On the Representation of Partial Spatial Information in Knowledge Bases

by

Theodoros Topaloglou

A thesis submitted in conformity with the requirements
for the degree of Doctor in Philoshophy
Graduate Department of Computer Science
University of Toronto

0-612-28307-0

ABSTRACT

On the Representation of Partial Spatial Information in Knowledge Bases

PhD, 1997

Theodoros Topaloglou
Graduate Department of Computer Science
University of Toronto

A fundamental requirement of advanced spatial applications is the capacity to deal with partial spatial information and multiple levels of granularity. This thesis studies the problem of representing and reasoning with partial spatial information in the context of knowledge bases. The thesis proposes a representation which views space as a totality of objects surrounded by a haze area and interrelated in terms of qualitative spatial relations. The most elementary object type in this representation, is the haze point. This is a non-zero sized object that is associated with an area of haze such that the point in question may be located anywhere inside it. Haze points are related in terms of an indistinguishability (called haze) or an order relation. The notion of haze can help us model situations where information is imprecise; the size of the haze area accounts for the degree of precision.

In the course of our study we present a formal axiomatization of the first-order theory of one-dimensional haze point space and develop several extensions of the theory for high dimensional space. We then define a set of topological and directional binary spatial relations in terms of the haze and order primitive relations and formalize spatial inferencing in a setting of varying degree of precision, as a constraint reasoning problem. Our reasoning algorithms make use of a data structure called haze-order graph which allows trading space for efficiency. Experimental results illustrate the efficiency of the proposed algorithms. Finally, we use these results in the development of a spatial data model which facilitates the representation of and reasoning with various forms of qualitatively and quantitatively incomplete spatial information, including indeterminate objects, multiple scales and granularity.

For my parents, Kosta and Hrysoula, with love

Αφιερωμένο με αγάπη στους γονείς μου Κώστα και Χρυσούλα

ACKNOWLEDGEMENTS

This dissertation is the product of the long educational exercise that I have been through in the last many years of my life. This exercise was full of adventure, full of knowledge and full of people who taught me, who encouraged me and who shared with me. I would like to thank each one of them.

First, I would like to thank my supervisor, Professor John Mylopoulos, for all his support and guidance throughout my graduate studies and the development of this dissertation. John inspired me both as a scientist and as a person.

I also want to thank my co-supervisor, Professor Janice Glasgow, for her continuous support and the confidence she showed in me and my work.

I wish to thank the members of my dissertation committee, Professors Alberto Mendelzon and Evangelos Milios for being available and supportive. The feedback they provided was invaluable in the development of this work. Sincere thanks go to my external examiners, Professors Christian Freksa and Vince Robinson. Professor Freksa got into deep questions and provided important feedback that led to significant improvements in the dissertation.

I am grateful to my friends and colleagues, Manolis Koubarakis, Dimitris Plexousakis and Dimitris Achlioptas for reading and critiquing my dissertation. Manolis, Dimitris and Dimitris, along with Nikos Liodakis, Olga Tsipou, Eftichis Sartzetakis, Costis Vezeridis and Panagiotis Louloudis are the people who stood by me during the good and the bad moments. I also want to thank my colleagues and lab-mates at the Knowledge Base Systems Laboratory at the University of Toronto, Brian Nixon, Adel Shrufi, Vinay Chaudhri, Martin Stanley, Igor Jurisica and Nikos Zahariadis. I should not forget to thank my many friends in Toronto for the great time that we had.

I have no words to thank my parents and my brother Niko for their love, support and encouragment through these years. A deep thank you to Karen, my partner and friend, for her love, patience and faith in me.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The need to represent and manipulate spatial information arises in many areas of computer science, including Artificial Intelligence and Databases.

In Artificial Intelligence, problem solving systems require sophisticated world models that can capture the notion of space. Likewise, high-level Machine Vision is interested in interpreting visual data on the basis of knowledge about shapes, positions and motions of objects. In a similar vein, commonsense reasoning about physical systems requires a rich geometric vocabulary and a powerful spatial reasoner since the behavior of many physical systems strongly depends on their spatial layout.

In Databases, the modeling, storage and retrieval of geometric data is becoming an important issue, particularly under the light of growing application areas such as Computer Aided Design/Manufacturing (CAD/CAM), Geographical Information Systems (GIS), and Multimedia applications.

Because the requirements for spatial information processing have arisen from many diverse domains, the models and techniques proposed for spatial representation and reasoning vary significantly depending on specialized concepts and solutions which apply to specific domains. There is much to be gained by integrating aspects of these various solutions into a coherent framework.

In this dissertation we propose an approach to the problem of spatial representation and reasoning support for advanced information processing tasks. The approach consists of a systematic integration of one or more abstractions of space, their underlying representational structures and reasoning algorithms, and an extensible data model which can represent information about spatial and non-spatial entities. In the rest of this chapter we identify the forms of spatial knowledge and spatial reasoning that we study in the dissertation, we state the research questions that we address, and summarize the contributions and outline the structure and contents of the dissertation.

1.2 The Challenges of Spatial Information

What are the forms of spatial information, and what kinds of problems arise when we try to represent it? Consider as an example a computerized system intended to coordinate first-aid vehicles that cover a geographic region (say, Metropolitan Toronto or Metro, for short). The region is divided into sections, which are further subdivided into subsections. For the coordinator and the vehicle drivers, *landmarks* serve as "constants" whose locations are precisely known by all concerned. Other spatial information is represented relative to landmarks. Each vehicle, V, has a spatial range of activity, denoted by *scope(V)*, which is the area that the vehicle can reach within, say, 2 minutes from its current position. Each vehicle reports its position to the coordinating station in imprecise terms (for example, "moving east", "at High Park", etc.). Hence, the vehicle position is represented by an *indeterminate* point. The scope of a vehicle is also partly known and is therefore represented by a rectangular region with an indeterminate boundary. Finally, the location of the trouble spot is reported by the coordinating station in imprecise terms, often through a reference to the nearest street intersection, and is represented by yet another indeterminate point. The reader should notice that indeterminate spatial objects in the example (vehicle positions, vehicle scopes and trouble spots) can be thought of as spatial variables which can take as values spatial positions within some spatial region. Moreover, indeterminate spatial objects are specified in varying degrees of precision, and for some of them only qualitative relationships may be known.

Figure 1.1(a) depicts the Metro region divided into sections. This information can be extracted from a city map and can be as precise as desired. To keep the example simple, we focus on the downtown section which is further subdivided into subsections (see figures 1.1(b-c)). The dividing lines are major streets of downtown Toronto. The different drawing styles distinguish between landmark objects and indeterminate objects. Figure 1.1(d) shows

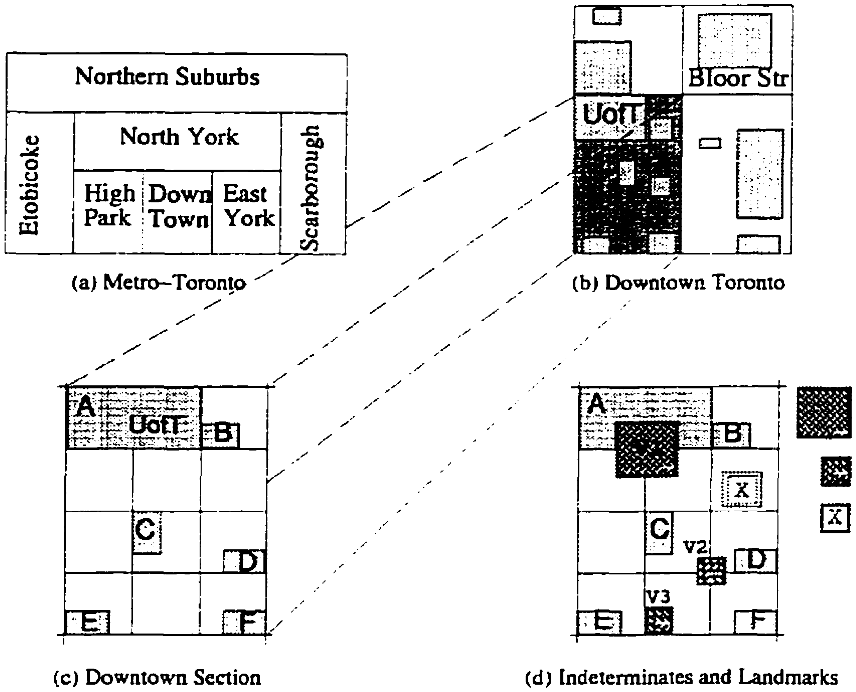


Figure 1.1: An example map at different scales

three vehicles and their scopes, along with a trouble spot (denoted by X).

Several forms of partial spatial information are revealed through this example.

- *Incomplete knowledge* – Knowledge about spatial individuals and spatial relationships is partially specified, e.g., "vehicle V_2 is either south or south-east of the University campus".
- *Imprecision* – Information about measurable entities is approximate, e.g., "the distance between places E (CN-Tower) and F (Union Station) is about two blocks".
- *Granularity* – Domain descriptions are specified in variable degrees of detail (*precision-related granularity*), or at different scales (*scale-related granularity*). In figures 1.1(a), (b) and (c) the scale changes from coarser to finer grain and so does our ability to talk about the details of the configuration.

The distinction between precision-related and scale-related granularity is fundamental. The former notion of granularity interferes with the resolution of the domain, e.g., highway distance is expressed in kilometers, any distance smaller than a kilometer is approximated by either 0 or 1 kilometers. The later notion refers to the multiple values that a spatial property might have depending on the system of reference, e.g., the highway distance between cities A and B is either 500Km or the distance of three counties, whatever that means.

Representing and reasoning with partial spatial information of the kind suggested by this example is one of the major goals of this study.

1.3 Spatial Representations

In general, spatial information in the context of spatial reasoning systems is represented in terms of either *knowledge-level* (also referred as *implicit*) or *symbol-level* (also referred as *explicit*) representations.[1] Implicit representation models, such as general first-order logic, Horn-logic and constraint languages, are common in commonsense reasoning and in reasoning about physical systems because they allow for partially specified spatial configurations. In addition, implicit representations capture relevant facts about the world. The explicit models of space are representations such as digitized maps and image arrays. Explicit representations are common in computer vision and image understanding, and in spatial databases. Their characteristic is that they assume complete information about space, which nevertheless is not always precise.

Another common distinction between representations of space is *qualitative* versus *quantitative* models. Qualitative representations focus on spatial features that are essential and have to be explicitly expressed. Quantitative representations express values of spatial properties, such as location, with respect to a predefined numerical scale.

These two classifications of spatial representations are orthogonal. An implicit representation can be expressed in terms of a qualitative language, e.g., topological relations, thus resulting in an *implicit-qualitative* representation. Alternatively, an implicit description might be relating an implicit spatial object [2] to a spatial landmark, hence defining an *implicit-quantitative* representation. Similarly, the deductive closure of all qualitative rela-

[1] We have adopted Allen Newell's *knowledge-level / symbol-level* distinction in representational systems [New82] as the most general terminology that subsumes Sloman's [Slo85] distinction between *propositional (or fregean)* and *analogical* representations [Slo85], Fleck's *symbolic / concrete* distinction of spatial representations [Fle87] and Chandrasekaran and Narayanan's *implicit / explicit* distinction of visual representations [NC91]. The key to understanding this less than clear-cut dichotomy is the way the structure of the represented world is mapped to the represented world. Knowledge-level representations need to define a mapping (interpretation) between the representation symbols and the objects in the represented world where in symbol-level representations the represented world is directly depicted by the representation.

[2] By an "implicit spatial object" we mean an object whose spatial extension is subject of an interpretation process, i.e., a mapping to concrete spatial location.

Implicit + Qualitative	Explicit + Qualitative
Implicit + Quantitative	Explicit + Quantitative

Table 1.1: A taxonomy of spatial representations

tionships between spatial objects in a certain chunk of space may be regarded as an *explicit-qualitative* representation (sometimes referred to as a cognitive map), and a numeric map or digitized image is regarded as an *explicit-quantitative* representation. A detailed review and a complete taxonomy of spatial representations is presented in Chapter 2. In this section, we simply use the composition of these two taxonomies to classify spatial representations that facilitate the forms of partial spatial information mined out of the example in Figure 1.1. Table 1.1 shows this composition.

Each entry in Table 1.1 stands for a representation that accommodates some form of the spatial information that is revealed in example of Figure 1.1:

- Indeterminate objects as well as topological or directional relationships between them are expressed using qualitative spatial constraints, i.e., an *implicit-qualitative* representation.
- Relationships between indeterminates and landscapes are expressed using spatial constraints with more quantitative flavor, i.e., an *implicit-quantitative* representation.
- A state of the system where all the implicit relationships between objects in a scene are computed forms an *explicit-qualitative* representation.
- Maintaining a computerized map of the entire area, or map fragments at various scales, conforms to a *explicit-quantitative* representation.

Existing database systems offer good support only for explicit-quantitative representations, such as maps and images. Recently, the integration of declarative constraint languages with databases opened new avenues for representing forms of time and space in databases. This dissertation will follow the same general direction in order to study rigorous and efficient representations of partial spatial information in databases.

1.4 The Problem Statement

This dissertation studies the representation of, and reasoning with spatial knowledge in databases (hereafter *spatial knowledge bases*). This study is based on the premise that the support of spatial information in knowledge bases is a three-facet problem consisting of a representation, a reasoning and a management component. The dissertation attempts to draw an integrated picture of spatial knowledge base systems, as well as to make contributions to each one of the three facets. In particular, this work investigates the following research topics:

- The representation of imprecise and incomplete information in knowledge bases.
- Formal and algorithmic aspects of spatial reasoning about imprecise space.
- Database models that materialize the integrated architecture, and allow for addressing consequent data management questions.

The dissertation consists of two main parts. In the first part, it sets out a formal presentation intended to reveal the theoretical underpinning of spatial knowledge representation and reasoning, as well as to exercise formal tools that have been successful in other related efforts to extend databases, such as temporal databases [TCG+93], and knowledge bases [MBJK90]. The formal tools employed in this part are mathematical logic, model theory and constraint reasoning. The second part of the dissertation touches upon practical aspects of spatial knowledge bases such as experimental performance evaluation of spatial reasoning algorithms, support of spatial relations, and the development of an expressive data model enhanced with facilities for dealing with granularity and scale.

1.5 Contributions of the Thesis

The research contributions of this dissertation span over the three facets of spatial knowledge processing that were identified in the problem statement: representation, reasoning and management.

This dissertation develops a representation which views space as a totality of objects surrounded by a haze area and connected in terms of qualitative spatial relations. A haze point is the most elementary object type in this representation since higher order objects are composed of haze points. A haze point is a non zero-sized object that is associated with an area of haze such that the point in question may be located anywhere inside it. Haze points are related in terms of an indistinguishability (called haze) or an order relation.

A formal treatment of imprecision in one-dimensional and two-dimensional space is presented. Specifically, we develop the first-order theory of one-dimensional space based on haze and order relations. We analyze the theory from the point of view of model theory and we show that its models are homomorphic to partial orders on a discrete domain. We propose a conservative two-dimensional extension of the theory of one-dimensional space, called independent combination, in which the evaluation of two-dimensional operators is reduced to the evaluation of projected one-dimensional operators over two coordinate copies of the one-dimensional theory. From the two-dimensional point theory, and by independent combination, we derive the theory of haze rectangles. Finally, we formalize the notion of varying granularity in a spatial representation.

Algorithms for reasoning about haze-order relations form the third contribution of this dissertation. We develop efficient algorithms for determining the consistency of a set of haze-order relations and deducing new relations from those that are already known. In addition, we define a quantitative index structure that supports constant-time retrievals. Our algorithms make use of a data structure called a haze-order graph, which trades space for efficiency. We implement and experimentally evaluate the performance of these algorithms.

Finally, a spatial data model is defined that facilitates the representation of and reasoning with various forms of qualitatively and quantitatively incomplete spatial information, including indeterminate objects, multiple scales and granularity. Representation of incomplete spatial information is accomplished through a spatial constraint language based on haze-order relations. We identify four reasoning tasks that are addressed during query processing in this representation and we offer efficient processing algorithms for each one of them. Our spatial representation model is integrated with an object-oriented data model by exploiting the meta-modeling facilities of the latter. The resulting spatial data model has unique features that make it applicable to a wide range of application involving imprecise dimensional data, such as temporal databases, genome databases and financial databases. [3]

[3]Time events as well as gemone fragments are one-dimensional entities arranged over an one-dimensional line and related via spatial an order relationships. Both of these data domains are rich in various forms of

1.6 Thesis Outline

The rest of this dissertation is organized as follows. In Chapter 2, we review research in the areas of spatial representations, spatial reasoning and spatial databases. In Chapter 3, we propose and study a representation for imprecise space. In Chapter 4, we present two different algorithms for reasoning about spatial relations in the context of the proposed representation. The implementation and the experimental evaluation of the algorithms is also discussed. In Chapter 5, we introduce a data model based an ideas developed in the earlier chapters. In Chapter 6, we show potential applications of the developed techniques in geographical information systems and genome databases. Finally, Chapter 7 concludes this dissertation with a summary of its contributions and an outlook to future research.

---

uncertainty. Financial data are also presented as two dimensional spatial configurations at multiple scales.

Chapter 2

The State of the Art

Σημείον εστίν, ου μέρος ουθέν	*A point is that which has no parts*
Ευκλείδη, "*Στοιχεία Γεωμετρίας*"	Euclides, *"Elements of Geometry"*
300 *π.Χ.*	300 B.C.

2.1 Introduction

Problem solving with spatial data can be decomposed into three fundamental research components, each one of which forms a field of study on its own right. These three components are:

Representation: We need a formalism which is able to represent spatial objects, their local geometry, their position in space, and spatial relationships among objects.

Reasoning: Given a configuration of objects in space and (some of the) spatial relationships between them, we need to be able to infer the spatial relationship among any pair of objects.

Management: We need to organize the spatial and non-spatial information so that it can be efficiently stored and searched.

In this chapter we present a review of the research issues and the solutions given to each of these components. The interested reader is also referred to survey works dedicated to each specific field of study, such as [MH95] on representation, [Spa95] on reasoning, and [Gut94] on management. In our review we give emphasis to the treatment of partial

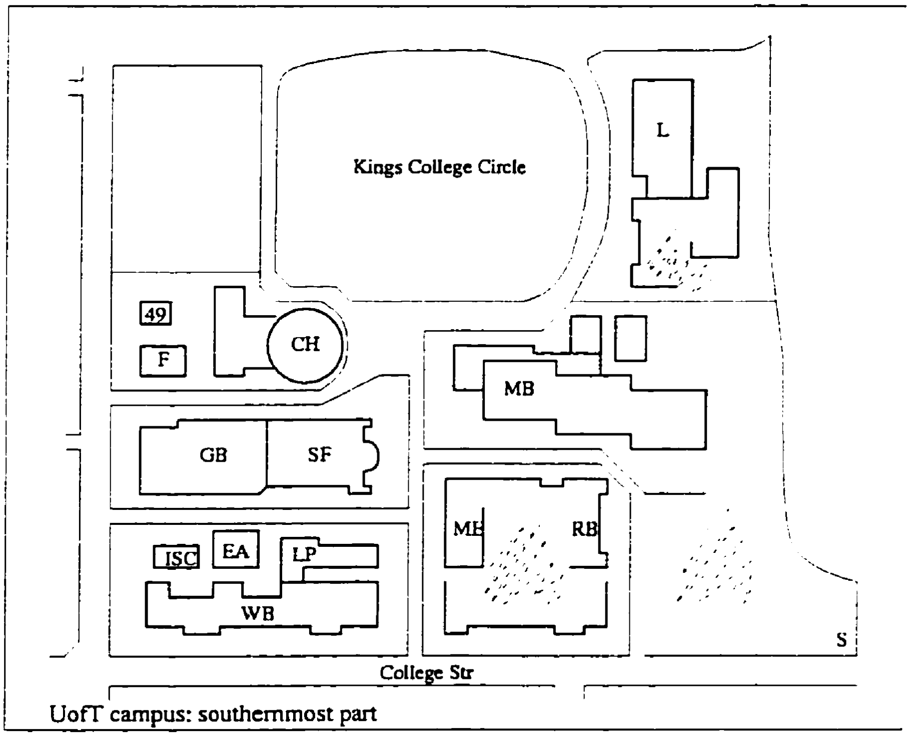


Figure 2.1: University of Toronto campus map

spatial information. The features of the examined methods are studied over the spatial configuration shown in Figure 2.1. This is a two-dimensional projection of a University of Toronto campus map. It depicts the buildings and the streets around the department of Computer Science. Buildings are described by their boundary (spatial description) and their abbreviated name (non-spatial description). Buildings are grouped into campus blocks forming a containment hierarchy. The map contains relative scales of the buildings but not exact metric information. In addition, the map contains various forms of partial information such as unknown areas, like the space behind the ME building, or unrecorded areas such as the top left corner of the map.

The rest of this chapter is organized as follows. Section 2 touches upon some ontological questions which arise in the representations of space. Section 3 reviews representations of spatial configurations. Section 4 presents various spatial reasoning methods. Finally, Section 5 reviews work done on spatial databases including data modeling, query processing and data organization issues.

2.2 Ontologies of Space

An *ontology* underlying a body of formally represented knowledge is defined in terms of two basic kinds of elements: a set of categories, and a set of relations that can hold between instances of these categories. In other words, an ontology determines a set of representational terms. In practice, these terms may not necessarily be primitives of the domain but rather, convenient abstractions which are built out of primitives.

An ontology of some discourse must be epistemologically and pragmatically adequate. An ontology is epistemologically adequate ontology if it can express everything we want to express in the universe of the discourse. An ontology is pragmatically adequate if it can express commonly-expressed things in a relatively easy fashion and support commonly-accepted inferences in a straightforward way.

Historically, the first known ontology of space is the point-based ontology. The first treatises on the study of space and the establishment of geometry date back to Ancient Greece. The Euclidean space is still the most favored abstraction of space in spatial databases. The most primitive notion in Euclidean geometry is the point and the distance between points. Hegel [Heg59], also accepts the *point* as the infinitesimal origin of any spatial configuration. Lines and more complex spatial configurations can be constructed by continuous movements of simpler spatial entities. Aside from point ontologies, solid body ontologies have also been investigated in this century when Tarski developed his geometry [Szc86].

The selection between points versus extended entities is not the only criterion in choosing an appropriate ontology for space. Other criteria are boundedness versus infiniteness, discreteness versus continuity of space, etc. These dichotomies have been the subject of ongoing philosophical debates. In the context of symbolic representations of space, selecting among these alternatives depends largely on specific application domain requirements. For instance, the point-based ontology is preferable in domains where location information is important, whereas the body ontology is preferred in physical systems modeling. Boundedness seems to generally win over infiniteness in spatial representations. Although there is always a point in space which is further than the furthest known point, this is not extremely useful in most applications. For instance, terrains, geometric configurations and land maps are always bounded. The choice between discreteness versus continuity depends on the universe within which spatial properties are being interpreted. Discreteness implies isomorphism to the cartesian product of integers and continuity to that of reals or rationals.

In some philosophical and psychological studies, the notion of relative distance and the notion of contact are treated as primitive concepts that underlie the definition of spatial relations. Van Benthem [vB91] states that the primary spatial operators are comparisons of relative distance. Clarke [Cla85] proposes a formal theory of mereology based on the notion of connection between individuals. The notion of intersection, as another form of contact, is adopted by Egenhofer [Ege89] as a primitive concept for the definition of topological relations.

Practical ontologies of space have been investigated in the context of knowledge representation research. In the CYC project, Lenat and Guha [LG89] consider different ontologies of space from a pragmatic adequacy perspective and discuss abstractions of space which would better serve the tasks that are performed in a CYC knowledge base environment, such as prediction of the behavior of some given device, diagnosis, manufacturing and design. The abstractions they consider include:

Set of points: The basic entities of the domain include points, set of points (lines), forces, masses, velocities, etc. The relations that hold between sets of points may be *spatial* relations, namely, above, below, etc., and *spatio-temporal relations*, such as, connectedTo, looselyConnectedTo, etc. The basic computations used in this abstraction are related to continuum mechanics.

Equations and diagrams: A large class of problems in mechanics uses an equation-level description of systems. Very often, equations are complemented by diagrams. The basic entities here are objects, forces, velocities, etc. The relationships between objects are geometric constraints between specific points on the objects. This abstraction emphasizes the behavioral aspects of a system.

Solids: This abstraction emphasizes the geometric properties of a system. The primitive elements are cubes, cylinders, etc. The relations between instances of these primitives are divided into *spatial* and *spatio-temporal*.

Devices: In a device level abstraction entities correspond to functional primitives; for example wheel, lever, etc. Relations at the device level are very specialized.

Finally, Gruber and Olsen [GO94] have designed a library of ontologies for the purpose of enabling knowledge exchange and reuse. Their work has made available a number of practical ontologies of space including conceptual foundations for physical dimensions and units of measure, in the form of Ontoligua [Gru93] classes and first-order logic axioms. In this study, we will take a similar axiomatic approach to define the abstract properties of an ontology for imprecise space.

2.3 Spatial Representations

Spatial data representations can be studied at two levels of abstraction, the physical and the logical level. At the physical level of a spatial representation one is interested in spatial data structures and access paths. At the logical level one is interested in logical models for spatial information and their associated inference mechanisms. This section focuses on logical representation issues.

At the logical level, the goal of a spatial representation is to establish a mapping between objects in a *modeling space*, M, and their representation in a symbolic structure which is called *representation space*, R. The mapping function s from a subset of the modeling space to a subset of the representation space

$$s : D \rightarrow V, \quad D \subseteq M, V \subseteq R$$

is called a *representation scheme* [Req80]. D is the domain of s and contains all the representable objects, and V is the range of s and it contains all the valid representations. The inverse function of s, $r = s^{-1}$, relates representations to objects. If $r(v)$ is a single element set for any $v \in V$, then the representation scheme s is said to be *unambiguous*. If $s(r(v)) = \{v\}$ for any $v \in V$ then the representation scheme is said to be *unique*.

The modeling space depends on the application area for which a spatial representation is targeted. For instance, the modeling space for CAD/CAM applications is three-dimensional solids. Geographical information systems concentrate on representations of two-dimensional regions. Reasoning about physical systems considers representations of three-dimensional space.

The representation space is characterized by the primitives and the abstractions that the representation structure uses, as well as its operations. In CAD/CAM modeling the representation should be able to handle objects' structure. In geographical information systems, it is important to represent points and regions related through topological and metric relationships. In reasoning about physical systems, aspects such as motion, time and imprecision are important.

The mapping function consists of a notation according to which objects are represented (syntax), and a methodology for associating meaning to the representations (semantics). The mapping function sets a formal framework for describing and evaluating spatial representation schemes.

In the rest of this section, we present a comparative review of spatial representation schemes based on formal properties such as uniqueness, ambiguity and inferential capabilities, but also on informal properties such as easiness of implementation, length of representation and understandability. The presentation focuses primarily upon schemes that are able to accommodate partial spatial information.

2.3.1 Spatial Representations in Artificial Intelligence

Spatial representations have been developed in many areas of Artificial Intelligence, including robot navigation [Kui78, McD80, MD84], qualitative mechanics [For80, FNF87], computational linguistics [Her85], computational imagery [GP92, Gla93b], image interpretation [RM89].

In route finding and robot navigation problems, the objective is to represent large scale space, usually called *working memory*, or *cognitive map*, or simply *map* [Kui78], [McD80], and to capture the ability to upgrade the map from observations. The latter is called the *assimilation* or *map learning* problem [Dav86], [Dut89], [KL88]. Given a map, the navigation problem consists of creating and successfully executing a plan to travel from one place to another.

The modeling space is a two-dimensional floor, called *terrain*. The terrain is either known or partially known (unexplored). The representation considers topological and geometrical properties of the objects in the terrain such as shape and position of an object in a room, position of the walls, etc. Non-spatial properties such as the color of the walls are not considered. Solid objects are represented by a two-dimensional projection on the terrain. In general, objects in a terrain are viewed as obstacles and thus the representation of their outline is more useful than their internal structure.

Kuipers and Levitt [KL88] defined a four-layer semantic hierarchy of descriptions of large scale space that supports map learning and navigation. The four layers are the sensorimotor, the procedural, the topological and the metric layer. The last two layers are relevant to the focus of this discussion. The topological description is a description of the environment in terms of fixed entities, such as places, paths, landmarks and regions linked by topological and order relations. The metric description is a description of the environment in terms of fixed entities, such as places, paths, landmarks and regions linked by metric relations such as relative distance or relative angle with respect to a frame of reference.

The spatial representation models developed for the robot exploration and the map learning problems range from analytic and quantitative, to qualitative models. The quantitative models are appropriate if exact metric information about the terrain is known (e.g., Configuration Space [LP81], Voronoi Diagram [Mil85], Polygonal Region Model [Mil85], etc). The advantage of these models is that they are supported by strong mathematical models, such as computational geometry and linear algebra. Their disadvantage is that are limited to a very specialized type of supported operations.

Qualitative models ([KB88], [KL88]) are better in describing environments with metrical inaccuracy. The TOUR model [Kui78] is an early, but very influential, spatial representation that is classified as a qualitative model. The TOUR model distinguishes between topological and metrical spatial information. Its topological component consists of a cognitive map represented as a network of places and paths linked by containment and boundary relations. The metrical component refers to the quantitative information that is integrated into the map. The metrical information is expressed either in terms of local geometry at places along paths, or in terms of local orientation frames with respect to a global frame of reference. Non-geometric knowledge about a particular environment is recorded in attribute/value pairs format.

McDermott and Davis [McD80, MD84] take an approach which combines the qualitative and the quantitative approach in cognitive map representations. According to this approach, a cognitive map consists of two components: a propositional component and a fuzzy map. The first corresponds to the topological level and the second is a repository of metric knowledge. This approach is an attempt to handle imprecision in spatial descriptions. Other interesting features include the support of multiple frames of reference and the support of object shape. Shapes are represented by a prototype and a modification. Their system, SPAM, implements a data structure along these lines. SPAM has a non-trivial query processor which is capable of three retrieval modes. Retrieval of assertional knowledge is handled by a special purpose theorem prover. Retrievals from the fuzzy map, term values and truth tests, are transformed into numerical optimization problems. The retrieval of objects with specified properties is handled by a structure which is called a discrimination tree which uses both qualitative and quantitative information. Performance and representational problems, observed in SPAM, are studied further by Davis [Dav85, Dav86] in the MERCATOR program.

Glasgow proposed a representation of cognitive maps based on symbolic arrays [Gla93a]. Symbolic arrays are nested structures where elements of the array denote meaningful components of a visual scene. In symbolic arrays, space is partitioned relative to the landmarks that it contains. Symbolic arrays provide an implicit representation of spatial and directional relations. An advantage of this approach is its ability to capture multiple levels of abstraction.

From the above classes of spatial representations, the qualitative and qualitative / quantitative cognitive maps are distinguished as more interesting from the viewpoint of this study.

2.3.2 Representation of Objects

In applications such as mechanical CAD/CAM, architectural design, and computer graphics, spatial representations focus on the representation of objects [RV82], usually three-dimensional solids. The representation of an object must be invariant of shape, location and orientation, occupy a finite portion of space, be finitely describable, have unambiguously defined boundaries, and finally must produce other objects when motion and boolean operations are applied to it.

Representation schemes for solid objects are distinguished into elementary and hierarchical [Gue88]. In a hierarchical representation scheme, the objects are represented by some combination of simpler objects of the same dimension. Elementary representation schemes cannot represent objects as compositions of simpler objects.

Boundary representations are the most noticeable elementary representation schemes because they find applications in the database modeling field [KW87], [AOG+88]. An object is represented by segmenting its boundary into a finite number of bounded subsets usually called faces. Each face is described by its bounding edges and vertices (see Fig-

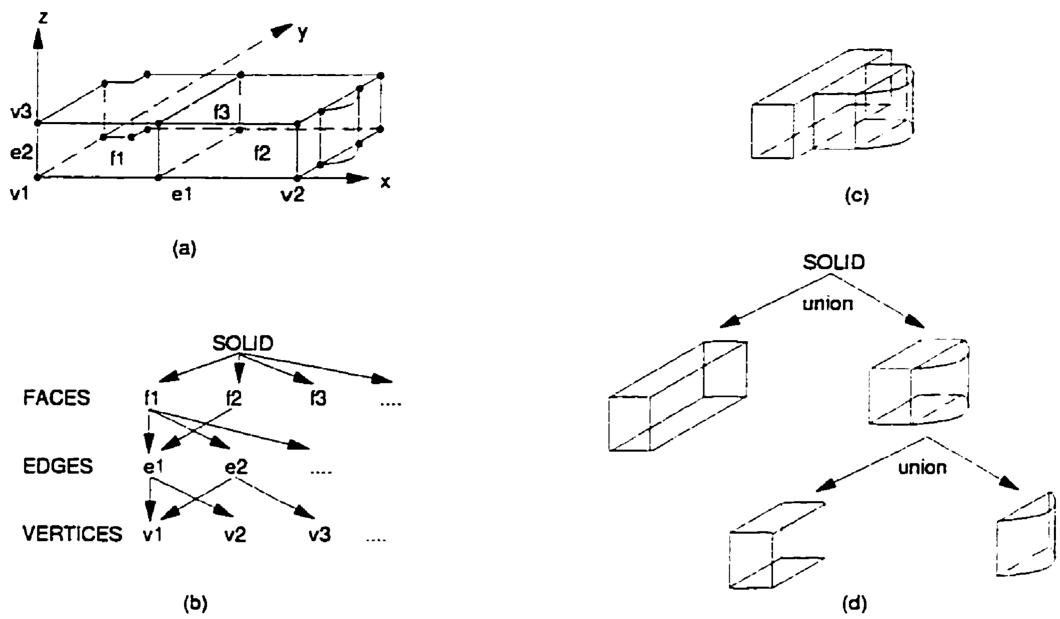


Figure 2.2: Representations of solid objects

ure 2.2(a-b)). Other boundary representations are vertex lists for general polygons and wireframe representations for 3-dimensional objects [Gue88]. Boundary representations are unambiguous if their faces are represented unambiguously. In general, they are not unique and cannot be used to represent objects with holes. They are not good for representing objects with concave faces. However, they are simple and easy to understand.

The most common hierarchical representation schemes are the *spatial occupancy* scheme, the *cell decomposition* and the *constructive solid geometry (CSG)* [Gue88].

In the spatial occupancy scheme an object is represented by a list of all spatial cells (voxels) that it occupies. Cells (voxels) may be cubes of fixed size and lie in a fixed spatial grid called spatial array. This scheme is unambiguous, unique, but quite verbose.

In the cell decomposition scheme an object is decomposed into cells which must be either disjoint or meet precisely at a common face. Quadtrees and octrees [Sam84] are examples of representation schemes which follow the hierarchical decomposition principle. Quadtrees are discussed in detail in section 2.3.3. Cell decompositions are unambiguous and non-unique.

The constructive solid geometry scheme represents solid objects in terms of a set of 3-dimensional volumetric primitives (blocks, cylinders, cones and spheres are typical examples) and a set of operators (set operators such as union, intersection, difference, and similarity operators such as rotation, translation and scaling). An object is represented by a binary tree (the CSG-tree) whose interior nodes correspond to operators and whose leaves correspond to primitive components or numerical arguments used by the operators. Figure 2.2(d) shows the CSG tree for the solid of Figure 2.2(c). CSG is an unambiguous but not unique object representation scheme. The difficulty of implementing search operators in CSG is its strongest drawback. Nevertheless, CSG is widely used in current CAD/CAM systems [Wil88].

2.3.3 Representation of Regions

Symbolic representations of regions are important in geographic information systems and pictorial databases. The following summarizes some of the requirements of region representations: The modeling domain is two-dimensional. Regions might have arbitrary complex shapes, and their dimensions might be known in limited or full precision. Usually, regions are represented in order to be stored in a database. There are a vast number of spatial operators that need to be supported: geographic operators, e.g., north-of, east-of, same-position-as, etc., local operators, e.g., overlap, adjacent, includes, etc., set operators, e.g., spatial union, intersection, difference, and similarity operators, e.g., rotation translation and scaling.

There are two approaches to region representation: those that specify the boundaries of a region and those that focus on the interior of the region.

In this section we survey four different region representation schemes. These are the *polygonal representation* scheme, the *symbolic projection* scheme, the *hierarchical partitioning* scheme and the *space filling curves* scheme. The first two are boundary representations whereas the last two organize the interior of a region.

Polygonal Representations

Polyhedral Chains. Guenther's polygonal (polyhedral) chains [Gue88] and Davis' polygonal approximation method [Dav85] are the most interesting general purpose polygonal representations. The minimum bounding rectangle (MBR) model is a simplified polygonal representation that approximates complex shaped objects by their enclosing rectangle. Minimum bounding rectangles are extensively used in the development of access methods for spatial objects.

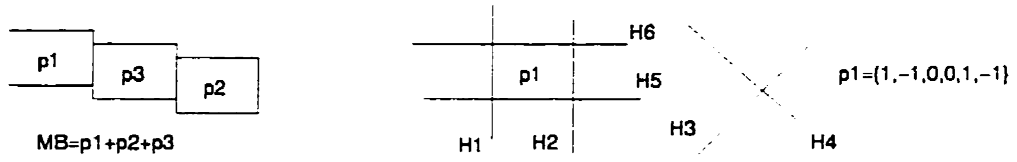


Figure 2.3: Polygonal chain

The polyhedral chains model can represent polyhedral objects of arbitrary dimensionality and arbitrary shape. A general polyhedron is represented as a convex polyhedral chain, that is, the algebraic sum of simple convex polyhedra, called *cells*. For instance, the general polygon of the "MB" building (in Figure 2.1) is written as the sum of three convex polygons, MB $= P_1 + P_2 + P_3$ (see Figure 2.3). The reason for this is that convex polyhedra are closed under all set operations (intersection, union, difference). Convex chains can be viewed as a special case of constructive solid geometry (CSG). Like CSG, the convex chains approach is a hierarchical representation scheme for polyhedra which is unambiguous but not necessarily unique.

Convex cells are represented by means of *h-vectors*. A convex polyhedron in d-dimensional space, E^d, is represented by the intersection of halfspaces in E^d. Each halfspace is viewed as a product $h \cdot H$ where H is an oriented (d-1)-dimensional hyperplane and h is one of {0,1,-1}. In particular, $-1 \cdot H$ defines the halfspace which stands left of H, $1 \cdot H$ the right halfspace and $0 \cdot H$ the entire E^d. If $\overline{H} = H_1 H_2 \ldots H_{|\overline{H}|}$ is the list of all (d-1)-dimensional hyperplanes such that for each face f of any spatial object there is a hyperplane in $\overline{H}$ that embeds f, then each polygon p is represented as a ternary vector, called h-vector, $h_p = \{0, 1, -1\}^{|\overline{H}|}$ such that $p = \bigcap_{i=1}^{|\overline{H}|} (h_p) H_i$. Likewise, the two-dimensional polygon p1 in Figure 2.3(b) is represented by the h-vector $\{1, -1, 0, 0, 1, -1\}$. It must ne noted that this approach abandons completely the notion of vertex in representing polyhedral objects.

The MERCATOR Representation. The representation model developed in the MERCATOR system [Dav86] deals with two-dimensional objects of arbitrary shape. In particular, the MERCATOR representation provides facilities for the representation of shape, multiple description, measurements at varying degree of precision, and approximate reasoning support.

MERCATOR's representational primitive is the line segment. Two-dimensional geometry is represented by straight line segments. Objects are represented by their boundaries and the description of their interior. An object's boundary is a set of edges (segments)

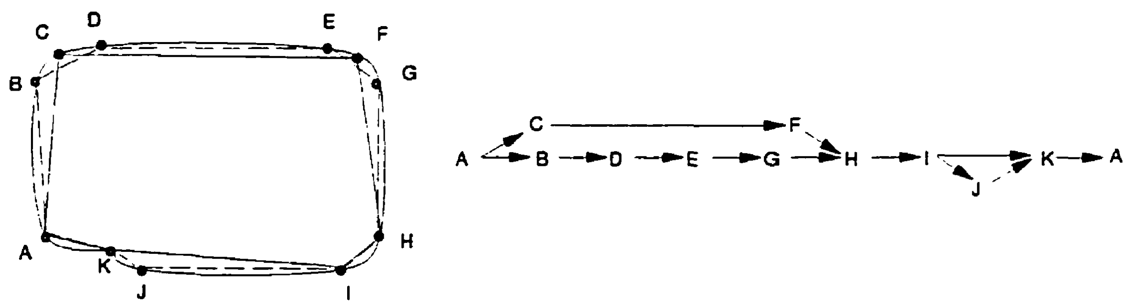


Figure 2.4: Multiple region representation

connecting vertices. An object's interior is a set of convex polygons. There is no notion of coordinate system because this presupposes full knowledge about space; instead, relative positions are supported. Local dimensions are recorded in terms of lengths and orientation of edges. To account for imprecision, MERCATOR uses range values expressed in a measurement scale of some *grain size*. A given object may have several region descriptions (which vary in the grain size) as shown in Figure 2.4. Objects in a map are hierarchically organized in a *containment* hierarchy. The non-geographic properties of objects are not part of the MERCATOR representation; these are recorded in a slot-filler pair representation scheme.

Space representations in MERCATOR are called *maps*. Maps are valid descriptions of the world in which an intelligent agent is situated. In such a setting, the MERCATOR representation suffers from three basic inadequacies. First, it can express only the presence of an object, not its absence. Second, it cannot express natural combinations of precise shape and imprecise dimensions. Third, it is limited to two-dimensional representations. From a database point of view, the MERCATOR model offers "dream" expressiveness, which unfortunately is penalized by an expensive computational model.

Symbolic Projections

The symbolic projection scheme was introduced by Chang et al. [CSY87] in the context of pictorial databases. In the symbolic projection scheme, any two-dimensional shape is projected to two strings, called *2D-string*, along the vertical and the horizontal direction. It is an approximate representation scheme because the size of the picture objects is known within a precision threshold that is determined at the creation time. It is also ambiguous

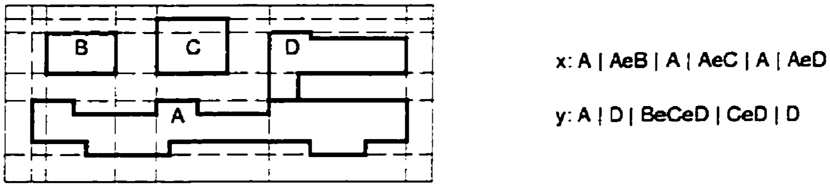


Figure 2.5: 2D G-string representation

since given a 2D-string many possible pictures can be constructed [1]. The original 2D-string supported only two spatial operators "<" (for right/left, bottom/up characterizations) and "=" (denoting same-position). Several extension of 2D strings have been proposed over the years, including the 2D H-strings [CL88], a hierarchical extension of 2D strings, and the 2D G-string [Jun88] which adds local operators such as the edge-to-edge operator, |. Figure 2.5 illustrates a 2D G-string representation of a part of the UofT map.

2D strings can support geographic inferences. For instance, the geographic relationship north-of between two objects o_1, o_2 is formulated by the rule:

$$\text{if } X : o_1 = o_2 \text{ and } Y : o_1 < o_2 \vee o_1 | o_2 \text{ then } (north, o_1, o_2)$$

It should be noted that the database representation of space proposed in this dissertation is an extension of the hierarchical symbolic projection scheme for multi-scale space.

Space Filling Curves

Space filling curves [Lau85] allow a one-dimensional representation of any kind of d-dimensional data. In particular, there exists a bijection between a k-dimensional space to 1-dimensional space by the means of a curve which passes only once through any point in the space. Any point on a curve is assigned a number, called *order number*. Continuous regions of space are mapped into a sequence of curve segments or continuous segments and therefore are characterized by a set of order numbers. The string length of the order number determines the resolution of the represented space. The Peano curve or *z-ordering* [OM84] is the most common space filling curve. Other known curves are the Gray-code curve [Fal88], and the Hilbert curve [FR89].

Space filling curves can support proximity searches and point-in-polygon queries. Space filling curves have been used extensively in spatial data handling systems. In particular, Orenstein and Merrett [OM84], [Ore86] applied them in the development of spatial access

[1] This claim is not true for the extended 2D strings [Jun88]

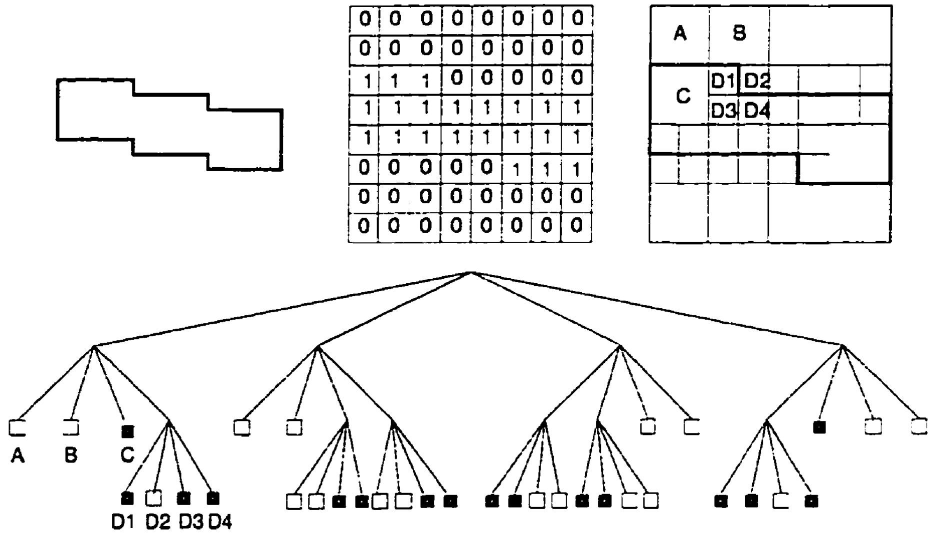


Figure 2.6: Region quadtree

methods, Orenstein and Manola [OM88] used z-orderings for the representation of spatial objects. Hilbert and Gray-code curves have also been used as spatial access methods [Fal88], [FR89] and [Jag90].

Among the representations of spatial content (objects in a spatial configuration), space filling curves are best for indexing spatial objects and provide fast access to them.

Hierarchical Partitioning

The hierarchical partitioning of space is the basis for a whole class of methods for representing and organizing spatial data. The *quadtree* [Sam84] is the most common representative of this class. Many quadtree variants have been proposed which differ according to the type of data they represent and their space partitioning method [Sam89].

The quadtree approach to region representation is called *region quadtree* (see Figure 2.6). The region quadtree performs successive subdivisions of a bounded (binary) image array into four equal sized quadrants. The quadrants that do not entirely contain 0 or 1 are further partitioned, in the same way. The region quadtree is a variable resolution representation method. The quadtree can also be extended to represent 3-dimensional binary region data. In this case, the resulting representation is called an *octree*.

The quadtree is a tree structure which admits a straight forward pointer-based implementation (i.e, non-leaf nodes are represented as records with four pointers to their children). However, pointer-based representations of quadtrees require a considerable amount of space, and thus large images cannot fit into core memory. Consequently, there has been a vast interest in pointerless representations [Gar82], [ABJN85].

Quadtrees are useful for performing set operations such as union (overlay) and intersection of several descriptions of the same region. The required time for these algorithms is proportional to the minimum of the number of nodes at the corresponding levels of the two quadtrees. It also supports operations that are common in computer graphics such as scaling by a power of 2 and rotations by multiples of 90 degrees. Area calculations are extremely easy, too. Lastly, it can be used as an image approximation device. Although the quadtree is rather an expressive representation, its dependence to main memory structures does not make it a good representation for large scale space.

2.3.4 Summary

We have presented several approaches to the representation of spatial information and a framework for characterizing them. Our presentation is clearly not exhaustive. The wide spectrum of uses for spatial information makes the enumeration of all different spatial representations a very challenging task. This presentation has focused on representations that are interesting from the knowledge representation and database perspectives. For an additional survey of the representations of spatial knowledge, the interested reader is referred to the work of Mukerjee and Hernandez [MH95]. A survey of spatial representations for databases, termed *geomatic models*, is presented by Paradaens [Par95].

Table 2.1 summarizes the results of our presentation. The abbreviated column headers UM, UQ, PK and IM stand for unambiguous, unique, partial knowledge and imprecise measurements respectively. The table also shows the modeling and the representation space for each representation scheme as well as the most important operations that each model supports.

2.4 Spatial Reasoning

In this section we review formalisms for spatial knowledge which are suitable for spatial reasoning. Spatial reasoning is a field which has defined itself over the last few years as researchers from a variety of subject areas have recognized the importance of automated

Representation Scheme	Application	Modeling Space	Representation Space	UM	UN	PK	IM	Coordinate System	Operators
TOUR [Kui78]	robotics	2D terrain	network of places and paths	No	No	Yes	Yes	many local, one global	containment, metrical
SPAM [MD84]	robotics	2D terrain	discrimination tree	No	No	Yes	Yes	many local	relative position
MERCATOR [Dav86]	robotics, cognitive maps	2D terrain, shapes	polygonal approximation	No	No	Yes	Yes	No	relative position, neighboring
Boundary Representation [KW87]	database modeling, geomatics	3D objects	set of polygons	Yes	No	No	No	No	set operations (union, intersection, difference), translation, rotation, scaling
Spatial Occupancy [GP92]	cognitive maps, geomatics	kD space, regions	kD arrays, quadtrees	No	Yes	No	Yes	Orthogonal	
Constructive Solid Geometry [Req80]	solid modeling, geomatics	3D objects	set of primitive solids	Yes	No	No	No	No	
Polyhedral Chains [Gue88]	solid modeling, geomatics	kD objects	polyhedral chains	Yes	No	No	No	No	
Quadtrees [Sam84]	geomatics	2D space, regions	quadtrees	No	Yes	No	Yes	Orthogonal	overlay, rotation, scaling
Symbolic Projections [CSY87]	image indexing	2D space, regions	2D strings	No	Yes	No	No	Orthogonal	spatial order
Space Filling Curves [OM84]	geomatics	kD space, regions	1D curves	Yes	Yes	No	Yes	Orthogonal	overlap, containment

Table 2.1: A classification of spatial representations schemes

reasoning about spatial relations between physical objects or regions of space. Examples of the kinds of questions for which spatial reasoning is required, include:

- Compute the relative position between two entities in space.
- Find whether an arrangement of entities in space is consistent with respect to a set of topological and geographic relationships that must hold between them.
- Find the route from point A to point B.
- Identify the spatial entities appearing in a certain range of space.

The following is a list of desirable requirements that a spatial reasoning framework can have:

- It must be capable of representing and reasoning about a variety of spatial constraints expressed in a qualitative or quantitative language.
- It must be capable of inferring spatial consequences given some specification of spatial and motional relationships.
- It must be capable of reasoning with partial information.
- It must be capable of reasoning about objects of various shapes and varying granularity.

In many cases, reasoning about the spatial relation between physical objects can not be done without precise quantitative information about these relations. The ability to reason with *partial* information is the essential requirement in this study. In the example of Section 1.2, we have identified several forms of partial spatial information, including *incompleteness*, *imprecision* and *granularity* related deficiencies. In the rest of this section we elaborate on methods that enable spatial reasoning with partial information.

Dealing with incomplete spatial knowledge is, in many respects, similar to the problem of incompleteness in symbolic knowledge representation. Techniques such as *completion assumptions* and *persistence rules* can be used for its solution. For instance, if it is stated that "object A is either to the left or to the right of object B", and there exists a completion assumption saying that "nothing exists to the right of the wall and B is the wall", then we can infer that "A is to the left of B". Similarly, if it is known that "regions C and D

are disjoint", then it can be assumed that "they have equal size" until further information is learned (persistence rule). These two examples do not really propose a solution to the spatial incompleteness problem. There is a lot more to be said if the properties of the spatial ontology are assessed. Nevertheless, they connect this problem to a mature body of work that exists in Artificial Intelligence [Lev81, Rei80].

Imprecision of spatial knowledge emanates either from limited accuracy of the perception or the measurement. For example, we can say: "John's height is about 1.80cm". "The measurement of the distance between atoms a_1 and a_2 in some crystallographic structure is between 3 and 5 $\mathring{A}$". "Point A is close/far to/from point B". "Region R is east/west/north/south of region S". Similarities with these types of examples can be found in the field of temporal reasoning [All83] or qualitative reasoning about physical systems [Kui86]. Formalisms such as quantitative and qualitative constraints, fuzzy sets, interval values, etc., are prime candidates for formalizing spatial imprecision.

In this section we review formalisms that are capable of representing and reasoning about spatial imprecision. Some of them formulate the problem as a constraint satisfaction problem over a network of spatial relations in the same way that reasoning with respect to time is formulated. Other approaches involve qualitative reasoning, approximate calculations based on fuzzy numbers, and numerical methods. In some cases, spatial reasoning can be thought of as a generalization of temporal reasoning in a multi-dimensional space.

2.4.1 Temporal Reasoning and Constraint Networks

Time is represented either in a change-based or a time-based fashion [SG88]. In this section we review the time-based approach because of its similarity to the spatial case. In the time-based approach, time is explicitly represented in terms of either points or intervals, and temporal events are related by the means of temporal relationships. A time-based representation can be seen as the one dimensional projection of a spatial representation that involves location and relative position operators. The most common representation for the time-based approach is that of binary constraint networks.

A temporal constraint network is a directed graph where nodes represent temporal entities, points or intervals, and the edges are labeled with temporal relationships holding between the connected nodes. The language used for the edge labels can be quantitative or qualitative, hence the constraint network is characterized as a quantitative or a qualitative network. Temporal reasoning is formalized as a constraint reasoning problem [DMP89], [vB90] or a label inferencing problem [Dav87].

Formally, a constraint network consists of a set of quantities $X_1, X_2, ..., X_n$, where D_i is the domain of each quantity, and a set of unary, $P(X_i)$, and binary, $C(X_i, X_j)$, constraints over these quantities [Mon74], [Mac77]. A solution of the network is a tuple $\{x_1, x_2, ..., x_n\}$ such that the assignment $X_i = x_i$ satisfies all the constraints. A network is consistent if at least one solution exists. A constraint C' is tighter than C'' if every pair of values for C' is allowed by C'', $C' \subseteq C''$. This notion is extended to networks. $\subseteq$ defines a partial order over networks. Two networks are equivalent if they have the same set of solutions. A network M is minimal if there is no equivalent network with tighter constraints.

The basic reasoning problems that are addressed in a constraint network, are: (a) given a network determine whether it is consistent or not, and (b) given a consistent network, compute the minimal network that is equivalent to it.

Allen, in his foundational work on temporal reasoning [All83], introduced a time-based representation based on intervals which are related by thirteen mutually exclusive relations and their disjunctions. These relations are: before, meets, overlaps, starts, during, finishes, equals, finished_by, over, started_by, overlapped_by, met_by and after. Two temporal events, "Anne has breakfast" and "Anne reads her morning newspaper", are represented by temporal intervals T_1 and T_2, and if we also know that Anne never reads while she is eating, then T_1 and T_2 are related by the following Allen's algebra expression

$$T_1 \text{ before } T_2 \vee T_1 \text{ meets } T_2 \vee T_1 \text{ met\_by } T_2 \vee T_1 \text{ after } T_2$$

Allen [All83] presented a constraint propagation algorithm for computing the minimum network equivalent to a given set of interval relations, or reporting inconsistency if it is unsatisfiable. The basic idea of this algorithm is based on previous algorithms for constraint satisfaction problems [Mon74], [Mac77], [MF85]. The algorithm runs in time $O(n^3)$, where n is the number of intervals in the network, however, it is incomplete. The incompleteness of this algorithm is not surprising since Vilain and Kautz [VK86] showed that computing the transitive closure of relations in Allen's interval Algebra is an NP-complete problem. Valdes-Perez in [VP86] showed also that Allen's constraint propagation algorithm is sound but not complete and he developed a dependency-directed backtracking algorithm [VP87] with exponential asymptotic complexity which is complete. In practice, Valdez-Perez's algorithm terminates early because of quick pruning and clever backtracking.

Vilain and Kautz [VK86] defined a temporal representation based on time points, which are related by the three binary qualitative relations $\{<,>,=\}$, and their disjunctive combinations. Then, they claimed that Allen's algorithm computes the minimal network in the point representation. VanBeek showed [vB89] that this is not the case if the network contains $\neq$ relationships. By excluding $\neq$, the source of incompleteness, from the point algebra, vanBeek defined a subclass of the point algebra, called a pointisable class, which is complete. He also presented an $O(n^4)$ algorithm which is complete for the point algebra. Recently, Nebel and Burckert [NB94] presented the maximum tractable subclass of Allen's interval algebra whose satisfiability can be determined by the $O(n^3)$ path consistency algorithm. A similar result was also obtained by Schubert and Gerevini [GS95a]. Various practical algorithms for reasoning in interval algebra [Koo89] and point algebra were proposed in [GA89], [MS88], [GS93].

All the above methods deal with abstract time references. Representations of time may involve absolute references and metric information. For instance, "The next AAAI conference starts on August 4, 1996, and lasts 3 to 5 days". In a quantitative representation of time, this is information is expressed as

$$start(AAAI96) = \text{“4/8/1996”} \wedge (3 \leq end(AAAI96) - start(AAAI96) \leq 5)$$

Dechter, Meiri and Pearl [DMP89], [DMP91] studied quantitative and metric temporal networks of the form

$$a_1 \leq T_i - T_j \leq b_1 \vee \ldots \vee a_n \leq T_i - T_j \leq b_n$$

where T_i and T_j are time points and $a_1, \ldots, a_n, b_1, \ldots, b_n$ are real numbers. Deciding the consistency of quantitative point constraint networks is also an intractable problem. A tractable subclass occurs when all constraints have only one disjunct, i.e., $a \leq T_i - T_j \leq b$. In this case, deciding consistency and computing the minimal network is done in $O(n^3)$. Dean [Dea89], [DM87] presented a linear time solution for a special case of the [DMP89] network, which is called distance graph. Dean's linear algorithm is based on indexing and caching of time intervals. Davis [Dav87] determined that the threshold from polynomial to exponential complexity in quantitative constraint networks, is the linear inequality constraints.

The combination of qualitative and quantitative constraints was studied by Kautz and Ladkin [KL91], Meiri [Mei91], and several tractable cases were identified.

2.4.2 Spatial Reasoning and Constraint Networks

Extending temporal representations to k dimensions has been a popular approach in spatial representation and reasoning [MB83], [VP86], [Lig90], [MJ90], [Gue89]. The main advantage of this approach is the reuse of the solid body of work developed for time. The adoption of temporal representations for representing space restricts reasoning to orthogonal domains. The key properties that time and space share, and which make methods from temporal reasoning valid for spatial reasoning, are order and strictly increasing continuity (in the calculus sense) [VP86]. Both of these properties are possessed by the orthogonal cartesian axes. Nonetheless, space, unlike time, is static in our world-view; therefore the provision for persistence that is made in many temporal representations, specifically in the change-based ones, does not carry for spatial representations, unless a notion of time is explicitly used, e.g., motion.

Allen's framework can directly support spatial reasoning in one dimension. Relationships such as front-of, back-of, and inside correspond directly to after, before and during. Genome maps [Fre91] and molecular sequences are typical domain examples which require one-dimensional spatial reasoning.

One-dimensional spatial relations based on Allen's framework were developed by Mukerjee and Joe [MJ90] and Guesgen [Gue89]. Mukerjee and Joe used end-points to express the thirteen Allen's relations, and further, they identified five point-to-interval relations: ahead, front, interior, back and posterior. Their reasoning algorithm is based on a composition table which determines the relative position between spatial entities A and C given the relative positions of (A, B) and (B, C). The spatial relation between two objects not orthogonally oriented is described by a pair comprised of relative positions and relative directions.

Guesgen, proposed a set of spatial relations, namely, left-of, attached-to, overlaps, inside and their inverses, based on a simplified set of interval relations and the underlying reasoning algorithm. Malik and Binford [MB83] came up with a set of "everyday" spatial operators {left_of/right_of, front_of/behind, bellow/above}, as counterparts of the before/after temporal relations for the x-, y- and z-axes by realizing their isomorphism to the *time*-axis. A similar argument is also made by Sistla et al. [SYH94].

All [MJ90, Gue89, MB83, LJ88] suggested that in k-dimensional orthogonal space, the spatial operators can be defined as vectors of length k one-dimensional operators. We call this approach the ***decomposition*** approach. In the decomposition approach, spatial reasoning is reduced to one-dimensional reasoning along k independent dimensions. The alternative to the decomposition approach is called ***unified***. In the unified approach, the representation and reasoning about spatial relations is based on topological properties of space. Clearly, the decomposition approaches have limited expressiveness when applied to higher dimensional spaces: they are restricted to orthogonal domains and rectangular shapes. However, in may applications [SYH94], including databases, their limitations are not restrictive and given their lower computational complexity [GPP95], are realistic choices.

2.4.3 Qualitative Spatial Reasoning

The qualitative reasoning approach is well suited to spatial problems because of its representational power for states of partial knowledge. Qualitative representations "make relevant distinctions" in a given context. In physical systems, "relevant distinctions" mean ordered sets of ***landmark*** values within ***quantity spaces*** of the values of parameters and functions [Kui89]. For instance, the temperature may be measured in the quantity space: {cold, cool, warm, hot}. In a representation of space based on the representation of objects in space and their interrelationships, a qualitative representation typically determines the relevant relations allowed by the representation, e.g., disjoint / not_disjoint. In a holistic representation of space, qualitativeness may be understood as the way in which space is partitioned into zones/areas with some distinct feature, e.g., type of land use. Grid representations are not qualitative because they use a metric condition in their partitioning, e.g, resolution.

In an object-based view of two-dimensional space, a variety of qualitative representations has been proposed. Guesgen's representation uses interval based qualitative relations on individual coordinates in order to capture the spatial relationships between higher dimensional objects. Chang et al. [CJL89] represented the content of an image on the basis of qualitative 2D strings Randell et al. [RCC92a], and Egenhofer [Ege89] developed sets of qualitative spatial relations based on the theory of mereology and topology, respectively. Finally, Hernandez [Her92], Freksa [Fre92b] and Frank [Fra91] proposed methods for qualitative spatial reasoning based on orientation/direction relations. With the exception of the 2D strings, all of these representations perform reasoning based on transitivity axioms.

2.4.4 Quantitative Methods

Quantitative methods in spatial reasoning make use of metrics, absolute values, range values and comparisons. Values are expressed in terms of some absolute unit. Quantitative methods are distinguished as propositional and analogical.

Propositional-quantitative methods are considered the models that formulate descriptions of space-based *metric* constraints on the endpoints of the x, y and z coordinates of objects. The computational methods that are suitable for spatial reasoning, in the context of metric information, include metric constraint networks [DM87, Dea89, DMP89] and linear programming [MB83, McD82, Dav86].

Analogical-quantitative representations such as the occupancy arrays and hierarchical partitioning represent space as a whole, including its objects, using metric information.

2.4.5 Semi-Qualitative Approaches

The combination of qualitative and quantitative representation is attempted in cognitive maps [McD80, MD84, KB88] and in constraint networks [Lad89]. A cognitive map which follows the layered model of [KL88] consists of a quantitative component (metric layer) and a qualitative component (topological layer). Reasoning in cognitive maps (spatial learning, path finding) exploits both representations. Kuipers [KB87, KB88] derives distinctive places on the map using hill-climbing search on the metric layer and then applies qualitative simulation [Kui89]. McDermott and Davis [McD80, MD84, Dav86] transform the problem to a network of fuzzy constraints and apply numerical techniques (Monte Carlo algorithm). The semi-qualitative approach gains support in qualitative kinematics (poverty conjecture) [FNF87].

An integration of the qualitative and the quantitative approach in spatial reasoning is proposed by Dutta [Dut89]. In this method, the relative position of objects is expressed in qualitative terms, e.g., "SF is near LP". Imprecision is expressed using range data, e.g. "the distance between the SF and the LP is between 4 and 6 meters". The general problem is, given a set of objects and a set of constraints (generally incomplete and sometimes conflicting), to find all the induced spatial constraints. Constraints are represented as *possibility assignment equations* of the form $p \rightarrow \Pi_{(x_1,..,x_n)} = F$, where p is a natural language proposition, Π is the possibility assignment and F is a fuzzy subset of the universe of discourse U. For example

$$\text{"Bob is short"} \rightarrow \Pi_{\mathrm{Height(Bob)}} = \mathrm{SHORT}$$

$SHORT$ is a fuzzy subset and $\mu_{SHORT}(Height(Bob))$ is the membership function of $Height(Bob)$ in the set $SHORT$. An approximation in the fuzziness expression is to use fuzzy numbers *(c,l,r)* (i.e., triangular distribution) instead of $\mu(u)$, e.g., about-five-miles=(5,1,1), north-eastern-direction=(45,10,10). For metric constraints the membership function is defined: $\mu_A(u) = 1$ if $u \in A$ and $\mu_A(u) = 0$ if $u \notin A$, therefore the fuzzy set has only one value ($\mu(u) = 1$) and the fuzzy number is *(c,0,0)*, e.g., "Object A is 5 miles NE of object B" $\rightarrow \Pi_{loc(A),loc(B)}=$ (5miles, NE), where, 5miles=(5,0,0) and NE=(45,0,0). The mathematical basis for the spatial reasoning algorithm is the composition of possibility assignment equations and the fuzzy set theory.

2.4.6 Qualitative Spatial Scenes

The common goal of many qualitative spatial reasoning methods is to relate objects in two, three or k-dimensional space, by means of spatial relationships, and to do inferences based on a composition operation (usually transitivity). Such representations are also referred as *spatial scenes* [FR93]. In this section we discuss the two-dimensional case of spatial scenes.

Two factors determine the relative position of objects in two-dimensional scenes: *topological relations* and *directional relations*. Topological relations describe how the objects "in-contact" are related to one another. Directional relations describe how the objects "at-a-distance" are related to one another.

Topological Relations

Topological relations are defined between objects with spatial extension and which are in contact with one another. There are many sets of topological relations proposed [Gut88, Ege89, RC89]. A representational concern in all of them is that the developed set of relations must be complete in the sense that it can describe all the qualitative distinct positionings of two objects in a scene, and, in addition, that these relations are mutually exclusive. Guting [Gut88] presented a set of topological operators based on set theory. Egenhofer [Ege89, Ege91], developed a set of binary topological relations based on the intersection of

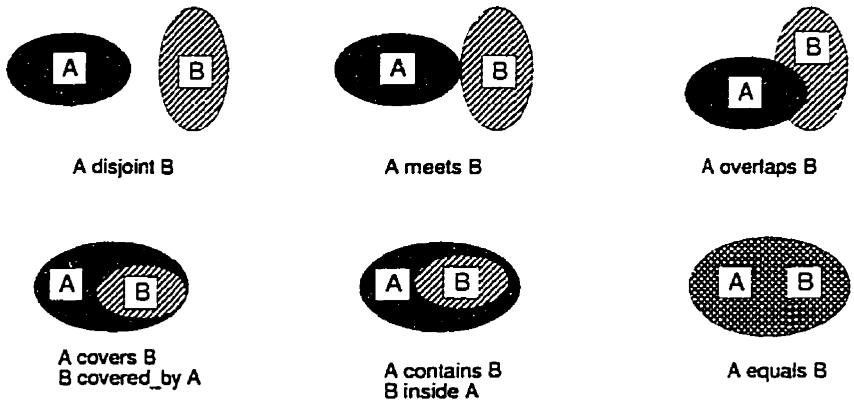


Figure 2.7: Egenhofer's topological relations

the boundaries and interiors of the two objects to be related (see Figure 2.7). Randell and Cohn [RC89, RCC92a], presented an alternative formalization of topological relations based on a primitive dyadic relation: $C(x, y)$, meaning region x connects with region y. These relations are usually referred to as RCC relations (see Figure 2.8).

Hernandez [Her92], Papadias [Pap94a, GPP95] studied transitivity-based reasoning using on Egenhofer's relations. Cui et al. [CCR92] and Bennett [Ben94] investigated the use of qualitative simulation and transitivity tables, respectively, as reasoning method for the RCC relations.

Directional Relations

Topological information alone is insufficient to express positional information, because topology has no means to distinguish two topologically disjoint objects which are the one "on the left of" and the other "on the right of" a third disjoint object. We need the concept of orientation in order to characterize relative positions in this situation. Orientation is the basis for defining directional relations.

Orientation is a property of points or extended objects in a scene. Orientation means that an object has a "point of view". In some approaches [Her92, Fra91], an object inherits its point of view from a global reference system; Freksa and Rohrig [FR93] call it *external* orientation. Other approaches, e.g., [Fre92b, LR93] assume that orientation is a local property. For instance, the orientation of an object is locally determined by some property of the object such as the "front of the building", or is determined by the position of a particular external viewer of the object. Figure 2.9 contrasts these two approaches. On the left side

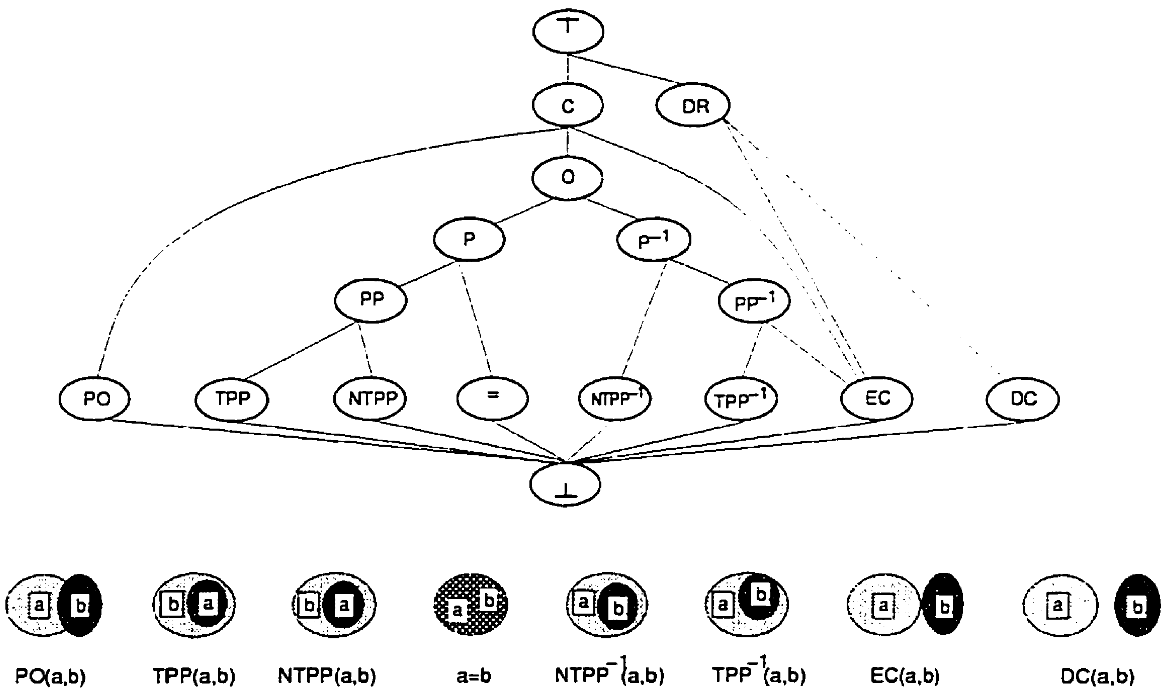


Figure 2.8: The RCC relations

of the figure the external orientation is illustrated. Object A induces four sectors around object B, each one denoting a directional relation. The rightmost part of the figure, we see a qualitative partitioning of the "directions" space by fixing "front" to be the top of the page.

Directional relations are derived through qualitative partitioning of two dimensional space. The degree of partitioning determines the representational granularity of a set of directional relations. Several sets of directional relations have been proposed. Figure 2.10 illustrates Hernadez's rod (relative orientation node) model in three granularities [Her92]; this is also termed the *anthropocentric* system. Figure 2.11 illustrates Frank's [Fra91] *orthogonal* system extended with levels of granularity: at granularity level 1, a distinguished front focus defines a left (or east) and a right (or west) sectors of space (similarly, a north/south partitioning is defined if the axis is rotated by 90 degrees). By superimposing the two, we obtain directional relations of level 2, etc.

Inferences of directional relations are based on a composition operation which is further reduced to order transitivity in a lower dimension structure [Roe94]. For example, in the case of two-dimensional space, the coordinate cartesian axes are the lower dimension

Figure 2.9: External versus internal orientation

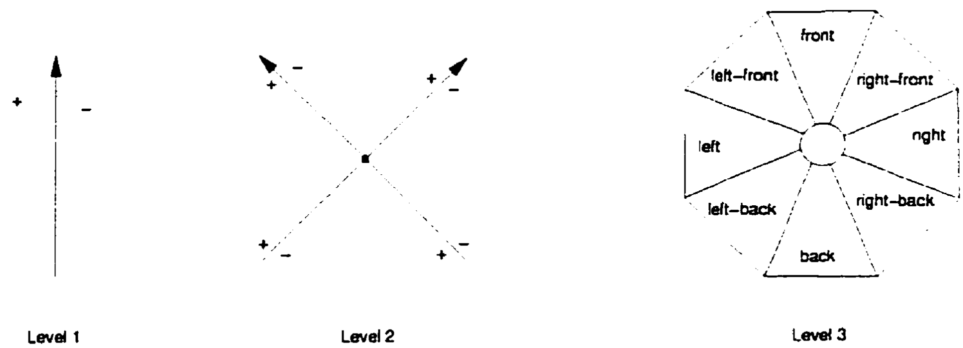


Figure 2.10: Hernandez's directional relations

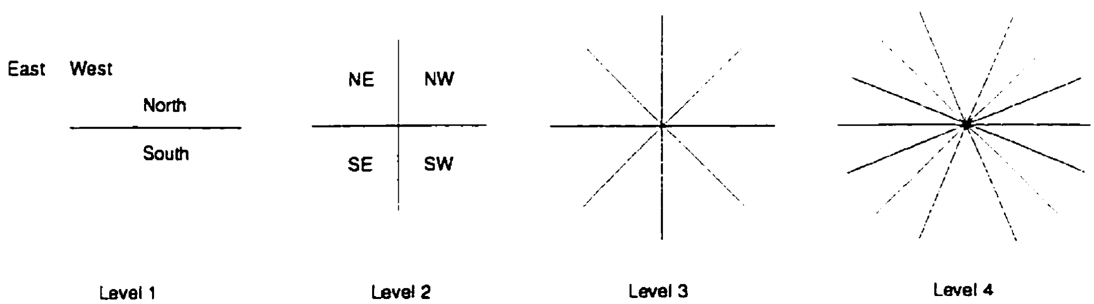


Figure 2.11: Frank's directional relations

structures. Freksa optimizes the size of composition tables used in reasoning, by exploring the structure in the directional relations set and introducing the notion of conceptual neighborhood [Fre92a]. The latter is a key concept that we will revisit in the course of this study.

2.4.7 Axiomatic Approaches

Formal logic has also been used as a framework for modeling spatial reasoning placing emphasis on the axiomatization of space. Some of the most notable axiomatic approaches include Kautz's [Kau85] and Shoham's [Sho85] work within the "commonsense summer" project [Hob85]. Roman's work in formalizing spatial inference in geographic information systems [Rom90], and Kaufman's work on formalizing imprecision in commonsense space [Kau91].

Kaufman proposed an ontology for time and space based on tolerance spaces. Tolerance spaces treat uncertainty as a fundamental principle. Previous models for approximate spatial reasoning, such as [MD84, Dav86], represented imprecise spatial information using interval values on measurements. In tolerance spaces the concept of uncertainty is deeply embedded in their definition. Every tolerance space is characterized by a tolerance relation. The selection of a tolerance relation is the same as selecting the granularity [Hob85] in which a spatial domain is viewed. A tolerance preserving function (i.e., a function which preserves closeness between different spaces) is used for mapping from one granularity level to another. Informally, a tolerance space is a collection of points with a symmetric and reflexive tolerance relation defined over them. At sufficiently small scales, positions or quantities are indistinguishable. Tolerance spaces have been proposed as an alternative reasoning system to deal with approximation and to complete the qualitative approach in reasoning about physical systems.

It should be noted that an axiomatic appoach similar to that of Kaufman, is taken in the development of the proposed theory of space in Chapter 3.

2.4.8 Summary

Table 2.2 summarizes approaches to spatial reasoning discussed in this section. For any method, the table shows the mechanisms it offers for specifying relative position, its representational primitives and modeling domain, and the computational method it employs for

Method	Relative Position	Representation	Modeling Domain	Computational Model
K-dim Allen's [MB83],[MJ90]	A_x left-of B_x	linear, metric or qualitative constraints	1D, 2D, 3D and orthogonal	constraint satisfaction
Topological Relations [Egen89]	A contains B	qualitative constraints	2D space	transitivity tables
Topological Relations [RCC91]	C(A,B)	qualitative constraints	2D space	transitivity tables, qaulitative simulation
Directional Relations [Hern92],[Frek92]	A left-of B	qualitative constraints	2D orthogonal	transitivity tables
Linguistic Variables [Dutt89]	P(A,B)=([5,0,0],[45,5,5])	semi-qualitative	kD space	Warshall's alg
SPAM [MD84]	x(A,B) in [2,4] y(A,B) in [5,9] scale(A,B) in [1.2,1.8] rot(A,B) in [30,35]	quantitative	2D space, multiple coord. frames	numerical optimization
MERCATOR [Davis86]	dist(A,B)=[6,7] rot(A,B)=[20,30]	quantitative	2D space	constraint reasoning + Monte Carlo searching
Tolerance Spaces [Kauf91]	N/A	tolerance relations	kD tolerance space	mathematical induction

Table 2.2: A classification of spatial reasoning methods

reasoning.

2.5 Spatial Databases

This section reviews research in the field of spatial databases. Spatial databases store k-dimensional data representing explicit knowledge about objects, their extent and their position in space [GB90]. Spatial databases are widely used in applications such as geographic information systems (GIS) [MP94], environmental protection [MGD91], CAD/CAM, robotics, medical imaging, etc. Work in this area is broadly divided into spatial data modeling, data structures and access methods, and query processing and optimization. In our discussion, we focus on data modeling and query processing aspects, and we simply mention references to the data structures and access methods work.

The requirements for data management techniques to deal with objects in space differ widely between applications. When the space of interest is a two-dimensional scene, i.e., geographic space, or a human-made space (e.g., VLSI layouts, CAD/CAM designs, drafted maps, etc), the requirement is to handle large collections of relatively simple geometric objects. When the space of interest involves digitized images sensed by satellites, medical scanners, etc., then a different functionality is intended by the database system. In particular, the system must be able to extract objects from images and treat images as discrete entities. Guting [Gut94] assessed these differences and proposed the terms ***spatial database system*** for the former and ***image database system*** for the latter. The definition that Guting gives for spatial database system is currently the most accepted by the research community [DK+94, Par95]. According to this definition, a spatial database system is a database system that offers spatial data types in its data model and query language. Its implementation supports spatial indexing and an efficient spatial join operation.

2.5.1 Spatial Data Models

The fundamental question in spatial database modeling is how to embed spatial aspects in a data model and the underlying database system such that acceptable interfaces (optimizable query languages and pictorial interfaces) can be defined. A spatial data model must support modeling of objects in space (e.g. points, lines, polygons, solids) and space itself, as well as modeling of the non-spatial aspects of objects in space. Finally, it must support basic spatial operators. Some important issues related to spatial data types and operations support as they are realized by Guting [Gut94] and Paradaens [Par95], include:

- *Extensibility.* Spatial operations are usually application dependent. Hence, an effective spatial data model should allow the definition of user defined operations.
- *Completeness.* The data model has to be "closed" for all its operations, i.e., the operations return representable types as answers. In addition, the operations must capture all the intended functionality of space. For instance, the topological operations in Section 2.4.6 can capture all the topologically distinct relative positions between two objects in space.
- *Genericity.* The operations supported by the spatial data model must be independent of the content of data [PVdBVG94].
- *Set Operations.* A spatial data model should support operations defined on individual objects as well as on sets of objects [GS95b].

All of the proposed spatial data models are associated with some dominant data modeling approach, namely, the relational model approach, the extended relational approach, the

MAP

OID	TYPE
B1	block
B2	block
SF	building
LP	building
S1	street

OBJECT

OID	EID
B1	e1
B1	e2
B1	e3
B1	e4
...	...

EDGES

EID	VID
e1	v1
e1	v2
e2	v2
e2	v3
...	...

VERTICES

VID	X	Y
v1	0	0
v2	0	20
v3	40	20
v4	40	0
...	...	...

Figure 2.12: A relational implementation of the BR scheme

object-oriented approach, and lately, with the knowledge representation approach to data modeling.

The relational model has been used in engineering databases [GS82] and in geographic information systems [CF81, CK81, AS86]. This approach has two major drawbacks. First, it cannot support the hierarchical construction of spatial objects. Second, geometric operations are very hard to compute and thus are not supported by the data model. For instance, the retrieval of the bounding edges of all blocks in Figure 2.1 requires an expensive three-way join operation between BLOCKS and EDGES and VERTICES relation.

Certain drawbacks of the relational approach are handled by extensions to the relational model. Useful relational extensions include the user defined types and functions (ADT-INGRES, "QUEL as a Datatype" [SRG83]), the use of surrogates and long fields in system R [LP83], the nested relations data model (NF^2) [PSS^+87], the entity relationship approach adopted by GEM [Zan83], the DAPLEX functional data model [CDF^+82] and many others.

Extended relational models are able to represent the structure and the manipulations over geometric entities. Those promoting the notion of abstract data types (ADTs), such as "QUEL as a Datatype" and GEM are able to model structurally complex entities. DAPLEX supports structure through the generalization and aggregation abstraction principles. The extensions of System R and the NF^2 data model provide constructs for modeling complex structures. A common drawback of these approaches is that they lack generality. Some of them are able to model CSG easier than BR or the other way around. Some operations are very hard to be computed and impossible to be extended. These drawbacks have led to a third approach to spatial data modeling. Instead of defining extensions it is preferable to build on extensibility. This approach is taken in the PROBE [OM88] object-oriented data model and is also featured by "knowledge models" such as Telos [MBJK90].

The *extensibility* of object-oriented databases allows the inclusion of specific object classes that support the required data types (spatial data) and the required manipulations. More specialized object classes are added as needed and customized optimizers are built along with them.

In the PROBE data model (PDM) [MO86], spatial characteristics of entities are represented and integrated with non-spatial characteristics described in the DAPLEX ordinary data model, by means of an entity type called PTSET (pointset). Entities of type PTSET are geometric entities such as lines, areas and volumes. Geometric entities serve as values of spatial attributes such as **boundary** or **shape** of ordinary database entities. The PTSET type can be specialized to obtain entity subtypes with more specific spatial features (e.g., specific shapes). Complex shapes can be built by combining primitive ones. Specialized types can also have specialized functions that are applied to them. A PTSET can contain other PTSETs and the container is called *space*. The operations that are associated with generic PTSET entities are either point set (e.g., spatial) or structural operations. Point set operations include point set intersection, union and difference, spatial selection, overlay and geometric transformations. The structural operations are concerned with the hierarchical structure of spaces and are defined in terms of non-spatial operators of the PDM algebra. The following example illustrates the structural facilities of PDM.

```
type MAP is ENTITY                     type FEATURE is ENTITY
   title(MAP) --> STRING                  entity-type(FEATURE) --> FEATURE-TYPE
   scale(MAP) --> INTEGER                 feature-id(FEATURE) --> INTEGER
   area(MAP) --> PTSET                    shape(FEATURE) --> PTSET
   feature(MAP) --> set of FEATURE

type ROAD is FEATURE                   create new R in ROAD (
   name(ROAD) --> STRING                  name ==> "College Str",
   crosses(ROAD) --> set of ROAD          crosses ==> {"Univ. Ave","St George"},
   length(ROAD) --> REAL                  length ==> 296.89 )
```

Point sets are implemented by the *Approximate Geometry* (AG) component of the PROBE database system. The AG component consists of a query processor and a storage facility for spatial objects. The storage structure of AG is based on a grid representation of spatial objects. Each PTSET entity is decomposed into box-shaped elements. The representation of PTSET entities by a union of box-shaped elements is approximate because it depends on the granularity of the grid decomposition. Each grid element is assigned a *z-ordering* number [Ore86]. The use of the z-ordering numbers reduce spatial operations such as precedence and containment operations, to simple bit-string operations. Hence, approximate spatial operations are computed by very simple algorithms [OM88].

An object-oriented approach in spatial data modeling has been taken by Mohan and Kashyap [MK88], van Oosterom and van den Bos [vOvdB89], Maier et al. [AOG+88], Scholl and Voisard [SV91] and many others.

The suitability of knowledge representation languages in spatial data modeling is acknowledged by Milios et al. in [MMT89] and Lu [Lu93]. Milios et al. introduce GeoTelos, an extension of the language Telos [MBJK90], to deal with spatial domains. GeoTelos allows one to organize objects with spatial features (*GeoConcepts*) using aggregation, generalization and classification abstraction principles, also to express integrity constraints and deductive rules. Deductive rules are seen as a means for specifying user-defined spatial operators. The early design of GeoTelos proposed a set of spatial operators which are a generalization of Allen's temporal relations for two dimensions.

Over the last five years, extended relational spatial models had an impressive comeback with systems such as Paradise [DK+94], GEO [VvO92] and Montage [Ube94]. These systems emphasize implementation technologies rather than data modeling issues.

2.5.2 Spatial Access Methods

At the physical level, spatial data are represented in terms of spatial data structures and access methods. Almost all known spatial data structures are based on the principle of partitioning the space into cells. A two way mapping relates regions of the space to the cells that they fill up. Cells or sets of cells are assigned to disk blocks. Access to spatial objects, given a region of space, consists of finding the cells that intersect the region (*cell addressing*) and then finding the objects that occupy those cells (*data access*) [Nie89]. The general concern during accessing of spatial objects is *minimal number of disk accesses*; therefore good access methods are necessary.

Spatial data objects in d-dimensional space, $d \geq 1$, are approximated by their minimum bounding rectangle (MBR). Methods for storage and accessing of (high dimensional) rectangles fall into three categories [SRF87]:

- Methods that transform the rectangles into points in a space of higher dimensionality; thus referred as *point access methods* [NHS84, SK88, Fre87, OSDM89].

- Methods that use space filling curves to map a k-dimensional space into a 1-dimensional space [OM84, Fal88, Jag90].

- Methods that decompose the space into sub-regions and distribute appropriately the data objects [Gut84, SRF87, BKSS90].

Decomposition-based access methods are further classified according to the techniques used for decomposing the data space. Seeger and Kriegel [SK88] and Sellis et al. [SRF87] presented two such classifications.

The most common spatial queries supported by spatial access methods include: (a) point search, (b) range search, (c) partial match and partial range queries, (d) nearest neighbour queries, (e) spatial join and (f) zoom-in queries. Point search retrieves the data object that meets a search criterion with respect to a certain point. Range search retrieves the data objects that fall into a specified region. Partial match refers to the case where one or more components of the search key are unspecified. The spatial join between two regions R and S returns all pairs of data objects r and s such that they are in R and S respectively and they overlap to one another.

2.5.3 Spatial Query Languages

The goal of a spatial database language is to allow the easy formulation of queries that involve both spatial and non-spatial predicates without loss of spatial semantics. A desired requirement is that the queries should be optimizable.

Existing spatial query languages are roughly divided into (a) extensions of the relational languages, (b) object-oriented languages and (c) pictorial languages.

Chang and Fu [CF81] developed the QPE (query by pictorial example) language as an extension of the QBE (query by example) and the predicate calculus. PSQL [RFS88], GEOQL [OSDM89] and *Spatial* SQL [Ege94] are more recent proposals based on SQL. [LM88b] used Peano tuple algebra, an extension of relational algebra, and computational geometry for processing an important subclass of spatial queries. Guting [Gut88] extended relational algebra in a many-sorted relational algebra with geometric types and geometric operators.

Examples of the object-oriented approach in spatial query languages are object logic [AOG$^+$88], the deductive, object-oriented language by Mohan and Kashyap [MK88], GeoTelos [MMT89], and O_2SQL [RS95].

Pictorial query languages emphasize direct manipulations of pictorial information integrated with ordinary querying facilities. Some known pictorial languages are the PSQL and the PICQUERY [JC88].

Optimization problems in spatial databases are significantly different than business oriented-databases. PROBE [OM88, Ore89] and GEOQL [OSDM89, OSD89] are known efforts which have progressed in this direction. In particular, PROBE offers a way of extending the query operations and the optimizer (see section 2.5.1). In GEOQL, Ooi et al. defined an optimizer which separates the spatial and the non-spatial predicates of the query, which are handled by different processors, and a high level optimization procedure selects the query execution plan with the lowest cost. Optimization of spatial subqueries is based on the use of a spatial index.

2.5.4 Summary

This section presented an overview of the research in spatial databases. Work in this area is broadly divided into spatial data modeling, data structures and access methods, and query processing and optimization. Object orientation and knowledge representation offer suitable modeling approaches for spatial databases because they can support the structural aspects and a wide range of spatial operations in spatial representation schemes. The spatial data structures and the access methods are well studied fields. There is virtually no work in query processing and optimization at the query language level although there is a huge amount of work at the data access level. The reason for this is that the corresponding problems have not yet received satisfactory answers even for object oriented databases or knowledge bases. Perhaps most importantly, the treatment of uncertainty and partial information has not received enough attention in spatial databases.

2.6 Conclusion

We have reviewed research in the fields of spatial representations, spatial reasoning and spatial databases. We have also motivated the importance of spatial knowledge management by presenting several applications which demand support for spatial data.

One first observation is that efforts in spatial representation and reasoning are driven by the requirements of specific applications. For instance, in robotics the emphasis is placed on the representation of the free space and the path finding problem. In qualitative mechanics, the representation and reasoning is mainly based on tangency relations. Geographic applications require relative position computations and representation of the extremely arbitrary shapes. As a result, there is no system that provides adequate functionality for a wide range of spatial applications.

The second observation is that many applications such as CAD/CAM and geographic in formation systems, have to cope with large amounts of spatial data. For this purpose, the functionality of a database system is necessary. At the same time, the support of spatial knowledge representation and reasoning techniques is generally acknowledged as desirable and important.

Chapter 3

The Haze-Order Space and its Axiomatization

3.1 Introduction

In this chapter we present a language for expressing spatial relations on points and regions in imprecise space. We start by introducing an appropriate ontology of space and then present the first-order theory of one-dimensional imprecise space. The theory of imprecise space is axiomatized in terms of haze points and the *haze* and *precedence* relations. We analyze the theory of one-dimensional imprecise space from the point of view of the model theory and we note useful facts about its models. These facts foreshadow some of the underlying principles of the spatial reasoning algorithms that are developed in Chapter 4. This analysis contributes to the theoretical work in spatial representations since it explores the limits of the *haze*-based approach. We also propose several extensions of the theory for higher dimensional space. In the two-dimensional case, we define a complete set of topological and directional relations that are useful for practical reasoning about space. The proposed formalism is strictly qualitative with a built-in concept for imprecision.

The rest of this chapter is organized as follows: Section 3.2 discusses motivation and proposes a new ontology of imprecise space. Section 3.3 reviews some definitions from first-order logic and model theory. Sections 3.4, 3.5 and 3.6 present the development of the theories of one and two-dimensional point and region space, respectively, along with an assessment of their models. Section 3.7 presents an application of the developed theories to the definition of binary spatial relations, and Section 3.8 discusses the problem of varying the granularity. Finally, Section 3.9 summarizes the technical results of the chapter. [1]

3.2 The Ontology of Haze-Order Space

An ontology of space is a conceptualization of space that includes a formal pattern for space, its objects and the defined spatial relations. This work is interested in an ontology of space with a built-in notion to account for imprecision.

The selected ontology of space must be characterized by simplicity and mathematical clarity. The most successful computational paradigms base their success on their mathematical clarity. Relational databases are a well known example. The study of formal aspects of spatial representation and reasoning, however, is a broad area without commonly accepted formal methods and theoretical tools. A simple and mathematical ontology should be the basis for proposing a comprehensive theory for spatial representation and reasoning. Such a theory is expected to provide answers to questions such as: Does a set of spatial relations allow us to specify all the qualitative distinct situations in space? Are inferences, in some given representation, sound and/or complete? Are certain theories of space complete and/or decidable?

We propose the ontology of *haze* space in which space is viewed as a totality of spatial objects connected by certain relations. The objects of haze space are either haze points or haze non-point objects. Haze points refer to points of space which are surrounded by a haze area, the smallest distinguishable quantity in the representation. Haze non-point objects are constructed by connecting such areas. A consequence of the haze area around points is that, at sufficiently small scales, objects are indistinguishable. Figure 3.1 illustrates haze points in one and two dimensions.

Other researchers used the terms "tolerance" or "noticeable difference" [Rob73], [Kau91] to describe a similar notion. The former is the closest concept to the proposed ontology however the formalization of two proposals differs significantly. Tolerance spaces are described in terms of inductive axioms. Worth mentioning is also the work on threshold and interval rerpesentations [KSLT89, Fis85] in the context of measurement theory as well as the work on logics for approximate reasoning [KH92]. All of the above study the mathe-

[1] A preliminary version of material presented in Chapter 3 has appeared in [Top94a].

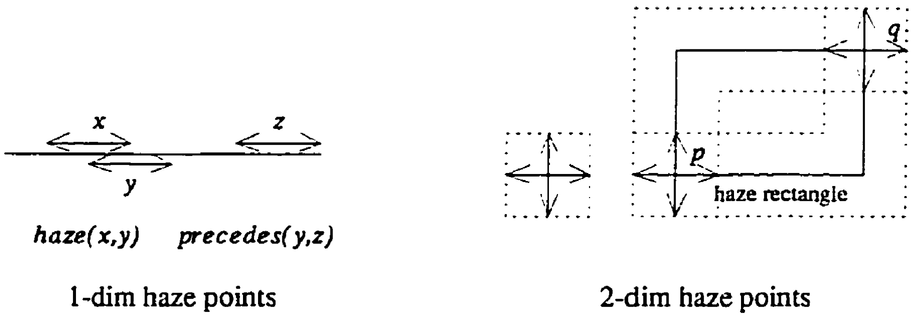


Figure 3.1: The *haze*-point ontology

matics of imprecision in quantitative represesentations. The fundamental difference of our work is that it is founded on a strictly qualitative basis. The haze-point ontology provides for a formal treatment of granularity along the lines of [Hob85] because the definitions of spatial relations are parameterized by a degree of detail. Finally, fuzzy set theory and fuzzy logic [Zad94, Fre94] have been used as formal models to exploit tolerance for imprecision and uncertainty leading to probabilistic models of reasoning. These approaches fall beyond the scope of our study.

We assume two primitive relations, *haze* and *precedence*. The haze relation models the indistinguishability of two objects being too close to each other. For instance, in a macroscopic view, one cannot determine whether two close points on a line, precede one another or are in the same place. If the two points start moving to opposite directions then a precedence relation between the two starts becoming clear. Still the threshold after which the two points can be sorted is not clear cut: it might depend on the distance from which the observation is made, or the discriminating power of the particular observers. Finding the threshold of indistinguishability is an interesting problem alone and it very much depends on particular application domains. For the purpose of a theoretical investigation of imprecise space, it is sufficient to assume that such a threshold is available.

The notion of haze can help us to model situations where input information is not precise such as linguistic descriptions, sonar sensors output, scientific experiment results, etc.; or situations where we are simply interested in limited degrees of precision. The size of the haze area accounts for the degree of precision.

3.3 Background

In this section, we present the definitions of first-order languages, structures and theories that are needed for the discussion in the rest of this chapter. The theoretical background of this chapter is based on material described in [End72], [Bri77], [CK77] and [Lad87].

A *many-sorted first-order* language $\mathcal{L}$ is a set of countably infinite symbols including non-logical symbols such as a set of *sort symbols*, the $\forall_s$ and $\exists_s$ *quantifiers* for each sort s, a set of *predicate, function* and *constant symbols*, and logical symbols including one countably infinite set of *variables*, for each sort, the standard *sentential connectives* $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$, *parentheses* (,), and the *equality symbol* $=_s$, for each sort s. *Atomic formulas*, *well-formed formulas* and *sentences* are defined in the usual way [End72].

A *first-order theory* T is a set of sentences expressed in a first-order language $\mathcal{L}$, that is closed under logical implication. That is, T is a theory iff T is a set of sentences such that for any sentence σ of $\mathcal{L}$, $T \models \sigma \Longrightarrow \sigma \in T$. A theory is usually specified by providing its language, a set of sentences called *axioms*, and a set of inference rules which, in the case of first-order thoeries, are the stardard first-order inference rules of *modus ponens* and *generalization*.

A *many-sorted structure* M is a set of objects along with relations and functions on those objects and distinguished constants that the symbols of a first-order language take their meaning on. M is an $\mathcal{L}$-structure if: for each sort s in $\mathcal{L}$ there is a domain set in M, $dom_M(s)$, where universally quantified variables of sort s take values from; for each predicate symbol p in $\mathcal{L}$ there is a relation p^M of appropriate arity and sort in M; for each function symbol f, there is a function f^M of the appropriate sort restrictions in M; and, for each constant c of sort s, there is an element c^M of $dom_M(s)$.

The truth of a sentence σ in $\mathcal{L}$ is evaluated with respect to an $\mathcal{L}$-structure M: $M \models \sigma$ reads "σ is true in M". The function v from the variables of $\mathcal{L}$ to their domains in M, is called *valuation*. We use the notation $M \models \phi[v]$ to say that M *satisfies* ϕ under valuation v. The definitions of truth and satisfaction are the standard ones found in [End72]. A *model of a theory* T expressed in language $\mathcal{L}$ is an $\mathcal{L}$-structure M, such that all sentences in T are true in M. A theory T is *consistent* if there is at least one model for T.

For a set of sentences Σ, $Mod(\Sigma)$ is defined to be the class of all models of Σ. The theory of a class of models M of Σ, denoted $Th(M)$, is the set of all sentences of the language of

M that are true in every member of M. $Th(Mod(\Sigma)$ is then the set of all sentences in all models of Σ. But this is the set of all sentences logically implied by Σ, called the set of *consequences* of Σ, $Cn(\Sigma)$. Thus $Cn(\Sigma) = \{\sigma : \Sigma \models \sigma\} = Th(Mod(\Sigma)$. A set of sentences T is a theory iff $T = Cn(T)$.

A theory T *complete* if it is consistent and for every sentence P in the language of T, either P or not $\neg P$, but not both, are in the theory. $Th(M)$ is complete by definition.

Two models M and M' are *isomorphic* if and only if there is a one-to-one onto function from M to M' such that any true statement about elements of M, in the language of M and M', is true also about their images in M', and vice-versa. The two models are *homomorphic* if the mapping function between M and M' preserves the structure but is not required to be one-to-one.

A theory T is *finitely axiomatizable* if there is a set of sentences Σ such that T is the set of all deductive consequences of Σ, that is $Cn(\Sigma) = Th(Mod(T))$. A finitely axiomatizable theory is axiomatizable, but not necessarily the reverse. A complete and axiomatizable theory is *decidable.*

3.4 One-dimensional Haze Space

We start the formal development by first considering haze space in a single dimension, denoted $H1$. The axioms of $H1$ are expressed in a many-sorted first-order language $\mathcal{L}$. $\mathcal{L}$ has two sorts of individuals: P for *points* and S for *scales*. The non-logical symbols of $\mathcal{L}$ include the predicate symbols $\prec$ and h both of sorts $P \times P \times S$, the predicate symbol $<_s$ relating individuals of sort $S \times S$, and equality for both sorts. Until Section 3.8, we will assume that sort S contains a single constant symbol g which denotes the size of the scope of the h(aze) relation. In the general case, relation h contains an explicit argument which quantifies its indistinguishability, i.e., its value can grow or shrink denoting that the imprecision of the representation increases or decreases, respectively. Although, relations $\prec$ and h contain a third argument of sort S, for readability purposes, we write them as binary relations, $x \prec y$ and $h(x,y)$. It should be noted that this is an equivalent notation that does not impose any technical limitations since g is a constant. In the following the letters x, y, z, u are taken as variables ranging over P.

The first two axioms of $H1$ state that relation h is reflexive and symmetric.

A1. $\forall x(h(x,x))$

A2. $\forall xy(h(x,y) \Rightarrow h(y,x))$

The next six axioms, with the exception of A6, express properties of the precedence relation. Axioms A3, A4 and A5 state that the precedence relation is irreflexive, asymmetric and transitive. A6 states that relation h is in the symmetric complement of $\prec$. Axiom A7 states that space extends in both directions. Finally, axiom A8 ensures that there is a "step-wise succession" with respect to the precedence relation which adds a discreteness property to the haze point space.

A3. $\forall x(\neg x \prec x)$

A4. $\forall xy(x \prec y \Rightarrow \neg y \prec x)$

A5. $\forall xyz(x \prec y \land y \prec z \Rightarrow x \prec z)$

A6. $\forall xy(h(x,y) \Rightarrow \neg x \prec y \land \neg y \prec x)$

A7.1 $\forall x \exists y(y \prec x)$

A7.2 $\forall x \exists y(x \prec y)$

A8.1 $\forall x \exists y(x \prec y \land \neg \exists z(x \prec z \land z \prec y))$

A8.2 $\forall x \exists y(y \prec x \land \neg \exists z(y \prec z \land z \prec x)))$

In addition, the equality axioms for each sort and linear order axioms for $<_s$ in S, are assumed.

The symbols of $\mathcal{L}$ are interpreted with respect to a fixed structure **P1** which captures our assumptions, i.e., space is discrete, unbounded, with precedence and haze relations defined as above. The domain that underlies **P1** is $\mathbb{Z}$. The constant symbol g is assigned a non-negative integer constant g under **P1**. The domain of P in **P1** is the set of all intervals of size 2g based on integers and it is denoted by $I(\mathbb{Z})$. To each of the predicate symbols $\prec$, h and $<_s$, **P1** assigns the relations p, r and $<$ (the "less than" relation in $\mathbb{Z}$)), respectively. Relations p and r are defined as follows:

$$p = \{(i,j,\mathrm{g}) | i,j \in I(\mathbb{Z}) \text{ and for every } x \in i \text{ it is the case that } x + \mathrm{g} \in j \lor x - \mathrm{g} \in j\}$$

$$r = \{(i,j,\mathrm{g}) | i,j \in I(\mathbb{Z}) \text{ and for every integer } z \text{ such that } z \in i \text{ it is either } (z \in j \land z + \mathrm{g} \in j \land z - \mathrm{g} \notin j) \text{ or } (z \notin j \land z < t \text{ for every } t \in j)\}$$

Note that an interpreted haze point over the integers line is an interval of length 2g. Two points that are in haze relation have their interpretation intervals overlapping by at least

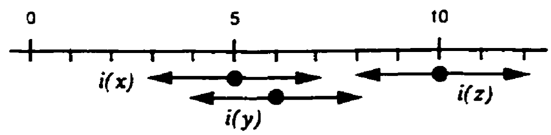


Figure 3.2: A model of T (Example 3.4.1)

half of their length. It is easy to show that for a given g, **P1** defines a class of models for $H1$. It suffices to show that all axioms are true in **P1**. We can easily verify this for each one of them. For instance, the meaning of A6 is that for every two intervals i, j, of size 2g, if there is an overlapping between them such that their overlapping part is equal or longer than g, i.e., for every $x \in i$ it is the case that $x + g \in j \vee x - g \in j$, then it cannot be the case that $i \prec j$ because we can find one $z \in i$ and $z \in j$ such that $z + g \in j \wedge z - g \in j$ contrary to what r states. Similarly, it cannot be the case that $j \prec i$. In the same way we can verify the truth of all other axioms under **P1**.

Example 3.4.1 Let $\mathbf{P1_2} = \langle I(\mathbb{Z}), p, r, 2\rangle$ [2] be an $\mathcal{L}$-structure and $T = \{x \prec z, h(x, y, g)\}$ be a set of $\mathcal{L}$ terms. Also, let i be an assignment of $\mathcal{L}$ variables to the domain of $\mathbf{P1_2}$ such that $i(x) = [3, 7], i(y) = [4, 8], i(z) = [8, 12]$ where $[l, u]$ denotes a closed integer interval. Then, the pair $(\mathbf{P1_2}, i)$ is a model of T (see Figure 3.2).

3.4.1 On the Models of $H1$

While in the previous section our objective was an explicit axiomatic view of the structure of haze space, in this section we take a view that combines representation and inference. This involves, first, the further analysis of the qualitative properties of $\prec$ and h and the definition of a quantitative representation (i.e., build the models of $H1$) which facilitates the problem of inference, i.e., it leads from recorded assertions to inferred ones. In addition, we note useful facts about the models of $H1$. These facts foreshadow some of the underlying principles of the algorithms presented in Chapter 4.

We start by looking into the haze relation. $h(x, y)$ is a symmetric but not transitive relation representing the indistinguishability of two points x and y. We can also define another indifference binary relation on P which is based on $\prec$ and is transitive. The new relation is called *neighbors*, written $n(x, y)$, and we say that x *neighbors* y if both x and y

[2] As was mentioned earlier, sort S contains a single constant and thus, for brevity, its interpretation domain $\mathbf{Z}$ and the interpretation function $<$ for $<_s$, are omitted from $\mathbf{P1_2}$. Thus, $\mathbf{P1_2}$ is written as $\langle I(\mathbf{Z}), p, r, 2\rangle$ instead of $\langle I(\mathbf{Z}), \mathbf{Z}, p, r, <_{\mathbf{Z}} 2\rangle$

are preceded and followed by exactly the same haze points. More formally, x *neighbors* y if for all z such that $x \prec z$ it is the case that $y \prec z$, and for all u such that $u \prec x$ it is the case that $u \prec y$, as well as, for all z such that $y \prec z$ it is the case that $x \prec z$, and for all u such that $u \prec y$ it is the case that $u \prec x$. That is, x, y are inferred not to satisfy $n(x, y)$ if some z can be found that stands in relation h to one of them but not the other. Once again, h is an abstraction of indistinguishability while n is a relation that denotes the indifference of two points which perhaps are not too close to each other but they are perceived indifferent when related to third points. n, as shown below, is proven very useful in drawing inferences in haze space.

Some ordering information embodied in h relation can also be recovered by examining how two hazy points are related to third points. We refer to the recovered ordering information using the relation symbol $\overline{\lhd}$ and we write $x\overline{\lhd}y$ to denote that x precedes everything that y does. The details of $\overline{\lhd}$ definition have as follows: First, if $x \prec y$, then by the transitivity of $\prec$, $y \prec z$ implies $x \prec z$, and therefore $x\overline{\lhd}y$. Next, if $h(x, y)$ suppose we find a z such that $x \prec z$ and $h(y, z)$. This is interpreted as evidence that y lies after x; and we assert that if opposite evidence is never found, then we can treat x as before or equal to y, which means $x\overline{\lhd}y$. One important point that we need to emphasize is the assymetric nature of the $\prec$ relation. This involves looking at both its decreasing and increasing direction in order to specify the delicate distinctions having to do with what exactly goes on the threshold that determines whether $x \prec y$ or $h(x, y)$. Therefore, $\unlhd$ denotes the ordering relation which is induced by examining how x, y are related to hazy points that lie before them: $x\unlhd y$ if there is no z that provides contrary evidence, lying after x but before y (i.e., whenever $z \prec x$, then $z \prec y$). By putting together $\overline{\lhd}$ and $\unlhd$ we get the two-sided ordering relation $\lhd$ which we will call an interval order. By symmetrizing $\overline{\lhd}$ we obtain an indifference relation: $\overline{n}(x, y)$ when both $x\overline{\lhd}y$ and $y\overline{\lhd}x$ hold. Similarly, we can define $\underline{n}$ and n, the *neighbors* relation. We embody these ideas in the following formal definition.

Definition 3.4.1 *Let $\prec$ be an assymetric binary relation on P. The following relations are defined in terms of $\prec$:*

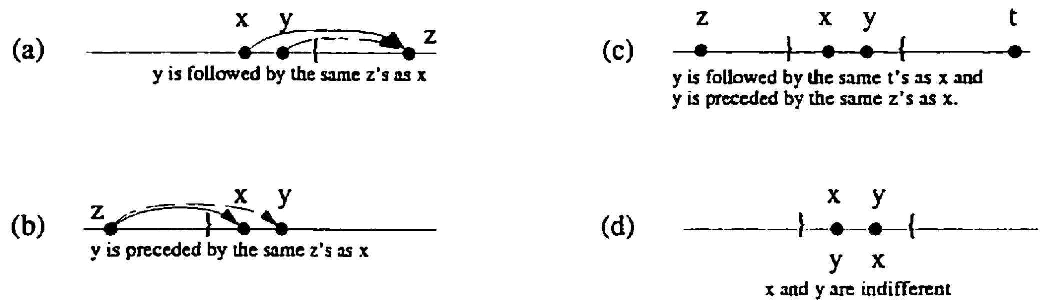


Figure 3.3: (a) x precedes_from_above y, (b) x precedes_from_below y, (c) x precedes y, (d) x neighbors y

x precedes_from_above y	:	$x\overline{\lhd}y \equiv \forall z(x \prec z \;\Rightarrow\; y \prec z)$
x precedes_from_below y	:	$x\underline{\lhd}y \equiv \forall z(z \prec x \;\Rightarrow\; z \prec y)$
x precedes y	:	$x \lhd y \equiv x\overline{\lhd}y \wedge x\underline{\lhd}y$
x neighbors y	:	$n(x,y) \equiv \forall x \lhd y \wedge y \lhd x$

The defined relations are illustrated in Figure 3.3.

We now build the quantitative representation for $H1$. Due to the assymetry of $\prec$ we need to take into account both the decreasing and increasing direction of $\prec$. We are basically investigating numeric functions from P to $\mathbb{Z}$ with the following property: $\overline{\phi}(x) + \delta < \overline{\phi}(y)$ for every $x \prec y$ and also $\underline{\phi}(x) < \underline{\phi}(y) - \delta$ for every $x < y$. Note that it is possible that $\overline{\phi}$ and $\underline{\phi}$ can induce different orders on $\mathbb{Z}$. A quantitative representation can capture the two different induced orders by considering $\overline{\phi} = \underline{\phi} = \phi$ and two different threshold values $\overline{\delta}$ and $\underline{\delta}$. In such case, $h(x,y)$ implies that $\phi(x)$ lies in the interval $[\phi(y) - \underline{\delta}, \phi(y) + \overline{\delta}]$. But the symmetry of h implies that $\overline{\delta} = \underline{\delta} = \delta$. All these concepts are formally summarized in the next definition.

Definition 3.4.2 *Let $\prec$ be an assymetric binary relation on P. An integer valued function ϕ and an integer constant δ form a quantitative representation if for all $x, y, z \in P$, the following hold:*

(a) $x \prec y$ iff $\phi(x) + \delta < \phi(y)$

(b) if $\phi(x) = \phi(y)$ then $n(x,y)$

To summarize the discussion until this point, we assumed a set of hazy points P and two primitive relations $\prec$ and h defined on it. $\prec$ is an assymetric and transitive relation, while

h is symmetric and non-transitive. Then, based on the primitive relations, we created two new relations n and $\lhd$ which both together embody the same information as $\prec$ and h, and finally we built a quantitative representation for them over the integers. Now, we can prove the following propositions:

Proposition 3.4.1 *Let ϕ be any integer valued function and δ any integer constant. If the pair (ϕ, δ) respects the Definition 3.4.2, then (ϕ, δ) defines a model for H1.*

Proof. In Section 3.4, we defined a class of models $H1$ based on the integers by mapping each element x of P to an interval, say $i(x) = [l_x, u_x]$, in $I(Z)$ such that $u_x - l_x = 2g$ (1). Let us take $\delta =$ g, $\phi(x) = u_x - \delta$ and $\phi(y) = l_y + \delta$. Note that $u_x - \delta = l_x + \delta$ because of (1).

$x \prec y$ means that either $i(x)$ precedes $i(y)$ or $i(x)$ overlaps $i(y)$ by less than half length i.e., $u_x - l_y < g$. This is written: $u_x - l_y = \phi(x) + \delta - \phi(y) + \delta < \delta$ or equivalently $\phi(x) + \delta < \phi(y)$ which by definition is the interpretation of $\prec$ under (ϕ, δ).

$h(x, y)$ means that $-g \leq l_x - l_y \leq g$. This is written: $-\delta \leq \phi(x) + \delta - \phi(y) - \delta \leq \delta$, i.e., $\phi(x) \geq (\phi(y) - \delta$ and $\phi(x) \leq \phi(y) + \delta$ meaning that $\phi(x)$ lies in the interval $[\phi(y) - \delta, \phi(y) + \delta]$ which is also an implication of Definition 3.4.2 and axiom A6. □.

Proposition 3.4.2 *Relation n (neighbors) is an equivalence relation.*

Proof. We need to show that n is reflexive, symmetric and transitive. Reflexivity and symmetry of n is trivially verified by Definition 3.4.1. Reflexivity: $n(x, x) \equiv \forall z(x \prec z \Leftrightarrow x \prec z) \land (z \prec x \Leftrightarrow z \prec x)$ is a tautology. Symmetry: $n(x, y) \equiv \forall z(x \prec z \Leftrightarrow y \prec z) \land (z \prec x \Leftrightarrow z \prec y) \equiv n(y, x)$. As for transitivity, suppose that $n(x, y)$ and $n(y, z)$. We must show that $\forall w(x \prec w \Leftrightarrow z \prec w) \land (w \prec x \Leftrightarrow w \prec z)$. But, from the hypothesis, for every w it is $x \prec w \Leftrightarrow y \prec w \Leftrightarrow z \prec w$, and $w \prec x \Leftrightarrow w \prec y \Leftrightarrow w \prec z$; hence $n(x, z)$ and thus n is transitive. □

Given that n is an equivalence relation on P, n partitions P into equivalent classes the set of which is denoted by P/n. As the following proposition affirms, the equivalent classes of n partially ordered by $\prec$ as well.

Proposition 3.4.3 *$\prec$ partially orders the equivalent classes of n.*

Proof. The equivalent classes defined by n on P are disjoined and by the definition of n and axioms A4 and A5 we can verify that $\prec$ is irreflexive and transitive on the equivalence classes and therefore it is a partial order. $\square$

The following example illustrates the practical implications of the until now discussion and also gives hints for an algorithm that computes the models of $H1$[3].

Example 3.4.1.1 Let $P = \{a, b, c, d\}$, $\prec = \{(a, d), (b, d)\}$ and $h(a, b)$. $\overline{\lhd}$ does not discriminate between a and b. If we add a fifth element e such that $e \prec a$ and $e \prec b$, then $\unlhd$ does not distinguish a and b either. Then, it is the case that $n(a.b)$. If we use capital letters to denote the equivalence classes, we get $P* = A, C, D, E$ and $\unlhd = \{(A, D), (E, A)\}$, where A is $\{a, b\}$.

We can also assign ranks 1 to 4 from low to high relatively to $\overline{\lhd}$ and $\unlhd$. These values will help us to come up with a ϕ function when we build a model. $\overline{\lhd}$ ties the ranks for a, b while suggests that c lies above a, b and below d. Thus we assign $a = 1, b = 1, c = 2, d = 3$. $\unlhd$ ties a, b and c since nothing is before them. So, we assign $a = 1, b = 1, c = 1, d = 2$. Note that the ranking is not a model. We also need to come up with an appropriate δ value that satisfies the conditions stated in the proof of Proposition 3.4.1. For $\delta = 1$, $i(a) = [0, 2], i(b) = [0, 2], i(c) = [2, 4]$ and $i(d) = [4, 6]$ is such a model. $\square$

Let us now investigate conditions under which the partial order structure that underlies $H1$ is extended to a linear order. A linear extension of $\prec$ is possible when h becomes the identity relation. Then, A6 axiom of Section 3.4 turns to a weak linearity axiom. An additional axiom stating that $\forall xy(\neg x \prec y \wedge \neg y \prec x \Rightarrow h(x, y))$ is needed to convert the weak-order to a linear order by adding antisymmetry.

To summarize this section, we have shown (Proposition 3.4.1) that there is a mapping between the class of models defined in Section 3.4 and the quantitative representation of structure $\langle P, \lhd, n\rangle$. In addition, we showed that P/n is partially ordered by $\prec$. This establishes state that the models of $H1$ consist of partially ordered neighborhoods. We have also outlined through Example 3.4.1.1 an algorithm to build models of $H1$.

[3]See Section 4.3.3.

3.5 The Theory of Two-dimensional Haze Points

In this section we extend the theory of haze space by adding a second dimension, thus deriving the theory of two-dimensional haze space. There are two distinct criteria defining a theory of two dimensional space. First, the alphabet of its language needs to be extended so that it contains two non-empty sets of operators. Second, the domain within which the symbols of the language are interpreted needs to be two-dimensional. In providing a method to define the two-dimensional theory of haze space, we have to pay attention in the following three points: extending the language, its axiom system and the method for defining its models. Our solution is based on the combination of two one-dimensional structures of space and it is termed *independent combination.*

We extend the language $\mathcal{L}$ with an additional sort of individuals, namely, the sort of two-dimensional points. The set of sorts in $\mathcal{L}$ now becomes $\{P_1, P_2, Q, S\}$, where P_1, P_2 are two disjoint copies of P, and Q is the newly introduced sort. For clarity, hereafter we require quantifiers to range over particular sorts. Let $sorts(\mathcal{L})$ be the set of all sorts in $\mathcal{L}$ and $\mathcal{F}(x)$ be a sentence. For each $s \in sorts(\mathcal{L})$:

$\forall x/s\ \mathcal{F}(x)$ stands for $\forall x(s(x) \Rightarrow \mathcal{F}(x))$, and

$\exists x/s\ \mathcal{F}(x)$ stands for $\exists x(s(x) \wedge \mathcal{F}(x))$

Symbols of each one copy of P are related by relation symbols $\prec$ and h appropriately subscripted with $_1$ or $_2$. A *pairing* function from sorts $P_1 \times P_2$ to Q, relates two-dimensional points with their one-dimensional coordinates.

$\langle x_1, x_2\rangle$ it is a pairing function which returns the two-dimensional point formed by its one-dimensional coordinates

Having augmented our theory vocabulary with new symbols we have to extend the axioms appropriately. The new theory is denoted by $H2$. Axiom B1 states that each two-dimensional point consists of two coordinate one-dimensional points. In addition, axiom B2 postulates that the pairing function "$\langle , \rangle$" is injective (i.e., $P1, P2$ and Q have the same cardinality.

B1. $\forall x/Q\ \exists x_1/P_1\ \exists x_2/P_2\ (x = \langle x_1, x_2\rangle)$

B2. $\forall x_1 y_1/P_1\ \forall x_2 y_2/P_2\ (\langle x_1, x_2\rangle = \langle y_1, y_2\rangle\ \Rightarrow\ x_1 = y_1 \wedge x_2 = y_2)$

The next axiom extends the notion of haze to two-dimensional points, hh:

B3. $\forall xy/Q\ \exists x_1y_1/P_1\ \exists x_2y_2/P_2\ (x = \langle x_1, x_2\rangle \wedge y = \langle y_1, y_2\rangle \wedge (hh(x,y)$
$\Leftrightarrow\ h_1(x_1,y_1) \wedge h_2(x_2,y_2)))$

We now define a notion of weak collinearity along each coordinate axis in two-dimensions. By convention, the two coordinate axes are subscripted by $_h$ (horizontal) and $_v$ (vertical).

B4.1 $\forall xy/Q\ \exists x_1y_1/P_1\ \exists x_2y_2/P_2\ (x = \langle x_1, x_2\rangle \wedge y = \langle y_1, y_2\rangle \wedge (w\_colinear_h(x,y)$
$\Leftrightarrow\ h_2(x_2,y_2)))$

B4.2 $\forall xy/Q\ \exists x_1y_1/P_1\ \exists x_2y_2/P_2\ (x = \langle x_1, x_2\rangle \wedge y = \langle y_1, y_2\rangle \wedge (w\_colinear_v(x,y)$
$\Leftrightarrow\ h_1(x_1,y_1)))$

We also define the $hor(x)$, $vert(x)$ functions from sort Q to P_1 and P_2, respectively, to return a point's projection on the horizontal and vertical axis.

B5. $\forall x_1/P_1\ \forall x_2/P_2\ (hor(\langle x_1, x_2\rangle) = x_1)$

B6. $\forall x_1/P_1\ \forall x_2/P_2\ (vert(\langle x_1, x_2\rangle) = x_2)$

The point decomposition property is expressed by the following sentence (B7); B7 is a valid sentence that follows from B1,B5,B6.

B7 $\forall x/Q\ (x = \langle hor(x), vert(x)\rangle)$

Order (precedence) in the two-dimensional space is defined in terms of the $\prec_1, \prec_2$ relations of the one-dimensional case.

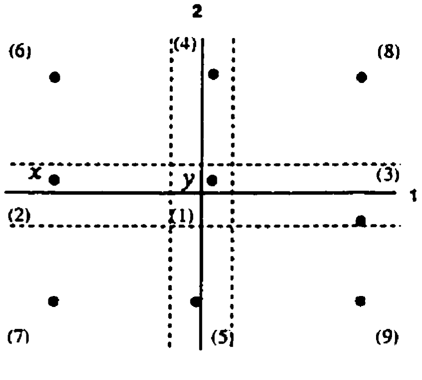
C1. $\forall xy/Q\ \exists x_1y_1/P_1\ \exists x_2y_2/P_2\ (x = \langle x_1, x_2\rangle \wedge y = \langle y_1, y_2\rangle \wedge (east(x,y)\ \Leftrightarrow\ x_1 \prec_1 y_1))$

C2. $\forall xy/Q\ \exists x_1y_1/P_1\ \exists x_2y_2/P_2\ (x = \langle x_1, x_2\rangle \wedge y = \langle y_1, y_2\rangle \wedge (west(x,y)\ \Leftrightarrow\ y_1 \prec_1 x_1))$

C3. $\forall xy/Q\ \exists x_1y_1/P_1\ \exists x_2y_2/P_2\ (x = \langle x_1, x_2\rangle \wedge y = \langle y_1, y_2\rangle \wedge (north(x,y)\ \Leftrightarrow\ y_2 \prec_2 x_2))$

C4. $\forall xy/Q\ \exists x_1y_1/P_1\ \exists x_2y_2/P_2\ (x = \langle x_1, x_2\rangle \wedge y = \langle y_1, y_2\rangle \wedge (south(x,y)\ \Leftrightarrow\ x_2 \prec_2 y_2))$

East, west, north and *south* are irreflexive, asymmetric and transitive, and, in addition, *east* is the inverse of *west* and *south* is the inverse of *north*. The following axiom states the totality property in the two-dimensional space, i.e., any two points are related with one of the nine disjunctive relationships. Figure 3.4(a) illustrates the nine distinct placements of two points in two dimensions.



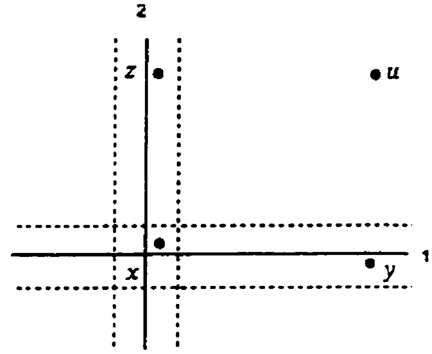


Figure 3.4: A graphical illustration of axioms C5 and C6

$$
\begin{aligned}
\text{C5.}\quad & \forall xy/Q\ (hh(x,y) \vee & (1)\\
& (w\_colinear_h(x,y)\ \wedge\ east(x,y)) \vee & (2)\\
& (w\_colinear_h(x,y)\ \wedge\ west(x,y)) \vee & (3)\\
& (w\_colinear_v(x,y)\ \wedge\ north(x,y)) \vee & (4)\\
& (w\_colinear_v(x,y)\ \wedge\ south(x,y)) \vee & (5)\\
& (north(x,y)\ \wedge\ east(x,y)) \vee & (6)\\
& (south(x,y)\ \wedge\ east(x,y)) \vee & (7)\\
& (north(x,y)\ \wedge\ west(x,y)) \vee & (8)\\
& (south(x,y)\ \wedge\ west(x,y))) & (9)
\end{aligned}
$$

Axiom C6 is to ensure the orthogonality property in $H2$ (see Figure 3.4(b)).

C6.1 $\forall xyz/Q\ (w\_colinear_h(x,y) \wedge w\_colinear_v(x,z)\ \Rightarrow\ \exists u/Q(w\_colinear_h(z,u) \wedge w\_colinear_v(y,u)))$

C6.2 $\forall xyz/Q\ (w\_colinear_v(x,y)\ \wedge\ w\_colinear_h(x,z)\ \Rightarrow\ \exists u/Q(w\_colinear_v(z,u) \wedge w\_colinear_h(y,u)))$

It can be easily seen that theory $H2$ has a class of models, i.e., is consistent, which are based on the cartesian plane $\mathbb{Z} \times \mathbb{Z}$. Next section precisely characterizes these models.

3.5.1 Models of $H2$

We developed a two-dimensional theory for space, $H2$, based on a combination of two theories that correspond to (independent) one-dimensional coordinates. This combination views each dimension independently. We require that the two-dimensional individuals are pairs of one-dimensional individuals for which a separate structure is assumed. Cross-domain operators defined between two-dimensional individuals are decomposable to primitive single domain operators.

We will now examine the models of $H2$ and in particular their structure and the semantics of evaluating two dimensional operators in them. First, we introduce the notation used in the rest of this section. $Op(L_C)$ denotes the set of non-logical symbols of a first-order language L_C which is used to express statements about a theory C. Let (A, op_a) and (B, op_b) are two $\mathcal{L}$-structures of some first order language $\mathcal{L}$ the cartesian product of $(A, op_a) \otimes (B, op_b)$ will be $(A \otimes B, op_a \circ op_b)$ such that for each $\langle a_1, b_1\rangle, \langle a_2, b_2\rangle \in A \otimes B$, $\langle a_1, b_1\rangle\ op_a \circ op_b\ \langle a_2, b_2\rangle$ is equivalent to $a_1 op_a a_2 \wedge b_1 op_b b_2$.

The idea is to define a combination of two logic systems. For that we assume that the language of a system is given by set of symbols and a set of formula building rules such as the ones described in Section 3.3. Let C_1, C_2 be two copies of the theory of one-dimensional space referring to different coordinate lines. Statements about C_1, C_2 are expressed in using the same language, i.e., the formula building rules are the same, but on different sets of symbols. In general, we do not want any non-boolean operator t be shares between the two copies of the language; this may cause problems when we combine their axiomatizations. To avoid such a behavior, independent combination imposes the restriction $Op(L_{C_1}) \cap Op(L_{C_2}) = \varnothing$. The combined theory is $H2$. As far as its axiomatization is concerned, let Σ_1, Σ_2 be axiom systems of C_1, C_2, respectively, then $\Sigma_1 \cup \Sigma_2 \cup \{B1, B2, \ldots, B6\}$ is the axiom system of $H2$.

The models of the independently combined theory of two-dimensional haze space are composed out of connecting the models of the two coordinate one-dimensional theories in the following way. Let the structures $\mathsf{P}_1 = \langle \mathsf{dom}(\mathsf{P}_1), \mathsf{h}_1, <_1, \mathsf{g}_1\rangle$ [4] and $\mathsf{P}_2 = \langle \mathsf{dom}(\mathsf{P}_2), \mathsf{h}_2, <_2, \mathsf{g}_2\rangle$ be models of C_1, C_2 (as defined in Section 3.4), then the models of the combined theory have the form $\langle \mathsf{dom}(\mathsf{P}_1), \mathsf{dom}(\mathsf{P}_2), \mathsf{dom}(\mathsf{P}_1) \otimes \mathsf{dom}(\mathsf{P}_2), <_1, <_2, <_1 \circ <_2, \mathsf{h}_1, \mathsf{h}_2, \mathsf{h}_1 \circ \mathsf{h}_2, \mathsf{g}_1, \mathsf{g}_2 \mathsf{g}_1 \circ \mathsf{g}_2\rangle$. Domains $\mathsf{dom}(\mathsf{P}_1), \mathsf{dom}(\mathsf{P}_2)$ are the domains that sorts P_1, P_2 are interpreted at, where domain $\mathsf{dom}(\mathsf{P}_1) \otimes \mathsf{dom}(\mathsf{P}_2)$ serves as an interpretation for sort Q. Due to axioms B1 and B2, $\mathsf{dom}(\mathsf{P}_1) \otimes \mathsf{dom}(\mathsf{P}_2)$ is nothing but the cartesian product of $\mathsf{dom}(\mathsf{P}_1)$ and $\mathsf{dom}(\mathsf{P}_2)$.

[4] Note that h_1, in this font, is the relation of P_1 that is used for the interpretation of the relation symbol h_1 of C_1.

Analogously, relation $<_1 \circ <_2$ stands as the interpretation of the vector operator $\langle <_1, <_2 \rangle$, and so on.

Given an independent combination, every sentence of the combined language is decomposed into a C_1 formula, a C_2 formula and a *P(airs)*-part. Two projection operators H and V separate the parts of the sentence that correspond to C_1 and C_2 respectively. In specific, any $H2$ sentence of the form $HV(\forall x_1..x_i/Q\ \Phi)$ is written as

$$\forall x_1..x_i/Q\ \exists x_{1_1}..x_{i_1}/P_1\ \exists x_{1_2}..x_{i_2}/P_2\ (x_1 = \langle x_{1_1}, x_{1_2}\rangle \wedge ... \wedge x_i = \langle x_{i_1}, x_{i_2}\rangle \wedge H(\Phi) \wedge V(\Phi))$$

where $H(\phi)$ (resp. $V(\phi)$) involves only C_1 (resp. C_2) symbols, and $\bigwedge_i x_i = \langle x_{i_1}, x_{i_2}\rangle$ is the P-part and $Q = P_1 \otimes P_2$. Subscripts $_1$, $_2$ denote symbols from C_1 and C_2, respectively.

The H and V projections are obtained as follows:

1. atomic formulas

ϕ	$H(\phi)$	$V(\phi)$	P
$x_1 \prec_1 y_1$	$x_1 \prec_1 y_1$	T	$x = \langle x_1, x_2\rangle \wedge y = \langle y_1, y_2\rangle$
$x_2 \prec_2 y_2$	T	$x_2 \prec_2 y_2$	$x = \langle x_1, x_2\rangle \wedge y = \langle y_1, y_2\rangle$
$h_1(x_1, y_1)$	$h_1(x_1, y_1)$	T	$x = \langle x_1, x_2\rangle \wedge y = \langle y_1, y_2\rangle$
$h_2(x_2, y_2)$	T	$h_2(x_2, y_2)$	$x = \langle x_1, x_2\rangle \wedge y = \langle y_1, y_2\rangle$
$op(x, y)$	$H(\phi_1)$	$V(\phi_2)$	$x = \langle x_1, x_2\rangle \wedge y = \langle y_1, y_2\rangle$

where op is one of the $\{hh,\ w\_colinear_h,\ w_colinear\_v,\ east,\ west,\ north,\ south\}$.

2. non-atomic formulas

Φ	$H(\Phi)$	$V(\Phi)$	$P(\Phi)$
$\neg\phi$	$\neg H(\phi)$	$\neg V(\phi)$	none
$\phi \wedge \psi$	$H(\phi) \wedge H(\psi)$	$V(\phi) \wedge V(\psi)$	none
$\forall x/Q(\phi(x))$	$\forall x_1/P_1(H(\phi(x_1))$	$\forall x_2/P_2(V(\phi(x_2))$	$x = \langle x_1, x_2\rangle$

$\exists$, $\vee$, $\Rightarrow$ formulas are derived in the usual way.

Proposition 3.5.1 *Let H_2 be an independent combination of C_1 and C_2, and $M = M_1 \otimes M_2$ the combined structure of M_1 and M_2 where M_1 and M_2 are models of C_1 and C_2, respectively. An $H2$ sentence ϕ is true in M under a valuation v, if and only if $H(\phi)$ and $V(\phi)$*

are true in M_1, M_2 with valuations v_1 and v_2 and $v = v_1 \circ v_2$, i.e.,

$$M \models \phi[v] \text{ iff } M_1 \models H(\phi)[v_1],\ M_2 \models V(\phi)[v_2] \text{ and } v = v_1 \circ v_2 = \{\langle x_1, x_2 \rangle | x_1 \in v_1 \text{ and } x_2 \in v_2\}$$

Proof: By induction on the structure of Φ.

Base case: ϕ is atomic. All five possibilities need to be considered. In fact, due to their similarity, we group the first four into one case, which, without any loss of generality, we assume to be the following: $H(\phi) = x_1 \prec_1 y_1$ and $V(\phi) = T$. Then

$M \models \forall xy/Q \exists x_1 y_1/P_1 \exists x_2 y_2/P_2 (x = \langle x_1, x_2 \rangle \wedge y = \langle y_1, y_2 \rangle \wedge x_1 \prec_1 y_1 \wedge T) \Leftrightarrow$

$M \models \forall xy/Q \exists x_1 y_1/P_1 (x = \langle x_1, vert(x) \rangle \wedge y = \langle y_1, vert(x) \rangle \wedge x_1 \prec_1 y_1 \wedge T) \Leftrightarrow$

$M \models (\forall xy/Q (x = \langle x_1, vert(x) \rangle \wedge y = \langle y_1, vert(x) \rangle)[v_1]$ and
$M \models (x_1 \prec_1 y_1)[v_1]$ and $M \models T \Leftrightarrow$

$M_1 \otimes M_2 \models (x = \langle x_1, vert(x) \rangle \wedge y = \langle y_1, vert(y) \rangle)[v]$ and
$M_1 \models (x_1 \prec_1 y_1)[v_1]$ and
$M_2 \models T$ and
$v_2 = \{a_2 | \langle a_1, a_2 \rangle \in v,\ a_1 \in v_1\}, a_i$'s are constants.

The remaining base case is if ϕ is atomic of the particular type $op(x, y)$ with $H(\phi)$ and $V(\phi)$ to be $op_1(x_1, y_1)$ and $op_2(x_2, y_2)$, respectively. Then

$M \models \forall xy/Q \exists x_1 y_1/P_1 \exists x_2 y_2/P_2 (x = \langle x_1, x_2 \rangle \wedge y = \langle y_1, y_2 \rangle \wedge op_1(x_1, y_1) \wedge op_2(x_2, y_2)) \Leftrightarrow$

$M \models (x = \langle hor(x), vert(x) \rangle \wedge y = \langle hor(y), vert(y) \rangle)[v]$ and
$M \models op_1(x_1, y_1)[v]$ and
$M \models op_2(x_2, y_2)[v] \Leftrightarrow$

$M \models (x = \langle hor(x), vert(x) \rangle \wedge y = \langle hor(y), vert(y) \rangle)[v]$ and
$M_1 \models op_1(x_1, y_1)[v_1]$ and
$M_2 \models op_2(x_2, y_2)[v_2]$ and
$v = \{x | hor(x) = x_1 \in v_1,\ vert(x) = x_2 \in v_2\}$

Inductive step: The inductive step is straightforward. To simplify the presentation, we will use the $\bar{\bar{x}}$ notation as short form of $x_1, x_2, \ldots, x_n$, and $\langle \bar{x}, \bar{y} \rangle$ as equivalent of $\bigwedge_i x_i = \langle x_{i_1}, x_{i_2} \rangle$. Then an $H2$ sentence is written as $\forall \bar{x}/Q \exists \bar{x}_1/P1 \exists \bar{x}_2/P2 (\bar{x} = \langle \bar{x}_1, \bar{x}_2 \rangle \wedge \Phi(\bar{x}))$.

$\Phi(\overline{x})$ might be one of $\Phi_1(\overline{x}) \wedge \Phi_2(\overline{x})$, $\Phi_1(\overline{x}) \vee \Phi_2(\overline{x})$ and $\neg\Phi_1(\overline{x})$. We perform the inductive step for first case only, since the rest of them are similar.

$M \models \forall\overline{x}/Q\exists\overline{x}_1/P1\exists\overline{x}_2/P2(\overline{x} = \langle\overline{x}_1, \overline{x}_2\rangle \wedge (\Phi_1(\overline{x}) \wedge \Phi_2(\overline{x})) \Leftrightarrow$

$M_1 \otimes M_2 \models \forall\overline{x}/Q\exists\overline{x}_1/P1\exists\overline{x}_2/P2(\overline{x} = \langle\overline{x}_1, \overline{x}_2\rangle \wedge \Phi_1(\overline{x}) \wedge \Phi_2(\overline{x}) \Leftrightarrow$

$M_1 \models \forall\overline{x}/Q\exists\overline{x}_1/P1\exists\overline{x}_2/P2(\overline{x} = \langle\overline{x}_1, \overline{x}_2\rangle \wedge \Phi_1(\overline{x}) \wedge \Phi_2(\overline{x})$ and
$M_2 \models \forall\overline{x}/Q\exists\overline{x}_1/P1\exists\overline{x}_2/P2(\overline{x} = \langle\overline{x}_1, \overline{x}_2\rangle \wedge \Phi_1(\overline{x}) \wedge \Phi_2(\overline{x})$,
which is true by the induction hypothesis. □

A similar method for combining two one-dimensional logic systems into a two-dimensional system in the context of temporal logics, has been proposed by Finger [Fin93]. Finger showed that the independent combination of two sound, complete and decidable logic systems carries the aforementioned properties.

3.6 The Theory of Two-dimensional Rectangles

We will now extend $\mathcal{L}$ with an additional sort, R, for rectangles. A haze rectangle is constructed by a pair of two-dimensional haze points. The new symbols in the language include two predicate symbols for *inclusion*, and a pairing function symbol:

- $in_0(p, a)$ point p is inside rectangle a
- $in_1(a, b)$ rectangle a is inside rectangle b
- $[p, q]$ is the pairing function which returns the convex rectangle which is formed by the two-dimensional points p, q.

The necessary axioms for the augmented theory are stated in the sequel. The new theory is called HR. For notational convenience, the letters x, y, z, u will be taken as individual variables ranging over Q, and the letters a, b, c, d will be taken as individual variables ranging over R.

Similarly to the point pairing function, rectangle pairing possesses the axioms:

D1. $\forall a/R\ \exists xy/Q\ (a = [x, y] \wedge east(x, y) \wedge south(x, y))$

D2. $\forall xyzu/Q\ ([x, y] = [z, u] \Rightarrow x = z \wedge y = u)$

D3. $\forall xy/Q\ (low([x, y]) = x)$

D4. $\forall xy/Q\ (high([x, y]) = y)$

Axiom D1 states that each rectangular region is composed of two two-dimensional points: a bottom-left x point and a top-right y point, i.e., x is in east-south relation to y. Due to axiom C5, x and y cannot be in the haze of each other (hh relation). To avoid inconsistencies we restrict the domain of R to consist of those pairs of $x, y \in Q$ for which $hh(x, y)$ does not hold. Axiom B2 postulates that there is a unique way to construct each distinct rectangle in R. Axioms D3 and D4 name the first and the second component point of a rectangle pair to by *low* and *high*, respectively. Then, the rectangle decomposition property follows from D1,D3,D4.

$$\forall a/R\ (a = [low(a), high(a)])$$

We abbreviate formula $east(x, y)\ \wedge\ south(x, y)$, by $x <_D y$, meaning that x "diagonally" precedes y. Then, we can state that all points included in a rectangle must be diagonally included by its low and high point. Also, any two rectangles are in inclusion relation, i.e., in_1, if and only if the inner rectangle's points are diagonally enclosed by the outer's extreme points.

D5. $\forall x/Q \forall a/R\ (in_0(x, a)\ \Leftrightarrow\ low(a) <_D x <_D high(a))$

D6. $\forall ab/R(in_1(a, b)\ \Leftrightarrow\ low(b) <_D low(a) <_D high(a) <_D high(b))$

In fact, D4,D5 define the two inclusion relations in terms of our ontological primitives, namely, points, haze and precedence. in_1 is a partial order relation (transitive, reflexive and antisymmetric) and in_0 is transitive over in_1.

The following topological relations are defined in terms of in_1. Their definitions are self explanatory.

E1. $overlaps(a, b) \equiv \exists c(in_1(c, a)\ \wedge\ in_1(c, b))$

E2. $disjoint(a, b) \equiv \forall c(\neg in_1(c, a)\ \vee\ \neg in_1(c, b))$

The smallest, non-decomposable area in our representation is the haze area that surrounds a point and is called *atomic*. Any other (non-atomic) region is called *properRegion*.

The precedence relations between rectangles are also defined in terms of the corresponding relations in Q. As each rectangle is seen as a pair of two haze point objects, then precedence is defined as:

$$a\ op\ b \equiv high(a)\ op\ low(b)$$

where $op \in \{east, south, <_D\}$ as defined for two-dimensional haze points. Transitivity, irreflexivity and asymmetry of these operators in R is easily derived.

The following axioms combine the notions of inclusion and precedence.

D7. $\forall ab/R\ (a\ op\ b\ \Rightarrow\ \neg overlaps(a,b))$

D8. $\forall ab/R\ (a\ op\ b\ \Rightarrow\ \forall c/R\ (in_1(c,a)\ \Rightarrow\ c\ op\ b))$

D9. $\forall ab/R\ (a\ op\ b\ \Rightarrow\ \forall c/R\ (in_1(c,b)\ \Rightarrow\ a\ op\ c))$

D10. $\forall abc/R\ (a\ op\ b\ op\ c\ \Rightarrow\ \forall d/R\ (in_1(a,d) \wedge in_1(c,d)\ \Rightarrow\ in_1(b,d)))$

3.6.1 Models of HR

Inclusion is the base primitive for axiomatizing the theory of rectangles. In our ontology, *inclusion*, a purely topological concept, is expressed in terms of the primitives of the ontology (axioms D5,D6), thus showing that HR is an extension of $H2$. In consequence, only the pairing function, "[,]", and the axioms D5-D10 are needed to define HR as an extension of $H2$, and they follow the syntactic rules of the independent combination. All the other sentences presented under HR are theorems in it.

3.7 Binary Spatial Relations

In this section we use the developed framework for the definition of *topological* and *directional relations*. In particular, we define Egenhofer's topological [EF91], [Ege91] and Hernandez's directional relations [Her92] within the same framework.

Topological relations are defined between pairs of rectangles. In the cartesian plane, a rectangle is defined by two points, the left-bottom and the right-top point. When points are assumed to be haze points the extension of the rectangle may grow by size g at each side. This means that there is uncertainty about the exact position of the rectangle and the size of its area. In our simplified domain we can visualize a rectangle having a stripe of width $2g$ as boundary. The real shape can be placed anywhere in this area.

$\mathtt{disjoint}(a,b) \equiv \neg in_1(a,b) \vee \neg in_1(b,a)$

$\mathtt{tangent_1}(a,b) \equiv \exists p,q(in_0(p,a) \wedge in_0(q,b) \wedge hh(p,q))$

$\mathtt{overlap}(a,b) \equiv \exists c(in_1(c,a) \wedge in_1(c,b)$

$\mathtt{inside_t}(a,b) \equiv in_1(a,b) \wedge \forall c(\mathtt{tangent_1}(c,a) \wedge \mathtt{tangent_1}(c,b))$

$\mathtt{inside_i}(a,b) \equiv in_1(a,b) \wedge \neg\exists c(\mathtt{tangent_1}(c,a) \wedge \mathtt{tangent_1}(c,b)$

$\texttt{contain}_t(\mathrm{a,b}) \equiv \texttt{inside}_t^{-1}(a, b)$

$\texttt{contain}_i(\mathrm{a,b}) \equiv \texttt{inside}_i^{-1}(a, b)$

$\texttt{equal}(\mathrm{a,b}) \equiv in_1(a, b) \wedge in_1(b, a)$

The notion of tangency as defined here is "loose" tangency in the sense that two the rectangles share a point of their haze. If the size of the haze decreases tangent relationship changes to disjointness.

Directional relations are defined between pairs of points. The first point is the reference and the second is the referencing point. The characterization of direction is done by means of the precedence and haze relations. Thus, granularity plays a role here, as well. Axiom C5 defines nine disjoint positionings of a haze point y with respect to a haze point x. The eight of them are directional: `east`(2), `west`(3), `north`(4), `south`(5), `north_east`(6), `south_east`(7), `north_west`(8) and `south_west`(9) [5].

3.8 Granularity

In this section we study the implications of supporting multiple granularity. There are two ways to support multiple granularities. The first is to change the constant g in the original theory $H1$ to a variable of sort S. Then, haze points are interpreted by integer intervals of arbitrary length.

The second way to incorporate granularity change is to increase or decrease the value of the constant g. This means that the haze area of all points in the representation increases or decreases respectively and therefore the truth/falsehood of its statements changes as well. The following axioms state these changes.

G1.($\rightarrow$) $\forall g_1 g_2/S\ (g_1 < g_2 \Rightarrow \forall xy/P\ (h(x, y, g_1) \Rightarrow h(x, y, g_2)))$ coarsening

G2.($\leftarrow$) $\forall g_1 g_2/S\ (g_2 < g_1 \Rightarrow \forall x/P \exists y/P\ (h(x, y, g_1) \wedge \neg h(x, y, g_2)))$ refinement

The idea is that when the granularity changes then the original theory, say T_0, has to change. In particular, G1 states that two points that are indistinguishable at g_1, they continue to be so at $g_2 > g_1$ ($\rightarrow$). G2 states that the transition from finer, g_1, to coarser granularity, $g_2 < g_1$, does not preserve indistinguishability ($\leftarrow$). Our future research goal is to investigate syntactic methods for computing the theories $T_{\leftarrow}$ and $T_{\rightarrow}$.

[5] The number enclosed in parentheses identifies the corresponding disjunct of axiom C5

An early work towards a formal treatment of granularity in reasoning systems was presented by Hobbs [Hob85], where the use of a transitive indistinguishability relation was proposed to select a local theory out of a global one. One of Hobbs conclusions was that grain size should be an explicit argument of many predications. In our case, we chose the h relation to be the grain-dependent predicate and we developed a theory using it as a primitive. One advantage of our solution is that local, and possibly minor, changes of the theory can reflect major modifications, because h influences the definition of many binary spatial relations. Our goal is to explore rewriting techniques for carrying out this task and provide soundness and completeness guarantees for theories of a certain syntactic type.

Our work is also compared to *temporal modules* and *time units* proposed by Wang et al [WJS93]. This work, deals with the problem of mismatches in temporal databases with different units of time. Their solution is to superimpose various units of time, finer or coarser, over the predefined unit that appears in the database. That method has a global effect in the sense that when the time unit changes the change affects the truth of all the domain facts. Our approach adds a granularity argument to spatial relations; therefore, changes of the grain size affect only the truth of neighbouring relations – assuming a notion of neighbouring relations similar to the one presented in [Fre92a].

Finally, the rationale for supporting granularity in our model is not only to allow for imprecise descriptions but also to support *indeterminate* retrievals. In the latter case, an indeterminate query, underconstrained in the AI terminology, is initially directed to the coarser theory and its answer is refined as the logical theory is refined.

3.9 Conclusion

In this chapter we presented a concise and formal treatment of imprecision in one-dimensional and two-dimensional space which appears useful as an underlying framework for addressing the spatial reasoning and management questions of the subsequent chapters. Our discussion extends to k dimensional spaces provided that the conditions of independent composition are preserved when adding dimensions. The chapter also developed a mathematical basis for the definition of spatial relations and the formal handling of concepts such as scale and granularity in space.

The technical results presented in this chapter are summarized as follows: Initially we developed a first-order theory of one-dimensional space, $H1$, with haze and precedence relations, and we showed that its models are partial orders on a discrete domain. We proposed a conservative two-dimensional extension of $H1$, called ***independent combination***, in which the evaluation of two-dimensional operators is reduced to the evaluation of projected one-dimensional operators over two coordinate copies of H1. From the two-dimensional point theory, $H2$, and by independent combination, we derived the theory of haze rectangles, HR. Finally, we outlined the effects of haze in reasoning with varying granularity.

Chapter 4

Reasoning with Qualitative Constraints

4.1 Introduction

This chapter presents efficient algorithms for qualitative spatial reasoning based on the haze-point ontology introduced in the previous chapter.

As established in Chapter 3, space in haze-point ontology is viewed as a totality of spatial objects connected in terms of spatial relations. A haze point is the most primitive object type which has non-zero size. A haze point can be thought in terms of an area of haze such that the point in question may be located anywhere inside it. Haze points are related in terms of haze or precedence relations. The former means that two objects are "close" to each other and therefore indistinguishable. The latter means that one object (one-dimensionally) precedes the other. Reasoning with spatial objects and their relationships in haze-space essentially amounts to reasoning with one-dimensional haze points and the haze and precedence relations (or the model of haze-orders, for short). Answers to the reasoning problem drawn in this context are then combined to create answers to the reasoning questions using higher-order constructs.

Reasoning about haze-orders involves, first, determining the consistency (satisfiability) of a set of haze-order assertions, and, second, deducing new relations from those that are already known (i.e., computing the closure of the input haze-order assertions). In this chapter we study both reasoning questions in the context of one-dimensional haze-order space.

The rest of this chapter is organized as follows. Section 4.2 introduces the structure of the qualitative graphs which underlie the reasoning algorithms developed in Section 4.3. Section 4.4 presents an alternative formulation of the haze-order reasoning problem through constraint relational algebras. Section 4.5 presents an experimental evaluation of the proposed algorithms. Finally, Section 4.6 summarizes the contributions and concludes the chapter. [1]

4.2 Haze-Order Graphs

In this section we summarize the definitions and the theorems on which reasoning about haze-orders is based. We also introduce a graph-based data structure, called *haze-order graph*, that is based on the haze space ontology presented in the previous chapter and used for the representation of a set of haze-order constraints.

Definition 4.2.1 *A haze-order constraint is a conjunction of haze-order terms. A haze-order term is an atomic term of one of the following types: $h(x, y)$ and $x \prec y$, where h and $\prec$ are the haze and precedence relations defined in Section 3.4.*

Haze-order constraints are a special case of the language of haze point space introduced in Section 3.4. The selected special case of the initial theory is motivated, first, by the starting requirement of this research which is to study efficient reasoning algorithms, and second, by the practical considerations that are discussed in Sections 5.4.2 and 5.4.3.

Haze-order constaints are represented in terms of haze-order graphs.

Definition 4.2.2 *A haze-order graph is a labeled graph whose vertices represent one-dimensional haze points and its edges represent binary haze-order relations that hold between points. Edges are denoted by triples (x, l, y) and are either directed (l is $<$) standing for $\prec$ relations or undirected (l is g) standing for h relations.*

Figure 4.1 illustrates several haze-order graphs. For instance, graph (e) represents the set of haze-order constraints $\{x \prec z, h(x, y), h(y, z)\}$. Every vertex of a haze-order graph is named

[1] The contents of Sections 4.2 and 4.3 have been published in [Top96a]. An preliminary version of the contents of Chapter 4 have also appeared in [Top94b].

by a distinct variable name. It must be noted that in the language of haze-order constraints we do not include constants to name specific haze points; instead, we use unbound variables to name points with the understanding that any particular model provides an interpretation for free variables.

Definition 4.2.3 *In a haze-order graph hG, a sequence of n successive edges* (x_i, l_i, x_{i+1}) *defines a path of length n. If all the labels along a path are* $<$*, then the path is a* $<$*-path. A path* $(x_1, l_1, x_2) \ldots (x_{i-1}, l_{i-1}, x_i), (x_i, l_i, x_1)$ *is a cyclic path. A cyclic path with both* $<$ *and g edges is called* $<$ *g-cycle. An* $<^i, g^j$*-cycle is a* $<, g$*-cycle with i occurrences of* $<$ *edges and j occurrences of g edges.*

The vertices of a haze-order graph are interpreted with respect to a totally ordered set, i.e., the integers with the "less than" order relation. According to Proposition 3.4.1 of Section 3.4.1, there is a duality in the way that the models of $H1$ are represented, i.e., either as same length intervals or points over the integers line which have to be in certain distance to each other. In this section we follow the second approach. Hence, a model of a haze-order constraint set is a mapping from vertices to the integers, such that the values assigned to any two vertices satisfy the relation represented by the edge connecting them. A model also assigns a value g to the parameter that stands for the size of the *haze*. All these are summarized formally in the following definition:

Definition 4.2.4 *Given a haze-order graph,* $hG = (V, E)$*, a model is a quintuple* $(v, P, R_<, R_g, \mathrm{g})$ *where P is a totally ordered set, v is a mapping from its vertices V to P,* g *is a constant denoting the haze size,* $R_<$ *is a binary relation whose elements are pairs of P elements such that for every* $(x, <, y) \in E$*,* $\langle v(x), v(y)\rangle \in R_<$ *and* R_g *is a binary relation whose elements are the pairs of P such that for every* $(x, g, y) \in E$*,* $\langle v(x), v(y)\rangle \in R_g$ *.*

The set of integers, $\mathbb{Z}$, along with the relations $\mathbb{Z}_<$ and $\mathbb{Z}_g$ establish the interpretation structure for haze-order graphs in this chapter. For some g, relations $\mathbb{Z}_<$ and $\mathbb{Z}_g$ are defined as follows:

$$\mathbb{Z}_< = \{(a, b) | a, b \in \mathbb{Z} \text{ and } b - a > \mathrm{g}\}$$

$$\mathbb{Z}_g = \{(a, b) | a, b \in \mathbb{Z} \text{ and } |a - b| \leq \mathrm{g}\}$$

Note that there is more than one ordering that can satisfy a set of haze-order constraints. For example, the constraints in $C = \{x \prec y, h(y, z)\}$ are satisfied by the orderings $v(x) <$

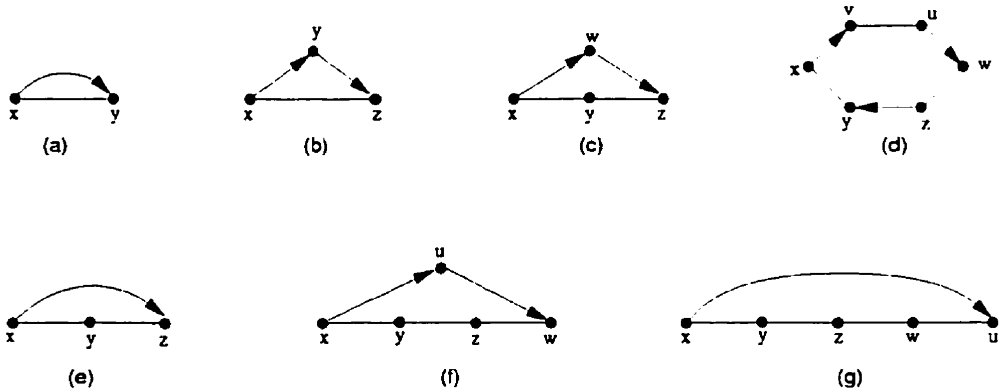


Figure 4.1: Inconsistent (a-d) and consistent (e-f) $<^i, g^j$-cycles.

$v(y) < v(z)$ and $v(x) < v(z) < v(y)$, where $v(x)$ denotes the integer value assigned to x. Both orderings are models of C. A haze-order graph is consistent if it has at least one model. Theorem 4.1 establishes a graph-theoretic condition which guarantees the consistency of a haze-order graph. This condition is enforced by the consistency checking algorithm of Section 4.3.1.

Theorem 4.1 *A haze-order graph is consistent if and only if it does not contain any $<^i, g^j$-cycles with $i \geq j$.*

Proof. **(Only if)** If $j = 0$ we end up with an $<$-cycle which is inconsistent due to the irreflexivity of $\prec$. If $j \neq 0$ and $i > j$, then we can cancel any g-edge on the path and its preceding $<$-edge and still obtain a $<$-cycle which induces inconsistency. If $i = j$ an extreme case is encountered in which all the g-edges, (x, g, y), need to be interpreted by an ordering where y precedes x (i.e., the g-edges materialize to the counter direction of the $<$-edges), and all the $<$-edges, $(z, <, x)$, are tight (i.e., the distance between z and x is just above the g threshold). As a result, for any pair of successive $<$ and g-edges, e.g., $(z, <, x) \wedge (x, g, y)$, z will precede y. Applying this inductive argument for the $<^i, g^i$-cycle starting from a vertex v, we will eventually find that v precedes itself, since the cycle closes at v, leading to a contradiction and therefore to inconsistency. The $i \geq j$ condition is illustrated in Figure 4.1, Graphs (a)-(d) satisfy the condition and therefore are inconsistent whereas graphs (e)-(g) are consistent. In particular, graph (f) is a $<^2, g^3$-cycle with the ordering $\{v(x), v(y), v(v), v(z), v(w)\}$ being a consistent model for it.

(If) We need to show that any consistent graph does not contain a $<^i, g^j$-cycle with $i \geq j$.

To show this, we use inductive on the number of edges. For the cases of 1 (trivial graph), 2 and 3 vertices, the base cases, this is verified by inspection of graphs (a), (b) and (e) in Figure 4.1. In the induction step, we assume a consistent graph without an $<^i, g^j$-cycle, $i \geq j$, and insert an edge. Adding an $<$-edge will leave the graph consistent as long as it doesn't close a $<$-cycle or become the ith $<$-edge in a $<^i, g^j$-cycle and $i = j$. In any other case a consistent model exist. The constraint imposed by the new edge refines the allowable existing orderings of the graph vertices by eliminating those that violate it. Adding a g-edge, as long as it is not the jth g-edge that closes a $<^i, g^j$-cycle and $i = j$, leaves the consistency of the graph unchanged. □

From the remarks made in the proof of theorem 4.1 we conjecture additional conditions that a consistent haze-order graph with $<^i, g^j$-cycles must satisfy. Observe the graph in Figure 4.2(a). It contains a $<^2, g^3$-cycle, and thus it is consistent. The ordering $\{v(x), v(w), v(y), v(v), v(u), v(s), v(z), v(t)\}$ is a potential model for it. The condition that must be satisfied in order for the above ordering to become a model is that all the order and the haze relations are satisfied. This requires that there exists an assignment from vertices to integer values and an integer value g that satisfies all the constraints. The selection of allowable g values involves resolving some additional metric constraints. The expression $i * (\mathrm{g} + 1) - j * \mathrm{g} = 0$ determines the lower bound for the g value. The meaning of the expression is that in the boundary case, that is, $i = j - 1$, all the g-edges are ordered in the counter direction of the $<$-edges, and $<$-edges are all tight (i.e., just above the g threshold). In this case a g exists such that it guarantees that the length cycle remains zero and therefore is consistent. If $i < j - 1$ (non-boundary case), this condition changes to $\sum_{n=1}^{i}(\mathrm{g} + a_n) - (j - k) * \mathrm{g} + b_k = 0$ where a_i is an integer increment (≥ 1) by which each $<$-constraint exceeds the g threshold, $k > j - i$ is the number of g-edges, (x, g, y), that are interpreted by a $\langle v(x), v(y)\rangle$ ordering, and $b_k = \sum_l^k [v(y)_l - c(x)_l]$. The cycle in Figure 4.2(a) requires that g is at least 2 due to the above condition. Hence, a model can be $\langle \mathrm{g} = 2, v(x) = 0, v(w) = 2, v(y) = 3, v(v) = 4, v(u) = 5, v(s) = 5, v(z) = 6, v(t) = 8\rangle$. Figure 4.2(b) shows the sorted cycle.

The minimum g value for each $<^i, g^j$-cycle, determined by the above expression, defines the maximum degree of detail for which a consistent assignment for an $<^i, g^j$-cycle exists. Therefore, a maximum g value that makes all $<^i, g^j$-cycles of a haze-order graph consistent will characterize the minimal model among the potentially many possible models of the

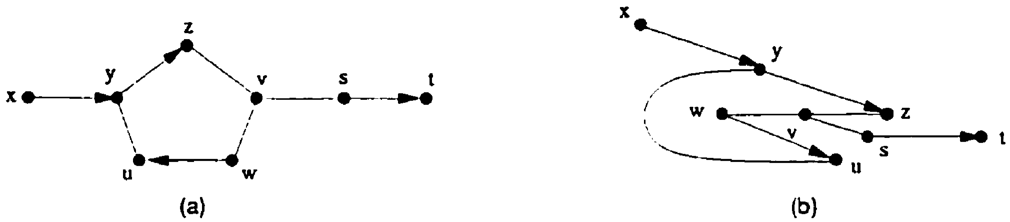


Figure 4.2: Ordering $<^i, g^j$-cycles.

graph.

Definition 4.2.5 *Two haze-order graphs are logically equivalent if they have the same set of models.*

Definition 4.2.6 *Given a haze-order graph, an edge (x, g, y) is disambiguated to the left (resp. right) if the ordering $\langle v(y), v(x)\rangle$ (resp. $\langle v(x), v(y)\rangle$) cannot occur in any of its graph's models.*

Notation: Label $\lhd$ denotes a "close-order" relationship; $(x, \lhd, y)$ says that although x, y are close to each other, x is slightly preceding y. Disambiguation of g-edges introduces a new edge label and therefore a new type of edges. $\lhd$-paths ands $<, \lhd, g$-paths are also defined in a similar fashion as the $<, g$-paths.

Proposition 4.2.1 *In a haze-order graph, a g-edge between vertices x, y is disambiguated, if*

1. *(g-edges over triangles) vertices z,x and y are connected by a g^2-path and z, y by an $<$-edge, then the g-edges disambiguate to $(z, \lhd, x)$ and $(x, \lhd, y)$ (Figure 4.3);*
2. *($g^3, <$-cycles) the two out of three vertices in a $g^3, <$-cycle ordered as the last g-edges are disambiguated in the counter direction of the $<$-edge;*
3. *($g^i, <^{i-1}$-cycles) all the i in the number g-edges of a $g^i, <^{i-1}$-cycle are disambiguated in the counter direction of the $<$-edges (Figure 4.2).*

Proof The proof is based on the same argument as the proof of Theorem 4.1 since all the above are special cases of $<^i, g^j$-cycles. 1 and 2 are trivial cases as illustrated in Figure 4.3. An informal proof justification for 3 is given in the post-Theorem 4.1 discussion. □

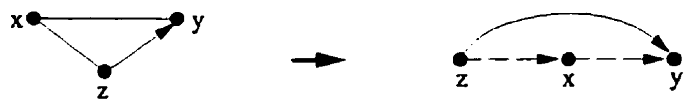


Figure 4.3: Disambiguating g-edges

Long g-paths represent many possible orderings. The consequence of this proposition is that in certain cases we can prune some impossible orderings by just using the structure of the haze-order graph. This is beneficial in the design of an order inferencing algorithm.

Proposition 4.2.2 *The vertices of any g-cycle with* $2n+1$ *edges can be interpreted inside an interval with maximum length* ng.

Proof In the extreme case that half of the g-edges are disambiguated as $\lhd$-edges and the other half as $\rhd$-edges (opposite $\lhd$-edges) and the maximum haze size is considered, their $2n+1$ vertices span over an interval with length ng. The same also holds for $2n$ edges. $\square$

Definition 4.2.7 *A disambiguated haze-order graph is a haze order graph with all the g-edges possible to disambiguate, disambiguated.*

The corresponding disambiguated graph, dG, of some haze-order graph, hG, has the same set of vertices and edges as hG and a richer set of labels.

In the following, we wish to entail the relations that hold between any two haze points x and y given a haze-order graph representation. Most importantly, we want to entail the strongest relations between any two points. For any two points, we want to decide whether the one precedes the other (i.e., $x \prec y$) or are in haze relation (i.e., $h(x,y)$) or if either relation can hold (i.e., $x \prec y \vee h(x,y)$) or, finally, if the universal relation holds (i.e, $x \prec y \vee h(x,y) \vee y \prec x$). The meaning of the universal relation is that any relative positioning of the two points is possible.

Notation: If the expression $r_1(x,y) \vee ... \vee r_n(x,y)$ is entailed by a haze-order graph, then we say that relation $R = \{r_1, .., r_n\}$ holds between the two points x, y.

The two lemmata below present a graph-theoretic definition for the notion of entailment.

Lemma 4.2.1 *Let hG be a consistent haze-order graph and let* x, y *be two vertices of hG; hG entails*

1. $x \prec y$, *if there is an* $<$*-edge or an* $<$*-path or a* $<^i, g^j$*-path with* $i > j$ *between* x *and* y;

2. *$x \prec y \vee h(x,y)$, if there is a $<^i, g^j$-path with $j = i \vee j = i-1$, between x and y;*

3. *$h(x,y)$, if there is a g-edge between x and y;*

4. *the universal relation in all other cases.*

Proof. Since the haze-order graph is consistent, there are no forbidden $<^i, g^j$-cycles. However, there might be multiple paths connecting any two of its nodes. First, we consider single paths. The entailment of $\prec$ relationships is justified by the transitivity of $\prec$ if the two nodes are connected either by a $<$-edge or an $<$-path, and by the quantitative constraint stated in the proof of Theorem 1. If there is a $<^i, g^j$-path between x and y, i.e., if all the g-edges disambiguate to the opposite direction of the $<$-edges, then still due to the the definition of $\mathbb{Z}_<$ and $\mathbb{Z}_g$ and even if the $<$-edges are defined just above the g threshold, the $\prec$ relation is the only possibility between x and y. The $\{h, \prec\}$ relation is derived if the $\prec$ relation can not win over a $<^i, g^j$-path. This happens if the number of g-edges is equal to or exceeds by one the number of $<$ edges as suggested by the quantitative formula given earlier. $\{h, \prec\}$ cannot be derived if $j > i+1$ since the $<^1, g^3$-path, $\{x \prec y, h(y,z), h(z,t), h(t,s)\}$, can have a model which satisfies $s \prec x$. Thus, $<^i, g^j$-paths with $j > i+1$ should entail the $\{\prec, h \succ\}$ relation. The entailment of h is a trivial case. If two nodes are connected by multiple paths then the entailed relation will be the intersection of the relations entailed for each path. □

Lemma 4.2.2 *Let dG be the disambiguated graph of a haze-order graph hG and let x, y be two vertices of dG; dG entails*

1. *$x \prec y$, if there is an $<$-edge or a $<$-path or a $<, \lhd$-path or a $<^i, \lhd^k, g^j$-path with $i + k > j, k \geq 1$, between x and y;*

2. *$x \prec y \vee h(x,y)$, if there is a $<^i, \lhd^k, g^j$-path with $i + k = j \vee i + k - 1 = j$ and $i \neq 0 \vee k \neq 0$, between x and y;*

3. *$h(x,y)$, if there is g-edge or a $\lhd$-edge between x and y;*

4. *the universal relation in all other cases.*

Proof. The proof of Lemma 4.2.2 is an extension of the proof of Lemma 4.2.1. In particular, $\lhd$-edges appear as result of disambiguating g-edges. These edges allow us to refine some of the $x \prec y \vee h(x,y)$ inferences to $x \prec y$ inferences since they provide evidence that certain haze relations are one-sided. Edges labeled by $\lhd$ entail $h(x,y)$ relations because of Definition 4.2.6. □

Proposition 4.2.3 *Let hG be a haze-order graph and dG its corresponding disambiguated haze-order graph. If dG entails xRy and hG entails xR′y then R implies R′ (but not necessarily the converse).*

Proof. By applying Lemmata 4.2.1 and 4.2.2. □

In the rest of this section, we show that the entailment notion based on haze-order graphs computes stronger relations between haze-points, than the relations computed by the path consistency algorithm.

Qualitative constraint networks are common tools in the study of constraint satisfaction problems and strongest relations entailment in the context of temporal and spatial reasoning. A qualitative constraint network for haze-order constraints is a haze-order graph with a complete set of edges, i.e., each vertex is connected to all other vertices. The intersection and composition operations are key concepts for the consistency theory developed in the context of qualitative constraint networks.

Definition 4.2.8 *If R1 and R2 are two relations holding between a pair of points x, y, then the combined relation between x,y is defined by their intersection, $R = R1 \cap R2$.*

Definition 4.2.9 *The composition of two relations R1 and R2 holding between two pairs of points x, y and y, z, respectively, is defined as $R = R1 \otimes R2 = \cup_{r_1 \in R1, r_2 \in R2} T(r_1, r_2)$, where T is the composition operation between atomic relations (defined by a composition table).*

Figure 4.4 shows the tables that define the composition operations in the relation set represented in a haze-order graph (T_1) and a disambiguated haze-order graph (T_2). Notice that table T_2 is more detailed around the *haze* relation (labels $\lhd, g, \rhd$).

The following definitions are "classics" in constraint networks references [Mac77], [Fre82], [vB92], [GS95a], etc.

A constraint network is *arc-consistent* if for each pair of vertices the entailed relation is not empty, and it is *path-consistent* if for any triple of vertices, x, y and z, the condition

T_1	$<$	g	$>$
$<$	$<$	$\{<,g\}$	$\{<,g,>\}$
g	$\{<,g\}$	$\{<,g,>\}$	$\{>,g\}$
$>$	$\{<,g,>\}$	$\{>,g\}$	$>$

T_2	$<$	$\lhd$	g	$\rhd$	$>$
$<$	$<$	$<$	$\{<,\lhd\}$	$\{<,\lhd\}$	ALL
$\lhd$	$<$	$\lhd$	$\{g,\lhd\}$	g	$\{\rhd,>\}$
g	$\{<,\lhd\}$	$\{<,\lhd\}$	ALL	$\{g,\rhd\}$	$\{\rhd,>\}$
$\rhd$	$\{<,\lhd\}$	g	$\{\rhd,>\}$	$\rhd$	$>$
$>$	ALL	$\{\rhd,>\}$	$\{\rhd,>\}$	$>$	$>$

Figure 4.4: Composition tables

$R3 \subseteq R1 \otimes R2$ holds, where $R1, R2, R3$ are the relationships between (x,y), (y,z) and (x,z), respectively. Path consistency can be checked in $O(n^3)$ time, where n is the number of vertices in the constraint graph, using the Mackworth's path consistency (PC) algorithm [Mac77].

A constraint network is *minimal* if the relations holding between each pairs of its vertices are the strongest possible. Relation $R1$ is stronger than $R2$ if $R1$ implies $R2$ but not the reverse. An equivalent definition for a constraint network to be minimal is if every subnetwork relative to the overall network is strongly consistent (the relations between its vertices are minimal). The size of the subnetwork depends on the deployed constraint language. [2]

In the following we investigate conditions that determine minimality in haze-order graphs. A haze-order graph is minimal if the relation entailed for each pair of its vertices is the strongest possible. Unfortunately path consistency can not guarantee minimality in haze-order graphs. Figure 4.5 shows counter examples. Path-consistency using the composition table T_1, will determine that relation $\{<,g\}$ holds between vertices x,y in the first graph, where, by Lemma 4.2.1, the relation $<$ is entailed. Path-consisteucy using the composition table T_2, however, yields the correct result for the first graph but it fails to ensure minimality for a graph with a g^3-path such as the the lower graph in Figure 4.5. Path consistency with T_2 guarantees 5-consistency (any sub-network with 5 vertices is minimal)

[2]For instance, in the (exact) point algebra (PA) the condition of minimality is that each 4-node subgraph has to be consistent. In the PA without $\neq$ relations the size of the consistent subgraph is 3, etc. [vB92].

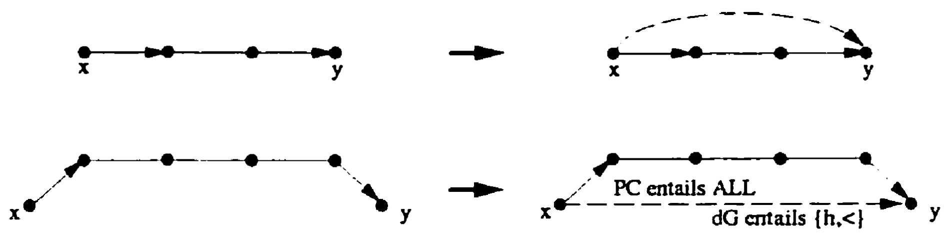


Figure 4.5: Minimal graphs

but not 6-consistency for this graph. An interesting observation that is revealed by this example is that the consistency condition in the presence of a haze relations, depends on the length of the g-paths in the constraint set. Another point is that by enriching the relation set with relations refining the qualitative scope of the original haze relation such as $\lhd$ and $\rhd$, we can increase the degree of consistency that the PC algorithm achieves. Naturally, this process has a limit which is the refinement of the approximate haze up to the exact equal and therefore the recovery of the $\{<, =, >\}$ relation set for which PC guarantees minimality (excluding the $\neq$ relations).

Proposition 4.2.4 *The path consistency property can not guarantee strongest relations in a haze-order graph.*

Proof. By the counter example of Figure 4.5. □

Theorem 4.2 *Any disambiguated haze-order graph entails minimal relations.*

Proof. We follow the same argument line as in [GS95a]. A minimal relation is the strongest possible relation entailed. We need to show that any stronger relation than the entailed by a haze-order graph is not feasible. We take the case of the haze-order graph with labels $\{<, g, >\}$, for which Lemma 4.2.1 establishes entailment. This entailment is strong. The lattice $\{\text{ALL} \rightarrow \{h, <\}, \{h, <\} \rightarrow h, \{h, <\} \rightarrow <\}$ represents all the "weaker to stronger" relation pairs. Let r be the relation on vertices x, y, as entailed by this lemma. Any solution for x, y that replaces r with a stronger relation, r', according to the above lattice will not be logically equivalent, i.e., will not satisfy all the models that a solution containing r satisfies. This is supported by a case analysis of the transitions and the checking of the quantitative expression that corresponds to each entailment as described in Lemma 4.2.1. A similar argument applies to labels $\{<, \lhd, g, \rhd, >\}$ and Lemma 4.2.2. □

The implication of Theorem 4.2 is that the minimal network representation for a haze-order graph can be constructed by simply computing the strongest entailed relation for each pair of nodes.

4.3 Efficient Algorithms for Qualitative Reasoning about Haze-Orders

4.3.1 Consistency

Theorem 4.1 suggests that a haze-order graph is consistent if and only if it does not contain any $<^i, g^j$-cycles with $i \geq j$. We now present an algorithm (referred to as **HOC** hereafter) which enforces this consistency detection criterion. The main idea is to reduce the haze-order graph to a graph that contains $<$-edges only and whose nodes are collections of nodes related by g-edges (hereafter we will refer to these nodes as *hyper-nodes*). Then, in a second phase, we detect cycles that satisfy the inconsistency criterion.

The algorithm operates on an adjacency list representation of the haze-order graph. Let $Alist(v)$ be the adjacency list that corresponds to node v. For every (v, g, u) edge, u appears in $Alist(v)$ and v appears in $Alist(u)$. For every $(v, <, u)$ edge, u appears in $Alist(v)$ but not the reverse.

A hyper-node is a node created as a result of collapsing nodes connected by g-edges. Each hyper node has a single "entry" point (node) and possibly many "exit" points. Each enclosed node, v, in hyper-node, h, has a value $d(v)$ denoting the distance of v from the entry point of h according to a breadth-first traversal. If v is an exit point then $d(v)$ is assigned as a label to the hyper-edge emanating from v. The set of nodes enclosed by a hyper-node h is denoted by $nodes(h)$.

The following notation is used in the consistency algorithm: $status(v)$ denotes the status of node v during the course of the consistency algorithm. Initially all nodes are marked as `closed`. If a node is reached by the algorithm, it is marked **open**, and when its processing is completed it is marked **done**. Function $in\_point(v)$ returns true if node v has an incoming $<$-edge, otherwise it returns false. Function $out\_point(v)$ returns true if there exists an outgoing $<$-edge from node v, otherwise returns false. *HyQueue* is a queue of potential hyper-nodes. *H_edgeQueue* is a queue for gathering boundary h-edges and $<$-edges that are to be be inserted in the hyper-graph. $queue(h)$ is a queue which stores the nodes that potentially become part of hyper-node h.

We can now outline the steps of the algorithm. A detailed description of the algorithm is given in Figure 4.6.

Step 1 (reduction) :

This step builds a graph. It starts from some initial node, v_0, and creates a hyper-node h with $queue(h) = Alist(v_0)$. Queued nodes are inserted in h if they are not *out_points* and connected to an enclosed node with a g-edge. *out_points* are queued in *HyQueue* since they are potential nuclei for subsequent hyper-nodes. Edges that lead to *out_points* are also queued for further processing. In particular, queued <-edges become edges of the hyper-graph. Queued g-edges are used to correct the d values. This step continues until all nodes of the haze-order graph are marked done, i.e., placed in some hyper-node.

Step 2 (detection) :

Let R be the reduced hyper-graph of hG. R is a directed graph. We run the strongly connected components (SCC) algorithm on R to detect cycles. For each cycle found, we test whether $\#edges \geq \Sigma_{i=1}^{\#edges} weight(edge_i)$ (1). If (1) holds the graph hG is inconsistent.

Theorem 4.3 *The* **HOC** *algorithm correctly detects inconsistency for any haze-order graph in $O(n+e)$ time, where n and e are the numbers of nodes and edges, respectively, in the haze-order graph.*

Proof. The correctness criterion for algorithm **HOC** is established in Theorem 4.1. The algorithm described in Figure 4.6 enforces exactly this criterion. First, it computes the maximum length of a g-paths that occur uninterrupted and encloses them in hyper-nodes. Hyper-nodes are linked by directed edges, thus forming a directed graph R. The **HOC** algorithm detects <-cycles on graph R by calling a procedure which finds the strongly connected components of R. At this step the algorithm may exit with a recommendation that the hG is consistent, if no cycles are found, or it enforces the consistency condition, otherwise. The running time of the algorithm is dominated by the graph building phase which is $O(n+e)$. The consistency detection phase takes $O(n'+e')$ time, where n' and e'

Algorithm: HOC
Input: A haze-order graph $hG = (V, E)$
Output: Success, if hG is consistent, failure otherwise.

```
Let R = (V', E') be the reduced graph, initially empty, i.e., V' := {}; E' := {}
for all nodes v ∈ V do d(v) := 0 ; status(v) :=closed end for
HyQueue := {} ; h_edgeQueue := {}
Let v := v0 be the starting node, status(v) :=open   //has to be either an in_ or out_ point
HyQueue := HyQueue ∪ {v}
while HyQueue ≠ ∅ do
    h := pop(HyQueue)
    Create hyper-node h with nodes(h) := v; queue(h) := Alist(v);
    V' := V' ∪ {h};
    status(v) :=open; d(v) := 0;
    while queue(h) ≠ ∅ do
        u := pop(queue(h))
        if in_point(u) then
            HyQueue := HyQueue ∪ {u}
            h_edgeQueue := h_edgeQueue ∪ {(h, l, u)}
        else if status(u) ≠ done then
            nodes(h) := nodes(h) ∪ {u}; status(u) := done
            for each w ∈ Alist(u) do
                if status(w) = closed then
                    status(w) :=open; d(w) := d(u) + 1; end if
                queue(h) := queue(h) ∪ {w}
            end for
        end if
    end while
end while
while h_edgeQueue ≠ ∅ do
    (v, l, u) := pop(h_edgeQueue)
    h_v, h_u are the hyper-nodes of v and u, respectively
    case l is
        g:  if h_u = next(h_v) then
                *correct* label for (h_v, h_u) hyper-edge
            else if h_u ≠ next(h_v) then
                E' = E' ∪ {(prev(h_v), -d(v) - 1, h_u)}
            else if |h_v| = |h_u| then merge(h_v, h_u) end if
        <:  E' = E' ∪ {(h_v, -d(v), h_u)}
    end case
end while
consistency := True
C = {c|c is cycle found while running SCC(R)}
while not (C = ∅ and consistency) do
    let c = (e', v') ∈ C; C := C - {c}
    if (|e'| ≥ Σ_i^{|e'|} weight(e_i)) then consistency := False end if
end while
return consistency
```

Figure 4.6: The haze-order consistency algorithm

are the number of nodes and edges of R. In effect, the overall time complexity of **HOC** is $O(n+e)$. □

4.3.2 Computing the "Tight" Graph

Having computed a consistent haze-order graph, the next step is to compute the tight graph, i.e., a haze-order graph with the most tight order relations [3]. The tight graph, T, underlying a haze-order graph, hG, has the following properties:

- all the disambiguations that are possible in hG, are made explicit in T;
- any order relation implied by hG, is derivable in T by a path traversal.

A key concern in the development of the algorithm for computing a tight-order graph, is to resolve ambiguities pertaining to $<^i, g^j$-cycles. An interesting situation arises when the $<^i, g^j$-cycles overlap.

The "tightening" algorithm operates on the directed graph R that is constructed during the reduction step of the consistency algorithm. A topological sort of R gives an initial ordering of the nodes which is then refined. The refinement step uses information about each node's connections. For each node v we maintain the following parameters: $next\_gtr(v)$, $prev\_lss(v)$, $next\_geq(v)$ and $prev\_leq(v)$, all with the obvious meaning. For each hyper-node we classify its enclosed nodes to those that are connected only with internal nodes, called the *inside* set, and those connected with nodes outside the hyper-node, the *frontier* set. From the connectivity of the frontier nodes to the outside ones we can deduce order information about them. For instance, if two nodes v and u of the same hyper-node h_1 are connected to the same node w of an adjacent hyper-node h_2, and it is $(v, <, w)$ and (u, g, w) then v precedes u; if it happens to be (v, g, u) this haze relationship disambiguates to a close-precedes ($\lhd$) relationship.

The refinement phase of the algorithm traverses the nodes of the haze-order graph twice. First, it walks the nodes starting from the nodes of the topologically last hyper-node going backwards. At each node, it determines the nodes pointing to it via a $next\_gtr$ link. All the g-neighbours of these nodes are examined and updated. In effect, all the implicit triangles of nodes involving at least one explicit g-edge, that is, the currently traversed, one

[3] For each vertex the immediate next and previous vertices are identified

of its predecessors and a g-neighbor of the later, are examined. The examination stage, determines whether the third node is *p_ordered*. A node is *p_ordered* if either its *next_gtr* link points to one of its original g-neighbors (i.e., a disambiguation has happened) or its *next_geq* link points beyond its *next_gtr* link. If the third node is not *p_ordered* then its *next_geq* link is directed to the first node. A similar traversal is done starting from the nodes of the topologically first hyper-node and going forward. This time the "previous" links are updated. At the end of this process we have information about the closest next and closest previous node for each each node v, as well as information about the possible next and possible previous node of v. To facilitate the presentation, we have presented a very high-level description of the algorithm. A detailed description of the "tightening" algorithm (**THO**) is presented in Appendix A.1. The next step is to create a representation of the tight-order graph which is suitable for query processing.

4.3.3 Computing the Query Graph

A query graph is used for answering queries. The query graph, Q, that corresponds to a tight-order graph, T, is a logically equivalent representation to T that supports constant-time retrievals. The query graph is computed using a ranking process similar to that of [GA89] and [GS93] which assigns a rank to each node, thus making query processing a lookup-and-compare operation.

In a haze-order graph the assignment of a single ranking value to each node will not produce the desirable result. For instance, due to the haze relations, there are cases where two points can be either before or after each other. The assignment of a single ranking value will not reflect this relationship. Although the disambiguation of haze relations was a major concern in the design of the "tightening" algorithm, such scenarios may still arise since it is always possible that some haze relationships have still remained. To handle such cases, the ranking process is designed to assign a low and a high value to each node.

The **H***aze* **I***ndex* algorithm computes a query graph. The algorithm receives as input a tight-order graph, i.e., a haze-order graph with previous (*prev_leq*, *prev_lss*) and next (*next_geq*, *next_gtr*) links computed for each node. The objective of the algorithm is to create a quantitative index of the tight-order graph, i.e., assign an interval value to each node of the graph such that the order relationship between any two nodes can be retrieved by comparing their respective index values. The algorithm trades space for time. In particular, it transforms the edges-based representation of the graph to an adjacency list representation. The $n \times n$ array option is rejected since the graph is expected to be sparse in most of the cases. Note that the tight-order graph is a multi-graph, i.e., two nodes may be connected by more than one edge. During the course of the transformation of the multi-graph to an adjacency list of a directed graph, we collapse all the edges from v to u in a single edge whose label is the composition of all single labels. The total cost of these preprocessing steps is $Q(n+e)$.

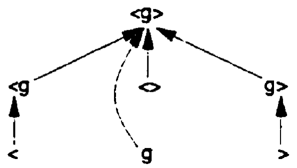
The main step of the algorithm assigns ranking information to the nodes. This step is essentially a depth first traversal algorithm starting from a special node called *start*. During the traversal each node is assigned a rank (the value of an incrementing counter) any time it is traversed. In effect, a node may have more than one rank. The minimum and maximum rank of the node define its index range. The value(s) assigned as rank of a node, depend(s) on the predecessor node rank and the type of the traversed edge. This step is also realized in $O(n+e)$ time. A detailed description of the **HI** algorithm is presented in appendix A.2.

4.4 Constraint Relation Algebras

In this section we introduce constraint relational algebras as an alternative formulation of the haze-order reasoning problem. The purpose of this formulation is twofold. First, it sets a framework that can accommodate and help in classifying various qualitative constraint languages based on haze relations. Second, it provides the background material for incorporating into our discussion a class of matrix-based sequential and parallel algorithms developed in the context of qualitative binary temporal constraint reasoning [LM94].

The $\uparrow$ operator is used in the definition of the *haze-order algebra*. Let B be a set of basic binary haze-order relations and $D = 2^B$ be the set of all their disjunctions. Members of D related by the *subsumes*, $\supset$, relation form a lattice with the universal relation as the top node. Given a member, a, of this lattice, the $\uparrow$ operator returns the relation of the next higher lattice level that subsumes a, i.e., $a \uparrow = b$ such that $b \supset a$.

Example. Let $B = \{<, g, >\}$, then $D = \{<, <g, g, g>, >, <>, <g>, \oslash\}$. For notational convenience, we write $<g$ to denote the relation $\{<, g\}$ which corresponds to the disjunction of $<$ and g basic relations. The *subsumes* lattice is



$<\uparrow$ is $<g$, $g\uparrow$ is $<g>$, $g>\uparrow$ is $<g>$ □

Definition 4.4.1 *A haze-order algebra, HO_B, over a set of basic haze-order relations, B, consists of an underlying set of all possible relations, D_B, that can hold between two haze points (usually $D_B = 2^B$), a unary inverse operator, $\smile$, and two binary operators for intersection, $\oplus$, and composition, $\otimes$, satisfying the following properties:*

1. *D is closed under the operations inverse, intersection and composition, i.e., $a^{\smile} \in D$, $a \oplus b \in D$, and $a \otimes b \in D$, for all $a, b \in D$;*

2. *intersection is associative, commutative, idempotent and has an identity element I, i.e., $a \oplus (b \oplus c) = (a \oplus b) \oplus c$, $a \oplus b = b \oplus a$, $a \oplus a = a$, and $a \oplus I = a$, for all $a, b, c \in D$;*

3. *composition is associative, commutative and has a quasi identity, Q, and an annihilator J, i.e., $a \otimes (b \otimes c) = (a \otimes b) \otimes c)$, $a \otimes b = b \otimes a$, $a \otimes Q = a\uparrow$, and $a \otimes J = J$, for all $a, b, c \in D$;*

4. *composition distributes over intersection provided that the intersection does not evaluate to the empty set, i.e., $a \otimes (b \oplus c) = a \otimes b \oplus a \otimes c$, and $(a \oplus b) \otimes c = a \otimes c \oplus b \otimes c$, for all $a, b, c \in D$.*

The elements of HO_B are constraints that hold between two haze points. Note that in a haze-order algebra, the composition operator has a quasi-identity, contrary to the composition identity of a regular relational algebra [vB90].

In the following we present specific haze-order algebras defined in the context of qualitative haze-order reasoning.

Basic Haze-Order Algebra HO_b:

$B_b = \{<, g, >\}$

$D_b = \{<,\ <g,\ g,\ g>,\ >,\ <>,\ <g>,\ \oslash\}$

$I = <g>$

$Q = g$

$J = \oslash$

Note that D_b is the set of all subsets of B ($D = 2^B$). The intersection and composition are defined as:

$$a \oplus b = a \cap b$$

$$a \otimes b = \cup_{i \in a, j \in b} T_1(i, j)$$

where $a, b \in D_b$ and T_1 is the topmost composition table in Figure 4.4. The inverse operation is defined as

op	$<$	$<g$	g	$g>$	$>$	$<>$	$<g>$	$\oslash$
$op^{\smile}$	$g>$	$>$	$<>$	$<$	$<g$	g	$\oslash$	$<g>$

Clearly, HO_b satisfies all the conditions of Definition 4.4.1.

Haze-Order Algebra with disambiguation , HO_d:

$B_d = \{<, \lhd, g, \rhd, >\}$

$D_d = \{<,\ <\lhd,\ <g,\ \lhd,\ <\lhd g,\ \lhd g,\ g,\ g\rhd,\ g\rhd>,\ \rhd,\ g>,\ \rhd>,\ >,\ <\rhd,\ \lhd>,$
$\lhd\rhd>,\ <\lhd\rhd>,\ <\lhd>,\ <\rhd>,\ <\lhd\rhd>,\ <\lhd g\rhd,\ \lhd g\rhd>,\ \lhd\rhd,\ <>,$
$<\lhd g>,\ <g\rhd>,\ <\lhd g\rhd>,\ \lhd g\rhd,\ <g>,\ \oslash\}$

$I = <\lhd g\rhd>$

$Q = g$

$J = \oslash$.

D_d contains all the subsets of B. However, many of the relations in D_d are synonyms of the same relation (e.g., $\lhd\rhd$ and $\lhd g\rhd$ are the same as g; $<g>, <\lhd g>, <g\rhd>, <\lhd\rhd>$ are the same as $<\lhd g\rhd>$ that we simply call ALL, etc.) By removing the symonyms in D_d, we obtain a subset D'_d which contains all the useful and qualitative distinct relations,

$$D'_d = \{<,\ <\lhd,\ <g,\ \lhd,\ \lhd g,\ g,\ g\rhd,\ \rhd,\ g>,\ \rhd>,\ >,\ <>,\ ALL, \oslash\}$$

The intersection and composition operations for HO_d are defined analogously to those of HO_b. Composition for HO_d uses the composition table T_2 (see Figure 4.4).

Haze-Order Algebra with multiple haze sizes , HO_N:

$B_N = \{<, g, >, N\}$, where N is a natural number,

$D_N = \{<,\ g,\ >,\ N,\ <g,\ <N,\ <>,\ gN,\ g>,\ N>,\ <gN,\ <N>,\ gN>,\ <g>,$
$<gN>, \oslash\}$

$I = \langle g \rangle$

$Q = g$

$J = \oslash$

The meaning of N is that it allows the representation of many haze relations whose size is a multiple of the size of g, e.g., $2g, 3g, \ldots$. Accordingly, $< N$ is the order relation corresponding to Ng, i.e., for some N the triple $\{< N, Ng, N >\}$ completely covers the one dimensional space. Some relations in D_N do not make sense in the physical space, e.g., N and $< N >$. By deleting these relations, we obtain

$$D'_N = \{<,\ g,\ >,\ < g,\ < N,\ <>,\ gN,\ g >,\ N >,\ < gN,\ gN >,\ < g >,\ \oslash\}.$$

Haze-Order Algebra with divisible haze , $HO_{N^{-1}}$:

$B_{N^{-1}} = \{<, g, >, N^{-1}\}$, where N is a natural number

$D_{N^{-1}} = \{<,\ g,\ >,\ < g,\ \frac{<}{N},\ <>,\ \frac{g}{N},\ g >,\ \frac{>}{N},\ \frac{<g}{N},\ < g >,\ \oslash\}$ [4]

$I = \langle g \rangle$

$Q = g$

$J = \oslash$

$HO_{N^{-1}}$ allows the representation of haze relations with size smaller than g. In this case, N^{-1} actually causes refinement of granularity g, where in HO_N, N causes coarsening of the granularity. In addition, $\frac{<}{1}$ is taken to be same as $\lhd$, $\frac{g}{2}$ a haze relation with half the size of g, and $\frac{<g}{2}$ is taken as the disjunction of the $<$ and the $\frac{g}{2}$ relations.

4.4.1 Haze-Order Reasoning and Relation Algebras

In the context of relation algebras, haze-order reasoning is expressed as a qualitative binary constraint reasoning problem represented by a binary constraint network.

A haze-order constraint network of size n (number of vertices) can be represented by an $n \times n$ matrix M over elements of a haze-order algebra HO_B. An $n \times n$ matrix M is path consistent if $M_{i,j} \subseteq M_{i,k} \otimes M_{k,j}$ for all i, k, j. Path consistency is a necessary condition for consistency in a constraint problem, but, in general, it does not imply consistency [LM94]. Instead, path consistency can be used as a pruning technique to narrow down the search for a solution to the constraint problem represented by M.

[4] Only the realizable relations are considered.

In the following we present a modified path consistency algorithm for haze-order constraints. As shown in Section 4.2, path consistency cannot compute a minimal haze-order graph (proposition 4.2.4).

The practical importance of this algorithm is that it can detect inconsistency of random haze-order networks. Ladkin and Reinefeld [LR92] use a path consistency algorithm as a pruning technique while searching for consistency of interval problems. Two remarkable results of their work are first that every random interval problem can be solved in reasonable time, given that the general problem is NP-complete, and second, path-consistency computation time dominates the overall computation time since inconsistency is almost guaranteed for large random networks. In our work, we are interested in studying the average time performance of inconsistency detection of the path consistency based algorithm, as well as studying the relative quality of the algorithm's output as we move along the hierarchy of the haze-order relation algebras defined in Section 4.4. The final objective is to establish an experimental test-bed (i.e., experiment parameters and criteria) that we can use for the experimental evaluation of the haze-order graph techniques described in Section 4.3.

Figure 4.7 presents the algorithmic scheme for path consistency of haze orders. Various versions of the same algorithmic scheme also appear in [Mac77], [Mei91], [vB90], etc. $\mathbf{PC2-HO_b}$ and $\mathbf{PC2-HO_N}$ are two derivatives of this scheme after substituting the $\oplus$ and $\otimes$ operations with the addition and the multiplication operation defined by the tables presented in Figures 4.8 and 4.9, respectively.

4.5 Experimental Evaluation

The algorithms presented in Sections 4.3 and 4.4 have been implemented and their performance has been experimentally evaluated. In this section we present an overview of their implementation and our experimental results.

4.5.1 Implementation of the Algorithms

The implementation of **PC2-HO** is based on a typical binary constraint networks implementation [All83], [VKvB89], [RCC92b]. The constraint network of size n is represented by an $n \times n$ array, each entry (i, j) of which represents a constraint relation between entities

Algorithm: PC2-Haze-Orders
Input: A matrix representation of a haze-order network, M
Output: "inconsistency" if M is inconsistent, or $M' \subseteq M$.

$Q := \{(i,k,j) | i < j, k \neq i,j\}$
while $Q \neq \oslash$ **do**
  select and delete any triplet (i,k,j) from Q
  if REVISE(i,k,j) **then**
    $Q := Q \cup \text{RELATED\_PATHS}(i,k,j)$
end while

REVISE(i,k,j)
$Z := M_{i,j} \oplus M_{i,k} \otimes M_{k,k} \otimes C_{k,j}$
if $Z = M_{i,k}$ **then** return **true**
  else return **false**
end

RELATED_PATHS(i,k,j)
return $\{(i,j,m)\} \cup \{(m,i,n)\} \cup \{(j,i,m)\} \cup \{(i,j,m)\}$

Figure 4.7: The path-consistency algorithm

i, j. The implemented algorithm follows the algorithmic scheme of Figure 4.7. The algorithm consults an addition and a multiplication table such as those presented in Figures 4.8 and 4.9. In fact, we have implemented two instances of this algorithm corresponding to haze-order algebras HO_b and HO_N, respectively.

The implementation of **HOC** algorithm is based on the algorithmic description of Figure 4.6. This part of our implementation uses heavily algorithmic techniques of graph theory [CLR91]. The implementation language for both algorithms is C.

4.5.2 Experimental Data

We ran the implemented algorithms using three classes of experimental data: randomly generated "consistent" haze-order graphs, randomly generated graphs, and, finally, test data from a real application.

The first category of test data refers to data that were created randomly and then filtered by the consistency algorithm so that only consistent sets were finally selected. According to the result of Ladkin and Reinefeld [LR92], randomly generated constraint satisfaction

xop_2z / xop_1y		$<$ 1	$<g$ 4	g 2	$g>$ 5	$>$ 3	$<>$ 7	$<g>$ 6	$\oslash$ 0
$<$	1	1	1	0	0	0	1	1	0
$<g$	4	1	4	2	2	0	1	4	0
g	2	0	2	2	2	0	0	2	0
$g>$	5	0	2	2	5	3	5	5	0
$>$	3	0	0	0	5	3	5	3	0
$<>$	7	1	1	0	3	3	7	7	0
$<g>$	6	1	4	2	5	3	7	6	0
$\oslash$	0	0	0	0	0	0	0	0	0

xop_2y / xop_1y		$<$ 1	$<g$ 4	g 2	$g>$ 5	$>$ 3	$<>$ 7	$<g>$ 6	$\oslash$ 0
$<$	1	1	4	2	6	6	6	6	6
$<g$	4	4	4	4	6	6	6	6	6
g	2	4	4	6	5	5	6	6	6
$g>$	5	6	6	5	6	5	6	6	6
$>$	3	6	6	5	3	5	6	6	6
$<>$	7	6	6	6	6	6	6	6	6
$<g>$	6	6	6	6	6	6	6	6	6
$\oslash$	0	6	6	6	6	6	6	6	6

Figure 4.8: The addition and multiplication tables for HO_b

x op1 y / y op2 y	< (1,0)	g (2,1)	Ng (2,N)	<g (4,1)	<Ng (4,N)	$<_N$ (1,N)	> (3,0)	g> (5,1)	gN> (5,N)	$N^>$ (3,N)	<g> (6,1)
< (1,0)	(1,0)	NIL	(1,0)	(1,0)	(1,N)	(1,N)	NIL	NIL	(1,0)	NIL	(1,0)
g (2,1)	NIL	(2,1)	(2,1)	(2,1)	(2,1)	NIL	NIL	(2,1)	(2,1)	NIL	(2,1)
Mg (2,M)	(1,0)	(2,1)	(2,min(N,M))	(4,1)	(2,min(N,M))	(1,N) if N>M NIL otherwise	(3,0)	(5,1)	(2,min(N,M))	(3,N) if N>M NIL otherwise	(2,M)
<g (4,1)	(1,0)	(2,1)	(4,1)	(4,1)	(4,1)	(1,N)	NIL	(2,1)	(4,1)	NIL	(4,1)
<Mg (4,M)	(1,M)	(2,1)	(2,min(N,M))	(4,1)	(4,min(N,M))	(1,N)	(3,0)	(2,M)	(2,min(N,M))	(3,N) if N>M NIL otherwise	(4,M)
$<_M$ (1,M)	(1,M)	NIL	(1,M) if M>N NIL otherwise	(1,M)	(1,M)	(1,max(N,M))	NIL	NIL	(1,M) if M>N NIL otherwise	NIL	(1,M)
> (3,0)	NIL	NIL	(3,0)	NIL	(3,0)	NIL	(3,0)	(3,0)	(3,0)	(3,N)	(3,0)
g> (5,1)	NIL	(2,1)	(5,1)	(2,1)	(2,N)	NIL	(3,0)	(5,1)	(5,1)	(3,N)	(5,1)
gM> (5,M)	(1,0)	(2,1)	(2,min(N,M))	(4,1)	(2,min(N,M))	(1,N) if N>M NIL otherwise	(3,0)	(5,1)	(5,min(N,M))	(3,N)	(5,N)
$M^>$ (3,M)	NIL	NIL	(3,M) if M>N NIL otherwise	NIL	(3,M) if M>N NIL otherwise	NIL	(3,M)	(3,M)	(3,N)	(3,max(N,M))	(3,M)
<g> (6,1)	(1,0)	(2,1)	(2,N)	(4,1)	(4,N)	(1,N)	(3,0)	(5,1)	(5,N)	(3,N)	(6,1)

x op1 y / y op2 z	< (1,0)	g (2,1)	Ng (2,N)	<g (4,1)	<Ng (4,N)	$<_N$ (1,N)	> (3,0)	g> (5,1)	gN> (5,N)	$N^>$ (3,N)	<g> (6,1)
< (1,0)	(1,0)	(4,1)	(4,N−1)	(4,1)	(4,N−1)	(4,N+1)	(6,1)	(6,1)	(6,1)	(6,1)	(6,1)
g (2,1)	(4,1)	(2,2)	(2,M+1)	(4,2)	(4,N+1)	(1,N−1)	(5,1)	(5,2)	(5,N+1)	(3,N−1)	(6,1)
Mg (2,N)	(4,M−1)	(2,M+1)	(2,N+M)	(4,M+1)	(4,N+M)	Exp1	(5,M−1)	(5,M+1)	(5,N+M)	Exp2	(6,1)
<g (4,1)	(4,1)	(4,2)	(4,N+1)	(4,2)	(4,N+1)	(1,N−1)	(6,1)	(6,1)	(6,1)	(6,1)	(6,1)
<Mg (4,N)	(4,M−1)	(4,M+1)	(4,N+1)	(4,M+1)	(4,N+M)	Exp1	(6,1)	(6,1)	(6,1)	(6,1)	(6,1)
$<_M$ (1,N)	(1,M+1)	(1,M−1)	Exp3	(1,N−1)	Exp3	(1,N+M)	(6,1)	(6,1)	(6,1)	(6,1)	(6,1)
> (3,0)	(6,1)	(5,1)	(5,N−1)	(6,1)	(6,1)	(6,1)	(3,0)	(5,1)	(5,N−1)	(3,N+1)	(6,1)
g> (5,1)	(6,1)	(5,2)	(5,N+1)	(6,1)	(6,1)	(6,1)	(5,1)	(5,2)	(5,N+1)	(3,N−1)	(6,1)
gM> (5,N)	(6,1)	(5,M+1)	(5,N+1)	(6,1)	(6,1)	(6,1)	(5,M−1)	(5,M+1)	(5,N+1)	Exp2	(6,1)
$M^>$ (3,N)	(6,1)	(5,M+1)	Exp4	(6,1)	(6,1)	(6,1)	(3,M+1)	(3,M−1)	Exp4	(3,N+M)	(6,1)
<g> (6,1)	(6,1)	(6,1)	(6,1)	(6,1)	(6,1)	(6,1)	(6,1)	(6,1)	(6,1)	(6,1)	(6,1)

Exp1 = (1,N−M) if N−M>0, (1,M−N) otherwise

Exp2 = (3,N−M) if N−M>0, (5,M−N) otherwise

Exp3 = (4,N−M) if N−M>0, (1,0) otherwise

Exp4 = (3,M−N) if N−M>0, (5,N−M) otherwise

Figure 4.9: The addition and multiplication tables for HO_N

Figure 4.10: Fragment of a genetic sequence (contig-369)

problems beyond a certain size are almost guaranteed to be inconsistent. This data set has been preprocessed in order to eliminate inconsistencies, since our purpose is to use the data for measuring the relative performance of the two different algorithms.

The second category of data is used for comparing the two algorithms with respect to their running time until they detect inconsistency. This dataset is program generated and consists of haze-order graphs with randomly generated connectivity. Each graph node has a fanout degree k in the range of 3 to 6 and each edge is labeled by either a haze or order relation with probabilities q and $1 - q$, respectively.

The third set of data is taken from the human genome project [Fre91] and in particular they have been retrieved using the AceDB system [TMD92]. This dataset originally contained ordering information which we have fuzzed thus creating haze-order relations. Figure 4.10 illustrates a fragment of the sequence of genetic intervals that has been used to create this dataset [5]. The genetic intervals shown in Figure 4.10 have been identified by biological experiments of limited precision. Particular problems of imprecision in genome data and possible solutions using our formalism are discussed in Chapter 6. Here, we only use genome data as suitable one-dimensional haze-order data for testing the variants of **PC2-HO** algorithms. A preprocessing transformation has been applied to these data which turned genetic intervals to haze points related by haze (overlapping intervals) and order relations.

4.5.3 Results

The graphs of Figure 4.11 illustrate the running time of the HOC algorithm as a function of the input size for program generated haze-order graphs. The graph on the left in Figure 4.11 displays HOC's running time in contrast to the *nlogn* line while the graph on the

[5]The genetic intervals (called YACs) depicted in Figure 4.10 belong to the contig-369 [DOE92].

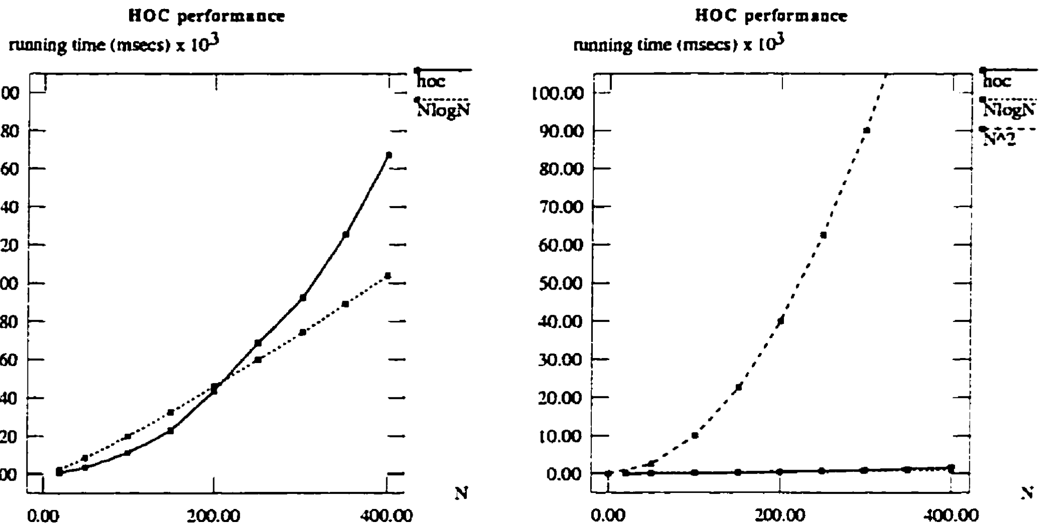


Figure 4.11: Performance of the Haze-Order Consistency algorithm

right in Figure 4.11 displays HOC's cost contrasted with the n^2 line. n^2 is the worst case complexity bound of HOC's for the case in which the input is a complete haze-order graph. The input haze-order graphs used in this experiment where sparse graphs. Each graph of Figure 4.11 summarizes a 400 runs of the HOC algorithm.

The graphs of Figure 4.12 show the dependence of HOC's performance on the type input relations, i.e., the labels of haze-graph edges. Two experiments were conducted to study this dependence. In the first experiment we varied the number of order relations/labels from 10% to 90% of the total number of edges for fixed size haze-order graphs. The results are illustrated by the graph on the left of Figure 4.12 and show a peak in the cost of HOC when the number of order relations (*#orders*) is equal to the number of haze relations (*#hazes*). When *#orders* > *#hazes*, the cost of HOC is a little higher. The explanation of this is that *#orders* controls the number of hyper-nodes that the algorithm creates, and that the detection part depends on this number. The second experiment plots the cost of HOC for various *#orders/#hazes* ratios as a function of the size of the graph. The finding in this experiment is that the cost increase with respect to N is sharper for bigger *#orders/#hazes*, exactly for the same reasons as in the previous experiment. The results are shown on the graph on the right in Figure 4.12.

The third series of experiments compares the cost of HOC and PC-HO-b algorithms. This part uses test data of the first and the second category. The graph on the left in

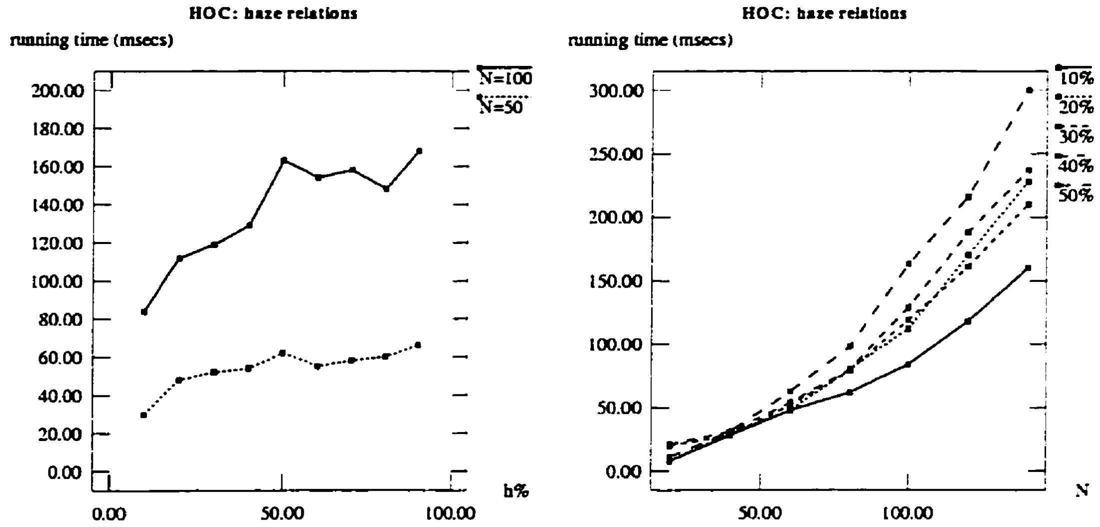


Figure 4.12: Dependence of HOC's performance on the relations' type

Figure 4.13 displays the relative cost of HOC and PC-HO-b for large and "consistent" data sets. As the data set is relatively sparse, the obtained running times for both algorithms are far from their upper complexity bound. HOC outperforms PC-H0-b, since it takes advantage of the structure of the input data. However, when the test data are random, then by a high probability they contain an inconsistency. In this case PC-HO-b is expected to detect it earlier than HOC. This is because HOC undergoes a preprocessing phase before its consistency detection phase. In our experiments we have modified PC-HO-b in two ways: (a) stop after the first inconsistency found, and (b) avoid updating the constraint network with relations that cause inconsistency so that it can finally process the entire input data set. These two variants are displayed on the graph on the right in 4.13 as PC-inconsistency and PC-all. The same modifications have also been made for HOC (denoted by HOC-inconsistency and HOC-all). In addition, we have measured the HOC's preprocessing cost. The results show that the preprocessing phase dominates the cost of HOC and also verified our intuition that if random input is used, then PC-inconsistency ceases very early.

The fourth series of experiments studies the differences between the variants of PC-HO algorithms. These experiments use the third dataset. The input for these experiments is a number of order (*#orders*) and haze (*#hazes*) relations on fixed sized haze-order graph (fragment of contig-369). The goal is to inspect the relations sitting on the edges of the constraint network when PC-HO-* stabilize. These relations are classified in the following types:

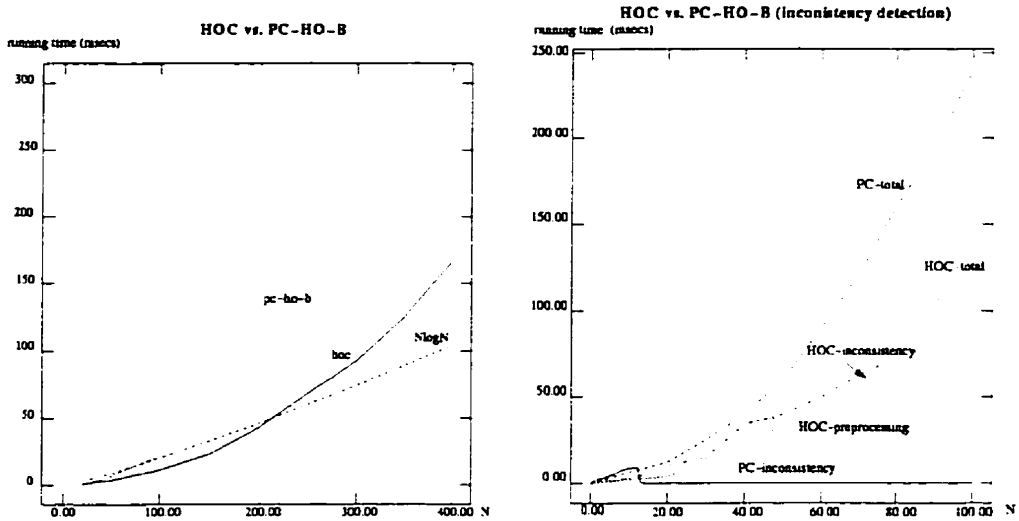


Figure 4.13: Comparison between HOC and PC-HO algorithms

ORD=$\{<, <_N, >, >_N\}$, HAZ=$\{g, Ng\}$, HLF=$\{< g, g >, < gN, gN >\}$ and NOR=$\{ALL\}$. The top two tables in Figure 4.14 show the results. Figures 4.15 and 4.15 (left) show the plots and a bar chart derived from these tables. Finally, the table at the bottom of Figure 4.14 presents the resulting output relations before and after the application of a preprocessing phase which enforces the rule described in Figure 4.3. This step attempts to increase the order information that is used as input by the PC-HO algorithms. The success of the preprocessing step is shown on the Gain column of the same table. The bar chart on the right of Figure 4.15 displays the results. As the results show, PC-HO-N succeeds in transforming many NOR constraints to either HAZ of HLF constraints. PC-HO-N becomes more successful when the number of input order relations increases.

4.6 Conclusions

We have studied the problem of qualitative spatial reasoning in one-dimensional haze space. Haze-space is a qualitative representation of space where entities are related in terms of an indistinguishability and a precedence relation.

The main contribution of this work is the definition of a computational model for reasoning about haze-orders. We have defined a data structure, called a haze-order graph, which is used to represent constraints on haze-space entities. We have addressed two reasoning questions, namely, the consistency of a set of haze-order constraints, and the strongest re-

PC-HO-b (No preprocessing)

Input		Output			
# haze	# order	ORD	HAZ	HLF	NOR
152	95	432	470	2202	8560
152	141	728	498	3200	7672
152	220	1784	812	5226	3842
152	241	2350	844	5110	3360
152	239	1262	588	4446	5366
152	323	2512	860	5050	3242

PC-HO-N (No preprocessing)

Input		Output			
# haze	# order	ORD	HAZ	HLF	NOR
152	98	378	3041	7041	1204
152	180	1146	4460	5916	142
152	231	1765	2599	6590	710
152	303	2308	1016	7049	1291
152	409	2902	1692	6492	654
152	531	3732	3109	4693	130

Experiment			Output			
Algorithm	Prepr.	Gain (%)	ORD	HAZ	HLF	NOR
PC-HO-b	No		1454	540	4432	5238
PC-HO-b	Yes	8%	1524	502	4154	5484
PC-HO-N	No		1200	3597	6361	506
PC-HO-N	Yes	8%	1324	1723	7467	1150

Figure 4.14: Experimental results from the PC-HO-b and PC-HO-N algorithms

lation computation between any two entities. We have developed polynomial algorithms for both problems, and in addition we have defined a quantitative index structure that can support constant time retrievals. Another interesting finding of our study is the proof that the classical path consistency algorithm can not be used for computing the strongest relation between any two entities.

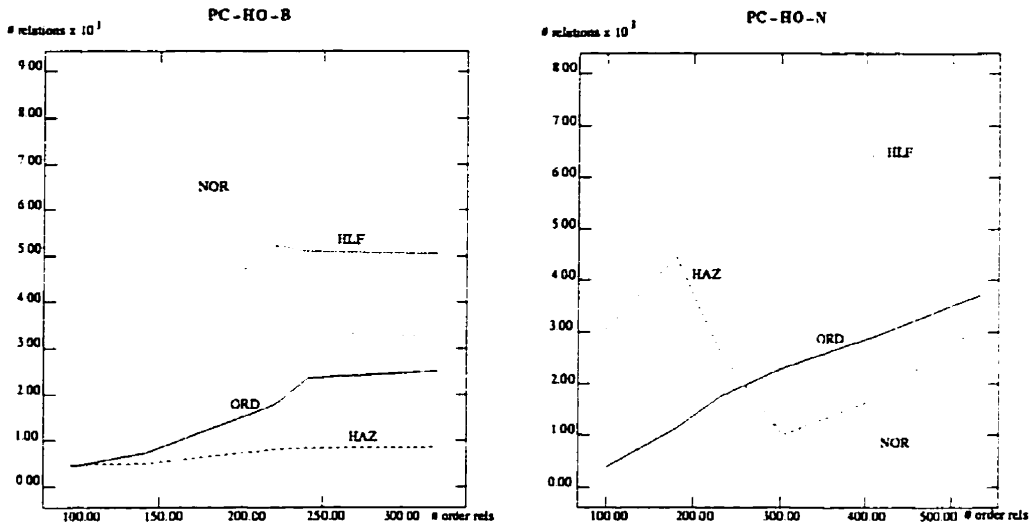


Figure 4.15: Relations distribution in the PC-HO-* output

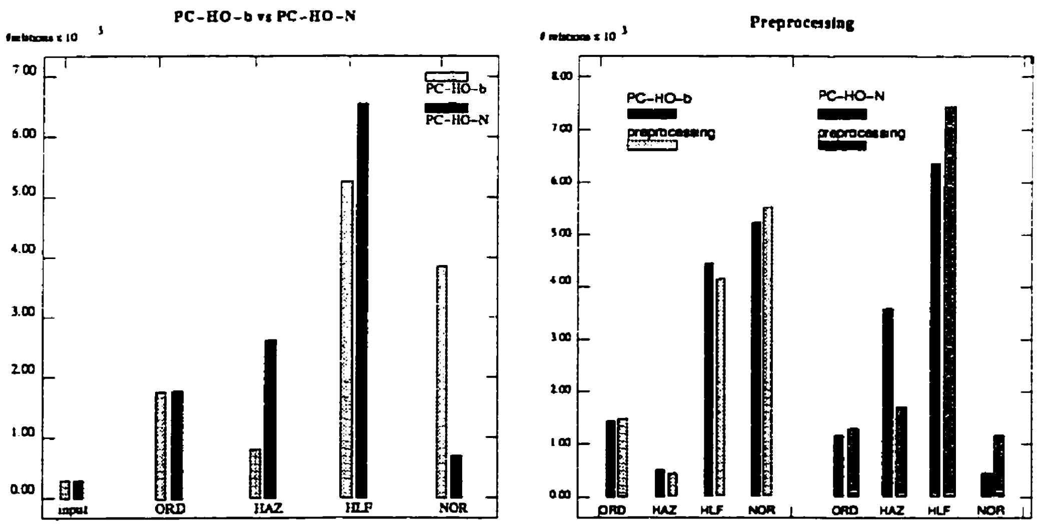


Figure 4.16: Relations "quality" in the PC-HO-* algorithms output

Chapter 5

Modeling Spatial Imprecision in Databases

5.1 Introduction

This chapter proposes a novel spatial data model which facilitates representation and reasoning with various forms of qualitatively and quantitatively incomplete spatial information. [1]

As mentioned in Chapters 1 and 2, existing spatial representations for databases can generally be classified into two broad categories. The first category includes models that focus on explicit representations of space usually in terms of a *quantitative* formalism such as a map or a digitized array. We call these models *complete* since they represent the entire modeling space (see for example, [OM88], [RFS88], [GS95b]). The second category includes models that focus on the representation of spatial features that are essential and are expressed in terms of a *qualitative* formalism such as symbolic arrays [GP92] or spatial relations [Her92]. The models are capable of reasoning about *partial* spatial information but, by-and-large, ignore quantitative spatial information and performance concerns.

The proposed spatial data model accommodates both qualitatively and quantitatively partial spatial information. The expressiveness of the model is further enhanced by the provision of facilities for dealing with granularity and scale within a single framework. The formal tools employed in the development of the data model include a conceptual

[1] The contents of Chapter 5 have appeared in [Top96b].

modeling language, where the features of the proposed spatial data model are embedded, and a constraint-based language that is suitable for representing partially specified spatial information.

The rest of this chapter is organized as follows. Section 5.2, includes an informal presentation of the features of the proposed spatial data model using the running example of Section 1.2. Section 5.3, integrates the proposed features with a conceptual modeling language, while Section 5.4 presents a formalization of the data model. Finally, Section 5.5, presents a summary of the work described in the chapter.

5.2 Granularity and Haze in Spatial Representations

This section first introduces the notions of scale and grain as basic concepts emerging from the example of Section 1.2 and then presents two fundamental constructs of the proposed spatial data model: the *spatial envelope* and the *map* structure.

5.2.1 Basic Concepts

A *spatial object* is a symbol structure representing a point or region of space.

Spatial Object Types. As suggested by the example of Section 1.2, spatial objects can be either *landmarks*, in which case there is complete information about the point or region of space being represented, or *indeterminate* objects (or indeterminates, for short), for which there is only partial information about the point or region being represented. Indeterminates are related to landmarks through constraints expressed in a qualitative constraint language.

Scale. In a representation space, a scale is a system of ordered marks used as a reference standard in determining the relationships between representations. For metric domains, a scale is defined as an ascending set of point values which differ by a fixed interval, called *grain* (or *unit*), denoting distance from a fixed constant of the system. In qualitative domains a scale is defined as a fixed order between landmarks which may differ by a variable size qualitative interval.

Scale Hierarchy. A scale hierarchy is an ordered set of scales, $S = \{s_1, s_2, \ldots, s_n\}$, such that for each spatial object α at scale s_i there exists a container object $container(\alpha)$ at scale s_{i-1} that contains α, i.e., $inside(\alpha, container(\alpha))$, for $i = 2 \ldots n$. The existence of a unique container requires that the scales are not overlapping and that scale s_{i-1} is "coarser" than

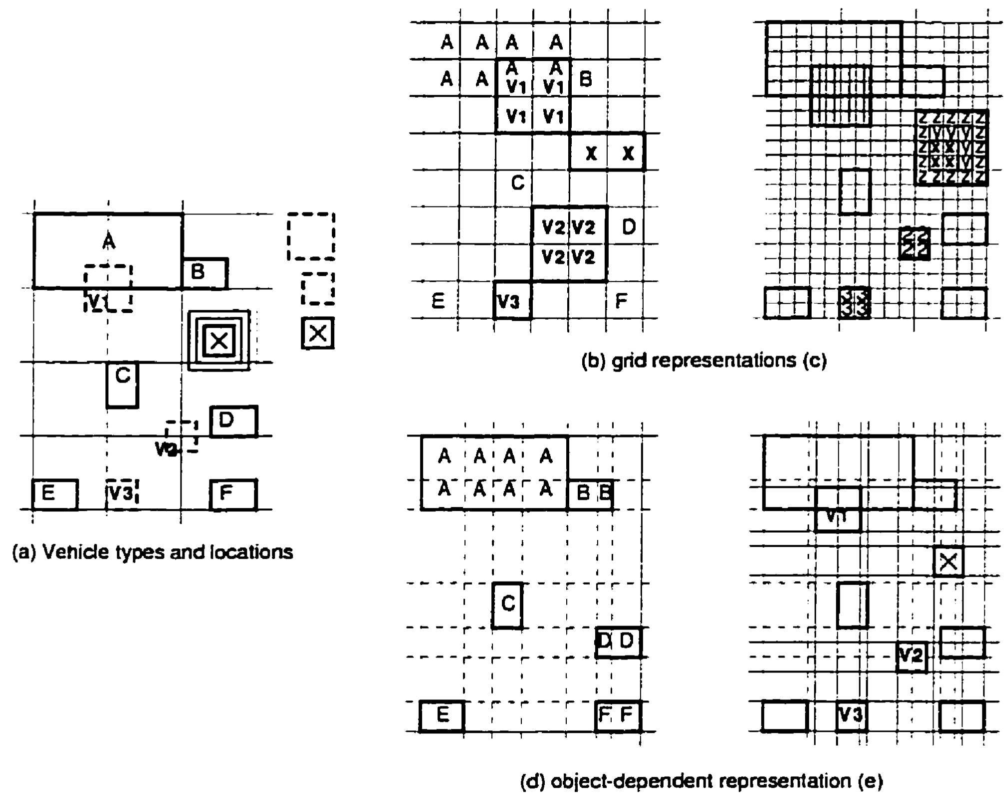


Figure 5.1: Representation of landmark and indeterminate spatial objects

s_i. The coarser relation is a total order. The ordered set $\{city, section, division\}$ of the above example, defines such a scale hierarchy.

Haze. Haze is a region which contains an indeterminate spatial object (point or region) and specifies the degree of indeterminacy associated with the object. In the example, assume that the position of V_2 is specified as "at University Ave. and Queen St." where V_1 is said to be "near College and St. George St.". In effect, V_2's position is given more precisely than that of V_1. Consequently, the haze size for V_1 is greater than the haze size for V_2.

5.2.2 Spatial Constraints

Constraints have been shown to be very useful in representing qualitative and quantitative temporal information [vB90], [KL91], [Kou94b]. This section develops a particular class of *spatial constraints*, which provide a convenient syntactic facility for expressing partial and relative information about spatial objects.

Spatial constraints are conjunctions of atomic formulas expressed in a constraint language $\mathcal{L}$. Section 5.4 presents such a language, namely, the language of qualitative and quantitative constraints on indeterminates and landmarks in one or two-dimensional space. In section 5.4, we also define a set of higher level topological and directional spatial relations which can serve as basic vocabulary for the constraint language. Here are some examples:

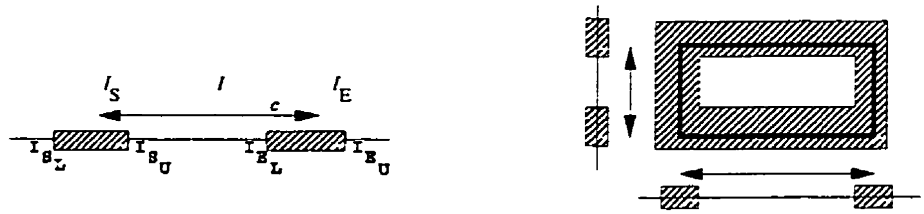
- $close(V2,$ "University Ave. and Queen St.")
- $south\_side(V1,$ "UofT") $\wedge\, east\_of(V1,$ "Spadina Ave.")

The discussion in the rest of this section is restricted to constraints on a single dimension, that are conjunctions of the following two types of atomic constraints: $x\ R\ c$, and $x - y\ R\ c$; where x, y are variables representing indeterminates, c is a constant representing a landmark, and R is one of $\{=, <, \leq\}$.

5.2.3 Spatial Envelopes

Spatial constraints can be used to define arbitrary types of spatial indeterminacy. *Spatial envelopes* provide a convenient mechanism for defining a useful and often-occurring type of spatial indeterminacy. In particular, spatial envelopes constrain an indeterminate spatial object to fall within a region. If x is an indeterminate spatial object, its spatial envelope is denoted by $env(x)$. The spatial envelope of a two-dimensional object is a rectangle characterized by two one-dimensional envelopes. A one-dimensional envelope constrains the exact position of an indeterminate spatial interval.

Indeterminacy in one-dimension is formulated as follows. Let I be an indeterminate interval whose (partly known) start and end points are denoted respectively by I_S and I_E. The length of I is denoted by some constant c, defined with respect to the scale of the metric domain of I. The envelope interval of I, $env(I)$, is a pair of point envelopes, $env(I_S)$, $env(I_E)$ (see Figure 5.2 for a graphical illustration). In a discrete domain, each point envelope, $env(P)$, is represented by two points, $\mathrm{P_L}$ and $\mathrm{P_U}$ which impose lower and upper bounds on the position of point P. Assuming that the size of the haze is g, then points $\mathrm{P_L}$ and $\mathrm{P_U}$ are related by the constraint $0 \leq \mathrm{P_U} - \mathrm{P_L} \leq 2g$, and the envelope definition $\mathrm{P_L} \leq P \leq \mathrm{P_U}$ for some point P. For interval envelopes, the length of the interval poses an additional metric constraint, i.e., $I_E = I_S + c$. As a result, the one-dimensional *interval envelope*, $env(I)$, can be characterized by four variables, $\mathrm{I_{S_L}}, \mathrm{I_{S_U}}, \mathrm{I_{E_L}}$, and $\mathrm{I_{E_U}}$, related by the following constraints:



one–dimensional envelope

two–dimensional envelope

Figure 5.2: Spatial envelopes

$$0 \le \mathrm{I}_{\mathrm{S_U}} - \mathrm{I}_{\mathrm{S_L}} \le 2g,$$
$$0 \le \mathrm{I}_{\mathrm{E_U}} - \mathrm{I}_{\mathrm{E_L}} \le 2g,$$
$$\mathrm{I}_{\mathrm{S_L}} \le I_S \le \mathrm{I}_{\mathrm{S_U}},$$
$$\mathrm{I}_{\mathrm{E_L}} \le I_E \le \mathrm{I}_{\mathrm{E_U}},$$
$$I_E - I_S = c$$

An envelope provides a convenient way for representing indeterminates after compiling-in their indeterminacy: a spatial object whose location is partially known is enclosed inside envelope parameters.

5.2.4 The Map Structure

The *map* structure is a logical data structure used to define a collection of spatial objects and their inter-relationships. Formally, a map structure m is a quintuple

$$m = \langle\ L,\ I,\ C,\ s,\ g\ \rangle$$

where L is a finite set of landmarks, I is a finite set of indeterminate spatial objects, C is a set of spatial constraints and s and g are its scale and grain, respectively. As is often the case, the same block of space may be represented by multiple maps of variable granularity. For example, Figure 5.1 shows several maps at various grains of scale *division* representing the same block of space. Specifically, $L = \{\mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{D}, \mathtt{E}, \mathtt{F}\}$, $I = \{\mathtt{V}_1, \mathtt{V}_2, \mathtt{V}_3, \mathtt{X}\}$ and $m_1 = \langle L, I, C, \textit{division}, g_{m_1}\rangle$, $m_2 = \langle L, I, C, \textit{division}, g_{m_2}\rangle$, $m_3 = \langle L, I, C, \textit{division}, g_{m_3}\rangle$ are maps corresponding to Figures 5.1(b), (c) and (e), respectively. The set of constraints, C, is explained below.

Every map definition must be such that its grain size can accommodate the haze size of its indeterminates. More formally, if *haze* is a function returning the haze size of an indeterminate and g_m is the grain size of map m, then this condition is written as $g_m \leq min\{haze(i)|i \in I\}$. The map corresponding to Figure 5.1(c) satisfies this condition since $haze(\mathtt{V}_1) = h_1$, $haze(\mathtt{V}_2) = haze(\mathtt{V}_3) = h_2$, $haze(\mathtt{X}) = h_x$, $g_{m_2} < h_2 < h_1 = h_x$. The map shown on Figure 5.1(b) is an example of a map which violates this condition.

The construction of a map involves two phases: First, the set of landmarks of the map are represented, followed by a definition of the map indeterminates. For the first phase, we assume that the input is a "segmented image", e.g., an $n \times m$ occupancy array, containing a set of landmarks, L. The grain, g, for the map is captured by the size of the array cells. A set of X and Y constraints can then be defined so that each one contains all the known order and distance relationships between landmarks along the X and the Y axes, respectively. Indeterminacy can now be circumscribed for map indeterminates through spatial envelope constraints.

Example 5.2.2.1 This example shows a constraint representation created for the static part of Figure 1.1(d). As indicated earlier, landmarks are approximated by their minimum bounding rectangle; we therefore need four parameters for representing them, namely, $A_{S_X}, A_{E_X}, A_{S_Y}$ and A_{E_Y}. Let g_m be the grain of the map. Then, with the help of an appropriate "cutting" function, we construct the array representation, say this is the array of Figure 5.1(c). The following two sets of equality constraints define X and Y axis projections of this array:

$$\begin{aligned}
Xcon : \{ & E_{S_X}=0, A_{S_X}=0, E_{E_X}=3, C_{S_X}=5,\ C_{E_X}=7,\ A_{E_X}=9, \\
& B_{S_X}=9, F_{S_X}=11, D_{S_X}=11,\ B_{E_X}=12, D_{E_X}=15, F_{E_X}=15\} \\
Ycon : \{ & E_{S_Y}=0, F_{S_Y}=0, E_{E_Y}=2, F_{E_Y}=2,\ D_{S_Y}=5, D_{E_Y}=7, \\
& C_{S_Y}=7, C_{E_Y}=10, A_{S_Y}=15,\ B_{S_Y}=15, B_{E_Y}=17, A_{E_Y}=20\}
\end{aligned}$$

□

Indeterminates are introduced next. According to the earlier discussion, a two-dimensional indeterminate is represented by a spatial envelope which consists of two coordinate one-dimensional envelopes. Each one-dimensional interval (resp. point) envelope is specified by four (resp. two) parameters, which are related by a fixed set of constraints as presented in section 5.2.3. The notion of envelope parameters used here (in `courier` font) is similar to Koubarakis' *e(xistential)-variables* [Kou94a], intended to represent values which are not completely known but for which a global constraint exists. The indeterminate parameters (in *italics* style) are also *e-variables* since they name a specific indeterminate and their possible values are bounded by envelope constraints.

Example 5.2.2.2 The insertion of a rectangular indeterminate into the map of Figure 5.1(b) (see Figure 6.1 for a magnified view) is demonstrated next: Let $V1 = \langle V1X_S, V1X_E, V1Y_S, V1Y_E \rangle$, $grain(V1) = g_1$, $size(V1) = c_1$; c_1 is a constant that denotes the size of $V1$'s scope. As for the discussion of section 5.2.3, the following constraints are introduced:

1.	$0 \leq \mathtt{V1X}_{\mathtt{S_U}} - \mathtt{V1X}_{\mathtt{S_L}} \leq 2g_1$	6.	$0 \leq \mathtt{V1Y}_{\mathtt{S_U}} - \mathtt{V1Y}_{\mathtt{S_L}} \leq 2g_1$
2.	$0 \leq \mathtt{V1X}_{\mathtt{E_U}} - \mathtt{V1X}_{\mathtt{E_L}} \leq 2g_1$	7.	$0 \leq \mathtt{V1Y}_{\mathtt{E_U}} - \mathtt{V1Y}_{\mathtt{E_L}} \leq 2g_1$
3.	$\mathtt{V1X}_{\mathtt{S_L}} \leq V1X_S \leq \mathtt{V1X}_{\mathtt{S_U}}$	8.	$\mathtt{V1Y}_{\mathtt{S_L}} \leq V1Y_S \leq \mathtt{V1X}_{\mathtt{S_U}}$
4.	$\mathtt{V1X}_{\mathtt{E_L}} \leq V1X_E \leq \mathtt{V1X}_{\mathtt{E_U}}$	9.	$\mathtt{V1Y}_{\mathtt{E_L}} \leq V1Y_E \leq \mathtt{V1X}_{\mathtt{E_U}}$
5.	$V1X_E - V1X_S = c_1$	10.	$V1Y_E - V1Y_S = c_1$

In addition, the position of $V1$ in the representation space is specified by constraints 11-14 (on the envelope parameters):

11.	$(E_{E_X} = \mathtt{V1X}_{\mathtt{S_L}}) \wedge (\mathtt{V1X}_{\mathtt{S_U}} < C_{S_X})$,	13.	$(C_{S_X} < \mathtt{V1X}_{\mathtt{E_L}}) \wedge (\mathtt{V1X}_{\mathtt{E_U}} = C_{E_X})$
12.	$(C_{E_Y} < \mathtt{V1Y}_{\mathtt{S_L}}) \wedge (\mathtt{V1Y}_{\mathtt{S_U}} < A_{S_Y})$,	14.	$(B_{E_Y} = \mathtt{V1Y}_{\mathtt{E_L}}) \wedge (\mathtt{V1Y}_{\mathtt{E_U}} < A_{E_Y})$

Constraints 1, 2, 11, and 13 are integrated into the *Xcon* constraint set of the earlier map and similarly, constraints 6, 7, 12 and 14 are integrated into its *Ycon* constraint set. Constraints 3, 4, 5 and 8, 9, 10 are local to object $V1$. □

5.3 Modeling Space in Telos

This section describes the integration of the proposed features for representing spatial information with the conceptual modeling language Telos [MBJK90]. This integration endows the resulting spatial data model with abstraction mechanisms such as generalization, classification and attribution, inherited from Telos, as well as facilities for expressing meta-concepts and for asserting constraints and rules.

Integration of spatial modeling facilities into Telos is accomplished through a library of meta-classes and meta-attributes that capture the semantics of the features presented in the previous section. The central class of the model is the `Map` class. Spatial information is attached to physical objects through a spatial object which participates in one or more maps.

It is assumed that the world being modeled includes, among other things, physical objects, which might have a temporal and a spatial aspect [Hay85]. The class `PhysicalObjectClass` is a metaclass whose instances include physical object classes such as the class of vehicles, and the class of buildings or parks. All these classes are also specializations of `PhysicalObject`, which is also an instance of `PhysicalObjectClass`. In addition, we introduce the metaclass `SpatialObjectClass`, whose instances are spatial object classes such as `Street`, `Lot`, `Parcel` etc. These classes are also specializations of `SpatialObject`, which is an instance of `SpatialObjectClass`. Figure 5.3 illustrates this class hierarchy. Physical objects can have an associated spatial object, about which information is represented in terms of one or more maps.

```
CLASS PhysicalObjectClass IN M1_CLASS
  WITH
    necessary,single
      when:TemporalClass
      where:SpatialClass
      what:OrdinaryClass
    attribute
      feature:AttributeClass
      time-feature:TemporalAttrClass
      space-feature:SpatialAttrClass
END PhysicalObjectClass

CLASS PhysicalObject IN PhysicalObjectClass
  WITH
    where
      place:SpatialObject
END PhysicalObject

CLASS SpatialObject IN SpatialObjectClass
  WITH
    attribute
       in-space:SpatialObject
       in-map:Map
END SpatialObject

CLASS UofT-Lot IN SpatialObject
  WITH
    in-space
       s1:division-City-Toronto-Parcel
    in-map
       m1:map1;
       m2:map2;
       m3:map3
```

```
END UofT-Lot
```

According to these definitions, the place attribute of PhysicalObject is declared to be an instance of the where meta-attribute of PhysicalObjectClass. A spatial object has an in-space attribute, which provides a spatial context, and zero or more associated maps that give information about the object. The next two definitions introduce different classes of spatial relationships. [2]

```
CLASS In-Map IN SpatialAttributeClass
  WITH
    components
       from:SpatialObject
       label:in-map
       to:Map
    attribute
       rtype:RepresentationType
END In-Map

CLASS In-Space IN SpatialAttributeClass
  WITH
    components
       from:SpatialObject
       label:in-space
       to:SpatialObject
    attribute
       stype:SpatialType
END In-Space
```

According to these definitions, in-map attributes have an associated attribute which specifies the representation type (landmark or indeterminate) of an object in a map. [3] Note that the same spatial object may have different types in different maps, i.e., be a landmark in one map and an indeterminate in another. Likewise, in-space associates a spatial type (region, point,...) to every spatial object / spatial context association. Again, the definition implies that a spatial object may have different types in different contexts.

A spatial object that serves as context for another object is itself described in terms of one or more maps. For example, the spatial token UofT-Lot is part of a division of the city of Toronto parcel, another spatial object, and participates in maps map1, map2 and map3 through relationships m1, m2 and m3. The two types of spatial object types supported in

[2] Telos does not have a built-in distinction between attributes and relationships.

[3] In Telos, all relationships are represented by a three-tuple, (from, label, to), which is called *proposition*. Intuitively, a proposition can be thought of as a link.

our model are declared as instances of RepresentationType. Analogously, the geometric types of objects (i.e., point vs. region) are defined as instances of SpatialType.

```
CLASS Map IN SpatialObjectClass
  WITH
    single,necessary
       grain: Grain
       scale: Scale
    attribute
       X-constraint: ConstraintSet
       Y-constraint: ConstraintSet
    deductiveRule
       R1:indeterminate(ThisClassInstance,X):-
                instance(X,SpatialObject), instance(Y,In-map),
                proposition(X,Y,ThisClassInstance), rtype(Y,Indeterminate).
       R2:landmark(ThisClassInstance,X):-
                instance(X,SpatialObject), instance(Y,In-map),
                proposition(X,Y,ThisClassInstance), rtype(Y,Landmark).
    integrityConstraint
       :indeterminate(ThisClassInstance,X) and X.haze > ThisClassInstance.grain
END Map
```

The Map class models the principal data structure of our spatial data model. A map is characterized by its grain and scale attributes and the set of objects it inter-relates. X- and Y-constraint attributes take as values object inter-relationships, where the objects included in a map are retrieved by deductive rules. [4] Additional integrity constraints specify properties that any Map instance needs to satisfy.

Granularity and scale is modeled in a similar fashion. Figure 5.4 illustrates graphically the Telos formalization.

```
CLASS MetricScale ISA Scale WITH
   attribute
      grain:Domain
      base :Domain
      unit :Domain
   integrityConstraint
      :in(X,MetricScale)==>(X.unit=X.grain.value)
END MetricScale

CLASS Measurement IN S_CLASS WITH
   attribute, single
      value: Domain
      inscale: MetricScale
END Measurement
```

[4] For simplicity, deductive rules R1 and R2 are specified in Prolog notation.

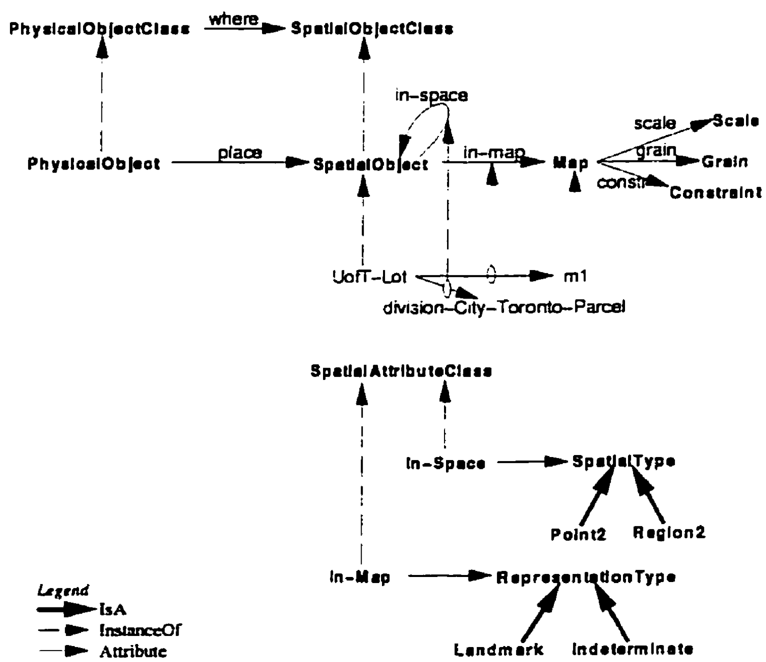


Figure 5.3: Modeling spatial information in Telos

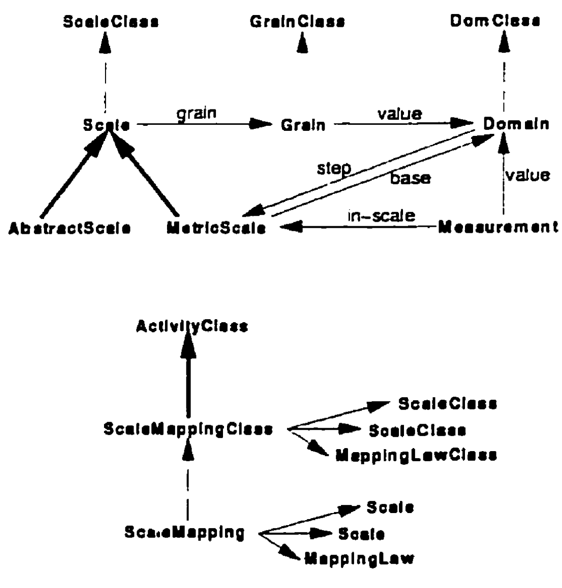


Figure 5.4: Modeling scale in Telos

5.4 Quantitative and Qualitative Spatial Constraints

This section extends the qualitative constraint language presented in Chapter 3 with a quantitative component. The section also presents an enumeration of four spatial reasoning tasks encountered in query processing and a discussion of algorithms for solving each one of them.

5.4.1 A Formal Language for Spatial Constraints

$\mathcal{L}_{\mathcal{H}}$ is intended as a language for specifying qualitative and quantitative constraints in one-dimensional space. $\mathcal{L}_{\mathcal{H}}$ is not a new language. $\mathcal{L}_{\mathcal{H}}$ extends the language of qualitative haze-order constraints of Section 4.2 (Definition 4.2.1) and it is also based on the axiomatization of haze-order space of Section 3.4. [5] $\mathcal{L}_{\mathcal{H}}$ has two sorts, a sort for haze points, H, and a sort for exact points, E; $x, y, z, ..$ are variables of sort H, $c_1, c_2, c_3, ..$ are constants (uninterpreted integers) of sort E, and g is a designated constant also of sort E. The non-logical symbols of $\mathcal{L}_{\mathcal{H}}$ include the predicate symbols $\approx$, $\prec$ relating objects of sorts $H \times H$ and $H \times E$, respectively, the predicate symbols $=$, $<$ relating objects of sorts $E \times E$, and the function symbol $-$ of sort $H \times H \rightarrow E$. [6]

An atomic formula of $\mathcal{L}_{\mathcal{H}}$ has one of the following forms: $x \approx y$, $x \prec y$, $c_1 = c_2$, $c_1 < c_2$, $x \approx c_i$, $x \prec c_i$, $y - x = c_i$, and $y - x < c_i$. The definition of terms and well-formed formulae in $\mathcal{L}_{\mathcal{H}}$ is the same as in Section 3.3.

The terms of $\mathcal{L}_{\mathcal{H}}$ are interpreted over the domain of integers. g is interpreted as the integer constant $\mathtt{g}$. Each constant c_i of the sort E is interpreted as the integer $\mathtt{i}$. Symbols $<$, $=$ and $-$ receive the standard interpretation of order, equality and subtraction over integers. Each variable x of sort H is interpreted as an integer $\mathbf{x}$ in the that ranges in the interval $[\mathbf{x}^* - \mathtt{g}, \mathbf{x}^* + \mathtt{g}]$, where $\mathbf{x}^*$ is some integer. Predicate symbols $\approx$ and $\prec$ are interpreted by the relations $R(\mathbf{x}, \mathbf{y}) = \{(\mathbf{x}, \mathbf{y}) : |\mathbf{x} - \mathbf{y}| \leq 2\mathtt{g}\}$ and $S(\mathbf{x}, \mathbf{y}) = \{(\mathbf{x}, \mathbf{y}) : \mathbf{y} - \mathbf{x} > \mathtt{g}\}$, respectively.

It should be noted that $\mathcal{L}_{\mathcal{H}}$ does not use constants to name a specific element of sort E; instead, it uses unbound variables which receive as interpretation the specific element of E

[5]The historical evolution of languages of haze-order space presented in this study, has as follows: In Section 3.4, we defined the first-order language of haze space and axiomatized the theory of one-dimensional qualitative haze space. In Section 4.2, we limited the language to conjunctive formulae and we called them qualitative haze-order constraints. In this section, we add a new sort for exact points to the language to provide for qualitative and quantitative haze-order constraints.

[6]The reader should notice that relation symbols $\sim$ and $\prec$ correspond to relation symbols h and $\prec$ of Section 3.4, respectively.

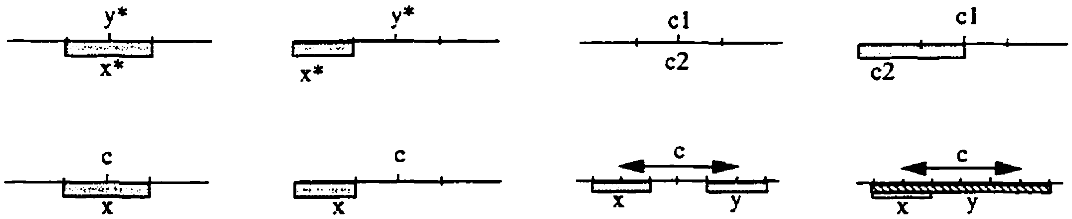


Figure 5.5: Graphical interpretation of $\mathcal{L}_{\mathcal{H}}$ terms

in any particular model of its theory.

Moreover, if g is assigned 0 in an interpretation, then $\approx$ and $\prec$ have "exact" meaning (i.e., same as $=$ and $<$), and $\mathcal{L}_{\mathcal{H}}$ becomes the language of discrete point order constraints. In fact, this is the language used for the spatial envelope constraints, since the envelopes have already "compiled away" spatial indeterminacy.

Example 5.4.1.1 Figure 5.5 illustrates graphically the meaning of the terms of $\mathcal{L}_{\mathcal{H}}$. Assume that g=2 throughout this example. If an interpretation v, assigns y to $\mathbf{y}^*$=6, then $\mathbf{y}$ can be one of {4,5,6,7,8}. If $x \approx y$, then the image of x has to be within at most 2 points from a $\mathbf{y}$, i.e., $\mathbf{x}$ can be one of {2,3,4,5,6} if $\mathbf{y} = 4$ and so on. The $\mathbf{x}^*, \mathbf{y}^*$ notation is used in the Figure 5.5 to show this relationship.

In the same model, $x \prec y$ means that $\mathbf{x}^* < 4$ if $\mathbf{y}^*$=6. The meaning of the terms $c_1 = c_2$ and $c_2 < c_1$ is the obvious one. The (hazy) equality between a haze point and an exact constant, $x \approx c_i$, means that $\mathrm{i} - \mathrm{g} \leq \mathbf{x} \leq \mathrm{i} + \mathrm{g}$, i.e., if i=8 then $\mathbf{x}$ can be one of {6,7,8,9,10}. Analogously, in the same model, x for $x \prec c_8$ is interpreted as an integer which is less than 6.

The subtraction operator "$-$" is useful in defining a notion of distance between haze points; "$-$" returns an exact quantity. [7] Let, for instance, $\mathbf{x}$ be interpreted in {1,2,3,4,5} and $\mathbf{y}$ in {7,8,9,10,11}, the term $y - x = c_6$ means that $\mathbf{x}$ and $\mathbf{y}$ are now restricted to be exactly 6 units apart, i.e., the following five pairs {(1,7),(2,8),(3,9),(4,10),(5,11)} are the only allowable interpretations for the pair x, y. □

The language of one-dimensional hazy-point space is extended to a language where one can state relationships between two-dimensional hazy points and rectangles. This is accom-

[7] The language is also extendible with a subtraction operator "$\sim$" which returns a hazy quantity. In the content of this example, the "hazy" subtraction, $x \sim y \approx c_6$, asserts that the distance between x and y has to be in the range [4,8] (i.e., 6 ±g) and therefore the following 25 pairs are possible models for the x, y pair: {(1,7), (1,8), (1,9), (2,7), (2,8), (2,9), (2,10), (3,7), ..., (3,11), (4,8), ..., (4,11), (5,9), (5,10), (5,11)}.

plished by using the technique of independent combination described in Section 3.5. By independent combination and Lemma 3.5.1, we can generate a calculus on two-dimensional spatial objects which has the computational properties of its one-dimensional coordinate calculi. Moreover, we can compose a solution for a two-dimensional constraint satisfaction problem by combining solutions of its coordinate problems. Of course, such a restricted form of combination imposes limitations to the expressiveness of the constraint language for the two-dimensional space. As already mentioned, one limitation of this method is that it limits two-dimensional spatial objects to either points or rectangles. In Section 6.3, we show how the language can be extended to model objects of arbitrary shape. The resulting language is expressive enough to cover an interesting set of spatial relationships encountered in geographic information systems [Pap94a] and picture retrieval systems [SYH94]. In the next section, we explore sets of spatial relations that are expressible in our formalism.

5.4.2 Spatial Relations

Egenhofer's proposal [Ege91] of eight fundamental topological relations for two planar regions is the most popular set of topological spatial relations. One advantage of his proposal is its clean topological semantics. Our work defines an alternative semantics for these relations based on the ontology of hazy points. The novelty in our approach is that it considers spatial relationships between objects with vague boundaries. In addition, a measure of precision appears as a parameter in the relations' definition, thus making reasoning about imprecise spatial information possible.

Figure 5.6 shows a graphical presentation of topological relations. Their formal definition requires the definitions of the helping relations in_0, in_1 and $close$. [8]

$$\begin{aligned}
in_0(P,R) &\equiv (R_{S_X} \prec_X P_X \wedge P_X \prec_X R_{E_X}) \wedge (R_{S_Y} \prec_Y P_Y \wedge P_Y \prec_Y R_{E_Y})\\
in_1(R1,R2) &\equiv (R2_{S_X} \prec_X R1_{S_X} \wedge R1_{E_X} \prec_X R2_{E_X}) \wedge (R2_{S_Y} \prec_Y R1_{S_Y} \wedge R1_{E_Y} \prec_Y R2_{E_Y})\\
close(P1,P2) &\equiv (P1_X \approx_X P2_X) \wedge (P1_Y \approx_Y P2_Y)
\end{aligned}$$

$P, P1$ and $P2$ are points and $R1, R2$ rectangles. [9]

[8] "Closeness" in the this definition is expressed in terms of the properties of the representation, e.g., the size of the haze relation. Other approaches define "closeness" in terms of the context of the representation [Rob89].

[9] Notation: $x_i \preceq_i y_i \equiv x_i \prec_i y_i \vee x_i \approx_i y_i$. Subscript $_i$ denotes the projection axis. $_i$ is either $_X$ or $_Y$.

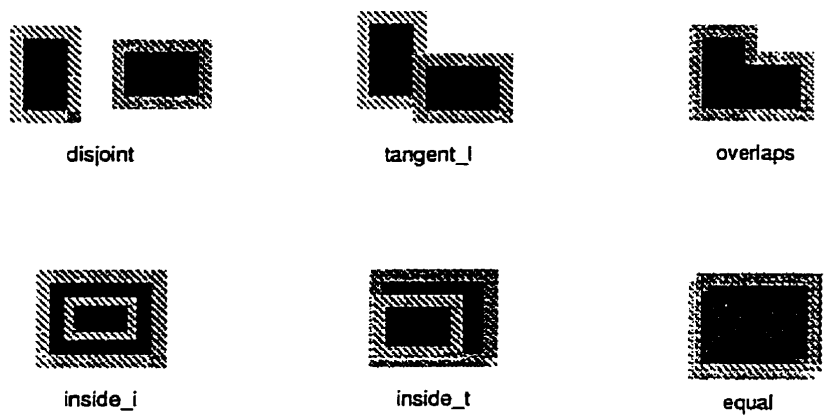


Figure 5.6: Hazy topological relations

$$\begin{aligned}
\texttt{disjoint}(R1, R2) &\equiv \neg\exists P(in_0(P, R1) \wedge in_0(P, R2)) \\
\texttt{tangent}_\texttt{l}(R1, R2) &\equiv \exists P1, P2(in_0(P1, R1) \wedge in_0(P2, R2) \wedge close(P1, P2)) \\
\texttt{overlap}(R1, R2) &\equiv \exists P(in_0(P, R1) \wedge in_0(P, R2)) \\
\texttt{inside}_\texttt{i}(R1, R2) &\equiv in_1(R1, R2) \\
\texttt{inside}_\texttt{t}(R1, R2) &\equiv (R2_{S_X} \preceq_X R1_{S_X} \wedge R1_{E_X} \preceq_X R2_{E_X}) \wedge (R2_{S_Y} \preceq_Y R1_{S_Y} \wedge R1_{E_Y} \preceq_Y R2_{E_Y}) \\
\texttt{contain}_\texttt{t}(R1, R2) &\equiv \texttt{inside}_\texttt{t}(R2, R1) \\
\texttt{contain}_\texttt{i}(R1, R2) &\equiv \texttt{inside}_\texttt{i}(R2, R1) \\
\texttt{equal}(R1, R2) &\equiv close(R1_S, R2_S) \wedge close(R1_E, R2_E)
\end{aligned}$$

The notion of tangency as defined here is "loose" tangency in the sense that two rectangles share a point of their haze. If the size of the haze decreases, a tangent relationship will change to disjointness. The above set of relations characterize all the qualitative distinct relative positionings of two object using the proposed ontology of space [Top94a].

Many researchers developed sets of directional relations exploring either characteristic points of the participant objects [PS94, Her92] or the order relation of the underlying domain [SYH94]. Our directional relations definition is closely related to the approach of [SYH94]; this model comes with a deductive reasoner that is shown to be sound and complete for the three-dimensions and sound only for the two dimensions. We show that the axioms of this deductive reasoner (except one) are theorems of $PR2$. Its incompleteness result does not affect our case since we do not use this rule system for inferencing.

The following non directional spatial relations complete Sistla's model: `inside` (same as the earlier $\texttt{inside}_\texttt{i}$), `outside` (same as `disjoint`) and `overlaps` (with the obvious

I	$left\_of(A,B) \wedge left\_of(B,C)$	$\Rightarrow$	$left\_of(A,C)$	†
II	$left\_of(A,B) \wedge overlaps(B,C) \wedge left\_of(C,D)$	$\Rightarrow$	$left\_of(A,D)$	‡
IIIa	$inside(A,B) \wedge left\_of(B,C)$	$\Rightarrow$	$left\_of(A,C)$	*
IIIb	$left\_of(A,B) \wedge inside(C,B)$	$\Rightarrow$	$left\_of(A,C)$	*
IV	$overlaps(B,A)$	$\Rightarrow$	$overlaps(A,B)$	**
V	$outside(A,B)$	$\Rightarrow$	$left\_of(A,B)$	‡
VI	$inside(A,B)$	$\Rightarrow$	$overlaps(A,B)$	***
VII	$inside(C,A) \wedge overlaps(C,B)$	$\Rightarrow$	$overlaps(A,B)$	
VIII			$inside(A,A)$	

†	The same rule scheme is repeated for relationship symbols *above*, and *inside*
‡	same for *above*
*	same for *above* and *outside*
**	same for *outside*
***	not express in our formalism

Table 5.1: Rules for deducing spatial relationships

meaning). The axioms of [SYH94] deductive system are summarized in table 5.1. The rules of table 5.1 (except VI) are theorems of $DR2$.

Finally, and as a result of the independent combination property, our model can handle relationships between one-dimensional and two-dimensional point or region objects.

Entry **R6**$'$ of table 5.2 could be completed in the same way as the preceding entries i.e., by taking all possible conjunctions of the $\prec$ and $\approx$ over the eight parameters defined by two regions; that would result in 64 different relative positions, many of them meaningless and therefore unsuitable to be used in a query language. Instead, we use a more succinct way to express the twelve aforementioned two-dimensional relations. Their formulation uses a language with existentially quantified variables and therefore are suited better for querying the database using a quantifier-elimination query processing strategy [Kou94b]. For insertions, we favor a subset, **R6**, of 38 meaningful relations out of the 64, which are shown on Table C.1 of appendix B. All relations in **R6** are translated in a conjunctive expression over hazy points.

5.4.3 Reasoning

A map structure is represented now in terms of two constraint sets, $Xcon$ and $Ycon$, each of which is a conjunction of $x\ R\ c$ and $x - y\ R\ d$ atomic constraints, where x, y are variables representing the parameters of a spatial envelope, R is one of $\{=, <, \leq\}$, d is a grain parameter, and c is a constant corresponding to a landmark parameter. Any map with an envelope representation for its indeterminate objects can be placed in this

R1: one-dimensional, point-to-point †	
$x_i \prec_i y_i$	$x_i \approx_i y_i$
R2: one-dimensional, point-to-interval †	
$before_i(x_i, I_i) \equiv x_i \prec_i I_{S_i}$	$touches_i(x_i, I_i) \equiv x_i \approx_i I_{S_i}$
$in_i(x_i, I_i) \equiv I_{S_i} \prec_i x_i \wedge x_i \prec_i I_{E_i}$	$touched\_by_i(x_i, I_i) \equiv x_i \approx_i I_{E_i}$
$after\_(x_i, I_i) \equiv I_{E_i} \prec_i x_i S_i$	
R3: one-dimensional, interval-to-interval ‡	
$same_i(I_i, J_i) \equiv I_{S_i} \approx_i J_{S_i} \wedge I_{E_i} \approx_i J_{E_i}$	$before_i(I_i, J_i) \equiv I_{E_i} \prec_i J_{S_i}$
$meets_i(I_i, J_i) \equiv I_{E_i} \approx_i J_{S_i}$	$finishes_i(I_i, J_i) \equiv J_{S_i} \prec_i I_{S_i} I_{E_i} \approx_i J_{E_i}$
$starts_i(I_i, J_i) \equiv I_{S_i} \approx_i J_{S_i} \wedge I_{E_i} \approx_i J_{E_i}$	$over_i(I_i, J_i) \equiv J_{S_i} \prec_i I_{S_i} \wedge I_{E_i} \prec_i J_{E_i}$
$overlaps_(I_i, J_i) \equiv I_{S_i} \prec_i J_{S_i} \wedge J_{S_i} \prec_i I_{E_i} \wedge I_{E_i} \prec_i J_{E_i}$	
R4: two-dimensional, point-to-point	
$close(P, Q) \equiv P_X \approx_X Q_X \wedge P_Y \approx_Y Q_Y$	
$north(P, Q) \equiv Q_Y \prec_Y P_Y$	$east(P, Q) \equiv P_X \prec_X Q_X$
$south(P, Q) \equiv P_Y \prec_Y Q_Y$	$west(P, Q) \equiv Q_X \prec_X P_X$
$north\_east(P, Q) \equiv Q_Y \prec_Y P_Y \wedge P_X \prec_X Q_X$	$south\_east(P, Q) \equiv P_Y \prec_Y Q_Y \wedge P_X \prec_X Q_X$
$north\_west(P, Q) \equiv Q_Y \prec_Y P_Y \wedge Q_X \prec_X P_X$	$south\_west(P, Q) \equiv P_Y \prec_Y Q_Y \wedge Q_X \prec_X P_X$
$x\_colinear(P, Q) \equiv P_Y \approx_Y Q_Y$	$y\_colinear(P, Q) \equiv P_X \approx_X Q_X$
R5: two-dimensional, point-to-region	
$inside(P, A) \equiv A_{S_X} \prec_X P_X \wedge P_X \prec_X A_{E_X} \wedge A_{S_Y} \prec_Y P_Y \wedge P_Y \prec_Y A_{E_Y}$	
$north(P, A) \equiv A_{E_Y} \prec_Y P_Y$	
$east(P, A) \equiv P_X \prec_X A_{S_X}$	
$south(P, A) \equiv P_Y \prec_Y A_{S_Y}$	
$west(P, A) \equiv A_{E_Y} \prec_X P_X$	
$north\_east(P, A) \equiv A_{E_X} \prec_Y P_Y \wedge P_X \prec_X A_{S_X}$	
$south\_east(P, A) \equiv P_Y \prec_Y A_{S_Y} \wedge P_X \prec_X A_{S_X}$	
$north\_west(P, A) \equiv A_{E_X} \prec_Y P_Y \wedge A_{E_X} \prec_X P_X$	
$south\_west(P, A) \equiv P_Y \prec_Y A_{S_Y} \wedge A_{E_X} \prec_X P_X$	
$on\_left\_side(P, A) \equiv P_X \approx_X A_{S_X} \wedge A_{S_Y} \prec_Y P_Y \wedge P_Y \prec_Y A_{E_Y}$	
$on\_top\_side(P, A) \equiv P_Y \approx_Y A_{E_Y} \wedge A_{S_X} \prec_X P_X \wedge P_X \prec_X A_{E_X}$	
$on\_right\_side(P, A) \equiv P_X \approx_X A_{E_X} \wedge A_{S_Y} \prec_Y P_Y \wedge P_Y \prec_Y A_{E_Y}$	
$on\_bottom\_side(P, A) \equiv P_Y \approx_Y A_{S_Y} \wedge A_{S_X} \prec_X P_X \wedge P_X \prec_X A_{E_X}$	
R6′ : two-dimensional, region-to-region **	
$disjoint(A, B)$	$tangent_l(A, B)$
$overlap(A, B)$	$inside_i(A, B)$
$inside_t(A, B)$	$contain_t(A, B)$
$contain_i(A, B)$	$equal(A, B)$
$left\_of(A, B)$	$right\_of(A, B)$
$above(A, B)$	$below(A, B)$

† subscript $_i$ is either $_X$ or $_Y$

‡ with their inverses, except $same_i$, they make the Allen's thirteen.

** the same as the topological and directional relations defined earlier.

Table 5.2: Spatial Relations

simple normal form. Note that the set of variables in the two sets is disjoint (except for d's). The reader should also recognize that the deployed language in the map constraints is the language of linear order constraints on integers as resulted from the compilation of indeterminates into envelope constraints.

The fundamental reasoning problems addressed in a constraint representation of a map structure are as follows:

P1. *Given the $Xcon$ and $Ycon$ constraint sets, decide if the constraint sets are satisfiable, i.e., there is an assignment for variables that satisfies every atomic formula in the $Xcon$ and $Ycon$ set.*

P2. *Given the $Xcon$ and $Ycon$ constraint sets, compute an assignment for all variables that satisfies every atomic formula in the $Xcon$ and $Ycon$ set.*

The type of the d parameters plays a pivotal role in the determination of the complexity of the above problems. If d's are integer constants (fixed grains) then both problems are solveable in polynomial time. For instance, problem **P1** could be solved using one of several path consistency algorithms proposed in [Mac77, DMP89, KL91, Kou94b]. The complexity of path consistency algorithms for the type of constraints considered is $O(n^3)$ where n is the number of variables in the constraint set. In the database literature, the classic results of [RH80] offer another alternative with the same complexity. Our guess is that even better performance can be achieved if one explores the structure and especially the sparsity of constraint sets. The second reasoning problem, is closely related to the first one. In fact, a solution for **P2** implies a solution for **P1**. A plethora of solutions is available for **P2**, including the above-mentioned path consistency algorithms as well as the dual method involving variable elimination algorithms (see [LM88a, Kou92]).

The two problems change complexity if the ds are taken to be integer variables. Then, for a single d value, efficient algorithms are still possible since computing a solution involves solving a system of linear inequalities (a known polynomial complexity problem). For more than one d value, on the other hand, the problem of computing a minimal solution becomes intractable since it is equivalent to integer programming [Pap94b]. In our future work we plan to investigate efficient special cases for the last problem.

In addition to **P1** and **P2**, there are two derivative spatial reasoning problems which require attention:

P3. *Given a consistent and minimal constraint set, $Xcon \cup Ycon$, of a map structure m, and i, an indeterminate of m, find the strongest possible bounds for the parameters of i.* [10]

Algorithmically, **P3** involves, first, projecting a solution of **P2** to the variables of i's envelope and, second, applying a path consistency algorithm on the selected set of constraints conjoined with i's local constraints. Both steps are realized in polynomial time. Note that in order to determine the consistency of a map, m, we need to test **P3** for all of its indeterminates.

P4. *Given a consistent and minimal constraint set, $Xcon \cup Ycon$, of a map structure m, and g', a new grain value (resolution) for m, recompute problem* **P2** *with grain value g'.*

Problem **P4** involves recomputing the constants for the landmark parameters in the $Xcon$ and $Ycon$ constraint sets and then computing **P2**. For the first step, the following linear time procedure applies. There are two directions to which the representation's grain can change

- *refinement* ($g' < g$): Let r be the refinement factor, $r = \frac{g}{g'}$ (assume g' divides g). Then each $P = v$ conjunct in the $Xcon$ and $Ycon$ constraint set is replaced by the constraint $(r * (v - 1) \leq P) \wedge (P \leq r * v)$.
- *coarsening* ($g' > g$): Let s be the coarseness factor, $s = \frac{g'}{g}$ (assume g divides g'). Replace each term $P = v$ with $P = \frac{v}{s}$.

There are various semantic issues that our quick coverage of the transition operations has overlooked. Ciapessoni et al. [CCMSP93] present an elaborate framework for scale-related granularity which is relevant to the above problem. Clifford and Rao [CR87] have studied the problem of scale transitions in discrete temporal domains.

[10] Recall that parameters of an indeterminate are constrained by its envelope and the size constraints. Problem **P3** calls for the determination of tight bounds, i.e., the smallest interval that a parameter can take values from, for the parameters of a specific indeterminate.

5.5 Conclusions

We have presented a spatial data model which facilitates the representation of and reasoning with various forms of qualitatively and quantitatively incomplete spatial information, including indeterminate objects, multiple scales and granularity. Representation of incomplete spatial information is accomplished through a spatial constraint language with built-in notions for representing partial spatial information. Reasoning with such representations is addressed by identifying four classes of reasoning tasks and offering efficient processing algorithms for each class. Our proposal accommodates object-orientation by embedding the proposed model within Telos and exploiting the meta-modeling facilities of the latter.

The proposed spatial data model is unique in the combination of features that it accommodates. In particular, it integrates ideas from object-oriented knowledge representation [MBJK90], constraint-based data models [KKR90], spatial knowledge representation [PS94], quantitative and qualitative temporal reasoning [KL91], and granularity modeling [CCMSP93].

Our spatial data model may be atypical of other proposals, however it complies with general structure of spatial data models. For example, the model matches features from three out of four abstractions of space that need to exist in spatial data model according to Guting's recent definition [Gut94]. In particular, it organizes the underlying space on a geometric basis (represented by constraints), offers a spatial relation-based language, and, integrates geometric types into the data model. Our model has limited spatial data types support, points and rectilinear regions, but as we show in Chapter 6, the integration of lines and polygons is straight forward.

We believe that the model could be of use in non-spatial applications as well, particularly ones involving dimensional data, such as temporal databases, genome databases and financial databases.

Chapter 6

Applications

In this chapter we present example applications of the proposed spatial data model and haze-order reasoning techniques in the fields of Geographic Information Systems and Genome Informatics. The presentation of each example application focuses on a specific feature of the developed methods and in addition it points to directions in which the methods can be extended.

The rest of this chapter is organized as follows: Section 6.1 presents a solution to problem of multiple representations of space. Section 6.2 applies the machinery developed in Chapter 5 in order to process queries with a granularity argument. Section 6.3 presents a solution to the problem of modeling objects with indeterminate boundaries in spatial databases. Section 6.4 discusses applications of the haze-order space in the context of the Human Genome Project. Finally, section 6.5 concludes the chapter with a discussion on the implementation of spatial knowledge bases within a knowledge base management system.

6.1 Multiple Representations of Space

In cartographic representations, an object's representation changes according to the level of abstraction at which data is represented. The process of converting spatial data from one scale-dependent to another is called generalization [SM89]. Recent geographic information systems aim to support cartographic generalization by maintaining multiple representations [RS95].

The spatial data model introduced in Chapter 5 supports multiple spatial representations. According to this model, every spatial object has an `in-space` attribute which specifies the object's spatial context, and one or more in-map attributes which specify the object's spatial type and location in different maps. A map is the data structure used to represent a chunk of space along with the objects that it contains. A map has a scale attribute. Hence, multiple scales representation support is equivalent to creating multiple instances of a map for different scales.

In Chapter 5, we have shown a constraint-based implementation of maps as well as operations for conversing scales in a constraint-based representation. Other data types are also candidates for implementing maps. For instance a map can be implemented as a binary array (see Figure 5.1(a)). In this case, the embedded objects are represented by an enumeration of the cells they occupy or represented by their boundary. Scale conversions are well studied in grid-like representations [Sam89].

6.2 Querying Spatial Data

Storing a granularity parameter together with the data forms an alternative solution to the similarity-based retrieval problem. In similarity-based retrieval, a query is associated with a similarity measure that specifies the degree of similarity between the retrieved times and the matching criteria [Jur95].

Our model associates a (spatial) granularity and a (spatial) scale parameter to the data being stored. Hence, the data "know" about their imprecise placement in space. A query can either be an exact query against imprecise data or it can specify the degree of precision at which an answer is sought. The following examples illustrate cases of granularity-based retrieval based on the map model developed in Chapter 5 and the example of Figure 6.1. A more comprehensive form of this example is presented in appendix D.

Example 6.2.1 Querying Spatial Relations with Granularity.

Assume the query: *Find if the Computer Science Department is inside the scope of* $V1$. This query may be initially evaluated with the highest possible precision (lowest grain), i.e., $g_q = 0$. Then the query is whether the Computer Science Department, an exact point with coordinates, say, $\langle 5, 16\rangle$, is inside the scope of $V1$. [1]

$$Q1_0 = \{true | P = \langle 5,16\rangle \wedge g_q = 0 \wedge (inside(P, V1, g_q) \vee on\_any\_side(inside(P, V1, g_q))\}$$

[1] on_any_side abbreviates $on\_left\_side \vee on\_right\_side \vee on\_top\_side \vee on\_bottom\_side$.

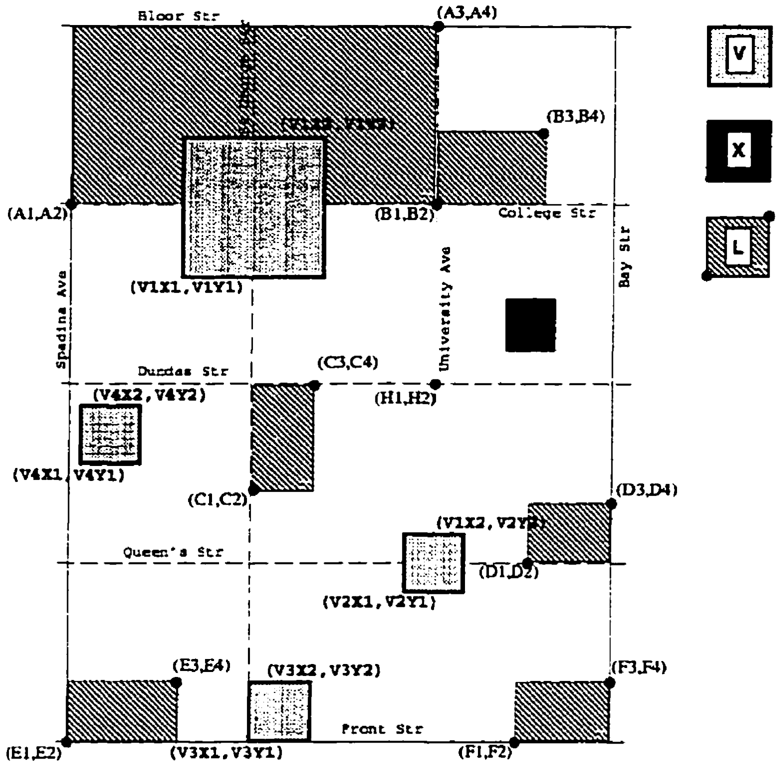


Figure 6.1: A map example

This query is an instance of problem **P3** for $V1$ (see Section 5.4) followed by the evaluation of the query predicates. As there is more than one database model describing a consistent set of parameters for $V1$, this query is ambiguous and can be answered with respect to one or all possible worlds. In constraint-based query processing there are two possible semantics that can be used for answering such a query: (a) truth in at least one possible world, and (b) truth in all possible worlds. Koubarakis [Kou94a] calls them the *possibility* and *certainty* problems, respectively. The existence of a grain notion in our representation makes the answers to these two problems Dependent on the grain size of the query or the representation. Below, we ask the same query assuming indeterminate position for the Computer Science Department

$$Q1_g = \{true | P = \langle 5, 16\rangle \wedge g_q = 1 \wedge (inside(P, V1, g_q) \vee on\_any\_side(inside(P, V1, g_q))\}$$

The implication of using an imprecise search point is to extend the selectivity of the query predicates. This a done by compiling their expressions for $g_q = 1$. One can obtain another variant of the query by changing the precision of $V1$. This means that its envelope constraints have to be recompiled, leading to new instances of problems **P1** and **P2**, i.e., the consistency and minimality of the map must be verified for the new envelope constraints. Then the query is evaluated following the steps described above. Other spatial relation-based queries can be expressed, and evaluated, along the same lines. □

In spatial databases, operations that relate sets of spatial objects are very important, e.g., overlay of two maps, merging of adjacent areas, etc. These operations are easily handled by constraint-based representation if no scale parameter is involved. In the case of multiple scale representations, a scale adjustment operation needs to be preceded.

Example 6.2.2 Map Overlay as Constraint Merging Operation.
Assume that we are given a map of Toronto hospitals and wish to find which hospital is closer to a trouble spot. This involves adjusting the scales, recomputing the constants (problem **P4**), merging the constraint sets of the two, and proceeding with problems **P1** and **P2** (i.e., verify the consistency of the merged map) and evaluating of the predicate $close(X, H)$. A number of optimizations are possible in this case, especially if we exploit the inclusion/nesting property of the scale hierarchy. □

Changing the scale of a map to a coarser one is known as the map simplification operation [PD95] in geography. The **P4** reasoning operation, mentioned in Chapter 5, computes a solution of the map simplification operation in the context of minimum boundary rectangle representations of spatial databases such as the one studied in [PTSE94].

6.3 Regions with Indeterminate Boundaries

Objects with indeterminate boundaries are not handled by today's Geographic Information Systems. Modeling of indeterminate boundaries is desired in many applications such as modeling habitats (the area that wheat grows may not be crisp), coast lines (the coast line changes with time or seasons), etc. A difficulty with habitats, for example, is that some of their boundary points are not connected, hence regions are not convex and not connected. The boundary of a habitat is intrinsically probabilistic. This property is nicely captured by the ontology of haze space.

Typical queries that can be asked against a habitat include topological queries such as "is place X part of the habitat?", and geometric queries such as "what is the area of the

$g2 \backslash g1$	1	0	−1
1	1		
0	.75	.5	.25
−1			0

Table 6.1: Conjoining certainty values of boundary points in Figure 6.2(b)

habitat?". The first query is expected to yield a yes/no/maybe answer while the second query is expected to return a range value for the habitat area consisting of a minimum and maximum value. Current GISs should give a yes/no (binary) answer and a real number result for the former and latter queries, respectively.

Hadzilacos [Had96] has suggested two alternative solutions towards the modeling of habitats in GISs: In the first solution, a habitat is seen as a sequence of regions with crisp boundary [Coh95], [PD95]. This is referred to as the matrioska (Russian doll) model. This solution operates under the binary query model. The second solution is based on the haze-order ontology of space and utilizes a query processing strategy which is capable of yes/no/maybe answers. The following two examples illustrate the application of the haze-order space in habitats modeling.

Example 6.3.1 (Rectangular Regions)
Figure 6.2(a) and (b) illustrate rectangular regions with crisp and multiple haze boundaries respectively. Figure 6.2(c) shows the steps needed to transform a crisp rectangle to a haze rectangle with haze size $g1$. The point-in-rectangle query at granularity $g_q = 0$, $Q1_0$, specified in Example 6.2.1 yields a yes or maybe answer if and only if its expression evaluates to true in all possible models or in some models, respectively. In all other cases, it returns a no answer. Finally, Figure 6.2(d) depicts the case of a rectangular region with a wider haze boundary, $g2 > g1$. Overlaying the two haze boundaries, we obtain a region with multiple haze boundaries that resembles the matrioskas model but with fewer represented rectangles. In a region with a single boundary, we have defined three qualitative distinct areas: the certain interior (1), the haze boundary (0), and the certain exterior (-1). Points of the haze boundary may be thought as of being part of the habitat with certainty 0.5. In a realistic habitat, however, the distant points of the haze boundary should have lower certainty of being part of the habitat. Overlaying two or more haze boundaries enforces this property. The point-to-rectangle query is now posed against both granularities, and the certainty of the boundary points is scaled after conjoining the two answers as the table 6.1 dictates. □

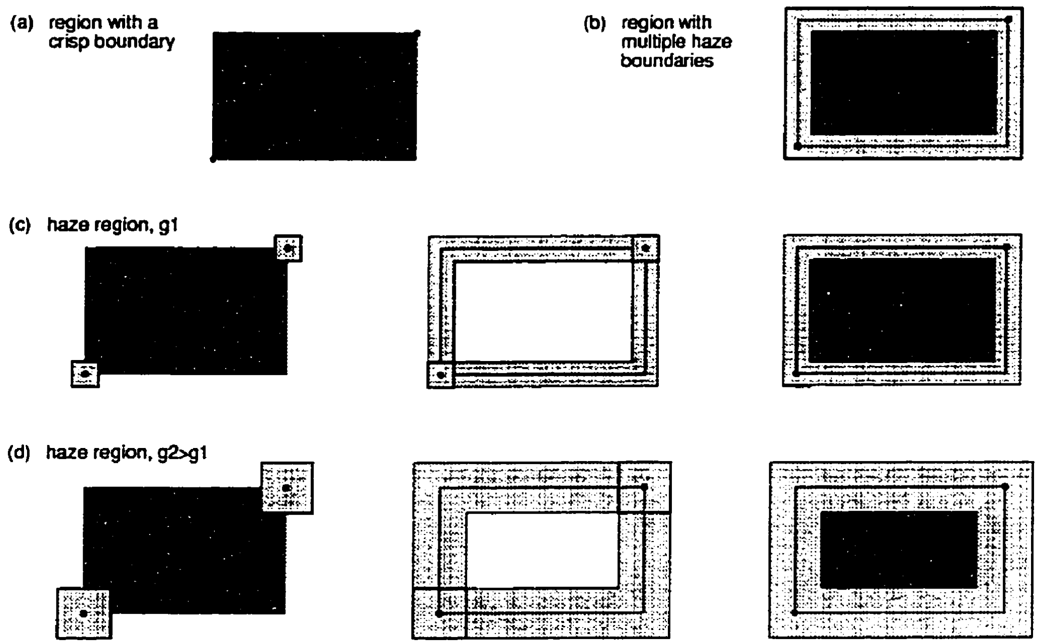


Figure 6.2: Rectangular region with multiple haze boundaries

An extension of our data model is to provide for the representation of arbitrary complexity (shape) convex objects. How is this done and what are the implications for the validity of the earlier discussion? The support of non-rectilinear spatial objects in two-dimensional haze space is a straightforward task. We first define a line to be a pair of two points, called endpoints. Then we define a chain to be a sequence of connected lines, and a polygon to be a closed polygon. A polygon in a plane defines two regions: the interior and the exterior. As the points used in these definitions have a haze, the defined types are inherently "haze". The representational model is now extended with construction operators such as $line(P_1, P_2)$ which constructs a line out of two points, etc. Each type now has a complexity identifier, for instance a two-dimensional point has complexity 0, a line has complexity 1, and a region has a complexity 2. Thus many interesting type constraints may be associated with types such as the boundary of an object o has to be an object with complexity one less than o's complexity etc.

Relations between convex objects of arbitrary shape are defined either in terms of Egenhofer's 3-intersection model [Ege91] or by relating characteristic points of objects [Pap94a]. A necessary extension to the underlying computational model is that the primitive *point_to_polygon* operation becomes a 3-valued operation. This change only affects reasoning

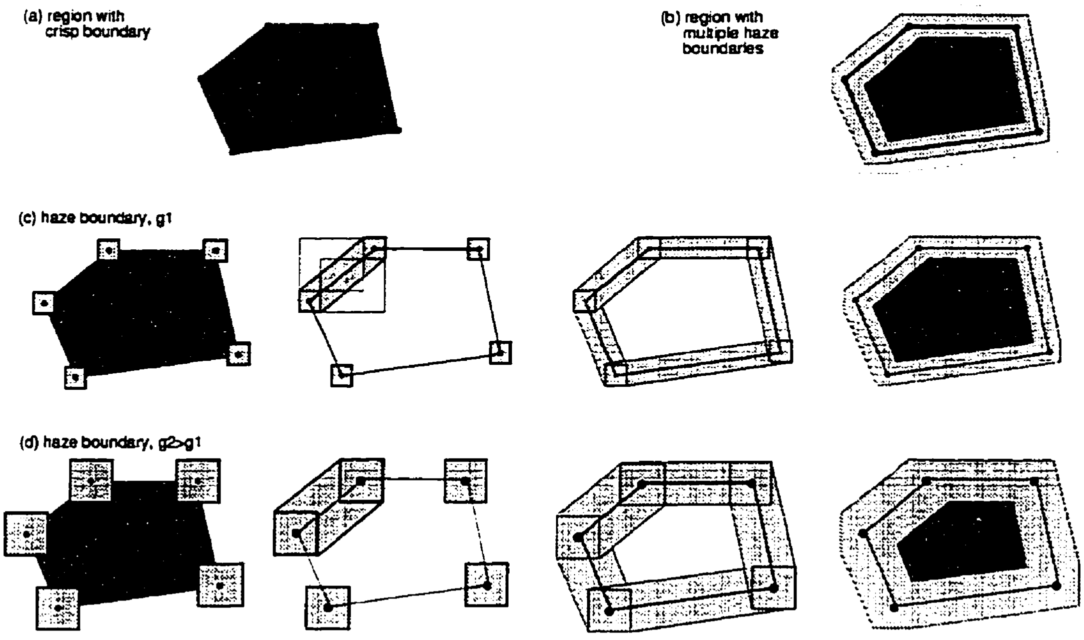


Figure 6.3: Region with multiple haze boundaries

with topological relations using the 3-intersection model. The proposed, constraint-based reasoning model based on two one-dimensional projections and minimum bounding rectangle (mbr) approximations is still valid. In addition, further non-mbr based computations using the one-dimensional projections are possible, provided that a new mapping between two-dimensional relations and their one-dimensional projections is defined.

Example 6.3.2 (Arbitrary Shapes)

Figure 6.3 outlines the same process for regions with arbitrary shape. A region with arbitrary shape is described by its boundary which is a closed sequence of connected line segments. The difference with previous example is the way that the point-in-polygon query is computed. □

6.4 Genome Mapping

A major goal of the Human Genome Project is to construct detailed physical maps of the human genome. A physical map is an assignment of DNA fragments to their locations on the genome. Complete maps of large genomes require the integration of many kinds of experimental data, each with its own forms of noise and experimental error [HB94]. In addition, the Human Genome Project has caused an incredible data explosion in the

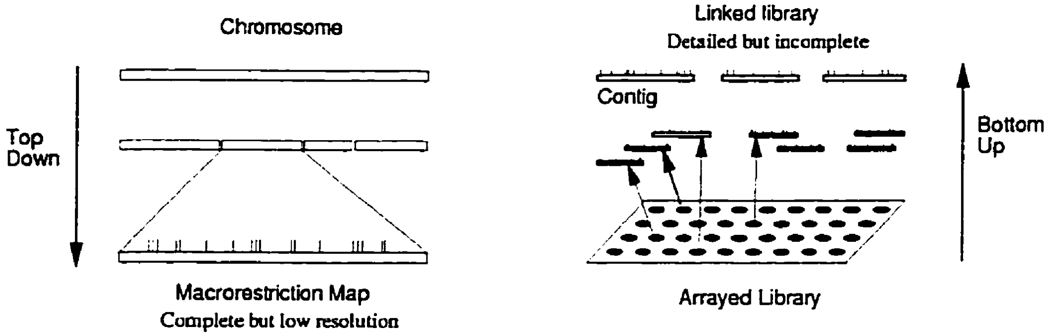


Figure 6.4: Physical mapping strategies (based on [DOE92])

biological sciences. New laboratory procedures and laboratory automation systematically produce large data sets that need to be stored in databases and processed by data-analysis programs [GRS94].

Our work is related both to the physical mapping problem and to the laboratory database support of the Human Genome. In the rest of this section we summarize the data requirements and outline the applicability of our methods for both problems. A detailed treatment of the proposed solutions constitutes future work of this dissertation.

Genome maps are constructed either in top-down or bottom-up fashion (see Figure 6.4, [DOE92]). In top-down mapping, a single chromosome is cut into large pieces which are ordered and subdivided; the smaller pieces are then mapped further. This approach yields maps without gaps but with low resolution. The bottom-up approach involves cutting the chromosome into small pieces each of which is cloned and ordered. The ordered fragments form DNA blocks, called contigs. Contig maps consist of a linked library of small overlapping clones representing a chromosome segment. Although this technique is useful for creating good local maps, contig maps are difficult to extend over large stretches of the chromosome. In this case, sophisticated order reasoning that utilizes biological laws and experimental data is sought [Cui94].

The problem of physical map assembly is illustrated in Figure 6.5. The same abstract scenario is encountered at different levels of resolution. Overlapping intervals of various resolutions (e.g. YACs, cosmids, plasmids) need to be ordered and put together in order to create longer fragments of sequenced genome. At the highest resolution, the bottom line, we see landmarks of the genetic sequence, called probes, whose position in the genetic sequence has been identified by experimental means. The linkage (intersection) of known

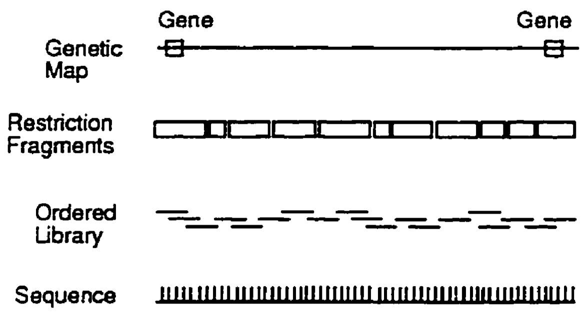


Figure 6.5: Types of genome maps (based on [DOE92])

probes with the overlapping interval is the input for the physical mapping problem.

The challenging part of the problem and its relevance to the techniques developed in this thesis, is that these genetic intervals are incomplete (orientation information might be missing, metric information is not always available), imprecise (their endpoints are not precise), ambiguous (high rate of false negatives), and often contradictory (due to the previous reasons). The input information for the physical mapping problem is a set of statements of the type: "probe p hits interval i". Since probes are ordered, this information can be used to order and assemble the intervals. Unfortunately, the data may contain up to 40% false negatives and false positives. In a false negative, probe p_i appears not to hit interval i_j although it should, and the opposite for false positives. Earlier map assembling techniques [WJ86] operate under the assumption on non-ambiguous data. Recent research addressed the problem by considering the ambiguity of data [GR93], [HB94], [Cui94]. This problems sounds as a very promising application for haze-order constraints.

Haze-order constraints is logic-based constraint language with a built-in concept for imprecision merging qualitative and quantitative terms. As a first step, we can use haze-order constraints as a succinct model to describe experimental genome map data that accounts for false negatives and measurement errors.

Genetic intervals may be represented as ordered pairs of haze points related to other points via haze and order relationships. We assume that genetic intervals are haze points. If any two of them are "hit" by the same probe, we say that they are in a haze relationship, else they are ordered. False positives are represented by "illegal" haze relations where false negatives may exists in the place of "absent" haze relations. The questions are which of the haze relations are "illegal" and where should there be "absent" haze relations. Biological knowledge can give hints for answering these questions. Such knowledge can be used in order to add/remove haze relations. The consistency checking algorithm will either accept or reject such updates. This type of analysis may also localize inconsistencies and insist on further laboratory experiments for this part of the genome.

A representation of incomplete and imprecise data has more than one model (orderings) that satisfies the specification. Existing databases for laboratory support use exact numerical representations and therefore maintain a single and perhaps faulty model for each set of experiments. Our proposal suggests storing uninterpreted data, such as haze-order constraints. Hence the data are represented together with all of their models. The "tight" graph algorithm defined in Section 4.3, can be used to minimize the number of possible models.

6.5 Implementing Multi-resolution Space in Telos

An implementation of the proposed data model is prototyped by first defining the non-spatial concepts using an existing data model (e.g., the one presented in section 5.3 or as relations in an object-based or relational system) and then implementing a constraint-based inference engine for spatial constraints.

To date, there is no generally accepted way for implementing large constraint databases. Logic programming and constraint logic programming are two obvious alternatives with respective limitations regarding their scalability for large workloads. Constraint-based reasoners are another option. We are currently exploring the last option. Our proposed solution to the scalability problem is to explore the scale and grain features in order to partition large chunks of space into many and small maps.

This section describes an implementation of the proposed features for representing spatial information using the conceptual modeling language Telos [MBJK90]. The integration of the spatial representation features with Telos endows the resulting spatial data model with abstraction mechanisms such as generalization, classification and attribution, inherited from Telos, as well as facilities for expressing meta-concepts and for asserting constraints and rules. The integrated system is then put together using the framework developed in the Telos knowledge base management project [Top93], [MCP+95], [ST95].

Integration of spatial modeling facilities into Telos is accomplished through a library of meta-classes and meta-attributes that capture the semantics of the features presented

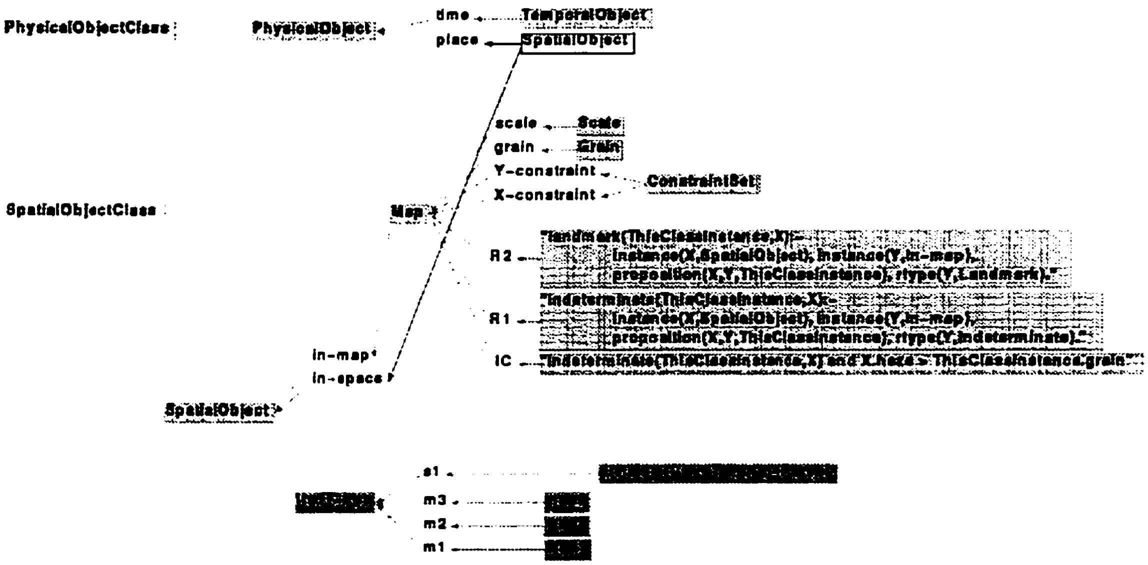


Figure 6.6: Representing spatial information in Telos

in Section 5.3. The central class of the model is the `Map` class (introduced in Section 5.2.4). Spatial information is attached to physical objects through a spatial object which participates in one or more maps. Figure 6.6 displays part of the knowledge base that implements the spatial data model using the the Telos Repository system [Sta95].

The second part of our implementation involves implementation of constraint reasoning in the context of haze orders and integrating it with the object model. Our achievements to date include the implementation of algorithms presented in Section 4.3. These algorithms include a qualitative constraint consistency checking algorithm based on the data structure of *haze-order* graphs [Top96a], and a ranking procedure which converts qualitative haze-order graphs to a quantitative form that is suitable for query processing [Top94b].

The implementation will be completed with the development of a third component which will be responsible for answering queries with spatial and non-spatial qualifications. To date, this component is not available. However, we envision this component as being a hybrid query processor in the spirit of [SPT87], [TIS92] which extends the Telos query processor [ST95] with a spatial reasoning capabilities.

6.6 Conclusions

We have presented applications of the proposed spatial data model and haze-order reasoning techniques in the fields of Geographic Information systems and the Human Genome Project. In addition, we have outlined a prototype implementation of the proposed data model based on the Telos knowledge base management framework.

Our main interest in this chapter has been to discuss the potentials for applications of the methods proposed in this dissertation rather than work out the technical details in each individual application. Consequently, this chapter contributes to the thesis by generating a plethora of open research questions related to several practical applications.

Chapter 7

Conclusions

In this final chapter we give a summary of the dissertation, highlight its main contributions and identify open problems.

7.1 Summary

This dissertation studies the problem of representing and reasoning with imprecise spatial information in knowledge bases. We have given examples of spatial information for which existing representations are unable to provide well-founded support. Generality and efficiency have been the main objectives in this work. Generality in information management issues is achieved through an extensible data model. Efficiency, on the other hand, is achieved through the selection of an appropriate ontology for space. This dissertation argues that the combination of artificial intelligence methods and databases offers a powerful framework for addressing the problems in question.

Initially, we have introduced a new ontology of imprecise space in which space is viewed as a totality of objects surrounded by a haze area and connected in terms of qualitative spatial relations. A haze point is the most elementary object type in this representation since higher order objects are composed of haze points. A haze point is a non zero-sized object that is associated with an area of haze such that the point in question may be located anywhere inside it. Haze points are related in terms of an indistinguishability (called haze) or an order relation.

We have developed a first-order theory of one-dimensional haze-order space and we have studied its models from the point of view of model theory. We have shown that its models are homomorphic to partial orders on a discrete domain. We have also proposed a conservative two-dimensional extension of the theory of one-dimensional space, called independent combination, in which the evaluation of two-dimensional operators is reduced to the evaluation of projected one-dimensional operators over two coordinate copies of the one-dimensional theory. This result is generalizable for k dimensional spaces provided that the conditions of independent composition are preserved when adding dimensions. The developed formalism is strictly qualitative with a built-in concept for imprecision. The account of imprecision within the representation language has allowed us to formalize the notion of granularity in spatial representation.

Next, we have developed algorithms for reasoning about relations in haze-order space. In particular we have developed efficient algorithms for determining the consistency of a set of haze-order relations, and deducing new relations from those that are already known. In addition we have defined a quantitative index structure that can support constant-time retrievals. Our algorithms make use of a data structure called haze-order graph which trades space for efficiency. We have also investigated adaptations of the path consistency algorithm for haze-order constraint networks and we have demonstrated that path consistency cannot compute minimal relations. Although incomplete, the development of path consistency based algorithms was motivated by pragmatic reasons since path consistency is proven to be an effective inconsistency detection technique for certain datasets. The developed algorithms have been implemented and their performance has been experimentally evaluated. The contribution in this part of the dissertation has been the development of a computational model for haze-order reasoning and the development of a testbed for evaluating alternative algorithms.

Another contribution is the development of a spatial data model which facilitates the representation of and reasoning with various forms of qualitatively and quantitatively incomplete spatial information, including indeterminate objects, multiple scales and granularity. Representation of incomplete spatial information is accomplished through a spatial constraint language based on haze-order relations. For pragmatic adequacy purposes, we have extended the qualitative constraint language of space with a quantitative component which allows us to relate indeterminate objects to landmarks. Finally, we have identified four reasoning tasks that are addressed during query processing in this representation and we have offered efficient processing algorithms for each one of them. We have integrated our spatial representation model with an object-oriented data model by exploiting the meta-modeling facilities of the latter. The resulting spatial data model has unique features that make it applicable to a wide range of applications involving imprecise dimensional data such as temporal databases, genome databases and financial databases.

Finally, we have presented four applications of the proposed spatial data model and haze-order reasoning techniques in the fields of geographic information systems and genome informatics. Specifically, we have outlined a solution to map generalization based on our model's ability to support multiple scales. We have illustrated the role of granularity in querying imprecise data. Haze-order semantics have been applied to model regions with indeterminate boundaries in geographic information systems. We have also demonstrated the use of the haze-order language in specifying experimental data in the Human Genome Project in the context of order inferencing and map assembly operations.

7.2 Future Work

The results of this work can be extended in different directions. We have shown a general and extensible framework that puts into a perspective all the relevant issues, (i.e., representation, reasoning and management) towards spatial data support in advanced information processing tasks. This work has touched upon imprecise, propositional spatial information. The same framework can be used for addressing images and image content information, or other types of dimensional non-spatial data, such as temporal or scientific data.

A starting point in this work was the selection of an appropriate ontology of space. The particular choice has influenced the solutions in the later steps. Extending the basis of the developed framework with more ontologies of space will have double impact. First, this will increase the scope of potential application of the framework. Second and more important, it will advance the knowledge level specification of the spatial domain. A new ontology of a subject matter, increases the terminology about the subject, but does not necessarily add any knowledge about it. The specification of an ontology needs to state axioms in order to constrain the interpretation of the defined terms. Going into the "intelligence agents" era, extended ontological bases increase the potentials of knowledge sharing and knowledge-based systems interoperation. As candidate ontologies of space to be studied under this light, we consider the topological, metric and linguistic space.

As far as spatial reasoning in databases is concerned, we believe that constraint databases is a prominent direction to follow. Earlier work in constraint databases [KKR90] and temporal constraint databases [Kou94b] have produced interesting query processing complexity results. The same stream of work may be extended for the case of spatial constraint object bases, i.e., spatial knowledge bases. Practical reasoning using haze order constraints requires the merging of qualitative and quantitative methods (the "poverty of qualitative reasoning"). As we pointed out in Chapter 5, query processing under such requirements is heavily based on a variable elimination algorithm. The development of one such algorithm is in our current research agenda. Another extension of the haze-order language that we are interested in exploiting is the accommodation of Datalog [BCW93] type rules.

The application of haze-orders and the integrated data model in multi-scale and granularity supporting geographic information systems, opens a number of issues which need to be looked into further. In particular, we intend to extend the current results to model objects with arbitrary shapes. Moreover, we plan to define a library of methods for geometric and scale-related operations. The latter is related to the implementation plans that were briefly discussed in Section 6.5. An alternative direction in our implementation plan is to use extended relational technology to replace Telos for production oriented applications.

Haze-order constraints are shown to be a representation model that can deal with imprecision and uncertainty of dimensional (geographic, temporal, genomic, financial) data. In its applications to a specific data domain, one early decision to be made is the selection of a "good" degree of precision, or in other words the scope of the haze relation. There is top-down and a bottom-up approach to give an answer to this question. In the top-down method, a continuous data domain such as the geographic space is discretized and hence the distinguisability of objects will be depenedent to the discritization operation. In the bottom-up approach, the data contain the imprecision as created (e.g., a photo-intensity map). The problem then is to find and apply an appropriate threshold value. Finding the discretization method or defining a threshold value for specific application domains is an interesting problem that is left for applied follow-up work.

User interface and visualization issues have not been considered in this dissertation. Visualization of the spatial information has always been an important issue. Thriving multimedia technologies and world wide web applications, make these issues even more important. For a long while, a logic-based interaction language with a spatial knowledge base was our preferred alternative. Not any longer. Its usability insufficiencies and the visualization requirements suggest a graphical alternative [CM93].

7.3 Conclusions

The guiding principle throughout this dissertation has been that spatial information management depends on the assumed ontology of space rather than the application domain. Consequently, we have proposed a three-layered framework for addressing spatial knowledge management in which the selection of a spatial ontology constitutes the first (bottom) layer. The second layer which is tightly connected to the selected ontology, consists of efficient algorithms that support the most common operations in this context. The third layer provides the glue for sticking together the parts of the architecture. This is an extensible data model which combines classic data model facilities with meta modeling features so that the ontology's primitives are expressed in it.

The approach just proposed is horizontally extensible at all layers. This proposal draws from our work in knowledge base management systems [MCP+95], [Top93], where a similar layered architecture has been proposed for the implementation of "any" knowledge base.

An interesting part of our work is the synthesis of many research areas. Most of our work is motivated by and focused on imprecise, propositional spatial information. Artificial Intelligence and constraint reasoning techniques were employed in the technical chapters of the work, while advanced data modeling techniques were used to interconnect the components. In addition to sound theoretical results and the methodological contribution, this study has attempted to address potential applications and implementation issues.

The final conclusion of this dissertation is that management issues for types of information found in advanced applications, imprecise spatial knowledge in this case, need to place emphasis on the integration of modeling features with rich, well-founded semantics as well as efficient implementations techniques with established good performance.

Appendix A

Algorithms

In this appendix, we present the algorithmic language description of the Tight Haze-Order (THO) and Haze-Index (HI) algorithms introduced in Chapter 4.

A.1 The Tight Haze-Order (THO) Algorithm

Input: A consistent haze-order graph $hG = (V, E)$
Output: A tight haze-order graph.

```
Let R = (V', E') be the graph constructed during the consistency algorithm.
Topologically sort R
Let start and end be two pseudo-points enclosing hG form both ends.
for each v ∈ V do
    prev_lss(v) := prev_leq(v) := start
    next_gtr(v) := next_geq(v) := end
end for
for each h ∈ V' do
    recover order relationships ({<, ◁}) by relating frontier nodes with the adjacent
    hypernodes propagate effects inside h by running a local CSP problem
    for each node v ∈ nodes(h) do
        if (v, {<, ◁}, u) ∈ edges(h) then
            next_gtr(v) := u
        else if (L := {u|(v, <, u) ∈ E} and |L| ≥ 1) then
            next_gtr(v) := topo_closer(L)
        else if ((v, g, u) ∈ E and u ∈ nodes(topo_next(h))) then
            next_geq(v) := s such that ∀t ∈ nodes(h)[(t, <, s) ∈ E and s ∈ nodes(h')
                           where h' = min{h_x is topologically after topo_next(h)}]
        else
            next_geq(v) := u such that u ∈ topo_next(h) and ∃s ∈ nodes(h)[(s, <, u) ∈ E]
            next_gtr(v) := u such that ∀s ∈ nodes(h)[(s, <, u) ∈ E and u ∈ nodes(h'),
                           where h' = min{h_x is topologically after topo_next(h)}]
        end if
    end for
    for each node v ∈ nodes(h) do
        if (u, {<, ◁}, v) ∈ edges(h) then
            prev_lss(v) := u
        else if (L := {u|(u, <, v) ∈ E} and |L| ≥ 1) then
            prev_lss(v) := topo_closer(L)
        else if ((u, g, v) ∈ E and u ∈ nodes(topo_prev(h))) then
            prev_lss(v) := s such that ∀t ∈ nodes(h)[(s, <, t) ∈ E and s ∈ nodes(h')
                           where h' = min{h_x is topologically before topo_prev(h)}]
        else
            prev_leq(v) := u such that u ∈ topo_prev(h) and ∃s ∈ nodes(h)[(u, <, s) ∈ E]
            prev_lss(v) := u such that ∀s ∈ nodes(h)[(s, <, u) ∈ E and u ∈ nodes(h'),
                           where h' = min{h_x is topologically before topo_prev(h)}]
        end if
    end for
end for
refinement phase (see procedure TRef)
```

Procedure: TRef
Input: Algorithm THO continued – the *refinement phase.*
Output: A tight haze-order graph.

```
% Refine the "next" links starting from the last topologically ordered hyper-node going backwards
  hy_node := topo_last(R)
  while all hyper-node are traversed do
      for each node v ∈ hy_node do
          W is the set of nodes pointing to v via next_gtr links
          for each node n ∈ W do
              if not n_ordered(n) then
                  for each g-neighbor of n, u do
                      if not n_ordered(u) then
                          u.next_geq := n.next_gtr
                      end if
                  end for
              end if
          end for
      end for
      hy_node := topo_prev(hy_node)
  end while
% Refine the "previous" links starting from the first topologically ordered hyper-node going forward
  hy_node := topo_first(R)
  while all hyper-nodes are traversed do
      for each node v ∈ hy_node do
          W is the set of nodes pointing to v via prev_lss links
          for each node n ∈ W do
              if not p_ordered(n) then
                  for each g-neighbor of n, u do
                      if not p_ordered(u) then
                          u.prev_leq := n.prev_lss
                      end if
                  end for
              end if
          end for
      end for
      hy_node := topo_next(hy_node)
  end while
```

A.2 The Haze-Index (HI) Algorithm

Input: A tight haze-order graph and a resolution parameter.
Output: A quantitative representation of a qualitative haze-order graph.

$E' := E \cup Next \cup Prev$;
for each node $v \in V$ **do**
    $alist(v) := \{\}$
end for
for each edge $e = (v, l, u) \in E'$ **do**
    if $u \in adjacents(v)$ **then**
        $label(v, u) := label(v, u) \circ l$
    else
        add u in $adjacents(v)$ with $label(v, u) := l$
    end if
end for
k is the resolution parameter
for each node $v \in V$ **do**
    $status(v) := closed$
    $hi(v) = \{\}$
end for
$v := start$; status(v):=open
while $P \neq \oslash$ **do**
    for each edge (v, l, u) **do**
        if status(u)=closed **then**
            status(v):=open;
            push_in_queue(Q,u)
        end if
        $hi(u) := hi(u) \cup \{hi(v) \uplus weight(l)\}$
    end for
    v:=pop(Q)
    $status(v) := done$
end while

Appendix B

Transitivity Tables

Figure B.1 displays the derivations of part of the multiplication table for HO_N defined in Figure 4.9. Index (i, j) refers to row i and column j of the table at the bottom of Fig. 4.9.

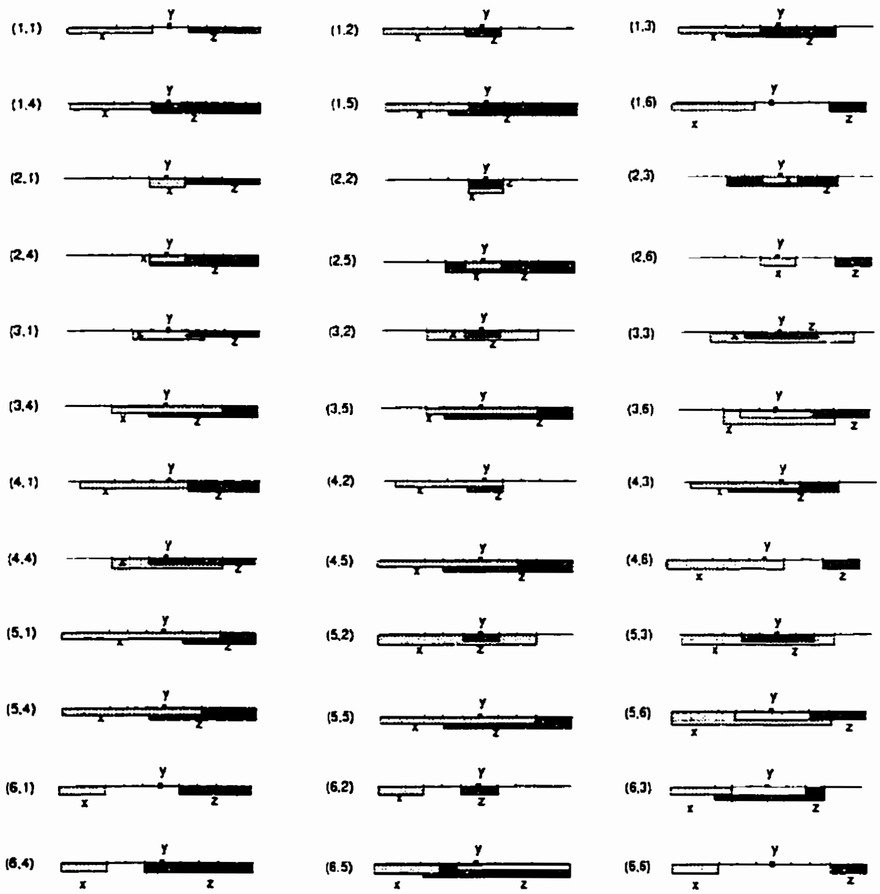


Figure B.1: Entries of the HO_N transitivity table

Appendix C

Canonical Relations

In this appendix, we present a set of binary spatial relations defined between two-dimensional rectangular regions, called *canonical relations*. A canonical relation can be translated into a conjuctive epxression involving haze and order relations on point arguments.

Some topological relations defined in 5.4.2, such as `contain`, `inside` and `equal`, are already in the canonical form. The rest of the topological relations are transformed to canonical relations by conjoining them with a directional constraint, called *directional inclination*. The following cases identify the type of directional inclination associated to each non-canonical topological relation. Table C.1 list the derived canonical relations.

case 1: The two regions are far apart. Then the directional relation between them is derived by the directional relation between designated points (the closest) which depends on the relation.

case 2: The two regions are close, i.e., in a contact relation (either `tangent_i` or `overlap`). Then the above principle is variated as follows: the directional relation between the regions is based on their most diametrical points.

case 3: The two regions are in a containment relation, then the one is bigger than the other. The characterization of the direction makes sense only if there is tangency form inside. The two regions agree in at least one side so their directional inclination is desided by the side of disagreement.

topological relation	inclination	canonical relation	expression
disjoint(a,b)	W_O(a_S, b_E)	disjoint_west(a,b)	$a_{S_X} - b_{E_X} \geq 2g$
	E_O(a_E, b_S)	disjoint_east(a,b)	$b_{S_X} - a_{E_X} \geq 2g$
	N_O(a_S, b_E)	disjoint_north(a,b)	$b_{S_Y} - a_{E_Y} \geq 2g$
	S_O(a_E, b_S)	disjoint_south(a,b)	$a_{S_Y} - b_{E_Y} \geq 2g$
tangent$_i$(a,b)	W_O(a_E, b_S)	tangent$_i$_west(a,b)	$0 \leq a_{S_X} - b_{E_X} < 2g$
	E_O(a_S, b_E)	tangent$_i$_east(a,b)	$0 \leq b_{S_X} - a_{E_X} < 2g$
	N_O(a_E, b_S)	tangent$_i$_north(a,b)	$0 \leq a_{S_Y} - b_{E_Y} < 2g$
	S_O(a_S, b_E)	tangent$_i$_south(a,b)	$0 \leq b_{S_Y} - a_{E_Y} < 2g$
overlap(a,b)	W_O(a_E, b_S)	overlap_west(a,b)	$a_{S_X} < b_{S_X} < a_{E_X}$
	E_O(a_S, b_E)	overlap_east(a,b)	$b_{S_X} < a_{S_X} < b_{E_X}$
	N_O(a_E, b_S)	overlap_north(a,b)	$b_{S_Y} < a_{S_Y} < b_{E_Y}$
	S_O(a_S, b_E)	overlap_south(a,b)	$a_{S_Y} < b_{S_Y} < b_{E_Y}$
inside_tangent(a,b)	NW_1(a_S, b_S)	inside$_t$_NW	$-g < a_{E_X} - b_{E_X} < g \wedge$ $-g < a_{E_Y} - b_{E_Y} < g \wedge$ $b_{S_X} > a_{S_X} \wedge$ $b_{S_Y} > a_{S_Y}$
	W_O(a_S, b_S)	inside$_t$_SW	$-g < a_{E_X} - b_{E_X} < g \wedge$ $-g < a_{S_Y} - b_{S_Y} < g \wedge$ $b_{S_X} > a_{S_X} \wedge$ $b_{E_Y} < a_{E_Y}$
	E_O(a_E, b_E)	inside$_t$_NE	$-g < a_{S_X} - b_{S_X} < g \wedge$ $-g < a_{E_Y} - b_{E_Y} < g \wedge$ $b_{E_X} < a_{E_X} \wedge$ $b_{S_Y} > a_{S_Y}$
	SE_1(a_E, b_E)	inside$_t$_SE	$-g < a_{S_X} - b_{S_X} < g \wedge$ $-g < a_{S_Y} - b_{S_Y} < g \wedge$ $b_{E_X} < a_{E_X} \wedge$ $b_{E_Y} < a_{E_Y}$
	W_O(a_S, b_S)	inside$_t$_west	$a_{S_Y} < b_{S_Y} < b_{E_Y} < a_{S_Y} \wedge$ $-g < a_{E_X} - b_{E_X} < g \wedge$ $a_{S_X} < b_{S_X}$
	E_O(a_E, b_E)	inside$_t$_east	$a_{S_Y} < b_{S_Y} < b_{E_Y} < a_{S_Y} \wedge$ $-g < a_{S_X} - b_{S_X} < g \wedge$ $a_{E_X} > b_{E_X}$
	N_O(a_S, b_S)	inside$_t$_north	$a_{S_X} < b_{S_X} < b_{E_X} < a_{S_X} \wedge$ $-g < a_{E_Y} - b_{E_Y} < g \wedge$ $a_{S_Y} < b_{S_Y}$
	S_O(a_E, b_E)	inside$_t$_south	$a_{S_X} < b_{S_X} < b_{E_X} < a_{S_X} \wedge$ $-g < a_{S_Y} - b_{S_Y} < g \wedge$ $a_{E_Y} > b_{E_Y}$
contain_tangent(a,b)	analogous to inside_tangent		

E_O$(p,q) \equiv p_X < q_X$
W_O$(p,q) \equiv p_X > q_X$
N_O$(p,q) \equiv p_Y > q_Y$
S_O$(p,q) \equiv p_Y < q_Y$
NW_1$(p,q) \equiv p_X > q_X \;\wedge\; p_Y > q_Y$
SE_1$(p,q) \equiv p_X < q_X \;\wedge\; p_Y < q_Y$

Table C.1: Canonical spatial relations

Appendix D

Map Example

The tables below represent the map $m5$ of Figure 6.1 using a constraint database in the spitit of [KKR90]. This is a more convenient form for displaying instance data than the representation proposed in Section 5.3. Following the definition os Section 5.2.4, the representation makes use of universal and existential variables as in [Kou93]. Universal variables appear only in an object's local constraint and they are enclosed in a $U()$ function term. Existensial variables represent values which are not completely known but for which a global constraint holds. Existensial variables are shown in courier font (e.g., `V1X1, XY, ..`). Subscripts $_L$ and $_U$ the lower and upper envelope value for indeterminates.

LANDMARKS

LID	MAP	TYPE	PARAMETERS	CONSTRAINTS
A	m5	rect	$\{a_1, a_2, a_3, a_4\}$	$\{a_1 < a_3,\ a_2 < a_4\}$
B	m5	rect	$\{b_1, b_2, b_3, b_4\}$	$\{b_1 < b_3,\ b_2 < b_4\}$
C	m5	rect	$\{c_1, c_2, c_3, c_4\}$	$\{c_1 < c_3,\ c_2 < c_4\}$
D	m5	rect	$\{d_1, d_2, d_3, d_4\}$	$\{d_1 < d_3,\ d_2 < d_4\}$
E	m5	rect	$\{e_1, e_2, e_3, e_4\}$	$\{e_1 < e_3,\ e_2 < e_4\}$
F	m5	rect	$\{f_1, f_2, f_3, f_4\}$	$\{f_1 < f_3,\ f_2 < f_4\}$
H	m5	pnt	$\{h_1, h_2\}$	$\{h_1 < h_2\}$

INDETERMINATES

I_ID	MAP	TYPE	PARAMETERS	GRAIN	X_CONSTR	Y_CONSTR
V1	m5	rect	{V1X1,V1X2 V1Y1,V1Y2}	$g_1 = 4$	$\{U(\mathrm{V1X2}) - U(\mathrm{V1X1}) = 6,$ $\mathrm{V1X1}_L \leq U(\mathrm{V1X1}) \leq \mathrm{V1X1}_U,$ $\mathrm{V1X2}_L \leq U(\mathrm{V1X2}) \leq \mathrm{V1X2}_U\}$	$\{U(\mathrm{V1Y2}) - U(\mathrm{V1Y1}) = 6,$ $\mathrm{V1Y1}_L \leq U(\mathrm{V1Y1}) \leq \mathrm{V1Y1}_U,$ $\mathrm{V1Y2}_L \leq U(\mathrm{V1Y2}) \leq \mathrm{V1Y2}_U\}$
V2	m5	rect	{V2X1,V2X2 V2Y1,V2Y2}	$g_2 = 1$	$\{U(\mathrm{V2X2}) - U(\mathrm{V2X1}) = 4,$ $\mathrm{V2X1}_L \leq U(\mathrm{V2X1}) \leq \mathrm{V2X1}_U,$ $\mathrm{V2X2}_L \leq U(\mathrm{V2X2}) \leq \mathrm{V2X2}_U\}$	$\{U(\mathrm{V2Y2}) - U(\mathrm{V2Y1}) = 4,$ $\mathrm{V2Y1}_L \leq U(\mathrm{V2Y1}) \leq \mathrm{V2Y1}_U,$ $\mathrm{V2Y2}_L \leq U(\mathrm{V2Y2}) \leq \mathrm{V2Y2}_U\}$
V3	m5	rect	{V3X1,V3X2 V3Y1,V3Y2}	$g_2 = 1$	$\{U(\mathrm{V3X2}) - U(\mathrm{V3X1}) = 4,$ $\mathrm{V3X1}_L \leq U(\mathrm{V3X1}) \leq \mathrm{V3X1}_U,$ $\mathrm{V3X2}_L \leq U(\mathrm{V3X2}) \leq \mathrm{V3X2}_U\}$	$\{U(\mathrm{V3Y2}) - U(\mathrm{V3Y1}) = 4,$ $\mathrm{V3Y1}_L \leq U(\mathrm{V3Y1}) \leq \mathrm{V3Y1}_U,$ $\mathrm{V3Y2}_L \leq U(\mathrm{V3Y2}) \leq \mathrm{V3Y2}_U\}$
X	m5	pnt	{XX,XY}	$g_3 = 3$	$\mathrm{XX}_L \leq U(\mathrm{XX}) \leq \mathrm{XX}_U,$	$\mathrm{XY}_L \leq U(\mathrm{XY}) \leq \mathrm{XY}_U,$

MAPS

M_ID	GRAIN	SCALE	X_CONSTR	Y_CONSTR
m5	$g_0 = 1$	s_1	CX_1	CY_1

$$
\begin{aligned}
CX_1 = \{\ & a_1 = 1, a_3 = 20, b_1 = 20, b_3 = 26, c_1 = 10, c_3 = 13,\\
& d_1 = 26, d_3 = 30, e_1 = 0, e_3 = 6, f_1 = 25, f_3 = 30, h_1 = 20,\\
& \mathrm{V1X1}_U - \mathrm{V1X1}_L \leq 2g_1,\ \mathrm{V1X2}_U - \mathrm{V1X2}_L \leq 2g_1,\\
& \mathrm{V1X1}_L - a_1 > g_1,\ c_1 - \mathrm{V1X1}_U > g_1,\ \mathrm{V1X2}_L - c_1 \leq 2g_1 \lor c_1 - \mathrm{V1X2}_L \leq 2g_1,\ a_3 - \mathrm{V1X2}_U > g_1,\\
& \mathrm{V2X1}_U - \mathrm{V2X1}_L \leq 2g_2,\ \mathrm{V2X2}_U - \mathrm{V2X2}_L \leq 2g_2,\\
& \mathrm{V2X1}_L - c_3 > g_2,\ h_1 - \mathrm{V2X1}_U > g_0,\ \mathrm{V2X2}_L - h_1 > g_0,\ d_1 - \mathrm{V2X2}_U > g_1,\\
& \mathrm{V3X1}_U - \mathrm{V3X1}_L \leq 2g_2,\ \mathrm{V3X2}_U - \mathrm{V3X2}_L \leq 2g_2,\\
& \mathrm{V3X1}_L - e_3 > g_2,\ \mathrm{V3X1}_U - c_1 \leq 2g_2 \ \lor\ c_1 - \mathrm{V3X1}_U \leq 2g_2,\\
& \mathrm{V3X2}_L - c_3 \leq 2g_2 \ \lor\ c_3 - \mathrm{V3X2}_L \leq 2g_2, h_1 - \mathrm{V3X2}_U > g_2\\
& \mathrm{XX}_U - \mathrm{XX}_L \leq 2g_3,\\
& \mathrm{XX}_L - b_1 > g_3,\ b_3 - \mathrm{XX}_U \leq 2g_3 \ \lor\ \mathrm{XX}_U - b_3 \leq 2g_3\ \}
\end{aligned}
$$

$$
\begin{aligned}
CY_1 = \{\ & a_2 = 30, a_4 = 40, b_2 = 30, b_4 = 34, c_2 = 14, c_4 = 20,\\
& d_2 = 10, d_4 = 14, e_2 = 0, e_4 = 4, f_2 = 0, f_4 = 4, h_2 = 20,\\
& \mathrm{V1Y1}_U - \mathrm{V1Y1}_L \leq 2g_1,\ \mathrm{V1Y2}_U - \mathrm{V1Y2}_L \leq 2g_1,\\
& \mathrm{V1Y1}_L - c_4 > g_1,\ a_2 - \mathrm{V1Y1}_U > g_1,\ \mathrm{V1Y2}_L - b_4 \leq 2g_1 \ \lor\ b_4 - \mathrm{V1Y2}_L \leq 2g_1,\ a_4 - \mathrm{V1Y2}_U > g_1,\\
& \mathrm{V2Y1}_U - \mathrm{V2Y1}_L \leq 2g_2,\ \mathrm{V2Y2}_U - \mathrm{V2Y2}_L \leq 2g_2,\\
& \mathrm{V2Y1}_L - f_4 > g_2,\ d_2 - \mathrm{V2Y1}_U > g_0,\\
& \mathrm{V2Y2}_L - d_2 > g_0,\ d_4 - \mathrm{V2Y2}_U > g_1,\\
& \mathrm{V3Y1}_U - \mathrm{V3Y1}_L \leq 2g_2,\ \mathrm{V3Y2}_U - \mathrm{V3Y2}_L \leq 2g_2,\\
& \mathrm{V3Y1}_L - e_2 \leq 2g_2 \ \lor\ e_2 - \mathrm{V3Y1}_L \leq 2g_2,\ e_4 - \mathrm{V3Y1}_U > g_2,\\
& \mathrm{V3X2}_L - e_4 \leq 2g_2 \ \lor\ e_4 - \mathrm{V3X2}_L \leq 2g_2,\ d_2 - \mathrm{V3X2}_U > g_2\ \mathrm{XY}_U - \mathrm{XY}_L \leq 2g_3,\\
& \mathrm{XY}_L - c_4 > g_3,\ b_2 - \mathrm{XX}_U - b_4 > g_3\ \}
\end{aligned}
$$

Bibliography

[ABJN85] D. Ayala, P. Brunet, R. Juan, and I. Navazo. Object Representation by Means of Nonminimal Division Quadtrees and Octrees. *ACM Transactions on Graphics*, 4(1):41–59, 1985.

[All83] J.F. Allen. Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, 26(11):832–843, November 1983.

[AOG+88] L. Anderson, H. Ohkawa, J. Gjovaag, D. Maier, and S. Shulman. Representing CSG Solids Using a Logic-Based Object Data Model. Technical Report CS/E 88-039, Oregon Graduate Center, 1988.

[AS86] D. Abel and J. Smith. A Relational GIS Database Accomodating Independent Partitionings of the Region. In *Proceedings of the 2nd International Symposium on Spatial Data Handling*, pages 213–224, 1986.

[BCW93] M. Baudinet, J. Chomicki, and P. Wolper. Temporal Deductive Databases. In *Temporal Databases: Theory, Design, and Implementation*, pages 294–320. Benjamin/Cummings, 1993.

[Ben94] Brandon Bennett. Spatial Reasoning with Propositional Logics. In *Proceedings of 4th Conference on Prinicіples of Knowledge Representation and Reasoning (KR94)*, pages 52-61, 1994.

[BKSS90] N. Beckman, H.-P. Kriegel, R. Schneider, and B. Seeger. The R* tree: An Efficient and Robust Access Method for Points and Rectangles. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 323–331, 1990.

[Bri77] Jane Bridge. *Beginning Model Theory.* Cladendon Press, Oxford, 1977.

[CCMSP93] E. Ciapessoni, E. Corsetti, A. Montanari, and P. San Pietro. Embedding Time Granularity in a Logical Specification Language for Synchronous Real-time Systems. *Science of Computer Programming*, 20:141–171, 1993.

[CCR92] Z. Cui, A. Cohn, and D. Randell. Qualitative Simulation Based on a Logical Formalism of Space and Time. In *Proceedings of AAAI92*, 1992.

[CDF+82] A. Chan, S. Danberg, S. Fox, W.-T. Lin, A. Nori, and D. Ries. Storage and Access Structures to Support a Semantic Data Model. In *Proceedings of the 8th VLDB Conference*, pages 308–316, 1982.

[CF81] N.S. Chang and K.S. Fu. Picture Query Languages for Pictorial Database Systems. *Computer*, 14(11):23–33, 1981.

[CJL89] S.-K. Chang, E. Jungert, and Y. Li. The Design of the Pictorial Databases Based upon the Theory of Symbolic Projections. In *Proceedings of the 1st Symposium on the Design and the Implementation of Large Spatial Databases (SSD'89)*, pages 303–323, 1989.

[CK77] C.C. Chang and H.J. Keisler. *Model Theory*. North Holland, 1977.

[CK81] S.-K. Chang and T. Kunii. Pictorial Database Systems. *Computer*, 14(11):13–21, 1981.

[CL88] S.-K. Chang and Y. Li. Representation of Multi-Resolution Symbolic and Binary Pictures Using 2DH Strings. In *Proceedings of IEEE Workshop on Language for Automation*, pages 190–195, 1988.

[Cla85] B.L. Clarke. Individuals and Points. *Norte Dame Journal of Formal Logic*, 26(1), 1985.

[CLR91] T. Cormen, C. Leiserson, and R. Rivest. *Inroduction to Algorithms*. MIT Press, 1991.

[CM93] P. Consens and A. Mendelzon. Hy+: A Hygraph-based Query and Visualization System. In *Proceedings of ACM SIGMOD Conference*, pages 511–516, 1993.

[Coh95] Anthony Cohn. A Hierarchical Representation of Qualitative Shape based on Connection and Convexity. In *Proceedings of COSIT-95*, pages 311–326, 1995.

[CR87] J. Clifford and A. Rao. A Simple, General Structure for Temporal Domains. In *Proceedings of the IFIP TC 8/WG 8.1 Working Conference on Temporal Aspects in Information Systems*, pages 17–28, 1987.

[CSY87] S.-K. Chang, Q.-Y. Shi, and C.-W. Yan. Iconic Indexing by 2-D Strings. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, PAMI-9(3):413–427, 1987.

[Cui94] Zhan Cui. Using Interval Logic for Order Assembly. In *Proceedings of the 2nd International Conference on Intelligent Systems for Molecular Biology*, pages 103–111, 1994.

[Dav85] Ernest Davis. The MERCATOR Representation of Spatial Knowledge. In *Proceedings of IJCAI-85*, pages 295–301, 1985.

[Dav86] Ernest Davis. *Representing and Acquiring Geographic Knowledge*. Pitman and Morgan Kaufmann Publishers Inc., 1986.

[Dav87] Ernest Davis. Constraint Propagation with Interval Labels. *Artificial Intelligence*, 32:281–331, 1987.

[Dea89] T. Dean. Using Temporal Hierarchies to Efficiently Maintain Large Temporal Databases. *Journal of ACM*, 36(4):687–718, 1989.

[DK+94] D. DeWitt, N. Kabra, , J. Luo, P. Jignesh, and J-B. Yu. Client-server paradise. In *Proceedings of the 20th VLDB Conference*, 1994.

[DM87] T. Dean and D.V. McDermott. Temporal Data Base Management. *Artificial Intelligence*, 32:1–55, 1987.

[DMP89] R. Dechter, I. Meiri, and J. Pearl. Temporal Constraint Networks. In *Proceedings of 1st Knowledge Representation and Reasoning Conference*, pages 83–93, 1989.

[DMP91] R. Dechter, I. Meiri, and J. Pearl. Temporal Constraint Networks. *Artificial Intelligence*, 49:61–95, 1991.

[DOE92] DOE92 .United States Department of Energy. Primer on molecular genetics. The DOE Human Genome 1991-92 Program Report, 1992.

[Dut89] Soumitra Dutta. Qualitative Spatial Reasoning: A semi-quantitative Approach Using Fuzzy Logic. In *Proceedings of the First Symposium on the Design and the Implementation of Large Spatial Databases (SSD'89)*, pages 345–364, 1989.

[EF91] M. Egenhofer and R. Franzosa. Point-set Topological Binary Relations. *International Journal of Geographical Information Systems*, 5(2):161–174, 1991.

[Ege89] Max Egenhofer. A Formal Definition of Binary Topological Relationships. In *Proceedings of the First Symposium on the Design and the Implementation of Large Spatial Databases (SSD'89)*, pages 457–472, 1989.

[Ege91] Max Egenhofer. Reasoning about Binary Topological Relations. In *Second Symposium on Large Spatial Databases (SSD'91)*, pages 143–160, 1991.

[Ege94] Max Egenhofer. Spatial SQL: A Query and Presentation Language. *IEEE Transactions on Data and Knowledge Engineering*, 6(1):86–95, 1994.

[End72] Herbert Enderton. *A Mathematical Introduction to Logic.* Academic Press, 1972.

[Fal88] Christos Faloutsos. Gray Codes for Partial Match and Range Queries. *IEEE Trans. on Software Engineering*, 14(10):1381–1393, 1988.

[Fin93] Marcelo Finger. *Changing the Past: Database Applications of Two-Dimensional Temporal Logics.* PhD thesis, Department of Computing, Imperial College, University of London, 1993.

[Fis85] Peter Fishburn. *Interval Orders and Interval Graphs: A Study of Partially Ordered Sets.* John Wiley & Sons, 1985.

[Fle87] Margaret Fleck. Representing Space for Practical Reasoning . In *Procedings of IJCAI-87*, pages 728–730, 1987.

[FNF87] K. Forbus, P. Nielsen, and B. Faltings. Qualitative Kinematics: A Framework. In *Proceedings of IJCAI-87*, pages 430–435, 1987.

[For80] K.D. Forbus. Spatial and Qualitative Aspects of Reasoning. In *Proceedings of AAAI-80*, 1980.

[FR89] C. Faloutsos and S. Roseman. Fractals for Secondary Key Retrieval. Technical Report UMIACS-TR-89-47, CS-TR-2242, University of Maryland, 1989.

[FR93] C. Freksa and R. Roehrig. Dimensions of Qualitative Spatial Reasoning. In *Workshop on Qualitative Reasoning and Decision Technologies*, 1993.

[Fra91] Andrew Frank. Qualitative Spatial Reasoning with Cardinal Directions. International Journal of Geographic Information Systems (to appear), 1995.

[Fre87] Michael Freeston. The BANG file: A New Kind of Grid File. In *Proceedings of the ACM-SIGMOD Inetrnational Conference on Management of Data*, 1987.

[Fre91] Karen Frenkel. The Human Genome Project and Informatics. *CACM*, 34(11):41–51, 1991.

[Fre92a] Christian Freksa. Temporal reasoning based on semi-intervals. *Artificial Intelligence*, 54:199–227, 1992.

[Fre92b] Christian Freksa. Using Orientation Information for Qualitative Spatial Reasoning. In *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space (COSIT-92)*, pages 162–178, 1992.

[Fre94] Christian Freksa. Fuzzy systems in ai. In *Fuzzy Systems in Computer Science*. Vieweg, 1994.

[Fre82] Eugene Freuder. A Sufficient Condition for Backtrack-Free Search. *Journal of ACM*, 21(11):958–966, 19982.

[GA89] M. Ghallab and M.A. Alaoui. Managing Efficiently Temporal Relations through Indexed Spanning Trees. In *Proceedings of IJCAI-89*, pages 1297–1303, 1989.

[Gar82] Irene Gargantini. An Effective Way to Represent Quadtrees. *Communications of ACM*, 25(12):905–910, 1982.

[GB90] O. Guenther and A. Buchmann. Research Issues in Spatial Databases. *SIGMOD Record*, 19(4):61–68, 1990.

[Gla93a] Janice Glasgow. Representation of Spatial Models for Geographic Information Systems. In *CIKM/ACM Workshop on Advances in Geographic Information Systems*, pages 105–110, 1993.

[Gla93b] Janice Glasgow. The Imagery Debate Revisited: A Computational Perpective. *Computational Intelligence*, 9(4), 1993.

[GO94] T. Gruber and G. Olsen. An Ontology for Engineering Mathematics. In *Proceedings of 4th Conference on Priniciples of Knowledge Representation and Reasoning (KR)*, 1994.

[GP92] J. Glasgow and D. Papadias. Computational Imagery. *Cognitive Science*, 16:355–394, 1992.

[GPP95] A. Gringi, D. Papadias, and C. Papadimitriou. Topological inference. In *Proceedings of IJCAI-95*, 1995.

[GR93] J. Guidi and T. Roderick. Inference of order in genetic systems. In *First International Conference on Intelligent Information Systems for Molecular Biology*, pages 163–171, 1993.

[GRS94] N. Goodman, S. Rozen, and L. Stein. Requirements for a query language in the mapbase genome-mapping database. In R. Ranakrishnan, editor, *Applications of Deducutive Databases*. Kluwer, 1994.

[Gru93] Tom Gruber. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.

[GS82] A. Guttman and M. Stonebraker. Using a Relational Database Management System for Computer Aided Design Data. Technical Report UCB/ERL M82/37, University of California, Berkeley, 1982.

[GS93] A. Gerevini and L. Schubert. Efficient Temporal Reasoning through Timegraphs. In *IJCAI-93*, pages 648–654, 1993.

[GS95a] A. Gerevini and L. Schubert. Efficient Algorithms for Qualitative Reasoning about Time. *Artificial Intelligence (to appear)*, 1995.

[GS95b] R-H. Guting and M. Schneider. Realm-Based Spatial Data Types: The ROSE Algebra. *The VLDB Journal*, 4:243–286, 1995.

[Gue88] Oliver Guenther. *Efficient Structures for Geometric Data Management*. Lecture Notes in Computer Science, 337. Springer Verlag, 1988.

[Gue89] H.W. Guesgen. Spatial Reasoning based on Allen's Temporal Logic. Techical Report TR-89-049, ICSI, Berkeley, 1989.

[Gut84] Antonin Guttman. R-Trees: A Dynamic Index Structure For Spatial Searching. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, pages 47–57, 1984.

[Gut88] Ralf H. Guting. Geo-Relational Algebra: A Model and Query Language for Geometric Database Systems. In *Proceedings of EDBT'88*, pages 506–527, 1988.

[Gut94] Ralf H. Guting. An Introduction to Spatial Database Systems. *The VLDB Journal*, 3(4):357–399, 1994.

[Had96] Thanasis Hadzilacos. Modeling Habitats. Personal Communication, 1996.

[Hay85] Pat Hayes. The Second Naive Physics Manifesto. In J. Hobbs and R. Moore, editors, *Formal Theories of the Commonsense World*. Ablex Publishing, 1985.

[HB94] E. Harley and T. Bonner. A Flexible Appoach to Genome Map Assembly. In *Proceedings of the 2nd International Conference on Intelligent Systems for Molecular Biology*, pages 161–169, 1994.

[Heg59] G.W.F. Hegel. *Encyclopedia of Philosophy, (translated and annotated by G.E. Mueller)*. Philosophical Library, New York, 1959.

[Her85] Annette Herskovits. Semantics and Pragmatics of Locative Expressions. *Cognitive Science*, 9:341–378, 1985.

[Her92] Daniel Hernandez. *Qualitative Representation of Spatial Knowledge*. PhD thesis, Faculty of Informatics, Technical University of Munich, 1992.

[Hob85] J. Hobbs. Granularity. In *Proceedings of IJCAI-85*, pages 432–435, 1985.

[Jag90] H.V. Jagadish. Linear Clustering of Objects with Multiple Attributes. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, pages 332–342, 1990.

[JC88] T. Joseph and A. Cardenas. PICQUERY: A High Level Query Language for Pictorial Database Management. *IEEE Transactions on Software Engineering*, 14(5):630–638, 1988.

[Jun88] E. Jungert. Extended Symbolic Projection Used in Knowledge Structure for Spatial Reasoning. In *Proceedings of 4th BPRA Conference on Pattern Recognition*, 1988.

[Jur95] Igor Jurisica. A Similarity-Based Retrieval Tool for Software Repositories. In *IJCAI-95 Workshop on AI and Software Engineering*, 1995.

[Kau85] Henry Kautz. Formalizing Spatial Concepts and the Spatial Language . In the Commonsense Summer Final Report, 1985.

[Kau91] Srephen Kaufman. A Formal Theory of Spatial Reasoning. In *Proceedings of AAAI Spring Symposium on Logical Foundations of Commonsense Reasoning*, pages 92–101, 1991.

[KB87] B.J. Kuipers and Y.-T. Byun. A Qualitative Approach to Robot Exploration and Map-Learning. In *Proceedings of Spatial Reasoning and Multi-Sensor Fusion Workshop*, pages 390–404, 1987.

[KB88] B.J. Kuipers and Y.-T. Byun. A Robust Qualitative Method for Robot Spatial Learning. In *Proceedings of AAAI-88*, pages 774–779, 1988.

[KH92] D Koller and J. Halpern. A logic for approximate reasoning. In *KR-92*, pages 153–163, 1992.

[KKR90] P. Kanellakis, G. Kuper, and P. Revesz. Constraint Query Languages. Long version. A preliminary version appearred in *Proc. of 9th ACM PODS*, 1990.

[KL88] B.J. Kuipers and S.T. Levitt. Navigation and Mapping in Large-Scale Space. *AI Magazine*, pages 25–43, Summer 1988.

[KL91] H. Kautz and P. Ladkin. Integrating Metric and Qualitative Temporal Reasoning. In *Proceedings of AAAI-91*, pages 241–246, 1991.

[Koo89] Johannes Koomen. Localizing Temporal Constraint Propagation. In R. Brachman, H. Levesque, and R. Reiter, editors, *Proceedings of 1st International Conference on Principles of Knowledge Representation and Reasoning*, pages 198–202, 1989.

[Kou92] Manolis Koubarakis. Dense Time and Temporal Constraints with $\neq$. In *Proceedings of KR-92*, pages 24–35, 1992.

[Kou93] Manolis Koubarakis. Representation and Querying in Temporal Databases. In *Proceedings of the 9th Int. Conf. on Data Engineering*, pages 327–334, 1993.

[Kou94a] Manolis Koubarakis. Database Models for Infinite and Indefinite Temporal Information. *Information Systems*, 19(2):141–173, March 1994.

[Kou94b] Manolis Koubarakis. *Foundations of Temporal Databases*. PhD thesis, National Technical University of Athens, 1994.

[KSLT89] D.H. Krantz, P. Suppes, R.D. Luce, and A. Tversky. *Foundations of Measurement*. Academic Press, 1989.

[Kui78] B.J. Kuipers. Modelling Spatial Knowledge. *Cognitive Science*, 2:129–153, 1978.

[Kui86] B.J. Kuipers. Qualitative Simulation. *Artificial Intelligence*, 29:289–338, 1986.

[Kui89] B.J. Kuipers. Qualitative Reasoning: Modelling, Simulation with Incomplete Knowledge. *Automatica*, 1989.

[KW87] A. Kemper and M. Wallrath. An Analysis of Geometric Modeling in Database Systems. *ACM Computing Surveys*, 19(1):47–91, 1987.

[Lad87] Peter Ladkin. Models of Axioms for Time Intervals. In *Proceedings of AAAI-87*, pages 234–239, 1987.

[Lad89] Peter Ladkin. Metric Constraint Satisfaction with Intervals. Technical Report TR-89-038, International Computer Science Center, Berkeley, 1989.

[Lau85] R. Laurini. Graphics Databases Built on Peano Space-Filling Curves. In *Proceedings of EUROGRAPHICS'85 Conference*, pages 327–338, 1985.

[Lev81] H.J. Levesque. *A Formal Treatment of Incomplete Knowledge Bases.* PhD thesis, Dept. of Computer Science, University of Toronto, 1981.

[LG89] D. Lenat and R. Guha. *Building Large Knowledge-Based Systems.* Addison-Wesley, 1989.

[Lig90] G. Ligozat. Weak Representations of Interval Algebras. In *Proceedings of AAAI-90*, pages 715–710, 1990.

[LJ88] N.A. Lorentzos and R.G. Johnson. Extending Relational Algebra To Manipulate Temporal Data. *Information Systems*, 13(3):289–296, 1988.

[LM88a] P. Ladkin and R. Maddux. Representation and Reasoning with Convex Time Intervals. Technical Report KES.U.88.2, Kestrel Institute Technical Report, 1988.

[LM88b] R. Laurini and F. Milleret. Relational Algebra versus Computational Geometry. In Rafanelli, Klensin, and Svensson, editors, *Proceedings of the 4th Int. Conference on Statistical and Scientific Database Management*, pages 291–313, 1988.

[LM94] P. Ladkin and R. Maddux. On Binary Constraint Problems. *Journal of the ACM*, 41(3):435–469, 1994.

[LP81] Tomas Lozano-Perez. Automatic Planning of Manipulator Transfer Movements. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-11(10):681–698, 1981.

[LP83] R. Lorie and W. Plouffe. Complex Objects and Their Use in Design Transaction. In *Proceedings of ACM-SIGMOD Conference on Engineering Design Application*, pages 115–121, 1983.

[LR92] P. Ladkin and A. Reinefeld. Effective Solution of Qualitative Interval Constraint Problems. *Artificial Intelligence*, 1(57):105–124, 1992.

[LR93] L. Latecki and R. Roehrig. Orientation and Qualitative Angle for Spatial Reasoning. In *Proceedings of IJCAI-93*, pages 1544–1549, 1993.

[Lu93] Wei Lu. *A Deductive and Object-Oriented Approach for Spatial Databases.* PhD thesis, Simon Fraser University, 1993.

[Mac77] A.K. Mackworth. Consistency in Networks of Relations. *Artificial Intelligence*, 8(1):99–118, 1977.

[MB83] J. Malik and T.O. Binford. Reasoning in Time and Space. In *Proceedings of IJCAI-83*, pages 343–345, 1983.

[MBJK90] J. Mylopoulos, A. Borgida, M. Jarke, and M. Koubarakis. Telos: A Language for Representing Knowledge About Information Systems. *ACM Transactions on Office Information Systems*, 8(4):325–362, 1990.

[McD80] Drew McDermott. A Theory of Metric Spatial Inference. In *Proceedings of AAAI-80*, pages 246–248, 1980.

[McD82] D.V. McDermott. A Temporal Logic for Reasoning About Processes and Plans. *Cognitive Science*, 6:101–155, 1982.

[MCP+95] J. Mylopoulos, V. Chaudhri, D. Plexousakis, A. Shrufi, and T. Topaloglou. Implementing Knowledge Management Systems . *The VLDB Journal (to appear)*, 1995.

[MD84] Drew McDermott and Ernest Davis. Planning Routes through Uncertain Territory. *Artificial Intelligence*, 22:107–156, 1984.

[Mei91] Itay Meiry. Combining Qualitative and Quantitative Constraints in Temporal Reasoning. In *Proceedings of AAAI-91*, pages 260–267, 1991.

[MF85] A.K. Mackworth and E.C. Freuder. The Complexity of Some Polynomial Network Consistency Algorithms for Constraint Satisfaction Problems. *Artificial Intelligence*, 25(1):65–74, 1985.

[MGD91] D. Manguire, M. Goodchild, and Rhind D. *Geographic Information Systems*. Longman Scientific & Technical and John Wiley & Sons, 1991.

[MH95] A. Mukerjee and D. Hernandez. Representations of Spatial Knowledge. Tutorial in the IJCAI-95 Conference, 1995.

[Mil85] David Miller. A Spatial Representation System for Mobile Robots. In *IEEE International Conference on Robotics and Automation*, pages 122–127, 1985.

[MJ90] A. Mukerjee and G. Joe. A Qualitative Model for Space. In *Proceedings of AAAI-90*, pages 721–727, 1990.

[MK88] Mohan and Kashyap. An Object-Oriented Knowledge Representation for Spatial Information. *IEEE Transactions on Software Engineering*, 14(5):675–681, 1988.

[MMT89] E. Milios, J. Mylopoulos, and J. Tsotsos. Towards Computational Geoscience: Issues of Knowledge Representation. technical report, 1989.

[MO86] Frank Manola and Jack Orenstein. Towards a General Spatial Data Model for an Object-Oriented DBMS. In *Proceedings of the 12th VLDB Conference*, pages 328–335, 1986.

[Mon74] U. Montanari. Networks of Constraints: Fundamental Properties and Applications to Picture Processing. *Information Sciences*, 7:95–132, 1974.

[MP94] C.B. Medeiros and F. Pires. Databases for GIS. *ACM SIGMOD Record*, 23:107–115, 1994.

[MS88] S. Miller and L. Schubert. Using Specialists to Accelerate General Reasoning. In *Proc. of AAAI-88*, pages 161–165, 1988.

[NB94] B. Nebel and H.-J. Burckert. Reasoning about temporal relations: A maximal tractable subclass of allen's interval algebra. In *Proceedings of AAAI-94*, 1994.

[NC91] N.H. Narayanan and B Chandraseckaran. Reasoning Visually about Spatial Interactions. In *Proceedings of IJCAI-91*, 1991.

[New82] Allen Newell. The Knowledge Level. *Artificial Intelligence*, 18(1):87–127, 1982. Also published in the AI Magazine 2(2), 1981.

[NHS84] J. Nievergelt, H. Hinterberger, and K. Sevcik. The Grid File: An Adaptable, Symmetric Multikey File Structure. *ACM Transactions on Database Systems*, 9(1):38–71, 1984.

[Nie89] Jurg Nievergelt. 7 ± 2 Criteria for Accessing and Comparing Spatial Data Structures. In *Proceedings of the First Symposium on the Design and the Implementation of Large Spatial Databases (SSD'89)*, pages 3–27, 1989.

[OM84] J.A. Orenstein and T.H. Merrett. A Class of Data Structures for Associative Searching. In *Proceedings of SIGMOD-84*, pages 181–190, 1984.

[OM88] J. Orenstein and F. Manola. PROBE: Spatial Data Modelling and Query Processing in an Image Database Application. *IEEE Transactions on Software Engineering*, 14(5):611–629, 1988.

[Ore86] Jack Orenstein. Spatial Query Processing in an Object-Oriented Database System. In *Proceedings of SIGMOD-86*, pages 326–333, 1986.

[Ore89] Jack Orenstein. Strategies for Optimizing the Use of Redundancy in Spatial Databases. In *Proceedings of the First Symposium on the Design and the Implementation of Large Spatial Databases (SSD'89)*, pages 115–134, 1989.

[OSD89] B.C. Ooi and R. Sacks-Davis. Query Optimization in an Extended DBMS. In *Proceedings of 3rd International Conference on Foundations of Data Organizations and Algorithms*, pages 48–63, 1989.

[OSDM89] B.C. Ooi, R. Sacks-Davis, and K.J. McDonell. Extending a DBMS for Geographic Applicaations. In *Proceedings of International Conference on Data Engineering*, pages 590–597, 1989.

[Pap94a] Dimitris Papadias. *Relation-based Representations for Spatial Databases.* PhD thesis, National Technical University of Athens, 1994.

[Pap94b] Christos Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[Par95] Jan Paradaens. Spatial Databases: The Final Frontier. In *Proceedings of the International Conference on Database Theory*, pages 14–32, 1995.

[PD95] E. Puppo and G. Dettori. Towards a formal model for multiresolution spatial maps. In *Proc. of the Fourth Symposium on Large Spatial Databases (SSD'95)*, pages 153–169, 1995.

[PS94] D. Papadias and T. Sellis. Qualitative Representation of Spatial Knowledge in Two-Dimensional Space. *The VLDB Journal*, 3(4):479–517, 1994.

[PSS+87] H.-B. Paul, H.-J. Schek, M.H. Scholl, G. Weikum, and U. Deppish. Architecture and Implementation of a Darmstadt Database Kernel System. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 196–207, 1987.

[PTSE94] D. Papadias, Y. Theodoridis, T. Sellis, and M. Egenhofer. Topological Relations in the World of Minimum Bounding Rectangles: A Study with R-trees. In *SIGMOD-95*, 1994.

[PVdBVG94] J. Paradaens, J. Van den Bussche, and D. Van Gucht. Towards a Theory of Spatial Database Queries. In *Proceedings of PODS*, 1994.

[RC89] D. Randell and A. Cohn. Modelling Topological and Metrical Properties in Physical Processes. In *Proceedings of KRR-89*, 1989.

[RCC92a] D. Randell, Z. Cui, and A. Cohn. A Spatial Logic based on Regions and Connection. In *Proceedings of KR-92*, 1992.

[RCC92b] D. Randell, Z. Cui, and A. Cohn. Computing Transitivity Tables. In *Proceedings of CADE11*, 1992.

[Rei80] Ray Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1):81–132, 1980.

[Req80] Aristides Requicha. Representations of Rigit Solids: Theory, Methods and Systems. *ACM Computing Surveys*, 12(4):337–365, 1980.

[RFS88] N. Roussopoulos, C. Faloutsos, and T. Sellis. An Efficient Pictorial Database System for PSQL. *IEEE Transactions on Software Engineering*, 14(5):639–650, 1988.

[RH80] D. Rozenkrantz and H. Hunt. Processing Conjunctive Predicate and Queries. In *VLDB-80*, pages 64–72, 1980.

[RM89] R. Reiter and A. Mackworth. A Logical Framework for Depiction and Image Interpretation. *Artificial Intelligence*, 41:125–155, 1989.

[Rob73] Fred S. Roberts. Tolerance Geometry. *Norte Dame Journal of Formal Logic*, 14(1):68–76, 1973.

[Rob89] Vince Robinson. Specification of Natural Language Concepts for Spatial Information Retrieval, 1989.

[Roe94] Ralf Roehrig. A Theory for Qualitative Spatial Reasoning Based on Order Relations. In *Proceedings of AAAI-94*, pages 1418–1423, 1994.

[Rom90] G.-C. Roman. Formal Specification of Geographic Data Processing Requirements. *IEEE Transactions on Knowledge and Data Engineering*, 2(4):370–380, 1990.

[RS95] P. Rigaux and M. Scholl. Multi-Scale Partitions: Applications to Spatial and Statistical Databases. In *Proceedings of the 4rth Symposium on Large Spatial Databases, SSD'95*, pages 171–183, 1995.

[RV82] A. Requicha and H. Voelcker. Solid Modeling: A Historical Summary and Contemporary Assessment. *IEEE Computer Graphics and Applications*, pages 9–24, March 1982.

[Sam84] Hanan Samet. The Quadtree and Related Hierarchical Data Structures. *ACM Computing Surveys*, 16(2):187–260, 1984.

[Sam89] Hanan Samet. *The Design and Analysis of Spatial Data Structures: Computer Graphics, Image Processing and GIS*. Addison-Wesley, 1989.

[SG88] Y. Shoham and D. Goyal. Temporal Reasoning in Artificial Intelligence. In H. Shrobe, editor, *Exploring Atrificial Intelligence*, pages 419–436. Morgan Kaufmann, 1988.

[Sho85] Yoan Shoham. Naive Kinematics: One Aspect of Shape. In *Proceedings of IJCAI-85*, pages 436–422, 1985.

[SK88] B. Seeger and H.-P. Kriegel. Techniques for Design and Implementation of Efficient Spatial Access Methods. In *Proceedings of the 14th VLDB Conference*, pages 360–371, 1988.

[Slo85] Aaron Sloman. Afterthoughts on Analogical Representations. In Ronald J. Brachman and Hector J. Levesque, editors, *Readings in Knowledge Representation*, pages 432–439. Morgan Kaufmann, 1985.

[SM89] K.S. Shea and R. Macmaster. Cartographic Generalizarion: When and How to Generalize. *Auto-Carto*, 9:56–67, 1989.

[Spa95] SpaceNet95. Workshop. `http://agora.leeds.ac.uk/spacenet/`, 1995.

[SPT87] L. Schubert, M.A. Papalaskaris, and J. Taugher. Accelerating Deductive Inference: Special Methods for Taxonomies, Colours and Times. In N. Cercone and G. McGalla, editors, *The Knowledge Frontier*. Springer-Verlag, 1987.

[SRF87] T. Sellis, N. Roussopoulos, and C. Faloutsos. The R^+-Tree: A Dynamic Index for Multidimensional Objects. In *Proceedings of the 13th VLDB Conference*, pages 507–518, 1987.

[SRG83] M. Stonebraker, B. Rubenstein, and A. Guttman. Application of Abstract Data Type and AbstractIndices to CAD Databases. In *Proceedings of ACM-SIGMOD Conference on Engineering Design Application*, pages 107–113, 1983.

[ST95] A. Shrufi and T. Topaloglou. Query Processing for Knowledge Bases Using Join Indices. In *Proceedings of the 4th International Conference on Information and Knowledge Management (CIKM'95)*, 1995.

[Sta95] Martin Stanley. The Telos Repository User's Manual. University of Toronto, Department of Computer Science, Technical Note, 1995.

[SV91] M. Scholl and A. Voisard. Object-Oriented Database Systems for Geographic Application: An Experiment with O2. In G. Gambrosi, H. Six, and M. Scholl, editors, *Proceedings of the International Workshop on Database Management Systems for Geographic Applications*, 1991.

[SYH94] P. Sistla, C. Yu, and R. Haddad. Reasoning about Spatial Relationships in Picture Retrieval Systems. In *VLDB-94*, 1994.

[Szc86] L.W. Szczerba. Tarski and Geometry. *The Journal od Symbolic Logic*, 51(4):907–912, 1986.

[TCG+93] A. Tansel, J. Clifford, S. Gadia, S. Jajodia, A. Segev, and R. Snodgrass, editors. *Temporal Databases: Theory, Design and Implementation.* Database Systems and Applications Series. Benjamin/Cummings Pub. Co., 1993.

[TIS92] T. Topaloglou, A. Illarramendi, and L. Sbattella. Query Optimization for KBMSs: Temporal, Syntactic and Semantic Transformation. In *Proceedings of the International Conference on Data Engineering*, pages 310–319, 1992.

[TMD92] J. Tierry-Mieg and R. Durbin. ACeDB: A Database for Genomic Information. In *Proceedings of the Genome Mapping and Sequencing Workshop*, 1992.

[Top93] Thodoros Topaloglou. Storage Management for Knowledge Bases. In *Proceedings of the 2nd International Conference on Information and Knowledge Management (CIKM'93)*, 1993.

[Top94a] Thodoros Topaloglou. First Order Theories of Approximate Space. In *AAAI-94 Workshop on Spatial and Temporal Reasoning*, pages 47–53, 1994.

[Top94b] Thodoros Topaloglou. Qualitative Reasoning about Imprecise Spatial Information. In *Proceedings of the 2nd ACM/CIKM Workshop on Advances in GIS*, 1994.

[Top96a] Thodoros Topaloglou. Efficient Algorithms for Qualitative Reasoning about Imprecise Space. In *Proceedings of the 11th Biennial Conference of the Canadian Society for Computational Studies of Intelligence (AI'96)*, 1996.

[Top96b] Topaloglou, T. and Mylopoulos, J. Representing Partial Spatial Information in Databases. In *Proceedings of the 15th International Conference on Conceptual Modeling (ER'96)*, 1996.

[Ube94] M. Ubell. The Montage Extensible Datablade Architecture. In *Proceedings of ACM SIGMOD Conference*, 1994.

[vB89] Peter van Beek. Approximation Algorithms for Temporal Reasoning. In *Proceedings of IJCAI-89*, pages 1291–1296, 1989.

[vB90] Peter van Beek. Exact and Approximate Reasoning About Qualitative Temporal Relations. Technical Report TR 90-29, Department of Computing Science, University of Alberta, August 1990.

[vB91] Johan van Benthem. *The Logic of Time*. Kluwer Academic Publishers, 1991.

[vB92] Peter van Beek. Reasoning About Qualitative Temporal Information. *Artificial Intelligence*, 58:297–326, 1992.

[VK86] M. Vilain and H. Kautz. Constraint Propagation Algorithms for Temporal Reasoning. In *Proceedings of AAAI-86*, pages 377–382, 1986.

[VKvB89] M. Vilain, H. Kautz, and P. van Beek. Constraint Propagation Algorithms for Temporal Reasoning: a Revised Report. In D.S. Weld and J. de Kleer, editors, *Readings in Qualitative Reasoning about Physical Systems*, pages 373–381. Morgan Kaufmann, 1989.

[vOvdB89] P. van Oosterom and J. van den Bos. An Object-Oriented Approach to the Design of Geographic Information Systems. *Computer & Graphics*, 13(4):409–418, 1989.

[VP86] R.E. Valdés-Pérez. Spatio-Temporal Reasoning and Linear Inequalities. Technical Report Memo 875, MIT Artificial Intelligence Laboratory, 1986.

[VP87] R.E. Valdés-Pérez. The Satisfiability of Temporal Constraint Networks. In *Proceedings of AAAI-87*, pages 256–260, 1987.

[VvO92] T. Vijlbrief and P. van Oosterom. The geo system: An extensible gis. In *Proceedings of the International Symposium on Spatial Data Handling*, 1992.

[Wil88] Peter Wilson. Solid Modeling Research and Applications in the U.S.A. In M.J. Wozny, H.W. McLaughin, and J.L. Encarnacao, editors, *Geometric Modeling for CAD Applications*, pages 69–89. Elsevier Sience Publishers (North-Holland), 1988.

[WJ86] M.S. Waterman and Griggs J.R. Interval Graphs and Maps of DNA. *Bulletin of Mathematical Biology*, 48(2):189–195, 1986.

[WJS93] X.S. Wang, S. Jajodia, and V.S. Subrahmanian. Temporal Modules: An Approach Towards Federated Temporal Databases. In *Proceedings of ACM/SIGMOD Conference*, pages 227–236, 1993.

[Zad94] L.H. Zadeh. Fuzzy logic, neural networks and soft computing. *Communication of ACM*, 37(3):77–84, 1994.

[Zan83] Carlo Zaniolo. The Database Language GEM. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, pages 207–218, 1983.

IMAGE EVALUATION TEST TARGET (QA-3)

150mm

6"

APPLIED IMAGE . Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989