enough for determining the coarse relations between a pair of regions (Sections 4.4-4.5), a description that captures how an object's boundary (region) or extent (line) relates to other scene objects is produced by sequencing the local relations.

Consider a scenario where two regions *meet*. The regions could *meet* at a single node, along an edge, or multiple times at any number of nodes and edges. Describing that sequence aids in uniquely representing the boundary of the objects participating in the *meet* relation (Section 3.3).

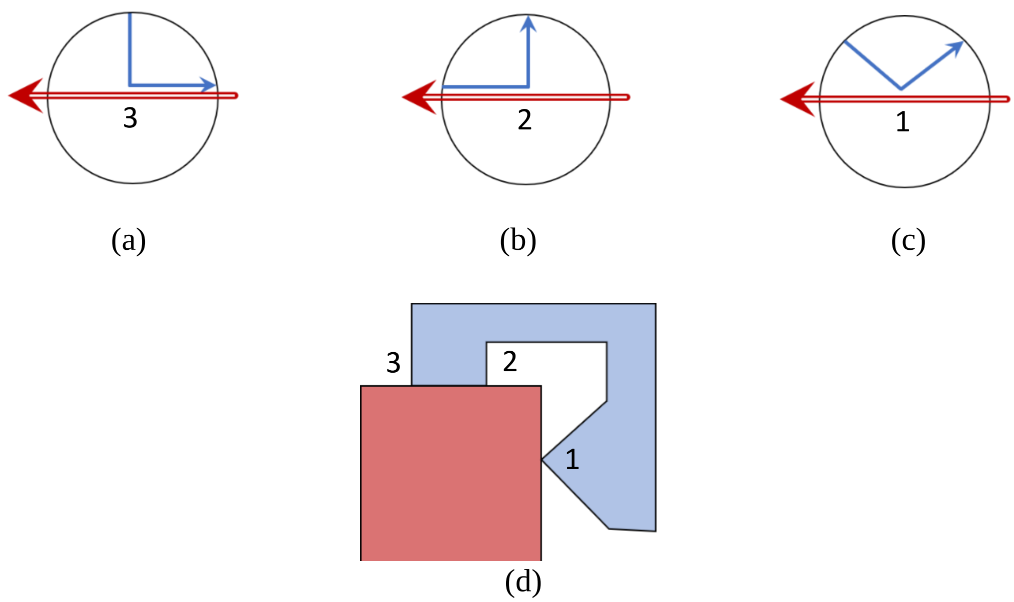


Figure 4.14 A sequence of local relations describes a boundary. (a) Two regions end a 1-dimensional *meet*, (b) two regions begin a 1-dimensional *meet*, (c) two regions *meet* at a point, and (d) all three local relations are combined into a depiction of the two regions.

For simple cases, listing the local relations between objects in the proper sequence is enough to describe the interrelation between the objects' boundaries; a region A *meets* a region B at a point (Figure 4.14c), a region A begins *meet*ing a region B along an edge (Figure 4.14b), and a region A stops *meet*ing a region B along an edge (Figure 4.14a). Together this 0-dimensional *meet,* along with the start and end points of a 1-dimensional *meet,* describes the boundary of the region A (Figure 4.14d).

It is straightforward that a sequence of *meet* relations indicates that two regions in fact *meet*. Less clear, however, is the case where a region A begins a local *meet* relation with a region B (Figure 4.15c), region A ends a local *entangled* relation with region B (Figure 4.15b), and region A ends a local *overlap* relation with an object B (Figure 4.15a). The start of a 1-dimensional *meet* relation necessarily begins in the exterior and converges on a shared boundary; the end of a 1-dimensional local *entangled* relation conversely begins from a shared boundary and diverges into the interior. Together, a pair of starting and ending local relations represent a shared boundary edge. If both local relations are of the same type (i.e., *meet* and *meet*, *covers* and *covers,* and so forth) the relation is straightforward, remaining unchanged. However, if the pairs of local relations are dissimilar (e.g., *meet* and *entangled*) the relation resolves to *overlap*.

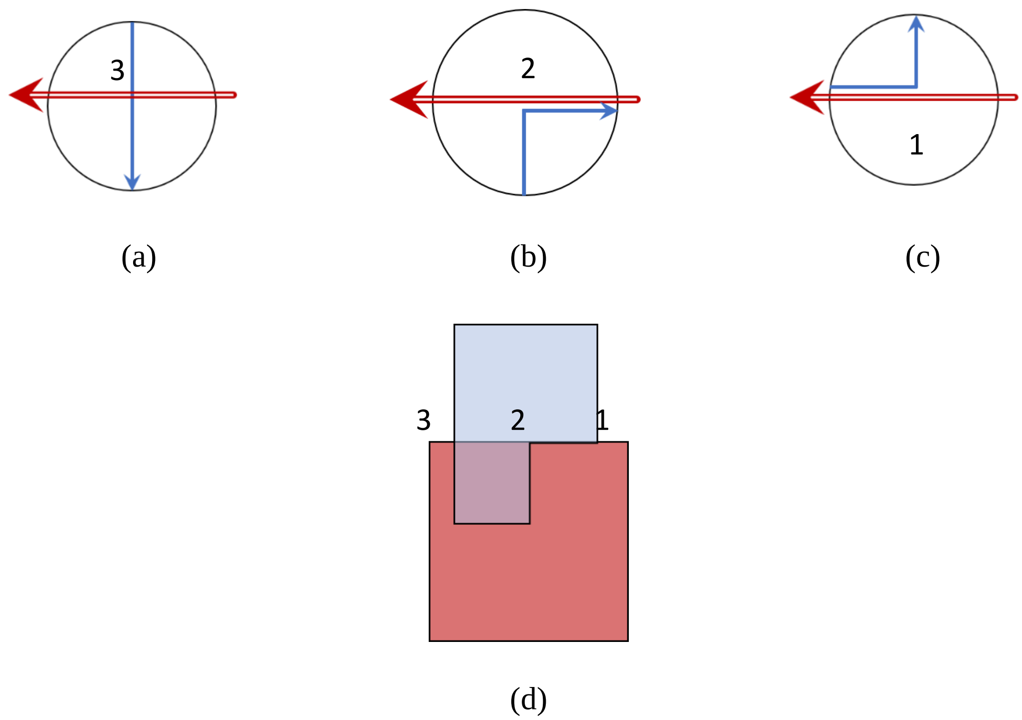


Figure 4.15 Another sequence of local relations describes a boundary. (a) Two regions end a 0-dimensional *overlap*, (b) two regions end a 1-dimensional *entangled*, (c) two regions begin *meeting* along an edge, and (d) all three local relations are combined into a depiction of the two regions.

Finally, the description of an object's boundary needs not be limited to a pair of objects: by listing the local relations between an object A and the other objects it intersects, for each boundary intersection in sequence the structure of A's boundary is fully described (Figure 4.16).

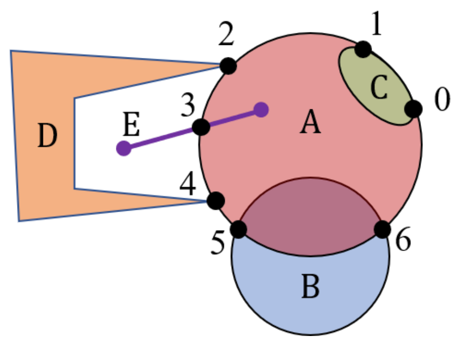


Figure 4.16 Local relations that describe the boundary of region A. 0: A *RR covers-start* C, 1: A *RR covers-end* C, 2: A *RR meet-point* D, 3: A *RL overlap* E, 4: A *RR meet-point* D, 5: A *RR overlap-start* B, 6: A *RR overlap-end* B.

By combining this boundary description (Figure 4.16) with information about the areas A contains, the entirety of A (or any other object) is represented. The boundary description also maps onto a set of coarse relations, enabling the objects to be reasoned about in a fashion like existing models, but with the benefit of more detailed structural information should it be desired. Three coarse 9-intersection relations demonstrate this correspondence (Figure 4.17), shown alongside the *local relations* used to describe a similar configuration.

As discussed in Section 4.4, however, this process has its limitations. When constructing scenes with lines it is necessary to account for both boundary endpoints of the line when constructing the 9-intersection matrix using the *ExteriorBoundary* and *BoundaryExterior* constraints (Section 4.4). When the endpoints of a line do not intersect the must still be recorded in the matrix (Figures 4.17b, 4.17c, Figures 4.18b, 4.18c).

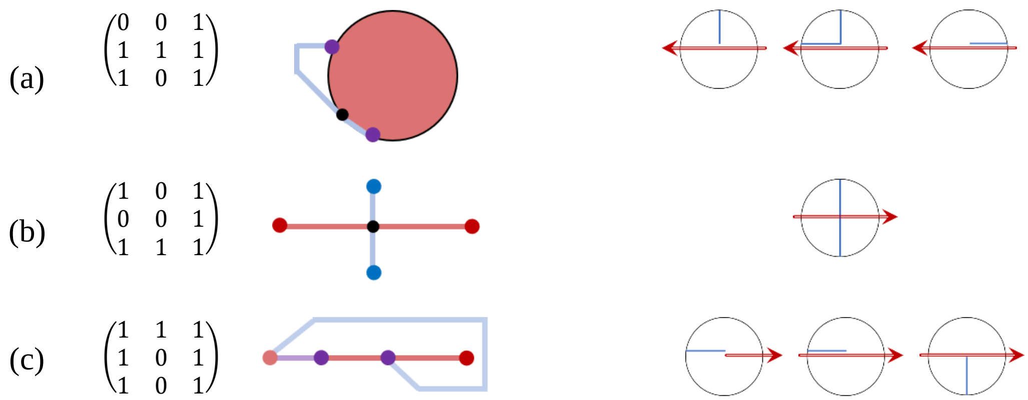


Figure 4.17 Three configurations of objects. (a) A line-region configuration with a 9-intersection and Scene Notation representation, and (b-c) two line-line configurations with their 9-intersection and *local relation* representations augmented with *BoundaryExterior* and *ExteriorBoundary* information.

Each of the three 9-intersection matrices is described by at least one local relation (Figure 4.17). The depiction of the 9-intersection relation is one of many that map onto its matrix. Specific matrices may have an infinite number of additional valid configurations, each of which leads to a unique sequence of *local relations* (Figure 4.18).

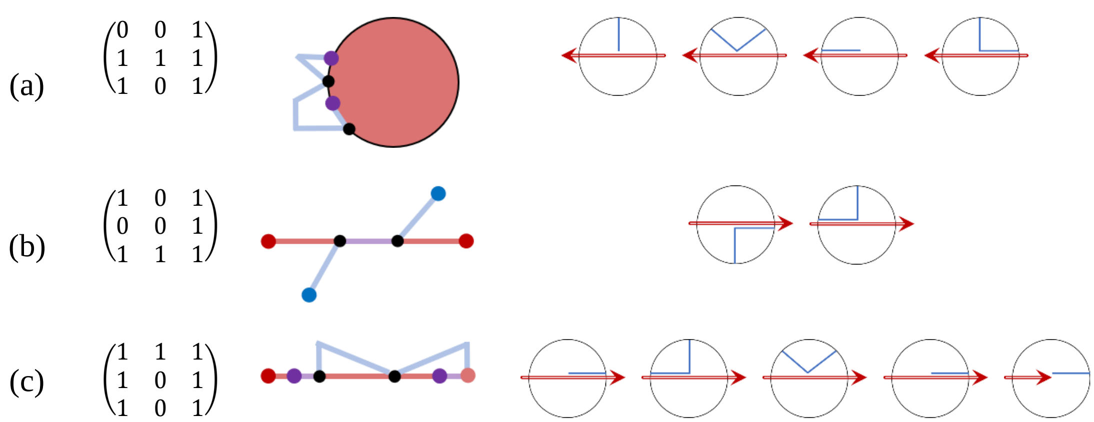


Figure 4.18 Three alternate configurations of objects. (a) A line-region configuration with a 9-intersection and Scene Notation representation, and (b-c) two line-line configurations with their 9-intersection and *local relation* representations.

These examples (Figures 4.17 and 4.18) demonstrate that the local relation, and by extension Scene Notation, can provide refinements of the coarse relations by capturing structural differences that map onto a specific matrix, allowing the number, sequence, and dimension of those intersections to be more fully modeled.

4.7 Summary

The sequence of edges that connect at a given intersection point form a set of 72 distinct local relations, providing a detailed description of the boundaries of regions or the extent of lines. Of these 72 relations, 12 exist between a region and a line, 16 exist between a pair of regions, 20 exist between a pair of lines, and 24 exist between a line and a region. Each local relation can also be mapped onto a 9-intersection analogue.

The local relations, however, only capture relations that involve boundary intersections, not those based on containment. Those relations can still be derived using the properties of objects under set intersection and difference. Such operations also enable relations involving boundary intersections to be captured, alternative to the local relations. The set operations also enable a set of *surrounds* relations to be derived for holed regions and gaps.

Regardless, the local relations do enable the boundary of a region or the extent of a line to be described as a sequence of detailed relations that add up to a structural description of the object, which is combined with information about the interior of an object.

CHAPTER 5

SCENE NOTATION IMPLEMENTATION

Having developed a strategy for describing the elements of a spatial scene, a software prototype implementation is developed, named *SceneAnalyze*. The software prototype serves to demonstrate that the approach described is realizable. To that end, a user is presented with an interface that enables collections of lines and regions to be sketched or modified based on the specifications presented in Chapter 3, and those sketches are then processed using the approaches developed in Section 3.5 and Chapter 4.

5.1 Implementation

The SceneAnalyze prototype was written in Python 3.6 due to the ease of development for that programming language. Many of the operations performed on the sketched geometry were in part enabled by the external Shapely library, which is itself based off GEOS, a C++ implementation of the Java Topology Suite. These include operations, such as identifying the cells within a scene by polygonizing its boundaries, to splitting edges at their intersection points.

The user interface was written using Kivy, an external Python library that enables the construction of highly interactive user interfaces. Tkinter, a standard GUI library included with Python, proved to be both more difficult to work with and more limited in its features (particularly the inability to handle partially transparent objects—useful when sketching overlapping shapes).

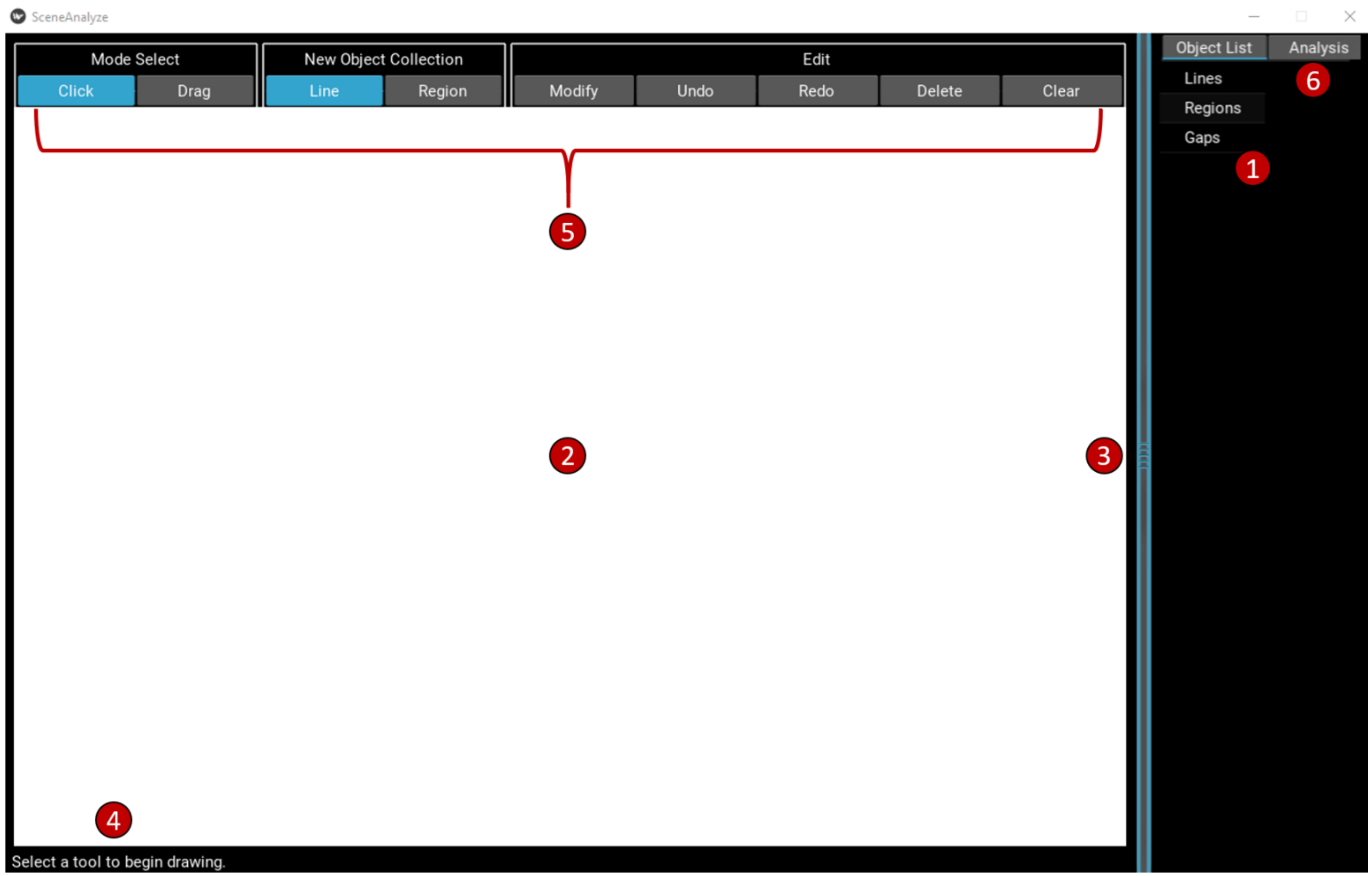
5.2 Interface Elements

The user interface was designed foremost for the purpose of enabling a user to draw collections of regions and lines, with those tools being presented initially upon running the program (Figure 5.1).

The *Object List* contains all the objects drawn by a user, segregated by type. These objects are sorted by collection, and each collection is represented as a hierarchy—the children of an object (holes) are listed after their parents at a different level of indentation. Different collections or children can be expanded or hidden from view as desired.

The *Drawing Canvas* is where the user draws a spatial scene consisting of regions and lines. All drawn objects are cropped to the canvas. The *Resize Bar* can be dragged to resize the canvas and the right-most menu relative to each other. Scrolling is also enabled for the canvas if the containing window is to small to accommodate it, and the main window itself starts at a resolution of 1,300 x 800 pixels, and can be expanded to accommodate different sized displays, with window contents scaling to fit the provided space.

The *Notification Bar* on the bottom of the window populates with error messages as they occur. Such errors include drawing invalid objects (drawing a polygon with two edges) and trying to delete an object with nothing selected. On the upper-right side of the window there are two tabs.



1: Object List

2: Drawing Canvas

3: Resize Bar

4: Notification Bar

5: Creation Tools

6: Analysis Tab

Figure 5.1 The SceneAnalyze prototype initial view displays the canvas and drawing tools.

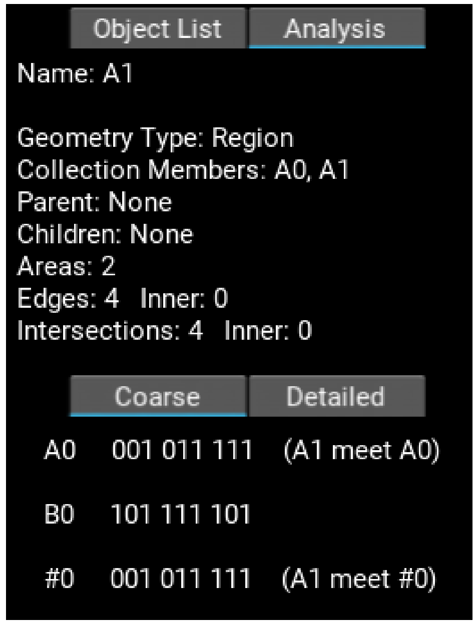


Figure 5.2 The SceneAnalyze prototype represents object properties as well as coarse and detailed boundary relations in the lower pane (initially blank).

The Creation Tab displays all the available drawing tools (visible by default), while the Analysis Tab displays details about a selected object, as well as it's relation to the other objects within the scene (Figure 5.2). Inner edges and inner intersections correspond to the edges and intersections that describe an object's holes. The object being described is highlighted on the canvas for added visual identification.

5.3 Drawing Interface

The set of drawing tools (Figure 5.1) allows the user to draw regions and lines by either clicking to place the endpoints of edges, or by holding the mouse down and dragging it across the canvas. Either way, double-clicking attempts to complete the object. The user can choose to draw a line, a region, or to modify a selected region. This modification allows holes to be carved out of regions, or additional objects to be added to collections.

A line requires at least one edge, while a region requires at least three edges. Simple lines and regions are unable to self-intersect. Failure to construct a valid region or line will result in the attempt being deleted and an error message being displayed at the bottom of the screen detailing the error, while a valid construction creates the object and outputs that the operation was successful. Additionally, the coordinates that define a region are reoriented to be counter-clockwise if they were drawn with a clockwise orientation. The full list of errors is detailed:

0. Success!
1. A line requires at least two points; a region requires at least three.
2. A line must not self-intersect.
3. A line may not have interior-interior or boundary-interior intersections with its collection.
4. A region may not have a disconnected interior.
5. A region may at most have point-connections with its collection.
6. A region/hole may not overlap its parent.
7. A region/hole may at most have point-connections with it's parent.
8. The modification tool requires an object be selected from the list.
9. The delete tool requires an object to be selected from the list.

In a screenshot of the interface (Figure 5.3), two regions (A0 and A1), belonging to the same collection (A), are drawn along with a line (B0). Together, A1 and B0 form an exterior gap (#0). Region A1 is selected in the Object List. The selected collection (A) is highlighted in red, and the specific selection gains a distinguishing border that is also thicker and brighter than the other members of its collection. The intersection points between these objects are labeled.

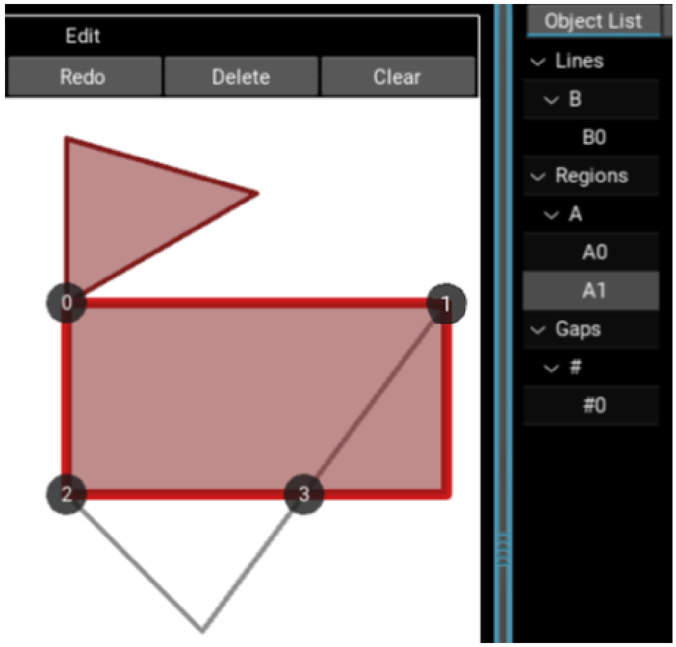


Figure 5.3 The interface depicting two regions belonging to collection A, with A1 highlighted. A line B0 intersects with A1, forming the gap #0.

By selecting A0 and choosing "Modify" another element of collection A was added. Since A1 exists outside of A0 it appears as a new region, rather than as a hole. In this sense the Modify tool is either additive or subtractive depending on the context of the new element relative to the collection it is being added to.

By selecting either A0 or A1 and choosing "Delete," either object would be removed while preserving the existence of the other. However, if region A1 contained a hole and was deleted, the hole object would be taken along with it—a hole has no reason for being without a host object as it would be a hole in nothing. Similarly, if a hole contained a disconnected piece of A (a region) and was deleted, that contained region would be deleted along with the hole. Otherwise the contained subpart of A would end up existing within the hole's parent, which is also a region belonging to collection A. A's interior would be coincident with itself, an invalid construction.

5.4 Extracting a Formal Representation of a Scene

The prototype identifies the components of the drawn objects, as well as their relations with other objects in the scene, from coarse relations to a rich description of the sequence of objects that intersect the boundary of a region. The prototype fully represents the coarse relations between regions, but reports line relations with the corresponding 9-intersection string rather than using explicit relation names like LL12 (which convey little surface meaning).

The interface for describing an object and its relations is simple (Figure 5.4). First, and object is chosen from the Object List on the far left. The upper half of the analysis pane on the right of the window lists details about the construction of the selected object—its ID, type, collection, the objects it is a part of, the object it is a child of, and how many intersections, edges, and areas it comprises (Figure 5.4b). The lower pane is divided into two tabs.

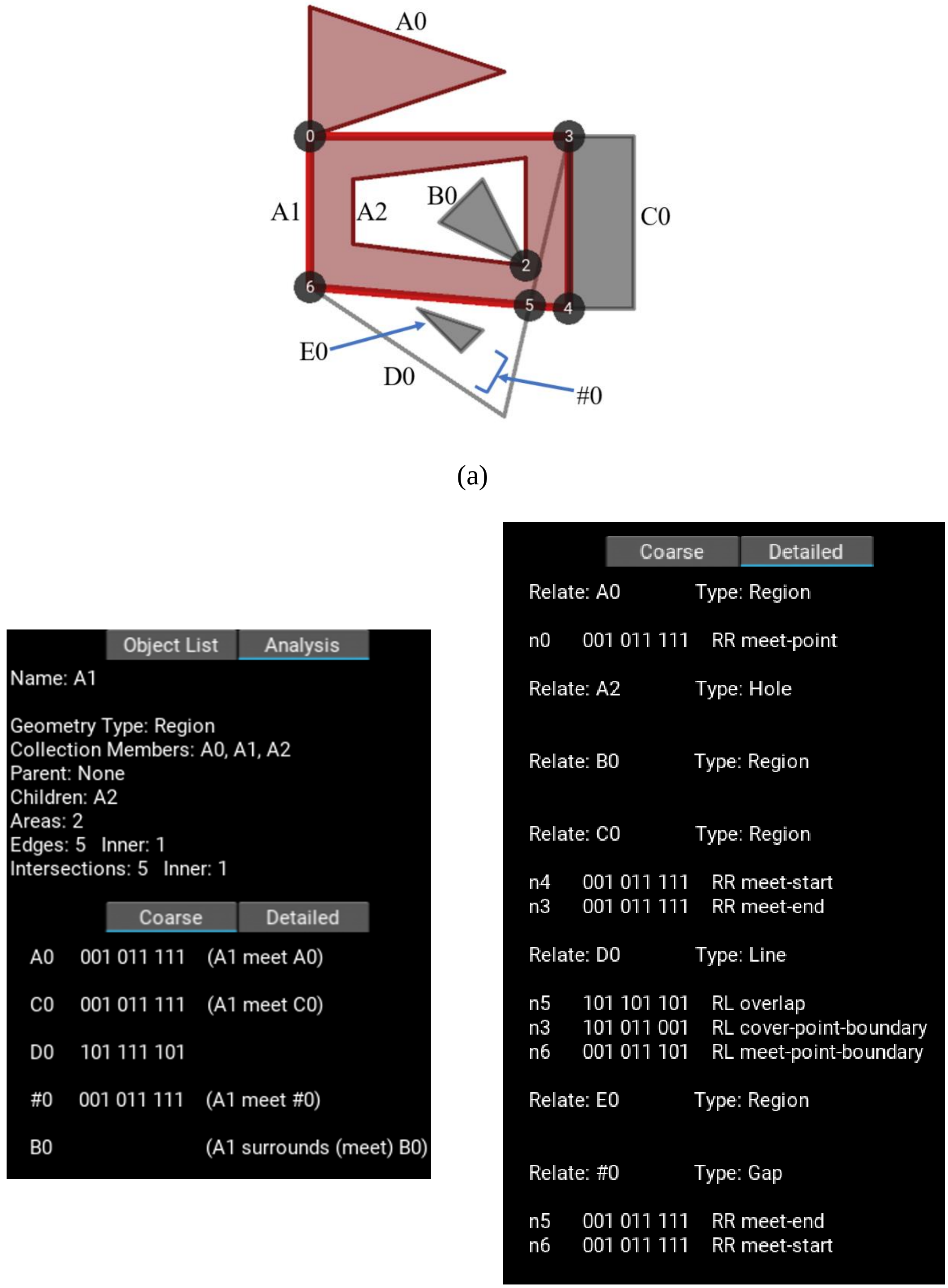
The first tab lists the coarse relations between the selected object and all other objects (Figure 5.4b), except *disjoint* objects (this was an omission to simplify the result), while the second tab describes the boundary sequence for the selected object, for each node (Figure 5.4c).

To begin, a scene must first be drawn (Figure 5.4a). The scene from Figure 5.3 has been modified by the addition of a hole in A1, and the addition of three new regions. In the screenshot, A1 is still selected. The interface reveals the details about the content of the region first. For instance, it has five intersections and is the parent of a hole A2.

The lower half of the window depicts the coarse relations that A1 participates in (excepting *disjoint*). Region A1 *meets* both A0 (a member of its collection) as well as E0, one of the newly added regions. It *contains* its hole (an object may *contain* or *cover* a hole since holes are treated as regions) and has a *coarse* region-line relation with the line B0. The specific coarse relation between A1 and B0 has not been given a natural-language label, but the 9-intersection string is shown.

By selecting the rightmost tab, a description of the object, its components, and its relation to other scene objects is displayed (Figure 5.4c). For each of the five intersection points A1 participates in (the hole is considered separately), the local relations are derived for all participant objects and listed in sequence. While the listed sequence starts at intersection three, the specific starting point is irrelevant if the ordering of the intersections that follow is consistent.

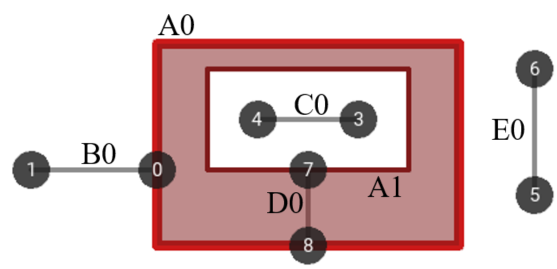
By traversing the outside of A1 its boundary structure is represented, including the beginning and ending of its *meet* relation with E0, which is split across two intersection points and a *meet* relation with the gap #0. Together the boundary description, the set of coarse relations, and the list of object-attributes describes each object within the scene in detail.



(b) (c)

Figure 5.4 The SceneAnalyze prototype interface displaying the coarse relations for a selected object within the scene. Arrows and labels added to the drawing for clarity.

Next, an example scene like the example in Chapter 3.6 is presented, with slight differences to object names. The SceneAnalyze sketch of the scene is first presented, along with a representation of both the coarse output from the prototype (Figure 5.56) as well as the detailed output for the same scene (Figure 5.6). The text has been simplified for presentation.

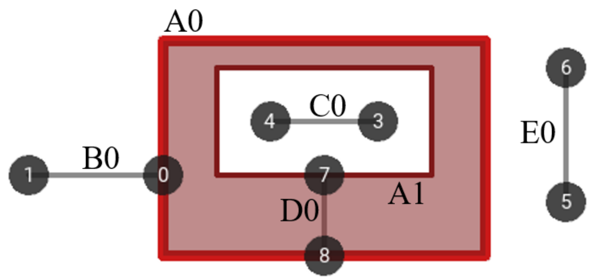


(a)

(Name: A0) B0 001 011 111 A0 D0 101 011 001 A0 C0 surroundedBy (disjoint) A0	(Name: A1) D0 101 011 001 A1 C0 inside* A1	(Name: B0) A0 001 011 111 B0
(b)	(c)	(d)

(Name: C0) A0 surrounds (disjoint) C0 A1 contains* C0	(Name: D0) A0 100 010 111 D0 A1 100 010 111 D0	(Name: E0)
(e)	(f)	(g)

Figure 5.5 A scene of objects and their coarse description. (a) Objects drawn using SceneAnalyze and (b-g) a representation of the coarse description generated for each object. Note that the boundary of A1 is an inner boundary of A0 and the container property of C0 determines its location within hole A1 (properties not depicted), using the semantics of containment as if the hole were a region (interior and exterior swapped).



(a)

(Name: A0) Relate: B0 Type: Line n0 001 011 101 Relate: E0 Type: Line N8 101 011 001	(Name: A1) Relate: D0 Type: Line n7 001 011 111	(Name: B0) Relate: A0 Type: Region n0 001 010 111
(b)	(c)	(d)

(Name: C0)	(Name: D0) Relate: A0 Type: Region n7 100 010 111 Relate: A0 Type: Region N8 100 010 111	(Name: E0)
(e)	(f)	(g)

Figure 5.6 A scene of objects and their detailed description. (a) Objects drawn using SceneAnalyze and (b-g) a representation of the detailed description generated for each object. Note the difference in the 9-interserction matrix for the boundaries of lines—nonintersecting boundary points are not captured—conversion to valid coarse relations is handled in Chapter 4.4.

Both the coarse and the detailed descriptions of these scene are less complex than the raw *Scene Notation* description presented in Chapter 3 and the possibility for representing the information in multiple levels of detail demonstrates that while verbose, the *Scene Notation* is also versatile. When multiple intersections occur between two objects (not depicted) each 9-intersection matrix can be combined to form a course relation, which additionally requires that all boundary nodes of a line be accounted for (Chapter 4.4). To reason with holes and gaps as if they were regions (i.e., the object is *inside* of the hole) the interior and exterior sets of the hole or gap must be swapped (Theorem 4.1), which the prototype does, although a more strict interpretation may be desirable.

5.5 Summary

The SceneAnalyze prototype for the spatial scene description described in this thesis allows a user to draw scenes and determine the construction of the depicted objects, as well as their relations to one another. It enables various configurations to be quickly encoded as notation with the derived properties and relations displayed.

The purpose of such a prototype is not to become a fully-fledged CAD program, or to perform rigorous spatial analysis on real-world spatial data (for instance). The prototype serves to demonstrate that the set of properties captured, and the relations derived from them can be used to describe a scene and that the underlying process is implementable. The step(s) beyond such a prototype may involve reducing input scenes to a qualitative description and relating them using the elements provided herein for use in some other process.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

This thesis is concerned with spatial scenes (Bruns and Egenhofer 1996), particularly collections of regions and lines along with their qualitative spatial relations. Regardless of how a spatial scene is represented (through geometric coordinates, a descriptive set of topological relations between objects, a verbal description, or with a depiction like a sketch), a description of the scene should have a correspondence—a mapping—between itself and the scene it purports to describe.

The representation of spatial scenes consisting of region and line objects is the basis for this work, by developing an approach that allows such a scene to be described while maintaining a specific set of topological properties. To this end the following hypothesis was presented:

> *When modeling an input scene [of lines and regions embedded in $\mathbb{R}^2$] by (1) decomposing the scene into a set of areas, edges, and nodes, and (2) recording the sequences of edges connecting each node and the area that contains each object, a detailed description of the scene is produced. The description enables three established topological invariants to be derived: (1) the dimension of the intersections between objects; (2) the containment relations between specific objects, holes and gaps; and (3) the relative ordering of intersecting objects around the boundary of a region and along the extent of a line. A detailed description requires all of these three properties in tandem—any omission may lead to ambiguity.*

The hypothesis leads to the following conjecture, which may be investigated further in future work:

Conjecture 1.1 The Scene Notation describes a scene uniquely, up to homeomorphism. Consequently, any two scenes produced from a given scene description are topologically identical.

6.1 Major Contributions

In order to satisfy the hypothesis, the construction of the objects themselves had to be considered, along with the topological properties that such a description would capture. It was shown that omitting any element from the developed *Scene Notation* would result in a description that was not equivalent to the input scene (Section 3.6)—each property is required along with all of the others.

6.1.1 Scene Notation

The scene description, called *Scene Notation*, captures the relations between lines, regions, holes, and exterior gaps within $\mathbb{R}^2$. The representation of these objects is conceptually based on algebraic topology, with each object being constructed from cell complexes. Collections of homogeneous objects (regions, for instance) are then able to be combined into collections of objects, to represent complex constructions (such as those with separations) using individual simple components.

These components are then described using a set of attributes for each object type (the edges and areas that make up each object, as well as its type and identity), as well as the sequence of intersections between objects (with intersections along an edge being broken up into individual nodes) and the sequence of edges around an intersection node.

Through the *Scene Notation*, the complex boundary, interior, and exterior intersections between objects is modeled.

Various operations developed provide further information, derived from the base set of attributes, such as the oriented sequence of edges surrounding a region, or the ordering of the edges between the end points of a line. The interiors and boundaries of holed regions were also derived, as distinct from the areas and edges that bound such an object. Finally, scene objects were split into their strongly connected components at articulation points, allowing subparts of a scene to be considered independently.

6.1.2 Local Relations and Containment Relations

As a scene may be made up of any number of regions and lines together, it was also necessary to consider how pairs of regions, pairs of lines, pairs regions and lines, and pairs of lines and regions might intersect. To this end 72 local relations are produced, with a single local relation depicting the sequence of edges that intersect a specific intersection node. There are 12 region-line relations, 16 region-region relations, 20 line-line relations, and 24 line-line relations.

The number of elements within each set is not symmetric (i.e., there are more line-region relations than there are region-line relations) due to such properties as the directness of a line affecting how a line is traversed from start to end, but not how it actually relates to other objects (i.e., if one boundary point of a line is in a region's boundary and the other is in the exterior there is no idea of a *first* or *second* boundary point—the line just has a boundary-boundary and a boundary-exterior relation).

Each of these local relations is also mapped onto a 9-intersection matrix, but the set of local relations is insufficient to capture cases where there are no point intersections between objects. To this end, the intersections between object components can be used in a more direct fashion to obtain the coarse 9-intersection relations between objects. Furthermore, by determining the coarse relations between an object and a hole (or a gap) the *surrounds* relations for those objects were determined.

Finally, the set of local relations is used to fully describe the boundary of an object in sequence, indicating the beginning and end of any number of relations between an object and potentially every other object within the scene, telling a narrative about that object (e.g., something approximating A begins *meet*ing B, A *covers* C at a point, A stops *meet*ing B, A begins *overlap*ping D, A ends *overlap*ping D, using local relations). Together, this specific boundary information, and the information about what an object *contains* (how many partitions it is broken into, what each of them contain, and so forth) allows an object to be robustly described.

6.1.3 Prototype Implementation

The SceneAnalyze software prototype, written in Python, allows a user to sketch scenes and determine the construction of the depicted objects (based on *Scene Notation*), as well as their relations to one another based on the developed model (the *local relations* and containment relations). It enables various configurations of objects to be quickly tested and described.

The purpose of the prototype is not to demonstrate a feature-rich design platform, or to intake real-world spatial data, but to demonstrate that the set of properties captured, and the relations derived from them can be used to describe a scene, and that the underlying process is implementable. Like the protype, the theory it is based on can be further expanded and refined. Additional objects may be added, additional use-cases considered, leading to the development of ever richer descriptions of spatial scenes.

6.2 Future Work

While the existing framework provides a rich description for regions and lines, it may be desirable to include both additional objects, as well as different embeddings for those objects. The prototype would necessarily also need to be modified to accommodate any change to the presented model, as well as being developed further to additional ends.

6.2.1 Points

Currently the only points considered in the scene description are intersection points and the endpoints of lines. The omission of points-as-objects is not due to any significant technical restriction, but rather due to a representational issue. As all intersecting edges are represented by pairs of points it becomes awkward to then also represent points as objects unto themselves, complete with the ability to also have their own intersection points with objects—intersections that are equivalent points. While not developed in detail, the local relations with points are nonetheless provided below (Figure 6.1).

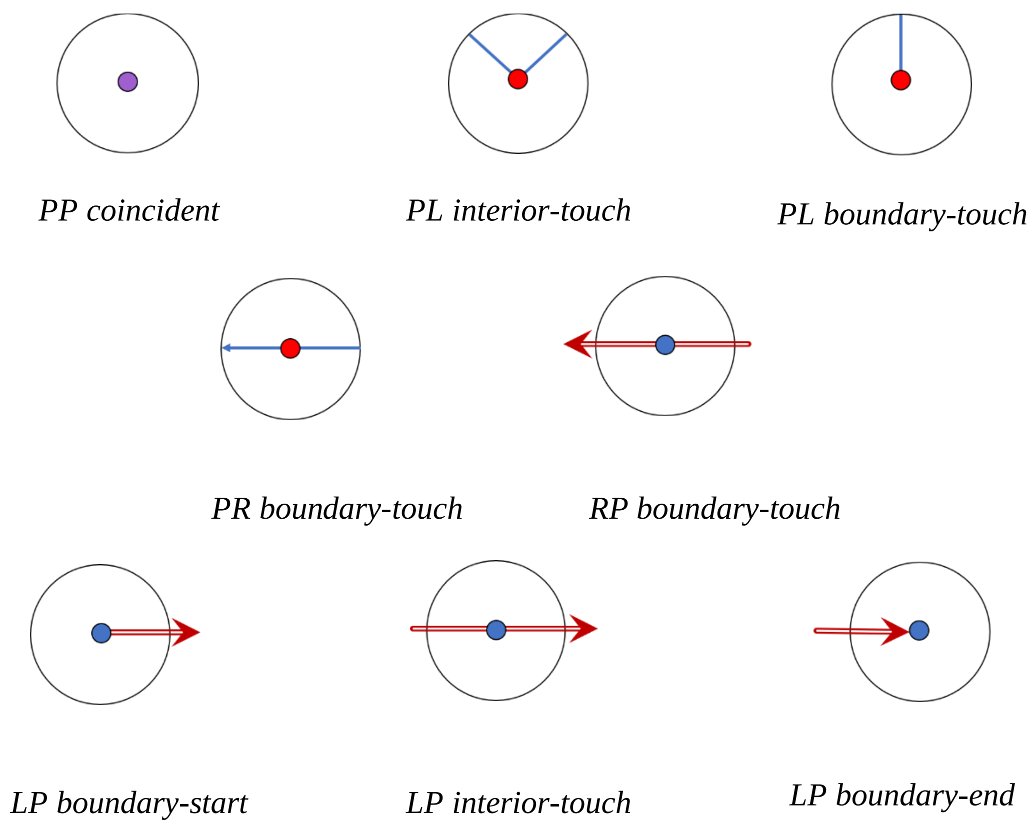


Figure 6.1 The eight local relations involving points.

6.2.2 Exteriors and Volumes

Adding additional objects in $\mathbb{R}^2$ need not be the end of an expansion, however. Other embedding and the objects they enable might also be considered, as well other modifications to the notation to make it consistent across the new cases. Volumes

As developed, the scene description ignores the exterior entirely until a hole or a gap is formed. From that point forward, only partitions of the exterior that are directly induced as a result of user action are represented. In order to more fully represent a scene, the representation of the exterior within a scene needs to be given a more complete description, which may allow $\mathbb{R}^3$ to be considered as an embedding space along with more complex objects, such as the torus (Figure 6.2b).

(a) (b)

Figure 6.2 A gap in $\mathbb{R}^3$ requires additional refinement (a) A gap between lines and (b) the gap in a torus are not disconnected from the rest of the exterior.

An issue arises with the representation gaps in $\mathbb{R}^3$, which will have to be overcome—while a gap in the plane partitions the exterior, such a configuration does not partition the exterior in $\mathbb{R}^3$. How to represent these new objects (as well as simple volumes) and their boundary relations will be necessary—hopefully in a form that is directly relatable to the work already completed for regions and lines in $\mathbb{R}^2$ (Figure 6.3).

(a) (b)

Figure 6.3 Two volumes intersect in two alternate views. (a) A diagram of their intersection and (b) a representation of the bottom object's boundary as a disk, with its intersection with the upper object.

While the example (Figure 6.3) is only a starting point, the boundaries of the two objects appear to be describable in the terms of this work with minor modification. The objects rest on a shared surface (the intersecting area) before one *crosses* into the other (the handle-like object connected to the area. This leads back to the idea of accommodating additional types of objects—beyond just points, accommodating collections of mixed type (regions with lines, for instance) additional scenarios like this may become easier to accommodate.

6.2.3 Toward a More Natural Description

The set of *local relations* (Sections 4.1-4.3) can be used in sequence to describe an object, in concert with information concerning the object's interior and its containment; however, these descriptions, while robust, are not at the level of natural language. Given that the structural details of an object and its relations are captured, it may be possible to transform a formal description of an object into one that that is closer to how a person would describe such a scene. This translation could be used to convey complex spatial information to a user when a visual representation is not a preferable modality for communication.

Given that the protype takes a user-generated sketch as input and represents it with a detailed description of all the sketched elements, a further step may be to expand the output to include different modalities beyond the textual description, such as a verbal description of the scene—a descriptive spoken narrative of the objects depicted. It is also worth considering whether a natural-language description could be used to generate a scene without the need to sketch it.

6.2.4 Continued Expansion of the Prototype

With the potential for expanding the types of objects represented and their descriptions, the prototype would necessarily need to be expanded to accommodate them, from specifying the rules for producing valid new objects to defining the relations that exist between them. In addition to that work, the prototype could also be expanded to accommodate additional ways of constructing existing objects, such as using set operations to produce new collections from already drawn regions and lines. The prototype could also be expanded to cover additional use cases—currently it is very focused on demonstrating the elements described within this dissertation with little consideration for any other purpose. As the purpose of the program may be expanded to fit additional needs, the underlying code may find additional uses without the UI wrapper (as a standalone library).

6.3 Summary

The spatial scene description developed herein, the Scene Notation, provides a means for regions and lines to be reasoned about, both regarding the sequences of intersections they have with other objects, the dimension of those intersections, and their containment relations with other objects—including subparts of themselves, such as holes and disconnected elements. These properties were shown to be necessary in order to describe an equivalent spatial scene through the provided intersection-based representation.

These considerations allowed several operations to be developed for the set of objects, allowing their interiors, boundaries, exteriors, containment, and connectedness to be used in later reasoning tasks. To this end, a set of 72 *local relations* was derived between combinations of lines and regions.

These local relations, when taken in sequence, allow the boundary of an object to be described based on its intersections. Furthermore, these local relations are mapped onto 9-intersection matrices by taking the union of all local relation matrices for an object, resulting in the identification of its 9-intersection matrix (if there is a boundary intersection).

For region-like objects without boundary intersections, the areas that partition the object are used to derive 9-intersection matrices and are also used to determine the *surrounds* relations between holed region or a gap and the objects *inside* the hole or gap.

A software prototype was then programmed to allow a user to sketch a spatial scene and immediately see the results of applying these relations on a set of objects. This demonstrated that the properties discussed could be captured, processed, and output in a manner like that described within this thesis.

REFERENCES

Adams, C. and Franzosa, R. (2008). *Introduction to Topology: Pure and Applied*. Pearson Prentice Hall.

Alexandroff, P. (1961). *Elementary Concepts of Topology*. Dover Publications, New York.

Allen, J. (1983). Maintaining knowledge about Temporal Intervals. *Communications of the ACM* 26, 832-843.

Avelar, S., and Hurni, L. (2000). On the Design of Schematic Transport Maps. In: *Cartographica* 41.3, 217-228.

Bateman, J., Hois, J., Ross, R. and Tenbrink, T. (2010). A Linguistic Ontology of Space for Natural Language Processing. *Artificial Intelligence* 174, 1027-1071.

Barkowsky, T., Latecki, L. and Richter, K. (2000). Schematizing Maps: Simplification of Geographic Shape by Discrete Curve Evolution. In: Freksa, C., Brauer, W., Habel, C. and Wender, K. (eds.), *Spatial Cognition* II, 41-53.

Bennett, B. (1998). Determining Consistency of Topological Relations. *Constraints* 3(2), 213-225.

Bertin, J. (1983). *Semiology of Graphics*, The University of Wisconsin Press, Madison, WI.

Blaser, A. (1998). Geo-Spatial Sketches. National Center of Geographic Information and Analysis, University of Maine, Orono, Technical Report TR98-1.

Blaser, A. and Egenhofer, M. (2000). A Visual Tool for Querying Geographic Databases. In: Di Gesù, V., Levialdi, S., and Tarantino, L. (eds.), *Working Conference on Advanced Visual Interfaces (AVI '00)*, Palermo, Italy, ACM Press, New York, NY, 211-216.

Brauer, W., Habel, C., Barkowsky, T., Latecki, L., and Richter, K-F. (2001). Schematizing Maps: Simplification of Geographic Shape by Discrete Curve Evolution. In: Freska, C., Habel, C., Brauer, W., and Wender, K. (eds.), *Spatial Cognition II*, Springer Berlin Heidelberg, 41-53.

Bruegger, B. and Kuhn, W. (1991). Multiple Topological Representations. National Center for Geographic Information and Analysis, University of Maine, Orono, ME, Technical Report, 1-17.

Bruns, T. and Egenhofer, M. (1996). Similarity of Spatial Scenes. *Seventh International Symposium on Spatial Data Handling*. Delft, The Netherlands, 31-42.

Caduff, D. and Egenhofer, M. (2005). Geo-Mobile Queries: Sketch-Based Queries in Mobile GIS Environments. In: Li, K.-J. and Vangenot, C. (eds.), *W2GIS: 5th International Workshop on Web and Wireless Geographical Information Systems*, Lausanne, Switzerland, *Lecture Notes in Computer Science*, Vol. 3633, Springer, 143-154.

Calcinelli, D. and Mainguenaud, M. (1994). Cigales, a Visual Query Language for a Geographical Information System: The User Interface. *Journal of Visual Languages and Computing* 5(2), 113-132.

Casati, R. and Varzi, A. (1994). Holes and other Superficialities. MIT Press, Cambridge, MA.

Chen, J., Cohn, A., Liu, D., Wang, S., Ouyang, J. and Yu, Q. (2015). A Survey of Qualitative Spatial Representations. *The Knowledge Engineering Review* 30, 106-136.

Clarke, B. (1981). A Calculus of Individuals Based on Connection. *Notre Dame Journal of Formal Logic*, 22(3), 204-218.

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 Beng Chin (eds), *Advances in Spatial Databases: Third International Symposium, SSD '93*, Singapore, *Lecture Notes in Computer Science* Vol. 692, Springer, 277–295.

Clementini, E., Di Felice, P., and Califano, G. (1995). Composite Regions in Topological Queries. *Information Systems* 20(7), 579-594.

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.

Clementini, E. and Di Felice, P. (1998): Topological Invariants for Lines. *IEEE Transactions on Knowledge and Data Engineering* 10(1), 38-54

Clementini, E., Di Felice, P. and Hernández, D. (1997). Qualitative Representation of Positional Information, *Artificial Intelligence* 95, 317-356

Coan, M., and Egenhofer, M. (1996). The Ontology of Land Boundaries under Natural Change: Erosion and Accretion of Sandy Shaols in Nantucket Sound. In: *ACSM/ASPRS Conference*, Baltimore, MD.

Cohn, A., Bennett, B., Gooday, J., and Gotts, N. (1997). Qualitative Spatial Representation and Reasoning with the Region Connection Calculus. *GeoInformatica* 1(3), 275-316.

Cohn, A. and Hazarika, S. (2001). Qualitative Spatial Representation and Reasoning: An Overview. *Fundamenta Informaticae* 46(1-2), 2-32.

Cohn, A. and Renz, J. (2008). Qualitative Spatial Representation and Reasoning. In: van Hermelen, F., Lifschitz, V., and Porter, B. (eds.), *Handbook of Knowledge Representation*, Elsevier, 551-596.

Cohn, A., Li, S., Liu, W., and Renz, J. (2014). Reasoning about Topological and Cardinal Direction Relations Between 2-Dimensional Spatial Objects. *Journal of Artificial Intelligence Research* 51(1), 493-532.

Dahlgren, K. (1988). *Naive Semantics for Natural Language Understanding*. Kluwer Academic Publishers, Norwell, MA.

De Kleer, J. and Brown, J. (1984). A Qualitative Physics Based on Confluences. *Artificial Intelligence,* Vol. 24, 7-83.

Di Loreto, F., Ferri, F., Massari, F., and Rafanelli, M. (1996). A Pictorial Query Language for Geographical Databases. In: Catarci, T., Costabile, M., Levialdi, S., and Santucci, G. (eds.), *Working Conference on Advanced Visual Interfaces (AVI '96)*, Gubbio, Italy, 233-244.

Dube, M. and Egenhofer, M. (2009). Establishing Similarity Across Multi-Granular Topological–Relation Ontologies. In: Rothermel, K., Fritsch, D., Blochinger, W. and Dürr, F. (eds.), *Quality of Context, Lecture Notes in Computer Science,* Vol. 5786, Springer Berlin, 98-108.

Dube, M. and Egenhofer, M. (2014). Surrounds in Partitions. In: Huang, Y., Schneider, M., Gertz, M., Krumm, J. and Sankaranarayanan J. (eds.), *ACM SIGSPATIAL GIS 2014—22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, Dallas, TX., ACM Press, 233-242.

Dube, M., Egenhofer, M., Lewis, J., Stephen, S. and Plummer, M. (2015). Swiss Canton Regions: A Model for Complex Objects in Geographic Partitions. In: Fabrikant, S., Raubal, M., Bertolotto, M., Davis, C., Freundschuh, S. and Bell S. (eds.), *Conference on Spatial Information Theory—COSIT 2015*, Santa Fe, NM., *Lecture Notes in Computer Science*, Vol. 9368, 309-330.

Egenhofer, M. (1989). A Formal Definition of Binary Topological Relationships. *FODO 1989*, 457-472.

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

Egenhofer, M. (1993). A Model for Detailed Binary Topological Relationships. *Geomatica* 47(3), 261-273.

Egenhofer, M. (1994a). Deriving the Composition of Binary Topological Relations. *Journal of Visual Languages and Computing* 5(2), 133-149.

Egenhofer, M. (1994b). Spatial SQL: A Query and Presentation Language. *IEEE Transactions on Knowledge and Data Engineering* 6(1), 86-95.

Egenhofer, M. (1996). Spatial-Query-By-Sketch. In: Burnett, M. and Citrin, W. (eds.), *VL'96: IEEE Symposium on Visual Languages*, Boulder, CO, 60-67.

Egenhofer, M. (1997). Query Processing in Spatial-Query-by-Sketch. *Journal of Visual Languages and Computing* 8(4), 403-424.

Egenhofer, M. (2005). Spherical Topological Relations. *Journal on Data Semantics III*, 25-49.

Egenhofer, M. (2009). A Reference System for Topological Relations between Compound Spatial Objects. In: Heuser, C. and Pernul, G. (eds.), *Advances in Conceptual Modeling—Challenging Perspectives, ER 2009 Workshops, Lecture Notes in Computer Science*, Vol. 5833, 307-316.

Egenhofer, M. (2010). The Family of Conceptual Neighborhood Graphs for Region-Region Relations. In: Fabrikant, S., Reichenbacher, T., van Kreveld, M., and Schlieder, C. (eds.), *6th International Conference, GIScience 2010* , Zurich, Switzerland, *Lecture Notes in Computer Science*, Vol. 6292, Springer, 42-55.

Egenhofer, M. and Al-Taha, K. (1992). Reasoning about Gradual Changes of Topological Relationships. In: Frank, A., Campari, I. and Formentini, U. (eds.), *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space,* Springer, Berlin, 196-219.

Egenhofer, M. and Frank, A. (1988). Towards a Spatial Query Language: User Interface Considerations. In: DeWitt, D. and Bancilhon, F. (eds.), *14th International Conference on Very Large Data Bases*, Long Beach, CA, 124-133.

Egenhofer, M. and Franzosa, R. (1991a) Point-Set Topological Spatial Relations. *International Journal of Geographical Information Systems* 5(2), 161-174.

Egenhofer, M. and Herring, J. (1991b). Categorizing Binary Topological Relationships between Regions, Lines and Points in Geographic Database. Technical Report, University of Maine.

Egenhofer, M. and Franzosa, R. (1995). On the Equivalence of Topological Relations. *International Journal of Geographical Information Systems* 9(2), 133-152.

Egenhofer M. and Mark, D. (1995a). Naive Geography. In: Frank, A. and Kuhn, W. (eds.), *Third European Conference on Spatial Information Theory (COSIT '95)*, Semmering, Austria, *Lecture Notes in Computer Science*, Vol. 988, Springer, 1-15.

Egenhofer, M. and Mark, D. (1995b). Modelling Conceptual Neighborhoods of Topological Line-Region Relations. *International Journal of Geographical Information Systems* 9(5), 555-565.

Egenhofer, M. and Shariff, R. (1998). Metric Details for Natural-Language Spatial Relations. *ACM Transactions on Information Systems* 16(4), 295-321.

Egenhofer, M. and Sharma, J. (1992). Topological Consistency. In: D. Cowen (ed.), *Fifth International Symposium on Spatial Data Handling*, Charleston, SC, 335-343.

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

Egenhofer, M. and Vasardani, M. (2007). Spatial Reasoning with a Hole. In: Winter, S., Duckham, M., Kulik, L., and Kuipers, B. (eds.), *Proceedings of the Spatial Information Theory, 8th International Conference, COSIT 2007*, Melbourne, Australia, *Lecture Notes in Computer Science*, Vol. 4736, Springer, 303–320.

Egenhofer, M., Frank, A., and Jackson, J. (1989). A Topological Data Model for Spatial Databases. In: Buchmann, A., Günther, O., Smith, T., and Wang, Y.-F. (eds.) *Design and Implementation of Large Spatial Databases, First Symposium* (SSD'89), Lecture Notes in Computer Science, Vol. 409, Springer, Santa Barbara, CA, 271-286.

Egenhofer, M., Sharma, J., and Mark, D. (1993). A Critical Comparison of the 4-Intersection and 9-Intersection Models for Spatial Relations: Formal Analysis. In: McMaster, R. and Armstrong, M. (eds.), *AUTOCARTO*, Minneapolis, MN, 1-11.

Frank, A. (1995). Qualitative Spatial Reasoning: Cardinal Directions as an Example. *International Journal of Geographical Information Systems* 10, 269-290.

Freksa, C. (1991). Conceptual Neighborhood and its Role in Temporal and Spatial Reasoning. In: Singh, M. and Travé-Massuyès, L. (eds.), *IMACS Workshop on Decision Support Systems and Qualitative Reasoning*, North-Holland, Amsterdam, Elsevier Science Publishers, 181-187.

Freksa, C. (1992). Using Orientation for Qualitative Spatial Reasoning. In: Frank, A., Campari, I., and Formentini, U. (eds.), *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space, Lecture Notes in Computer Science*, Vol. 639, Springer, 162-178.

Freksa, C. and Röhrig, R. (1993). Dimensions of Qualitative Spatial Reasoning. In: Carreté, P. and Singh, M.G. (eds.), *Qualitative Reasoning in Decision Technologies QUARDET '93*, CIMNE, Barcelona, 483-492.

Freundschuh, S. and Egenhofer, M. (1997). Human Conceptions of Space: Implications for GIS. *Transactions in GIS* 2(4), 361-375.

Galton, A. (1998). Modes of Overlap. *Journal of Visual Languages and Computing* 9(1), 61-79.

Galton, A. (2004). Multidimensional Mereotopology. *Principles of Knowledge Representation and Reasoning*, 45-54.

Galton, A. (2009). Spatial and Temporal Knowledge Representation. *Earth Science Informatics* 2(3), 169-187.

Gotts, N. (1996). Formalizing Commonsense Topology: The INCH Calculus. *International Symposium on Artificial Intelligence and Math*, 72-75.

Goyal, R. (2000). Similarity Assessment for Cardinal Directions between Extended Spatial Objects. PhD Thesis, Spatial Information Science and Engineering, University of Maine, Orono, ME.

Goyal, R. and Egenhofer, M. (2000). Consistent Queries over Cardinal Directions across Different levels of Detail. In: Tjoa, A., Wagner, R. and Al-Zobaidie, A. (eds.), *11th International Workshop on Database and Expert Systems Applications*, Greenwich, U.K., IEEE Computer Society, 876-880.

Gross, M. (1996). The Electronic Cocktail Napkin-Computer Support for Working with Diagrams. *Design Studies* 17(1), 53-69.

Haarslev, V. and Wessel, M. (1997). Querying GIS with Animated Spatial Sketches. In: Burnett, M. and Citrin, W. (eds.), *IEEE Symposium on Visual Languages*, Capri, Italy, 197-204.

Hahmann, T. (2013). A Reconciliation of Logical Representations of Space: From Multidimensional Mereotopology to Geometry. Ph.D Thesis, University of Toronto, Department of Computer Science.

Hahmann, T. and Brodaric B. (2012). The Void in Hydro Ontology. In: Donnely, M. and Guizzardi, G. (eds.), *Proceedings of the 7th International Conference on Formal Ontology in Information Systems*, IOS Press, 45-58.

Hahmann, T. and Gruninger, M. (2009). Detecting Physical Defects: A Practical 2D-Study of Cracks and Holes. *AAAI Spring Symposium: Benchmarking of Qualitative Spatial and Temporal Reasoning Systems*, Stanford, California, 11-16.

Hahmann, T. and Gruninger, M. (2011a). Multidimensional Mereotopology with Betweenness. *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-11),* AAAI Press, 906-911.

Hahmann, T. and Gruninger, M. (2011b). A Naïve Theory of Dimension for Qualitative Spatial Relations. In: Symposium on Logical Formalizations of Commonsense Reasoning (CommonSense 2011). AAAI Press.

Hayes, P. (1978). The Naïve Physics Manifesto. In: D. Michie (eds.), *Expert Systems in the Microelectric Age*. Edinburgh, Scotland: Edinburgh University Press, 242-270.

Hayes, P. (1985). The Second Naïve Physics Manifesto. In: Hobbs, J. and Moore, R. (eds.), *Formal Theories of the Commonsense World*. Norwood, NJ, Ablex, 1-36.

Hayward, W. and Tarr, M. (1995) Spatial Language Representation. *Cognition* 55(1), 39-84.

Hernández, D. (1991). Relative Representations of Spatial Knowledge. In: Mark, D. and Frank, A. (eds.), *Cognitive and Linguistic Aspects of Geographic Space*, Kluwer Academic, Dordrecht, 373-385.

Hernández, D. (1994). Qualitative Representation of Spatial Knowledge. Lecture Notes in Artificial Intelligence, Vol. 804, Springer-Verlag, Berlin.

Herring, J. (1991). The Mathematical Modeling of Spatial and Non-Spatial Information in Geographic Information Systems. In: Mark, D. and Frank, A. (eds.), *Cognitive and Linguistic Aspects of Geographic Space,* Springer Netherlands, 313-350.

Kim, J., Vasardani, M. and Winter, S. (2016). From Descriptions to Depictions: A Dynamic Sketch Map Drawing Strategy. *Spatial Cognition and Computation* 16(1), 29-53.

Klippel, A., Worboys, M. and Duckham, M. (2008). Identifying Factors of Geographic Event Conceptualization. *International Journal of Geographical Information Science* 22(2), 183-204.

Klippel, A. and Li, R. (2009). The Endpoint Hypothesis: A Topological-Cognitive Assessment of Geographic Scale Movement Patterns. In: Hornsby, K., Claramunt, C., Denis, M., and Ligozat, G. (eds.), *Spatial Information Theory COSIT '09*, Aber Wrac'h, France, *Lecture Notes in Computer Science*, Vol. 5756, Springer, 177-194.

Kordjamshidi, P., Van Otterlo and M., Moens, M. (2011). Spatial Role Labeling: Towards Extraction of Spatial Relations from Natural Language. *ACM Transactions on Speech and Language Processing* 8(3), 1-36.

Kritzman, G. and Hahmann, T. (2018). Using Commonsensical Cardinal Directions to Describe Bordering Objects. *Proceedings of the 26th International Conference on Advances in Geographic Information Science* (*ACM SIGSPATIAL '18*), ACM, New York, NY, 412-415.

Kurata, Y. (2010). From Three-Dimensional Topological Relations to Contact Relations. In: Neutens, T. and de Maeyer, P. (eds.), *Developments in 3D Geo-Information Sciences.* Springer, Berlin, 123-142.

Kurata, Y. and Egenhofer, M. (2007). The 9+-Intersection for Topological Relations Between a Directed Line Segment and a Region. In: Gottfried, B. (ed.), *Workshop on Behaviour and Monitoring Interpretation*, Technical Report 42, Technologie-Zentrum Informatik, University of Bremen, Germany, 62-76.

Kurata, Y. (2008a). The 9+-Intersection: A Universal Framework for Modeling Topological Relations. In: Cova, T., Miller, H., Beard, K., Frank, A., Goodchild, M. (eds.), *Geographic Information Science*. Park City, UT, *Lecture Notes in Computer Science*, Vol. 5266, Springer, 181-198.

Kurata, Y. and Egenhofer, M. (2008b). The Arrow Semantic Interpreter. *Spatial Cognition and Computing* 8(4), 306-332.

Lewis, J., Dube, M. and Egenhofer, M. (2013). The Topology of Spatial Scenes in $\mathbb{R}^2$. In: Tenbrink, T., Stell, J., Galton, A., Wood, Z. (eds.), *Spatial Information Theory COSIT '13*, Scarborough, UK, *Lecture Notes in Computer Science*, Vol. 8116, Springer, 495-515.

Lewis, J. and Egenhofer, M. (2014). Oriented Regions for Linearly Conceptualized Features. In: M. Duckham, K. Stewart, and E. Pebesma (eds.), *Geographic Information Science–Eighth International Conference, GIScience 2014*, Vienna, Austria, *Lecture Notes in Computer Science*, Vol. 8728, Springer, 333-348.

Li, S. (2006). A Complete Classification of Topological Relations Using the 9-Intersection Method. *International Journal of Geographical Information Science* 20(6), 589-610.

Li, S. (2007). Combining Topological and Directional Information for Spatial Reasoning. In: Veloso, M. (ed.), *20th International Joint Conference on Artificial Intelligence*, Hyderabad, India, 435-440.

Mackworth, A. (1977). On Reading Sketch Maps. In: *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, MIT Press, Cambridge, MA, 598-606.

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

Mark, D. (1993). Human Spatial Cognition. In: Medyckyj-Scott, D. and Hearnshaw, M.H. (eds.), *Human Factors in Geographical Information Systems*. Belhaven Press, 51.

Mark, D. and Freundschuh, S. (1995). Spatial Concepts and Cognitive Models for Geographic Information Use. In: Nyerges, T., Mark, D., Laurini, R., and Egenhofer, M. (eds.), *Cognitive Aspects of Human-Computer Interaction for Geographic Information Systems*. Kluwer Academic Publishers, Dordrecht, 21-28.

McKenney, M., Pauly, A., Praing, R., and Schneider, M. (2005). Dimension-Refined Topological Predicates in GIS. In: *Conference on Advances in Geographic Information Systems* (GIS-05), ACM Press, 240-249.

McKenney, M., Pauly, A., Praing, R., and Schneider, M. (2007). Local Topological Relationships for Complex Regions. In: Papadias, D., Zhang, D., and Kollios, G. (eds.), *10th International Symposium on Spatial and Temporal Databases*, 203-220.

Montanari, U. (1974). Network Constraints: Fundamental Properties and Applications to Picture Processing. *Information Sciences* (7), 95-132.

Montello, D. (1993). Scale and Multiple Psychologies of Space. In: Frank, A. and Campari, I. (eds.), *Spatial Information Theory: A Theoretical Basis for GIS*. *Lecture Notes in Computer Science*, Vol. 716, Springer-Verlag Berlin, 312-321.

Moratz, R. and Ragni, M. (2008). Qualitative Spatial Reasoning about Relative Point Position. *Journal of Visual Language Processing and Computation* 19(1), 75-98.

Moratz, R., Lücke, D., and Massakowski, T. (2011). A Condensed Semantics for Qualitative Spatial Reasoning About Oriented Straight Line Segments. Artificial Intelligence (175), 2099-2127.

Morehouse, S. (1995). GIS-Based Map Compilation and Generalization. In: Müller, J. C., Lagrange, J. and Weibel, R. (eds.), *GIS and Generalization: Methodology and Practice*, Taylor&Francis, Bristol, PA, 21-30.

Munkress, J. (2000). *Topology*, Upper Saddle River, NJ, Prentice Hall, Inc.

Nedas, K. and Egenhofer, M. (2008). Spatial-Scene Similarity Queries. *Transactions in GIS* 12(6), 661-681.

Nedas, K. and Egenhofer, M. (2003). Spatial Similarity Queries with Logical Operators. In: Hadzilacos, T., Manolopoulos, Y., Roddick, J. and Theodoridis, Y. (eds.), *Advances in Spatial and Temporal Databases, Lecture Notes in Computer Science*, Vol. 2750, Springer Berlin, 430-448.

OpenStreetMap Project (2016a) https://www.openstreetmap.org/

OpenStreetMap Wiki, OSM XL (2016b) http://wiki.openstreetmap.org/wiki/OSM_XML

Papadias, D. and Sellis, T. (1994). Qualitative Representation of Spatial Knowledge in Two-Dimensional Space, *VLDB Journal*, Vol. 3(4), 479-516.

Papadimitriou, C., Suciu, D. and Vianu, V. (1996). Topological Queries in Spatial Databases. In: *ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, ACM Press*, 81-92.

Peuquet, D. and Zhan, C. (1987). An Algorithm to Determine the Directional Relationship Between Arbitrarily-Shaped Polygons in the Plane. *Pattern Recognition*, 20(1), 65-74.

Randell, D., Cui, Z., and Cohn, A. (1992). A Spatial Logic Based on Regions and Connection. In: Rich, C., Nebel, B., and Swartout, W. (eds.), *Third International Conference on Knowledge Representation and Reasoning*, 165-176.

Reis, R., Egenhofer, M., and Matos, J. (2008). Conceptual Neighborhoods of Topological Relations Between Lines. In: A. Ruas and C. Gold (eds.), *The 13th International Symposium on Spatial Data Handling (SDH 2008)*, Montpellier, France, Springer, 557-574.

Reiter, R. and Mackworth A. (1989). A Logical Framework for Depiction and Image Interpretation. *Artificial Intelligence* Vol. 41, 125-155.

Renz, J., and Ligozat, G. (2005). Weak Composition for Qualitative Spatial and Temporal Reasoning. In: Goos, G., Hartmanis, J., and van Leeuwen, J. (eds.), *Principles and Practices of Constraint Programming* Vol. 3709, Sitges, Spain, Springer, 534-548.

Rodríguez, M., Egenhofer, M., and Blaser, A. (2003). Query Pre-Processing of Topological Constraints: Comparing a Composition-Based with Neighborhood-Based Approach. In: Hadzilacos, T., Manolopoulos, Y., Roddick, J., and Theodoridis, Y. (eds.), *Advances in Spatial and Temporal Databases*, Santorini Island, Greece, *Lecture Notes in Computer Science,* Vol. 2750, Springer, Berlin, 362-379.

Ruas, A. and Lagrange, J. (1995). Data and Knowledge Modelling for Generalization. In: Müller, J., Lagrange, J., Weibel, R. (eds.), GIS and *Generalization: Methodology and Practice*. Taylor&Francis, Bristol, PA, 73-90.

Sharma, J., Flewelling, D. and Egenhofer, M. (1994). A Qualitative Spatial Reasoner. In: *Sixth International Symposium on Spatial Data Handling*, Edinburgh, Scotland, 665-681.

Schlaisich, I. and Egenhofer, M. (2001). Multimodal Spatial Querying: What People Sketch and Talk About. In: C. Stephanidis (ed.), *1st International Conference on Universal Access in Human-Computer Interaction*, New Orleans, LA, 732-736.

Schneider, M. and Behr, T. (2006). Topological Relationships between Complex Spatial Objects. *ACM Transactions on Database Systems* 31(1), 39-81.

Smith, B. (1995). On Drawing Lines on a Map. In: Frank, A. and Kuhn, W. (eds.), *Spatial Information Theory: A Theoretical Basis for GIS, International Conference COSIT '95,* Semmering, Austria, *Lecture Notes in Computer Science*, Vol. 988, Springer-Verlag, 475-484.

Spanier, E. (1966). *Algebraic Topology*. McGraw-Hill Book Company, New York.

Tarski, A. (1941). On the Calculus of Relations. *The Journal of Symbolic Logic* 6(3), 73-89.

Timpf, S., Volta, G., Pollock D., and Egenhofer, M. (1992). A Conceptual Model of Wayfinding Using Multiple Levels of Abstraction. In: Frank A., Campari, I., and Formentini, U. (eds.), *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*, Pisa, Italy, *Lecture Notes in Computer Science,* Vol. 639, 348-367.

USGS Surface-Water Data for the Nation http://water.usgs.gov/GIS/metadata/usgswrd/XML/nwis_surface_water.xml

Vasardani, M. and Egenhofer, M. (2009). Comparing Relations with a Multi-Holed Region. In: Hornsby, K., Claramunt, C., Denis, M., and Ligozat, G. (eds.), *Spatial Information Theory, 9th International Conference*, *COSIT 2009,* Aber Wrac'h, France, *Lecture Notes in Computer Science,* Vol. 5756, Springer, 159-176.

Vasardani, M. and Egenhofer, M. (2008). Single-Holed Regions: Their Relations and Inferences. In: Cova, T., Miller, H., Beard, K., Frank, A., and Goodchild, M. (eds.), *Proceedings of the Geographic Information Science, 5th International Conference*, *GIScience 2008*, Park City, UT, USA, *Lecture Notes in Computer Science*, Vol. 5266, Springer, 337-353.

Vasardani, M, Timpf, S., Winter, S., and Tomko, M. (2013). From Descriptions to Depictions: A Conceptual Framework. In*:* T. Thenbrink, J. Stell, A. Galton, and Z. Wood (eds.), *Spatial Information Theory COSIT '13*, Scarborough, UK, *Lecture Notes in Computer Science*, Vol. 8116, Springer, 99-319.

Worboys, M. (2012). The Maptree: A Fine-Grained Formal Representation of Space. In: Xiao, N., Kwan, M., Goodchild, M., Shekhar, S. (eds.), *Geographic Information Science*. Columbus, OH, *Lecture Notes in Computer Science*, Vol. 7478, 298-310.

Worboys, M. and Bofakos, P. (1993) A Canonical Model for a Class of Areal Spatial Objects. In: Abel, D., Ooi, B. (eds.), *Third International Symposium on Advances in Spatial Databases*, Singapore, *Lecture Notes in Computer Science,* Vol. 692, 6-52.

Worboys, M. and Duckham, M. (2004). *GIS: A Computing Perspective*. CRC Press, Boca Raton, FL.

Worboys, M., Duckham, M. and Kulik, L. (2004). Commonsense Notions of Proximity and Direction in Environmental Space. *Spatial Cognition and Computation* 4(4), 285-312.

Wuersch, M. (2003). *Perceptual Sketch Interpretation*. Master's Thesis, The University of Maine.

Zubin, K. (1989). Untitled. In: Mark, D., Frank, A., Egenhofer, M., Freundschuh, S., McGranaghan, M. and White, R.M. (eds.), *Languages of Spatial Relations: Initiative Two Specialist Meeting Report.* Technical Paper 89-2, National Center for Geographic Information and Analysis, Santa Barbara, CA, 13-17.

BIOGRAPHY OF THE AUTHOR

Joshua Lewis was born on January 17th, 1986 in Portland, Maine. After graduating from Boothbay Region High School, he attended the University of Maine, receiving a double degree in Anthropology and Political Science in 2009. In May of 2012 he completed a Master of Science in Information Systems degree from the University of Maine and began the Ph.D. program in Spatial Information Science and Engineering that fall.

Coinciding with the end of his MSIS degree and the beginning of his Ph.D. enrollment, Joshua was heavily involved with EqualityMaine, an organization working to further the civil rights of the LGBT population of Maine. He personally spoke to over 5,000 people, visiting over 11,000 homes in central Maine to discuss the importance of equal rights, leading to the only successful referendum in the United States expanding the right to marriage to the LGBT population. Joshua was also involved in protecting Maine's same-day voter registration and various environmental campaigns.

Joshua is a candidate for the Doctor of Philosophy degree in Spatial Information Science and Engineering from the University of Maine in December 2019.