Reasoning about Betweenness and RCC8 Constraints in Qualitative Conceptual Spaces

Steven Schockaert
School of Computer Science and Informatics
Cardiff University, UK
schockaerts1@cardiff.ac.uk

Sanjiang Li
Centre for Quantum Software and Information
University of Technology Sydney, Australia
sanjiang.li@uts.edu.au

Abstract

Conceptual spaces are a knowledge representation framework in which concepts are represented geometrically, using convex regions. Motivated by the fact that exact conceptual spaces are usually difficult to obtain, we study the problem of spatial reasoning about qualitative abstractions of such representations. In particular, we consider the problem of deciding whether an RCC8 network extended with constraints about betweenness can be realized using bounded and convex regions in a high-dimensional Euclidean space. After showing that this decision problem is PSPACE-hard in general, we introduce an important fragment for which deciding realizability is NP-complete.

1 Introduction

Knowledge about concepts can be represented in at least two radically different ways. On the one hand, ontology languages are often used for this purpose. Such languages allow us to describe, in a transparent and precise way, how different concepts are related, but they offer only limited means for supporting inductive inferences. On the other hand, in fields such as natural language processing (NLP) or machine learning, knowledge about concepts is usually encoded in the form of vector space embeddings. These embeddings typically encode both concepts and objects as vectors in a high-dimensional Euclidean space [Mikolov et al., 2013; Bordes et al., 2013]. They are very effective at supporting inductive inferences (e.g. generalizing a learned model to words not encountered during training in NLP applications), but are not able to encode exact logical relationships; vector representations of concepts can encode how similar two concepts are, but not e.g. whether one is subsumed by the other.

Conceptual spaces [Gärdenfors, 2000] are geometric representations, in which the properties and concepts from a given domain are modeled as convex regions in a potentially high-dimensional space. In some sense, conceptual spaces combine the best of both worlds: their geometric nature allows for elegant formalizations of inductive arguments, while the use of regions means that logical relationships such as disjointness and subsumption can be encoded in an exact way.

Unfortunately, conceptual space representations are difficult to learn from data. A promising alternative, which we consider in this paper, is to rely on qualitative abstractions of conceptual spaces to encode knowledge about concepts and properties. A standard taxonomy, viewed as a set of statements of the form “A is subsumed by B”, written $A \subseteq B$, can be seen as a simple example of such a qualitative abstraction, if we view $A \subseteq B$ as the spatial constraint $\text{reg}(A) \subseteq \text{reg}(B)$, where we write $\text{reg}(X)$ for the (unknown) conceptual space representation of the concept $X$. Similarly, the assertion that two concepts cannot share any instances corresponds to the spatial constraint that $\text{reg}(A)$ and $\text{reg}(B)$ should be disjoint.

By encoding knowledge using qualitative spatial constraints, we can also encode information that goes beyond what is normally considered in ontologies. One example is conceptual neighborhood [Freksa, 1991]. Two concepts $A$ and $B$ are called conceptual neighbors if there is a continuous transition from instances of $A$ to instances of $B$, without having to go via a third concept. For example, the colors red and orange, the place types pub and restaurant, and the adjectives moderate and severe could all be regarded as conceptual neighbors. The assertion that $A$ and $B$ are conceptual neighbors naturally corresponds to the spatial constraint that $\text{reg}(A)$ and $\text{reg}(B)$ are adjacent (i.e. externally connected) to each other. This type of information is important in applications where information from multiple sources needs to be combined, as different sources may disagree about where exactly the boundary between two conceptual neighbors should be drawn [Schockaert and Prade, 2011]; e.g. a place which is regarded as a pub by one source might be regarded as a restaurant by another source, even if both sources agree on the fact that pub and restaurant are disjoint concepts.

So far, we have only considered qualitative spatial constraints that can be modeled in the well-known RCC8 calculus [Randell et al., 1992]. Such constraints are not sufficient for assessing the plausibility of inductive arguments. In particular, qualitative conceptual space representations should allow us to determine whether a given property $P$ is likely to apply to instances of a given concept $C$, knowing only that
it applies to instances of some other concepts $C_1, \ldots, C_k$ [Osherson et al., 1990]. For instance, knowing that fine-dining restaurants and fast-food restaurants in Sweden both need a license to sell alcohol, we can plausibly conclude that bistros would as well. Intuitively, this is plausible because bistros are conceptually in between fine-dining restaurants and fast-food restaurants. It was found in [Derrac and Schockaert, 2015] that this qualitative notion of conceptual betweenness is often sufficient to make highly accurate predictions.

To further enrich qualitative conceptual space representations, we can consider constraints that only refer to particular facets of concepts\(^2\) (e.g. encoding that the price range of a bistros is in between that of fast-food and fine-dining restaurants). Moreover, concepts with gradual boundaries can be represented by considering nested sets of regions [Gärdenfors and Williams, 2001]. This would allow us, for instance, to express that the typical size of a bird is in between that of mice and dogs. However, as such modeling constructs do not affect our technical results, we will not explicitly consider them.

Qualitative conceptual spaces can be obtained in similar ways as traditional taxonomies, e.g. manually encoded by domain experts, extracted from text, or learned using Bayesian methods [Griffiths et al., 2008]. Approximate information about betweenness can furthermore be derived from vector space embeddings. However, as most of these methods are noisy, we need effective mechanisms for checking the consistency of the resulting representations, to allow repair mechanisms similar to the techniques which are used for taxonomy learning when cycles are detected [Merhej et al., 2014].

To this end, we study the problem of deciding whether a set of RCC8 constraints with betweenness information can be realized using convex regions. First, we show that this decision problem is PSPACE-hard in general. We then introduce an important fragment, and show that deciding realizability for this fragment is NP-complete. The basis of this result comes from the observation that the models of (atomic) sets of constraints from this fragment can be characterized using a number of rankings. This ranking representation is significant for applications, as it allows us to compactly encode which betweenness relations hold between arbitrary sets of regions, whereas naive encodings would be exponentially large.

The most closely related work is [Schockaert and Li, 2013], where an extension of RCC5 with betweenness constraints was proposed. However, while adding betweenness constraints to RCC5 networks only affects reasoning in a minimal way (e.g. most transitivity rules that we intuitively expect to hold for betweenness are only valid for disconnected regions). As we will see, the situation for RCC8 networks is entirely different. Beyond betweenness, a number of other works have considered forms of qualitative spatial reasoning that are important for reasoning about concepts. For example, [Wolter and Zakharyaschev, 2000] introduced an extension of RCC8 in which constraints can relate to unions and intersections of regions. In [Sheremet et al., 2007] a logic is introduced which can be used to reason about relative similarity (i.e. region $A$ is closer to $B$ than to $C$). The problem of realizing RCC8 networks using convex regions has been studied in [Davis et al., 1999] for two-dimensional spaces and in [Schockaert and Li, 2015] for general Euclidean spaces.

## 2 Preliminaries

In this section we recall some basic notions from the region connection calculus.

RCC8 is a constraint language based on the Region Connection Calculus [Randell et al., 1992]. Constraints are built from a set of 8 primitive relations $\mathcal{R}_8 = \{ \text{DC}, \text{EC}, \text{PO}, \text{EQ}, \text{TPP}, \text{TPP}^{-1}, \text{NTPP}, \text{NTPP}^{-1} \}$ and a set of variables $V$. For each $R \in \mathcal{R}_8$ and $u, v \in V$, the expression $uRv$ is called an RCC8 constraint. An RCC8 network is a set of RCC8 constraints in which each pair of variables occurs in at most one constraint. If $R = \{R_0\}$ is a singleton then $uRv$ is called an atomic RCC8 constraint, and we usually write $u R_0 v$ instead of $u \{R_0\} v$. We will use some common abbreviations to denote particular non-atomic RCC8 relations: $\text{DR} = \{\text{DC}, \text{EC}\}$, $\text{PP} = \{\text{TPP}, \text{NTPP}\}$, $\text{P} = \text{PP} \cup \{\text{EQ}\}$, and $\mathcal{O} = \mathcal{R}_8 \setminus \{\text{DC}, \text{EC}\}$. An RCC8 network $\Theta$ is called atomic if every constraint in $\Theta$ is atomic.

Let $\text{Reg}_n$ be the set of non-empty regular closed\(^3\) subsets of $\mathbb{R}^n$. A mapping $S$ from $V$ to $\text{Reg}_n$ is called an (n-dimensional) interpretation. Table 1 lists the conditions under which $S$ satisfies each of the atomic RCC8 constraints, except for $u$ $\text{NTPP}^{-1} v$ and $u$ $\text{TPP}^{-1} v$, which are equivalent to $v$ $\text{NTPP} u$ and $v$ $\text{TPP} u$ respectively. When $S$ satisfies an atomic RCC8 constraint $u R v$, we write $S \models u R v$. We have that $S$ satisfies a non-atomic constraint $u R v$, written $S \models u R v$, if $S \models R_0$ for some $R_0 \in R$. An RCC8 network $\Theta$ is called satisfiable (or consistent or realizable) if there is an interpretation $S$ that satisfies every constraint in $\Theta$. In that case $S$ is called an (RCC8) solution (or realization). While we only consider Euclidean spaces in this paper, it should be noted that any RCC8 network that can be realized in a topological space that satisfies the axioms of the Region Connection Calculus also has a Euclidean solution, even if we fix the number of dimensions [Renz, 2002].

## 3 RCC8\(^8\)

We now introduce the notion of RCC8\(^8\)-network, which is the RCC8 analogue of the RCC5\(^8\)-networks from [Schockaert and Li, 2013]. To encode betweenness information, we will use RCC8 constraints in which the arguments can be expressions of the form $u \bowtie w$, denoting the convex hull of the union of $u$ and $w$. For instance, the constraint $u$ $\text{PO} u \bowtie w$ encodes that $v$ is completely between $u$ and $w$, while $v$ $\text{PO} u \bowtie w$ encodes that $v$ is partially between $u$ and $w$. In particular, let $V^8 = V \cup \{u \bowtie v \mid u, v \in V\}$. An RCC8\(^8\)-network over $V$ is then defined as an RCC8 network over $V^8$. An interpretation $\mathcal{S}$ is called an RCC8\(^8\)-solution of an RCC8\(^8\)-network $\Theta$ if (i) $\mathcal{S}$ is an RCC8\(^8\)-solution of $\Theta$ and (ii) for each $u, v \in V$ it holds that $\mathcal{S}(u \bowtie v) = \text{CH}(\mathcal{S}(u) \cup \mathcal{S}(v))$, where CH denotes the convex hull operator. Note that while $\bowtie$ is limited to two regions, we can consider the convex hull of an arbitrary number of regions by introducing fresh variables, e.g. in any RCC8\(^8\)-solution

\(^2\)Note that is a direct counter-part to multi-domain conceptual space representations.

\(^3\)A is regular closed if $A$ is equal to the closure of its interior.
of \{x \text{ EQ } a_1 \times y_1, y_1 \text{ EQ } a_2 \times y_2, ..., y_{n-1} \text{ EQ } a_{n-1} \times a_n\} it holds that \(S(x) = \text{CH}(S(a_1), ..., S(a_n))\). We write \(\Theta \models \Phi\)
for \(\Theta\) and \(\Phi\) (sets of) RCC8 constraints, to denote that every RCC8 solution of \(\Theta\) is also an RCC8 solution of \(\Phi\). Note that this is different from requiring that every RCC8 solution of \(\Theta\) is an RCC8 solution of \(\Phi\).

The problem we will focus on in this paper is deciding whether an RCC8 has a convex and bounded solution, i.e. a solution in which every variable is mapped to a convex and bounded region. The reason why we require convexity stems from our motivation in terms of conceptual spaces. Furthermore, note that because we can always add constraints of the form \(a \text{ EQ } a_a\), encoding that \(a\) has to be convex, the hardness results we show in this paper remain valid even when convexity is not required. From a practical point of view, this also means that we need to understand how convex solutions can be found, before arbitrary solutions can be considered. The restriction to bounded regions is because the \(\rtimes\) operator can otherwise lead to counterintuitive behaviour\(^5\).

We can show the following result\(^5\).\(^6\).

**Theorem 1.** Deciding whether a RCC8 network has a convex and bounded solution is PSPACE-hard.

When an RCC8 network \(\Theta\) contains a constraint such as \(u \text{ EC } v\), then in any solution \(S\) there is a hyperplane \(H\) separating \(S(u)\) from \(S(v)\). For each region \(v\), let \(v_H\) be a new variable corresponding to the (lower-dimensional and possibly empty) region \(S(v) \cap H\). The constraints in \(\Theta\) can impose constraints between these new variables \(v_H\), and checking their satisfiability is similar to checking the satisfiability of the constraints between the initial set of variables. Moreover, these new constraints may induce further constraints, between intersections of the regions \(S(v_H)\) with another separating hyperplane. Intuitively, it is this recursive process that causes the PSPACE-hardness.

### 3.1 Basic Networks

We call an RCC8 network \(\Theta\) basic if each constraint in \(\Theta\) has one of the following forms:

- \(u \text{ DC } v\), with \(R \in \{\text{DC, EC, NTTP, P}\}\) and \(u, v \in V\)
- \(v \text{ PO } u \rtimes w\), with \(u, v, w \in V\)

Note that any atomic RCC8 network can be represented as a basic network by introducing fresh variables, e.g.

\(^5\)The fact that \(S(u)\) and \(S(v)\) are regular closed is not sufficient for \(\text{CH}(S(u), S(v))\) to be regular closed, which leads to particular restrictions on that mappings that we would be able to consider as interpretations.

\(^6\)The proofs of all results from this paper are available in an online appendix at: http://users.cs.cf.ac.uk/S.Schockaert/reports/betweennessRCC8-supplement.pdf

### 4 An NP-Complete Fragment

We now introduce a restriction on RCC8 networks to prevent the aforementioned recursive process that is responsible for the PSPACE-hardness result.

**Definition 1.** An RCC8 network \(\Theta\) is non-recursive if there are no \(u, v, w \in V\) such that \(\Theta \models \{u \text{ NTTP } v, v \text{ EC } w\}\).

One straightforward way to guarantee non-recursiveness is by either disallowing EC constraints or NTTP constraints. In applications, EC constraints are important for modeling conceptual neighborhood, while NTTP relations are important for modeling concepts with gradual boundaries. However, we would normally only need one of these modeling constructs, as the notion of conceptual neighborhood is based on the assumption that there is a crisp (i.e. non-gradual) but ill-defined boundary between two concepts.

Next we show that checking if a basic non-recursive network has a convex and bounded solution is NP-complete.

### 4.1 Ranking-Based Models

Let us assume that there are regions \(a_1, ..., a_m, b_1, ..., b_n\) and some \(p \in \{1, ..., m\}\) satisfying the following assumptions:

- **AB\(_1\)** For each \(i \in \{1, ..., p\}\) it holds that \(\Theta \models a_i \text{ DC } b_i\).
- **AB\(_2\)** For each \(i \in \{p+1, ..., m\}\) it holds that \(\Theta \models a_i \text{ EC } b_i\).
- **AB\(_3\)** For every \(u, v \in V\) such that \(\Theta \models u \text{ DC } v\) (resp. \(\Theta \models u \text{ EC } v\)) there exists an \(i \in \{1, ..., p\}\) (resp. \(i \in \{p+1, ..., m\}\)) such that \(\Theta \models \{a_i \text{ PO } a_i, v \text{ PO } b_i\}\) or \(\Theta \models \{v \text{ PO } a_i, u \text{ PO } b_i\}\).
For each $v \in V$ and $i \in \{1, \ldots, m\}$ it holds that (i) $\Theta \models \forall v \forall i \left( \forall a_i, \exists b_i \right)$ or (ii) $\Theta \models \forall v \forall i \left( \exists a_i \right)$ and (iii) $\Theta \models \forall v \forall i \left( \exists a_i \land \exists b_i \right)$.

These assumptions do not affect generality, as we can always add fresh regions and guess the required constraints to make $AB_1 - AB_4$ satisfied. Note that while this process involves making a guess, it does not affect the NP-completeness result.

Let us write $M = \{a_1, \ldots, a_m, b_1, \ldots, b_m\}$. We also write $\text{inv}(a_i)$ for $b_i$ and $\text{inv}(b_i)$ for $a_i$. The following proposition introduces a necessary condition for RCC8-networks to have a convex solution. In Section 4.3, we will see that this condition is also sufficient for basic non-recursive networks.

**Proposition 1.** Suppose that the RCC8-network $\Theta$ has a convex and bounded RCC8-network solution. For each $x \in M$ there exists a mapping $\rho_x : V \rightarrow \{0, \ldots, |V| - 1\}$ such that for every $u, v, w \in V$ it holds that:

$$
\Theta \models \forall v \forall u \forall w \left( \rho_x(v) \geq \min(\rho_x(u), \rho_x(w)) \right) \quad (1)
$$

$$
\Theta \models \forall v \forall u \forall w \left( \rho_x(u) > \rho_x(v) \right) \quad (2)
$$

$$
\Theta \models \forall v \forall u \forall w \left( \rho_x(v) < \rho_x(x) \right) \quad (3)
$$

and for $x, y \in M$ and $u, v \in V$ such that $\Theta \models y \text{DC inv}(y)$:

$$
\rho_x(u) = \rho_x(v) \land \rho_y(u) = \rho_y(v) \Rightarrow \rho_{\text{inv}(y)}(v) < \rho_{\text{inv}(y)}(y) \quad (4)
$$

The mappings $\rho_x$ essentially define a set of rankings over the variables from $V$, as only the relative ordering of the $\rho_x(v)$ values matters. Note that the existence of suitable rankings can be straightforwardly checked using integer linear programming. Because the existence of these rankings is equivalent to the realizability of $\Theta$ using convex and bounded regions (as we will show), we can effectively view them as models of RCC8-networks. From an application point of view, this ranking based representation is also important, as it will allow us to easily determine for arbitrary variables $v_1, \ldots, v_k, u$ whether $S(u) \subseteq \text{CH}(S(v_1) \cup \ldots \cup S(v_k))$, for $S$ the solution that we will associate with the rankings, which will be the case iff $\rho_x(v) \geq \min(\rho_x(v_1), \ldots, \rho_x(v_k))$ holds for all $x \in M$. By encoding qualitative conceptual spaces in terms of such rankings, we can thus compactly encode exponentially many betweenness statements.

**Example 1.** It follows from Proposition 1 that the following set of constraints is not satisfiable:

$$
\begin{align*}
a_1 \text{PO } a_2 & \quad a_1 \text{PO } b_2 & \quad b_1 \text{PO } a_2 & \quad b_1 \text{PO } b_2 \\
a_1 \text{DC } b_1 & \quad a_2 \text{DC } b_2 & \quad a_1 \text{PO } b_1 & \quad b_1 \text{PO } a_1 & \quad b_1 \text{PO } b_2
\end{align*}
$$

Indeed, using (1) and (3) we find:

$$
\rho_{a_1}(b_1) \geq \min(\rho_{b_2}(a_1), \rho_{b_2}(b_2)) = \rho_{b_2}(a_1)
$$

$$
\rho_{b_2}(a_2) \geq \min(\rho_{b_1}(b_1), \rho_{b_1}(b_2)) = \rho_{b_1}(b_1)
$$

It follows that $\rho_{a_1}(a_1) = \rho_{b_2}(b_1)$, and thus by (5) (for $y = u = b_1$) that $\rho_{a_1}(a_1) < \rho_{a_1}(a_1)$, which is a contradiction.

Even for basic non-recursive networks, checking whether Proposition 1 is satisfied is NP-hard. In fact, we can show the following, stronger property.

**Proposition 2.** Let $\Theta$ be an RCC8-network. The problem of checking whether $\Theta$ satisfies the conditions from Proposition 1 is NP-complete, even if $\Theta$ is a set of basic network and does not contain any occurrences of NTPP or EC.

### 4.2 Properties of Hyperrectangles

In Section 4.3, we will show that when the conditions from Proposition 1 are satisfied, and $\Theta$ is a basic non-recursive RCC8-network, we can construct a convex and bounded solution from the mappings $\rho_x$. The main idea will be to realize variables using axis-aligned hyperrectangles, where $\rho_{a_i}(v)$ and $\rho_{b_i}(v)$ respectively determine the lower and upper bound of the $i$th coordinate for the hyperrectangles associated with $v$. In this section, we first discuss how hyperrectangles can be chosen such that betweenness constraints can easily be satisfied. The main idea is illustrated in the following example.

**Example 2.** Let $\Theta$ contain the following constraints:

$$
\begin{align*}
a_1 \text{DC } b_1 & \quad a_2 \text{DC } b_2 & \quad a_1 \text{PO } a_2 & \quad a_1 \text{PO } b_2 \\
b_1 \text{PO } a_2 & \quad b_1 \text{PO } b_2 & \quad c \text{P } a_1 & \quad c \text{P } b_2 \\
d \text{P } a_1 & \quad d \text{P } a_2 & \quad e \text{P } b_1 & \quad e \text{P } a_2 \\
f \text{P } b_1 & \quad f \text{P } b_2 & \quad d \text{P } c \text{e } & \quad f \text{P } c \text{e }
\end{align*}
$$

It can be verified that the following choices for $\rho_x$ satisfy the conditions of Proposition 1:

$$
\begin{align*}
\rho_{a_1}(b_1) & = 1 & \rho_{a_1}(a_1) & = 1 & \rho_{a_2}(a_1) & = 1 & \rho_{b_2}(a_1) & = 1 \\
\rho_{a_1}(a_2) & = 2 & \rho_{a_2}(a_2) & = 2 & \rho_{b_2}(b_1) & = 2 & \rho_{a_2}(b_1) & = 2 \\
\rho_{a_1}(b_2) & = 3 & \rho_{a_1}(b_2) & = 3 & \rho_{a_2}(a_2) & = 3 & \rho_{b_2}(a_2) & = 3 \\
\rho_{a_1}(c) & = 4 & \rho_{b_1}(c) & = 4 & \rho_{a_2}(c) & = 4 & \rho_{b_2}(c) & = 4 \\
\rho_{a_1}(f) & = 5 & \rho_{b_1}(d) & = 5 & \rho_{a_2}(f) & = 5 & \rho_{b_2}(d) & = 5 \\
\rho_{a_1}(a_1) & = 6 & \rho_{b_1}(b_1) & = 6 & \rho_{a_2}(a_2) & = 6 & \rho_{b_2}(b_2) & = 6 \\
\rho_{a_1}(d) & = 7 & \rho_{b_1}(e) & = 7 & \rho_{a_2}(d) & = 7 & \rho_{b_2}(e) & = 7 \\
\rho_{a_1}(e) & = 8 & \rho_{b_1}(f) & = 8 & \rho_{a_2}(e) & = 8 & \rho_{b_2}(c) & = 8
\end{align*}
$$

A solution $S$ is shown in Figure 1, which corresponds to the rankings $\rho_x$ in the following sense: there are decreasing functions $f_1, f_2, f_3, f_4$ such that $S(v)$ for any $v \in V$ is given by:

$$
\left[ - f_1(\rho_{a_1}(v)), f_2(\rho_{a_1}(v)) \right] \times \left[ - f_3(\rho_{a_2}(v)), f_4(\rho_{b_2}(v)) \right]
$$

As illustrated in Example 2, up to a monotonic transformation, the rankings $\rho_x$ will determine a solution (although it

---

*This is shown in the online appendix.*
will not always be possible to realize variables using a single hyperrectangle). In accordance with condition (1), we need to choose this monotonic transformation such that we are able to realize a region \( b \) as being between two other regions \( a \) and \( c \), as soon as \( \rho_c(b) \geq \min(\rho_a(a), \rho_c(c)) \) for all \( x \in M \). To this end we will use the \( \{0, ..., \eta\} \rightarrow [0, +\infty[ \) functions \( f \) and \( g \) defined as follows:

\[
\begin{align*}
f(i) &= \begin{cases} 2f(i + 1) + 1 & \text{if } i < \eta \\ 3 & \text{if } i = \eta \end{cases} \\
g(i) &= \left(1 - \frac{1}{2^i}\right)
\end{align*}
\]

where \( \eta \in \mathbb{N} \) will be assumed to be sufficiently large (see below for details).

The definition of \( S \) will be such that all points in \( S(a_i) \) are positive in the \( i^{th} \) coordinate and all points in \( S(b_i) \) are negative in this coordinate (as in Figure 1). Furthermore, if \( \rho_{a_i}(u) < \rho_{a_i}(a_i) \) then \( S(u) \) will overlap with \( S(b_i) \), which means that the lower bound on the \( i^{th} \) coordinate of the hyperrectangles defining \( S(u) \) will be negative. On the other hand, if \( \rho_{a_i}(u) \geq \rho_{a_i}(a_i) \) then \( S(u) \) will be included in \( S(a_i) \) and thus this lower bound will be positive. This difference matters for the following reason. If the lower bound is negative, we can always make a constraint such as \( v \mathbf{P} u \mathbf{w} \mathbf{w} \) satisfied by making the lower bound of the \( i^{th} \) coordinate of \( S(u) \) sufficiently small (and appropriately defining the other coordinates). In such cases we use the function \( f \) to define the lower bound. However, if the lower bound needs to be positive, a different strategy is needed, and we instead use the function \( g \) to define lower bounds that can be arbitrarily close to 1. For the ease of presentation, we will define upper bounds using the following abbreviation, for \( \lambda \in \{0, ..., \eta\} \):

\[
\begin{align*}
h^{+}_\lambda(r) &= \begin{cases} f(r) & \text{if } r < \lambda \\ -g(r - \lambda + 1) & \text{otherwise} \end{cases} \\
h^{-}_\lambda(r) &= \begin{cases} -f(r) & \text{if } r < \lambda \\ g(r - \lambda + 1) & \text{otherwise} \end{cases}
\end{align*}
\]

Similarly we will define lower bounds using the abbreviation \( h^{-}_\lambda(r) \), defined as \( -h^{+}_\lambda(r) \). Intuitively, we can think of \( h^{+}_\rho_{a_i}(b_i)(\rho_{b_i}(u)) \) and \( h^{-}_\rho_{a_i}(a_i)(\rho_{a_i}(v)) \) as the upper and lower bounds on the \( i^{th} \) coordinate for the hyperrectangle corresponding to \( v \). However, the actual construction in Section 4.3 will be slightly more complicated, among others because we will need more than one hyperrectangle for each variable.

We call a hyperrectangle \( X_i[l_i, u_i] \) regular if each \( l_i \) is of the form \( h^{+}_\lambda(r_i) \) and each \( u_i \) is of the form \( h^{-}_\lambda(r_i) \). The main motivation for using regular hyperrectangles is that they allow us to easily determine which points are in between them. In particular, the construction in Section 4.3 will crucially rely on the following propery about regular hyperrectangles.

**Lemma 1.** Let \( X_i[l_i, u_i] \) and \( X_i[l'_i, u'_i] \) be regular hyperrectangles. Furthermore, we consider the point \((p_1, ..., p_m)\) where each \( p_i \) is of the form \( f(r_i) - f(r_i), g(r) \) or \( -g(r) \). We have \((p_1, ..., p_m) \in \text{CH}((X_i[l_i, u_i] \cup (X_i[l'_i, u'_i])) \) if the following holds for each \( i \):

- \( \min(l_i, l'_i) < p_i \) or \( l_i = l'_i = p_i \); and
- \( p_i < \max(u_i, u'_i) \) or \( u_i = u'_i = p_i \)

![Figure 2: Solution of the RCC8* network from Example 3. For simplicity, the realizations of \( a_1, b_1, a_2, b_2 \) are not shown.](image)

### 4.3 Constructing Convex Solutions

As already mentioned, it is not always possible to construct a solution in which each variable is realized as a single hyperrectangle. This is illustrated in the following example.

**Example 3.** Consider the following RCC8* constraints:

\[
\begin{align*}
v_0 &\mathbf{P} a_1 & v_0 &\mathbf{P} b_2 & v_1 &\mathbf{P} a_1 & v_1 &\mathbf{P} a_2 & w_0 &\mathbf{P} b_1 \\
v_0 &\mathbf{P} b_2 & v_1 &\mathbf{P} b_1 & w_1 &\mathbf{P} a_1 & u_0 &\mathbf{P} a_1 & u &\mathbf{P} b_2 \\
v &\mathbf{P} a_1 & v_0 &\mathbf{P} a_2 & v_1 &\mathbf{P} b_2 & w &\mathbf{P} b_1 & w_0 &\mathbf{P} w \\
w_1 &\mathbf{P} w & a_1 &\mathbf{D} & b_1 & a_2 &\mathbf{D} & b_2 & a_1 &\mathbf{P} & a_2 & a_1 &\mathbf{P} & b_2 & b_1 &\mathbf{P} & a_2 & b_1 &\mathbf{P} & b_2 & u &\mathbf{P} & w & w & u &\mathbf{P} & w & u &\mathbf{P} & w
\end{align*}
\]

A possible solution \( S \) is shown in Figure 2, where \( v \) is realized as the convex hull of the two rectangles that are labelled with \( v \). The remaining variables are realized using single rectangles, as before. To see why it is not possible to realize \( v \) using a single rectangle, note that because of the constraints \( u \mathbf{P} v \mathbf{w} \mathbf{w} \) and \( v \mathbf{P} u \mathbf{w} \mathbf{w} \), the points from \( S(u) \) that are at maximal distance from \( S(w) \) have to belong to \( S(v) \) and vice versa. In particular, this means that none of the points from \( S(v) \) that are at maximal distance from \( S(w) \) can belong to \( S(a_2) \). Hence, with a single axis-aligned rectangle for \( v \), we would not be able to also satisfy the constraint \( v_1 \mathbf{P} v \).

We will construct a solution \( S \) of the following form \((v \in V)\):

\[
S(v) = \text{CH}( \bigcup_{x \in M} R_x(v) )
\]

where \( R_x(v) \) is a hyperrectangle of the following form:

\[
[(l(v, x, 1), u(v, x, 1))] \times \ldots \times [(l(v, x, m), u(v, x, m))]
\]

Next we explain how the lower bounds \( l(v, x, i) \) and upper bounds \( u(v, x, i) \) are chosen. Throughout the section, we will assume that \( \eta \geq 3mn + 2m \). Intuitively, we want the hyperrectangle \( R_x(v) \) to correspond to the set of points from \( S(v) \) whose \( i^{th} \) coordinate is minimal. For \( i \leq p \), that minimal value will be chosen as \( h^{3m\rho_{a_i}(a_i)}(3m\rho_{a_i}(v)) \). The reason why we choose this bound in terms of \( 3m\rho_{a_i}(v) \) rather than simply \( \rho_{a_i}(v) \) relates to the fact that we need to consider a number of additional hyperrectangles in the proof. For \( i > p \), some additional care
is needed to ensure that all EC relations are satisfied. To this end, if \( i > p \), we will ensure that \( l(v, a_i, i) \geq 0 \) for all regions \( v \). In other words, for \( x = a_i \) we have:

\[
l(v, a_i, i) = \begin{cases} 
\min(0, h_{3m\rho_a(a_i)}(3m\rho_a(v))) & \text{if } i > p \\
h_{3m\rho_a(a_i)}(3m\rho_a(v)) & \text{otherwise} 
\end{cases}
\]

The upper bound \( u(v, a_i, i) \) is essentially irrelevant, as the rectangle \( R_{a_i}(v) \) is only aimed at encoding the set of points from \( S(v) \) whose \( i^{th} \) coordinate is minimal. In principle, we could define \( R_{a_i}(v) \) as a degenerate hyperrectangle and choose \( l(v, a_i, i) = u(v, a_i, i) \). However, to be in accordance with the conditions of Lemma 1, if \( i < p \), we will choose \( u(v, a_i, i) \) as the lowest possible value that makes \( R_{a_i}(v) \) a regular hyperrectangle. For \( i \geq p \) we can simply choose \( u(v, a_i, i) = 0 \):

\[
u(v, a_i, i) = \begin{cases} 
0 & \text{if } i > p \\
\kappa^+(l(v, a_i, i)) & \text{otherwise} 
\end{cases}
\]

where

\[
\kappa^+(x) = \min(\{f(r) \mid f(r) \geq x\} \cup \{-g(r) - g(r) \geq x\})
\]

The values \( l(v, b_i, i) \) and \( u(v, b_i, i) \) are chosen in a similar way, but here it is rather the upper bound that matters:

\[
l(v, b_i, i) = \begin{cases} 
0 & \text{if } i > p \\
\kappa^-(u(v, b_i, i)) & \text{otherwise} 
\end{cases}
\]

\[
u(v, b_i, i) = \begin{cases} 
\max(0, h_{3m\rho_b(b_i)}(3m\rho_b(v))) & \text{if } i > p \\
h_{3m\rho_b(b_i)}(3m\rho_b(v)) & \text{otherwise} 
\end{cases}
\]

where

\[
\kappa^-(x) = \max(\{-f(r) \mid f(r) \leq x\} \cup \{g(r) \mid g(r) \leq x\})
\]

Finally, for \( x \notin \{a_i, b_i\} \), the bounds \( l(v, x, i) \) and \( u(v, x, i) \) are chosen such that \( R_x(v) = R_x(u) \) if \( \rho_x(u) = \rho_x(v) \):

\[
l(v, x, i) = \max\{l_1(v', i) \mid \rho_x(v') = \rho_x(v)\}
\]

\[
u(v, x, i) = \min\{u_1(v', i) \mid \rho_x(v') = \rho_x(v)\}
\]

where

\[
l_1(v, i) = \begin{cases} 
\min(0, h_{3m\rho_a(a_i)}(3m\rho_a(v)+2m)) & \text{if } i > p \\
h_{3m\rho_a(a_i)}(3m\rho_a(v)+2m) & \text{otherwise} 
\end{cases}
\]

\[
u_1(v, i) = \begin{cases} 
\max(0, h_{3m\rho_b(b_i)}(3m\rho_b(v)+2m)) & \text{if } i > p \\
h_{3m\rho_b(b_i)}(3m\rho_b(v)+2m) & \text{otherwise} 
\end{cases}
\]

We can then show the following result.

**Theorem 2.** Let \( \Theta \) be a basic non-recursive set of RCC8 constraints. Let \( S(v) \) be defined as in (6) and \( S(u \cup v) = CH(S(u) \cup S(v)) \) for \( u, v \in V \). If the mappings \( \rho_x \) used in the definition of the hyperrectangles \( R_x \) satisfy the conditions from Proposition 1, then \( S \) is a solution of \( \Theta \).

**Corollary 1.** Deciding whether a basic non-recursive RCC8 network has a convex and bounded solution is NP-complete.

The following example illustrates that non-recursiveisiveness is indeed required to guarantee the existence of a solution.

**Example 4.** Consider the following RCC8 constraints:

\[
a_1 \text{ EC } b_1 \quad a_2 \text{ EC } b_2 \quad a_1 \text{ PO } a_2 \quad a_1 \text{ PO } b_2 \\
b_1 \text{ PO } a_2 \quad b_1 \text{ PO } b_1 \quad a \text{ PO } a_1 \quad a \text{ PO } b_2 \\
b_0 \text{ PO } a_1 \quad b_0 \text{ PO } b_1 \quad b_0 \text{ PO } b_2 \quad c_0 \text{ PO } a_2 \\
c_0 \text{ PO } b_2 \quad c_0 \text{ PO } a_1 \quad d \text{ PO } b_1 \quad d \text{ PO } a_2 \\
b_0 \text{ NTPP } a \quad b \text{ NTPP } c \quad a \text{ EC } d
\]

Despite the fact that these constraints satisfy the conditions from Proposition 1, they do not admit a convex solution. Figure 3 shows an interpretation which satisfies all of these constraints, except for a EC d, which cannot be jointly satisfied with the other constraints. To see why this is the case, let us consider hyperplanes \( H_i \) separating \( a_i \) and \( b_i \). The problem is that \( c \) cannot contain any point from \( H_3 \) due to the constraint \( c \text{ NTPP } c_0 \) and similarly \( b \) cannot contain any point from \( H_2 \) due to the constraint \( b \text{ NTPP } b_0 \). As a result, given the constraint \( a \text{ PO } b \), \( a \) cannot contain any points that belong to \( H_1 \cap H_2 \), which would be required to satisfy a EC d.

## 5 Conclusions

We have studied the problem of reasoning about betweenness in the context of RCC8. Among others, we have shown that satisifability checking in this setting is PSPACE-hard in general, and we have introduced a fragment for which satisfiability checking is NP-complete. As this fragment appears general enough for most applications, one of the most important questions for future work is how RCC8 encodings of conceptual knowledge could be learned from text. Several interesting theoretical questions also remain. For example, it is still an open problem how consistency can be decided when convexity is not required. Another natural question is how betweenness could be combined with the Boolean region terms from [Wolter and Zakharyaschev, 2000].

## Acknowledgments

Steven Schockaert was supported by ERC Starting Grant 637277. Sanjiang Li was partially supported by NSFC (No. 11671244).

## References


