Qualitative Constraint Satisfaction Problems: Algorithms, Computational Complexity, and Extended Framework

Weiming Liu

Faculty of Engineering and Information Technology

University of Technology, Sydney

A thesis submitted for the degree of

Doctor of Philosophy

March 2013

CERTIFICATE OF AUTHORSHIP/ORIGINALITY

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Signature of Student

____

Acknowledgements

I would like to thank all the people who have supported this research work over the past three years. My special thanks are due to my supervisor, A/Prof. Sanjiang Li for his extraordinary and valuable guidance, support and encouragement throughout my PhD candidature. I am very grateful to the three reviewers of my thesis: Prof. Anthony G. Cohn, Prof. Christian Freska, and Prof. Abdul Sattar, for their valuable remarks and suggestions about the thesis. My thanks also belong to Sue Felix, for her excellent editing work. I am grateful to Prof. Shengsheng Wang and all the students and staff in QCIS at the University of Technology, Sydney, with whom I had interesting and inspiring discussions. Last but not least, I want to thank my parents and my wife for their constant and unwavering support over all these years.

Contents

List of Figures

Abstract

Qualitative Spatial and Temporal Reasoning (QSTR) is a subfield of artificial intelligence that represents and reasons with spatial/temporal knowledge in a qualitative way. In the past three decades, researchers have proposed dozens of relational models (known as qualitative calculi), including, among others, Point Algebra (PA) and Interval Algebra (IA) for temporal knowledge, Cardinal Relation Algebra (CRA) and Cardinal Direction Calculus (CDC) for directional spatial knowledge, and the Region Connection Calculus RCC-5/RCC-8 for topological spatial knowledge. *Relations* are used in qualitative calculi for representing spatial/temporal information (e.g. Germany is to the east of France) and constraints (e.g. the to-be-established landfill should be disjoint from any lake).

The reasoning tasks in QSTR are formalised via the *qualitative constraint satisfaction problem* (QCSP). As the central reasoning problem in QCSP, the *consistency problem* (which decides the consistency of a number of constraints in certain qualitative calculi) has been extensively investigated in the literature. For PA, IA, CRA, and RCC-5/RCC-8, the consistency problem can be solved by composition-based reasoning. For CDC, however, composition-based reasoning is incomplete, and the consistency problem in CDC remains challenging.

Previous works in QCSP assume that qualitative constraints only concern completely unknown entities. Therefore, constraints about *landmarks* (i.e., fixed entities) cannot be properly expressed. This has significantly restricted the usefulness of QSTR in real-world applications.

The main contributions of this thesis are as follows.

(i) The composition-based method is one of the most important reasoning methods in QSTR. This thesis designs a semi-automatic algo-

rithm for generating composition tables for general qualitative calculi. This provides a partial answer to the challenge proposed by Cohn in 1995.

(ii) Schockaert et al. (2008) extend the RCC models interpreted in Euclidean topologies to the fuzzy context and show that composition-based reasoning is sufficient to solve fuzzy QCSP, where 31 composition rules are used. This thesis first shows that only six of the 31 composition rules are necessary, and then introduces a method which consistently fuzzifies any classical RCC models. This thesis also proposes a polynomial algorithm for realizing solutions of consistent fuzzy RCC constraints.

(iii) Composition-based reasoning is incomplete for solving QCSP over the CDC. This thesis provides a cubic algorithm which for the first time solves the consistency problem of complete basic CDC networks, and further shows that the problem becomes NP-complete if the networks are allowed to be incomplete. This draws a sharp boundary between the tractable and intractable subclasses of the CDC.

(iv) This thesis proposes a more general and more expressive QCSP framework, in which a variable is allowed to be a landmark (i.e., a fixed object), or to be chosen among several landmarks. The computational complexity of the consistency problems in the new framework is then investigated, covering all qualitative calculi mentioned above. For basic networks, the consistency problem remains tractable for Point Algebra, but becomes NP-complete for all the remaining qualitative calculi. A special case in which a variable is either a landmark or is totally unknown has also been studied.

(v) A qualitative network is *minimal* if it cannot be refined without changing its solution set. Unlike the assumptions in the literature, this thesis shows that computing a solution of minimal networks is NP-complete for (partially ordered) PA, CRA, IA, and RCC-5/RCC-8. As a by-product, it has also been proved that determining the minimality of networks in these qualitative calculi is NP-complete.

Chapter 1

Introduction

Space and time are the most fundamental elements of the world. All our physical experience and practice are embedded in space and time, and our recognition of the world starts with an understanding of space and time. The ability to perceive, process, maintain, retrieve, and (most significantly) reason with spatial/temporal knowledge is indispensable for any intelligent creature. In fact, intelligence may develop rapidly only when an individual has a certain level of ability to handle spatial and temporal knowledge. Therefore, a sophisticated mechanism for dealing with spatial and/or temporal information is decisive and of key importance for the design of general artificial intelligent agents, and in disciplines including robot planning and navigation, natural language understanding, scene understanding and even fields outside AI (such as geographic information science (GIS) [32]).

The foundation for all reasoning tasks is the establishment of a formalism for the representation of spatial and temporal information. There are basically two approaches for this purpose: the quantitative approach and the qualitative approach. The next subsection provides an overview of both approaches.

1.1 Approaches for Spatial and Temporal Information

The quantitative approach, generally speaking, adopts a coordinate system to maintain spatial/temporal information. Considerable research has been conducted in the quantitative approach within traditional disciplines, such as computer vision [31, 69], computational geometry [18], and GIS [113]. Take computer vision for example. Raw digital images are acquired by image sensors. After pre-processing (e.g., noise reduction), features of the images (e.g., lines or textures) are extracted which may be used for further tasks such as region detection. Spatial information (e.g., images, features or regions) is represented by quantitative numbers, and reasoning tasks (e.g., noise reduction) are carried out mainly by numerical algorithms. Therefore we consider the common approach in computer vision a quantitative one.

The qualitative approach, on the other hand, represents spatial information by human comprehensible terms instead of exact numerical values. For example, in qualitative spatial representations [50], one may say "The Alps partially overlap Switzerland", "Vatican is surrounded by Rome", "Switzerland is to the north of Italy", and "Russia is larger than any other country", where spatial knowledge is represented by the spatial relations *partially overlap*, *surrounded by*, *to the north of* and *larger than*, rather than by exact geographical locations. Qualitative spatial representations [50] are pervasively involved in (and usually sufficient for) human activities and communications, e.g., to identify described objects or follow described routes. The qualitative approach is inherently succinct and efficient in reasoning, because only decisive knowledge is kept and details that are less important are ignored. Furthermore, the qualitative approach is closer to human cognition and may help build user-friendly interfaces (e.g., in database queries). It is believed that the qualitative approach is more suitable and promising for research and applications in artificial intelligence, and the goal of the QSTR community is to build a formal framework for representing and reasoning with qualitative spatial and/or temporal knowledge. As we have seen, the relations (such as *partially overlap*) are used as basic components for qualitative spatial/temporal knowledge representation. The formal frameworks developed by the QSTR community are essentially relational models, which are known as *qualitative calculi*. Cohn and Renz have summarised the goal of QSTR [12]:

> The challenge of QSR (Qualitative Spatial Reasoning) then is to provide calculi which allow a machine to represent and reason with spatial entities without resort to the traditional quantitative techniques prevalent in, for, e.g., the computer graphics or computer vision communities.

Both the quantitative and the qualitative approaches have advantages, and are prefer-able in different applications. The quantitative approach is ideal when spatial information is complete, precise and without ambiguity. The qualitative approach, on the other hand, may better suit applications where: (i) Precise values are less important. This is because the qualitative approach focuses on high-level knowledge; thus reasoning tasks may also be carried out at a high-level (e.g., logics, or the consistency problem, discussed in the next section) without concern for the less important details. (ii) Indefinite knowledge is involved. The qualitative approach may directly represent indefinite knowledge by non-basic relations, without jeopardizing the framework of all the reasoning tasks (although possibly increasing the reasoning complexity). (iii) Human interaction is involved. For example, in a navigation system, qualitative instructions may be preferred by people (although it is even better to adopt both approaches) [89].

These two approaches are not exclusive: they can (and perhaps should, if possible) be adopted at the same time. For example, if the knowledge base only contains the information that the Netherlands is *to the west of* Germany and Belgium is *to the west of* Germany, we cannot infer the cardinal relation between the Netherlands and Belgium. However, if we also have a map, we are able to determine that the Netherlands is to the north of Belgium by locating the two countries and applying an algorithm, which may cost more time but provide a satisfactory outcome. In fact, [36] has made the "poverty conjecture", claiming that no powerful, general-purpose, purely qualitative representation of spatial properties exists, especially for spatial and kinematic mechanisms. Although the conjecture itself is arguable [12], it may motivate QSTR communication to consider integrating both the quantitative and qualitative approaches. The extended framework of the consistency problem (see the next section and Chapter 6) may be viewed as a such attempt.

There are many existing applications of qualitative spatial representation and reasoning that are worthy of consideration, in areas such as GIS [33], navigation [34], high-level computer vision, natural language processing, cognitive systems [79], document analysis [1], visual (programming) languages [42], qualitative simulation of physical processes [17, 83, 54], location-based service [51], (spatial and/or temporal) databases [109], (spatial and/or temporal) data mining [49], and even biology [14].

Traditional GISs, with gigabytes of data, do not support intuitive and commonsense-oriented interfaces sufficiently well. Users are likely to expect to be able to specify queries in a qualitative way. The so-called *Naive Geography* [32] and the *qualitative*

GIS [16], in which the qualitative approach is fundamental, may be promising directions to follow to reduce the conflict.

1.2 Reasoning in a Qualitative Way

This section provides an overview of qualitative calculi and the reasoning problems in QSTR.

Since Allen's seminal work on Interval Algebra [3], researchers have proposed dozens of qualitative calculi (relational models) in the past three decades. Each qualitative calculus deals with a particular aspect of spatial/temporal knowledge. This thesis will focus on the following qualitative calculi: Point Algebra (PA) and Interval Algebra (IA) for temporal knowledge, Cardinal Relation Algebra (CRA) (for point-like objects) and Cardinal Direction Calculus (CDC) (for extended objects) for directional spatial knowledge, and the Region Connection Calculus RCC-5 and RCC-8 for topological spatial knowledge. There are many other important qualitative calculi that are not discussed in this thesis. These include are the point-and-interval calculus [22], the generalised interval algebra [59], the INDU calculus [82], the directed interval calculus [88], the cyclic interval calculus [6] for temporal knowledge, and the Flip-Flop Calculus [60] and its refinement the $\mathcal{LR}$ calculus [98], the Single Cross Calculus and Double Cross Calculus [97], the Oriented Point Relation Algebra ($\mathcal{OPRA}_m$) [72], the Dipole Relation Algebra [92, 73] with its fine grained variants $\mathcal{DRA}_f$ and $\mathcal{DRA}_{fp}$ [30], and the Qualitative Trajectory Calculus [20], for directional knowledge.

Generally speaking, a qualitative calculus is characterised by its *universe* and *basic relations*. The universe $\mathbf{U}$ of a qualitative calculus $\mathcal{M}$ is the domain of entities that are concerned in $\mathcal{M}$, and the basic relations are a jointly exhaustive and pairwise disjoint (JEPD) set of binary relations on $\mathbf{U}$ (i.e., a partition of $\mathbf{U} \times \mathbf{U}$), denoted by $\mathcal{B}_{\mathcal{M}}$. [1] The union of a subset of $\mathcal{B}_{\mathcal{M}}$ is called a non-basic relation. Note that a qualitative calculus provides a language for knowledge representation: a basic relation may be used to represent definite knowledge (e.g. Germany is *to the west of* France), while non-basic relations may be used for indefinite knowledge. Now we discuss three important reasoning problems: the weak composition problem, the consistency problem and the

[1]We only introduce binary qualitative calculi for simplicity, because only binary qualitative calculi will be discussed in this thesis.

minimal labeling problem.

The weak composition problem seeks to decide the possible relations between two entities a and c, if the relation between a and b, and that between b and c, are given for an entity b. More formally, it is to decide the *weak composition* of given relations α and β, which is defined as the smallest (possibly non-basic) relation in the qualitative calculus that contains the composition of α and β. For example, Interval Algebra may express temporal knowledge such as: World War Two occurred and ended *during* the 20th century, and the Industry Revolution ended *before* the 20th century. We may then infer that the Industry Revolution ended *before* World War Two. The weak composition problem reflects the most simple reasoning procedure. For a qualitative calculus $\mathcal{M}$, the weak composition problem is equivalent to computing the *composition table* (CT) of $\mathcal{M}$. A composition table records the weak composition of each pair of basic relations in the qualitative calculus. The importance of the composition table is much more than the simple reasoning tasks shown above. In fact, it plays a significant role in the consistency problem (see below). Because CTs are usually obtained manually, which is an error-prone procedure (especially for large qualitative calculi containing dozens or even hundreds of basic relations), the weak composition problem has been identified as a challenge for computer scientists in [13]. Chapter 3 of this thesis will provide a semi-automatic algorithm for computing composition tables.

The consistency problem aims to decide the consistency of a set of constraints over a certain qualitative calculus. Here a constraint is a formula that specifies the imposed requirement of the variables (e.g, x north y requires that x should be located to the north of y). The consistency problem is more general (and more difficult) than the weak composition problem. The consistency problem is the central reasoning problem in QSTR, as many other reasoning problems (e.g., the minimal labeling problem: see below) can be reduced to the consistency problem. The consistency problem has been extensively investigated and successfully solved for a number of qualitative calculi. For PA, IA, CRA, RCC-5, and RCC-8, it is well-known that composition-based reasoning (together with backtracking) can efficiently solve the consistency problems for these calculi. In particular, within a fuzzy context of RCC, Chapter 4 will prove that the composition-based method is still sufficient for determining the consistency problem. For CDC, however, composition-based reasoning is incomplete and the consistency problem remains challenging. Even worse, it is not clear whether the consistency problem is tractable when only basic CDC relations and the universal relation are considered. Chapter 5 will address the consistency problem in CDC.

Previous works about the consistency problem only consider qualitative constraints between completely unknown entities; that is to say, variables can be interpreted by arbitrary elements from the universe. In such a framework, we cannot express qualitative constraints between, say, a landmark (or a fixed entity) and a completely unknown entity. Consider the following example. Suppose you are recommended a restaurant in Sydney by a friend who has dined there before. The spatial information about the restaurant may be similar to "it is *in* downtown and *close to* a MacDonald's, and it is to the *west of* or *southwest of* Central Station." Furthermore, while the position of the restaurant may be totally unknown, the position of Central Station is fixed as a landmark, and the position of downtown is also fixed somehow, but the position of "MacDonald's" is only confined to a finite number of possibilities because there are several branches of MacDonald's in downtown Sydney. The traditional framework can hardly express such constraints properly, because it cannot fix the location of Central Station, or confine the location of MacDonald's. This has significantly reduced the usefulness of QSTR in real-world applications and we will propose a more general framework of the consistency problem in Chapter 6.

The minimal labeling problem [105] asks what information can be inferred between two of the entities, with respect to a given knowledge base. In QSTR, a knowledge base may be expressed by a set of constraints, and the possible relation between two entities under such constraints is called the strongest implied relation. The problem of computing the strongest implied relation between each pair of variables is called the minimal labeling problem. The minimal labeling problem is closely related to the consistency problem, as it is known that the minimal labeling problem and the consistency problem are equivalent with respect to Turing-reductions [107]. Even so, a closely related problem, i.e., computing solutions of minimal networks, remains open. Here, a *minimal* constraint network is a set of constraints in which each pair of the involved variables is related with the strongest implied relation. Note that a minimal constraint network is consistent by definition. Although it was assumed in [9] that computing solutions of minimal networks can be easily accomplished, the recent work by Gottlob [43] has proved that the problem in the classical constraint satisfaction problem is NP-hard. Chapter 7 will prove that, computing solutions of minimal networks in qualitative calculi PA, CRA, IA, RCC-5, and RCC-8 is also NP-hard, i.e., although the existence of solutions is ensured, finding these solutions is intractable.

1.3 Overview of This Thesis

In Chapter 2, we provide necessary and general notions used in the thesis, including preliminary mathematics, reasoning tasks in qualitative calculi, and all the qualitative calculi to be discussed later (i.e., Point Algebra, Interval Algebra, Cardinal Relation Algebra, Cardinal Direction Calculus and Region Connection Calculus). The following chapters which address the main contributions of the thesis, are organised as follows.

We design a semi-automatic algorithm for computing composition tables, which can be used to generate the composition tables of many qualitative calculi proposed in the literature. This provides a partial answer to the challenge proposed by Cohn [13]. In particular, we give for the first time the correct composition table of the INDU calculus. This is provided in Chapter 3.

Schockaert et al. [96] extend the region connection calculus to the fuzzy context and show that composition-based reasoning is sufficient to solve the consistency problem in a fuzzy context, where they use 31 composition rules. Chapter 4 will prove that only six out of the 31 composition rules are necessary, and will then introduce standard models for the fuzzy RCC, which are the fuzzy extensions of standard RCC models, and prove that any satisfiable set of fuzzy RCC-8 constraints has a solution in each standard fuzzy RCC model by designing a cubic realisation algorithm.

Composition-based reasoning is incomplete for solving the consistency problem over the CDC. Chapter 5 will show that, we still have a cubic algorithm to solve the consistency problem of complete networks of basic CDC constraints. Chapter 5 will further prove that if some constraints are allowed to be unspecified, the consistency problem of possibly incomplete networks of basic CDC constraints is NP-hard. This draws a sharp boundary between the tractable and intractable subclasses of the CDC.

We propose a more general framework for the consistency problem, in which a variable can be a fixed entity (called a landmark), chosen from several landmarks, or an arbitrary entity. In the extended framework we can express qualitative constraints, such as the queried hotel is *to the west of* a MacDonald's. The computational complexity of reasoning in the extended framework has also been studied for all qualitative calculi mentioned above. In the new framework, we show that the consistency problem is NP-complete for basic networks except for Point Algebra, and, if a variable is either a landmark or totally unknown, then the consistency problem is tractable for basic networks in all qualitative calculi mentioned above except RCC-8.

Minimal (qualitative) networks are consistent qualitative networks in which each basic relation in the qualitative constraint between any two variables can be realised by a solution. Unlike the assumptions in the literature, we show that computing a solution of minimal networks is NP-complete for partially ordered PA, CRA, IA, RCC-5, and RCC-8. As a by-product, we also prove that it is NP-complete to determine the minimality of networks for these qualitative calculi.

Chapter 2

Concepts and Notations

This chapter first provides brief introduction to the mathematical tools that are used in the thesis, and then introduces several qualitative calculi, including Point Algebra, Interval Algebra, Region Connection Calculus, and Cardinal Direction Calculus. We then discuss fundamental reasoning problems in QSTR, including the weak composition problem, the consistency problem, and the minimal labeling problem. The last section summaries the results obtained in this thesis.

2.1 Preliminaries

This section provides definitions of a number of basic concepts that will be used in this thesis.

2.1.1 Relations

First we recall *relations* and the *composition* and *converse* of relations.

Definition 2.1 (binary relation, composition, converse) *Suppose X is a nonempty set. For positive integer k, a binary relation on X is a subset of $X \times X$. Let α,β be two binary relations on a set X. The composition of α and β, denoted by $\alpha \circ \beta$, is the binary relation on X defined by the following equation:*

$$\alpha \circ \beta := \{(a,c) \in X \times X : \exists b \in X \ \text{ such that } \ (a,b) \in \alpha, (b,c) \in \beta\}. \tag{2.1}$$

The converse of a binary relation α, denoted by α˘, is defined as

$$\alpha^{\smile} := \{(b,a) \in X \times X : (a,b) \in \alpha\}. \tag{2.2}$$

Notation Suppose α is a binary relation on X, we usually write $a\alpha b$ instead of $(a,b) \in \alpha$.

2.1.2 Boolean Algebra

Definition 2.2 (Boolean Algebra [70]) *A Boolean Algebra is a structure $(A, +, \cdot, -, \bot, \top)$, where A is a set, $+$ and $\cdot$ are two binary operations on A, $-$ is a unary operation on A, and $\bot$ and $\top$ are two distinguished elements in A, such that for any $x, y, z \in A$, the following axioms are satisfied.*

$$x + (y + z) = (x + y) + z, \qquad x \cdot (y \cdot z) = (x \cdot y) \cdot z, \tag{2.3}$$

$$x + y = y + z, \qquad x \cdot y = y \cdot x, \tag{2.4}$$

$$x + (x \cdot y) = x, \qquad x \cdot (x + y) = x, \tag{2.5}$$

$$x \cdot (y + z) = (x \cdot y) + (x \cdot z), \qquad x + (y \cdot z) = (x + y) \cdot (x + z), \tag{2.6}$$

$$x + (-x) = \top, \qquad x \cdot (-x) = \bot. \tag{2.7}$$

Suppose $(A, +, \cdot, -, \bot, \top)$ is a Boolean Algebra, and S is a subset of A. There exists a minimal subset of A that contains $S \cup \{\bot, \top\}$ and is closed under operations $+, \cdot$ and $-$. In fact, this set is the intersection of all subsets of A that satisfy the above properties. We denote this set by S'. It is clear that $(S', +, \cdot, -, \bot, \top)$ is a subalgebra of Boolean algebra $(A, +, \cdot, -, \bot, \top)$, which is called the Boolean Algebra *generated by S*.

As an example of Boolean Algebra, suppose X is a nonempty set and $\mathcal{P}(X)$ is the power set of X, i.e., the set of all subsets of X. It can be directly proved that $(\mathcal{P}(X), \cup, \cap, -, \varnothing, X)$ is a Boolean Algebra, where $\cup$ and $\cap$ are the union operation and the intersection operation on sets respectively, and the unary operation $-$ is defined as $-a = X \backslash a$, the complement of a with respect to X. This Boolean Algebra is called the *power set algebra* of X. A subalgebra of the power set algebra of set X is called an *algebra of subsets* of X.

Notation We denote $\mathbf{Rel}(A)$ as the set of all binary relations on set A, viz., the power set of $A \times A$.

By the arguments above, we know that $(\mathbf{Rel}(A), \cup, \cap, -, \varnothing, A \times A)$ is a Boolean Algebra. Furthermore, suppose $\mathcal{S}$ is a set of binary relations on A. Then there is a Boolean Algebra generated by $\mathcal{S}$ which is an algebra of subsets of $\mathbf{Rel}(A)$. This is a convenient way to define Boolean Algebras and will be used frequently when defining qualitative calculi.

2.1.3 Topology

Next we introduce basic topological concepts, such as can be found in any textbook on point-set topology (e.g., [76]).

Definition 2.3 (topology, open sets, closed sets) *Let X be a set, and τ be a family of subsets of X. We say τ is a topology on X, if the following axioms are satisfied:*

$$X \text{ and } \varnothing \text{ are elements in } \tau, \tag{2.8}$$

$$\text{any intersection of finite elements in } \tau \text{ is an element in } \tau, \tag{2.9}$$

$$\text{any union of elements in } \tau \text{ is an element in } \tau. \tag{2.10}$$

We call (X, τ), or simply X if τ is clear, a topological space. *An element in τ is called an* open set *in the topology. A subset of X is called a* closed set *in the topology if its complement is an open set.*

Definition 2.4 (interior, closure, regular closed sets) *Suppose X is a set and τ is a topology on X. Let r be a subset of X. We say a point P in X is an* interior point *of r, if there exists an open set s such that $P \in s \subseteq r$. We say P an* exterior point *of r, if there exists an open set s such that $P \in s \subseteq r^c$, where r^c is the complement of r. We say P is a* boundary point *of r, if it is neither an interior point nor an exterior point.*

The interior *of a set r, denoted by r°, is defined as the set of all the interior points of r. The* closure *of r, denoted by $\overline{r}$, is defined as the complement of the interior of r^c. We say a set r is* regular closed, *if it equals to the closure of its interior, i.e., $r = \overline{r^\circ}$.*

Definition 2.5 (connected set) *We say an open set r in a topological space X is* connected, *if there do not exist two open sets s and t in X such that $u \cup v = r$ and $u \cap v = \varnothing$.*

If X itself is a connected open set, then se say X is a connected topological space.

Let $B(P, \epsilon)$ be the open disk on the plane that is centred at P and with radius ϵ, where P is a point on the plane and $\epsilon > 0$. That is $B(P, \epsilon) = \{Q : |P - Q| < r\}$, where $|P - Q|$ is the distance function defined as $|P - Q| = \sqrt{(P_x - Q_x)^2 + (P_y - Q_y)^2}$, assuming $P = (P_x, P_y)$ and $Q = (Q_x, Q_y)$.

Definition 2.6 (2-dim Euclidean topology) *Let τ be the family of subsets of $\mathbb{R}^2$ defined by: a subset r of $\mathbb{R}^2$ is in τ if and only if, for any $P \in r$, there exists some $\epsilon > 0$ such that $B(P, \epsilon) \subseteq r$. The topology τ is called the* 2-dim Euclidean topology.

Definition 2.7 (region) *A* region *is a nonempty regular closed set in the 2-dimensional Euclidean topology. We say a region is* connected*, if it has a connected interior. We say a region is* bounded*, if all the points in it are contained by some rectangle $[x, x'] \times [y, y']$. We say a bounded region* has a hole*, if its complement is not connected. We say a region is a* simple *region, if it is connected and has no holes. We say a region is* convex*, if for any two points P, Q in it, the straight line segment that has P and Q as its end points is also contained in the region.*

The following figure illustrates some of the above concepts.

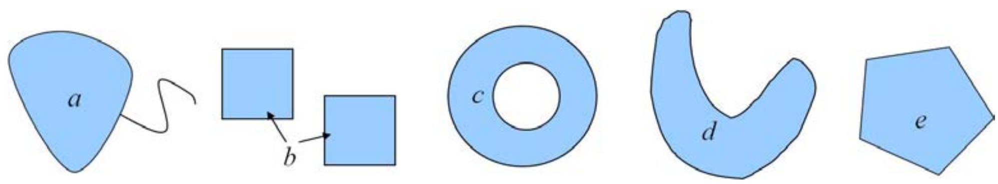


Figure 2.1: Illustrations of (a) a closed set (b) a disconnected region (c) a connected region with a hole (d) a simple region (e) a convex region.

Suppose X is a topological space. Let A be the set of regular closed sets in X. Then $(\mathrm{A}, +, \cdot, -, \varnothing, X)$ is a Boolean Algebra, where $x + y = x \cup y$, $x \cdot y = \overline{(x \cap y)^\circ}$, $-x = \overline{X \setminus x}$. This is called the *regular closed algebra* of X.

The minimal bounding rectangle is an approximation of a region.

Definition 2.8 (minimal bounding rectangle) *For a bounded (connected or disconnected) region r in the real plane, define* $x^-(r), x^+(r), y^-(r)$ *and* $y^+(r)$ *as*

$$x^-(r) = \inf\{x : (x,y) \in r\}, \quad x^+(b) = \sup\{x : (x,y) \in r\}, \tag{2.11}$$
$$y^-(r) = \inf\{y : (x,y) \in r\}, \quad y^+(b) = \sup\{y : (x,y) \in r\}. \tag{2.12}$$

The x- and y-projections of r, denoted by $I_x(r)$ *and* $I_y(r)$ *respectively, are then defined as*

$$I_x(r) = [x^-(r), x^+(r)], \tag{2.13}$$
$$I_y(r) = [y^-(r), y^+(r)]. \tag{2.14}$$

The minimum bounding rectangle *(mbr) of b, denoted by* $\boldsymbol{mbr}(r)$*, is defined to be*

$$\boldsymbol{mbr}(r) = I_x(r) \times I_y(r). \tag{2.15}$$

2.2 CSP and SAT

This section briefly introduces the classical constraint satisfaction problem (CSP) and the well-known SAT problem, which will be used in later chapters.

Definition 2.9 (constraint satisfaction problem [21])

Instance: *A 3-tuple* (V, D, C)*, where* $V = \{v_1, \cdots, v_n\}$ *is a set of variables,* $D = \{D_1, \cdots, D_n\}$ *is the set of domains of variables, and C is a set of binary constraints of the form* $v_i c_{ij} v_j$*, where* $c_{ij} \subseteq D_i \times D_j$*.*

Question: *Deciding the* satisfiability *of instance, i.e., whether there exists an assignment* $\tau : V \to D^*$ *where* D^* *is the union of* $D_1, \cdots, D_n$*, such that* $\tau(v_i) \in D_i$ *and each constraint* $v_I c_{ij} v_j$ *in C,* $(\tau(v_i), \tau(v_j)) \in c_{ij}$*.*

The 3-SAT problem can be viewed as a special case of CSP, where variables can only take values in $\{\mathsf{true}, \mathsf{false}\}$, and the constraints are of certain form. Formally, suppose V is a set of *atomic propositional variables*. Then a *literal* is a formula of the form either p or $\neg p$, where p is a variable in V. A *disjunctive clause* (or simply a clause) is a formula of the form $c = l_1 \vee l_2 \vee \cdots l_m$ (also denoted by $\bigvee_{k=1}^{m} l_k$), where $l_k (1 \leq k \leq m)$ are literals. A *conjunctive normal form* (CNF) is of the form $\phi = c_1 \wedge c_2 \cdots c_n$ (or $\bigwedge_{k=1}^{n} c_k$), where $c_k (1 \leq k \leq n)$ are disjunctive clauses. An *assignment* is a function $f : V \to \{0, 1\}$. A literal l is assigned true under assignment f, if $l = p$ and $f(p) = \mathsf{true}$, or $l = \neg p$ and $f(p) = \mathsf{false}$. Clause $c = \bigvee_{k=1}^{m} l_k$ is satisfied under assignment f, if any clause l_j is assigned true. A CNF $\bigwedge_{k=1}^{n} c_k$ is assigned true under f, if all clauses c_k are assigned true. We say a clause c or a CNF ϕ is *satisfiable*, if there exists an assignment under which c or ϕ is assigned true.

Definition 2.10 (SAT, 3-SAT) *The 3-SAT problem is defined as follows:*

> ***Instance****: A CNF $\phi = \bigwedge_{j=1}^{m} c_j$. In particular, if each c_j has only three literals (i.e., c_j can be written as $l_{j,1} \vee l_{j,2} \vee l_{j,3}$), then the instance is also a 3-SAT instance.*
>
> ***Question****: Deciding the* satisfiability *of ϕ, i.e., whether ϕ is satisfiable.*

It is well-known that the 3-SAT problem is NP-Complete [15]. That is to say, the 3-SAT problem can be solved by a nondeterministic algorithm in polynomial time, and if there is a polynomial deterministic algorithm that solves 3-SAT, all the other NP-Complete problems can also be solved in polynomial time.

The NP-hardness results provided in the following sections are proved by polynomial reductions from 3-SAT. We fix some notations here. Suppose $\phi = \bigwedge_{j=1}^{m} c_j$ is a 3-SAT instance over propositional variables $p_1, p_2, \cdots, p_n$, where c_j is a clause with three literals. Write $\mathrm{Var}(c_j)$ for the set of propositional variables appearing in c_j, and write $\mathrm{Var}^+(c_j)$ ($\mathrm{Var}^-(c_j)$ resp.) for the set of propositional variables that occur as positive (negative resp.) literals in c_j. Formally,

$$\mathrm{Var}^+(c_j) := \{p_i : p_i \text{ is a literal in } c_j\}, \tag{2.16}$$

$$\mathrm{Var}^-(c_j) := \{p_i : \neg p_i \text{ is a literal in } c_j\}, \tag{2.17}$$

$$\mathrm{Var}(c_j) := \mathrm{Var}^+(c_j) \cup \mathrm{Var}^-(c_j). \tag{2.18}$$

2.3 Qualitative Calculi

The qualitative approach to spatial and temporal reasoning often uses qualitative calculi to represent qualitative information. A qualitative calculus provides a language (focusing on some spatial or temporal properties of entities) which can be used for both knowledge representation and reasoning.

Definition 2.11 (qualitative calculus [63].) *Suppose $\boldsymbol{U}$ is a set of spatial or temporal entities. A* qualitative calculus *$\mathcal{M}$ on $\boldsymbol{U}$ is a tuple $(U, \mathcal{R})$, where $\mathcal{R}$ is a finite Boolean subalgebra of $\boldsymbol{Rel}(\boldsymbol{U})$*[1].

$\boldsymbol{U}$ is called the universe *of the qualitative calculus $\mathcal{M}$. An atom in the Boolean algebra R is called a* basic *relation, or an* atomic *relation in $\mathcal{M}$. If a relation in $\mathcal{M}$ is nonempty and not a basic relation, it is called a non-basic relation. In particular, $\boldsymbol{U} \times \boldsymbol{U}$ is called the* universal *relation and usually denoted by symbol* ?.

We say a qualitative calculus $\mathcal{M}$ is closed under converse*, if the converse of any relation in $\mathcal{M}$ is also a relation in $\mathcal{M}$. Similarly, we say $\mathcal{M}$ is* closed under composition*, if the composition of any two relations in $\mathcal{M}$ is also in $\mathcal{M}$.*

Notation Generally, we will use $\mathbf{U}_{\mathcal{M}}$ to denote the universe of qualitative calculus $\mathcal{M}$, and use $\mathcal{B}_{\mathcal{M}}$, $\mathcal{B}^*_{\mathcal{M}}$ to denote respectively the set of basic relations and the set of all relations in qualitative calculus $\mathcal{M}$. The subscript may be omitted when the semantic is clear.

As the basic relations in a (binary) qualitative calculus $\mathcal{M}$ (with universe $\mathbf{U}$) are the atoms in a finite Boolean subalgebra of $\mathbf{Rel}(\mathbf{U})$, they form a partition of $\mathbf{U} \times \mathbf{U}$. In other words, they are a set of JEPD (jointly exhaustive and pairwise disjoint) relations. On the other side, given a set $\mathbf{U}$ and a JEPD set of relations B on $\mathbf{U} \times \mathbf{U}$, we may generate a Boolean Algebra B^* by B. Because B is a JEPD set of relations, the atoms of B^* are exactly the relations B, and B^* has $2^m - 1$ nonempty relations in total, where m is the cardinality of B. In this way, a qualitative calculus $(\mathbf{U}, B^*)$ may be defined according to the JEPD set of relations B. The qualitative calculi introduced below are all defined in this way.

We next recall the well-known Point Algebra (PA) [107, 106, 8], the Cardinal Relation Algebra (CRA) [37, 62], the Interval Algebra (IA) [3], the RCC-8 and RCC-5 algebras [84], and the Cardinal Direction Calculus [46].

[1]We here only define binary qualitative calculus, as the qualitative calculi discussed in this thesis are all binary ones. This definition can be easily generalised to k-ary qualitative calculus.

2.3.1 Point Algebras

A Point Algebra is defined over a partial order or a total order.

Definition 2.12 (partial order, total order) *Suppose A is a set, and $\leq$ is a binary relation on A. We say structure $(A, \leq)$ is a* partial order*, if the following axioms are satisfied, for any $a, b, c \in A$:*

$$a \leq a, \tag{2.19}$$

$$\text{if } a \leq b \text{ and } b \leq a, \text{ then } a = b, \tag{2.20}$$

$$\text{if } a \leq b \text{ and } b \leq c, \text{ then } a \leq c. \tag{2.21}$$

We say a partial order $(A, \leq)$ is a total order*, if it also satisfies the following axiom:*

$$a \leq b \quad \text{or} \quad b \leq a. \tag{2.22}$$

Definition 2.13 (Point Algebras [8]) *Let $(\boldsymbol{U_{PA}}, \geq)$ be a nontrivial partial order. Let $\mathcal{B}^*_{\boldsymbol{PA}}$ be the algebra of subsets of $\boldsymbol{Rel}(\boldsymbol{U_{PA}})$ which is generated by $\{\geq\}$. Then we call $(\boldsymbol{U_{PA}}, \mathcal{B}^*_{\boldsymbol{PA}})$ the* Point Algebra *with respect to $(\boldsymbol{U_{PA}}, \geq)$.*

Note that in general the Point Algebra with respect to $(\mathbf{U_{PA}}, \geq)$ has four basic relations, viz., the identity relation $=$ and the relations defined below[1]:

$$> := \{(a, b) \in \mathbf{U_{PA}} \times \mathbf{U_{PA}} : a \geq b, b \not\geq a\}, \tag{2.23}$$

$$< := \{(a, b) \in \mathbf{U_{PA}} \times \mathbf{U_{PA}} : a \not\geq b, b \geq a\}, \tag{2.24}$$

$$\| := \{(a, b) \in \mathbf{U_{PA}} \times \mathbf{U_{PA}} : a \not\geq b, b \not\geq a\}. \tag{2.25}$$

If $(\mathbf{U_{PA}}, \geq)$ is a total order, then $\|$ is empty and the Point Algebra has only three basic relations $<, >$ and $=$.

As defined above, a Point Algebra is determined by a partial order $(\mathbf{U_{PA}}, \geq)$. Therefore, we may call $(\mathbf{U_{PA}}, \geq)$ as the *underlying structure* of the Point Algebra. The most common totally ordered Point Algebra is the one with underlying structure $(\mathbb{R}, \geq)$, where $\mathbb{R}$ is the set of real numbers and $\geq$ is the standard greater than or equal relation.

[1] We assume both $\mathbf{U_{PA}}$ and $\geq$ are not empty.

We here provide another example of Point Algebra which has the underlying structure $(\mathcal{N}^+, \leq)$, where $\mathcal{N}^+$ is the set of positive integers and $\leq$ is defined as

$$\leq := \{(a,b) \in \mathcal{N}^+ \times \mathcal{N}^+ : \text{there exists some integer } k \text{ s.t. } b = ka\}. \tag{2.26}$$

In fact, any finite Point Algebra can be embedded into this one. This is because, suppose $\mathcal{M}$ is a finite Point Algebra with underlying partial order $(A, \leq)$, where $A = (x_1, \cdots, x_n)$. We construct a mapping $f : A \to \mathcal{N}^+$ as follows. Let $t_1, \cdots, t_n$ be the first n prime numbers respectively. Let t_i' be $t_i \times \Pi\{t_j : a_j < a_i\}$. It is clear that $a_i \leq a_j$ iff $t_i \leq t_j$, where binary relation $\leq$ on $\mathcal{N}^+$ is defined above. Therefore $f(x_i) = t_i$ is the desired embedding.

Point Algebra is closed under converse, as the converses of $<$, $>$, $\|$ and $=$ are $>$, $<$, $\|$ and $=$ respectively by definition. A totally ordered Point Algebra is closed under composition if its universe is dense in the sense that $a < b$ implies the existence of some c such that $a < c < b$.

Notation In the following chapters, we call the totally ordered Point Algebra with underlying structure $(\mathbb{R}, \geq)$ simply the Point Algebra if not clarified otherwise.

2.3.2 Interval Algebra

Definition 2.14 (Interval Algebra [3]) *Let $\boldsymbol{U_{IA}}$ be the set of closed intervals on the real line, i.e., $\boldsymbol{U_{IA}} = \{[x^-, x^+] : -\infty < x^- < x^+ < \infty\}$. The following thirteen JEPD relations between two intervals $x = [x^-, x^+]$ and $y = [y^-, y^+]$ are defined by the order of the four endpoints of x and y (see the table below for definitions).*

$$\mathsf{p}, \mathsf{pi}, \mathsf{m}, \mathsf{mi}, \mathsf{o}, \mathsf{oi}, \mathsf{s}, \mathsf{si}, \mathsf{d}, \mathsf{di}, \mathsf{f}, \mathsf{fi}, \mathsf{eq}.$$

The Interval Algebra is defined to be $\mathcal{M}_{\boldsymbol{IA}} = (\boldsymbol{U_{IA}}, \mathcal{B}^*_{\boldsymbol{IA}})$, where $\mathcal{B}^*_{\boldsymbol{IA}}$ is the Boolean Algebra generated by the thirteen JEPD relations.

The Interval Algebra is closed under both converse and composition.

Relation	Symbol	Converse	Definition
before	p	pi	$x^- < x^+ < y^- < y^+$
meets	m	mi	$x^- < x^+ = y^- < y^+$
overlaps	o	oi	$x^- < y^- < x^+ < y^+$
starts	s	si	$x^- = y^- < x^+ < y^+$
during	d	di	$y^- < x^- < x^+ < y^+$
finishes	f	fi	$y^- < x^- < x^+ = y^+$
equals	eq	eq	$x^- = y^- < x^+ = y^+$

Table 2.1: Basic IA relations and their converses, where $x = [x^-, x^+], y = [y^-, y^+]$ are two intervals.

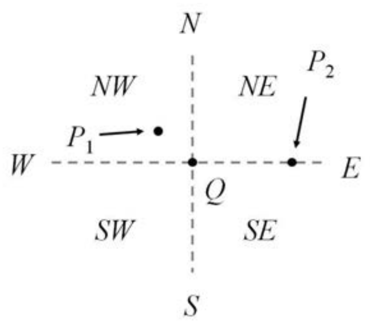
2.3.3 Cardinal Relation Algebra and Rectangle Algebra

The Cardinal Relation Algebra focuses on the cardinal relation between two points on the real plane, which can be viewed as the Cartesian product of two totally ordered Point Algebras with underlying structure $(\mathbb{R}, \geq)$.

Definition 2.15 (Cardinal Relation Algebra [37, 62]) *Define the following nine binary relations on the points on the real plane* $\mathbb{R}^2$*:*

$$NW, N, NE, W, EQ, E, SW, S, SE$$

as in Table 2.2. The Cardinal Relation Algebra (CRA) $\mathcal{M}_{CRA}$ *is defined to be* $(\mathbb{R}^2, \mathcal{B}^*_{CRA})$*, where* $\mathcal{B}^*_{CRA}$ *is the Boolean Algebra generated by the nine JEPD relations above.*



Relation	Definition
NW	$x < x', y > y'$
N	$x = x', y > y'$
NW	$x > x', y > y'$
W	$x < x', y = y'$
EQ	$x = x', y = y'$
E	$x > x', y = y'$
SW	$x < x', y < y'$
S	$x = x', y < y'$
SW	$x > x', y < y'$

Table 2.2: Definitions and illustrations of basic relations of Cardinal Relation Algebra, where in the left figure we have P_1 NW Q and P_2 E Q

Similarly, we may extend the Interval Algebra to two dimensions and get the Rectangle Algebra, which is the Cartesian product of two Interval Algebras.

Definition 2.16 (Rectangle Algebra, [4]) *Let $\boldsymbol{U_{RA}}$ be the set of all rectangles on the plane that have sides parallel to the axes, that is*

$$\boldsymbol{U_{RA}} = \{[x^-, x^+] \times [y^-, y^+] : -\infty < x^- < x^+ < \infty, -\infty < y^- < y^+ < \infty\}. \tag{2.27}$$

For each pair of basic IA relations α and β, define a binary relations $\alpha \otimes \beta$ on $\boldsymbol{U_{RA}}$ as follows, where $I_x(r)$ and $I_y(r)$ are the x- and y-projections of rectangle r respectively:

$$\alpha \otimes \beta = \{(r_1, r_2) \in \boldsymbol{U_{RA}} \times \boldsymbol{U_{RA}} : I_x(r_1)\ \alpha\ I_x(r_2)\ \ \textit{and}\ \ I_y(r_1)\ \beta\ I_y(r_2)\}. \tag{2.28}$$

Let $\mathcal{B}^*_{\boldsymbol{RA}}$ be the Boolean Algebra generated by the $13 \times 13 = 169$ relations defined above. The qualitative calculus $(\boldsymbol{U_{RA}}, \mathcal{B}^*_{\boldsymbol{RA}})$ is called the Rectangle Algebra.

Clearly, each relation $\alpha \otimes \beta$ defined in Equation 2.28 is a basic relation in the Rectangle Algebra.

We may extend the universe of the Rectangle Algebra to all bounded regions. Let a and b are two bounded regions on the plane. Let $\mathbf{mbr}(a)$ and $\mathbf{mbr}(b)$ be the rectangles defined as in Equation 2.15. Then the generalised rectangle relation between a and b is defined as the rectangle relation between $\mathbf{mbr}(a)$ and $\mathbf{mbr}(b)$. It is clear that the generalised RA also has 169 basic relations.

2.3.4 Region Connection Calculus

The Region Connection Calculus was defined originally as a first order theory [84]. Here we adopt an equivalent definition based on the so-called *Boolean connection algebra* [101].

Definition 2.17 (RCC model [101]) *Let $(A; +, \cdot, -, \bot, \top)$ be a nontrivial Boolean algebra, and write $\boldsymbol{U_{RCC}}$ for $A \setminus \{\bot\}$, and $\boldsymbol{U^*_{RCC}}$ for $\boldsymbol{U_{RCC}} \setminus \{\top\}$. We say structure $(\boldsymbol{U_{RCC}}, \mathsf{C})$ is a* Boolean connection algebra*, or an RCC model, if C is a binary relation on $\boldsymbol{U_{RCC}}$*

which satisfies the following axioms:

$$(\forall x \in \boldsymbol{U_{RCC}})\ C(x,x), \tag{2.29}$$

$$(\forall x,y \in \boldsymbol{U_{RCC}})\ C(x,y) \to C(y,x), \tag{2.30}$$

$$(\forall x \in \boldsymbol{U^{\star}_{RCC}})\ C(x,-x), \tag{2.31}$$

$$(\forall x,y,z \in \boldsymbol{U_{RCC}})\ C(x,y+z) \leftrightarrow C(x,y) \vee C(x,z), \tag{2.32}$$

$$(\forall x \in \boldsymbol{U^{\star}_{RCC}})(\exists y \in \boldsymbol{U_{RCC}})\ \neg C(x,y). \tag{2.33}$$

A number of binary relations are defined by C by first order logic[1]:

$$\mathsf{DC} = \neg\mathsf{C}(r,s), \tag{2.34}$$

$$\mathsf{P} = (\forall t \in \mathbf{U_{RCC}})(\mathsf{C}(t,r) \to \mathsf{C}(t,s)), \tag{2.35}$$

$$\mathsf{O} = (\exists t \in \mathbf{U_{RCC}})(\mathsf{P}(t,r) \wedge \mathsf{P}(t,s)), \tag{2.36}$$

$$\mathsf{PO} = \mathsf{O}(r,s) \wedge \neg\mathsf{P}(r,s) \wedge \neg\mathsf{P}(s,r), \tag{2.37}$$

$$\mathsf{EC} = \mathsf{C}(r,s) \wedge \neg\mathsf{O}(r,s), \tag{2.38}$$

$$\mathsf{DR} = \mathsf{DC}(r,s) \vee \mathsf{EC}(r,s), \tag{2.39}$$

$$\mathsf{EQ} = \mathsf{P}(r,s) \wedge \mathsf{P}(s,r), \tag{2.40}$$

$$\mathsf{PP} = \mathsf{P}(r,s) \wedge \neg\mathsf{P}(s,r), \tag{2.41}$$

$$\mathsf{NTP} = (\forall t \in \mathbf{U_{RCC}})(\mathsf{C}(t,r) \to \mathsf{O}(t,s)), \tag{2.42}$$

$$\mathsf{TPP} = \mathsf{PP}(r,s) \wedge \neg\mathsf{NTP}(r,s), \tag{2.43}$$

$$\mathsf{NTPP} = \mathsf{PP}(r,s) \wedge \mathsf{NTP}(r,s). \tag{2.44}$$

Besides, relations $\mathsf{Pi}, \mathsf{PPi}, \mathsf{TPPi}, \mathsf{NTPPi}$ are defined as the converse relations of P, PP,TPP and NTPP respectively.

Definition 2.18 (RCC-8 and RCC-5 Algebras) *Suppose $(\boldsymbol{U_{RCC}}, C)$ is an RCC model. The RCC-5 Algebra and the RCC-8 Algebra are the qualitative calculi $(\boldsymbol{U_{RCC}}, \mathcal{B}^{*}_{\boldsymbol{RCC5}})$ and $(\boldsymbol{U_{RCC}}, \mathcal{B}^{*}_{\boldsymbol{RCC8}})$ respectively, where $\mathcal{B}^{*}_{\boldsymbol{RCC5}}$ and $\mathcal{B}^{*}_{\boldsymbol{RCC8}}$ are the Boolean Algebras generated by relations $\{O, P, Pi\}$ and $\{C, O, P, NTPP, Pi, NTPPi\}$ defined above respectively.*

[1] This is the reason why C is called the *primitive* predicate in the original formalisation.

The basic relations of the RCC-5 Algebra are $\mathsf{DR}, \mathsf{PO}, \mathsf{EQ}, \mathsf{PP}$ and PPi, while those of the RCC-8 Algebra are $\mathsf{DC}, \mathsf{EC}, \mathsf{PO}, \mathsf{EQ}, \mathsf{TPP}, \mathsf{NTPP}, \mathsf{TPPi}$ and NTPPi.

RCC-5 may be viewed as a coarse version of RCC-8 which only deals with mereological (part-whole) aspect of entities.

Recall that Point Algebras are defined on their underlying structures which are partial orders. The case for RCC-5 and RCC-8 Algebras is similar: each RCC-5 or RCC-8 Algebra relies on an RCC model as its underlying structure, which is composed of a Boolean Algebra and a binary relation on it. Recall that given a topological space X, the regular closed sets in X compose a Boolean Algebra. The standard interpretation of the RCC model is composed of such a Boolean Algebra generated by the regular closed sets in the 2-dim Euclidean topology, and a connectedness relation C on the Boolean Algebra, as defined in the following definition.

Definition 2.19 (standard interpretation for RCC model) *Suppose $\boldsymbol{U_{Reg}}$ is the set of all regions (nonempty regular closed sets) on the 2-dim Euclidean space. Define binary relation C as*

$$\mathsf{C} = \{(a,b) \in \boldsymbol{U_{Reg}} \times \boldsymbol{U_{Reg}} : a \cap b \neq \emptyset\}. \tag{2.45}$$

Other RCC relations are defined as in Equations 2.34 - 2.44, where **DC**, **EC**, **PO**, **EQ**, **TPP** *and* **NTPP** *are equivalent to those defined in Table 2.3. Relations* **TPPi** *and* **NTPPi** *to be the converses of* **TPP** *and* **NTPP** *respectively. The RCC-8 algebra is the qualitative calculus $(\boldsymbol{U_{Reg}}, \mathcal{B}^*_{\boldsymbol{RCC8}})$, where $\mathcal{B}^*_{\boldsymbol{RCC8}}$ is the Boolean Algebra generated by*

$$\mathbf{DC}, \mathbf{EC}, \mathbf{PO}, \mathbf{EQ}, \mathbf{TPP}, \mathbf{NTPP}, \mathbf{TPPi}, \mathbf{NTPPi}.$$

Relation	Meaning	Relation	Meaning
EQ	$a = b$	**EC**	$a \cap b \neq \emptyset,\ a^\circ \cap b^\circ = \emptyset$
DC	$a \cap b = \emptyset$	**PO**	$a \nsubseteq b,\ b \nsubseteq a,\ a^\circ \cap b^\circ \neq \emptyset$
TPP	$a \subset b,\ a \nsubseteq b^\circ$	**NTPP**	$a \subset b^\circ$

Table 2.3: Topological interpretation of basic RCC-8 relations on plane regions a, b

The RCC-5 algebra is the sub-algebra of RCC-8 generated by the following five relations

$$\mathbf{DR}, \mathbf{PO}, \mathbf{EQ}, \mathbf{PP}, \mathbf{PPi},$$

where **DR** = **DC** ∪ **EC**, **PP** = **TPP** ∪ **NTPP**, *and* **PPi** = **TPPi** ∪ **NTPPi**.

Figure 2.2 provides illustrations of basic RCC-8 and RCC-5 relations. Note that only simple regions are used in the figure; nonetheless, the regions in general may contain multiple pieces or have holes.

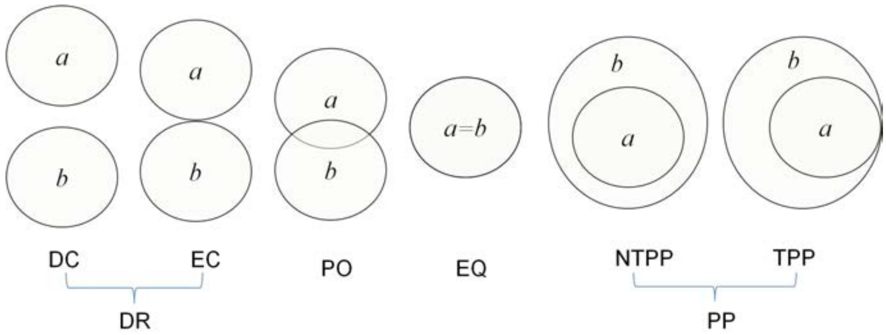


Figure 2.2: Illustration of the basic relations in RCC-5 / RCC-8

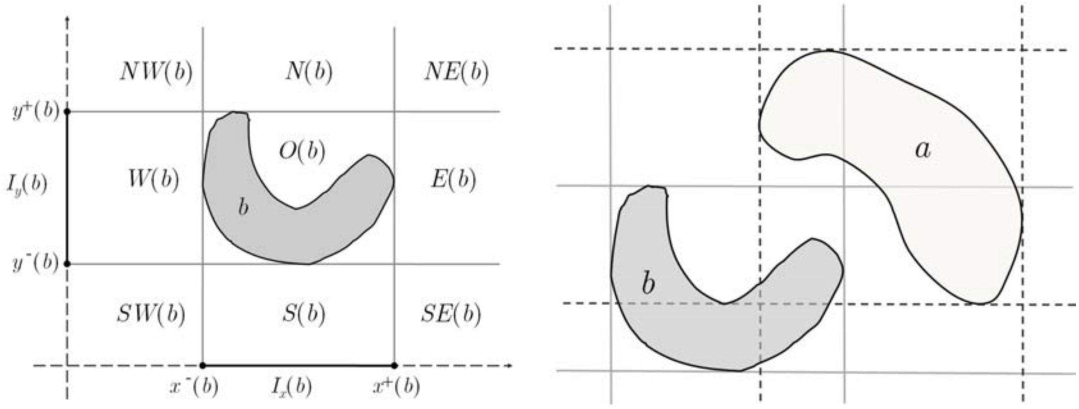
2.3.5 Cardinal Direction Calculus

The Cardinal Direction Calculus (CDC) was proposed in [46]. The universe of CDC is the set of bounded connected plane regions, denoted by $\mathbf{U}_{\mathbf{CDC}}$. To define the basic relations of CDC, recall that $\mathbf{mbr}(r)$ is the smallest rectangle that contains r and has sides parallel to the axes. Suppose b is a region. We extend the four edges of $\mathbf{mbr}(b)$ and thus partition the plane into nine tiles, which are denoted as

$$NW(b), N(b), NE(b), W(b), O(b), E(b), SW(b), S(b), SE(b)$$

respectively, see Figure 2.3 (a) for illustration. Note that each tile is a bounded or unbounded connected region, and the intersection of two tiles is of a dimension lower than two.

The notion of the *direction relation matrix* was first proposed in [46] to represent the cardinal direction between a primary object and a reference object. The matrix records which tiles (generated by the reference object) are fully or partially occupied by the primary object.



(a) (b)

Figure 2.3: (a) A bounded region b and its 9-tiles; (b) a pair of regions (a, b).

Definition 2.20 (direction relation matrix) *Suppose bounded connected regions a and b are the primary object and reference object respectively. The* cardinal direction *of a to b is a* 3×3 *Boolean matrix*

$$\mathsf{dir}(a, b) = \begin{bmatrix} d^{NW} & d^{N} & d^{NE} \\ d^{W} & d^{O} & d^{E} \\ d^{SW} & d^{S} & d^{SE} \end{bmatrix}, \tag{2.46}$$

where for each tile name $\chi \in \{NW, N, NE, W, O, E, SW, S, SE\}$

$$d^{\chi} = 1 \quad \textit{iff} \quad a^{\circ} \cap \chi(b) \neq \varnothing, \tag{2.47}$$

where $\chi(b)$ *denotes the* χ*-tile of b.*

We say a 3×3 *Boolean matrix d is a* valid *CDC matrix, if there exist bounded connected regions a and b such that* $\mathsf{dir}(a, b) = d$*.*

Notation The cardinal direction relation of a to b can also be compactly represented in the form $\delta_1{:}\delta_2{:}\cdots{:}\delta_k$, where $\{\delta_1, \delta_2, \cdots, \delta_k\}$ is the set of tile names χ such that $d^{\chi} = 1$.

Take the two regions a, b in Figure 2.3 (b) as example. The cardinal direction relation of a to b is $N{:}NE{:}E$ and that of b to a is $W{:}O{:}SW{:}S$.

Note that not all 3×3 Boolean matrices are valid, i.e., some of them are not the direction relation matrices of any two regions.

Proposition 2.1 ([47]) *There are altogether 218 valid CDC matrices.*

Definition 2.21 (Cardinal Direction Calculus) *Let d be a valid CDC matrix. The* cardinal direction relation *with respect to d, is defined as $\{(a,b) \in \boldsymbol{U}_{\boldsymbol{CDC}} \times \boldsymbol{U}_{\boldsymbol{CDC}} : \mathsf{dir}(a,b) = d\}$. Write $\mathcal{B}_{\boldsymbol{CDC}}$ for the JEPD set of all 218 cardinal direction relations. The* Cardinal Direction Calculus *is defined to be $(\boldsymbol{U}_{\boldsymbol{CDC}}, \mathcal{B}^*_{\boldsymbol{CDC}})$, where $\mathcal{B}^*_{\boldsymbol{CDC}}$ is the Boolean algebra generated by $\mathcal{B}_{\boldsymbol{CDC}}$.*

Note that we will also use a valid matrix to denote a basic CDC relation, as the correspondence is clear.

2.4 Reasoning with qualitative calculi

A qualitative calculus defines a language for expressing qualitative spatial knowledge. Furthermore, a number of reasoning tasks can be formalised based on the language, among which the most important are the weak composition problem, the consistency problem, and the minimal labeling problem. The weak composition problem asks what the possible relations are between two entities a and c, if the relation between a and b, and that between b and c are given for an entity b. The consistency problem, which is more general, aims to decide whether a set of constraints concerning variables in set V are consistent, i.e., whether they can be satisfied simultaneously by an assignment of variables in V. The minimal labeling problem is related to the consistency problem. Given a set of constraints, the minimal labeling problem is to decide the best information about two variables we may infer. Now we explore the detail of these problems.

2.4.1 Weak Composition

A qualitative calculus provides a language for expressing knowledge that may be either complete or incomplete. For example, we may say in the RCC-8 language USA **EC** *Canada* to represent the fact that America and Canada share part of their boundaries, or say in the IA language $World\_War\_Two$ d $20century$ to represent that World War Two occurred and ended in the 20th century. Incomplete knowledge may be expressed by nonbasic relations, for example, $a \leq b$ stands for a is smaller or equals to b, where $\leq$ is a nonbasic relation in Point Algebra.

Now suppose we know the following facts expressed by the Interval Algebra.

- $World\_War\_Two$ d $20th\_century$,
- $Industry\_Revolution$ p $20th\_century$ (claiming that the industry revolution ended before the 20th century).

From the above knowledge, we can easily infer that the Industry Revolution should also ended before World War Two, or written in the IA language,

$$Industry\_Revolution \ \mathsf{p} \ World\_War\_Two.$$

The weak composition problem formalises this kind of simple reasoning task, i.e., in a given qualitative calculus, if the relation between objects a and b is α, and that between b and c is β, then what are the possible relations between a and c? The weak composition problem is essentially computing the weak compositions of relations in the following sense.

Definition 2.22 (weak composition) *Suppose $\mathcal{M} = (\boldsymbol{U}_{\mathcal{M}}, \mathcal{B}^*_{\mathcal{M}})$ is a qualitative calculus, where $\mathcal{B}^*_{\mathcal{M}}$ is a Boolean Algebra with atoms $\mathcal{B}_{\mathcal{M}}$. The* weak composition *of two relation $\alpha, \beta \in \mathcal{B}^*_{\mathcal{M}}$, denoted by $\alpha \circ_w \beta$[1], is defined as*

$$\alpha \circ_w \beta := \bigcup \{\gamma \in \mathcal{B}_{\mathcal{M}} : \exists a,b,c \ \ s.t. \ (a,b) \in \alpha, (b,c) \in \beta, (a,c) \in \gamma\} \tag{2.48}$$
$$= \bigcup \{\gamma \in \mathcal{B}_{\mathcal{M}} : (\alpha \circ \beta) \cap \gamma \neq \varnothing\}. \tag{2.49}$$

Note that a qualitative calculus is not necessarily closed under composition. The weak composition is the smallest relation in the calculus that contains the actual composition. In other words, weak composition is the best approximation to the actual composition in the language provided by the qualitative calculus.

Now the composition problem can be rephrased as 'what are the weak compositions of relations in the qualitative calculi?' Since a qualitative calculus has only a finite number of relations, we can compute (usually by case analysis or theoretical deduction) and record all the weak compositions in advance. In fact, it is sufficient

[1]In some literature, the weak composition is written as $\alpha \diamond \beta$.

to compute the weak compositions of basic relations only. This is because, the weak composition is distributive over $\cup$ (the set union operation), i.e., for $\alpha, \beta, \gamma \in \mathcal{B}^*_{\mathcal{M}}$,

$$\alpha \circ_w (\beta \cup \gamma) = (\alpha \circ_w \beta) \cup (\alpha \circ_w \gamma), \tag{2.50}$$

$$(\alpha \cup \beta) \circ_w \gamma = (\alpha \circ_w \gamma) \cup (\beta \circ_w \gamma). \tag{2.51}$$

Therefore, the weak composition of two general relations can be obtained by computing the union of a number of weak compositions of basic relations. This greatly reduces the number of weak compositions we need to record in advance (from 2^{2m} down to m^2, where m is the number of basic relations in the qualitative calculus). The *composition table*, or CT for short, is the table that stores the weak compositions of all basic relations in a qualitative calculus.

When a new qualitative calculus is proposed, the first problem is very likely to be how to compute its composition table. The composition tables of Point Algebras, Interval Algebra and RCC-8/RCC-5 Algebras have been computed by the QSTR community (see e.g., [28]). How to compute the CT automatically or semi-automatically is still a challenging problem.

2.4.2 The Consistency Problem

We have seen in the previous section that very simple reasoning tasks can be carried out via the composition problem. This section introduces a more general and powerful reasoning problem, viz., the consistency problem, which can be viewed as a *qualitative* constraint satisfaction problem.

The classical CSP (cf. Definition 2.9) the domains D_i are all finite, therefore each constraint c_{ij} is also a finite set, because it is a subset of the Cartesian product of two finite sets. This means that a CSP instance can be specified by enumerating the elements in domains D_i and constraints c_{ij}. The consistency problem in QSTR differs from the classical CSP in the following aspects, where a qualitative calculus is presumed:

- First, the domain is usually infinite. In a qualitative constraint satisfaction problem, it is assumed that each variable can be interpreted by any value in the universe.

- Second, for the consistency problem in QSTR, the relation in a constraint is restricted to the relations in the qualitative calculus, whereas for classical CSP, the relation is explicitly and arbitrarily specified.

The second difference is reasonable: in most cases we cannot enumerate all instances that satisfy a qualitative property (e.g. partially overlapping), as such instances maybe infinite. However, the first assumption may be questioned, as it is quite possible that our enquiries involve *known* entities like landmarks. This issue will be discussed in Chapter 6. We now give the formal definition of the consistency problem in QSTR.

We give the concept of *constraint*, the basic element in the consistency problem:

Definition 2.23 (constraint) *Suppose* $\mathcal{M} = (\boldsymbol{U}_{\mathcal{M}}, \mathcal{B}^*_{\mathcal{M}})$ *is a qualitative calculus. A constraint (in* $\mathcal{M}$*) is a formula of the form* $v_i \alpha v_j$*, where* v_i *and* v_j *are spatial or temporal variables that take values in* $\boldsymbol{U}_{\mathcal{M}}$ *and* α *is a relation in* $\mathcal{B}^*_{\mathcal{M}}$*. If* α *is a basic relation,* $v_i \alpha v_j$ *is called a* basic constraint.

Note that here v_i and v_j are variable symbols, which are different from entities in the universe of the qualitative calculus. Spatial or temporal variables are supposed to be interpreted by entities in the universe that satisfy the corresponding relations in the constraints.

The consistency problem over $\mathcal{M}$ can then be formulated as below.

Definition 2.24 (consistency problem [12]) *Let* $\mathcal{M} = (\boldsymbol{U}_{\mathcal{M}}, \mathcal{B}^*_{\mathcal{M}})$ *be a qualitative calculus. Suppose* $\mathcal{S}$ *is a subset of* $\mathcal{B}^*_{\mathcal{M}}$*. The consistency problem* $\textsc{cspsat}_{\mathcal{M}}(\mathcal{S})$*, or simply* $\textsc{cspsat}(\mathcal{S})$ *if* $\mathcal{M}$ *is clear in the context, is the problem defined as:*

> ***Instance****: A 2-tuple* (V, Γ)*. Here* V *is a finite set of variables, and* Γ *is a finite set of constraints of the form* $x\alpha y$*, where* $\alpha \in \mathcal{S}$ *and* $x, y \in V$*. Note that if the variable set* V *is clear in the context, we may simply use* Γ *to refer the CSP instance.*
>
> ***Question****: Deciding the* consistency *(or* satisfiability*) of* (V, Γ)*. That is, deciding whether there exists an interpretation* $\mathcal{I} : V \to \boldsymbol{U}_{\mathcal{M}}$*, s.t., all constraints in* Γ *are satisfied, i.e., for each constraint* $x\alpha y$ *in* Γ*, the basic relation between* $\mathcal{I}(x)$ *and* $\mathcal{I}(y)$ *is contained in relation* α*.*

If an interpretation $\mathcal{I}$ *satisfies all constraints in* Γ*, we say* $\mathcal{I}$ *is a solution of* Γ*.*

Notation We usually suppose the variable set $V = \{v_1, v_2, \cdots, v_n\}$. In this case, we also use simply $(\mathcal{I}(v_1), \cdots, \mathcal{I}(v_n))$ to denote the interpretation $\mathcal{I}$.

Definition 2.25 (equivalent CSP instances) *Let $\mathcal{M} = (\boldsymbol{U}_{\mathcal{M}}, \mathcal{B}^*_{\mathcal{M}})$ be a qualitative calculus. Suppose $\mathcal{S}$ is a subset of $\mathcal{B}^*_{\mathcal{M}}$. Two* $\text{CSPSAT}_{\mathcal{M}}(\mathcal{S})$ *instances (V,Γ) and (V',Γ') are called* equivalent*, if (i) $V = V'$, and (ii) an interpretation of V is a solution of (V,Γ) iff it is a solution of (V',Γ').*

In other words, equivalent instances are semantically the same. As the consistency problem is essentially a semantic problem, we will focus on a special kind of instance, i.e., constraint networks.

Definition 2.26 (constraint network) *Suppose (V,Γ) is an instance of* $\text{CSPSAT}_{\mathcal{M}}(\mathcal{S})$*. We say (V,Γ) is a* constraint network *over $\mathcal{S}$ (with respect to qualitative calculus $\mathcal{M}$), if Γ contains exactly one constraint of the form $x\alpha y$ for each pair of variables (x,y) in V. Furthermore, we say a constraint network (V,Γ) is a* basic *network, if Γ only contains basic constraints.*

Notation We usually write $\Gamma = \{v_i \alpha_{ij} v_j\}_{i,j=1}^n$ for the constraint set of a constraint network (V,Γ), where we assume $V = \{v_1, v_2, \cdots, v_n\}$.

Proposition 2.2 ([71]) *Suppose (V,Γ) is an instance of* $\text{CSPSAT}_{\mathcal{M}}(\mathcal{S})$*. Then we can get an equivalent constraint network (V,Γ') over $\mathcal{S}'$ in polynomial time, where $\mathcal{S}'$ is the closure of $\mathcal{S} \cup \{?\}$ with respect to the intersection operation (recall that $?$ is the universal relation).*

Next we introduce the concepts of refinement and scenario, followed by the motivation.

Definition 2.27 (refinement, scenario) *Let $\mathcal{M}$ be a qualitative calculus. Suppose (V,Γ) and (V,Γ') are two constraint networks over the same variable set V in $\mathcal{M}$, where $\Gamma = \{v_i \alpha_{ij} v_j : 1 \leq i,j \leq n\}$ and $\Gamma' = \{v_i \beta_{ij} v_j : 1 \leq i,j \leq n\}$. We say (V,Γ') is a* refinement *of (V,Γ), if for any $1 \leq i,j \leq n$ it holds that $\beta_{ij} \subseteq \alpha_{ij}$. We say (V,Γ') is a* scenario *of (V,Γ) if it is a basic constraint network.*

A refinement of a constraint network is a network with stronger constraints. Scenarios are the finest refinements of the original network. Although 'scenario' is defined as a syntax concept, it has a direct semantic connection with constraint network, as stated in the following proposition.

Proposition 2.3 *A constraint network is consistent iff it has a consistent scenario.*

In general, constraint networks may still be difficult to solve. In this case, we would tend to solve the CSP over basic constraint networks first. If we find an algorithm for the CSP over basic constraint networks, the CSP over general constraint networks can be solved by backtracking all the scenarios and Proposition 2.3. Furthermore, if a polynomial algorithm is found for basic constraint networks, then the general CSP is consequently in NP [12]

For several important qualitative calculi, polynomial algorithms have been found for the CSP over basic constraint networks (i.e., constraints restricted to $\mathcal{S} = \mathcal{B}_{\mathcal{M}}$). However, the CSP over general constraint networks (i.e., constraints restricted to $\mathcal{S} = \mathcal{B}^*_{\mathcal{M}}$) turns out to be NP-Complete. The *maximal tractable subclasses* have been identified for several qualitative calculi, in the following sense.

Definition 2.28 (maximal tractable subclass) *Suppose $\mathcal{M} = (\boldsymbol{U}_{\mathcal{M}}, \mathcal{B}^*_{\mathcal{M}})$ is a qualitative calculus, and $\mathcal{S}$ is a subset of $\mathcal{B}^*_{\mathcal{M}}$. We say $\mathcal{S}$ is a* tractable subclass*, if the consistency problem* $\textsc{cspsat}_{\mathcal{M}}(\mathcal{S})$ *has a polynomial algorithm. We say $\mathcal{S}$ is an* intractable subclass*, if* $\textsc{cspsat}_{\mathcal{M}}(\mathcal{S})$ *is NP-hard. We say $\mathcal{S}$ is a* maximal tractable subclass*, if $\mathcal{S}$ is tractable and its any proper superset is intractable.*

If a qualitative calculus is proved to be intractable, an important problem is to decide its maximal tractable subclasses which are not necessarily unique.

Remark 2.1 *Note that, strictly speaking, the consistency problem is different from the problem of* computing a solution *of a set of constraints. The consistency problem is weaker, since we need not actually work out a solution to show the constraints are satisfiable. This is like it is sufficient to check whether $b^2 - 4ac \geq 0$ to decide whether $ax^2 + bx + c = 0$ has real roots. However, for basic constraint networks in qualitative calculi including Point Algebra, Interval Algebra and RCC-8, if the network is satisfiable, we can always construct a solution in polynomial time [61, 87].*

As the central reasoning problem, the consistency problem for different calculi has been investigated extensively in the literature (see e.g. [3, 106, 77, 86, 67, 8]). We here summarise the results for qualitative calculi introduced above. [106] proves that the consistency problem for the totally ordered Point Algebras can be solved in $O(n^2)$, where n is the number of variables. For other qualitative calculi including the general PAs, IA, CRA, and RCC-8, the consistency problems are NP-hard. Nonetheless, maximal subclasses of IA and RCC-8 have been identified in [77, 23, 86]. In particular, for constraint networks from a tractable subclass in PA, IA, RCC-5/RCC-8, the consistency problem can be decided in $O(n^3)$ time by checking its path-consistency (cf. [57]) in the following sense.

Definition 2.29 (path-consistency) *Let (V, Γ) be a constraint network in some qualitative calculus, where $V = \{v_1, \cdots, v_n\}$ and $\Gamma = \{v_i \alpha_{ij} v_j\}_{i,j=1}^n$. We say (V, Γ) is path-consistent, if for any v_i, v_j, v_k in V, we have*

$$\alpha_{ij} = \alpha_{ji}^{\sim},$$

$$\alpha_{ij} \subseteq \alpha_{ik} \circ_w \alpha_{kj},$$

where $\alpha \circ_w \beta$ is the weak composition of α, β in $\mathcal{M}$, as defined in Definition 2.22.[1]

If a constraint network is not path-consistent, say, α_{ij} is not contained in $\alpha_{ik} \circ_w \alpha_{kj}$ for some i, j, k, we may replace α_{ij} with $\alpha_{ij} \cap (\alpha_{ik} \circ_w \alpha_{kj})$. This clearly does not change the semantic of the constraint network (i.e., the refined network is equivalent to the original one). Based on this observation, *composition-based reasoning* is developed, via the *Path Consistency Algorithm* (PCA). PCA iteratively applies the following updating rules until a constraint becomes empty, or a non-empty fix point (which is a path-consistent network) is reached.

$$\alpha_{ij} \leftarrow \alpha_{ij} \cap \alpha_{ji}^{\sim}, \tag{2.52}$$

$$\alpha_{ij} \leftarrow \alpha_{ij} \cap \alpha_{ik} \circ_w \alpha_{kj}. \tag{2.53}$$

Note that if a constraint becomes empty, we may infer that the constraint network is inconsistent, because the PCA reserves consistency. On the other hand, if a non-empty fix point is reached in PCA, we cannot ensure that the original constraint network is consistent. However, it has been shown that the existence of non-empty fix point implies consistency for qualitative calculi including IA, the Point Algebra, Rectangle

[1]Note that we assume the qualitative calculus is closed under converse. If not, the first condition should be replaced with α_{ij} is contained in the closure of $\alpha_{ji}^{\sim}$ (i.e., the smallest relation that contains $\alpha_{ji}^{\sim}$ in the qualitative calculus).

Algebra and RCC-5/8, when restricted to tractable subclasses. For intractable subclasses in these qualitative calculi, we may also use PCA and the backtracking method to solve the consistency problem (e.g., the qualitative constraint solvers in [108, 110]).

2.4.3 Strongest Implied Relation and Minimal Labeling Problem

In applications, we are often interested in inferring the best knowledge about the relation between two certain entities from a given knowledge base. In QSTR, a knowledge base may be expressed by a set of constraints, and the possible relations between two entities under such constraints is called the *strongest implied relation*. The problem of computing the strongest implied relation is called the minimal labeling problem. In fact, the composition problem may be viewed as a special minimal labeling problem that only involves three variables. Now we provide formal definitions.

Definition 2.30 (feasible constraint, strongest implied relation) *Suppose $\mathcal{M}$ is a qualitative calculus, V is a set of variables and Γ is a set of constraint. For a constraint $x\alpha y$ in Γ, and a basic relation β in $\mathcal{M}$, β is called a* feasible relation *between x and y, if it is contained in α and there is a solution $\mathcal{I}$ of (V, Γ) such that relation β holds between $\mathcal{I}(x)$ and $\mathcal{I}(y)$. The* strongest implied relation *between x and y is the union of all feasible relation between them.*

Given a set of variables V and a set of constraints Γ, the *minimal labeling problem* is to determine for each pair of variables in V the strongest implied relation between them. It is known that the minimal labeling problem and the consistency problem are equivalent with respect to Turing-reductions [107]. That is to say, if we have a polynomial algorithm that solves one of them, we may also design a polynomial algorithm that solves another. Furthermore, the two problems are still equivalent if the constraints are subject to a subclass $\mathcal{S}$, provided $\mathcal{S}$ contains all the basic relations [77].

Given a constraint network, if we replace each constraint $x\alpha y$ with constraint $x\beta y$, where β is the strongest implied relation between x and y, then we get a minimal network in the following sense (cf. [71]).

Definition 2.31 (minimal network) *Let $\Gamma = \{v_i\alpha_{ij}v_j\}_{i,j=1}^n$ be a constraint network in a qualitative calculus. We say Γ is* minimal*, if for any $1 \leq i, j \leq n$, the strongest implied relation between v_i and v_j is α_{ij} itself.*

A minimal network may be regarded as the most compact representation of a certain knowledge base. There are several problems regarding minimal networks:

(i) How to decide whether a network is minimal? (the *minimality problem*)

(ii) How to compute the equivalent minimal network of a given network?

(iii) How to obtain one (or all) solution(s) of a constraint network which is known to be minimal?

The minimality problem can be decided in polynomial time for *tractable* subclasses. This is because, as stated above, we can compute the strongest implied relation between each pair of variables in polynomial time for constraint network from a tractable subclass, and thus we can check the minimality of the network by definition in polynomial time. However, the complexity of the minimality problem remains open if the qualitative calculus is not tractable.

The second problem is greater than the minimality problem, in the following sense. Suppose we are able to compute the equivalent minimal network, say Γ', of a given network Γ in polynomial time, then we may also decide the minimality of Γ in polynomial time, by simply compare Γ and Γ'. Therefore, computing the equivalent minimal network is in general NP-hard.

We observe that the first two problems can be reduced to the consistency problem. However, the third problem has not yet been considered. Although several authors tend to believe that this is simple (i.e., given a minimal network, we may compute a solution of it in a backtracking-free manner), this thesis will show this is not the case.

2.5 Solution construction for RCC-5/RCC-8 networks

When we claim that certain conditions are sufficient for a network to be consistent, a most common way is to construct a solution of the network. This section briefly reviews previous results of [55, 56, 87] about constructing a solution of basic consistent RCC-5 or RCC-8 networks. This idea will also be used in the solution construction for fuzzy RCC-8 networks and in that for RCC-5/RCC-8 networks in the extended framework of qualitative CSP.

First we discuss the RCC-5 case. Suppose (V, Γ) is a basic RCC-5 constraint network which is consistent, where $V = \{v_1, v_2, \cdots, v_n\}$ and $\Gamma = \{v_i \alpha_{ij} v_j\}_{i,j=1}^n$. We initialise each variable with a starting region, and then modify the region to meet the PO constraints and the PP constraints in two steps. A number of *base regions* will be used to compose the regions to be constructed in the procedure. The base regions can be easily selected because the only requirement is that they are pairwise disjoint. For example, we may use disks $B(P_{ij}, r)$ as base regions, where $B(P_{ij}, r)$ is the disk centred at point $P_{ij} = (i, j)$ with radius r. In detail,

- $a_i \leftarrow$ some base region that has not been used before (e.g., $B(P_{ii}, 1/4)$, the disk centred at (i, i) with radius $1/4$).
- If $(v_i \mathbf{PO} v_j) \in \Gamma$, then $a_i \leftarrow a_i \cup r$, $a_j \leftarrow a_j \cup r$, where r is an unused base region (e.g., $B(P_{ij}, 1/4)$).
- $a_i \leftarrow a_j \cup \bigcup \{a_j : v_j \mathbf{PP} v_j \in \Gamma\}$.

It can be checked that the assignment is a solution of Γ, if Γ is consistent. Note that we need only $O(n^2)$ base regions, and thus the procedure takes $O(n^2)$ time.

In the RCC-8 case, the EC constraints and the NTPP constraints will make the construction slightly complicated. For the EC constraints, we need a pair of base regions that are externally connected. For the NTPP constraints, we need a series of n nested regions (called *ntpp-shells*) instead of a base region, where n is the number of the variables in the constraint network. We use r or r^0 to denote the core of the nested regions and use r^k to denote the $(k+1)$-th innermost region in the series. Now suppose (V, Γ) is a consistent basic RCC-8 constraint network. We first define the ntpp-level l_i of each variable v_i as the maximum length of the ntpp chains contained in Γ that ends with variable v_i as in the following equation, where $\max \varnothing$ is considered as 0.

$$l_i = \max\{k : \exists v_{j_0}, \cdots v_{j_k} \text{ such that } v_{j,l} \textbf{ NTPP } v_{j,l+1} \in \Gamma \text{ and } v_{j_k} = v_i.\} \tag{2.54}$$

Note that the consistency guarantees that the ntpp-levels are well-defined, as the **NTPP** constraints cannot contain cycles. A solution may then be constructed by the following procedure.

- $a_i \leftarrow$ an unused base region (e.g., $B(P_{ii}, 1/4)$).
- If $(v_i \mathsf{EC} v_j) \in \Gamma$, then $a_i \leftarrow a_i \cup r$, $a_j \leftarrow a_j \cup r'$, where r and r' are two unused base region that are externally connected (e.g., r is the left half of disk $B(P_{ij}, 1/4)$ and r' is the right half of the same disk).
- If $(v_i \mathbf{PO} v_j) \in \Gamma$, then $a_i \leftarrow a_i \cup r$, $a_j \leftarrow a_j \cup r$, where r is a unselected base region.
- $a_i \leftarrow a_j \cup \bigcup\{a_j : v_j \mathbf{TPP} v_j \in \Gamma \text{ or } v_j \mathbf{NTPP} v_j \in \Gamma\}$.
- $a_i \leftarrow \bigcup\{r^{l_i} : r^0 \text{ is a base region and } r^0 \subseteq a_i\}$.

In this case, we need $O(n^2)$ ntpp-shells in which each ntpp-shell contains n-nested regions. Therefore $O(n^3)$ regions are used to compose the regions in the solution, and thus the time complexity is $O(n^3)$.

2.6 Chapter Summary

This chapter has provided the necessary background for later discussion. The concept of qualitative calculus has been introduced, followed by the definitions of qualitative calculi discussed in the thesis. The reasoning problems in QSTR, including the weak composition problem, the consistency problem and the minimal labeling problem, were given. A solution construction method for basic RCC-5/RCC-8 consistent constraint networks was reviewed at the end of this chapter.

Chapter 3

Computing Composition Tables Semi-Automatically

3.1 Introduction

Since Allen's seminal work of Interval Algebra (IA) [2, 3], qualitative calculi have been widely used to represent and reason about temporal and spatial knowledge. In the past three decades, dozens of qualitative calculi have been proposed in the AI area "Qualitative Spatial & Temporal Reasoning" and Geographic Information Science. For example, The Netherlands is *west* of Germany, The Alps *partially overlap* Italy, I have an appointment today with my doctor *followed by* a check-up.

Given a set of qualitative knowledge, new knowledge can be derived by using constraint propagation. Consider an example in RCC-5 [84]. Given that The Alps *partially overlap* Italy and Switzerland, and Italy is a *proper part of* the European Union (EU), and Switzerland is *discrete from* the EU, we may infer that The Alps *partially overlap* the EU. The above inference can be obtained by using *composition-based reasoning*, which is typically carried out by the *Path-Consistency Algorithm* (PCA). Please refer to Section 2.4.2 for details of composition-based reasoning and PCA.

To apply the Path-Consistency Algorithm, one prerequisite is that we must know the weak composition of any two basic relations in the qualitative calculus. For efficiency, it is common to store these weak compositions in a $n \times n$ table, where n is the number of the basic relations in the calculus. This table, called the *composition ta-*

ble (CT), is usually obtained by manually checking the consistency of $\{x\alpha y, y\beta z, x\gamma z\}$ for each triple of basic relations (α, γ, β). When the concerned qualitative calculus contains dozens or even hundreds of basic relations, this consistency-based method is undesirable and error-prone. This problem is identified by Cohn in [13] as a challenge for computer scientists.

This problem remains a challenge today. We here consider several examples. The Interval Algebra and the RCC-8 algebra contain, respectively, 13 and 8 basic relations. Their CTs were established manually. If a calculus contains a hundred basic relations, we need to determine the consistency of one million such basic networks. This is manually impossible. The $\mathcal{OPRA}$ calculi and the CDC are large qualitative spatial calculi that have drawn increasing interest. $\mathcal{OPRA}_m$ contains $4m \times (4m + 1)$ (i.e. 72, 156, 272 for $m = 2, 3, 4$, respectively) basic relations [72], while the CDC contains 218 basic relations [45]. Sometimes we need ingenious and special methods to establish the CT for such a calculus. For the $\mathcal{OPRA}$ calculi, the algorithm presented in the original paper [72] contains gaps and errors. An algorithm was later presented in [40]. This algorithm is lengthy and cumbersome. Another simple algorithm has also been proposed recently [75]. Given the huge number of basic relations of $\mathcal{OPRA}_m$, the validity of these algorithms needs further verification. As for the CDC, [45] first studied the weak composition. Later, [100] noticed errors in Goyal's method and presented a new algorithm to compute the weak composition. Unfortunately, in several cases, their algorithm does not generate the correct weak composition (see [67]).

In this chapter, we respond to this challenge and propose a semi-automatic approach to generate CT for general qualitative calculi. In the remainder of this chapter, we first recall basic notions and propose our algorithm for computing CT in Section 2, and then apply our method to IA, INDU, $\mathcal{OPRA}_1$, and $\mathcal{OPRA}_2$ in Section 3. Section 4 concludes the chapter.

3.2 A Random Method for Computing CT

Recall that in a qualitative calculus $\mathcal{M} = (\mathbf{U}, \mathcal{B}^*)$, the *weak composition* of two basic relations α and β in $\mathcal{M}$, denoted as $\alpha \circ_w \beta$, is defined as the smallest relation in $\mathcal{M}$ which contains $\alpha \circ \beta$. The weak composition operation of $\mathcal{M}$ can be summarised in an $n \times n$ table, where n is the cardinality of $\mathcal{B}$, and the cell specified by α and β contains all basic relations γ in $\mathcal{B}$ such that $\gamma \subseteq \alpha \circ_w \beta$. For example, the CT of Point Algebra is given in Table 3.1.

$\circ$	$<$	$=$	$>$
$<$	$<$	$<$	$*$
$=$	$<$	$=$	$>$
$>$	$*$	$>$	$>$

Table 3.1: The CT of the Point Algebra, where $*$ is the universal relation

Definition 3.1 *Suppose $\mathcal{M}$ is a qualitative calculus on $\mathbf{U}$ with basic relation set $\mathcal{B}$. For basic relations α,β,γ, we call $\langle\alpha,\gamma,\beta\rangle$ a* composition triad*, or* c-triad*, if $\gamma \subseteq \alpha \circ_w \beta$.*

We can determine if a 3-tuple is a c-triad as follows.

Proposition 3.1 *A 3-tuple $\langle\alpha,\gamma,\beta\rangle$ of basic relations in $\mathcal{M}$ is a c-triad iff $\gamma \cap \alpha \circ \beta \neq \varnothing$, which is equivalent to saying that the basic constraint network*

$$\{x\alpha y, y\beta z, x\gamma z\} \tag{3.1}$$

is consistent, i.e. it has a solution in U.

To compute the weak composition of α and β, one straightforward method is to find all basic relations γ such that $\langle\alpha,\gamma,\beta\rangle$ is a c-triad. In what follows, we address this as the *consistency approach* to computing the composition table.

In this section, we propose a novel approach to compute the composition table of $\mathcal{M}$. The approach is based on the observation that each triple of objects in **U** derives a unique c-triad.

Proposition 3.2 *Suppose a,b,c are three objects in $\mathbf{U}$. Then*

$$\langle\rho(a,b),\rho(a,c),\rho(b,c)\rangle$$

is a c-triad, where $\rho(x,y)$ is the basic relation in $\mathcal{M}$ that relates x to y.

If we consider all permutations of a,b,c, then obviously the following five more (different or not) c-triads can be generated:

$$\begin{aligned}
&\langle \rho(a,c), \rho(a,b), \rho(c,b)\rangle,\\
&\langle \rho(b,a), \rho(b,c), \rho(a,c)\rangle,\\
&\langle \rho(b,c), \rho(b,a), \rho(c,a)\rangle,\\
&\langle \rho(c,a), \rho(c,b), \rho(a,b)\rangle,\\
&\langle \rho(c,b), \rho(c,a), \rho(b,a)\rangle.
\end{aligned}$$

To compute the CT of $\mathcal{M}$, the idea is to randomly choose a triple of elements in $\mathbf{U}$ and then compute and record the c-triads related to these objects in a dynamic table. Continuing in this way, we will obtain more and more c-triads until the dynamic table becomes stable after sufficiently many loops. The basic algorithm is given in Algorithm 1, where D is a subdomain of $\mathcal{M}$, Ψ decides when the procedure terminates, TRIAD records the number of c-triads obtained when the procedure terminates, and LASTFOUND records the time when the last triad is first recorded. For a calculus with unknown CT, the condition may be assigned with the form LOOP $\leq 1,000,000$ (i.e., the algorithm loops one million times), or LOOP $\leq$ LASTFOUND $+ 100,000$ (i.e., until no new c-triad is found in the last one hundred thousand loops), or both. If the CT is known, then the boundary condition could be set to TRIAD $< N$ to save time, where N is the number of c-triads of the calculus. We note that the converse of a basic relation is not necessarily a relation in $\mathcal{M}$.

We offer further explanations here. Suppose $\mathcal{M}$ is a qualitative calculus on $\mathbf{U}$. Recall $\mathbf{U}$ is often an infinite set. We need to find a finite subdomain D of $\mathbf{U}$, as computers only deal with numbers with finite precision. Once D is chosen, we run the loop, say, one million times. Therefore, one million instances of triples of elements in D are generated. We then record all computed c-triads in a dynamic table. It is reasonable to claim that the table is stable if no new entry has been recorded after a long time (e.g. as long as the time has passed to obtain all recorded c-triads). Because D is finite, Algorithm 1 will generate a stable table after a sufficiently large number of iterations.

We observe that a finite subdomain D may restrict the possible c-triads if it is selected inappropriately. We introduce a notion to characterise the appropriateness of

Input: A subdomain D of $\mathcal{M}$, and a boundary condition Ψ related to $\mathcal{M}$

Output: The Composition Table CT of $\mathcal{M}$

Initialise CT;
LOOP $\leftarrow 0$;
TRIAD $\leftarrow 0$;
LASTFOUND $\leftarrow 0$;
while Ψ **do**
    LOOP $\leftarrow$ LOOP $+ 1$;
    Generate triple of objects (a, b, c) randomly;
    $\alpha \leftarrow$ the basic relation between a and b;
    $\beta \leftarrow$ the basic relation between b and c;
    $\gamma \leftarrow$ the basic relation between a and c;
    $\alpha' \leftarrow$ the basic relation between b and a;
    $\beta' \leftarrow$ the basic relation between c and b;
    $\gamma' \leftarrow$ the basic relation between c and a;
    for $\langle r, s, t\rangle \in \{\langle\alpha,\gamma,\beta\rangle, \langle\alpha',\beta,\gamma\rangle, \langle\gamma,\alpha,\beta'\rangle, \langle\beta,\alpha',\gamma'\rangle, \langle\beta',\gamma',\alpha'\rangle, \langle\gamma',\beta',\alpha\rangle\}$ **do**
        if $\langle r, s, t\rangle$ *is not in* CT **then**
            Record triad $\langle r, s, t\rangle$ to CT;
            TRIAD $\leftarrow$ TRIAD $+ 1$; LASTFOUND $\leftarrow$ LOOP;
        end
    end
end
return CT.

Algorithm 1: Computing the Composition Table of $\mathcal{M}$

a subdomain.

Definition 3.2 *Suppose $\mathcal{M}$ is a qualitative calculus defined on the universe $\boldsymbol{U}$. A nonempty subset D of $\boldsymbol{U}$ is called a* 3-complete *subdomain of $\mathcal{M}$ if each consistent basic network as specified in Eq. 3.1 has a solution in D.*

If D is a 3-complete subdomain, then, for each c-triad $\langle\alpha,\gamma,\beta\rangle$, there are a,b,c in D such that $(a,b)\in\alpha$, $(b,c)\in\beta$, and $(a,c)\in\gamma$. Therefore, to determine the CT of $\mathcal{M}$, we need only consider instances of triples in D.

Nevertheless, how should we determine whether a subdomain is 3-complete? There is no general answer for arbitrary qualitative calculi. For a particular calculus, e.g. IA, this can be verified by formal analysis. Note that a superset of a 3-complete subdomain is also 3-complete. To make sure a chosen subdomain D is 3-complete, we often apply the algorithm on several of its supersets at the same time. If the same number is generated for all subdomains, we tend to believe that D is 3-complete and the generated table is the CT of $\mathcal{M}$. Naturally, a formal proof is still necessary to guarantee the 3-completeness of D. Once an appropriate D is chosen, this is usually not very difficult.

Even if a CT of $\mathcal{M}$ has been somehow obtained, our method can be used to verify its correctness. Double-checking is necessary, since computing the CT is error-prone. It should be stressed that c-triads computed by Algorithm 1 are always true c-triads of $\mathcal{M}$. If there is a recorded c-triad that does not appear in the previously given table, something must be wrong.

Another thing we should keep in mind is how to generate a triple of elements (a,b,c) from D. The distribution over D may affect the efficiency of the algorithm. Assuming that we have very limited knowledge of the calculus $\mathcal{M}$, it is natural to take a,b and c independently with respect to the uniform distribution. We note that the better we understand the calculus, the more appropriate is the distribution we may choose.

To increase the efficiency of the algorithm, we sometimes use the algebraic properties of the calculus. For example, if the identity relation id is a basic relation, then by $\alpha\circ_w id=\alpha=id\circ_w\alpha$ and $id\subseteq\alpha\circ_w\alpha^{\sim}$, we need not compute the c-triads involving id, where $\alpha^{\sim}$ is the converse of α. As another example, suppose that the calculus is closed under converse, i.e. the converse of a basic relation is still a basic relation. Then in

Algorithm 1 we need only compute α, β, γ. The other relations and c-triads can be obtained by replacing α', β', γ' by, respectively, $\alpha^{\sim}, \beta^{\sim}, \gamma^{\sim}$.

3.3 Experimental Results

This section applies the proposed algorithm to three qualitative calculi: IA, INDU and $\mathcal{OPRA}_m$. All experiments were conducted on a 3.16 GHZ Intel Core 2 Duo CPU with 3.25 GB RAM running Windows XP.

3.3.1 The Interval Algebra and the INDU Calculus

We start with Interval Algebra, the best known qualitative calculus. The CT for IA was computed in 1983 in Allen's famous work. When applying Algorithm 1 to IA, we do not consider all intervals. Instead, we restrict the domain to the set of all intervals contained in $[0, M)$ that have integer nodes

$$D_M = \{[p, q] | p, q \in \mathbb{Z}, 0 \le p < q < M\},$$

and use a uniform distribution to choose random intervals. It is easy to see that the size of the domain is $M(M-1)/2$. Note that to converge quickly and generate all entries, we need to choose an appropriate M. Table 3.2 summarises the results for $M = 4$ to $M = 20$. In the experiment, we generate one million instances of triples of elements for each domain D_M. In all cases the dynamic table becomes stable in less than 50,000 loops. When the table becomes stable, the number of triads computed is not always correct (that is, 409). This is mainly because the domain is small. For M bigger than or equal to six, however, we always obtain the correct number of triads. The loops needed (i.e. LastFound) vary from less than a thousand to more than 43 thousand (see Table 3.2). In general, the smaller the domain is, the more efficient the algorithm is.

Definition 3.3 (INDU calculus) *The INDU calculus [81] is a refinement of IA. The duration of an interval $a = [x, y]$ is defined as $d(a) := y - x$. For IA relation $\alpha = \mathsf{p}, \mathsf{m}, \mathsf{o}$*

M	4	5	6	7	8	9	10	11	12
TRIAD	139	319	409	409	409	409	409	409	409
LASTFOUND	92	629	1501	878	2111	3517	728	697	932

M	13	14	15	16	17	18	19	20
TRIAD	409	409	409	409	409	409	409	409
LASTFOUND	11212	20249	7335	4343	3632	17862	5533	43875

Table 3.2: Implementation for IA, where TRIAD is the number of c-triads recorded by running the algorithm on D_M for $M = 4$ to $M = 20$, LASTFOUND is the loop when the last triad is first recorded

and their converses, INDU calculus refines α into three relations

$$\alpha^< := \{(a,b) : (a,b) \in \alpha, d(a) < d(b)\}, \tag{3.2}$$

$$\alpha^= := \{(a,b) : (a,b) \in \alpha, d(a) = d(b)\}, \tag{3.3}$$

$$\alpha^> := \{(a,b) : (a,b) \in \alpha, d(a) > d(b)\}. \tag{3.4}$$

The INDU calculus is generated by above 18 relations and the other 7 basic relations of IA that have no proper sub-relations.

INDU is quite unlike IA. For example, it is not closed under composition, and a path-consistent basic network is not necessarily consistent [5].

Applying our algorithm to INDU, we use the same subdomain D_M as for IA. From Table 3.3 we can see that D_6 is no longer 3-complete: more than 1000 c-triads do not appear in the stable table. The table becomes complete in D_{11}, which has 2053 c-triads. The 3-completeness of D_{11} is confirmed by the following proposition.

M	6	7	8	9	10	11	12	13
TRIAD	1045	1531	1819	1987	2041	2053	2053	2053
LASTFOUND	3766	5753	10417	35201	35891	25031	12512	27728

M	14	15	16	17	18	19	20
TRIAD	2053	2053	2053	2053	2053	2053	2053
LASTFOUND	17223	24578	14758	22491	29034	49693	19772

Table 3.3: Implementation for INDU, where TRIAD is the number of c-triads recorded by running the algorithm on D_M for $M = 6$ to $M = 20$, LASTFOUND is the loop when the last triad is first recorded

Proposition 3.3 *The INDU calculus has at most 2053 c-triads.*

Proof For any three INDU relations $\alpha^{*1}, \beta^{*2}, \gamma^{*3}$ where $*1, *2, *3 \in \{<, =, >\}$, it is easy to see that $(\alpha^{*1}, \gamma^{*3}, \beta^{*2})$ is a valid c-triad of INDU only if (α, γ, β) is valid c-triad of IA and $(*1, *3, *2)$ is a valid c-triad of PA. We note that for IA relations in $\{\mathsf{d}, \mathsf{s\,f}, \mathsf{eq}, \mathsf{si}, \mathsf{fi}, \mathsf{di}\}$, only $\mathsf{d}^<, \mathsf{s}^<, \mathsf{f}^<, \mathsf{eq}^=, \mathsf{si}^>, \mathsf{fi}^>, \mathsf{di}^>$ are valid INDU relations. It is routine to check that there are only 2053 triples of INDU relations that satisfy the above two constraints. We recall that IA has 409 c-triads and PA has 13 c-triads.

The composition of INDU calculus is provided below in Table 3.4. Note we use $\star$ to denote the disjunction of all relations, and use $x \in \{\mathsf{p}, \mathsf{pi}, \mathsf{m}, \mathsf{mi}, \mathsf{o}, \mathsf{oi}\}$ to denote the disjunction of $x^<$, $x^=$ and $x^>$. Other abbreviations are given in Table 3.5.

3.3.2 The Oriented Point Relation Algebra

In the $\mathcal{OPRA}_m$ calculus, each object is represented as an oriented point (o-point for short) in the plane. Each o-point has an orientation. Based on which, $2m-1$ other directions are introduced according to the chosen granularity. Any other o-point is located on either a ray or in a section between two consecutive rays. Each of these rays and sections is assigned an integer from 0 to $4m-1$. The relative directional information of two o-points A, B is uniquely encoded in a pair of integer numbers (s, t), where s is the ray or section of A in which B is located, and t is the ray or section of B in which A is located. Such a relation is also written as $A_m\angle_s^t B$.

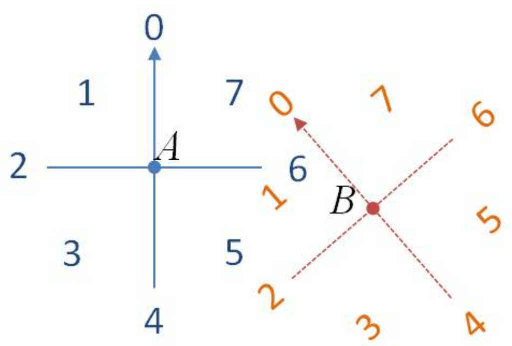


Figure 3.1: Two o-points A, B with the $\mathcal{OPRA}_2$ relation ${}_2\angle_5^1$

	$p^<$	$p^=$	$p^>$	$o^<$	$o^=$	$o^>$	$m^<$	$m^=$	$m^>$	s	d	f	fi	di	si	$mi^<$	$mi^=$	$mi^>$	$oi^<$	$oi^=$	$oi^>$	$pi^<$	$pi^=$	$pi^>$
$p^<$	$p^<$	$p^<$	p	$p^<$	$p^<$	p	$p^<$	$p^<$	p	$p^<$	$\gamma^<$	$\gamma^<$	p	p	p	$\gamma^<$	$\gamma^<$	γ	$\gamma^<$	$\gamma^<$	γ	$\star^<$	$\star^<$	$\star$
$p^=$	$p^<$	$p^=$	$p^>$	$p^<$	$p^=$	$p^>$	$p^<$	$p^=$	$p^>$	$p^<$	$\gamma^<$	$\gamma^<$	$p^>$	$p^>$	$p^>$	$\gamma^<$	$\phi_1^=$	$\phi_1^>$	$\gamma^<$	$\phi_1^=$	$\phi_1^>$	$\star^<$	$\star^=$	$\star^>$
$p^>$	p	$p^>$	$p^>$	p	$p^>$	$p^>$	p	$p^>$	$p^>$	p	γ	γ	$p^>$	$p^>$	$p^>$	γ	$\phi_1^>$	$\phi_1^>$	γ	$\phi_1^>$	$\phi_1^>$	$\star$	$\star^>$	$\star^>$
$o^<$	$p^<$	$p^<$	p	$\phi_1^<$	$\phi_1^<$	ϕ_1	$p^<$	$p^<$	p	$o^<$	$\phi_3^<$	$\phi_3^<$	ϕ_1	α	ϕ_6	$oi^<$	$oi^<$	ϕ_4	$\mu^<$	$\mu^<$	μ	$\phi_2^<$	$\phi_2^<$	δ
$o^=$	$p^<$	$p^=$	$p^>$	$\phi_1^<$	$\phi_1^=$	$\phi_1^>$	$p^<$	$p^=$	$p^>$	$o^<$	$\phi_3^<$	$\phi_3^<$	$\phi_1^>$	$\alpha^>$	$\phi_6^>$	$oi^<$	$oi^=$	$\phi_4^>$	$\mu^<$	$\mu^=$	$\mu^>$	$\phi_2^<$	$\phi_2^=$	$\delta^>$
$o^>$	p	$p^>$	$p^>$	ϕ_1	$\phi_1^>$	$\phi_1^>$	p	$p^>$	$p^>$	o	ϕ_3	ϕ_3	$\phi_1^>$	$\alpha^>$	$\phi_6^>$	ϕ_4	$\phi_4^>$	$\phi_4^>$	μ	$\mu^>$	$\mu^>$	δ	$\delta^>$	$\delta^>$
$m^<$	$p^<$	$p^<$	p	$p^<$	$p^<$	p	$p^<$	$p^<$	p	$m^<$	$\phi_3^<$	$\phi_3^<$	p	p	m	f	f	ϕ_8	$\phi_3^<$	$\phi_3^<$	ϕ_3	$\phi_2^<$	$\phi_2^<$	δ
$m^=$	$p^<$	$p^=$	$p^>$	$p^<$	$p^=$	$p^>$	$p^<$	$p^=$	$p^>$	$m^<$	$\phi_3^<$	$\phi_3^<$	$p^>$	$p^>$	$m^>$	f	eq	fi	$\phi_3^<$	$o^=$	$o^>$	$\phi_2^<$	$\phi_2^=$	$\delta^>$
$m^>$	p	$p^>$	$p^>$	p	$p^>$	$p^>$	p	$p^>$	$p^>$	m	ϕ_3	ϕ_3	$p^>$	$p^>$	$m^>$	ϕ_8	fi	fi	ϕ_3	$o^>$	$o^>$	δ	$\delta^>$	$\delta^>$
s	$p^<$	$p^<$	p	$\phi_1^<$	$\phi_1^<$	ϕ_1	$p^<$	$p^<$	p	s	d	d	ϕ_1	α	ϕ_7	$mi^<$	$mi^<$	mi	$\phi_5^<$	$\phi_5^<$	ϕ_5	$pi^<$	$pi^<$	pi
d	$p^<$	$p^<$	p	$\gamma^<$	$\gamma^<$	γ	$p^<$	$p^<$	p	d	d	d	γ	$\star$	β	$pi^<$	$pi^<$	pi	$\beta^<$	$\beta^<$	β	$pi^<$	$pi^<$	pi
f	$p^<$	$p^<$	p	$\phi_3^<$	$\phi_3^<$	ϕ_3	$m^<$	$m^<$	m	d	d	f	ϕ_8	δ	ϕ_2	$pi^<$	$pi^<$	pi	$\phi_2^<$	$\phi_2^<$	ϕ_2	$pi^<$	$pi^<$	pi
fi	p	$p^>$	$p^>$	o	$o^>$	$o^>$	m	$m^>$	$m^>$	o	ϕ_3	ϕ_8	fi	di	di	ϕ_4	$\phi_4^>$	$\phi_4^>$	ϕ_4	$\phi_4^>$	$\phi_4^>$	δ	$\delta^>$	$\delta^>$
di	α	$\alpha^>$	$\alpha^>$	ϕ_6	$\phi_6^>$	$\phi_6^>$	ϕ_6	$\phi_6^>$	$\phi_6^>$	ϕ_6	μ	ϕ_4	di	di	di	ϕ_4	$\phi_4^>$	$\phi_4^>$	ϕ_4	$\phi_4^>$	$\phi_4^>$	δ	$\delta^>$	$\delta^>$
si	α	$\alpha^>$	$\alpha^>$	ϕ_6	$\phi_6^>$	$\phi_6^>$	ϕ_6	$\phi_6^>$	$\phi_6^>$	ϕ_7	ϕ_5	oi	di	di	si	mi	$mi^>$	$mi^>$	oi	$oi^>$	$oi^>$	pi	$pi^>$	$pi^>$
$mi^<$	$\phi_1^<$	$\phi_1^<$	α	$\phi_5^<$	$\phi_5^<$	ϕ_5	s	s	ϕ_7	$\phi_5^<$	$\phi_5^<$	$mi^<$	mi	pi	pi	$pi^<$	$pi^<$	pi	$pi^<$	$pi^<$	pi	$pi^<$	$pi^<$	pi
$mi^=$	$\phi_1^<$	$\phi_1^=$	$\alpha^>$	$\phi_5^<$	$oi^=$	$oi^>$	s	eq	si	$\phi_5^<$	$\phi_5^<$	$mi^<$	$mi^>$	$pi^>$	$pi^>$	$pi^<$	$pi^=$	$pi^>$	$pi^<$	$pi^=$	$pi^>$	$pi^<$	$pi^=$	$pi^>$
$mi^>$	α	$\alpha^>$	$\alpha^>$	ϕ_5	$oi^>$	$oi^>$	ϕ_7	si	si	ϕ_5	ϕ_5	mi	$mi^>$	$pi^>$	$pi^>$	pi	$pi^>$	$pi^>$	pi	$pi^>$	$pi^>$	pi	$pi^>$	$pi^>$
$oi^<$	$\phi_1^<$	$\phi_1^<$	α	$\mu^<$	$\mu^<$	μ	$o^<$	$o^<$	ϕ_6	$\phi_5^<$	$\phi_5^<$	$oi^<$	ϕ_4	δ	ϕ_2	$pi^<$	$pi^<$	pi	$\phi_2^<$	$\phi_2^<$	ϕ_2	$pi^<$	$pi^<$	pi
$oi^=$	$\phi_1^<$	$\phi_1^=$	$\alpha^>$	$\mu^<$	$\mu^=$	$\mu^>$	$o^<$	$o^=$	$\phi_6^>$	$\phi_5^<$	$\phi_5^<$	$oi^<$	$\phi_4^>$	$\delta^>$	$\phi_2^>$	$pi^<$	$pi^=$	$pi^>$	$\phi_2^<$	$\phi_2^=$	$\phi_2^>$	$pi^<$	$pi^=$	$pi^>$
$oi^>$	α	$\alpha^>$	$\alpha^>$	μ	$\mu^>$	$\mu^>$	ϕ_6	$\phi_6^>$	$\phi_6^>$	ϕ_5	ϕ_5	oi	$\phi_4^>$	$\delta^>$	$\phi_2^>$	pi	$pi^>$	$pi^>$	ϕ_2	$\phi_2^>$	$\phi_2^>$	pi	$pi^>$	$pi^>$
$pi^<$	$\star^<$	$\star^<$	$\star$	$\beta^<$	$\beta^<$	β	$\beta^<$	$\beta^<$	β	$\beta^<$	$\beta^<$	$pi^<$	pi	pi	pi	$pi^<$	$pi^<$	pi	$pi^<$	$pi^<$	pi	$pi^<$	$pi^<$	pi
$pi^=$	$\star^<$	$\star^=$	$\star^>$	$\beta^<$	$\phi_2^=$	$\phi_2^>$	$\beta^<$	$\phi_2^=$	$\phi_2^>$	$\beta^<$	$\beta^<$	$pi^<$	$pi^>$	$pi^>$	$pi^>$	$pi^<$	$pi^=$	$pi^>$	$pi^<$	$pi^=$	$pi^>$	$pi^<$	$pi^=$	$pi^>$
$pi^>$	$\star$	$\star^>$	$\star^>$	β	$\phi_2^>$	$\phi_2^>$	β	$\phi_2^>$	$\phi_2^>$	β	β	pi	$pi^>$	$pi^>$	$pi^>$	pi	$pi^>$	$pi^>$	pi	$pi^>$	$pi^>$	pi	$pi^>$	$pi^>$

Table 3.4: The composition table of the INDU calculus, where cell z in row x and column y stands for $x \circ y = z$.

abbreviations	definition
$\star$	p, m, o, s, d, f, eq, fi, di, si, oi, mi, pi
$\star^<$	$p^<$, $m^<$, $o^<$, s, d, f, $oi^<$, $mi^<$, $pi^<$
$\star^>$	$p^>$, $m^>$, $o^>$, fi, di, si, $oi^>$, $mi^>$, $pi^>$
$\star^=$	$p^=$, $m^=$, $o^=$, eq, $oi^=$, $mi^=$, $pi^=$
μ	o, s, d, f, eq, fi, di, si, oi
$\mu^<$	$o^<$, s, d, f, $oi^<$
$\mu^>$	$o^>$, fi, di, si, $oi^>$
$\mu^=$	$o^=$, eq, $oi^=$
α	p, m, o, fi, di
$\alpha^>$	$p^>$, $m^>$, $o^>$, fi, di
β	d, f, oi, mi, pi
$\beta^<$	d, f, $oi^<$, $mi^<$, $pi^<$
γ	p, m, o, s, d
$\gamma^<$	$p^<$, $m^<$, $o^<$, s, d
δ	di, si, oi, mi, pi
$\delta^>$	di, si, $oi^>$, $mi^>$, $pi^>$
ϕ_1	p, m, o
$\phi_1^<$	$p^<$, $m^<$, $o^<$
$\phi_1^>$	$p^>$, $m^>$, $o^>$
$\phi_1^=$	$p^=$, $m^=$, $o^=$
ϕ_2	oi, mi, pi
$\phi_2^<$	$oi^<$, $mi^<$, $pi^<$
$\phi_2^>$	$oi^>$, $mi^>$, $pi^>$
$\phi_2^=$	$oi^=$, $mi^=$, $pi^=$
ϕ_3	o, s, d
$\phi_3^<$	$o^<$, s, d
ϕ_4	di, si, oi
$\phi_4^>$	di, si, $oi^>$
ϕ_5	d, f, oi
$\phi_5^<$	d, f, $oi^<$
ϕ_6	o, fi, di
$\phi_6^>$	$o^>$, fi, di
ϕ_7	s, eq, si
ϕ_8	f, eq, fi

Table 3.5: The abbreviations used in Table 3.4.

There are two natural ways to represent o-points. One uses the Cartesian coordinate system, the other uses the polar coordinate system. We next show how the choice of coordinate system will significantly affect the experimental results, which are compared with that of [75].

In the Cartesian coordinate system, an o-point P is represented by its coordination (x, y) and its orientation ϕ.

Definition 3.4 *Let M_1 and M_2 be two positive integers. We define a Cartesian based subdomain of $\mathcal{OPRA}_m$ as*

$$D_c(M_1, M_2) = \{((x, y), \phi) : x, y \in [-M_1, M_1] \cap \mathbb{Z}, \phi \in \Phi_{M_2}\}, \tag{3.5}$$

where $\Phi_{M_2} := \{0, 2\pi/M_2, \cdots, (M_2 - 1)/M_2 \times 2\pi\}$.

M_2	2	3	4	5	6	8	10	12	16
TRIAD	148	1024	1056	1024	1024	$\boxed{1440}$	1024	1408	$\boxed{1440}$

M_1	2	4	6	8	10
LASTFOUND ($M_2 = 8$)	8082	35932	411893	881787	> 1000000
LASTFOUND ($M_2 = 16$)	18618	295936	174490	> 1000000	> 1000000

Table 3.6: Implementation for $\mathcal{OPRA}_1$ on a Cartesian coordinated domain $D_c(M_1, M_2)$, where TRIAD is the number of c-triads computed by running the algorithm on $D_c(M_1, M_2)$ for $M_1 = 6$; LASTFOUND is the loop when the last triad is first recorded for $M_2 = 8$ (in the 2nd last row) and $M_2 = 16$ (in the last row)

Our experimental results show that, for $\mathcal{OPRA}_1$, the algorithm converges and generates the correct CT for subdomains with $M_1 \geq 2$ and $M_2 \in \{8, 16\}$. That is, the smallest 3-complete subdomain is $D_c(2, 8)$.

For $\mathcal{OPRA}_2$, however, the algorithm does not compute the desired CT in ten million loops. In fact, it is impossible to compute the desired CT if we use Cartesian coordination. Consider the following example. Suppose A, B, C are three o-points, such that $\triangle ABC$ is an acute triangle, and the orientation of A is the same as the direction from A to B, the orientations of B and C are similar. In this configuration, we have $A\,{}_2\!\angle^1_0 B$, $B\,{}_2\!\angle^1_0 C$, and $A\,{}_2\!\angle^0_1 C$. This configuration, however, cannot be realised in a Cartesian based subdomain.

M_2	2	4	6	8	10	12	16
TRIAD	2704	2704	21792	23616	21792	21792	35232

Table 3.7: Implementation for $\mathcal{OPRA}_2$ on a Cartesian coordinated domain $D_c(M_1, M_2)$, where TRIAD is the number of c-triads computed by running Algorithm 1 ten million loops on $D_c(M_1, M_2)$ for $M_1 = 6$

Based on the above observation, we turn to the polar coordinated representation. In the polar coordinate system, an o-point P is represented by its polar coordination (ρ, θ) and its orientation ϕ.

Definition 3.5 *Let M_1 and M_2 be two positive integers. We define a polar coordinated subdomain of $\mathcal{OPRA}_m$ as*

$$D_p(M_1, M_2) = \{((\rho, \theta), \phi) : \rho \in [0, M_1] \cap \mathbb{Z}, \theta, \phi \in \Phi_{M_2}\}, \tag{3.6}$$

where $\Phi_{M_2} := \{0, 2\pi/M_2, \cdots, (M_2 - 1)/M_2 \times 2\pi\}$.

As in Cartesian based subdomains, the parameter M_2 determines if a domain is complete, while M_1 determines the efficiency of the algorithm. For $\mathcal{OPRA}_1$, we have $D(M_1, M_2)$ is a 3-complete subdomain if $M_1 \geq 2$ and $M_2 = 6, 8, 10, 12, 16$ (see Table 3.8); for $\mathcal{OPRA}_2$, we have $D(M_1, M_2)$ is 3-complete if $M_1 \geq 4$ and $M_2 = 6, 10, 12, 16$ (see Table 3.9).

M_2	2	3	4	5	6	8	10	12	16
TRIAD	52	1024	1032	1408	$\boxed{1440}$	$\boxed{1440}$	$\boxed{1440}$	$\boxed{1440}$	$\boxed{1440}$

M_1	4	6	8	10	16
LASTFOUND ($M_2 = 8$)	3072	4868	22327	10363	38843
LASTFOUND ($M_2 = 16$)	26219	45831	121542	71205	146536

Table 3.8: Implementation for $\mathcal{OPRA}_1$ on a polar coordinated domain $D_p(M_1, M_2)$, where TRIAD is the number of c-triads computed by running the algorithm on $D_p(M_1, M_2)$ for $M_1 = 6$; LASTFOUND is the loop when the last triad is first recorded for $M_2 = 8$ (in the 2nd last row) and $M_2 = 16$ (in the last row)

M_2	2	3	4	6	8	10	12	16
Triad	400	24672	2128	$\boxed{36256}$	23616	$\boxed{36256}$	$\boxed{36256}$	$\boxed{36256}$

Table 3.9: Implementation for $\mathcal{OPRA}_2$ on a polar coordinated domain $D_p(M_1, M_2)$, where Triad is the number of c-triads computed by running the algorithm on $D_p(M_1, M_2)$ for $M_1 = 6$

3.4 Chapter Summary

In this chapter, we introduced a novel and simple semi-automatic method for computing the composition tables of qualitative calculi. We implemented the basic algorithm for several well-known qualitative calculi, including the Interval Algebra, INDU, and $\mathcal{OPRA}_m$ for $m = 1, 2$. In particular, we established for the first time, as far as we know, the correct CT for INDU, and verified the validity of the algorithm reported for the $\mathcal{OPRA}$ calculi [75]. Our method can be easily integrated into existing qualitative solvers e.g. SparQ [108] or GQR [110]. This provides a partial answer to the challenge proposed in [13].

Our method relies on the assumption that the qualitative calculus has a small 'discretised' 3-complete subdomain. The concept of the 3-complete subdomain can be generlised to k-complete as follows. A subset of the universe is called a *k-complete subdomain* if any consistent constraint network that involves no more than k variables has a solution in it. The k-complete subdomain always exists. This is because there are only finitely many constraint networks with no more than k variables (denoted by Σ_k). For each consistent network in Σ_k, we select a representative solution. Then the union of all the representative solutions is clearly a k-complete subdomain. More importantly, if a k-complete subdomain is somehow provided in advance, the consistency problem may be reduced to the classical CSP such that well-developed CSP solvers can be made use of. For Interval Algebra, the integer numbers in $[1, 2k]$ are a k-complete subdomain. For other qualitative calculi, how to find an explicit k-complete subdomain remains open. Future work might also address the application of our method for reasoning with a customised composition table.

Chapter 4

Introducing Fuzziness to RCC

4.1 Introduction

Temporal and spatial knowledge often involves vagueness. The vagueness can be roughly classified into two types: uncertainty and imprecision. Uncertainty usually comes from the lack of knowledge or the ambiguity of natural languages. For instance, the location of *the MacDonald's in downtown* may be uncertain since there may be multiple MacDonald's; *Sydney* may refer to the city in Australia, or the city in Canada. Imprecision may occur when metric quantities are approximated by numbers with limited precisions. For one hand, the quantities in real applications are usually not finitely representable (such as the area of Australia), so it is inevitable to use approximations instead. For the other hand, when very accurate data are available, people may still use rough approximations when the approximations are sufficient for the tasks or in the contexts – this is even more important in computer science as dealing with more precise data needs almost all kinds of computing resources (e.g., storage, communication channels and CPU) as a trade-off. Imprecision is also rooted in the semantics of natural languages. The meaning of a word (or words) may be qualitative and variable in different contexts. For example, the volumes of *a large rock* and *a large diamond* are likely to be quite different; 'Sydney' may only refer to the downtown area, or it may include suburbs as well. These two kinds of vagueness may get tangled. The geographical location of *Sydney* is an example as shown above. The meaning of *spring* is another example: it varies in different hemispheres and there is no hard boundary between *spring* and *winter* (or *summer*).

The vagueness has hampered the applications of qualitative calculi (such as RCC-8 and IA) in real-world scenarios, such as extracting and integrating spatial information from the numerous texts on the Internet [96]. Therefore, the QSR community has been increasing interested in how to revise or generalise qualitative calculi, to cope with the vagueness of spatial information [11, 10, 90, 24, 48, 115, 58, 7]. While Chapter 6 of this thesis can be viewed as an approach for handling uncertainty, this chapter focuses on the issue of imprecision.

Due to its expressive power and flexibility, the Region Connection Calculus (RCC) proposed by Randell, Cui, and Cohn [84], has attracted special attention. One remarkable approach first proposed by Cohn and Gotts [11], known as the *Egg-Yolk model*, is to consider regions as having 'thick' boundaries. Meanwhile, as RCC is established on first-order logic, fuzzy set theory has naturally become another candidate for generalising the RCC, see e.g. [115, 58, 7][1]. Providing an effective and universal approach to deal with vagueness, fuzzy set theory no longer crisply asserts whether or not an element is in a set. Instead, it makes use of a quantity in $I = [0, 1]$ to denote the *degree* to which the element is considered as belonging to the set.

Although regions can be fuzzified in a natural way, there are a number of different methods for extending crisp relations to fuzzy ones [25, 52, 115, 58] (see Section 2 for more detail). Recently, Schockaert et al. [95] proposed a novel fuzzification of the RCC theory. Analogous to the crisp RCC, a reflexive, symmetric but fuzzy relation $\mathbf{C}$ on universe $\mathbf{U}$ is assumed. They regard $(\mathbf{U}, \mathbf{C})$ as a fuzzy connection structure, and further define other well-known RCC-8 relations in terms of $\mathbf{C}$, by generalizing logical conjunction and implication to fuzzy logic connectives, and universal and existential quantification to the infima and suprema of truth degrees. This fuzzy RCC turns out to have several attractive properties. First, it makes no assumptions on how regions should be represented and thus keeps the maximum generality of the original RCC. Second, the weak-composition based reasoning technique developed for RCC-8 algebra [86, 26, 57] is extended to determine the consistency of fuzzy RCC constraints in NP time complexity. In fact, the consistency of a normalised fuzzy RCC-8 network can be determined by checking the validity of a set of basic conditions and transitivity rules [96].

[1] Interested readers may consult [95] for more information.

The connection between the fuzzy RCC models and crisp RCC models is not completely clear. Schockaert et al. [94] define special fuzzy connectedness in terms of the *closeness* between fuzzy sets, and prove in [96] that any consistent set of fuzzy RCC constraints has a solution in these models. On the other hand, as admitted in [96], this closeness-based interpretation is counterintuitive in some situations. For example, two externally connected closed disks will be interpreted as overlapping in the closeness-based interpretation.

The main contribution of this chapter is to present a close connection between crisp RCC models and fuzzy RCC models. Precisely speaking, it will be shown that any crisp RCC model can be naturally generalised to a fuzzy RCC model, which is called the *standard fuzzy RCC model*. We then show that each standard fuzzy RCC model (in particular the one defined on $\mathbb{R}^n$) is *canonical* in the sense that any consistent set of fuzzy RCC constraints has a solution in it. To this end, we devise a polynomial realisation algorithm for consistent and normalised fuzzy RCC-8 networks. The algorithm is similar to the cubic realisation algorithm devised for classical RCC-8 constraints [55], described in Section 2.5. Moreover, the construction algorithm enables us to prove that two *similar* sets of fuzzy constraints have *similar* solutions if both are satisfiable, where fuzzy sets A, B are similar if $\sup\{|\mathbf{r}(x) - \mathbf{s}(x)| : x \in X\}$ is smaller than a predefined number.

The rest of this chapter proceeds as follows. Section 2 provides fundamental knowledge of fuzzy set theory and defines fuzzy regions. Section 3 presents standard fuzzy RCC models. Section 4 discusses the consistency problem of the fuzzy RCC constraints, and claims that normalised fuzzy networks are tractable by identifying a set of necessary and sufficient conditions. To show the sufficiency of the conditions, Section 5 presents a polynomial algorithm for solution construction, the soundness of which is proved in Section 6. Section 7 shows that a small perturbation will not greatly change the solution of a consistent fuzzy RCC-8 network and, thus, justifies the rationality of approximating fuzzy constraints by constraints whose bounds are taken from a finite domain Σ_k. Note that the results in this chapter have been published in [64].

4.2 Fuzzy Set Theory and Fuzzy Regions

First we briefly introduce the basics of fuzzy set theory [114]. Let X be the universe of discourse. A *fuzzy set* $\mathbf{r}$ of X is defined as a function from X to $[0,1]$. For $x \in X$, the value $\mathbf{r}(x)$ reflects to what extent x belongs to $\mathbf{r}$. We use $\mathsf{ran}(\mathbf{r})$ to denote the range of $\mathbf{r}$. For $\alpha \in [0,1]$, the α-cut of $\mathbf{r}$ is defined as $\mathbf{r}_\alpha := \{x \in X : \mathbf{r}(x) \geq \alpha\}$. We call $\mathbf{r}_1$ the *core* of $\mathbf{r}$ and call $\{x \in X : \mathbf{r}(x) > 0\}$ the *support* of $\mathbf{r}$. A fuzzy set is called *normalised* if its core is nonempty. For a crisp set $r \subseteq X$, we write $\chi_r : X \to \{0,1\}$ for the *characteristic function* of r, i.e., $\chi_r(x)$ is 1 if $x \in r$ and 0 otherwise. Note that the characteristic function of r is a fuzzy set which can be regarded as a representation of the crisp set r.

For two fuzzy sets $\mathbf{r}$ and $\mathbf{s}$, we say $\mathbf{r}$ is a subset of $\mathbf{s}$, denoted by $\mathbf{r} \subseteq \mathbf{s}$, iff $\mathbf{r}(x) \leq \mathbf{s}(x)$ for any $x \in X$. Their (standard) intersection $\mathbf{r} \wedge \mathbf{s}$ and union $\mathbf{r} \vee \mathbf{s}$ are fuzzy sets defined by the following equations:

$$(\mathbf{r} \wedge \mathbf{s})(x) := \min(\mathbf{r}(x), \mathbf{s}(x)), \tag{4.1}$$

$$(\mathbf{r} \vee \mathbf{s})(x) := \max(\mathbf{r}(x), \mathbf{s}(x)). \tag{4.2}$$

Notation For $\kappa \in [0,1]$, we write $\chi_r \wedge \kappa$ for the fuzzy set $\mathbf{r}$ such that $\mathbf{r}(x)$ is κ if $x \in r$ and 0 otherwise.

For a set X, a *fuzzy relation* on X is a function from $X \times X$ into $[0,1]$. For a fuzzy relation $\mathbf{R}$ and $x, y \in X$, $\mathbf{R}(x,y)$ usually denotes to what degree x is related to y by $\mathbf{R}$.

Notation We use α,β to denote a value in $I = [0,1]$, and write r,s,t for crisp subsets of X, $\mathbf{r},\mathbf{s},\mathbf{t}$ for fuzzy subsets of X; and use R,S,T to denote crisp relations, $\mathbf{R},\mathbf{S},\mathbf{T}$ to denote fuzzy relations.

Propositional formulas can also be generalised to fuzzy ones. The well known Łukasiewicz t-norm T_W and its corresponding residual implicator I_W, which are the generalisations of the standard logical *conjunction* and *implication*, are defined as

$$T_W(a,b) = \max\{0, a+b-1\}, \tag{4.3}$$

$$I_W(a,b) = \min\{1, 1-a+b\}. \tag{4.4}$$

Now we are able to formalise the concept of *fuzzy regions*. Recall that each RCC model is defined on a Boolean Algebra. Suppose X is a topological space, and Boolean Algebra A is the regular closed algebra of X. It can be proved that if X is a connected regular T_1 space [1], then $(\mathbf{U}, \mathsf{C})$ is an RCC model, where $\mathbf{U} = A \setminus \{\varnothing\}$ and binary relation C on $\mathbf{U}$ is defined as $\mathsf{C}(a,b)$ iff $a \cap b \neq \varnothing$. This is known as the standard (crisp) RCC model associated to topological space X. Examples include all n-dimensional Euclidean spaces. To the converse, it is proved that every RCC model can be embedded in such a standard model defined by some connected T_1 space X which is weakly regular [27]. Henceforth, we always assume that a connection structure (or an RCC model) is defined on some underlying topological space X that is a connected regular T_1 space, e.g. the 2-dimensional Euclidean Space $\mathbb{R}^2$.

Definition 4.1 (fuzzy regions) *Let $(\boldsymbol{U}, \mathsf{C})$ be an RCC model with underlying space is X. A fuzzy subset $\boldsymbol{r}$ of X is called a* fuzzy region *if its support and all α-cuts ($\alpha > 0$) of $\boldsymbol{r}$ are regions in $\boldsymbol{U}$. We call fuzzy region $\boldsymbol{r}$ a Σ-region, if $\mathsf{ran}(\boldsymbol{r})$ is contained in set $\Sigma \subseteq [0,1]$.*

Suppose $\mathbf{r}$ is a fuzzy set such that $\mathsf{ran}(\mathbf{r})$ is contained in a finite set $\Sigma \subseteq [0,1]$. It is direct to show that $\mathbf{r}$ is a fuzzy region if and only if all α-cuts ($\alpha \in \Sigma \setminus \{0\}$) of $\mathbf{r}$ are regions in $\mathbf{U}$, as the union of finite regions is a still a region. It is obvious that for a Σ-region $\mathbf{r}$, we have $\mathbf{r} = \bigvee_{\alpha\in\Sigma}(\mathbf{r}_\alpha \wedge \alpha)$.

Notation We write $\mathbf{U}_I$ for the class of all fuzzy regions over $\mathbf{U}$ and write $\mathbf{U}_\Sigma$ for the class of Σ-regions.

4.3 Standard Fuzzy RCC Models

Given a crisp RCC model $(\mathbf{U}, \mathsf{C})$, we have generalised the crisp regions in $\mathbf{U}$ to fuzzy regions in $\mathbf{U}_I$. The next problem is to extend the crisp RCC relations to fuzzy ones. A number of formalisms have been introduced to fuzzify crisp relations. Suppose R is a binary crisp relation on U, and $\mathbf{r}, \mathbf{s}$ are two fuzzy sets on U, which take values in $V = \{1 = \alpha_1 > \alpha_2 > \cdots > \alpha_k > \alpha_{k+1} = 0\}$. Then the fuzzy relation between $\mathbf{r}$ and $\mathbf{s}$ can be

[1] Suppose $\langle X, \tau\rangle$ is a topological space. X is called a T_1 *space*, if for any distinct points $x, y \in X$, there exists an open set $u \in \tau$ such that $x \in u$ and $y \notin u$. A topological space X is called *regular*, if for any point $x \in X$ and any closed set u, there exist open sets $v_1, v_2 \in \tau$ such that $x \in v_1, u \subset v_2, v_1 \cap v_2 = \varnothing$.

Rel.	RCC definition	Fuzzy RCC definition
DC	$\neg\mathsf{C}(r,s)$	$1-\mathbf{C}(\mathbf{r},\mathbf{s})$
P	$(\forall t\in\mathbf{U})(\mathsf{C}(t,r)\rightarrow\mathsf{C}(t,s))$	$\inf_{\mathbf{t}\in\mathbf{U}_I}\{I_W(\mathbf{C}(\mathbf{t},\mathbf{r}),\mathbf{C}(\mathbf{t},\mathbf{s}))\}$
O	$(\exists t\in\mathbf{U})(\mathsf{P}(t,r)\wedge\mathsf{P}(t,s))$	$\sup_{\mathbf{t}\in\mathbf{U}_I}\{T_W(\mathbf{P}(\mathbf{t},\mathbf{r}),\mathbf{P}(\mathbf{t},\mathbf{s}))\}$
PO	$\mathsf{O}(r,s)\wedge\neg\mathsf{P}(r,s)\wedge\neg\mathsf{P}(s,r)$	$\min(\mathbf{O}(\mathbf{r},\mathbf{s}),1-\mathbf{P}(\mathbf{r},\mathbf{s}),1-\mathbf{P}(\mathbf{s},\mathbf{r}))$
EC	$\mathsf{C}(r,s)\wedge\neg\mathsf{O}(r,s)$	$\min(\mathbf{C}(\mathbf{r},\mathbf{s}),1-\mathbf{O}(\mathbf{r},\mathbf{s}))$
EQ	$\mathsf{P}(r,s)\wedge\mathsf{P}(s,r)$	$\min(\mathbf{P}(\mathbf{r},\mathbf{s}),\mathbf{P}(\mathbf{s},\mathbf{r}))$
PP	$\mathsf{P}(r,s)\wedge\neg\mathsf{P}(s,r)$	$\min(\mathbf{P}(\mathbf{r},\mathbf{s}),1-\mathbf{P}(\mathbf{s},\mathbf{r}))$
NTP	$(\forall t\in\mathbf{U})(\mathsf{C}(t,r)\rightarrow\mathsf{O}(t,s))$	$\inf_{\mathbf{t}\in\mathbf{U}_I}\{I_W(\mathbf{C}(\mathbf{t},\mathbf{r}),\mathbf{O}(\mathbf{t},\mathbf{s}))\}$
TPP	$\mathsf{PP}(r,s)\wedge\neg\mathsf{NTP}(r,s)$	$\min(\mathbf{PP}(\mathbf{r},\mathbf{s}),1-\mathbf{NTP}(\mathbf{r},\mathbf{s}))$
NTPP	$\mathsf{PP}(r,s)\wedge\mathsf{NTP}(r,s)$	$\min(1-\mathbf{P}(\mathbf{s},\mathbf{r}),\mathbf{NTP}(\mathbf{r},\mathbf{s}))$

Table 4.1: Definitions of relations in the original RCC and the fuzzy RCC [96]

defined by the following equations [25, 52]

$$F_R(\mathbf{r},\mathbf{s}) = \Sigma_{i,j=1}^{k}(\alpha_i-\alpha_{i+1})(\alpha_j-\alpha_{j+1})R(\mathbf{r}_i,\mathbf{s}_j), \tag{4.5}$$

$$F_R^*(\mathbf{r},\mathbf{s}) = \Sigma_{i=1}^{k}(\alpha_i-\alpha_{i+1})R(\mathbf{r}_i,\mathbf{s}_i), \tag{4.6}$$

where $\mathbf{r}_i=\{x\in U:\mathbf{r}(x)\geq\alpha_i\}$ is the α_i-cut of $\mathbf{r}$ and $\mathbf{s}_j=\{x\in U:\mathbf{s}(x)\geq\alpha_j\}$ is the α_j-cut of $\mathbf{s}$. These general methods have been applied to extend topological relations by Zhan [115] and Li & Li [58]. In particular, [58] makes complete classifications of the topological relations between fuzzy regions using RCC-5 and RCC-8. However, generalising all the RCC relations in the above way jeopardises the simplicity of the reasoning rules of RCC (such as weak composition and consistency).

Recall that (crisp) RCC is a first-order theory based on a primitive connectedness relation C. That is to say, all other RCC relations can be defined by C and first-order formulas. Therefore, suppose we have defined $\mathbf{C}$ as the fuzzy connectedness relation which extends the crisp connectedness relation C in some manner, it is natural to require that all other fuzzy RCC relations can be derived from $\mathbf{C}$ in a consistent way. In fact, Schockaert et al. [95] have proposed such a method, as shown in Table 4.1, and justified its rationality. Note that the T_W and I_W in the table are respectively the Łukasiewicz t-norm and its corresponding residual implicator. For clarity, we use T and I to denote T_W and I_W from now on.

Note that Schockaert et al. [94] have also defined a special fuzzy connectedness relation in terms of the *closeness* between fuzzy regions on the 2-dim Euclidean space.

However, such a definition cannot be applied to an arbitrary RCC model, and a general method of fuzzifying connectedness relation $\mathbf{C}$ is still missing.

For two fuzzy regions $\mathbf{r}, \mathbf{s}$ with $\mathsf{C}(\mathbf{r}_\alpha, \mathbf{s}_\beta)$ for some α and β, it is reasonable to require that $\mathbf{C}(\mathbf{r}, \mathbf{s}) \geq T(\alpha, \beta)$. Note that this implies in particular that $\mathbf{C}(\mathbf{r}, \mathbf{s}) = 1$ if $\mathsf{C}(\mathbf{r}_1, \mathbf{s}_1)$, which coincides with the crisp case. Based on this assumption, we propose our novel formalisation of $\mathbf{C}$, by amalgamating all connectedness evidences from α-cuts of $\mathbf{r}$ and β-cuts of $\mathbf{s}$ in $(\mathbf{U}, \mathsf{C})$.

Definition 4.2 *Let (U, C) be an RCC model with underlying space X. We call $(\boldsymbol{U}_I, \mathbf{C})$ the* standard fuzzy RCC model on *X, where $\mathbf{C}$ is the fuzzy relation on $\boldsymbol{U}_I$ defined as:*

$$\mathbf{C}(\boldsymbol{r}, \boldsymbol{s}) = \sup\{T(\alpha, \beta) : C(\boldsymbol{r}_\alpha, \boldsymbol{s}_\beta)\}, \quad (\boldsymbol{r}, \boldsymbol{s} \in \boldsymbol{U}_I). \tag{4.7}$$

Write $\mathbf{P}, \mathbf{O}$, respectively, for the fuzzy extension of RCC-8 relations P, O, and write $\mathbf{NTP}$, or $\mathbf{N}$ for short, for the fuzzy extension of NTP (cf. Table 4.1). Note that besides the primitive fuzzy relation $\mathbf{C}$, these fuzzy relations are the only three which have quantifiers in their definitions. Furthermore, all the other fuzzy RCC-8 relations can be calculated from $\mathbf{C}, \mathbf{O}, \mathbf{P}, \mathbf{N}$ directly according to Table 4.1. Therefore, these four relations are of particular importance. The fuzzy relation $\mathbf{C}$ between two fuzzy regions $\mathbf{r}, \mathbf{s}$ is defined by amalgamating all connectedness evidence from α-cuts of $\mathbf{r}$ and β-cuts of $\mathbf{s}$. However, the fuzzy relations $\mathbf{O}$, $\mathbf{P}$ and $\mathbf{N}$ are defined in Table 4.1 as the infimum or supremum which involves a fuzzy region running all over the domain and thus cannot be directly computed. The following theorem answers this problem and allows us to compute $\mathbf{O}$, $\mathbf{P}$ and $\mathbf{N}$ in a direct way.

Theorem 4.1 *For fuzzy regions $\boldsymbol{r}, \boldsymbol{s}$ in a standard fuzzy RCC model $(\boldsymbol{U}_I, \mathbf{C})$, we have*

$$\mathbf{O}(\boldsymbol{r}, \boldsymbol{s}) = \sup\{T(\alpha, \beta) : O(\boldsymbol{r}_\alpha, \boldsymbol{s}_\beta)\}, \tag{4.8}$$

$$\mathbf{P}(\boldsymbol{r}, \boldsymbol{s}) = \sup\{\alpha : (\forall \beta) P(\boldsymbol{r}_\beta, \boldsymbol{s}_{T(\alpha, \beta)})\}, \tag{4.9}$$

$$\mathbf{N}(\boldsymbol{r}, \boldsymbol{s}) = \sup\{\alpha : (\forall \beta) NTPP(\boldsymbol{r}_\beta, \boldsymbol{s}_{T(\alpha, \beta)})\}. \tag{4.10}$$

Proof We first show Eq. 4.9 holds, and then use Eq. 4.9 to prove Eq. 4.8. Let $\alpha^* = \sup\{\alpha : (\forall \beta) \mathsf{P}(\mathbf{r}_\beta, \mathbf{s}_{T(\alpha, \beta)})\}$. We show that

$$\mathbf{P}(\mathbf{r}, \mathbf{s}) = \inf_{\mathbf{t} \in \mathbf{U}_I} \{I(\mathbf{C}(\mathbf{t}, \mathbf{r}), \mathbf{C}(\mathbf{t}, \mathbf{s}))\} \geq \alpha^*.$$

For any $\alpha < \alpha^*$, we assert $\alpha \le I(\mathbf{C}(\mathbf{t},\mathbf{r}),\mathbf{C}(\mathbf{t},\mathbf{s}))$ holds for any fuzzy region $\mathbf{t} \in \mathbf{U}_I$. In fact, as $\alpha < 1$, we only need to prove

$$\mathbf{C}(\mathbf{t},\mathbf{r}) + \alpha \le 1 + \mathbf{C}(\mathbf{t},\mathbf{s}).$$

Recall $\mathbf{C}(\mathbf{t},\mathbf{r}) = \sup\{T(\gamma,\delta) : \mathsf{C}(\mathbf{t}_\gamma,\mathbf{r}_\delta)\}$ by definition. For any γ and δ such that $\mathsf{C}(\mathbf{t}_\gamma,\mathbf{r}_\delta)$ holds, if $\alpha+\delta \le 1$, then $T(\gamma,\delta)+\alpha \le 1+\mathbf{C}(\mathbf{t},\mathbf{s})$ holds. Suppose $\alpha+\delta > 1$. By $\mathsf{P}(\mathbf{r}_\delta,\mathbf{s}_{\delta+\alpha-1})$, we have $\mathsf{C}(\mathbf{t}_\gamma,\mathbf{s}_{\delta+\alpha-1})$. So $T(\gamma,\delta+\alpha-1) \le \mathbf{C}(\mathbf{t},\mathbf{s})$, i.e., $T(\gamma,\delta)+\alpha \le 1+\mathbf{C}(\mathbf{t},\mathbf{s})$. Therefore $\mathbf{C}(\mathbf{t},\mathbf{r})+\alpha \le 1+\mathbf{C}(\mathbf{t},\mathbf{s})$.

It remains to prove

$$\mathbf{P}(\mathbf{r},\mathbf{s}) = \inf_{\mathbf{t}\in\mathbf{U}_I}\{I(\mathbf{C}(\mathbf{t},\mathbf{r}),\mathbf{C}(\mathbf{t},\mathbf{s}))\} \le \alpha^*.$$

For any $\epsilon > 0$, we prove there exists a fuzzy region $\mathbf{t}$ such that $I(\mathbf{C}(\mathbf{t},\mathbf{r}),\mathbf{C}(\mathbf{t},\mathbf{s})) \le \alpha^* + \epsilon$. By the definition of α^*, we know that there exists some γ such that $\neg\mathsf{P}(\mathbf{r}_\gamma,\mathbf{s}_{\gamma+\alpha^*+\epsilon-1})$, i.e., there exists a crisp region t such that

$$\mathsf{C}(t,\mathbf{r}_\gamma) \text{ and } \neg\mathsf{C}(t,\mathbf{s}_{\gamma+\alpha^*+\epsilon-1}).$$

Let $\mathbf{t} = \chi_t$ be the characteristic function of t. Then $\mathbf{C}(\mathbf{t},\mathbf{r}) \ge \gamma$ and $\mathbf{C}(\mathbf{t},\mathbf{s}) \le \gamma+\alpha^*+\epsilon-1$ hold. Therefore $\mathbf{C}(\mathbf{t},\mathbf{s}) \le \mathbf{C}(\mathbf{t},\mathbf{r})+\alpha^*+\epsilon-1$. So $I(\mathbf{C}(\mathbf{t},\mathbf{r}),\mathbf{C}(\mathbf{t},\mathbf{s})) \le \alpha^*+\epsilon$. Due to the arbitrariness of ϵ, we know $\mathbf{P}(\mathbf{r},\mathbf{s}) \le \alpha^*$.

The proof of Eq. 4.10 is similar to that of Eq. 4.9.

We next prove Eq. 4.8. To this end, we first show

$$\mathbf{O}(\mathbf{r},\mathbf{s}) = \sup_{\mathbf{t}\in\mathbf{U}_I}\{T(\mathbf{P}(\mathbf{t},\mathbf{r}),\mathbf{P}(\mathbf{t},\mathbf{s}))\} \le \sup\{T(\alpha,\beta) : \mathsf{O}(\mathbf{r}_\alpha,\mathbf{s}_\beta)\}.$$

For any $\mathbf{t} \in \mathbf{U}_I$, let $\mathbf{t}^*$ be the fuzzy region defined by $\chi_{\mathbf{t}_1}$, i.e., $\mathbf{t}^*(x) = 1$ if $\mathbf{t}(x) = 1$, otherwise $\mathbf{t}^*(x) = 0$. By Eq. 4.9, it is straightforward to prove that $\mathbf{P}(\mathbf{t}^*,\mathbf{r}) = \sup\{\gamma : \mathsf{P}(\mathbf{t}_1,\mathbf{r}_\gamma)\}$ and $\mathbf{P}(\mathbf{t}^*,\mathbf{s}) = \sup\{\gamma : \mathsf{P}(\mathbf{t}_1,\mathbf{s}_\gamma)\}$.

Obviously, we have $\mathbf{t}^* \subseteq \mathbf{t}$. Suppose $\alpha = \mathbf{P}(\mathbf{t},\mathbf{r})$ and $\beta = \mathbf{P}(\mathbf{t},\mathbf{s})$. From Eq. 4.12 (see below) we know that $\alpha \le \mathbf{P}(\mathbf{t}^*,\mathbf{s})$ and $\beta \le \mathbf{P}(\mathbf{t}^*,\mathbf{s})$. Consequently, we have $\mathsf{P}(\mathbf{t}_1,\mathbf{r}_\alpha)$ and $\mathsf{P}(\mathbf{t}_1,\mathbf{s}_\beta)$. This implies $\mathsf{O}(\mathbf{r}_\alpha,\mathbf{s}_\beta)$.

On the other side, assume $\mathsf{O}(\mathbf{r}_\alpha,\mathbf{s}_\beta)$ holds for some α and β. That is, there exists some crisp region $t \in \mathbf{U}$ such that $\mathsf{P}(t, \mathbf{r}_\alpha)$ and $\mathsf{P}(t, \mathbf{s}_\beta)$. Consider the fuzzy region $\mathbf{t}^*$ defined by the characteristic function of t, i.e., $\mathbf{t}^*(x) = 1$ if $x \in t$, $\mathbf{t}^*(x) = 0$ otherwise. By Eq. 4.9, we have $\mathbf{P}(\mathbf{t}^*, \mathbf{r}) \geq \alpha$ and $\mathbf{P}(\mathbf{t}^*, \mathbf{s}) \geq \beta$. Therefore, $\mathbf{O}(\mathbf{r}, \mathbf{s}) = \sup_{\mathbf{t} \in \mathbf{U}_I} \{T(\mathbf{P}(\mathbf{t}, \mathbf{r}), \mathbf{P}(\mathbf{t}, \mathbf{s}))\} \geq T(\mathbf{P}(\mathbf{t}^*, \mathbf{r}), \mathbf{P}(\mathbf{t}^*, \mathbf{s})) \geq T(\alpha, \beta)$.

As a direct corollary, we have

Corollary 4.1 *Suppose $\boldsymbol{r}$ and $\mathbf{s}$ are two fuzzy regions in a standard fuzzy RCC model $(\boldsymbol{U}_I, \mathbf{C})$. If $\mathsf{ran}(\boldsymbol{r}) = \Sigma_1$ and $\mathsf{ran}(\boldsymbol{s}) = \Sigma_2$ are both finite, then the following equations hold:*

$$
\begin{aligned}
\mathbf{C}(\boldsymbol{r}, \boldsymbol{s}) &= \max\{T(\alpha, \beta) : C(\boldsymbol{r}_\alpha, \boldsymbol{s}_\beta), \alpha \in \Sigma_1, \beta \in \Sigma_2\}. \\
\mathbf{O}(\boldsymbol{r}, \boldsymbol{s}) &= \max\{T(\alpha, \beta) : O(\boldsymbol{r}_\alpha, \boldsymbol{s}_\beta), \alpha \in \Sigma_1, \beta \in \Sigma_2\}, \\
\mathbf{P}(\boldsymbol{r}, \boldsymbol{s}) &= \min\{I(\alpha, \max\{\beta : P(\boldsymbol{r}_\alpha, \boldsymbol{s}_\beta), \beta \in \Sigma_2\}) : \alpha \in \Sigma_1\}, \\
\mathbf{N}(\boldsymbol{r}, \boldsymbol{s}) &= \min\{I(\alpha, \max\{\beta : NTPP(\boldsymbol{r}_\alpha, \boldsymbol{s}_\beta), \beta \in \Sigma_2\}) : \alpha \in \Sigma_1\}.
\end{aligned}
$$

Remark 4.1 *This corollary gives an effective way for computing the fuzzy topological relation between two finite valued fuzzy regions. The well-known Egg-Yolk model, first introduced in [11] and then extended in [96], represents each vague region as k nested crisp regions. The vague regions in this model, called* Egg-Yolk regions*, can be naturally regarded as specific fuzzy regions that take values from $\Sigma_k = \{0, 1/k, 2/k, , (k-1)/k, 1\}$. The fuzzy connectedness of two Egg-Yolk regions is defined in terms of the standard connectedness on the n-dimensional Euclidean space [96, Eq. 85]. Schockaert et al. also show how to compute the corresponding* $\mathbf{O}, \mathbf{P}, \mathbf{N}$ *relations between two Egg-Yolk regions [96, Page 280]. Finally, they prove that any satisfiable fuzzy RCC-8 network also has solutions in Egg-Yolk models [96, Prop. 6].*

It is worth pointing out that the Egg-Yolk models in [96] resemble the standard fuzzy RCC models. In fact the Egg-Yolk models can be regarded as prototypes of standard fuzzy RCC models in some sense, despite the following differences. First of all, a standard fuzzy RCC model considers the class of all fuzzy regions, while an Egg-Yolk model is defined for Egg-Yolk regions, which can be regarded as specific fuzzy regions that take values from Σ_k. Second, the fuzzy connectedness relations of these two kinds of models are defined in a similar way from some standard (classical) RCC models (cf. Eq. 4.7 and [96, Eq. 85]), but on different domains: the Egg-Yolk *models are defined in the n-dimensional Euclidean space, while our standard fuzzy models are defined on connected regular T_1 topological spaces. One can easily check that Eq. 4.7 is equivalent to, but more concise than [96, Eq. 85] as far as Egg-Yolk regions are concerned.*

Our Theorem 4.1 shows that the fuzzy RCC-8 relations $\mathbf{O},\mathbf{P}$ *and* $\mathbf{N}$ *defined as in Table 4.1 have similar characterisations as* $\mathbf{C}$*. Compared with similar results claimed in [96, Page 280] for the Egg-Yolk models, our result is more general and is proved without assuming a certain underlying topological space, e.g. the Euclidean space. Note that the proof of Theorem 4.1 is applicable to the Egg-Yolk models.*

4.4 The Consistency Problem

The composition problem and the consistency problem are respectively the most simple and fundamental reasoning problems in QSTR. As we have fuzzified the RCC model and the RCC-8 Algebra, it is natural to study these problems in the fuzzy context. In fact, the composition problem has already been addressed in [95], and we now summarise some results obtained there.

Proposition 4.1 *Suppose* $\boldsymbol{r} \subseteq \boldsymbol{s}$ *are fuzzy regions in a fuzzy RCC model. Then*

$$\mathbf{C}(\boldsymbol{r},\boldsymbol{t}) \leq \mathbf{C}(\boldsymbol{s},\boldsymbol{t}), \qquad \mathbf{O}(\boldsymbol{r},\boldsymbol{t}) \leq \mathbf{O}(\boldsymbol{s},\boldsymbol{t}), \tag{4.11}$$

$$\mathbf{P}(\boldsymbol{r},\boldsymbol{t}) \geq \mathbf{P}(\boldsymbol{s},\boldsymbol{t}), \qquad \mathbf{N}(\boldsymbol{r},\boldsymbol{t}) \geq \mathbf{N}(\boldsymbol{s},\boldsymbol{t}). \tag{4.12}$$

This proposition follows directly from the definitions of **P**, **O** and **N**.

Lemma 4.1 ([95]) *For fuzzy regions* $\boldsymbol{r},\boldsymbol{s}$ *in a fuzzy RCC model, we have* [1]

$$\mathbf{C}(\boldsymbol{r},\boldsymbol{s}) = \mathbf{C}(\boldsymbol{s},\boldsymbol{r}), \quad \mathbf{O}(\boldsymbol{r},\boldsymbol{s}) = \mathbf{O}(\boldsymbol{s},\boldsymbol{r}), \tag{4.13}$$

$$\mathbf{N}(\boldsymbol{r},\boldsymbol{s}) \leq \mathbf{P}(\boldsymbol{r},\boldsymbol{s}) \leq \mathbf{O}(\boldsymbol{r},\boldsymbol{s}) \leq \mathbf{C}(\boldsymbol{r},\boldsymbol{s}), \tag{4.14}$$

$$\mathbf{N}(\boldsymbol{r},\boldsymbol{r}) \leq \mathbf{P}(\boldsymbol{r},\boldsymbol{r}) = \mathbf{O}(\boldsymbol{r},\boldsymbol{r}) = \mathbf{C}(\boldsymbol{r},\boldsymbol{r}) = 1. \tag{4.15}$$

[1] Note although not proved in [95], $\mathbf{N}(\mathbf{r},\mathbf{s}) \leq \mathbf{P}(\mathbf{r},\mathbf{s})$ follows directly from their definitions and the fact that $\mathbf{O}(\mathbf{r},\mathbf{s}) \leq \mathbf{C}(\mathbf{r},\mathbf{s})$.

	DC	EC	PO	EQ	TPP	NTPP	TPP^{-1}	NTPP^{-1}
$\mathbf{C}(a,b)$	0	1	1	1	1	1	1	1
$\mathbf{O}(a,b)$	0	0	1	1	1	1	1	1
$\mathbf{P}(a,b)$	0	0	0	1	1	1	0	0
$\mathbf{N}(a,b)$	0	0	0	0	0	1	0	0
$\mathbf{P}(b,a)$	0	0	0	1	0	0	1	1
$\mathbf{N}(b,a)$	0	0	0	0	0	0	0	1

Table 4.2: Relations between the crisp and fuzzy RCC-8 relations for crisp regions a and b

Furthermore, we have the following composition rules which correspond to the composition table of the classical RCC-8 Algebra.

Lemma 4.2 ([96]) *For fuzzy regions* $\boldsymbol{r},\boldsymbol{s}$ *and* $\boldsymbol{t}$ *(which could be identical) in a fuzzy RCC model, we have*

$$T(\mathbf{C}(\boldsymbol{r},\boldsymbol{s}),\mathbf{P}(\boldsymbol{s},\boldsymbol{t})) \le \mathbf{C}(\boldsymbol{r},\boldsymbol{t}), \tag{4.16}$$

$$T(\mathbf{C}(\boldsymbol{r},\boldsymbol{s}),\mathbf{N}(\boldsymbol{s},\boldsymbol{t})) \le \mathbf{O}(\boldsymbol{r},\boldsymbol{t}), \tag{4.17}$$

$$T(\mathbf{O}(\boldsymbol{r},\boldsymbol{s}),\mathbf{P}(\boldsymbol{s},\boldsymbol{t})) \le \mathbf{O}(\boldsymbol{r},\boldsymbol{t}), \tag{4.18}$$

$$T(\mathbf{P}(\boldsymbol{r},\boldsymbol{s}),\mathbf{P}(\boldsymbol{s},\boldsymbol{t})) \le \mathbf{P}(\boldsymbol{r},\boldsymbol{t}), \tag{4.19}$$

$$T(\mathbf{P}(\boldsymbol{r},\boldsymbol{s}),\mathbf{N}(\boldsymbol{s},\boldsymbol{t})) \le \mathbf{N}(\boldsymbol{r},\boldsymbol{t}), \tag{4.20}$$

$$T(\mathbf{N}(\boldsymbol{r},\boldsymbol{s}),\mathbf{P}(\boldsymbol{s},\boldsymbol{t})) \le \mathbf{N}(\boldsymbol{r},\boldsymbol{t}). \tag{4.21}$$

The fuzzy RCC-8 relations between fuzzy regions $\mathbf{r}$ and $\mathbf{s}$ can be completely captured by a 6-tuple $\langle \mathbf{C}(\mathbf{r},\mathbf{s}), \mathbf{O}(\mathbf{r},\mathbf{s}), \mathbf{P}(\mathbf{r},\mathbf{s}), \mathbf{N}(\mathbf{r},\mathbf{s}), \mathbf{P}(\mathbf{s},\mathbf{r}), \mathbf{N}(\mathbf{s},\mathbf{r})\rangle$. In particular, if $\mathbf{r}$ and $\mathbf{s}$ are crisp regions, then the elements in the 6-tuple take values from $\{0,1\}$. The constraints in Lemma 4.1 confines all the $2^6 = 32$ possibilities of the 6-tuple to 11, three ($\langle 1,1,1,0,1,1\rangle$, $\langle 1,1,1,1,1,0\rangle$, and $\langle 1,1,1,1,1,1\rangle$) of which are not realizable by crisp regions. Each of the remaining 8 possible 6-tuples corresponds to a crisp RCC-8 relation, shown in Table 4.2. Furthermore, it can be verified that if only crisp regions are considered, then the RCC-8 composition table can be derived by Lemma 4.2. Therefore, the equations in the above lemma are the fuzzified composition table of RCC-8.

It is worth noting that for the finite valued fuzzy region $\mathbf{r}$, we always have $\mathbf{N}(\mathbf{r},\mathbf{r}) < 1$. Although it is possible for some region $\mathbf{r}$ such that $\mathbf{N}(\mathbf{r},\mathbf{r}) = 1$, this requires a "continuous" characteristic function in some sense (e.g., a fuzzy region with a trapezoid-shape characteristic function).

The definitions of constraints and consistency can be generalised to fuzzy case in the following way.

Definition 4.3 ([96]) *Let $V = \{v_i\}_{i=1}^n$ be a set of spatial variables. An* atomic fuzzy RCC formula *over V has the form $\boldsymbol{R}(v_i,v_j) \leq \lambda$ or $\boldsymbol{R}(v_i,v_j) \geq \lambda$, where $\lambda \in [0,1]$, v_i, v_j are spatial variables in V, and $\boldsymbol{R}$ is a fuzzy RCC-8 relation defined in Table 4.1.*

Definition 4.4 ([96]) *A set Γ of atomic fuzzy RCC formula is* consistent *or* satisfiable*, if there exists a fuzzy connection structure $(\boldsymbol{U},\mathbf{C})$ and a set of regions $\{\boldsymbol{a}_i\}_{i=1}^n$ in $\boldsymbol{U}$, such that $\boldsymbol{R}(\boldsymbol{a}_i,\boldsymbol{a}_j) \leq \lambda$ (or $\boldsymbol{R}(\boldsymbol{a}_i,\boldsymbol{a}_j) \geq \lambda$) holds for each formula $\boldsymbol{R}(v_i,v_j) \leq \lambda$ (or $\boldsymbol{R}(v_i,v_j) \geq \lambda$) in Γ. In this case, we call $\{\boldsymbol{a}_i\}_{i=1}^n$ a* solution *of Γ.*

Definition 4.5 ([96]) *A set Γ of atomic fuzzy RCC-8 formula over V is called a* normalised fuzzy RCC-8 network *or a* normalised network *for short, if (i) there exist two formulae of the form $\boldsymbol{R}(v_i,v_j) \geq \lambda$ and $\boldsymbol{R}(v_i,v_j) \leq \lambda$ ($\lambda \in [0,1]$) for each fuzzy RCC relation $\boldsymbol{R}$ in $\{\mathbf{C},\mathbf{P},\mathbf{O},\mathbf{N}\}$ and each pair of v_i, v_j, and (ii) Γ contains no other formulae.*

From now on, we assume a normalised network Γ has the form

$$\Gamma = \{\mathbf{C}(v_i,v_j) = \lambda_{ij}^{\mathbf{C}}, \mathbf{O}(v_i,v_j) = \lambda_{ij}^{\mathbf{O}}, \mathbf{P}(v_i,v_j) = \lambda_{ij}^{\mathbf{P}}, \mathbf{N}(v_i,v_j) = \lambda_{ij}^{\mathbf{N}}\}_{i,j=1}^n. \tag{4.22}$$

The consistency problem of normalised/standard fuzzy RCC-8 networks has been studied by Schockaert et al. in [96]. In detail, it is shown by [96, Corollary 1] that a normalised fuzzy RCC-8 network Γ with all $\lambda_{ii}^{\mathbf{N}} < 1$ is satisfiable iff Γ satisfies (i) Eq.s 4.23-4.25, which will henceforth be called *basic conditions*.

$$\lambda_{ij}^{\mathbf{C}} = \lambda_{ji}^{\mathbf{C}}, \qquad \lambda_{ij}^{\mathbf{O}} = \lambda_{ji}^{\mathbf{O}}, \tag{4.23}$$

$$\lambda_{ij}^{\mathbf{N}} \leq \lambda_{ij}^{\mathbf{P}} \leq \lambda_{ij}^{\mathbf{O}} \leq \lambda_{ij}^{\mathbf{C}}, \tag{4.24}$$

$$\lambda_{ii}^{\mathbf{N}} < \lambda_{ii}^{\mathbf{C}} = \lambda_{ii}^{\mathbf{O}} = \lambda_{ii}^{\mathbf{P}} = 1, \tag{4.25}$$

and (ii) a set of 31 transitivity rules. Each transitivity rule has the form $T(\delta_{ij},\delta_{jk}) \leq \delta_{ik}$, where either δ_{ij} or $1 - \delta_{ij}$ takes a value from $\{\lambda_{ij}^{\mathbf{C}}, \lambda_{ij}^{\mathbf{O}}, \lambda_{ij}^{\mathbf{P}}, \lambda_{ij}^{\mathbf{N}}, \lambda_{ji}^{\mathbf{P}}, \lambda_{ji}^{\mathbf{N}}\}$.

The sufficiency part of their result ([96, Proposition 1]) relies on the construction of a solution in a particular kind of fuzzy connection structure, called the $(n;\alpha,0)$-model, which is defined as follows. The universe $\mathbf{U}_I$ is the set of all normalised fuzzy sets in $\mathbb{R}^n$, and the connectedness relation $\mathbf{C}_\alpha$ is defined by

$$\mathbf{C}_\alpha(\mathbf{r},\mathbf{s}) = \sup_{p\in\mathbb{R}^n} T(\mathbf{r}(p), \sup_{q\in\mathbb{R}^n} T(R_\alpha(p,q),\mathbf{s}(q))), \tag{4.26}$$

where $R_\alpha(p,q)$ is 1 if $d(p,q) \le \alpha$, and 0 otherwise. Intuitively, two points are considered as *close* if their distance is no more than α, and *far* otherwise. The degree of the connectedness of two fuzzy regions $\mathbf{r},\mathbf{s}$ is decided by the supremum of $T(\mathbf{r}(p),\mathbf{s}(q))$, where points p and q are considered as close.

Given a particular standard fuzzy RCC model (fuzzy connection structure) rather than the $(n;\alpha,0)$-model, however, the problem of whether a normalised fuzzy RCC-8 network has a solution in this standard fuzzy RCC model (fuzzy connection structure) remains open. We will show that if a normalised fuzzy RCC-8 network is satisfiable (i.e., it has a solution in *some* standard fuzzy RCC model), then it also has a solution in *any* standard fuzzy RCC model (see Theorem 4.2). This generalises the result of [96, Proposition 1]. Furthermore, we find that only the following 6 rules (Eq.s 4.27-4.32) of the 31 transitivity rules in [96] are actually necessary, because the other 25 rules can be derived from these 6 rules and the basic conditions. Besides, the fuzzy RCC-8 networks considered in [96] are assumed to satisfy $\lambda_{ii}^{\mathbf{N}} < 1$. We show that the $\lambda_{ii}^{\mathbf{N}} < 1$ restriction can be removed, although a slightly different construction is needed as a trade-off (see Remark 4.3).

$$T(\lambda_{ij}^{\mathbf{C}},\lambda_{jk}^{\mathbf{P}}) \le \lambda_{ik}^{\mathbf{C}}, \tag{4.27}$$

$$T(\lambda_{ij}^{\mathbf{C}},\lambda_{jk}^{\mathbf{N}}) \le \lambda_{ik}^{\mathbf{O}}, \tag{4.28}$$

$$T(\lambda_{ij}^{\mathbf{O}},\lambda_{jk}^{\mathbf{P}}) \le \lambda_{ik}^{\mathbf{O}}, \tag{4.29}$$

$$T(\lambda_{ij}^{\mathbf{P}},\lambda_{jk}^{\mathbf{P}}) \le \lambda_{ik}^{\mathbf{P}}, \tag{4.30}$$

$$T(\lambda_{ij}^{\mathbf{P}},\lambda_{jk}^{\mathbf{N}}) \le \lambda_{ik}^{\mathbf{N}}, \tag{4.31}$$

$$T(\lambda_{ij}^{\mathbf{N}},\lambda_{jk}^{\mathbf{P}}) \le \lambda_{ik}^{\mathbf{N}}. \tag{4.32}$$

The following theorem shows that the basic conditions and the reduced transitivity conditions are sufficient for Γ to be satisfiable in any standard fuzzy RCC model, generalizing the results of [96] of particular models.

Theorem 4.2 *Let $(\boldsymbol{U}_I, \mathbf{C})$ be a standard fuzzy RCC model. Suppose Γ in Eq. 4.22 is a normalised fuzzy network with all $\lambda_{ii}^{\mathbf{N}} < 1$. Then Γ is satisfiable in $(\boldsymbol{U}_I, \mathbf{C})$ iff it satisfies the basic conditions (Eq.s 4.23-4.25), and the transitivity rules (Eq.s 4.27-4.32).*

The necessity of Theorem 4.2 follows directly from Lemmas 4.1 and 4.2. We leave the sufficiency part to the next section, where we describe a model-independent method for constructing a solution of Γ.

Proposition 6 of [96] states that, if Γ is a consistent fuzzy RCC network and formulas in Γ take bounds from certain finite set, then it has an Egg-Yolk model. Compared with this proposition, Theorem 4.2 is more general: it holds for any consistent fuzzy network and any standard fuzzy RCC model. We prove this result by constructing a solution in polynomial time.

4.5 A Polynomial Realisation Algorithm

The realisation algorithm is inspired by the algorithm for crisp RCC (see Section 2.5). Note that for any regions a and b in a crisp $(\mathbf{U}, \mathsf{C})$ RCC model, the following claims hold:

(P1) There exists c s.t. $\mathsf{NTPP}(c, a)$.

(P2) If $\mathsf{DC}(a, b)$, then there exists c s.t. $\mathsf{NTPP}(a, c)$ and $\mathsf{DC}(c, b)$.

(P3) If $\mathsf{PO}(a, b)$ or $\mathsf{EC}(a, b)$, then there exists c s.t. $\mathsf{NTPP}(a, c)$ and $\mathsf{PO}(c, b)$.

Now we turn to the fuzzy case. Let $(\mathbf{U}_I, \mathbf{C})$ be a standard fuzzy RCC model. Suppose Γ is a normalised fuzzy RCC-8 network defined as in Eq. 4.22. Assume all the $\lambda_{ii}^{\mathbf{N}}$ is strictly less than 1 and Γ satisfies Eq.s 4.23-4.32. We next construct a solution of Γ in $(\mathbf{U}_I, \mathbf{C})$.

Although n ntpp-shells are enough in the crisp case, more ntpp-shells (depending on the $\mathbf{N}$ constraints) are needed here. This is because the degree of a fuzzy region has to be lowered gradually to satisfy the **N** constraints. Consider constraint $\lambda^{\mathbf{N}}_{11} = 1 - 1/k$. According to Eq. 4.10, this constraint requires that, for two β_1-cut and β_2-cut, if there is no other different α-cut between them, then the difference of β_1 and β_2 is no more than $1/k$. In other words, the characteristic function of the region cannot contain "steps" higher than $1/k$. Therefore the fuzzy region has at least k different α-cuts, and we have to construct at least k ntpp-shells.

Before we go into the details of the construction, we first introduce some notations which clarify the argument.

Definition 4.6 *Let $(\boldsymbol{U}, \mathsf{C})$ be a crisp RCC model, with the introduced fuzzy RCC model $(\boldsymbol{U}_I, \mathbf{C})$. Assume Ω is a set of crisp regions in $\mathbf{U}$, and $\mathbf{r}$ a fuzzy region in $\mathbf{U}_I$. $\mathbf{r}$ is said to be* based on Ω*, if $\mathsf{ran}(\mathbf{r})$ is finite and every α-cut of $\mathbf{r}$ is the union of some regions in Ω. The* standard decomposition *of $\mathbf{r}$ (with respect to Ω) is defined as*

$$\bigvee_{u \in \Omega} (\chi_u \wedge \kappa(u)), \tag{4.33}$$

where $\kappa(u) = \max\{\alpha \in \mathsf{ran}(\mathbf{r}) : u \subseteq \mathbf{r}_\alpha\}$.

It is straightforward to show that $\mathbf{r}$ is equal to its standard decomposition.

We first select a set of crisp regions Ω^0 from the underlying crisp RCC model $(\mathbf{U}, \mathsf{C})$:

$$\Omega^0 = \{u_i\}_{i=1}^n \cup \{u^{\mathbf{C}}_{ij} : i \neq j\} \cup \{u^{\mathbf{O}}_{ij} : i < j\}, \tag{4.34}$$

such that the RCC-8 relation between any two different u, u' in Ω^0 is DC, except we require $\mathsf{EC}(u^{\mathbf{C}}_{ij}, u^{\mathbf{C}}_{ji})$. The pair of crisp regions $u^{\mathbf{C}}_{ij}$ and $u^{\mathbf{C}}_{ji}$ are selected to fulfil the **C** constraints between v_i and v_j, while the crisp region $u^{\mathbf{O}}_{ij}$ is for the **O** constraints. In the crisp case, similar regions are selected for the EC and PO constraints respectively.

Next we select the ntpp-shells. The number of the shells depends on the **N** constraints, with upper bound $L = n\lceil 1/\Delta \rceil$, where

$$\Delta = 1 - \max\{\lambda^{\mathbf{N}}_{ij} : \lambda^{\mathbf{N}}_{ij} < 1\}. \tag{4.35}$$

To simplify the proof, we construct $L + 1$ ntpp-shells. In detail, we associate each region $u = u^0 \in \Omega^0$ with $L + 1$ (crisp) nested regions $u^1, u^2, \cdots, u^{L+1}$, such that

- $\mathsf{NTPP}(u^h, u^{h+1})$ for $0 \leq h \leq L$;

- If $\mathsf{DC}(u^0, v^0)$, then $\mathsf{DC}(u^{L+1}, v^{L+1})$; and
- If $\mathsf{EC}(u^0, v^0)$, then $\mathsf{PO}(u^{L+1}, v^0)$ and $\mathsf{PO}(u^0, v^{L+1})$.

In other words, we also require u^h is a non-tangential part of u^{h+1}, and u^{L+1} is close enough to u^0 such that it does not meet or overlap any other regions if possible. It can be proved that these regions can be selected in an arbitrary RCC model, which has properties P1-P3 (see §2.2).

Let $\Omega^h = \{u^k : u \in \Omega^0, 0 \le k \le h\}$, where $0 \le h \le L+1$ be the $(k+1)$-innermost shells. We call Ω^{L+1} the set of *base regions*.

Define the *one-shell-expansion function* $\mathsf{expand} : \Omega^L \to \Omega^{L+1}$ as

$$\mathsf{expand}(u^h) = u^{h+1} \qquad (u^h \in \Omega^L). \tag{4.36}$$

This function can be extended to the fuzzy regions based on Ω^L. For a fuzzy region $\mathbf{r}$ which is based on Ω^L and has standard decomposition $\bigvee_{u \in \Omega^L} \chi_u \wedge \kappa(u)$, the one-shell-expansion of $\mathbf{r}$ is defined to be the fuzzy region

$$\mathsf{expand}(\mathbf{r}) = \bigvee_{u \in \Omega^L} \chi_{\mathsf{expand}(u)} \wedge \kappa(u). \tag{4.37}$$

For fuzzy regions $\mathbf{r}$ and $\mathbf{s}$ based on Ω^L, the fuzzy RCC-8 relations between them can be computed as follows.

Lemma 4.3 *Let* $\boldsymbol{r}$ *and* $\boldsymbol{s}$ *be two fuzzy regions based on* Ω^L*. Suppose the standard decomposition of* $\boldsymbol{r}$ *and* $\boldsymbol{s}$ *are* $\bigvee_{u \in \Omega^L} \chi_u \wedge \kappa(u)$ *and* $\bigvee_{u \in \Omega^L} \chi_u \wedge \mu(u)$*, respectively. Then*

$$\begin{aligned}
\mathbf{C}(\boldsymbol{r},\boldsymbol{s}) &= \max\{T(\kappa(u),\mu(v)) : u,v \in \Omega^L, C(u,v)\}, && (4.38)\\
\mathbf{O}(\boldsymbol{r},\boldsymbol{s}) &= \max\{T(\kappa(u),\mu(v)) : u,v \in \Omega^L, O(u,v)\}, && (4.39)\\
\mathbf{P}(\boldsymbol{r},\boldsymbol{s}) &= \min\{I(\kappa(u),\mu(u)) : u \in \Omega^L\}, && (4.40)\\
\mathbf{N}(\boldsymbol{r},\boldsymbol{s}) &= \min(\{I(\kappa(u),\mu(\mathit{expand}(u)) : u \in \Omega^L\}\\
&= \mathbf{P}(\mathit{expand}(\boldsymbol{r}),\boldsymbol{s}), && (4.41)
\end{aligned}$$

where we assume $\mu(u^{L+1}) = 0$ *for any* $u^0 \in \Omega^0$ *in the last equation.*

Proof Note that for any $\alpha,\beta > 0$, $\mathbf{r}_\alpha$ and $\mathbf{s}_\beta$ are unions of subsets of Ω^L. Therefore, $\mathsf{C}(\mathbf{r}_\alpha, \mathbf{s}_\beta)$ iff there exist some $u, v \in \Omega^L$ such that $u \subseteq \mathbf{r}_\alpha$, $v \subseteq \mathbf{s}_\beta$ and $\mathsf{C}(u, v)$. Therefore,

$$\begin{aligned}
\mathbf{C}(\mathbf{r},\mathbf{s}) &= \sup\{T(\alpha,\beta) : \mathsf{C}(\mathbf{r}_\alpha,\mathbf{s}_\beta)\}\\
&= \sup\{T(\alpha,\beta) : (\exists u, v \in \Omega^L)(\kappa(u) \geq \alpha, \mu(v) \geq \beta, \mathsf{C}(u,v))\}\\
&= \max\{T(\kappa(u),\mu(v)) : u, v \in \Omega^L, \mathsf{C}(u,v)\}.
\end{aligned}$$

The second equation can be similarly proved. For the last two equations, we have

$$\begin{aligned}
\mathbf{P}(\mathbf{r},\mathbf{s}) &= \sup\{\alpha : (\forall\beta)\mathsf{P}(\mathbf{r}_\beta,\mathbf{s}_{T(\alpha,\beta)})\}\\
&= \sup\{\alpha : (\forall\beta)(\forall u \in \Omega^L)(u \subseteq \mathbf{r}_\beta \to u \subseteq \mathbf{s}_{T(\alpha,\beta)})\}\\
&= \sup\{\alpha : (\forall u \in \Omega^L)(\forall\beta)(\kappa(u) \geq \beta \to \mu(u) \geq T(\alpha,\beta))\}\\
&= \sup\{\alpha : (\forall u \in \Omega^L)(\mu(u) \geq T(\alpha,\kappa(u)))\}\\
&= \min\{I(\kappa(u),\mu(u)) : u \in \Omega^L\}.\\
\mathbf{N}(\mathbf{r},\mathbf{s}) &= \sup\{\alpha : (\forall\beta)\mathsf{NTPP}(\mathbf{r}_\beta,\mathbf{s}_{T(\alpha,\beta)})\}\\
&= \sup\{\alpha : (\forall\beta)(\forall u \in \Omega^L)(u \subseteq \mathbf{r}_\beta \to \mathsf{NTPP}(u,\mathbf{s}_{T(\alpha,\beta)}))\}\\
&= \sup\{\alpha : (\forall\beta)(\forall u \in \Omega^L)(u \subseteq \mathbf{r}_\beta \to \mathsf{expand}(u) \subseteq \mathbf{s}_{T(\alpha,\beta)})\}\\
&= \sup\{\alpha : (\forall u \in \Omega^L)(\forall\beta)(\kappa(u) \geq \beta \to \mu(\mathsf{expand}(u)) \geq T(\alpha,\beta))\}\\
&= \sup\{\alpha : (\forall u \in \Omega^L)(\mu(\mathsf{expand}(u)) \geq T(\alpha,\kappa(u)))\}\\
&= \min\{I(\kappa(u),\mu(\mathsf{expand}(u))) : u \in \Omega^L\}.\\
&= \mathbf{P}(\mathsf{expand}(\mathbf{r}),\mathbf{s}).
\end{aligned}$$

This ends the proof.

For a fuzzy region $\mathbf{r}$, let $\mathsf{sub}(\mathbf{r},\alpha)$ be the fuzzy region defined as

$$\mathsf{sub}(\mathbf{r},\alpha)(x) = \max(0, \mathbf{r}(x) - \alpha). \tag{4.42}$$

(i,j)	$(1,1)$	$(1,2)$	$(2,1)$	$(2,2)$
$\lambda_{ij}^{\mathbf{C}}$	1	1	1	1
$\lambda_{ij}^{\mathbf{O}}$	1	0.8	0.8	1
$\lambda_{ij}^{\mathbf{P}}$	1	0.6	0.4	1
$\lambda_{ij}^{\mathbf{N}}$	0.8	0.4	0	0.6

Table 4.3: A normalised fuzzy RCC-8 network Γ.

We now construct for each spatial variable a fuzzy region which is based on Ω^{L+1}. Let

$$\mathbf{a}_i^{\mathbf{C}} = \chi_{r_i} \vee \bigvee_{j<i} (\chi_{r_{ij}^{\mathbf{C}}} \wedge \lambda_{ij}^{\mathbf{C}}) \vee \bigvee_{j>i} \chi_{r_{ij}^{\mathbf{C}}}, \tag{4.43}$$

$$\mathbf{a}_i^{\mathbf{O}} = \mathbf{a}_i^{\mathbf{C}} \vee \bigvee_{j<i} (\chi_{r_{ij}^{\mathbf{O}}} \wedge \lambda_{ij}^{\mathbf{O}}) \vee \bigvee_{j>i} \chi_{r_{ij}^{\mathbf{O}}}, \tag{4.44}$$

$$\mathbf{a}_i^{\mathbf{N},0} = \mathbf{a}_i^{\mathbf{P}} = \bigvee_{j} \mathsf{sub}(\mathbf{a}_j^{\mathbf{O}}, 1-\lambda_{ji}^{\mathbf{P}}), \tag{4.45}$$

$$\mathbf{a}_i^{\mathbf{N},h+1} = \mathbf{a}_i^{\mathbf{N},h} \vee \bigvee_{j} \mathsf{sub}(\mathsf{expand}(\mathbf{a}_j^{\mathbf{N},h}), 1-\lambda_{ji}^{\mathbf{N}}), \tag{4.46}$$

where $0 \leq h \leq L-1$ in Eq. 4.46.

Theorem 4.3 *Let Γ be a normalised fuzzy network with all $\lambda_{ii}^{\mathbf{N}} < 1$. Suppose Γ satisfies the basic conditions (Eq.s 4.23-4.25), and the transitivity rules (Eq.s 4.27-4.32). Then the set of fuzzy regions $\{\boldsymbol{a}_i^{\mathbf{N},L}\}_{i=1}^n$ constructed in Eq.s 4.43-4.46 is a solution of Γ.*

Proof A sketch of the proof is provided in Section 4.6.

Note that in the **N** steps of the construction, if a fixed point has been reached, i.e., $\mathbf{a}_i^{N,h+1} = \mathbf{a}_i^{N,h}$ for all $1 \leq i \leq n$, then the algorithm can stop immediately.

Example 4.1 *Consider the normalised network defined in Table 4.3. It is easy to verify that Γ satisfies all the basic conditions and transitivity rules. We use the above method to construct a solution in the standard RCC model on the real line. There are in total seven steps of the procedure (four of which are for the **N** constraints), shown in Fig. 4.1, where the white region and the shaded region stand for fuzzy regions constructed for v_1 and v_2 respectively. We have $\boldsymbol{a}_i^{\mathbf{N},4} = \boldsymbol{a}_i^{\mathbf{N},L}$ for $i = 1, 2$, so $\{\boldsymbol{a}_1^{\mathbf{N},4}, \boldsymbol{a}_2^{\mathbf{N},4}\}$ is a solution of Γ.*

The pseudo-code of the construction procedure is given in Algorithm 2. Note that each fuzzy region $\mathbf{r}$ in the algorithm is based on Ω^L, and can be represented by an array with size $|\Omega^L| = O(Ln^2)$ storing the coefficients in its standard decomposition (see Eq. 4.33). Furthermore, the operations over fuzzy regions in the algorithms (e.g. intersection) can be carried out by manipulating the coefficients of the operands, which takes $O(Ln^2)$ time. There are $O(Ln^2)$ operations in total, so the time complexity of the algorithm is no more than $O(L^2n^4)$. The following proposition provides a better result.

Proposition 4.2 *The complexity of the construction procedure is $O(Ln^4)$, i.e. $O(\Delta n^5)$, where n is the cardinality of V, and Δ is defined in Eq. 4.35.*

Proof For any $0 \leq h < L$ and $1 \leq i \leq n$, we consider the coefficients of $\mathbf{a}_i^{\mathbf{N},h+1}$ and $\mathbf{a}_i^{\mathbf{N},h}$. By Proposition 4.5, we know that there are only $O(n^2)$ different coefficients (i.e. $\kappa_i^h(u^{h+1})$ and $\kappa_i^{h+1}(u^{h+1})$, for $u^0 \in \Omega^0$). Therefore the body of the last loop in the algorithm which updates $\mathbf{a}_i^{\mathbf{N},h}$ can be implemented in $O(n^2)$ time instead of $O(Ln^2)$. So the complexity of Algorithm 2 is $O(Ln^4)$.

The following proposition gives an estimation of the range of the constructed solution.

Proposition 4.3 *Let Γ be a consistent normalised network with $\lambda_{ii}^{\mathbf{N}} < 1$. Let $\boldsymbol{a}_i^{\mathbf{N},L}$ be the fuzzy regions defined as Eq.4.46. Let $\Sigma^{\mathbf{C},\mathbf{O}} = \{\lambda_{ij}^{\mathbf{C}}, \lambda_{ij}^{\mathbf{O}}\} \cup \{0\}$, $\Sigma^{\mathbf{P}} = \{\lambda_{ij}^{\mathbf{P}}\}$, $\Sigma^{\mathbf{N}} = \{\lambda_{ij}^{\mathbf{N}}\} \cup \{1\}$. Then we have*

$$\mathsf{ran}(\boldsymbol{a}_i^{\mathbf{N},L}) \subseteq \Sigma^L, \tag{4.47}$$

where $\Sigma^0 = T(\Sigma^{\mathbf{C},\mathbf{O}}, \Sigma^{\mathbf{P}})$, and $\Sigma^{h+1} = T(\Sigma^h, \Sigma^{\mathbf{N}})$ for $0 \leq h \leq L-1$.

In particular, if Σ contains $\{\lambda_{ij}^{\mathbf{C}}, \lambda_{ij}^{\mathbf{O}}, \lambda_{ij}^{\mathbf{P}}, \lambda_{ij}^{\mathbf{N}}\} \cup \{0, 1\}$ and is closed under operation T (e.g., $\Sigma_k = \{0, 1/k, 2/k, \ldots, (k-1)/k, 1\}$), then the degrees of all the fuzzy regions constructed above are also in Σ.

Remark 4.2 *It is worth noting that the proof of Theorem 4.1 and that of Lemma 4.1 only use Axiom C3 of the RCC model, which states that $\mathsf{C}(x, y+z)$ iff $\mathsf{C}(x, y)$ or $\mathsf{C}(x, z)$. Therefore, Theorem 4.1 still holds if a standard fuzzy RCC model is replaced with a fuzzy connection structure induced by some connection structure which satisfies C3.*

Furthermore, once the base regions are chosen, the proof of Theorem 4.3 does not rely on other axioms of the RCC structure either. This implies that Theorem 4.2 holds for general fuzzy connection structure as long as it supplies a sufficient number of base regions.

We consider the $(n;\alpha,0)$-model introduced by Schockaert et al. [94, 96]. Let $\boldsymbol{U}$ be the powerset of $\mathbb{R}^n$. Define C_α as

$$\mathsf{C}_\alpha = \{(r,s) : d(r,s) \leq \alpha\}, \tag{4.48}$$

where $d(r,s) = \inf\{d(p,q) : p \in r, q \in s\}$. It is clear that C_α is a connection structure that satisfies the axiom C3.

It is straightforward to verify that $(\boldsymbol{U}_I, \mathbf{C}_\alpha)$ is the fuzzy connection structure induced by $(\boldsymbol{U}, \mathsf{C}_\alpha)$. Moreover, it is always possible to choose base regions from $\mathbb{R}^n$ for any $n \geq 1$. Therefore, a network has a solution in the $(n;\alpha,0)$-model for any $n \geq 1$ iff it satisfies the basic conditions and the transitivity rules. This shows the Propositions 1 and 2 of [96] can be viewed as a special case of Theorem 4.2.

Remark 4.3 *Theorem 2 (and [96, Prop. 1] requires that $\lambda_{ii}^{\mathbf{N}} < 1$ for all $1 \leq i \leq n$. It is argued in [96] that this requirement has no effect if all* **N** *constraints are introduced by* **TPP** *and* **NTPP** *constraints [96, Propsition 2], which is usually the case in applications.*

Nonetheless, the satisfiability of fuzzy RCC network Γ without restriction $\lambda_{ii}^{\mathbf{N}} < 1$ is still interesting. It turns out that the basic conditions and the transitivity rules are also sufficient for the unrestricted case. However, "continuous" ntpp-shells may be needed in the constructions. To be precise, for each base region $u \in \Omega^0$, we need a function $f_u : [0,1] \to \boldsymbol{U}$, such that

- *$f_u(0) = u$;*
- *$\mathsf{NTPP}(f_u(p), f_u(q))$ for any $0 \leq p < q \leq 1$;*
- *If $\mathsf{DC}(u,v)$ for $u,v \in \Omega^0$, then $\mathsf{DC}(f_u(1), f_v(1))$; and*
- *If $\mathsf{EC}(u^0,v^0)$ for $u,v \in \Omega^0$, then $\mathsf{PO}(f_u(1), v)$ and $\mathsf{PO}(u, f_v(1))$.*

Let $\boldsymbol{a}_i^{\mathbf{P}} = \bigvee_{u\in\Omega^0} u \wedge \kappa_i(u)$ be the fuzzy region with standard decomposition as defined in Eq. 4.45, Define

$$\boldsymbol{a}_i^{\mathbf{N}} = \boldsymbol{a}_i^{\mathbf{P}} \vee \bigvee_{u\in\Omega^0} \Big(\bigvee_{p\in[0,1]} f_u(p) \wedge (\mu_i(u) - p) \Big), \tag{4.49}$$

where $\mu_i(u) = \max_{1\le j\le n}\{T(\kappa_j(u), \lambda_{ji}^{\mathbf{N}})\}$. In other words, the fuzzy region $\boldsymbol{a}_i^{\mathbf{N}}$ jumps down at the boundary of the regions in Ω^0 from degree $\kappa(u)$ to $\mu(u)$, then goes gradually to zero. It can be proved that $\{\boldsymbol{a}_i^{\mathbf{N}}\}_{i=1}^n$ is a solution of Γ. Interestingly, this construction is somewhat simpler than that used in the original case: $\boldsymbol{a}_i^{\mathbf{N}}$ has a direct expression instead of an iterated one. As a trade-off, the fuzzy regions in the solution are no longer finite valued.

4.6 Proof for Theorem 4.3

It is clear that each $\mathbf{a}_i^{\mathbf{N},h}$ is a fuzzy region based on Ω^h. Let $\bigvee_{u\in\Omega^h} \chi_u \wedge \kappa_i^h(u)$ be the standard decomposition of $\mathbf{a}_i^{\mathbf{N},h}$. First we give two technical results about the coefficients $\kappa_i^h(u)$.

Proposition 4.4 *For any $0 \le h < L$ and $1 \le i \le n$, we have*

$$\kappa_i^{h+1}(u^0) = \kappa_i^h(u^0), \tag{4.50}$$

$$\kappa_i^{h+1}(u^k) = \max_{1\le j\le n}\{\kappa_j^h(u^{k-1}) + \lambda_{ji}^{\mathbf{N}} - 1\} \cup \{\kappa_i^h(u^k)\}, 1 \le k \le L. \tag{4.51}$$

Proof The fact that $\kappa_i^1(u^0) = \kappa_i^0(u^0)$ can be proved by an exhaustive case analysis for $u^0 \in \Omega^0$ using the transitivity rules. To show Eq. 4.50 is not hard by an inductive method. Eq. 4.51 follows directly from Eq. 4.46.

The following proposition shows that the coefficients of a base region u^h are always zero until the h-step in the ntpp-construction, and remain the same value (possibly still 0) afterwards.

Proposition 4.5 *For $1 \le i \le n, u^0 \in \Omega^0, 0 \le h \le L$, we have*

$$\kappa_i^0(u^h) = \kappa_i^1(u^h) = \cdots = \kappa_i^{h-1}(u^h) = 0, \tag{4.52}$$

$$\kappa_i^h(u^h) = \kappa_i^{h+1}(u^h) = \cdots = \kappa_i^L(u^h). \tag{4.53}$$

Proof We prove Eq. 4.52 by showing the following equivalent equation:

$$\kappa_i^k(u^{k+1}) = \kappa_i^k(u^{k+2}) = \cdots = \kappa_i^k(u^L) = 0 \quad (0 \le k \le L).$$

By Eq.s 4.43-4.45 we know this equation holds for $k = 0$, because only regions in Ω^0 have been used. Assume the equation holds for some $0 \le k < L$. For $k' > k+1$, by Eq. 4.51,

$$\begin{aligned}\kappa_i^{k+1}(u^{k'}) &= \max\,\{\kappa_j^k(u^{k'-1}) + \lambda_{ji}^{\mathbf{N}} - 1 : 1 \le j \le n\} \cup \{\kappa_i^k(u^{k'})\}\\ &= \max\,\{0 + \lambda_{ji}^{\mathbf{N}} - 1 : 1 \le j \le n\} \cup \{0\}\\ &= 0.\end{aligned}$$

Therefore the equation also holds for $k+1$. So Eq. 4.52 has been proved.

For Eq. 4.53, the case $h = 0$ has already been proved in Prop. 4.4. We now prove $\kappa_i^2(u^1) = \kappa_i^1(u^1)$. By Eq. 4.53 we know that

$$\kappa_i^2(u^1) = \max\,\{\kappa_j^1(u^0) + \lambda_{ji}^{\mathbf{N}} - 1 : 1 \le j \le n\} \cup \{\kappa_i^1(u^1)\}.$$

Because $\kappa_j^1(u^0) = \kappa_j^0(u^0)$, we only need to prove for any $1 \le j \le n$,

$$\kappa_j^0(u^0) + \lambda_{ji}^{\mathbf{N}} - 1 \le \kappa_i^1(u^1),$$

which follows direct from Eq. 4.53.

By induction, similarly we can prove $\kappa_i^L(u^1) = \cdots = \kappa_i^2(u^1) = \kappa_i^1(u^1)$. Eq. 4.53 can be proved using induction one more time (on index h).

We now prove Theorem 3 by the following three lemmas.

Lemma 4.4 $\{\boldsymbol{a}_i^{\mathbf{N},L}\}_{i=1}^n$ *satisfies all the* $\mathbf{C}, \mathbf{O}$ *constraints in* Γ.

Proof It is obvious that $\mathbf{a}_i^{\mathbf{C}}, \mathbf{a}_i^{\mathbf{O}}, \mathbf{a}_i^{\mathbf{P}}$ are based on Ω^0. According to Lemma 4.3, it is straightforward to verify that $\mathbf{C}(\mathbf{a}_i^{\mathbf{O}}, \mathbf{a}_j^{\mathbf{O}}) = \lambda_{ij}^{\mathbf{C}}$ and $\mathbf{O}(\mathbf{a}_i^{\mathbf{O}}, \mathbf{a}_j^{\mathbf{O}}) = \lambda_{ij}^{\mathbf{O}}$.

We now show $\mathbf{O}(\mathbf{a}_i^{\mathbf{P}}, \mathbf{a}_j^{\mathbf{P}}) = \lambda_{ij}^{\mathbf{O}}$. It is clear that $\mathbf{a}_i^{\mathbf{O}} \subseteq \mathbf{a}_i^{\mathbf{P}}$, so $\mathbf{O}(\mathbf{a}_i^{\mathbf{P}}, \mathbf{a}_j^{\mathbf{P}}) \ge \mathbf{O}(\mathbf{a}_i^{\mathbf{O}}, \mathbf{a}_j^{\mathbf{O}}) = \lambda_{ij}^{\mathbf{O}}$.

Suppose $\mathbf{a}_i^{\mathbf{P}} = \bigvee_{u \in \Omega^0} \kappa(u)$ and $\mathbf{a}_j^{\mathbf{P}} = \bigvee_{u \in \Omega^0} \mu(u)$. Note that for any base region u, v in Ω^0, $\mathsf{O}(u, v)$ iff $u = v$. Therefore from Lemma 4.3 we have

$$\mathbf{O}(\mathbf{a}_i^{\mathbf{P}}, \mathbf{a}_j^{\mathbf{P}}) = \max\{T(\kappa(u), \mu(u)) : u \in \Omega^0\}.$$

It remains to show that $T(\kappa(u), \mu(u)) \leq \lambda_{ij}^{\mathbf{O}}$ for any $u \in \Omega^0$. This can be proved by case analysis, and we only discuss the case that $u = u_k$ where $k \neq i, j$. It can be computed that $\kappa(u_k) = \lambda_{ki}^{\mathbf{P}}$ and $\mu(u_k) = \lambda_{kj}^{\mathbf{P}}$. By the transitivity rule $T(\lambda_{ik}^{\mathbf{O}}, \lambda_{kj}^{\mathbf{P}}) \leq \lambda_{ij}^{\mathbf{P}}$ and basic condition $\lambda_{ki}^{\mathbf{P}} \leq \lambda_{ki}^{\mathbf{O}} = \lambda_{ik}^{\mathbf{O}}, \lambda_{ij}^{\mathbf{P}} \leq \lambda_{ij}^{\mathbf{O}}$, we have

$$T(\kappa(u_k), \mu(u_k)) = T(\lambda_{ki}^{\mathbf{P}}, \lambda_{kj}^{\mathbf{P}}) \leq T(\lambda_{ik}^{\mathbf{O}}, \lambda_{kj}^{\mathbf{P}}) \leq \lambda_{ij}^{\mathbf{P}} \leq \lambda_{ij}^{\mathbf{O}}.$$

That $\mathbf{C}(\mathbf{a}_i^{\mathbf{P}}, \mathbf{a}_j^{\mathbf{P}}) = \lambda_{ij}^{\mathbf{C}}$ and $\mathbf{P}(\mathbf{a}_i^{\mathbf{P}}, \mathbf{a}_j^{\mathbf{P}}) = \lambda_{ij}^{\mathbf{P}}$ can be similarly proved by exploiting Lemma 4.3, the basic conditions and the transitivity rules of the network Γ. We omit the details.

We next prove $\{\mathbf{a}_i^{\mathbf{N},L}\}_{i=1}^n$ satisfies all the $\mathbf{C}$ constraints. Note that $\mathsf{C}(u^k, v^l)$ iff $\mathsf{C}(u^0, v^0)$ for any two $u^k, v^l \in \Omega^L$. Moreover, since $\mathsf{NTPP}(u^0, u^k)$, we have $\kappa_i^h(u^k) \leq \kappa_i^h(u^0)$ for any $u \in \Omega, 0 \leq k \leq L, 0 \leq h \leq L$. So

$$\begin{aligned}
\mathbf{C}(\mathbf{a}_i^{\mathbf{N},L}, \mathbf{a}_j^{\mathbf{N},L}) &= \max\{T(\kappa_i^L(u^k), \kappa_j^L(v^l)) : u^k, v^l \in \Omega^L, \mathsf{C}(u^k, v^l)\} \\
&= \max\{T(\kappa_i^L(u), \kappa_j^L(v)) : u, v \in \Omega, \mathsf{C}(u, v)\} \\
&= \max\{T(\kappa_i^0(u), \kappa_j^0(v)) : u, v \in \Omega, \mathsf{C}(u, v)\} \\
&= \mathbf{C}(\mathbf{a}_i^{\mathbf{P}}, \mathbf{a}_j^{\mathbf{P}}) = \lambda_{ij}^{\mathbf{C}}.
\end{aligned}$$

Similarly we can prove $\{\mathbf{a}_i^{\mathbf{N},L}\}_{i=1}^n$ satisfies all the $\mathbf{O}$ constraints.

Lemma 4.5 *$\{\mathbf{a}_i^{\mathbf{N},L}\}_{i=1}^n$ satisfies all the* $\mathbf{P}$ *constraints in* Γ.

Proof We already have that $\mathbf{P}(\mathbf{a}_i^{\mathbf{N},0}, \mathbf{a}_j^{\mathbf{N},0}) = \lambda_{ij}^{\mathbf{P}}$. Assume $\mathbf{P}(\mathbf{a}_i^{\mathbf{N},h-1}, \mathbf{a}_j^{\mathbf{N},h-1}) = \lambda_{ij}^{\mathbf{P}}$ for some $0 \leq h < L$.

By Lemma 4.3 and Proposition 4.5, for any $0 < h \le L$, we have

$$\begin{aligned}
&\mathbf{P}(\mathbf{a}_i^{\mathbf{N},h}, \mathbf{a}_j^{\mathbf{N},h})\\
=&\min\{I(\kappa_i^h(u), \kappa_i^h(u)) : u \in \Omega^h\}\\
=&\min\{I(\kappa_i^h(u), \kappa_j^h(u)) : u \in \Omega^{h-1}\} \cup \{I(\kappa_i^h(u^h), \kappa_j^h(u^h)) : u \in \Omega^0\}\\
=&\min\{I(\kappa_i^{h-1}(u), \kappa_j^{h-1}(u)) : u \in \Omega^{h-1}\} \cup \{I(\kappa_i^h(u^h), \kappa_j^h(u^h)) : u \in \Omega^0\}
\end{aligned}$$

The following equation holds by the assumption:

$$\min\{I(\kappa_i^{h-1}(u^k), \kappa_j^{h-1}(u^k)) : u \in \Omega^{h-1}\} = \mathbf{P}(\mathbf{a}_i^{\mathbf{N},h-1}, \mathbf{a}_j^{\mathbf{N},h-1}) = \lambda_{ij}^{\mathbf{P}}.$$

We now show $I(\kappa_i(u^h), \kappa_j(u^h)) \ge \lambda_{ij}^{\mathbf{P}}$ for any $u \in \Omega^0$. By Proposition 4.4 and Proposition 4.5, we have

$$\begin{aligned}
\kappa_i^h(u^h) &= \max\{\kappa_i^{h-1}(u^h)\} \cup \{\kappa_k^{h-1}(u^{h-1}) + \lambda_{ki}^{\mathbf{N}} - 1 : 1 \le k \le n\}\\
&= \max\{\kappa_k^{h-1}(u^{h-1}) + \lambda_{ki}^{\mathbf{N}} - 1 : 1 \le k \le n\}.
\end{aligned}$$

Suppose $\kappa_k^{h-1}(r^{h-1}) + \lambda_{ki}^{\mathbf{N}} - 1$ gets the maximum value when $k = k_0$. Because the constraint network satisfies the transitivity rule $T(\lambda_{k_0 i}^{\mathbf{N}}, \lambda_{ij}^{\mathbf{P}}) \le \lambda_{k_0 j}^{\mathbf{N}}$, we have

$$\begin{aligned}
&1 - \kappa_i(u^h) + \kappa_j(u^h)\\
=&1 - \max\{\kappa_k^{h-1}(u^{h-1}) + \lambda_{ki}^{\mathbf{N}} - 1 : 1 \le k \le n\}\\
&+ \max\{\kappa_k^{h-1}(u^{h-1}) + \lambda_{kj}^{\mathbf{N}} - 1 : 1 \le k \le n\}\\
\ge&1 - (\kappa_{k_0}^{h-1}(u^{h-1}) + \lambda_{k_0 i}^{\mathbf{N}} - 1) + (\kappa_{k_0}^{h-1}(u^{h-1}) + \lambda_{k_0 j}^{\mathbf{N}} - 1)\\
=&1 - \lambda_{k_0 i}^{\mathbf{N}} + \lambda_{k_0 j}^{\mathbf{N}}\\
\ge&\lambda_{ij}^{\mathbf{P}}.
\end{aligned}$$

Therefore $I(\kappa_i(u^h), \kappa_j(u^h)) \ge \lambda_{ij}^{\mathbf{P}}$, i.e., $\mathbf{P}(\mathbf{a}_i^{\mathbf{N},h}, \mathbf{a}_j^{\mathbf{N},h}) = \lambda_{ij}^{\mathbf{P}}$.

We have the following proposition estimating the upper bound of $\max\{\kappa_i^h(u^h) : 0 \le i \le n, u \in \Omega^0\}$, i.e., the greatest degree of the outmost shell of region $\mathbf{a}_i^{\mathbf{N},h}$.

Proposition 4.6 *Suppose $0 \le h = pn + q \le L$, where $0 \le q < n$. Then*

$$\max\{\kappa_i^h(u^h) : 0 \le i \le n, u \in \Omega^0\} \le 1 - p(1 - m), \tag{4.54}$$

where $m = \max\{\lambda^{\mathbf{N}}_{ij} : \lambda^{\mathbf{N}}_{ij} < 1\}$. *Moreover, there are at most q different base regions* $u \in \Omega^0$ *such that* $\kappa^h_i(u^h)$ *gets the above maximum value.*

Proof The idea is to construct a superset of $\mathbf{a}^{\mathbf{N},h}_i$ with standard decomposition $\bigvee_{u\in\Omega^L}(\chi_u \wedge \iota^h_i(u))$, and to prove ι^h_i satisfies the proposition.

Let $\iota^0_i(u^0) = 1$, $\iota^0_i(u^k) = 0$ for any $1 \le i \le n$, $u \in \Omega^0$, $0 < k \le L$. Moreover, for any $0 \le h < L$, $\iota^{h+1}_i(u^k) = \max\{\iota^h_i(u^k)\} \cup \{\iota^h_j(u^{k-1}) + \lambda_{ji} - 1 : 1 \le j \le n\}$, where $\lambda_{ji} = 1$ iff $\lambda^{\mathbf{N}}_{ji} = 1$, otherwise $\lambda_{ji} = m$.

Therefore, for any $1 \le i \le n$, $0 \le h \le L$ and $u^k \in \Omega^L$, we have $\kappa^h_i(u^k) \le \iota^h_i(u^k)$, $\iota^h_i(u^{k+1}) \le \iota^h_i(u^k)$, and $\kappa^h_i(u^k)$ is either 0 or $1 - p(1-m)$ for some non-negative integer $p \le \lfloor 1/(1-m) \rfloor$. Similarly to Prop 4.5, we also have

$$\iota^0_i(u^k) = \iota^1_i(u^k) = \cdots = \iota^{k-1}_i(u^k) = 0, \tag{4.55}$$

$$\iota^k_i(u^k) = \iota^{k+1}_i(u^k) = \cdots = \iota^l_i(u^k). \tag{4.56}$$

For any $u \in \Omega^0$, let $\Delta_k = \max\{\iota^k_i(u^k) : 1 \le i \le n\}$. As $\iota^{k+1}_i(u^{k+1}) \le \iota^{k+1}_i(u^k) = \iota^k_i(u^k)$, we have $\Delta_{k+1} \le \Delta_k$. If $\Delta_{k+1} < \Delta_k$, apparently $\Delta_k - \Delta_{k+1} \ge 1 - m$ unless $\Delta_{k+1} = 0$.

In the case $\Delta_{k+1} = \Delta_k$, we assume $\mathcal{I} = \{i : \Delta_k = \iota^k_i(u^k)\}$ and $\mathcal{J} = \{i : \Delta_{k+1} = \iota^{k+1}_i(u^{k+1})\}$. By $\iota^k_i(u^k) = \iota^{k+1}_i(u^k) \ge \iota^{k+1}_i(u^{k+1})$, we have $\mathcal{J} \subseteq \mathcal{I}$. We now prove $\mathcal{J} \subsetneq \mathcal{I}$. Otherwise for any $i \in \mathcal{I}$, $\iota^k_i(u^k) = \iota^{k+1}_i(u^{k+1})$. Moreover, by $\iota^{k+1}_i(u^{k+1}) = \max\{\iota^k_j(u^k) + \lambda_{ji} - 1 : 1 \le j \le n\}$, we know for any $i_1 \in \mathcal{J}$, there exists some $i_2 \in \mathcal{I}$ such that $\lambda_{i_1 i_2} = 1$. This implies that there exist some i and j such that $\lambda_{ij} = \lambda_{ji} = 1$. Therefore, $\lambda^{\mathbf{N}}_{ij} = \lambda^{\mathbf{N}}_{ji} = 1$. As a consequence, $\lambda^{\mathbf{N}}_{ii} \ge T(\lambda^{\mathbf{P}}_{ij}, \lambda^{\mathbf{N}}_{ji}) \ge T(\lambda^{\mathbf{N}}_{ij}, \lambda^{\mathbf{N}}_{ji}) = 1$, which contradicts with $\lambda^{\mathbf{N}}_{ii} < 1$, one of the basic conditions of the constraints network. So $\mathcal{J} \subsetneq \mathcal{I}$, i.e., the number of the elements getting the maximum value is strictly decreasing.

Therefore ι^h_i satisfies the proposition, which is a stronger result as $\kappa^h_i(u^k) \le \iota^h_i(u^k)$.

Lemma 4.6 $\{\boldsymbol{a}^{\mathbf{N},L}_i\}^n_{i=1}$ *satisfies all the* $\mathbf{N}$ *constraints in* Γ.

Proof We first show that, for any $0 < h \le L$, the following equation holds.

$$\min\{I(\kappa^h_i(u), \kappa^h_j(\mathsf{expand}(u))) : u \in \Omega^{h-1}\} = \lambda^{\mathbf{N}}_{ij}. \tag{4.57}$$

For the case that $h = 1$,

$$
\begin{aligned}
&\min\{I(\kappa_i^1(u^0), \kappa_j^1(u^1)) : u \in \Omega^0\} \\
=&\min\{I(\kappa_i^0(u^0), \max\{\kappa_k^0(u^0) + \lambda_{kj}^{\mathbf{N}} - 1\}) : u \in \Omega^0\} \\
\geq&\min\{I(\kappa_i^0(u^0), \kappa_i^0(u^0) + \lambda_{ij}^{\mathbf{N}} - 1) : u \in \Omega^0\} \\
=&\lambda_{ij}^{\mathbf{N}}.
\end{aligned}
$$

To prove that the left hand side of Eq. 4.57 equals $\lambda_{ij}^{\mathbf{N}}$, we consider base region u_i. By the assumption that the constraint network satisfies the transitivity rule $T(\lambda_{ik}^{\mathbf{P}}, \lambda_{kj}^{\mathbf{N}}) \leq \lambda_{ij}^{\mathbf{N}}$, we have

$$
\begin{aligned}
\kappa_j^1(u_i^1) &= \max\{\kappa_k^0(u_i) + \lambda_{kj}^{\mathbf{N}} - 1 : 1 \leq k \leq n\} \\
&= \max\{\lambda_{ik}^{\mathbf{P}} + \lambda_{kj}^{\mathbf{N}} - 1 : 1 \leq k \leq n\} \\
&\leq \max\{\lambda_{ij}^{\mathbf{N}}\} = \lambda_{ij}^{\mathbf{N}}.
\end{aligned}
$$

Meanwhile, as $\lambda_{ii}^{\mathbf{P}} = 1$,

$$
\begin{aligned}
\kappa_j^1(u_i^1) &= \max\{\lambda_{ik}^{\mathbf{P}} + \lambda_{kj}^{\mathbf{N}} - 1 : 1 \leq k \leq n\} \\
&\geq \lambda_{ii}^{\mathbf{P}} + \lambda_{ij}^{\mathbf{N}} - 1 = \lambda_{ij}^{\mathbf{N}}.
\end{aligned}
$$

Therefore $\kappa_j^1(u_i^1) = \lambda_{ij}^{\mathbf{N}}$. So

$$
\min\{I(\kappa_i^1(u^0), \kappa_j^1(u^1)) : u \in \Omega^0\} \leq I(\kappa_i^1(u_i^0), \kappa_j^1(u_i^1)) = I(1, \lambda_{ij}^{\mathbf{N}}) = \lambda_{ij}^{\mathbf{N}}.
$$

Let $0 < h < L$. Suppose Eq. 4.57 holds for some h, we next show it also holds for $h + 1$.

$$
\begin{aligned}
&\min\{I(\kappa_i^{h+1}(u), \kappa_j^{h+1}(\mathsf{expand}(u))) : u \in \Omega^h\} \\
=&\min\{I(\kappa_i^{h+1}(u^k), \kappa_j^{h+1}(u^{k+1})) : u \in \Omega^0, 0 \leq k \leq h-1\} \\
&\cup \{I(\kappa_i^{h+1}(u^h), \kappa_j^{h+1}(u^{h+1})) : u \in \Omega^0\} \\
=&\min\{I(\kappa_i^{h}(u^k), \kappa_j^{h}(u^{k+1})) : u \in \Omega^0, 0 \leq k \leq h-1\} \\
&\cup \{I(\kappa_i^{h+1}(u^h), \kappa_j^{h+1}(u^{h+1})) : u \in \Omega^0\} \\
=&\min\{\lambda_{ij}^{\mathbf{N}}\} \cup \{I(\kappa_i^{h+1}(u^h), \kappa_j^{h+1}(u^{h+1})) : u \in \Omega^0\}.
\end{aligned}
$$

So we only need to show $I(\kappa_i^{h+1}(u^h), \kappa_j^{h+1}(u^{h+1})) \geq \lambda_{ij}^{\mathbf{N}}$ for any $u \in \Omega^0$. This follows from

$$\begin{aligned}
I(\kappa_i^{h+1}(u^h), \kappa_j^{h+1}(u^{h+1})) =& 1 - \kappa_i^{h+1}(u^h) + \kappa_j^{h+1}(u^{h+1}) \\
=& 1 - \kappa_i^h(u^h) + \max\{\kappa_k^h(u^h) + \lambda_{kj}^{\mathbf{N}} - 1 : 1 \le k \le n\} \\
\ge& 1 - \kappa_i^h(u^h) + (\kappa_i^h(u^h) + \lambda_{ij}^{\mathbf{N}} - 1) = \lambda_{ij}^{\mathbf{N}}.
\end{aligned}$$

It remains to prove that

$$\min\{I(\kappa_i^L(u), \kappa_j^L(\mathsf{expand}(u))) : u \in \Omega^L \smallsetminus \Omega^{L-1}\} \ge \lambda_{ij}^{\mathbf{N}}. \tag{4.58}$$

By the fact that $\kappa_j^L(\mathsf{expand}(u^L)) = 0$ for any $u \in \Omega^0$, we only need to show $\kappa_i^L(u^L) \le 1 - \lambda_{ij}^{\mathbf{N}}$. It follows directly from Proposition 4.6, that we have

$$\max\{\kappa_i^L(u^L) : 0 \le i \le n, u \in \Omega^0\} \le 1 - (1-m)\lceil 1/(1-m) \rceil \le 0.$$

Therefore, $\kappa_i^L(u^L) = 0 \le 1 - \lambda_{ij}^{\mathbf{N}}$ for every $1 \le i \le n$. This completes the proof.

4.7 Approximation

As a result of the limited precision of computers, we usually have to approximate real numbers (to rational numbers, for example). As a consequence, fuzzy regions and fuzzy constraints are also approximated. An important question then is *how the consistency and solutions change when we approximate fuzzy constraints.*

When reasoning with the fuzzy RCC, Schockaert et al. [96] assume that all bounds of the fuzzy RCC-8 constraints are taken from a finite $\Sigma_k = \{0, 1/k, 2/k, \ldots, (k-1)/k, 1\}$. Suppose Θ is an arbitrary set of fuzzy RCC-8 constraints. They do not explain why it is possible, or how one could transform Θ into another set of fuzzy RCC-8 constraints which take values in Σ_k for some $k \ge 1$, without changing the consistency of Θ. In this section, we justify the rationality of such an approximation.

We first define the distance function d between two fuzzy sets $\mathbf{r}, \mathbf{s}$ on X as

$$d(\mathbf{r}, \mathbf{s}) = \sup\{|\mathbf{r}(x) - \mathbf{s}(x)| : x \in X\}. \tag{4.59}$$

Using this distance function, it is possible to say that two fuzzy sets (fuzzy regions, fuzzy constraints) are *similar* if their distance is very small. We say two sets of fuzzy RCC constraints are similar if the distance between any two corresponding constraints is similar.

The following proposition is straightforward to prove.

Proposition 4.7 *The distance function d defined above satisfies*

- $d(\boldsymbol{r},\boldsymbol{s}) + d(\boldsymbol{s},\boldsymbol{t}) \geq d(\boldsymbol{r},\boldsymbol{t})$,
- $d(\boldsymbol{r} \vee \boldsymbol{s}, \boldsymbol{r}' \vee \boldsymbol{s}') \leq \max\{d(\boldsymbol{r},\boldsymbol{r}'), d(\boldsymbol{s},\boldsymbol{s}')\}$,
- $d(\chi_r \wedge \lambda, \chi_r \wedge \gamma) = |\lambda - \gamma|$,
- $d(\mathsf{sub}(\boldsymbol{r},\lambda), \mathsf{sub}(\boldsymbol{s},\gamma)) \leq d(\boldsymbol{r},\boldsymbol{s}) + |\lambda - \gamma|$,
- $d(\mathsf{expand}(\boldsymbol{r}), \mathsf{expand}(\boldsymbol{s})) = d(\boldsymbol{r},\boldsymbol{s})$.

The following result shows that the fuzzy RCC relation changes continuously as we approximate the fuzzy regions.

Proposition 4.8 *Let $\boldsymbol{r}, \boldsymbol{r}', \boldsymbol{s}$ and $\boldsymbol{s}'$ be fuzzy regions. Suppose $d(\boldsymbol{r},\boldsymbol{r}') \leq \epsilon$ and $d(\boldsymbol{s},\boldsymbol{s}') \leq \epsilon$. Then $|\boldsymbol{R}(\boldsymbol{r},\boldsymbol{s}) - \boldsymbol{R}(\boldsymbol{r}',\boldsymbol{s}')| \leq 2\epsilon$ for any $\boldsymbol{R}$ in $\{\mathbf{C},\mathbf{O},\mathbf{P},\mathbf{N}\}$.*

Proof By $d(\mathbf{r},\mathbf{r}') \leq \epsilon$, we have $\mathbf{r}'_{\alpha+\epsilon} \subseteq \mathbf{r}_\alpha \subseteq \mathbf{r}'_{\alpha-\epsilon}$. The proposition follows from Definition 4.2 and Theorem 4.1.

The following result shows that if two networks of fuzzy RCC constraints are similar, then their solutions constructed as in Theorem 4.3 are also similar. Note that the set Ω^0 of base regions selected in Eq. 4.34 is irrelevant to the particular normalised network Γ. It is only related to the number of spatial variables in Γ.

Proposition 4.9 *Let*

$$\begin{aligned} \Gamma &= \{\boldsymbol{R}(v_i,v_j) = \lambda_{ij}^{\boldsymbol{R}} : \boldsymbol{R} \in \{\mathbf{C},\mathbf{O},\mathbf{P},\mathbf{N}\}\}_{i,j=1}^n \\ \Gamma' &= \{\boldsymbol{R}(v_i,v_j) = \rho_{ij}^{\boldsymbol{R}} : \boldsymbol{R} \in \{\mathbf{C},\mathbf{O},\mathbf{P},\mathbf{N}\}\}_{i,j=1}^n \end{aligned}$$

be two consistent normalised fuzzy RCC networks, where $|\lambda_{ij}^{\boldsymbol{R}} - \rho_{ij}^{\boldsymbol{R}}| \le \epsilon$ *for any* $\boldsymbol{R} \in \{\mathbf{C}, \mathbf{O}, \mathbf{P}, \mathbf{N}\}$ *and* $1 \le i, j \le n$*. Let* L *and* L' *be the ntpp-shell numbers of* Γ *and* Γ' *respectively. Let* $\boldsymbol{a}_i^{\mathbf{N},L}$ *and* $\boldsymbol{b}_i^{\mathbf{N},L'}$ *be the solutions of* Γ *and* Γ' *constructed as in Theorem 4.3. Then we have*

$$d(\boldsymbol{a}_i^{\mathbf{N},L}, \boldsymbol{b}_i^{\mathbf{N},L'}) \le (\max\{L, L'\} + 2)\epsilon. \tag{4.60}$$

Proof By Proposition 4.7, we have $d(\mathbf{a}_i^{\mathbf{N},0}, \mathbf{b}^{\mathbf{N},0}) \le 2\epsilon$. Suppose $L \le L'$. For $h = L, \cdots, L' - 1$, let $\mathbf{a}_i^{\mathbf{N},h+1}$ be the fuzzy regions defined by Eq. 4.46. It can be proved that $d(\mathbf{a}_i^{\mathbf{N},k}, \mathbf{b}^{\mathbf{N},k}) \le (k+2)\epsilon$ for any $0 \le k \le L'$, and $\mathbf{a}_i^{\mathbf{N},L} = \mathbf{a}_i^{\mathbf{N},L+1} = \cdots = \mathbf{a}_i^{\mathbf{N},L'}$. Therefore $d(\mathbf{a}_i^{\mathbf{N},L}, \mathbf{b}_i^{\mathbf{N},L'}) \le (L' + 2)\epsilon$.

Suppose Γ is a normalised network and all bounds of the constraints in Γ are taken from Σ. Give a deviation tolerance bound $\epsilon > 0$, we now discuss whether it is possible to approximate Γ with a network Γ' such that

- all bounds of constraints in Γ' are taken from Σ_k for some k;
- Γ' is also consistent;
- the difference of the solutions of Γ and Γ' constructed in Theorem 4.3 is no more than ϵ.

A natural idea is to replace each bound α in Σ with $\lfloor k\alpha \rfloor / k$ to get a new normalised network Γ'. The difference of the two networks is clearly no more than $1/k$, and the solutions (if they exist) will differ at most $(L+2)/k$. However, we cannot assure that the new normalised network is also consistent. For example, if $T(\lambda_{ij}^{\mathbf{P}}, \lambda_{jk}^{\mathbf{P}})$ is exactly $\lambda_{ik}^{\mathbf{P}}$ for some i, j, k in Γ, this Γ' may possible violate the transitivity rule $T(\lambda_{ij}^{\mathbf{P}}, \lambda_{jk}^{\mathbf{P}}) \le \lambda_{ik}^{\mathbf{P}}$.

Suppose $\Sigma = \{\alpha_i\}_{i=0}^{l}$, where

$$0 = \alpha_0 < \alpha_1 < \cdots < \alpha_{l-1} < \alpha_l = 1.$$

Choose $0 < \delta < \alpha_1$. Let $\alpha_i' = \alpha_i - \delta$ for $i = 1, 2, \cdots, l-1$ and $\alpha_0' = 0$, $\alpha_l' = 1$. Then we have

- $\alpha_i' < \alpha_j'$ iff $i < j$;
- If $T(\alpha_i, \alpha_j) \le \alpha_k$, then $T(\alpha_i', \alpha_j') \le \alpha_k'$. In particular, if $\alpha_k \ne 0, 1$, then $T(\alpha_i', \alpha_j') + \delta \le \alpha_k'$

By Theorem 4.2, we can replace each α_i with α_i' without jeopardizing the consistency. Therefore, the consistency is not violated if we further change α_i' into $\alpha_i'' = \lfloor k\alpha_i' \rfloor / k \in \Sigma_k$ for any $k > 1/\delta$.

It is straightforward to see that

$$|\alpha_i - \alpha_i''| \leq |\alpha_i - \alpha_i'| + |\alpha_i' - \alpha_i''| < \delta + 1/k < 2\delta.$$

This means that the difference of the solutions is no more than $2(L+2)\delta$. In other words, if the deviation tolerance bound is ϵ, then we can safely approximate Γ with a normalised network Γ' bounds of which are all taken from Σ_k for

$$k > \max\{1/\alpha_1, 2(L+2)/\epsilon\}.$$

4.8 Chapter Summary

This chapter defined the standard fuzzy RCC models for the fuzzy region connection calculus proposed by Schockaert et al., and proved that each standard model is canonical in the sense that any consistent network of fuzzy RCC-8 constraints has a solution in it. We also proved that the consistency of a normalised fuzzy RCC-8 network can be determined by checking a set of three basic conditions and six transitivity rules. This implies that composition-based reasoning is also complete for solving fuzzy RCC-8 constraints in any standard fuzzy RCC model. For a consistent normalised fuzzy RCC-8 network, a solution can be constructed in polynomial time. Our construction is robust in the sense that similar normalised networks may have similar solutions if both normalised networks are consistent. Roughly speaking, the $(n;\alpha,0)$-model of Schockaert et al. [96] can be taken as a special case of our standard fuzzy RCC model.

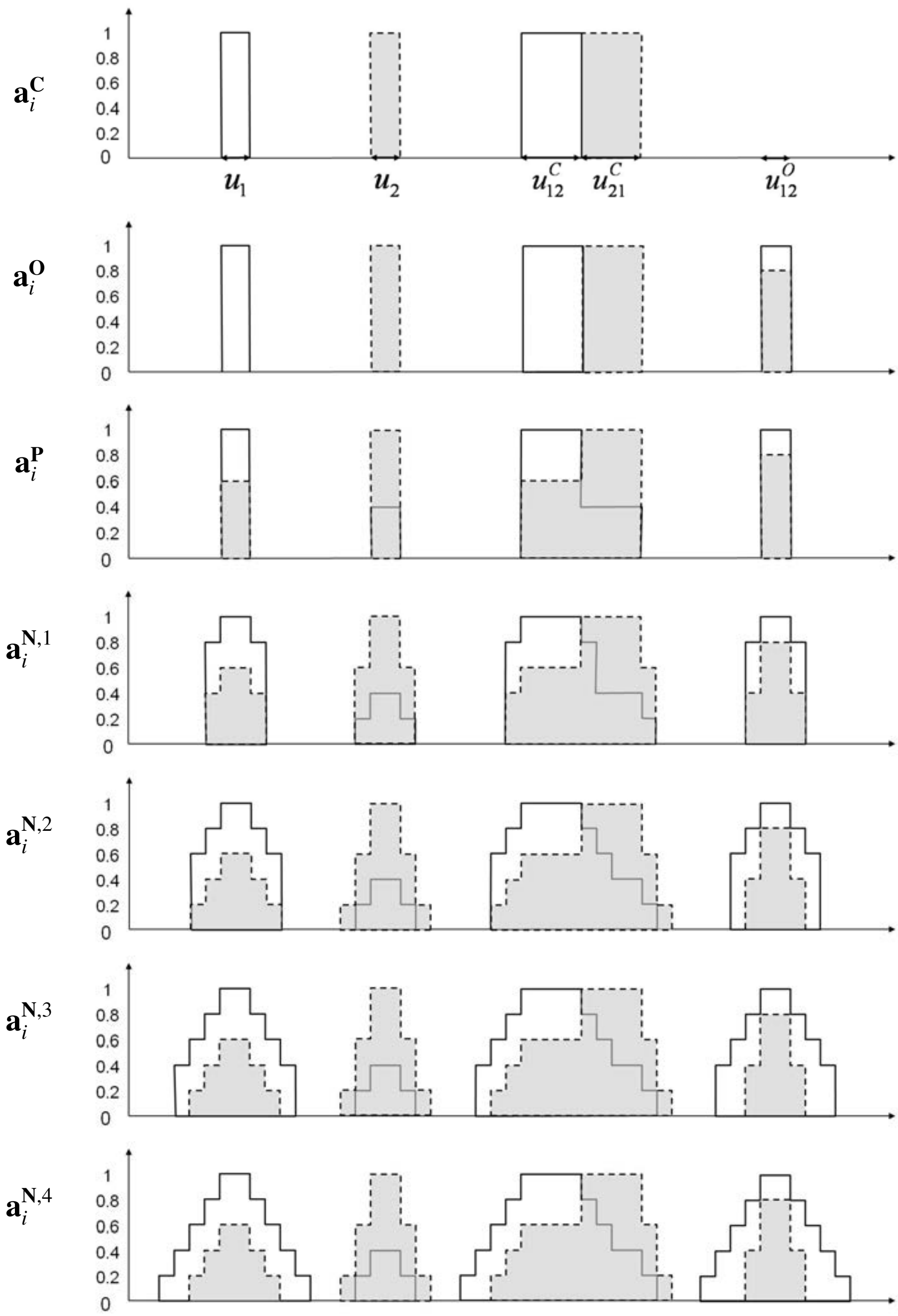


Figure 4.1: Illustration of the construction procedure of network $\{\lambda_{12}^{\mathbf{C}} = 1, \lambda_{12}^{\mathbf{O}} = 0.8, \lambda_{12}^{\mathbf{P}} = 0.6, \lambda_{21}^{\mathbf{P}} = 0.4, \lambda_{11}^{\mathbf{N}} = 0.8, \lambda_{12}^{\mathbf{N}} = 0.4, \lambda_{21}^{\mathbf{N}} = 0, \lambda_{22}^{\mathbf{N}} = 0.6\}$. The white region stands for $\mathbf{a}_1$, and the shaded region stands for $\mathbf{a}_2$.

Input: $\Gamma = \{\lambda^{\mathbf{C}}_{ij}, \lambda^{\mathbf{O}}_{ij}, \lambda^{\mathbf{P}}_{ij}, \lambda^{\mathbf{P}}_{ij}\}_{i,j=1}^{n}$
Output: A solution of Γ

for $1 \leq i \leq n$ **do**
    $\mathbf{a}^{\mathbf{C}}_i \leftarrow \chi_{r_i}$;
    for $1 \leq j \leq n, j \neq i$ **do**
        if $j < i$ **then** $\mathbf{a}^{\mathbf{C}}_i \leftarrow \mathbf{a}^{\mathbf{C}}_i \vee (\chi_{r^{\mathbf{C}}_{ij}} \wedge \lambda^{\mathbf{C}}_{ij})$;
        else $\mathbf{a}^{\mathbf{C}}_i \leftarrow \mathbf{a}^{\mathbf{C}}_i \vee \chi_{r^{\mathbf{C}}_{ij}}$;
    end
end
for $1 \leq i \leq n$ **do**
    $\mathbf{a}^{\mathbf{O}}_i \leftarrow \mathbf{a}^{\mathbf{C}}_i$;
    for $1 \leq j \leq n, j \neq i$ **do**
        if $j < i$ **then** $\mathbf{a}^{\mathbf{O}}_i \leftarrow \mathbf{a}^{\mathbf{O}}_i \vee (\chi_{r^{\mathbf{O}}_{ij}} \wedge \lambda^{\mathbf{O}}_{ij})$;
        else $\mathbf{a}^{\mathbf{O}}_i \leftarrow \mathbf{a}^{\mathbf{O}}_i \vee \chi_{r^{\mathbf{O}}_{ij}}$;
    end
end
for $1 \leq i \leq n$ **do**
    $\mathbf{a}^{\mathbf{N},0}_i \leftarrow 0$;
    for $1 \leq j \leq n$ **do**
        $\mathbf{a}^{\mathbf{N},0}_i \leftarrow \mathbf{a}^{\mathbf{N},0}_i \vee \mathsf{sub}(\mathbf{a}^{\mathbf{O}}_j, 1 - \lambda^{\mathbf{P}}_{ji})$;
    end
end
$L \leftarrow n\lceil 1/(1 - \max\{\lambda^{\mathbf{N}}_{ij} : \lambda^{\mathbf{N}}_{ij} < 1\})\rceil$;
for $1 \leq h \leq L$ **do**
    for $1 \leq i \leq n$ **do**
        for $1 \leq j \leq n$ **do**
            $\mathbf{a}^{\mathbf{N},h+1}_i \leftarrow \mathbf{a}^{\mathbf{N},h}_i \vee \mathsf{sub}(\mathsf{expand}(\mathbf{a}^{\mathbf{N},h}_j), 1 - \lambda^{\mathbf{N}}_{ji})$;
        end
    end
end
return $\{\mathbf{a}^{\mathbf{N},L}_i\}_{i=1}^{n}$.

Algorithm 2: Generating a solution of a satisfiable fuzzy network Γ

Chapter 5

Reasoning in the Cardinal Direction Calculus

5.1 Introduction

Direction relations between extended spatial objects are an important kind of common-sense knowledge. Most existing direction relation models approximate a spatial object by a point (e.g., its centroid) or a box. This is certainly imprecise in real-world applications, such as describing the directional information between two countries, say, Portugal and Spain [100].

Goyal and Egenhofer [46, 47] proposed the *direction relation matrix* (DRM) for representing direction relations between *connected* plane regions. The original description of the DRM lacks formality and does not consider limit cases. This problem was fixed in [100], in which the model is called the *cardinal direction calculus* (CDC) [1]. When representing the direction of the primary object to a reference object, the CDC approximates the reference object by a box while leaving the primary object unaltered; therefore, the exact geometry of the primary object is considered in the representation of the direction. The CDC has 218 basic relations [47], each of which represents certain *definite* directional information between objects. We write $\mathcal{B}_{\mathbf{CDC}}$ for the set of basic CDC relations; additionally, we write $\mathcal{B}^{*}_{\mathbf{CDC}}$ for the set of all CDC relations, and $\mathcal{B}^{\star}_{\mathbf{CDC}}$ for $\mathcal{B}_{\mathbf{CDC}} \cup \{?\}$, where ? is the universal relation.

[1]A number of cardinal relation models including CDC have been discussed and compared in [93].

As introduced in Chapter 2, the consistency problem of CDC is to decide the consistency of a given CDC constraint network

$$\Gamma = \{v_i \delta_{ij} v_j\}_{i,j=1}^{n} \quad \text{(each } \delta_{ij} \text{ is a CDC relation)}$$

over n spatial variables $v_1, \cdots, v_n$: we say Γ is *consistent* if there exist n *connected* plane regions $a_1, \cdots, a_n$ such that (a_i, a_j) is an instance of δ_{ij} for any $1 \leq i, j \leq n$. In particular, if the relations δ_{ij} are all from a subclass $\mathcal{S}$ of $\mathcal{B}^*_{\mathbf{CDC}}$, we write $\textsc{cspsat}_{\mathbf{CDC}}(\mathcal{S})$, or simply $\textsc{cspsat}(\mathcal{S})$, for the consistency problem restricted to $\mathcal{S}$. Then $\textsc{cspsat}(\mathcal{B}^*_{\mathbf{CDC}})$ is the general consistency problem of CDC, while $\textsc{cspsat}(\mathcal{B}_{\mathbf{CDC}})$ and $\textsc{cspsat}(\mathcal{B}^{\star}_{\mathbf{CDC}})$ denote the consistency problems restricted to respectively *complete* basic CDC networks and *incomplete* basic CDC networks. Note that we call an instance in $\textsc{cspsat}(\mathcal{B}^{\star}_{\mathbf{CDC}})$ an *incomplete* network, as two variables related by the universal relation can be regarded as not directly constrained.

It is clear that the complexity of $\textsc{cspsat}(\mathcal{B}_{\mathbf{CDC}})$, $\textsc{cspsat}(\mathcal{B}^{\star}_{\mathbf{CDC}})$ and $\textsc{cspsat}(\mathcal{B}^*_{\mathbf{CDC}})$ is increasing. Therefore, complete basic constraint networks are a good starting point for the investigation of the consistency problem. For basic IA and RCC-8 networks, it has been shown that path-consistency is sufficient to decide the consistency, hence the consistency problem can be decided in polynomial time. For basic complete CDC networks, however, examples (such as [100, Example 9] and [67, Example 4]) show that local consistency, in particular path-consistency, fails to imply consistency. This suggests that reasoning with CDC requires a novel approach or algorithm.

This chapter will first solve the $\textsc{cspsat}(\mathcal{B}_{\mathbf{CDC}})$ by providing a cubic algorithm for checking the consistency of basic complete CDC constraint networks, and will prove that reasoning with the CDC in general is an NP-Complete problem. This means that $\textsc{cspsat}_{\mathbf{CDC}}(\mathcal{B}_{\mathbf{CDC}})$ is tractable but $\textsc{cspsat}(\mathcal{B}^*_{\mathbf{CDC}})$ is not. Although it would be natural to conjecture that the tractability of $\mathcal{B}_{\mathbf{CDC}}$ implies that $\mathcal{B}^{\star}_{\mathbf{CDC}}$ is also tractable (as it is the case for IA and RCC-8), however, this chapter will prove the NP-hardness of $\textsc{cspsat}(\mathcal{B}^{\star}_{\mathbf{CDC}})$ [1]. The CDC is the first qualitative calculus in which reasoning with complete and incomplete basic networks has different complexity.

The remainder of this chapter proceeds as follows. Section 2 introduces the Car-

[1]Note that the universal relation is the weak composition of two basic CDC relations [67]. This suggests that the propagation technique used in [77, 86, 56, 85] fails to find the maximal tractable subclasses of the CDC.

dinal Direction Calculus and establishes its relation with Interval Algebra. Section 3 proposes the concept of a *maximal canonical solution* and proves that a consistent complete basic CDC network has a unique maximal canonical solution. Based on this concept, Section 4 provides a $O(n^4)$ algorithm for $\textsc{cspsat}(\mathcal{B}_{\mathbf{CDC}})$ and refines it to the $O(n^3)$. Section 5 introduces a number of non-CDC relations that can be defined in CDC. The NP-hardness result and related discussion are given in Section 6. Section 7 addresses a variant of CDC in which regions are allowed to be disconnected. The last section concludes the chapter. We note that a preliminary version of the cubic algorithm for solving $\textsc{cspsat}(\mathcal{B}_{\mathbf{CDC}})$ was first reported in [116], and then significantly extended in [67]. The NP-hardness result was obtained in [65].

Table 5.1 summarises major and special notations used in this chapter.

5.2 CDC and Projective Interval Relations

This section aims to establish the connection between CDC networks and IA networks. We first give a brief review of the definition of CDC, which has been defined in Section 2.3.5.

The universe of the CDC is the set of bounded connected plane regions. Note that such regions may have holes. The minimum bounding rectangle (mbr) of a region is the smallest rectangle which contains the region and has sides parallel to the axes. Given a bounded region b and its mbr $\mathbf{mbr}(b)$ (see Definition 2.8), we may partition the plane into nine tiles, denoted as $NW(b), N(b), NE(b), W(b), O(b), E(b), SW(b), S(b), SE(b)$, by extending the four edges of $\mathbf{mbr}(b)$. The following figure, which is a duplication of Figure 2.3, provides illustration.

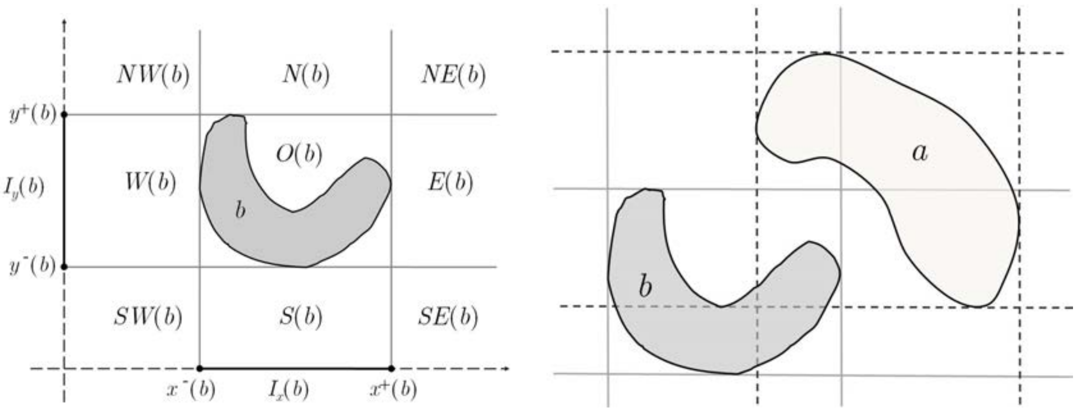
Suppose bounded connected regions a, b are respectively the primary object and the reference object. Then the *cardinal direction* of a to b is defined as the following Boolean matrix,

$$\mathsf{dir}(a,b) = \begin{bmatrix} d^{NW} & d^{N} & d^{NE} \\ d^{W} & d^{O} & d^{E} \\ d^{SW} & d^{S} & d^{SE} \end{bmatrix}, \tag{5.1}$$

where for each tile name $\chi \in \{NW, N, NE, W, O, E, SW, S, SE\}$, $d^{\chi} = 1$ iff $a^{\circ} \cap \chi(b) \neq \varnothing$.

Notation	Meaning
a, b, c	regions
$I_x(a), I_y(a)$	the x- and y-projections of a
$\mathbf{mbr}(a)$	the minimum bounding rectangle (mbr) of a
α, β	basic IA relations
$\alpha \otimes \beta$	a basic RA relation
δ, γ, δ_{ij}	basic CDC relations
$\mathcal{B}_{\mathbf{CDC}}$, $\mathcal{B}^{*}_{\mathbf{CDC}}$, $\mathcal{B}^{\star}_{\mathbf{CDC}}$	the set of basic CDC relations, all CDC relations, $\mathcal{B}_{\mathbf{CDC}} \cup \{?\}$
$\delta_1{:}\cdots{:}\delta_k$	a concrete basic CDC relation
$\mathsf{dir}(a, b)$	the cardinal direction of a to b
$\iota^x(\delta)$, $\iota^y(\delta)$	the x- and y-projective interval relations of δ
$\iota^x(\delta, \gamma)$, ρ^x_{ij}	$\iota^x(\delta) \cap \iota^x(\gamma)^\smallsmile$, $\iota^x(\delta_{ij}, \delta_{ji}) \setminus (\mathsf{m} \cup \mathsf{mi})$
Γ	(complete) basic CDC network
Γ_x, Γ_y	the x- and y-projective IA networks of Γ
w_i	spatial variables
p, p_{ij}	pixels
$\mathfrak{a}$	a set of (connected) regions
$S(\mathfrak{a})$, c_{ij}, $C(\mathfrak{a})$	the frame of $\mathfrak{a}$, a cell of $\mathfrak{a}$, the cell set of $\mathfrak{a}$
$\mathfrak{a}^r$	the regularisation of $\mathfrak{a}$
$\mathfrak{a}^+$	the digital solution obtained from $\mathfrak{a}$
$\{I_i\}_{i=1}^n$, $\{J_i\}_{i=1}^n$	the canonical interval solutions of Γ_x and Γ_y
m_i	$I_i \times J_i$
D_i	the disallowed pixels for variable v_i
b_i	the upper bound region for variable v_i
c_i	the maximal connected component of b_i with the mbr m_i
$\Vert$	the right-side parallel relation with gap (*parallel relation* for short)
$\ulcorner$	the upper left corner (ULC) relation
$\Gamma_{\Vert}$, $\Gamma_{\ulcorner}$, $\Gamma_{\alpha\otimes\beta}$	the basic CDC networks that define $\Vert$, $\ulcorner$ and a subset of $\alpha \otimes \beta$
ϕ, c, p, p^*	3-SAT instance, clause, propositional variable, literal
Γ_p, Γ_V, Γ_c, Γ_ϕ	the basic CDC networks for p, V, c and ϕ
$f_p, f_{\neg p}, f^0_p$	the frame spatial variables for propositional variable p
$u_p, u_{\neg p}$	the dual spatial variables for propositional variable p
v_c	the spatial variable for propositional clause c
$w^c_0, w^c_{rs}, w^c_{st}, w^c_1$	the 'pier' spatial variables for propositional clause c
u^*_r	the spatial variable corresponding to p^*_r (it is either u_r or $u_{\neg r}$)
X_c	the spatial variable set $\{w^c_0, u^*_r, w^c_{rs}, u^*_s, w^c_{st}, u^*_t, w^c_1\}$

Table 5.1: Notation.



(a) region b and its 9-tiles (b) a pair of regions a and b.

The matrix can be compactly and equivalently represented in the form $\delta_1{:}\delta_2{:}\cdots{:}\delta_k$, where $\{\delta_1, \delta_2, \cdots, \delta_k\}$ is the set of tile names χ such that $d^\chi = 1$. For example, in the above figure $\mathsf{dir}(a,b) = N{:}NE{:}E$ and $\mathsf{dir}(b,a) = W{:}O{:}SW{:}S$.

We call a 3×3 Boolean matrix a *valid matrix* if it is the cardinal direction of two regions. Each valid matrix corresponds to a basic CDC relation. Therefore, we do not distinguish a valid matrix and a basic CDC relation. There are altogether 218 cardinal direction relations [47] (between connected regions). Recall that we use $\mathcal{B}_{\mathbf{CDC}}$ to denote the set of all 218 basic relations of CDC. It is worth mentioning that the CDC is not closed under converse.

We now establish the connection between CDC and IA. Observing that the CDC relation of a to b and that of b to a can (almost) determine the IA relation between $I_x(a)$ and $I_x(b)$, we introduce the notion of *projective interval relations*. Note that interval $I_x(r)$ is the x-projection of region r, defined in Definition 2.8.

Definition 5.1 (projective interval relation) *Let δ be a basic CDC relation. The x-*projective interval relation *of δ is a binary relation on intervals, defined as*

$$\iota^x(\delta) := \{(I_x(a), I_x(b)) : (a,b) \in \delta\}. \tag{5.2}$$

Furthermore, for basic CDC relations δ and γ, we define $\iota^x(\delta,\gamma)$ as $\iota^x(\delta) \cap \iota^x(\gamma)^\sim$.

The following proposition confirms that $\iota^x(\delta)$ is indeed a relation in IA, and provides a procedure for calculating the x-projective interval relation of a given basic CDC relation. The y-projective interval relation has similar properties.

Proposition 5.1 *Suppose δ is a basic CDC relation which as the matrix representation as in Equation 5.1. Define*

$$
\begin{aligned}
d_1 &:= \max\{d^{NW}, d^{W}, d^{SW}\},\\
d_2 &:= \max\{d^{N}, d^{O}, d^{S}\},\\
d_3 &:= \max\{d^{NE}, d^{E}, d^{SE}\}.
\end{aligned}
$$

Then the x-projective interval relation $\iota^x(\delta)$ is one of the following IA relations

$$\mathsf{p} \cup \mathsf{m}, \mathsf{s} \cup \mathsf{d} \cup \mathsf{f} \cup \mathsf{eq}, \mathsf{pi} \cup \mathsf{mi}, \mathsf{o} \cup \mathsf{fi}, \mathsf{oi} \cup \mathsf{si}, \mathsf{di}, \tag{5.3}$$

decided by the following table.

(d_1, d_2, d_3)	(1,0,0)	(0,1,0)	(0,0,1)	(1,1,0)	(0,1,1)	(1,1,1)
$\iota^x(\delta)$	p ∪ m	s ∪ d ∪ f ∪ eq	pi ∪ mi	o ∪ fi	oi ∪ si	di

Table 5.2: Computing the x-projective IA relations of a basic CDC relation

sketch Suppose α is a non-basic IA relation that corresponds to basic CDC relation δ as in the statement of the proposition. We need to prove (i) if $\mathsf{dir}(a,b) = \delta$, then $I_x(a)\alpha I_x(b)$; (ii) for any interval J_1 and J_2 such that $J_1\alpha J_2$, there exists two regions such that $\mathsf{dir}(a,b) = \delta$ and $I_x(a) = J_1$, $I_x(b) = J_2$.

The first part can be proved following the definition of cardinal direction. For example, if $d_1 = 0$, i.e., $d^{NW} = d^{W} = d^{SW} = 0$, we know that $a^\circ \cap \chi(b) = 0$, where $\chi \in \{NW, W, SW\}$. This implies $a \cap (-\infty, x^-(b)) \times (-\infty, +\infty) = \varnothing$, i.e., $x^-(a) \geq x^-(b)$, which rules out the IA relations $\mathsf{p}, \mathsf{m}, \mathsf{o}, \mathsf{fi}, \mathsf{di}$ of $I_x(a)$ and $I_x(b)$.

The second part can be proved by case analysis. Suppose the endpoints of J_1 and J_2 are $\{x_0, x_1, x_2, x_3\}$ (some of which may be identical), then we can compose rectangles of the form $[x_i, x_j] \times [k, l]$ to obtain the target regions, where $i, j, k, l \in \{0, 1, 2, 3\}$. For example, consider basic CDC relation $\delta = NW : N : O : E$, for which $(d_1, d_2, d_3) = (1, 1, 1)$. We aim to prove that for any intervals $J_1 = (x_0, x_1)$ and $J_2 = (x_2, x_3)$ with $J_1 \mathsf{di} J_2$, there exists connected regions a and b such that $\mathsf{dir}(a,b) = \delta$ and $I_x(a) = J_1$, $I_x(b) = J_2$. Note that $J_1 \mathsf{di} J_2$ implies that $x_0 < x_2 < x_3 < x_1$. In such case, regions $a = ([x_0, x_3] \times [2, 3]) \cup ([x_2, x_1] \times [1, 2])$ and $b = [x_2, x_3] \times [1, 2]$ satisfy the requirement.

The IA relation between $I_x(a)$ and $I_x(b)$ can be further refined if we know both the cardinal relation of a to b and that of b to a.

Proposition 5.2 *For a pair of basic CDC relations (δ, γ), $\iota^x(\delta, \gamma)$ is either empty or an IA relation in $\mathcal{A}_{int}$, where*

$$\mathcal{A}_{int} = \{\mathsf{o}, \mathsf{s}, \mathsf{d}, \mathsf{f}, \mathsf{eq}, \mathsf{fi}, \mathsf{di}, \mathsf{si}, \mathsf{oi}\} \cup \{\mathsf{p} \cup \mathsf{m}, \mathsf{pi} \cup \mathsf{mi}\}. \tag{5.4}$$

Furthermore, if regions a and b satisfy $\mathsf{dir}(a, b) = \delta$ and $\mathsf{dir}(b, a) = \gamma$, then $\iota^x(\delta, \gamma)$ is not empty and $I_x(a)\iota^x(\delta, \gamma)I_x(b)$.

Proof Proposition 5.1 has proved that $\iota^x(\delta)$ ($\iota^x(\gamma)$) is one of the six IA relations in $\{\mathsf{p}\cup\mathsf{m}, \mathsf{s}\cup\mathsf{d}\cup\mathsf{f}\cup\mathsf{eq}, \mathsf{pi}\cup\mathsf{mi}, \mathsf{o}\cup\mathsf{fi}, \mathsf{oi}\cup\mathsf{si}, \mathsf{di}\}$. Therefore, the first part of this proposition can be proved by enumerating all the $6 \times 6 = 36$ possible cases of $\iota^x(\delta)$ and $\iota^x(\gamma)$. The second part follows directly the definition of the x-projective interval relation.

The x-projective IA network is obtained from a complete basic CDC network $\Gamma = \{v_i\delta_{ij}v_j\}_{i,j=1}^n$ by replacing each δ_{ij} with $\iota^x(\delta_{ij}, \delta_{ji})$, and then further refining each non-basic IA constraint (viz, $\mathsf{p} \cup \mathsf{m}$ or $\mathsf{pi} \cup \mathsf{mi}$) to basic IA constraint by excluding m and mi.

Definition 5.2 (projective IA networks) *Let $\Gamma = \{v_i\delta_{ij}v_j\}_{i,j=1}^n$ be a complete basic CDC constraint network. For each i, j, let ρ_{ij}^x and ρ_{ij}^y be basic IA relations defined as follows,*

$$\rho_{ij}^x = \iota^x(\delta_{ij}, \delta_{ji}) \setminus (\mathsf{m} \cup \mathsf{mi}), \tag{5.5}$$

$$\rho_{ij}^y = \iota^y(\delta_{ij}, \delta_{ji}) \setminus (\mathsf{m} \cup \mathsf{mi}). \tag{5.6}$$

*We call $\Gamma_x = \{v_i\rho_{ij}^x v_j\}_{i,j=1}^n$ and $\Gamma_y = \{v_i\rho_{ij}^y v_j\}_{i,j=1}^n$ the x- and y-*projective IA networks *of Γ respectively.*

We have the following theorem characterising the relationship between a basic CDC constraint network and its projective IA networks.

Theorem 5.1 *A basic CDC network of constraints $\Gamma = \{v_i\delta_{ij}v_j\}_{i,j=1}^n$ is consistent only if the projective IA networks $\Gamma_x = \{v_i\rho_{ij}^x v_j\}_{i,j=1}^n$ and $\Gamma_y = \{v_i\rho_{ij}^y v_j\}_{i,j=1}^n$ are consistent.*

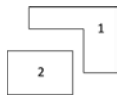
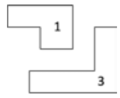
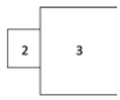
(i,j)	δ_{ij}	δ_{ji}	illus.	ρ^x_{ij}	ρ^y_{ij}
(1,2)	$\begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix}$		oi	oi
(1,3)	$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$		o	oi
(2,3)	$\begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$		p	d

Figure 5.1: A complete basic CDC network and its projective IA basic networks

Proof Let Θ_x be the non-basic IA constraint network $\{v_i \iota^x_{ij} v_j\}^n_{i,j=1}$, where ι^x_{ij} is defined as $\iota^x(\delta_{ij}, \delta_{ji})$. Suppose $\{a_i\}^n_{i=1}$ is a solution of Γ, then Proposition 5.2 shows that $\{I_x(a_i)\}^n_{i=1}$ is a solution of Θ_x, which implies the consistency of Θ_x. Therefore, we only need to prove that the consistency of IA constraint network Θ_x implies the consistency of basic IA constraint network Γ_x.

Suppose $\{I_i = [u^-_i, u^+_i]\}^n_{i=1}$ is a solution of Θ_x. We now construct a solution of Γ_x. We call a point $u \in M = \{u^-_i, u^+_i\}^n_{i=1}$ a *meet point* if $u = u^+_i = u^-_j$, for some i, j. That is to say, if I_i meets I_j then the larger end point of I_i is a meet point. Clearly, if $\{I_i\}^n_{i=1}$ contains no meet point, then it is also a solution of Γ_x. Now suppose u is a meet point, and we show that u can be eliminated by slightly modifying $\{I_i\}^n_{i=1}$ as follows. For each interval $I_i = [u^-_i, u^+_i]$ where u^-_i is equal to the meet point u, we replace it by the interval $[u + \epsilon, u^+_i]$, where ϵ is a sufficient small positive number (e.g., $\epsilon = \min\{|w - w'| : w \neq w', w, w' \in M\}/2$). Noticing that Θ_x is an IA constraint network over $\mathcal{A}_{int}$, we can straightforwardly verify by case analysis that the intervals are still a solution of Θ_x after such modification. Repeating this procedure until no meet point is left, we get a solution of Γ_x. Therefore, the consistency of Θ_x implies the consistency of Γ_x.

Example 5.1 *Fig. 5.1 specifies a complete basic CDC network $\Gamma = \{v_i \delta_{ij} v_j\}^3_{i,j=1}$, and gives its projective IA networks $\Gamma_x = \{v_i \rho^x_{ij} v_j\}^3_{i,j=1}$ and $\Gamma_y = \{v_i \rho^y_{ij} v_j\}^3_{i,j=1}$. For each pair* *of $i \neq j$, a solution of $\{v_i \delta_{ij} v_j, v_j \delta_{ji} v_i\}$ is illustrated.*

5.3 Maximal Canonical Solutions of CDC Networks

Based on the results established in the previous section, this section will prove that if a complete basic CDC constraint network Γ is consistent, then it has a unique solution which is called the *maximal canonical solution*. The x-projections of maximal canonical solution will be a solution of Γ_x, the x-projective IA network of Γ. The concept of a maximal canonical solution will be exploited in the next section to devise a polynomial algorithm for solving the consistency problem.

First we introduce the notion of a *canonical solution* of complete basic IA networks.

Definition 5.3 (canonical set of intervals) *Suppose $\mathfrak{I} = \{[x_i^-, x_i^+]\}_{i=1}^n$ is a set of intervals. Let $E(\mathfrak{I}) = \bigcup_{i=1}^n \{x_i^-, x_i^+\}$ be the set of endpoints of intervals in $\mathfrak{I}$. We say $\mathfrak{I}$ is a* canonical set of intervals *iff $E(\mathfrak{I}) = [0, M] \cap \mathbb{Z}$, where M is the largest number in $E(\mathfrak{I})$. A solution of a complete basic IA network is called a* canonical (interval) solution *if it is a canonical set of intervals.*

We have the following theorem claiming that every consistent basic IA constraint network has a unique canonical solution.

Theorem 5.2 *Suppose $\Gamma = \{v_i \lambda_{ij} v_j\}_{i,j=1}^n$ is a basic IA constraint network. If Γ is consistent, then it has a unique canonical interval solution.*

Proof Suppose $\mathfrak{I} = \{[x_i^-, x_i^+]\}_{i=1}^n$ is a solution of Γ. Write $\alpha_0 < \alpha_1 < \cdots < \alpha_{n^*}$ for the ordering of $E(\mathfrak{I})$. Define $h : E(\mathfrak{I}) \to \{0, 1, \cdots, n^*\}$ as $h(x) = k$ if $x = \alpha_k$. Because only the ordering of endpoints of intervals matters in a solution, $\mathfrak{h} = \{[h(x_i^-), h(x_i^+)]\}_{i=1}^n$ is also a solution of Γ. Moreover, it is clear that $E(\mathfrak{h}) = [0, n^*] \cap \mathbb{Z}$. Therefore, $\mathfrak{h}$ is a canonical interval solution of Γ. It remains to prove that all any canonical solutions $\mathfrak{I} = \{[x_i^-, x_i^+]\}_{i=1}^n$ and $\mathfrak{h} = \{[y_i^-, y_i^+]\}_{i=1}^n$ are identical. Note that the two canonical solution have the same orderings of the endpoints of the intervals as they are solutions of the same basic IA constraint network. Based on this observation, it is straightforward to prove that they are identical according to the definition of canonical solutions.

Note that the canonical solution of a consistent basic IA network with n variables can be constructed in $O(n^2)$ time by Algorithm 3.

Input: A complete basic consistent IA network $\Gamma = \{v_i \rho_{ij} v_j\}_{i,j=1}^n$.
Output: The unique canonical solution of Γ.

$V \leftarrow \{x_i^-, x_i^+\}_{i=1}^n$;
foreach $p, q \in V$ **do**
  calculate $Order[p,q]$ by constraints in Γ;
end
sort V by $Order$ ascending;
$k \leftarrow 0$, $p \leftarrow$ the smallest element in V, $r[p] \leftarrow k$;
while *p is not the greatest element in V* **do**
  $q \leftarrow$ the next element of p in V;
  if *$Order[q,p]$ is $>$* **then**
    $k \leftarrow k+1$;
  end
  $p \leftarrow q$, $r[p] \leftarrow k$;
end
return $\{[r[x_i^-], r[x_i^+]]\}_{i=1}^n$.

Algorithm 3: the canonical solution of a complete basic IA network

The regions in a canonical solution of basic CDC network are required to be digital regions in the following sense.

Definition 5.4 (pixel, digital region, digital solution) *A* pixel *is a rectangle $p_{ij} = [i, i+1] \times [j, j+1]$, where i, j are integers. A region a is* digital *if a is composed of pixels,* i.e. *$p_{ij} \cap a^\circ \neq \varnothing$ iff $p_{ij} \subseteq a$. A solution $\mathfrak{a} = \{a_i\}_{i=1}^n$ of a basic CDC network is* digital *if each a_i is a digital region.*

Fig. 5.2 illustrates a pixel p and a digital region a. We now can formally define the canonical solution of basic CDC networks.

Definition 5.5 (canonical solution) *Let Γ be a consistent basic CDC constraint network, and Γ_x, Γ_y be the x- and y-projective IA networks of Γ (cf. Dfn. 5.2) respectively. A solution $\mathfrak{a} = \{a_i\}_{i=1}^n$ of Γ is said to be* canonical *if (i) it is a digital solution, and (ii) $\{I_x(a_i)\}_{i=1}^n$ and $\{I_y(a_i)\}_{i=1}^n$ are the canonical solutions of Γ_x and Γ_y respectively.*

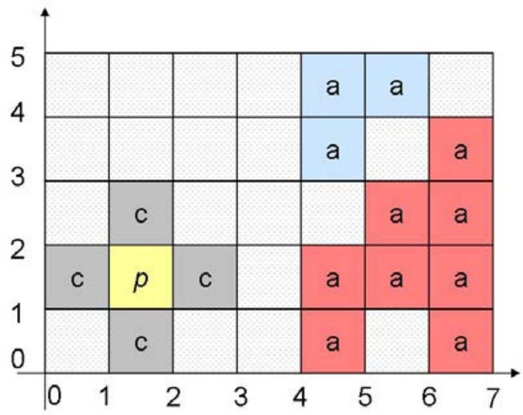


Figure 5.2: A pixel p and a digital region a with two pieces

In the following subsections, we show that each consistent CDC network has a maximal canonical solution by transforming an arbitrary solution into a maximal canonical one.

5.3.1 Regular Solutions

Suppose $\mathfrak{a} = \{a_i\}_{i=1}^n$ is a set of n bounded connected regions. We show $\mathfrak{a}$ can be *regularised* without changing any CDC relation between regions in $\mathfrak{a}$. Roughly speaking, regions in a regular solution are required to be composed of rectangles, rather than have arbitrary shapes.

Suppose $\mathbf{mbr}(a_i)$, the minimal bounding rectangle of a_i, is $[x_i^-, x_i^+] \times [y_i^-, y_i^+]$. Furthermore, suppose $\alpha_0 < \alpha_1 < \cdots < \alpha_{n_x}$ and $\beta_0 < \beta_1 < \cdots < \beta_{n_y}$ are respectively the orderings of $\{x_i^-, x_i^+ : 1 \leq i \leq n\}$ and $\{y_i^-, y_i^+ : 1 \leq i \leq n\}$., where $n_x + 1$ and $n_y + 1$ are the cardinalities of $\{x_i^-, x_i^+ : 1 \leq i \leq n\}$ and $\{y_i^-, y_i^+ : 1 \leq i \leq n\}$ respectively. Extending edges of each rectangle $\mathbf{mbr}(a_i)$ until meeting the boundary of the rectangle $[\alpha_0, \alpha_{n_x}] \times [\beta_0, \beta_{n_y}]$, we partition $[\alpha_0, \alpha_{n_x}] \times [\beta_0, \beta_{n_y}]$ into cells. We next show that these cells can be used to compose another set of regions that have the same CDC relations as $\mathfrak{a}$.

Definition 5.6 (frame, cell set, regularisation) *Suppose* $\mathfrak{a} = \{a_i\}_{i=1}^n$ *is a set of bounded*

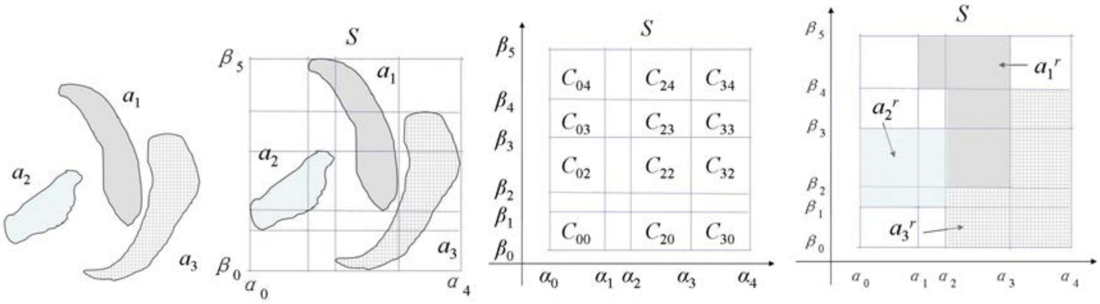


Figure 5.3: Illustration of regularisation

connected regions. Denote

$$S(\mathfrak{a}) := [\alpha_0, \alpha_{n_x}] \times [\beta_0, \beta_{n_y}], \tag{5.7}$$

$$c_{ij} := [\alpha_i, \alpha_{i+1}] \times [\beta_j, \beta_{j+1}] \ \ (0 \le i < n_x, 0 \le j < n_y), \tag{5.8}$$

$$C(\mathfrak{a}) := \{c_{ij} : 0 \le i < n_x, 0 \le j < n_y\}, \tag{5.9}$$

$$a_i^r := \bigcup \{c \in C(\mathfrak{a}) : c \cap a_i^\circ \ne \varnothing\} \ (i = 1, \cdots, n), \tag{5.10}$$

$$\mathfrak{a}^r := \{a_i^r : 1 \le i \le n\}. \tag{5.11}$$

We call $S(\mathfrak{a})$ *the* frame *of* $\mathfrak{a}$*, call* c_{ij} *a* cell *of* $\mathfrak{a}$*, and call* $C(\mathfrak{a})$ *the* cell set *of* $\mathfrak{a}$*, and call* $\mathfrak{a}^r$ *the* regularisation *of* $\mathfrak{a}$ *(see Fig. 5.3).*

Proposition 5.3 *Suppose* $\mathfrak{a} = \{a_i\}_{i=1}^n$ *is a set of connected regions and* $\mathfrak{a}^r = \{a_i^r\}_{i=1}^n$ *is the regularisation of* $\mathfrak{a}$*. Then we have*

- $a_i \subseteq a_i^r$ *and* $\mathbf{mbr}(a_i) = \mathbf{mbr}(a_i^r)$ *for each i.*
- $S(\mathfrak{a}) = S(\mathfrak{a}^r)$ *and* $C(\mathfrak{a}) = C(\mathfrak{a}^r)$.
- *The CDC relation of* a_i^r *to* a_j^r *is the same as that of* a_i *to* a_j.

Proof It is clear that each a_i^r is connected. For each pair of i, j, to show $\mathsf{dir}(a_i, a_j) = \mathsf{dir}(a_i^r, a_j^r)$, it suffices to show for each tile name χ that $a_i^\circ \cap \chi(a_j) = \varnothing$ iff $(a_i^r)^\circ \cap \chi(a_j^r) = \varnothing$. Because $\mathbf{mbr}(a_j^r) = \mathbf{mbr}(a_j)$, we know that $\chi(a_j^r) = \chi(a_j)$ for any χ. Therefore, we need to only show $a_i^\circ \cap \chi(a_j) = \varnothing$ iff $(a_i^r)^\circ \cap \chi(a_j) = \varnothing$. If $(a_i^r)^\circ \cap \chi(a_j) = \varnothing$, by $a_i \subseteq a_i^r$, we know $a_i^\circ \cap \chi(a_j) = \varnothing$. On the other hand, suppose $(a_i^r)^\circ \cap \chi(a_j) \ne \varnothing$. There exists a cell $c_{st} \subseteq \chi(a_j)$ such that $c_{st} \subseteq a_i^r$. This is possible iff $c_{st} \cap a_i^\circ$ is nonempty. Therefore, we know $a_i^\circ \cap \chi(a_j) \neq \varnothing$.

Definition 5.7 (regular solution) *A solution $\mathfrak{a} = \{a_i\}_{i=1}^n$ of a basic CDC network Γ is called* regular *if $\mathfrak{a}$ is the same as its regularisation $\mathfrak{a}^r = \{a_i^r\}_{i=1}^n$.*

It is easy to see that the regularisation of a solution is regular, *i.e.* $(a_i^r)^r = a_i^r$ for each $1 \leq r \leq n$. This is due to that $\mathfrak{a}^r$ and $\mathfrak{a}$ have the same frame and the same cell set.

Example 5.2 *Fig. 5.3 illustrates how to transform a solution $\{a_1, a_2, a_3\}$ (Fig. 5.3 (a)) of the constraint network specified in Fig. 5.1 into a regular solution $\{a_1^r, a_2^r, a_3^r\}$ (Fig. 5.3 (d)). The frame and the cell set are illustrated in Fig. 5.3 (b) and (c), respectively.*

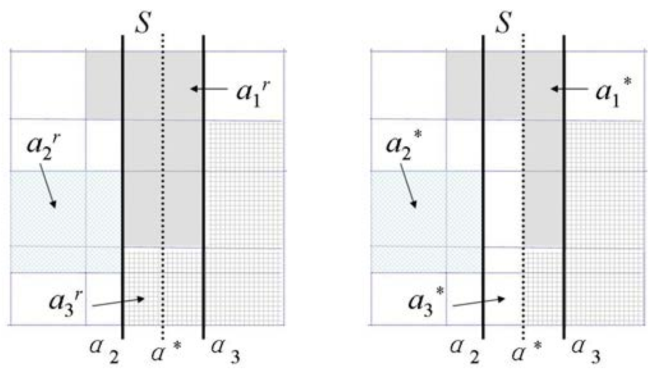
The following proposition asserts that canonical solutions (see Dfn. 5.5) are regular solutions.

Proposition 5.4 *Let Γ be a basic CDC network. Suppose $\mathfrak{a} = \{a_i\}_{i=1}^n$ is a canonical solution of Γ. Then $\mathfrak{a}$ is a regular solution of Γ.*

Proof Construct the frame and the cell set of $\mathfrak{a}$ as in Dfn. 5.6. By Dfn. 5.5, we know $\{I_x(a_i)\}_{i=1}^n$ and $\{I_y(a_i)\}_{i=1}^n$ are the canonical solutions of Γ_x and Γ_y. It is easy to see that $\alpha_i = i$ and $\beta_j = j$ for any $0 \leq i \leq n_x$ and $0 \leq j \leq n_y$. Moreover, each cell c_{ij} is exactly the pixel $p_{ij} = [i, i+1] \times [j, j+1]$. Because a_i is a digital region, it is clear $a_i^\circ \cap c_{st} = a_i^\circ \cap p_{st}$ is nonempty iff $p_{st} \subseteq a_i$. This implies $a_i = a_i^r$ for each $1 \leq i \leq n$. Therefore, $\mathfrak{a}$ is a regular solution of Γ.

5.3.2 Meet-Free Solution

Suppose $\mathfrak{a} = \{a_i\}_{i=1}^n$ is a regular solution of a basic CDC network $\Gamma = \{v_i \delta_{ij} v_j\}_{i,j=1}^n$. Let ι_{ij}^x be IA relation $\iota^x(\delta_{ij}) \cap \iota^x(\delta_{ji})^\sim$, and ρ_{ij}^x be $\iota_{ij} \setminus (\mathsf{m} \cup \mathsf{mi})$. By Prop. 5.2 we know that $\{I_x(a_i)\}_{i=1}^n$ is a solution of $\{v_i \iota_{ij}^x v_j\}_{i,j=1}^n$. Note that $\{I_x(a_i)\}_{i=1}^n$ is not necessarily a solution of $\Gamma_x = \{v_i \rho_{ij}^x v_j\}_{i,j=1}^n$, the x-projective IA network of Γ. We now show that we may transform the solution $\mathfrak{a}$ into a *meet-free* solution such that $\{I_x(a_i)\}_{i=1}^n$ is a solution of Γ_x, while keeping all the CDC relations between regions in $\mathfrak{a}$. Figure 5.4



(a) (b)

Figure 5.4: Illustration of meet-freeing

illustrates the transformation, in which the x-meet point α_2 between a_2^r and a_3^r is eliminated by removing two half-cells from a_3^r. Note two half-cells are removed from a_1^r in the procedure as well.

Definition 5.8 (meet-free solution) *A solution* $\mathfrak{a} = \{a_i\}_{i=1}^n$ *of* Γ *is* meet-free *if for any* i, j, $I_x(a_i)$ *does not meet* $I_x(a_j)$*, and* $I_y(a_i)$ *does not meet* $I_y(a_j)$*.*

Suppose $\mathfrak{a} = \{a_i\}_{i=1}^n$ is a solution of Γ and $\mathfrak{a}^r = \{a_i^r\}_{i=1}^n$ is its regularisation (cf. Fig. 5.3 (a) and (d)). Suppose $\mathbf{mbr}(a_i) = [x_i^-, x_i^+] \times [y_i^-, y_i^+]$ and $\mathbf{mbr}(a_j) = [x_j^-, x_j^+] \times [y_j^-, y_j^+]$ meet at x direction, i.e., $x_i^+ = x_j^-$. Recall that $\alpha_0 < \cdots < \alpha_{n_x}$ is the ordering of $\{x_i^-, x_i^+ : 1 \le i \le n\}$. Therefore, there exists some $0 < k < n_x$ such that $\alpha_k = x_i^+ = x_j^-$. We call α_k an *x-meet point* (cf. Fig. 5.4 (a)). Clearly,

$$x_i^- \le \alpha_{k-1} < x_i^+ = x_j^- = \alpha_k < \alpha_{k+1} \le x_j^+.$$

We aim to remove this meet point by transforming $\mathfrak{a}^r$ into another regular solution.

Write $\alpha^* = (\alpha_k + \alpha_{k+1})/2$. The line $x = \alpha^*$ divides each cell c_{kl} $(0 \le l < n_y)$ into two equal parts, written in order c_{kl}^- and c_{kl}^+. For each $1 \le s \le n$ and each $0 \le l < n_y$, if $c_{kl} \subseteq a_s^r$ but $c_{k-1,l} \notin a_s^r$ then delete c_{kl}^- from a_s^r (cf. Fig. 5.4 (a)). The remaining part of a_s^r, written as a_s^*, is still connected, and it is straightforward to show that $\mathfrak{a}^* = \{a_s^*\}_{s=1}^n$ is also a regular solution of Γ. Such a modification introduces no new meet points.

Continuing this process for at most n times, we will achieve a solution that has no x-meet points. The same procedures can be applied to y-meet points. In this way we may obtain a meet-free solution. Note that the obtained meet-free solution has the same frame but different cell set to $\mathfrak{a}$.

Proposition 5.5 *Each consistent basic CDC network has a regular solution that is meet-free.*

Proof We only need to show that above procedure eliminates a meet point, and the modified regions are still a regular solution.

Note that a_i^* is formally defined by

$$a_i^* = \bigcup\{c_{st} \subseteq a_i : s \neq k\} \cup \bigcup\{c_{kt}^+ : c_{kt} \subseteq a_i\} \cup \bigcup\{c_{kt}^- : c_{kt}, c_{k-1,t} \subseteq a_i\} \tag{5.12}$$

Clearly, each a_i^* is contained in a_i. We claim $\mathfrak{a}^* = \{a_i^*\}_{i=1}^n$ is a regular solution which has fewer x-meet points than $\mathfrak{a}$. Note that it can be straightforwardly proved that each a_i^* is a connected region.

Write $C^0 = \{c_{st} \in C(\mathfrak{a}) : s \neq k\}$, $C^+ = \{c_{kt}^+ : c_{kt} \in C(\mathfrak{a})\}$, $C^- = \{c_{kt}^- : c_{kt} \in C(\mathfrak{a})\}$. It is clear that $C(\mathfrak{a}^*)$, the cell set of $\mathfrak{a}^*$, is the union of C^0, C^+, and C^-. Moreover, each a_i^* is composed of cells in $C(\mathfrak{a}^*)$. This shows that $\mathfrak{a}^*$ is also regular.

It can be directly prove that $I_x(a_i^*)$ meets $I_x(a_j^*)$ only if $I_x(a_i)$ meets $I_x(a_j)$, and α^* is no longer meet point. It remains to verify that $\mathsf{dir}(a_i^*, a_j^*) = \mathsf{dir}(a_i, a_j)$ for each pair of $i \neq j$. This is equivalent to proving that

$$(\exists c \in C(\mathfrak{a}))c \subseteq a_i \cap \chi(a_j) \text{ iff } (\exists c^* \in C(\mathfrak{a}^*))c^* \subseteq a_i^* \cap \chi(a_j^*) \tag{5.13}$$

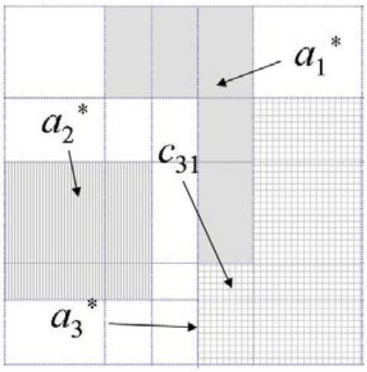
holds for all $1 \leq i, j \leq n$ and all $\chi \in \textsc{TileName}$. For each $c \in C(\mathfrak{a})$, define $c^* = c$ if $c \in C^0$ and define $c^* = c_{kt}^+$ if $c = c_{kt}$. It is straightforward to prove that $c \subseteq \chi(a_j)$ iff $c^* \subseteq \chi(a_j^*)$. Note that $c \subseteq a_i$ iff $c^* \subseteq a_i^*$ is also clear. This shows that if there exists $c \in C(\mathfrak{a})$ s.t. $c \subseteq a_i \cap \chi(a_j)$ then there exists c^* in $C(\mathfrak{a}^*)$ s.t. $c^* \subseteq a_i^* \cap \chi(a_j^*)$. On the other hand, suppose there exists $c' \in C(\mathfrak{a}^*)$ s.t. $c' \subseteq a_i^* \cap \chi(a_j^*)$. If $c' \in C_0$, then $c' \subseteq a_i \cap \chi(a_j)$; if $c' = c_{kt}^+$ for some t, then $c_{kt} \subseteq a_i \cap \chi(a_j)$. If $c' = c_{kt}^-$ for some t, then by $c' \subseteq a_i^*$ and the definition of a_i^*, both $c_{kt}, c_{k-1,t}$ are contained in a_i. It is easy to see that either $c_{kt} \subseteq \chi(a_j)$ or $c_{k-1,t} \subseteq \chi(a_j)$. This implies that either $c_{kt} \subseteq a_i \cap \chi(a_j)$ or $c_{k-1,t} \subseteq a_i \cap \chi(a_j)$. Therefore, Eq. 5.13 holds for every i, j, χ.

Therefore, we obtain another regular solution which has fewer meet-points. Repeat the above procedure until all meet points are eliminated and we get a meet-free, regular solution.

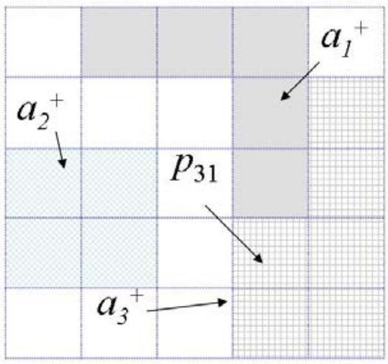
5.3.3 Canonical Solutions

The next step is to transform regions in a regular, meet-free solution into digital regions, and we will prove that the obtained regions are a canonical solution.

Suppose $\mathfrak{a} = \{a_s\}_{s=1}^n$ is a regular solution of Γ. As usual, we write $\mathbf{mbr}(a_i) = [x_i^-, x_i^+] \times [y_i^-, y_i^+]$, $S(\mathfrak{a}) = [\alpha_0, \alpha_{n_x}] \times [\beta_0, \beta_{n_y}]$, and $C(\mathfrak{a}) = \{c_{ij} : 0 \leq i < n_x, 0 \leq j < n_y\}$ for the frame and the cell set of $\mathfrak{a}$, respectively. Since $\mathfrak{a}$ is regular, each a_i is composed of a subset of cells in $C(\mathfrak{a})$.



(a)



(b)

Figure 5.5: Transform a regular solution (a) into a digital one (b)

We transform $\mathfrak{a} = \{a_s\}_{s=1}^n$ into a digital solution $\mathfrak{a}^+ = \{a_s^+\}_{s=1}^n$, where each a_s^+ is a digital region contained in $S(\mathfrak{a}^+) = [0, n_x] \times [0, n_y]$ such that

A pixel p_{ij} is contained in a_s^+ iff the cell c_{ij} in $C(\mathfrak{a})$ is contained in a_s.

That is,

$$a_s^+ = \bigcup \{p_{ij} : c_{ij} \subseteq a_s, 0 \leq i < n_x, 0 \leq j < n_y\}. \tag{5.14}$$

Fig. 5.5 illustrates the process. Clearly, a_s^+ is digital region. Furthermore, it can proved that a_s^+ is connected (meet-free resp.) if a_s is connected (meet-free resp.), and that $\mathfrak{a}^+$

is also a solution of Γ.

Proposition 5.6 *Suppose $\mathfrak{a} = \{a_s\}_{s=1}^n$ is a regular meet-free solution of a basic CDC network Γ. Then $\mathfrak{a}^+ = \{a_s^+\}_{s=1}^n$ constructed as above is also a regular and meet-free solution of Γ. Furthermore, $\mathfrak{a}^+$ is canonical solution.*

Proof It is clear that $S(\mathfrak{a}^+) = [0, n_x] \times [0, n_y]$ and $C(\mathfrak{a}^+) = \{p_{st} : 0 \leq s < n_x, 0 \leq t < n_y\}$, and $p_{st} \cap (a_i^+)^\circ$ is nonempty iff p_{st} is contained in a_i^+. Hence, $(a_i^+)^r = a_i^+$ for each i, and $\mathfrak{a}^+$ is regular. It is also clear that $I_x(a_i^+) = [s_1, s_2]$ iff $I_x(a_i) = [\alpha_{s_1}, \alpha_{s_2}]$. Therefore, $I_x(a_i^+)$ meets $I_x(a_j^+)$ iff $I_x(a_i)$ meets $I_x(a_j)$. Similar conclusion holds for the y-direction. Note that $\mathfrak{a}$ is meet-free, so $\mathfrak{a}^+$ is also meet-free.

For each i, j and each tile name χ, by Eq. 5.14 we know a pixel p_{st} is contained in a tile $\chi(a_j^+)$ iff the cell c_{st} is contained in $\chi(a_j)$. Therefore, a pixel p_{st} is contained in $a_i^+ \cap \chi(a_j^+)$ iff c_{st} is contained in $a_i \cap \chi(a_j)$. By definition we have $\mathsf{dir}(a_i, a_j) = \mathsf{dir}(a_i^+, a_j^+)$ for any i, j. Therefore, $\mathfrak{a}^+ = \{a_s^+\}_{s=1}^n$ is also a solution of Γ.

It remains to prove that $\mathfrak{a}^+$ is canonical. Note that $\mathfrak{a}^+$ is obviously a digital solution. We next prove $\{I_x(a_i^+)\}_{i=1}^n$ is a canonical solution of Γ_x. The y-direction can be proved the same way.

Because $\mathfrak{a}^+ = \{a_i^+\}_{i=1}^n$ is a solution to Γ, $I_x(a_i^+)\iota^x(\delta_{ij}, \delta_{ji})I_x(a_j^+)$ follows from Proposition 5.2. Note that $\mathfrak{a}^+$ is meet-free, therefore $I_x(a_i^+)$ cannot meet $I_x(a_j^+)$ for any i, j. This implies that $\{I_x(a_i^+)\}_{i=1}^n$ is a solution of the x-projective IA network Γ_x. Let E be the set of endpoints of intervals in $\{I_x(a_i^+)\}_{i=1}^n$. Note that $I_x(a_i^+) = [r, s]$ iff $I_x(a_i) = [\alpha_r, \alpha_s]$, where $\alpha_0 < \alpha_1 < \cdots < \alpha_{n_x}$ is the ordering of the all endpoints of intervals in $\{I_x(a_i)\}_{i=1}^n$. This implies E equals to $[0, n_x] \cap \mathbb{Z}$. Therefore, $\{I_x(a_i^+)\}_{i=1}^n$ is the canonical solution of Γ_x.

As a corollary, we have,

Proposition 5.7 *Each consistent basic CDC network has a canonical solution.*

5.3.4 Maximal Canonical Solution

This subsection shows how to construct a particular canonical solution (viz., the maximal canonical solution) directly, given a consistent basic CDC network $\Gamma = \{v_i \delta_{ij} v_j\}_{i,j=1}^n$.

By Theorem 5.1, we know the projective IA networks Γ_x and Γ_y are consistent. Therefore Γ_x and Γ_y have unique canonical interval solutions, say, $\{I_i\}_{i=1}^n$ and $\{J_i\}_{i=1}^n$. The following lemma shows that in any canonical solution of Γ, the mbr of the region interpreting variable v_i is $I_i \times J_i$.

Lemma 5.1 *Let $\Gamma = \{v_i \delta_{ij} v_j\}_{i,j=1}^n$ be a consistent basic CDC network. Suppose $\{I_i\}_{i=1}^n$ and $\{J_i\}_{i=1}^n$ are the canonical interval solutions of Γ_x and Γ_y, respectively. Let $m_i = I_i \times J_i$ for each i. If $\{a_i\}_{i=1}^n$ is a canonical solution of Γ, then we have $a_i \subseteq \boldsymbol{mbr}(a_i) = m_i$ for $1 \leq i \leq n$.*

Proof By Dfn. 5.5, we know $I_x(a_i) = I_i$ and $I_y(a_i) = J_i$ for each i. Therefore, $a_i \subseteq \mathbf{mbr}(a_i) = I_i \times J_i = m_i$.

For each constraint δ_{ij} in Γ, we assume

$$\delta_{ij} = \begin{bmatrix} d_{ij}^{NW} & d_{ij}^{N} & d_{ij}^{NE} \\ d_{ij}^{W} & d_{ij}^{O} & d_{ij}^{E} \\ d_{ij}^{SW} & d_{ij}^{S} & d_{ij}^{SE} \end{bmatrix}, \tag{5.15}$$

where $d_{ij}^{\chi} \in \{0, 1\}$ for each tile name $\chi \in$ TileName. Suppose $\{a_i\}_{i=1}^n$ is a canonical solution of Γ. By Lemma 5.1, we know $\mathbf{mbr}(a_j) = m_j$ for each j. Therefore, $\chi(a_j) = \chi(\mathbf{mbr}(a_j)) = \chi(m_j)$ for each $\chi \in$ TileName. Moreover, if $d_{ij}^{\chi} = 0$, then $a_i^{\circ} \cap \chi(m_j) = \varnothing$ because $(a_i, a_j) \in \delta_{ij}$. This means that a_i is disjoint with pixels which are contained in $\chi(m_j)$. Formally, write D_i for the set of all pixels in m_i that cannot appear in a_i,

$$D_i := \{p_{st} \subseteq m_i : (\exists\, j, \chi) \text{ such that } p_{st} \subseteq \chi(m_j) \text{ and } d_{ij}^{\chi} = 0\}. \tag{5.16}$$

We have the following lemma.

Lemma 5.2 *Let Γ and m_i be as in Lemma 5.1. For each i, define*

$$b_i := \bigcup \{p_{st} \subseteq m_i : p_{st} \notin D_i\}. \tag{5.17}$$

Suppose $\{a_i\}_{i=1}^n$ is a canonical solution of Γ. Then $a_i \subseteq b_i$ for each i.

Note that b_i is not always a connected region. However, as $a_i \subseteq b_i$ and a_i is connected, we know that b_i has a maximal connected component, say c_i, that contains a_i. The following lemma shows that $\{c_i\}_{i=1}^n$ is also a solution of Γ.

Lemma 5.3 *Let Γ, a_i and m_i be as in Lemma 5.1, and let b_i be as in Eq. 5.17. Assume c_i is the maximal connected component of b_i such that $a_i \subseteq c_i$. Then $\{c_i\}_{i=1}^n$ is a canonical solution of Γ.*

Proof We first prove that $\{c_i\}_{i=1}^n$ is a solution Γ, i.e., $\mathsf{dir}(a_i, a_j) = \mathsf{dir}(c_i, c_j)$. Note that $a_i \subseteq c_i \subseteq b_i$ and $\mathbf{mbr}(a_i) = \mathbf{mbr}(b_i) = m_i$, we know that $\mathbf{mbr}(c_i)$ is also m_i. Note that a_i, a_j, c_i, c_j are all digital regions, we know that $\mathsf{dir}(a_i, a_j) = \mathsf{dir}(c_i, c_j)$ equals to $a_i \cap \chi(m_j)$ contains a pixel iff $c_i \cap \chi(m_j)$ contains a pixel. Because $a_i \subseteq c_i \subseteq b_i$, it is sufficient to show that $b_i \cap \chi(m_j)$ contains a pixel implies $a_i \cap \chi(m_j)$ contains a pixel. Note that $b_i \cap \chi(m_j)$ contains a pixel implies $d_{ij}^{\chi} = 1$ by Equations 5.16 and 5.17, which further implies $a_i \cap \chi(m_j)$ has a pixel as $a_i \delta_{ij} a_j$ holds. Therefore $\{c_i\}_{i=1}^n$ is a solution Γ. That $\{c_i\}_{i=1}^n$ is canonical can be proved directly from the fact that $\{a_i\}_{i=1}^n$ is a canonical solution, due to the fact that $\mathbf{mbr}(a_i) = \mathbf{mbr}(c_i)$.

It is clear that $\mathbf{mbr}(c_i) = \mathbf{mbr}(b_i) = m_i$. Note that for any digital region b, there exists at most one connected component c, such that $\mathbf{mbr}(c) = \mathbf{mbr}(b)$. We may obtain a more general result which does not involve any particular canonical solution $\{a_i\}_{i=1}^n$.

Theorem 5.3 *Suppose $\Gamma = \{v_i \delta_{ij} v_j\}_{i,j=1}^n$ is a consistent basic CDC network, and b_i is defined as in Eq. 5.17. Then b_i has a unique maximal connected component, denoted as c_i, such that $\mathbf{mbr}(c_i) = \mathbf{mbr}(b_i) = m_i$. Furthermore, $\{c_i\}_{i=1}^n$ is the* maximal *canonical solution of Γ in the sense that for any canonical solution $\{a_i\}_{i=1}^n$, we have that $a_i \subseteq c_i$.*

So far we have shown that if Γ is consistent, then it has a maximal canonical solution $\{c_i\}_{i=1}^n$. The theorem also provides a method for computing the maximal canonical solution of a consistent basic CDC network:

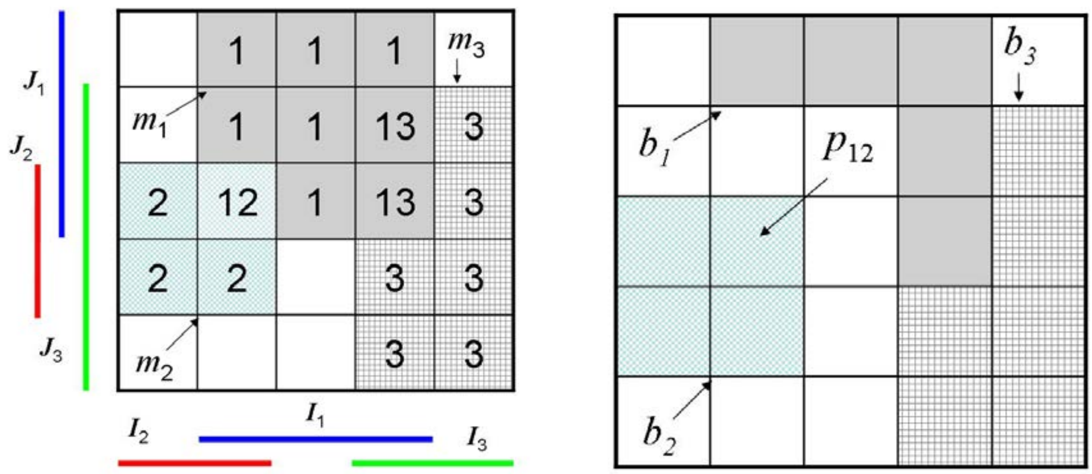
1. Compute the projective interval networks Γ_x and Γ_y of Γ;
2. Compute the canonical solutions $\{I_i\}_{i=1}^n$ and $\{J_i\}_{i=1}^n$ of Γ_x and Γ_y;
3. Compute b_i for each i, as in Eq. 5.17;
4. Find the unique connected component c_i of b_i such that $\mathbf{mbr}(c_i) = m_i$.

Example 5.3 *Consider the basic CDC constraint network specified in Fig. 5.1 which is consistent. We have*

$$x_2^- < x_1^- < x_2^+ < x_3^- < x_1^+ < x_3^+ \tag{5.18}$$

$$y_3^- < y_2^- < y_1^- < y_2^+ < y_3^+ < y_1^+ \tag{5.19}$$

The canonical interval solutions of Γ_x *and* Γ_y *are illustrated in Fig. 5.6 (a). Note that* d_{12}^O *(cf. Eq. 5.15) is 0. This excludes pixel* $p_{12} \subseteq O(m_2) = m_2$ *from* m_1 *(see Fig. 5.6 (a)). Note that each* b_i *obtained in this example happens to be connected (see Fig. 5.6 (b)). The maximal canonical solution of* Γ *is* $\{b_1, b_2, b_3\}$*.*



(a) (b)

Figure 5.6: Canonical interval solutions (a) and the maximal canonical solution (b)

5.4 A Cubic Algorithm for CSPSAT$_{\text{CDC}}(\mathcal{B}_{\text{CDC}})$

In this section, we describe an algorithm with time complexity $O(n^4)$ for determining the consistency of basic CDC network $\Gamma = \{v_i \delta_{ij} v_j\}_{i,j=1}^n$, and then refine it to cubic time complexity.

5.4.1 An Intuitive $O(n^4)$ Algorithm

The key idea of the algorithm is that we suppose Γ is consistent and try to compute the maximal canonical solution in the way described in the previous section. If the procedure fails at some step then we conclude that Γ is inconsistent. Otherwise, we succeed in obtaining a solution (in fact, the maximal canonical solution), then Γ is consistent. Note that the procedure may fail because of: (1) the projective IA network Γ_x or Γ_y is not consistent; or (2) some b_i does not have a maximal connected component which has the same mbr as b_i does. Furthermore, we need an extra step (Step 5) to check that whether $\{c_i\}_{i=1}^n$ is a solution of Γ. In conclusion, we have the following algorithm that decides the consistency of a basic CDC constraint network Γ that involves n variables. Figure 5.7 provides a flowchart of the algorithm.

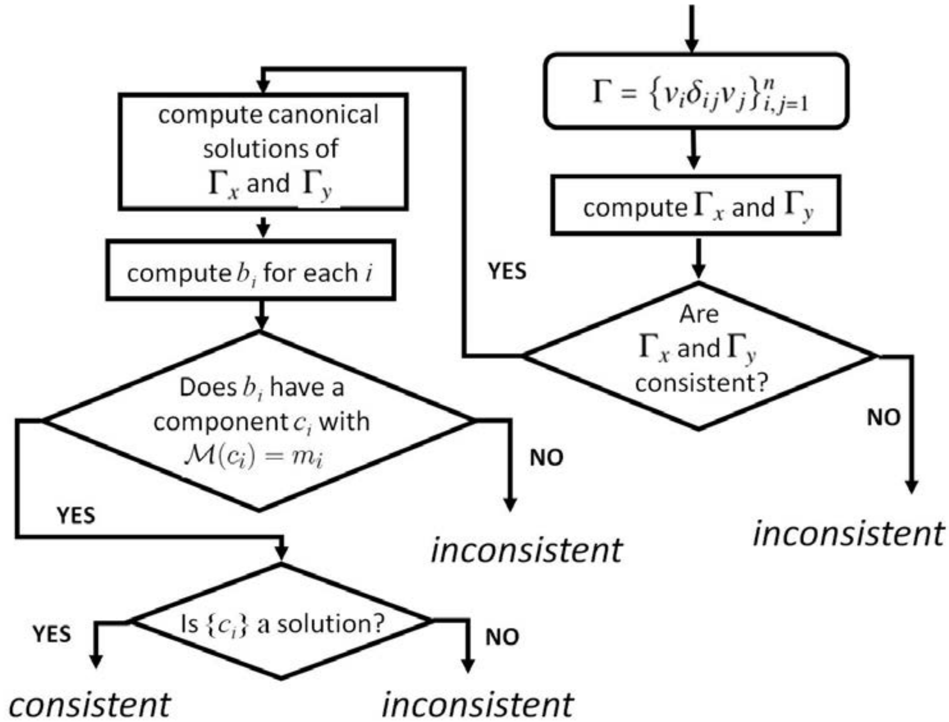


Figure 5.7: Flowchart of the main algorithm

We now explain the algorithm and discuss the time complexity step by step. Note that all the digital regions are contained in $[0, n_x] \times [0, n_y]$ and thus can be presented by $n_x \times n_y$ Boolean matrices.

Step 1. Computing projective IA networks (time complexity $O(n^2)$)

Input: Basic CDC network $\Gamma = \{v_i \delta_{ij} v_j\}_{i,j=1}^n$
Output: The consistency of Γ

Compute the projective networks Γ_x and Γ_y;
if *any of* Γ_x *and* Γ_y *is inconsistent* **then**
    return 'Inconsistent';
$\{I_i\}_{i=1}^n, \{J_i\}_{i=1}^n \leftarrow$ canonical interval solutions of Γ_x, Γ_y;
for $1 \leq i \leq n$ **do**
    Compute b_i;
end
for $1 \leq i \leq n$ **do**
    if b_i *has no maximal connected component with mbr* $I_i \times J_i$ **then**
        return 'Inconsistent';
    $c_i \leftarrow$ the maximal connected component of b_i with mbr $I_i \times J_i$;
end
if $\{c_i\}_{i=1}^n$ *is not a solution of* Γ **then**
    return 'Inconsistent';
return'Consistent'.

Algorithm 4: Deciding the consistency of basic CDC network Γ

For a pair of basic CDC constraints $x_i \delta_{ij} x_j$ and $x_j \delta_{ji} x_i$, the x- and y-projective IA relations ρ_{ij}^x and ρ_{ij}^y (Eq.s 5.5 and 5.6) can be computed in constant time as shown in Propositions 5.1 and 5.2. So the projective IA networks Γ_x and Γ_y can be constructed in $O(n^2)$ time.

Step 2. Computing canonical interval solutions (time complexity $O(n^3)$)

The consistency of Γ_x and Γ_y may be decided in cubic time by a path-consistency algorithm [3, 103]. If either of the two is inconsistent, then Γ is also inconsistent because of Theorem 5.1. Suppose Γ_x and Γ_y are consistent. Their canonical solutions $\{I_i\}_{i=1}^n$ and $\{J_i\}_{i=1}^n$ can be constructed in cubic time, where we assume intervals $I_i = [x_i^-, x_i^+]$ and $J_i = [y_i^-, y_i^+]$. Write $m_i = [x_i^-, x_i^+] \times [y_i^-, y_i^+]$.

Step 3. Computing b_i (time complexity $O(n^4)$)

Proposition 5.8 *For each pixel p contained in m_i, it takes $O(n)$ time to determine whether p is in b_i.*

Proof To determine whether p is in b_i, by definition we need to check whether p is not contained in $\chi(m_j)$ for all j, χ with $d_{ij}^\chi = 0$. Note that it needs constant time to decide whether a pixel is contained in a tile. Because there are at most $9n$ different tiles $\chi(m_j)$, it needs at most $O(n)$ time to determine whether p is contained in b_i.

Note that there are at most $O(n^2)$ pixels contained in m_i. As a consequence, it takes $O(n^3)$ time to compute b_i for each i. Therefore, we need $O(n^4)$ time in total to compute all b_i.

Step 4. Computing maximal connected components (time complexity $O(n^3)$)

For each i, all the maximal connected components of b_i may be computed in $O(n^2)$ time by a general Breadth-First Search algorithm. The mbrs of the maximal connected components can be obtained along the search without increasing time complexity. Therefore, it takes $O(n^2)$ time to find the desired maximal connected components of b_i for each i (if it exists), and this step needs $O(n^3)$ time in total.

Step 5. Verifying the candidate solution (time complexity $O(n^3)$)

We still need to check whether $\{c_i\}_{i=1}^n$ is a solution of Γ. Note that if the answer is positive, then $\{c_i\}_{i=1}^n$ is in fact the maximal canonical solution of Γ (see Theorem. 5.3). For each pair of c_i and c_j, we need to check for each $\chi \in$ TileName whether the following equation holds

$$d_{ij}^{\chi} = 0 \Leftrightarrow c_i^{\circ} \cap \chi(m_j) = \varnothing. \tag{5.20}$$

Because b_i and $\chi(m_j)$ have no common pixels if $d_{ij}^{\chi} = 0$, and c_i is a connected component of b_i, we only need to check

$$d_{ij}^{\chi} = 1 \Rightarrow c_i \cap \chi(m_j) \text{ contains a pixel.} \tag{5.21}$$

Proposition 5.9 *Let m_i, c_i be as constructed in Step 2 and Step 4, respectively. Let χ be a tile name. Suppose $d_{ij}^{\chi} = 1$. Then whether $c_i \cap \chi(m_j)$ contains a pixel can be checked in $O(n)$ time.*

Proof Suppose $m_i \cap \chi(m_j) = [x^-, x^+] \times [y^-, y^+]$. We do not need to check for every pixel p in $m_i \cap \chi(m_j)$ whether $p \subseteq c_i$. Instead, we only need to check this for all *boundary* pixels of $m_i \cap \chi(m_j)$, i.e., pixels $p_{k,l}$ with $(k,l) \in H_1 \cup H_2$, where

$$H_1 = \{(k,l) : k \in \{x^-, x^+ - 1\} \text{ and } y^- \leq l < y^+\},$$
$$H_2 = \{(k,l) : x^- \leq k < x^+ \text{ and } l \in \{y^-, y^+ - 1\}\}.$$

We justify the above statement as follows. If $m_i \cap \chi(m_j) = m_i$, then by $\mathbf{mbr}(c_i) = m_i$ we know there exists a boundary pixel which is contained in c_i. Otherwise, $m_i \cap \chi(m_j)$ is a rectangle strictly contained in m_i. Because $\mathbf{mbr}(c_i) = m_i$, this implies that c_i contains a pixel p out of $m_i \cap \chi(m_j)$. If c_i contains no boundary pixel of $m_i \cap \chi(m_j)$, then c_i, as a connected digital region, contains no pixel of $m_i \cap \chi(m_j)$ at all.

Since $H_1 \cup H_2$ contains $O(n)$ pixels and checking if a pixel is contained in c_i needs constant time, $\mathsf{dir}(c_i, c_j) = \delta_{ij}$ can be checked in $O(n)$ time.

Therefore, we can determine in $O(n)$ time whether $\mathsf{dir}(c_i, c_j) = \delta_{ij}$ for each pair of i, j. As a consequence, whether $\{c_i\}_{i=1}^n$ is a solution of Γ can checked in cubic time. This means Step 5 can be carried out in cubic time.

The algorithm takes $O(n^4)$ time. This result may be improved to $O(n^3)$, as shown in the following subsection.

5.4.2 Improvement to Cubic Time Complexity

All five steps except Step 3 in the main algorithm are of time complexity $O(n^3)$. This subsection will compute each b_i in $O(n^2)$ time, and thus obtain a cubic algorithm.

Recall that a digital region confined in $T = [0, n_x] \times [0, n_y]$ is represented by an $n_x \times n_y$ Boolean matrix. We assume digital region a confined in T is represented by $n_x \times n_y$ Boolean matrix $B(a)$, and that $B(a)$ is the zero matrix if a is a degenerate rectangle. For each $1 \le i \le n$, denote

$$Q_i = \sum \{B(T \cap \chi(m_j)) : 1 \le j \le n, j \ne i, \chi \in \textsc{TileName}, d_{ij}^{\chi} = 0\}. \tag{5.22}$$

Note that Q_i is not a Boolean matrix, as some of its entries may be integers larger than 1. Then the following equation about $B(b_i)$ follows from the definition of b_i directly

$$B(b_i)[k, l] = \begin{cases} 1, & \text{if } B(m_i)[k, l] = 1 \text{ and } Q_i[k, l] = 0; \\ 0, & \text{otherwise.} \end{cases} \tag{5.23}$$

Therefore, each $B(b_i)$ can be computed from $B(m_i)$ and Q_i in $O(n^2)$ time. We next show how to compute each Q_i in $O(n^2)$ time. To this end, we introduce the following operations on matrices.

Definition 5.9 (cumulative matrix and difference matrix) *For a matrix N, we define its* cumulative matrix *as*

$$\mathsf{acc}(N)[k,l] = \sum_{t=0}^{k} N[t,l]. \tag{5.24}$$

and its difference matrix *as*

$$\mathsf{diff}(N)[k,l] = \begin{cases} N[k,l], & \text{if } k = 0; \\ N[k,l] - N[k-1,l], & \text{otherwise.} \end{cases} \tag{5.25}$$

The cumulative matrix $\mathsf{acc}(N)$ can be computed column by column. We first add the first column to the second one, and then add the updated second column to the third, *etc.* (see Table 5.3 for an example.) In this way, cumulative matrix $\mathsf{acc}(N)$ can be computed linearly with respect to the number of elements of N.

$$\underset{\text{(a)}}{\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}} \quad \underset{\text{(b)}}{\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}} \quad \underset{\text{(c)}}{\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}}$$

Table 5.3: A matrix N (a) and its difference matrix $\mathsf{diff}(N)$ (b) and $\mathsf{acc}(\mathsf{diff}(N))$, the cumulative matrix of $\mathsf{diff}(N)$ (c)

It is clear that

- $N = \mathsf{acc}(\mathsf{diff}(N)) = \mathsf{diff}(\mathsf{acc}(N))$;
- the difference operation is additive, i.e., $\mathsf{diff}(N_1 + N_2) = \mathsf{diff}(N_1) + \mathsf{diff}(N_2)$.

Therefore, we have

$$\begin{aligned} Q_i &= \mathsf{acc}(\mathsf{diff}(Q_i)) \\ &= \mathsf{acc}(\mathsf{diff}(\textstyle\sum\{B(T \cap \chi(m_j)) : d^{\chi}_{ij} = 0 \text{ and } j \neq i\})) \\ &= \mathsf{acc}(\textstyle\sum\{\mathsf{diff}(B(T \cap \chi(m_j))) : d^{\chi}_{ij} = 0 \text{ and } j \neq i\}) \end{aligned}$$

Note that each digital region $T \cap \chi(m_j)$ is a rectangle (possibly degenerate), therefore, the non-zero elements in $\mathsf{diff}(B(T \cap \chi(m_j)))$ is of order $O(n)$. Thus $\mathsf{diff}(B(T \cap \chi(m_j)))$ can be generated in time $O(n)$. Therefore, $\sum\{\mathsf{diff}(B(T \cap \chi(m_j))) : d_{ij}^{\chi} = 0 \text{ and } j \neq i\}$, and hence Q_i and $B(b_i)$ can be computed in $O(n^2)$ time. In this way we improve Step 3 of the main algorithm from $O(n^4)$ to cubic time.

Theorem 5.4 *The consistency of a complete basic CDC constraint network can be decided in cubic time. If the network is consistent, then its maximal canonical solution may also be computed in cubic time.*

5.5 Define Relations outside the CDC

We have shown that $\textsc{cspsat}(\mathcal{B}_{\mathbf{CDC}})$ can be solved in cubic time. The following two sections will prove that $\textsc{cspsat}(\mathcal{B}^{\star}_{\mathbf{CDC}})$ is NP-hard by devising a polynomial reduction from the 3-SAT problem to $\textsc{cspsat}_{\mathbf{CDC}}(\mathcal{B}^{\star}_{\mathbf{CDC}})$. The reduction is based on the observation that some relations outside the CDC are *definable* in the CDC (see Definition 5.10). One such relation, called the *upper left corner* (ULC) relation, will play a critical role in the reduction. It is worth mentioning that such a technique for defining relations outside a qualitative calculus has been also used in [53] for generalizing the tractability of subclasses of the IA.

We say that two bounded regions have the ULC relation if their minimum bounding rectangles (mbrs) are incomparable, i.e., neither rectangle is contained in the other, although they have the same upper left corner point (see Figure 5.8 for illustrations). When considering only rectangles, the ULC relation is exactly the union of two basic rectangle relations, namely $\mathsf{s} \otimes \mathsf{fi}$ and its converse $\mathsf{si} \otimes \mathsf{f}$, where s, f are basic relations in the IA, and si and fi are their converses. Write $\ulcorner$ for the ULC relation. We note that $\ulcorner$ is not a CDC relation. The consistency decision problem over $\mathcal{B}^{\star}_{\mathbf{CDC}}$ is, informally speaking, equivalent to that over $\mathcal{B}^{\star}_{\mathbf{CDC}} \cup \{\ulcorner\}$. The NP-hardness of $\mathcal{B}^{\star}_{\mathbf{CDC}} \cup \{\ulcorner\}$ is then what one would expect.

Now we formally introduce the concept of *definable relation*.

Definition 5.10 (definable relation) *We say a relation γ is* definable *in the CDC, if there exists a CDC network Γ over $\mathcal{B}^{\star}_{\mathbf{CDC}}$ and over variables $\{u, v, w_1, \cdots, w_k\}$ $(k \geq 0)$,*

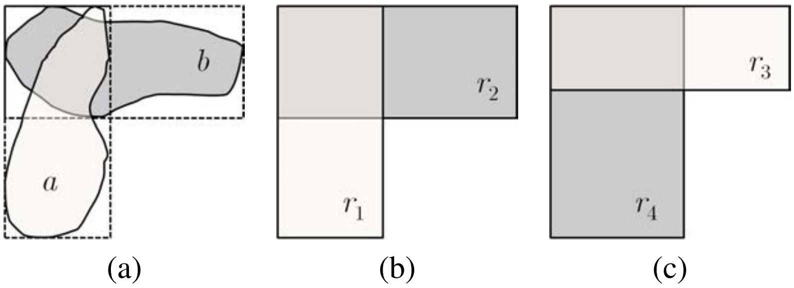


Figure 5.8: Illustrations of the symmetric ULC relation: (a) an instance (a,b) of the ULC relation; (b) an instance (r_1,r_2) of the rectangle relation $\mathsf{s}\otimes\mathsf{fi}$; (c) an instance (r_3,r_4) of the rectangle relation $\mathsf{si}\otimes\mathsf{f}$.

such that

$$\gamma = \{(a,b) : (\exists c_1,\cdots,c_k) \text{ s.t. } (a,b,c_1,\cdots,c_k) \text{ is a solution of } \Gamma\}. \tag{5.26}$$

In such a case, we also say that the relation γ is defined *by Γ (with respect to variables u and v). Here variables $w_1,\cdots,w_k$ are called auxiliary variables.*

A relation which is definable in the CDC is not necessarily a CDC relation.

The polynomial reduction from 3-SAT to CSPSAT$_{\mathbf{CDC}}(\mathcal{B}^{\star}_{\mathbf{CDC}}$ relies heavily on one particular relation which is definable in the CDC, namely, the upper left corner (ULC) relation (see Figure 5.8). Before discussing the ULC relation, we first review basic notions of the Rectangle Algebra and then introduce some examples of CDC definable relations that are closely related to basic rectangle relations.

Recall that for two bounded regions a,b, the basic RA relation of a to b is $\alpha\otimes\beta$, where α,β are basic IA relations such that $I_x(a)\alpha I_x(b)$ and $I_y(a)\beta I_y(b)$. We next introduce some relations that are definable in the CDC, which are proper subsets of basic RA relations.

Example 5.4 *Consider the relation defined by the following basic CDC network*

$$\Gamma_{\mathsf{s}\otimes\mathsf{f}} := \{u\ O\ v, v\ E{:}S\ E{:}S{:}O\ u\}. \tag{5.27}$$

It is easy to see that if (a,b) satisfies $\Gamma_{\mathsf{s}\otimes\mathsf{f}}$, then $\boldsymbol{mbr}(a)$ is contained in, and shares the upper left corner point with, $\boldsymbol{mbr}(b)$ (cf. Figure 5.9(a)). In terms of the RA language, we have $(\boldsymbol{mbr}(a),\boldsymbol{mbr}(b))\in\mathsf{s}\otimes\mathsf{f}$. The converse does not always hold, i.e., (a,b) may *not be a solution of* $\Gamma_{s\otimes f}$ *even if* $(\boldsymbol{mbr}(a), \boldsymbol{mbr}(b)) \in \mathsf{s} \otimes \mathsf{f}$*. This is because the CDC relation of b to a could be, for example,* $E{:}S\,E{:}S$*. In other words, the relation defined by* $\Gamma_{s\otimes f}$ *is actually a* proper *subset of* $\mathsf{s} \otimes \mathsf{f}$*. When only rectangles are considered, it is straightforward to see that* $\mathsf{s} \otimes \mathsf{f}$ *is exactly the relation defined by* $\Gamma_{s\otimes f}$*.*

Similarly, we define

$$\Gamma_{o\otimes f} \coloneqq \{u\ W{:}O\ v, v\ E{:}S\,E{:}S{:}O\ u\}, \tag{5.28}$$

$$\Gamma_{o\otimes fi} \coloneqq \{u\ S{:}S\,W{:}W{:}O\ v, v\ E{:}O\ u\}, \tag{5.29}$$

$$\Gamma_{o\otimes eq} \coloneqq \{u\ W{:}O\ v, v\ E{:}O\ u\}. \tag{5.30}$$

Figure 5.9 shows illustrations for the relations defined by these networks.

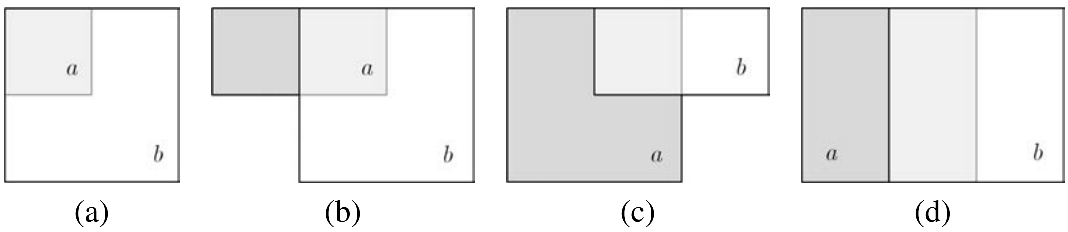


Figure 5.9: Illustrations of relations defined by CDC networks (a) $\Gamma_{\mathsf{s}\otimes\mathsf{f}}$, (b) $\Gamma_{\mathsf{o}\otimes\mathsf{f}}$, (c) $\Gamma_{\mathsf{o}\otimes\mathsf{fi}}$, and (d) $\Gamma_{\mathsf{o}\otimes\mathsf{eq}}$

The above relations do not involve auxiliary variables. We next introduce two relations that are defined by CDC networks involving auxiliary variables.

Example 5.5 *We say a is* right-side parallel with gap *(or* parallel *for short) to b, if* $I_x(a)$ pi $I_x(b)$ *and* $I_y(a) = I_y(b)$*, i.e., a is to the east of b (with gap) and has the same y-projection as b (see Figure 5.10). This relation is defined by the following basic CDC network*

$$\Gamma_{\parallel} = \{u\ E\ w, w\ E\ v, v\ W\ u\}, \tag{5.31}$$

where w is an auxiliary variable.

The parallel relation is strictly contained in the basic CDC relation E. Our next example is the upper left corner relation.

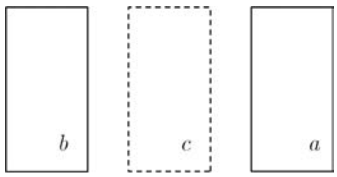


Figure 5.10: Illustration of a solution $\{a, b, c\}$ of $\Gamma_{\|}$, where c corresponds to the auxiliary variable w.

Definition 5.11 *Two bounded plane regions a, b are said to have the* upper left corner (ULC) relation, *denoted as $\ulcorner(a, b)$, if the mbrs of a, b are incomparable, i.e., $\boldsymbol{mbr}(a) \nsubseteq \boldsymbol{mbr}(b)$ and $\boldsymbol{mbr}(b) \nsubseteq \boldsymbol{mbr}(a)$, and have the same upper left corner point, or in the RA language, $(\boldsymbol{mbr}(a), \boldsymbol{mbr}(b))$ is an instance of either $\mathsf{s} \otimes \mathsf{fi}$ or its converse $\mathsf{si} \otimes \mathsf{f}$ (see Figure 5.8).*

The two possibilities of the ULC relation (see Figure 5.8) correspond to the two truth values of a propositional variable. This correspondence will be exploited in the design of the polynomial reduction from 3-SAT. For convenience, we introduce the following terminology.

Definition 5.12 *Suppose $\ulcorner(a, b)$. We say a is* horizontal *(*vertical*, resp.) with respect to b, or a is* horizontally instantiated *(*vertically instantiated*, resp.), if $\boldsymbol{mbr}(a)$ is related to $\boldsymbol{mbr}(b)$ by the RA relation $\mathsf{si} \otimes \mathsf{f}$ ($\mathsf{s} \otimes \mathsf{fi}$, resp.).*

The following proposition is the basis of the reduction to be introduced in Section 4.

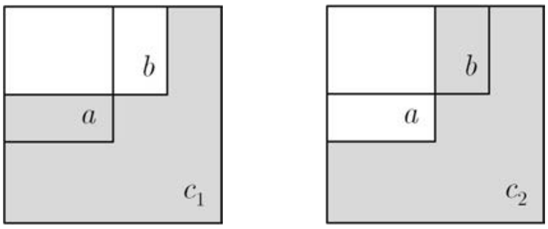
Proposition 5.10 *The ULC relation $\ulcorner$ is definable in the CDC.*

Proof Two auxiliary variables w_1 and w_2 are introduced. Let

$$\begin{aligned}\Gamma_{\ulcorner} = \{&u\ O\ w_1, w_1\ E{:}S\ E{:}S{:}O\ u, v\ O\ w_1, w_1\ E{:}S\ E{:}S\ v,\\ &v\ O\ w_2, w_2\ E{:}S\ E{:}S{:}O\ v, u\ O\ w_2, w_2\ E{:}S\ E{:}S\ u\}\end{aligned} \tag{5.32}$$

A basic constraint is imposed to each pair of variables in $\{u, v\} \times \{w_1, w_2\} \cup \{w_1, w_2\} \times \{u, v\}$. Note the constraints between u and v are $u?v$ and $v?u$, which are omitted in the equation.

First, we show that if $\{a,b,c_1,c_2\}$ is a solution of $\Gamma^{\ulcorner}$ (see Figure 5.11(a) and (b) for illustration), then a and b have the ULC relation. Therefore we need to prove (i) $\mathbf{mbr}(a)$ and $\mathbf{mbr}(b)$ have the same upper left corner point, and (ii) $\mathbf{mbr}(a)$ is not contained in $\mathbf{mbr}(b)$, and vice versa. According to the network $\Gamma^{\ulcorner}$, we have that $a\ O\ c_1$, $c_1\ E{:}S\,E{:}S{:}O\ a$, and $b\ O\ c_1$, $c_1\ E{:}S\,E{:}S\ b$, which directly imply that $\mathbf{mbr}(a)$ and $\mathbf{mbr}(b)$ have the same upper left corner point as $\mathbf{mbr}(c_1)$ does (see Figure 5.11(a)). Furthermore, because $c_1\ E{:}S\,E{:}S{:}O\ a$ and $c_1\ E{:}S\,E{:}S\ b$, we know $\mathbf{mbr}(a)$ is not contained in $\mathbf{mbr}(b)$. Similarly, $\mathbf{mbr}(b)$ is not contained in $\mathbf{mbr}(a)$ (see Figure 5.11(b)). Therefore, a and b have the ULC relation.



(a) (b)

Figure 5.11: Illustrations for a solution of $\Gamma^{\ulcorner}$, where a,b are rectangles, and c_1,c_2 are the shaded region in (a) and (b) respectively.

On the other hand, if (a,b) is an instance of $^{\ulcorner}$, then it is easy to find c_1,c_2 such that $\{a,b,c_1,c_2\}$ is a solution of $\Gamma^{\ulcorner}$.

5.6 Consistency Checking of Incomplete Networks of Basic CDC Constraints

This section proves that consistency checking of incomplete basic CDC networks, i.e., CSPSAT$(\mathcal{B}^{\star}_{\mathbf{CDC}})$,is an NP-hard problem. We achieve this by reducing the 3-SAT problem to CSPSAT$_{\mathbf{CDC}}(\mathcal{B}^{\star}_{\mathbf{CDC}})$. For each 3-SAT instance ϕ, we construct an incomplete basic CDC network Γ_ϕ in polynomial time, and show that ϕ is satisfiable if and only if Γ_ϕ is consistent.

In this section, we assume $V = \{p_1, p_2, \cdots, p_n\}$ is a set of propositional variables. Suppose $\phi = c_1 \wedge c_2 \wedge \cdots \wedge c_m$, where clause c_j is of the form $p^*_r \vee p^*_s \vee p^*_t$ and p^*_r, p^*_s, p^*_t are literals over V. We first introduce a basic CDC network Γ_p for each propositional variable $p \in V$, and then introduce a basic CDC network Γ_c for each clause c. The basic CDC network Γ_ϕ is defined as the union of all Γ_{c_j} $(1 \leq j \leq m)$. Note that when expressing the constraints in Γ_c, for simplicity, we often use non-CDC constraints which are definable in the CDC. We stress that if such a constraint, e.g., $\ulcorner(u, v)$, appears, we always assume that it is replaced by the constraints in the CDC network that defines it, e.g., constraints in $\Gamma_{\ulcorner}$. In particular, constraints in the forms of $\mathsf{s} \otimes \mathsf{f}(u, v)$ ($\mathsf{o} \otimes \mathsf{f}(u, v)$, $\mathsf{o} \otimes \mathsf{fi}(u, v)$, $\mathsf{o} \otimes \mathsf{eq}(u, v)$, resp.) should be replaced by the constraints in the network $\Gamma_{\mathsf{s}\otimes\mathsf{f}}$ ($\Gamma_{\mathsf{o}\otimes\mathsf{f}}$, $\Gamma_{\mathsf{o}\otimes\mathsf{fi}}$,$\Gamma_{\mathsf{o}\otimes\mathsf{eq}}$, resp.), defined in Eq.s 5.27-5.30. Furthermore, auxiliary variables for different non-CDC constraints should be different.

Note. In a slight abuse of notation, we will use variables f, u, v, w in this section with subscripts and/or superscripts to denote *both* spatial variables and the regions in a solution (if any) that correspond to them, which will make the statements much clearer.

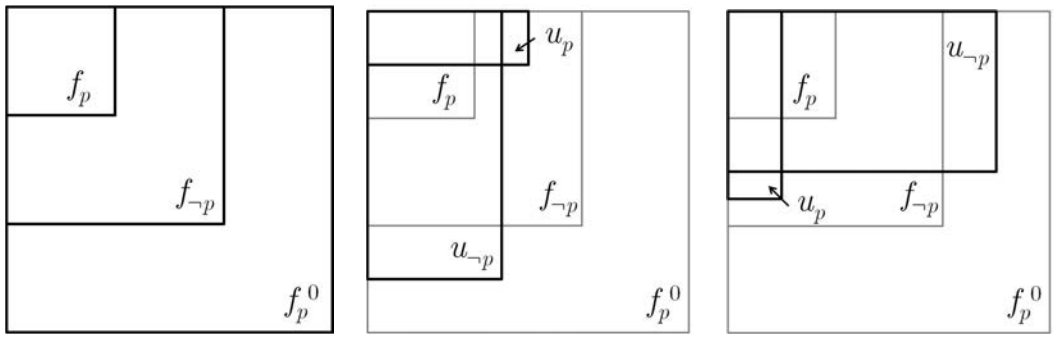
5.6.1 CDC Constraints Related to Propositional Variables

For each propositional variable p, we introduce five spatial variables u_p, $u_{\neg p}$, f_p, $f_{\neg p}$, and f_p^0, and define a basic CDC network Γ_p which is incomplete. Shortly we will show that Γ_p is consistent and has a solution in which all the above five spatial variables are rectangles. In the following informal description, we assume the five variables are all rectangles for simplicity. The network Γ_p will ensure the two requirements:

- The configuration of $f_p, f_{\neg p}, f_p^0$ is as shown in Figure 5.12(a);
- Suppose $f_p, f_{\neg p}, f_p^0$ are predefined. The configuration of $u_p, u_{\neg p}$ has two possibilities (cf. Figure 5.12(b) and (c)).

The second condition is mainly achieved by the fact that u_p is horizontal w.r.t. f_p iff $u_{\neg p}$ is vertical w.r.t. $f_{\neg p}$, which is guaranteed by the following constraints (see Figure 5.12(b) and (c)):

- u_p is contained in $f_{\neg p}$ and has the ULC relation with f_p,
- $u_{\neg p}$ contains f_p, and is contained in f_p^0, and has the ULC relation with $f_{\neg p}$,
- u_p and $u_{\neg p}$ have the ULC relation.



(a) (b) (c)

Figure 5.12: Illustrations of spatial variables in $\{f_p, f_{\neg p}, f_p^0, u_p, u_{\neg p}\}$: (a) the frame spatial variables $f_p, f_{\neg p}, f_p^0$; (b) a solution of Γ_p where u_p is horizontally instantiated; (c) a solution of Γ_p where u_p is vertically instantiated.

The following definition specifies constraints in Γ_p formally.

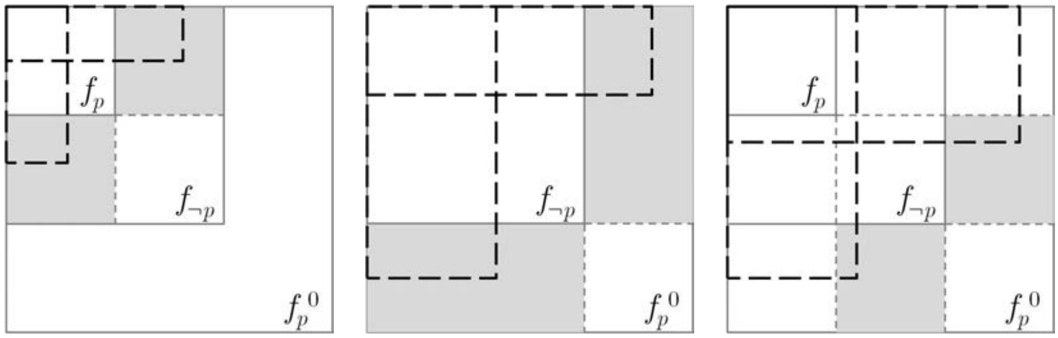
Definition 5.13 *Let p be a propositional variable, and $u_p, u_{\neg p}, f_p, f_{\neg p}, f_p^0$ be five spatial variables. The basic CDC network Γ_p has exactly the following constraints*

$$\begin{array}{lll} \mathsf{s} \otimes \mathsf{f}(f_p, f_{\neg p}), & \mathsf{s} \otimes \mathsf{f}(f_{\neg p}, f_p^0), & \\ \mathsf{s} \otimes \mathsf{f}(u_p, f_{\neg p}), & \mathsf{s} \otimes \mathsf{f}(f_p, u_{\neg p}), & \mathsf{s} \otimes \mathsf{f}(u_{\neg p}, f_p^0) \\ \ulcorner(u_p, f_p), & \ulcorner(u_{\neg p}, f_{\neg p}), & \ulcorner(u_p, u_{\neg p}), \end{array} \tag{5.33}$$

where $\mathsf{s} \otimes \mathsf{f}$ and $\ulcorner$ should be replaced by constraints in Eq. 5.27 and Eq. 5.32 respectively. The non-CDC constraints appearing in Γ_p are replaced by the basic CDC constraints that define them. We call u_p and $u_{\neg p}$ the dual spatial variables *for p, and call f_p, $f_{\neg p}$, and f_p^0 the* frame spatial variables *for p.*

We note that apart from $u_p, u_{\neg p}, f_p, f_{\neg p}, f_p^0$, the network Γ_p also involves six other auxiliary spatial variables, which are introduced by the three ULC constraints.

Two solutions of Γ_p are shown in Figure 5.12, where u_p is horizontally instantiated (w.r.t both f_p and $u_{\neg p}$) in the solution shown in Figure 5.12(b), but vertically instantiated in the solution shown in Figure 5.12(c). Although its position is not determined, we know that the lower right corner of **mbr**(u_p) is in the interior of the shaded upper left (lower right, resp.) sub-rectangle if u_p is horizontal (vertical, resp.) w.r.t. f_p (see Figure 5.13(a)). We next show that in any solution of Γ_p, u_p is vertical w.r.t. f_p if and



(a) (b) (c)

Figure 5.13: Possible positions for the lower right corner points of u_p (a) and $u_{\neg p}$ (b) (c)

only if $u_{\neg p}$ is horizontal w.r.t. $f_{\neg p}$.

Proposition 5.11 *Let Γ_p be the basic CDC network for propositional variable p. Suppose $\{u_p, u_{\neg p}, f_p, f_{\neg p}, f_p^0\}$ is a solution of Γ_p. Then u_p is vertical w.r.t f_p iff $u_{\neg p}$ is horizontal w.r.t. $f_{\neg p}$.*

Proof It is clear that the mbrs of $u_p, u_{\neg p}, f_p, f_{\neg p}, f_p^0$ have the same upper left corner point. We now consider the lower right corner points of $\mathbf{mbr}(u_p)$ and $\mathbf{mbr}(u_{\neg p})$. The constraints $\ulcorner(u_p, f_p)$ and $\mathsf{s} \otimes \mathsf{f}(u_p, f_{\neg p})$ restrict the lower right corner point of $\mathbf{mbr}(u_p)$ to the shaded part in Figure 5.13(a). Similarly, the constraints $\ulcorner(u_{\neg p}, f_{\neg p})$ and $\mathsf{s} \otimes \mathsf{f}(u_{\neg p}, f_p^0)$ restrict the lower right corner point of $\mathbf{mbr}(u_{\neg p})$ to the shaded part in Figure 5.13(b). By constraint $\mathsf{s} \otimes \mathsf{f}(f_p, u_{\neg p})$, the possible area of the lower right corner point of $\mathbf{mbr}(u_{\neg p})$ is further restricted to the shaded part in Figure 5.13(c).

Therefore, if u_p and $u_{\neg p}$ are both vertical or both horizontal, then the lower right corner point of $\mathbf{mbr}(u_p)$ must be in $\mathbf{mbr}(u_{\neg p})$, which implies $\mathbf{mbr}(u_p) \subset \mathbf{mbr}(u_{\neg p})$ as their upper left corner points are the same. This contradicts the constraint $\ulcorner(u_p, u_{\neg p})$ in Γ_p, which requires that $\mathbf{mbr}(u_p)$ and $\mathbf{mbr}(u_{\neg p})$ are partially overlapping. Therefore, u_p and $u_{\neg p}$ cannot be vertical or horizontal at the same time, i.e., u_p is vertical w.r.t f_p iff $u_{\neg p}$ is horizontal w.r.t. $f_{\neg p}$.

The mutual exclusion of vertically and horizontally instantiations of dual variables $u_p, u_{\neg p}$ corresponds to the mutual exclusion of the truth values of p and its negation $\neg p$.

For each propositional variable $p_i \in V = \{p_1, p_2, \cdots, p_n\}$, we introduce dual spatial variables

$$u_i \equiv u_{p_i} \text{ and } u_{\neg i} \equiv u_{\neg p_i}, \tag{5.34}$$

and three frame spatial variables

$$f_i \equiv f_{p_i}, f_{\neg i} \equiv f_{\neg p_i} \text{ and } f_i^0 \equiv f_{p_i}^0, \tag{5.35}$$

and construct, as described above in Definition 5.13, a basic CDC network Γ_{p_i} over spatial variables $\{u_i, u_{\neg i}, f_i, f_{\neg i}, f_i^0\}$.

In order to fix the relative direction between frame spatial variables for different propositional variables (cf. Figure 5.14), we introduce a set of reference spatial variables. Precisely, let

$$V_{ref} = \{w_{ref}, f_{ref}, f_{\neg ref}, f_{ref}^0\}, \tag{5.36}$$

$$\Gamma_{ref} = \{w_{ref}\ O\ f_{ref}\ O\ f_{\neg ref}\ O\ f_{ref}^0,\ f_{ref}^0\ S{:}O\ f_{\neg ref}\ S{:}O\ f_{ref}\ S{:}O\ w_{ref}\}, \tag{5.37}$$

where the shorthand, say, $x\ S{:}O\ y\ S{:}O\ z$ denotes that $x\ S{:}O\ y$ and $y\ S{:}O\ z$. Note that the reference variable w_{ref} in V_{ref} will be used in the next subsection when constructing the basic CDC networks for propositional clauses. Furthermore, we require

$$\parallel (f_1, f_{ref}),\ \parallel (f_{\neg 1},\ f_{\neg ref}),\ \parallel (f_1^0, f_{ref}^0), \tag{5.38}$$

$$\parallel (f_{i+1}, f_i),\ \parallel (f_{\neg(i+1)}, f_{\neg i}),\ \parallel (f_{i+1}^0, f_i^0,), \qquad (1 \le i < n) \tag{5.39}$$

where the relation $\parallel$ is defined in Eq. 5.31. Note that these $3n$ parallel constraints introduce $3n$ new auxiliary variables.

Definition 5.14 *We write Γ_V for the set of basic CDC constraints that includes those in Γ_{ref} and Γ_p for each $p \in V$, and those basic CDC constraints that define the parallel relations specified in Eq.s 5.38 and 5.39.*

Example 5.6 *A solution of Γ_V is constructed as follows (see Figure 5.14 for an illus-*

tration).

$$w_{ref} = [0, 0.5] \times [0.9, 1], \tag{5.40}$$
$$f_{ref} = [0, 0.5] \times [0.7, 1], \tag{5.41}$$
$$f_{\neg ref} = [0, 0.5] \times [0.4, 1], \tag{5.42}$$
$$f^0_{ref} = [0, 0.5] \times [0.2, 1]. \tag{5.43}$$

For each propositional variable p_i, we define $f_i, f_{\neg i}$, and f^0_i as follows.

$$f_i = [i, i + 0.3] \times [0.7, 1] \tag{5.44}$$
$$f_{\neg i} = [i, i + 0.6] \times [0.4, 1] \tag{5.45}$$
$$f^0_i = [i, i + 0.8] \times [0.2, 1] \tag{5.46}$$

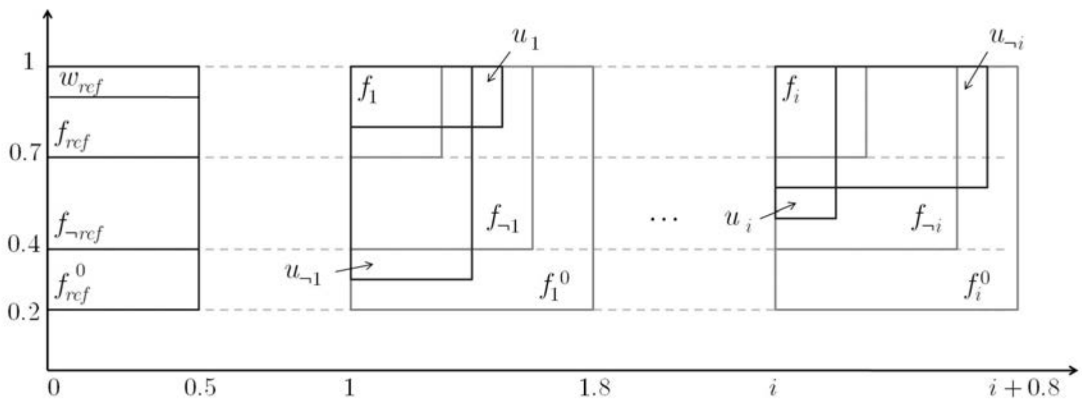


Figure 5.14: Illustration of a solution of Γ_V

The network Γ_V does not impose new constraints on u_i and $u_{\neg i}$. Therefore, we can lay u_i horizontally and $u_{\neg i}$ vertically, or vice versa. For example, we may define (see u_1 and $u_{\neg 1}$ in Figure 5.14 for illustration).

$$u_i = [i, i + 0.5] \times [0.8, 1], \quad u_{\neg i} = [i, i + 0.4] \times [0.3, 1], \tag{5.47}$$

or vice versa (see u_i and $u_{\neg i}$ in Figure 5.14 for illustration),

$$u_i = [i, i + 0.2] \times [0.5, 1], \quad u_{\neg i} = [i, i + 0.7] \times [0.6, 1]. \tag{5.48}$$

5.6.2 CDC Constraints Related to Clauses

In the above subsection, we have set up the correspondence between the truth value (true/false) of a propositional variable p and the vertical/horizontal state of the corresponding spatial variable u_p. This subsection introduces for each clause $c = p_r^* \vee p_s^* \vee p_t^*$ $(1 \leq r < s < t \leq n)$ a basic CDC constraint network Γ_c.

Note that clause c rules out those assignments π such that $\pi(p_i) =$ false iff p_i^* is positive in c for $i = r, s, t$. We next construct a network Γ_c which excludes the configurations in which u_i is horizontally instantiated iff p_i^* is positive in c for $i = r, s, t$. Write u_i^* for the spatial variable that corresponds to p_i^*, i.e.,

$$u_i^* = \begin{cases} u_i, & \text{if } p_i^* = p_i, \\ u_{\neg i}, & \text{if } p_i^* = \neg p_i. \end{cases} \tag{5.49}$$

Then, Γ_c should rule out the configurations in which u_i^* are horizontally instantiated for $i = r, s, t$. Assume that $state_i$ is a vertical/horizontal state that takes values in $\{$vertical, horizontal$\}$. It will be proved that for each 3-tuple $(state_r, state_s, state_t)$ of vertical/horizontal states, Γ_c has a solution in which u_i^* is in $state_i$ for $i = r, s, t$ if and only if $state_r, state_s, state_t$ are not all horizontal.

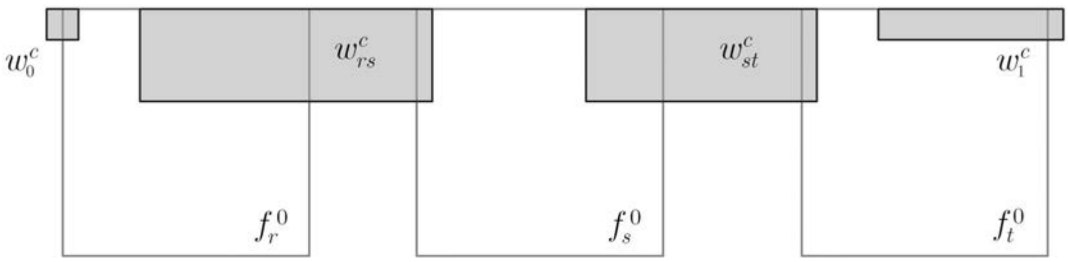


Figure 5.15: Positions of $w_0^c, w_{rs}^c, w_{st}^c, w_1^c$.

We next give an intuitive explanation for the construction of constraints in Γ_c. Consider the frames of p_r, p_s, and p_t. We introduce four auxiliary spatial variables $w_0^c, w_{rs}^c, w_{st}^c$, and w_1^c such that they are "bridged" by f_r^0, f_s^0, and f_t^0 in the sense that their x-projections are overlapped one by one in the ordering $w_0^c, f_r^0, w_{rs}^c, f_s^0, w_{st}^c, f_t^0, w_1^c$ (see Figure 5.15). The spatial variables u_i^* $(i = r, s, t)$ may be either horizontally or vertically instantiated. To exclude the case in which u_r^*, u_s^*, and u_t^* are all horizontally instantiated, we introduce a new spatial variable v_c and several related constraints. Intuitively, the mbr of v_c has the form as shown in Figure 5.16 (a), but the interior of v_c is disjoint from spatial variables in

$$X_c \equiv \{w_0^c, u_r^*, w_{rs}^c, u_s^*, w_{st}^c, u_t^*, w_1^c\}. \tag{5.50}$$

Note that this happens only if u_r^*, u_s^*, and u_t^* do not bridge all the gaps between w_0^c, w_{rs}^c, w_{st}^c, and w_1^c. In what follows, we refer to this as the *gap condition*.

The gap condition is fulfilled by imposing the following constraints:

$$x \; O \; v_c \qquad\qquad (x \in X_c), \tag{5.51}$$

$$v_c \; E{:}S\,E{:}S \; w_0^c, \qquad v_c \; S{:}S\,W{:}W \; w_1^c, \tag{5.52}$$

$$v_c \; E{:}S\,E{:}S{:}S\,W{:}W \; x \qquad\qquad (x \in \{u_r^*, w_{rs}^c, u_s^*, w_{st}^c, u_t^*\}). \tag{5.53}$$

Note that the constraint of v_c to a spatial variable x in X_c does not contain tile name O. This means that the interior of v_c is disjoint from $\mathbf{mbr}(x)$. From constraints $w_0^c \; O \; v_c$ and $v_c \; E{:}S\,E{:}S \; w_0^c$, we know $\mathbf{mbr}(v_c)$ has the same upper left corner point as $\mathbf{mbr}(w_0^c)$. Similarly, $\mathbf{mbr}(v_c)$ has the same upper right corner point as $\mathbf{mbr}(w_1^c)$.

Assume the gap condition is violated. This means that there is no gap between any two consecutive regions in X_c (see Figure 5.16 (b)). In this case, the union of the mbrs of these regions contains the rectangle $I_x(v_c) \times I_y(w_0^c)$, and it should be excluded from the interior of v_c. This contradicts the requirement that $\mathbf{mbr}(v_c)$ shares the same upper left corner point with w_0^c. Therefore, the constraints in Eq.s 5.51-5.53 are not satisfiable if the gap condition is violated.

On the other hand, suppose there is a gap between two consecutive regions in X_c (see Figure 5.16 (a)). It is straightforward to check that all the constraints are satisfied if we let v_c be the region obtained (after necessary regularisation) from subtracting regions in X_c from a rectangle which has the same upper left (upper right resp.) corner point as w_0^c (w_1^c, resp.), and is "deeper" than any regions in X.

After an intuitive description, we formally define the basic CDC constraints in Γ_c for clause $c \equiv p_r^* \vee p_s^* \vee p_t^*$ in ϕ, introducing spatial variables $w_0^c, w_{rs}^c, w_{st}^c, w_1^c$ and v_c.

We begin with the constraints involving the four "pier" spatial variables w_0^c, w_{rs}^c, w_{st}^c, and w_1^c. As shown in Figure 5.15, these variables are interpreted as rectangles that

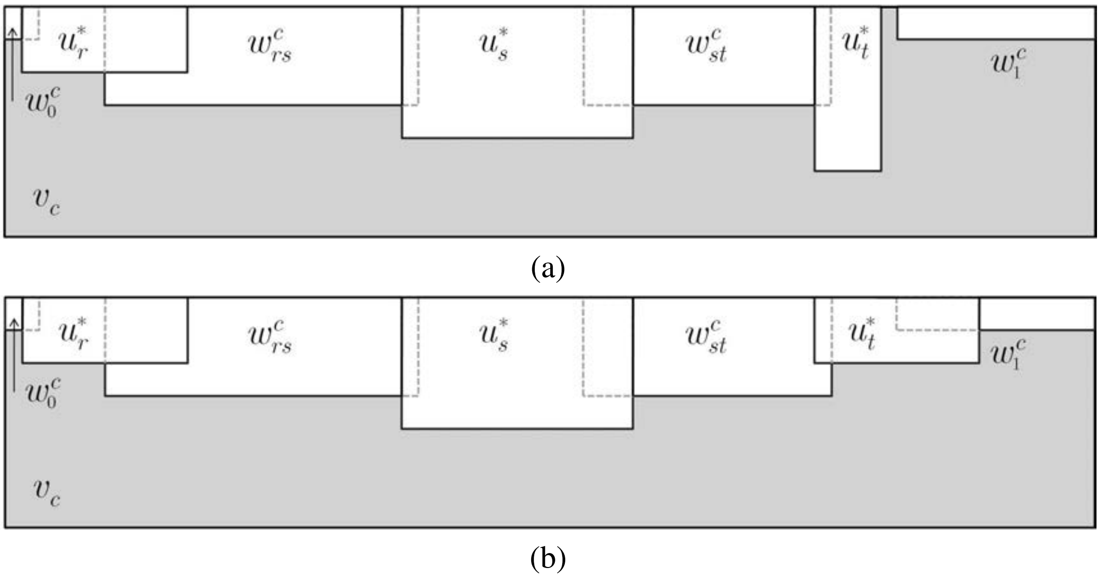


Figure 5.16: Illustrations of the situations in which (a) the gap condition is satisfied, and (b) the gap condition is violated

are bridged by the frames f_r^0, f_s^0, and f_t^0.

$$\| \ (w_0^c, w_{ref}), \quad \| \ (w_1^c, w_0^c), \tag{5.54}$$

$$\mathsf{o} \otimes \mathsf{f}(w_0^c, f_r), \tag{5.55}$$

$$\begin{cases} \mathsf{o} \otimes \mathsf{eq}(f_r, w_{rs}^c), & \mathsf{o} \otimes \mathsf{eq}(w_{rs}^c, f_s), & \text{if } p_r^* = p_r, \\ \mathsf{o} \otimes \mathsf{fi}(f_{\neg r}, w_{rs}^c), & \mathsf{o} \otimes \mathsf{eq}(w_{rs}^c, f_s), & \text{if } p_r^* = \neg p_r, \end{cases} \tag{5.56}$$

$$\begin{cases} \mathsf{o} \otimes \mathsf{eq}(f_s, w_{st}^c), & \mathsf{o} \otimes \mathsf{eq}(w_{st}^c, f_t), & \text{if } p_s^* = p_s, \\ \mathsf{o} \otimes \mathsf{fi}(f_{\neg s}, w_{st}^c), & \mathsf{o} \otimes \mathsf{eq}(w_{st}^c, f_t), & \text{if } p_s^* = \neg p_s, \end{cases} \tag{5.57}$$

$$\begin{cases} \mathsf{o} \otimes \mathsf{fi}(f_t, w_1^c), & \text{if } p_t^* = p_t, \\ \mathsf{o} \otimes \mathsf{fi}(f_{\neg t}, w_1^c), & \text{if } p_t^* = \neg p_t. \end{cases} \tag{5.58}$$

Note that the constraints $\|$ and $\mathsf{o} \otimes \mathsf{f}$, $\mathsf{o} \otimes \mathsf{eq}$, $\mathsf{o} \otimes \mathsf{fi}$ in the above equations are shorthand expressions of the basic CDC constraints that define them (see Examples 5.4 and 5.5). The first equation (Eq. 5.54) requires that w_0^c and w_1^c are of the same height as the reference spatial variable w_{ref}. The second equation (Eq. 5.55) specifies the rectangle relation between w_0^c and f_r. The third equation (Eq. 5.56) specifies that w_{rs}^c is of the same height as the inner frame f_i of u_i. The position of w_{rs}^c, however, depends on the sign of literal p_r^*. If p_r^* is positive, then we require w_{rs}^c to bridge the gap between f_r and f_s; otherwise, we require w_{rs}^c to bridge the gap between $f_{\neg r}$ and f_s. The constraints involving w_{st}^c, specified in Eq. 5.57, are similar. The last equation (Eq. 5.58) specifies that f_t overlaps w_1^c if p_t^* is positive, and $f_{\neg t}$ overlaps w_1^c otherwise.

We illustrate the construction of the above constraints with an example.

Example 5.7 *Consider the clause $c = p_r \vee \neg p_s \vee p_t$. Figure 5.17 illustrates the configuration of variables $w_0^c, w_{rs}^c, w_{st}^c, w_1^c$ with respect to the frame variables for p_r, p_s and p_t. The network Γ_c specifies that $\mathsf{O} \otimes \mathsf{f}(w_0^c, f_r)$; $\mathsf{O} \otimes \mathsf{eq}(f_r, w_{rs}^c), \mathsf{O} \otimes \mathsf{eq}(w_{rs}^c, f_s)$; $\mathsf{O} \otimes \mathsf{fi}(f_{\neg s}, w_{st}^c)$, $\mathsf{O} \otimes \mathsf{eq}(w_{st}^c, f_t)$; and $\mathsf{O} \otimes \mathsf{fi}(f_t, w_1^c)$.*

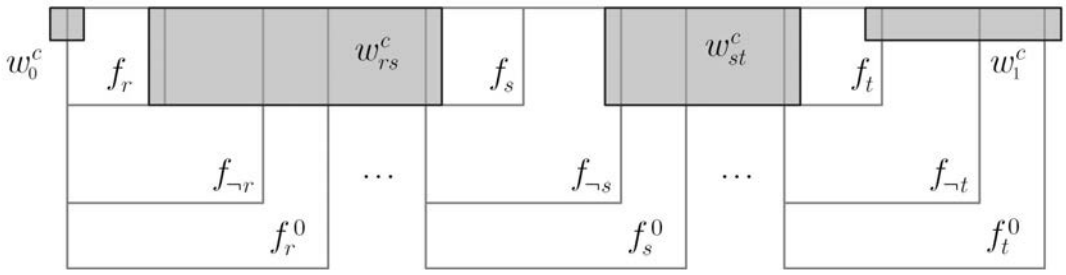


Figure 5.17: Configurations of $w_0^c, w_{rs}^c, w_{st}^c, w_1^c$ for clause $c = p_r \vee \neg p_s \vee p_t$.

Figure 5.18 examines the possible position of $u_r^* = u_r$ and $u_s^* = u_{\neg s}$.

If u_r is horizontally instantiated, then the gap between u_r and $u_{\neg s}$ is certainly bridged by w_{rs}^c (Figure 5.18(a)); if u_r is vertically instantiated, then it is possible to make u_r "thin" enough so that the gap between u_r and $u_{\neg s}$ is maintained (Figure 5.18(b)). Similar results hold for $u_{\neg s}$ (see Figure 5.18(c,d) for illustration). In case u_r, $u_{\neg s}$, and u_t are all horizontally instantiated, then there is no gap between any consecutive two of the seven regions in X_c. Otherwise, if any of u_r, $u_{\neg s}$, and u_t is vertically instantiated, then it is possible to maintain some gap.

Combining these with the constraints in Eq.s 5.51-5.53 involving the spatial variable v_c, we are now ready to introduce Γ_c.

Definition 5.15 *The basic CDC network Γ_c for $c = p_r^* \vee p_s^* \vee p_t^*$ has exactly the basic CDC constraints, explicitly or implicitly, in Γ_V (see Definition 5.14), and those in Eq.s 5.51-5.58.*

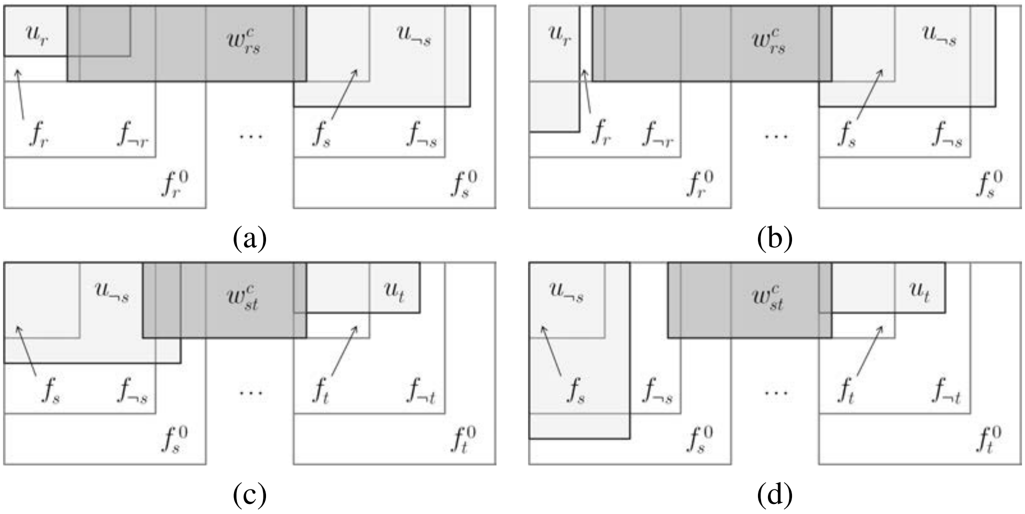


Figure 5.18: Possible configurations of u_r and $u_{\neg s}$: (a) u_r is horizontally instantiated; (b) u_r is vertically instantiated; (c) $u_{\neg s}$ is horizontally instantiated; (d) $u_{\neg s}$ is vertically instantiated.

Note that two new parallel relations are introduced in Γ_c. The spatial variable set of Γ_c includes those in Γ_V, and $v_c, w^c_0, w^c_{rs}, w^c_{st}, w^c_1$, and two auxiliary variables for constructing parallel relations.

Proposition 5.12 *Suppose Γ_c is the basic CDC network defined for clause $c \equiv p^*_r \vee p^*_s \vee p^*_t$. In any solution of Γ_c, if u^*_r (u^*_s, u^*_t, resp.) is horizontally instantiated, then its mbr bridges the gap between* $\mathbf{mbr}(w^c_0)$ *(*$\mathbf{mbr}(w^c_{rs})$*,* $\mathbf{mbr}(w^c_{st})$*, resp.) and* $\mathbf{mbr}(w^c_{rs})$ *(*$\mathbf{mbr}(w^c_{st})$*,* $\mathbf{mbr}(w^c_1)$*, resp.).*

Proof From the constraints in Γ_c, we know that the top edges of the mbrs of $f_r, f_{\neg r}, u_r, u_{\neg r}$ are on the same line. If p^*_r is positive, then $u^*_r = u_r$ by definition (see Eq. 5.49), and f_r bridges the gap between $\mathbf{mbr}(w^c_0)$ and $\mathbf{mbr}(w^c_{rs})$ by the construction of Γ_c (see Eq.s 5.55 and 5.56). Assume u^*_r is horizontally instantiated, i.e., u_r is horizontally instantiated. Since $I_x(f_r)\ \mathsf{s}\ I_x(u_r)\ \mathsf{s}\ I_x(f_{\neg r})$, we know u^*_r also bridges the gap between $\mathbf{mbr}(w^c_0)$ and $\mathbf{mbr}(w^c_{rs})$. If p^*_r is negative, then u^*_r is $u_{\neg r}$ and $f_{\neg r}$ bridges the gap between $\mathbf{mbr}(w^c_0)$ and $\mathbf{mbr}(w^c_{rs})$. Assume u^*_r is horizontally instantiated, i.e., $u_{\neg r}$ is horizontally instantiated. Since $I_x(f_{\neg r})\ \mathsf{s}\ I_x(u_{\neg r})$, we still have u^*_r bridges the gap between $\mathbf{mbr}(w^c_0)$ and $\mathbf{mbr}(w^c_{rs})$. The cases for u^*_s and u^*_t can be similarly proved.

We now show that Γ_c has the following property.

Proposition 5.13 *Suppose Γ_c is the basic CDC network defined for clause $c = p_r^* \vee p_s^* \vee p_t^*$ ($r < s < t$). Assume, moreover, $\pi : V \to \{\textsf{true}, \textsf{false}\}$ is a truth assignment. Then the following statements are equivalent:*

- *π satisfies c, i.e., at least one of the three literals p_r^*, p_s^*, and p_t^* is* true *under π;*
- *Γ_c has a solution such that u_i is vertically instantiated in the solution iff $\pi(p_i) =$* true *for $i = r, s, t$.*

Proof We first prove that, if π does not satisfy c, then Γ_c has no solution in which u_i is vertically instantiated iff $\pi(p_i) = \textsf{true}$ for $i = r, s, t$. Or equivalently, if $\pi(p_i^*) = \textsf{false}$ for $i = r, s, t$, then Γ_c has no solution in which u_i^* is horizontally instantiated for $i = r, s, t$. We prove this statement by contradiction. Suppose Γ_c has a solution in which u_i^* is horizontally instantiated for all $i = r, s, t$. Then, by Proposition 5.12, u_r^* bridges the gap between $\mathbf{mbr}(w_0^c)$ and $\mathbf{mbr}(w_{rs}^c)$; u_s^* bridges the gap between $\mathbf{mbr}(w_{rs}^c)$ and $\mathbf{mbr}(w_{st}^c)$; and u_t^* bridges the gap between $\mathbf{mbr}(w_{st}^c)$ and $\mathbf{mbr}(w_1^c)$ (cf. Figure 5.16 (b)). Let

$$A_c = \mathbf{mbr}(v_c) \smallsetminus (\bigcup\{\mathbf{mbr}(x) : x \in X_c\}),$$

where X_c is defined as in Eq. 5.50. It is clear that $\mathbf{mbr}(A_c)$ is a proper subset of $\mathbf{mbr}(v_c)$. Because the interior of v_c is disjoint from $\mathbf{mbr}(x)$ ($x \in X_c$), we have $v_c \subseteq A_c$. This leads to a contradiction. Therefore, if π does not satisfy c, then Γ_c has no solution in which u_i is vertically instantiated iff $\pi(p_i) = \textsf{true}$ for $i = r, s, t$.

On the other hand, suppose π satisfies c. We construct a solution of Γ_c in which u_i is vertically instantiated iff $\pi(p_i) = \textsf{true}$ for $i = r, s, t$. Variables in Γ_V other than $u_i, u_{\neg i}$ ($1 \leq i \leq n$) are defined as the same in Example 5.6.

For each propositional variable p_i, we define u_i and $u_{\neg i}$ as follows (cf. Figure 5.12).

$$u_i = \begin{cases} [i, i+0.2] \times [0.5, 1], & \text{if } \pi(p_i) \text{ is } \textsf{true}, \\ [i, i+0.5] \times [0.8, 1], & \text{otherwise}, \end{cases} \tag{5.59}$$

$$u_{\neg i} = \begin{cases} [i, i+0.7] \times [0.6, 1], & \text{if } \pi(p_i) \text{ is } \textsf{true}, \\ [i, i+0.4] \times [0.3, 1], & \text{otherwise}. \end{cases} \tag{5.60}$$

For clause $c \equiv p_r^* \vee p_s^* \vee p_t^*$, we define the "pier" variables as follows.

$$w_0^c = [r - 0.05, r + 0.05] \times [0.9, 1], \tag{5.61}$$

$$w_{rs}^c = \begin{cases} [r + 0.25, s + 0.05] \times [0.7, 1], & \text{if } p_r^* = p_r, \\ [r + 0.55, s + 0.05] \times [0.7, 1], & \text{otherwise,} \end{cases} \tag{5.62}$$

$$w_{st}^c = \begin{cases} [s + 0.25, t + 0.05] \times [0.7, 1], & \text{if } p_s^* = p_s, \\ [s + 0.55, t + 0.05] \times [0.7, 1], & \text{otherwise,} \end{cases} \tag{5.63}$$

$$w_1^c = \begin{cases} [t + 0.25, t + 0.85] \times [0.9, 1], & \text{if } p_t^* = p_t, \\ [t + 0.55, t + 0.85] \times [0.9, 1], & \text{otherwise.} \end{cases} \tag{5.64}$$

For example, the solution constructed for clause $c = p_r \vee \neg p_s \vee p_t$ and truth assignment π:

$$\pi(p_r) = \mathsf{false}, \pi(p_s) = \mathsf{true}, \pi(p_t) = \mathsf{true},$$

is shown in Figure 5.19 (a) (regions $w_0^c, w_{rs}^c, w_{st}^c, w_1^c$ and v_c) and (b) (regions $u_r, u_{\neg r}$, $u_s, u_{\neg s}, u_t, u_{\neg t}$).

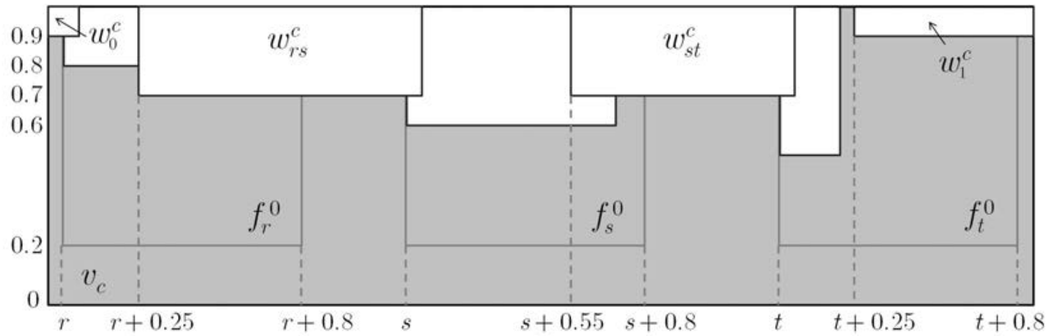
It is straightforward to verify that the three gaps between $w_0^c, w_{rs}^c, w_{st}^c, w_1^c$ are all bridged iff u_r^*, u_s^*, u_t^* are all horizontally instantiated. Let v_c be the region obtained by subtracting from $[r - 0.05, t + 0.85] \times [0, 1]$ the union of x ($x \in X_c$). Since the values of p_r^*, p_s^*, and p_t^* are not all false, we know that at least one of the gaps between the "pier" variables is maintained. This guarantees that the above instantiation of v_c satisfies all constraints in Γ_c involving v_c. Therefore, we have constructed a solution of Γ_c with the desired property.

As a corollary, we know in particular that Γ_c is consistent, and at least one of u_r^*, u_s^*, and u_t^* is vertically instantiated in any solution of Γ_c.

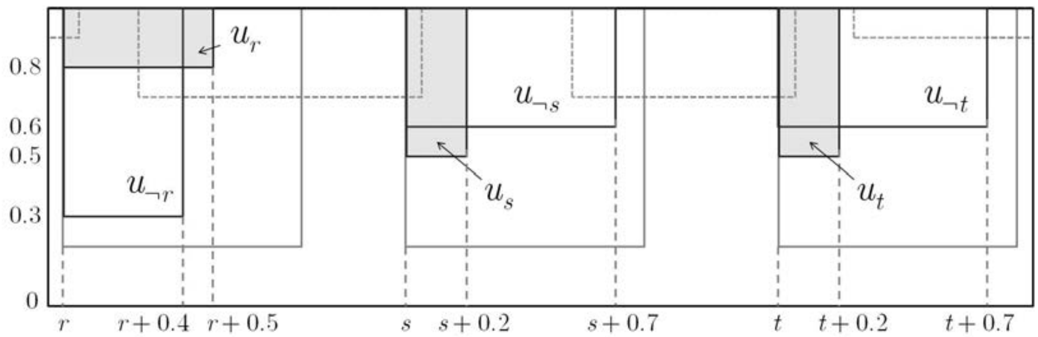
Definition 5.16 *For a 3-SAT instance $\phi = \bigwedge_{j=1}^m c_j$ over $V = \{p_1, \cdots, p_n\}$, we define Γ_{c_j} as the basic CDC network for clause c_j as in Definition 5.15, and define Γ_ϕ as the (incomplete) basic CDC network that is the union of all Γ_{c_j} ($1 \le j \le m$).*

We next show ϕ is satisfiable if Γ_ϕ is satisfiable.

Lemma 5.4 *Let ϕ be a 3-SAT instance and let Γ_ϕ be the basic CDC network of ϕ. If Γ_ϕ is satisfiable, then ϕ is also satisfiable.*



(a)



(b)

Figure 5.19: Illustration of solution for Γ_c: (a) regions $w_0^c, w_{rs}^c, w_{st}^c, w_1^c$ and v_c; (b) regions $u_r, u_{\neg r}, u_s, u_{\neg s}, u_t, u_{\neg t}$

Proof Let $\mathfrak{a}$ be a solution of Γ_ϕ. Define a truth assignment $\pi : \{p_1, \cdots, p_n\} \to \{\mathsf{true}, \mathsf{false}\}$ as: $\pi(p_i) = \mathsf{true}$ if and only if u_i is vertically instantiated in $\mathfrak{a}$. For each clause c of ϕ, we know $\mathfrak{a}$ is also a solution of Γ_c. By definition of π, we know in particular that u_i is vertically instantiated in $\mathfrak{a}$ if and only if $\pi(p_i)$ is true for $i = r, s, t$. By Proposition 5.13, π satisfies c. Due to the arbitrariness of c, we know π satisfies ϕ. Therefore, ϕ is satisfiable.

On the other hand, we show Γ_ϕ is satisfiable if ϕ is satisfiable.

Lemma 5.5 *Let ϕ be a 3-SAT instance and let Γ_ϕ be the basic CDC network of ϕ. If ϕ is satisfiable, then Γ_ϕ is also satisfiable.*

Proof Suppose $\pi : V \to \{\mathsf{true}, \mathsf{false}\}$ is a truth assignment. A solution for Γ_ϕ can be constructed by following exactly the same procedures as we have used in Proposition 5.13. Note that there are no direct constraints between variables in X_c and $X_{c'}$, where c, c' are two different clauses. We instantiate v_c as in Proposition 5.13. It is easy to see that the assignment satisfies Γ_c for each clause c of ϕ. Therefore, the network Γ_ϕ is also satisfiable.

As a consequence of the above results, we have

Theorem 5.5 *Deciding the consistency of incomplete basic CDC networks is an NP-hard problem.*

Proof We prove the NP-hardness of consistency checking of incomplete basic CDC networks by a reduction from 3-SAT. For each 3-SAT instance ϕ, we define an incomplete basic CDC network Γ_ϕ. Lemmas 5.4 and 5.5 show that the 3-SAT instance ϕ is satisfiable if and only if the basic CDC network Γ_ϕ is satisfiable. It is not hard to show that the number of variables in Γ_ϕ is linear to the number of variables and clauses in ϕ [1]. This shows that the size of Γ_ϕ is polynomial in the size of ϕ. So we have reduced 3-SAT in polynomial time to the consistency problem of incomplete basic CDC networks.

We have shown that CSPSAT($\mathcal{B}_{\mathbf{CDC}}$) has a cubic algorithm and hence is in P. Therefore, CSPSAT($\mathcal{B}^{\star}_{\mathbf{CDC}}$) is also also NP.

[1] In fact there are $14n + 7m + 4$ spatial variables and $9n + 2m$ auxiliary variables in Γ_ϕ, where n and m are the number of variables and clauses, respectively, of ϕ.

Corollary 5.1 *Deciding the consistency of an incomplete basic CDC network is NP-Complete.*

Remark 5.1 *Let*

$$\begin{aligned}\mathcal{S} = \{&O,\ E,\ W,\ E{:}O,\ W{:}O,\ S{:}O,\ E{:}SE{:}S,\ W{:}SW{:}S,\\ &E{:}SE{:}S{:}O,\ W{:}SW{:}S{:}O,\ E{:}SE{:}S{:}SW{:}W\}.\end{aligned} \tag{5.65}$$

It is worth noting that, in the reduction, only the universal relation and the 11 basic CDC relations in $\mathcal{S}$ are used to construct constraints in the network Γ_ϕ. Therefore, we actually have proved that ***RSAT***$(\mathcal{S} \cup \{?\})$ *is NP-hard.*

5.7 The Cardinal Direction Calculus over Disconnected Regions CDC_d

The CDC introduced above requires regions to be connected. As proposed in [99], a variant of CDC, denoted by CDC_d, allows regions to be disconnected. In this section, we show the consistency result obtained in previous sections also hold for this variant, viz., $\textsc{cspsat}(\mathcal{B}_{\mathbf{CDC}_d})$ can be solved in cubic time and $\textsc{cspsat}(\mathcal{B}^{\star}_{\mathbf{CDC}_d})$ is NP-complete.

The domain of CDC_d is the set of all bounded regions, which are allowed to be disconnected. For such regions, the *cardinal direction* can be defined in the same as in Equation 5.1. However, there are 511 valid matrices, i.e., all 3×3 Boolean matrix except the one with all entries 0 are valid. That is to say, CDC_d has 511 basic relations. We discuss $\textsc{cspsat}(\mathcal{B}_{\mathbf{CDC}_d})$ and $\textsc{cspsat}(\mathcal{B}^{\star}_{\mathbf{CDC}_d})$ in separate subsections.

5.7.1 Cubic Algorithm for $\textsc{cspsat}(\mathcal{B}_{\mathbf{CDC}_d})$

The approach for dealing with CDC_d is the same as that for CDC: we establish the connection of CDC_d and IA, show that each consistent complete basic CDC_d network has a maximal canonical solution, and devise a cubic algorithm which tries to find the maximal canonical solution of an input network.

The definition of projective interval relations is the same, although the mapping table in Proposition 5.1 needs a new column for (d_1, d_2, d_3) to be $(1, 0, 1)$ and $\iota^x(\delta) =$

di. This is because, CDC_d permits more valid matrices and (d_1, d_2, d_3) is possibly to be $(1, 0, 1)$. Theorem 5.1 also holds for CDC_d, i.e., a basic complete CDC_d network Γ is consistent only if its x- and y-projective IA networks Γ_x and Γ_y are both consistent.

We then define regular solutions, meet-free solutions, and canonical solutions for CDC_d in the same way as for CDC. Moreover, a general solution of a complete basic CDC_d network can be transformed into a canonical solution as we have done in Section 5.3. therefore, a consistent complete basic CDC_d network always has a canonical solution (as Proposition 5.7 for CDC). However, the maximal canonical solution in CDC_d differs slightly.

Lemmas 5.1 and 5.2 (with their proof) still hold in CDC_d. We here repeat statements. Suppose $\Gamma = \{v_i \delta_{ij} v_j\}_{i,j=1}^n$ is a consistent basic complete CDC_d network, with Γ_x and Γ_y respectively being its x- and y-projective IA networks. Let $\{I_i\}_{i=1}^n$ and $\{J_i\}_{i=1}^n$ be the canonical solutions of Γ_x and Γ_y respectively, and $m_i = I_i \times J_i$. Then for any canonical solution $\mathfrak{a} = \{a_i\}_{i=1}^n$ of Γ, we have $\mathbf{mbr}(a_i) = m_i$. Furthermore, define D_i and b_i the same way as in Equations 5.16 and 5.17,

$$
\begin{aligned}
D_i &:= \{p_{st} \subseteq m_i : (\exists\ j, \chi) \text{ such that } p_{st} \subseteq \chi(m_j) \text{ and } d_{ij}^{\chi} = 0\}, \\
b_i &:= \bigcup \{p_{st} \subseteq m_i : p_{st} \notin D_i\}.
\end{aligned}
$$

The following theorem which asserts that $\mathfrak{b} = \{b_i\}_{i=1}^n$ is the maximal canonical solution of Γ.

Theorem 5.6 *Suppose $\Gamma = \{v_i \delta_{ij} v_j\}_{i,j=1}^n$ is a complete basic CDC_d network. If $\mathcal{N}$ is consistent, then $\mathfrak{b} = \{b_i\}_{i=1}^n$ as defined above, is a canonical solution of Γ. Furthermore, it is the maximal canonical solution in the sense that for any canonical solution $\mathfrak{a} = \{a_i\}_{i=1}^n$ of Γ, we have $a_i \subseteq b_i$.*

The theorem can be proved similarly to Lemma 5.3. Note that we *do not* need to compute the maximal connected component of b_i, as we no longer require regions in the solution to be connected.

We next adapt Algorithm 4 for CDC_d. Recall that Algorithm 4 basically contains five steps:

1. Compute projective IA networks;

2. Compute canonical interval solutions;

3. Compute b_i;

4. Compute c_i;

5. Verify whether $\{c_i\}_{i=1}^n$ is a solution.

The first three steps do not need modification[1]. Step 4 is no longer necessary because $\{b_i\}_{i=1}^n$ would be the maximal canonical solution if the input network is consistent. The last step should consequently use $\{b_i\}_{i=1}^n$ instead of $\{c_i\}_{i=1}^n$. Therefore, the procedure for CDC_d should be as follows.

1. Compute projective IA networks;

2. Compute canonical interval solutions;

3. Compute b_i;

4' Verify whether $\{b_i\}_{i=1}^n$ is a solution.

Recall that when we verify $\{c_i\}_{i=1}^n$ in Algorithm 4, we have made use of the connectedness of c_i to get a cubic complexity. This suggests we need a new subroutine here for CDC_d. To verify whether $\mathsf{dir}(b_i, b_j) = \delta_{ij}$ holds for each pair of i, j, we need to check whether $b_i \cap \chi(m_j)$ contains a pixel iff $d_{ij}^{\chi} = 1$, for $1 \leq i, j \leq n, \chi \in \textsc{TileName}$. The intuitive idea is to check all the pixels in b_i, pixel by pixel. However, as b_i contains $O(n^2)$ pixels, this intuitive approach would result $O(n^4)$ time complexity of Step 4'. We next show that this can be also refined to $O(n^3)$.

Recall that a digital region b contained in $T = [0, n_x] \times [0, n_y]$ is represented by the $n_x \times n_y$ Boolean matrix $B(b)$. We now define a new matrix representation $M(b)$ for b, where for each $0 \leq k < n_x$ and $0 \leq l < n_y$,

$$M(b)[k,l] = \sum \{B(b)[p,q] : 0 \leq p \leq k, 0 \leq q \leq l\}. \tag{5.66}$$

That is to say, $M(b)[k,l]$ is the number of pixels contained in $([0,k+1] \times [0,l+1]) \cap b$. Note that $M(b)$ is an $n_x \times n_y$ *integer* matrix. Given the Boolean matrix $B(b)$ for a digital region b, $M(b)$ can be computed in $O(n_x \times n_y) = O(n^2)$ time by iteratively adding the

[1]As mentioned above, the computation of projective IA networks slightly differs, as the case that $(d_1, d_2, d_3) = (1, 0, 1)$ should be considered.

k-th column to the $(k+1)$-th, and then iteratively adding the p-th row to the $(p+1)$-th. Therefore, all $M(b_i)$ can be computed in cubic time.

The following proposition shows that $M(b)$ can accelerate the calculation of the number of pixels contained in b and an arbitrary rectangle.

Proposition 5.14 *Let $r = [x^-, x^+] \times [y^-, y^+]$ be a rectangle and let b be a digital region, both contained in $T = [0, n_x] \times [0, n_y]$. Then $b \cap r$ contains a pixel iff*

$$M(b)[x^- - 1, y^- - 1] + M(b)[x^+ - 1, y^+ - 1] > M(b)[x^+ - 1, y^- - 1] + M(b)[x^- - 1, y^+ - 1], \tag{5.67}$$

where $M(b)[-1, l] = M(b)[k, -1] = 0$.

Proof Because $M(b)[k, l]$ denotes the number of pixels contained in $([0, k+1] \times [0, l+1]) \cap b$, the number of pixels in $([x^-, x^+] \times [y^-, y^+]) \cap b$ is $M(b)[x^+ - 1, y^+ - 1] - (M(b)[x^- - 1, y^+ - 1] + M(b)[x^+ - 1, y^- - 1]) + M(b)[x^- - 1, y^- - 1]$. The conclusion follows directly.

Therefore, given $M(b_i)$, it takes constant time to decide whether $b_i \cap \chi(m_j)$ contains a pixel for each χ with $d^{\chi}_{ij} = 1$. This implies that whether $\mathsf{dir}(b_i, b_j) = \delta_{ij}$ can be checked in $O(n)$ time. This implies whether $\{b_i\}_{i=1}^{n}$ is a solution of Γ can be decided in cubic time, provided $M(b_i)$ are given. Recall that all $M(b_i)$ can also be obtained in cubic time. Therefore, the time complexity of Step 5, and hence the whole algorithm, is $O(n^3)$ time.

Theorem 5.7 *The consistency of a complete basic CDC_d constraint network can be decided in cubic time. If the network is consistent, then its maximal canonical solution may also be computed in cubic time.*

5.7.2 NP-hardness for CSPSAT$(\mathcal{B}^{\star}_{\mathbf{CDC}_d})$

Now we discuss the consistency problem CSPSAT$(\mathcal{B}^{\star}_{\mathbf{CDC}_d})$ in CDC_d. Note that the reduction for CSPSAT$(\mathcal{B}^{\star}_{\mathbf{CDC}})$ does not require regions to be connected. Therefore, the same reduction is also applicable to CDC_d, which implies that CSPSAT$(\mathcal{B}^{\star}_{\mathbf{CDC}_d})$ is NP-hard. Because we have shown that CSPSAT$(\mathcal{B}_{\mathbf{CDC}_d})$ is in P, we have the following theorem.

Theorem 5.8 *Deciding the consistency of incomplete basic CDC_d networks is NP-complete.*

This suggests that the $O(n^5)$ algorithm Consistency devised in [99] for determining the consistency of basic CDC_d networks is incomplete. We now present a brief discussion for the algorithm Consistency.

Suppose Γ is a possibly incomplete basic CDC_d network. Consistency first transforms constraints in Γ into a network Γ_{pa} of Point Algebra (PA) constraints (which is possibly also incomplete). It then calls the Cspan algorithm of van Beek [104] to compute a solution of Γ_{pa} and transforms the solution into a maximal one (see [99, Definition 10 and Theorem 2] for the details). The algorithm then returns "consistency" if this particular maximal solution satisfies a property called the "NTB property" [99, Section 4.3], and returns "inconsistency" otherwise.

Because the PA network Γ_{pa} is possibly incomplete, it may have exponentially many different (maximal) solutions.[1] As a polynomial algorithm, Cspan returns only one solution of Γ_{pa}. It has been proved [99, Theorem 3] that Γ is consistent if and only if Γ_{pa} has *a* maximal solution which satisfies the NTB property. However, for some PA network Γ_{pa}, only *some*, instead of all, maximal solutions satisfy the NTB property. So to assure that Γ is inconsistent, we need to try all different maximal solutions of Γ_{pa}, which may take exponential time. Consistency, however, checks the NTB property for only one maximal solution (i.e., the one constructed based on the result of Cspan). This explains why it is an incomplete algorithm for checking the consistency of basic CDC_d networks.

Take the inconsistent basic network in [99, Example 13] as an example. This network is defined as

$$\Gamma = \{x\ N{:}E{:}O\ y, x\ O{:}S{:}W\ z, y\ S\ W\ z\}. \tag{5.68}$$

The inconsistency of Γ is detected by Cspan. In fact, Consistency first transforms Γ into a network Γ_{pa} of PA constraints, and then computes a maximal solution of Γ_{pa} based on the result of Cspan with input Γ_{pa}. Write $\mathfrak{m}$ for this maximal solution. Consistency finally returns "inconsistency" after finding that $\mathfrak{m}$ does not satisfy the

[1] Two solutions of a PA network are regarded as different if the orderings of the points in the two solutions are different.

NTB property. So far so good. Let Γ' be the network obtained by removing the third constraint $(y\ SW\ z)$ from Γ. It is easy to see that Γ' is consistent. For this network, Consistency first computes a set Γ'_{pa} of PA constraints, which is a subset of Γ_{pa}, and then tries to find a maximal solution of Γ'_{pa}, which could possibly be $\mathfrak{m}$. If this happens, the algorithm would return "inconsistency" for Γ' because $\mathfrak{m}$ does not satisfy the NTB property. This is, however, incorrect.

Even so, the algorithm in [99] works correctly for *complete* basic CDC networks. This is because, for a complete basic CDC network, the maximal solution (if any) of the PA network Γ_{pa} is unique, and Consistency then checks whether this unique maximal solution satisfies the NTB property. If the solution does not satisfy the NTB property, Consistency would correctly return "inconsistency", because there is no other maximal solution which can satisfy the NTB property; in fact, there is no other maximal solution at all.

5.8 Chapter Summary

In this chapter, we have proposed a cubic algorithm that decides the consistency of complete basic CDC constraint networks, and then proved that deciding the consistency of incomplete basic CDC networks is an NP-hard problem. These results draw a sharp boundary between the tractable and intractable subclasses of the CDC. The CDC is the first known qualitative calculus in which reasoning with conjunctive constraints is NP-hard, while reasoning with explicit constraints is in P. These results are also applicable to CDC_d, the cardinal direction calculus for possibly disconnected regions. This suggests that the $O(n^5)$ algorithm in [99] is incomplete for checking the consistency of basic CDC networks. Future work will consider approximating methods for solving the consistency decision problem in the CDC.

Chapter 6

Landmarks and Restricted Domains

6.1 Introduction

Landmarks (e.g., Sydney Opera House or Big Ben), are outstanding features in an environment [111], or visual points of reference that enhance the image ability of urban spaces [68]. Research has shown that people prefer referring to landmarks in most cases [102], e.g., in navigation or route planning. In fact, computer-based navigation systems (e.g., Google Maps) are trying to integrate landmarks into traditional frameworks to provide more effective and user-friendly interfaces.

Although the QSTR has been increasingly influential in the past, there is a consensus that breakthroughs are necessary to bring spatial/temporal reasoning theory closer to practical applications. One reason might be that the language used in the qualitative reasoning scheme is somewhat restricted: constraints in a qualitative CSP only relate variables that can be interpreted by arbitrary elements from the universe, and thus they cannot express landmarks properly. Consider the following example. Suppose you are recommended a restaurant in Sydney by a friend who has dined there before. The spatial information about the restaurant may be similar to "it is *in* downtown and *close to* a MacDonald's, and it is to the *west of* or *southwest of* Central Station." In this example, topological, directional, and distance information appears together. Furthermore, while the position of the restaurant may be totally unknown, the position of Central Station is fixed as a landmark, and the position of downtown is also fixed somehow, but the position of "MacDonald's" is only *finitely fixed* because there are several branches of

MacDonald's in downtown Sydney.

This chapter extends the framework of qualitative constraint satisfaction problems by allowing variables to be *finitely restricted*, i.e., a variable can be interpreted by an entity from a finite subset of the universe in a qualitative calculus. In particular, a variable is called a *landmark* if it can only be interpreted by one fixed entity in the universe. In other words, a landmark is a symbol representing a constant entity in the universe. An important question is, *how does this extension (i.e., introducing finitely restricted variables, or landmarks) affect the computational complexity of deciding the consistency of qualitative CSPs*? This chapter answers this question for several of the most important qualitative calculi, viz. Point Algebra (PA), Interval Algebra (IA), Cardinal Relation Algebra (CRA), and RCC-5/RCC-8, and is organised as follows. Section 2 formalises the extended consistency problem and provides a summary of novel results. Section 3 discusses the point-based qualitative calculi PA, IA and CRA. For PA, we show that the extended consistency problem is still tractable. For IA and CRA, because the classical consistency problem is already NP-hard in general, we focus on basic networks. It will be shown that, for basic IA or CRA networks introduced with landmarks (finitely restricted variables resp.), the consistency problem is still tractable (NP-complete resp.). Section 4 studies the consistency problems in the standard RCC-5/RCC-8 models (viz., interpreted in the 2-dim Euclidean topology), including one subsection for background knowledge in computational geometry. For RCC-5, the complexity of the extended consistency problem is similar to that of IA and CRA. For RCC-8, it will be shown that the extended consistency problem is NP-hard even in the most simple case (i.e., basic networks involving only landmarks).

6.2 Formalisation and Summary of Results

As discussed in the preliminary chapter, there are two differences between the consistency problem in QSTR and the classical CSP problem:

- The consistency problem in QSTR assumes that every variable may take *any* value in the universe, while the classical CSP specifies for each variable a domain of its own.

- The constraints in the consistency problem in QSTR are restricted to the relations in the qualitative calculus.

The first assumption is quite a restriction of the expressive power of the consistency problem in QSTR, but it can be generalised in a direct way, as shown in the following definition.

Definition 6.1 *Let $\mathcal{M}$ be a qualitative calculus on universe $\boldsymbol{U}$. Suppose $\mathcal{S}$ is a subset of $\mathcal{M}$. The* extended consistency problem*, denoted by* $\textsc{cspsat}_{\mathcal{M},f}(\mathcal{S})$*, is defined as follows, where the subscript 'f' stands for 'finite':*

> ***Instance****: A 3-tuple $(V,\Gamma,\boldsymbol{D})$. Here V is a finite set of variables $\{v_1,v_2,\cdots,v_n\}$, $\boldsymbol{D}$ is an n-tuple $(D_1,D_2,\cdots,D_n)$ where each D_i is either the universe $\boldsymbol{U}$, or a nonempty finite subset of $\boldsymbol{U}$, and Γ is a constraint network with respect to variables in V.*
>
> ***Question****: Is there an interpretation $\mathcal{I}: V \to \boldsymbol{U}$ such that (i) $\mathcal{I}(v_i) \in D_i$ for each v_i in V, and (ii) all constraints in Γ are satisfied?*

We say variable v_i appearing in such an instance is finitely restricted *if its domain D_i is finite. In particular, if D_i contains only one element, we say v_i is* uniquely restricted *or a* landmark*. If interpretation $\mathcal{I}$ satisfies all constraints in Γ and $\mathcal{I}(v_i) \in D_i$ for each i, then we say $\mathcal{I}$ is a solution of $(V,\Gamma,\boldsymbol{D})$ and say $(V,\Gamma,\boldsymbol{D})$ is* consistent *or* satisfiable*.*

As a special case, if each D_i is either the universe $\boldsymbol{U}$ or a singleton, we write the corresponding consistency problem $\textsc{cspsat}_{\mathcal{M},s}(\mathcal{S})$, where the subscript 's' denotes 'singleton'.

Notation If the qualitative calculus $\mathcal{M}$ is clear from the context, we simply write $\textsc{cspsat}_f$ and $\textsc{cspsat}_s$ for $\textsc{cspsat}_{\mathcal{M},f}$ and $\textsc{cspsat}_{\mathcal{M},s}$ respectively.

We also call $\textsc{cspsat}_{\mathcal{M}}(\mathcal{S})$ the *classic consistency problem* to distinguish it from the extended consistency problem. An instance of the classic consistency problem $\textsc{cspsat}_{\mathcal{M}}(\mathcal{S})$ is clearly also an instance of $\textsc{cspsat}_{\mathcal{M},f}(\mathcal{S})$ or $\textsc{cspsat}_{\mathcal{M},s}(\mathcal{S})$: We only need to let all D_i be the universe. We have the following general results.

Proposition 6.1 *Suppose $\mathcal{M} = (\boldsymbol{U},\mathcal{B}^*)$ is a qualitative calculus with basic relation set $\mathcal{B}$, and $\mathcal{S}$ is a subset of $\mathcal{B}^*$. Then we have*

i) $\textsc{cspsat}(\mathcal{S}) \subset \textsc{cspsat}_s(\mathcal{S}) \subset \textsc{cspsat}_f(\mathcal{S})$*;*

ii) $\textsc{cspsat}_f(\mathcal{S})$ *is in NP if* $\textsc{cspsat}_f(\mathcal{B})$ *is in NP;*

iii) $\textsc{cspsat}_f(\mathcal{S})$ *is in NP if* $\textsc{cspsat}_s(\mathcal{S})$ *is in NP.*

Proof (i) follows directly from the definition. As for (ii), suppose we already have a nondeterministic Turing machine T_0 which solves $\textsc{cspsat}_f(\mathcal{B})$ in polynomial time. Given a non-basic constraint network $(V, \Gamma, \mathbf{D})$, it is consistent iff there is a consistent basic constraint network Γ' that refines Γ in the sense that for each constraint $x\alpha y$ in Γ there exists a constraint $x\alpha' y$ in Γ' such that $\alpha' \subseteq \alpha$. So we devise a nondeterministic Turing machine T as follows. T first guesses a refinement $(V, \Gamma', \mathbf{D})$ of $(V, \Gamma, \mathbf{D})$, and then calls T_0 to decide the consistency of $(V, \Gamma', \mathbf{D})$. T' asserts the instance to be consistent if T returns consistent in any branch. It is clear that the nondeterministic Turing machine T decides the consistency of $(V, \Gamma, \mathbf{D})$ in polynomial time. Similar argument applies to (iii).

By the above proposition, the computational complexity of $\textsc{cspsat}_f$ is in general higher than that of $\textsc{cspsat}_s$ and $\textsc{cspsat}$, as far as the same subset $\mathcal{S}$ of the same calculus is considered. In particular, recall that the classic consistency problems for CRA, IA, RCC5 and RCC8 are all NP-complete. We have the following corollary.

Corollary 6.1 *Suppose qualitative calculus $\mathcal{M}$ is one of IA, CRA, RCC-5, and RCC-8. Then the extended consistency problems* $\textsc{cspsat}_{\mathcal{M},s}(\mathcal{B}^*_{\mathcal{M}})$ *and* $\textsc{cspsat}_{\mathcal{M},f}(\mathcal{B}^*_{\mathcal{M}})$ *are both NP-hard.*

Table 6.1 provides a summary of complexity results of $\textsc{cspsat}_{\mathcal{M}}(\mathcal{S})$, $\textsc{cspsat}_{\mathcal{M},s}(\mathcal{S})$ and $\textsc{cspsat}_{\mathcal{M},f}(\mathcal{S})$, where qualitative calculus $\mathcal{M}$ is Point Algebra, Cardinal Relation Algebra, Interval Algebra, RCC-5 or RCC-8, and $\mathcal{S}$ is either $\mathcal{B}_{\mathcal{M}}$ or $\mathcal{B}^*_{\mathcal{M}}$ (i.e., we consider the basic networks and the most general case). Original nontrivial results are underlined.

In the following sections, we first consider point-based calculi PA, CRA, and IA, and then consider region-based calculi RCC5 and RCC8. Unlike point-based calculi, the *geometrical representation* (in particular, shape and location) of the landmarks may affect the existence of solutions in the plane. To make the analysis more meaningful,

$\mathcal{M}$	PA		CRA, IA		RCC-5		RCC-8	
$\mathcal{S}$	$\mathcal{B}_{\mathbf{PA}}$	$\mathcal{B}^*_{\mathbf{PA}}$	$\mathcal{B}_{\mathcal{M}}$	$\mathcal{B}^*_{\mathcal{M}}$	$\mathcal{B}_{\mathbf{RCC5}}$	$\mathcal{B}^*_{\mathbf{RCC5}}$	$\mathcal{B}_{\mathbf{RCC8}}$	$\mathcal{B}^*_{\mathbf{RCC8}}$
$\text{CSPSAT}_{\mathcal{M}}(\mathcal{S})$	P	P	P	NP-C	P	NP-C	P	NP-C
$\text{CSPSAT}_{\mathcal{M},s}(\mathcal{S})$	P	P	P	NP-C	$\underline{\text{P}}$	NP-C	$\underline{\text{NP-C}}$	NP-C
$\text{CSPSAT}_{\mathcal{M},f}(\mathcal{S})$	$\underline{\text{P}}$	$\underline{\text{NP-C}}$	$\underline{\text{NP-C}}$	NP-C	$\underline{\text{NP-C}}$	NP-C	NP-C	NP-C

Table 6.1: Computational complexity results summary

we assume that all the landmarks in RCC5 and RCC8 constraint networks are represented as polygons which may have different connected components and holes. This assumption is practical because polygons are the most widely used approximations of regions in spatial databases.

6.3 Point-based Qualitative Calculi

This section discusses the extended problems for point-based qualitative calculi, viz., the Point Algebra, Interval Algebra, and Cardinal Relation Algebra.

6.3.1 Some simple results

To prove the computational complexity results, we will need the following convention for specifying instances and the notion of a *finitely restricted sub-instance*.

Notation Suppose $(X, \Gamma, \mathbf{D})$ is an instance of $\text{CSPSAT}_{\mathcal{M},f}(\mathcal{S})$, where $X = \{x_1, x_2, \cdots, x_t\}$, $\mathbf{D} = (D_1, \cdots, D_t)$, and $\Gamma = \{x_i \alpha_{ij} x_j\}$. Let U be the set of variables in X the domains of which are finitely restricted, i.e., $U = \{x_i \in X : D_i \text{ is a finite subset of } \mathbf{U}_{\mathcal{M}}\}$. Moreover, denote $V = X \setminus U$, and $\mathbf{D}_U = \{D_i : x_i \in U\}$. Then the instance $(X, \Gamma, \mathbf{D})$ is also specified by or referred as $(V \uplus U, \Gamma, \mathbf{D}_U)$.

Definition 6.2 *Suppose* $(V \uplus U, \Gamma, \boldsymbol{D}_U)$ *is an instance of* $\text{CSPSAT}_{\mathcal{M},f}(\mathcal{S})$*. Let* Γ_U *be the restriction of* Γ *with respect to* U*, i.e.,* $\Gamma_U = \{u_i \alpha_{ij} u_j : u_i \alpha_{ij} u_j \in \Gamma, u_i, u_j \in U\}$*. Then* $(U, \Gamma_U, \boldsymbol{D}_U)$ *is also an instance of* $\text{CSPSAT}_{\mathcal{M},f}(\mathcal{S})$*, and we call it the* finitely restricted sub-instance *of* $(V \uplus U, \Gamma, \boldsymbol{D}_U)$.

For basic constraint networks in PA, IA and CRA, we have the following general results.

Lemma 6.1 *Let qualitative calculus $\mathcal{M}$ be one of PA, IA, and CRA. Suppose* $(V \uplus U, \Gamma, \boldsymbol{D}_U)$ *is an instance of* $\textsc{cspsat}_{\mathcal{M},f}(\mathcal{B}_\mathcal{M})$*, where* $V = \{v_1, v_2, \cdots, v_n\}$*,* $U = \{u_1, u_2, \cdots, u_m\}$*,* $\boldsymbol{D}_U = (D_i, \cdots, D_m)$*, ($D_j$ is a finite subset of $\boldsymbol{U}_\mathcal{M}$). Then* $(V \uplus U, \Gamma, \boldsymbol{D}_U)$ *is consistent iff (i) its the finitely restricted sub-instance* $(U, \Gamma_U, \boldsymbol{D}_U)$ *is consistent, and (ii)* Γ *is path-consistent.*

Proof The necessity of conditions (i) and (ii) is clear. We next prove the sufficiency for Point Algebra.

Because the finitely restricted sub-instance $(U, \Gamma_U, \mathbf{D}_U)$ is consistent, it has a solution, say $\mathfrak{b} = (b_1, \cdots, b_m)$. Note that $\mathfrak{b}$ is a partial solution of $\textsc{cspsat}(\mathcal{B}_{\mathbf{PA}})$ instance $(V \cup U, \Gamma)$, and thus can be extended to a solution $\mathfrak{b}'$ of $(V \cup U, \Gamma)$. It can be directly proved that $\mathfrak{b}'$ is also a solution of $(V \uplus U, \Gamma, \mathbf{D}_U)$. Therefore $(V \uplus U, \Gamma, \mathbf{D}_U)$ is consistent.

The cases for IA and CRA can be proved in the same method.

The following conclusion follows directly.

Corollary 6.2 *The extended consistency problems* $\textsc{cspsat}_{\mathcal{M},s}(\mathcal{B}_\mathcal{M})$ *are in P for qualitative calculus $\mathcal{M}$ being one of PA, IA, and CRA.*

Proof The path-consistency of Γ. can be checked in $O((n+m)^3)$ time. The consistency of $(U, \Gamma_U, \mathbf{D}_U)$ can be decided in $O(m^2)$ time, as we only need to check for each pair of variables u_i and u_j in U whether the values specified for them, say, l_i and l_j respectively, satisfy the constraint between them. Therefore, $\textsc{cspsat}_{\mathcal{M},s}(\mathcal{B}_\mathcal{M})$ is in P.

Lemma 6.2 *Let qualitative calculus $\mathcal{M}$ be one of PA, IA, and CRA. Let $\mathcal{S}$ be a tractable subclass of $\mathcal{M}$ which contains all the basic relations. Then* $\textsc{cspsat}_{\mathcal{M},s}(\mathcal{S})$ *is tractable.*

Proof We prove the case of PA. Suppose $(V \uplus U, \Gamma, \mathbf{D}_U)$ is an instance of $\textsc{cspsat}_s(\mathcal{S})$, where $U = \{u_1, \cdots, u_m\}$. Suppose landmark u_i can only be assigned with value l_i. Furthermore, suppose the relation between l_i and l_j is β_{ij}, and we then replace the constraint about u_i and u_j in Γ by $u_i \beta_{ij} u_j$, for all pairs u_i and u_j. Denote the obtained constraint network by Γ'. Note that $(V \uplus U, \Gamma', \mathbf{D}_U)$ is still an instance of $\textsc{cspsat}_s(\mathcal{S})$, since $\mathcal{S}$ contains all the basic relations. Clearly $(V \uplus U, \Gamma, \mathbf{D}_U)$ is consistent iff (i) the constraint $u_i \alpha u_j$ about u_i and u_j in Γ satisfies $(\beta_{ij} \subseteq \alpha$, and (ii) $(V \uplus U, \Gamma, \mathbf{D}_U)$ is consistent.

We claim that, $(V \uplus U, \Gamma', \mathbf{D}_U)$ is consistent iff $(V \cup U, \Gamma')$ is consistent. The necessity is obvious. For the sufficiency, suppose $(V \cup U, \Gamma')$ has a solution $\mathfrak{a} = (a_1, \cdots, a_n, b_1, \cdots, b_m)$ where $v_i \in V$ is assigned with a_i and $u_j \in U$ is assigned with b_j. We now transform $\mathfrak{a}$ to make it consistent with the fix values specified for landmarks. W.o.l.g., suppose $s < b_1 \leq b_2 \leq \cdots \leq b_m < t$ (this can be achieved by renaming variables in U), and b_1 is larger than any l_j (this can be achieved by adding a sufficient large number to all a_i and b_j). For each a_i, define a'_i as follows. If there exists some b_j such that $a_i = b_j$, then $a'_i = l_j$. Otherwise, there exists some t such that $b_t < a_i < b_{t+1}$ (supposing $b_0 = s$ and $b_{m+1} = t$), and a'_i is defined as $l_t + (a_i - b_t) \times (l_{t+1} - l_t)/(b_{t+1} - b_t)$ (supposing $l_0 = l_1 - 1$ and $l_{m+1} = l_m + 1$). It is clear that $\mathfrak{a}$ and $\mathfrak{a}' = (a'_1, \cdots, a'_n, l_1, \cdots, l_m)$ respect the same ordering. Therefore $\mathfrak{a}'$ is also a solution of $(V \cup U, \Gamma')$, and hence solution of $(V \uplus U, \Gamma', \mathbf{D}_U)$.

Therefore, $(V \uplus U, \Gamma, \mathbf{D}_U)$ is consistent iff (i) the constraint $u_i \alpha u_j$ about u_i and u_j in Γ satisfies $(\beta_{ij} \subseteq \alpha$, and (ii) $(V \cap U, \Gamma')$ is consistent. The first condition can be checked in $O(m^2)$ time. The second condition can be decided in polynomial time, because $(V \cap U, \Gamma')$ is an instance of $\textsc{cspsat}_{\mathbf{PA}}(\mathcal{S})$ and $\mathcal{S}$ is tractable. Therefore, $\textsc{cspsat}_{\mathbf{PA},s}(\mathcal{S})$ is tractable.

The cases of IA and CRA can be similarly proved.

Because $\textsc{cspsat}_{\mathbf{PA}}(\mathcal{B}^*_{\mathbf{PA}})$ is tractable, we have the following result.

Corollary 6.3 $\textsc{cspsat}_{\boldsymbol{PA},s}(\mathcal{B}^*_{\boldsymbol{PA}})$ *is also tractable.*

For qualitative calculus $\mathcal{M}$ being IA, CRA, the NP-completeness of $\textsc{cspsat}_{\mathcal{M}}(\mathcal{B}^*_{\mathcal{M}})$ and the tractability of $\textsc{cspsat}_{\mathcal{M},s}(\mathcal{B}_{\mathcal{M}})$ imply that $\textsc{cspsat}_{\mathcal{M},f}(\mathcal{B}^*_{\mathcal{M}})$ is NP-complete. Therefore, it remains to determine:

- The complexities of $\textsc{cspsat}_{\mathbf{PA},f}(\mathcal{B}_{\mathbf{PA}})$ and $\textsc{cspsat}_{\mathbf{PA},f}(\mathcal{B}^*_{\mathbf{PA}})$.

- The complexities of $\textsc{cspsat}_{\mathcal{M},f}(\mathcal{B}_{\mathcal{M}})$ for qualitative calculus $\mathcal{M}$ being IA, CRA.

We will address these two problem in separate subsections.

6.3.2 Point Algebra

In this section we prove that $\textsc{cspsat}_f(\mathcal{B}_{\mathbf{PA}})$ is in P but $\textsc{cspsat}_f(\mathcal{B}^*_{\mathbf{PA}})$ is NP-complete.

The following proposition shows that any consistent instance of $\textsc{cspsat}_f(\mathcal{B}_{\mathbf{PA}})$ has a *minimal* solution in a sense.

Proposition 6.2 *Suppose (V, Γ, D) is an instance of $\textsc{cspsat}_f(\mathcal{B}_{\mathbf{PA}})$ such that each D_i is a finite subset of $\mathbb{R}$. If (V, Γ, D) is consistent, then there is a unique solution $(a_1, \cdots, a_n)$ such that $a_i \le a'_i$ $(1 \le i \le n)$ holds for any other solution $(a'_1, a'_2, \cdots, a'_n)$. Furthermore, if $\Gamma = \{v_i < v_j\}_{1 \le i < j \le n}$, then*

- $a_1 = \min D_1$;
- $a_k = \min\{x \in D_k : x > a_{k-1}\}$ *for* $k = 2, 3, \cdots, n$.

Proof Assume $\Gamma = \{v_i < v_j\}_{1 \le i < j \le n}$. This does not lose generality because we can combine variables related by $=$ constraints and then sort the variables by the $<, >$ constraints. Every D_i is a finite set, so (V, Γ, D) has only finitely many, say k, solutions. Suppose $(a^i_1, a^i_2, \cdots, a^i_n)$ $(i = 1, 2, \cdots, k)$ enumerate all the solutions. Let $a_j = \min\{a^i_j\}_{1 \le i \le k}$. It can be proved that $(a_1, a_2, \cdots, a_n)$ is the minimal solution and satisfies the property. The proof is simple and omitted here.

We propose a polynomial algorithm that solves $\textsc{cspsat}_f(\mathcal{B}_{\mathbf{PA}})$. For an instance (V, Γ, D), we first transform it into its finitely restricted sub-instance (V', Γ', D'). We then decide the consistency of the sub-instance by attempting to compute $(a_1, \cdots, a_n)$ by equations in Proposition 6.2. If we fail in the k-th step due to the emptiness of $\{x \in D_k : x > a_{k-1}\}$, we may conclude that the sub-instance, and thus the original instance, is inconsistent. If we succeed in computing $(a_1, a_2, \cdots, a_n)$, then it is a solution of the sub-instance and can be extended to a solution of the original instance. The soundness of the algorithm is clear by the above argument.

Theorem 6.1 *Algorithm 5 solves the* $\textsc{cspsat}_f(\mathcal{B}_{\mathbf{PA}})$.

We next analyse the complexity of the algorithm. Suppose there are n variables in V, and the sum of the cardinalities of all finite D_i is L. Then the input size is $O(n^2 + L)$ (n^2 constraints and L points). The following proposition shows the optimality of the algorithm. The proposition adopts a reduction from the 3-SAT problem, though a simpler reduction from the graph 3-colouring problem is also available.

Input: $\textsc{cspsat}_f(\mathcal{B}_{\mathbf{PA}})$ instance (V, Γ, D)
Output: The consistency of (V, Γ, D)

if Γ *is not consistent* **then**
    return 'Inconsistent';
$(V', \Gamma', D') \leftarrow$ finitely restricted sub-instance of (V, Γ, D);
Sort V' to $v'_1 < \cdots < v'_{n'}$ by Γ', modify D' correspondingly;
$a_1 \leftarrow \min D'_1$;
for $2 \leq k \leq n'$ **do**
    if $a_{k-1} \geq \max D'_k$ **then**
        return 'Inconsistent';
    $a_k \leftarrow \min\{x \in D'_k : x > a_{k-1}\}$;
end
return 'Consistent'.

Algorithm 5: Solving $\textsc{cspsat}_f(\mathcal{B}_{\mathbf{PA}})$

Proposition 6.3 *The complexity of Algorithm 5 is $O(n^2 + L)$.*

Proof The consistency of Γ can be computed in $O(n^2)$ time by Algorithm $\textsc{cspan}$ proposed in [106]. Sorting V' takes $O(n \log n)$ time. Let l_i be the cardinality of D'_i. Then step '$a_1 \leftarrow D'_1$' takes $O(l_1)$ time, and the i-th loop body takes $O(l_{i+1})$ time $(i = 1, 2, \cdots, n' - 1)$. Therefore, the complexity of the algorithm is $O(n^2 + n \log n + l_1 + l_2 + \cdots + l_{n'}) = O(n^2 + L)$.

Despite the fact that both $\textsc{cspsat}(\mathcal{B}^*_{\mathbf{PA}})$ and $\textsc{cspsat}_f(\mathcal{B}_{\mathbf{PA}})$ are in P, the next theorem shows that $\textsc{cspsat}_f(\mathcal{B}^*_{\mathbf{PA}})$ is NP-hard.

Theorem 6.2 *The consistency problem $\textsc{cspsat}_f(\mathcal{B}^*_{\mathbf{PA}})$ is NP-complete.*

Proof The NP-membership of $\textsc{cspsat}_f(\mathcal{B}^*_{\mathbf{PA}})$ follows from Proposition 6.1(ii) and that $\textsc{cspsat}_f(\mathcal{B}_{\mathbf{PA}})$ is in P. We provide a polynomial reduction from 3-SAT to $\textsc{cspsat}_f(\mathcal{B}^*_{\mathbf{PA}})$. Suppose $\phi = \bigwedge_{j=1}^{m} c_j$ is a 3-SAT instance involving propositional variables $p_1, p_2, \cdots, p_n$.

A $\textsc{cspsat}_f(\mathcal{B}^*_{\mathbf{PA}})$ instance (V, Γ, D) is constructed as follows:

$$
\begin{aligned}
V &= \{v_1, v_2, \cdots, v_n\} \cup \{u_1, u_2, \cdots u_m\},\\
D &= (D_{v_1}, D_{v_2}, \cdots, D_{v_n}, D_{u_1}, D_{u_2}, \cdots, D_{u_m}), \text{ where}\\
&\quad D_{v_i} = \{2i-1, 2i\}\\
&\quad D_{u_j} = \{2i : p_i \in \mathrm{Var}^+(c_j)\} \cup \{2i-1 : p_i \in \mathrm{Var}^-(c_j)\}\\
\Gamma &= \{v_i \neq u_j : p_i \in \mathrm{Var}(c_j)\}.
\end{aligned}
$$

Note that constraints that are not explicitly specified are assumed to be universal ones.

Intuitively, a temporal variable v_i taking the value $2i-1$ ($2i$ resp.) corresponds to propositional variable p_i being assigned true(false, resp.). Suppose $\mathrm{Var}(c_j) = \{p_{j_1}, p_{j_2}, p_{j_3}\}$. Note D_{u_j} has only three elements, and the following three constraints

$$u_j \neq v_{j_1},\ u_j \neq v_{j_2},\ u_j \neq v_{j_3}$$

exclude the case that v_{j_t} takes $2j_t$ for $p_{j_t} \in \mathrm{Var}^+(c_j)$ or $2j_t - 1$ for $p_{j_t} \in \mathrm{Var}^-(c_j)$, where $t = 1, 2, 3$ (otherwise, $v_{j_1}, v_{j_2}, v_{j_3}$ occupy all the elements in D_{u_j}. In other words, these constraints exclude the assignments that make all literals in c_j false. It is straightforward to show that ϕ is satisfiable if and only if (V, Γ, D) is consistent. The reduction is clearly polynomial. Therefore, the consistency problem $\textsc{cspsat}_f(\mathcal{B}^*_{\mathbf{PA}})$ is NP-complete.

Remark 6.1 *The NP-hardness is due to the uncertainty of the $\neq$ constraints and the finiteness of the domains. In fact it can be proved that $\textsc{cspsat}_f(\mathcal{S})$ is in P for $\mathcal{S} = \{<, =, >, \leq, \geq, ?\}$ (i.e., only the $\neq$ constraints are prohibited). A polynomial algorithm can be devised with similar idea, except that the variables are sorted into a partial order instead of a total order.*

6.3.3 Cardinal Relation Algebra

This section provides a polynomial reduction from 3-SAT to $\textsc{cspsat}_f(\mathcal{B}_{\mathbf{CRA}})$. [1] Suppose $\phi = \bigwedge_{j=1}^{m} c_j$ is a 3-SAT instance over propositional variables $p_1, p_2, \cdots p_n$, where

[1] A much clearer and simpler reduction from the graph 3-colouring problem is suggested by an review when the content of this chapter was submitted the *Artificial Intelligence* Journal. We here, however, still provide the original reduction from 3-SAT.

clause $c_j = l_{j,1} \vee l_{j,2} \vee l_{j,3}$. We construct in polynomial time an instance $(V_\phi, \Gamma_\phi, D_\phi)$ of $\textsc{cspsat}_f(\mathcal{B}_{\mathbf{CRA}})$ such that $(V_\phi, \Gamma_\phi, D_\phi)$ has a solution ν if and only if ϕ is satisfied by a propositional assignment $\pi : \{p_1, p_2, \cdots p_n\} \to \{\mathsf{true}, \mathsf{false}\}$.

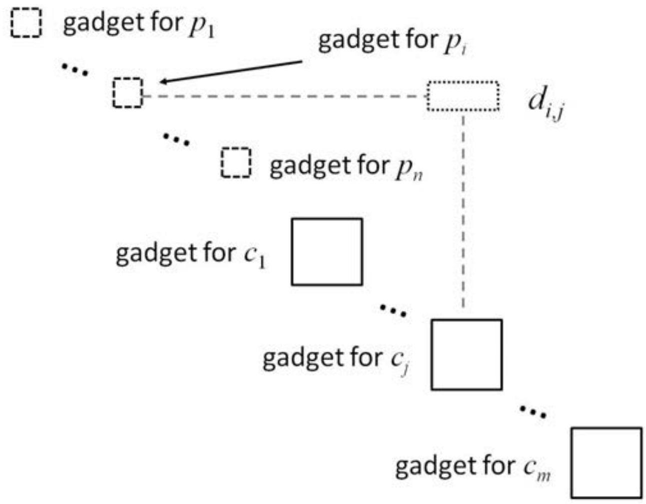


Figure 6.1: Overview of the configuration of all spatial variables in CRA, where we assume $p_i \in \mathrm{Var}(c_j)$

We introduce for each propositional variable p_i and each propositional clause c_j a 'gadget'. The gadget for p_i is a spatial variable v_i which is located in the i-th top dashed box in Figure 6.1; and the gadget for c_j consists of five spatial variables $u_{j,1}, u_{j,2}, u_{j,3}, x_j$ and y_j, which are all located in the j-th top solid box in Figure 6.1. Suppose $l_{j,s}$, the s-th literal in clause c_j, is p_i or $\neg p_i$. To establish the connection between spatial variable v_i and spatial variable $u_{j,s}$, we introduce the spatial variable $d_{i,j}$, which is located in the dots box.

The domain of v_i is $D_{v_i} = \{V_i^+, V_i^-\}$, and the domain of u_j^s is $D_{u_{j,s}} = \{U_{j,s}^+, U_{j,s}^-\}$ $(1 \leq j \leq m, s = 1, 2, 3)$ (see Figure 6.2 (a) for illustration). We intend to translate a propositional assignment π into a spatial assignment ν, such that $\pi(p_i) = \mathsf{true}$ if $\nu(v_i) = V_i^+$ and $\pi(p_i) = \mathsf{false}$ if $\nu(v_i) = V_i^-$. Assuming the literal $l_{j,s}$ is either p_i or $\neg p_i$, the point assigned to $u_{j,s}$ should be somehow decided by the point assigned to v_i. In detail, we would like to require that

- If $l_{j,s} = p_i$, then $\nu(u_{j,s}) = U_{j,s}^+$ iff $\nu(v_i) = V_i^+$;

- If $l_{j,s} = \neg p_i$, then $\nu(u_{j,s}) = U^+_{j,s}$ iff $\nu(v_i) = V^-_i$.

The spatial variable $d_{i,j}$ is introduced to transfer the position of v_i to that of $u_{j,s}$. The domain of $d_{i,j}$ is specified as $D_{d_{i,j}} = \{D^+_{i,j}, D^-_{i,j}\}$, where the positions of $D^+_{i,j}$ and $D^-_{i,j}$ depend on $l_{j,s}$ being p_i or $\neg p_i$ (see Figure 6.2) (c)(d). With constraints

$$v_i \text{ W } d_{i,j}, \quad d_{i,j} \text{ N } u_{j,s},$$

it is clear that the requirements are fulfilled (cf. Figure 6.2 (c)(d)).

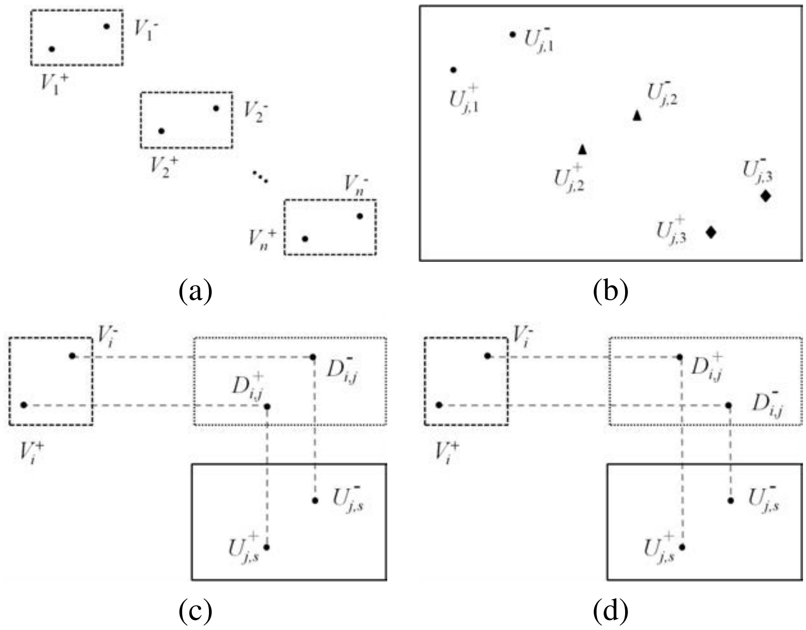


Figure 6.2: Illustrations of the domains of (a) v_i, (b) $u_{j,s}$, (c) (d) $d_{i,j}$, where $l_{j,s} = p_i$ in (c) and $l_{j,s} = \neg p_i$ in (d)

Clause $c_j = l_{j,1} \vee l_{j,2} \vee l_{j,3}$ is not satisfied by a propositional assignment π if and only if none of $l_{j,s}$ is satisfied by π, which corresponds to that $u_{j,s}$ all take positions $U^-_{j,s}$. Such configurations are excluded by spatial variables x_j and y_j in the gadget for c_j. Their domains are respectively $D_{x_j} = \{X^1_j, X^2_j, X^3_j\}$ and $D_{y_j} = \{Y^1_j, Y^2_j\}$, depicted in Figure 6.3. The following constraints about x_j and y_j are imposed:

$$c_{j,1} \text{ NW } x_j, \quad x_j \text{ NW } c_{j,2}, \quad x_j \text{ NW } y_j, \quad y_j \text{ NW } c_{j,3}.$$

The first two constraints imply that $\nu(x_j) = X^1_j$ if $\nu(u_{j,1}) = U^-_{j,1}$ and $\nu(u_{j,2}) = U^-_{j,2}$. The latter two further imply that, if $\nu(x_j) = X_j^1$ and $\nu(u_{j,3}) = U_{j,3}^-$, then y_j is not realisable, because neither Y_j^1 nor Y_j^2 can satisfy the constraints. These four constraints together enforce that y_j is not realisable if $\nu(u_{j,s}) = U_{j,s}^-$ hold for $s = 1, 2, 3$. Furthermore, it can be verified that x_j and y_j are realisable in any other configuration of $u_{j,1}, u_{j,2}, u_{j,3}$.

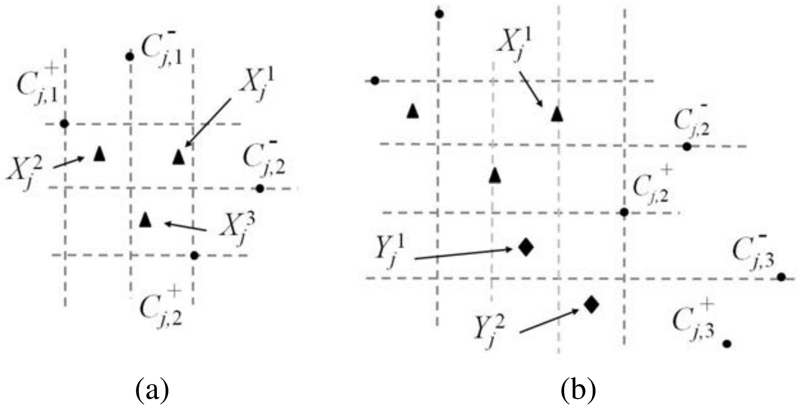


Figure 6.3: Illustrations for the domain of (a) X_j and (b) Y_j

Now we have finished the construction of the $\text{CSPSAT}_f(\mathcal{B}_{\mathbf{CRA}})$ instance $(V_\phi, \Gamma_\phi, D_\phi)$. In summary,

$$
\begin{aligned}
V_\phi =& \{v_1, \cdots, v_n\} \cup \{u_{j,1}, u_{j,2}, u_{j,3}, x_j, y_j\}_{1 \le j \le m} \\
& \cup \{d_{i,j} : p_i \in \mathrm{Var}(c_j)\}, \\
D_{v_i} =& \{V_i^+, V_i^-\}, \quad D_{u_{j,s}} = \{U_{j,s}^+, U_{j,s}^-\} \ (s = 1, 2, 3), \\
D_{x_j} =& \{X_j^1, X_j^2, X_j^3\}, \quad D_{y_j} = \{Y_j^1, Y_j^2\}, \quad D_{d_{i,j}} = \{D_{i,j}^+, D_{i,j}^-\}.
\end{aligned}
$$

Note that only a part of the constraints between variables in V_ϕ have been explicitly expressed. The rest of the constraints can be trivially inferred from the configurations of the domains given in Figures 6.1 - 6.3. For completeness, we list all constraints in Table 6.2.

Following the idea clarified in the above construction, the following proposition can be proved. We only need to translate a satisfying truth assignment π of ϕ to a solution ν of $(V_\phi, \Gamma_\phi, D_\phi)$, and vice versa.

Proposition 6.4 *Given a 3-SAT instance ϕ, suppose $(V_\phi, \Gamma_\phi, D_\phi)$ is the $\text{CSPSAT}_f(\mathcal{B}_{\mathbf{CRA}})$ instance as constructed above. Then ϕ is satisfiable if and only if $(V_\phi, \Gamma_\phi, D_\phi)$ is con-*

	v_i	$c_{i,l}, x_i, y_i$	$d_{i,j}$
$v_{i'}$	NW ($i' < i$) EQ ($i' = i$) SE ($i' > i$)	NW	W($i' = i$) NW($i' < i$) SW($i' > i$)
$c_{i',l'}$, $x_{i'}$, $y_{i'}$		NW ($i' < i$) see (b) ($i' = i$) SE ($i' > i$)	SW ($i' < i$) see (b) ($i' = i$) SE ($i' > i$)
$d_{i',j'}$			see (c)

(a)

	$c_{i,1}$	$c_{i,2}$	$c_{i,3}$	x_i	y_i	$d_{i,1}$	$d_{i,2}$	$d_{i,3}$
$c_{i,1}$	EQ	NW	NW	NW	NW	S	SW	SW
$c_{i,2}$	SE	EQ	NW	SE	NE	SE	S	SW
$c_{i,3}$	SE	SE	EQ	SE	SE	SE	SE	S
x_i	SE	NW	NW	EQ	NW	SE	SW	SW
y_i	SE	SW	NW	SE	EQ	SE	SW	SW

(b)

	$i' < i$	$i' = i$	$i' > i$
$j' < j$	$d_{i',j'} NW d_{i,j}$	$d_{i',j'} W d_{i,j}$	$d_{i',j'} SW d_{i,j}$
$j' = j$	$d_{i',j'} NW d_{i,j}$	$d_{i',j'} EQ d_{i,j}$	$d_{i',j'} SE d_{i,j}$
$j' > j$	$d_{i',j'} NE d_{i,j}$	$d_{i',j'} E d_{i,j}$	$d_{i',j'} SE d_{i,j}$

(c)

Table 6.2: Constraints between variables in V_ϕ

Variable	Assignments						
$c_{j,1}$	+	+	+	+	-	-	-
$c_{j,2}$	+	+	-	-	+	+	-
$c_{j,3}$	+	-	+	-	+	-	+
x_j	2	2	2	2	3	3	1
y_j	1	1	1	1	1	1	2

Table 6.3: The assignments of x_j and y_j, determined by those of $c_{j,s}$, where symbol '+' ('-' resp.) stands for $c_{j,s}$ being assigned $C^+_{j,s}$ ($C^-_{j,s}$ resp.), and the number k (1,2 or 3) stands for x_j (y_j, resp.) being assigned X^k_j (Y^k_j, resp.).

sistent.

Proof Suppose ϕ is satisfied by a truth assignment $\pi : \{p_1, \cdots, p_n\} \rightarrow \{\mathsf{true}, \mathsf{false}\}$. Recall the spatial variables $c_{j,1}$, $c_{j,2}$, and $c_{j,3}$ cannot be, simultaneously, assigned $C^-_{j,1}$, $C^-_{j,2}$, and $C^-_{j,3}$ respectively. We construct a solution ν of $(V_\phi, \Gamma_\phi, D_\phi)$ as follows.

- $\nu(v_i) = V^+_i$ if $\pi(p_i) = \mathsf{true}$, and $\nu(v_i) = V^-_i$ if $\pi(p_i) = \mathsf{false}$.
- $\nu(c_{j,s}) = C^+_{j,s}$ if the s-th literal $l_{j,s}$ in clause c_j is satisfied by π. Otherwise, $\nu(c_{j,s}) = C^-_{j,s}$.
- $\nu(d_{i,j}) = D^+_{i,j}$ if either (i) $\pi(p_i) = \mathsf{true}$ and $p_i \in \mathrm{Var}^+(c_j)$, or (ii) $\pi(p_i) = \mathsf{false}$ and $p_i \in \mathrm{Var}^-(c_j)$. Otherwise $\nu(d_{i,j}) = D^-_{i,j}$.
- $\nu(x_j)$ and $\nu(y_j)$ are decided by $\nu(c_{j,1})$, $\nu(c_{j,2})$ and $\nu(c_{j,3})$, see Table 6.3 for details.

It can be verified that ν satisfies all the constraints in Γ_ϕ.

On the other hand, suppose $(V_\phi, \Gamma_\phi, D_\phi)$ has a solution $\nu : V_\phi \rightarrow \bigcup_{v \in V_\phi} D_v$. Let π be the propositional assignment for the propositional variables in ϕ defined by $\pi(p_i) = \mathsf{true}$ iff $\nu(v_i) = V^+_i$. We claim that π satisfies ϕ. Otherwise suppose π does not satisfy $c_j = l_{j,1} \vee l_{j,2} \vee l_{j,3}$, i.e., $\pi(l_{j,1}) = \pi(l_{j,2}) = \pi(l_{j,3}) = \mathsf{false}$. In this case, we can prove that $\nu(c_{j,1}) = C^-_{j,1}$, $\nu(c_{j,2}) = C^-_{j,2}$, and $\nu(c_{j,3}) = C^-_{j,3}$, which implies that ν is not a solution since x_j and y_j are not realisable simultaneously under such configuration.

Therefore ϕ is satisfiable if and only if $(V_\phi, \Gamma_\phi, D_\phi)$ is consistent.

Note that for an 3-SAT instance ϕ with n propositional variables and m clauses, the $\textsc{cspsat}_f(\mathcal{B}_{\mathbf{CRA}})$ instance $(V_\phi, \Gamma_\phi, D_\phi)$ contains in total $n+8m$ spatial variables, $2n+17m$ specified points in all domains, and $(n+8m)^2$ basic CRA constraints. Therefore, the reduction is polynomial. Together with Proposition 6.1, we have

Corollary 6.4 *The problem $\textsc{cspsat}_f(\mathcal{B}_{\mathbf{CRA}})$ is NP-complete.*

6.3.4 Interval Algebra

Now we turn to the consistency problem $\textsc{cspsat}_f(\mathcal{B}_{\mathbf{IA}})$. Noticing that an interval $[x,y]$ naturally corresponds to the point (x,y) on the half-plane $\{(x,y) : x < y\}$, we devise the following reduction from 3-SAT by reusing the reduction for CRA above. Assume ϕ is a 3-SAT instance and $(V_\phi, \Gamma_\phi, D_\phi)$ is the corresponding $\textsc{cspsat}_f(\mathcal{B}_{\mathbf{CRA}})$ instance. Suppose $V_\phi = \{u_1, \cdots, u_n\}$, $\Gamma_\phi = \{u_i \alpha_{ij} u_j : 1 \le i, j \le n\}$ and $D_\phi = (D_1, \cdots, D_n\}$, where $D_i = \{d_i^1, \cdots, d_i^{t_i}\}$. We now translate $(V_\phi, \Gamma_\phi, D_\phi)$ into a $\textsc{cspsat}_f(\mathcal{B}_{\mathbf{IA}})$ instance $(V'_\phi, \Gamma'_\phi, D'_\phi)$, where Γ'_ϕ is a basic IA network. The translation maps

- each variable u_i in V_ϕ to variable u'_i in V'_ϕ;
- each basic CRA relation α_{ij} to a basic IA relation β_{ij} as specified in Table 6.4;
- each point (x,y) in D_i to an interval $[x, y+\Delta]$ in D'_i, where Δ is a large number such that $x < y + \Delta$ for any (x,y) in $\bigcup_{i=1}^n D_i$.

α_{ij}	NW	N	NE	W	EQ	E	SW	S	SE
β_{ij}	di	si	oi	fi	eq	f	o	s	d

Table 6.4: Translation of the constraints

We show that the translation reserves consistency.

Proposition 6.5 *The instance $(V_\phi, \Gamma_\phi, D_\phi)$ in $\textsc{cspsat}_f(\mathcal{B}_{\mathbf{CRA}})$ is consistent if and only if the instance $(V'_\phi, \Gamma'_\phi, D'_\phi)$ in $\textsc{cspsat}_f(\mathcal{B}_{\mathbf{IA}})$ is consistent.*

Proof Suppose $(a_1, \cdots, a_n)$ is a solution of $(V_\phi, \Gamma_\phi, D_\phi)$, where $a_i = (x_i, y_i) \in D_i$. Define interval $a'_i = [x_i, y_i + \Delta] \in D'_i$. We prove that $(a'_1, \cdots, a'_i)$ is a solution of $(V'_\phi, \Gamma'_\phi, D'_\phi)$. It is clear that $a'_i \in D'_i$ by the translation from D_i to D'_i. We need only verify that all constraints in Γ' are satisfied by $(a'_1, \cdots, a'_i)$. This can be done by discussing each of the nine kinds of basic IA constraints in Γ'_ϕ.

Suppose u'_i di u'_j is a constraint in Γ'_ϕ. We need prove that $[x_i, y_i + \Delta]$ di $[x_j, y_j + \Delta]$, i.e., $x_i < x_j < y_j + \Delta < y_i + \Delta$. By the translation we know that u_i NW u_j is in Γ_ϕ. Therefore (x_i, y_i) NW (x_j, y_j), i.e., $x_i < x_j$ and $y_i > y_j$. Meanwhile $x_j < y_j + \Delta$ is guaranteed by the selection of Δ. So the constraint u'_i di u'_j is satisfied by $(a'_1, \cdots, a'_n)$. The rest eight cases can be proved analogously.

Therefore we obtain the following corollary.

Corollary 6.5 *The problem* $\text{CSPSAT}_f(\mathcal{B}_{IA})$ *is NP-complete.*

Proof We have provided a reduction from 3-SAT which is clearly polynomial. Meanwhile, the problem is in NP by Proposition 6.1.

So far, we have finished the discussion for point-based qualitative calculi PA, IA and CRA. The next section will address region-based qualitative calculi RCC-5 and RCC-8.

6.4 Qualitative Calculi RCC-5 and RCC8

This section discusses the extended consistency problem in RCC-5 and RCC-8 [1]. Note that although the universe of RCC-5 (or RCC-8) is the set of all regions, this section assumes that all the landmarks are polygons. This is because landmarks, as inputs of instances, are required to be representable by computers, (or in other words, they should be finitely representable). Meanwhile, general polygons (which may have holes or be disconnected) are the most widely used approximations of regions: they are simple, intuitive, and expressive [2].

[1] We consider standard RCC models interpreted on the 2-dim Euclidean topology, which is the most influential and useful RCC model. Note that different RCC models may result in different computational complexity (see Section 6.4.3.3 for an example.

[2] Semi-algebraic curve is another method for region representation. Although semi-algebraic curves are more expressive than polygons, the operations (such as union and intersection) on them are much more complicated, and the topological expressive ability of semi-algebraic curves and polygons are the same.

Under the assumption that all landmarks are represented by general polygons, we show in this section that all these consistency problems are in NP. In particular, we show that $\textsc{cspsat}_s(\mathcal{B}_{RCC5})$ is in P, but that $\textsc{cspsat}_f(\mathcal{B}_{RCC5})$ and $\textsc{cspsat}_s(\mathcal{B}_{RCC8})$ are all NP-complete. It is not surprising that $\textsc{cspsat}_f(\mathcal{B}_{RCC5})$ is NP-complete if we regard the finitely restricted sub-instance of each instance of $\textsc{cspsat}_f(\mathcal{B}_{RCC5})$ as a classical CSP, but the NP-hardness of $\textsc{cspsat}_s(\mathcal{B}_{RCC8})$ is quite undesirable. One way to circumvent this obstacle is to use a stronger connectedness instead of the one used in Definition 6.3.

The remainder of this section is organised as follows. Several of our computational complexity results are related to computing the intersection of landmarks (represented as polygons), so we first analyse the computational complexity of computing the intersection of multiple polygons in Subsection 6.4.1. The tractability of $\textsc{cspsat}_s(\mathcal{B}_{RCC5})$ is then proved in Subsection 6.4.2. Subsection 6.4.3 shows that $\textsc{cspsat}_s(\mathcal{B}_{RCC8})$ is NP-complete if the RCC8 relations are interpreted as in Definition 6.3, and proves that the same problem is in P (i.e. tractable) if we adopt another interpretation that uses a stronger connectedness.

6.4.1 Planar subdivision and overlay computation

In the following subsections we will see that computing the intersection of landmarks (represented as polygons) is critically important when solving the consistency problems for RCC5 and RCC8 in the extended QCSP framework. To facilitate the discussion, this subsection analyses the computational complexity of computing the intersection of multiple polygons. Our discussion is based on the *doubly-connected edge list* (DCEL) structure for representing planar subdivisions (cf. e.g. [18]).

Informally speaking, a subdivision is a general plane object composed of points and line segments, and an overlay is the collection of multiple subdivisions. In detail, a *subdivision*, is the map induced by a planar embedding of a graph. The embedding of nodes (arcs, resp.) of the graph is called *vertices* (*edges*, resp.) in the subdivision, where each edge is required to be a straight line segment. A *face* of the subdivision is a maximal connected subset of the remaining part of the plane excluded by all the edges and vertices. Note that faces in a subdivision are decided by the vertices and edges, and may have holes [1]. The *complexity of a subdivision* is defined as the sum

[1] Note that there is a unique unbounded face.

of the number of vertices, the number of edges, and the number of faces in the subdivision. For example, the subdivision of the rectangle in Figure 6.4 (a) has two faces (Figure 6.4 (a)), four vertices (Figure 6.4 (b)) and four edges (Figure 6.4 (c)), and thus its complexity is ten.

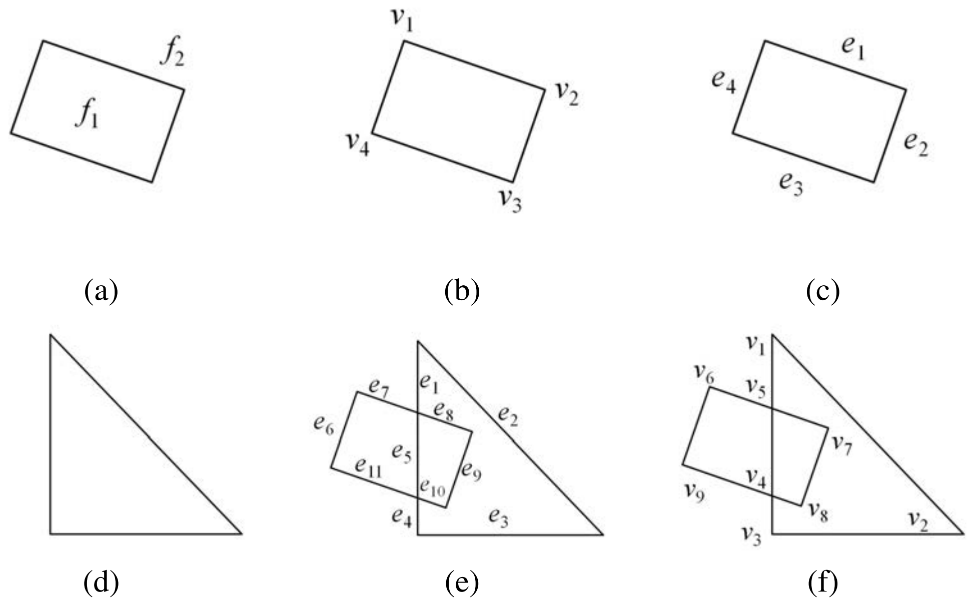


Figure 6.4: An example of subdivision

Note that the subdivisions (which are composed of vertices, edges and faces) need to be represented by data structures in computers. Though various kinds of data structures may be used to represent subdivisions, the *doubly-connected edge list* (DCEL) is quite popular as the overlay of two subdivisions which are represented by DCELs can be efficiently computed (see below). The DCEL considers each edge in a subdivision as two directed half-edges with opposite directions. A half-edge has a vertex as its origin, and another vertex as its end. Briefly, the DCEL maintains a set of records for faces, half-edges, and vertices respectively, which takes $O(K)$ space for a subdivision with complexity K :

(i) For each vertex, the DCEL stores its coordinates, and a pointer to one half-edge (arbitrarily chosen if not unique) that has the vertex as its origin.

(ii) For each face, the DCEL stores a pointer to one (arbitrarily chosen) edge which is on its outer boundary, and also records all its holes by maintaining a list which contains a pointer to one (arbitrarily chosen) half-edge on the boundary of each hole.

(iii) For each half-edge, the DCEL stores a pointer to its origin vertex, a pointer to its dual half-edge, and a pointer to its bounding face (i.e., the face at the left side of the directed half-edge) as well as two pointers to the next and previous half-edges on the boundary of the above bounding face.

The *overlay* of two subdivisions S_1 and S_2 is defined to be the subdivision of the plane induced by all the edges from S_1 and S_2. Figure 6.4 (e) and (f) illustrate the overlay of the rectangle in Figure 6.4 (a) and the triangle in Figure 6.4 (d). The overlay has four faces, eleven edges and nine vertices, and hence its complexity is twenty-four. In the sequel, we write FACE, EDGE, and VTX respectively for the set of faces, the set of edges, and the set of vertices in an overlay subdivision. In general, we have the following result about the complexity of overlay computation.

Theorem 6.3 *[18, Theorem 2.6] Let S_1 be a planar subdivision of complexity k_1, let S_2 be a subdivision of complexity k_2. The overlay of S_1 and S_2 can be constructed in $O((k_1 + k_2 + k)\log(k_1 + k_2))$ time, where k is the complexity of the overlay.*

The following corollary relaxed the above result.

Corollary 6.6 *Let S be the overlay of subdivisions S_1 and S_2. Assume that the complexity of S is k. Then S can be computed in $O(k \log k)$ time from S_1 and S_2.*

Proof Clearly S is at least complex as each of S_1 and S_2, so k is no less than $\max(k_1, k_2)$. Therefore k is $\Omega(k_1 + k_2)$. By Theorem 6.3, S can be computed in $O(k_1 + k_2 + k \log(k_1 + k_2)) = O(k \log k)$ time.

Note that we are only able to guarantee that k is $O(k_1 k_2)$ (see the proof of Lemma 6.3 below).

Theorem 6.3 only considers the overlay of two subdivisions. For the problem $\textsc{cspsat}_s(\mathcal{B}_{RCC5})$ (or $\textsc{cspsat}_s(\mathcal{B}_{RCC8})$), we may need to compute the overlay $\mathcal{O}$ of $m \geq 3$ landmarks. This will be achieved progressively by computing the overlays $\mathcal{O}_i$ of the first i landmarks for i increasing from 2 to m. The computational complexity, however, is not quite trivial. Assume for simplicity that the complexity of each landmark is bounded by k. A simple induction implies that the complexity of $\mathcal{O}_i$ is $O(k^i)$, and thus computing $\mathcal{O}$ takes $O(k^m \log k)$ time. This result is exponential and thus quite unsatisfactory. In fact, $\mathcal{O}$ can be computed in polynomial time, observing that a much better estimation on the complexity of $\mathcal{O}$ exists.

Lemma 6.3 *Let $L = \{l_1, \cdots, l_m\}$ be a set of subdivisions (landmarks) in the plane, where the complexity of l_i is k_i. Let $\mathcal{O}$ be the overlay of all subdivisions in L, and K be the sum of all k_i. Then the complexity of $\mathcal{O}$ is $O(K^2)$, and $\mathcal{O}$ can be computed in $O(mK^2 \log K)$ time.*

Proof We first show that the complexity of $\mathcal{O}$ is $O(K^2)$.

It is clear that there are at most K vertices (or edges, faces respectively) in the subdivisions in L in total. Note that each vertex in the overlay $\mathcal{O}$ is either a vertex of a subdivision, or the intersection point of two edges of different subdivisions. Therefore we know that $|\textsc{Vtx}|$ is $O(K^2)$. It is clear that each edge in $\mathcal{O}$ must be a part of an edge from some subdivision in L. Moreover, an edge in a subdivision in L may be split into at most K edges in the overlay. Therefore $|\textsc{Edge}|$ is $O(K^2)$. The number of faces $|\textsc{Face}|$ in $\mathcal{O}$ is also bounded by $O(K^2)$. To see this, let l_i' be the planar subdivision obtained by replacing the line segments in l_i with lines.[1] It is clear that the overlay $\mathcal{O}'$ obtained by combining all l_i' is finer than $\mathcal{O}$. Because there are K lines in $\mathcal{O}'$ and K lines partition the plane into at most $1+1+2+\cdots+K = O(K^2)$ faces, we know that the number of faces in $\mathcal{O}'$ is $O(K^2)$. Therefore the number of faces in $\mathcal{O}$ is also $O(K^2)$, thus the complexity of subdivision overlay $\mathcal{O}$ is $O(K^2)$.

Let $\mathcal{O}_i$ be the overlay of the first i subdivisions $l_1, \cdots, l_i$. Note that the complexity of $\mathcal{O}_i$ is no more than that of $\mathcal{O} = \mathcal{O}_m$. By Corollary 6.6, each subdivision $\mathcal{O}_i$ can be computed in $O(K^2 \log K)$ time. Therefore, computing the overlay $\mathcal{O}$ takes $O(mK^2 \log K)$ time in total.

We note that the DCEL of $\mathcal{O}$ contains incidence and adjacency information between two elements in $\textsc{Face}$, $\textsc{Edge}$, and $\textsc{Vtx}$. The relationship between such an element and a polygon in L, however, is not provided. For example, the DCEL does not tell us whether an edge lies inside, outside, or on the boundary of a polygon l_i. To represent

[1]Note here that we temporally allow the edges in a planar subdivision to be rays.

the complete topological information of the polygon system L, we introduce the following functions, which can be computed by supplying a number of attributes to the DCEL of the overlay.

For each polygon $l_i \in L$, we write $\mathcal{I}_{\text{FACE}}(l_i)$ for the set of faces in $\mathcal{O}$ that lie in the interior of l_i, and write $\mathcal{E}_{\text{FACE}}(l_i)$ for the set of the faces that lie in the exterior of l_i:

$$\mathcal{I}_{\text{FACE}}(l_i) \coloneqq \{f \in \text{FACE} : f \subseteq l_i^\circ\}, \tag{6.1}$$

$$\mathcal{E}_{\text{FACE}}(l_i) \coloneqq \{f \in \text{FACE} : f \cap l_i = \varnothing\}. \tag{6.2}$$

It is clear that $\mathcal{I}_{\text{FACE}}(l_i) \cup \mathcal{E}_{\text{FACE}}(l_i) = \text{FACE}$ and $\mathcal{I}_{\text{FACE}}(l_i) \cap \mathcal{E}_{\text{FACE}}(l_i) = \varnothing$.

For an edge or a vertex in $\mathcal{O}$, it may be in the interior, in the exterior, or on the boundary of l_i. Therefore we define:

$$\mathcal{I}_{\text{EDGE}}(l_i) \coloneqq \{e \in \text{EDGE} : e \subseteq l_i^\circ\}, \tag{6.3}$$

$$\mathcal{E}_{\text{EDGE}}(l_i) \coloneqq \{e \in \text{EDGE} : e \cap l_i = \varnothing\}, \tag{6.4}$$

$$\mathcal{B}_{\text{EDGE}}(l_i) \coloneqq \{e \in \text{EDGE} : e \subseteq \partial l_i\}, \tag{6.5}$$

and similarly,

$$\mathcal{I}_{\text{VTX}}(l_i) \coloneqq \{v \in \text{VTX} : v \in l_i^\circ\}, \tag{6.6}$$

$$\mathcal{E}_{\text{VTX}}(l_i) \coloneqq \{v \in \text{VTX} : v \notin l_i\}, \tag{6.7}$$

$$\mathcal{B}_{\text{VTX}}(l_i) \coloneqq \{v \in \text{VTX} : v \in \partial l_i\}. \tag{6.8}$$

We provide an example for intuitive explanation. Suppose landmark set $L = \{l_1, l_2, l_3\}$ consists three landmarks as illustrated in Figure 6.5(a). Then for the overlay of all three landmarks, we have $\text{FACE} = \{f_0, \cdots, f_4\}$, $\text{VTX} = \{v_1, \cdots, v_{11}\}$ and $\text{EDGE} = \{e_1, \cdots, e_{14}\}$, as shown in Figure 6.5(b-d). Furthermore, we have, for landmark l_1,

$\mathcal{I}_{\text{FACE}}(l_1) = \{f_1, f_2\}$, $\mathcal{I}_{\text{VTX}}(l_1) = \{v_6, v_{11}\}$, $\mathcal{I}_{\text{EDGE}}(l_1) = \{e_6, e_{10}, e_{11}\}$,

$\mathcal{E}_{\text{FACE}}(l_1) = \{f_0, f_3, f_4\}$, $\mathcal{E}_{\text{VTX}}(l_1) = \{v_3, v_8, v_9\}$, $\mathcal{E}_{\text{EDGE}}(l_1) = \{e_2, e_3, e_7, e_8, e_9\}$,

$\mathcal{B}_{\text{VTX}}(l_1) = \{v_1, v_2, v_7, v_{10}, v_4, v_5\}$, $\mathcal{B}_{\text{EDGE}}(l_1) = \{e_1, e_{12}, e_{13}, e_{14}, e_4, e_5\}$.

Functions 6.1 to 6.8 defined above and the DCEL completely describe the topo-

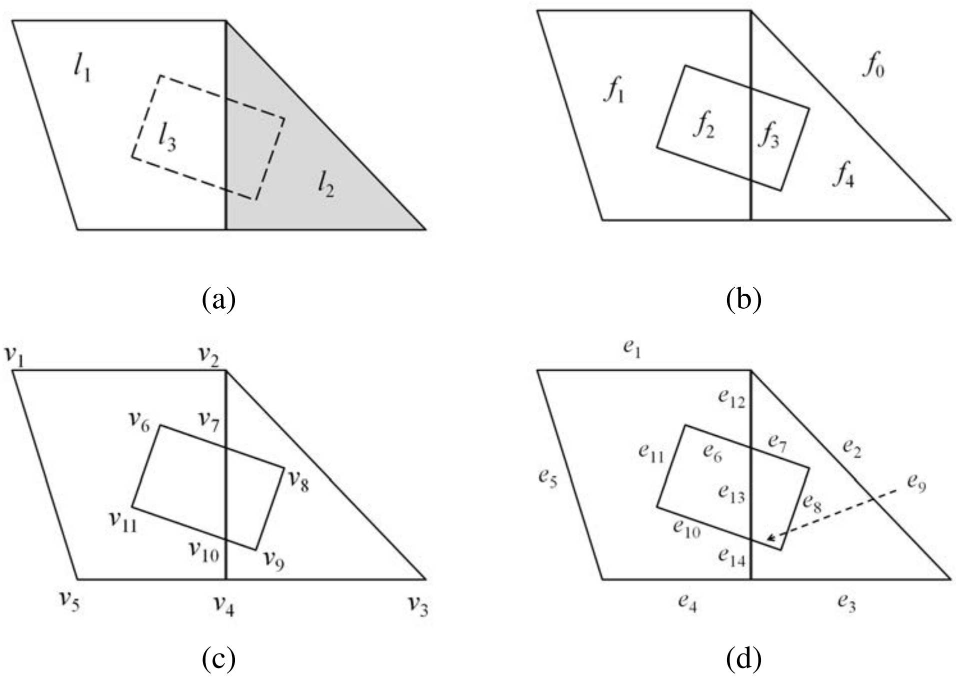


Figure 6.5: Example of the overlay of $L = \{l_1, l_2, l_3\}$.

logical information of all the landmarks. For clarity, we also define a function $S_{\textsc{Face}}$: $\textsc{Edge} \cup \textsc{Vtx} \rightarrow 2^{\textsc{Face}}$ which provides all the surrounding faces of a vertex or an edge:

$$S_{\textsc{Face}}(x) := \{f \in \textsc{Face} : x \text{ lies on the boundary of } f\}. \tag{6.9}$$

Note that the above function can be obtained directly from the DCEL.

Now we discuss the computational complexity of computing the DCEL of overlay $\mathcal{O}$, with additional attributes introduced for the functions defined above.

Lemma 6.4 *Suppose $L = \{l_1, \cdots, l_m\}$ is a set of polygons, where each l_i is represented by a planar subdivision with complexity k_i. Let $\mathcal{O}$ be the overlay of all polygons in L, and K be the sum of all k_i. Then the DCEL of the overlay $\mathcal{O}$ and the functions defined in Eq.s 6.1 to 6.8 can be computed in $O(m^2K^2 \log K)$ time.*

Proof We already know that the overlay $\mathcal{O}$ of all the polygons can be computed in $O(mK^2 \log K)$ time, if we do not need to compute functions defined in Eq.s 6.1 to 6.8.

Computing these functions would increase the complexity to $\mathcal{O}$ to $O(m^2K^2 \log K)$. In detail, we use an additional vector to represent the relation (i.e., inside, outside or on the boundary) between a component (i.e., a face, edge or vertex) in overlay $\mathcal{O}_i$ and the polygons $l_1, l_2, \cdots, l_i$. When updating the overlay $\mathcal{O}_i$ to $\mathcal{O}_{i+1}$, we need to update these vectors correspondingly. Note that for each $\mathcal{O}_i$, there are $O(K^2)$ components as discussed above, and hence $O(K^2)$ vectors, each of which has i (which is $O(m)$) indices. Therefore we need $O(mK^2)$ time to update all the vectors for each overlay $\mathcal{O}_i$, and thus $O(m^2K^2)$ time in total for $\mathcal{O}$. The functions in Eq.s 6.1 to 6.8 can be computed from the vectors for $\mathcal{O}$ directly in $O(mK^2)$ time. In summary, it takes additional $O(m^2K^2)$ time to compute all the functions. Therefore, the total time complexity is $O(mK^2 \log K + m^2K^2)$ which can be relaxed to $O(m^2K^2 \log K)$.

6.4.2 Introducing Landmarks to RCC-5

This section focuses on the complexity of the extended consistency problem in RCC-5, assuming all the landmarks are polygons. We first give a number of necessary and sufficient conditions for deciding the consistency of $\text{CSPSAT}_{\mathbf{RCC5},s}(\mathcal{B}_{\mathbf{RCC5}})$ instances in the first subsection, followed by the proof of its sufficiency. Note that the conditions can be checked in polynomial time, therefore $\text{CSPSAT}_{\mathbf{RCC5},s}(\mathcal{B}_{\mathbf{RCC5}})$ is tractable. The last subsection proves $\text{CSPSAT}_{\mathbf{RCC5},s}(\mathcal{B}_{\mathbf{RCC5}})$ is NP-hard by providing a reduction from 3-SAT.

Here we make a convention when specifying $\text{CSPSAT}_{\mathbf{RCC5},s}(\mathcal{B}_{\mathbf{RCC5}})$ instances. Suppose $(V \uplus U, \Gamma, \mathbf{D}_U)$ is such an instance, where $V = \{v_1, v_2, \cdots, v_n\}$ is the set of unrestricted variable and $U = \{u_1, u_2, \cdots, u_m\}$ is the set of uniquely restricted variables. Assume that variable u_i can only be interpreted by polygonal landmark l_i. In such case, we will use symbol l_i to denote both the polygon and variable u, and thus, we use $(V \uplus L, \Gamma)$ to denote the instance, where $L = \{l_1, l_2, \cdots, l_m\}$ is the set of all landmarks.

6.4.2.1 Necessary and sufficient conditions

Recall that we have defined the $\mathcal{I}_{\text{FACE}}(l_j)$ ($\mathcal{E}_{\text{FACE}}(l_j)$ resp.) to denote the faces that are parts of landmark l_j (lie outside of landmark l_j resp.). Now we consider the imposed relationship between components in the overlay of landmarks and variables. For a variable v_i, the constraints about v_i may force some faces to be part of v_i, and some faces to be outside of v_i. Precisely, face f is required to be part of v_i if there is some landmark l_j such that $f \in \mathcal{I}_{\textsc{Face}}(l_j)$ and $v_i\mathbf{PPi}l_j$ (for the RCC-8 case, $v_i\mathbf{TPPi}$ or $\mathbf{NTPPi}l_j$). On the other side, face f is required to lie outside v_i in the following two cases, (i) $v_i\mathbf{DR}l_j$ (for RCC-8 case, $v_i\mathbf{DC}l_j$ or $v_i\mathbf{EC}l_j$) and $f \in \mathcal{I}_{\textsc{Face}}(l_j)$, or (ii) $v_i\mathbf{PP}l_j$ (for the RCC-8 case, $v_i\mathbf{TPP}l_j$ or $v_i\mathbf{NTPP}l_j$) and $f \in \mathcal{E}_{\textsc{Face}}(l_j)$. So we define of $\mathcal{I}_{\textsc{Face}}(v_i)$ and $\mathcal{E}_{\textsc{Face}}(v_i)$ as follows,

$$\mathcal{I}_{\textsc{Face}}(v_i) := \bigcup\{\mathcal{I}_{\textsc{Face}}(l_j) : v_i\mathbf{PPi}l_j\}, \tag{6.10}$$

$$\mathcal{E}_{\textsc{Face}}(v_i) := \bigcup\{\mathcal{I}_{\textsc{Face}}(l_j) : v_i\mathbf{DR}l_j\} \cup \bigcup\{\mathcal{E}_{\textsc{Face}}(l_j) : v_i\mathbf{PP}l_j\}. \tag{6.11}$$

Note that for RCC-8, the constraints in Eq.s 6.10 and 6.11 (e.g. $v_i\mathbf{PP}l_j$) should be replaced by the disjunction of corresponding constraints (e.g. $v_i\mathbf{TPP}l_j$ or $v_i\mathbf{NTPP}l_j$).

For example, suppose Γ is a constraint network that involves three landmarks l_1, l_2, l_3 which are shown in Figure 6.5(a), and one variable v_1. The constraints related to v_1 are specified as $v_1\mathbf{PPi}l_1, v_1\mathbf{PPi}l_2, v_1\mathbf{PO}l_3$. Then in such case,

$$\mathcal{I}_{\textsc{Face}}(v_1) = \mathcal{I}_{\textsc{Face}}(l_1) \cup \mathcal{I}_{\textsc{Face}}(l_2) = \{f_1, f_2, f_3, f_4\}, \quad \mathcal{E}_{\textsc{Face}}(v_1) = \varnothing.$$

The following proposition claims that no face belongs to both $\mathcal{I}_{\textsc{Face}}(v_i)$ and $\mathcal{E}_{\textsc{Face}}(v_i)$, given that the constraint network is path-consistent.

Proposition 6.6 *Suppose Γ is a basic RCC-5 constraint network that involves at least one landmark. If Γ is path-consistent, then $\mathcal{I}_{\textsc{Face}}(v_i) \cap \mathcal{E}_{\textsc{Face}}(v_i) = \varnothing$.*

Proof Assume $f \in \mathcal{I}_{\textsc{Face}}(v_i) \cap \mathcal{E}_{\textsc{Face}}(v_i)$. There exists some l_j such that $v_i\mathbf{PPi}l_j$ and $f \in \mathcal{I}_{\textsc{Face}}(l_j)$. Furthermore, there exists some l_k such that either (i) $v_i\mathbf{DR}l_k$ and $f \in \mathcal{I}_{\textsc{Face}}(l_k)$, or (ii) $v_i\mathbf{PP}l_k$ and $f \in \mathcal{E}_{\textsc{Face}}(l_k)$.

Both cases lead to contradiction. For the first case, we know that $f \subseteq l_j \cap l_k$, while the path-consistency of Γ implies that $l_j\mathbf{DR}l_k$. For the second case, the path-consistency of Γ implies $l_j\mathbf{PP}l_k$, but $f \subseteq l_j$ and $f \cap l_k = \varnothing$.

The following theorem provides a necessary and sufficient condition that decides $\textsc{cspsat}_{\mathbf{RCC5},s}(\mathcal{B}_{\mathbf{RCC5}})$. Note that the condition only involves

$$\textsc{Face}, \mathcal{I}_{\textsc{Face}}(l_j), \mathcal{E}_{\textsc{Face}}(l_j), \mathcal{I}_{\textsc{Face}}(v_i), \mathcal{E}_{\textsc{Face}}(v_i),$$

Constraint	Conditions
$v_i\mathbf{PO}l_j$	$\mathcal{E}_{\textsc{Face}}(v_i) \cup \mathcal{E}_{\textsc{Face}}(l_j) \neq \textsc{Face}$, $\mathcal{E}_{\textsc{Face}}(v_i) \cup \mathcal{I}_{\textsc{Face}}(l_j) \neq \textsc{Face}$, $\mathcal{I}_{\textsc{Face}}(v_i) \cup \mathcal{E}_{\textsc{Face}}(l_j) \neq \textsc{Face}$
$v_i\mathbf{PP}l_j$	$\mathcal{I}_{\textsc{Face}}(v_i) \neq \mathcal{I}_{\textsc{Face}}(l_j)$
$v_i\mathbf{PPi}l_j$	$\mathcal{E}_{\textsc{Face}}(v_i) \neq \mathcal{E}_{\textsc{Face}}(l_j)$
$v_i\mathbf{PO}v_j$	$\mathcal{E}_{\textsc{Face}}(v_i) \cup \mathcal{E}_{\textsc{Face}}(v_j) \neq \textsc{Face}$, $\mathcal{E}_{\textsc{Face}}(v_i) \cup \mathcal{I}_{\textsc{Face}}(v_j) \neq \textsc{Face}$, $\mathcal{I}_{\textsc{Face}}(v_i) \cup \mathcal{E}_{\textsc{Face}}(v_j) \neq \textsc{Face}$
$v_i\mathbf{PP}v_j$	$\mathcal{I}_{\textsc{Face}}(v_i) \cup \mathcal{E}_{\textsc{Face}}(v_j) \neq \textsc{Face}$

Table 6.5: Conditions for extended RCC-5 constraint network

and the constraints in the network, hence can be checked after computing the overlay of all landmarks and $\mathcal{I}_{\textsc{Face}}(v_i)$ and $\mathcal{E}_{\textsc{Face}}(v_i)$.

Theorem 6.4 *Suppose Γ is a basic RCC-5 constraint network that involves at least one landmark. Then Γ is consistent, if and only if*

- *Γ is path-consistent.*
- *For any $v_i \in V$, $\mathcal{E}_{\textsc{Face}}(v_i) \neq \textsc{Face}$.*
- *All the conditions in Table 6.5 hold.*

We delay the proof of the sufficiency to a later subsection. We give the necessity part here.

Necessity Suppose $\{\bar{v}_1, \cdots, \bar{v}_n\}$ is a solution of Γ. Because each $\bar{v}_i$ has nonempty interior, there exists at least one face f such that $f \cap \bar{v}_i$ is nonempty. Clearly, $f \notin \mathcal{E}_{\textsc{Face}}(v_i)$ since faces in $\mathcal{E}_{\textsc{Face}}(v_i)$ are all disjoint from $\bar{v}_i$. Therefore, $\mathcal{E}_{\textsc{Face}}(v_i) \neq \textsc{Face}$.

If $(v_i\mathbf{PO}l_j) \in \Gamma$, then by assumption we have $\bar{v}_i\mathbf{PO}l_j$. By definition of **PO** (see Table 2.3), we know $\bar{v}_i$ and l_j have a common interior point. This implies that there exists some face f that contains an interior point of $\bar{v}_i \cap l_j$. Face f is neither in $\mathcal{E}_{\textsc{Face}}(v_i)$ nor in $\mathcal{E}_{\textsc{Face}}(l_j)$. That is, $\mathcal{E}_{\textsc{Face}}(v_i) \cup \mathcal{E}_{\textsc{Face}}(l_j) \neq \textsc{Face}$. Similarly, we know neither $\mathcal{E}_{\textsc{Face}}(v_i) \cup \mathcal{I}_{\textsc{Face}}(l_j)$ nor $\mathcal{I}_{\textsc{Face}}(v_i) \cup \mathcal{E}_{\textsc{Face}}(l_j)$ is $\textsc{Face}$.

If $(v_i\mathbf{PP}l_j) \in \Gamma$, then $\bar{v}_i\mathbf{PP}l_j$. Because l_j is the regularised union of all the blocks it contains, we know there exists at least one face in $\mathcal{I}_{\textsc{Face}}(l_j)$ that is not in $\mathcal{I}_{\textsc{Face}}(v_i)$. This shows $\mathcal{I}_{\textsc{Face}}(v_i) \neq \mathcal{I}_{\textsc{Face}}(l_j)$.

The rest situations are either quite straightforward or similar to the above two cases.

Now we are able to decide the consistency of an instance in the following procedure:

- Compute $\mathcal{I}_{\text{FACE}}(l_j)$ and $\mathcal{E}_{\text{FACE}}(l_j)$ for each landmark l_j (this relies on the computation of the overlay subdivision $\mathcal{O}$).
- Compute $\mathcal{I}_{\text{FACE}}(v_i)$ and $\mathcal{E}_{\text{FACE}}(v_i)$ for each variable v_i.
- Check the conditions in Theorem 6.4.

Therefore the complexity of solving $\textsc{cspsat}_{\mathbf{RCC5},s}(\mathcal{B}_{\mathbf{RCC5}})$ consists three parts, viz., the complexity of computing the $\mathcal{I}_{\text{FACE}}(l_j)$ and $\mathcal{E}_{\text{FACE}}(l_j)$,, the complexity of computing $\mathcal{I}_{\text{FACE}}(v_i)$ and $\mathcal{E}_{\text{FACE}}(v_i)$, and the complexity of checking the condition in Theorem 6.4. Put them together, we come to the following theorem.

Theorem 6.5 *Suppose $(V \cup L, \Gamma)$ is a basic RCC-5 constraint network, and $V = \{v_1, \cdots, v_n\}$ and $L = \{l_1, \cdots, l_m\}$ are the set of variables and, respectively, the set of landmarks appearing in Γ. Assume each landmark l_i is represented by a planar subdivision with complexity k_i. Let K be the sum of all k_i. Then the consistency of Γ can be decided in $O((n+m)^3 + n(n+m)K^2 + m^2K^2 \log K)$ time, which may be simplified to $O(n^3 + n^2K^2 + m^2K^2 \log K)$.*

Proof Lemma 6.4 shows that $\mathcal{O}$, the overlay of all the landmarks, together with $\mathcal{I}_{\text{FACE}}(l_j)$ and $\mathcal{E}_{\text{FACE}}(l_j)$, can be computed in $O(m^2K^2 \log K)$ time.

All $\mathcal{I}_{\text{FACE}}(v_i)$ and $\mathcal{E}_{\text{FACE}}(v_i)$ can be computed in $O(nmK^2)$ time by definition.

For the conditions in Theorem 6.4, the path-consistency checking takes $O((n+m)^3)$ time. Each of the rest $O(n(n+m))$ conditions can be checked in $O(K^2)$ time directly. Therefore, checking all the conditions in Theorem 6.4 takes $O((n+m)^3 + n(n+m)K^2)$ time.

In conclusion, the consistency can be checked in $O((n+m)^3 + n(n+m)K^2 + m^2K^2 \log K)$ time, which can be reduced to $O(n^3 + n^2K^2 + m^2K^2 \log K)$.

Remark 6.2 *If the number of landmarks is small and the landmarks are simple (i.e. m and K are both small), then the computational complexity can be simplified to $O(n^3)$.*

When the landmarks are fixed and we need to solve many instances, we may compute and store the DCEL of the overlay once and for all. In such a case, the computational complexity of solving an instance involving n variables becomes $O(n^3 + n^2K^2)$, where we do not neglect the n^2K^2 term as K could be large.

6.4.2.2 Sufficiency proof for Theorem 6.4

The sufficiency part is proved by a realisation algorithm which generates a solution of the constraint network, which is similar to the classic realisation algorithm introduced in Subsection 2.5. First we define $\mathcal{P}_{\textsc{Face}}(v_i) = \textsc{Face} - \mathcal{I}_{\textsc{Face}}(v_i) - \mathcal{E}_{\textsc{Face}}(v_i)$. We will construct for each variable v_i a region c_i as follows

- For each face f in $\mathcal{P}_{\textsc{Face}}(v_i)$, select a base region contained in f and put it into a_i (initialised with empty set).
- For any $j > i$ such that $(v_i\mathbf{PO}v_j) \in \Gamma$ and $\mathcal{P}_{\textsc{Face}}(v_i) \cap \mathcal{P}_{\textsc{Face}}(v_j) \neq \varnothing$, select a base region in some face $f \in \mathcal{P}_{\textsc{Face}}(v_i) \cap \mathcal{P}_{\textsc{Face}}(v_j)$, and put it into both a_i and a_j.
- For each i, let $b_i \leftarrow a_i \cup \bigcup\{a_j : v_j\mathbf{PP}v_i \in \Gamma\}$.
- For each i, let $c_i \leftarrow b_i \cup \bigcup\{l_j : l_j\mathbf{PP}v_i \in \Gamma\}$.

Note that the 'base regions' in above procedure are required to be pairwise disjoint polygons, and the union of all base regions should not contain any face. These requirements can be easily achieved as we may select arbitrarily small polygons.

Lemma 6.5 *Let Γ be a path-consistent basic RCC-5 constraint network that involves at least one landmark. Suppose* Face *is the set of faces of Γ. Then, for each face $f \in \textsc{Face}$, we have*

- $f \in \mathcal{I}_{\textsc{Face}}(v_i)$ *iff* $f \subseteq c_i^\circ$.
- $f \in \mathcal{E}_{\textsc{Face}}(v_i)$ *iff* $f \cap c_i = \varnothing$.
- $f \in \mathcal{P}_{\textsc{Face}}(v_i)$ *iff* $f \nsubseteq c_i$ *and* $f \cap c_i \neq \varnothing$.

Proof We first prove the necessity part. Note we write $x\alpha y$ instead of $x\alpha y \in \Gamma$ for simplicity.

Suppose $f \in \mathcal{I}_{\textsc{Face}}(v_i)$. There exists a landmark l such that $f \in \mathcal{I}_{\textsc{Face}}(l)$ and $v_i\mathbf{PPi}l$. Because $l \subseteq c_i$, the first statement holds directly.

Assume $f \in \mathcal{E}_{\textsc{Face}}(v_i)$. Clearly, we have $f \cap a_i = \varnothing$. Suppose $v_j\mathbf{PP}v_i$. By definition, there is a landmark l such that either (i) $f \in \mathcal{I}_{\textsc{Face}}(l)$ and $v_i\mathbf{DR}l$ or (ii) $f \in \mathcal{E}_{\textsc{Face}}(l)$ and $v_i\mathbf{PP}l$. In case (i), we have $v_j\mathbf{DR}l$ by path-consistency and thus $f \in \mathcal{E}_{\textsc{Face}}(v_j)$. In case (ii), we have $v_j\mathbf{PP}l$ by path-consistency and thus also $f \in \mathcal{E}_{\textsc{Face}}(v_j)$. Therefore we have $f \cap a_j = \varnothing$ and consequently $f \cap b_i = \varnothing$. Similarly, we can prove that f is not contained in any landmark l' if $l'\mathbf{PP}v_i$ due to path-consistency. Therefore, we have $f \cap c_i = \varnothing$ whenever $f \in \mathcal{E}_{\textsc{Face}}(v_i)$.

Now assume $f \in \mathcal{P}_{\textsc{Face}}(v_j)$. Clearly we have $f \cap a_j \neq \varnothing$ according to the first step. We only need to prove $f \nsubseteq c_i$. We have that $f \nsubseteq b_i$ as the union of all the base regions does not contain any face (by the requirements of base regions). Since $f \notin \mathcal{I}_{\textsc{Face}}(v_j)$, we know f is not contained in any landmark l with $v_i\mathbf{PPi}l$. This implies $f \nsubseteq c_i$.

The sufficiency part follows from the fact $\mathcal{I}_{\textsc{Face}}(v_i) \cup \mathcal{E}_{\textsc{Face}}(v_i) \cup \mathcal{P}_{\textsc{Face}}(v_i) = \textsc{Face}$ directly.

We next prove that $\{c_1, \cdots, c_t\}$ is a solution of Γ.

Lemma 6.6 *Suppose Γ is a basic RCC-5 constraint network involving landmarks L and variables V. Assume Γ satisfies the conditions in Theorem 6.4. Then $\{c_1, \cdots, c_t\}$, as constructed above, is a solution of Γ.*

Proof By the conditions in Theorem 6.4 we know $\mathcal{E}_{\textsc{Face}}(v_i) \neq \textsc{Face}$, i.e., there is at least one face f in $\mathcal{I}_{\textsc{Face}}(v_i) \cup \mathcal{P}_{\textsc{Face}}(v_i)$. By Lemma 6.5 we know c_i is nonempty. We next prove all constraints in Γ are satisfied.

We first consider the constraint $v_i\alpha l_j$ between variable v_i and landmark l_j. The cases that $\alpha = \mathbf{PP}, \mathbf{PP}^{\sim}, \mathbf{DR}$ can be directly checked by Lemma 6.5. Now suppose $v_i\mathbf{PO}l_j$. By Row 1 in Table 6.5, we know that $\mathcal{E}_{\textsc{Face}}(v_i) \cup \mathcal{E}_{\textsc{Face}}(l_j) \subsetneq \textsc{Face}$. That is, there is some face f in $\mathcal{I}_{\textsc{Face}}(l_j)$ but not in $\mathcal{E}_{\textsc{Face}}(v_i)$. By Lemma 6.5, we know $f \cap c_i \neq \varnothing$. By $f \subseteq l_j$, c_i and l_j have a common interior point. Furthermore, by $\mathcal{E}_{\textsc{Face}}(v_i) \cup \mathcal{I}_{\textsc{Face}}(l_j) \subsetneq \textsc{Face}$, we know there is a face f' in $\mathcal{E}_{\textsc{Face}}(l_j)$ that is not in $\mathcal{E}_{\textsc{Face}}(v_i)$. By $f' \in \mathcal{E}_{\textsc{Face}}(l_j)$, we have $f' \cap l_j = \varnothing$; by $f' \notin E(v_i)$ and Lemma 6.5, we have $f' \cap c_i \neq \varnothing$. So $c_i \nsubseteq l_j$. Similarly, we can show $l_j \nsubseteq c_i$. Therefore, $c_i\mathbf{PO}l_j$.

Now we consider constraints between two variables v_i and v_j.

(1) If $(v_i\mathbf{PP}v_j) \in \Gamma$, clearly we have $b_i \subseteq b_j$. By path-consistency, we know that $l\mathbf{PP}v_i$ implies $l\mathbf{PP}v_j$. Therefore, $c_i \subseteq c_j$. It remains to prove that $c_i \neq c_j$. By $\mathcal{I}_{\textsc{Face}}(v_i) \cup \mathcal{E}_{\textsc{Face}}(v_j) \subsetneq \textsc{Face}$ (Row 4 in Table 6.5), we know there is a face f that is in neither $\mathcal{I}_{\textsc{Face}}(v_i)$ nor $\mathcal{E}_{\textsc{Face}}(v_j)$. If $f \in \mathcal{E}_{\textsc{Face}}(v_i)$, we know that $c_i \neq c_j$ since $f \cap c_i = \varnothing$ and $f \cap c_j \neq \varnothing$. If $f \in \mathcal{P}_{\textsc{Face}}(v_i)$ and $f \in \mathcal{I}_{\textsc{Face}}(v_j)$, we have $f \nsubseteq c_i$ and $f \subseteq c_j$ and thus $c_i \neq c_j$. If $f \in \mathcal{P}_{\textsc{Face}}(v_i)$ and $f \in \mathcal{P}_{\textsc{Face}}(v_j)$, there exists some base region, say r, contained in f selected for v_j, thus $r \subseteq a_j \subseteq c_j$. Note that r is disjoint with a_i (due to the selection of base regions). Path-consistency implies that r is also disjoint with b_j or c_j. Therefore in such case we also have $c_i \neq c_j$. As a conclusion, $c_i \subsetneq c_j$.

(2) If $(v_i\mathbf{PP}^{\sim}v_j) \in \Gamma$, we know that Γ also contains constraint $(v_j\mathbf{PP}v_i)$. Because we have proved that $c_j\mathbf{PP}c_i$, constraint $v_i\mathbf{PP}^{\sim}v_j$ is also satisfied by c_i and c_j.

(3) If $(v_i\mathbf{DR}v_j) \in \Gamma$, we show that $c_i \cap c_j = \varnothing$. First it is clear that $a_i \cap a_j = \varnothing$ since all the base regions are pairwise disjoint and the constraint between v_i and v_j is not $\mathbf{PO}$. Note that if $v_k\mathbf{PP}v_i$, then we have $v_k\mathbf{DR}v_j$ by path-consistency and thus $a_k \cap a_i = \varnothing$. Therefore we have $b_i \cap a_j = \varnothing$. Similarly we can prove $b_i \cap b_j = \varnothing$ and $c_i \cap c_j = \varnothing$.

(4) If $(v_i\mathbf{PO}v_j) \in \Gamma$, we first show c_i and c_j have a common interior point. By $\mathcal{E}_{\textsc{Face}}(v_i) \cup \mathcal{E}_{\textsc{Face}}(v_j) \neq \textsc{Face}$ (Row 6 in Table 6.5), we have $\mathcal{I}_{\textsc{Face}}(v_i) \cup \mathcal{P}_{\textsc{Face}}(v_i)$ and $\mathcal{I}_{\textsc{Face}}(v_j) \cup \mathcal{P}_{\textsc{Face}}(v_j)$ are not disjoint. If $\mathcal{P}_{\textsc{Face}}(v_i) \cap \mathcal{P}_{\textsc{Face}}(v_j) \neq \varnothing$, there exists some base region contained both in a_i and a_j. Therefore, this base region is contained in both c_i and c_j. If $\mathcal{I}_{\textsc{Face}}(v_i) \cap \mathcal{P}_{\textsc{Face}}(v_j) \neq \varnothing$, suppose face f is in both $\mathcal{I}_{\textsc{Face}}(v_i)$ and $\mathcal{P}_{\textsc{Face}}(v_j)$. Then we know that $f \subseteq c_i$ and $f \cap c_j \neq \varnothing$, thus c_i and c_j have common interior point. Other two cases can be proved in the same way. It remains to show that c_i and c_j are incomparable (i.e., one is not contained in another), which can be proved in the same way as in the case of $(v_i\mathbf{PO}l_j)$. In conclusion, we know $c_i\mathbf{PO}c_j$.

In summary, all constraints are satisfied and $\{c_1, \cdots, c_t\}$ is a solution of Γ.

We now finish the sufficiency part of the proof: we have shown that, if the conditions are met, then we can construct a solution of the constraint network, therefore the constraint network is consistent.

Remark 6.3 *In fact the conditions for constraints of the form $v_i\mathbf{PP}l_j$, $v_i\mathbf{PPi}l_j$, $v_i\mathbf{PP}v_j$ are needed only because we need guarantee that the variable cannot be identical to a landmark (or another variable). The three conditions for the $v_i\mathbf{PO}l_j$ constraints* *guarantee that (i) v_i is not a proper subset of l_j, (ii) v_i is not a proper superset of l_j, and (iii) v_i may overlap with l_j (i.e., not all the faces in $\mathcal{I}_{\text{FACE}}(l_j)$ are excluded from v_i) respectively. The case for $v_i\mathbf{PO}v_j$ is analogous.*

Example 6.1 *In Example 2, we have that $\mathcal{I}_{\text{FACE}}(v_1) = \{f_1, f_2, f_3, f_4\}$ and $\mathcal{E}_{\text{FACE}}(v_1) = \varnothing$. There is constraint $v_1\mathbf{PO}v_3$, but $\mathcal{I}_{\text{FACE}}(v_1) \cap \mathcal{E}_{\text{FACE}}(l_3) = \text{FACE}$ which violates Row 1 in Table 6.5, so we know this network is inconsistent.*

6.4.2.3 Generalizing landmarks to finite domains in RCC-5

We have proved that the extended consistency problem $\text{CSPSAT}_s(\mathcal{B}_{\mathbf{RCC5}})$ is tractable, therefore $\text{CSPSAT}_f(\mathcal{B}_{\mathbf{RCC5}})$ is in NP. This section further shows that $\text{CSPSAT}_f(\mathcal{B}_{\mathbf{RCC5}})$ is NP-hard by devising a reduction from 3-SAT.

Suppose $\phi = \bigwedge_{j=1}^{m} c_j$ is a 3-SAT instance that involves n propositional variables $p_1, p_2, \cdots p_n$. We first take $2n$ pairwise disjoint rectangles $r_1^+, r_1^-, \cdots, r_n^+, r_n^-$ on the plane, and then construct a $\text{CSPSAT}_f(\mathcal{B}_{\mathbf{RCC5}})$ instance (V, Γ, D) in which the domains are all unions of the selected rectangles, where we suppose $\text{Var}(c_j) = \{p_{j_1}, p_{j_2}, p_{j_3}\}$ and where $a_{j_k}^*$ is $r_{j_k}^-$ if $p_{j_k} \in \text{Var}^+(c_j)$, and is $r_{j_k}^+$ if $p_{j_k} \in \text{Var}^-(c_j)$, for $k = 1, 2, 3$.

$$
\begin{aligned}
V &= \{v_1, v_2, \cdots, v_n\} \cup \{u_1, \cdots u_m\}, \\
D &= \{V_1, V_2, \cdots, V_n, U_1, U_2, \cdots, U_m\}, \text{where} \\
&\quad V_i = \{r_i^+, r_i^-\}, \\
&\quad U_j = \{a_{j_1} \cup a_{j_2} \cup a_{j_3} : a_{j_k} \in V_{j_k}\} - \{a_{j_1}^* \cup a_{j_2}^* \cup a_{j_3}^*\}, \\
\Gamma &= \{v_i\mathbf{DR}v_j : i \neq j\} \\
&\quad \cup \{u_i\mathbf{DR}u_j : \text{Var}(c_i) \cap \text{Var}(c_j) = \varnothing\} \\
&\quad \cup \{u_i\mathbf{PO}u_j : \text{Var}(c_i) \cap \text{Var}(c_j) \neq \varnothing\} \\
&\quad \cup \{v_i\mathbf{PP}u_j : p_i \in \text{Var}(c_j)\} \cup \{v_i\mathbf{DR}u_j : p_i \notin \text{Var}(c_j)\}.
\end{aligned}
$$

The above construction relates the truth value assigned to propositional variable p_i (trueor false) to the region assigned to spatial variable v_i (r_i^+ or r_i^-). Spatial variable u_j is required to contain spatial variables in set $\text{Var}(c_j) = \{p_{j_1}, p_{j_2}, p_{j_3}\}$ by the **PP** constraints. In fact, u_j is forced to be exactly the union of variables in $\text{Var}(c_j)$ according to the first term in U_j. Moreover, the second term in u_j forbids the one configuration among the eight which corresponds to the truth assignment that falsifies clause c_j.

It is routine to verify that ϕ is satisfiable if and only if the $\text{CSPSAT}_f(\mathcal{B}_{\mathbf{RCC5}})$ instance (V, Γ, D) is consistent. This reduction is polynomial, therefore $\text{CSPSAT}_f(\mathcal{B}_{\mathbf{RCC5}})$ is NP-hard.

Theorem 6.6 *The problem* $\text{CSPSAT}_f(\mathcal{B}_{RCC5})$ *is NP-complete, if the regions mentioned in the instance are (general) polygons.*

6.4.3 Introducing landmarks to RCC-8

This section investigates the consistency problem $\text{CSPSAT}_{\mathbf{RCC8},s}(\mathcal{B}_{\mathbf{RCC8}})$. First, we show that in general the problem is NP-hard due to the fact that two polygons may meet at discrete points. Second, we show that the problem is still in NP by providing a nondeterministic algorithm. We also show that the problem is still in another RCC model, where two regions are considered to be connected only if they share at least a curve. Note that the standard RCC model only requires two regions to share at least a point to regard them as being connected.

Note that the NP-completeness of $\text{CSPSAT}_{\mathbf{RCC8},s}(\mathcal{B}_{\mathbf{RCC8}})$ implies that consistency problems $\text{CSPSAT}_{\mathbf{RCC8},s}(\mathcal{B}^*_{\mathbf{RCC8}})$, $\text{CSPSAT}_{\mathbf{RCC8},f}(\mathcal{B}_{\mathbf{RCC8}})$ and $\text{CSPSAT}_{\mathbf{RCC8},f}(\mathcal{B}^*_{\mathbf{RCC8}})$ are also NP-complete.

6.4.3.1 The NP-hardness

Suppose Γ is a basic RCC-8 constraint network that involves no landmarks. Then Γ is consistent if it is path-consistent [78, 86]. Moreover, a solution can be constructed for each path-consistent basic network in cubic time [55, 56]. This section shows that, however, when considering polygons, it is NP-hard to determine if a basic RCC-8 network involving landmarks has a solution. We achieve this by devising a polynomial reduction from 3-SAT.

In this section, for clarity, we use upper case letters A, B, C (with indices) to denote landmarks (plane polygons), and use lower case letters u, v, w (with indices) to denote spatial variables.

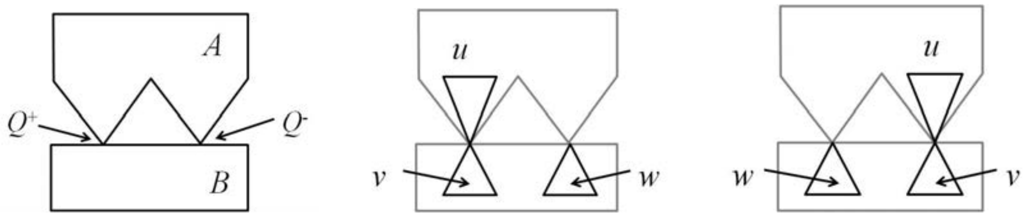
The NP-hardness exploits the fact that for two externally connected polygonal landmarks, say A, B, it is possible that

$$\mathcal{B}_{\text{EDGE}}(A) \cap \mathcal{B}_{\text{EDGE}}(B) = \varnothing \quad \text{and} \quad |\mathcal{B}_{\text{VTX}}(A) \cap \mathcal{B}_{\text{VTX}}(B)| > 1. \tag{6.12}$$

In this case assume v is a spatial variable that is required to be a tangentially proper part of A and externally connected with B. It is obvious that the meeting point(s) of v and B could be any meeting points of A and B. Furthermore let us consider the configuration shown in Fig. 6.6 (a), where A and B are two externally connected landmarks meeting at vertices Q^+ and Q^-. Assume $\{u, v, w\}$ are variables that are subject to constraints

$$u\mathbf{TPP}A, u\mathbf{EC}B,$$
$$v\mathbf{TPP}B, v\mathbf{EC}A, w\mathbf{TPP}B, w\mathbf{EC}A,$$
$$u\mathbf{EC}v, u\mathbf{DC}w, v\mathbf{DC}w.$$

In such case u is required to meet B at either Q^+ or Q^-, but not both (cf Fig. 6.6(b,c)). This correspondence between the two configurations and the two truth values (trueor false) of a propositional variable is exploited in the following reduction.



(a) (b) (c)

Figure 6.6: Landmarks externally connected at two tangential points

Let $\phi = \bigwedge_{k=1}^{m} c_k$ be a 3-SAT instance over propositional variables set $\{p_1, \cdots, p_n\}$. Each clause c_k has the form $p_r^* \vee p_s^* \vee p_t^*$, where literal p_i^* is either p_i or $\neg p_i$ for $i = r, s, t$. We next construct a set of polygons L and an instance $(V \cup L, \Gamma)$ of $\text{CSPSAT}_{\mathbf{RCC8},s}(\mathcal{B}_{\mathbf{RCC8}})$ such that $(V \cup L, \Gamma)$ is satisfiable iff Γ_ϕ is satisfiable.

First, we define $A, B_1, B_2, \cdots, B_n$ such that for each $1 \leq i \leq n$, A is externally connected to B_i at two vertices Q_i^+ and Q_i^-, as shown in Fig. 6.7.

The variable set of Γ is $V = \{u, v_1, \cdots, v_n, w_1, \cdots, w_n\}$. We impose the following

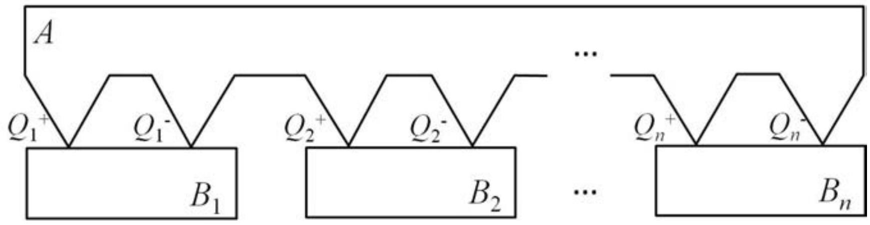


Figure 6.7: Illustration of landmarks $A, B_1, \cdots, B_n$.

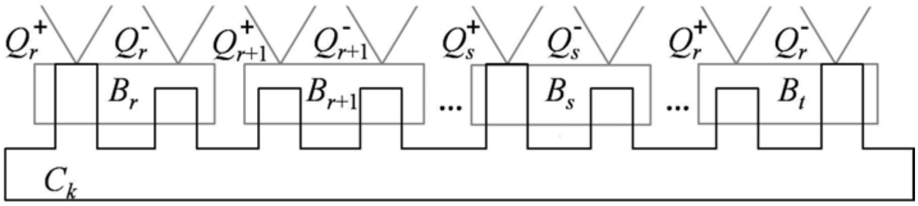


Figure 6.8: Illustration of landmark C_k.

constraints to the variables in V.

$$u\mathbf{TPP}A, \qquad u\mathbf{EC}B_i, \tag{6.13}$$

$$v_i\mathbf{EC}A, \qquad v_i\mathbf{TPP}B_i, \qquad v_i\mathbf{DC}B_j\ (j \neq i), \tag{6.14}$$

$$w_i\mathbf{EC}A, \qquad w_i\mathbf{TPP}B_i, \qquad w_i\mathbf{DC}B_j\ (j \neq i), \tag{6.15}$$

$$u\mathbf{EC}v_i, \qquad u\mathbf{DC}w_i, \tag{6.16}$$

$$v_i\mathbf{DC}w_j, \qquad v_i\mathbf{DC}v_j\ (j \neq i), \qquad w_i\mathbf{DC}w_j\ (j \neq i). \tag{6.17}$$

Therefore, u is required to meet each B_i at either Q_i^- or Q_i^+ but not both.

For each clause c_k, we introduce an additional landmark C_k, which externally connects A at three vertices (determined by the literals in c_k), and partially overlaps B_i. Formally, suppose $c_k = p_r^* \vee p_s^* \vee p_t^*$, then the first meeting vertices of A and C_k is constructed to be Q_r^+ if $p_r^* = p_r$, or Q_r^- if $p_r^* = \neg p_r$. The second and the third meeting vertices of A and C_k are selected from $\{Q_s^+, Q_s^-\}$ and $\{Q_t^+, Q_t^-\}$ similarly. Take clause $p_r \vee \neg p_s \vee p_t$ for example, the meeting vertices of landmarks C_k and A should be Q_r^+, Q_s^-, and Q_t^+, as shown in Fig. 6.8.

The constraints between C_k and variables in V are specified as

$$u\mathbf{EC}C_k, \quad v_i\mathbf{PO}C_k, \quad w_i\mathbf{PO}C_k. \tag{6.18}$$

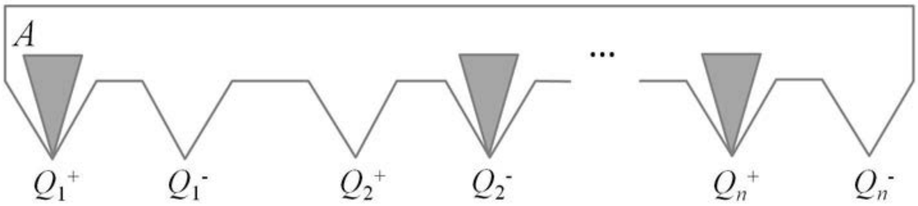


Figure 6.9: Construction of variable u.

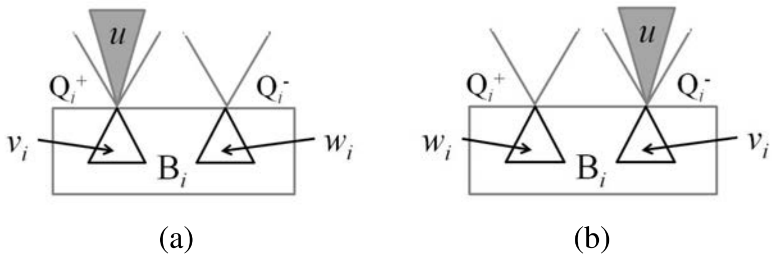


Figure 6.10: Construction of variable v_i and w_i. $\pi(p_i) = \mathsf{true}$ (a), $\pi(p_i) = \mathsf{false}$ (b).

Since C_k and A meets only at three vertices, the constraints $u\mathbf{TPP}A$ and $u\mathbf{EC}C_k$ imply that u should occupy at least one of the three. This corresponds to the fact that if c_k is trueunder some assignment, then at least one of its three literals is assigned true.

Lemma 6.7 *Suppose $\phi = \bigwedge_{k=1}^{m} c_k$ is a 3-SAT instance over propositional variables set $\{p_1, p_2, \cdots, p_n\}$. Let Γ_ϕ be the basic RCC-8 network composed with constraints in (6.13)-(6.18), involving landmarks $\{A, B_1, \cdots, B_n, C_1, \cdots, C_m\}$ and spatial variables $\{u, v_1, \cdots, v_n, w_1, \cdots, w_n\}$. Then ϕ is satisfiable iff Γ_ϕ is satisfiable.*

Proof Suppose ϕ is satisfiable and $\pi : P \to \{\mathsf{true}, \mathsf{false}\}$ is a truth value assignment that satisfies ϕ. We construct regions $\bar{u}, \bar{v}_1, \cdots, \bar{v}_n, \bar{w}_1, \cdots, \bar{w}_m$ that satisfy all constraints in Γ_ϕ.

Region $\bar{u}$ is composed of n pairwise disjoint triangles in A. The lower vertex of the i-th triangle is Q_i^+ if $\pi(p_i) = \mathsf{true}$, and Q_i^- otherwise. Fig. 6.9 shows the case that $\pi(p_1) = \mathsf{true}, \pi(p_2) = \mathsf{false}, \pi(p_n) = \mathsf{true}$.

Regions $\bar{v}_i$ and $\bar{w}_i$ are constructed as, respectively, a triangle inside B_i. If $\pi(p_i) = \mathsf{true}$, then Q_i^+ is a vertex of $\bar{v}_i$ and Q_i^- is a vertex of $\bar{w}_i$ (see Fig. 6.10(a)). Oppositely, if $\pi(p_i) = \mathsf{false}$, then Q_i^- is a vertex of $\bar{v}_i$ and Q_i^+ is a vertex of $\bar{w}_i$ (see Fig. 6.10(b)). Moreover, $\bar{v}_i$ and $\bar{w}_i$ are properly chosen to make them partially overlap with each C_k.

By the construction, it is easy to see that all constraints in (6.13)-(6.18), except $u\mathbf{EC}C_k$, are satisfied. We next show $u\mathbf{EC}C_k$ is also satisfied. That is, $\bar{u}\mathbf{EC}C_k$. Because π satisfies ϕ, it also satisfies c_k. That is, at least one of the three literals in c_k, say p_r^*, is trueunder the assignment π. If $p_r^* = p_r$, then Q_r^+ is at the boundary of C_k by construction. In this case, we have $\pi(p_r) = \mathsf{true}$. By the construction of $\bar{u}$, we know Q_r^+ is also at the boundary of $\bar{u}$. Similarly, if $p_r^* = \neg p_r$, then we can prove Q_r^- is a meeting vertex of C_k and $\bar{u}$. Therefore, in both cases, the RCC-8 relation between C_k and u is $\mathbf{EC}$. This shows that the constructed regions $\bar{u}, \bar{v}_1, \cdots, \bar{v}_n, \bar{w}_1, \cdots, \bar{w}_m$ satisfy all constraints in Γ_ϕ. Hence, Γ_ϕ is satisfiable.

On the other hand, suppose $\{\bar{u}, \bar{v}_1, \cdots, \bar{v}_n, \bar{w}_1, \cdots, \bar{w}_n\}$ is a solution of the network Γ_ϕ. It is straightforward to verify that $\bar{v}_i$ has exactly one meeting point with A, namely either Q_i^- or Q_i^+. We define a truth value assignment $\pi : P \to \{\mathsf{true}, \mathsf{false}\}$ as

$$\pi(p_i) = \begin{cases} \mathsf{true}, & \text{if } \bar{v}_i \cap A = Q_i^+, \\ \mathsf{false}, & \text{otherwise.} \end{cases} \tag{6.19}$$

We assert that $\pi(c_k) = \mathsf{true}$ for each c_k in ϕ. Otherwise, suppose $\pi(c_k) = \mathsf{false}$ for some $c_k = p_r^* \vee p_s^* \vee p_t^*$ in ϕ. This only happens when $\pi(p_i^*) = \mathsf{false}$ for $i = r, s, t$. Therefore, for $i = r, s, t$, if p_i^* is positive, then by (6.19), we know that $\bar{v}_i \cap A = Q_i^-$. Since $\bar{u} \subset A$ and $\bar{v}_i\mathbf{EC}\bar{u}$, we have $Q_i^- \in \bar{u}$, which implies Q_i^+ is not in $\bar{u}$. Similarly if p_i^* is negative, then Q_i^- is not in $\bar{u}$. This is to say, all the three meeting vertices of A and C_k are not in $\bar{u}$, which contradicts with $\bar{u}\mathbf{EC}C_k$. Therefore, ϕ is satisfiable.

Lemma 6.8 *The consistency problem* CSPSAT$_s(\mathcal{B}_{\boldsymbol{RCC8}})$ *are NP-hard.*

6.4.3.2 A nondeterministic algorithm

This section shows that the extended consistency problem of basic RCC-8 networks is still in NP. Recall that we write $\mathcal{O}$ for the overlay of all landmarks, and we have defined

$$\mathcal{I}_{\text{FACE}}(l_i), \mathcal{E}_{\text{FACE}}(l_i), \mathcal{I}_{\text{EDGE}}(l_i), \mathcal{E}_{\text{EDGE}}(l_i), \mathcal{B}_{\text{EDGE}}(l_i), \mathcal{I}_{\text{VTX}}(l_i), \mathcal{E}_{\text{VTX}}(l_i), \mathcal{B}_{\text{VTX}}(l_i)$$

for representing the topological relationship between the faces, edges, vertices in $\mathcal{O}$ and landmark l_i. We have also defined $\mathcal{I}_{\text{FACE}}(v_i)$ and $\mathcal{E}_{\text{FACE}}(v_i)$ for the faces that should be part of v_i and the faces that should excluded from v_i, respectively.

For the same consideration, we now define $\mathcal{I}_{\text{EDGE}}(v_i)$ as the set of edges that should lie in the interior of v_i. For an edge e, if both of its adjacent faces (i.e., $S_{\text{FACE}}(e)$) are in $\mathcal{I}_{\text{FACE}}(v_i)$, then edge e is consequently required to be in the interior of v_i. Furthermore, constraint of the form $l_j\mathbf{NTPP}v_i$ also requires that all the boundary edges of l_j (i.e., $\mathcal{B}_{\text{EDGE}}(l_j)$) should lie in the interior of v_i. Similarly, $\mathcal{E}_{\text{EDGE}}(v_i)$ is defined as the set of edges that should be excluded from v_i. We also write $\mathcal{B}_{\text{EDGE}}(v_i)$ for the set of edges that are required to be parts of boundary of v_i, which only happens when its two adjacent faces are in $\mathcal{I}_{\text{FACE}}(v_i)$ and $\mathcal{E}_{\text{FACE}}(v_i)$ respectively. The formal definitions of $\mathcal{I}_{\text{EDGE}}(v_i)$, $\mathcal{E}_{\text{EDGE}}(v_i)$ and $\mathcal{B}_{\text{EDGE}}(v_i)$ are as follows.

$$\mathcal{I}_{\text{EDGE}}(v_i) :=\{e \in \text{EDGE} : S_{\text{FACE}}(e) \subseteq \mathcal{I}_{\text{FACE}}(v_i)\} \cup \bigcup\{\mathcal{B}_{\text{EDGE}}(l_j) : v_i\mathbf{NTPPi}l_j\}, \tag{6.20}$$

$$\mathcal{E}_{\text{EDGE}}(v_i) :=\{e \in \text{EDGE} : S_{\text{FACE}}(e) \subseteq \mathcal{E}_{\text{FACE}}(v_i)\} \cup \bigcup\{\mathcal{B}_{\text{EDGE}}(l_j) : v_i\mathbf{DC}\text{ or }\mathbf{NTPP}l_j\}. \tag{6.21}$$

$$\mathcal{B}_{\text{EDGE}}(v_i) :=\{e \in \text{EDGE} : S_{\text{FACE}}(e) \cap \mathcal{I}_{\text{FACE}}(v_i) \neq \varnothing, S_{\text{FACE}}(e) \cap \mathcal{E}_{\text{FACE}}(v_i) \neq \varnothing\}. \tag{6.22}$$

For the vertices, $\mathcal{I}_{\text{VTX}}(v_i)$, $\mathcal{E}_{\text{VTX}}(v_i)$ and $\mathcal{B}_{\text{VTX}}(v_i)$ are defined in the same way as $\mathcal{I}_{\text{EDGE}}(v_i)$, $\mathcal{E}_{\text{EDGE}}(v_i)$ and $\mathcal{B}_{\text{EDGE}}(v_i)$

$$\mathcal{I}_{\text{VTX}}(v_i) :=\{v \in \text{VTX} : S_{\text{FACE}}(v) \subseteq \mathcal{I}_{\text{FACE}}(v_i)\} \cup \bigcup\{\mathcal{B}_{\text{VTX}}(l_j) : v_i\mathbf{NTPPi}l_j\}, \tag{6.23}$$

$$\mathcal{E}_{\text{VTX}}(v_i) :=\{v \in \text{VTX} : S_{\text{FACE}}(v) \subseteq \mathcal{E}_{\text{FACE}}(v_i)\} \cup \bigcup\{\mathcal{B}_{\text{VTX}}(l_j) : v_i\mathbf{DC}\text{ or }\mathbf{NTPP}l_j\}, \tag{6.24}$$

$$\mathcal{B}_{\text{VTX}}(v_i) :=\{v \in \text{VTX} : S_{\text{FACE}}(v) \cap \mathcal{I}_{\text{FACE}}(v_i) \neq \varnothing, S_{\text{FACE}}(v) \cap \mathcal{E}_{\text{FACE}}(v_i) \neq \varnothing\}. \tag{6.25}$$

For convenience, we introduce the following abbreviations ($\mathcal{P}$ standing for 'pending').

$$\mathcal{P}_{\text{FACE}}(v_i) := \text{FACE} - \mathcal{I}_{\text{FACE}}(v_i) - \mathcal{E}_{\text{FACE}}(v_i), \tag{6.26}$$

$$\mathcal{P}_{\text{EDGE}}(v_i) := \text{EDGE} - \mathcal{I}_{\text{EDGE}}(v_i) - \mathcal{E}_{\text{EDGE}}(v_i), \tag{6.27}$$

$$\mathcal{P}_{\text{VTX}}(v_i) := \text{VTX} - \mathcal{I}_{\text{VTX}}(v_i) - \mathcal{E}_{\text{VTX}}(v_i). \tag{6.28}$$

Note that $\mathcal{B}_{\text{VTX}}(v_i)$ is the set of vertices that are required to lie on the boundary of v_i, which is because that, for any $v \in \mathcal{B}_{\text{VTX}}(v_i)$, there are faces f and f' in $S_{\text{FACE}}(v)$ such that $f \in \mathcal{I}_{\text{FACE}}(v_i)$ and $f' \in \mathcal{E}_{\text{FACE}}(v_i)$ (i.e., f should be part of v_i and f' should lie outside of v_i). Meanwhile, $\mathcal{P}_{\text{VTX}}(v_i)$ contains all the vertices that might lie on the boundary of v_i (i.e., the vertices not in $\mathcal{I}_{\text{VTX}}(v_i) \cup \mathcal{E}_{\text{VTX}}(v_i)$).

Proposition 6.7 *Suppose* Γ *is a basic RCC-8 constraint network involving landmarks* $L = \{l_1, \cdots, l_m\}$ *and variables* $V = \{v_1, \cdots, v_n\}$. *If* Γ *is path-consistent, then for each variable* v_i*,*

- $\mathcal{I}_{\text{FACE}}(v_i) \cap \mathcal{E}_{\text{FACE}}(v_i) = \varnothing$.
- $\mathcal{I}_{\text{VTX}}(v_i)$*,* $\mathcal{E}_{\text{VTX}}(v_i)$*, and* $\mathcal{B}_{\text{VTX}}(v_i)$ *are pairwise disjoint.*
- $\mathcal{I}_{\text{EDGE}}(v_i)$*,* $\mathcal{E}_{\text{EDGE}}(v_i)$*, and* $\mathcal{B}_{\text{EDGE}}(v_i)$ *are pairwise disjoint.*

Proof The first conclusion can be proved in the same way as in Proposition 6.6, and the rest two can be proved similarly.

Here we only show that $\mathcal{I}_{\text{VTX}}(v_i) \cap \mathcal{B}_{\text{VTX}}(v_i) = \varnothing$. Otherwise, suppose some vertex $v \in \text{VTX}$ is contained in both $\mathcal{I}_{\text{VTX}}(v_i)$ and $\mathcal{B}_{\text{VTX}}(v_i)$. Therefore, it follows from $v \in \mathcal{B}_{\text{VTX}}(v_i)$ that both $\mathcal{I}_{\text{VTX}}(v_i)$ and $\mathcal{E}_{\text{VTX}}(v_i)$ have some surrounding face of v. Meanwhile, $v \in \mathcal{I}_{\text{VTX}}(v_i)$ implies that either (i) all the surrounding faces of v are in $\mathcal{I}_{\text{FACE}}(v_i)$, or (ii) there exists some l_j such that $v_i\mathbf{NTPPi}l_j$ and v is on the boundary of l_j. Because we already have $\mathcal{I}_{\text{VTX}}(v_i)$ and $\mathcal{E}_{\text{VTX}}(v_i)$ are disjoint, case (i) is impossible. We next prove that case (ii) is also impossible.

As $\mathcal{E}_{\text{VTX}}(v_i)$ has some surrounding face, say f, of v, by the definition of $\mathcal{E}_{\text{VTX}}(v_i)$, we know that there exists some landmark l_k such that either (1) face f is contained by l_k, and $v_i\mathbf{DC}l_k$ or $v_i\mathbf{EC}l_k$, or (2) f lies outside of l_k, and $v_i\mathbf{NTPP}l_k$ or $v_i\mathbf{TPP}l_k$. We show in both subcases the constraints between v_i, l_j, l_k are not path-consistent. In subcase (1), we have shown that vertex v is on the boundary of l_j and contained by (i.e., inside or on the boundary of) l_k. Therefore, l_j and l_k are connected, which is not path-consistent with constraints $v_i\mathbf{NTPPi}l_j$ and $v_i\mathbf{DR}l_k$. In subcase (2), we have that vertex v is on the boundary of l_j and outside l_k. So l_j cannot be a proper part of l_k, but this is not path-consistent with constraints $l_j\mathbf{NTPP}v_i$ and $v_i\mathbf{PP}l_k$. Therefore both subcases violate the assumption of the path-consistency of Γ, hence we have $\mathcal{I}_{\text{VTX}}(v_i) \cap \mathcal{B}_{\text{VTX}}(v_i) = \varnothing$.

Note that the second property implies that $\mathcal{B}_{\text{VTX}}(v_i) \subseteq \mathcal{P}_{\text{VTX}}(v_i)$.

Suppose the network is consistent and has a solution, say $\bar{v}_1, \bar{v}_2, \cdots, \bar{v}_n$. Let $\bar{S}_i$ denote the set of vertices in VTX that lie on the boundary of $\bar{v}_i$. Clearly we have

$$\mathcal{B}_{\text{VTX}}(v_i) \subseteq \bar{S}_i \subseteq \mathcal{P}_{\text{VTX}}(v_i).$$

Furthermore we have the following lemma.

Lemma 6.9 *Suppose Γ is a basic RCC-8 network with variable set $V = \{v_1, v_2, \cdots, v_n\}$ and landmark set $L = \{l_1, l_2, \cdots, l_m\}$. $\mathcal{B}_{\text{VTX}}(v_i)$ and $\mathcal{P}_{\text{VTX}}(v_i)$ are as defined in Eq.s 6.25 and 6.28. Suppose Γ is path-consistent, and let S_i be a subset of* VTX *such that $\mathcal{B}_{\text{VTX}}(v_i) \subseteq S_i \subseteq \mathcal{P}_{\text{VTX}}(v_i)$ for $i = 1, 2, \cdots, n$. Then the problem:*

whether there is a solution $\{\bar{v}_1, \bar{v}_2, \cdots, \bar{v}_n\}$ such that $\partial\bar{v}_i \cap \text{VTX} = S_i$

can be decided in polynomial time, by checking the following conditions.

- *For any $v_i \in V$, $\mathcal{E}_{\text{FACE}}(v_i) \neq$ FACE.*
- *All the conditions in Table 6.6 hold, where*

$$\mathcal{P}_{\text{EDGE}}(v_i) \cap \mathcal{B}_{\text{EDGE}}(l_j) \neq \varnothing \quad or \quad S_i \cap \mathcal{B}_{\text{VTX}}(l_j) \neq \varnothing, \tag{6.29}$$

$$\mathcal{P}_{\text{FACE}}(v_i) \cap \mathcal{P}_{\text{FACE}}(v_j) \neq \varnothing \quad or \quad \mathcal{P}_{\text{EDGE}}(v_i)\mathcal{P}_{\text{EDGE}}(v_j) \neq \varnothing \quad or \quad S_i \cap S_j \neq \varnothing. \tag{6.30}$$

As it is similar to the RCC-5 case, we here first prove the necessity part. The constructive proof of sufficiency is provide later.

Proof Suppose the extended has a solution $\{\bar{v}_1, \bar{v}_2, \cdots, \bar{v}_n\}$ such that $\bar{v}_i \cap \text{VTX} = S_i$. It is straightforward to see that Γ must be path-consistent, and for any $v_i \in V$, $\mathcal{E}_{\text{FACE}}(v_i) \neq$ FACE (as in the RCC-5 case). We next show that all the conditions in the table also hold.

First we argue that for any face f in $\mathcal{I}_{\text{FACE}}(v_i)$, we have $f \subseteq (\bar{v}_i)^\circ$. This is clear because by the definition of $\mathcal{I}_{\text{FACE}}(v_i)$ (see Eq. 6.10), there exists some landmark l_k such that $f \in \mathcal{I}_{\text{FACE}}(l_k)$ and v_i**TPPi**l_k or v_i**NTPPi**l_k. So we know that $f \subseteq l_k^\circ$ and $l_k \subseteq \bar{v}_i$, which implies $f \subseteq (\bar{v}_i)^\circ$. It can be similarly proved that for any face f' in $\mathcal{E}_{\text{FACE}}(v_i)$, we have $f' \cap \bar{v}_i = \varnothing$.

Constraint	Conditions
$v_i\mathbf{EC}l_j$	Eq. 6.29
$v_i\mathbf{PO}l_j$	$\mathcal{E}_{\text{FACE}}(v_i) \cup \mathcal{E}_{\text{FACE}}(l_j) \neq \text{FACE}$, $\mathcal{E}_{\text{FACE}}(v_i) \cup \mathcal{I}_{\text{FACE}}(l_j) \neq \text{FACE}$, $\mathcal{I}_{\text{FACE}}(v_i) \cup \mathcal{E}_{\text{FACE}}(l_j) \neq \text{FACE}$
$v_i\mathbf{TPP}l_j$	$\mathcal{I}_{\text{FACE}}(v_i) \neq \mathcal{I}_{\text{FACE}}(l_j)$ and Eq. 6.29
$v_i\mathbf{TPPi}l_j$	$\mathcal{E}_{\text{FACE}}(v_i) \neq \mathcal{E}_{\text{FACE}}(l_j)$ and Eq. 6.29
$v_i\mathbf{DC}v_j$	$S_i \cap S_j = \varnothing$
$v_i\mathbf{EC}v_j$	Eq. 6.30
$v_i\mathbf{PO}v_j$	$\mathcal{E}_{\text{FACE}}(v_i) \cup \mathcal{E}_{\text{FACE}}(v_j) \neq \text{FACE}$, $\mathcal{E}_{\text{FACE}}(v_i) \cup \mathcal{I}_{\text{FACE}}(v_j) \neq \text{FACE}$, $\mathcal{I}_{\text{FACE}}(v_i) \cup \mathcal{E}_{\text{FACE}}(v_j) \neq \text{FACE}$
$v_i\mathbf{TPP}v_j$	$\mathcal{P}_{\text{FACE}}(v_j) \neq \varnothing$ or $\mathcal{I}_{\text{FACE}}(v_i) \neq \mathcal{I}_{\text{FACE}}(v_j)$, and Eq. 6.30

Table 6.6: Conditions for extended RCC-5 constraint network

Second we show that for any edge e in $\mathcal{I}_{\text{EDGE}}(v_i)$, we have $e \subseteq (\bar{v}_i)^{\circ}$. From the definition of $\in \mathcal{I}_{\text{EDGE}}(v_I)$ (see Eq. 6.20), it is either the case that $S_{\text{FACE}}(e) \subseteq \mathcal{I}_{\text{EDGE}}(v_i)$, or $e \in \mathcal{B}_{\text{EDGE}}(l_k)$ and $v_i\mathbf{NTPPi}l_k$. For the first case, the two neighbouring faces of e are both in $\mathcal{I}_{\text{FACE}}(v_i)$ hence contained in the interior of $\bar{v}_i$, so e is also contained in the interior of $\bar{v}_i$. For the second case we have $e \subseteq l_k \subseteq (\bar{v}_i)^{\circ}$. Therefore $e \subseteq (\bar{v}_i)^{\circ}$ in both cases. Similarly for any edge e' in $\mathcal{E}_{\text{EDGE}}(v_i)$, we have $e' \cap \bar{v}_i = \varnothing$.

In the same way we can prove the for any vertex v in $\mathcal{I}_{\text{VTX}}(v_i)$, $v \in (\bar{v}_i)^{\circ}$, and for any vertex v in $\mathcal{E}_{\text{VTX}}(v_i)$ $v \notin \bar{v}_i$.

Now suppose $\bar{v}_i\mathbf{EC}l_j$, so there is a point P is on $\partial l_j \cap \partial \bar{v}_i$. It is clear that P is either a vertex in $\mathcal{B}_{\text{VTX}}(l_j)$, or on an edge which is in $\mathcal{B}_{\text{EDGE}}(l_j)$. In the first case, we know that $P \in S_i$ because $\partial \bar{v}_i \cap \text{VTX} = S_i$. Therefore $P \in S_i \cap \mathcal{B}_{\text{VTX}}(l_j)$ and Eq. 6.29 is satisfied. In the second case, suppose $p \in e \in \mathcal{B}_{\text{EDGE}}(l_j)$ where e is an edge. Because edge e is neither in the interior of $\bar{v}_i$ nor in the exterior of $\bar{v}_i$ (see above), we know that $e \in \mathcal{P}_{\text{EDGE}}(v_i)$. Therefore $\mathcal{P}_{\text{EDGE}}(v_i) \cap \mathcal{B}_{\text{EDGE}}(l_j) \neq \varnothing$.

The rest cases can be proved in the similar way.

We now constructed a solution which satisfies the requirement in Lemma 6.9, provided all the conditions in the lemma are met. For each variable v_i, we select a set of small triangles, denoted by X_i, in the following way.

- For each face $f \in \mathcal{P}_{\text{FACE}}(v_i)$, select a small triangle in the interior of $\mathcal{P}_{\text{FACE}}(v_i)$ and put it in X_i, see Figure 6.11(a).

- For each vertex $v \in S_i - \mathcal{B}_{\text{VTX}}(v_i)$, it is clear that v is not in $\mathcal{B}_{\text{VTX}}(v_i) \cup \mathcal{I}_{\text{VTX}}(v_i) \cup \mathcal{E}_{\text{VTX}}(v_i)$ by Proposition 6.7. Therefore, there is some face $f \in S_{\text{FACE}}(v)$ such that $f \in \mathcal{P}_{\text{FACE}}(v_i)$. Select a small triangle in face f which has v as one of its vertices. Put this triangle in X_i, see Figure 6.11(b).
- For constraint $v_i\mathbf{EC}l_j$, we have that $\mathcal{P}_{\text{EDGE}}(v_i) \cap \mathcal{B}_{\text{EDGE}}(l_j) \neq \varnothing$ or $S_i \cap \mathcal{B}_{\text{VTX}}(l_j) \neq \varnothing$ (Row 1 in Table 6.6).
  - Case 1. $S_i \cap \mathcal{B}_{\text{VTX}}(l_j) \neq \varnothing$. Do nothing. Otherwise,
  - Case 2. Select edge e from $\mathcal{P}_{\text{EDGE}}(v_i) \cap \mathcal{B}_{\text{EDGE}}(l_j)$. Let f and f' be the two neighbouring faces of e such that $f \in \mathcal{I}_{\text{FACE}}(l_j)$ and $f' \in \mathcal{E}_{\text{FACE}}(l_j)$. If $f' \in \mathcal{I}_{\text{FACE}}(v_i)$, do nothing. If $f' \in \mathcal{P}_{\text{FACE}}(v_i)$, select a triangle in face f' with one of its edges on e and put this triangle in X_i, see Figure 6.11(c). Note that f' cannot be in $\mathcal{E}_{\text{FACE}}(v_i)$, otherwise we can prove e is in $\mathcal{E}_{\text{FACE}}(v_i)$.
- For constraint $v_i\mathbf{TPP}l_j$, we also have $\mathcal{P}_{\text{EDGE}}(v_i) \cap \mathcal{B}_{\text{EDGE}}(l_j) \neq \varnothing$ or $S_i \cap \mathcal{B}_{\text{VTX}}(l_j) \neq \varnothing$ (Row 3 in Table 6.6).
  - Case 1. $S_i \cap \mathcal{B}_{\text{VTX}}(l_j) \neq \varnothing$. Do nothing. Otherwise,
  - Case 2. Select edge e from $\mathcal{P}_{\text{EDGE}}(v_i) \cap \mathcal{B}_{\text{EDGE}}(l_j)$. Let f and f' be the two neighbouring faces of e such that $f \in \mathcal{I}_{\text{FACE}}(l_j)$ and $f' \in \mathcal{E}_{\text{FACE}}(l_j)$. If $f \in \mathcal{I}_{\text{FACE}}(v_i)$ do nothing. If $f \in \mathcal{P}_{\text{FACE}}(v_i)$, select a triangle in f with one edge on e and put this triangle in X_i. Note that f cannot be in $\mathcal{E}_{\text{FACE}}(v_i)$.
- For constraint $v_i\mathbf{EC}v_j$, we have $\mathcal{P}_{\text{FACE}}(v_i) \cap \mathcal{P}_{\text{FACE}}(v_j) \neq \varnothing$, or $\mathcal{P}_{\text{EDGE}}(v_i) \cap \mathcal{P}_{\text{EDGE}}(v_j) \neq \varnothing$, or $S_i \cap S_j \neq \varnothing$ (Row 6 in Table 6.6).
  - Case 1. $S_i \cap S_j \neq \varnothing$, do nothing. Otherwise,
  - Case 2. $\mathcal{P}_{\text{FACE}}(v_i) \cap \mathcal{P}_{\text{FACE}}(v_j) \neq \varnothing$, select a face $f \in \mathcal{P}_{\text{FACE}}(v_i) \cap \mathcal{P}_{\text{FACE}}(v_j)$ and two externally connected triangles in f. Put one triangle in X_i and another in X_j, see Figure 6.11(d). Otherwise,
  - Case 3. $\mathcal{P}_{\text{EDGE}}(v_i) \cap \mathcal{P}_{\text{EDGE}}(v_j) \neq \varnothing$. Select edge $e \in \mathcal{P}_{\text{EDGE}}(v_i) \cap \mathcal{P}_{\text{EDGE}}(v_j)$ with neighbouring faces f and f'. Here we have four subcases depending on whether e is in $\mathcal{B}_{\text{EDGE}}(v_i)$ and $\mathcal{B}_{\text{EDGE}}(v_j)$.

* Subcase 1. $e \in \mathcal{B}_{\text{EDGE}}(v_i)$ and $e \in \mathcal{B}_{\text{EDGE}}(v_j)$. Do nothing.
* Subcase 2. $e \in \mathcal{B}_{\text{EDGE}}(v_i)$ and $e \notin \mathcal{B}_{\text{EDGE}}(v_j)$. Suppose $f \in \mathcal{I}_{\text{FACE}}(v_i)$ and $f' \in \mathcal{E}_{\text{FACE}}(v_i)$. Select a triangle in face f' with one edge on e, and put this triangle in X_j.
* Subcase 3. $e \notin \mathcal{B}_{\text{EDGE}}(v_i)$ and $e \in \mathcal{B}_{\text{EDGE}}(v_j)$. The same as Subcase 2, except the exchange of v_i and v_j.
* Subcase 4. $e \notin \mathcal{B}_{\text{EDGE}}(v_i)$ and $e \notin \mathcal{B}_{\text{EDGE}}(v_j)$. Select two triangles in f and f' respectively such that the triangles have a common edge on e, see Figure 6.11(e)

- For constraint $v_i\mathbf{TPP}v_j$, we also have $\mathcal{P}_{\text{FACE}}(v_i) \cap \mathcal{P}_{\text{FACE}}(v_j) \neq \emptyset$, or $\mathcal{P}_{\text{EDGE}}(v_i)\mathcal{P}_{\text{EDGE}}(v_j) \neq \emptyset$, or $S_i \cap S_j \neq \emptyset$ (Row 8 in Table 6.6).
  - Case 1. $S_i \cap S_j \neq \emptyset$, do nothing. Otherwise,
  - Case 2. $\mathcal{P}_{\text{FACE}}(v_i) \cap \mathcal{P}_{\text{FACE}}(v_j) \neq \emptyset$, select face $f \in \mathcal{P}_{\text{FACE}}(v_i) \cap \mathcal{P}_{\text{FACE}}(v_j)$ and one triangle in f. Put the triangle in both X_i and X_j. Otherwise,
  - Case 3. $\mathcal{P}_{\text{EDGE}}(v_i) \cap \mathcal{P}_{\text{EDGE}}(v_j) \neq \emptyset$. Select edge $e \in \mathcal{P}_{\text{EDGE}}(v_i) \cap \mathcal{P}_{\text{EDGE}}(v_j)$ with neighbouring faces f and f'. We can prove that either $f \in \mathcal{E}_{\text{FACE}}(v_i)$ or $f' \in \mathcal{E}_{\text{FACE}}(v_i)$ [1]. W.l.o.g., we assume $f \in \mathcal{E}_{\text{FACE}}(v_i)$. If $f' \in \mathcal{I}_{\text{FACE}}(v_i)$, do nothing. Otherwise $f' \in \mathcal{P}_{\text{FACE}}(v_i)$, and we select a triangle in f' with one edge on e. Put this triangle in X_i.

- For constraint $v_i\mathbf{PO}v_j$, we have $\mathcal{E}_{\text{FACE}}(v_i) \cup \mathcal{E}_{\text{FACE}}(v_j) \neq \text{FACE}$ (Row 7 in Table 6.6), which means there is some face f in $(\mathcal{I}_{\text{FACE}}(v_i) \cup \mathcal{P}_{\text{FACE}}(v_i)) \cap (\mathcal{I}_{\text{FACE}}(v_j) \cup \mathcal{P}_{\text{FACE}}(v_j))$. If f is in $\mathcal{P}_{\text{FACE}}(v_i)$ and in $\mathcal{P}_{\text{FACE}}(v_j)$, select a triangle in face f and put the triangle in both X_i and X_j. In any other case, do nothing.

Note that all the triangles are small enough that they are pairwise disjoint and the union of all the triangle does not entirely occupy any face or any edge. Now X_i contains all the triangles we need for variable v_i. Denote X for the union of all X_i, i.e., X is the set of all the triangles selected for variables.

[1] It is clear that $f \in \mathcal{I}_{\text{FACE}}(v_i)$ or $f \in \mathcal{P}_{\text{FACE}}(v_i)$ implies $f \in \mathcal{I}_{\text{FACE}}(v_j)$. So if neither f nor f' is in $\mathcal{E}_{\text{FACE}}(v_i)$, we know that f and f' are both in $\mathcal{I}_{\text{FACE}}(v_j)$, which implies $e \in \mathcal{I}_{\text{EDGE}}(v_j)$.

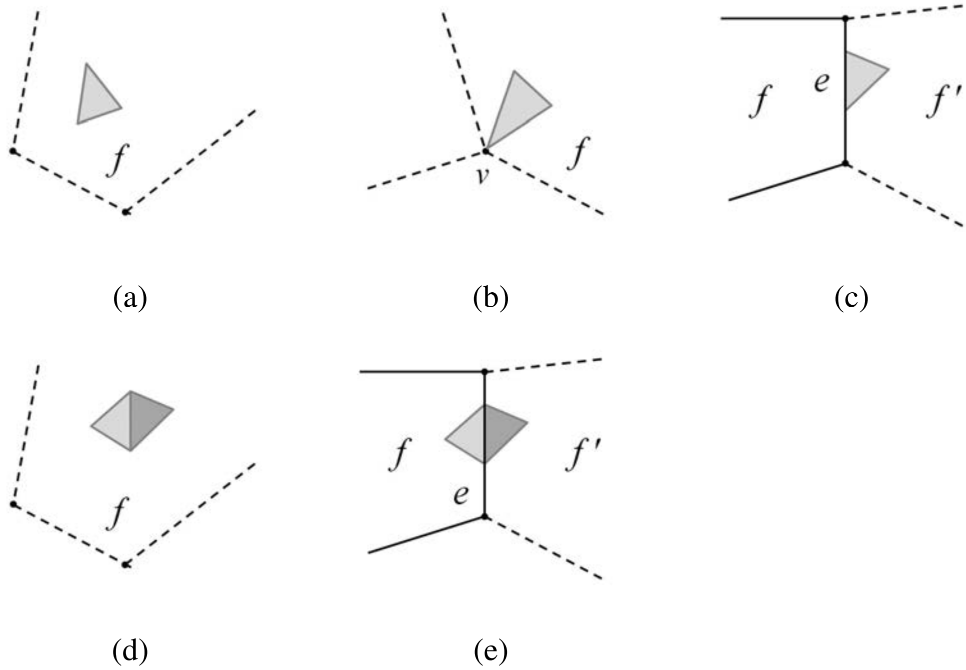


Figure 6.11: Illustration of the selection of triangles

Define

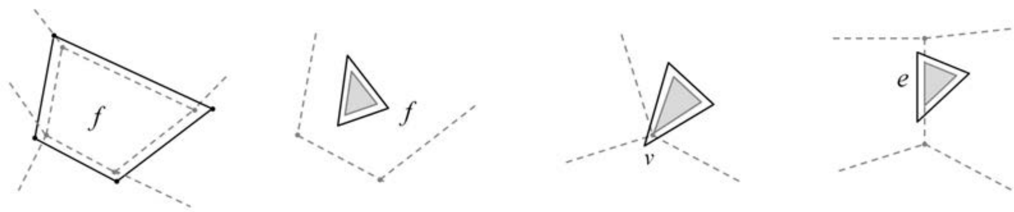
$$a_i = \bigcup \mathcal{I}_{\textsc{Face}}(v_i) \cup \bigcup X_i. \tag{6.31}$$

The following lemma can be proved in the same way as in the proof of the Lemma 6.6.

Lemma 6.10 $\{a_1, \cdots, a_n\}$ *satisfies all the constraints in* Γ *except that some* **NTPP** *constraints may only be realised as* **TPP**.

To cope with the **NTPP** constraints, we introduce the expand function from $(\textsc{Face} \cup X) \times \{1, 2, \cdots, n\}$ to plane regions such that

- $\mathsf{expand}(x, i)$ **NTPP** $\mathsf{expand}(x, i+1)$ for any $x \in \textsc{Face} \cup X$ and $i = 1, 2, \cdots, n-1$.
- $\mathsf{expand}(x, i)$ **DC** $\mathsf{expand}(x', i')$ for any $x, x' \in \textsc{Face} \cup X$ such that $x\mathbf{DC}x'$, and $i, i' = 1, 2, \cdots, n$.
- $\mathsf{expand}(x, i)$ **PO** $\mathsf{expand}(x', i')$ for any $x, x' \in \textsc{Face} \cup X$ such that $x\mathbf{EC}x'$, and $i, i' = 1, 2, \cdots, n$.



(a) $x \in$ FACE (b) x inside face f (c) x on vertex v (d) x on edge e

Figure 6.12: Illustration of function $\mathsf{expand}(x, 1)$

That is to say, $\mathsf{expand}(x, i)$ are a series of nested regions with x as the innermost core. Meanwhile, $\mathsf{expand}(x, i)$ should be small enough in order not to touch or overlap any other regions or any other $\mathsf{expand}(x', i')$ if possible. Figure 6.12 provides illustrations for $\mathsf{expand}(x, 1)$.

We can extend the domain of the function expand to include all the a_i defined in Eq. 6.31 and all the landmarks by

$$\mathsf{expand}(y, i) = \bigcup\{\mathsf{expand}(x, i) : x \subseteq y, x \in \textsc{Face} \cup X\}, \tag{6.32}$$

where $y \in \{a_1, \cdots, a_n, l_1, \cdots, l_m\}$ and $i = 1, 2, \cdots, n$.

Define a function from $d_{\mathbf{NTPP}} : V \times (V \cup L)$ to $\mathbb{N}$, such that $d_{\mathbf{NTPP}}(v_i, w)$ is the length of the longest **NTPPi** chain from v_i to w, where w is either some v_j or some l_j.

Let

$$\begin{aligned} b_i = a_i &\cup \bigcup\{\mathsf{expand}(a_j, d_{\mathbf{NTPP}}(v_i, v_j)) : v_j \mathbf{NTPP} v_i\} \\ &\cup \bigcup\{\mathsf{expand}(l_j, d_{\mathbf{NTPP}}(v_i, l_j)) : l_j \mathbf{NTPP} v_i\}. \end{aligned} \tag{6.33}$$

Then $\{b_1, \cdots, b_n\}$ is a solution that satisfies the requirements in Lemma6.9

The above lemma indicates that we are able to decide the existence of solutions of Γ, as long as the intersection of the boundary of each variable and the vertex set VTX is fixed in advance (namely, S_i). Note that each S_i has $O(K^2)$ points which are polynomial with respect to the input size. Therefore, by guessing S_i in a nondeterministic algorithm, we come to the conclusion that the extended consistency problem for basic RCC-8 networks is in NP.

Theorem 6.7 *The extended consistency problem* $\textsc{cspsat}_{\mathcal{RCC8},s}(\mathcal{B}_{\mathcal{RCC8}})$ *is NP-complete,* *provided that the landmarks are represented by polygons.*

6.4.3.3 RCC-8 model based on strong connectedness

In the standard RCC-8 model, two regions are considered to be connected if their intersection is not empty. As we have shown in subsection 5.1, the possibility of the finite intersection of two **EC** regions results in the NP-hardness of the extended consistency problem. In this subsection, we consider another RCC-8 model in which two regions are considered to be connected, if and only if there is at least a *curve* (rather than a point) belonged to both regions. The different definition of connectedness results in different **DC**, **EC**, **NTPP**, **TPP** relations.

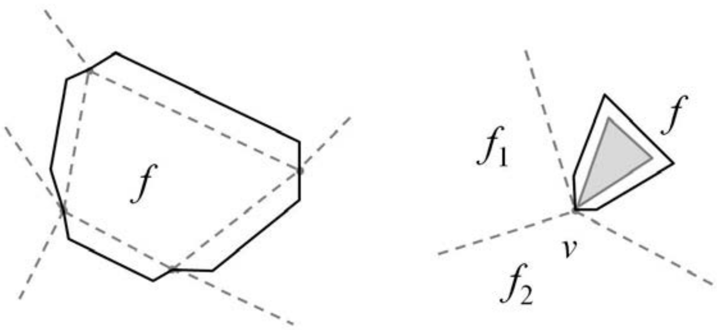
Definition 6.3 (strong-connectedness RCC-8 algebra) *A subset X of the plane is called a* curve *if there is a continuous mapping $\gamma : [0, 1] \to X$. The* **PO**, **EQ** *relations in the strong RCC-8 algebra is the same as those in the standard RCC-8 algebra, while the* **DC**, **EC**, **NTPP**, **TPP** *relations are defined as follows, and* **TPPi**, **NTPPi** *are the converses of* **TPP** *and* **NTPP**.

- a**DC**b *if and only if $a \cap b$ doesn't contain any curve.*
- a**EC**b *if and only if $a^\circ \cap b^\circ = \varnothing$ and $a \cap b$ contains at least one curve.*
- a**NTPP**b *if and only if $a \subset b$ and $\partial a \cap \partial b$ doesn't contain any curve.*
- a**TPP**b *if and only if $a \subset b$ and $\partial a \cap \partial b$ contains at least one curve.*

The following theorem shows that under this RCC-8 model, the extended consistency can be decided in polynomial time.

Theorem 6.8 *The extended consistency problem of the strong-connectedness RCC-8 algebra can be decided by checking all the conditions in Lemma 6.9, assuming that $S_i = \varnothing$ for all variables v_i.*

Intuitively, the NP-hardness of the extended consistency problem in the standard RCC model is due to the exponential possibilities of S_i, the intersection of VTX and the boundary of v_i. This matters because the points in S_i may be evidence of **EC**



(a) $x \in$ FACE (c) x on vertex v

Figure 6.13: Illustration of function $\mathsf{expand}(x, 1)$ in strong connectedness RCC model

constraints. However, in the strong connectedness model, discrete points no longer account for **EC** relations, so S_i may simply be regarded as empty.

The proof can be obtained by slightly modifying the proof of Lemma 6.9. Note that we need to adjust the construction in the sufficiency part to cope with the strong connectedness.

The only differences here from the standard RCC-8 interpretation are: (i) we assume $S_i = \varnothing$ for all variables v_i, as in the theorem; (ii) although the requirements for $\mathsf{expand}(\cdot, \cdot)$ maintain, the construction of the function needs modification due to different interpretations of RCC-8 relations. If x is a face in FACE, or a triangle in X on some vertex v, $\mathsf{expand}(x, 1)$ should be modified as shown in Figure 6.13 (a) and (c) respectively. Note that in Figure 6.13 (c), it holds that $x\mathbf{DC}f_1$ since their intersection is a point (not a curve). Therefore, $\mathsf{expand}(x, 1)$ is supposed to be disjoint with f_1 (under the strong connectedness interpretation of RCC) due to the requirement of $\mathsf{expand}(\cdot, \cdot)$. The case in Figure 6.13 (a) is similar: the boundary of the expanded face does not intersect with any face that are disjoint with the original face.

All the remaining parts of the construction, including the selection of triangles (note that $S_i = \varnothing$ here), the definition of a_i, b_i, and the verification of b_i as a solution of Γ, are completely the same as in the standard RCC model.

The computational complexity is $O((n + m)^3 + n(n + m)K^2 + m^2K^2 \log K)$, the same as for the RCC-5 case. The arguments there also apply for RCC-8 in the strong connectedness model.

6.5 Chapter Summary

This chapter extends the consistency problem in qualitative spatial and temporal reasoning. In the former consistency problem, all the entities are variables which could be interpreted by arbitrary elements in the universe. This cannot characterise all situations in applications in which completely known entities (landmarks) or partially known entities are involved. By equipping each variable with a domain, being either the universe or a finite subset of the universe, the expressive ability of the consistency problem is enhanced. In particular, we distinguish the landmark case, that is, each variable in the constraint network is either totally unknown or completely known as a landmark. There are several reasons for this: first, the landmark case is simpler than the general case and is a good starting point. Second, the general case can always be reduced to the landmark case (though possibly exponentially), i.e., if we have an algorithm that solves the landmark case, we may also solve the general case in a backtracking manner. Furthermore, landmarks are widely involved in real applications.

The computational complexity of solving the extended consistency problem is then investigated for qualitative calculi including Point Algebras, Interval Algebra, Cardinal Direction Calculus and RCC-5/RCC-8. We show that, if only basic networks are considered, the landmark case is tractable except for RCC-8. For the RCC-8 (under standard interpretation), the problem is in general NP-complete; but it becomes tractable if two regions are considered as being connected when they share at least a curve. For the general case in which variables are allowed to be finitely restricted, the consistency problems are NP-complete even for basic networks in all qualitative calculi mentioned above except Point Algebra. For PA, we devise a polynomial algorithm that solves the consistency problem over a maximal tractable subclass $\{<, >, =, \leq, \geq, ?\}$. For other qualitative calculi, we construct polynomial reductions from the well-known NP-complete 3-SAT problem to the extended consistency problem and thus prove its NP-hardness.

Chapter 7

Solving Minimal Constraint Networks

7.1 Introduction

In previous chapters, we have considered the consistency problems for several qualitative calculi. There are a number of other related reasoning problems. In this chapter, we will consider two reasoning problems involving the concept of *minimal labels*. Given a set of constraints Γ over variables set V, the minimal label between variable x and y is defined as the strongest relation between them implied by Γ. The *minimal labeling problem* (MLP) [105] [1], which computes the minimal label turns out to be equivalent to the consistency problem with respect to polynomial Turing-reductions. Another related problem is the *minimality problem*, which is to decide whether a constraint network Γ is *minimal*, where we say Γ is minimal if for all constraints $x\alpha y$ in Γ, α is the minimal label of x and y [71]. Although the minimality problem can be reduced to consistency problem (but not vice versa), its tractability remains unsolved when the consistency problem is NP-hard. Meanwhile, noticing that minimal networks are always consistent, it would be natural to ask the following question: is there a polynomial algorithm that may compute a solution of a given minimal constraint network?

In the context of classical CSP, Gottlob has recently proved the NP-hardness of computing a solution of minimal networks [43, 44], which confirms the conjecture proposed by Dechter [21]. However, related work in QSTR is quite limited [9, 41]. Inspired by Gottlob's work, this chapter will investigate the minimality problem and

[1] Also known as the *deductive closure problem* in [107].

the solution computation problem in the context of QSTR. For partially ordered Point Algebras (in which two points can be incomparable), Interval Algebra, Cardinal Relation Algebra, and RCC-8, we will show that deciding the minimality of constraint networks is in general NP-hard. Furthermore, we will prove the NP-hardness of computing solutions of minimal constraint networks in each qualitative calculi mentioned above, although the problem was assumed to be easy in [9].

The remainder of this chapter proceeds as follows. Section 2 reviews basic notions and discusses Gottlob's approach. Section 3 shows that computing a solution of a minimal constraint network in partially ordered Point Algebras and RCC-8 is NP-hard, while Section 4 proves the same result for Cardinal Relation Algebra and Interval Algebra. The last section concludes this chapter. Note that the results in this chapter have been accepted by conference CP-2012 [66].

7.2 Preliminaries

We recall the concept of minimal networks which has been introduced in Definition 2.31. Suppose $\mathcal{M}$ is a qualitative calculus, V is a set of variables, and Γ is a constraint network over V. If for any constraint $x\alpha y$ in Γ and any basic relation β contained in α, the constraint network $\Gamma \setminus \{x\alpha y\} \cup \{x\beta y\}$ (i.e., replace $x\alpha y$ in Γ with $x\beta y$) is consistent, then we say Γ is a *minimal network*.

The main aim of this chapter is to discuss the complexity of computing a solution of a given constraint network which is known to be minimal. We prove that, for all qualitative calculi mentioned above except the totally ordered Point Algebra, it is NP-hard to compute a single solution of a minimal constraint network.[1] We do not distinguish the semantic difference between 'computing a solution' and 'computing a consistent scenario'. This is because, on one hand, polynomial algorithms to construct solutions for consistent scenarios have been proposed for all qualitative calculi discussed in this paper (see e.g.[55]); and on the other hand, a consistent scenario can always be computed in polynomial time from a solution.

[1]Note that computing a solution of a constraint network is not a decision problem (i.e., problem with either a 'yes' or 'no' answer). So the 'NP-hardness' here means that, if we have a polynomial algorithm that computes a solution of a minimal network, then we can provide a polynomial algorithm that solves an NP-complete problem [43].

A strategy has been introduced in [43] to prove the NP-hardness of computing a solution of a minimal constraint network in classical CSP. This strategy can also be applied to the same problem in QSTR. The general framework is as follows, where $\mathcal{M}$ is a qualitative calculus in which the consistency of any basic network (scenario) can be decided in polynomial time.

- Construct a polynomial reduction R from an NP-hard problem $\mathcal{N}$ (variants of the SAT problem in this paper) to the consistency problem CSPSAT($\mathcal{M}$) for $\mathcal{M}$.
- Show any CSPSAT($\mathcal{M}$) instance generated by R is either inconsistent or minimal.

We claim that the existence of such a reduction R implies the NP-hardness of computing a solution of a minimal constraint network in $\mathcal{M}$. This is because, if there exists a polynomial algorithm $\mathcal{A}$ which can compute a solution (i.e. a consistent scenario) of a minimal constraint network with upper bounding time $p(x)$ (where x is the size of input), then the following polynomial algorithm $\mathcal{A}^*$ would solve the NP-hard problem $\mathcal{N}$, where i is an instance of $\mathcal{N}$.

- Compute the CSPSAT($\mathcal{M}$) instance $R(i)$ by the reduction R.
- Call Algorithm $\mathcal{A}$ with input $R(i)$.
- If $\mathcal{A}$ does not halt in $p(size(R(i)))$ time, return 'No'.
- Verify whether the output of $\mathcal{A}$ is a solution of $R(i)$. If not, return 'No'.
- Return 'Yes'.

Because the consistency of any scenario in $\mathcal{M}$ can be determined in polynomial time, we know that $\mathcal{A}^*$ is a polynomial algorithm. We next show that Algorithm $\mathcal{A}^*$ is sound. First, suppose i is a positive instance of $\mathcal{N}$. Then $R(i)$ is a minimal constraint network by the assumption of R. So Algorithm $\mathcal{A}$ should return a solution of $R(i)$ in $p(size(R(i)))$ time, and thus $\mathcal{A}^*$ returns the correct output 'Yes'. Second, suppose i is a negative instance of problem $\mathcal{N}$, in which case $R(i)$ is inconsistent, and Algorithm $\mathcal{A}$ gets an invalid input. So Algorithm $\mathcal{A}$ may either not halt in $p(size(R(i)))$ time, or halt with some output which is not a solution of $R(i)$ (as $R(i)$ is inconsistent). In both cases, Algorithm $\mathcal{A}^*$ returns 'No'. Therefore, $\mathcal{A}^*$ is sound and we conclude that computing a solution of a minimal network in qualitative calculus $\mathcal{M}$ is NP-hard.

Note that the reduction R above (if it exists) is also a polynomial reduction from the NP-hard problem $\mathcal{N}$ to the minimality problem in $\mathcal{M}$. Therefore the existence of R also implies the NP-hardness of the minimality problem in $\mathcal{M}$.

Theorem 7.1 *Let $\mathcal{M}$ be a qualitative calculus in which the consistency of basic networks can be decided in polynomial time. Suppose there exists a polynomial reduction R from an NP-hard problem to the consistency problem in $\mathcal{M}$ such that any* CSPSAT$(\mathcal{M})$ *instance generated by R is either inconsistent or minimal. Then the minimality problem in $\mathcal{M}$ is NP-complete. Furthermore, it is also NP-complete to compute a solution of a minimal constraint network in $\mathcal{M}$.*

The NP-hardness results provided in the following sections are achieved by polynomial reductions from special variants of the SAT problem introduced below. Note that Definition 7.1 is original while Definition 7.2 comes from [43].

Definition 7.1 (symmetric SAT) *We say a SAT instance ϕ is* symmetric, *if for any truth value assignment $v : Var(\phi) \to \{\textsf{true}, \textsf{false}\}$, v satisfies ϕ iff $\overline{v}$ satisfies ϕ, where $\overline{v}$ is defined by $\overline{v}(p) = \textsf{true}$ if $v(p) = \textsf{false}$, and $\overline{v}(p) = \textsf{false}$ if $v(p) = \textsf{true}$, for $p \in Var(\phi)$.*

It is clear that each unsatisfiable SAT instance is a symmetric instance.

Lemma 7.1 *There exists a polynomial-time transformation that transforms each SAT instance ϕ into a symmetric instance ϕ', such that ϕ is satisfiable iff ϕ' is satisfiable.*

Proof Suppose $\phi = \bigwedge_{j=1}^{m} c_j$ is a SAT instance with propositional variables $\mathrm{Var}(\phi) = \{p_1, p_2, \cdots, p_n\}$ and clauses $c_j = \bigvee_{k=1}^{t_j} l_{j,k}$, where $l_{j,k}$ are literals. For a literal l, its *negation*, denoted by $\bar{l}$, is defined to be $\neg p_i$ if $l = p_i$, or p_i if $l = \neg p_i$. Define $\overline{c_j} = \bigvee_{k=1}^{t_j} \overline{l_{j,k}}$. Clearly, a truth value assignment v satisfies c_j iff $\overline{v}$ satisfies $\overline{c_j}$.

Now we define a SAT instance ϕ' with $\mathrm{Var}(\phi') = \mathrm{Var}(\phi) \cup \{q\}$ and $2m$ clauses,

$$\phi' = \bigwedge_{i=1}^{m}(c_i \vee q) \wedge \bigwedge_{i=1}^{m}(\overline{c_i} \vee \neg q).$$

It is straightforward to verify that ϕ is satisfiable iff ϕ' is satisfiable.

Gottlob introduced the following concept of k-supersymmetry of SAT instances ($k \geq 1$) and proved that any SAT instance can be transformed into a k-supersymmetric instance while preserving its satisfiability.

Definition 7.2 (k-supersymmetric SAT, [43]) *Let ϕ be a SAT instance. We say ϕ is k-*supersymmetric*, if either it is unsatisfiable, or for any set of k propositional variables $V_k \subseteq Var(\phi)$, and any partial truth value assignment v to V_k, there is an extension of v which satisfies ϕ.*

Lemma 7.2 ([43]) *For each fixed integer $k \geq 1$, there exists a polynomial-time transformation that transforms a SAT instance ϕ into a k-supersymmetric instance ϕ^k, such that ϕ is satisfiable iff ϕ^k is satisfiable.*

It is clear that a symmetric SAT instance is also 1-supersymmetric and a $(k+1)$-supersymmetric instance is also k-supersymmetric. The following lemma synthesises Lemmas 7.1 and 7.2.

Lemma 7.3 *For each fixed integer $k \geq 1$, there exists a polynomial-time transformation that transforms a SAT instance ϕ into a symmetric and k-supersymmetric instance ϕ^*, such that ϕ is satisfiable iff ϕ^* is satisfiable.*

Proof We first transform ϕ into a symmetric SAT instance ϕ' by Lemma 7.1, then transform ϕ' into a k-supersymmetric instance ϕ^* using the second transformation described in [43, Lemma 7.2]. It can be straightforwardly verified that this transformation preserves symmetry. Therefore ϕ^* is symmetric and k-supersymmetric. This procedure is polynomial as both of the two transformations are polynomial.

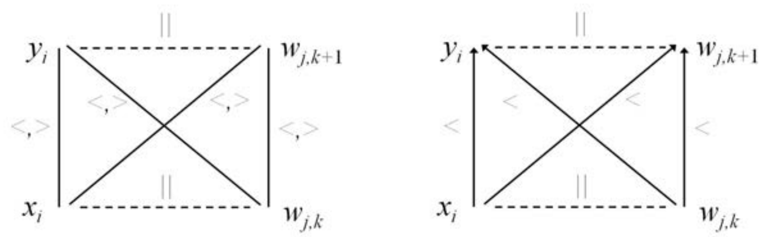
7.3 The Partially Ordered Point Algebras and RCC-8

We first prove that computing a solution of a minimal partially ordered PA constraint network is NP-hard. We achieve this by devising a reduction from the symmetric and 3-supersymmetric SAT problem to the consistency problem in the partially ordered PA. Let $\phi = \bigwedge_{j=1}^{m} c_j$ be a symmetric and 3-supersymmetric SAT instance with $\mathrm{Var}(\phi) = \{p_1, p_2, \cdots, p_n\}$ and clauses $c_j = \bigvee_{k=1}^{t_j} l_{j,k}$. We construct a constraint network (V_ϕ, Γ_ϕ) in the partially ordered PA.

The variable set is $V_\phi = V_0 \cup V_1 \cup \cdots \cup V_m$, where $V_0 = \{x_i, y_i : 1 \leq i \leq n\}$ and $V_j = \{w_{j,k} : 1 \leq k \leq t_j\}$. The constraints in Γ_ϕ are as follows, where $w_{j,t_j+1} = w_{j,1}$.

- For $x_i, y_i \in V_0$, $x_i \ \{<, >\} \ y_i$,
- For $k \neq k'$, $w_{j,k} \ \{<, >\} \ w_{j,k'}$,
- If $l_{j,k} = p_i$, then $x_i \ \{<, >\} \ w_{j,k+1}, y_i \ \{<, >\} \ w_{j,k}, x_i \parallel w_{j,k}, y_i \parallel w_{j,k+1}$,
- If $l_{j,k} = \neg p_i$, then $x_i \ \{<, >\} \ w_{j,k}, y_i \ \{<, >\} \ w_{j,k+1}, x_i \parallel w_{j,k+1}, y_i \parallel w_{j,k}$,
- For $j \neq j'$ and any $w \in V_j, w' \in V_{j'}$, $w \parallel w'$,
- For any other pair of variables $(u, v) \in V_\phi$, $u \ \{<, >, \parallel\} \ v$.

We next provide a brief explanation. We use variables $x_i, y_i \in V_0$ to simulate propositional variable p_i. The case that $x_i < y_i$ ($x_i > y_i$, resp.) corresponds to p_i being assigned true (false, resp.). Variables in V_j simulate clause c_j in ϕ, where $w_{j,k}$ and $w_{j,k+1}$ correspond to literal $l_{j,k}$ in c_j. Suppose $l_{j,k} \in \{p_i, \neg p_i\}$. Then the constraints between $x_i, y_i, w_{j,k}, w_{j,k+1}$ as specified above establish the connection between the relation of x_i and y_i and that of $w_{j,k}$ and $w_{j,k+1}$, which simulates the dependency of $l_{j,k}$ on p_i. In detail, if $l_{j,k} = p_i$ ($\neg p_i$, resp.), then the relation between $w_{j,k}$ and $w_{j,k+1}$ should be the same as (opposite to, resp.) that between x_i and y_i. For example, if $l_{j,k} = p_i$, then $x_i < y_i$ implies $x_i < w_{j,k+1}$ (otherwise, we shall have $w_{j,k+1} < x_i < y_i$ and $y_i \parallel w_{j,k+1}$, which is inconsistent). Similarly we have $y_i > w_{j,k}$ and $w_{j,k} < w_{j,k+1}$ in such case. See Figure 7.1 for illustration. By these constraints, we relate the case $w_{j,k} < w_{j,k+1}$ ($w_{j,k} > w_{j,k+1}$, resp.) to that $l_{j,k}$ being assigned true (false, resp.).



(a) constraints (b) a partial solution

Figure 7.1: Passing the relation between x_i and y_i to that between $w_{j,k}$ and $w_{j,k+1}$

Clause c_j rules out the assignments that assign all the literals $l_{j,k}$ false. Because ϕ is a symmetric SAT instance, any assignment ν that assigns all literals $l_{j,k}$ true would also falsify ϕ (otherwise, ϕ should also be satisfied by $\bar{\nu}$ which fails to satisfy c_j). Correspondingly, it is inconsistent if $w_{j,1} > w_{j,2} > \cdots > w_{j,t_j} > w_{j,1}$ or $w_{j,1} < w_{j,2} < \cdots < w_{j,t_j} < w_{j,1}$. Note that the direct constraint between any two variables in V_j is never explicitly specified. Instead, we constrain variables in V_j by enforcing a total order and transmitting the configuration of V_0 to it. In summary, by introducing of variables in V_j (and related constraints), we forbid two certain kinds of configurations of V_0, which correspond to assignments of $\mathrm{Var}(\phi)$ that assign all literals in c_j true or false.

Proposition 7.1 *Let ϕ be a symmetric SAT instance, and (V_ϕ, Γ_ϕ) be the constraint network as constructed above. Then ϕ is satisfiable iff (V_ϕ, Γ_ϕ) has a consistent scenario.*

Proof Suppose ϕ is satisfiable and $\nu : \mathrm{Var}(\phi) \to \{\mathsf{true}, \mathsf{false}\}$ is a truth value assignment that satisfies ϕ. We construct a consistent scenario for (V_ϕ, Γ_ϕ).

The constraints between variables in V_0 are as follows, see Figure 7.2 for illustration.

- If $\nu(p_i) = \mathsf{true}$, then $x_i < y_i$; otherwise $x_i > y_i$,
- Let big_i ($small_i$ resp.) be the big one (small one resp.) in x_i, y_i. Then for $i \neq i'$,

$$big_i \parallel big_{i'}, \quad small_i \parallel small_{i'}, \quad small_i < big_{i'}.$$

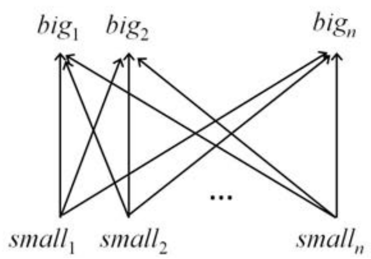


Figure 7.2: Constraints between variables in V_0 in the scenario, where $\{big_i, small_i\} = \{x_i, y_i\}$

Assume $l_{j,k} \in \{p_i, \neg p_i\}$. The constraints between $x_i, y_i, w_{j,k}$ and $w_{j,k+1}$ are decided by the constraint between x_i and y_i, as we have explained before the proof. The scenario should provide a total order on V_j with determined constraints between $w_{j,k}$ and $w_{j,k+1}$. Such a total order always exists, because the determined constraints contain no circles (which is because v satisfies clause c_j). Pick arbitrary one if such total orders are not unique.

It remains to refine constraints of the form $u \{<,>,||\} v$, where $u \in V_0$ and $v \in V_j$ for some j. Part of these constraints need to be refined to $<$ (or $>$) due to the transitivity of $<$ (or $>$). The rest can be refined to $||$.

We now get a scenario, the consistency of which can be verified by checking its path-consistency. We omit the details here.

Now suppose (V_ϕ, Γ_ϕ) has a consistent scenario. We define a truth value assignment v by $v(p_i) = \mathsf{true}$ if $x_i < y_i$ in the scenario, or $v(p_i) = \mathsf{false}$ otherwise. For each clause c_j in ϕ, it can be checked that v satisfies c_j because the constraints about variables in V_j in the scenario are consistent. Therefore, ϕ is satisfiable.

The PA network (V_ϕ, Γ_ϕ) in the reduction above has $2n + T$ variables, where $n = |\mathrm{Var}(\phi)|$, and T is $t_1 + t_2 + \cdots + t_m$, the number of all literals in the SAT instance. Therefore the reduction is polynomial. We next prove, by exploiting the 3-supersymmetry of the SAT instance ϕ, that (V_ϕ, Γ_ϕ) is minimal if ϕ is consistent.

Proposition 7.2 *Suppose ϕ is a symmetric and 3-supersymmetric SAT instance. Let (V_ϕ, Γ_ϕ) be the constraint network as constructed above. If ϕ is satisfiable, then (V_ϕ, Γ_ϕ) is a minimal constraint network.*

Proof First note that (V_ϕ, Γ_ϕ) contains three forms of constraints, viz. $u \,||\, v$, $u\{<,>\}v$ and $u\{<,>,||\}v$. We aim to show the latter two forms of non-basic constraints can be refined to any basic constraint without violating the consistency. We have proved that (V_ϕ, Γ_ϕ) has a consistent scenario, say, (V_ϕ, Γ_0). Let (V_ϕ, Γ_1) be the scenario of (V_ϕ, Γ_ϕ) which refines non-basic constraints in the opposite way, i.e.,

- Constraint $u\{<,>\}v$ is refined to $u < v$ ($u > v$ resp.) if it is refined to $u > v$ ($u < v$ resp.) in Γ_0.
- Constraint $u\{<,>,||\}v$ is refined to $u < v$ ($u > v$, $u \,||\, v$ resp.) if it is refined to $u > v$ ($u < v$, $u \,||\, v$ resp.) in Γ_0.

It can be proved that scenario (V_ϕ, Γ_1) is path-consistent (otherwise (V_ϕ, Γ_0) would not be path-consistent) and hence consistent.

Now we only need to show that constraint $u\{<, >, ||\}v$ in (V_ϕ, Γ_ϕ) can be refined to any of $u < v$, $u > v$ and $u \parallel v$. There are only two possible cases of such u and v:

- $u \in \{x_i, y_i\}$ and $v \in \{x_{i'}, y_{i'}\}$ for some $i \neq i'$.
- $u \in \{x_i, y_i\}$ and $v = w_{j,k}$ for some i, j, k such that $l_{j,k}, l_{j,k+1} \notin \{p_i, \neg p_i\}$.

For the first case, w.l.o.g., we may suppose $u = x_i$ and $v = x_{i'}$. Because ϕ is 3-supersymmetric and hence 2-supersymmetric, there exists a truth value assignment ν which satisfies ϕ and $\nu(p_i) = \mathsf{true}$, $\nu(p_{i'}) = \mathsf{true}$. So we may get a consistent scenario of (V_ϕ, Γ_ϕ) as in the proof of Proposition 7.1 according to ν. In this scenario, we have $x_i < y_i, x_{i'} < y_{i'}$ and $x_i \parallel x_{i'}$. Therefore, there exists a consistent scenario of (V_ϕ, Γ_ϕ) in which $x_i \parallel x_{i'}$. Similarly, we may obtain consistent scenarios in which $x_i < x_{i'}$ or $x_i > x_{i'}$, by other truth value assignments on p_i and $p_{i'}$.

The second case is slightly complicated and is briefly described here. W.o.l.g., suppose we want to refine constraint $x_i\{<, >, ||\}w_{j,k}$ to $x_i < w_{j,k}$, $x_i > w_{j,k}$ and $x_i \parallel w_{j,k}$ respectively. For $x_i < w_{j,k}$, we need an assignment ν which satisfies ϕ and $\nu(p_i) = \mathsf{true}$, $\nu(l_{j,k-1}) = \mathsf{true}$ and $\nu(l_{j,k}) = \mathsf{false}$. Such assignment ν exists as ϕ is 3-supersymmetric. We are able to get a scenario as in the proof of Proposition 7.1 in which we further require $w_{j,k}$ to be the maximal element in V_j (note $w_{j,k} > w_{j,k-1}, w_{j,k+1}$ as $\nu(l_{j,k-1}) = \mathsf{true}$ and $\nu(l_{j,k}) = \mathsf{false}$). In this scenario either $x_i < w_{j,k}$ or $x_i \parallel w_{j,k}$. In the latter case, we replace it with $x_i < w_{j,k}$, which does not jeopardise the consistency as x_i is in the smaller part of V_0 and $w_{j,k}$ is the maximal element in V_j. So we get a consistent scenario in which $x_i < w_{j,k}$. The other two cases are similar, where for $x_i > w_{j,k}$ we need $\nu(p_i) = \mathsf{false}$ and $w_{j,k}$ to be the minimal element in V_j, and for $x_i \parallel w_{j,k}$ we need $\nu(p_i) = \mathsf{true}$ and $w_{j,k}$ to be the minimal element.

Now we have the following conclusion for partially ordered Point Algebra.

Theorem 7.2 *Computing a solution of a minimal network in the partially ordered Point Algebra is NP-complete.*

Proof By Propositions 7.1, 7.2 and Theorem 7.1, we know that computing a solution (or a consistent scenario) of a minimal network in partially ordered PA is NP-hard. The problem is in NP as we may guess a consistent scenario by indeterminism.

The above technique can be directly applied to the RCC-8 algebra.

Theorem 7.3 *Computing a solution of a minimal network in RCC-8 is NP-complete.*

Proof Given a symmetric and 3-supersymmetric 3-SAT instance ϕ, we may construct an instance of the consistency problem in RCC-8 by substituting PA relations $<, >, ||$ in the reduction provided above with RCC-8 relations **NTPP**, **NTPPi**, **PO** respectively. Propositions 7.1 and 7.2 still hold and can be proved in the same way. Therefore, computing a solution of a minimal RCC-8 network is NP-complete.

7.4 Cardinal Relation Algebra and Interval Algebra

This section shows that computing a solution for a minimal CRA or IA constraint network is also NP-hard. The proof is similar to but simpler than the previous one, as 3-supersymmetry is not necessary here. We begin with the Cardinal Relation Algebra.

Before going into the detail, we introduce some abbreviations that may clarify the specifications of constraints. For a CRA relation α, we use

$$a|b \;\; \alpha \;\; c|d \;\text{ to denote }\; a\,\alpha\,c, a\,\alpha\,d \;\text{ and }\; b\,\alpha\,c, b\,\alpha\,d,$$
$$\frac{a}{b} \;\; \alpha \;\; \frac{c}{d} \;\text{ to denote }\; a\,\alpha\,c \;\text{ and }\; b\,\alpha\,d.$$

Suppose $\phi = \bigwedge_{j=1}^{m} c_j$ is a symmetric SAT instance with $\mathrm{Var}(\phi) = \{p_1, p_2, \cdots p_n\}$ and clauses $c_j = \bigvee_{k=1}^{t_j} l_{j,k}$. We now construct the CRA constraint network (V_ϕ, Γ_ϕ). The spatial variables for propositional variable p_i are still x_i and y_i, while $2t_j$ spatial variables $c_{j,k}, d_{j,k} (k = 1, 2, \cdots, t_j)$ are introduced for clause c_j (which has t_j literals). So the variable set is $V_\phi = V_0 \cup V_1 \cup \cdots \cup V_m$, where $V_0 = \{x_i, y_i : 1 \leq i \leq n\}$ and $V_j = \{c_{j,k}, d_{j,k} : 1 \leq k \leq t_j\}$.

We describe the relative locations of variables which are implied by the CRA constraints in Γ_ϕ. The variables in V_0 will be located in the leftmost column (Column 0) in Figure 7.3. Among them, x_i and y_i will be located in the small dashed box which is in the i-th row. Meanwhile, x_i is either to the northwest of or to the southeast of y_i. Formally, we impose the following CRA constraints, where $1 \leq i < i' \leq n$,

$$x_i \; \{\mathrm{NW}, \mathrm{SE}\} \; y_i, \qquad x_i|y_i \; \mathrm{NW} \; x_{i'}|y_{i'}.$$

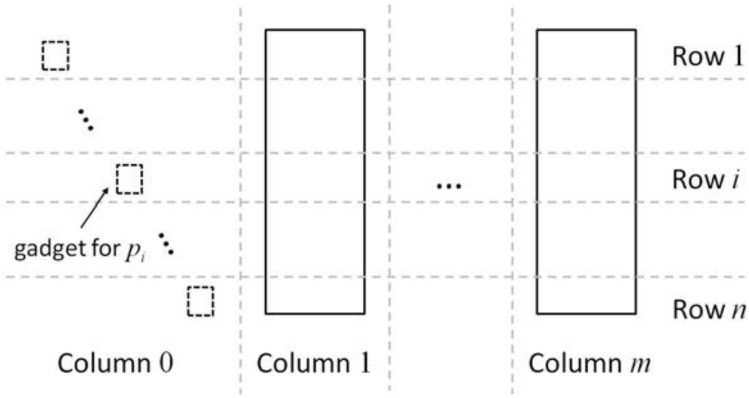


Figure 7.3: Overview of the configuration of (V_ϕ, Γ_ϕ)

The case that x_i is to the northwest (southeast resp.) of y_i corresponds to the case that p_i is assigned true (false resp.).

The $2t_j$ variables $c_{j,k}, d_{j,k} (1 \leq k \leq t_j)$ in V_j are all located in Column j in Figure 7.3. For the vertical positions, if $l_{j,k} \in \{p_i, \neg p_i\}$, then $c_{j,k}$ and $d_{j,k}$ are located in the i-th row. Precisely, we impose

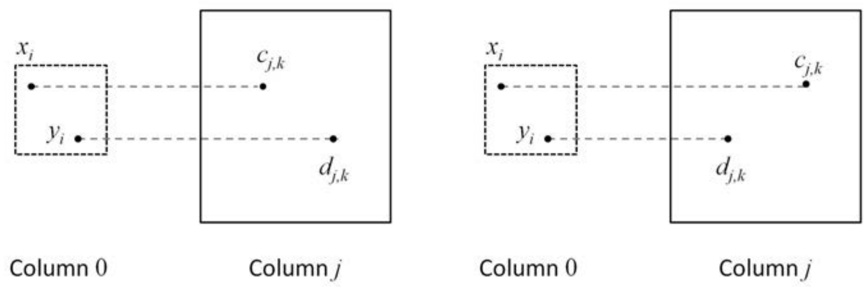
$$\frac{x_i}{y_i} \text{ W } \frac{c_{j,k}}{d_{j,k}}, \quad \frac{x_i}{y_i} \text{ \{NW, SW\} } \frac{d_{j,k}}{c_{j,k}},$$

and

$$c_{j,k} \text{ \{NW, SE\} } d_{j,k} \quad \text{if } l_{j,k} = p_i, \quad \text{or} \quad c_{j,k} \text{ \{NE, SW\} } d_{j,k} \quad \text{if } l_{j,k} = \neg p_i.$$

That is to say, $c_{j,k}$ ($d_{j,k}$ resp.) is to the east of x_i (y_i resp.). By these constraints, the CRA relation between $c_{j,k}$ and $d_{j,k}$ is decided by the relation between x_i and y_i and whether literal $l_{j,k}$ is positive or negative. In fact, it is straightforward to check that the horizontal relation (i.e., west or east) between $c_{j,k}$ and $d_{j,k}$ is in accordance with the truth value assigned to literal $l_{j,k}$ (true or false), see Figure 7.4. The relative positions (or constraints) between $c_{j,k}$, $d_{j,k}$ and $x_{i'}$, $y_{i'}$ where $i \neq i'$ can be completely decided by the constraints above.

We now discuss the constraints between variables in V_j. Suppose $u \in \{c_{j,k}, d_{j,k}\}$ and $v \in \{c_{j,k'}, d_{j,k'}\}$ are two variables in V_j, where $k \neq k'$. Suppose $l_{j,k} \in \{p_i, \neg p_i\}$ and $l_{j,k'} \in \{p_{i'}, \neg p_{i'}\}$ for some $i \neq i'$ (if $i = i'$ then either one literal can be removed, or the



(a) $l_{j,k} = p_i$ (b) $l_{j,k} = \neg p_i$

Figure 7.4: Passing the relation between x_i and y_i to that between $c_{j,k}$ and $d_{j,k}$, assuming x_i NW y_i

clause is unsatisfiable and we may simply construct an inconsistent CRA instance). We consider the constraint between u and v in the vertical direction and in the horizontal direction separately. The vertical relation between u and v is determined by i and i', as u is in the i-th row and v is in the i'-th row.

The horizontal relation between u and v is the key point which connects the SAT instance ϕ and the CRA constraint network (V_ϕ, Γ_ϕ). Note that the j-th clause c_j in ϕ rules out the assignments that assign all literals $l_{j,k}$ false. Therefore, we expect (V_ϕ, Γ_ϕ) may forbid the case that $d_{j,k}$ is to the left (i.e., northwest or southwest) of $c_{j,k}$ for all $k \in \{1, 2, \cdots, t_j\}$. To this end, $d_{j,k}$ and $c_{j,k+1}$ are required to lie on the same vertical line, while any other pair of variables in V_j are required not to, where t_j+1 is considered as 1. It is clear that $d_{j,k}$ (or equivalently $c_{j,k+1}$) being to the west of $c_{j,k}$ for all $k \in \{1, 2, \cdots, t_j\}$ is not realisable.

Note that the above constraints do not permit the case when $d_{j,k}$ is to the right (i.e., northeast of southeast) of $c_{j,k}$ for all $k \in \{1, 2, \cdots, t_j\}$. This does not cause problems since ϕ also rules out the assignments that assign all literals $l_{j,k}$ in c_j true by its symmetry.

Now we consider the constraint between $u \in V_j$ and $v \in V_{j'}$ for $j \neq j'$. Suppose $u \in \{c_{j,k}, d_{j,k}\}$, $v \in \{c_{j',k'}, d_{j',k'}\}$, where $l_{j,k} \in \{p_i, \neg p_i\}$, $l_{j',k'} \in \{p_{i'}, \neg p_{i'}\}$. The horizontal relation between u and v is determined by j and j' (u is in Column j and v is in Column j'). For the vertical constraint, note that u and v are located in the i-th and i'-th rows respectively. Therefore the case that $i \neq i'$ is clear. If $i = i'$, $c_{j,k}$ and $c_{j',k'}$ are both to the east of x_i, while $d_{j,k}$ and $d_{j',k'}$ are both to the east of y_i. So we specify the following constraints if $j < j'$. The case when $j > j'$ is similar.

$$\frac{c_{j,k}}{d_{j,k}} \; \mathrm{E} \; \frac{c_{j',k'}}{d_{j',k'}}, \qquad \frac{c_{j,k}}{d_{j,k}} \; \{\mathrm{NW},\mathrm{SW}\} \; \frac{d_{j',k'}}{c_{j',k'}}.$$

Proposition 7.3 *Given a symmetric SAT instance ϕ, suppose (V_ϕ, Γ_ϕ) is the CRA instance as constructed above. Then ϕ is satisfiable iff (V_ϕ, Γ_ϕ) has a consistent scenario.*

Proof This part can be straightforwardly proved by the connection between ϕ and (V_ϕ, Γ_ϕ) described above.

Therefore, we have a reduction from symmetric SAT to CSPSAT(*CRA*). Note that for a symmetric SAT instance ϕ with n propositional variables and T literals, the CRA instance (V_ϕ, Γ_ϕ) consists $2n + 2T$ spatial variables. So the reduction is polynomial.

Proposition 7.4 *Suppose ϕ is a satisfiable symmetric SAT instance. Then the CRA constraint network (V_ϕ, Γ_ϕ) constructed above is minimal.*

Proof We need to prove that, after refining a non-basic constraint to any basic relation it contains, the network is still consistent. Note that the CRA relation in any non-basic constraints of (V_ϕ, Γ_ϕ) is exactly the union of two basic relations. Suppose Γ_0 is a consistent scenario of (V_ϕ, Γ_ϕ). Let Γ_1 be the scenario obtained by refining non-basic constraints in (V_ϕ, Γ_ϕ) to the basic constraint different from the one in Γ_0. Scenario Γ_1 is also consistent by path-consistency (otherwise, Γ_0 cannot be path-consistent). Therefore, each non-basic constraint in (V_ϕ, Γ_ϕ) can be refined to any basic constraint (either in scenario Γ_0 or in scenario Γ_1). So (V_ϕ, Γ_ϕ) is a minimal CRA network.

Theorem 7.4 *Computing a solution of a minimal CRA network is NP-complete.*

Proof The theorem can be proved in the same way as Theorem 7.2.

For Interval Algebra, note that an interval $[a, b]$ naturally corresponds to the point (a, b) on the half plane. Based on this observation, we may translate the above reduction into a reduction for Interval Algebra directly.

Theorem 7.5 *Computing a solution of a minimal IA network is NP-complete.*

Proof We may translate the reduction for CRA into a reduction for IA by replacing CRA relations with IA relations as in the following table. Therefore, computing a solution of a minimal IA network is NP-complete.

CRA relation	NW	N	NE	W	EQ	E	SW	S	SE
IA relation	di	si	oi	fi	eq	f	o	s	d

7.5 Chapter Summary

In this chapter, we have discussed minimal constraint networks in qualitative spatial and temporal reasoning. We have proved for four major qualitative calculi (viz., partially ordered Point Algebras, Cardinal Relation Algebra, Interval Algebra, and RCC-8) that deciding the minimality of networks and computing solutions of minimal constraint networks are both NP-complete problems. We have provided a polynomial reduction from a specialised SAT problem to the consistency problem of each qualitative calculus, which only maps positive SAT instances to minimal constraint networks. The reduction exploits the symmetry of qualitative calculi, because it uses 'symmetric' intractable subclasses of relations, for example, $\{||, \{<,>\}, \{<,>,||\}\}$ in partially ordered Point Algebra, and $\{\mathbf{PO}, \{\mathbf{NTPP}, \mathbf{NTPPi}\}, \{\mathbf{NTPP}, \mathbf{NTPPi}, \mathbf{PO}\}\}$ in RCC-8.

The work of Gottlob [43] reveals the intractability of solving minimal constraint networks in classical CSP (with finite domains). This chapter has discussed the same problem, but in the context of QSTR. In this situation, the domains are infinite, and the constraints are all taken from a fixed and finite set of relations in a qualitative calculus. The minimality problem in such a qualitative calculus can also be regarded as a 'special' problem in classical CSP, but the NP-hardness of the general problem proved by Gottlob (where 2-supersymmetry is enough for the binary case) by no means implies the NP-hardness of the special problems considered in this paper, where symmetry (and 3-supersymmetry for CRA and IA) are required.

Chapter 8

Conclusion

8.1 Thesis Contributions

The qualitative approach is proved to be effective and efficient in representing and reasoning with spatial and/or temporal knowledge. This thesis has made a number of contributions concerning reasoning problems in qualitative spatial and temporal reasoning.

For the weak composition problem, we have designed a semi-automatic algorithm for computing composition tables. The algorithm repetitively generates random objects in the universe of the qualitative calculus and fills the composition table with entries that are verified by the objects, until the composition table is stable. Because the universe of a qualitative calculus is usually infinite, the question of how to generate the objects has become the most critical issue: it may affect the completeness of the CT and the efficiency of the algorithm. For IA and INDU, we have identified their 3-complete domains and thus obtained complete CTs. For $\mathcal{OPRA}_m$ with $m = 1, 2$, we have exploited the symmetry and confirmed the CT obtained in [75].

Schockaert et al. [95, 94] extend the Region Connection Calculus to the fuzzy context. However, their work has used a special fuzzy connectedness relation which can only be applied to the RCC model established on Euclidean topologies. We have proposed a general method and thus made it possible to fuzzify any crisp RCC model and obtain the *standard* fuzzy RCC model. Schockaert et al.[96] show that composition-based reasoning is still sufficient to solve fuzzy QCSP, where 31 composition rules are used. We have shown that their result is also applicable for our standard fuzzy RCC models, although only six out of the 31 composition rules are necessary. We have further proved that any satisfiable set of fuzzy RCC-8 constraints has a solution in any standard fuzzy RCC model by designing a cubic realisation algorithm.

We have discussed the consistency problem of Cardinal Direction Calculus. For complete basic constraint networks, we have proved that the problem is tractable by providing a cubic algorithm. We have further shown that, if the universal relation is allowed (or equivalently, if we allow some constraints to be unspecified), then the problem becomes NP-complete. This draws a sharp boundary between tractable and intractable subclasses in CDC.

We have observed that the original framework of the consistency problem allows variables to take *any* values in the universe, which may be improper in applications since partial knowledge may be available. Motivated by this, we have generalised the consistency problem, enabling a variable to take a fixed value (a landmark), or to take a value from a finite set. The computational complexity of the generalised consistency problems has been studied for qualitative calculi PA, IA, RCC-5 and RCC-8.

Gottlob [43, 44] has recently proven that solving minimal networks in classical CSP is NP-hard. We have investigated the same problem, but in the context of QSTR. Unlike the assumptions in the literature, we have found that solving minimal (qualitative) networks is NP-complete for partially ordered PA, CRA, IA, RCC-5, and RCC-8. As a by-product, the NP-completeness of determining the minimality of networks in these qualitative calculi has also been proved.

8.2 Future Directions

The results of this thesis are mainly of theoretical interests, therefore, making use of these results in applications is a natural but meaningful research direction. Two other directions in QSTR relating to this thesis are given in the following subsections.

8.2.1 Reasoning with Point-based directional qualitative calculi

A large number of *point-based directional* qualitative calculi have been proposed for various applications. This kind of qualitative calculi are of particular importance mainly due to two reasons: (i) spatial objects (e.g., a car or a building) can be abstracted to points in many real-world applications (e.g., high-level agent control [29] and natural language processing for robot instructions [74]), and reasoning with points are in most cases (if not all) much simpler than reasoning with regions; (ii) directional information is one of the most common and expressive kinds spatial information, and is indispensable for human-machine interfaces. Therefore, the study of this family of qualitative calculi is absolutely crucial and has significant potential in applications.

Representative point-based directional qualitative calculi qualitative calculi include CRA, $\mathcal{OPRA}_m$, LR calculus [98], the dipole calculus [73], the double-cross calculus [38] and the Qualitative Trajectory Calculus (QTC) [19]. As discussed in previous chapters, the CRA is a binary qualitative calculus that expresses cardinal direction knowledge like '*the Central Station is to the south of Town Hall*'. That is to say, the CRA uses a uniform orientation system for all the points. Differently, the $\mathcal{OPRA}_m$ equips with each object an independent orientation and is able to express directional knowledge like '*the car is running towards the building*'. The LR calculus (as well as its refinement the double-cross calculus) is a ternary qualitative calculus which can express the relative direction of an object with respect to two other objects, such as '*Go from UTS Tower Building towards Town Hall, you will find Central Station to the right of you*'. The dipole calculus is similar to the $\mathcal{OPRA}_m$ calculus, but it takes all vectors, called *dipoles*, on the plane as its universe and considers the relative directional relations between two vectors. A vector is essentially a pair of points, so in this sense the dipole calculus is a point-based qualitative calculus.

Proposing qualitative calculi is mainly an issue on how to *represent* spatial or temporal knowledge. Meanwhile, how to *reason with* spatial or temporal information in given qualitative calculi is an even more important problem. Unfortunately, only limited results have been obtained concerning point-based directional qualitative calculi. The composition problem has been investigated in some of the qualitative calculi mentioned above [75, 20], which is a good starting point for the study of the consistency problem. One recent and important progress has been made by Wolter and Lee in [112], pointing out that the consistency problem in $\mathcal{OPRA}_m$, the LR calculus, the dipole calculus and the double-cross calculus are all NP-hard. Despite of the negative NP-hardness result, the paper also reveals the connection among these qualitative calculi. The NP-hardness result suggests that there is no polynomial algorithm (unless

P=NP) for the consistency problem in this family of qualitative calculi. As a matter of fact, up to now there is even no decision procedure, regardless its computational complexity. This puts two critical problems on the table: (i) how to design algorithms to solve the consistency problems for the qualitative calculi mentioned above; and (ii) how to improve the efficiency of the algorithms. Note that whether the consistency problems are in NP is still unsolved. This is not only an important theoretical problem, but also may provide insights on problem (i). For the efficiency issue, the common and effective approach is to use heuristics. Another approach has been proposed by Pham et al. [80] and proved to be successful for the Interval Algebra: by encoding an IA constraints network into Boolean formulas, they reduce the consistency problem of IA to the SAT problem, which allows them to use cutting-edge SAT solvers to efficiently solve the former problem. Whether this approach may apply in point-based qualitative still needs to be investigated. Furthermore, approximate algorithms are also an direction. This is because, in practise (especially real-time applications), finding acceptable solutions in seconds may be much more meaningful than obtaining a best solution in days. Note that the connections between these qualitative calculi revealed in [112] suggests that, an algorithm for one of the qualitative calculi can be hopefully adapted to handle other qualitative calculi. Therefore it is reasonable to investigate a relatively simple qualitative calculus, say, the LR calculus.

8.2.2 Spatial planning

Exploiting the methodology and results of QSTR in broader practical applications has become the trend of the QSTR community. An interesting and potential area would be spatial planning. Generally speaking, automatic planning aims to find a sequence of actions which may lead a system from an initial state to a given goal. Computer scientists have begun the study of automated planning since early 1970s [35], and developed it in both theoretical areas (such as theorem proving and control theory) and practical domains (such as robotics and scheduling). In the classical setting of planning, it is assumed that the system has a finite set of states, which allows a search schema to find the desired sequence. Since the *STanford Research Institute Problems Solver* (STRIPS), a number of remarkable algorithms (called planners) for automated planning tasks have been developed, and remarkable problem specification languages such as *Action Description Language* (ADL) and the *Problem Domain Description Language* (PDDL) have been proposed for formally describing planning tasks. During the last four decades (please refer to [91] for more details), automatic planning has become an indispensable part in a huge number of human activities, covering from industry (e.g., goods transportation, car assembling), to scientific research (e.g., scheduling the assignments of the Hubble Space Telescope), or even to macro-economic and military operations.

A dominant part of practical planning tasks involve space. One significant difference between classical planning and spatial planning is that, spatial configurations (e.g., the locations of spatial entities) can hardly be described by finite states, and the actions may also be equipped with continuous parameters (e.g., a robot control system may decide to *move forward for 2.45 meters*). Therefore, classical planning techniques or planners cannot be directly applied when solving spatial planning tasks. Despite the research on QSTR has been mainly focusing on static configurations and lacks the ability to handle dynamic systems, the qualitative approach is a promising potential direction to circumvent the difficulty of spatial planning mentioned above, as it provides an elegant way to discretise the continuous space of spatial configurations and/or the action set. Even though, there is still a long way to go. The foundation and the starting point would be a well-designed, general, expressive specification language for spatial planning. The specification language should be able to handle both quantitative and qualitative spatial knowledge (e.g., the initial state is usually quantitative while the goal state such as *the ball in the basket* may be qualitative), and be preferably compatible with classical specification languages. Based on the sematic of the specification language, how to design solvers for spatial planning tasks and how to integrate the solver into real-world applications are even more challenging topics. Note that the spatial solvers may make use of the reasoning formalisms (e.g., the consistency problems) in QSTR.

References

[1] Marco Aiello, Christof Monz, and Leon Todoran. Combining linguistic and spatial information for document analysis. In Joseph-Jean Mariani and Donna Harman, editors, *RIAO*, pages 266–275. CID, 2000. 3

[2] James F. Allen. An interval-based representation of temporal knowledge. In Patrick J. Hayes, editor, *IJCAI*, pages 221–226. William Kaufmann, 1981. 35

[3] James F. Allen. Maintaining knowledge about temporal intervals. *Commun. ACM*, **26**[11]:832–843, 1983. 4, 15, 17, 29, 35, 102

[4] P. Balbiani, J.-F. Condotta, and L. Fariñas del Cerro. A new tractable subclass of the rectangle algebra. In D. Dean, editor, *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 442–447. Morgan Kaufmann, 1999. 19

[5] Philippe Balbiani, Jean-François Condotta, and Gérard Ligozat. On the consistency problem for the INDU calculus. *J. Applied Logic*, **4**[2]:119–140, 2006. 42

[6] Philippe Balbiani and Aomar Osmani. A model for reasoning about topologic relations between cyclic intervals. In Anthony G. Cohn, Fausto Giunchiglia, and Bart Selman, editors, *KR*, pages 378–385. Morgan Kaufmann, 2000. 4

[7] I. Bloch. Spatial reasoning under imprecision using fuzzy set theory, formal logics and mathematical morphology. *International Journal of Approximate Reasoning*, **41**[2]:77–95, 2006. 50

[8] Mathias Broxvall and Peter Jonsson. Point algebras for temporal reasoning: Algorithms and complexity. *Artif. Intell.*, **149**[2]:179–220, 2003. 15, 16, 29

[9] Priti Chandra and Arun K. Pujari. Minimality and convexity properties in spatial CSPs. In *ICTAI*, pages 589–593. IEEE Computer Society, 2005. 6, 178, 179

[10] E. Clementini and P. Di Felice. Approximate topological relations. *International Journal of Approximate Reasoning*, **16**[2]:173–204, 1997. 50

[11] A. Cohn and N. Gotts. The 'egg-yolk' representation of regions with indeterminate boundaries. In P. Burrough and A. Frank, editors, *Geographical Objects with Undetermined Boundaries*, pages 171–187. Taylor and Francis, London, England, 1996. 50, 57

[12] A.G Cohn and J Renz. Qualitative spatial reasoning. In Frank van Harmelen, Vladimir Lifschitz, and Bruce Porter, editors, *Handbook of Knowledge Representation*. Elsevier, 2007. 2, 3, 27, 29

[13] Anthony G. Cohn. The challenge of qualitative spatial reasoning. *ACM Comput. Surv.*, **27**[3]:323–325, 1995. 5, 7, 36, 48

[14] Anthony G. Cohn. Formalising bio-spatial knowledge. In *FOIS*, pages 198–209, 2001. 3

[15] Stephen A. Cook. Characterizations of pushdown machines in terms of time-bounded computers. *J. ACM*, **18**[1]:4–18, 1971. 14

[16] M. Cope and S. Elwood. *Qualitative GIS: a mixed methods approach*. SAGE, 2009. 4

[17] Zhan Cui, Anthony G. Cohn, and David A. Randell. Qualitative simulation based on a logical formalism of space and time. In William R. Swartout, editor, *AAAI*, pages 679–684. AAAI Press / The MIT Press, 1992. 3

[18] Mark de Berg, Otfried Ceong, Marc van Kreveld, and Mark Overmars. *Computational Geometry : Algorithms and Applications*. Springer, 3rd edition, 2008. 2, 148, 150

[19] Nico Van de Weghe, Anthony G. Cohn, and Philippe De Maeyer. A qualitative representation of trajectory pairs. In Ramon López de Mántaras and Lorenza Saitta, editors, *ECAI*, pages 1103–1104. IOS Press, 2004. 194

[20] Nico Van de Weghe, Bart Kuijpers, Peter Bogaert, and Philippe De Maeyer. A qualitative trajectory calculus and the composition of its relations. In M. Andrea Rodríguez, Isabel F. Cruz, Max J. Egenhofer, and Sergei Levashkin, editors, *GeoS*, **3799** of *Lecture Notes in Computer Science*, pages 60–76. Springer, 2005. 4, 194

[21] Rina Dechter. *Constraint Processing*. Elsevier Morgan Kaufmann, San Francisco, CA, 2003. 13, 178

[22] Rina Dechter, Itay Meiri, and Judea Pearl. Temporal constraint networks. *Artif. Intell.*, **49**[1-3]:61–95, 1991. 4

[23] Thomas Drakengren and Peter Jonsson. A complete classification of tractability in Allen's algebra relative to subsets of basic relations. *Artif. Intell.*, **106**[2]:205–219, 1998. 30

[24] S. Du, Q. Qin, Q. Wang, and H. Ma. Reasoning about topological relations between regions with broad boundaries. *International Journal of Approximate Reasoning*, **47**[2]:219–232, 2008. 50

[25] D. Dubois and M. C. Jaulent. A general approach to parameter evaluation in fuzzy digital pictures. *Pattern Recognition Letters*, **6**[4]:251–259, 1987. 50, 54

[26] I. Düntsch, H. Wang, and S. McCloskey. A relation — algebraic approach to the region connection calculus. *Theoretical Computer Science*, **255**[1-2]:63–83, 2001. 50

[27] I. Düntsch and M. Winter. A representation theorem for Boolean contact algebras. *Theoretical Computer Science*, **347**[3]:498–512, 2005. 53

[28] Ivo Düntsch. Relation algebras and their application in temporal and spatial reasoning. *Artif. Intell. Rev.*, **23**[4]:315–357, 2005. 26

[29] F. Dylla, L. Frommberger, J. O. Wallgrün, D. Wolter, B. Nebel, and S. Wölfl. Sailaway: Formalizing navigation rules. In *Proceedings of the Artificial and Ambient Intelligence Symposium on Spatial Reasoning and Communication, AISB*, 2007. 194

[30] Frank Dylla and Reinhard Moratz. Exploiting qualitative spatial neighborhoods in the situation calculus. In Freksa et al. [39], pages 304–322. 4

[31] Shimon Edelman. *Representation and Recognition in Vision*. The MIT Press, Cambridge, MA, 1999. 1

[32] Max Egenhofer and David Mark. Naive geography. In Andrew Frank and Werner Kuhn, editors, *Spatial Information Theory A Theoretical Basis for GIS*, **988** of *Lecture Notes in Computer Science*, pages 1–15. Springer Berlin / Heidelberg, 1995. 1, 3

[33] M.J. Egenhofer and R.G. Golledge. *Spatial and Temporal Reasoning in Geographic Information Systems*. Spatial Information Systems. Oxford University Press, 1998. 3

[34] T. Escrig and F. Toledo. *Qualitative Spatial Reasoning: Theory and Practice : Application to Robot Navigation*. Qualitative Spatial Reasoning Theory and Practice. IOS Press, 1998. 3

[35] Richard Fikes, Peter E. Hart, and Nils J. Nilsson. Learning and executing generalized robot plans. *Artif. Intell.*, **3**[1-3]:251–288, 1972. 195

[36] Kenneth D. Forbus, Paul Nielsen, and Boi Faltings. Qualitative spatial reasoning: The clock project. *Artif. Intell.*, **51**[1-3]:417–471, 1991. 3

[37] Andrew U. Frank. Qualitative spatial reasoning with cardinal directions. In Hermann Kaindl, editor, *ÖGAI*, **287** of *Informatik-Fachberichte*, pages 157–167. Springer, 1991. 15, 18

[38] Christian Freksa. Using orientation information for qualitative spatial reasoning. In Andrew U. Frank, Irene Campari, and Ubaldo Formentini, editors, *Spatio-Temporal Reasoning*, **639** of *Lecture Notes in Computer Science*, pages 162–178. Springer, 1992. 194

[39] Christian Freksa, Markus Knauff, Bernd Krieg-Brückner, Bernhard Nebel, and Thomas Barkowsky, editors. *Spatial Cognition IV: Reasoning, Action, Interaction, International Conference Spatial Cognition 2004, Frauenchiemsee, Germany, October 11-13, 2004, Revised Selected Papers*, **3343** of *Lecture Notes in Computer Science*. Springer, 2004. 200, 206

[40] Lutz Frommberger, Jae Hee Lee, Jan Oliver Wallgrn, and Frank Dylla. Composition in $\mathcal{OPRA}_m$. Technical report, SFB/TR 8, University of Bremen, 2007. 36

[41] Alfonso Gerevini and Alessandro Saetti. Computing the minimal relations in point-based qualitative temporal reasoning through metagraph closure. *Artif. Intell.*, **175**[2]:556–585, 2011. 178

[42] John Gooday and Anthony G. Cohn. Using spatial logic to describe visual languages. *Artif. Intell. Rev.*, **10**[3-4]:171–186, 1996. 3

[43] Georg Gottlob. On minimal constraint networks. In Jimmy Ho-Man Lee, editor, *CP*, **6876** of *Lecture Notes in Computer Science*, pages 325–339. Springer, 2011. 6, 178, 179, 180, 181, 182, 191, 193

[44] Georg Gottlob. On minimal constraint networks. *Artif. Intell.*, **191-192**[18]:42–60, 2012. 178, 193

[45] Goyal and Roop Kishor. *Similarity assessment for cardinal directions between extended spatial objects*. PhD thesis, The University of Maine, 2000. 36

[46] R. Goyal and M.J. Egenhofer. The direction-relation matrix: A representation for directions relations between extended spatial objects. In *The Annual Assembly and the Summer Retreat of University Consortium for Geographic Information Systems Science*, 1997. 15, 22, 81

[47] R.K. Goyal. *Similarity assessment for cardinal directions between extended spatial objects*. PhD thesis, The University of Maine, 2000. 24, 81, 85

[48] H. W. Guesgen. From the egg-yolk to the scrambled-egg theory. In Susan M. Haller and Gene Simmons, editors, *FLAIRS Conference*, pages 476–480. AAAI Press, 2002. 50

[49] S.M. Hazarika. *Qualitative Spatio-Temporal Representation and Reasoning: Trends and Future Directions*. Igi Global, 2012. 3

[50] Daniel Hernández. *Qualitative Representation of Spatial Knowledge*, **804** of *Lecture Notes in Computer Science*. Springer, 1994. 2

[51] Bin Jiang and Xiaobai Yao. Location-based services and gis in perspective. *Computers, Environment and Urban Systems*, **30**[6]:712–725, 2006. 3

[52] R. Krishnapuram, J.M. Keller, and Y. Ma. Quantitative analysis of properties and spatial relations of fuzzy image regions. *IEEE Transactions on Fuzzy Systems*, **1**[3]:222–233, 1993. 50, 54

[53] A. Krokhin, P. Jeavons, and P. Jonsson. Reasoning about temporal relations: The tractable subalgebras of Allen's interval algebra. *Journal of the ACM*, **50**[5]:591–640, 2003. 106

[54] B. Kuipers. *Qualitative Reasoning: Modeling and Simulation With Incomplete Knowledge*. Artificial Intelligence. MIT Press, 1994. 3

[55] Sanjiang Li. On topological consistency and realization. *Constraints*, **11**[1]:31–51, 2006. 32, 51, 162, 179

[56] Sanjiang Li and Huaiqing Wang. RCC8 binary constraint network can be consistently extended. *Artif. Intell.*, **170**[1]:1–18, 2006. 32, 82, 162

[57] Sanjiang Li and Mingsheng Ying. Region connection calculus: Its models and composition table. *Artif. Intell.*, **145**[1-2]:121–146, 2003. 30, 50

[58] Y. Li and S. Li. A fuzzy sets theoretic approach to approximate spatial reasoning. *IEEE Transactions on Fuzzy Systems*, **12**[6]:745–754, 2004. 50, 54

[59] Gerard Ligozat. On generalized interval calculi. In Thomas L. Dean and Kathleen McKeown, editors, *AAAI*, pages 234–240. AAAI Press / The MIT Press, 1991. 4

[60] Gerard Ligozat. Qualitative triangulation for spatial reasoning. In *COSIT*, pages 54–68, 1993. 4

[61] GERARD LIGOZAT. A new proof of tractability for ORD-Horn relations. In WILLIAM J. CLANCEY AND DANIEL S. WELD, editors, *AAAI/IAAI, Vol. 1*, pages 395–401. AAAI Press / The MIT Press, 1996. 29

[62] GERARD LIGOZAT. Reasoning about cardinal directions. *J. Vis. Lang. Comput.*, **9**[1]:23–44, 1998. 15, 18

[63] GÉRARD LIGOZAT AND JOCHEN RENZ. What is a qualitative calculus? A general framework. In CHENGQI ZHANG, HANS W. GUESGEN, AND WAI-KIANG YEAP, editors, *PRICAI*, **3157** of *Lecture Notes in Computer Science*, pages 53–64. Springer, 2004. 15

[64] WEIMING LIU AND SANJIANG LI. On standard models of fuzzy region connection calculus. *Int. J. Approx. Reasoning*, **52**[9]:1337–1354, 2011. 51

[65] WEIMING LIU AND SANJIANG LI. Reasoning about cardinal directions between extended objects: The NP-hardness result. *Artif. Intell.*, **175**[18]:2155–2169, 2011. 83

[66] WEIMING LIU AND SANJIANG LI. Solving minimal constraint networks in qualitative spatial and temporal reasoning. In *CP*, 2012. To appear. 179

[67] WEIMING LIU, XIAOTONG ZHANG, SANJIANG LI, AND MINGSHENG YING. Reasoning about cardinal directions between extended objects. *Artificial Intelligence*, **174**[12-13]:951–983, 2010. 29, 36, 82, 83

[68] K. LYNCH. *The Image of the City*. The MIT Press, 1960. 131

[69] H.A. MALLOT. *Computational Vision: Information Processing in Perception and Visual Behavior*. Computational Neuroscience. Mit Press, 2000. 1

[70] J.D. MONK, S. KOPPELBERG, AND R. BONNET. *Handbook of Boolean algebras*. Handbook of Boolean Algebras: General Theory of Boolean Algebras. North-Holland, Amsterdam, 1989. 10

[71] UGO MONTANARI. Networks of constraints: Fundamental properties and applications to picture processing. *Inf. Sci.*, **7**:95–132, 1974. 28, 31, 178

[72] Reinhard Moratz. Representing relative direction as a binary relation of oriented points. In Gerhard Brewka, Silvia Coradeschi, Anna Perini, and Paolo Traverso, editors, *ECAI*, **141** of *Frontiers in Artificial Intelligence and Applications*, pages 407–411. IOS Press, 2006. 4, 36

[73] Reinhard Moratz, Jochen Renz, and Diedrich Wolter. Qualitative spatial reasoning about line segments. In Werner Horn, editor, *ECAI*, pages 234–238. IOS Press, 2000. 4, 194

[74] Reinhard Moratz and Thora Tenbrink. Spatial reference in linguistic human-robot interaction: Iterative, empirically supported development of a model of projective relations. *Spatial Cognition & Computation*, **6**[1]:63–107, 2006. 194

[75] Till Mossakowski and Reinhard Moratz. Qualitative reasoning about relative direction of oriented points. *Artif. Intell.*, **180-181**:34–45, 2012. 36, 46, 48, 192, 194

[76] J.R. Munkres. *Topology; a first course*. Prentice-Hall, 1974. 11

[77] B. Nebel and H.-J. Bürckert. Reasoning about temporal relations: A maximal tractable subclass of Allen's interval algebra. *Journal of the ACM*, **42**[1]:43–66, 1995. 29, 30, 31, 82

[78] Bernhard Nebel. Computational properties of qualitative spatial reasoning: First results. In Ipke Wachsmuth, Claus-Rainer Rollinger, and Wilfried Brauer, editors, *KI*, pages 233–244. Springer, 1995. 162

[79] Bernhard Nebel and Christian Freksa. Ai approaches to cognitive systems c the example of spatial cognition. *Informatik-Spektrum*, **34**:462–468, 2011. 10.1007/s00287-011-0555-6. 3

[80] Duc Nghia Pham, John Thornton, and Abdul Sattar. Modelling and solving temporal reasoning as propositional satisfiability. *Artif. Intell.*, **172**[15]:1752–1782, 2008. 195

[81] Arun K. Pujari, G. Vijaya Kumari, and Abdul Sattar. Indu: An interval and duration network. In Norman Y. Foo, editor, *Australian Joint Conference on*

Artificial Intelligence, **1747** of *Lecture Notes in Computer Science*, pages 291–303. Springer, 1999. 41

[82] Arun K. Pujari and Abdul Sattar. A new framework for reasoning about points, intervals and durations. In Thomas Dean, editor, *IJCAI*, pages 1259–1267. Morgan Kaufmann, 1999. 4

[83] D. A. Randell, A. G. Cohn, and Z. Cui. Recent advances in qualitative physics. chapter Naive topology: modeling the force pump, pages 177–192. MIT Press, Cambridge, MA, USA, 1993. 3

[84] David A. Randell, Zhan Cui, and Anthony G. Cohn. A spatial logic based on regions and connection. In *KR*, pages 165–176, 1992. 15, 19, 35, 50

[85] J. Renz. Qualitative spatial and temporal reasoning: Efficient algorithms for everyone. In Manuela M. Veloso, editor, *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI-07)*, pages 526–531, 2007. 82

[86] J. Renz and B. Nebel. On the complexity of qualitative spatial reasoning: A maximal tractable fragment of the Region Connection Calculus. *Artificial Intelligence*, **108**[1-2]:69–123, 1999. 29, 30, 50, 82, 162

[87] Jochen Renz. A canonical model of the region connection calculus. In *KR*, pages 330–341, 1998. 29, 32

[88] Jochen Renz. A spatial odyssey of the interval algebra: 1. directed intervals. In Bernhard Nebel, editor, *IJCAI*, pages 51–56. Morgan Kaufmann, 2001. 4

[89] Jochen Renz. *Qualitative Spatial Reasoning with Topological Information*, **2293** of *Lecture Notes in Computer Science*. Springer, 2002. 3

[90] A. J. Roy and J. G. Stell. Spatial relations between indeterminate regions. *International Journal of Approximate Reasoning*, **27**[3]:205–234, 2001. 50

[91] Stuart J. Russell and Peter Norvig. *Artificial Intelligence - A Modern Approach (3. internat. ed.)*. Pearson Education, 2010. 196

[92] CHRISTOPH SCHLIEDER. Reasoning about ordering. In *COSIT*, pages 341–349, 1995. 4

[93] MARKUS SCHNEIDER, TAO CHEN, GANESH VISWANATHAN, AND WENJIE YUAN. Cardinal directions between complex regions. *ACM Trans. Database Syst.*, **37**[2]:8:1–8:40, June 2012. 81

[94] S. SCHOCKAERT, M. DE COCK, C. CORNELIS, AND E.E. KERRE. Fuzzy region connection calculus: An interpretation based on closeness. *International Journal of Approximate Reasoning*, **48**[1]:332–347, 2008. 51, 54, 68, 192

[95] S. SCHOCKAERT, M. DE COCK, C. CORNELIS, AND E.E. KERRE. Fuzzy region connection calculus: Representing vague topological information. *International Journal of Approximate Reasoning*, **48**[1]:314–331, 2008. 50, 54, 58, 192

[96] S. SCHOCKAERT, M. DE COCK, AND E.E. KERRE. Spatial reasoning in a fuzzy region connection calculus. *Artificial Intelligence*, **173**[2]:258–298, 2009. 7, 50, 51, 54, 57, 58, 59, 60, 61, 62, 68, 75, 78, 192

[97] ALEXANDER SCIVOS AND BERNHARD NEBEL. Double-crossing: Decidability and computational complexity of a qualitative calculus for navigation. In DANIEL R. MONTELLO, editor, *COSIT*, **2205** of *Lecture Notes in Computer Science*, pages 431–446. Springer, 2001. 4

[98] ALEXANDER SCIVOS AND BERNHARD NEBEL. The finest of its class: The natural point-based ternary calculus for qualitative spatial reasoning. In Freksa et al. [39], pages 283–303. 4, 194

[99] S. SKIADOPOULOS AND M. KOUBARAKIS. On the consistency of cardinal direction constraints. *Artificial Intelligence*, **163**[1]:91–135, 2005. 125, 129, 130

[100] SPIROS SKIADOPOULOS AND MANOLIS KOUBARAKIS. Composing cardinal direction relations. *Artif. Intell.*, **152**[2]:143–171, 2004. 36, 81, 82

[101] J.G. STELL. Boolean connection algebras: A new approach to the region-connection calculus. *Artificial Intelligence*, **122**[1-2]:111–136, 2000. 19

[102] Ariane Tom and Michel Denis. Referring to landmark or street information in route directions: What difference does it make? In Walter Kuhn, Michael Worboys, and Sabine Timpf, editors, *Spatial Information Theory. Foundations of Geographic Information Science*, **2825** of *Lecture Notes in Computer Science*, pages 362–374. Springer Berlin / Heidelberg, 2003. 131

[103] R.E. Valdés-Pérez. The satisfiability of temporal constraint networks. In *Proceedings of the 6th National Conference on Artificial Intelligence (AAAI-87)*, pages 256–260, 1987. 102

[104] P. van Beek. Reasoning about qualitative temporal information. *Artificial Intelligence*, **58**[1-3]:297 – 326, 1992. 129

[105] Peter van Beek. Approximation algorithms for temporal reasoning. In N. S. Sridharan, editor, *IJCAI*, pages 1291–1296. Morgan Kaufmann, 1989. 6, 178

[106] Peter van Beek. Reasoning about qualitative temporal information. *Artif. Intell.*, **58**[1-3]:297–326, 1992. 15, 29, 139

[107] Marc B. Vilain and Henry A. Kautz. Constraint propagation algorithms for temporal reasoning. In *AAAI*, pages 377–382, 1986. 6, 15, 31, 178

[108] Jan Oliver Wallgrün, Lutz Frommberger, Diedrich Wolter, Frank Dylla, and Christian Freksa. Qualitative spatial representation and reasoning in the sparq-toolbox. In Thomas Barkowsky, Markus Knauff, Gérard Ligozat, and Daniel R. Montello, editors, *Spatial Cognition*, **4387** of *Lecture Notes in Computer Science*, pages 39–58. Springer, 2006. 31, 48

[109] Sheng-Sheng Wang, Da-You Liu, Juan Chen, and Hai-Yang Jia. Qualitative spatio-temporal database. In *Machine Learning and Cybernetics, 2006 International Conference on*, pages 2002 –2007, aug. 2006. 3

[110] M. Westphal, S. Wölfl, and Z. Gantner. GQR: A fast solver for binary qualitative constraint networks. In *AAAI Spring Symposium on Benchmarking of Qualitative Spatial and Temporal Reasoning Systems*, 2009. 31, 48

[111] S. Winter, M. Duckham, and M. Robinson. Routing by landmarks. *Geoinformatics Magazine*, **12**[7]:58–59, 2009. 131

[112] Diedrich Wolter and J. H. Lee. Qualitative reasoning with directional relations. *Artif. Intell.*, **174**[18]:1498–1507, 2010. 194, 195

[113] M. Worboys and M. Duckham. *GIS: A Computing Perspective*. CRC Press, 2004. 2

[114] L.A. Zadeh. Fuzzy sets. *Information and Control*, **8**[3]:338–353, 1965. 52

[115] F.B. Zhan. Approximate analysis of binary topological relations between geographic regions with indeterminate boundaries. *Soft Computing*, **2**[2]:28–34, 1998. 50, 54

[116] X. Zhang, W. Liu, S. Li, and M. Ying. Reasoning with cardinal directions: An efficient algorithm. In D. Fox and C. Gomes, editors, *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (AAAI-08)*, pages 387–392. AAAI, 2008. 83