On Redundant Topological Constraints (Extended Abstract)*

Sanjiang Li\textsuperscript{a,b}, Zhiguo Long\textsuperscript{a}, Weiming Liu\textsuperscript{a}, Matt Duckham\textsuperscript{c}, Alan Both\textsuperscript{c}
\textsuperscript{a} Centre for Quantum Software & Information, FEIT, University of Technology Sydney, Australia
\textsuperscript{b} UTS-AMSS Joint Research Laboratory, AMSS, Chinese Academy of Sciences, China
\textsuperscript{c} School of Science, RMIT University, Australia
sanjiang.li@uts.edu.au

Abstract

Redundancy checking is an important task in AI subfields such as knowledge representation and constraint solving. This paper considers redundant topological constraints, defined in the region connection calculus RCC8. We say a constraint in a set $\Gamma$ of RCC8 constraints is redundant if it is entailed by the rest of $\Gamma$. A prime subnetwork of $\Gamma$ is a subset of $\Gamma$ which contains no redundant constraints and has the same solution set as $\Gamma$. It is natural to ask how to compute such a prime subnetwork, and when it is unique. While this problem is in general intractable, we show that, if $\mathcal{S}$ is a subalgebra of RCC8 in which weak composition distributes over nonempty intersections, then $\Gamma$ has a unique prime subnetwork, which can be obtained in cubic time by removing all redundant constraints simultaneously from $\Gamma$. As a by-product, we show that any path-consistent network over such a distributive subalgebra is minimal.

1 Introduction

Qualitative spatial reasoning (QSR) is a common subfield of artificial intelligence and geographical information science, and has applications in GIS, cognitive robotics, high-level understanding of video data etc. The region connection calculus (RCC) [Randell et al., 1992] is perhaps the most well-known calculus for representing qualitative spatial information. Based on a binary connectedness relation, it defines a class of binary topological relations between regions in a connected topological space (e.g., the real plane). The RCC is an expressive formalism for representing topological information, and the computational complexity of reasoning with RCC has been investigated in depth in the literature. Most of these works focus on the consistency or satisfiability of RCC constraint networks.

In this paper, we consider the important problem of redundant RCC constraints. Given a set $\Gamma$ of RCC constraints, we say a constraint $(xRy)$ in $\Gamma$ is redundant if it can be entailed by the rest of $\Gamma$, i.e., removing $(xRy)$ from $\Gamma$ will not change the solution set of $\Gamma$. It is natural to ask when a network is redundant and how to get an irredudant subset without changing the solution set. We call a subset of $\Gamma$ a prime subnetwork of $\Gamma$ if it contains no redundant constraints but has the same solution set as $\Gamma$.

We show that it is in general co-NP hard to determine if a constraint is redundant in a network of RCC constraints, but if $\Gamma$ is over a tractable subclass, then a prime subnetwork can be found in $O(n^3)$ time. If in addition weak composition distributes over non-empty intersections of relations in $\mathcal{S}$, then $\Gamma$ has a unique prime subnetwork, which is obtained by removing all redundant constraints from $\Gamma$.

As in the case of propositional logic formulas [Liben-Nowell, 2005], redundancy of RCC constraints “often leads to unnecessary computation, wasted storage, and may obscure the structure of the problem” [Belov et al., 2012]. Finding a prime subnetwork can be useful in at least the following aspects: a) computing and storing the relationships between spatial objects and hence saving space for storage and communication; b) facilitating comparison between different constraint networks; c) unveiling the essential graphical structure of a network; and d) adjusting geometrical objects to meet topological constraints [Wallgrün, 2012].

1.1 Motivational Example: Placename Footprints

To motivate our discussion, we focus briefly on one specific application to illustrate how the use of prime subnetworks can save space for storage. Figure 1 gives a small example of a set of spatial regions formed by the geographic “footprints” associated with placenames in the Southampton area of the UK. The footprints are derived from crowd-sourced data, formed from the convex hull of the sets of coordinate locations at which individuals used the placenames on social media. Using such data sets in natural language placename searches frequently requires queries over the topological relationships between footprints (e.g., “is Clarence Pier in Southampton?”). Computing such relationships on-the-fly requires computationally intensive and slow geometric operations; by contrast Web-search queries demand rapid responses.

One potential solution is to cache the topological relations between all footprints of interest. However, even the small example in Figure 1, the 84 footprints then require $84 \times 83/2 = 3486$ stored relations. The moderate-sized footprint data set from which Figure 1 is adapted contains

*This is an extended abstract of the same titled article published in Artificial Intelligence, 225: 51-76 (2015).
a total of 3443 footprints leads to a constraint network with 5,925, 403 relations. It is easy to see that as crowd-sourced data sources continue to grow, the volumes of such data is set to explode. In the case of footprints, many of the relationships can be inferred, and computing the prime subnetwork can reduce the number of stored relationships to be approximately linear in the number of footprints. In the case of the Southampton constraint network, 1324 redundant relations lead to a prime subnetwork with only 2162 relations needing to be stored. For the full data set, 5, 604, 200 redundant relations lead to a prime subnetwork of just 321, 203 relations (in contrast to the full network of almost 6 million relations).

In Section 2 we recall the RCC constraint language and then discuss the redundancy and prime subnetwork problem in general in Section 3. We present our major results and an efficient algorithm in Section 4, and present the empirical evaluation in Section 5. The last section concludes the paper.

## 2 RCC Constraint Language

The RCC was introduced in [Randell et al., 1992]. Let \( U \) be the set of nonempty proper closed sets of \( \mathbb{R}^2 \). We call each element in \( U \) a region. For two regions \( a, b \), we say \( a \) is a part of \( b \), written \( a \sqsubseteq b \); say \( a \) is connected to \( b \), written \( a \sqsubset b \), if \( a \cap b \neq \emptyset \). Using \( C \) and \( P \), we define

\[
\begin{align*}
ex \, P \, y &\equiv x \lor \neg(y \land x) \\
ex \, O \, y &\equiv (\exists z)(z \land x \land z \land y) \\
ex \, D \, R \, y &\equiv \neg(x \land y) \\
ex \, P \, O \, y &\equiv x \land \neg(y \land x) \\
ex \, E \, Q \, y &\equiv x \land y \\
ex \, D \, C \, y &\equiv \neg(x \land y) \\
ex \, E \, C \, y &\equiv x \land \neg(y \land x) \\
ex \, T \, P \, P \, y &\equiv x \land y \\
ex \, N \, T \, P \, P \, y &\equiv x \land y \\
\end{align*}
\]

Write \( PP^{-1} \), \( TPP^{-1} \) and \( N^{-1} \) for the converses of \( PP \), \( TPP \) and \( NTPP \), respectively. Then \( B_5 = \{DR, PO, EQ, PP, PP^{-1}\} \)

\[
B_8 = \{DC, EC, PO, EQ, TPP, NTPP, TPP^{-1}, N^{-1}\}
\]

are two jointly exhaustive and pairwise disjoint (JEPD) sets of relations, i.e., for any two regions \( a, b \in U \), \( a \) is related by exactly one relation in \( B_i \) (\( i = 5, 8 \)). Figure 2 illustrates these basic RCC5/8 relations. We call the Boolean algebra generated by relations in \( B_i \) the RCCI algebra, which consists all relations that are unions of the basic relations in \( B_i \). For convenience, we denote a non-basic RCCI relation \( R \) as the subset of \( B_i \) it contains. For example, we write \( \{DR, PO, PP\} \) for the relation \( DR \cup PO \cup PP \), and write \( \ast_5 \) and \( \ast_8 \) for the corresponding universal relation in RCC5 and RCC8.

The composition of two basic RCC5/8 relations is not necessarily a relation in RCC5/8. For two RCC5/8 relations \( R \) and \( S \), we call the smallest relation in RCC5/8 that contains \( R \circ S \) the weak composition of \( R \) and \( S \), written \( R \circ S \) [Düntsch et al., 2001; Li and Ying, 2003].

### 2.1 RCC5/8 Constraint Network

An RCC5/8 constraint has the form \((x R y)_t\), where \( x, y \) are variables taking values from \( U \), the set of regions, \( R \) is an RCC5/8 relation (not necessarily basic). Given a set \( \Gamma \) of RCC5/8 constraints over variables \( V = \{v_1, v_2, \ldots, v_n\} \), we say \( \Gamma \) is consistent or satisfiable if there is an assignment \( \sigma \): \( V \rightarrow U \) such that \((\sigma(v_i), \sigma(v_j)) \) satisfies the constraint in \( \Gamma \) that relates \( v_i \) to \( v_j \).

Without loss of generality, we assume \( \Gamma \) has the form \((x_i R_{ij} x_j)_{t_{ij} = 1}\), where, for any \( 1 \leq i, j \leq n \), there is a unique constraint \( R_{ij} \), and \( R_{ii} = EQ \). In this sense, we call \( \Gamma \) a constraint network. Let \( \Gamma = \{x_i R_{ij} x_j\}_{i, j = 1}^n \) and \( \Gamma' = \{x_i R_{ij} x_j\}_{i, j = 1}^n \) be two RCC5/8 constraint networks. We say \( \Gamma \) and \( \Gamma' \) are equivalent if they have the same set of solutions; and say \( \Gamma \) refines \( \Gamma' \) if \( R_{ij} \sqsubseteq R'_{ij} \) for all \( (i, j) \). We say an RCC5/8 network \( \Gamma \) is a basic network if each constraint is either the universal relation or a basic relation; and say a basic network complete if there are no universal relations.

Suppose \( S \) is a subset of RCC5/8. We say an RCC5/8 network \( \Gamma = \{v_i R_{ij} v_j\} \) is over \( S \) if \( R_{ij} \in S \) for every pair of variables \( v_i, v_j \). The consistency problem over \( S \), written as \( CSP(S) \), is the decision problem of the consistency of an arbitrary constraint network over \( S \). It is well known that the consistency problem over RCC5/8, i.e., \( CSP(RCC5/8) \), is NP-complete and RCC8 has three maximal tractable subclasses that contain all basic relations [Renz, 1999] and RCC5 has only one [Jonsson and Drakengren, 1997].

We say a network \( \Gamma = \{v_i R_{ij} v_j\} \) path-consistent if for every \( 1 \leq i, j, k \leq n \), we have \( R_{ik} \sqsubseteq R_{ik} \circ R_{kj} \). A cubic

Figure 1: Examples of crowd-sourced geographic placename “footprints” around Southampton, UK

Figure 2: RCC5/8 basic relations
algorithm, henceforth called the path-consistency algorithm or PCA, has been devised to enforce path-consistency. For any RCC5/8 network \( \Gamma \), the PCA either detects inconsistency of \( \Gamma \) or returns a path-consistent network, written \( \Gamma_p \), which is equivalent to \( \Gamma \) and known as the algebraic closure or \( a \)-closure of \( \Gamma \) [Ligozat and Renz, 2004]. It is easy to see that in this case \( \Gamma_p \) also refines \( \Gamma \), i.e., we have \( S_{ij} \leq R_{ij} \) for each constraint \( (x_i, S_{ij}, x_j) \) in \( \Gamma_p \).

**Proposition 1.** Let \( S \) be a tractable subclass of RCC5/8 which contains all basic relations. An RCC5/8 network \( \Gamma \) over \( S \) is consistent if applying PCA to \( \Gamma \) does not result in inconsistency.

This is particularly true for basic RCC5/8 networks.

### 2.2 Distributive Subalgebra

Write \( \hat{B}_S \) for the closure of \( B_S \) under converse, intersection, and weak composition in RCC5. Then \( \hat{B}_S \) contains the basic relations as well as:

\[
\{ PO, PP \}, \{ PO, PP^{-1} \}, \{ PO, PP, PP^{-1}, EQ \},
\{ DR, PO, PP \}, \{ DR, PO, PP^{-1} \}, \{ DR, PO \}, \ast_5,
\]

where \( \ast_5 = \{ DR, PO, PP, PP^{-1}, EQ \} \). It is interesting to note in that in \( \hat{B}_S \) the weak composition operation is distributive over nonempty intersections in the following sense.

**Lemma 2.** Let \( R, S, T \) be three relations in \( \hat{B}_S \). Suppose \( S \cap T \) is nonempty. Then we have:

\[
R \circ (S \cap T) = R \circ S \cap R \circ T \quad (1)
\]

\[
(S \cap T) \circ R = S \circ R \cap T \circ R. \quad (2)
\]

In general, we have the following definition.

**Definition 1.** Let \( S \) be a subclass of RCC5/8. We say \( S \) is a subalgebra if \( S \) contains all basic relations, and is closed under converse, weak composition, and intersection. We say a subalgebra \( S \) is distributive if weak composition distributes over nonempty intersections of relations in \( S \).

Clearly, every distributive subalgebra of RCC5 contains \( \hat{B}_S \). The following lemma shows that relations in a distributive subalgebra have the Helly property.

**Lemma 3.** Let \( S \) be a distributive subalgebra of RCC5/8. Suppose \( R, S, T \) are three relations in \( S \). Then \( R \cap S \cap T = \emptyset \) iff \( R \cap S = \emptyset \) or \( R \cap T = \emptyset \), or \( S \cap T = \emptyset \).

Actually, the inverse of the above result is also true (see [Long and Li, 2015]). We say a distributive subalgebra \( S \) is maximal if there is no other distributive subalgebra that properly contains \( S \). To find all maximal distributive subalgebras, we start with \( \hat{B}_S \) and then try to add other relations to this subalgebra to get larger distributive subalgebras. It turns out that RCC5 (RCC8, resp.) has only two maximal distributive subalgebras, which are all contained in \( H_8 \), the maximal tractable subclass of RCC5 identified in [Renz and Nebel, 1997; Jonsson and Drakengren, 1997] \( (H_8, \) one of the three maximal subclasses of RCC8 identified in [Renz, 1999], resp.).

### 3 Prime Subnetwork

Given an RCC5/8 network \( \Gamma \), a very interesting question is, how to find a prime subnetwork of \( \Gamma \)? This problem is clearly at least as hard as determining if \( \Gamma \) has a redundant constraint. Similar to the case of propositional logic formulas [Libera-tore, 2005], we have the following result for RCC5/8.

**Proposition 4.** Let \( \Gamma \) be an RCC5/8 network and suppose \( (xRy) \) is a constraint in \( \Gamma \). It is co-NP-complete to decide if \( (xRy) \) is redundant in \( \Gamma \).

A naive method to obtain a prime subnetwork is to remove redundant constraints one by one from \( \Gamma \) until we get an irreducible network. Suppose we have an oracle which can tell if a constraint is redundant. Then in an additional \( O(n^3) \) time we can find an irreducible network that is equivalent to \( \Gamma \) by removing several constraints from \( \Gamma \).

Suppose \( S \) is a tractable subalgebra of RCC5/8. Then, for any network \( \Gamma \) over \( S \), we can determine whether a constraint is redundant in \( \Gamma \) in \( O(n^3) \) time and find all redundant constraints of \( \Gamma \) in \( O(n^5) \) time. A prime subnetwork for \( \Gamma \) can also be found in \( O(n^5) \) time.

Are prime subnetworks unique? In general this is not the case, but it is easy to see that the core of \( \Gamma \), i.e. the set of non-redundant constraints in \( \Gamma \), is contained in every prime subnetwork of \( \Gamma \). In the following we assume that \( \Gamma \) is an all-different constraint network, i.e. it has the following property:

\[
(\forall i,j)[(i \neq j) \rightarrow (\Gamma \neq \{ v_i \text{EQ} v_j \})]. \quad (3)
\]

This property ensures that no two variables have to be identical. In other words, there is no ‘redundant’ variables.

We next show that, if \( \Gamma \) is an all-different constraint network over a distributive subalgebra of RCC5/8, then \( \Gamma_c \) is the unique prime network of \( \Gamma \). This is quite surprising, as, in general, knowing that both \( (xRy) \) and \( (uSv) \) are redundant in \( \Gamma \) does not imply that \( (uSv) \) is redundant in \( \Gamma \setminus \{(xRy)\} \).

### 4 Networks over a Distributive Subalgebra

In this section, we assume \( S \) is a distributive subalgebra of RCC5/8. Let \( \Gamma \) be an all-different constraint network over \( S \). We show that \( \Gamma_c \) is equivalent to \( \Gamma \) and hence the unique prime network of \( \Gamma \).

**Definition 2** (cf. [Chandra and Pujari, 2005; Liu and Li, 2012]). Suppose \( \Theta = \{ v_i T_{ij} v_j \}_{i,j=1}^n \) is an RCC5/8 network. We say \( \Theta \) is minimal if for every pair of variables \( v_i, v_j \) \((i \neq j)\) and every basic relation \( \alpha \) in \( T_{ij} \), there exists a solution \( \{ a_1, a_2, \cdots, a_n \} \) of \( \Theta \) s.t. \( (a_i, a_j) \) is an instance of \( \alpha \).

Each consistent RCC5/8 constraint network has a unique minimal network, but it is in general NP-hard to compute it (see e.g. [Liu and Li, 2012]).

**Notation:** We write \( \Gamma_m \) for the minimal network of \( \Gamma \), \( \Gamma_p \) for the \( a \)-closure of \( \Gamma \), and \( \Gamma_c \) for the core of \( \Gamma \).

To prove that \( \Gamma_c \) is equivalent to \( \Gamma \), we need two important results. The first result, stated in Thm. 5, shows that \( \Gamma_m \) is exactly \( \Gamma_p \). The second result, stated in Prop. 6, shows that a particular constraint \( (xRy) \) is redundant in \( \Gamma \) iff its corresponding constraint in \( \Gamma_p \) is redundant. Our main result, stated in Thm. 7, then follows immediately.

**Theorem 5.** Suppose \( \Gamma \) is a consistent network over \( S \) and \( \Gamma_p \) its \( a \)-closure. Then \( \Gamma_p = \Gamma_m \), the minimal network of \( \Gamma \).
Proposition 6. Suppose $\Gamma$ is a consistent network over $S$ which satisfies (3). Assume that $(xRy)$ and $(xSy)$ are the constraints from $x$ to $y$ in $\Gamma$ and $\Gamma_p$ respectively. Then $(xRy)$ is redundant in $\Gamma$ iff $(xSy)$ is redundant in $\Gamma_p$.

Recall that Thm. 5 asserts that $\Gamma_p$ is minimal. As a consequence, we have our main result.

Theorem 7. Suppose $\Gamma$ is a consistent network over $S$ which satisfies (3) and $\Gamma_c$ the core of $\Gamma$. Then $\Gamma_c$ is equivalent to $\Gamma$ and hence the unique prime network of $\Gamma$.

In general, the core of an RCC5/8 network over a tractable subclass can be found in $O(n^3)$ time. This can be improved for networks over a distributive subalgebra. To show this, we need the following result for path-consistent networks.

Lemma 8. Suppose $\Gamma$ is a path-consistent network over $S$. Then a constraint $(v_iRijv_j)$ is redundant in $\Gamma$ iff $R_{ij} = \bigcap \{R_{ik} \cap R_{kj} : k \neq i, j\}$, i.e., $R_{ij}$ is the intersection of the weak compositions of all paths from $v_i$ to $v_j$ with length 2.

Algorithm 1: Algorithm for finding all redundant constraints, where $\pi_i$ is the universal relation in RCC.

Input: An all-different consistent RCC5/8 network $\Gamma = \{v_iR_{ij}v_j : 1 \leq i, j \leq n\}$ over $S$ and $V = \{v_i : 1 \leq i \leq n\}$.

Output: Redun: the set of redundant constraints of $\Gamma$, and Core: the core of $\Gamma$.

1. Redun $\leftarrow \emptyset$
2. Core $\leftarrow \Gamma$
3. $\Gamma_p \leftarrow$ the a-closure of $\Gamma$
4. for each constraint $(v_iS_{ij}v_j) \in \Gamma_p$ do
5. $Q_{ij} \leftarrow \pi_i$
6. for each variable $v_k \in V \setminus \{v_i, v_j\}$ do
7. $Q_{ij} \leftarrow Q_{ij} \cap S_{ik} \cap S_{kj}$
8. if $Q_{ij} = S_{ij}$ then
9. Redun $\leftarrow$ Redun $\cup \{(v_iR_{ij}v_j)\}$
10. Core $\leftarrow$ Core $\cup \{(v_iR_{ij}v_j)\}$
11. break the inner loop;

Suppose $\Gamma$ is an all-different consistent network over a distributive subalgebra of RCC5/8. Prop. 6 and Lemma 8 suggest a simple way for computing $\Gamma_c$, the unique prime network of $\Gamma$. By Prop. 6, a constraint $(v_iR_{ij}v_j)$ in $\Gamma$ is redundant iff the corresponding constraint $(v_iS_{ij}v_j)$ in $\Gamma_p$ is redundant. Furthermore, Lemma 8 shows that $(v_iS_{ij}v_j)$ is redundant in $\Gamma_p$ iff $S_{ij}$ is the intersection of all $S_{ik} \cap S_{kj}$ ($k \neq i, j$). We hereby have the cubic algorithm Algorithm 1 for finding all redundant constraints in $\Gamma_p$. For each constraint $(v_iS_{ij}v_j)$, to verify if $S_{ij} = \bigcap \{S_{ik} \cap S_{kj} : k \neq i, j\}$, we introduce a relation $P_{ij}$ which consists of all basic relations $\alpha$ that are not in $S_{ik} \cap S_{kj}$ for some $k \neq i, j$ and then check if $P_{ij} \cup S_{ij}$ is the universal relation.

5 Empirical evaluation

In this section, we empirically evaluate our method in comparison with the two greedy methods for removing redundant constraints proposed in [Wallgrön, 2012]: the basic and extended simplification algorithms (hereafter SIMPLE and SIMPLEX). SIMPLE loops through all triples of regions $i, j, k$ and $k$ and identifies as redundant any constraints $R_{ij}$ such that $R_{ij} \cap R_{jk} \subseteq R_{ik}$. A drawback of SIMPLE is that redundant relations removed may affect subsequent iterations of the algorithm. Hence, the order in which triples are visited by SIMPLE can alter the resulting subnetwork. SIMPLEX solves this issue by first marking potentially redundant relations for removal, subject to a consistency check, before removing all marked relations in a final loop. SIMPLE and SIMPLEX are not guaranteed to provide a prime subnetwork.

In the evaluation, two real data sets were used: the UK geographic “footprint” dataset introduced in Section 1.1 (total 3443 regions) and the statistical areas levels 1–4 dataset for Tasmania (in total 1559 regions), provided by the Australian Bureau of Statistics. Both datasets are complete basic constraint networks, i.e., there is a single basic relation between each pair of regions. Derived from social media, the footprint data set contains a variety of regions of differing sizes and shapes, but relatively unstructured sharing almost no adjacent boundaries. In stark contrast the Tasmanian statistical areas are highly structured, made up of four levels of spatially contiguous and nested but non-overlapping regions. To aid in our analysis, five subsets of each of the two datasets were generated in addition to the full datasets.

Our empirical analysis showed that for real geographic data sets the prime subnetwork can lead to significant increases in the number of redundant relations identified when compared with the approximations proposed by [Wallgrön, 2012]. In practice, the algorithm was efficient, exhibiting average case $O(n^2)$ scalability. The redundant relations identified by the prime subnetwork can also significantly outnumber DC relations, especially in less structured geographic data sets that may contain a significant minority of PO relations.

6 Conclusion and Further Development

In this paper, we have systematically investigated the computational complexity of redundancy checking for RCC5/8 constraints. Although it is in general intractable, we have shown that, if the constraints are taken from a distributive subalgebra, then the core of the constraint network is the unique prime network and can be found in cubic time.

Algorithm 1 can be significantly improved if we enforce partial path-consistency [Bliik and Sam-Haroud, 1999] instead of path-consistency. Indeed, it is shown in [Sioutis et al., 2015] that the thus revised redundancy removing algorithm significantly progresses the state-of-the-art for practical reasoning with very large real RCC8 networks.

Some of our results (e.g., all results before Thm. 5) can be immediately applied to several other qualitative calculi e.g. Interval Algebra [Allen, 1983], but Prop. 6 and Thm. 7 do use the particular algebraic properties of RCC5/8. However, the same results actually apply to the simple temporal problem (STP) [Dechter et al., 1991]. In [Lee et al., 2016], it was shown that every non-degenerated STP instance has a unique prime subnetwork and evaluation on a large benchmark dataset of STP exhibits a significant reduction in redundant information for the involved instances.
References


