Region Connection Calculus: Composition Tables and Constraint Satisfaction Problems

Manas Ghosh

Department of Computer Science

Submitted in partial fulfillment
of the requirements for the degree of

Master of Science

Faculty of Mathematics and Science, Brock University
St. Catharines, Ontario

To Professor Michael Winter

&

My Parents

Abstract

Qualitative spatial reasoning (QSR) is an important field of AI that deals with qualitative aspects of spatial entities. Regions and their relationships are described in qualitative terms instead of numerical values. This approach models human based reasoning about such entities closer than other approaches. Any relationships between regions that we encounter in our daily life situations are normally formulated in natural language. For example, one can outline one's room plan to an expert by indicating which rooms should be connected to each other. Mereotopology as an area of QSR combines mereology, topology and algebraic methods. As mereotopology plays an important role in region based theories of space, our focus is on one of the most widely referenced formalisms for QSR, the region connection calculus (RCC).

RCC is a first order theory based on a primitive connectedness relation, which is a binary symmetric relation satisfying some additional properties. By using this relation we can define a set of basic binary relations which have the property of being jointly exhaustive and pairwise disjoint (JEPD), which means that between any two spatial entities exactly one of the basic relations hold. Basic reasoning can now be done by using the composition operation on relations whose results are stored in a composition table. Relation algebras (RAs) have become a main entity for spatial reasoning in the area of QSR. These algebras are based on equational reasoning which can be used to derive further relations between regions in a certain situation. Any of those algebras describe the relation between regions up to a certain degree of detail. In this thesis we will use the method of splitting atoms in a RA in order to reproduce known algebras such as RCC15 and RCC25 systematically and to generate new algebras, and hence a more detailed description of regions, beyond RCC25.

Acknowledgements

I would like to express my profound gratitude to my supervisor, Dr. Michael Winter for his continuous supervision, advice, motivation, and encouragement from the very beginning of this research work. Throughout the duration of my master program, his constant support, valuable suggestions and efforts to clarify concepts and simplify terms have inspired and enriched my development as a young researcher. It would have never been possible for me to finish this thesis work without his guidance and help.

I would like to thank my supervisory committee members- Dr. Brian Ross and Dr. Ke Qui for their support, guidance and helpful suggestions. Their guidance has served me well and I owe them my heartfelt appreciation.

My sincere thanks goes to Dr. Wendy MacCaull who took her time to read through my thesis for her insightful comments.

I would also like to thank the Department of Computer Science of Brock University for the financial and academic support.

Last but not the least, I would like to thank my family members and friends for their constant support, assistantship and inspiration with the best wishes.

M.G

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Qualitative reasoning is an approach where reasoning is based not on numbers, but on a range of more abstract or sophisticated data. The qualitative approach is considered to be closer to how humans represent and reason about commonsense knowledge. Qualitative spatial reasoning (QSR) is an important subfield of AI which is concerned with the qualitative aspects of representing and reasoning about spatial entities. Non-numerical relationships among spatial objects can be expressed through QSR. Most of the work carried out in QSR has focused on single aspects of space. The most studied, and probably most important, aspect is based on topology, the spatial relationship between regions. Relation algebras (RAs) are interesting to researchers of spatial reasoning because a large part of contemporary spatial reasoning is based on the investigations of the behavior of "part of" relations and their extensions to "contact" relations in various domains [7, 23, 24, 56]. Using the techniques of relation algebras the consistency of topological relations can be checked. From the definition of the Boolean operations , the composition operation, and the converse operation on relations we can derive which relationships between two regions are possible in a given situation. Relation algebras were introduced into spatial reasoning in [24] with additional results published in [25, 26]. We would like to refer the reader to these papers for additional motivation.

The most popular reasoning methods used in qualitative spatial reasoning are constraint based techniques. In order to apply them, it is necessary to have a set of basic qualitative binary relations which have the property of being jointly exhaustive and pairwise disjoint (JEPD). The set of all possible relations is then the set of all possible unions of the basic relations, given that reasoning can be done by exploiting composition of relations. Pre-computed compositions of relations are stored in a composition table which can serve as a look-up table for the relations. For example, if binary relation R holds between entities A and B and the binary relation S holds between B and C, then the composition of R and S restricts the possible relationship between A and C.

A constraint will be a subset of regions for a particular selected algebra. Two operators, *composition* and *join*, will be used for forming the constraint. For example a constraint is given below. In that constraint *washroom*, *bedroom* and *drawingroom* are variables ranging over regions and TPP and ECN are atomic relations. Multiple atomic relations are joined by ',' in the constraint string which means essentially '*and*'. In a constraint string, it is also possible that two entities are related by non-atomic relationships. This is indicated by combing the appropriate atomic relations using OR.

$$\begin{aligned}&washroom\ TPP\ bedroom,\\&bedroom\ ECN\ drawingroom,\\&washroom(TPP\ OR\ ECN)drawingroom\end{aligned}$$

As an area within QSR, mereotopology combines mereology, topology and algebraic reasoning. Formalisms for reasoning about spatial entities can be developed using mereotopology [4, 8, 45, 47]. Many possible theories have been proposed for mereotopology, among them, the most prominent theory is the region connection calculus (RCC) [7], which is originated from Clarke's theory [5]. Randell in [48, 49] first proposed RCC to describe a logical framework for mereotopology. It was shown in [55] that models of the RCC are isomorphic to Boolean connection algebras (or Boolean contact algebras). As lattices and Boolean algebras in particular are well-known mathematical structures, this led towards an intensive study of the properties of the RCC including several topological representation theorems [12, 13, 21, 27].

RCC is one of the widely studied systems of QSR. In RCC, regions are used as a fundamental notion. This region-based approach to spatial reasoning closely mirrors Allen's [1] interval-based approach to temporal reasoning. The JEPD set of topological relations known as RCC8 were identified as being of particular importance in the RCC theory. RCC8 consists of the relations "x is disconnected from y", "x is externally connected to y", "x partially overlaps y", "x is equal to y", "x is tangential proper part of y", "x is non-tangential proper part of y", and the inverses of the latter two relations. A relation algebra was developed based on these 8-atomic relations. These relations are defined by $\{$ DC, EC, PO, EQ, TPP, $NTPP$, $TPP^{\smile}$, $NTPP^{\smile}$ $\}$. This kind of categorization of topological relations was independently given by Egenhofer [27] in the context of geographical information systems (GIS). The same set of relations has been independently identified in [2, 29] as significant in the context of GIS. RCC8 supports the notion of a composition table since it is a JEPD. To study contact relations, Düntsch [14, 15, 16] used methods of relation algebras and explored their expressive power with respect to topological domains.

It has been shown in [14, 15, 17] that after several refinements of the eight atomic relations it is possible to produce new algebras of up to 25 atoms. New relations were obtained by splitting certain atoms from the previous algebra into two new relations and simultaneously removing certain entries in the composition table for one of the new atoms.

In [3] a method for splitting atoms in relation algebras was introduced. This method was then used in the theory of cylindric algebras to obtain nonrepresentable cylindric algebras from representable ones. In this approach a condition of splittability on the atoms was used in order ensure associativity of the composition operation after splitting. Unfortunately, this condition is violated by all the RCC tables in consideration starting with RCC11 that is also known as complemented closed disc algebra [16, 17].

Siddavaatam and Winter [51] proved a theorem for splitting atoms in a more general setting and explore ways to accomodate functional elements like bijections during the splitting process. Our contributions in the thesis are listed below.

- A proof is given for removing the additional cycle $\langle TPPA, TPPA, TPPB\rangle$ from RCC15 and RCC25 in the Lemma 3.2.2.
- In section 3.3, we have produced know algebras RCC11, RCC15 and RCC25 from RCC15 based on [51].
- In section 3.4, we have produced RCC27 from RCC25 by splitting $PONXB2$.
- In section 3.5, RCC29 is generated from RCC25 by splitting $PONXB1$.
- In section 3.6, RCC31 is generated by combining RCC27 and RCC29.
- Splitting of ECNB is not possible shown in the section 3.7.
- We have developed a system to check whether a given constraint satisfiable or not based on composition as well as based on particular selected relation algebra.

The remainder of the thesis is structured as follows. In Chapter 2 we will first introduce the region connection calculus and its basic properties. Then we will discuss relation algebras in order to study contact relations within the RCC. We will also define constraint satisfaction problems (CSP) in the context of spatial reasoning. In Chapter 3 we describe existing composition tables and using the system [51] for splitting atoms develop composition tables for RCC25, RCC27, RCC29 and RCC31, where each composition table induces a particular relation algebra. In Chapter 4 we focus on CSP for different relation algebras, finally in Chapter 5 we present our conclusion and future work.

Chapter 2

Background

Relational methods have been the basis for many conceptual and methodological tools in computer science since the 1970's. In logic and computer science there are many applications for the calculus of relations, references to many of these can be found in Németi's survey [44]. Another excellent source of applications can be found in the publications of the International Seminar on Relational Methods in Computer Science.

For analyzing, modeling or resolving several computer science problems such as program specification, heuristic approaches for program derivation, automatic prover design, database and software decomposition, program fault tolerance, testing, data abstraction and information coding, and more importantly in the area of QSR, relation algebra has been used as a basic tool.

The relation algebras that we are interested in are based on finite sets of JEPD relations which are also basic relations. If R and S are members of a JEPD relation set then $R \cap S = \emptyset$ for each pair of relations and the union of all is the greatest relation on the set. The relations of such a set are the atoms of a subalgebra of the Boolean algebra of all relations on the structure in question. If two entities are related by one of the basic relations, this can be used to represent specific knowledge. Unclear knowledge can be specified by unions of possible basic or atomic relations. If the basic relations are closed under composition and converse, then the Boolean algebra induced by the basic relations forms a relation algebra. Converse, complement, intersection and union of relations can easily be obtained by performing the corresponding set theoretic operations.

Allen's interval algebra introduced in [1] is considered as a best known example of such a relation algebra, which defines different basic relations between convex intervals on a directed line. Though the interval algebra was introduced for temporal representation and reasoning, there is a number of spatial calculi which are derived from the interval algebra. Allen identified a set of thirteen JEPD relations those are given in Figure 2.1. Those relations exist between two interpreted time intervals and reasoning can be done based on the composition of relations. If a time interval is denoted by X then X is an ordered pair (X^-, X^+) such that $X^- < X^+$, where X^- and X^+ are taken points on the real line. Basic interval relations are defined in terms of its endpoint relations. Let us consider the set $\mathbf{B}$ of those thirteen basic interval relations, then an atomic formula of the form XBY, where X and Y are intervals and $B \in \mathbf{B}$, is said to be satisfied by an interpretation if the interpretation of the intervals satisfies the endpoint relation specified in Figure 2.1.

First-order logic is of great importance to the foundations of mathematics as it is the standard formal logic for axiomatic systems and it is different from propositional logic by its use of quantified variables. It is also known as first-order predicate calculus, the lower predicate calculus quantification theory, and predicate logic.

The equational theory of the calculus of binary relations is equivalent to the three variable fragments of the first-order logic with at most binary relations [52]. Thus it is imperative to use relation algebraic methods, initiated by Tarski [53], to explore their expressive power in the topological domains based on contact relations.

Mereotopology is part of qualitative spatial reasoning which combines mereology, topology and algebraic reasoning. Mereology is a collection of axiomatic first-order theories dealing with parts and their respective wholes. The algebraic part is an atomless Boolean algebra. Topological approaches for qualitative spatial reasoning generally describe relationships between spatial regions. Here spatial regions are subsets of some topological space. Existing approaches for formalizing topological properties of spatial regions are based on the work of Whitehead [58].

RCC is based on the primitive connectedness relation, C, which is a binary symmetric relation. Using this primitive relation it is possible to define many other relations. RCC theory of spatial regions was greatly influenced by the works of Allen and Hayes [2, 31, 32, 46]. Later, its development followed based on first-order theory. Bennett [6] investigates logical representations for describing and reasoning about spatial situations. Egenhofer and Sharma [28] used relation algebras for spatial reasoning.

Reasoning which can be done in RCC through composition tables and these composition tables have become a key technique in providing an efficient inference mechanism for a wide class of theories. Cui, Cohn, Randell [10] and Egenhofer [29] independently established the composition table for basic topological relations for RCC8.

Basic Interval Relation	Symbol	Pictorial Example	Endpoint Relations
X before Y	$\prec$	xxx	$X^- < Y^-$, $X^- < Y^+$,
Y after X	$\succ$	yyy	$X^+ < Y^-$, $X^+ < Y^+$
X meets Y	m	xxxx	$X^- < Y^-$, $X^- < Y^+$,
Y met-by X	$\mathsf{m}^\smile$	yyyy	$X^+ = Y^-$, $X^+ < Y^+$
X overlaps Y	o	xxxx	$X^- < Y^-$, $X^- < Y^+$,
Y overlapped-by X	$\mathsf{o}^\smile$	yyyy	$X^+ > Y^-$, $X^+ < Y^+$
X during Y	d	xxx	$X^- > Y^-$, $X^- < Y^+$,
Y includes X	$\mathsf{d}^\smile$	yyyyyyy	$X^+ > Y^-$, $X^+ < Y^+$
X starts Y	s	xxx	$X^- = Y^-$, $X^- < Y^+$,
Y started-by X	$\mathsf{s}^\smile$	yyyyyyy	$X^+ > Y^-$, $X^+ < Y^+$
X finishes Y	f	xxx	$X^- > Y^-$, $X^- < Y^+$,
Y finished-by X	$\mathsf{f}^\smile$	yyyyyyy	$X^+ > Y^-$, $X^+ = Y^+$
X equals Y	$\equiv$	xxxx	$X^- = Y^-$, $X^- < Y^+$,
		yyyy	$X^+ > Y^-$, $X^+ = Y^+$

Figure 2.1: Thirteen basic relations of Allen's interval algebra

2.1 Binary Relations and Their Algebras

2.1.1 Definitions

Binary relations and their algebras have become essential entities for researchers especially in the field of QSR. For QSR researchers composition based reasoning with binary relations has been of great interest. Expressive power, consistency and complexity of relational reasoning have also become topics of study today.

Definition 1. *A binary relation on a set U is a subset of $U \times U$. If R,$S \subseteq U \times U$, and $x, y, z \in U$, we generally write xRy for $\langle x, y\rangle \in R$, $-xRy$ for $\langle x, y\rangle \notin R$ and $xRySz$ for xRy and yRz. We will denote the set of all relations on U by $Rel(U)$.*

The following definitions for a relation R on U are frequently used:

1. R is $reflexive$ if xRx for all $x \in U$.

2. R is $irreflexive$ if xRx for no $x \in U$.

3. R is *symmetric* if for all $x, y \in U$, xRy implies yRx.

4. R is *antisymmetric* if for all $x, y \in U$, xRy and yRx implies $x = y$.

5. R is *asymmetric* if for all $x, y \in U$, xRy implies $-yRx$.

6. R is *transitive* if for all $x, y, z \in U$, xRy and yRz implies xRz.

7. R is *functional* if for all $x, y, z \in U$, xRy and xRz implies $y = z$.

In this thesis we will use multiple algebras of different kinds. In order to define algebras in the sense of universal algebra we will use the notation A^n for Cartesian product $\underbrace{A \times \ldots \times A}_{n \text{ times}}$. Using this notation $f : A^n \to A$ denotes an n-ary function on A. Notice that we will use the convention that a 0-ary function is considered to be an element of A.

Definition 2. *An algebra $\mathfrak{A} = \langle A, \mathfrak{F} \rangle$ consists of a set A and a set of operations $\mathfrak{F}$, i.e., each $f \in \mathfrak{F}$ is a function $f : A^{n_f} \to A$. If the set $\mathfrak{F}$ is finite we will also use the notation $\mathfrak{A} = \langle A, f_1, \ldots, f_m \rangle$. In this case we say that $\mathfrak{A}$ is of type $\langle n_1, \ldots, n_m \rangle$ if f_i is an n_i-ary function for all $1 \leq i \leq m$.*

In the following we will also use the following specific types of algebras. We call an algebra $\mathfrak{A}$

- of unitary type iff its type is $\langle 1 \rangle$,
- of Boolean type iff its type is $\langle 2, 1 \rangle$,
- of monoid type iff its type is $\langle 2, 0 \rangle$,
- of relational type iff its type is $\langle 2, 1, 2, 1, 0 \rangle$.

Since relations are sets, the collection of all relations on U forms a Boolean algebra.

Definition 3. *A structure $\mathfrak{B} = \langle B, +, ^- \rangle$ of Boolean type is called a Boolean algebra (BA) iff it satisfies the following for all $x, y, z \in B$:*

B1 $x + y = y + x$.

B2 $x + (y + z) = (x + y) + z$.

B3 $\overline{\overline{x} + y} + \overline{\overline{x} + \overline{y}} = x$

In a Boolean algebra we can define a *meet* $(\cdot)$ by $x \cdot y = \overline{\overline{x} + \overline{y}}$. The *least element* 0 and the *greatest element* 1 are defined respectively as $0 = x \cdot \overline{x}$ and $1 = x + \overline{x}$ where $x \in B$. A Boolean algebra is also equipped with a *partial order* $\leq$ definable as $x \leq y$ iff $x + y = y$ $(x \cdot y = x)$. Notice that $x + y$ and $x \cdot y$ are the least upper bound (supremum) and the greatest lower bound (infimum) of x and y. A Boolean algebra $\mathfrak{B}$ is called complete iff every subset $M \subseteq B$ has a supremum $\sum M$ and and infimum

$\prod M$ with respect to the partial order $\leq$. Notice that a finite Boolean algebra is always complete.

An element $q \in B$ of a Boolean algebra $\mathfrak{B}$ is called an atom iff $q \neq 0$ and if $p \leq q$ implies $p = 0$ or $p = q$. We denote the set of all atoms by $\mathrm{At}(\mathfrak{B})$. In the Boolean algebra of all subsets of a given set the singleton sets are the atoms of the algebra. A complete Boolean algebra is called atomic iff every element is the supremum of the atoms below it, i.e., if $\sum\{q \in \mathrm{At}(\mathfrak{B}) \mid q \leq x\} = x$.

Several operations may be defined on relations: Let $R, S, T \subseteq U \times U$ and $x, y, z \in U$ then we can define:

1. Transpose or converse: $x\breve{R}y$ iff yRx
2. Complement: $x\bar{R}y$ iff $-xRy$
3. Union: $x(R \cup T)y$ iff xRy *or* xTy
4. Intersection: $x(R \cap T)y$ iff xRy *and* xTy
5. Composition: $x(R;T)z$ iff there exist y in U : xRy *and* yTz
6. Inclusion: $R \subseteq T \Leftrightarrow R \cap T = R$, i.e., $R \subseteq T$ iff xRy implies xTy for all x and y.
7. Empty relation: R $= \emptyset$, the empty relation.
8. Universal relation: $V = U \times U$.
9. Identity relation: $x1'y$ iff $x = y$.

Definition 4. *An algebra* $\mathfrak{A}= \langle B, ;, 1\text{'}\rangle$ *of monoid type is called a monoid, i.e.*

- *; is associative.*
- $1\text{'}; R = R; 1\text{'} = R$ *for all* $R \in B$.

We now provide the definition of an abstract relation algebra. The elements of such an algebra need not to be relations as defined above. However, the set of relations on a set U satisfies the axioms of a relation algebra (see Lemma 2.1.2 below).

Definition 5. *A structure* $\mathfrak{B}=\langle B, +, ^{-}, ;, ^{\smile}, 1\text{'}\rangle$ *of relational type is called an (abstract) relation algebra (RA) if it satisfies the following:*

R1 $\langle B, +, ^{-} \rangle$ *is a Boolean algebra.*

R2 $\langle B, ;, \breve{}, 1' \rangle$ *is an algebra of type* $\langle 2, 1, 0 \rangle$ *so that,*

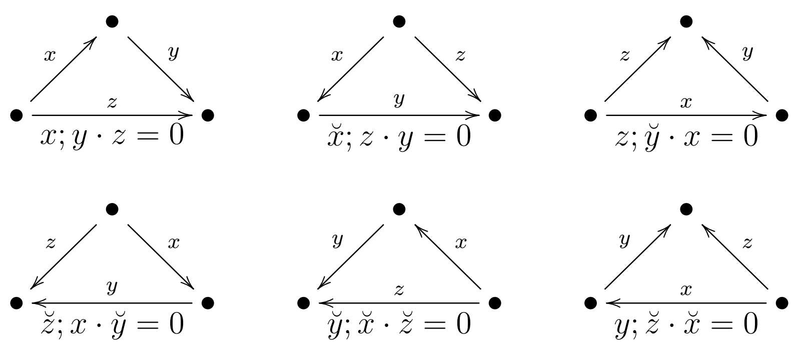
- $\langle B, ;, 1' \rangle$ *is a monoid.*
- $\breve{\breve{a}} = a$ *and* $(a;b)\breve{} = \breve{b};\breve{a}$.

R3 *For all* $x, y, z \in B$ *the following formulas are equivalent:*

$$x;y \cdot z = 0 \iff \breve{x};z \cdot y = 0 \iff z;\breve{y} \cdot x = 0.$$

We will use the term relation for elements of a relation algebra. We call an algebra of relational type that satisfies all the axioms of a relation algebra except the associativity of composition ; a nonassociative relation algebra (NA).

Oriented triangles or the *cycle law* can be used to visualize **R3**, which is de Morgan's theorem K [11]. **R3** express the fact that if one of the directed triangles below is satisfiable (in the sense of the equation below the triangle), then the others are also satisfable. In the figure below the first row of triangles was directly obtained for **R3**, and we get the second row by applying converse to the corresponding diagram of the first row.



Lemma 2.1.1. *[17] If we denote the set of all relations on the set* U *by Rel(U) then the structure denoted by* $\langle Rel(U), \cup, ^{-}, ;, \breve{}, 1' \rangle$ *is a relation algebra.*

The above algebra is called the full algebra of relations on U. This algebra is large even in the case of a small underlying set U. The cardinality of Rel(U) is given by $2^{|U|^2}$, where $|U|$ denotes the cardinality of U. We call a relation algebra $\mathfrak{B}$ representable if it is isomorphic to a subalgebra of a product of full algebras of binary relations.

Example 1. *Consider a full algebra Rel(3) where* $3 = \{0,\ 1,\ 2\}$. *Rel(3) has 512 elements and 9 atoms. Each atom is a set containing exactly one pair, e.g., one atom* *would be* $\{(0,0)\}$. *Atoms are singleton sets, i.e., the sets of the form* $\{(x,y)\}$ *with* $x, y \in U$.

In the next lemma, some properties of relation algebras are given below. A proof can be found in [39, 53].

Lemma 2.1.2. *Let* $\mathfrak{B}$ *be a relation algebra, and let* $x, y, z \in B$. *Then we have:*

1. $\breve{0} = 0, \breve{1} = 1, \breve{1}\text{'} = 1\text{'}$.
2. $x \leq x; \breve{x}; x$.
3. $x; (y; z) = (x; y); z$.
4. $x; (y + z) = x; y + x; z$.
5. $x; 1\text{'} = x$.
6. $\breve{\breve{x}} = x$.
7. $(x + y)\breve{} = \breve{x} + \breve{y}$.
8. $(x; y)\breve{} = \breve{y}; \breve{x}$
9. *if* $\breve{x}; x \leq 1\text{'}$, *then* $x; (y \cdot z) = x; y \cdot x; z$.
10. *if* $\breve{x}; x \leq 1\text{'}$, *then* $x; \overline{y} \leq \overline{x; y}$.
11. $x; y \leq z$ *iff*[1] $\breve{x}; \overline{z} \leq \overline{y}$ *iff* $\overline{z}; \breve{y} \leq \overline{x}$.

We call a relation x univalent if $\breve{x}; x \leq 1\text{'}$. A relation x is called injective if $\breve{x}$ is univalent. A bijective relation is both univalent and injective. A relation x is defined total if $1\text{'} \leq x; \breve{x}$ and x is surjective if $\breve{x}$ is total.

Integral relation algebras form basic building blocks in constructing arbitrary algebras. For details on their importance, we refer to [57].

Definition 6. *A relation algebra* $\mathfrak{B}$ *is called integral iff for all* $x, y \in B$, $x; y = 0$ *implies that* $x = 0$ *or* $y = 0$.

A relation algebra $\mathfrak{B}$ is integral if and only if the identity is an atom of $\mathfrak{B}$. Another equivalent property is the requirement that all relations of the algebra are total [38].

Properties of atoms in relation algebras are given in the next lemma. We will denote the set of atoms of a relation algebra $\mathfrak{B}$ by $At\mathfrak{B}$. Again, a proof of the lemma can be found in [38].

---

[1] We use iff as an abbreviation for if and only if.

Lemma 2.1.3. *[39, 40, 41, 44] Let $\mathfrak{B}$ be a relation algebra, and $x, y, z \in At\mathfrak{B}$. Then we have:*

1. *There is an atom $i \leq 1$' with $x; i = x$.*

2. $z \leq x; y$ *iff* $y \leq \breve{x}; z$ *iff* $x \leq z; \breve{y}$ *iff* $\breve{z} \leq \breve{y}; \breve{x}$ *iff* $\breve{y} \leq \breve{z}; x$ *iff* $\breve{x} \leq y; \breve{z}$.

3. *If y is a bijection and $x; y \neq 0$, then $x; y$ is an atom.*

If a relation algebra $\mathfrak{B}$ is finite, then the actions of Boolean operators are uniquely determined by the atoms of it. We denote the set of all bijections i.e. bijective elements of a relation algebra $\mathfrak{B}$ by $B\mathfrak{ij}\mathfrak{B}$.

If a, b are elements of a relation algebra $\mathfrak{B}$. The equation $a; x = b$ may not always have a solution but there is always a greatest solution to $a; x \leq b$. This solution is obtained using Lemma 2.1.2(11) by

$$\begin{aligned} a; x \leq b \quad &\Longleftrightarrow \quad \breve{a}; \overline{b} \leq \overline{x} \\ &\Longleftrightarrow \quad x \leq \overline{\breve{a}; \overline{b}} \end{aligned}$$

This solution is called the right residual of b over a and we denote it by $a \backslash b = \overline{\breve{a}; \overline{b}}$. In a similar way the inclusion $x; a \leq b$ has a greatest solution $b/a = \overline{\overline{b}; \breve{a}}$ called the left residual of b over a.

$a \setminus b$= -($\breve{a}$; -b) and b/a= -(b ; $\breve{a}$)

Lemma 2.1.4. *[15]*

1. *$a \setminus a$ and a/a are reflexive and transitive.*

2. *If a is reflexive, then $a \setminus a \leq a$.*

3. *If a is symmetric, then $a \backslash a \leq a$ iff $(a \setminus a)^{\smile}; (a \setminus a) \leq a$.*

Topological distinctions became particularly interesting for QSR because of its inherently qualitative nature. In qualitative spatial reasoning, topological approaches usually describe relationships between spatial regions rather than points, where spatial regions are subsets of some topological space.

Definition 7. *Let X be a non-empty set. A collection $\mathcal{T}$ of subsets of X is said to be a topology on X if*

- *X and empty set $\emptyset$, belong to $\mathcal{T}$*

- *the union of any (finite or infinite) number of sets in $\mathcal{T}$ belongs to $\mathcal{T}$*
- *the intersection of any two sets in $\mathcal{T}$ belongs to $\mathcal{T}$*

The pair $(X,\mathcal{T})$ is called a topological space.

The largest and smallest set defined above do exist since open sets are closed under arbitrary unions and closed sets are closed under arbitrary intersections. Members of $\mathcal{T}$ are called open and those sets S with $X \setminus S = \{x \in X \mid x \notin S\} \in \mathcal{T}$ are called closed.

Example 2. *Let X be a nonempty set. The collection $\{\emptyset, X\}$, consisting of the empty set and the whole set, is a topology on X, called the trivial topology or indiscrete topology. The power set $\mathcal{P}(X)$ of X, consisting of all subsets of X, is a topology on X, called the discrete topology.*

Models for mereotopological structures are collections of regular closed (or regular open) sets of topological spaces $(X, \mathcal{T})$. We will look at some definitions which are related to topological spaces. The purpose is to highlight the characterization of the models of RCC from a topological perspective.

Definition 8. *For any subset S of X, the interior of S denoted as $Int(S)$, is the largest open set contained in S, and for any subset S of X, the closure of S denoted as $Cl(S)$ is the smallest closed set contained in S.*

Let $x, y \in \mathcal{T}$, and if $Cl(x) \cap y = x \cap Cl(y) = \emptyset$, then x and y are called separated. An open set x that is nonempty is called connected if it is not the union of two separated nonempty open sets. A set $u \subseteq X$ is called regular open if $u = Int(Cl(u))$, and regular closed, if $u = Cl(Int(u))$.

Lemma 2.1.5. *[19] Let $RC(X)$ be a collection of regular closed sets of $(X, \mathcal{T})$, then $RC(X)$ is a complete Boolean algebra under set inclusion. We then have the following:*

1. $v + w = v \cup w$.
2. $v \cdot w = Cl(Int(v \cap w))$
3. $\overline{v} = Cl(X \setminus v)$

From the above lemma it is important to note that $v \cdot w \subseteq v \cap w$. In Figure 2.2 we see that the intersection of v and w contains exactly one element - the point on the border of the circles where they touch each other. But we have $v \cdot w = \emptyset$ if we use the regular topology of the Euclidean plane. This difference is the basis of external contact.

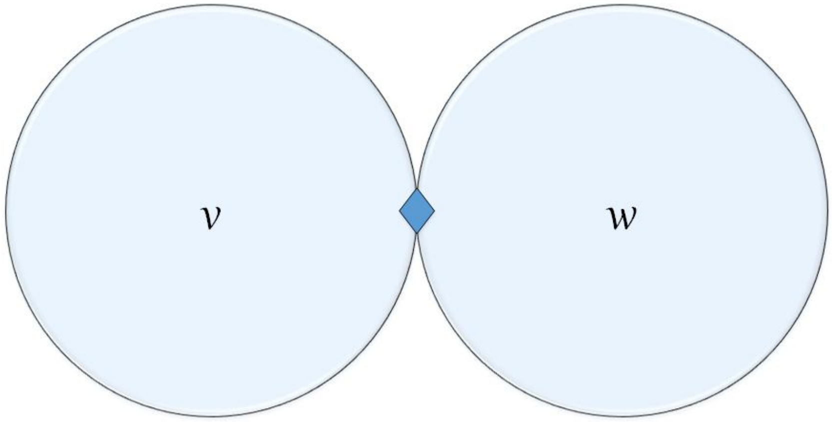


Figure 2.2: Regions v and w are in contact by means of a point

2.2 Contact Algebra

De Laguna (1922) and Whitehead (1929) [22, 58] first used contact relations in their works. They tend to use regions instead of points as the basic entity of geometry. Whitehead [58] has defined that two regular closed sets are in contact, if they have a non-empty intersection. The notion of a contact is basically reflexive and symmetric relation C among non empty regions, satisfying an additional extensionality axiom. Leśniewski's classical mereology was generalised by Clarke [9] by taking a contact relation C as the basic structural element. Clarke proposed additional axioms such as compatibility and summation in [4], in order to formalize mereological structures which are essentially complete Boolean algebras without a least element together with Whitehead's connection relation C. Nowadays the study of "part-of" and "contact" relations are used interchangely for the term "mereology" in QSR.

Definition 9. *[18] Let $\mathfrak{B}$ be a Boolean algebra, and $C \in Rel(B)$. Then we define the following properties for all $x, y, z \in B$:*

$\mathbf{C_0}$. $0\ \bar{C}\ x$ *(Null disconnectedness)*

$\mathbf{C_1}$. if $x \neq 0$, then xCx *(Reflexivity)*

$\mathbf{C_2}$. if xCy, then yCx *(Symmetry)*

$\mathbf{C_3}$. if xCy and $y \leq z$, then xCz *(Compatibility)*

$\mathbf{C_4}$. if $xC(y+z)$, then xCy or xCz *(Summation)*

$\mathbf{C_5}$. if $C(x) = C(y)$, then $x = y$ *(Extensionality)*

$\mathbf{C_6}$. if xCz or $yC\overline{z}$, then xCy *(Interpolation)*

$\mathbf{C_7}$. if $x \neq 0$ and $x \neq 1$, then $xC\overline{x}$ *(Connection)*

A relation C on a Boolean algebra $\mathfrak{B}$ is called a contact relation if it satisfies $\mathbf{C_0}$-$\mathbf{C_4}$. In this case the pair $\langle \mathfrak{B}, C \rangle$ is called a Boolean contact algebra (BCA). Notice that a Boolean contact algebra is not an algebra in the sense of Definition 3 because C is a relation and not a function. If C satisfies $\mathbf{C_5}$ in addition, it is called an extensional contact relation. In this case $C \setminus C$ is equal to the partial order of the Boolean algebra. A Boolean contact algebra is called connected if C also satisfies $\mathbf{C_7}$.

2.3 Region Connection Calculus

2.3.1 Definitions and Axioms

The RCC is a very appropriate description for a spatial formalism. Spatial entities, i.e, regions of space are extended by the fundamental approach of RCC. The primitive relation between regions - giving the language the ability to represent the structure of spatial entities - is that of *connection*.

2.3.2 RCC Axioms

A model of the RCC consists of a set R, an element $u \in R$, a singleton set {n} disjoint from R, a unary operation complement ($^-$): $R \setminus \{u\} \to R \setminus \{u\}$, a binary operation sum : $R \times R \to R$, and prod : $R \times R \to R \cup \{n\}$, and a binary relation $\mathbf{C}$ on R.

These data are required to satisfy the following axioms, $x, y, z \in R$ and $v \in R \setminus \{u\}$ which make use of the relations derived from C defined in Figure 2.3.

$\mathbf{R_1}$. xCx.

$\mathbf{R_2}$. if xCy then, yCx.

$\mathbf{R_3}$. xCu.

$\mathbf{R_{4a}}$. $xC\overline{v}$ iff $x\overline{NTPP}v$.

$\mathbf{R_{4b}}$. $xO\overline{v}$ iff $x\overline{P}v$.

$\mathbf{R_5}$. $xCsum(y, z)$ iff xCy or yCx.

$\mathbf{R_6}$. if $prod(y, z) \in R$ then $xCprod(y, z)$ is equivalent to wPy, wPz and xCw for a $w \in P$.

$\mathbf{R_7}$. $prod(x, y) \in R$ iff xOy.

$\mathbf{R_8}$. if xPy and yPx, then $x = y$.

$\mathbf{R_1}$ and $\mathbf{R_2}$ make sure that the connection relation **C** is a reflexive and symmetric binary relation. $\mathbf{R_3}$ ensures the university of the region u. $\mathbf{R_{4a}}$ and $\mathbf{R_{4b}}$ capture the ideas of the complement of a region. $\mathbf{R_5}$ represents the sum of regions. The product of two regions are represented by $\mathbf{R_6}$ and $\mathbf{R_7}$.

It has been shown that RCC models are equivalent to BCAs without least elements, i.e., that the two notations are essentially the same. Because of this we will refer to either or both sets of axioms interchangeably.

Definitions and intended meanings of the relations definable in terms of C are summarized in Figure 2.3. How one region is connected with other region is given in Figure 2.4 based on the relations defined in Figure 2.3.

2.4 Composition Table

2.4.1 Definitions

A composition table (**CT**) can be described as a matrix whose rows and columns are marked by atoms. If identity ($1'$) is an atom in our considered relation algebra, then we will discard the column and row pertaining to $1'$ from the composition table.

Mereological Relation	Connection Interpretation	Definition using Ordered Pairs
$DC \overset{def}{=} -C$	x is disconnected from y	$x\overline{C}y$
$P \overset{def}{=} C \setminus C$	x is part of y	$\forall z[zCx \rightarrow zCy]$
$PP \overset{def}{=} P \cap -1'$	x is proper part y	xPy and $x\overline{P}y$
$O \overset{def}{=} \breve{P}\ ; P$	x overlaps y	$\exists z[zPx$ and $zPy]$
$PO \overset{def}{=} O \cap (P \cup \breve{P}\ )$	x partially overlaps y	xOy and $x\overline{P}y$ and $y\overline{P}x$
$EC \overset{def}{=} C \cap -O$	x is externally connected to y	xCy and $x\overline{O}y$
$TPP \overset{def}{=} PP \cap (EC\ ;\ EC)$	x is a tangential proper part of y	$xPPy$ and $\exists z[xECz$ and $zECy]$
$NTPP \overset{def}{=} PP \cap -TPP$	x is non-tangential proper part of y	$xPPy$ and $-(\exists z[xECz$ and $zECy])$
$DR \overset{def}{=} -O$	x is discrete from y	$x\overline{O}y$

Figure 2.3: Relations between regions defined in terms of C

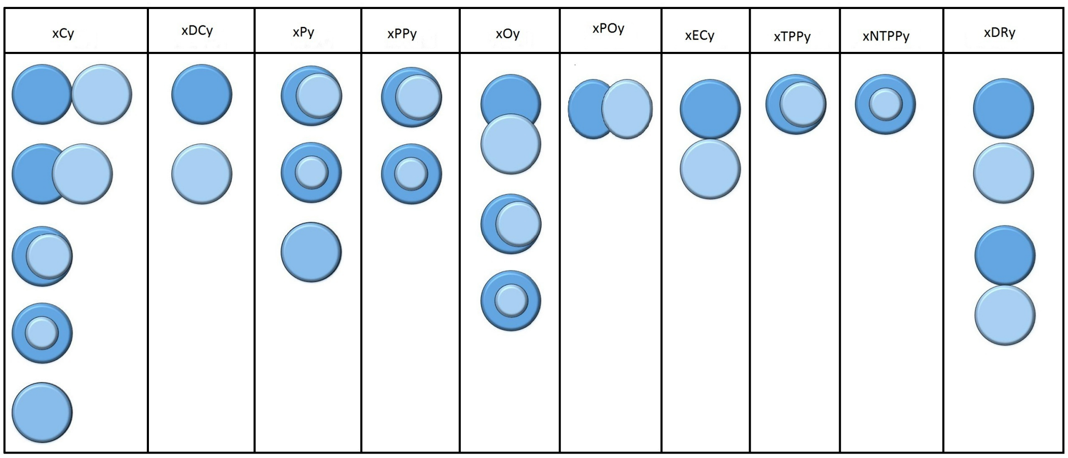


Figure 2.4: Relations definable in terms of C

Definition 10. *A composition table CT is a mapping $CT : Rels \times Rels \rightarrow 2^{Rels}$, where $Rels$ is a set of relation symbols. A model of CT is a pair (U,v), where U is a set and v is a mapping from $Rels$ to the set of binary relations on U such that, $\{v(R) : R \in Rels\}$ is partition of $U \times U$ and $v(R)\ ;v(S) \subseteq \bigcup_{T \in CT(R,S)} v(T)$.*

For three relational symbols R, S and T, if $T \in CT(R, S)$, we say T is a cell entry in the composition table specified by R and S. In this case we will also write $\langle R, S, T\rangle$ and call this triple a composition triad of the table. A model of a composition table is consistent if $T \in CT(R, S)$ implies that there are elements $a, b, c \in U$ with $av(R)b$, $bv(S)c$, and $av(T)c$, or, equivalently, if $v(R); v(S) \cap v(T) \neq \emptyset$. A consistent model is called extensional if the following condition is satisfied

$$v(R); v(S) = \bigcup_{T \in CT(R,S)} v(T) \tag{2.1}$$

In such an extensional model if T is an entry in the cell specified by R and S, then whenever $T(a, c)$ holds, there must exist some b in U such that $R(a, b)$ and $S(b, c)$. Suppose that $\mathcal{R}$ is a set of relations on U, and $R, S \in \mathcal{R}$. Now we can define weak composition in the following way.

$$R;_w S = \bigcup\{T \in \mathcal{R} : T \cap R; S \neq \phi\} \tag{2.2}$$

Weak composition is of importance for us if the JEPD set of relation $\mathcal{R}$ is the image of a model of a composition table, i.e., $\mathcal{R} = \{v(R) \mid R \in Rels\}$. Notice that in this case we have $v(R);_w v(S) = \bigcup_{T \in CT(R,S)} v(T)$, and we call $;_w$ the weak composition induced by the table. The table is extensional iff weak composition and composition coincide [16].

Consider the composition table for RCC8 relations as shown in Table 2.1. It is shown as a (7,7) matrix where an entry (i, j) contains the list of atoms of the composition of relations x_i and x_j i.e., $(x_i; x_j)$. Suppose $x_i = EC$ and $x_j = TPP$, then we find the list EC,PO,TPP,$NTPP$ in the corresponding entry, i.e., $CT(EC, TPP) = \{EC, PO, TPP, NTPP\}$.

;	**DC**	**EC**	**PO**	**TPP**	**TPP˘**	**NTPP**	**NTPP˘**
DC	DC, EC, PO TPP, TPP˘, 1' NTPP, NTPP˘	DC, EC, PO TPP NTPP	DC, EC, PO TPP NTPP	DC, EC, PO TPP NTPP	DC	DC, EC, PO TPP NTPP	DC
EC	DC, EC, PO TPP˘ NTPP˘	DC, EC, PO TPP, TPP˘ 1'	DC, EC, PO TPP NTPP	EC, PO TPP NTPP	DC, EC	PO TPP NTPP	DC
PO	DC, EC, PO TPP˘ NTPP˘	DC, EC, PO TPP˘ NTPP˘	DC, EC, PO TPP, TPP˘, 1' NTPP, NTPP˘	PO TPP NTPP	DC, EC, PO TPP˘ NTPP˘	PO TPP NTPP	DC, EC, PO TPP˘ NTPP˘
TPP	DC	DC, EC	DC, EC, PO TPP NTPP	TPP NTPP	DC, EC, PO TPP, TPP˘ 1'	NTPP	DC, EC, PO TPP˘ NTPP˘
TPP˘	DC, EC, PO TPP˘ NTPP˘	EC, PO TPP˘ NTPP˘	PO TPP˘ NTPP˘	PO TPP, TPP˘	TPP˘ NTPP˘	PO TPP NTPP	NTPP˘
NTPP	DC	DC	DC, EC, PO TPP NTPP	NTPP	DC, EC, PO TPP NTPP	NTPP	DC, EC, PO TPP, TPP˘, 1' NTPP, NTPP˘
NTPP˘	DC, EC, PO TPP˘ NTPP˘	PO TPP˘ NTPP˘	PO TPP˘ NTPP˘	PO TPP˘ NTPP˘	NTPP˘	PO TPP, TPP˘, 1' NTPP, NTPP˘	NTPP˘

Table 2.1: RCC8 Composition Table.

Composition tables are of particular interest if the corresponding weak composition always induces a relation algebra. In this case a composition table together with some additional information is equivalent to a structure known as an atom structure (see definition below). The additional information is related to the identity relation and the converse operation. In our examples this information is always implicitly given for any composition table. The identity will always be an atom, i.e., a basic symbol of the composition table. The converse of an atom in a relation algebra is again an atom. Therefore, we indicate in a composition table the converse of a symbol R by either naming its converse $R^{\smile}$ or assuming that R is its own converse. As an example the converse of the non-symmetric atom TPP is $TPP^{\smile}$ and the converse of symmetric atom PON is PON.

2.5 Splitting Atoms in Relation Algebra

It is possible to recover a complete and atomic relation algebra from a suitable structure based on its atoms with the aid of its complex algebra (see the definition below). Atom structures are very useful for storing the relation algebra as its taking less storage space for the entire algebra, where composition is carried out by a ternary relation and converse is done by a function.

We consider a relational structure $\mathfrak{G} = \langle U, C, f, I\rangle$, where C is a ternary relation on U, a unary function $f\colon U \to U$, and I is a subset of U. It is possible to construct an algebra of relational type on $Rel(U)$ of U as follows.

Definition 11. *[39] Given a relational structure $\mathfrak{G} = \langle U, C, f, I\rangle$, the complex algebra $\mathfrak{CmG} = \langle \mathcal{P}(U), \cup, \cap, ^{-}, \emptyset, U, ^{\smile}, ;, 1'\rangle$ is defined by*

$$X;Y = \{z \in U : \exists x \in X, \exists y \in Y, \langle x, y, z\rangle \in C\} \text{ and } \breve{X} = \{f(x) : x \in X\}.$$

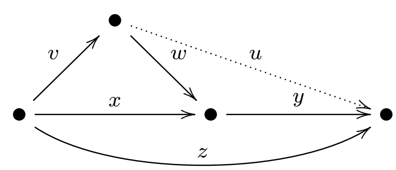
Now we will look at some definitions and theorems that are already discussed in Siddavaatam and Winter's paper [51] which will act as a basic apparatus for algorithm implementation to split atoms based on relation algebra.

Definition 12. *[39] An atom structure $\mathfrak{AtA} = \langle At\mathfrak{A}, C(\mathfrak{A}), f, I(\mathfrak{A})\rangle$ of a NA relation algebra ($\mathfrak{A}$) consists of a non-empty set $At\mathfrak{A}$ of atoms, a unary predicate $I(\mathfrak{A}) = \{x \in At\mathfrak{A} : x \leq 1'\}$, a unary function $f : At\mathfrak{A} \to At\mathfrak{A}$ defined by $f(x) = x^{\smile}$, and a ternary relation $C(\mathfrak{A}) = \{\langle x, y, z\rangle : x, y, z \in At\mathfrak{A}, x;y \geq z\}$*

Theorem 2.5.1. *[39] Let $\mathfrak{G} = \langle U, C, f, I\rangle$ be a relational structure consisting of a set U together with a ternary relation C on U, a unary function $f : U \to U$, and a subset I of U.*

1. The following three conditions are equivalent:
   (i) $\mathfrak{G}$ is an atom structure of some complete atomic NA
   (ii) $\mathfrak{Cm}\mathfrak{G}$ is a non-associative (NA).
   (iii) $\mathfrak{G}$ satisfies condition (a) and (b)
   (a) if $\langle x, y, z\rangle \in C$, then $\langle f(x), z, y\rangle \in C$ and $\langle z, f(y), x\rangle \in C$.
   (b) for all $x, y \in U, x = y$ iff there is some $w \in I$ such that $\langle x, w, y\rangle \in C$

2. $\mathfrak{Cm}\mathfrak{G}$ is a relation algebra iff $\mathfrak{Cm}\mathfrak{G}$ is a NA which also satisfies condition (c):
   (c) for all $x, v, w, x, y, z \in U$, if $\langle v, w, x\rangle \in C$ and $\langle x, y, z\rangle \in C$, then there is some $u \in U$ such that $\langle w, y, u\rangle \in C$ and $\langle v, u, z\rangle \in C$



An atom structure of a relation algebra is made up of cycles. We refer to property (a) of Theorem 2.5.1. for cycles. The notion of cycles basically reflect the *cycle law* introduced earlier. A set of cycles is actually the set of triples that are contained in those cycles. For three elements x, y, z of a relational structure $\mathfrak{G} = \langle U, C, f, I\rangle$ we write *cycle,* $\langle x, y, z\rangle$ for the following set up to six triples:

$$\langle x, y, z\rangle = \{(x, y, z), (\breve{x}, z, y), (y, \breve{z}, \breve{x}), (\breve{y}, \breve{x}, \breve{z}), (\breve{z}, x, \breve{y}), (z, \breve{y}, x)\}. \tag{2.3}$$

L. Henkin et al. [30] introduced the method of splitting in cylindric algebra theory. He showed how to obtain nonrepresentable cylindric algebras from representables ones. Later on H. Andréka et al. [3] formulated the way of splitting atoms in relation algebras.

Definition 13. *[3] For atomic NA relation algebras $\mathfrak{A}$ and $\mathfrak{B}$, $\mathfrak{A}$ is obtained from $\mathfrak{B}$ by splitting if the following conditions are satisfied:*

1. $\mathfrak{A} \supseteq \mathfrak{B}$

2. *every atom $x \in \mathfrak{A}$ is contained in an atom $c(x) \in \mathfrak{B}$, called the cover of x; and*

3. *for all x,y $\in At\mathfrak{A}$, if x,y $\leq$ 0', then*

$$x;y = \begin{cases} c(x);c(y)\cdot 0' & \text{if } x \neq \breve{y} \\ c(x);c(y) & \text{if } x = \breve{y} \end{cases}$$

If η and θ are functions mapping $At\mathfrak{B}$ to cardinals, we say that $\mathfrak{A}$ is obtained from $\mathfrak{B}$ by splitting along η and θ if $\mathfrak{A}$ is obtained from $\mathfrak{B}$ by splitting and for all $x \in At\mathfrak{B}$,

$$\begin{aligned}\eta(x) &= |\{y \in At\mathfrak{A} : y \leq x, y \neq \breve{y}\}|, \\ \theta(x) &= |\{y \in At\mathfrak{A} : y \leq x, y = \breve{y}\}|\end{aligned}$$

Andréka and Maddux [3] discussed a theorem about splitting along with two functions $\eta(x)$ and $\theta(x)$ described above. Since RCC11 is an integral relation algebra and contains the bijection relation ECD, Siddavaatam and Winter [51] showed that there is no splittable atom in RCC11 based on the Theorem 2.5.1. So splitting is not always possible because associativity might be lost. The following theorem provides some conditions under which splitting is possible.

Definition 14. *[3] Let $\mathfrak{A}$ and $\mathfrak{B}$ be atomic integral RA′s. We say that $\mathfrak{A}$ is an extension of $\mathfrak{B}$ if the following conditions are satisfied:*

1. $\mathfrak{A} \supseteq \mathfrak{B}$
2. *every atom $x \in \mathfrak{A}$ is contained in an atom $c(x) \in \mathfrak{B}$, called the cover of x.*

If the atoms in $\mathfrak{A}$ satisfy the condition imposed by two functions $\eta(x)$ and $\theta(x)$ then we can say that $\mathfrak{A}$ is an extension of $\mathfrak{B}$.

Theorem 2.5.2. *[51] Let $\mathfrak{B}$ be a complete atomic integral RA and let η, Θ be the functions mapping $At\mathfrak{B}$ to cardinals, and let $\alpha(x) = \Theta(x) + \eta(x)$. Then there is a complete atomic integral RA $\mathfrak{A}$ that is an extension of $\mathfrak{B}$ along η and Θ if the following conditions hold for all $x, y \in At\mathfrak{B}$:*

1. $\alpha(x) \geq 1$
2. $\eta(x) = \eta(\breve{x})$
3. $x \in B\mathbf{ij}\mathfrak{B}$ implies $\alpha(x) = 1$
4. $x = \breve{x}$ implies $\eta(x) =$ even, i.e. $\eta(x) = 2 * \beta$ for some ordinal β.
5. $x \neq \breve{x}$ implies $\Theta(x) = 0$.
6. $y \in B\mathbf{ij}\mathfrak{B}$ implies $\alpha(x; y) = \alpha(x)$
7. $y \in B\mathbf{ij}\mathfrak{B}$ $x = \breve{x}$ and $\eta(x) > 0$ implies $x; y = (x; y)^{\smile}$ and $\theta(x) = \theta(x; y)$

8. $\alpha(x) > 1, y; x \neq 0$ and $\notin B\mathfrak{ij}\mathfrak{B}$ implies $y \leq y; (x; x\breve{x} \cap 0')$

Let us consider the RCC11 relation algebra where the number of total atoms, $n = 11$ and the number of symmetric atoms, $s = 7$. The diversity cycles, i.e., those cycles that do not contain the identity are given below.

$C(\mathfrak{A}) = \{\langle TPP, TPP, TPP\rangle, \langle TPP, TPP, NTPP\rangle, \langle TPP, TPP^{\smile}, DC\rangle,$
$\langle TPP, TPP^{\smile}, PON\rangle, \langle TPP, TPP^{\smile}, ECN\rangle, \langle TPP, NTPP, NTPP\rangle,$
$\langle TPP, NTPP^{\smile}, NTPP^{\smile}\rangle, \langle TPP, NTPP^{\smile}, PON\rangle, \langle TPP, NTPP^{\smile}, DC\rangle,$
$\langle TPP, NTPP^{\smile}, ECN\rangle, \langle TPP, PON, TPP\rangle, \langle TPP, PON, NTPP\rangle,$
$\langle TPP, PON, PON\rangle, \langle TPP, PON, ECN\rangle, \langle TPP, PON, DC\rangle,$
$\langle TPP, PODY, TPP\rangle, \langle TPP, PODY, PON\rangle, \langle TPP, PODY, NTPP\rangle,$
$\langle TPP, PODY, PODY\rangle, \langle TPP, PODY, ECN\rangle, \langle TPP, PODY, ECD\rangle,$
$\langle TPP, PODZ, TPP\rangle, \langle TPP, PODZ, NTPP\rangle, \langle TPP, PODZ, PON\rangle,$
$\langle TPP, PODZ, PODY\rangle, \langle TPP, PODZ, PODZ\rangle, \langle TPP, ECN, ECN\rangle,$
$\langle TPP, ECN, DC\rangle, \langle TPP, ECD, ECN\rangle, \langle TPP, DC, DC\rangle,$
$\langle NTPP, NTPP, NTPP\rangle, \langle NTPP, NTPP^{\smile}, PON\rangle, \langle NTPP, NTPP^{\smile}, ECN\rangle,$
$\langle NTPP, NTPP^{\smile}, DC\rangle, \langle NTPP, PON, NTPP\rangle, \langle NTPP, PON, PON\rangle,$
$\langle NTPP, PON, ECN\rangle, \langle NTPP, PON, DC\rangle, \langle NTPP, PODY, NTPP\rangle,$
$\langle NTPP, PODY, PON\rangle, \langle NTPP, PODY, ECN\rangle, \langle NTPP, PODY, DC\rangle,$
$\langle NTPP, PODZ, NTPP\rangle, \langle NTPP, PODZ, PON\rangle, \langle NTPP, PODZ, PODY\rangle,$
$\langle NTPP, PODZ, PODZ\rangle, \langle NTPP, PODZ, ECN\rangle, \langle NTPP, PODZ, ECD\rangle,$
$\langle NTPP, PODZ, DC\rangle, \langle NTPP, ECN, DC\rangle, \langle NTPP, ECD, DC\rangle,$
$\langle NTPP, DC, DC\rangle, \langle PON, PON, PON\rangle, \langle PON, PON, PODY\rangle,$
$\langle PON, PON, PODZ\rangle, \langle PON, PON, DC\rangle, \langle PON, PON, ECN\rangle,$
$\langle PON, PON, ECD\rangle, \langle PON, PODY, PODY\rangle, \langle PON, PODY, PODZ\rangle,$
$\langle PON, PODZ, PODZ\rangle, \langle PON, ECN, ECN\rangle, \langle PON, ECN, DC\rangle,$
$\langle PON, DC, DC\rangle, \langle PODY, PODY, PODY\rangle, \langle PODY, PODY, PODZ\rangle,$
$\langle PODY, PODZ, PODZ\rangle, \langle PODZ, PODZ, PODZ\rangle, \langle ECN, ECN, ECN\rangle,$
$\langle ECN, ECN, DC\rangle, \langle ECN, DC, DC\rangle, \langle DC, DC, DC\rangle\}$

For the above relation algebra the non symmetric atoms are TPP and $NTPP$. Based on the Theorem 2.5.2 TPP is split into two new atoms called $TPPA$ and $TPPB$. As TPP is a non symmetric atom and ECD is a bijection relation then according to properties (2) and (6) of Theorem 2.5.2 we also have to split $TPP^{\smile}$, ECN and $PODY$ each into two new relations $TPPA^{\smile}$, $TPPB^{\smile}$, $ECNA$, $ECNB$, $PODYA$ and $PODYB$. After splitting we get a new algebra for which the number of total atoms $n = 15$ and the number of symmetric atoms $s = 9$. Detailed description of the splitting along with a diagram is given the next chapter. The diversity cycles of the new algebra are as follows:

$C(\mathfrak{B}) = \{\langle TPPA, TPPA, TPPA\rangle, \langle TPPA, TPPA, TPPB\rangle,$
$\langle TPPA, TPPB, TPPA\rangle, \langle TPPA, TPPB, TPPB\rangle,$
$\langle TPPB, TPPA, TPPA\rangle, \langle TPPB, TPPA, TPPB\rangle,$
$\langle TPPB, TPPB, TPPA\rangle, \langle TPPB, TPPB, TPPB\rangle, \langle TPPA, TPPA, NTPP\rangle,$
$\langle TPPA, TPPB, NTPP\rangle, \langle TPPB, TPPA, NTPP\rangle, \langle TPPB, TPPB, NTPP\rangle,$
$\langle TPPA, TPPA^{\smile}, DC\rangle, \langle TPPA, TPPB^{\smile}, DC\rangle, \langle TPPB, TPPB^{\smile}, DC\rangle,$
$\langle TPPA, TPPA^{\smile}, PON\rangle, \langle TPPA, TPPB^{\smile}, PON\rangle, \langle TPPB, TPPB^{\smile}, PON\rangle,$
$\langle TPPA, TPPA^{\smile}, ECNA\rangle, \langle TPPA, NTPP, NTPP\rangle, \langle TPPB, NTPP, NTPP\rangle,$
$\langle TPPA, NTPP^{\smile}, NTPP^{\smile}\rangle, \langle TPPB, NTPP^{\smile}, NTPP^{\smile}\rangle, \langle TPPA, NTPP^{\smile}, PON\rangle,$
$\langle TPPB, NTPP^{\smile}, PON\rangle, \langle TPPA, NTPP^{\smile}, DC\rangle, \langle TPPB, NTPP^{\smile}, DC\rangle,$
$\langle TPPA, NTPP^{\smile}, ECNA\rangle, \langle TPPA, NTPP^{\smile}, ECNB\rangle, \langle TPPB, NTPP^{\smile}, ECNA\rangle,$
$\langle TPPB, NTPP^{\smile}, ECNB\rangle, \langle TPPA, PON, TPPA\rangle, \langle TPPA, PON, TPPB\rangle,$
$\langle TPPB, PON, TPPB\rangle, \langle TPPA, PON, NTPP\rangle, \langle TPPB, PON, NTPP\rangle,$
$\langle TPPA, PON, PON\rangle, \langle TPPB, PON, PON\rangle, \langle TPPA, PON, ECNA\rangle,$
$\langle TPPA, PON, ECNB\rangle, \langle TPPB, PON, ECNA\rangle, \langle TPPB, PON, ECNB\rangle,$
$\langle TPPA, PON, DC\rangle, \langle TPPB, PON, DC\rangle, \langle TPPA, PODYA, TPPA\rangle,$
$\langle TPPA, PODYA, PON\rangle, \langle TPPA, PODYB, PON\rangle, \langle TPPB, PODYA, PON\rangle,$
$\langle TPPB, PODYB, PON\rangle, \langle TPPA, PODYA, NTPP\rangle,$
$\langle TPPA, PODYB, NTPP\rangle, \langle TPPB, PODYA, NTPP\rangle,$
$\langle TPPB, PODYB, NTPP\rangle, \langle TPPA, PODYA, PODYA\rangle,$
$\langle TPPA, PODYA, PODYB\rangle, \langle TPPA, PODYB, PODYA\rangle,$
$\langle TPPA, PODYB, PODYB\rangle, \langle TPPB, PODYA, PODYA\rangle,$
$\langle TPPB, PODYA, PODYB\rangle, \langle TPPB, PODYB, PODYA\rangle,$
$\langle TPPB, PODYB, PODYB\rangle, \langle TPPA, PODYA, ECNA\rangle,$
$\langle TPPA, PODYA, ECNB\rangle, \langle TPPA, PODYB, ECNA\rangle,$
$\langle TPPA, PODYB, ECNB\rangle, \langle TPPB, PODYA, ECNA\rangle,$
$\langle TPPB, PODYA, ECNB\rangle, \langle TPPB, PODYB, ECNA\rangle,$
$\langle TPPB, PODYB, ECNB\rangle, \langle TPPA, PODZ, TPPA\rangle, \langle TPPA, PODZ, TPPB\rangle,$

$\langle TPPB, PODZ, TPPB\rangle, \langle TPPA, PODZ, NTPP\rangle, \langle TPPB, PODZ, NTPP\rangle,$
$\langle TPPA, PODZ, PON\rangle, \langle TPPB, PODZ, PON\rangle, \langle TPPA, PODZ, PODYA\rangle,$
$\langle TPPA, PODZ, PODYB\rangle, \langle TPPB, PODZ, PODYA\rangle,$
$\langle TPPB, PODZ, PODYB\rangle, \langle TPPA, PODZ, PODZ\rangle,$
$\langle TPPB, PODZ, PODZ\rangle, \langle TPPA, ECNA, ECNA\rangle,$
$\langle TPPA, ECNA, ECNB\rangle, \langle TPPA, ECNB, ECNA\rangle, \langle TPPA, ECNB, ECNB\rangle$
$, \langle TPPB, ECNA, ECNA\rangle, \langle TPPB, ECNA, ECNB\rangle, \langle TPPB, ECNB, ECNA\rangle,$
$\langle TPPB, ECNB, ECNB\rangle, \langle TPPA, ECNA, DC\rangle, \langle TPPA, ECNB, DC\rangle,$
$\langle TPPB, ECNA, DC\rangle, \langle TPPB, ECNB, DC\rangle, \langle TPPA, DC, DC\rangle,$
$\langle TPPB, DC, DC\rangle, \langle NTPP, NTPP, NTPP\rangle, \langle NTPP, NTPP\breve{}, PON\rangle,$
$\langle NTPP, NTPP\breve{}, ECNA\rangle, \langle NTPP, NTPP\breve{}, ECNB\rangle, \langle NTPP, NTPP\breve{}, DC\rangle,$
$\langle NTPP, PON, NTPP\rangle, \langle NTPP, PON, PON\rangle, \langle NTPP, PON, ECNA\rangle,$
$\langle NTPP, PON, ECNB\rangle, \langle NTPP, PON, DC\rangle,$
$\langle NTPP, PODYA, NTPP\rangle, \langle NTPP, PODYB, NTPP\rangle,$
$\langle NTPP, PODYA, PON\rangle, \langle NTPP, PODYB, PON\rangle, \langle NTPP, PODYA, ECNA\rangle,$
$\langle NTPP, PODYA, ECNB\rangle, \langle NTPP, PODYB, ECNA\rangle,$
$\langle NTPP, PODYB, ECNB\rangle, \langle NTPP, PODYA, DC\rangle,$
$\langle NTPP, PODYB, DC\rangle, \langle NTPP, PODZ, NTPP\rangle,$
$\langle NTPP, PODZ, PON\rangle, \langle NTPP, PODZ, PODYA\rangle, \langle NTPP, PODZ, PODYB\rangle,$
$\langle NTPP, PODZ, PODZ\rangle, \langle NTPP, PODZ, ECNA\rangle, \langle NTPP, PODZ, ECNB\rangle,$
$\langle NTPP, PODZ, DC\rangle, \langle NTPP, ECNA, DC\rangle, \langle NTPP, ECNB, DC\rangle,$
$\langle NTPP, DC, DC\rangle, \langle PON, PON, PON\rangle, \langle PON, PON, PODYA\rangle,$
$\langle PON, PON, PODYB\rangle, \langle PON, PON, PODZ\rangle, \langle PON, PON, DC\rangle,$
$\langle PON, PON, ECNA\rangle, \langle PON, PON, ECNB\rangle, \langle PON, PODYA, PODYA\rangle,$
$\langle PON, PODYA, PODYB\rangle, \langle PON, PODYB, PODYB\rangle,$
$\langle PON, PODYA, PODZ\rangle, \langle PON, PODYB, PODZ\rangle,$
$\langle PON, PODZ, PODZ\rangle, \langle PON, ECNA, ECNA\rangle,$
$\langle PON, ECNA, ECNB\rangle, \langle PON, ECNB, ECNB\rangle, \langle PON, ECNA, DC\rangle,$
$\langle PON, ECNB, DC\rangle, \langle PON, DC, DC\rangle, \langle PODYA, PODYA, PODYA\rangle,$
$\langle PODYA, PODYA, PODZ\rangle, \langle PODYA, PODYB, PODZ\rangle,$
$\langle PODYB, PODYB, PODZ\rangle, \langle PODYA, PODZ, PODZ\rangle,$
$\langle PODYB, PODZ, PODZ\rangle, \langle PODZ, PODZ, PODZ\rangle,$
$\langle ECNA, ECNA, ECNA\rangle, \langle ECNA, ECNA, DC\rangle, \langle ECNA, ECNB, DC\rangle,$
$\langle ECNB, ECNB, DC\rangle, \langle ECNA, DC, DC\rangle, \langle ECNB, DC, DC\rangle,$

$\langle DC, DC, DC\rangle, \langle TPPA\breve{}, ECD, PODYA\rangle, \langle TPPB\breve{}, ECD, PODYB\rangle,$
$\langle TPPA, ECD, ECNA\rangle, \langle TPPB, ECD, ECNB\rangle, \langle NTPP\breve{}, ECD, PODZ\rangle,$
$\langle NTPP, ECD, DC\rangle, \langle PON, ECD, PON\rangle\}$

2.6 Constraint Satisfaction Problem

2.6.1 Definitions and Axioms

The most popular reasoning methods used in QSR are constraint based techniques. It is necessary to have a set of qualitative binary basic relations which have the property of $JEPD$ in order to apply those reasoning methods. The set of all relations considered is then the set of all possible unions of the basic relations. Reasoning can be done by exploiting composition of relations. The composition operation is generally pre-computed and stored in a composition table.

Relationships between entities is often given in the form of constraints. For example, a customer specifies the outline of his future home to the architect by indicating which rooms should be close to each other. From this kind of specification binary constraints can be formed. A binary constraint is "washroom shall be away from kitchen" and a ternary constraint is "dining room should be between drawing and bedroom".

Binary constraints are consist of variables and relational expression. Relational expressions are recursively defined by the following:

- R where R is an atomic relation,
- $S \cup T$ where S and T are relational expressions,
- $R \cap S$ where R and S are relational expressions,
- $\breve{R}$ where R is a relational expressions,
- $\overline{R}$ where R is a relational expressions.

We can define a CSP consisting of a finite set of variables V, a domain D with possible instantiations for each variable $v_i \in V$ and a finite set C of constraints betweeen the variables of V. A solution of a CSP is an instantiation of each variable $v_i \in V$ with a value $d_i \in D$ such that all constraints of C are satisfied, i.e., for each constraint $v_i R v_j \in C$ we have $(d_i, d_j) \in R$. If a CSP has a solution, it is called consistent or satisfiable. A simple binary constraint is a constraint of the form xRy where R is an atomic relation. A CSP with simple constraints only is called a simple CSP. The set of constraints of an arbitrary CSP can be transformed into a set of simple CSP problems. The original problem is equivalent to this set in the following sense. The problem is satisfiable if one of the simple problems is satisfiable. The transformation is based on the following replacement rules where C is a simple binary constraint and $\{R, S, S_1 .... S_n\}$ is a set of atomic relations from the composition table and x and y are variables:

- $\{x(R \cup S)y\} \cup C$ is transformed into the two problems $\{xRy\} \cup C$ and $\{xSy\} \cup C$,
- $\{x(R \cap S)y\} \cup C$ is transformed to $\{xRy, xSy\} \cup C$,
- $\{x(\breve{R})y\} \cup C$ is transformed to $\{yRx\} \cup C$,
- $\{x\overline{R}y\} \cup C$ is transformed to $\{xS_1y, ..., xS_ny\} \cup C$.

As an example if we consider a set $A = \{xRy, x(S \cap (R \cup T))z, y\breve{S}z\}$ from set of constraints. The set A can be simplified to $\{xRy, xSz, x(R \cup T)z, zSy\}$. Now it can be represented by two sets as follows.

- $\{\ xRy, xSz, xRz, zSy\}$
- $\{\ xRy, xSz, xTz, zSy\}$

If there is a solution for any of those sets, then we can say CSP has a solution or it is satisfiable.

2.6.2 Path-consistency

As deciding consistency is highly complex, different forms of local consistency and algorithms were introduced for achieving local consistency. Path-consistency was developed as a local consistency by Montanir [43].

Definition 15. *[37] A CSP is path-consistent, if for every instantiation of two variables $v_i, v_j \in V$ that satisfies $v_iR_{ij}v_j \in C$ there exists an instantiation of every third variable $v_k \in V$ such that $v_iR_{ik}v_k \in C$ and $v_kR_{kj}v_j \in C$ are also satisfied.*

One algorithm that was developed by Montanir is the path consistency algorithm. The path-consistency algorithm removes locally inconsistent tuples from the relations between the variables by successively applying the $R_{ij} = R_{ij} \cap (R_{ik}; R_{kj})$ to all triples of variables $v_i, v_j \in V$ until a fixed point is reached. CSP is inconsistent if the empty relation occurs. Otherwise the resulting CSP is path-consistent.

Chapter 3

Composition Tables for RCC

In this chapter we want to generate more detailed composition tables for RCC starting with RCC8. After a brief review of how RCC11 was constructed from RCC8 we will reconstruct the tables for RCC15 and RCC25 which have been studied in [17]. Then we will continue to produce new tables by splitting atoms in RCC25. This way we obtain the new composition tables for RCC27, RCC29 and RCC31.

3.1 From RCC8 to RCC11

The composition table of RCC8 is given in Table 2.1. The universal or largest region 1 in a model of RCC can be characterized algebraically (or relationally). It was determined in [15] that the investigation of RCC can be restricted to the set $U = R \cap \setminus\{1\}$. Therefore, the relations EC and PO split into two disjoint non-empty relations ECN and ECD and PON and POD, respectively. Whenever x and y are related by EC or PO we can distinguish two situations depending on whether the union x and y is equal to the whole space or not, i.e., whether $x + y = 1$ or not. This leads to the following equivalent definitions for ECN and ECD, where P is restricted to $R \setminus \{0, 1\}$.

$$ECD = -(P; P^{\smile}) \cap -(P^{\smile}; P) \qquad xECDy \Longleftrightarrow y = \overline{x} \tag{3.1}$$

$$ECN = EC \cap -ECD \qquad xECNy \Longleftrightarrow x \cdot y = 0, x + y \not\leq 1, xCy \tag{3.2}$$

Figure 3.2 (Page 29) shows the diagram for ECN and ECD. In that figure two circles x and y are externally connected that is indicated by ECN. On the other hand for ECD a different shading issued for y in order to indicate, that is everything else x.

$$POD = PO \cap -(P;P^{\smile}) \qquad xPODy \Longleftrightarrow xPOy, x+y=1 \tag{3.3}$$

$$PON = PO \cap -POD \qquad xPONy \Longleftrightarrow xPOy, x+y \leq 1 \tag{3.4}$$

Using these definitions we get 10 disjoint atomic relations that are referred to as RCC10 relations [17]. We refer to [17] for the composition table of RCC10. The composition table of RCC10 does not have an extensional interpretation. In addition, the weak composition induced by this table is not associative. Properties of RCC10 relations are given in the Lemma 3.2.1. By splitting the relation POD of RCC10 into two new atoms $PODY$ and $PODZ$ we obtain RCC11 [59]. Diagrams of $PODY$ and $PODZ$ are shown in Figure 3.3 (Page 29). In that figure for $PODY$, y is indicated by everything else the white circle and x is indicated by different shading than y, that touches the border of y. For $PODZ$, x does not touch the white border of y. Among the 11 atoms of RCC11 seven are symmetric and four are non-symmetric atoms. The atoms of RCC11 are { $1'$, DC, ECN, ECD, PON, $PODY$, $PODZ$, TPP, $TPP^{\smile}$, $NTPP$, $NTPP^{\smile}$ }. The iterative splitting of EC and PO in the transition from RCC8 to RCC11 is shown in Figure 3.1 (Page 28). Finally, the composition table of RCC11 is given in Figure 3.5 (Page 30).

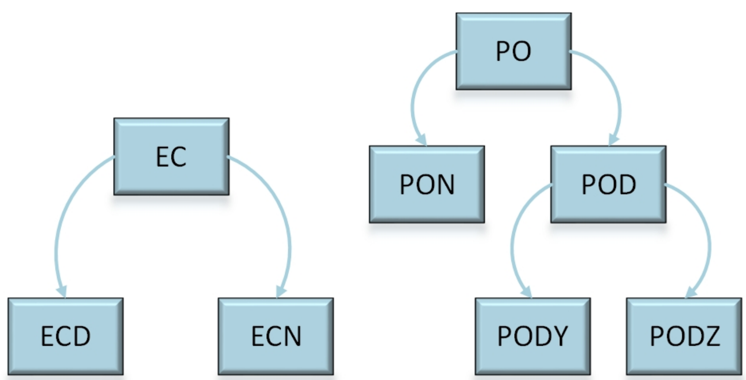


Figure 3.1: Splitting of atoms EC and PO

3.2 From RCC11 to RCC15

In order to obtain the composition table RCC15 consider the composition of ECN; TPP from the RCC11 composition table. The atomic relation TPP is the result from the composition, i.e., we have $TPP \cap ECN;TPP \neq \emptyset$ since the RCC11 table is consistent. We now provide an example where RCC11 cannot be extensional for

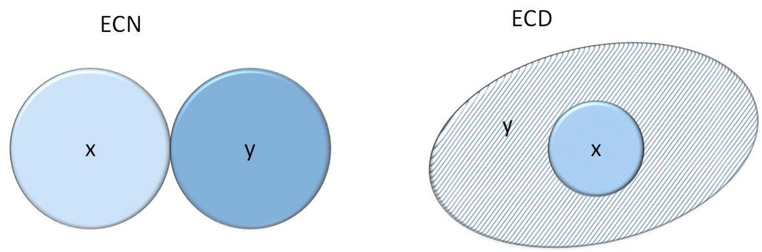


Figure 3.2: $xECNy$ and $xECDy$

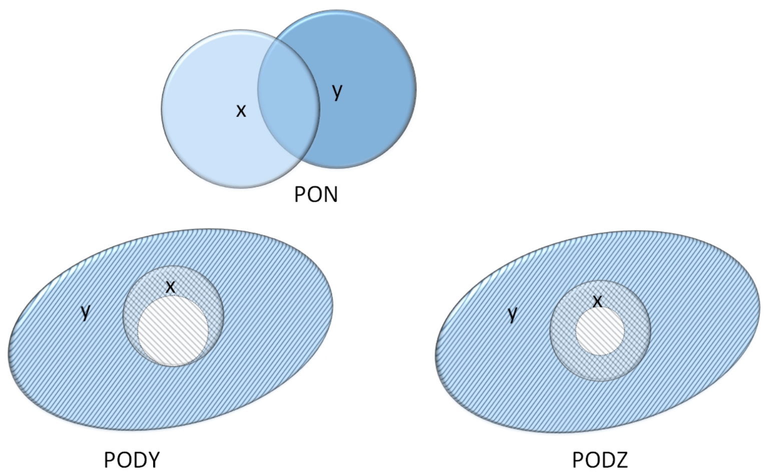


Figure 3.3: $xPONy$, $xPODYy$ and $xPODZy$

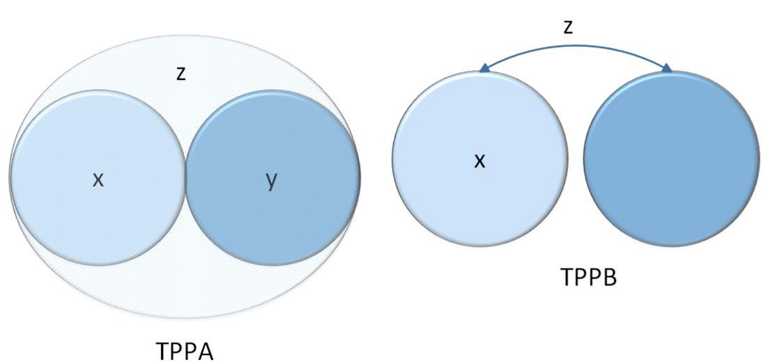


Figure 3.4: $xTPPAz$ and $xTPPBz$

any RCC model. For a detailed proof we refer to [59]. In order to do so we have to provide two regions so that $xTPPz$ but there is no y with $xECNyTPPz$. For completeness we also provide an example where such a y exists. These examples are provided in Figure 3.4 (Page 29). Notice that the figure only shows the situation in the Euclidean plane. However, the situation can be constructed in every RCC model [59]. As a consequence the relation TPP can be split into two new versions of TPP,

;	DC	ECN	ECD	PON	PODY	PODZ	TPP	TPP˘	NTPP	NTPP˘
DC	1', TPP, TPP˘ NTPP, NTPP˘ PON, ECN, DC	TPP, NTPP PON, ECN DC	NTPP	TPP, NTPP PON, ECN DC	NTPP	NTPP	TPP, NTPP PON, ECN, DC	DC	TPP, NTPP PON, PODY PODZ, ECN ECD, DC	DC
ECN	TPP˘, NTPP˘ PON, ECN, DC	1', TPP, TPP˘ PON, ECN, DC	TPP	TPP, NTPP PON, ECN, DC	TPP, NTPP	NTPP	TPP, NTPP PON, PODY ECN, ECD	ECN, DC	TPP, NTPP PON, PODY PODZ	DC
ECD	NTPP˘	TPP˘	1'	PON	TPP	NTPP	PODY	ECN	PODZ	DC
PON	TPP˘, NTPP˘ PON, ECN DC	TPP˘, NTPP˘ PON, ECN DC	PON	1', TPP, TPP˘ NTPP, NTPP˘ PON, PODY PODZ, ECN ECD, DC	TPP, NTPP PON, PODY PODZ	TPP, NTPP PON, PODY PODZ	TPP, NTPP PON, PODY PODZ	TPP˘, NTPP˘ PON, ECN, DC	TPP, NTPP PON, PODY PODZ	TPP˘, NTPP˘ PON, ECN DC
PODY	NTPP˘	TPP˘, NTPP˘	TPP˘	TPP˘, NTPP˘ PON, PODY PODZ	1', TPP, TPP˘ PON. PODY PODZ	TPP, NTPP PON, PODY PODZ	PODY, PODZ	TPP˘, NTPP˘ PON, PODY ECN, ECD	PODZ	TPP˘, NTPP˘ PON, ECN, DC
PODZ	NTPP˘	NTPP˘	NTPP˘	TPP˘, NTPP˘ PON, PODY PODZ	TPP˘, NTPP˘ PON, PODY PODZ	1', TPP, TPP˘ NTPP, NTPP˘ PON, PODY PODZ	PODZ	TPP˘ NTPP˘, PON PODY, PODZ	PODZ	TPP˘, NTPP˘ PON, PODY PODZ, ECN ECD, DC
TPP	DC	ECN, DC	ECN	TPP, NTPP PON, ECN, DC	TPP, NTPP PON, PODY ECN, ECD	TPP, NTPP PON, PODY PODZ	TPP, NTPP	1', TPP, TPP˘ PON, ECN DC	NTPP	TPP˘, NTPP˘ PON, ECN DC
TPP˘	TPP˘, NTPP˘ PON, ECN, DC	TPP˘, NTPP˘ PON, PODY ECN, ECD	PODY	TPP˘, NTPP˘ PON, PODY PODZ	PODY, PODZ	PODZ	1', TPP, TPP˘ PON,PODY PODZ	TPP˘, NTPP˘	TPP, NTPP PON, PODY PODZ	NTPP˘
NTPP	DC	DC	DC	TPP, NTPP PON, ECN, DC	TPP, NTPP PON, ECN, DC	TPP, NTPP PON, PODY PODZ, ECN ECD, DC	NTPP	TPP, NTPP PON, ECN DC	NTPP	1', TPP, TPP˘ NTPP, NTPP˘ PON, ECN DC
NTPP˘	TPP˘, NTPP˘ PON, PODY PODZ, ECN ECD, DC	TPP˘, NTPP˘ PON, PODY PODZ	PODZ	TPP˘, NTPP˘ PON, PODY PODZ	PODZ	PODZ	TPP˘, NTPP˘ PON, PODY PODZ	NTPP˘	1', TPP, TPP˘ NTPP, NTPP˘ PON, PODZ PODZ	NTPP˘

Figure 3.5: Composition Table for RCC11

one version for which the y in question always exists and one version for which the y never exists. In Figure 3.4, for $TPPA$, x is externally connected with y, i.e ECN. For $TPPB$, x is a touching proper part of z, where z is the union of two circles.

In the first step a new algebra is obtained by splitting TPP into a pair $TPPA$, $TPPB$ of identical copies of TPP. In the second step we remove the corresponding triple (ECN, TPP, TPP) for one of the copies from the composition table, i.e., we use the definitions

$$TPPA = TPP \cap ECN;TPP \tag{3.5}$$

$$TPPB = TPP \cap \overline{ECN;TPP} \tag{3.6}$$

During the first step of the process indicated above it may become necessary to split more atoms than originally intended. With respect to our example we know that TPP is non symmetric i.e., $TPP \neq TPP^{\smile}$. Since the converse of an atom needs to be an atom again, we need to split $TPP^{\smile}$ also, i.e., $TPP^{\smile} = TPPA^{\smile} \cup TPPB^{\smile}$ with,

$$TPPA^{\smile} = TPP^{\smile} \cap (ECN;TPP)^{\smile} \tag{3.7}$$

$$TPPB^{\smile} = TPP^{\smile} \cap \overline{(ECN;TPP)}^{\smile} \tag{3.8}$$

Furthermore, the composition table RCC11 shows that ECD is a bijection. It is easy to verify that the composition of an atom in a RA with a bijection from the left or the right is an atom. Therefore, we also need to split the $ECN = TPP;ECD$ and $PODY = ECD;TPP$. We name the new relations $ECNA$, $ECNB$ and $PODYA$, $PODYB$ respectively. Form the definition of $TPPA$ and $TPPB$ we obtain

$$ECNA = ECN \cap (TPP;TPP^{\smile}) \tag{3.9}$$

$$ECNB = ECN \cap \overline{(TPP;TPP^{\smile})} \tag{3.10}$$

$$PODYA = PODY \cap (TPP^{\smile};TPP) \tag{3.11}$$

$$PODYB = PODY \cap \overline{(TPP^{\smile};TPP)} \tag{3.12}$$

The splitting of the TPP relation of RCC11 algebra leads to the RCC15 relation algebra. Now in addition it is required to remove triples ($ECNA$, $TPPA$, $TPPB$), ($ECNA, TPPB, TPPB$), ($ECNB, TPPA, TPPB$) and ($ECNB, TPPB, TPPB$)., which are related to $TPPB$. Removing a triple requires the removal of a cycle because of the a Theorem 2.5.1(a). For a given triple ($ECNA$,$TPPA$,$TPPB$) related triples are obtained by composition with the bijection ECD from the left and/or right. For the previous mentioned triple, related triples are ($ECNA$, $TPPA$, $TPPB$), ($ECNA$, $ECNA$, $ECNB$), ($TPPA^{\smile}$, $TPPA$, $PODYB$), ($TPPA^{\smile}$, $ECNA$, $TPPB^{\smile}$), ($TPPA$, $PODYA$, $TPPB$), ($TPPA$, $TPPA^{\smile}$, $ECNB$), ($PODYA$, $PODYA$, $PODYB$) and ($PODYA$, $TPPA^{\smile}$, $TPPB^{\smile}$). Some properties of RCC15 relations are listed in the next lemma. A proof can be found in [15].

Lemma 3.2.1. *[15] Let $\mathfrak{B}$ be a relation algebra, and let $x, y, z \in B$. Then we have:*

1. $1' \in NTPP^{\smile};NTPP$, *i.e. for all z there is some x with $xNTPPz$.*
2. $ECN = TPP;ECD$, *i.e. $xECNz$ iff $xTPP\overline{z}$.*
3. *If $xDCz$, then $xTPP(x+z)$.*
4. *$xNTPPz$ and $yNTPPz$ iff $(x+y)NTPPz$.*
5. *If $xNTPPz$, then $(\overline{x} \cdot z)TPPz$*
6. $DC;P^{\smile} \in DC$, *i.e. $xDCy$ and $z \leq y$ imply $xDCz$.*
7. $NTPP = ECD;NTPP^{\smile};ECD$, *i.e. $xNTPPy$ iff $\overline{y}NTPP\overline{x}$.*

8. $P;NTPP \leq NTPP$, *i.e.* $x \leq y$ *and* $yNTPPz$ *imply* $xNTPPz$.

9. $NTPP;TPP = NTPP$

10. $NTPP;P = NTPP$

11. $TPP;NTPP = NTPP$

12. $1' \leq NTPP;NTPP\breve{}$, *i.e. for all* x *there is some* z *with* $xNTPPz$

13. $xNTPPy$ *and* $xNTPPz$ *iff* $xNTPPy \cdot z$.

14. $ECD;DC = NTPP\breve{}$,*i.e* $\overline{x}DCz$ *iff* $zNTPPx$.

15. $PON;ECD = PON$, *i.e.* $xPONz$ *iff* $xPON\overline{z}$

16. $TPP\breve{};ECD = POD \cap -(ECD;NTPP)$

17. $xECN;TPPz$ *iff* $xECN(\overline{x} \cdot z)TPPz$.

18. *If* $x \cdot z \neq 0$ *then* $x - (TPP\breve{};TPP)z$ *iff* $(x \cdot z)NTPPx$ *or* $(x \cdot z)NTPPz$

19. $xTPP\breve{};TPPz$ *iff* $xTPP\breve{}x \cdot zTPPz$

20. $xTPP;TPP\breve{}z$ *iff* $xTPP(x+z)TPP\breve{}z$

21. $yNTPP(x+z)$ *and* $yDCz$ *implies* $yNTPPx$

22. $PONZ \subseteq TPP;TPP\breve{}$.

23. $PONYB \subseteq TPP\breve{};TPP$.

24. $PONZ \subseteq TPP\breve{};TPP$.

25. $PODZ \subseteq POD$.

26. $PODZ \subseteq TPP\breve{};TPP$

In [17] Düntsch and Winter presented the RCC25 composition table. From this table as well as from the RCC15 table the additional cycle $\langle TPPA, TPPA, TPPB\rangle$ was removed. Unfortunately, the paper did not provide a proof that removal of this triple is correct, i.e., $TPPA;TPPA \cap TPPB = \emptyset$. We give the proof in the following lemma.

Lemma 3.2.2. $TPPA;TPPA \cap TPP \subseteq ECN;TPP$

Proof:

To prove the above lemma, assume $x(TPPA;TPPA \cap TPP)z$. Then there is a y so that $xTPPAyTPPAz$, i.e., we have (1) $xTPPy$ (2) $yTPPz$ (3) $xTPPz$ (4) $xECN(\overline{x} \cdot y)TPPy$ (5) $yECN(\overline{y} \cdot z)TPPz$ by Lemma 3.2.1 (17). We want to show (a) $xECN(\overline{x} \cdot z)$ and (b) $(\overline{x} \cdot z)TPPz$. To prove (a) at first we show $xTPP(x + \overline{z})$ which is equivalent to (a) by Lemma 3.2.1 (2). We have $x \leq x + \overline{z}$. If $x = x + \overline{z}$, then $\overline{z} \leq x$, and hence $\overline{x} \leq z$. But this implies $z = 1$, and, hence, $xTPP1$ by (3), which is a contradiction to $xNTPP1$ for all x. Therefore we have $xPP(x + \overline{z})$. Now assume $xNTPP(x + \overline{z})$. From the computation

$$\begin{aligned}(x + \overline{y}) \cdot (y + z) &= x \cdot (y + \overline{z}) + \overline{y} \cdot (y + \overline{z}) \\ &= x \cdot y + x \cdot \overline{z} + \overline{y} \cdot y + \overline{y} \cdot \overline{z} \\ &= x + 0 + 0 + \overline{z} \quad by \quad (1), (2) \quad and \quad (3) \\ &= x + \overline{z}\end{aligned}$$

we obtain $xNTPP(x+\overline{y}) \cdot (y+\overline{z})$, which implies $xNTPP(x+\overline{y})$ and $xNTPP(y+\overline{z})$ by Lemma 3.2.1 (13). The first property is equivalent to $xECD(\overline{x} \cdot y)$ by Lemma 3.2.1 (14), a contradiction to (3). Therefore we have $xTPP(x + \overline{z})$.

Now, in order to prove (b) we already have $\overline{x} \cdot z \leq z$. If $\overline{x} \cdot z = z$, then $z \leq \overline{x}$, and, hence, $x \leq z \cdot \overline{z} = 0$ by (3). But this is a contradiction to (3) since $0NTTPz$ for all z. We conclude $(\overline{x} \cdot z)PPz$. Now, assume $(\overline{x} \cdot z)NTPPz$. From the computation

$$\begin{aligned}\overline{x} \cdot y + \overline{y} \cdot z &= \overline{x} \cdot y + \overline{x} \cdot (x + \overline{y}) \cdot z \\ &= \overline{x} \cdot (y + (x + \overline{y}) \cdot z) \\ &= \overline{x} \cdot (y \cdot z + (x + \overline{y}) \cdot z) \\ &= \overline{x} \cdot (y + x + \overline{y}) \cdot z \\ &= \overline{x} \cdot z\end{aligned}$$

we get $(\overline{x} \cdot y + \overline{y} \cdot z)NTPPz$, and, hence, $\overline{x} \cdot yNTPPz$ and $\overline{y} \cdot zNTPPz$ by Lemma 3.2.1 (4). The second property is a contradiction to (5). So we conclude that $(\overline{x} \cdot z)TPPz$. All these facts prove the above lemma. $\square$

3.3 From RCC15 to RCC25

To generate the RCC25 table we started from RCC15. Düntsch and Winter [17] showed that RCC25 is generated based on splitting of atom PON. To split PON the composition of $(ECN;TPP)$ and $(TPP;TPP^{\smile})$ as well as their converses are taken into consideration and that shows 11 new atomic relations, among those the first 5 are symmetric atoms and the remaining 6 are non-symmetric atoms. We used Siddavaatam's system [50] that is developed based on Theorem 2.5.2. Figure 3.12 shows the splitting of PON using his system, where the field eta(η) and theta (θ) indicates non-symmetric and symmetric atoms. Symmetric atoms are $PONXA1, PONXA2, PONXB1, PONXB2$ and $PONZ$ and non-symmetric atoms are $PONYA1$, $PONYA2$, $PONYA1^{\smile}$, $PONYA2^{\smile}$, $PONYB$, $PONYB^{\smile}$. Diagrams of $PONXB2$, $PONXA1$, $PONXA2$, $PONXB1$, $PONYA1$ and $PONYA2$ are given in the Figure 3.6 (Page 36), Figure 3.7 (Page 37), Figure 3.8 (Page 37), Figure 3.9 (Page 37), Figure 3.10 (Page 37) and Figure 3.11 (Page 38) respectively. Düntsch and Winter [17] showed that each relation is non-empty and since ECD is a bijection relation we have to consider the following compositions related to ECD.

a) $TPPA; ECD = ECNA$

b) $TPPB; ECD = ECNB$

c) $TPPA^{\smile}; ECD = PODYA$

d) $TPPB^{\smile}; ECD = PODYB$

e) $PONXA1; ECD = PONXA1$

f) $PONXA2; ECD = PONYA1^{\smile}$

g) $PONXB1; ECD = PONYA1$

h) $PONXB2; ECD = PONZ$

i) $PONYA1; ECD = PONXB1$

j) $PONYA2; ECD = PONYB^{\smile}$

k) $PONYA1^{\smile}; ECD = PONXA2$

l) $PONYA2^{\smile}; ECD = PONYA2^{\smile}$

m) $PONYB; ECD = PONYB$

n) $PONYB^{\smile}; ECD = PONYA2$

o) $PONZ; ECD = PONXB2$

As PON is split into eleven relations for each PON we will keep the compositions related to ECD mentioned above and remove the other ten compositions as a result of other PONs to make the relation ECD a bijection. As an example, if we consider the relation $PONXA1$. We will remove the following triples:

- $PONXA1; ECD = PONXA2$
- $PONXA1; ECD = PONXB1$
- $PONXA1; ECD = PONXB2$
- $PONXA1; ECD = PONYA1$
- $PONXA1; ECD = PONYA2$
- $PONXA1; ECD = PONYB$
- $PONXA1; ECD = PONYA1^{\smile}$
- $PONXA1; ECD = PONYA2^{\smile}$
- $PONXA1; ECD = PONYB^{\smile}$
- $PONXA1; ECD = PONZ$

The same procedure is applied for all other relations $PONXA2$, $PONXB1$, $PONXB2$, $PONYA1$, $PONYA2$, $PONYA1^{\smile}$, $PONYA2^{\smile}$, $PONYB$, $PONYB^{\smile}$ and $PONZ$. So, in this way the total number of triples that we are removing is 110. Definitions of all atomic relations related to RCC25 are given in Table 3.1. Lemma 3.2.1 shows the reason of definition of relations $PONYB$, $PONZ$ and $PODZ$.

Considering the definitions of all relations of RCC25 from Table B.1, we also remove triples which are listed in Table 3.2. However while removing TPP, $TPP^{\smile}$ and ECN, we consider those relations as $TPPA$, $TPPB$, $TPPA^{\smile}$, $TPPB^{\smile}$, $ECNA$ and $ECNB$ because of their splitting. After splitting and removing all those triples we are checking the associativity of the algebra based on Theorem 2.5.1. A pair of triples will trigger if that pair is not associative for a given algebra. After that we check which triple should remain and removing other triple. We remove triple with its related triples, which are the products of relative multiplication by isomorphism of the algebra. As an example while generating RCC25 for a triggered pair $[(TPPA^{\smile}, PONZ, PODYA)$, $(PODYA$, TPPA$^{\smile}$, $ECD)]$, we are removing all isomorphic triples of $(TPPA^{\smile}, PONZ, PODYA)$, as this triple must not exist in the relation algebra because there is no such y for which x is related to $TPPA^{\smile}$ and z is related to $PONZ$. To generate RCC25 the list of triggered pairs with removed triple is given in Table B.2.

Relation	Triple Removed
$PONXA2$	$(TPP^{\smile}, TPP, PONXA2)$
$PONXB1$	$(TPP,TPP^{\smile},PONXB1)$
$PONXB2$	$(TPP, TPP^{\smile}, PONXB2)$, $(TPP^{\smile},TPP,PONXB2)$
$PONYA1$	$(ECN,TPP, PONYA1)$
$PONYA2$	$(ECN, TPP, PONYA2)$, $(TPP^{\smile}, TPP, PONYA2)$
$PONYA1^{\smile}$	$(TPP^{\smile},ECN, PONYA1^{\smile})$
$PONYA2^{\smile}$	$(TPP^{\smile}, ECN, PONYA2^{\smile})$, $(TPP^{\smile}$, ECN, $PONYA2^{\smile})$
$PONYB$	$(ECN, TPP, PONYB)$, $(TPP, TPP^{\smile}, PONYB)$
$PONYB^{\smile}$	$(TPP^{\smile}, ECN, PONYB^{\smile})$, $(TPP, TPP^{\smile}, PONYB^{\smile})$
$PONZ$	$(ECN, TPP, PONZ)$, $(TPP^{\smile}, ECN, PONZ)$

Table 3.1: Triple removed considering definitions of RCC25

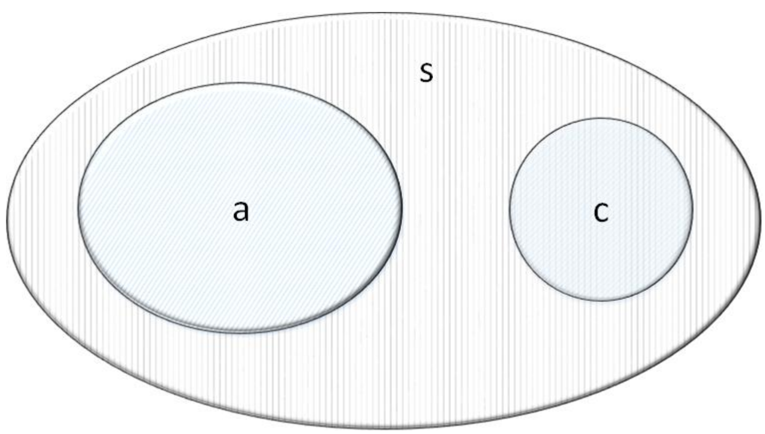


Figure 3.6: $(a + c)PONXB2(\overline{a} \cdot s)$

3.4 From RCC25 to RCC27

Now considering the diagram of $PONXB2$ that is given in Figure 3.6 (Page 36) we see that $xNTPP(x + z)$ where $x = (a + c)$ and $z = \overline{a} \cdot s$, but $z\overline{NTPP}(x + z)$. So this fact implies that $PONXB2$ can be split into two parts as $PONXB2H$ and $PONXB2H^{\smile}$.

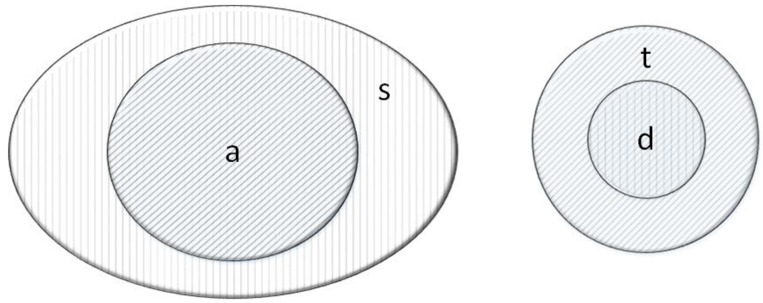


Figure 3.7: $(a + t)PONXA1(d + s)$

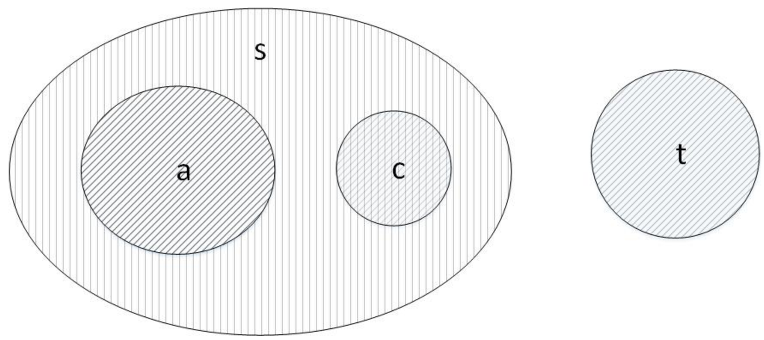


Figure 3.8: $(\overline{a} \cdot s)PONXA2(a + c + t)$

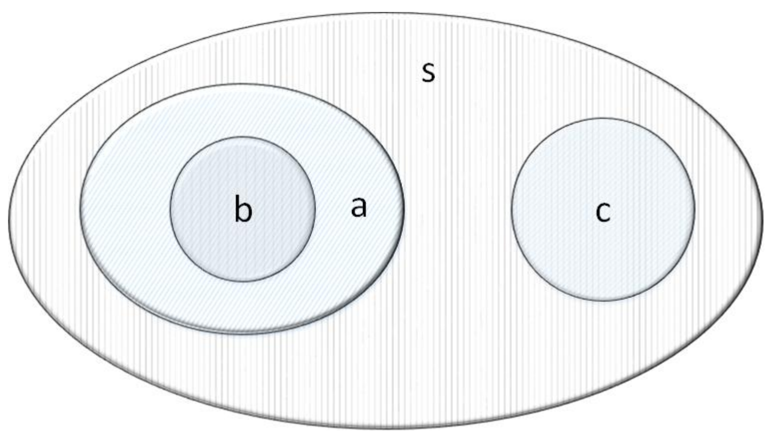


Figure 3.9: $(a + c)PONXB1\overline{a} \cdot (s + b)$

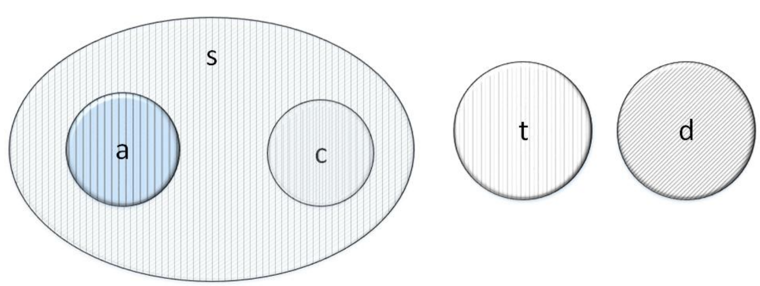


Figure 3.10: $(s + t)PONYA1(\overline{a} \cdot (s + d))$

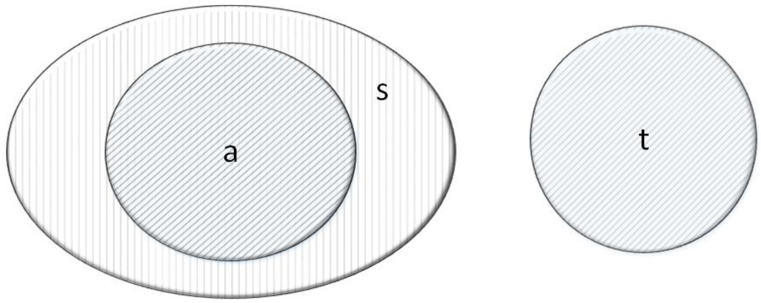


Figure 3.11: $sPONYA2(a+t)$

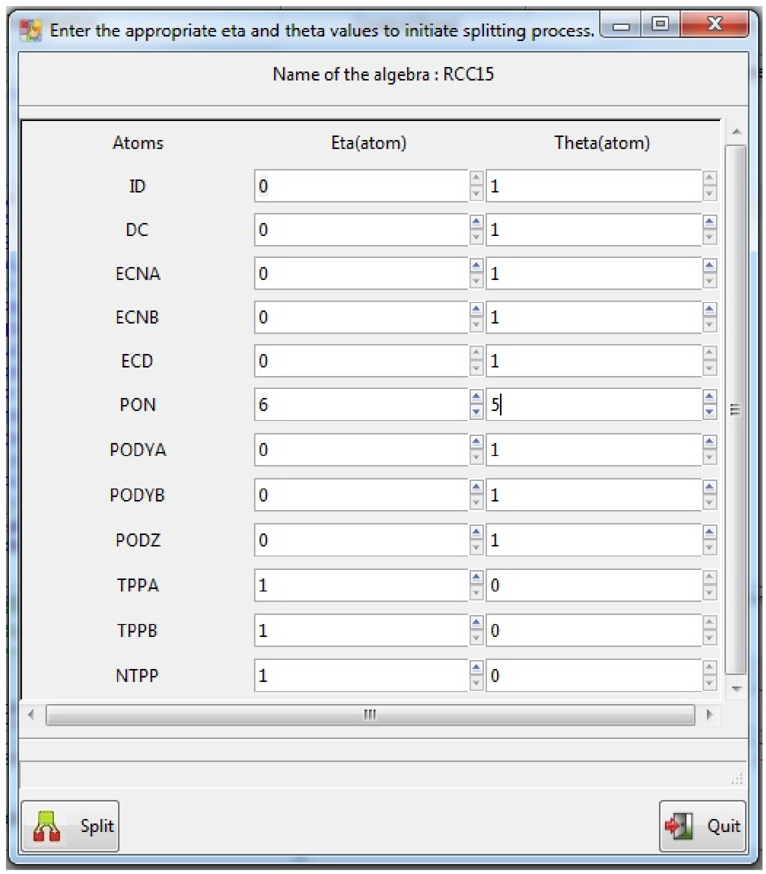


Figure 3.12: Splitting of the RCC15 relation algebra

$$xPONXB2Hz = xPONXB2z \cap xNTPP(x+z) \tag{3.13}$$

$$xPONXB2H\breve{}z = xPONXB2z \cap x\overline{NTPP}(x+z) \tag{3.14}$$

Later we will provide a justification for the name $PONXB2H\breve{}$ by showing that $PONXB2H\breve{}$ is indeed the converse of $PONXB2H$.

The definition of $PONXB2H$ is based on the condition $xNTPP(x+z)$. This property cannot be used to remove triples because it is not based on the composition of atomic relations. Instead it uses the algebraic operation $+$. In the following we want to show that $PONXB2H$ can be written in suitable way.

Lemma 3.4.1. $x(ECN;\overline{O})z \Leftrightarrow xTPP(x+z)$
Proof:

First we want to prove the implication $\Rightarrow$. For this purpose we have to find a region y with $xECNy\overline{O}z$, i.e., y has to satisfy (1) $x \cdot y = 0$ (2) xCy (3) $x + y \neq 1$ (4) $z \cdot y = 0$. Now choose $y = \overline{x} \cdot \overline{z}$. The properties (1) and (4) follow immediately from the definition of y. From the assumption we conclude that $x\overline{NTPP}(x+z)$, which is equivalent to (2) by Lemma 3.2.1. In order to show (3) assume that $x + y = 1$. Then we have

$$\begin{aligned} 1 &= x + (\overline{x} \cdot \overline{z}) \\ &= (x + \overline{x}) \cdot (x + \overline{z}) \\ &= (x + \overline{z}). \end{aligned}$$

This implies $z \leq x$, and, hence, $x + z = x$. But this is a contradiction to the assumption. For the other implication assume $xNTPP(x+z)$. From the assumption we obtain a y with (1)-(4) as listed above. First, we want to show that $xTPP\overline{y}$. From (1) we get $x \leq \overline{y}$. If $x = \overline{y}$, then $xECDy$, a contradiction to the assumption $xECNy$. Since xCy we conclude $xTPP\overline{y}$. On the other hand (1) and (4) show that $x+z \leq \overline{y}$. By our assumption $xNTPP(x+z)$ and Lemma 3.2.1(10) we get $xNTPP\overline{y}$, a contradiction. $\square$

Lemma 3.4.2. $xNTPP(x+z) \Leftrightarrow x(ECN \setminus O)z$
Proof:

First consider direction $\Rightarrow$. From $xNTPP(x+z)$ we get $x\overline{TPP}(x+z)$ since $x \leq x+z$. This is equivalent to $x(\overline{ECN;\overline{\overline{O}}})(x+z)$ by Lemma 3.4.1, and, hence, we have $x(ECN \setminus O)z$. Now consider the other implication. Let us assume $xTPP(x+z)$. Then we get $x(ECN;\overline{O})z$ from Lemma 3.4.1. But this contradicts with $(ECN \setminus O)$. If, $x = x+z$ then $z \leq x$. Now choose an a with $aNTPP\overline{x}$, which is possible because of Lemma 3.2.1(1). Then define $y = \overline{x} \cdot \overline{a}$. We want to show that $xECNy$. First, we have $x \cdot y \leq x \cdot \overline{x} = 0$. Now, assume $x\overline{C}y$ then $yNTPP\overline{x}$ from Lemma 3.2.1 (14) and hence we have $(y+a)NTPP\overline{x}$ from Lemma 3.2.1(4). Then we have

$$\begin{aligned} y+a &= (\overline{x}\cdot\overline{a})+a \\ &= (\overline{x}+a)\cdot(\overline{a}+a) \\ &= (\overline{x}+a) \\ &= \overline{x} \quad \text{since} \quad a \leq \overline{x}. \end{aligned}$$

This implies $\overline{x}NTPP\overline{x}$. But that is contradiction to the assumption. This fact implies xCy. If $x+y=1$ then $1 = x+(\overline{x}\cdot\overline{a})$, which is equivalent to $x+\overline{a}$ and that implis $a \leq x$. So we have $a \leq x+\overline{x} = 1$ and hence $aNTPP\overline{x}$ i.e $xECNy$. Now $y.z$ is equivalent to $\overline{x}\cdot\overline{a}\cdot z$ and that is 0. Which implies $z \leq x$. So, we can conclude that $y\overline{O}z$. □

The previous two lemmas show that

$$PONXB2H = PONXB2 \cap (ECN \setminus O) \tag{3.15}$$

This formula can be used for our method of splitting. But before we proceed with this procedure, we want to show that $PONXB2H\breve{}$ is indeed the converse of $PONXB2H$.

Lemma 3.4.3. $(ECN \setminus O) \cap (ECN \setminus O)\breve{} = \emptyset$
Proof:

From Lemma 3.4.2 $x(ECN\setminus O)z$ is equivalent to $xNTPP(x+z)$ and $z(ECN\setminus O)x$ is equivalent to $zNTPP(x+z)$. The latter two imply $(x+z)NTPP(x+z)$ by Lemma 3.2.1(4), which is a contradiction. □

Lemma 3.4.4. $\overline{(ECN \setminus O)} \cap \overline{(ECN \setminus O)\breve{}} \subseteq TPP;TPP\breve{}$
Proof:

Suppose we have $x\overline{(ECN \setminus O)}z$ and $x\overline{(ECN \setminus O)\breve{}}z$. Then Lemma 3.4.2 implies $x\overline{NTPP}(x+z)$ and $z\overline{NTPP}(x+z)$. But $x\overline{NTPP}(x+z)$ is equivalent to $xTPP(x+z)$ and $z\overline{NTPP}(x+z)$ equivalent to $zTPP(x+z)$ since $xPP(x+z)$ and $zPP(x+z)$. □

We can write Lemma 3.4.4 in the following ways

- $\overline{TPP;TPP\breve{}} \subseteq (ECN \setminus O) \cup (ECN \setminus O)\breve{}$
- $\overline{TPP;TPP\breve{}} \cap \overline{(ECN \setminus O)} \subseteq (ECN \setminus O)\breve{}$

Lemma 3.4.5. $PONXB2H^{\smile} = PONXB2 \cap (ECN \setminus O)^{\smile}$

Proof:

$$\begin{aligned} PONXB2H^{\smile} &= PONXB2 \cap \overline{(ECN \setminus O)} \\ &= PONXB2 \cap \overline{(ECN \setminus O)} \cap \overline{(TPP; TPP^{\smile})} \quad \textit{(by definition of } PONXB2) \\ &= PONXB2 \cap (ECN \setminus O)^{\smile}, \end{aligned}$$

where the last line follows from previous two lemmas. □

The next lemma will show precisely which cycles have to be removed during the splitting process for $PONXB2$ and $PONZ$.

Lemma 3.4.6. $PONXB2H = PONXB2 \cap \overline{(ECN; DC)}$

Proof:

$$\begin{aligned} PONXB2H &= PONXB2 \cap (ECN \setminus O) \\ &= PONXB2 \cap \overline{ECN; \overline{O}} \\ &= PONXB2 \cap \overline{ECN; (DC \cup ECD \cup ECN)} \\ &= PONXB2 \cap \overline{(ECN; DC) \cup (ECN; ECD) \cup (ECN; ECN)} \\ &= PONXB2 \cap \overline{(ECN; DC)} \cap \overline{(ECN; ECD)} \cap \overline{(ECN; ECN)} \\ &= PONXB2 \cap \overline{(ECN; DC)} \cap \overline{TPP} \cap \overline{(ECN; ECN)} \\ &= PONXB2 \cap \overline{(ECN; DC)}, \end{aligned}$$

where the last line follows from $PONXB2 \neq TPP$ and $PONXB2 \notin ECN; ECN$ □

Again from the RCC25 relation algebra we know that,

- $PONXB2; ECD = PONZ$
- $PONZ; ECD = PONXB2$
- $ECD; PONXB2 = PONZ$
- $ECD; PONZ = PONXB2$

So we also need to split $PONZ$. We are splitting $PONZ$ into $PONZH$ and $PONZH^{\smile}$ because $PONXB2$ is already split into two. The definitions of $PONZH$ and $PONZH^{\smile}$ are given below. Figure 3.13 shows the diagram of $PONZH$.

$$PONZH = PONXB2H; ECD \tag{3.16}$$

$$PONZH^{\smile} = PONXB2H^{\smile}; ECD \tag{3.17}$$

As before the lemma will show precisely which cycles have to be removed.

Lemma 3.4.7. $PONZH = PONZ \cap \overline{(ECN; NTPP)}$

Proof:

$$\begin{aligned} PONZH &= PONXB2; ECD \\ &= PONXB2 \cap \overline{(ECN; DC)}; ECD \\ &= PONXB2; ECD \cap \overline{(ECN; DC; ECD)} \\ &= PONZ \cap \overline{(ECN; NTPP)} \qquad \square \end{aligned}$$

If we split $PONXB2$ and $PONZ$ in RCC25, we obtain an algebra with 27 atoms, which we will call RCC27. During the splitting process we will remove initially the cycles listed below plus the cycles obtained from them by composing a cycle from the left and/or right with the bijection ECD. The first two cycles are removed in order to make sure that ECD remains a bijection. The other cycles are removed considering the definition of $PONXB2H$, $PONXB2H^{\smile}$, $PONZH$ and $PONZH^{\smile}$.

- $\langle PONXB2H, ECD, PONZH^{\smile} \rangle$
- $\langle PONXB2H^{\smile}, ECD, PONZH \rangle$
- $\langle ECNA, DC, PONXB2H \rangle$
- $\langle ECNB, DC, PONXB2H \rangle$
- $\langle DC, ECNA, PONXB2H^{\smile} \rangle$
- $\langle DC, ECNB, PONXB2H^{\smile} \rangle$
- $\langle ECNA, NTPP, PONZH \rangle$
- $\langle ECNB, NTPP, PONZH \rangle$
- $\langle NTPP^{\smile}, ECNA, PONZH^{\smile} \rangle$
- $\langle NTPP^{\smile}, ECNB, PONZH^{\smile} \rangle$

Then we continue by checking the associativity of RCC27. During this process additional cycles are being removed in the same way as we did in the case of RCC25. Triggered triples and the removed triples are listed in Table B.3.

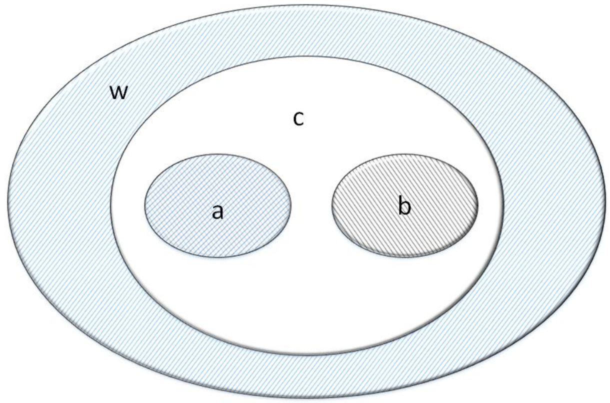


Figure 3.13: $(a+b)PONZH(a+\overline{c})$

3.5 From RCC25 to RCC29

Now let us look at the diagram of $PONXB1$ that is given in Figure 3.9 (Page 37). It is obvious that $xNTPP(x+z)$ holds for $PONXB1$ where $x = a+c$ and $z = \overline{a} \cdot (s+b)$ but, $zNTPP(x+z)$ not hold, so this indicates that we can also split $PONXB1$. $PONXB1$ is being split into two parts $PONXB1H$ and $PONXB1H^{\smile}$. Definitions of $PONXB1H$ and $PONXB1H^{\smile}$ are given below.

$$xPONXB1Hz = xPONXB1z \cap xNTPP(x+z) \tag{3.18}$$

$$xPONXB1H^{\smile}z = xPONXB1z \cap x\overline{NTPP}(x+z) \tag{3.19}$$

The same reasoning as for RCC27 leads to the following equations for the two relations.

$$PONXB1H = PONXB1 \cap (ECN \setminus O) \tag{3.20}$$

$$PONXB1H^{\smile} = PONXB1 \cap (ECN \setminus O)^{\smile} \tag{3.21}$$

The next lemmas will tell us which cycles have to be removed concretely.

Lemma 3.5.1. $PONXB1H = PONXB1 \cap \overline{(ECN; DC)}$

Proof:

$$\begin{aligned}
PONXB1H &= PONXB1 \cap (ECN \setminus O) \\
&= PONXB1 \cap \overline{ECN;\overline{\overline{O}}} \\
&= PONXB1 \cap \overline{ECN;(DC \cup ECD \cup ECN)} \\
&= PONXB1 \cap \overline{(ECN;DC) \cup (ECN;ECD) \cup (ECN;ECN)} \\
&= PONXB1 \cap \overline{(ECN;DC)} \cap \overline{(ECN;ECD)} \cap \overline{(ECN;ECN)} \\
&= PONXB1 \cap \overline{(ECN;DC)} \cap \overline{TPP} \cap \overline{(ECN;ECN)} \\
&= PONXB1 \cap \overline{(ECN;DC)},
\end{aligned}$$

where the last line follows from $PONXB1 \neq TPP$ and $PONXB1 \notin ECN;ECN$. □

From RCC25 we know that:

- $PONXB1;ECD = PONYA1$
- $PONYA1;ECD = PONXB1$
- $PONYA1^{\smile};ECD = PONXA2$
- $PONXA2;ECD = PONYA1^{\smile}$
- $ECD;PONXB1 = PONYA1^{\smile}$
- $ECD;PONYA1 = PONXA2$
- $ECD;PONYA1^{\smile} = PONXB1$
- $ECD;PONXA2 = PONYA1$

So now we can define $PONYA1H$, $PONYA1H^{\smile}$, $PONYA1tH$, $PONYA1tH^{\smile}$, $PONXA2H$ and $PONXA2H^{\smile}$ in the following ways:

- $PONYA1H = PONXB1H;ECD$
- $PONYA1H^{\smile} = ECD;PONXB1H^{\smile}$
- $PONYA1tH = PONXB1H^{\smile};ECD$
- $PONYA1tH^{\smile} = ECD;PONXB1H$
- $PONXA2H = ECD;PONXB1H^{\smile};ECD$
- $PONXA2H^{\smile} = ECD;PONXB1H;ECD$

Based on the definition we obtain the following equations that make the cycles, which have to be removed, explicitly.

Lemma 3.5.2. $PONYA1H = PONYA1 \cap \overline{(ECN; NTPP)}$
Proof:

$$\begin{aligned} PONYA1H &= PONXB1H; ECD \\ &= PONXB1 \cap \overline{(ECN; DC)}; ECD \\ &= PONXB1; ECD \cap \overline{(ECN; DC; ECD)} \\ &= PONYA1 \cap \overline{(ECN; NTPP)} \qquad \square \end{aligned}$$

Lemma 3.5.3. $PONYA1tH = PONYA1 \cap \overline{(DC; TPP)}$
Proof:

$$\begin{aligned} PONYA1tH &= PONXB1H^{\smile}; ECD \\ &= PONXB1 \cap \overline{(DC; ECN)}; ECD \\ &= PONXB1; ECD \cap \overline{(DC; ECN; ECD)} \\ &= PONYA1 \cap \overline{(DC; TPP)} \qquad \square \end{aligned}$$

Lemma 3.5.4. $PONXA2H = PONXA2 \cap \overline{NTPP^{\smile}; TPP}$
Proof:

$$\begin{aligned} PONXA2H &= PONYA1H^{\smile}; ECD \\ &= PONYA1^{\smile} \cap \overline{(NTPP^{\smile}; ECN)}; ECD \\ &= PONYA1^{\smile}; ECD \cap \overline{(NTPP^{\smile}; ECN; ECD)} \\ &= PONXA2 \cap \overline{(NTPP^{\smile}; TPP)} \qquad \square \end{aligned}$$

Lemma 3.5.5. *a)* $PONYA1H^{\smile} = PONYA1^{\smile} \cap \overline{NTPP^{\smile}; ECN}$

b) $PONYA1tH^{\smile} = PONYA1^{\smile} \cap \overline{(TPP^{\smile}; DC)}$

c) $PONXA2H^{\smile} = PONXA2 \cap \overline{(TPP^{\smile}; NTPP)}$

Proof:

Similar to the proofs of lemma 3.5.2, 3.5.3 and 3.5.4. □

RCC29 will be obtained by splitting $PONXB1$, $PONYA1$, $PONYA1^{\smile}$ and $PONXA2$. Removing the following cycles and those how are related by composing the bijection from the left and/or right will result in an algebra with 29 atoms. We call this algebra RCC29.

- $\langle PONYA1H, ECD, PONXB1H^{\smile}\rangle$
- $\langle PONYA1H^{\smile}, ECD, PONXA2H^{\smile}\rangle$
- $\langle PONYA1tH, ECD, PONXB1H\rangle$
- $\langle PONYA1tH^{\smile}, ECD, PONXA2H\rangle$
- $\langle PONXA2H, ECD, PONYA1tH^{\smile}\rangle$
- $\langle PONXA2H^{\smile}, ECD, PONYA1H^{\smile}\rangle$
- $\langle PONXB1H, ECD, PONYA1tH\rangle$
- $\langle PONXB1H^{\smile}, ECD, PONYA1H\rangle$
- $\langle ECNA, DC, PONXB1H\rangle$
- $\langle ECNB, DC, PONXB1H\rangle$
- $\langle DC, ECNA, PONXB1H^{\smile}\rangle$
- $\langle DC, ECNB, PONXB1H^{\smile}\rangle$

Diagrams showing examples for $PONYA1H$, $PONYA1tH$ and $PONXA2H$ are given in Figure 3.19 (Page 54), Figure 3.20 (Page 54) and Figure 3.21 (Page 54) respectively. Now we continue by checking the associativity of RCC29. During this process additional cycles are being removed in the same way as we did in the case of RCC25. Triggered triples and the removed triples are listed in Table B.4.

3.6 Generating RCC31

Now by combining the RCC27 and RCC29 relation algebras we will get RCC31. RCC27 is obtained by splitting $PONXB2$ and $PONZ$. In order to obtain RCC31 it is, therefore, sufficient to split these two relations in RCC29. An alternative way would be to split the new $PONXB1$, $PONYA1$, $PONYA1^{\smile}$, and $PONXA2$ in RCC27. During the associativity test for RCC31, only one pair of triples is triggered and that is given in Table 3.2.

No.	Triggered Pair	Removed Triple
1	$[(TPPA, TPPA\breve{}, PONYA1tH), (TPPB, PONYA1tH, PONXB2H\breve{})]$	$(TPPB, PONYA1tH, PONXB2H\breve{})$

Table 3.2: Triggered pair for the RCC31 algebra

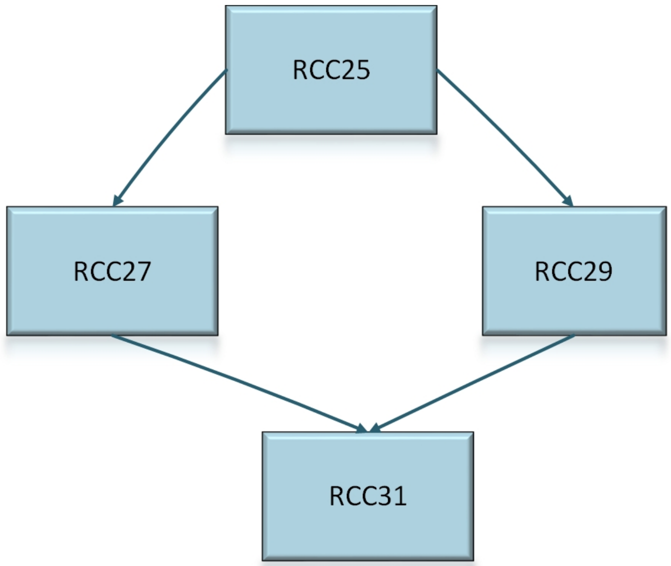


Figure 3.14: Generation of the RCC31 relation algebra

3.7 Splitting ECNB

Mormann [42] introduces the concept of a hole relation. The hole relation was defined by: $H = EC \cap \overline{(EC;\overline{O})}$. A restricted version of hole relation $H' = ECN \cap H = ECN \cap EC \cap \overline{(EC;\overline{O})}$ is also introduced in the same paper. $H' = ECN \cap H = ECN \cap EC \cap \overline{(EC;\overline{O})}$ was defined as $H' = ECN \cap H$, where $ECN = \{(x,y) : xECy, x \neq y'\}$ An example for the hole relation is given in Figure 3.15, where x is a Hole of z. Some basic properties of hole relations already presented in [42] are given below.

- H and H' are nonempty relations on U.
- xHy iff $xECy$ and $xNTPPx$ or $xNTPPy$
- xHy iff there is some $z \in U$ such that $xNTPPz$ and $y = z - x$
- The relation $ECNB$ splits as $ECNB = H' \cap H'^{\smile}$

The last property from above shows that the restricted hole relation can be obtained by splitting ECNB. In the remainder of the thesis we will focus on the restricted hole relation and write H instead of H'. We get

$$H = ECNB \cap (ECN \diagdown O) \tag{3.22}$$

$$H^{\smile} = ECNB \cap (ECN \diagdown O)^{\smile} \tag{3.23}$$

As before we want to make the cycles that have to be removed explicit.

Lemma 3.7.1. $H = ECNB \cap \overline{(ECN; DC)}$
Proof:

$$\begin{aligned}
ECNB \cap (ECN \setminus O) &= ECNB \cap \overline{ECN; \overline{O}} \\
&= ECNB \cap \overline{[ECN; (DC \cup ECD \cup ECN)]} \\
&= ECNB \cap \overline{(ECN; DC) \cup (ECN; ECD) \cup (ECN; ECN)} \\
&= ECNB \cap \overline{(ECN; DC)} \cap \overline{(ECN; ECD)} \cap \overline{(ECN; ECN)} \\
&= ECNB \cap \overline{(ECN; DC)} \cap \overline{TPP} \cap \overline{(ECN; ECN)}] \\
&= ECNB \cap \overline{(ECN; DC)},
\end{aligned}$$

where the last equation follows from $ECNB \neq TPP$ and $ECNB \notin ECN; ECN$. $\square$

From RCC25 we know that

- $ECNB; ECD = TPPB$
- $TPPB; ECD = ECNB$
- $TPPB^{\smile}; ECD = PODYB$
- $ECD; TPPB^{\smile} = ECNB$
- $PODYB; ECD = TPPB^{\smile}$
- $ECD; PODYB = TPPB$

Therefore, we also need to split $TPPB, TPPB^{\smile}$ and $PODYB$. This leads to the following definitions:

- $TPPB1 = H^{\smile}; ECD$
- $TPPB1^{\smile} = ECD; H$
- $TPPB2 = H; ECD$
- $TPPB2^{\smile} = ECD; H^{\smile}$

- $PODYBH = ECD; H; ECD$
- $PODYBH^{\smile} = ECD; H^{\smile}; ECD$

Again, in the following lemmas we make the cycles that have to be removed explicitly.

Lemma 3.7.2. $TPPB1 = TPPB \cap \overline{DC; TPP}$

Proof:

$$\begin{aligned} TPPB1 &= H^{\smile}; ECD \\ &= ECNB \cap \overline{(DC; ECN)}; ECD \\ &= ECNB; ECD \cap \overline{(DC; ECN; ECD)} \\ &= TPPB \cap \overline{(DC; TPP)} \qquad \square \end{aligned}$$

Lemma 3.7.3. $TPPB2 = TPPB \cap \overline{ECN; NTPP}$

Proof:

$$\begin{aligned} TPPB2 &= H; ECD \\ &= ECNB \cap \overline{(ECN; DC)}; ECD \\ &= ECNB; ECD \cap \overline{(ECN; DC; ECD)} \\ &= TPPB \cap \overline{(ECN; NTPP)} \qquad \square \end{aligned}$$

Lemma 3.7.4. $PODYBH = PODYB \cap \overline{(TPP^{\smile}; NTPP)}$

Proof:

$$\begin{aligned} PODYBH &= PPB1^{\smile}; ECD \\ &= TPPB^{\smile} \cap \overline{(TPP^{\smile}; DC)}; ECD \\ &= TPPB^{\smile}; ECD \cap \overline{(TPP^{\smile}; DC; ECD)} \\ &= PODYB \cap \overline{(TPP^{\smile}; NTPP)} \qquad \square \end{aligned}$$

To make sure ECD will act as a bijection relation, we remove the following cycles.

- $\langle TPPB1, ECD, H \rangle$
- $\langle TPPB1^{\smile}, ECD, PODYBH^{\smile} \rangle$
- $\langle TPPB2, ECD, H^{\smile} \rangle$

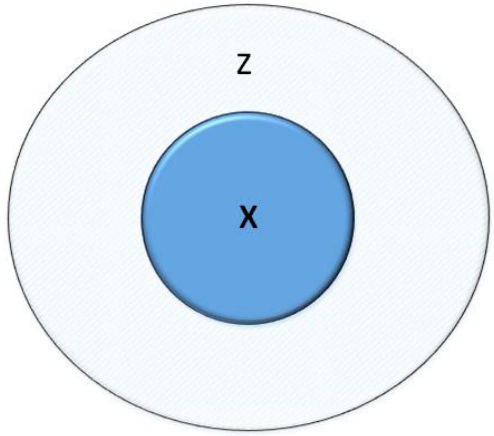


Figure 3.15: xHz

- $\langle TPPB2\breve{}, ECD, PODYBH\rangle$
- $\langle PODYBH, ECD, TPPB2\breve{}\rangle$
- $\langle PODYBH\breve{}, ECD, TPPB1\breve{}\rangle$

As $ECNB$ is split into two atomic relation H and $H\breve{}$, we are removing following cycles considering definitions of H and $H\breve{}$.

- $\langle ECNA, DC, H\rangle$
- $\langle H, DC, H\rangle$
- $\langle H\breve{}, DC, H\rangle$
- $\langle DC, ECNA, H\breve{}\rangle$
- $\langle DC, H, H\breve{}\rangle$
- $\langle DC, H\breve{}, H\breve{}\rangle$

We are also removing cycles considering the definiton of $TPPB1, TPPB1\breve{}, TPPB2$, $TPPB2\breve{}$, $PODYBH$ and $PODYBH\breve{}$

Table B.5 shows the list of triggered pairs during the splitting process for $ECNB$. For the last triggered pair it is not possible to remove any one of the pair as both triples are possible in any model of RCC (see Figure 3.17 (Page 51) and Figure 3.18) (Page 52). We are interested what the cause of this inconsistency is. Since cycles represent up to six triples, these cycles correspond to multiple situations in the associativity condition of Theorem 2.5.1. All those situations are listed in Table 3.3. In

No.	Matching Cycle Set	u exist
1	[$(TPPA, TPPB1, TPPA, TPPB2, TPPA)$]	0
2	[$(TPPA, TPPB1, TPPA,$ $TPPB2\breve{}$,TPPA)]	1
3	[$(TPPB1$, $TPPA\breve{}, TPPA\breve{}$, $TPPA, TPPB2)$]	1
4	[$(TPPB1$, $TPPA\breve{}, TPPA\breve{}$, $TPPA$, $TPPB2\breve{})$]	1
5	[$(TPPB1\breve{}, TPPA\breve{}, TPPA\breve{}, TPPA, TPPB2$)]	0
6	[$(TPPB1\breve{}, TPPA\breve{}, TPPA\breve{}$, $TPPA$, $TPPB2\breve{})$]	1
7	[$(TPPA$, $TPPB1\breve{}$, $TPPA, TPPB2, TPPA$)]	1
8	[$(TPPA$, $TPPB1\breve{}$, $TPPA$, $TPPB2\breve{}$, $TPPA$)]	1

Table 3.3: Match Table

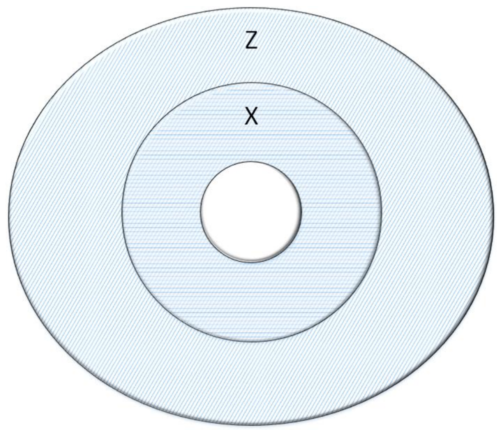


Figure 3.16: $xTPPB1z$

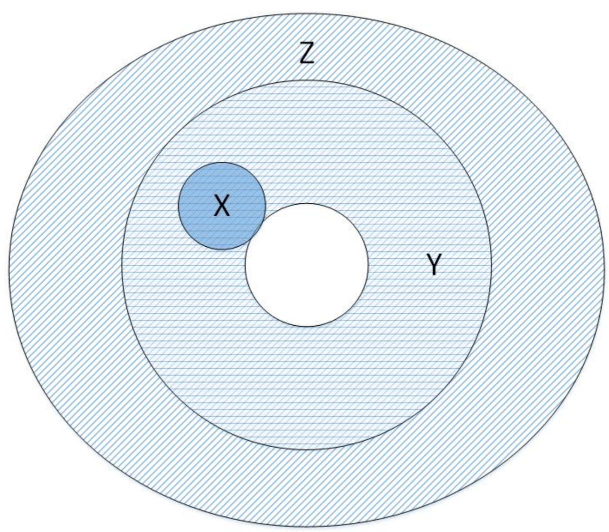


Figure 3.17: $xTPPAz$

that table, the column "u exists" indicates whether the u required by Theorem 2.5.1 exists or not. Obviously, we are interested in the first and fifth row containing a zero, i.e., indicating that such a u does not exist. The first row can be visualized by the following diagram:

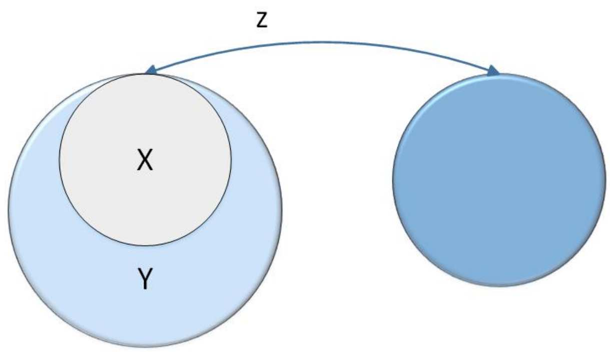


Figure 3.18: $xTPPAz$

$$\begin{tikzcd}
 & \bullet \arrow[d, "TPPB1"] \arrow[dr, dotted, "u"] & \\
\bullet \arrow[ur, "TPPA"] \arrow[r, "TPPA"'] \arrow[rr, bend right, "TPPA"'] & \bullet \arrow[r, "TPPB2"'] & \bullet
\end{tikzcd}$$

Possible candidates for u with respect to the composition $TPPB1; TPPB2$ are $TPPA$, $TPPB1$, $TPPB2$ and $NTPP$. On the other hand, if we require that $TPPB2$ is contained in $TPPB1^{\smile}; u$, we have the following candidates for u:

- $PONXA1$, $PONXB1$, $PONZ$, $PODYA$, $PODZ$, $PONYA1$, $PONYA1^{\smile}$, $PONYB$, $PONYB^{\smile}$, $TPPA$, $TPPA^{\smile}$, $TPPB1$, $TPPB1^{\smile}$, $TPPB2$, $TPPB2^{\smile}$

$TPPA, TPPB1$ and $TPPB2$ are the common candidates. Therefore, we will consider triples those are related with these three relations only. Among our considered triples, $(TPPB1, TPPB2, TPPA)$ was removed in Step 61 and $(TPPB1, TPPB2, TPPB1)$ and $(TPPB1, TPPB2, TPPB2)$ are removed in Step 5 of Table B.5. The second situation in which the required u does not exist is visualized as follows:

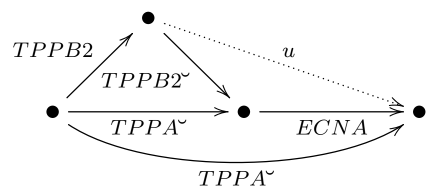
$$\begin{tikzcd}
 & \bullet \arrow[d, "TPPA^{\smile}"] \arrow[dr, dotted, "u"] & \\
\bullet \arrow[ur, "TPPB1^{\smile}"] \arrow[r, "TPPA^{\smile}"'] \arrow[rr, bend right, "TPPB2"'] & \bullet \arrow[r, "TPPA"'] & \bullet
\end{tikzcd}$$

Triple $(TPPB1^{\smile}, TPPA, TPPB2)$ is removed from step 61. Triples $(TPPB1^{\smile}, TPPB1, TPPB2)$ and $(TPPB1^{\smile}, TPPB2, TPPB2)$ are related to the removing triple of row 5 from Table B.5.

If we continue to chase the problem backwards, i.e., analyzing what the cause for Step 5 and Step 61 is, we come back to Step 3. In step 3 we get the following matching for which the required u does not exist:

- $(TPPB2, TPPB2^{\smile}, TPPA^{\smile}, ECNA, TPPA^{\smile})$
- $(TPPB2, TPPB2^{\smile}, TPPA, TPPA^{\smile}, ECNA)$

The following diagram is obtained from the first situation:



The composition result of $TPPB2^{\smile}; ECNA$ is given as follows

- $ECNA$, $PONXA1$, $PONXA2$, $PONXB1$, $PONXB2$, H, $H^{\smile}$, $PODYBH$, $PODYBH^{\smile}$, $PONYA1$, $PONYA2$, $PONYB$

On the other hand, candidates for u so that $TPPA^{\smile}$ is included in $TPPB2; u$ are $TPPA^{\smile}$, $TPPB1^{\smile}$, $TPPB2^{\smile}$, $NTPP^{\smile}$.

If we look at both results there is no common atomic relation. Since $NTPP^{\smile}$ is included in the second set, let us consider $NTPP^{\smile}$ as a result of the composition of $TPPB2^{\smile}; ECNA$. We obtain the following cycles:

- $\langle ECNA, DC, H \rangle$
- $\langle DC, ECNA, H^{\smile} \rangle$
- $\langle DC, TPPA, TPPB1 \rangle$
- $\langle ECNA, NTPP, TPPB2 \rangle$
- $\langle TPPA^{\smile}, NTPP, PODYBH \rangle$
- $\langle NTPP^{\smile}, TPPA, PODYBH^{\smile} \rangle$

The above list is related to the definitions of H, $H^{\smile}$, $TPPB1$, $TPPB2$, $PODYBH$ and $PODYBH^{\smile}$. A similar argument applies to all other potential candidates. This means that the definition of H and $H^{\smile}$ and related relations by splitting does not lead to a relation algebra, a situation similar to RCC10. It should be possible to obtain a relation algebra by splitting at least one atom in addition. So far we were not able to identify that atom and the condition defining its subrelations. So we leave this problem for future investigation.

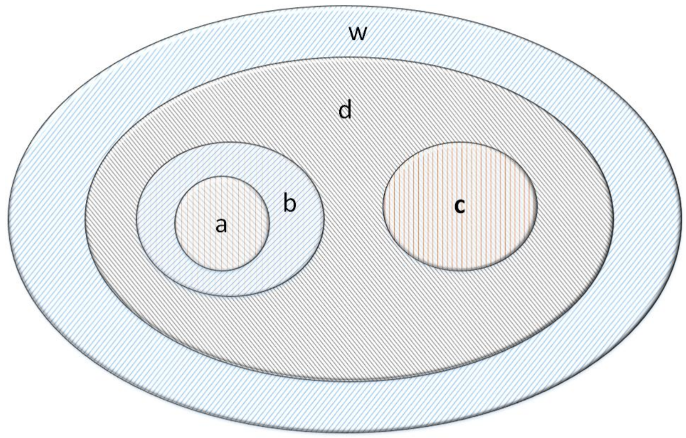


Figure 3.19: $(b + c)PONYA1H(\overline{b} \cdot d + a)$

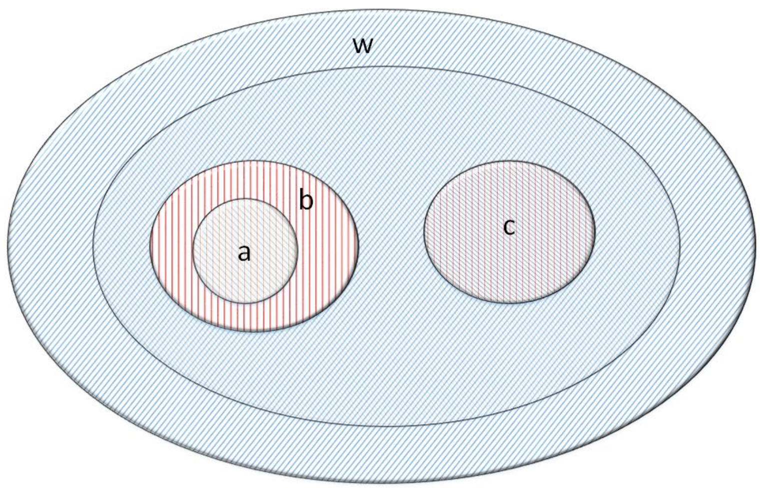


Figure 3.20: $(\overline{b} \cdot d + a)PONYA1tH(\overline{b} + \overline{c})$

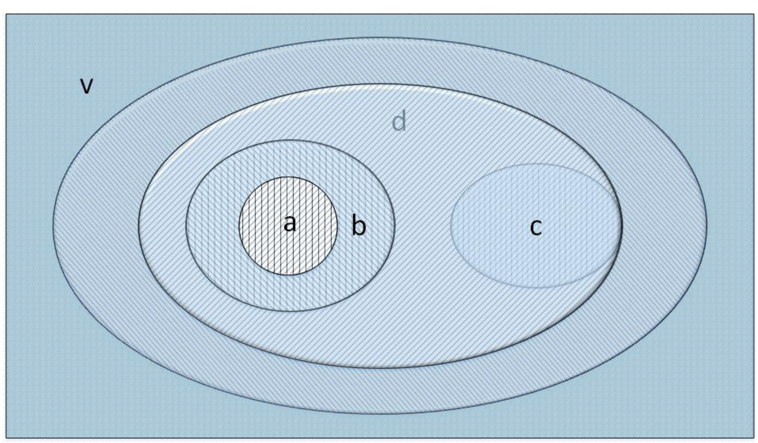


Figure 3.21: $(\overline{a} \cdot b + \overline{d} \cdot v)PONXA2H(\overline{b} + \overline{c})$

Chapter 4

Constraint Satisfaction Problem for RCC

Knowledge between different entities or knowledge about relations between entities can be represented by constraints. To formulate constraints about spatial entities, spatial calculi are used, which can be represented as constraint satisfaction problem. A CSP can be represented as a graph with the nodes corresponding to the variables and the arcs corresponding to constraints. If an assignment for all variables to values of the domain can be found that satisfies all constraints, then we can say CSP is consistent. Otherwise it is not.

Different algorithms such as path consistency, arc consistency and k-consistency have been extensively studied for this kind of problem. Renz [37] has shown that if the consistency problem for CSP is decidable for a certain subset $S \subseteq 2^B$, where B is a set of atomic relations, then the solution remains for other subsets of 2^B by using a non-deterministic algorithm. Trudel [54] proved that a constraint is part of a consistent scenario of a non-finite interval algebra network if and only if it is a consistent scenario of a finite interval algebra network. The CSP is a more appropriate and successful approach for reasoning about spatial qualitative constraint networks. In classical CSPs relations are finite, and they can be explicitly manipulated as a set of tuples of a finite domain.

In spatial CSPs the domains of spatial variables are usually infinite. A usual way to deal with relations of qualitative spatial variables is to have a finite set of JEPD relations. The relations of a JEPD set are atomic relations. To represent the knowledge, we can use these relations by using CSPs and use constraint based techniques to decide whether such a problem is consistent or not. Operators such as union, complement, converse, intersection and composition are connected with relations. Composition is not as straight forward as other operators, because it has to be computed only for pairs of atomic relations. Computing the composition may not be feasible for domains of arbitrary spatial regions that are not well structured and if there is no common representation of the region. Therefore the composition can be approximated by using weak composition. The point of using weak composition is that the result will remain within the given set of relations.

If the given constraint is satisfied then we can say constraints that we are considering will be subsets of regions for a particular selected algebra. The constraint is composed of *composition* and *join* operators. Relation algebra that we are taking into account are $RCC8$, $RCC11$, $RCC15$, $RCC25$, $RCC27$, $RCC29$ and $RCC31$. For example we consider a constraint like "$washroomTPPbedroom$, $bedroomECNdrawingroom$, $washroomECNdrawingroom$" where $washroom$, $bedroom$ and $drawingroom$ are variables, and TPP and ECN are relations from RCC11.

The way our developed system works to check constraints is given in Figure 4.1. For example the constraint may be

$$\begin{aligned}&washroomDCbedroom,\\&bedroomPODYdrawingroom,\\&washroomTPPdrawingroom\ OR\ washroomNTPPdrawingroom\end{aligned}$$

From the above constraint it is clear that if the composition of DC and $PODY$ is either TPP or $NTPP$, then the given constraint is satisfiable based on the composition. Now it is required to check whether the entered constraint is satisfiable or not based on the composition table of a particular selected algebra. The flowchart for manipulating the constraint is given in Figure 4.2. For example, if the constraint is "washroomDCbedroom, bedroomPODYdrawingroom, washroomTPPbedroom" then it is not a satisfiable constraint as there is no relation with 'washroom' and 'drawingroom'. So we don't need to check this constraint by the composition table of a particular algebra.

There are different sections like 'Select Relation Algebra', 'Enter Name of Variable', 'Enter Constraint' and 'Test Constraint' in the user interface of the application 'Constraint Satisfaction Checking'. They are marked by red square in Figure B.1. The 'Add variable' button adds variables for the constraint string that we need to check. While entering the name for variables, we have to make sure the name of variables should start with a lowercase letter. For example in Figure B.2 we have entered four variables in our system. In the 'Select Relation Algebra' section there are seven

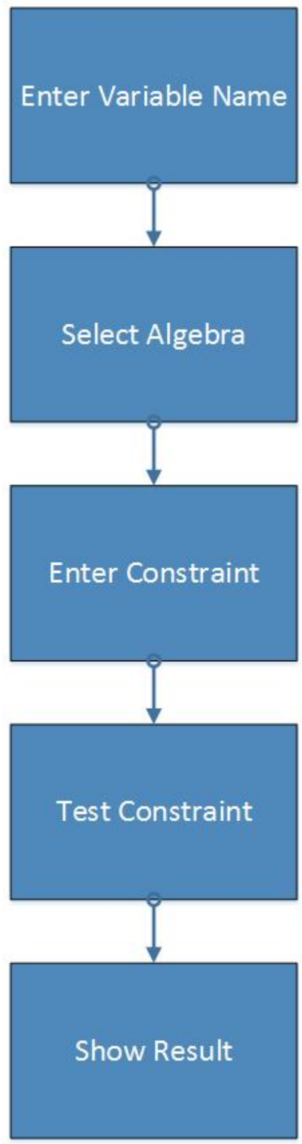


Figure 4.1: Constraint string manipulation

radio buttons named with different relation algebras. After selecting a particular algebra, all the atomic relations related to that algebra will appear below the text box of where to enter a constraint. For example if 'RCC8' is selected eight atomic relations will appear (Figure B.3). Those atomic relations are 'ID, DC, EC, PO, TPP, $TPP^{\smile}$, NTPP and $NTPP^{\smile}$'. By clicking on the buttons of the variable and the relation we can enter a constraint. We have to insert ',' after inserting a relation with two variables.

For example, we select a relation algebra "RCC11" and enter four variables: "washroom", "bedroom", "drawingroom" and "kitchen". After that we enter a constraint string "washroomTPPbedroom, bedroomECNdrawingroom, washroomECNdrawingroom" in the text area of "Enter Constraint". As the relation "washroomECNdrawingroom" is present in the constraint string, this is a satisfiable constraint. Next we check the constraint by the composition table of RCC11. In the composition table of RCC11 there exists the relation ECN as the result of composition of TPP and ECN. So the given constraint satisfies the selected algebra (Figure B.4). Again, if we enter a constraint like "washroomTPPbedroom, bedroomECNdrawingroom, washroomTPPdrawingroom" then the given constraint is satisfiable based on compostion but it would not satisfy the selected algebra RCC11. This is because TPP is not there as a result of composition of TPP and ECN. Figure B.5 shows this scenario.

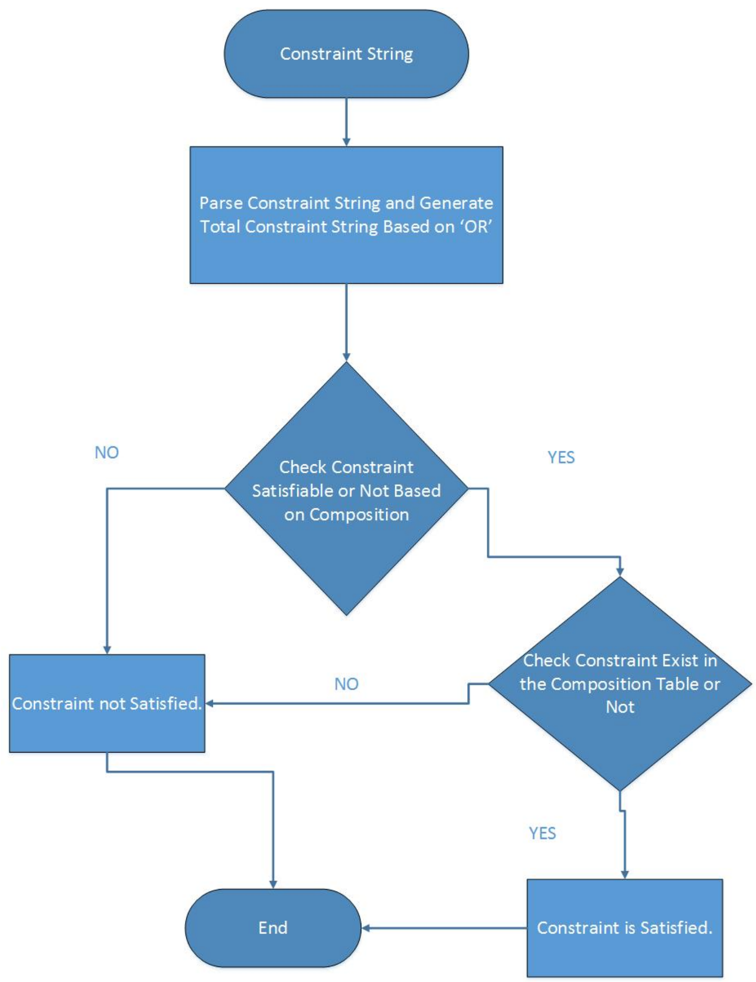


Figure 4.2: Constraint string manipulation

Chapter 5

Conclusion and Future Work

In this chapter we will review the contents covered in this thesis. We will suggest some work and investigation that can be carried out in the future.

We started by introducing Allen's [1] interval calculus that lead us to define the composition table. Then we have shown binary relations, Boolean algebra, and their properties from where we have defined relation algebra and their properties. We have discussed contact algebra that is based on contact relations where spatial regions are used instead of points. Region connection calculus is defined based on contact relation "C'. We have also discussed atom structure and how an atomic relation can be split from old algebras to form new algebras. Based on the splitting mechanism, atomic relations $PONXB1$, $PONXB2$ and $ECNB$ were split from relation algebra RCC25. However we are not able to split $ECNB$ successfully and leave this problem for future investigation. We have also given a proof showing that the triple ($TPPA$, $TPPA$, $TPPB$) can be removed from RCC15 as well as from RCC25. In the context of spatial reasoning we also defined the constraint satisfaction problem. We have developed a system in Java to check whether the given constraint is satisfied or not.

There are also two $TPPA$ situations that are given below.

$$TPPA1 = TPPA \cap TPPA; TPPB2 \tag{5.1}$$

$$TPPA2 = TPPA \cap TPPA; TPPB2 \tag{5.2}$$

So if we are splitting $TPPA$ we also need to split $ECNA$, as $ECNA$ is related to $TPPA$ by ECD. We are also not able to split $ECNB$. Further investigation can be done and may be it would be possible to generate more atoms beyond atoms of RCC31 algebra. Spatial regions that we are considering are circles. So polygons can be considered instead of circles in future endeavors.

Siddavaatam's system [50] was developed in the functional programming language Haskell. One of the drawbacks of his system that it is very slow for relation algebras with large N values, mainly where N$\geq$15. For graphical user interface design he used the GTK+ toolkit along with Haskell library Gtk2Hs. However it is tiresome to install this open source GTK+ toolkit and integrate with Haskell. The method to install Glade and GTK for Haskell is given in the appendix. We have converted some basic functions in C that are used by Siddavaatam's system to check the associativity of the algebra. Basic functions related to splitting and associativity can be implemented in other languages like Java and $C\sharp$. If that works fine, then a full system can be developed with that language.

Appendix A

Installing Glade and GTK for Haskell

It would be recommended to install all files related to Haskell, MINGW, GTK and GLADE in the same directory. At first Haskell and MinGW need to be installed. Haskell can be downloaded from [33] and after installing Haskell, MinGW needs to be installed and it can be downloaded from [34]. Now to install libxml download and unzip the latest libxml2 and libxml2-dev from the Gnome site, that is [35], to the folder where we are installing all files related to Haskell, Glade and GTK. After unzipping the libxml contents the name of the folder would be "libxml22.7.7-1win32" and "libxml2-dev2.7.7-1win32 ". We also need to copy the contents of /bin and /manifest of "libxml22.7.7-1win32" to "libxml2-dev2.7.7-1win32 ". To install the GTK/Glade bundle we have to download that from [35]. We also have to set values for environment variables. As an example for the environment variable 'PKGCONFIGPATH' we have set the location of 'pkgconfig' directory. There would be not any 'PKGCONFIGPATH' in environment variables so we have to create that variable. For example we have assigned the value " D:/Program/Gtk+/lib/pkgconfig;D:/Program/libxml2-dev/lib/pkgconfig" for the 'PKGCONFIGPATH' variable. For the INCLUDE environment variable we add the location of 'libglade-2.0' directory. Our environment variable should be look like:
"D:\ Program\ Gtk+\ include\ libglade-2.0;D:\ Program \ libxml2-dev\ include;D:\ Program\ Gtk+\ include".

Appendix B

Tables and Figures

No.	Definition
1	1’
2	$TPPA = TPP \cap (ECN; TPP)$
3	$TPPA^{\smile} = TPP^{\smile} \cap (ECN; TPP)^{\smile}$
4	$TPPB = TPP \cap -(ECN; TPP)$
5	$TPPB^{\smile} = TPP^{\smile} \cap -(ECN; TPP)^{\smile}$
6	$NTPP$
7	$NTPP^{\smile}$
8	$PONXA1 = PON \cap (ECN; TPP) \cap (ECN; TPP)^{\smile} \cap (TPP; TPP^{\smile}) \cap (TPP^{\smile}; TPP)$
9	$PONXA2 = PON \cap (ECN; TPP) \cap (ECN; TPP)^{\smile} \cap (TPP; TPP^{\smile}) \cap -(TPP^{\smile}; TPP)$
10	$PONXB1 = PON \cap (ECN; TPP) \cap (ECN; TPP)^{\smile} \cap -(TPP; TPP^{\smile}) \cap (TPP^{\smile}; TPP)$
11	$PONXB2 = PON \cap (ECN; TPP) \cap (ECN; TPP)^{\smile} \cap -(TPP; TPP^{\smile}) \cap -(TPP^{\smile}; TPP)$
12	$PONYA1 = PON \cap -(ECN; TPP) \cap (ECN; TPP)^{\smile} \cap (TPP; TPP^{\smile}) \cap (TPP^{\smile}; TPP)$
13	$PONYA2 = PON \cap -(ECN; TPP) \cap (ECN; TPP)^{\smile} \cap (TPP; TPP^{\smile}) \cap -(TPP^{\smile}; TPP)$
14	$PONYA1^{\smile} = PON \cap (ECN; TPP) \cap -(ECN; TPP)^{\smile} \cap (TPP; TPP^{\smile}) \cap (TPP^{\smile}; TPP)$
15	$PONYA2^{\smile} = PON \cap (ECN; TPP) \cap -(ECN; TPP)^{\smile} \cap (TPP; TPP^{\smile}) \cap -(TPP^{\smile}; TPP)$
16	$PONYB = PON \cap -(ECN; TPP) \cap (ECN; TPP)^{\smile} \cap -(TPP; TPP^{\smile})$
17	$PONYB^{\smile} = PON \cap (ECN; TPP) \cap -(ECN; TPP)^{\smile} \cap -(TPP; TPP^{\smile})$
18	$PONZ = PON \cap -(ECN; TPP) \cap -(ECN; TPP)^{\smile}$
19	$PODYA = POD \cap -(ECD; NTPP) \cap (TPP^{\smile}; TPP)$
20	$PODYB = POD \cap -(ECD; NTPP) \cap -(TPP^{\smile}; TPP)$
21	$PODZ = ECD; NTPP$
22	$ECNA = ECN \cap (TPP; TPP^{\smile})$
23	$ECNB = ECN \cap -(TPP; TPP^{\smile})$
24	ECD
25	DC

Table B.1: Definitions of RCC25 atoms

No.	Triggered Pair	Removed Triple
1	$[(TPPA^{\smile}, PONZ, PODYA),(PODYA,TPPA^{\smile},ECD)]$	$(TPPA^{\smile}, PONZ, PODYA)$
2	$[(TPPA^{\smile}, PONYA1, PODYA),(PODYA,TPPA^{\smile},ECD)]$	$(TPPA^{\smile}, PONYA1, PODYA)$
3	$[(TPPA^{\smile}, PONYA2, PODYA),(PODYA,TPPA^{\smile},ECD)]$	$(TPPA^{\smile}, PONYA2, PODYA)$
4	$[(TPPA^{\smile}, PONYB, PODYA),(PODYA,TPPA^{\smile},ECD)]$	$(TPPA^{\smile}, PONYB, PODYA)$
5	$[(TPPA^{\smile}, PONZ, PODYB),(PODYB,TPPB^{\smile},ECD)]$	$(TPPA^{\smile}, PONZ, PODYB)$
6	$[(TPPA^{\smile}, PONYA1, PODYB),(PODYB, TPPB^{\smile},ECD)]$	$(TPPA^{\smile}, PONYA1, PODYB)$
7	$[(TPPA^{\smile}, PONYA2, PODYB),(PODYB,TPPB^{\smile},ECD)]$	$(TPPA^{\smile}, PONYA2, PODYB)$
8	$[(TPPA^{\smile}, PONYB, PODYB),(PODYB,TPPB^{\smile},ECD)]$	$(TPPA^{\smile}, PONYB, PODYB)$
9	$[(TPPB^{\smile}, PONZ, PODYA),(PODYA,TPPA^{\smile},ECD)]$	$(TPPB^{\smile}, PONZ, PODYA)$
10	$[(TPPB^{\smile}, PONYA1, PODYA),(PODYA,TPPA^{\smile},ECD)]$	$(TPPB^{\smile}, PONYA1, PODYA)$
11	$[(TPPB^{\smile}, PONYA2, PODYA),(PODYA,TPPA^{\smile},ECD)]$	$(TPPB^{\smile}, PONYA2, PODYA)$
12	$[(TPPB^{\smile}, PONYB, PODYA),(PODYA,TPPA^{\smile},ECD)]$	$(TPPB^{\smile}, PONYB, PODYA)$
13	$[(TPPB^{\smile}, PONZ, PODYB),(PODYB,TPPB^{\smile},ECD)]$	$(TPPB^{\smile}, PONZ, PODYB)$
14	$[(TPPB^{\smile}, PONYA1, PODYB),(PODYB,TPPB^{\smile},ECD)]$	$(TPPB^{\smile}, PONYA1, PODYB)$
15	$[(TPPB^{\smile}, PONYA2, PODYB), (PODYB, TPPB^{\smile},ECD)]$	$(TPPB^{\smile}, PONYA2, PODYB)$
16	$[(TPPB^{\smile}, PONYB, PODYB), (PODYB, TPPB^{\smile}, ECD)]$	$(TPPB^{\smile}, PONYB, PODYB)$
17	$[(PONXA2, PODYA, PODYA), (PODYA, TPPA^{\smile},ECD)]$	$(PONXA2, PODYA, PODYA)$
18	$[(PONXB2, PODYA, PODYA),(PODYA, TPPA^{\smile},ECD)]$	$(PONXB2, PODYA, PODYA)$
19	$[(PONYA2, PODYA, PODYA), (PODYA, TPPA^{\smile},ECD)]$	$(PONYA2, PODYA, PODYA)$
20	$[(PONXA2, PODYB, PODYA),(PODYA,TPPA^{\smile},ECD)]$	$(PONXA2,PODYB, PODYA)$
21	$[(PONXB2, PODYB, PODYA),(PODYA,TPPA^{\smile},ECD)]$	$(PONXB2, PODYB, PODYA)$
22	$[(PONYA2, PODYB, PODYA),(PODYA,TPPA^{\smile},ECD)]$	$(PONYA2,PODYB, PODYA)$
23	$[(PONYA2, PODYB, PODYB),(PODYA,TPPB^{\smile},ECD)]$	$(PONYA2, PODYB, PODYB)$
24	$[(PONXA2, PODYB, PODYB),(PODYB,TPPB^{\smile},ECD)]$	$(PONXA2, PONYB, PODYB)$
25	$[(PONXB2, PODYB, PODYB),(PODYB,TPPB^{\smile},ECD)]$	$(PONXB2, PONZ, PODYA)$
26	$[(PONXB1, ECNA, ECNA),(ECNA, TPPA, ECD)]$	$(PONXB1, ECNA, ECNA)$
27	$[(PONXB2, ECNA, ECNA), (ECNA, TPPA, ECD)]$	$(PONXB2, ECNA, ECNA)$
28	$[(PONYB, ECNA, ECNA), (ECNA, TPPA, ECD)]$	$(PONYB, ECNA, ECNA)$
29	$[(PONYA2, PODYB, PODYB),(TPPB^{\smile},ECD, PODYB)]$	$(PONYA2, PODYB, PODYB)$
30	$[(PONYA2, PODYA, PODYB),(TPPA^{\smile}, ECD, PODYA)]$	$(PONYA2, PODYA, PODYB)$
31	$[(PONXB1, ECNA, ECNB), (TPPA,ECD, ECNA)]$	$(PONXB1, ECNA, ECNB)$
32	$[(PONXB1, ECNB, ECNA), (ECNA, TPPA, ECD)]$	$(PONXB1, ECNB, ECNA)$
33	$[(PONYB^{\smile}, ECNB, ECNA),(ECNA,TPPA,ECD)]$	$(PONYB^{\smile}, ECNB, ECNA)$
34	$[(PONYB, ECNB, ECNA),(ECNA, TPPA, ECD)]$	$(PONYB, ECNB, ECNA)$
35	$[(PONXB2, ECNA, ECNB),(TPPA, ECD, ECNA)]$	$(PONXB2, ECNA, ECNB)$
36	$[(PONXB1, ECNB, ECNB),(TPPB,ECD,ECNB)]$	$(PONXB1, ECNB, ECNB)$
37	$[(PONXB2, ECNB, ECNB),(TPPB,ECD,ECNB)]$	$(PONXB2, ECNB, ECNB)$
38	$[(PONYB, ECNB, ECNB),(TPPB,ECD,ECNB)]$	$(PONYB, ECNB, ECNB)$

Table B.2: Triggered pairs for the RCC25 algebra

No.	Triggered Pair	Removed Triple
1	$[(TPPA, NTPP^{\smile}, PONXB2H), (TPPA, ECNA, ECNA)]$	$(TPPA, NTPP^{\smile}, PONXB2H)$
2	$[(TPPA, TPPB, TPPA), (TPPB, PONXB2H, NTPP)]$	$(TPPB, PONXB2H, NTPP)$
3	$[(TPPA, TPPA^{\smile}, PONXA1), (TPPA, PONXA1, PONXB2H^{\smile})]$	$(TPPA, PONXA1, PONXB2H^{\smile})$
4	$[(TPPA, TPPA^{\smile}, PONXA1), (TPPB, PONXA1, PONXB2H^{\smile})]$	$(TPPB, PONXA1, PONXB2H^{\smile})$
5	$[(TPPA, TPPA^{\smile}, PONXA1), (NTPP, PONXA1, PONXB2H^{\smile})]$	$(NTPP, PONXA1, PONXB2H^{\smile})$
6	$[(TPPA, TPPA^{\smile}, PONXA2), (TPPA, PONXA2, PONXB2H^{\smile})]$	$(TPPA, PONXA2, PONXB2H^{\smile})$
7	$[(TPPA, TPPA^{\smile}, PONXA2), (TPPB, PONXA2, PONXB2H^{\smile})]$	$(TPPB, PONXA2, PONXB2H^{\smile})$
8	$[(TPPA, TPPA^{\smile}, PONXA2), (NTPP, PONXA2, PONXB2H^{\smile})]$	$(NTPP, PONXA2, PONXB2H^{\smile})$
9	$[(TPPA, TPPA^{\smile}, PONZH), (TPPA, PONZH, PONXB2H^{\smile})]$	$(TPPA, PONZH, PONXB2H^{\smile})$
10	$[(TPPA, TPPA^{\smile}, PONZH), (TPPA, PONZH^{\smile}, PONXB2H^{\smile})]$	$(TPPA, PONZH^{\smile}, PONXB2H^{\smile})$
11	$[(TPPA, TPPA^{\smile}, PONZH), (TPPB, PONZH, PONXB2H^{\smile})]$	$(TPPB, PONZH, PONXB2H^{\smile}, ECD)$
12	$[(TPPA, TPPA^{\smile}, PONZH), (TPPB, PONZH^{\smile}, PONXB2H^{\smile})]$	$(TPPB, PONZH^{\smile}, PONXB2H^{\smile})$
13	$[(TPPA, TPPA^{\smile}, PONZH), (NTPP, PONZH, PONXB2H^{\smile})]$	$(NTPP, PONZH, PONXB2H^{\smile})$
14	$[(TPPA, TPPA^{\smile}, PONZH), (TPPB, PONZH^{\smile}, PONXB2H^{\smile})]$	$(TPPB, PONZH^{\smile}, PONXB2H^{\smile})$
15	$[(TPPA, TPPA^{\smile}, PONYA1), (TPPA, PONYA1, PONXB2H^{\smile})]$	$(TPPA, PONYA1, PONXB2H^{\smile})$
16	$[(TPPA, TPPA^{\smile}, PONYA1), (TPPA, PONYA1^{\smile}, PONXB2H^{\smile})]$	$(TPPA, PONYA1^{\smile}, PONXB2H^{\smile})$
17	$[(TPPA, TPPA^{\smile}, PONYA1), (TPPB, PONYA1, PONXB2H^{\smile})]$	$(TPPB, PONYA1, PONXB2H^{\smile})$
18	$[(TPPA, TPPA^{\smile}, PONYA1), (TPPB, PONYA1^{\smile}, PONXB2H^{\smile})]$	$(TPPB, PONYA1^{\smile}, PONXB2H^{\smile})$
19	$[(TPPA, TPPA^{\smile}, PONYA1), (NTPP, PONYA1, PONXB2H^{\smile})]$	$(NTPP, PONYA1, PONXB2H^{\smile})$
20	$[(TPPA, TPPA^{\smile}, PONYA1), (NTPP, PONYA1^{\smile}, PONXB2H^{\smile})]$	$(NTPP, PONYA1^{\smile}, PONXB2H^{\smile})$
21	$[(TPPA, TPPA^{\smile}, PONYA2), (TPPA, PONYA2, PONXB2H^{\smile})]$	$(TPPA, PONYA2, PONXB2H^{\smile})$
22	$[(TPPA, TPPA^{\smile}, PONYA2), (TPPA, PONYA2^{\smile}, PONXB2H^{\smile})]$	$(TPPA, PONYA2^{\smile}, PONXB2H^{\smile})$
23	$[(TPPA, TPPA^{\smile}, PONYA2), (TPPB, PONYA2, PONXB2H^{\smile})]$	$(TPPB, PONYA2, PONXB2H^{\smile})$
24	$[(TPPA, TPPA^{\smile}, PONYA2), (TPPB, PONYA2^{\smile}, PONXB2H^{\smile})]$	$(TPPB, PONYA2^{\smile}, PONXB2H^{\smile})$
25	$[(TPPA, TPPA^{\smile}, PONYA2), (NTPP, PONYA2, PONXB2H^{\smile})]$	$(NTPP, PONYA2, PONXB2H^{\smile})$
26	$[(TPPA, TPPA^{\smile}, PONYA2), (NTPP, PONYA2^{\smile}, PONXB2H^{\smile})]$	$(NTPP, PONYA2^{\smile}, PONXB2H^{\smile})$
27	$[(TPPA, NTPP^{\smile}, PONXB1), (TPPA, PONXB1, PONXB2H^{\smile})]$	$(TPPA, PONXB1, PONXB2H^{\smile})$
28	$[(TPPA, NTPP^{\smile}, PONXB1), (TPPB, PONXB1, PONXB2H^{\smile})]$	$(TPPB, PONXB1, PONXB2H^{\smile})$
29	$[(TPPA, NTPP^{\smile}, PONXB1), (NTPP, PONXB1, PONXB2H^{\smile})]$	$(NTPP, PONXB1, PONXB2H^{\smile})$
30	$[(TPPA, NTPP^{\smile}, PONXB2H^{\smile}), (TPPA, PONXB2H, PONXB2H^{\smile})]$	$(TPPA, PONXB2H, PONXB2H^{\smile})$
31	$[(TPPA, NTPP^{\smile}, PONXB2H^{\smile}), (TPPB, PONXB2H, PONXB2H^{\smile})]$	$(TPPB, PONXB2H, PONXB2H^{\smile})$
32	$[(TPPA, NTPP^{\smile}, PONXB2H^{\smile}), (NTPP, PONXB2H, PONXB2H^{\smile})]$	$(NTPP, PONXB2H, PONXB2H^{\smile})$
33	$[(TPPA, NTPP^{\smile}, PONYB), (TPPA, PONYB^{\smile}, PONXB2H^{\smile})]$	$(TPPA, PONYB^{\smile}, PONXB2H^{\smile})$
34	$[(TPPA, NTPP^{\smile}, PONYB), (TPPB, PONYB^{\smile}, PONXB2H^{\smile})]$	$(TPPB, PONYB^{\smile}, PONXB2H^{\smile})$
35	$[(TPPA, NTPP^{\smile}, PONYB), (NTPP, PONYB^{\smile}, PONXB2H^{\smile})]$	$(NTPP, PONYB^{\smile}, PONXB2H^{\smile})$
36	$[(TPPA, NTPP^{\smile}, PONYB^{\smile}), (TPPA, PONYB, PONXB2H^{\smile})]$	$(TPPA, PONYB, PONXB2H^{\smile})$
37	$[(TPPA, TPPA, NTPP), (NTPP, PONXB2H^{\smile}, PONYA2^{\smile})]$	$(NTPP, PONXB2H^{\smile}, PONYA2^{\smile})$
38	$[(TPPA, NTPP^{\smile}, PONYB^{\smile}), (TPPB, PONYB, PONXB2H^{\smile})]$	$(TPPB, PONYB, PONXB2H^{\smile})$

Table B.3: Triggered pairs for the RCC27 algebra

No.	Triggered Pair	Removed Triple
1	$[(TPPA,NTPP^{\smile}, PONXB1H),(TPPA, ECNA, ECNA)]$	$(TPPA,NTPP^{\smile}, PONXB1H)$
2	$[(TPPA,TPPB,TPPA),(TPPB,PONXA2H^{\smile},NTPP)]$	$(TPPB,PONXA2H^{\smile},NTPP)$
3	$[(TPPA,TPPA^{\smile}, PONXA1),(TPPA, PONXA1,PONXB1H^{\smile})]$	$(TPPA, PONXA1, PONXB1H^{\smile})$
4	$[(TPPA, TPPA^{\smile}, PONXA1),(TPPB,PONXA1,PONXB1H^{\smile})]$	$(TPPB, PONXA1, PONXB2H^{\smile})$
5	$[(TPPA,TPPA^{\smile}, PONXA1),(NTPP,PONXA1,PONXB1H^{\smile})]$	$(NTPP, PONXA1, PONXB1H^{\smile})$
6	$[(TPPA, TPPA^{\smile}, PONXA2H),(TPPA, PONXA2H, PONXB1H^{\smile})]$	$(TPPA, PONXA2H, PONXB1H^{\smile})$
7	$[(TPPA, TPPA^{\smile}, PONXA2H),(TPPA,PONXA2^{\smile},PONXB1H^{\smile})]$	$(TPPA,PONXA2^{\smile},PONXB1H^{\smile})$
8	$[(TPPA,TPPA^{\smile}, PONXA2H),(TPPB, PONXA2H,PONXB1H^{\smile})]$	$(TPPB, PONXA2H,PONXB1H^{\smile})$
9	$[(TPPA,TPPA^{\smile}, PONXA2H),(TPPB,PONXA2H^{\smile},PONXB1H^{\smile})]$	$(TPPB,PONXA2H^{\smile},PONXB1H^{\smile})$
10	$[(TPPA,TPPA^{\smile}, PONXA2H),(NTPP,PONXA2H,PONXB1H^{\smile})]$	$(NTPP,PONXA2H,PONXB1H^{\smile})$
11	$[(TPPA,TPPA^{\smile}, PONXA2H),(NTPP,PONXA2H^{\smile},PONXB1H^{\smile})]$	$(NTPP,PONXA2H^{\smile},PONXB1H^{\smile})$
12	$[(TPPA,TPPA^{\smile}, PONZH),(TPPA, PONZH ,PONXB1H^{\smile})]$	$(TPPA, PONZH, PONXB1H^{\smile})$
13	$[(TPPA, TPPA^{\smile}, PONZH),(TPPB, PONZ,PONXB1H^{\smile})]$	$(TPPB, PONZ,PONXB1H^{\smile})$
14	$[(TPPA,TPPA^{\smile}, PONZ),(NTPP, PONZ, PONXB1H^{\smile})]$	$(NTPP, PONZ, PONXB1H^{\smile})$
15	$[(TPPA, TPPA^{\smile}, PONYA1H),(TPPA,PONYA1H,PONXB1H^{\smile})]$	$(TPPA,PONYA1H,PONXB1H^{\smile})$
16	$[(TPPA,TPPA^{\smile}, PONYA1H),(TPPA,PONYA1H^{\smile},PONXB1H^{\smile})]$	$(TPPA,PONYA1H^{\smile},PONXB1H^{\smile})$
17	$[(TPPA,TPPA^{\smile}, PONYA1H),(TPPB,PONYA1H,PONXB1H^{\smile})]$	$(TPPB,PONYA1H,PONXB1H^{\smile})$
18	$[(TPPA,TPPA^{\smile}, PONYA1H),(TPPB,PONYA1H^{\smile},PONXB1H^{\smile})]$	$(TPPB,PONYA1H^{\smile},PONXB1H^{\smile})$
19	$[(TPPA,TPPA^{\smile}, PONYA1H),(NTPP,PONYA1H,PONXB1H^{\smile})]$	$(NTPP,PONYA1H,PONXB1H^{\smile})$
20	$[(TPPA,TPPA^{\smile}, PONYA1H),(NTPP,PONYA1H^{\smile},PONXB1H^{\smile})]$	$(NTPP,PONYA1H^{\smile},PONXB1H^{\smile})$
21	$[(TPPA,TPPA^{\smile}, PONYA1tH),(TPPA,PONYA1tH,PONXB1H^{\smile})]$	$(TPPA,PONYA1tH,PONXB1H^{\smile})$
22	$[(TPPA,TPPA^{\smile}, PONYA1tH),(TPPA,PONYA1tH^{\smile},PONXB1H^{\smile})]$	$(TPPA,PONYA1tH^{\smile},PONXB1H^{\smile})$
23	$[(TPPA,TPPA^{\smile}, PONYA1tH),(TPPB,PONYA1tH,PONXB1H^{\smile})]$	$(TPPB,PONYA1tH,PONXB1H^{\smile})$
24	$[(TPPA,TPPA^{\smile}, PONYA1tH),(TPPB,PONYA1tH^{\smile},PONXB1H^{\smile})]$	$(TPPB,PONYA1tH^{\smile},PONXB1H^{\smile})$
25	$[(TPPA,TPPA^{\smile}, PONYA1tH),(NTPP,PONYA1tH,PONXB1H^{\smile})]$	$(NTPP,PONYA1tH,PONXB1H^{\smile})$
26	$[(TPPA,TPPA^{\smile}, PONYA1tH),(NTPP,PONYA1tH^{\smile},PONXB1H^{\smile})]$	$(NTPP,PONYA1tH^{\smile},PONXB1H^{\smile})$
27	$[(TPPA,TPPA^{\smile}, PONYA2),(TPPA, PONYA2,PONXB1H^{\smile})]$	$(TPPA, PONYA2, PONXB1H^{\smile})$
28	$[(TPPA,TPPA^{\smile}, PONYA2),(TPPA,PONYA2^{\smile},PONXB1H^{\smile})]$	$(TPPA,PONYA12^{\smile},PONXB1H^{\smile})$
29	$[(TPPA,TPPA^{\smile}, PONYA2),(TPPB, PONYA2,PONXB1H^{\smile})]$	$(TPPB, PONYA2,PONXB1H^{\smile})$
30	$[(TPPA,TPPA^{\smile}, PONYA2),(TPPB,PONYA2H^{\smile},PONXB1H^{\smile})]$	$(TPPB,PONYA2^{\smile},PONXB1H^{\smile})$
31	$[(TPPA,TPPA^{\smile}, PONYA2),(NTPP, PONYA2,PONXB1H^{\smile})]$	$(NTPP, PONYA2,PONXB1H^{\smile})$
32	$[(TPPA,TPPA^{\smile}, PONYA2),(NTPP,PONYA2^{\smile},PONXB1H^{\smile})]$	$(NTPP,PONYA2^{\smile},PONXB1H^{\smile})$
33	$[(TPPA,NTPP^{\smile}, PONXB1H^{\smile}),(TPPA, PONXB1H,PONXB1H^{\smile})]$	$(TPPA, PONXB1H, PONXB1H^{\smile})$
34	$[(TPPA, NTPP^{\smile}, PONXB1H^{\smile}),(TPPB, PONXB1H, PONXB1H^{\smile})]$	$(TPPB, PONXB1H, PONXB1H^{\smile})$
35	$[(TPPA, NTPP^{\smile}, PONXB1H^{\smile}),(NTPP, PONXB1H, PONXB1H^{\smile})]$	$(NTPP, PONXB1H, PONXB1H^{\smile})$
36	$[(TPPA, NTPP^{\smile}, PONXB2),(TPPA, PONXB2,PONXB1H^{\smile})]$	$(TPPA, PONXB2, PONXB1H^{\smile})$
37	$[(TPPA, NTPP^{\smile}, PONXB2),(TPPB, PONXB2, PONXB1H^{\smile})]$	$(TPPB, PONXB2, PONXB1H^{\smile})$
38	$[(TPPA, NTPP^{\smile}, PONXB2),(NTPP, PONXB2,PONXB1H^{\smile})]$	$(NTPP, PONXB2,PONXB1H^{\smile})$
39	$[(TPPA, NTPP^{\smile}, PONYB),(TPPA,PONYB^{\smile},PONXB1H^{\smile})]$	$(TPPA,PONYB^{\smile},PONXB1H^{\smile})$
40	$[(TPPA, NTPP^{\smile}, PONYB),(TPPA,PONYB^{\smile},PONXB1H^{\smile})]$	$(TPPB,PONYB^{\smile},PONXB1H^{\smile})$
41	$[(TPPA, NTPP^{\smile}, PONYB),(NTPP,PONYB^{\smile},PONXB1H^{\smile})]$	$(NTPP,PONYB^{\smile},PONXB1H^{\smile})$
42	$[(TPPA, NTPP^{\smile}, PONYB^{\smile}),(TPPA, PONYB,PONXB1H^{\smile})]$	$(TPPA, PONYB,PONXB1H^{\smile})$
43	$[(TPPA,TPPA,NTPP),(NTPP, PONXA2H,PONYA2^{\smile})]$	$(NTPP, PONXA2H, PONYA2^{\smile})$
44	$[(TPPA, NTPP^{\smile}, PONYB^{\smile}),(TPPB, PONYB, PONXB1H^{\smile})]$	$(TPPB, PONYB, PONXB1H^{\smile})$

Table B.4: Triggered pairs for the RCC29 algebra

No.	Triggered Pair	Removed Triple
1	$[(TPPA,TPPB1,TPPB2),(TPPA,TPPA^{\smile},ECNA)]$	$(TPPA,TPPA^{\smile},ECNA)$
2	$[(TPPA,TPPB2,TPPB1),(TPPA,TPPB2^{\smile},\mathrm{DC})]$	$(TPPA,TPPB2,TPPB1)$
3	$[(TPPA,TPPB2,TPPB2),(TPPA,TPPA^{\smile},\mathrm{ECNA})]$	$(TPPA,TPPA^{\smile},ECNA)$
4	$[(TPPA,TPPB1,TPPB1),(TPPB1,TPPB1,TPPB2)]$	$(TPPB1,TPPB1,TPPB2)$
5	$[(TPPA,TPPB1,TPPB1),(TPPB1,TPPB2,TPPB2)]$	$(TPPB1,TPPB2,TPPB2)$
6	$[(TPPA,TPPB2^{\smile},PONXA1),(TPPA,PONXA1,ECNA)]$	$(TPPA,PONXA1,ECNA)$
7	$[(TPPA,TPPB2^{\smile},PONXA2),(TPPA,PONXA2,ECNA)]$	$(TPPA,TPPB2^{\smile},PONXA2)$
8	$[(TPPA,TPPB2^{\smile},PONZ),(NTPP,PONZ,ECNA)]$	$(TPPA,TPPB2^{\smile},PONZ)$
9	$[(TPPA,TPPB2^{\smile},PONYA1),(TPPA,PONYA1,ECNA)]$	$(TPPA,TPPB2^{\smile},PONYA1)$
10	$[(TPPA,TPPB2^{\smile},PONYA1^{\smile}),(NTPP,PONYA1^{\smile},ECNA)]$	$(TPPA,TPPB2^{\smile},PONYA1^{\smile})$
11	$[(TPPA,TPPB2,TPPA),(TPPB1,TPPB1,TPPA)]$	$(TPPB1,TPPB1,TPPA)$
12	$[(TPPA,TPPA,NTPP),(TPPB1,TPPB1,NTPP)]$	$(TPPB1,TPPB1,NTPP)$
13	$[(TPPA,TPPB1,TPPA),(TPPB1,TPPB2,NTPP)]$	$(TPPB1,TPPB2,NTPP)$
14	$[(TPPA,TPPB1,TPPB1),(TPPB1,PONXA1,TPPB2)]$	$(TPPB1,PONXA1,TPPB2)$
15	$[(TPPA,TPPB1,TPPB1),(TPPB1,PONXB1,TPPB2)]$	$(TPPB1,PONXB1,TPPB2)$
16	$[(TPPA,TPPB1,TPPB1),(TPPB1,PONZ,TPPB2)]$	$(TPPB1,PONZ,TPPB2)$
17	$[(TPPA,TPPB1,TPPB1),(TPPB1,PONYA1,TPPB2)]$	$(TPPB1,PONYA1,TPPB2)$
18	$[(TPPA,TPPB1,TPPB1),(TPPB2,PONYA1,TPPB1)]$	$(TPPB2,PONYA1,TPPB1)$
19	$[(TPPA,TPPB2,TPPA),(TPPB2,TPPB1,TPPA)]$	$(TPPB2,TPPB1,TPPA)$
20	$[(TPPB2,TPPA,TPPB2),(TPPA,NTPP^{\smile},H)]$	$(TPPA,NTPP^{\smile},H)$
21	$[(TPPB1,TPPA,TPPA),(TPPB1,NTPP^{\smile},H)]$	$(TPPB1,NTPP^{\smile},H)$
22	$[(TPPB2,TPPA,TPPA),(TPPB2,PODZ,TPPB2)]$	$(TPPB2,PODZ,TPPB2)$
23	$[(TPPA,TPPA^{\smile},PONXA1),(TPPB1,PONXA1,NTPP)]$	$(TPPB1,PONXA1,NTPP)$
24	$[(TPPA,TPPA^{\smile},PONXA1),(TPPB1,PONXA1,PONXB1)]$	$(TPPB1,PONXA1,PONXB1)$
25	$[(TPPA,TPPA^{\smile},PONXA1),(TPPB1,PONXA1,PONXB2)]$	$(TPPB1,PONXA1,PONXB2)$
26	$[(TPPA,TPPA^{\smile},PONXA1),(TPPB1,PONXA1,PONYB)]$	$(TPPB1,PONXA1,PONYB)$
27	$[(TPPA,TPPA^{\smile},PONXA1),(TPPB1,PONXA1,PONYB^{\smile})]$	$(TPPB1,PONXA1,PONYB^{\smile})$
28	$[(TPPA,TPPA^{\smile},PONXA1),(TPPB1,PONXA1,H^{\smile})]$	$(TPPB1,PONXA1,H^{\smile})$
29	$[(TPPA,TPPA^{\smile},PONXA2),(TPPB1,PONXA2,NTPP)]$	$(TPPB1,PONXA2,NTPP)$
30	$[(TPPA,TPPA^{\smile},PONXA2),(TPPB1,PONXA2,PONXB1)]$	$(TPPB1,PONXA2,PONXB1)$
31	$[(TPPA,TPPA^{\smile},PONXA2),(TPPB1,PONXA2,PONXB2)]$	$(TPPB1,PONXA2,PONXB2)$
32	$[(TPPA,TPPA^{\smile},PONXA2),(TPPB1,PONXA2,PONYB)]$	$(TPPB1,PONXA2,PONYB)$
33	$[(TPPA,TPPA^{\smile},PONXA2),(TPPB1,PONXA2,PONYB^{\smile})]$	$(TPPB1,PONXA2,PONYB^{\smile})$
34	$[(TPPA,TPPA^{\smile},PONXA2),(TPPB1,PONXA2,H^{\smile})]$	$(TPPB1,PONXA2,H^{\smile})$
35	$[(TPPA,TPPA^{\smile},PONXA2),(TPPB2,PODZ,PONXA2)]$	$(TPPB2,PODZ,PONXA2)$
36	$[(TPPA,TPPA^{\smile},PONZ),(TPPB1,PONZ,NTPP)]$	$(TPPB1,PONZ,NTPP)$
37	$[(TPPA,TPPA^{\smile},PONZ),(TPPB1,PONZ,PONXB1)]$	$(TPPB1,PONZ,PONXB1)$
38	$[(TPPA,TPPA^{\smile},PONZ),(TPPB1,PONZ,PONXB2)]$	$(TPPB1,PONZ,PONXB2)$
39	$[(TPPA,TPPA^{\smile},PONZ),(TPPB1,PONZ,PONYB)]$	$(TPPB1,PONZ,PONYB)$
40	$[(TPPA,TPPA^{\smile},PONZ),(TPPB1,PONZ,PONYB^{\smile})]$	$(TPPB1,PONZ,PONYB^{\smile})$
41	$[(TPPA,TPPA^{\smile},PONZ),(TPPB2,PODZ,PONZ)]$	$(TPPB2,PODZ,PONZ)$
42	$[(TPPA,TPPA^{\smile},PONYA1),(TPPB1,PONYA1^{\smile},NTPP)]$	$(TPPB1,PONYA1^{\smile},NTPP)$
43	$[(TPPA,TPPA^{\smile},PONYA1),(TPPB1,PONYA1,PONXB1)]$	$(TPPB1,PONYA1,PONXB1)$
44	$[(TPPA,TPPA^{\smile},PONYA1),(TPPB1,PONYA1,PONXB2)]$	$(TPPB1,PONYA1,PONXB2)$
45	$[(TPPA,TPPA^{\smile},PONYA1),(TPPB1,PONYA1,PONYB)]$	$(TPPB1,PONYA1,PONYB)$
46	$[(TPPA,TPPA^{\smile},PONYA1),(TPPB1,PONYA1,PONYB^{\smile})]$	$(TPPB1,PONYA1,PONYB^{\smile})$
47	$[(TPPA,TPPA^{\smile},PONYA1),(TPPB1,PONYA1^{\smile},PONXB1)]$	$(TPPB1,PONYA1^{\smile},PONXB1)$
48	$[(TPPA,TPPA^{\smile},PONYA1),(TPPB1,PONYA1^{\smile},PONXB2)]$	$(TPPB1,PONYA1^{\smile},PONXB2)$
49	$[(TPPA,TPPA^{\smile},PONYA1),(TPPB1,PONYA1^{\smile},PONYB^{\smile})]$	$(TPPB1,PONYA1^{\smile},PONYB^{\smile})$
50	$[(TPPA,TPPA^{\smile},PONYA1),(TPPB1,PONYA1,H^{\smile})]$	$(TPPB1,PONYA1,H^{\smile})$
51	$[(TPPA,TPPA^{\smile},PONYA1),(TPPB2,PODZ,PONYA1^{\smile})]$	$(TPPB2,PODZ,PONYA1^{\smile})$
52	$[(TPPA,TPPA^{\smile},PONYA2),(TPPB1,PONYA2,NTPP)]$	$(TPPB1,PONYA2,NTPP)$
53	$[(TPPA,TPPA^{\smile},PONYA2),(TPPB1,PONYA2^{\smile},NTPP)]$	$(TPPB1,PONYA2^{\smile},NTPP)$
54	$[(TPPA,TPPA^{\smile},PONYA2),(TPPB1,PONYA2,PONXB1)]$	$(TPPB1,PONYA2,PONXB1)$
55	$[(TPPA,TPPA^{\smile},PONYA2),(TPPB1,PONYA2,PONXB2)]$	$(TPPB1,PONYA2,PONXB2)$
56	$[(TPPA,TPPA^{\smile},PONYA2),(TPPB1,PONYA2,PONYB)]$	$(TPPB1,PONYA2,PONYB)$
57	$[(TPPA,TPPA^{\smile},PONYA2),(TPPB1,PONYA2,PONYB^{\smile})]$	$(TPPB1,PONYA2,PONYB^{\smile})$
58	$[(TPPA,TPPA^{\smile},PONYA2),(TPPB1,PONYA2^{\smile},PONXB1)]$	$(TPPB1,PONYA2^{\smile},PONXB1)$
59	$[(TPPA,TPPA^{\smile},PONYA2),(TPPB1,PONYA2^{\smile},PONXB2)]$	$(TPPB1,PONYA2^{\smile},PONXB2)$
60	$[(TPPA,TPPA^{\smile},PONYA2),(TPPB1,PONYA2^{\smile},PONYB)]$	$(TPPB1,PONYA2^{\smile},PONYB)$
61	$[(TPPB1,TPPB2,TPPA),(TPPA,PONXA2,PONYB)]$	$(TPPA,PONXA2,PONYB)$
62	$[(TPPA,TPPB1,TPPA),(TPPA,TPPB2,TPPA)]$	$------------$

Table B.5: Triggered pairs for the ECNB Splitting

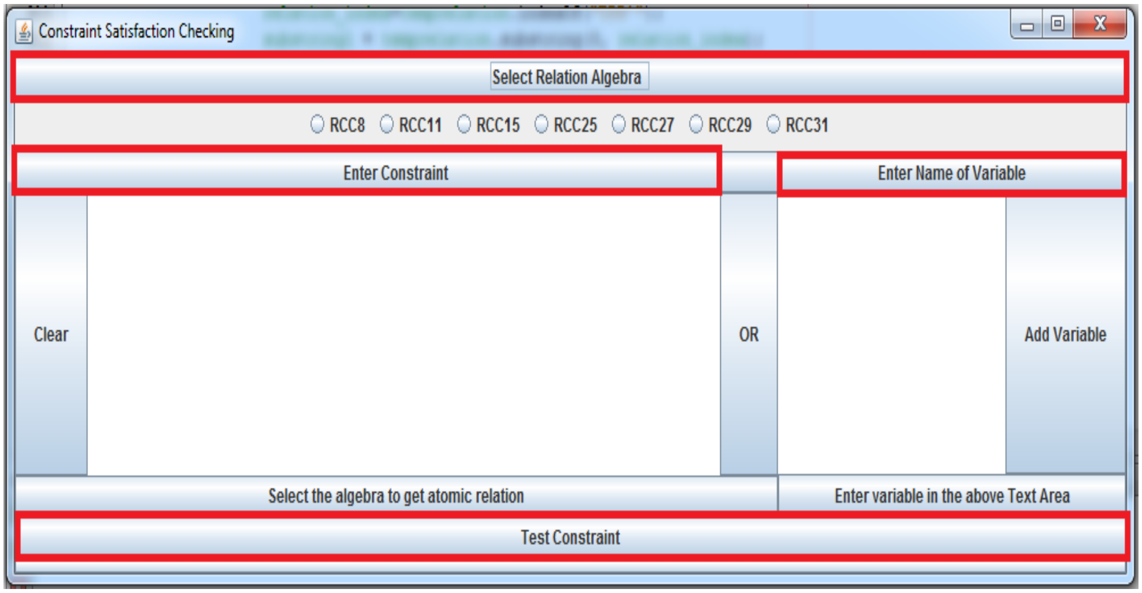


Figure B.1: Constraint satisfaction checking interface

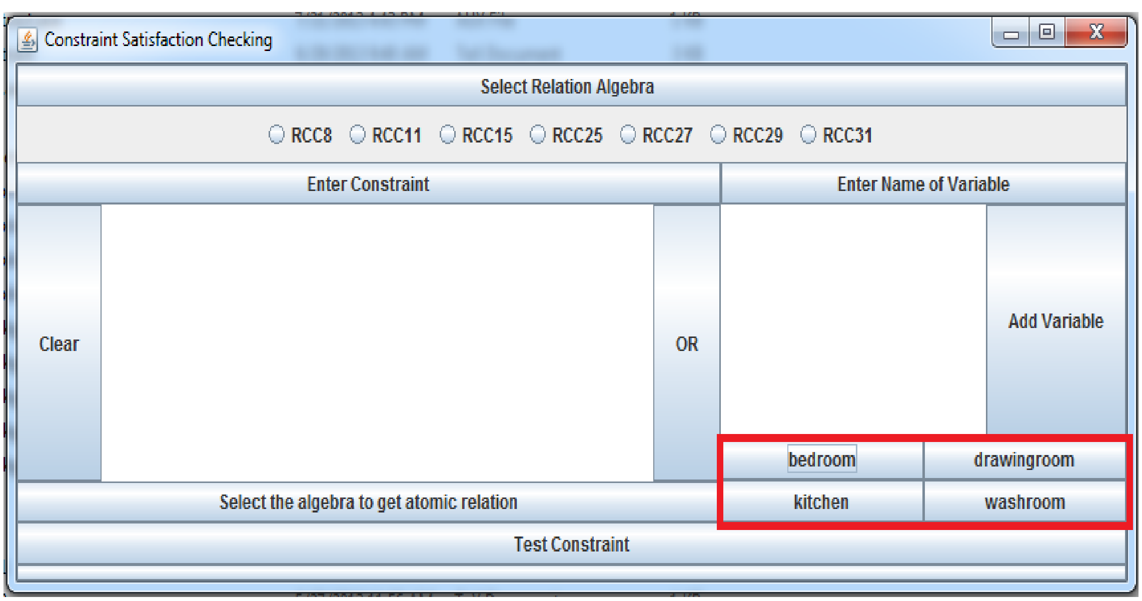


Figure B.2: Variables entered for constraint string

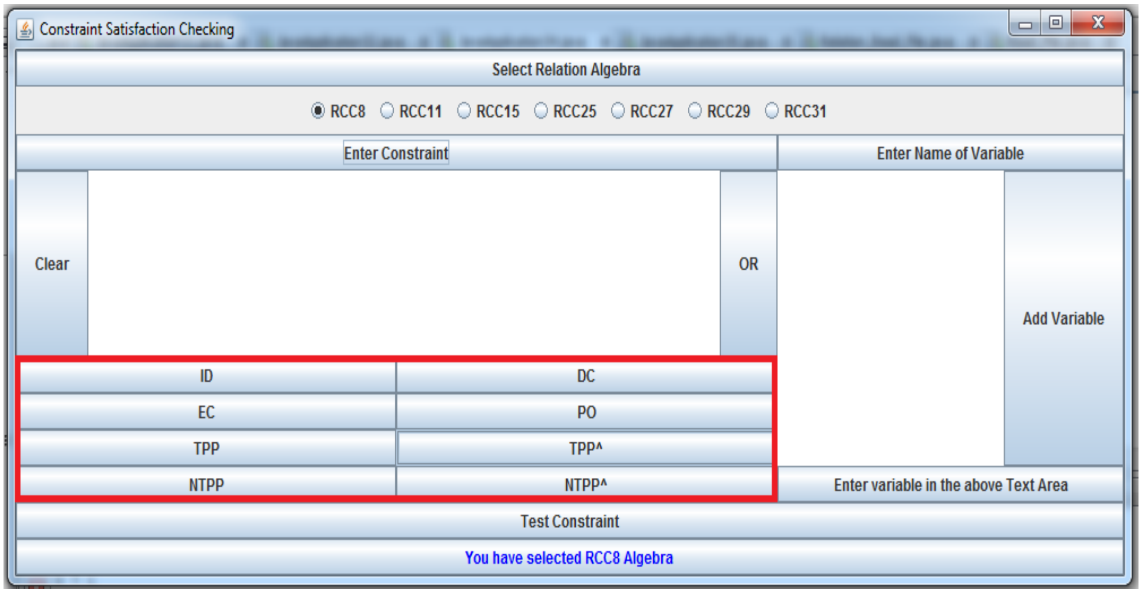


Figure B.3: Atomic relations for RCC8

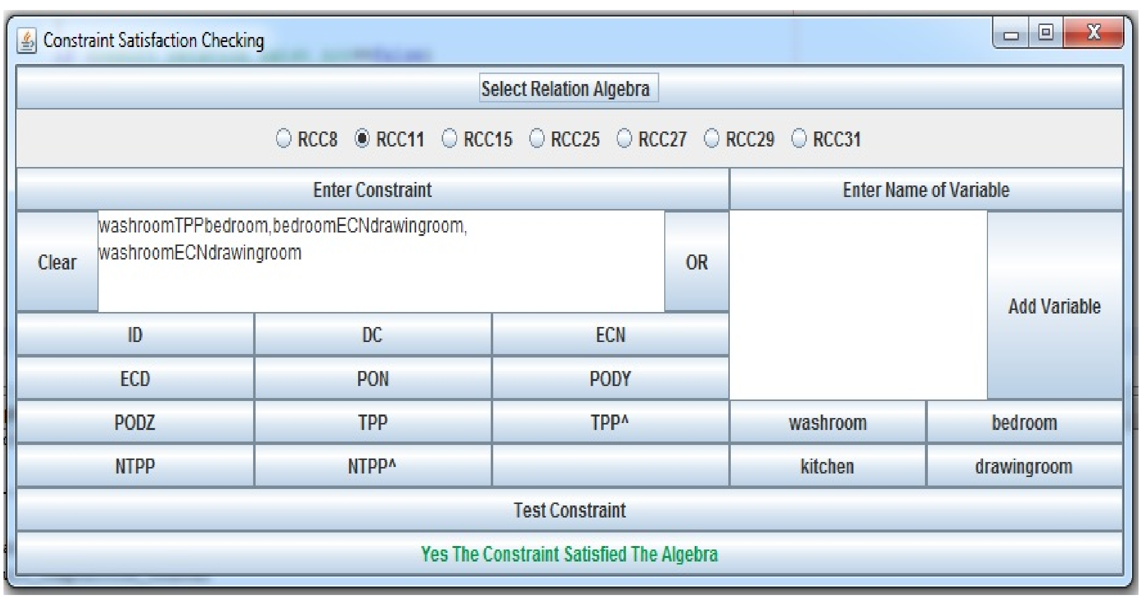


Figure B.4: Constraint satisfied based on the RCC11 algebra

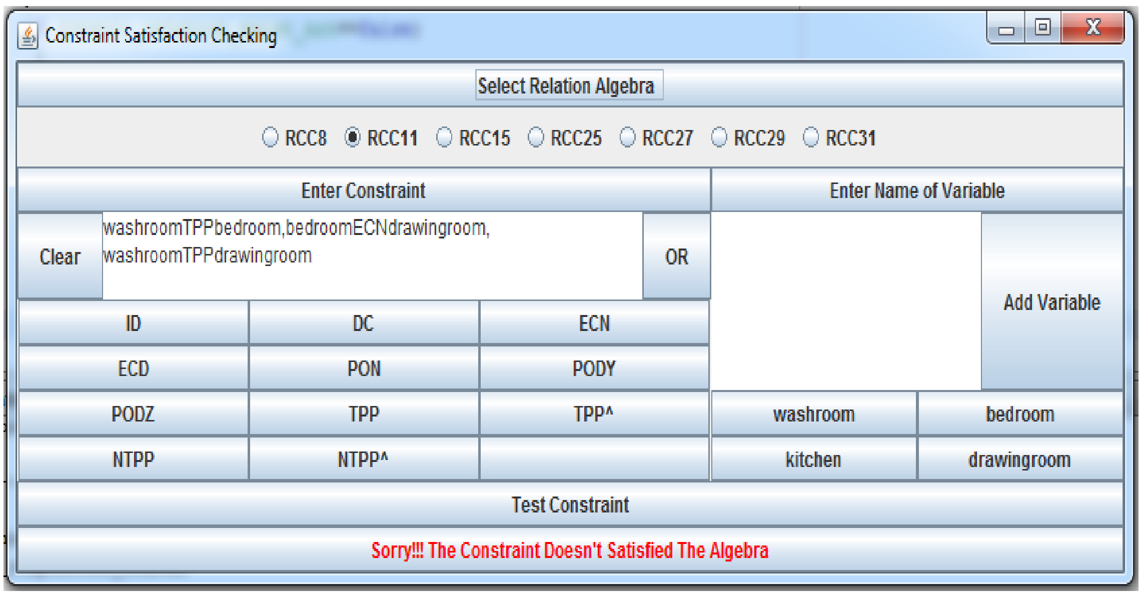


Figure B.5: Constraint not satisfied based on the RCC11 algebra

Bibliography

[1] Allen, J. F. : *Maintaining knowledge about temporal intervals.*, Communications of the ACM, 26(11), 832-843 (1983).

[2] Allen J.F., P.J. Hayes : *A common sense theory of time.*, Proceedings 9th IJCAI, Los Angeles, 528-531 (1985).

[3] Andréka, H. , Maddux, R. D. , and Nemeti, I. : *Splitting in relation algebras*, Proceedings of The American Mathematical Society 111, (1991).

[4] Asher, N. and Vieu, L. : *Toward a geometry of common sense: A semantics and a complete axiomatization of mereotopology*, In Mellish, C., editor, IJCAI 95, Proceedings of the 14th International Joint Conference on Artificial Intelligence, (1995).

[5] Bowman, C. L. : *A calculus of individuals based on 'connection'*, Notre Dame J. Formal Logic 22, pp. 204-218, (1981).

[6] Bennett, B. : *Logical Representations for Automated Reasoning about Spatial Relationships* , PhD thesis, University of Leeds, UK, (1998).

[7] Cohn, A. G., Bennett, B., Gooday, J. and Gotts, N. M. : *Representing and reasoning with qualitative spatial relations about regions*, In O. Stock (Ed.), Spatial and Temporal Reasoning, (Stock, O. , ed.), Kluwer, IRST, pp. 97-134, (1997).

[8] Cohn, A. G. : *Qualitative spatial representation and reasoning techniques. Research report*, School of Computer Studies, University of Leeds, (1997).

[9] Clarke, B. L. : *A calculus of individuals based on connection.*, Notre Dame Journal of Formal Logic 22, pp. 204-218, (1981).

[10] Cui, Z., Cohn, A.G., Randell, D.A. : *Qualitative and topological relation-ships in spatial databases D. Abel and B. C. Ooi (Eds.)*, Advances in Spatial Databases, Vol. 692 of Lecture Notes in Computer Science, Springer Verlag, Berlin, pp. 293-315, (1993).

[11] de Morgan, A. : *On the syllogism: IV, and on the logic of relations.*, Transactions of the Cambridge Philosophical Society 10, pp. 331-358, (1860).

[12] Düntsch, I. and Winter, M. : *A representation theorem for boolean contact algebras*, Theoretical Computer Science 347, pp. 498-512, (2003).

[13] Düntsch, I. and Winter, M. : *Weak contact structures*, Relational Methods in Computer Science, pp. 73-82, (2005).

[14] Düntsch, I. , Wang, H. , and Mccloskey, S. : *Relations algebras in qualitative spatial reasoning*, Fundamenta Informaticae, pp. 229-248, (2000).

[15] Düntsch, I. , Wang, H. , and Mccloskey, S. : *A relation - algebraic approach to the region connection calculus*, Theoretical Computer Science 255, pp. 63-83, (2001).

[16] Düntsch, I. : *Relation algebras and their application in temporal and spatial reasoning*, Artificial Intelligence Review 23, pp. 315-357, (2005).

[17] Düntsch, I. , Schmidt, G. , and Winter, M. : *A necessary relation algebra for mereotopology*, Studia Logica - An International Journal for Symbolic Logic 69, pp. 381-409, (2001).

[18] Düntsch, I. and Winter, M. : *The lattice of contact relations on a boolean algebra*, Relational Methods in Computer Science, pp. 99-109, (2008).

[19] Düntsch, I. and Winter, M. : *Algebraization and representation of mereotopological structures,*, Relational Methods in Computer Science, (2004).

[20] Düntsch, I. , Wang, H. , and Mccloskey, S. : *Relations algebras in qualitative spatial reasoning*, Fundamenta Informaticae 39, pp. 229-248, (1999).

[21] Dimov, G. and Vakarelov, D. : *Contact algebras and region-based theory of space*, Proximity approach - ii, Fundamenta Informaticae 74, pp. 251-282, (2006).

[22] De Laguna, T. : *Point, line and surface as sets of solids.*, The Journal of Philosophy 19, pp. 449-461, (1922).

[23] Egenhofer, M. and Franzosa, R. : *Point-set topological spatial relations*, International Journal of Geographic Information Systems 5, pp. 161-174, (1991).

[24] Egenhofer, M. and Sharma, J. : *Topological consistency*, In Fifth International Symposium on Spatial Data Handling, Charleston, SC. (1992).

[25] Egenhofer, M. : *Deriving the composition of binary topological relations*, Journal of Visual Languages and Computing 5, pp. 133-149, (1994).

[26] Egenhofer, M. and Sharma, J. : *Assessing the consistency of complete and incomplete topological information*, Geographical Systems 1, pp. 47-68, (1993).

[27] Egenhofer, M. J. : *Reasoning about binary topological relations*, Symposium on Large Spatial Databases, pp. 143-160, (1991).

[28] Egenhofer, M. J. and Sharma, J. : *Topological consistency*, University of South Carolina, Columbia, SC, (1992).

[29] Egenhofer, M.J. : *Reasoning about binary topological relations.*, Advances in Spatial Databases, Springer, New York, pp. 143-160, (1991).

[30] Henkin, L. , Monk, J. D. , and Tarski, A. : *Cylindric algebras*, Studies in logic and the foundations of mathematics, no. v. 2, North-Holland Pub. Co., (1985).

[31] Hayes P.J. : *Naive physics I: Ontology for liquids.*, J.R. Hobbs and B. Moore (Eds.) Formal Theories of the Commonsense World, Ablex, pp. 71-89, (1985).

[32] Hayes P.J. : *The second naive physics manifesto*, J.R. Hobbs and B. Moore (Eds.) Formal Theories of the Commonsense World, Ablex, pp. 1-46, (1985).

[33] http://hackage.haskell.org/platform/windows.html.

[34] http://sourceforge.net/projects/mingw/files/

[35] http://ftp.gnome.org/pub/GNOME/binaries/win32/dependencies/

[36] http://ftp.gnome.org/pub/GNOME/binaries/win32/glade3/3.6/glade3-3.6.6-with-GTK+.exe.

[37] Jochen Renz, Gérard Ligozat : *Weak Composition for Qualitative Spatial and Temporal Reasoning*, Principles and Practice of Constraint Programming, Volume 3709, pp. 534-548, (2005).

[38] Jonsson, B. and Tarski, A. : *Boolean algebras with operators part ii, Amer*, Journal of Mathematics 73, pp. 127-162, (1952).

[39] Maddux, R. D. : *Some varieties containing relation algebras*, Transactions of The American Mathematical Society 272, pp. 501-501, (1982).

[40] Maddux, R. D. : *Finite integral relation algebras*, Proceedings of a Conference held at Charleston 1149, pp. 175-197, (1985).

[41] Maddux, R. D. : *Relation algebras: Studies in logic and the foundations of mathematics*, vol. 150, Elsevier Science, (2006).

[42] Mormann, T. : *Holes in the region connection calculus*, Preprint, Presented at RelMiCS 6, Oisterwijk ,(2001).

[43] Montanari, U. : *Networks of constraints :Fundamental properties and applications to picture processing.*, Information Sciences 7, pp. 95-132,(1974).

[44] Németi, I. : *Algebraizations of quantifier logics*, Studia Logica, 50. :485-569 (1991). Updated version 12.1 (January 1997) is available from ftp.math-inst.hu/pub/ algebraic-logic/survey.ps.

[45] Pratt, I. and Schoop, D. : *Expressivity in polygonal, plane mereotopology*, Journal of Symbolic Logic, 65(2), pp. 822-838. (2000).

[46] P.J. Hayes: *The naive physics manifesto*, D. Mitchie (Ed.) Expert systems in the micro-electronic age, Edinburgh University Press, (1979).

[47] Pratt, I. and Schoop, D. : *A complete axiom system for polygonal mereotopology of the real plane.*, Journal of Philosophical Logic, 27(6), pp. 621-658, (1998).

[48] Randell, D. A. and Cohn, A. G. : *Modelling topological and metrical properties in physical processes*, Principles of Knowledge Representation and Reasoning, pp. 357-368, (1989).

[49] Randell, D. A. , Cui, Z. , and Cohn, A. G. : *A spatial logic based on regions and connection*, Principles of Knowledge Representation and Reasoning, pp. 165-176, (1992).

[50] Siddavaatam, P.: *Generating Relation Algebras for Qualitative Spatial Reasoning*, Maste's Thesis, Department of Computer Science, Brock University, St. Catharines, Ontario, (2011).

[51] Siddavaatam, P. , and Winter, M.: *Splitting atoms in relation algebras*, Proceedings of the 12th international conference on Relational and algebraic methods. :331-346 (2011).

[52] Tarski, A. and Givant, S. : *A formalization of set theory without variables*, American Mathematical Society, (1996).

[53] Tarski, A. : *On the calculus of relations*, Journal of Symbolic Logic 6, pp. 73-89, (1941).

[54] Trudel, André : *Interval Algebra Networks with Infinite Intervals*, 16th International Symposium on Temporal Representation and Reasoning, pp. 141-146, (2009).

[55] Vakarelov, D. , Dimov, G. , Düntsch, I. , and Bennett, B : *A proximity approach to some region-based theories of space*, Journal of Applied Non-classical Logics 12, pp. 527-559, (2002).

[56] Varzi, A. C. : *Parts, wholes, and part-whole relations*, The prospect of mereotopology Data Knowledge Engineering, 20, pp. 259-286, (1996).

[57] Winter, M. : *Relation algebras are matrix algebras over a suitable basis*, Tech. Report 1998-05, UBwM, (1998).

[58] Whitehead, A.N. : *Process and reality.*, New York: MacMillan. (1929).

[59] Yongming, Li. , Sanjiang, Li. and Mingsheng, Ying : *Relational reasoning in the Region Connection Calculus.*, eprint arXiv:cs/0505041. (2005).