Qualitative Constraint Satisfaction Problems: An Extended Framework with Landmarks ★

Sanjiang Li*a, Weiming Liu*a,∗, Shengsheng Wangb

aCentre for Quantum Computation & Intelligent Systems, Faculty of Engineering and Information Technology, University of Technology Sydney, Australia
bCollege of Computer Science and Technology, Jilin University, Changchun, China

Abstract

Dealing with spatial and temporal knowledge is an indispensable part of almost all aspects of human activity. The qualitative approach to spatial and temporal reasoning, known as Qualitative Spatial and Temporal Reasoning (QSTR), typically represents spatial/temporal knowledge in terms of qualitative relations (e.g., to the east of, after), and reasons with spatial/temporal knowledge by solving qualitative constraints.

When formulating qualitative constraint satisfaction problems (CSPs), it is usually assumed that each variable could be “here, there and everywhere”1. Practical applications such as urban planning, however, often require a variable to take its value from a certain finite domain, i.e. it is required to be ‘here or there, but not everywhere’. Entities in such a finite domain often act as reference objects and are called “landmarks” in this paper. The paper extends the classical framework of qualitative CSPs by allowing variables to take values from finite domains. The computational complexity of the consistency problem in this extended framework is examined for the five most important qualitative calculi, viz. Point Algebra, Interval Algebra, Cardinal Relation Algebra, RCC5, and RCC8. We show that all these consistency problems remain in NP and provide, under practical assumptions, efficient algorithms for solving basic constraints involving landmarks for all these calculi.

✩This is an extended and revised version of two conference papers [29, 27] presented at CP-2011 and ECAI-2012.

∗Corresponding Author

Email addresses: sanjiang.li@uts.edu.au (Sanjiang Li), weiming.liu@uts.edu.au (Weiming Liu), wss@jlu.edu.cn (Shengsheng Wang)

1A song by The Beatles from the album Revolver (1966).
Keywords: qualitative spatial and temporal reasoning; qualitative calculi; constraint satisfaction; landmarks

1. Introduction

Spatial and temporal information is pervasive and forms an increasing part of our everyday life. Many tasks in the real or virtual world require sophisticated spatial and temporal reasoning abilities. Furthermore, the rapid progress in science and technology in this century continues to present new challenges for spatial and temporal reasoning. Taking spatial information as an example, on one hand, people can now easily acquire location information with the help of GPS-enabled mobile equipment and web GISs such as Google Maps. This has greatly increased the public’s demand for location-based services. On the other hand, the development of technologies such as remote sensing, medical imaging, and sensor networks has generated volumes of spatial data, which makes the phenomenon of ‘rich data but poor knowledge’ particularly serious in the area of spatial knowledge management.

The qualitative approach to spatial and temporal reasoning, known as Qualitative Spatial and Temporal Reasoning (QSTR), has the potential to resolve the conflict between data and knowledge. This is because the main aims of QSTR research are to design (i) human comprehensible and cognitively plausible spatial relation models (or query languages), and (ii) efficient algorithms for consistency checking (or query preprocessing). For intelligent systems, the ability to understand and process qualitative, vague or even inconsistent (textual, graphical or speech) information collected from human beings or the Web is very important. This is because ‘the input and the output of spatial processes is often qualitative rather than quantitative’ [36].

QSTR represents spatial/temporal information in terms of human comprehensible qualitative relations (e.g. partially overlaps, west of, after) and reduces spatial/temporal reasoning to solving constraint satisfaction problems (CSPs). Qualitative relations are widely used in temporal and spatial reasoning (see e.g. [1, 38, 24]). This is partially because they are close to the way humans represent and reason about commonsense knowledge, are easy to specify, and provide a flexible way to deal with incomplete knowledge. Usually, these relations are taken from a qualitative calculus, which is a finite set of relations defined on an infinite universe \( U \) of entities [25]. Well-known qualitative calculi include among others Point Algebra (PA) [44], Interval Algebra (IA) [1], Cardinal Relation Algebra (CRA) [24], RCC5, and RCC8 [38, 24].
A central reasoning problem of QSTR is the **consistency problem**. An instance of the consistency problem is a set \( \Gamma \) of constraints like \((x \alpha y)\), where \( x, y \) are variables taken from a finite set \( V \), and \( \alpha \) is a qualitative relation. Unlike classical CSPs, the domain of a variable appearing in a qualitative constraint is usually infinite, and Hirsch [16] has shown that it may be undecidable for determining consistency for binary CSPs with infinite domains. However, for the five qualitative calculi that we have mentioned above, the consistency problems are all in NP and can be solved by using path consistency and backtracking (cf. [7, 41]).

In the past three decades, QSTR has made significant progress, and prominent qualitative calculi such as IA and RCC8 have been applied in areas such as natural language processing, geographical information systems, robotics, and content-based image retrieval (see e.g. [7]). There is a growing consensus, however, that breakthroughs are necessary to bring spatial/temporal reasoning theory closer to practical applications. One reason might be that the current qualitative reasoning scheme uses a rather restricted constraint language: constraints in a qualitative CSP are always taken from the *same* calculus and only relate variables from the same *infinite* domain. This is highly undesirable, as constraints involving restricted variables and/or multiple aspects of information frequently appear in practical tasks such as urban planning and spatial query processing.

Consider the following example. Suppose you are recommended a restaurant in Sydney by a friend who has dined there before. The spatial information about the restaurant may be similar to “it is in downtown and close to a MacDonald’s, and it is to the west of or southwest of Central Station.” In this example, topological, directional, and distance information appears together. While the position of the restaurant may be completely unknown, the position of Central Station is fixed as a landmark, and the position of downtown is also fixed somehow, but the position of “MacDonald’s” is only *finitely fixed* because there are several branches of MacDonald’s in downtown Sydney.

While some recent works have considered how to reason with qualitative constraints from different spatial or temporal calculi [13, 20, 28, 45, 21], the importance of solving constraints that involve restricted variables has been totally neglected. Cohn and Renz regarded this as a major future challenge, and commented in their chapter [7, page 578] in “Handbook of Knowledge Representation” that

One problem with this [constraint-based] approach is that spatial entities are treated as variables which have to be instantiated using values of an infinite domain. How to integrate this with settings where some spatial entities are known or can only be from a small domain is still
unknown and is one of the main future challenges of constraint-based spatial reasoning.

This paper aims to address the above challenge. We say that a variable is finitely restricted if it can only take its value from a finite subset of the universe in a qualitative calculus. We propose to extend the qualitative CSP framework by allowing variables to be finitely restricted. In such a qualitative CSP, the constraints are taken from a fixed qualitative calculus, and the domain of each variable is either the universe of the calculus or a (nonempty) finite subset of the universe. The entities in each finite domain usually act as reference spatial/temporal objects in the constraint network. In this paper, we address these entities as “landmarks”.

Landmarks (e.g. Sydney Opera House or Big Ben) are external, outstanding physical objects that act as reference objects. As found in many spatial discourses, landmarks play a fundamental role in cognitive spatial representations, in particular in human navigation and route planning. There are many works in geographical information science that are devoted to characterising or generating landmarks. Lynch [31] is perhaps the first such attempt, which although informal is very influential. Grabler et al. [14] developed a system to generate tourist maps enriched with landmarks. Duckham, Winter, and Robinson [11] considered how to incorporate cognitively salient landmarks in computer-generated navigation instructions. Landmarks are also used as a metaphor in automatic planning, where a landmark acts as an auxiliary sub-goal [15, 42].

In this paper, landmarks are used as reference objects for formulating constraints. This is related to but different from Allen’s ‘reference intervals’ [1], which are used to group clusters of intervals, and the intervals in one cluster are related to intervals outside the cluster only indirectly via the reference intervals.

An important research question is, how does this extension affect the computational complexity of deciding the consistency of qualitative CSPs? This paper examines the effect for the five most important qualitative calculi, viz. PA, IA, CRA, RCC5 and RCC8. We show that in the extended framework the consistency problem remains in NP for each calculus. Moreover, we propose practical efficient algorithms for solving basic constraints involving landmarks for these qualitative calculi.

The remainder of this paper proceeds as follows. Section 2 introduces basic notions in qualitative constraint solving as well as the five qualitative calculi discussed in this paper. The extended qualitative CSP framework is also presented there. Section 3 discusses the computational complexity of reasoning with the point-based calculi PA, IA, and CRA, and Section 4 considers the same prob-
lem for the region-based calculi RCC5 and RCC8. The last section concludes the paper and outlines problems for future study.

2. Preliminaries

In this section, we first recall several well-known qualitative calculi and basic notions in qualitative constraint solving, and then introduce the extended qualitative CSP framework.

2.1. Qualitative calculi

The qualitative approach to spatial and temporal knowledge representation and reasoning is mainly based on qualitative calculi. In this paper, we only consider binary relations, but the extended qualitative CSP framework can be straightforwardly extended to ternary and any \( n \)-ary relations.

Suppose \( U \) is the universe of spatial or temporal entities. Write \( \text{Rel}(U) \) for the Boolean algebra of binary relations on \( U \). A qualitative calculus on \( U \) is defined as a finite Boolean subalgebra of \( \text{Rel}(U) \). Let \( M \) be a qualitative calculus on \( U \). A relation \( \alpha \) in \( M \) is a basic relation if it is an atom in \( M \). We write \( B_M \) for the set of basic relations in \( M \).

We next recall the well-known Point Algebra (PA) [44, 43], Cardinal Relation Algebra (CRA) [12, 24], Interval Algebra (IA) [1], and RCC5 and RCC8 [38].

**Definition 1** (Point Algebra [44]). Let \( U \) be the set of real numbers. The Point Algebra is the Boolean subalgebra generated by the jointly exhaustive and pairwise disjoint (JEPD) set of relations \( \{<, >, =\} \), where \(<, >, = \) are defined as usual.

PA contains eight relations, viz. the three basic relations \(<, >, =\), the empty relation, and the four non-basic nonempty relations \(\leq, \geq, \neq, ?\), where ? stands for the universal relation.

**Definition 2** (Cardinal Relation Algebra [12, 24]). Let \( U \) be the real plane. Define binary relations NW, N, NE, W, EQ, E, SW, S, SE as in Table 1. The Cardinal Relation Algebra (CRA) is generated by these nine JEPD relations.

CRA can be viewed as the Cartesian product of two PAs.

**Definition 3** (Interval Algebra [1]). Let \( U \) be the set of closed intervals on the real line. Thirteen binary relations between two intervals \( x = [x^-, x^+] \) and \( y = [y^-, y^+] \) are defined by the order of the four endpoints of \( x \) and \( y \), see Table 2. The Interval Algebra (IA) is generated by these JEPD relations.
<table>
<thead>
<tr>
<th>Relation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NW</td>
<td>$x &lt; x', y &gt; y'$</td>
</tr>
<tr>
<td>N</td>
<td>$x = x', y &gt; y'$</td>
</tr>
<tr>
<td>NW</td>
<td>$x &gt; x', y &gt; y'$</td>
</tr>
<tr>
<td>W</td>
<td>$x &lt; x', y = y'$</td>
</tr>
<tr>
<td>EQ</td>
<td>$x = x', y = y'$</td>
</tr>
<tr>
<td>E</td>
<td>$x &gt; x', y = y'$</td>
</tr>
<tr>
<td>SW</td>
<td>$x &lt; x', y &lt; y'$</td>
</tr>
<tr>
<td>S</td>
<td>$x = x', y &lt; y'$</td>
</tr>
<tr>
<td>SW</td>
<td>$x &gt; x', y &lt; y'$</td>
</tr>
</tbody>
</table>

Table 1: Basic relations of CRA.

<table>
<thead>
<tr>
<th>Relation</th>
<th>Symbol</th>
<th>Converse</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>before</td>
<td>b</td>
<td>bi</td>
<td>$x^- &lt; x^+ &lt; y^- &lt; y^+$</td>
</tr>
<tr>
<td>meets</td>
<td>m</td>
<td>mi</td>
<td>$x^- &lt; x^+ = y^- &lt; y^+$</td>
</tr>
<tr>
<td>overlaps</td>
<td>o</td>
<td>oi</td>
<td>$x^- &lt; y^- &lt; x^+ &lt; y^+$</td>
</tr>
<tr>
<td>starts</td>
<td>s</td>
<td>si</td>
<td>$x^- = y^- &lt; x^+ &lt; y^+$</td>
</tr>
<tr>
<td>during</td>
<td>d</td>
<td>di</td>
<td>$y^- &lt; x^- &lt; x^+ &lt; y^+$</td>
</tr>
<tr>
<td>finishes</td>
<td>f</td>
<td>fi</td>
<td>$y^- &lt; x^- &lt; x^+ = y^+$</td>
</tr>
<tr>
<td>equals</td>
<td>eq</td>
<td>eq</td>
<td>$x^- = y^- &lt; x^+ = y^+$</td>
</tr>
</tbody>
</table>

Table 2: Basic IA relations and their converses, where $x = [x^-, x^+]$, $y = [y^-, y^+]$ are two intervals.
Unlike the above qualitative calculi, the RCC algebras have interpretations in arbitrary topological spaces. Since applications in GIS and many other spatial reasoning tasks mainly consider objects represented in the real plane, in this paper, we only consider interpretations where regions are interpreted as nonempty regular closed sets, and two regions are connected if they somehow intersect.

**Definition 4 (RCC5 and RCC8 Algebras).** Let \( U \) be the set of nonempty regular closed sets, or regions, in the real plane. The RCC8 algebra is generated by the eight topological relations

\[
\text{DC, EC, PO, EQ, TPP, NTPP, TPPi, NTPPi},
\]

where \( \text{DC, EC, PO, TPP and NTPP} \) are defined in Table 3. \( \text{EQ} \) is the identity relation, and \( \text{TPPi} \) and \( \text{NTPPi} \) are the converses of \( \text{TPP} \) and \( \text{NTPP} \) respectively. See Figure 2 for illustration. It is worth mentioning that these eight relations are all definable by the connectedness relation \( C \), which is the complement of \( \text{DC} \) and two regions are connected if they have nonempty intersection.

The RCC5 algebra is the sub-algebra of RCC8 generated by the five part-whole relations

\[
\text{DR, PO, EQ, PP, PPi},
\]

where \( \text{DR} = \text{DC} \cup \text{EC}, \text{PP} = \text{TPP} \cup \text{NTPP}, \) and \( \text{PPi} = \text{TPPi} \cup \text{NTPPi}. \)

While the RCC algebras defined as above using a ‘weak’ connectedness relation, we will introduce another interpretation in Section 4.4.3 based on a ‘strong’ connectedness relation.

### 2.2. Qualitative constraint satisfaction problem

A qualitative calculus \( \mathcal{M} \) provides a constraint language by using formulas of the form \( (v_i \alpha v_j) \), where \( \alpha \) is a relation in \( \mathcal{M} \) and \( v_i, v_j \) are variables taking values

---

\[\footnote{We note that restricting the underlying topological space may drastically change the computational properties of calculi like RCC8 [4, 39].} \]
from the universe of $M$. Formulas of the form $(v_i \alpha v_j)$ are called constraints
(over $M$). If $\alpha$ is a basic relation in $M$, $(v_i \alpha v_j)$ is called a basic constraint. The
classical consistency problem over $M$ can then be formulated as below.

**Definition 5.** [2] Let $M$ be a qualitative calculus on universe $U$. Suppose $S$ is a
subset of $M$. The consistency problem CSPSAT($S$) is defined as follows:

**Instance:** A 2-tuple $(V, \Gamma)$. Here $V$ is a finite set of variables \{v_1, v_2, \ldots, v_n\},
and $\Gamma = \{v_i \gamma_{ij} v_j : 1 \leq i, j \leq n\}$ is a binary constraint network and each $\gamma_{ij}$
is in $S$.

**Question:** Is there an instantiation $\nu : V \rightarrow U$ such that all the constraints
in $\Gamma$ are satisfied?

If $\nu$ satisfies all the constraints in $\Gamma$, then we say $\nu$ is a solution of $\Gamma$ and say
$\Gamma$ is consistent or satisfiable.

**Notation.** In this paper, we also represent an instantiation $\nu : V \rightarrow U$ as an $n$-tuple
$(\nu(v_1), \nu(v_2), \ldots, \nu(v_n))$.

We note that each instance $(V, \Gamma)$ in CSPSAT($S$) is complete in the sense that
the relation $\gamma_{ij}$ between any two variables $v_i, v_j$ is taken from $S$. Given a binary
constraint work $\Gamma = \{v_i \gamma_{ij} v_j : 1 \leq i, j \leq n\}$, we say $\Gamma$ is a basic constraint network
if $\gamma_{ij}$ is either the universal relation or a basic relation for any two variables $v_i, v_j$;
and say $\Gamma$ is a complete basic constraint network if $\gamma_{ij}$ is a basic relation for any
two variables $v_i, v_j$. In other words, each complete basic constraint network is an
instance of CSPSAT($B_M$), while each basic constraint network is an instance of
CSPSAT($B_M \cup \{*_M\}$), where $B_M$ is the set of basic relations in $M$, and $*_M$ is the
universal relation of $\mathcal{M}$.

The consistency problem as defined in Definition 5 has been investigated for many calculi (see e.g. [1, 43, 35, 40, 30, 26]). In particular, the consistency problem $\text{CSPSAT}(PA)$ can be solved in $O(n^2)$ time, where $n$ is the number of variables [43]. For most other qualitative calculi, including IA, CRA, RCC5, and RCC8, the consistency problem $\text{CSPSAT}(\mathcal{M})$ is NP-complete.

When only basic constraint networks are considered, however, the consistency problem over each of these four calculi becomes tractable. In fact, it can be decided by checking whether the network is path-consistent. For binary relations $\alpha$ and $\beta$, we write $\alpha^{-}$ for the converse of $\alpha$, and $\alpha \circ \beta$ for the usual composition of $\alpha$ and $\beta$. We say a complete basic constraint network $\Gamma = \{v_i \alpha_{ij} v_j : 1 \leq i, j \leq n\}$ is path-consistent, if for any three variables $v_i, v_j, v_k$, we have

$$\alpha_{ij} = \alpha_{ji}^{-} \quad \text{and} \quad \alpha_{ij} \cap (\alpha_{ik} \circ \alpha_{kj}) \neq \emptyset$$

for any $1 \leq i, j, k \leq n$.

Note that for complete basic constraint networks, path-consistency is equivalent to saying that every subnetwork with three variables is consistent. As a local property, path-consistency can be enforced in cubic time.

We summarise the computational complexity results of these calculi in the following theorem.

**Theorem 1.** [35, 24, 40] The consistency problem $\text{CSPSAT}(PA)$ is in P. Let $\mathcal{M}$ be one of IA, CRA, RCC5, and RCC8. Then $\text{CSPSAT}(\mathcal{B}_M)$ is in P and $\text{CSPSAT}(\mathcal{M})$ is NP-complete.

A complete basic network is globally consistent if any partial solution can be extended to a global solution. The following theorem can be directly proven by exploiting the density of real numbers.

**Theorem 2.** Let $\mathcal{M}$ be one of PA, IA, and CRA. Then a complete basic network is globally consistent if it is path-consistent.

We note that RCC5 and RCC8 do not have this property.

---

3The consistency problems $\text{CSPSAT}(\mathcal{B}_M)$ and $\text{CSPSAT}(\mathcal{B}_M \cup \{\ast_M\})$ may have different complexities. For example, there exists a cubic algorithm for solving complete basic CDC (cardinal direction calculus) networks [30], but it is NP-hard to solve basic CDC networks [26].

4This definition of path-consistency is different from the same notion for finite CSPs [33, 32].
2.3. Extended qualitative CSP

By Definition 5 in the classical consistency problem, each variable can in principle take any value in the universe. In many practical applications, however, it is very common to have additional knowledge about some variables (cf. the restaurant and MacDonald’s example in the Introduction), which will affect the consistency of qualitative CSPs. It is therefore necessary to extend the qualitative CSP framework to allow restricted domains of variables.

Definition 6. Let \( \mathcal{M} \) be a qualitative calculus on universe \( U \). Suppose \( S \) is a subset of \( \mathcal{M} \). The consistency problem \( \text{CSPSAT}_f(S) \) is defined as follows, where the subscript ‘\( f \)’ stands for ‘finite’:

**Instance:** A 3-tuple \((V, \Gamma, D)\). Here \( V \) is a finite set of variables \( \{v_1, v_2, \ldots, v_n\} \), \( D \) is an \( n \)-tuple \((D_1, D_2, \ldots, D_n)\), where each \( D_i \) is either \( U \) or a nonempty finite subset of \( U \), and \( \Gamma = \{v_i \gamma_{ij} v_j : 1 \leq i, j \leq n\} \) is a binary constraint network and each \( \gamma_{ij} \) is in \( S \).

**Question:** Is there an instantiation \( \nu : V \to U \) such that \( \nu(v_i) \in D_i \) for each \( i \) and all the constraints in \( \Gamma \) are satisfied?

We say that a variable \( v_i \) appearing in the instance \((V, \Gamma, D)\) is finitely restricted if its domain \( D_i \) is finite. If \( \nu \) satisfies all the constraints in \( \Gamma \) and \( \nu(v_i) \in D_i \) for each \( i \), then we say \( \nu \) is a solution of \((V, \Gamma, D)\) and say \((V, \Gamma, D)\) is consistent or satisfiable. We call elements of each finite domain \( D_i \) landmarks of \((V, \Gamma, D)\).

As a special case, if each finite domain \( D_i \) is required to be a singleton, we write the corresponding consistency problem \( \text{CSPSAT}_s(S) \), where the subscript ‘\( s \)’ denotes ‘singleton’.

An instance of \( \text{CSPSAT}(S) \) is clearly an instance of both \( \text{CSPSAT}_s(S) \) and \( \text{CSPSAT}_f(S) \): we only need to let each \( D_i \) be the universe.

**Proposition 1.** Suppose \( \mathcal{B}_M \) is the set of basic relations in a qualitative calculus \( \mathcal{M} \), and \( S \) is a subclass of \( \mathcal{M} \). Then we have

i) \( \text{CSPSAT}(S) \subseteq \text{CSPSAT}_s(S) \subseteq \text{CSPSAT}_f(S) \);

ii) \( \text{CSPSAT}_f(\mathcal{M}) \) is in \( \text{NP} \) if \( \text{CSPSAT}_f(\mathcal{B}_M) \) is in \( \text{NP} \);

iii) \( \text{CSPSAT}_f(S) \) is in \( \text{NP} \) if \( \text{CSPSAT}_s(S) \) is in \( \text{NP} \);

iv) \( \text{CSPSAT}_f(\mathcal{M}) \) is in \( \text{NP} \) if \( \text{CSPSAT}_s(\mathcal{B}_M) \) is in \( \text{NP} \).
Proof. i) follows directly from the definition. As for ii), suppose we already have a nondeterministic Turing machine $T_0$ which solves $\text{CSPSAT}_f(B_M)$ in polynomial time. Given a non-basic constraint network $(V, \Gamma, D)$, it is consistent iff there is a consistent basic constraint network $\Gamma'$ that refines $\Gamma$ in the sense that for each constraint $(x\alpha'y)$ in $\Gamma$ there exists a constraint $(x\alpha'y)$ in $\Gamma'$ such that $\alpha' \subseteq \alpha$. A basic constraint network that refines $\Gamma$ is often called a scenario of $\Gamma$. We devise a nondeterministic Turing machine $T$ as follows. $T$ first guesses a scenario $(V, \Gamma', D)$ of $(V, \Gamma, D)$, and then calls $T_0$ to decide the consistency of $(V, \Gamma', D)$. $T'$ asserts the instance to be consistent if $T$ returns consistent in any branch. It is clear that the nondeterministic Turing machine $T$ decides the consistency of $(V, \Gamma, D)$ in polynomial time. Similar argument applies to iii), and iv) follows from ii) and iii) directly.

By the above proposition, the computational complexity of $\text{CSPSAT}_f$ is in general higher than that of $\text{CSPSAT}_s$ and $\text{CSPSAT}$, as far as the same subset $S$ of the same calculus is considered. In particular, recall that the classical consistency problems for CRA, IA, RCC5 and RCC8 are all NP-complete. We have the following corollary.

Corollary 1. The consistency problem $\text{CSPSAT}_s(M)$ and $\text{CSPSAT}_f(M)$ are all NP-hard for $M$ being any one of IA, CRA, RCC5, and RCC8.

To determine the computational complexity of reasoning with a qualitative calculus $M$, we will begin with $\text{CSPSAT}_s(B_M)$.

Our computational complexity results are summarised in Table 4, where qualitative calculus $M$ is PA, CRA, IA, RCC5 or RCC8, and $S$ is either $B_M$ or $M$ itself (i.e., we consider either complete basic networks or the most general case).

<table>
<thead>
<tr>
<th>$M$</th>
<th>PA</th>
<th>CRA</th>
<th>IA</th>
<th>RCC5</th>
<th>RCC8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{CSPSAT}(S)$</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>NP-C</td>
<td>P</td>
</tr>
<tr>
<td>$\text{CSPSAT}_s(S)$</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>NP-C</td>
<td>P</td>
</tr>
<tr>
<td>$\text{CSPSAT}_f(S)$</td>
<td>P</td>
<td>NP-C</td>
<td>NP-C</td>
<td>NP-C</td>
<td>NP-C</td>
</tr>
</tbody>
</table>

Table 4: Computational complexity results summary

In the following sections, we first consider point-based calculi PA, CRA, and IA, and then consider region-based calculi RCC5 and RCC8. Unlike point-based calculi, the geometrical representation (in particular, shape and location) of the landmarks may affect the existence of solutions in the plane. To make the analysis more meaningful, we assume that all the landmarks in RCC5 and RCC8 constraint
networks are represented as polygons which may have different connected components and holes. This assumption is practical because polygons are the most widely used approximations of regions in spatial databases.

The NP-hardness results in Table 1 obtained in this paper are mainly achieved by designing polynomial reductions from the Graph 3-Colouring problem, which is a well-known NP-complete problem. Recall that a graph \( G = (V, E) \) is 3-colourable if there is a function \( f : V \rightarrow \{0, 1, 2\} \) such that \( f(v) \neq f(v') \) for each edge \( (v, v') \in E \). The Graph 3-Colouring problem is to decide whether a graph is 3-colourable.

3. Point-based Qualitative Calculi

This section discusses the consistency problems in the extended framework for the three point-based qualitative calculi, viz. Point Algebra, Interval Algebra, and Cardinal Relation Algebra.

3.1. Some simple results

To prove the computational complexity results, we will need the following notion of a finitely restricted sub-instance.

**Definition 7.** Let \( \mathcal{M} \) be a qualitative calculus with universe \( U \), and let \( \mathcal{S} \) be a subclass of \( \mathcal{M} \). Suppose \((V, \Gamma, D)\) is an instance of CSPSAT\(_f(\mathcal{S})\), where \( V = \{v_1, \ldots, v_n\} \), \( D = (D_1, \ldots, D_n) \) and \( \Gamma = \{v_i \alpha_{ij} v_j\}_{1 \leq i, j \leq n} \). Let \( V' = \{v_i : D_i \neq U\} \) be the set of finitely restricted variables in \( V \). Suppose \( V' = \{v_{i_1}, v_{i_2}, \ldots, v_{i_k}\} \). Let \( \Gamma' = \{v_{i_r} \alpha_{rs} v_{i_s}\}_{1 \leq r, s \leq k} \) and \( D' = (D_{i_1}, D_{i_2}, \ldots, D_{i_k}) \). We call \((V', \Gamma', D')\), which is also an instance of CSPSAT\(_f(\mathcal{S})\), the finitely restricted sub-instance of \((V, \Gamma, D)\).

For complete basic constraint networks, we have the following general result.

**Lemma 1.** Let \( \mathcal{M} \) be one of PA, IA, and CRA. Suppose \((V, \Gamma, D)\) is an instance of CSPSAT\(_f(\mathcal{B}_M)\). Then \((V, \Gamma, D)\) is consistent iff \( \Gamma \) is path-consistent and the finitely restricted sub-instance of \((V, \Gamma, D)\) is consistent.

**Proof.** The necessity is clear. We prove the sufficiency, which uses the property that any consistent basic PA (IA or CRA) network is also globally consistent.

Because the finitely restricted sub-instance \((V', \Gamma', D')\) is consistent, it has a solution, say \( b = (b_1, \ldots, b_k) \). Note that \( b \) is a partial solution of the CSPSAT\(_f(\mathcal{B}_{PA})\) instance \((V, \Gamma)\), and thus, by Theorem 2, can be extended to a solution \( b' \) of \((V, \Gamma)\). It is clear that \( b' \) is also a solution of \((V, \Gamma, D)\). Therefore \((V, \Gamma, D)\) is consistent.

The cases for IA and CRA can be proven in the same way. \( \square \)
Using Lemma [1] we are able to show the following computational complexity results.

**Theorem 3.** For PA, we have \( \text{CSPSAT}_s(\mathcal{B}_{PA}) \) and \( \text{CSPSAT}_s(\mathcal{P}A) \) are in P and \( \text{CSPSAT}_f(\mathcal{P}A) \) is in NP. Let \( \mathcal{M} \) be IA or CRA. Then \( \text{CSPSAT}_s(\mathcal{B}_{\mathcal{M}}) \) is in P, and \( \text{CSPSAT}_s(\mathcal{M}) \) and \( \text{CSPSAT}_f(\mathcal{M}) \) are NP-complete.

**Proof.** For PA, we recall that \( \text{CSPSAT}(\mathcal{P}A) \) can be solved in \( O(n^2) \) time \([43]\). Suppose \((V, \Gamma, D)\) is an instance of \( \text{CSPSAT}_s(\mathcal{P}A) \). We show that the consistency of \((V, \Gamma, D)\) can be determined in polynomial time. For a pair of variables \( v_i \) and \( v_j \) such that \( D_i = \{d_i\} \) and \( D_j = \{d_j\} \) are both singletons, suppose \((v_i \alpha v_j)\) is in \( \Gamma \), and \( \beta \) is the basic PA relation between \( d_i \) and \( d_j \). It is clear that \((V, \Gamma, D)\) is inconsistent if \( \beta \) is not included in \( \alpha \). Without loss of generality, we assume \( \alpha \) is a basic relation and \( \alpha = \beta \). Under this assumption, we show that \((V, \Gamma, D)\) is consistent iff the \( \text{CSPSAT}(\mathcal{P}A) \) instance \((V, \Gamma')\) is consistent. The necessity is clear. For the sufficiency, suppose \((V, \Gamma)\) is consistent and has a consistent scenario \((V, \Gamma_0)\). Note that the finitely restricted sub-instance of \((V, \Gamma_0, D)\) is consistent, as the constraint between any two variables with a singleton domain is the actual relation between the corresponding landmarks. By Lemma [1] we have \((V, \Gamma, D)\) is consistent. Because the consistency of \((V, \Gamma)\) can be decided in polynomial time \([43]\), we know that \( \text{CSPSAT}_s(\mathcal{P}A) \) is in P and consequently \( \text{CSPSAT}_s(\mathcal{B}_{PA}) \) is in P and \( \text{CSPSAT}_f(\mathcal{P}A) \) is in NP. \( \square \)

For \( \mathcal{M} \) being IA or CRA, suppose \((V, \Gamma, D)\) is an instance of \( \text{CSPSAT}_s(\mathcal{B}_{\mathcal{M}}) \), and \((V', \Gamma', D')\) is its finitely restricted sub-instance. Assume that \( V \) has \( n \) variables and \( V' \) has \( m \leq n \) variables. The path-consistency of \( \Gamma \) can be checked in \( O(n^3) \) time. Moreover, the consistency of \((V', \Gamma', D')\) can be decided in \( O(m^2) \) time, as we only need to check for each pair of variables \( v_i \) and \( v_j \) in \( V' \) whether the unique landmarks specified for them satisfy the constraint between them. By Lemma [1] the consistency of \((V, \Gamma, D)\) can be determined in \( O(n^3) \) time. Therefore, \( \text{CSPSAT}_s(\mathcal{B}_{\mathcal{M}}) \) is in P. By Proposition [1] we know \( \text{CSPSAT}_s(\mathcal{M}) \) and \( \text{CSPSAT}_f(\mathcal{M}) \) are all in NP. Meanwhile, the NP-completeness of \( \text{CSPSAT}(\mathcal{M}) \) implies that \( \text{CSPSAT}_s(\mathcal{M}) \) and \( \text{CSPSAT}_f(\mathcal{M}) \) are all NP-complete. \( \square \)

The following subsections will respectively show that (i) \( \text{CSPSAT}_f(\mathcal{B}_{PA}) \) is in P but \( \text{CSPSAT}_f(\mathcal{P}A) \) is NP-complete, and (ii) \( \text{CSPSAT}_f(\mathcal{B}_{\mathcal{M}}) \) is NP-complete for \( \mathcal{M} \) being IA or CRA.

---

Suppose \( \mathcal{M} \) is one of PA, IA, or CRA. Then this result can be generalised to any tractable subclass \( S \) of \( \mathcal{M} \) that contains all basic relations.
3.2. Point Algebra

We first propose a polynomial algorithm that solves $\text{CSPSAT}_f(B_{PA})$ and then provide a polynomial reduction from Graph 3-Colouring to $\text{CSPSAT}_f(PA)$.

Let $(V, \Gamma, D)$ be an instance of $\text{CSPSAT}_f(B_{PA})$. By Lemma 1 we know that $(V, \Gamma, D)$ is consistent iff $\Gamma$ is path-consistent and the finitely restricted sub-instance $(V', \Gamma', D')$ of $(V, \Gamma, D)$ is consistent. Because path-consistency can be determined in cubic time, we only need to devise a polynomial algorithm for checking whether $(V', \Gamma', D')$ is consistent. To this end, we show that such a consistent instance of $\text{CSPSAT}_f(B_{PA})$ has a minimal solution in a sense.

**Proposition 2.** Suppose $(V, \Gamma, D)$ is an instance of $\text{CSPSAT}_f(B_{PA})$ such that $D = \{D_1, D_2, \ldots, D_n\}$ and each $D_i$ is a finite set of real numbers. If $(V, \Gamma, D)$ is consistent, then there is a unique solution $(a_1, \ldots, a_n)$ such that $a_i \leq a'_i$ $(1 \leq i \leq n)$ for any other solution $(a'_1, a'_2, \ldots, a'_n)$. Furthermore, if $\Gamma = \{v_i < v_j\}_{1 \leq i < j \leq n}$, then

- $a_1 = \min D_1$;
- $a_k = \min \{x \in D_k : x > a_{k-1}\}$ for $k = 2, 3, \ldots, n$.

**Proof.** Assume $\Gamma = \{v_i < v_j\}_{1 \leq i < j \leq n}$. This does not lose generality because we can combine variables related by the ‘$=$’ constraint and then sort the variables by the ‘$<$’ and ‘$>$’ constraints. Every $D_i$ is a finite set, so $(V, \Gamma, D)$ has at most finitely many, say $k$, solutions. Suppose $(a^1_i, a^2_i, \ldots, a^s_i)$ $(i = 1, 2, \ldots, k)$ enumerate all solutions. Let $a_j = \min \{a^j_i\}_{1 \leq i \leq s}$. We claim that $(a_1, a_2, \ldots, a_n)$ is the minimal solution. We only need to prove that it is a solution of $(V, \Gamma, D)$, i.e. to show (i) each $a_j$ is in $D_j$; and (ii) $a_1 < a_2 < \ldots < a_n$. Because $a^j_i \in D_j$, we know $a_j = \min \{a^j_i\}_{1 \leq i \leq s}$ is in $D_j$. We next prove $a_1 < a_2$. Suppose $a_2 = a^j_2$ for some $j$ by definition. Then $a_1 = \min \{a^j_1\}_{1 \leq i \leq s} \leq a^j_1 < a^j_2 = a_2$. By using induction, we can also prove $a_2 < a_3 < \ldots < a_n$. Therefore, $(a_1, a_2, \ldots, a_n)$ is the minimal solution of $(V, \Gamma, D)$. \hfill $\square$

We next propose a polynomial algorithm that solves $\text{CSPSAT}_f(B_{PA})$ based on Proposition 2. For any instance $(V, \Gamma, D)$ of $\text{CSPSAT}_f(B_{PA})$, we first check whether $\Gamma$ is consistent. If it is inconsistent, then so is $(V, \Gamma, D)$. Otherwise, we check whether the finitely restricted sub-instance $(V', \Gamma', D')$ of $(V, \Gamma, D)$ is consistent. To this end, we attempt to compute the minimal solution $(a_1, \ldots, a_n)$ by procedures described in Proposition 2. If in the $k$-th step $\{x \in D_k : x > a_{k-1}\}$ is empty, then we conclude that the sub-instance, and thus the original instance, is inconsistent. If we succeed in computing $(a_1, a_2, \ldots, a_n)$, then it is a solution.
of the sub-instance and can be extended to a solution of the original instance. The soundness of the algorithm is clear by the above argument.

**Input:** \(\text{CSPSAT}_f(B_{PA})\) instance \((V, \Gamma, D)\)

**Output:** The consistency of \((V, \Gamma, D)\)

1. if \(\Gamma\) is not consistent then
2. \(\text{return} \text{ ‘Inconsistent’};\)
3. \((V', \Gamma', D') \leftarrow \) finitely restricted sub-instance of \((V, \Gamma, D)\);
4. Sort \(V'\) to \(v'_1 < \ldots < v'_{n'}\) by \(\Gamma'\), modify \(D'\) correspondingly;
5. \(a_1 \leftarrow \min D'_1;\)
6. for \(2 \leq k \leq n'\) do
7. \(\text{if } a_{k-1} \geq \max D'_k \text{ then}\)
8. \(\text{return} \text{ ‘Inconsistent’};\)
9. \(a_k \leftarrow \min \{x \in D'_k : x > a_{k-1}\};\)
10. \(\text{end}\)
11. \(\text{return} \text{ ‘Consistent’}.\)

<table>
<thead>
<tr>
<th>Algorithm 1: \text{SOLVING CSPSAT}<em>f(B</em>{PA})</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> (\text{CSPSAT}<em>f(B</em>{PA})) instance ((V, \Gamma, D))</td>
</tr>
<tr>
<td><strong>Output:</strong> The consistency of ((V, \Gamma, D))</td>
</tr>
<tr>
<td>1. if (\Gamma) is not consistent then</td>
</tr>
<tr>
<td>2. (\text{return} \text{ ‘Inconsistent’};)</td>
</tr>
<tr>
<td>3. ((V', \Gamma', D') \leftarrow ) finitely restricted sub-instance of ((V, \Gamma, D));</td>
</tr>
<tr>
<td>4. Sort (V') to (v'<em>1 &lt; \ldots &lt; v'</em>{n'}) by (\Gamma'), modify (D') correspondingly;</td>
</tr>
<tr>
<td>5. (a_1 \leftarrow \min D'_1;)</td>
</tr>
<tr>
<td>6. for (2 \leq k \leq n') do</td>
</tr>
<tr>
<td>7. (\text{if } a_{k-1} \geq \max D'_k \text{ then})</td>
</tr>
<tr>
<td>8. (\text{return} \text{ ‘Inconsistent’};)</td>
</tr>
<tr>
<td>9. (a_k \leftarrow \min {x \in D'<em>k : x &gt; a</em>{k-1}};)</td>
</tr>
<tr>
<td>10. (\text{end})</td>
</tr>
<tr>
<td>11. (\text{return} \text{ ‘Consistent’}.)</td>
</tr>
</tbody>
</table>

**Theorem 4.** Algorithm[?] solves \(\text{CSPSAT}_f(B_{PA})\).

We next analyse the computational complexity of the algorithm. Suppose there are \(n\) variables in \(V\), and the sum of the cardinalities of all finite \(D_i\) is \(L\). Then the input size is \(O(n^2 + L)\) \((n^2\) constraints and \(L\) points\). The following proposition shows the optimality of the algorithm.

**Proposition 3.** The computational complexity of Algorithm[?] is \(O(n^2 + L)\).

**Proof.** Let \((V, \Gamma, D)\) be an instance of \(\text{CSPSAT}_f(B_{PA})\). The consistency of \(\Gamma\) can be computed in \(O(n^2)\) time by Algorithm \text{CSPAN} proposed in [43]. Sorting \(V'\) takes \(O(n \log n)\) time. Let \(l_i\) be the cardinality of \(D'_i\). Then step ‘\(a_1 \leftarrow \min D'_1\)’ takes \(O(l_1)\) time, and the \(i\)-th loop body takes \(O(l_{i+1})\) time \((i = 1, 2, \ldots, n' - 1)\). Therefore, the computational complexity of the algorithm is \(O(n^2 + n \log n + l_1 + l_2 + \ldots + l_{n'}) = O(n^2 + L)\). □

Despite the fact that both \(\text{CSPSAT}(PA)\) and \(\text{CSPSAT}_f(B_{PA})\) are in P, the next theorem shows that \(\text{CSPSAT}_f(PA)\) is NP-hard. We prove this by using a polynomial reduction from the Graph 3-Colouring problem to \(\text{CSPSAT}_f(PA)\).

**Theorem 5.** The consistency problem \(\text{CSPSAT}_f(PA)\) is NP-complete.
Proof. Let $G = (V, E)$ be a graph, where $V = \{v_0, \ldots, v_n\}$. Define a CSPSAT$_f(PA)$ instance $(U_G, \Gamma_G, D_G)$ as follows:

\[
U_G = \{u_0, \ldots, u_n\},
\]
\[
D_G = \{D_{u_0}, \ldots, D_{u_n}\}, \quad \text{where} \quad D_{u_i} = \{0, 1, 2\},
\]
\[
\Gamma_G = \{u_i \neq u_{i'} : (v_i, v_{i'}) \in E\}.
\]

That is, we construct for each vertex $v_i \in V$ a corresponding temporal variable $u_i$ which takes value from $\{0, 1, 2\}$; and we specify for each edge $(v_i, v_{i'}) \in E$ a constraint $(u_i \neq u_{i'})$. It is clear that $G = (V, E)$ can be 3-colourable iff $(U_G, \Gamma_G, D_G)$ is satisfiable. Therefore the consistency problem CSPSAT$_f(PA)$ is NP-hard, and hence NP-complete as its NP-membership has been identified in Theorem 3.

Remark 1. The NP-hardness of CSPSAT$_f(PA)$ is due to the uncertainty of the non-equal ($\neq$) constraints and the finiteness of the domains. It can be proven that CSPSAT$_f(S)$ is in P for $S = \{<,\geq,\leq,\geq,\neq\}$ (i.e., with $\pm$ removed from PA). A polynomial algorithm can be devised based on the observation that the concept of a minimal solution still applies. The algorithm first merges the variables which are required to be equal by the constraints (see [43]). Note the domains of the merged variables should also be revised as the intersection of their original domains. The algorithm then adopts a topological sort, during which each finitely restricted variable is assigned a value in its domain as small as possible.

3.3. Cardinal Relation Algebra

To show that CSPSAT$_f(B_{CRA})$ is NP-hard, we design a polynomial reduction from Graph 3-Colouring to CSPSAT$_f(B_{CRA})$. Suppose $G = (V, E)$ is a graph with vertex set $V = \{v_0, \ldots, v_n\}$. We construct an instance $(U_G, \Gamma_G, D_G)$ of CSPSAT$_f(B_{CRA})$ such that $(U_G, \Gamma_G, D_G)$ is satisfiable iff $G$ is 3-colourable.

First, for each vertex $v_i \in V$, we introduce a spatial variable $u_i$ with domain

\[
D_{u_i} = \{(3i, 3i), (3i + 1, 3i + 1), (3i + 2, 3i + 2)\}.
\]

We say $u_i$ is at position $p$ (where $p \in \{0, 1, 2\}$), if it takes the point $(3i + p, 3i + p)$ in $D_{u_i}$. Second, for each edge $e_j = (v_i, v_{i'}) \in E$ (assuming $i < i'$), we introduce a spatial variable $w_j$ with domain

\[
D_{w_j} = \{(3i + p, 3i' + q) : p, q \in \{0, 1, 2\}, p \neq q\},
\]

---

6 We assume that the constraint between two variables is the universal constraint if it is not specified in $\Gamma_G$.

7 We here specially thank the reviewer who suggested this elegant reduction to us.
and add two constraints \( (w_j \in u_i) \) and \( (w_j \in u'_i) \) to \( \Gamma_G \). That is to say, \( w_j \) should be to the east of \( u_i \) and to the south of \( u'_i \). The domain of \( w_j \) is used to rule out the cases when \( u_i \) and \( u'_i \) are at the same position (with respect to their own domains), which correspond to the requirement that vertices \( v_i \) and \( v'_i \) cannot be coloured the same as they are connected by edge \( e_j \).

Note that each \( \text{CSPSAT}_f(B_{CRA}) \) instance is a complete network. This means that we should specify for each pair of variables in \( U_G \) a basic CRA constraint. In above we have specified such a constraint for two spatial variables \( u_i \) and \( w_j \) when \( v_i \) is a vertex incident to edge \( e_j \) in \( G \). There are three other cases unspecified:

- The constraint between \( u_i \) and \( w_i' \);
- The constraint between \( u_i \) and \( w_j \), where \( v_i \) is not incident to edge \( e_j \) in \( G \);
- The constraint between \( w_j \) and \( w_j' \).

In each case it is straightforward to specify a basic constraint between the two spatial variables.

**Example 1.** Suppose \( G = (V, E) \) is a graph, where \( V = \{v_0, v_1, v_2\} \) and \( E = \{(v_0, v_1), (v_1, v_2)\} \). Let \( (U_G, \Gamma_G, D_G) \) be the \( \text{CSPSAT}_f(B_{CRA}) \) instance constructed as above for \( G \). Then \( U_G = \{u_0, u_1, u_2, w_0, w_1\} \), with their domains shown in Figure 3. The constraints in \( \Gamma_G \) are given in Table 5 where constraints in black are those corresponding to edges in \( E \).

![Figure 3: Domains of \((U_G, \Gamma_G, D_G)\)](image)

<table>
<thead>
<tr>
<th>( u_0 )</th>
<th>( u_1 )</th>
<th>( u_2 )</th>
<th>( w_0 )</th>
<th>( w_1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>EQ</td>
<td>SW</td>
<td>SW</td>
<td>W</td>
<td>SW</td>
</tr>
<tr>
<td>EQ</td>
<td>SW</td>
<td>N</td>
<td>W</td>
<td></td>
</tr>
<tr>
<td>EQ</td>
<td>NE</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EQ</td>
<td>SW</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EQ</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5: Constraints of \((U_G, \Gamma_G, D_G)\)

**Proposition 4.** Graph \( G = (V, E) \) is 3-colourable iff \( (U_G, \Gamma_G, D_G) \) is satisfiable.

**Proof.** Straightforward. □

As a consequence, we have:
\[
\begin{array}{c|cccccccc}
\alpha_{ij} & NW & N & NE & W & EQ & E & SW & S & SE \\
\beta_{ij} & di & si & oi & fi & eq & f & o & s & d \\
\end{array}
\]

Table 6: Translation of the constraints

**Theorem 6.** The problem \(\text{CSPSAT}_f(B_{CRA})\) is NP-complete.

**Proof.** Since the reduction above is polynomial, we know that \(\text{CSPSAT}_f(B_{CRA})\) is NP-hard. Meanwhile, the NP-membership of \(\text{CSPSAT}_f(B_{CRA})\) follows from Theorem 3. Therefore, \(\text{CSPSAT}_f(B_{CRA})\) is an NP-complete problem.

---

### 3.4. Interval Algebra

To show that \(\text{CSPSAT}_f(B_{IA})\) is NP-hard, we design a polynomial reduction from \(\text{CSPSAT}_f(B_{CRA})\). Note that an interval \([x, y]\) corresponds to the point \((x, y)\) on the half-plane \(\{(x, y) : x < y\}\). Suppose \((V, \Gamma, D)\) is a \(\text{CSPSAT}_f(B_{CRA})\) instance, where \(V = \{u_1, \ldots, u_n\}\), \(\Gamma = \{u_i \alpha_{ij} u_j : 1 \leq i, j \leq n\}\), \(D = \{D_1, \ldots, D_n\}\). Note that \(D_i\) is either the universe of CRA \(U_{CRA}\) (viz. the real plane), or a finite subset of \(U_{CRA}\). We now translate \((V, \Gamma, D)\) into a \(\text{CSPSAT}_f(B_{IA})\) instance \((V', \Gamma', D')\), where \(\Gamma'\) is a complete basic IA network. The translation maps:

- each variable \(u_i\) in \(V\) to variable \(u'_i\) in \(V'\);
- each basic CRA relation \(\alpha_{ij}\) to a basic IA relation \(\beta_{ij}\) as specified in Table 6;
- each \(D_i\) to \(D'_i\) such that if \(D_i = U_{CRA}\) then \(D'_i\) is the universe of IA \(U_{IA}\);
- if \(D_i\) is finite, then \(D'_i = \{[x, y + \Delta] : (x, y) \in D_i\}\). Here \(\Delta\) is a fixed large number such that \(x < y + \Delta\) for any point \((x, y)\) in any restricted domain \(D_i\).

We show that the translation preserves consistency.

**Proposition 5.** An instance \((V, \Gamma, D)\) in \(\text{CSPSAT}_f(B_{CRA})\) is consistent iff the corresponding instance \((V', \Gamma', D')\) in \(\text{CSPSAT}_f(B_{IA})\) as constructed above is consistent.

**Proof.** Suppose \((a_1, \ldots, a_n)\) is a solution of \((V, \Gamma, D)\), where \(a_i = (x_i, y_i) \in D_i\). Define interval \(a'_i = [x_i, y_i + \Delta] \in D'_i\). We prove that \((a'_1, \ldots, a'_n)\) is a solution of \((V', \Gamma', D')\). It is clear that \(a'_i \in D'_i\) by the translation from \(D_i\) to \(D'_i\). We only need to verify that all the constraints in \(\Gamma'\) are satisfied by \((a'_1, \ldots, a'_n)\). This can be done by discussing each of the nine kinds of basic IA constraints in \(\Gamma'\).

Suppose \((u'_1 \equiv u'_2)\) is a constraint in \(\Gamma'\). We need to prove \([x_i, y_i + \Delta] \equiv [x_j, y_j + \Delta]\), i.e., \(x_i < x_j < y_j + \Delta < y_i + \Delta\). By the translation we know that \((u_i\, NW\, u_j)\) is in \(\Gamma\). Therefore \((x_i, y_i)\, NW\, (x_j, y_j)\), i.e., \(x_i < x_j\) and \(y_i > y_j\). Meanwhile \(x_j < y_j + \Delta\) is guaranteed by the selection of \(\Delta\), so the constraint \((u'_1 \equiv u'_2)\) is satisfied by \((a'_1, \ldots, a'_n)\). The remaining eight cases can be proven analogously. \(\square\)
Therefore we obtain the following result.

**Theorem 7.** The consistency problem CSPSAT\(_f(B_{IA})\) is NP-complete.

**Proof.** Since the reduction from CSPSAT\(_f(B_{CRA})\) to CSPSAT\(_f(B_{IA})\) is polynomial, we know that CSPSAT\(_f(B_{IA})\) is NP-hard. Moreover, by Theorem 3, we have CSPSAT\(_f(B_{IA})\) is in NP. This shows that CSPSAT\(_f(B_{IA})\) is NP-complete. \(\Box\)

So far, we have completed the discussion for the three point-based qualitative calculi. The next section will address region-based qualitative calculi.

### 4. Region-based Qualitative Calculi RCC5 and RCC8

This section discusses the consistency problems over RCC5 and RCC8 in the extended qualitative CSP framework. Note that although the universe of RCC5 (or RCC8) is the set of all regions in the plane, it is reasonable to assume that all landmarks are represented as polygons. This is because landmarks, as inputs of instances, are required to be representable in computers. In other words, they should be finitely representable. Meanwhile, general polygons (which may have holes or multiple components) are the most widely used approximations of regions: they are simple, intuitive, and expressive.

Under the assumption that all landmarks are represented by general polygons, we show in this section that all these consistency problems are in NP. In particular, we show that CSPSAT\(_s(B_{RCC5})\) is in P, but that CSPSAT\(_f(B_{RCC5})\) and CSPSAT\(_s(B_{RCC8})\) are all NP-complete. It is not surprising that CSPSAT\(_f(B_{RCC5})\) is NP-complete if we regard the finitely restricted sub-instance of each instance of CSPSAT\(_f(B_{RCC5})\) as a classical CSP, but the NP-hardness of CSPSAT\(_s(B_{RCC8})\) is quite undesirable. One way to circumvent this obstacle is to use a stronger connectedness instead of the one used in Definition 4.

The remainder of this section is organised as follows. We first introduce a simple computational complexity result in Section 4.1 showing that CSPSAT\(_f(B_{RCC5})\) is NP-hard. Several of our results are related to computing the intersection of landmarks (represented as polygons), so we analyse its computational complexity in Section 4.2. The tractability of CSPSAT\(_s(B_{RCC5})\) is then proven in Section 4.3. Section 4.4 shows that CSPSAT\(_s(B_{RCC8})\) is NP-complete if the RCC8 relations are interpreted as in Definition 4 and proves that the same problem is in P (i.e. tractable) if we adopt another interpretation that uses a stronger connectedness.

---

\(^8\)Another way to represent regions is to use semi-algebraic sets, which are more expressive than polygons but the set operations are much more complicated.
4.1. The NP-hardness of $\text{CSPSAT}_f(B_{RCC5})$

We prove the NP-hardness of $\text{CSPSAT}_f(B_{RCC5})$ by designing a polynomial reduction from the Graph 3-Colouring problem.

**Proposition 6.** The consistency problem $\text{CSPSAT}_f(B_{RCC5})$ is NP-hard.

**Proof.** Let $G = (V, E)$ be a graph, where $V = \{v_0, \ldots, v_n\}$ and $E = \{e_0, \ldots, e_m\}$. For each vertex $v \in V$, we introduce three regions (represented by rectangles) $r^0_v, r^1_v$ and $r^2_v$; for each edge $e \in E$, we introduce three regions (represented by rectangles) $s^0_e, s^1_e$ and $s^2_e$. These rectangles are required to be pairwise disjoint.

For any $1 \leq i \leq n$ and $0 \leq p \leq 2$, we define a landmark $l^p_i$ as

$$l^p_i = r^p_{v_i} \cup \bigcup \{s^p_e : \text{edge } e \text{ is incident to vertex } v_i\}.$$  

Because rectangles $r^p_e, s^p_e$ are pairwise disjoint for $v \in V, e \in E$ and $p \in \{0, 1, 2\}$, it is clear that $l^p_i \cap l^p_j = \emptyset$ if $p \neq q$. For $i \neq j$ and $p = q$, it is also straightforward to see that $l^p_i \cap l^p_j \neq \emptyset$ is a rectangle if $e = (v_i, v_j) \in E$ and $l^p_i \cap l^p_j = \emptyset$ otherwise.

The $\text{CSPSAT}_s(B_{RCC5})$ instance $(V_G, \Gamma_G, D_G)$ is constructed as follows.

- $V_G = \{u_0, u_1, \ldots, u_n\}$,
- $D_G = \{D_{u_0}, D_{u_1}, \ldots, D_{u_n}\}$, where $D_{u_i} = \{l^0_{v_i}, l^1_{v_i}, l^2_{v_i}\}$,
- $\Gamma_G = \{u_i \text{DR} u_j\}$.

Note that spatial variable $u_i$ corresponds to vertex $v_i$, and “vertex $v_i$ is coloured with colour $p$” corresponds to that “variable $u_i$ takes value $l^p_i$.” It is routine to show that $G$ is 3-colourable iff $(V_G, \Gamma_G, D_G)$ is consistent. Because the reduction is polynomial, we know the consistency problem $\text{CSPSAT}_f(B_{RCC5})$ is NP-hard. □

4.2. Planar subdivision and overlay computation

In the following subsections we will see that computing the intersection of landmarks (represented as polygons) is critically important when solving the consistency problems for RCC5 and RCC8 in the extended qualitative CSP framework. To facilitate the discussion, this subsection analyses the computational complexity of computing the intersection of multiple polygons. Our discussion is based on the doubly-connected edge list (DCEL) structure for representing planar subdivisions (cf. e.g. [9]).

A planar subdivision is an embedding of a planar graph in the plane such that its edges are mapped into straight line segments. It consists of vertices, edges, and faces. Vertices are endpoints of line segments, edges are interiors of line segments,
and faces are maximally connected subsets of the plane with all edges and vertices removed. In particular, each face is a connected open set, which may have holes. The outer face is unbounded, but every other face is bounded and its boundary consists of vertices and edges. The complexity of a planar subdivision is defined as the sum of the number of its vertices, the number of its edges, and the number of its faces. For example, the planar subdivision of the rectangle in Figure 4(a) has two faces (Figure 4(a)), four vertices (Figure 4(b)) and four edges (Figure 4(c)), and thus has a complexity of ten.

In what follows, we write \( \text{FACE} \), \( \text{EDGE} \), and \( \text{VTX} \) respectively for the set of faces, the set of edges, and the set of vertices in a planar subdivision, and use lower Fraktur symbols \( f \), \( e \), \( v \) (possibly with indices) to denote, respectively, faces, edges, and vertices in the subdivision.

The following lemma shows that the complexity of a planar subdivision is of the same order as the number of its vertices.

**Lemma 2.** Let \( S \) be a planar subdivision with \( k \) vertices. Then the complexity of \( S \) is \( O(k) \).

**Proof.** Recall that each planar subdivision is an embedding of a planar graph in the plane. By Euler’s formula (cf. [10]), if \( S \) has \( C \) connected components then

\[
|\text{VTX}| - |\text{EDGE}| + |\text{FACE}| = C + 1.
\]

Furthermore, since each face is bounded by at least three edges, and each edge touches at most two faces, it is straightforward to prove that

\[
|\text{EDGE}| < 3|\text{VTX}| \quad \text{and} \quad |\text{FACE}| < 2|\text{VTX}|.
\]

Therefore the complexity of \( S \) is \( O(k) \). \( \square \)

In Computational Geometry, a planar subdivision is usually represented by the doubly-connected edge list (DCEL), where each edge is considered as two directed half-edges with opposite directions. The DCEL of a subdivision maintains a table for each vertex, each half-edge, and each face. The table allows the retrieve from an object (viz. vertex, half-edge, or face) to its incident (or adjacent) objects efficiently. For a planar subdivision \( S \) with complexity \( k \), the DCEL of \( S \) takes \( O(k) \) space.

---

9To avoid potential confusion, when discussing the time resource it takes for computing an overlay, we always explicitly use the term *computational complexity*. 
The overlay of two planar subdivisions $S_1$ and $S_2$ is the planar subdivision $S$ induced by all edges from $S_1$ and $S_2$. Each vertex of $S$ is either a vertex of $S_1$ or $S_2$, or the intersection point of two edges from $S_1$ and $S_2$. Each edge is either an edge of $S_1$ or $S_2$, or a part of an edge of $S_1$ cut by an edge of $S_2$, or vice versa. Similarly, each face of $S$ is either a face of $S_1$ or $S_2$, or the intersection of two faces from $S_1$ and $S_2$. Figures 4(e) and (f) illustrate the overlay of the rectangle in Figure 4(a) and the triangle in Figure 4(d), which has four faces, eleven edges and nine vertices, and hence has a complexity of 24.

We have the following result about the complexity of the overlay.

**Lemma 3.** Let $S_1$ and $S_2$ be two planar subdivisions of complexity $k_1$ and $k_2$ respectively. Then the overlay of $S_1$ and $S_2$ has complexity $O(k_1k_2)$.

**Proof.** Note that each vertex in the overlay is either a vertex of $S_1$, or a vertex of $S_2$, or the intersection point of two edges from different subdivisions. As the numbers of vertices and edges of $S_i$ are less than $k_i$, the overlay has $O(k_1k_2)$ vertices. The complexity of the overlay then follows from Lemma 2. □

The computational complexity of the overlay computation is as follows.

Figure 4: An example of subdivision
Proposition 7. [9, Theorem 2.6] Let $S_1$ and $S_2$ be two planar subdivisions of complexity $k_1$ and $k_2$ respectively. Then the overlay of $S_1$ and $S_2$ can be constructed in $O((k_1 + k_2 + k)\log(k_1 + k_2))$ time, where $k$ is the complexity of the overlay.

Proposition 7 only considers the overlay of two subdivisions. For the consistency problems CSPSAT$_s(B_{RCC5})$ and CSPSAT$_s(B_{RCC8})$, we need to compute the overlay $\mathcal{O}$ of the subdivisions induced by landmarks $l_1, \ldots, l_m$ ($m \geq 3$). At first glance, the computational complexity seems to be very high. Suppose each landmark is represented by a polygon with $k$ vertices. If we use Lemma 2 successively then the overlay will have complexity $O(k^m)$. As a consequence, the computational complexity of computing the overlay will be exponential if we use Proposition 7 successively. The following result shows that, however, $\mathcal{O}$ can be computed in polynomial time. The key idea is that the complexity of the overlay of the $m$ subdivisions is, instead of $O(k^m)$, polynomial in $m$ and $k$ (if we assume each landmark has $k$ vertices).

Lemma 4. Suppose $l_i$ is a polygon with $k_i$ vertices for each $1 \leq i \leq m$. Let $K = \sum_{i=1}^{m} k_i$ and $\mathcal{O}$ be the overlay of the subdivisions induced by these polygons. Then $\mathcal{O}$ has complexity $O(K^2)$ and can be computed in $O(mK^2 \log K)$ time.

Proof. It is clear that there are in total $O(K)$ vertices and, by Lemma 2, $O(K)$ edges in the subdivisions induced by these polygons. As each vertex in the overlay $\mathcal{O}$ is either a vertex of a subdivision, or the intersection point of two edges from different subdivisions, we know that $\mathcal{O}$ has $O(K^2)$ vertices. By Lemma 2, the complexity of $\mathcal{O}$ is also $O(K^2)$.

Write $\mathcal{O}_i$ for the overlay of the subdivisions induced by the first $i$ polygons $l_1, \ldots, l_i$. The complexity of each $\mathcal{O}_i$ is no more than that of $\mathcal{O} = \mathcal{O}_m$. By Proposition 7 we know $\mathcal{O}_{i+1}$ can be computed in $O((K^2 + K + K^2) \log(K^2 + K)) = O(K^2 \log K)$ time from $\mathcal{O}_{i+1}$ and $l_{i+1}$. Therefore, the overlay $\mathcal{O}$ can be computed in $O(mK^2 \log K)$ time from $l_1, \ldots, l_m$.  

We note that the DCEL of $\mathcal{O}$ contains incidence and adjacency information between two elements in FACE, EDGE, and VTX. The relationship between such an element and a polygon in $L$, however, is not provided. For example, the DCEL does not tell us whether an edge lies inside, outside, or on the boundary of a polygon $l_i$. To represent the complete topological information of the polygon system $L$, we introduce the following functions, which can be computed by supplying a number of attributes to each object in the DCEL of the overlay.
For each polygon \( l_i \in L \), we write \( \mathcal{I}_{\text{FACE}}(l_i) \) (\( \mathcal{E}_{\text{FACE}}(l_i) \), resp.) for the set of faces in \( O \) that lie in the interior (exterior, resp.) of \( l_i \):

\[
\mathcal{I}_{\text{FACE}}(l_i) = \{ f \in \text{FACE} : f \subseteq l_i \}, \\
\mathcal{E}_{\text{FACE}}(l_i) = \{ f \in \text{FACE} : f \cap l_i = \emptyset \}.
\]

(1) \hspace{10cm} (2)

It is clear that \( \mathcal{I}_{\text{FACE}}(l_i) \cup \mathcal{E}_{\text{FACE}}(l_i) = \text{FACE} \) and \( \mathcal{I}_{\text{FACE}}(l_i) \cap \mathcal{E}_{\text{FACE}}(l_i) = \emptyset \).

For each polygon \( l_i \), we define

\[
\mathcal{I}_{\text{EDGE}}(l_i) = \{ e \in \text{EDGE} : e \subseteq l_i \}, \\
\mathcal{E}_{\text{EDGE}}(l_i) = \{ e \in \text{EDGE} : e \cap l_i = \emptyset \}, \\
\mathcal{B}_{\text{EDGE}}(l_i) = \{ e \in \text{EDGE} : e \subseteq \partial l_i \},
\]

(3) \hspace{10cm} (4) \hspace{10cm} (5)

and similarly,

\[
\mathcal{I}_{\text{VRTX}}(l_i) = \{ v \in \text{VRTX} : v \subseteq l_i \}, \\
\mathcal{E}_{\text{VRTX}}(l_i) = \{ v \in \text{VRTX} : v \subseteq l_i \}, \\
\mathcal{B}_{\text{VRTX}}(l_i) = \{ v \in \text{VRTX} : v \in \partial l_i \}.
\]

(6) \hspace{10cm} (7) \hspace{10cm} (8)

Because each edge and each vertex is either in the interior of \( l_i \), or in the exterior of \( l_i \), or on the boundary of \( l_i \), we know that \( \{ \mathcal{I}_{\text{EDGE}}(l_i), \mathcal{E}_{\text{EDGE}}(l_i), \mathcal{B}_{\text{EDGE}}(l_i) \} \) is a partition of \( \text{EDGE} \), and \( \{ \mathcal{I}_{\text{VRTX}}(l_i), \mathcal{E}_{\text{VRTX}}(l_i), \mathcal{B}_{\text{VRTX}}(l_i) \} \) is a partition of \( \text{VRTX} \).

We provide an example to illustrate these functions.

**Example 2.** Suppose \( L = \{ l_1, l_2, l_3 \} \) consists of the three polygons illustrated in Figure 5(a). Then we have \( \text{FACE} = \{ f_0, \ldots, f_3 \} \), \( \text{VRTX} = \{ v_1, \ldots, v_{11} \} \) and \( \text{EDGE} = \{ e_1, \ldots, e_{14} \} \), as shown in Figure 5(b-d). In particular, for landmark \( l_1 \), we have

\[
\mathcal{I}_{\text{FACE}}(l_1) = \{ f_1, f_2 \}, \quad \mathcal{I}_{\text{VRTX}}(l_1) = \{ v_6, v_{11} \}, \quad \mathcal{I}_{\text{EDGE}}(l_1) = \{ e_6, e_{10}, e_{11} \}, \\
\mathcal{E}_{\text{FACE}}(l_1) = \{ f_0, f_3, f_4 \}, \quad \mathcal{E}_{\text{VRTX}}(l_1) = \{ v_3, v_8, v_9 \}, \quad \mathcal{E}_{\text{EDGE}}(l_1) = \{ e_2, e_3, e_7, e_8, e_9 \}, \\
\mathcal{B}_{\text{VERTEX}}(l_1) = \{ v_1, v_2, v_7, v_{10}, v_4, v_5 \}, \quad \mathcal{B}_{\text{EDGE}}(l_1) = \{ e_1, e_{12}, e_{13}, e_{14}, e_4, e_5 \}.
\]

Together with the functions defined in (1)-(8), the DCEL of the overlay of polygons in \( L \) completely describes the topological information of polygons in \( L \). The following lemma shows that these functions can also be computed in polynomial time.

**Lemma 5.** Suppose \( l_i \) is a polygon with \( k_i > 2 \) vertices for each \( 1 \leq i \leq m \). Let \( O \) be the overlay of all these polygons, and \( K \) be the sum of all \( k_i \). Then the functions defined in (1)-(8) for all \( 1 \leq i \leq m \) can be computed in \( O(m^2 K^2) \) time in total.
Proof. As in the proof of Lemma 4, suppose $O_k$ is the overlay of the first $k$ polygons $l_1, \ldots, l_k$ and $O = O_m$. For each element (i.e., a face, edge or vertex) $c$ in overlay $O_i$, we introduce an additional vector to represent the relation between $c$ and polygons $l_1, l_2, \ldots, l_i$. When updating the overlay $O_i$ to $O_{i+1}$, we need to update these vectors correspondingly. Note that each $O_i$ has $O(K^2)$ elements. There are $O(K^2)$ vectors, each of which has $i \leq m$ indices. Therefore we need $O(mK^2)$ time to update all vectors for each overlay $O_i$, and thus $O(m^2K^2)$ time in total for $O$. The functions in (1)-(8) can be computed from the vectors for $O$ directly in $O(mK^2)$ time. In summary, it takes an additional $O(m^2K^2)$ time to compute all the functions. \hfill $\square$

Combined with Lemma 4, this shows that the overlay and the functions can be computed in $O(m^2K^2 \log K)$ time.

4.3. Solving basic RCC5 constraints involving polygonal landmarks

This subsection shows that CSPSAT$_s(B_{RCC5})$ is in P, provided that all landmarks are represented as polygons. We obtain this by giving a necessary and
sufficient condition for deciding the consistency of $\text{CSPSAT}_s(B_{RCC5})$ instances, which can be checked in polynomial time.

In what follows, we write an instance of $\text{CSPSAT}_s(B_{RCC5})$ or $\text{CSPSAT}_s(B_{RCC8})$ explicitly as $(V \cup L, \Gamma)$, where $V = \{v_1, v_2, \ldots, v_n\}$ is the set of unrestricted variables, $L = \{l_1, l_2, \ldots, l_m\}$ is the set of uniquely restricted variables. We write, for simplicity, $l_i$ for the only value (viz. a polygonal landmark) it takes and assume that the constraint between two landmarks is the actual relation between them.

4.3.1. A necessary and sufficient condition

Suppose $(V \cup L, \Gamma)$ is an instance of $\text{CSPSAT}_s(B_{RCC5})$, where $V = \{v_1, v_2, \ldots, v_n\}$ and $L = \{l_1, l_2, \ldots, l_m\}$. Let $\mathcal{O}$ be the overlay of polygons in $L$. Recall that for each $l_j$ and each face $f$ in $\mathcal{O}$, $f$ is either in $\mathcal{I}_{\text{FACE}}(l_j)$ (the set of faces contained in $l_j$) or in $\mathcal{E}_{\text{FACE}}(l_j)$ (the set of faces that lie outside $l_j$). Constraints in $\Gamma$ may impose similar relationships between $f$ and the variables in $V$. For a variable $v_i$, the constraints about $v_i$ may force $f$ to be part of $v_i$, or outside $v_i$. Precisely, $f$ is required to be part of $v_i$ if there is a landmark $l_j$ such that $f \in \mathcal{I}_{\text{FACE}}(l_j)$ and $l_j \mathcal{P} \mathcal{P} v_i$, and $f$ is required to lie outside $v_i$ if either $v_i \mathcal{D} \mathcal{R} l_j$ and $f \in \mathcal{I}_{\text{FACE}}(l_j)$, or $v_i \mathcal{P} \mathcal{P} l_j$ and $f \in \mathcal{E}_{\text{FACE}}(l_j)$. For each variable $v_i \in V$, we thus define $\mathcal{I}_{\text{FACE}}(v_i)$ and $\mathcal{E}_{\text{FACE}}(v_i)$ as follows:

\[\mathcal{I}_{\text{FACE}}(v_i) = \bigcup \{\mathcal{I}_{\text{FACE}}(l_j) : l_j \mathcal{P} \mathcal{P} v_i\}, \quad \text{(9)}\]
\[\mathcal{E}_{\text{FACE}}(v_i) = \bigcup \{\mathcal{I}_{\text{FACE}}(l_j) : v_i \mathcal{D} \mathcal{R} l_j\} \cup \bigcup \{\mathcal{E}_{\text{FACE}}(l_j) : v_i \mathcal{P} \mathcal{P} l_j\}. \quad \text{(10)}\]

**Example 3.** Suppose $(V \cup L, \Gamma)$ is an instance of $\text{CSPSAT}_s(B_{RCC5})$, where $V = \{v_1\}$ and $L = \{l_1, l_2, l_3\}$. Landmarks $l_1, l_2, l_3$ are shown in Figure 5(a). The constraints related to $v_1$ are specified as $l_1 \mathcal{P} \mathcal{P} v_1, l_2 \mathcal{P} \mathcal{P} v_1, v_1 \mathcal{P} \mathcal{O} l_3$. Then we have

\[\mathcal{I}_{\text{FACE}}(v_1) = \mathcal{I}_{\text{FACE}}(l_1) \cup \mathcal{I}_{\text{FACE}}(l_2) = \{f_1, f_2, f_3, f_4\}, \quad \mathcal{E}_{\text{FACE}}(v_1) = \emptyset.\]

The following proposition asserts that no face belongs to both $\mathcal{I}_{\text{FACE}}(v_i)$ and $\mathcal{E}_{\text{FACE}}(v_i)$, given that the constraint network is path-consistent.

**Proposition 8.** Suppose $(V \cup L, \Gamma)$ is an instance of $\text{CSPSAT}_s(B_{RCC5})$, where $V = \{v_1, v_2, \ldots, v_n\}$, $L = \{l_1, l_2, \ldots, l_m\}$, and each $l_i$ is a polygon. If $\Gamma$ is path-consistent, then $\mathcal{I}_{\text{FACE}}(v_i) \cap \mathcal{E}_{\text{FACE}}(v_i) = \emptyset$.

**Proof.** Assume $f \in \mathcal{I}_{\text{FACE}}(v_i) \cap \mathcal{E}_{\text{FACE}}(v_i)$. By definition there exist $l_j$ and $l_k$ such that $l_j \mathcal{P} \mathcal{P} v_i$ and $f \in \mathcal{I}_{\text{FACE}}(l_j)$, and either (i) $v_i \mathcal{D} \mathcal{R} l_k$ and $f \in \mathcal{I}_{\text{FACE}}(l_k)$ or (ii) $v_i \mathcal{P} \mathcal{P} l_k$ and $f \in \mathcal{E}_{\text{FACE}}(l_k)$. We show that both cases lead to a contradiction. For the first
case, we know \( f \subseteq l_j \cap l_k \), while the path-consistency of \( \Gamma \) implies that \( l_j \text{DR} l_k \) since \( l_j \text{PP} v_i \) and \( v_i \text{DR} l_k \). For the second case, we have \( f \subseteq l_j \) and \( f \cap l_k = \emptyset \), but the path-consistency of \( \Gamma \) implies \( l_j \text{PP} l_k \) since \( l_j \text{PP} v_i \) and \( v_i \text{PP} l_k \).

The following theorem provides a necessary and sufficient condition that decides \( \text{CSPSAT}_s(B_{RCC5}) \). Note that the condition only involves \( \text{FACE}, \mathcal{I}_{\text{FACE}}(l_j), \mathcal{E}_{\text{FACE}}(l_j), \mathcal{I}_{\text{FACE}}(v_i), \mathcal{E}_{\text{FACE}}(v_i) \), and constraints in the network, hence it can be checked after constructing the overlay of all landmarks and computing \( \mathcal{I}_{\text{FACE}}(v_i) \) and \( \mathcal{E}_{\text{FACE}}(v_i) \) for each \( v_i \).

**Theorem 8.** Suppose \( (V \cup L, \Gamma) \) is an instance of \( \text{CSPSAT}_s(B_{RCC5}) \), where \( V = \{v_1, v_2, \ldots, v_n\} \), \( L = \{l_1, l_2, \ldots, l_m\} \), and each \( l_i \) is a polygon. Then \( (V \cup L, \Gamma) \) is consistent, if and only if

- \( \Gamma \) is path-consistent.
- For any \( v_i \in V \), \( \mathcal{E}_{\text{FACE}}(v_i) \neq \text{FACE} \).
- All the conditions in Table 7 hold.

Conditions in Table 7 are very natural. For instance, the three conditions for constraint \( (v_i \text{PO} l_j) \) guarantee, respectively, that (i) \( v_i \) is not a proper subset of \( l_j \), (ii) \( v_i \) is not a proper superset of \( l_j \), and (iii) \( v_i \) may overlap with \( l_j \), i.e., not every face in \( \mathcal{I}_{\text{FACE}}(l_j) \) is excluded from \( v_i \). Consider Example 3 again.

<table>
<thead>
<tr>
<th>Constraint</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>( v_i \text{PO} l_j )</td>
<td>( \mathcal{E}<em>{\text{FACE}}(v_i) \cup \mathcal{E}</em>{\text{FACE}}(l_j) \neq \text{FACE}, \mathcal{I}<em>{\text{FACE}}(v_i) \cup \mathcal{E}</em>{\text{FACE}}(l_j) \neq \text{FACE}, \mathcal{I}<em>{\text{FACE}}(v_i) \cup \mathcal{E}</em>{\text{FACE}}(l_j) \neq \text{FACE} )</td>
</tr>
<tr>
<td>( v_i \text{PP} l_j )</td>
<td>( \mathcal{I}<em>{\text{FACE}}(v_i) \neq \mathcal{I}</em>{\text{FACE}}(l_j) )</td>
</tr>
<tr>
<td>( l_j \text{PP} v_i )</td>
<td>( \mathcal{E}<em>{\text{FACE}}(v_i) \neq \mathcal{E}</em>{\text{FACE}}(l_j) )</td>
</tr>
<tr>
<td>( v_i \text{PO} v_j )</td>
<td>( \mathcal{E}<em>{\text{FACE}}(v_i) \cup \mathcal{E}</em>{\text{FACE}}(v_j) \neq \text{FACE}, \mathcal{E}<em>{\text{FACE}}(v_i) \cup \mathcal{I}</em>{\text{FACE}}(v_j) \neq \text{FACE}, \mathcal{I}<em>{\text{FACE}}(v_i) \cup \mathcal{E}</em>{\text{FACE}}(v_j) \neq \text{FACE} )</td>
</tr>
<tr>
<td>( v_i \text{PP} v_j )</td>
<td>( \mathcal{I}<em>{\text{FACE}}(v_i) \cup \mathcal{E}</em>{\text{FACE}}(v_j) \neq \text{FACE} )</td>
</tr>
</tbody>
</table>

Table 7: Conditions for extended RCC5 constraint network
Example 3 (continued)

In this example, we have \( \mathcal{I}_{\text{FACE}}(v_1) = \{f_1, f_2, f_3, f_4\} \) and \( \mathcal{E}_{\text{FACE}}(v_1) = \emptyset \). Since \( \mathcal{E}_{\text{FACE}}(l_3) = \{f_0, f_1, f_3\} \), we know \( \mathcal{I}_{\text{FACE}}(v_1) \cup \mathcal{E}_{\text{FACE}}(l_3) = \{f_0, f_1, f_2, f_3, f_4\} = \text{FACE} \). Because \( (v_1 \text{PO} l_3) \in \Gamma \), Row 3 of Table 7 is violated. By Theorem 8 we know this instance is inconsistent.

We prove the necessity part here and leave the sufficiency part to Appendix A.

Proof of Theorem 8 (Necessity). Suppose \( (a_1, \ldots, a_n) \) is a solution of \( \Gamma \), where \( a_i \) is assigned to \( v_i \). Because each \( a_i \) has a nonempty interior, there exists at least one face \( f \) such that \( f \cap a_i \) is nonempty. Clearly, \( f \notin \mathcal{E}_{\text{FACE}}(v_i) \) since faces in \( \mathcal{E}_{\text{FACE}}(v_i) \) are all disjoint from \( a_i \) (otherwise a DR or PP constraint is violated). Therefore, \( \mathcal{E}_{\text{FACE}}(v_i) \neq \text{FACE} \).

If \( (v_i \text{PO} l_j) \in \Gamma \), then by assumption we have \( a_i \text{PO} l_j \). By definition of PO (see Table 3), we know that \( a_i \) and \( l_j \) have a common interior point. This implies that there exists a face \( f \) that contains an interior point of \( a_i \cap l_j \). Face \( f \) is neither in \( \mathcal{E}_{\text{FACE}}(v_i) \) nor in \( \mathcal{E}_{\text{FACE}}(l_j) \). That is, \( \mathcal{E}_{\text{FACE}}(v_i) \cup \mathcal{E}_{\text{FACE}}(l_j) \neq \text{FACE} \). Similarly, we know that neither \( \mathcal{E}_{\text{FACE}}(v_i) \cup \mathcal{I}_{\text{FACE}}(l_j) = \text{FACE} \) nor \( \mathcal{I}_{\text{FACE}}(v_i) \cup \mathcal{E}_{\text{FACE}}(l_j) = \text{FACE} \).

If \( (v_i \text{PP} l_j) \in \Gamma \), then \( a_i \text{PP} l_j \). Because \( l_j \) is the regularised union of all faces it contains, i.e. \( l_j = \bigcup \{f : f \in \mathcal{I}_{\text{FACE}}(l_j)\} \), we know there exists at least one face in \( \mathcal{I}_{\text{FACE}}(l_j) \) that is not in \( \mathcal{I}_{\text{FACE}}(v_i) \). This shows \( \mathcal{I}_{\text{FACE}}(v_i) \neq \mathcal{I}_{\text{FACE}}(l_j) \).

The remaining cases are either straightforward or similar to the above two cases.

Using Theorem 8, we are able to determine the consistency of any instance of \( \text{CSPSAT}_s(B_{RCCS}) \) in the following procedure:

- Compute \( \mathcal{I}_{\text{FACE}}(l_j) \) and \( \mathcal{E}_{\text{FACE}}(l_j) \) for each landmark \( l_j \) (this relies on the computation of the overlay planar subdivision \( \mathcal{O} \)).

- Compute \( \mathcal{I}_{\text{FACE}}(v_i) \) and \( \mathcal{E}_{\text{FACE}}(v_i) \) for each variable \( v_i \).

- Check the conditions in Theorem 8

Therefore the computational complexity of solving \( \text{CSPSAT}_s(B_{RCCS}) \) consists of three parts, corresponding to (i) computing \( \mathcal{I}_{\text{FACE}}(l_j) \) and \( \mathcal{E}_{\text{FACE}}(l_j) \), (ii) computing \( \mathcal{I}_{\text{FACE}}(v_i) \) and \( \mathcal{E}_{\text{FACE}}(v_i) \), and (iii) checking the conditions in Theorem 8. Putting them together, we come to the following theorem.
Theorem 9. Suppose \((V \cup L, \Gamma)\) is an instance of CSPSAT\(_{s}(B_{RCC5})\), where \(V = \{v_1, v_2, \ldots, v_n\}\), \(L = \{l_1, l_2, \ldots, l_m\}\), and each \(l_i\) is a polygon. Let \(k_i\) be the complexity of the planar subdivision induced by \(l_i\), and let \(K = \sum_{i=1}^{m} k_i\). Then the consistency of \((V \cup L, \Gamma)\) can be decided in \(O(n^3 + n^2K^2 + m^2K^2\log K)\) time.

Proof. Lemmas \(4\) and \(5\) show that \(O\), the overlay of all landmarks in \(L\), together with \(I_{\text{FACE}}(l_j)\) and \(E_{\text{FACE}}(l_j)\), can be computed in \(O(m^2K^2\log K)\) time. Moreover, all \(I_{\text{FACE}}(v_i)\) and \(E_{\text{FACE}}(v_i)\) can be computed in \(O(nmK^2)\) time by definition. For the conditions in Theorem \(8\), it takes \(O((n + m)^3)\) time to check the path-consistency of \(\Gamma\), and \(O(K^2)\) time to check each of the remaining \(O(n(n + m))\) conditions. Therefore, it takes \(O((n + m)^3 + n(n + m)K^2 + m^2K^2\log K)\) time to check all the conditions in Theorem \(8\). Summing these up, the consistency of \((V \cup L, \Gamma)\) can be determined in \(O((n + m)^3 + n(n + m)K^2 + m^2K^2\log K)\) time. Note that \(m \leq \sum_{i=1}^{m} k_i = K\). If \(m \leq n\), then \(O((m + n)^3) = O(n^3)\); if \(n \leq m\), then \(O((m + n)^3) = O(m^3)\). In both cases we have \(O((m + n)^3) = O(m^3 + n^3)\).

Similarly we have \(O(mnK^2) = O(m^2K^2 + n^2K^2)\). Therefore,

\[
O((n + m)^3 + n(n + m)K^2 + m^2K^2\log K) \\
= O((m^3 + n^3) + (n^2K^2 + m^2K^2) + m^2K^2\log K) \\
= O(n^3 + n^2K^2 + m^2K^2\log K)
\]

and the consistency of \((V \cup L, \Gamma)\) can be decided in \(O(n^3 + n^2K^2 + m^2K^2\log K)\) time. \(\square\)

As a direct consequence, we have

Theorem 10. Assuming that all landmarks are represented by polygons, then the consistency problem CSPSAT\(_{s}(B_{RCC5})\) is in \(P\), and the consistency problems CSPSAT\(_{f}(B_{RCC5})\), CSPSAT\(_{s}(RCC5)\), and CSPSAT\(_{f}(RCC5)\) are all \(NP\)-complete.

Proof. It follows directly from Theorem \(9\) that CSPSAT\(_{s}(B_{RCC5})\) is in \(P\). Moreover, by Proposition \(12\) we know that CSPSAT\(_{f}(B_{RCC5})\), CSPSAT\(_{s}(RCC5)\), and CSPSAT\(_{f}(RCC5)\) are all in \(NP\). The \(NP\)-hardness of CSPSAT\(_{f}(B_{RCC5})\) is proven in Proposition \(16\) and the \(NP\)-hardness of CSPSAT\(_{s}(RCC5)\) and CSPSAT\(_{f}(RCC5)\) follows from the \(NP\)-hardness of CSPSAT\(_{s}(RCC5)\). \(\square\)

Although CSPSAT\(_{s}(B_{RCC5})\) is in \(P\), we show in the next subsection that the consistency problem CSPSAT\(_{s}(B_{RCC8})\) is \(NP\)-hard.
4.4. Solving basic RCC8 constraints involving landmarks

This subsection investigates the consistency problem \( \text{CSPSAT}(\mathcal{B}_{\text{RCC8}}) \). First, we show that the problem is NP-hard by exploiting the fact that two polygons may have multiple ‘meeting’ points. Second, we show that the problem is still in NP by providing a polynomial nondeterministic algorithm. We then consider another interpretation of the RCC8 model by using a stronger connectedness. Under this interpretation, we show that \( \text{CSPSAT}(\mathcal{B}_{\text{RCC8}}) \) is still tractable.

4.4.1. The NP-hardness of \( \text{CSPSAT}(\mathcal{B}_{\text{RCC8}}) \)

We reduce the Graph 3-Colouring problem to the \( \text{CSPSAT}(\mathcal{B}_{\text{RCC8}}) \) problem.

**Proposition 9.** Assuming that all landmarks are represented by polygons, the consistency problem \( \text{CSPSAT}(\mathcal{B}_{\text{RCC8}}) \) is NP-hard.

**Proof.** Suppose \( G = (V, E) \) is a graph and \( V = \{v_0, \ldots, v_n\} \). We construct a \( \text{CSPSAT}(\mathcal{B}_{\text{RCC8}}) \) instance \( (V_G \cup L, \Gamma_G) \) as follows. The landmark set \( L \) is independent of the choice of \( G \) and contains the two polygons \( l \) and \( l' \) in Figure 6 (a). Note that \( l \) and \( l' \) are externally connected and have exactly three meeting points \( Q_0, Q_1 \) and \( Q_2 \), which are used to mimic the three colours in the Graph 3-Colouring problem.

The spatial variable set \( V_G \) is defined as \( \{u_0, u_1, \ldots, u_n\} \), where spatial variable \( u_i \) corresponds to vertex \( v_i \) in \( V \). The constraint network \( \Gamma_G \) is defined as follows.

\[
\Gamma_G = \{u_i \text{TPP}l\} \cup \{u_i \text{EC}l'\} \cup \{u_i \text{DC}u_j : (v_i, v_j) \in E\} \cup \{u_i \text{EC}u_j : (v_i, v_j) \notin E\}.
\]

We have finished the construction of the instance. The idea behind this reduction is as follows. Because \( l \) and \( l' \) have only three meeting points (viz. \( Q_0, Q_1 \) and \( Q_2 \)), each \( u_i \) can be connected to \( l' \) only via (one or more of) the three points \( Q_0, Q_1, Q_2 \). Determining which point \( u_i \) should occupy is essentially equivalent to
choosing a colour for vertex \( v_i \). For \( v_i \) and \( v_j \), if \( (v_i, v_j) \) is an edge in \( E \), then they cannot be coloured the same. Correspondingly, in such a case there is a constraint \( u_i \text{DC} u_j \), which forbids that \( u_i \) and \( u_j \) occupy the same point in \( \{Q_0, Q_1, Q_2\} \).

We now prove that \( G \) is 3-colourable iff \( (V_G \cup L, \Gamma_G) \) is consistent. Suppose \( \pi : V \rightarrow \{0, 1, 2\} \) is a valid 3-colouring of \( G \). We choose three candidate regions \( r_i^0, r_i^1 \) and \( r_i^2 \) for each variable \( u_i \), where \( r_i^0 \) is a triangle contained in \( l \) with a vertex being \( Q_p \). The candidate regions \( r_i^0, r_i^1, \ldots, r_i^n \) are externally connected at \( Q_p \), as illustrated in Figure 6(b). If we assign \( r_{\pi(v_i)}^i \) to \( u_i \), then all the DC constraints are satisfied. This is because, \( r_{\pi(v_i)}^i \) and \( r_{\pi(v_j)}^j \) are connected iff \( \pi(v_i) = \pi(v_j) \). This assignment, however, cannot fulfil all the EC constraints. For each unsatisfied EC constraint \( (u_i \text{EC} u_j) \), we introduce a pair of rectangles \( r_{ij} \) and \( r_{ij}' \), which are external and disjoint from any other rectangles \( r_{ij'}, r_{ij'}' \) and any triangle \( r_k^p \). We then add \( r_{ij} \) and \( r_{ij}' \) into, respectively, the candidate regions we have selected for \( u_i \) and \( u_j \). It is routine to verify that the modified assignment satisfies all constraints in \( \Gamma_G \) and hence is a solution of \( (V_G \cup L, \Gamma_G) \).

For the other direction, suppose \( (a_0, \ldots, a_n) \) is a solution of \( (V_G \cup L, \Gamma_G) \). Note that each \( a_i \) occupies at least one point in \( \{Q_0, Q_1, Q_2\} \). Define \( \pi : V \rightarrow \{0, 1, 2\} \) by assigning \( v_i \) the smallest index \( q \) such that \( a_i \) occupies \( Q_q \). The assignment \( \pi \) is a valid 3-colouring for graph \( G \). In fact, suppose \( \pi(v_i) = \pi(v_j) = p \). Then by definition both \( a_i \) and \( a_j \) occupies \( Q_p \). Hence \( (u_i \text{DC} u_j) \) is not a constraint in \( \Gamma_G \), which happens only when \( (v_i, v_j) \notin E \).

The reduction given above is polynomial because there are only two landmarks and \( |V| \) spatial variables in \( (V_G \cup L, \Gamma_G) \). Therefore, the consistency problem \( \text{CSPSAT}_s(\mathcal{B}_{RCC8}) \) is NP-hard. \( \square \)

In the next subsection we show that \( \text{CSPSAT}_s(\mathcal{B}_{RCC8}) \) is still in NP by designing a nondeterministic algorithm.

4.4.2. A nondeterministic algorithm for \( \text{CSPSAT}_s(\mathcal{B}_{RCC8}) \)

Suppose \( (V \cup L, \Gamma) \) is an instance of \( \text{CSPSAT}_s(\mathcal{B}_{RCC8}) \), where \( V = \{v_1, v_2, \ldots, v_n\} \), \( L = \{l_1, l_2, \ldots, l_m\} \), and each \( l_i \) is a polygon. We write \( \mathcal{O} \) for the overlay of all landmarks in \( L \), and define

\[
\mathcal{I}_{\text{FACE}}(l_i), \mathcal{I}_{\text{FACE}}(l_i), \mathcal{I}_{\text{EDGE}}(l_i), \mathcal{I}_{\text{EDGEX}}(l_i), \mathcal{I}_{\text{VTX}}(l_i), \mathcal{E}_{\text{VTX}}(l_i)
\]

as in \([1]-[8]\) for representing the topological relations between faces, edges, vertices in \( \mathcal{O} \) and landmarks in \( L \). As in the case of \( \text{CSPSAT}_s(\mathcal{B}_{RCC5}) \), we extend

31
these definitions from landmarks to variables. In the following, we say an edge \( e \) or a vertex \( v \) in \( \mathcal{O} \) is incident to a face \( f \) in \( \mathcal{O} \) if \( e \) or \( v \) is contained in the boundary of \( f \), and write

\[
S_{\text{FACE}}(v) = \{ f \in \text{FACE} : v \text{ is incident to } f \}, \quad \text{(11)}
\]

\[
S_{\text{FACE}}(e) = \{ f \in \text{FACE} : e \text{ is incident to } f \}. \quad \text{(12)}
\]

Note that \( S_{\text{FACE}}(e) \) has exactly two faces and \( S_{\text{FACE}}(v) \) may have more than two faces. These two functions can be directly obtained from the DCEL of the overlay.

Similarly as in the RCC5 case, we define \( I_{\text{FACE}}(v_i) \) as the set of faces that should be part of \( v_i \) and define \( E_{\text{FACE}}(v_i) \) as the set of faces that should be excluded from \( v_i \).

\[
I_{\text{FACE}}(v_i) = \bigcup \{ I_{\text{FACE}}(l_j) : l_j \text{ TPP } v_i \text{ or } l_j \text{ TPP } v_i \}, \quad \text{(13)}
\]

\[
E_{\text{FACE}}(v_i) = \bigcup \{ E_{\text{FACE}}(l_j) : v_i \text{ DCL } l_j \text{ or } v_i \text{ ECL } l_j \} \cup \bigcup \{ E_{\text{FACE}}(l_j) : v_i \text{ TPP } l_j \text{ or } v_i \text{ NTPP } l_j \}. \quad \text{(14)}
\]

Moreover, we define \( I_{\text{EDGE}}(v_i) \) as the set of edges that should lie in the interior of \( v_i \), \( E_{\text{EDGE}}(v_i) \) as the set of edges that should lie in the exterior of \( v_i \), and \( B_{\text{EDGE}}(v_i) \) as the set of edges that are required to be parts of the boundary of \( v_i \).

\[
I_{\text{EDGE}}(v_i) = \{ e \in \text{EDGE} : S_{\text{FACE}}(e) \subseteq I_{\text{FACE}}(v_i) \} \cup \bigcup \{ B_{\text{EDGE}}(l_j) : l_j \text{ NTPP } v_i \}; \quad \text{(15)}
\]

\[
E_{\text{EDGE}}(v_i) = \{ e \in \text{EDGE} : S_{\text{FACE}}(e) \not\subseteq E_{\text{FACE}}(v_i) \} \cup \bigcup \{ B_{\text{EDGE}}(l_j) : v_i \text{ DCL } l_j \text{ or } v_i \text{ NTPP } l_j \}; \quad \text{(16)}
\]

\[
B_{\text{EDGE}}(v_i) = \{ e \in \text{EDGE} : S_{\text{FACE}}(e) \cap I_{\text{FACE}}(v_i) \neq \emptyset, S_{\text{FACE}}(e) \cap E_{\text{FACE}}(v_i) \neq \emptyset \}. \quad \text{(17)}
\]

A brief explanation for the above notions follows. For an edge \( e \), if its two incident faces (i.e., faces in \( S_{\text{FACE}}(e) \)) are both in \( I_{\text{FACE}}(v_i) \) (\( E_{\text{FACE}}(v_i) \), resp.), then \( e \) itself should be in the interior (exterior, resp.) of \( v_i \). If one incident face of \( e \) is in \( I_{\text{FACE}}(v_i) \) while the other is in \( E_{\text{FACE}}(v_i) \), we know that \( e \) should be on the boundary of \( v_i \) (i.e. \( e \in B_{\text{EDGE}}(v_i) \)). Moreover, suppose \( e \) is a boundary edge of \( l_j \) (i.e. \( e \in B_{\text{EDGE}}(l_j) \)). If \( l_j \text{ NTPP } v_i \), then \( e \) should lie in the interior of \( v_i \) (i.e. \( e \in I_{\text{EDGE}}(v_i) \)); if \( v_i \text{ DCL } l_j \) or \( v_i \text{ NTPP } l_j \), then \( e \) should lie in the exterior of \( v_i \) (i.e. \( e \in E_{\text{EDGE}}(v_i) \)).
In the same way, we define $I_{VTX}(v_i)$, $E_{VTX}(v_i)$ and $B_{VTX}(v_i)$:

\[ I_{VTX}(v_i) = \{ v \in VTX : S_{FACE}(v) \subseteq I_{FACE}(v_i) \} \cup \{ B_{VTX}(l_j) : l_j \cap \text{B} \neq \emptyset \}, \]

\[ E_{VTX}(v_i) = \{ v \in VTX : S_{FACE}(v) \subseteq E_{FACE}(v_i) \} \cup \{ B_{VTX}(l_j) : v_i \cap \text{B} \neq \emptyset \}, \]

\[ B_{VTX}(v_i) = \{ v \in VTX : S_{FACE}(v) \cap I_{FACE}(v_i) \neq \emptyset, S_{FACE}(v) \cap E_{FACE}(v_i) \neq \emptyset \}. \]

Note that $S_{FACE}(v)$ may contain multiple faces while $S_{FACE}(v_i)$ contains exactly two faces.

**Proposition 10.** Suppose $(V \cup L, \Gamma)$ is an instance of $\text{CSPSAT}(\mathcal{B}_{RCC8})$, where $V = \{ v_1, v_2, \ldots, v_n \}$, $L = \{ l_1, l_2, \ldots, l_m \}$, and each $l_i$ is a polygon. If $\Gamma$ is path-consistent, then for each variable $v_i$ we have

1. $I_{FACE}(v_i) \cap E_{FACE}(v_i) = \emptyset$.
2. $I_{VTX}(v_i), E_{VTX}(v_i),$ and $B_{VTX}(v_i)$ are pairwise disjoint.
3. $I_{EDGE}(v_i), E_{EDGE}(v_i),$ and $B_{EDGE}(v_i)$ are pairwise disjoint.

**Proof.** (1) can be proven in the same way as Proposition 8. The remaining two can be similarly proven. Here we only show $I_{VTX}(v_i) \cap B_{VTX}(v_i) = \emptyset$ as an example.

Suppose otherwise that there exists a vertex $v \in VTX$ such that $v \in I_{VTX}(v_i)$ and $v \in B_{VTX}(v_i)$. Because $v \in B_{VTX}(v_i)$ we know there exist $f_1, f_2$ that are incident to $v$ and $f_1 \in I_{FACE}(v_i)$, $f_2 \in E_{FACE}(v_i)$. This implies that not all incident faces of $v$ are in $I_{FACE}(v_i)$. Therefore, by $v \in I_{VTX}(v_i)$, we know there exists a landmark $l_j$ such that $l_j \cap \text{B} \neq \emptyset$ and $v \in B_{VTX}(l_j)$.

As $f_2 \in E_{FACE}(v_i)$, by definition, we know that there exists a landmark $l_k$ such that either (i) $f_2 \in I_{FACE}(l_k)$ and $v_i \cap \text{B} \neq \emptyset$ or (ii) $f_2 \in E_{FACE}(l_k)$ and $v_i \cap \text{B} \neq \emptyset$. Note that $\Gamma$ is path-consistent. Case (i) implies that $f_2 \cap l_k$ and $l_j \cap \text{B} \neq \emptyset$. Because $v$ is incident to $f_2$, this shows that $v$ is in $l_k$. By $v \in B_{VTX}(l_j)$ we also have $v \in f_2$. This contradicts the conclusion $l_j \cap \text{B}$. In Case (ii), we have $l_j \cap \text{B} \neq \emptyset$ and $f_2 \cap l_k = \emptyset$. This also leads to a contradiction, because $l_j \cap \text{B}$ implies $v$ is in the interior of $l_k$, and $f_2 \cap l_k = \emptyset$ implies $v$ is not in the interior of $l_k$.

Therefore, we have $I_{VTX}(v_i) \cap B_{VTX}(v_i) = \emptyset$. \[\square\]

For convenience, we define

\[ P_{FACE}(v_i) = \text{FACE} - I_{FACE}(v_i) - E_{FACE}(v_i), \]

\[ P_{EDGE}(v_i) = \text{EDGE} - I_{EDGE}(v_i) - E_{EDGE}(v_i), \]

\[ P_{VTX}(v_i) = VTX - I_{VTX}(v_i) - E_{VTX}(v_i), \]

33
Table 8: Conditions for extended RCC5 constraint network

<table>
<thead>
<tr>
<th>Constraint</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_i \mathcal{E}C/l_j$</td>
<td>$\mathcal{E}<em>{\text{FACE}}(v_i) \cup \mathcal{E}</em>{\text{FACE}}(l_j) \neq \text{FACE}$, $\mathcal{I}<em>{\text{FACE}}(v_i) \cup \mathcal{E}</em>{\text{FACE}}(l_j) \neq \text{FACE}$, and (25)</td>
</tr>
<tr>
<td>$v_i \mathcal{P}O/l_j$</td>
<td>$\mathcal{E}<em>{\text{FACE}}(v_i) \cup \mathcal{I}</em>{\text{FACE}}(l_j) \neq \text{FACE}$, $\mathcal{I}<em>{\text{FACE}}(v_i) \cup \mathcal{E}</em>{\text{FACE}}(l_j) \neq \text{FACE}$, and (25)</td>
</tr>
<tr>
<td>$v_i \mathcal{T}P\mathcal{P}l_j$</td>
<td>$\mathcal{I}<em>{\text{FACE}}(v_i) \neq \mathcal{I}</em>{\text{FACE}}(l_j)$ and (25)</td>
</tr>
<tr>
<td>$l_j \mathcal{T}P\mathcal{P}v_i$</td>
<td>$\mathcal{E}<em>{\text{FACE}}(v_i) \neq \mathcal{E}</em>{\text{FACE}}(l_j)$ and (25)</td>
</tr>
<tr>
<td>$v_i \mathcal{D}Cv_j$</td>
<td>$S_i \cap S_j = \emptyset$</td>
</tr>
<tr>
<td>$v_i \mathcal{E}Cv_j$</td>
<td>$\mathcal{P}<em>{\text{FACE}}(v_j) \neq \emptyset$ or $\mathcal{I}</em>{\text{FACE}}(v_i) \neq \mathcal{I}_{\text{FACE}}(v_j)$, and (26)</td>
</tr>
</tbody>
</table>

where $\mathcal{P}$ denotes ‘pending’. We note that while $B_{\text{VTX}}(v_i)$ is the set of vertices that must lie on the boundary of $v_i$, $\mathcal{P}_{\text{VTX}}(v_i)$ contains all the vertices that may lie on the boundary of $v_i$. The pairwise disjointness of $\mathcal{I}_{\text{VTX}}(v_i), \mathcal{E}_{\text{VTX}}(v_i)$ and $B_{\text{VTX}}(v_i)$ implies $B_{\text{VTX}}(v_i) \subseteq \mathcal{P}_{\text{VTX}}(v_i)$.

Suppose $\Gamma$ is consistent and has a solution $\bar{v}_1, \bar{v}_2, \ldots, \bar{v}_n$. Write $\bar{S}_i$ for the set of vertices on the boundary of $\bar{v}_i$, i.e., $\bar{S}_i = \{v \in \text{VTX} : v \in \partial \bar{v}_i\}$. Then it is straightforward to show that

$$\bar{S}_i \cap \mathcal{I}_{\text{VTX}}(v_i) = \emptyset, \bar{S}_i \cap \mathcal{E}_{\text{VTX}}(v_i) = \emptyset, \text{ and } B_{\text{VTX}}(v_i) \subseteq \bar{S}_i \subseteq \mathcal{P}_{\text{VTX}}(v_i). \quad (24)$$

As we have seen in the reduction, determining $\bar{S}_i$ could be intractable. If all $\bar{S}_i$ are given in advance as a constraint for spatial variable $v_i$ (i.e., we explicitly specify whether vertex $v$ in the overlay is on the boundary of $v_i$ for all $v$ and $v_i$), then the existence of such a solution can be determined in polynomial time.

**Lemma 6.** Suppose $(V \cup L, \Gamma)$ is an instance of CSPSAT$_s(B_{\text{RCC5}})$, where $V = \{v_1, v_2, \ldots, v_n\}$, $L = \{l_1, l_2, \ldots, l_m\}$, and each $l_i$ is a polygon. Assume furthermore that $S_i$ is a subset of $\text{VTX}$ for $i = 1, 2, \ldots, n$. If $\Gamma$ is path-consistent, then $(V \cup L, \Gamma)$ has a solution $\{\bar{v}_1, \bar{v}_2, \ldots, \bar{v}_n\}$ such that $\partial \bar{v}_i \cap \text{VTX} = S_i$ if and only if

1. $\mathcal{E}_{\text{FACE}}(v_i) \neq \text{FACE}$ and $B_{\text{VTX}}(v_i) \subseteq S_i \subseteq \mathcal{P}_{\text{VTX}}(v_i)$ for each $v_i$.  


(b) All the conditions in Table 8 hold, where
\[ P_{\text{EDGE}}(v_i) \cap B_{\text{EDGE}}(l_j) \neq \emptyset \quad \text{or} \quad S_i \cap B_{\text{VTX}}(l_j) \neq \emptyset, \]  
\[ P_{\text{FACE}}(v_i) \cap P_{\text{FACE}}(v_j) \neq \emptyset \quad \text{or} \quad P_{\text{EDGE}}(v_i) \cap P_{\text{EDGE}}(v_j) \neq \emptyset \quad \text{or} \quad S_i \cap S_j \neq \emptyset. \]  
\[ (25) \quad (26) \]

Proof. See Appendix B.

Based on this result, we have the following theorem.

Theorem 11. Suppose all landmarks are represented by polygons. Then the consistency problem CSPSAT_s(B_{RCC8}) is NP-complete. Moreover, the consistency problems CSPSAT_f(B_{RCC8}), CSPSAT_s(RCC8), and CSPSAT_f(RCC8) are all NP-complete.

Proof. We propose a nondeterministic algorithm which solves CSPSAT_s(B_{RCC8}). The algorithm first guesses a configuration of \( S_i \) and uses it as an additional constraint, then determines the consistency by Lemma 11. Note that each \( S_i \) has \( O(K^2) \) points, which are polynomial in the input size. Thus guessing a configuration of \( S_i \) takes polynomial time. Meanwhile, checking all the conditions also takes polynomial time. Therefore, the extended consistency problem CSPSAT_s(B_{RCC8}) is in NP, and hence NP-complete as its NP-hardness has been confirmed in Proposition 9.

By Proposition 1 (ii) and (iii) we know CSPSAT_f(B_{RCC8}), CSPSAT_s(RCC8), and CSPSAT_f(RCC8) are all in NP. Meanwhile, they are also NP-hard because they all contain the NP-hard problem CSPSAT_s(B_{RCC8}) as a sub-problem. Therefore, they are all NP-complete.

Remark 2. Recall that in the reduction from the Graph 3-Colouring problem to CSPSAT_s(B_{RCC8}) the landmark \( l \) is a concave polygon which has three meeting points with landmark \( l' \) (see Figure 6(a)). This property of landmarks plays a critical role in designing the reduction. Another reduction from the 3-SAT problem to CSPSAT_s(B_{RCC8}), given in [29], also uses concave landmarks. Note that two convex polygons cannot have multiple isolated meeting points (i.e., they either have only one meeting point or share a line segment). One may conjecture that the consistency problem CSPSAT_s(B_{RCC8}) becomes tractable if all landmarks are represented as convex polygons. This, however, is not true.

In fact, a polynomial reduction from 3-SAT to CSPSAT_s(B_{RCC8}) exists even if all landmarks are represented by rectangles with edges parallel to the coordinate
axes. The reduction is more complicated than the reduction provided in the proof of Proposition 9. The main idea is, although landmarks are all convex regions, spatial variables can be interpreted as arbitrary regions, and we can constrain a spatial variable by using these rectangular landmarks in a way such that it may have multiple meeting points with some landmark. For example, suppose \( l_0, l_1, l_2 \) are three rectangles as shown in Figure 7, where \( l_1 \, \text{TPP} \, l_0, \, l_0 \, \text{EC} \, l_2 \) and \( l_1 \, \text{EC} \, l_2 \).

Assume that \( v \) is a spatial variable and \( v \, \text{TPP} \, l_0, \, v \, \text{EC} \, l_1 \) and \( v \, \text{EC} \, l_2 \). These constraints require \( v \) to contain (at least) one of the two points \( Q^+ \) and \( Q^- \), which may be used to simulate a propositional variable. Based on this observation, a reduction from 3-SAT can be devised. Therefore, the consistency problem \( \text{CSPSAT}_s(B_{RCC8}) \) remains NP-hard even for rectangular landmarks.

![Figure 7: Illustration for simulating a propositional variable using rectangular landmarks](image)

**Remark 3.** In practice, we may reduce the problem \( \text{CSPSAT}_s(B_{RCC8}) \) to SAT (i.e. deciding the satisfiability of propositional formulas in conjunctive normal form). As stated in the proof of Theorem 11, \( \text{CSPSAT}_s(B_{RCC8}) \) is equivalent to deciding whether there exist \( S_i \subseteq V_{TX} \) for each \( i \) such that all the conditions in Lemma 6 are satisfied. Note that, once the instance is given, the conditions in the lemma can be simplified (in polynomial time) into a set of conditions concerning \( S_i \) of the following forms: \( R \subseteq S_i \subseteq R', S_i \cap R \neq \emptyset, S_i \cap S_j = \emptyset, \) and \( S_i \cap S_j \neq \emptyset \), where \( R \) and \( R' \) are subsets of \( V_{TX} \) determined by the instance. For each \( S_i \) and each vertex \( v \in V_{TX} \), we introduce a propositional variable which is assigned true iff \( v \) is in \( S_i \). In this way, each condition in one of the above forms is transformed into a disjunction clause or a number of disjunction clauses, and thus a \( \text{CSPSAT}_s(B_{RCC8}) \) instance is transformed into an equivalent SAT instance. Therefore, \( \text{CSPSAT}_s(B_{RCC8}) \) can be reduced to SAT, which enables us to solve the problem by the well-developed SAT solvers.

The NP-hardness of \( \text{CSPSAT}_s(B_{RCC8}) \) is quite undesirable, as it is the simplest and most fundamental case of introducing landmarks to reasoning with RCC8. In
the following subsection, we show that the same problem becomes tractable if we interpret RCC8 relations by using a stronger connectedness relation.

4.4.3. RCC8 model based on strong connectedness

In the standard RCC8 model, two regions are considered to be connected if they have a common point. Consequently, two externally connected (EC) regions may share one or more isolated boundary points (see Figure 6(a)). In this subsection, we turn to another interpretation of RCC8, which uses a stronger version of connectedness: two regions are considered as connected if they share a common curve, where a curve is defined as a topological embedding of the closed interval [0,1] in the plane. As a result, two non-overlapping regions are externally connected iff their boundaries share at least a curve. Formally, we have

**Definition 8** (RCC8 algebra based on strong connectedness). Let \( U \) be the set of nonempty regular closed sets, or regions, in the real plane. The RCC8 algebra based on strong connectedness, written \( \text{RCC8}' \), is generated by the following eight topological relations

\[
\text{DC, EC, PO, EQ, TPP, NTPP, TPPi, NTPPi},
\]

where \( \text{TPPi} \) and \( \text{NTPPi} \) are the converses of \( \text{TPP} \) and \( \text{NTPP} \) respectively, and \( \text{EQ} \) is the identity relation, and for two regions \( a, b \),

- \( a \text{DC} b \) iff \( a \cap b \) does not contain any curve;
- \( a \text{EC} b \) iff \( a^c \cap b^c = \emptyset \) and \( a \cap b \) contains at least one curve;
- \( a \text{NTPP} b \) iff \( a \subset b \) and \( \partial a \cap \partial b \) does not contain any curve;
- \( a \text{TPP} b \) iff \( a \subset b \) and \( \partial a \cap \partial b \) contains at least one curve;
- \( a \text{PO} b \) iff \( a^c \cap b^c \neq \emptyset \) and \( a \not\subseteq b, b \not\subseteq a \).

It is easy to see that this connectedness relation (i.e. the complement of \( \text{DC} \)) is stronger than (i.e. contained in) the connectedness relation given in **Definition 4**.

Intuitively, the NP-hardness of \( \text{CSPSAT}_s(\mathcal{B}_{\text{RCC8}}) \) (for weak connectedness) is due to that there are exponentially many possibilities of \( S_i \) (the intersection of \( \text{VTX} \) and the boundary of \( v_i \)), since points in \( S_i \) may be evidences of \( \text{EC} \) constraints (cf. the reduction in **Section 4.4.1**). In the strong connectedness interpretation, however, isolated meeting points have no effects on RCC8 relations. Therefore \( S_i \) may be ignored safely and the problem \( \text{CSPSAT}_s(\mathcal{B}_{\text{RCC8}'}) \) becomes tractable, as shown in the following theorem.
Theorem 12. The consistency problem CSPSAT_s(B_{RCC^8'}) can be decided in polynomial time.

The computational complexity of CSPSAT_s(B_{RCC^8'}) is the same as that of CSPSAT_s(B_{RCC5}) (see Theorem 9), as the argument for RCC5 still applies here. Precisely, the consistency of an instance of CSPSAT_s(B_{RCC^8'}) can be decided by checking the conditions in Lemma 6 and neglecting all conditions involving $S_i$. That is, we discard the following conditions:

- the condition $B_{V_{TX}}(v_i) \subseteq S_i \subseteq P_{V_{TX}}(v_i)$ in condition (a);
- the condition $S_i \cap S_j = \emptyset$ whenever $(v_i; DC v_j) \in \Gamma$ in Row 4 of Table 8;
- the disjunct $S_i \cap B_{V_{TX}}(l_j) \neq \emptyset$ in (25);
- the disjunct $S_i \cap S_j \neq \emptyset$ in (26).

The above theorem can be proven by modifying the proof of Lemma 6 with a slightly different construction. The proof sketch is provided in Appendix C.

Remark 4. The strong connectedness introduced above has been considered in [3, 8]. In particular, in [3], Borgo, Guarino, and Masolo argued that the classical Whiteheadian connectedness may be considered too weak in many cases. For example, “a worm cannot pass from the interior of one apple to another, which touch just at a point, without becoming visible to the exterior – so from the worm’s point of view we might as well say that the apples are not ‘sufficiently’ connected.”

As far as consistency and realisations are concerned, Li [19] has shown that any consistent RCC8 network has a solution in any RCC model. The cubic realisation algorithm described there can be easily adapted to construct a solution in the RCC8 model based on strong connectedness. This implies in particular that an RCC8 network (without landmarks) has a solution in the RCC8 model with ‘weak’ connectedness iff it has a solution in the RCC8 model with ‘strong’ connectedness.

5. Conclusion and future work

One major difference between qualitative CSPs and classical CSPs is that the domain of a qualitative CSP is always infinite, while that of a classical CSP is usually finite. In this paper we proposed an extended framework for qualitative CSPs
that supports finite domains. In the extended framework, a spatial/temporal variable could take values from a finite domain or even a singleton. This reflects demands in applications such as urban planning and spatial query processing where additional knowledge about variables may be available. We believe this extension is necessary to bring QSTR closer to real-world applications.

We then investigated the computational complexity of solving the extended consistency problem for five very important qualitative calculi, viz. PA, IA, CRA, RCC5 and RCC8. The results were summarised in Table 4, where for each calculus, we determined whether each of the four variants of the consistency problem is in P or NP-complete. Recall that the classical consistency problem is NP-complete for IA, CRA, RCC5 and RCC8. This shows that, in general, the expressiveness of the extended framework of qualitative CSP does not incur additional cost in computational complexity for these calculi. Under practical assumptions, we also provided efficient algorithms for solving basic constraints involving landmarks for all these calculi.

While this paper introduces landmarks in qualitative CSPs, there is a related work in classical CSPs. Recently, Bulatov [5] has given a full classification of computational complexity for conservative constraint satisfaction problems with finite values, in which the set of values for each individual variable can be restricted arbitrarily. The solving algorithm and the proofs given there heavily use the algebraic approach to (classical) CSP developed in [17, 6]. One interesting future research direction will be investigating the possibility of applying the solving algorithm given in [5], and, more generally, the algebraic approach, to solving qualitative CSPs involving landmarks. We refer the reader to [2] for recent progresses of applying the algebraic approach for attacking qualitative CSPs.

In this paper, we have confined ourselves to the five most important qualitative calculi, which are all binary calculi. The framework can be straightforwardly extended to any other qualitative calculus, binary or ternary, but the computational complexity has to be examined case by case. Take the ternary calculus LR [23] as an example. It has been shown that reasoning with complete basic and landmark-free LR networks is already at least NP-hard and its NP-membership is still open [46]. As a consequence, reasoning with complete basic LR networks involving landmarks is also NP-hard. Another direction of future research will be investigating the computation complexity for other well-known calculi, individually or combined together. Because most of these consistency problems are at least NP-hard, it is also necessary to develop either approximate methods or practical methods (e.g. those in [37, 18]) for solving qualitative (binary or ternary) CSPs involving landmarks.
Acknowledgements

We thank the three anonymous referees and the handling associate editor for their invaluable suggestions, and, especially, for their suggestion that using Graph 3-Colouring instead of 3-SAT for proving the NP-hardness. We also thank Steven Schockaert for suggesting us the MacDonald’s example and thank Jochen Renz and Tony Cohn for fruitful discussions on our joint ARC Discovery Project “Approximate Reasoning with Qualitative Spatial Constraints Involving Landmarks.”

This work was partially supported by Australian Research Council (FT0990811, DP120104159) and National Natural Science Foundation of China (61133011, 61228305).

Appendix A. Proof of Theorem 8 (Sufficiency)

The sufficiency part is proven by a realisation algorithm which generates a solution of the constraint network. The algorithm is similar to the classical realisation algorithm introduced in [13, 22]. We first construct for each variable $v_i$ a region $a_i$ such that $\{a_1, a_2, \ldots, a_n\}$ satisfies all except the PP constraints, and then construct regions $\{c_1, c_2, \ldots, c_n\}$ which is a solution of $\Gamma$.

For each variable $v_i$, we define

$$\mathcal{P}_{\text{FACE}}(v_i) = \text{FACE} - \mathcal{I}_{\text{FACE}}(v_i) - \mathcal{E}_{\text{FACE}}(v_i).$$  \hfill (A.1)

A number of ‘base regions’ are necessary in the construction of $\{a_1, a_2, \ldots, a_n\}$. Base regions are arbitrarily selected, as long as they are pairwise disjoint polygons and are so small that their union does not contain any face. We use $X_i$ to denote the set of base regions being selected for variable $v_i$. The construction is as follows, where each $X_i$ is initialised as the empty set.

1. For each face $f \in \mathcal{P}_{\text{FACE}}(v_i)$, select a base region contained in $f$ and put it into $X_i$.

2. For any $i < j$ such that $(v_i, \text{PO}v_j) \in \Gamma$ and $\mathcal{P}_{\text{FACE}}(v_i) \cap \mathcal{P}_{\text{FACE}}(v_j) \neq \emptyset$, select a face $\hat{f}$ in $\mathcal{P}_{\text{FACE}}(v_i) \cap \mathcal{P}_{\text{FACE}}(v_j)$ and a base region contained in $\hat{f}$. Put the base region into both $X_i$ and $X_j$.

3. For each $i$, let $a_i = \bigcup X_i$.

4. For each $i$, let $b_i = a_i \cup \{a_j : (v_j, \text{PP}v_i) \in \Gamma\}$.

5. For each $i$, let $c_i = b_i \cup \{l_j : (l_j, \text{PP}v_i) \in \Gamma\}$.

40
Lemma 7. Suppose \((V \cup L, \Gamma)\) is an instance of \(\text{CSPSAT}_4(B_{\text{RCC5}})\), where \(V = \{v_1, v_2, \ldots, v_n\}\), \(L = \{l_1, l_2, \ldots, l_m\}\), and each \(l_i\) is a polygon. Suppose \(\Gamma\) is path-consistent. Assume that \(a_i, b_i, c_i\) \((1 \leq i \leq n)\) are as in the construction given above. Then for each face \(f \in \text{FACE}\) we have

- \(f \in \mathcal{I}_{\text{FACE}}(v_i)\) iff \(f \subseteq c_i\).
- \(f \in \mathcal{E}_{\text{FACE}}(v_i)\) iff \(f \cap c_i = \emptyset\).
- \(f \in \mathcal{P}_{\text{FACE}}(v_i)\) iff \(f \nsubseteq c_i\) and \(f \cap c_i \neq \emptyset\).

Proof. We first prove the necessity part.

Suppose \(f \in \mathcal{I}_{\text{FACE}}(v_i)\). There exists a landmark \(l\) such that \(f \in \mathcal{I}_{\text{FACE}}(l)\) and \(l PP v_i\). Because \(l \subseteq c_i\), the first statement holds directly.

Assume \(f \in \mathcal{E}_{\text{FACE}}(v_i)\). Because each base region in \(X_i\) is contained in a face in \(\mathcal{P}_{\text{FACE}}(v_i)\), we know that \(f \cap a_i = \emptyset\). Suppose \((v_j PP v_i) \in \Gamma\). By the definition of \(\mathcal{E}_{\text{FACE}}(f)\) and the path-consistency of \(\Gamma\), it is direct to prove that \(f\) is also in \(\mathcal{E}_{\text{FACE}}(v_j)\). Therefore we have \(f \cap a_j = \emptyset\), and thus \(f \cap b_i = \emptyset\) by the construction of \(b_i\). Similarly, for any landmark \(l\) such that \((l PP v_i) \in \Gamma\), we can prove that \(f \cap l = \emptyset\). Therefore, we have \(f \cap c_i = \emptyset\).

Now assume \(f \in \mathcal{P}_{\text{FACE}}(v_i)\). Clearly we have \(f \cap a_i \neq \emptyset\) because \(X_i\) has a base region contained in \(f\). We only need to prove \(f \nsubseteq c_i\). By the selection of base regions, \(f\) is not contained in the union of all base regions, and hence it is not contained in \(b_i\). Moreover, for any landmark \(l_j\), if \((l_j PP v_i) \in \Gamma\), then \(f \in \mathcal{E}_{\text{FACE}}(l_j)\) (otherwise, \(f \in \mathcal{I}_{\text{FACE}}(l_j) \subseteq \mathcal{I}_{\text{FACE}}(v_i)\)). That is to say, \(f\) is disjoint with \(l_j\). Therefore, \(f \nsubseteq c_i\).

The sufficiency part follows from \(\mathcal{I}_{\text{FACE}}(v_i) \cup \mathcal{E}_{\text{FACE}}(v_i) \cup \mathcal{P}_{\text{FACE}}(v_i) = \text{FACE}\). □

Corollary 2. Let \((V \cup L, \Gamma)\) and \(c_i\) be as in Lemma 7. Furthermore, suppose \((V \cup L, \Gamma)\) satisfies all the conditions in Theorem 8. Then \(\{c_1, c_2, \ldots, c_n\}\) satisfies all the constraints in \(\Gamma\) of the form \(v_i \alpha l_j\).

Proof. Because \((V \cup L, \Gamma)\) satisfies the conditions in Theorem 8, we know in particular that \(\mathcal{E}_{\text{FACE}}(v_i) \neq \text{FACE}\) for each \(1 \leq i \leq n\). That is, there exists a face \(f\) in \(\mathcal{I}_{\text{FACE}}(v_i) \cup \mathcal{P}_{\text{FACE}}(v_i)\). By Lemma 7 this implies that each \(c_i\) is nonempty.

1. If \((v_i PP l_j) \in \Gamma\), then we have \(\mathcal{E}_{\text{FACE}}(l_j) \subseteq \mathcal{E}_{\text{FACE}}(v_i)\) by (10). Lemma 7 directly implies that \(c_i \subseteq l_j\). Because \(\mathcal{I}_{\text{FACE}}(v_i) \neq \mathcal{I}_{\text{FACE}}(l_j)\) (Row 2 in Table 7), there exists a face \(f\) which is in \(\mathcal{I}_{\text{FACE}}(l_j)\) but not in \(\mathcal{I}_{\text{FACE}}(v_i)\). By Lemma 7 \(f\) is not contained in \(c_i\). Therefore, \(c_i \subseteq l_j\), i.e. \(c_i PP l_j\).
(2) If \((l_j PP v_i) \in \Gamma\), clearly we have \(l_j \subseteq c_i\). Because \(E_{FACE}(v_i) \neq E_{FACE}(l_j)\) (Row 3 in Table 7) and \(E_{FACE}(v_i) \subseteq \text{FACE} - I_{FACE}(v_i) \subseteq \text{FACE} - I_{FACE}(l_j) = E_{FACE}(l_j)\), we know that \(E_{FACE}(v_i) \subset E_{FACE}(l_j)\), i.e. there exists a face \(\tilde{f}\) in \(E_{FACE}(l_j)\) but not in \(E_{FACE}(v_i)\). Therefore \(\tilde{f} \cap l_j = \emptyset\) and \(\tilde{f} \cap c_i \neq \emptyset\). That is, \(l_j \subset c_i\), i.e. \(l_j PP c_i\).

(3) If \((v_i DR l_j) \in \Gamma\), then we have \(I_{FACE}(l_j) \subseteq E_{FACE}(v_i)\). Lemma 7 directly implies that \(c_i \cap l_j = \emptyset\), i.e. \(c_i DR l_j\).

(4) If \((v_i PO l_j) \in \Gamma\), then by Row 1 in Table 7, we know that \(E_{FACE}(v_i) \cup E_{FACE}(l_j) \neq \text{FACE}\). That is, there exists a face \(f\) such that \(f \notin E_{FACE}(v_i)\) and \(f \notin E_{FACE}(l_j)\) (hence \(f \in I_{FACE}(l_j)\)). Therefore \(f \subseteq l_j\) and \(f \cap c_i \neq \emptyset\) by Lemma 7 and hence \(c_i\) overlaps \(l_j\), i.e. they have a common interior point. It can be proven that \(c_i \not\subset l_j\) and \(l_j \not\subset c_i\) as in the first two cases above. Therefore, \(c_i PO l_j\) holds.

We next prove that \(\{c_1, \ldots, c_n\}\) is a solution of \(\Gamma\).

**Lemma 8.** Let \((V \cup L, \Gamma)\) and \(c_i\) be as in Corollary 2. Then \(\{c_1, \ldots, c_n\}\) is a solution of \((V \cup L, \Gamma)\).

**Proof.** We only need to prove that constraints of the form \((v_i \triangleright v_j)\) are satisfied.

(1) If \((v_i PP v_j) \in \Gamma\), it can be proven that \(b_i \subseteq b_j\) and \(c_i \subseteq c_j\) by the path-consistency of \(\Gamma\). We next prove \(c_i \not\subset c_j\). By \(I_{FACE}(v_i) \cup E_{FACE}(v_j) \neq \text{FACE}\) (last row in Table 7), there exists a face \(f\) that is in neither \(I_{FACE}(v_i)\) nor \(E_{FACE}(v_j)\). Therefore \(f\) is either in \(E_{FACE}(v_i)\) or \(P_{FACE}(v_i)\). If \(f \notin E_{FACE}(v_i)\), then \(f \cap c_i = \emptyset\). By Lemma 2 and \(f \notin E_{FACE}(v_j)\), we also know \(f \cap c_j \neq \emptyset\), and thus \(c_i \not\subset c_j\). Now suppose \(f \in P_{FACE}(v_i)\). By Lemma 7 we have \(f \subset c_i\). Note that \(f\) is in either \(I_{FACE}(v_j)\) or \(P_{FACE}(v_j)\). In the first case, we have \(f \subseteq c_j\) and thus \(c_i \not\subset c_j\). In the second case, by the construction of \(X_j\) we know that there exists some base region \(r\) contained in \(f\) that belongs to \(X_j\) only. Therefore \(r\) is disjoint with \(a_i\) and hence disjoint with \(b_i\). Moreover, \(r\) cannot be contained in \(c_j\). Otherwise, there must exist some landmark \(l\) such that \(l PP v_i\) and \(r \subseteq l\). This implies that \(f \in I_{FACE}(l)\), which further implies \(f \in I_{FACE}(v_i)\), a contradiction. Therefore, we have \(r \not\subset c_i\) and \(r \subseteq c_j\) and thus \(c_i \not\subset c_j\).

(2) If \((v_i DR v_j) \in \Gamma\), we show \(c_i \cap c_j = \emptyset\). By construction we have \(a_i \cap a_j = \emptyset\), because \(X_i \cap X_j = \emptyset\) unless \((v_i PO v_j) \in \Gamma\). Note that \((v_i PP v_j) \in \Gamma\) implies \((v_i DR v_j) \in \Gamma\) by path-consistency. Therefore, we also have \(a_k \cap a_j = \emptyset\). By the construction of \(b_i\) we know \(b_i \cap a_j = \emptyset\). Similarly we can prove that \(b_i \cap b_j = \emptyset\). In the same way, it can be further proven that \(c_i \cap c_j = \emptyset\), i.e. \(c_i DR c_j\).

(3) If \((v_i PO v_j) \in \Gamma\), we first show that \(c_i\) overlaps \(c_j\). By \(E_{FACE}(v_i) \cup E_{FACE}(v_j) \neq \text{FACE}\) (Row 6 in Table 7), there exists a face \(f\) such that \(f \notin E_{FACE}(v_i)\) and \(f \notin E_{FACE}(v_j)\). In other words, we have \(f \in I_{FACE}(v_i) \cup P_{FACE}(v_i)\) and \(f \in I_{FACE}(v_j) \cup P_{FACE}(v_j)\).
Appendix B. Proof of Lemma 6

Appendix B.1. Necessity

Suppose \( \{\bar{v}_1, \bar{v}_2, \ldots, \bar{v}_n\} \) is a solution of \( \Gamma \) and \( \bar{v}_i \cap VTX = S_i \) for each \( i \). By the definitions of \( B_{VTX}(v_i) \) and \( P_{VTX}(v_i) \), it is straightforward to show that \( B_{VTX}(v_i) \subseteq S_i \subseteq P_{VTX}(v_i) \). Similarly to the RCC5 case, we can prove that \( E_{FACE}(v_i) \neq \text{FACE} \) for any \( v_i \in V \). We first prove the following lemmas.

Lemma 9. Suppose \( \{\bar{v}_1, \bar{v}_2, \ldots, \bar{v}_n\} \) is a solution of \( \Gamma \), then for any \( \bar{v}_1 \) we have

(i) \( \bar{v}_i \cap \bar{v}_i = \emptyset \) for any face \( \bar{v} \in I_{FACE}(v_i) \);

(ii) \( \bar{v} \cap \bar{v}_i = \emptyset \) for any face \( \bar{v} \in E_{FACE}(v_i) \);

(iii) \( \bar{v} \subseteq (\bar{v}_i) \) for any edge \( \bar{v} \in I_{EDGE}(v_i) \);

(iv) \( \bar{v} \cap \bar{v}_i = \emptyset \) for any edge \( \bar{v} \in E_{EDGE}(v_i) \);

(v) \( \bar{v} \in (\bar{v}_i) \) for any vertex \( \bar{v} \in I_{VTX}(v_i) \);

(vi) \( \bar{v} \not\in \bar{v}_i \) for any vertex \( \bar{v} \in E_{VTX}(v_i) \).

Proof. For (i), by the definition of \( I_{FACE}(v_i) \) (see (9)), there exists a landmark \( l_k \) such that \( \bar{v} \in I_{FACE}(l_k) \) and \( l_k \cap TP(v_i) \) or \( l_k \cap NTPP(v_i) \). Thus we have \( \bar{v} \subseteq l_k^{0} \) and \( l_k \subseteq \bar{v}_i \), and, therefore, \( \bar{v} \subseteq (\bar{v}_i)^{0} \). Similarly we have \( \bar{v} \cap \bar{v}_i = \emptyset \) for any \( \bar{v} \in E_{FACE}(v_i) \).

For (ii), by the definition of \( I_{EDGE}(v_i) \) (see (15)), we have either \( S_{FACE}(e) \subseteq I_{FACE}(v_i) \), or \( e \in B_{EDGE}(l_k) \) for some landmark \( l_k \) with \( l_k \cap TP(v_i) \). In the first case, because the two incident faces of \( e \) are both in \( I_{FACE}(v_i) \), they are contained in
the interior of $\bar{v}_i$. Because $e$ is the common boundary of its two incident faces, we know $e$ is also contained in $(\bar{v}_i)^o$. In the second case, we have $e \subseteq l_k \subseteq (\bar{v}_i)^o$. Therefore $e \subseteq (\bar{v}_i)^o$ holds in both cases. Similarly we have $e' \cap \bar{v}_i = \emptyset$ for any edge $e'$ in $E_{\text{EDGE}}(v_i)$.

(v) and (vi) can be proven in the same way. \hfill \Box

**Lemma 10.** Suppose $\{\bar{v}_1, \bar{v}_2, \ldots, \bar{v}_n\}$ is a solution of $\Gamma$, and $S_i = \bar{v}_i \cap \text{VTX}$ for each $i$. Then for any $v_i$ and $l_j$, if $(v_i E\Gamma_l)_j$, $(v_i T\Pi_p l_j)$ or $(v_i T\Pi_l l_j)$ is a constraint in $\Gamma$, then (25) holds; for any $v_i$ and $v_j$, if $(v_i E\Gamma v) j$ or $(v_i T\Pi p v) j \in \Gamma$ is a constraint in $\Gamma$, then (26) holds.

**Proof.** Suppose one of $(v_i E\Gamma_l)_j$, $(v_i T\Pi_p l_j)$, and $(v_i T\Pi_l l_j)$ is a constraint in $\Gamma$. We show (25) holds. Because $\{\bar{v}_1, \bar{v}_2, \ldots, \bar{v}_n\}$ is a solution, we know that $\bar{v}_i$ and $l_j$ have a common boundary point, say $P$. It is clear that $P$ is either a vertex in $B_{\text{VTX}}(l_j)$, or on an edge $e \in B_{\text{EDGE}}(l_j)$. In the first case, we have $P \in \partial \bar{v}_i \cap \text{VTX} = S_i$. Therefore $P \in S_i \cap B_{\text{VTX}}(l_j)$ and thus (25) is satisfied. In the second case, because $P \in e$ and $P \in \partial \bar{v}_i$, we know edge $e$ cannot be in the interior of $\bar{v}_i$ or in the exterior of $\bar{v}_i$. By Lemma 9 $e$ is in neither $L_{\text{EDGE}}(v_i)$ nor $E_{\text{EDGE}}(v_i)$, hence $e \in P_{\text{EDGE}}(v_i)$. Therefore we have $P_{\text{EDGE}}(v_i) \cap B_{\text{EDGE}}(l_j) = \emptyset$ and thus (25) is also satisfied.

The other part of the lemma can be proven similarly. \hfill \Box

The necessity of conditions in Table 8 can then be proven straightforwardly.

**Appendix B.2. Sufficiency**

Suppose $(V \cup L, \Gamma)$ and $S_i$ $(i = 1, \ldots, n)$ satisfy the conditions in Lemma 6, we construct a solution $\{\bar{v}_1, \ldots, \bar{v}_n\}$ of $\Gamma$ such that $S_i = \partial \bar{v}_i \cap \text{VTX}$. The construction procedure is similar to that in [19, 22]. For each spatial variable $v_i$, we select a set of small triangles, denoted by $X_i$, in the following way.

- For each face $f \in P_{\text{FACE}}(v_i)$, select a small triangle in $f$ and put it in $X_i$, see Figure B.8(a).

- For each vertex $v \in S_i - B_{\text{VTX}}(v_i) \subseteq P_{\text{VTX}}(v_i) - B_{\text{VTX}}(v_i)$, by Proposition 10 we know that $v$ is not in $B_{\text{VTX}}(v_i) \cup L_{\text{VTX}}(v_i) \cup E_{\text{VTX}}(v_i)$. We have that $S_{\text{FACE}}(v) \cap P_{\text{FACE}}(v_i) = \emptyset$. Otherwise, $S_{\text{FACE}}(v)$ is contained in $L_{\text{FACE}}(v_i) \cup E_{\text{FACE}}(v_i)$, which implies that $v$ is either in $L_{\text{VTX}}(v_i)$, or in $E_{\text{VTX}}(v_i)$, or in $B_{\text{VTX}}(v_i)$. We select a face $f$ from $S_{\text{FACE}}(v) \cap P_{\text{FACE}}(v_i)$, and select a small triangle in $f$ that contains $v$. Put the triangle in $X_i$, see Figure B.8(b).


• If \( v_1 \in EC_{l_1} \) is in \( \Gamma \), then by Table 8, we have either \( \mathcal{P}_{EDGE}(v_1) \cap B_{EDGE}(l_1) \neq \emptyset \) or \( S_i \cap B_{VTX}(l_1) \neq \emptyset \) (i.e. (25)). If \( S_i \cap B_{VTX}(l_1) \neq \emptyset \), do nothing. Otherwise, we select an edge \( e \) from \( \mathcal{P}_{EDGE}(v_1) \cap B_{EDGE}(l_1) \). Let \( f \) and \( f' \) be the two incident faces of \( e \) such that \( f \in I_{FACE}(l_1) \) and \( f' \in E_{FACE}(l_1) \). By definition, we know \( f \in E_{FACE}(v_i) \). We note that \( f' \) cannot be in \( E_{FACE}(v_i) \). This is because, otherwise, we have \( S_{FACE}(e) = \{ f, f' \} \subseteq E_{FACE}(v_i) \) and hence \( e \in E_{EDGE}(v_i) \), which contradicts the assumption that \( e \in \mathcal{P}_{EDGE}(v_i) \). If \( f' \in E_{FACE}(v_i) \), do nothing. If \( f' \in P_{FACE}(v_i) \), select a triangle in face \( f' \) with one edge on \( e \) and put it in \( X_i \), see Figure B.8(c).

• If \( v_1 \in TPP_{l_1} \) is in \( \Gamma \), then by Table 8, we also have \( \mathcal{P}_{EDGE}(v_1) \cap B_{EDGE}(l_1) \neq \emptyset \) or \( S_i \cap B_{VTX}(l_1) \neq \emptyset \) (i.e. (25)). If \( S_i \cap B_{VTX}(l_1) \neq \emptyset \), do nothing. Otherwise, select an edge \( e \) from \( \mathcal{P}_{EDGE}(v_1) \cap B_{EDGE}(l_1) \). Let \( f \) and \( f' \) be the two incident faces of \( e \) such that \( f \in I_{FACE}(l_1) \) and \( f' \in E_{FACE}(l_1) \). By definition, we know \( f' \in E_{FACE}(v_i) \). Similar to the case of \( v_1 \in EC_{l_1} \), \( f \) cannot be in \( E_{FACE}(v_i) \). If \( f \in I_{FACE}(v_i) \), do nothing. If \( f \in P_{FACE}(v_i) \), select a triangle in face \( f \) with one edge on \( e \) and put it in \( X_i \).

• If \( v_1 \in EC_{v_j} \) is in \( \Gamma \), then by Table 8, we have \( \mathcal{P}_{FACE}(v_i) \cap \mathcal{P}_{FACE}(v_j) \neq \emptyset \), or \( \mathcal{P}_{EDGE}(v_i) \cap \mathcal{P}_{FACE}(v_j) \neq \emptyset \), or \( S_i \cap S_j \neq \emptyset \). If \( S_i \cap S_j \neq \emptyset \), do nothing. If \( S_i \cap S_j = \emptyset \) and \( \mathcal{P}_{FACE}(v_i) \cap \mathcal{P}_{FACE}(v_j) \neq \emptyset \), select a face \( f \in \mathcal{P}_{FACE}(v_i) \cap \mathcal{P}_{FACE}(v_j) \) and two externally connected triangles in \( f \). Put one triangle in \( X_i \) and put the other in \( X_j \), see Figure B.8(d). If \( S_i \cap S_j = \emptyset \), \( \mathcal{P}_{FACE}(v_i) \cap \mathcal{P}_{FACE}(v_j) = \emptyset \), and \( \mathcal{P}_{EDGE}(v_i) \cap \mathcal{P}_{EDGE}(v_j) \neq \emptyset \), then select edge \( e \in \mathcal{P}_{EDGE}(v_i) \cap \mathcal{P}_{EDGE}(v_j) \). Suppose \( f \) and \( f' \) are the two incident faces of \( e \). We have four subcases depending on whether \( e \) is in \( B_{EDGE}(v_i) \) and \( B_{EDGE}(v_j) \).

  - If \( e \in B_{EDGE}(v_i) \) and \( e \in B_{EDGE}(v_j) \), then do nothing.
  - If \( e \notin B_{EDGE}(v_i) \) and \( e \in B_{EDGE}(v_j) \), suppose \( f \in I_{FACE}(v_i) \) and \( f' \in E_{FACE}(v_i) \). Select a triangle in \( f' \) with one edge on \( e \) and put it in \( X_j \).
  - If \( e \notin B_{EDGE}(v_i) \) and \( e \notin B_{EDGE}(v_j) \), suppose \( f \in I_{FACE}(v_j) \) and \( f' \in E_{FACE}(v_j) \). Select a triangle in \( f' \) with one edge on \( e \) and put it in \( X_i \).
  - If \( e \notin B_{EDGE}(v_i) \) and \( e \notin B_{EDGE}(v_j) \), then select two triangles in \( f \) and \( f' \) respectively such that the triangles have a common edge on \( e \), see Figure B.8(e).

• If \( v_1 \in TPP_{v_j} \) is in \( \Gamma \), then by Table 8, we also have \( \mathcal{P}_{FACE}(v_i) \cap \mathcal{P}_{FACE}(v_j) \neq \emptyset \), or \( \mathcal{P}_{EDGE}(v_i) \cap \mathcal{P}_{EDGE}(v_j) \neq \emptyset \), or \( S_i \cap S_j \neq \emptyset \). If \( S_i \cap S_j = \emptyset \), then do
nothing. If \( S_i \cap S_j = \emptyset \) and \( P_{\text{FACE}}(v_i) \cap P_{\text{FACE}}(v_j) \neq \emptyset \), then select a face \( f \in P_{\text{FACE}}(v_i) \cap P_{\text{FACE}}(v_j) \) and one triangle in \( f \). Put the triangle in both \( X_i \) and \( X_j \). If \( S_i \cap S_j = \emptyset \), \( P_{\text{FACE}}(v_i) \cap P_{\text{FACE}}(v_j) = \emptyset \), and \( P_{\text{EDGE}}(v_i) \cap P_{\text{EDGE}}(v_j) \neq \emptyset \), then select an edge \( e \in P_{\text{EDGE}}(v_i) \cap P_{\text{EDGE}}(v_j) \). Suppose \( f \) and \( f' \) are the two incident faces of \( e \). At least one of \( f \) and \( f' \) is not in \( E_{\text{FACE}}(v_i) \) (otherwise \( e \) is in \( E_{\text{EDGE}}(v_i) \)). W.l.o.g., suppose \( f \notin E_{\text{FACE}}(v_i) \). If \( f \in I_{\text{FACE}}(v_i) \), then do nothing. If \( f \in P_{\text{FACE}}(v_i) \), we select a triangle in \( f \) with one edge on \( e \) and put it in \( X_i \).

- If \( v_i \text{PO} v_j \) is in \( \Gamma \), then by Table 8 we have \( E_{\text{FACE}}(v_i) \cup E_{\text{FACE}}(v_j) \neq E_{\text{FACE}} \). There exists a face \( f \) in \( (I_{\text{FACE}}(v_i) \cup P_{\text{FACE}}(v_i)) \cap (I_{\text{FACE}}(v_j) \cup P_{\text{FACE}}(v_j)) \). If \( f \) is in \( P_{\text{FACE}}(v_i) \cap P_{\text{FACE}}(v_j) \), then select a triangle in face \( f \) and put it in both \( X_i \) and \( X_j \). Otherwise, we do nothing.

We assume that all the triangles are pairwise disjoint and are sufficiently small such that the union of all the triangles does not entirely occupy any face or any edge. Now \( X_i \) contains all the triangles we need for spatial variable \( v_i \). For clarity, we now consider each face as its closure, and we use \( (v_j \text{PP} v_i) \in \Gamma \) to denote that
$v_j \text{TPP} v_i$ or $v_j \text{NTPP} v_i$ is a constraint in $\Gamma$. Define $a_i$ and $b_i$ as follows:

\[
a_i = \bigcup X_i, \quad \text{(B.1)}
\]
\[
b_i = a_i \cup \bigcup \mathcal{I}_{\text{FACE}}(v_i) \cup \bigcup \{a_j : (v_j \text{PP} v_i) \in \Gamma\}. \quad \text{(B.2)}
\]

We assert that $\{b_1, b_2, \ldots, b_n\}$ satisfies all the constraints in $\Gamma$ except that some NTPP constraints may be realised as TPP. This assertion can be proven in the same way as in the proof of Lemma 3.

Let $X$ be the union of all $X_i$, i.e. $X$ is the set of all the triangles selected for spatial variables. To cope with the NTPP constraints, we introduce the expand function from $(X \cup \text{FACE}) \times \{1, 2, \ldots, n\}$ to regions in the plane such that for any $x, x' \in X \cup \text{FACE}$,

- expand($x, 1$) = $x$.
- expand($x, i$) NTPP \(\text{expand}(x, i + 1)\) for $i = 1, 2, \ldots, n - 1$.
- expand($x, i$) DC expand($x', i'$) if $x \text{DC} x'$, for $i, i' = 1, 2, \ldots, n$.
- expand($x, i$) PO expand($x', i'$) if $x \text{EC} x'$, for $i, i' = 1, 2, \ldots, n$.

That is to say, expand($x, i$) ($i = 1, 2, \ldots, n$) is a series of nested regions among which $x$ is the innermost core. Meanwhile, expand($x, i$) should be small enough to not touch or overlap any other regions or any other expand($x', i'$) whenever possible. Figure B.9 provides illustrations for expand($x, 1$).

We can extend the domain of the function expand to include all $b_i$ defined above and all landmarks by

\[
\text{expand}(y, i) = \bigcup \{\text{expand}(x, i) : x \subseteq y, x \in X \cup \text{FACE}\}, \quad \text{(B.3)}
\]

where $y \in \{b_1, \ldots, b_n, l_1, \ldots, l_m\}$ and $i = 1, 2, \ldots, n$. 

Figure B.9: Illustration of function expand($x, 1$)
Define a function $d_{NTPP} : V \times (V \cup L) \to \mathbb{N}$, such that $d_{NTPP}(v_i, w)$ is the length of the longest NTPPi chain from $v_i$ to $w$, where $w$ is either variable $v_j$ or landmark $l_j$. Furthermore, define

$$c_i = b_i \cup \bigcup \{\text{expand}(b_j, d_{NTPP}(v_i, v_j)) : v_j \text{NTPP} v_i\}$$
$$\cup \bigcup \{\text{expand}(l_j, d_{NTPP}(v_i, l_j)) : l_j \text{NTPP} v_i\}. \quad \text{(B.4)}$$

It can be proven that $\{c_1, \ldots, c_n\}$ is a solution of $\Gamma$ such that $S_i = \partial c_i \cap \text{VTX}$ for $i = 1, 2, \ldots, n$ in the same way as in [19, 22]. We omit the details here.

Appendix C. Proof sketch of Theorem 12

We need to adjust the construction given in the sufficiency part to cope with the strong connectedness. The only differences from the standard RCC8 interpretation are: (i) we assume $S_i = \emptyset$ for each variable $v_i$; (ii) although the requirements for $\text{expand}(\cdot, \cdot)$ still apply, we need to modify the construction of this function to cater for the change in the interpretations of RCC8 relations. If $x$ is a face in $\text{FACE}$, or a triangle in $X$ on some vertex $v$, $\text{expand}(x, 1)$ should be modified as shown in Figures C.10(a) and (c) respectively, which can be contrasted with Figures B.9(a) and (c). Note that in Figure C.10(c), it holds that $x \text{DC} f_1$ because their intersection is a point (not a curve). Therefore, $\text{expand}(x, 1)$ is supposed to be disjoint with $f_1$ (under the strong connectedness interpretation of RCC8) due to the requirement of $\text{expand}(\cdot, \cdot)$. The case in Figure C.10(a) is similar: the boundary of the expanded face does not intersect with any face which is disjoint with the original face.

All the remaining parts of the construction, including the selection of triangles (note that $S_i = \emptyset$ here), definitions of $a_i$, $b_i$, and verification of $b_i$ as a solution of $\Gamma$, are completely the same as in the standard interpretation of RCC8 relations.

References


Figure C.10: Illustration of function $\text{expand}(x, 1)$ in the RCC model based on strong connectedness.


