ECAI 2014
T. Schaub et al. (Eds.)

doi:10.3233/978-1-61499-419-0-495

Qualitative Spatial and Temporal Reasoning with AND/OR Linear Programming[1]

Arne Kreutzmann[2] and **Diedrich Wolter**[3]

Abstract. This paper explores the use of generalized linear programming techniques to tackle two long-standing problems in qualitative spatio-temporal reasoning: Using LP as a unifying basis for reasoning, one can jointly reason about relations from different qualitative calculi. Also, concrete entities (fixed points, regions fixed in shape and/or position, etc.) can be mixed with free variables. Both features are important for applications but cannot be handled by existing techniques. In this paper we discuss properties of encoding constraint problems involving spatial and temporal relations. We advocate the use of AND/OR graphs to facilitate efficient reasoning and we show feasibility of our approach.

1 Introduction

Qualitative spatial and temporal reasoning (QSTR) is involved with knowledge representations that explicate relational knowledge between (spatial or temporal) entities [11, 15]. QSTR has several important application areas both inside and outside AI. Over the past two decades of research, a rich repertoire of specialized representations has been proposed (see [4] for recent summary). Aside from the development of individually successful representations, called *qualitative calculi*, there are two penetrating and long-standing research questions that apply to all representations.

- How can qualitative calculi be combined, i.e., how can one jointly reason with knowledge represented in distinct calculi?
- How can qualitative representations incorporate grounded information, i.e., how can free-ranging and constrained variable domains (singleton, finite, numerical constraints) be mixed?

For the first question, two algebraic approaches have been considered, the loose and the tight coupling of spatial calculi [17]. While the loose coupling is too weak to obtain sound and complete reasoning, the tight coupling essentially means to manually develop a combined calculus. Combining individual approaches by translation into a common, expressive formalism would provide an answer to the question. However, formalisms expressive enough to capture a multitude of spatial and temporal relations such as algebraic geometry (e.g., see [3, 19]) lead to infeasible complexity which limits applicability to toy problems.

The second question addresses needs of practical applications in which it is common that some objects to be reasoned about are already identified with concrete entities. This question has recently received attention [10], revealing the specific answer for the region connection calculus (RCC) [13]. For other calculi, this question remains open.

In this paper we are concerned with developing a unified framework for QSTR that provides a solution to both questions and which is applicable to a wide range of qualitative calculi. To this end, we further explore the use linear programming (LP). LP is interesting since it can capture several calculi in an efficient framework, either exactly or by tight approximations. While LP techniques have already been used in QSTR for selected tasks (e.g., [8, 11, 7]), potentials of LP frameworks have not yet been explored thoroughly. We propose a basic language Q_{basic} for QSTR and describe how selected qualitative calculi can be encoded in it. For reasoning with Q_{basic}, translations into LP frameworks are performed. Comparing mixed integer linear programming (MILP) and AND/OR graphs combined with LP, we advocate the latter since it allows sophisticated optimizations that foster efficient reasoning. To further motivate our aims, let us outline a problem from the field of safety in autonomous mobile systems.

1.1 Motivating Problem

Täubig et al. [18] present in “Guaranteeing functional safety: design for provability and computer-aided verification” a supervisory method for an autonomous vehicle to ensure that the vehicle does not issue commands which could (potentially) lead to a collision with a static obstacle. The particular contribution is a formal method for which certification according to IEC 61508 was achieved.

From a QSTR perspective, safe navigation could have been formalized using RCC relations. Considering the primitives illustrated in Fig. 1, we call free_space_sensed the region within sensor range that is free of obstacles. Using $\mathfrak{r}$ as reference to the position of the robot, an intuitive formalization could start as follows:

$$\phi_{\text{safe}} = (\text{braking\_region}(\mathfrak{r}) \ \texttt{pp} \ \text{sensor\_region}(\mathfrak{r})) \tag{1a}$$

The specification would also identify potentially dangerous locations (denoted $\mathfrak{h}$), i.e., positions of obstacles within the braking region but outside sensor range, e.g., due to occlusion. Using $\text{reg}()$ to refer to the region occupied by an obstacle, we obtain

$$\begin{aligned}\phi_{\text{dangerous}} = &((\text{reg}(\mathfrak{h}) \ \texttt{PP} \ \text{braking\_region}(\mathfrak{r}) \\ &\vee (\text{reg}(\mathfrak{h}) \ \texttt{PO} \ \text{braking\_region}(\mathfrak{r}))) \\ &\wedge (\text{reg}(\mathfrak{h}) \ \texttt{DR} \ \text{sensor\_region}(\mathfrak{r}))\end{aligned} \tag{1b}$$

The above formulae essentially describe safety of navigation as considered in [18], they are valid for both static and dynamic obstacles. Extending the specification to consider a moving object $\mathfrak{m}$, its respective braking region needs to be considered too:

$$\psi_{\text{dangerous}} = (\text{braking\_region}(\mathfrak{r}) \ \texttt{PO} \ \text{braking\_region}(\mathfrak{m})) \tag{2}$$

---

[1] This work is partially funded by the DFG, financial support is gratefully acknowledged.
[2] University of Bremen, Germany, email: kreutzma@tzi.de
[3] University of Bamberg, Germany, email: diedrich.wolter@uni-bamberg.de

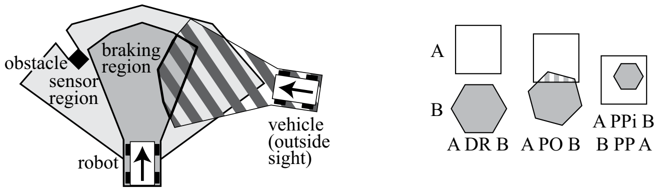

Figure 1. **Left:** regions in safe navigation, overlapping braking regions are dangerous. **Right:** RCC-5 topological relations discrete (DR), partial overlap (PO) and proper part (inverse) (PP, PPi); equality (EQ) not shown

Observe that $\mathrm{braking\_region}(\mathfrak{m})$ may either refer to a concrete region if $\mathfrak{m}$ is observed, but it may also be unknown if $\mathfrak{m}$ is positioned outside sensor range, i.e., $(\mathrm{sensor\_region}(\mathfrak{r})\ \mathtt{DR}\ \mathrm{reg}(\mathfrak{m}))$.

A next step in a formalization could involve traffic rules such "left shall yield to right", saying that the robot has to let vehicles pass which approach from the right, but in turn the robot is allowed to pass by a vehicle approaching from the left:

$$\begin{aligned}\psi'_{\text{dangerous}} = &(\mathrm{braking\_region}(\mathfrak{r})\ \mathtt{PO}\ \mathrm{braking\_region}(\mathfrak{m})) \\ &\wedge (\mathrm{sensor\_region}(\mathfrak{r})\ \mathtt{DR}\ \mathrm{reg}(\mathfrak{m})) \\ &\wedge (\mathfrak{r}\ \mathtt{right}\ \mathfrak{m}),\end{aligned} \tag{3}$$

As can be seen, the example of safe navigation from the literature can be represented with qualitative relations and easily be advanced beyond [18] by considering moving obstacles. However, in order to decide whether an issued driving command is safe, we require means to handle partially grounded information such as the polygonal braking area alongside unknown regions such as the breaking area of a hidden object $\mathfrak{m}$. For considering traffic rules, qualitative representations for region topology (e.g., RCC) and directional knowledge (e.g., OPRA [12]) would need to be mixed. As we will see, the techniques proposed in this paper provide a solution to both problems.

2 Qualitative Spatial and Temporal Reasoning

We briefly introduce key concepts from the field of QSTR necessary in our context. For more detailed coverage we kindly refer to the literature, e.g., [11, 15, 4].

In QSTR, one is involved with representations that are based on finite sets of relations called *base relations* which partition a spatial or temporal domain into meaningful parts. Technically speaking, the set of base relations is jointly exhaustive and pairwise disjoint (JEPD). Due to the set-theoretic semantics of relations, any set of base relations $\mathcal{B}$ induces a Boolean set algebra of qualitative relations $\mathcal{R}_\mathcal{B} = \bigcup_{R \in 2^\mathcal{B}} \left(\bigcup_{r \in R} r\right)$. The Boolean set algebra, in conjunction with relation operation converse $\smile$: $\mathcal{R} \to \mathcal{R}$, $r^\smile = \{(x,y) | (y,x) \in r\}$ and weak composition $\diamond : \mathcal{R} \times \mathcal{R} \to \mathcal{R}$, $r \diamond s = \{(\bigcap_q | (r \circ s) \subseteq q,\ q \in \mathcal{R}\}$ constitutes the algebraic structure of the representation which is also called a *qualitative calculus* [4].

These qualitative relations serve as constraint language to represent constraints like $(X\ \mathtt{DR}\ Y)$, or $(X\ (\mathtt{DR} \cup \mathtt{PO})\ Z)$ whereby DR is a base relation in RCC-5 [13] and $(\mathtt{DR} \cup \mathtt{PO})$ is a respective qualitative relation (see Fig. 1.). Constraint-based reasoning is the single most important form of QSR and it is considered as a decision problem.

Definition 1 **(QCSP).** Given a constraint satisfaction problem (CSP) with variables X ranging over domain D that involves only binary constraints that are qualitative relations in one calculus over domain D, i.e., $c_{i,j} \in \mathcal{R}_\mathcal{B}$ for some set of base relations $\mathcal{B}$ over D. The problem QCSP is then to decide whether there exists a valuation of X with values from D such that all constraints are satisfied.

Since D is typically infinite, special techniques are necessary that allow QCSP to be solved efficiently for various qualitative calculi. The complexity of QCSP is usually NP-complete, while reasoning with base relations only may be in P. There exist however calculi that involve directional relations such as `right` from the motivating example that are inherently NP-hard and, assuming P$\neq$NP, require exponential time algorithms [20].

3 Approaches to Unifying QSTR

With respect to capturing semantics of QSTR, expressive and hence computationally very hard languages are commonly used. For example, algebraic geometry provides a suitable basis to represent many qualitative calculi, but reasoning is only feasible for toy problems [19]. In order to obtain an efficient unified approach to reasoning, few approaches have been proposed so far.

A decomposition of the algebraic structure of calculi has been proposed in [6] that allows QCSP instances to be encoded as SAT instances. However, the method is limited to calculi in which composition-based reasoning can be used to decide consistency (see [15]) which, e.g., excludes RCC in the domain of polygons [16] or calculi involving directional relations.

Linear programming has previously been considered to tackle selected, isolated problems in QSTR. Lee et. al. [8] describe a reasoning method for directional relations that employs an LP solver to check consistency of STAR [14] QCSPs and to compute a realization. In temporal reasoning, LP has previously been considered as a backbone to unifying temporal reasoning, since temporal relations are largely based on linear inequalities. Jonsson and Bäckström [7] describe an approach based on disjunctive linear relations that is similar to ours. In order to extend their idea to spatial relations, we introduce *oracles* that allows us to cope with the higher expressiveness of spatial relations. This requires a new approach to reasoning.

4 A unifying language for QSTR

We now introduce the new language Q_{basic}. The motivation of this language is to separate the translation from QSTR into a common language from translation into a specific LP framework in order to allow different LP backends to be used without the need of re-encoding all spatial calculi. Moreover, Q_{basic} explicates some nice features we obtain as side effects but which are helpful on their own, most notably the propositional closure of qualitative constraints that is not expressible in standard QSTR, e.g., in Q_{basic} we can express $((x\,\alpha\,y) \wedge (y\,\beta\,z)) \vee (x\,\gamma\,y)$.

The primitives of the new language Q_{basic} are systems of inequalities that may contain non-linear elements. When the non-linear elements are externally grounded, the resulting system of inequalities becomes linear. By restricting the domains of the non-linear elements to finite sets we obtain a flexible discretization scheme that easily outperforms any fixed discretization of a spatial or temporal domain. For example, we can choose a finite set of 360 angular 2D directions of lines $\{(\sin(\frac{k}{180}\pi), \cos(\frac{k}{180}\pi)) | k = 0, 1, \ldots 359\}$ when reasoning about lines in the plane, while realizing these directions on a discrete coordinate would require a grid that grows with the number of lines to be positioned.

Definition 2. We call $S^n = \langle \mathcal{O}, G \rangle$ a *system of finite disjunctive linear inequalites over* $\mathbb{R}^n$ *with oracle values* $\mathcal{O}$, where $\mathcal{O}$ is a finite set and G is a mapping $G : \mathcal{O} \to \langle \mathbb{R}^{m_G \times n}, \mathbb{R}^{m_G} \rangle$. We say $s = \langle x, o \rangle \in \langle \mathbb{R}^n, \mathcal{O} \rangle$ is a solution of S^n iff $G(o) = \langle A_o, b_o \rangle$

and $\quad A_o \cdot x \leq b_o$, using the component-wise interpretation of $\leq$ used in LP, i.e., $(x_1, \ldots, x_n) \leq (y_1, \ldots, y_n)$ iff $x_i \leq y_i$ for all $i = 1, \ldots, n$.

Definition 3 (Q_{basic})**.** We call $\langle \mathbb{R}^n, \mathcal{O} \rangle$ the *domain* and $\mathcal{S} = \{S_1^n, \ldots\}$ the set of *symbols*, whereby any symbol S_i^n is a system of finite disjunctive linear inequalities sharing the same oracle $\mathcal{O}$ as defined above. A choice of D and $\mathcal{S}$ is called the signature of our language. Given a signature, we define a Q_{basic} formula ϕ as follows:

$$\phi =_{\text{def}} S_i^n \mid \top \mid \bot \mid \neg\phi \mid \phi \wedge \psi.$$

Given $x \in D$ and a $o \in \mathcal{O}$, we inductively define the notion of a formula ϕ being satisfied in $\langle x, o \rangle$ as follows:

$$
\begin{array}{lll}
x, o \models S^n & \text{iff } \langle x, o \rangle \text{ is a solution of } S^n & (4)\\
x, o \models \top & \text{always} & (5)\\
x, o \models \bot & \text{never} & (6)\\
x, o \models \neg\phi & \text{iff } \langle x, o \rangle \text{ is \textit{not} a solution of } S^n & (7)\\
x, o \models \phi \wedge \psi & \text{iff } x, o \models \phi \text{ and } x, o \models \psi & (8)
\end{array}
$$

The other Boolean connectives are defined as usual.

Corollary 1. *Deciding satisfiability of a Q_{basic} formula is NP-complete*

5 Encoding QCSP in Q_{basic}

This section provides an overview of how QCSP instances for several calculi can be encoded in Q_{basic}. We show how qualitative relations can be represented as systems of finite disjunctive linear inequalites. Due to space constraints, definitions of the individual calculi are omitted. Refer to [11, 4] for definitions and further references.

5.1 Temporal Calculi

As pointed out in [7], temporal relations can be described by linear inequalities. Strictness in the sense $x < y$ can be resolved by introducing a fixed $\varepsilon > 0$ and rewriting to $x + \varepsilon \leq y$ since the qualitative temporal relations considered do not rely on absolute values.

5.2 Direction Calculi

Given a vector $\vec{v} \in \mathbb{R}^2$, we call $\vec{v}^{\perp}$ its left normal obtained by 90° counter-clockwise rotation. Given two (variable) points $p, q \in \mathbb{R}^2$ and a fixed orientation expressed as a vector $\vec{v} \in \mathbb{R}^2$, we define the following constraints by translation to Q_{basic}:

$$
\begin{array}{llll}
p \,\texttt{left}_{\vec{v}}\, q & =_{\text{def}} \vec{q}^T \cdot \vec{v}^{\perp} - \vec{p}^T \cdot v^{\perp} \leq 0 & (q \text{ left of } p) & \\
p \,\texttt{right}_{\vec{v}}\, q & =_{\text{def}} \vec{p}^T \cdot \vec{v}^{\perp} - \vec{q}^T \cdot v^{\perp} \leq 0 & (q \text{ right of } p) & \\
p \,\texttt{front}_{\vec{v}}\, q & =_{\text{def}} \quad \vec{p}^T \cdot \vec{v} - \vec{q}^T \cdot v \leq 0 & (q \text{ in front of } p) & (9)\\
p \,\texttt{back}_{\vec{v}}\, q & =_{\text{def}} \quad \vec{q}^T \cdot \vec{v} - \vec{p}^T \cdot v \leq 0 & (q \text{ behind } p) &
\end{array}
$$

The relations $\texttt{left}_{\vec{v}}$, $\texttt{right}_{\vec{v}}$, $\texttt{front}_{\vec{v}}$, $\texttt{back}_{\vec{v}}$ are not *pairwise disjoint* (they overlap in one quadrant) but they are *jointly exhaustive*.

Theorem 1. *Let ϕ be a propositional formula with atoms of the kind $(\mathbf{x}\ \mathtt{R}\ \mathbf{y})$, where* R *is a relation as defined above. Let* $\text{var}(\phi)$ *denote the number of (distinct) variables in ϕ and let* $\text{rel}(\phi)$ *denote the number of (distinct) relations in ϕ, then ϕ can be translated into a Q_{basic} formula with signature $D = \mathbb{R}^{2\,\text{var}(\phi)}$, $|(S)| = \text{rel}(\phi)$, and $\mathcal{O} = \emptyset$.*

Proof. Let $I : V \to \{1, \ldots, n\}$ be a bijective mapping between the variables and corresponding dimension in $\mathbb{R}^{2\,\text{var}(\phi)}$. We define

$$H_i =_{\text{def}} \begin{pmatrix} \underbrace{0 \ldots 0}_{2\cdot(I(i)-1)} & 1 & 0 & 0 \ldots \\ 0 \ldots 0 & 0 & 1 & 0 \ldots \end{pmatrix}, \quad H_{i,j} =_{\text{def}} \begin{pmatrix} H_i \\ H_j \end{pmatrix}.$$

In the given formula ϕ, replace all atoms $(x_i\ R_{\vec{v}}\ x_j)$ by $S_k = \langle \{\}, \langle H_{i,j}^T A_{R_{\vec{v}}} H_{i,j}, 0 \rangle \rangle$, where $A_{R_{\vec{v}}}$ is the corresponding matrix to represent inequality as given by Eq. 9. This yields a Q_{basic} formula with the signature, $D = \mathbb{R}^{2\,\text{var}(\phi)}$, $\mathcal{O} = \{\}$, and $\mathcal{S}$ as the set comprising all S_k defined above. □

Consider two arbitrarily fixed vectors $\vec{s}$ and $\vec{t}$ such that the counter-clockwise angle between $\vec{s}$ and $\vec{t}$ does not exceed 180°. A (variable) point q with respect to a (variable) point p is said to be inside the sector spanned by $\vec{s}$ and $\vec{t}$, iff:

$$(p \,\texttt{left}_{\vec{s}}\, q) \wedge (p \,\texttt{right}_{\vec{t}}\, q) \tag{10}$$

All cardinal direction calculi considered in the literature are either based on half-plane or sector membership, whereby half-plane normals and sectors are globally aligned to one of finitely many directions. This makes mapping QCSP instances to Q_{basic} with any of these calculi straightforward using either Eq. 10 or $\texttt{front}_{\vec{n}}$ where $\vec{n}$ denotes the respective half-plane normal. No oracle needs to be introduced. Since all these calculi are scale-invariant like temporal calculi, the same approach of introducing ε can be applied to represent truly $\texttt{left}_{\vec{v}}$, $\texttt{right}_{\vec{v}}$, etc. Applicability to the most important cardinal direction calculi is shown in Tab. 1.

Theorem 2. *StarVars [8] can be represented by Q_{basic}.*

Proof. StarVars, like Star [14], employs sector-shaped spatial relations. The sectors in StarVars are rotated by an undetermined angle $\frac{2i}{2^N}\pi,\ i = 0, \ldots 2^N - 1$ for a fixed N. Choosing these angles as oracles, the construction of the Q_{basic} formula follows directly from [8] which also employs an LP algorithm to decide consistency. □

Theorem 3. *$\mathcal{OPRA}$ can be mapped to Q_{basic} if the domain of directions is restricted to a finite set.*

Proof. Interpreted over finite domain of directions, $\mathcal{OPRA}$ relations can be represented as two conjuncts of StarVars relations [8]. □

5.3 Region Connection Calculus

In this work we only consider planar regions in form of simple, i.e, not self-intersecting polygons. We start with convex polygons since the mappings can then be generalized to non-convex polygons by considering a convex partitioning and disjunctively adjoining the linear programs.

First note that the relation saying that a point is located inside a simple convex polygon positioned at an unknown origin can be represented by a LP. This is due to the point-in-polygon test being based on half-plane membership tests which are linear inequalities and stay linear if the whole polygon is translated by unknown x, y. For convex polygons, point-outside-polygon can also be modeled by disjunctively adjoining the negated clauses of the point-in-polygon test.

Corollary 2. *If two simple convex polygons do not share a common point, then there exists a line parallel to one edge which separates the space between both polygons.*

This fact grants a mapping for the RCC relation *discrete* saying that regions do not share a common interior part. For simple convex polygons, we disjunctively choose one edge as the dividing line. Let two simple convex polygons P and Q be defined by vertices $v_1^P, \ldots, v_k^P$ and $v_1^Q, \ldots, v_m^Q$ in counter-clockwise orientation. We write e_i^P to refer to edge $v_i^P, v_{(i+1) \mod k}^P$ and d_i^P to refer to direction $v_{(i+1) \mod k}^P - v_i^P$ and obtain:

$$\begin{aligned}(P \ \mathtt{dr}_{\text{conv}}\ Q) =_{\text{def}} & \bigvee_{\vec{e}_i^P} \bigwedge_{v_j^Q} (v_i^P \ \mathtt{right}_{\vec{d}_i^P}\ v_j^Q) \\ & \vee \bigvee_{\vec{e}_i^Q} \bigwedge_{v_j^P} (v_i^Q \ \mathtt{right}_{\vec{d}_i^Q}\ v_j^P)\end{aligned} \tag{11}$$

Analogously, $\mathtt{dc}_{\text{conv}}$ can be defined, except that touching points need to be excluded by using $\neg(v_i^P \ \mathtt{left}\ v_j^Q)$ instead of $(v_i^P \ \mathtt{right}\ v_j^Q)$. Given P as above we can express that point x lies on the edge e_i^P, i.e., between v_i^P and v_{i+1}^P, including both vertices.

$$\begin{aligned}(e_i^P \ \mathtt{cont}\ x) =_{\text{def}} & (v_i^P \ \text{left}_{(\vec{v}_{i+1}^P - \vec{v}_i^P)}\ x) \wedge (v_i^P \ \text{right}_{(\vec{v}_{i+1}^P - \vec{v}_i^P)}\ x) \\ & \wedge (v_i^P \ \text{front}_{(\vec{v}_{i+1}^P - \vec{v}_i^P)}\ x) \wedge (v_{i+1}^P \ \text{back}_{(\vec{v}_{i+1}^P - \vec{v}_i^P)}\ x),\end{aligned} \tag{12}$$

External connection can be mapped to Q_{basic} as follows:

$$\begin{aligned}(P \ \mathtt{t}_{\text{conv}}\ Q) =_{\text{def}} & \bigvee_{e_i^P} \Big[\bigwedge_{v_j^Q} (v_i^P \ \mathtt{right}_{(\vec{v}_{i+1}^P - \vec{v}_i^P)}\ v_j^Q) \\ & \wedge \bigvee_{v_j^Q} (e_i^P \ \mathtt{cont}\ v_j^Q) \Big] \\ (P \ \mathtt{ec}_{\text{conv}}\ Q) =_{\text{def}} & (P \ \mathtt{t}_{\text{conv}}\ Q) \vee (Q \ \mathtt{t}_{\text{conv}}\ P)\end{aligned} \tag{13}$$

Theorem 4. *RCC-5 and RCC-8 [13] can be mapped to Q_{basic} for the domain of simple (i.e., not self-intersecting) polygons in 2D space that involve at most N vertices each.*

Proof. We need to show how the relations of RCC-8 can be stated in Q_{basic}, RCC-5 relations can then be obtained by disjunctive combinations, e.g., $(P \ \mathtt{DR}_{\text{RCC-5}}\ Q) = (P \ \mathtt{DC}_{\text{RCC-8}}\ Q) \vee (P \ \mathtt{EC}_{\text{RCC-8}}\ Q)$. The vertex limit N is required to obtain finite formulae. For RCC-8, the following mapping can be employed:

$$(P \ \mathtt{dc}\ Q) =_{\text{def}} \bigwedge_{P^C \in C_P} \bigwedge_{Q^C \in C_Q} (P^C \ \mathtt{dc}_{\text{conv}}\ Q^C) \tag{14}$$

$$\begin{aligned}(P \ \mathtt{ec}\ Q) =_{\text{def}} & \bigvee_{P^C \in C_P} \bigvee_{Q^C \in C_Q} (P^C \ \mathtt{ec}_{\text{conv}}\ Q^C) \\ & \wedge \bigwedge_{P^C \in C_P} \bigwedge_{Q^C \in C_Q} (P^C \ \mathtt{dr}_{\text{conv}}\ Q^C)\end{aligned} \tag{15}$$

Given three fresh variables τ_1, τ_2, τ_3 denoting points:

$$\begin{aligned}(P \ \mathtt{po}\ Q) =_{\text{def}} & \big((\tau_1 \ \mathtt{inside}\ P) \wedge (\tau_1 \ \mathtt{inside}\ Q)\big) \\ & \wedge\big((\tau_2 \ \mathtt{inside}\ P) \wedge \neg(\tau_2 \ \mathtt{inside}\ Q)\big) \\ & \wedge\big(\neg(\tau_3 \ \mathtt{inside}\ P) \wedge (\tau_3 \ \mathtt{inside}\ Q)\big)\end{aligned} \tag{16}$$

For containment it is not sufficient that all vertices of one polygon P are inside another polygon Q, see Fig. 2. Let I_Q denote edges introduced by the convex partitioning. If an edge E of P overlaps with a sequence of adjacent convex parts of Q, all I_Q's of this sequence need to cross, i.e, one endpoint of I_Q lies left of and the other right of E. In the following, this is denoted by the formula $(P \otimes Q)$.

$$(P \ \mathtt{pp}\ Q) =_{\text{def}} \bigwedge_{v_j^P} (v_j^P \ \mathtt{inside}\ Q) \wedge (P \otimes Q)$$

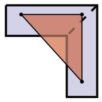

Figure 2. Convex region (red) partially overlapping a non-convex region (blue) although all vertices of the red region are inside the blue region.

$$\begin{aligned}(P \ \mathtt{tpp}\ Q) =_{\text{def}} & (P \ \mathtt{pp}\ Q) \\ & \wedge \Big[\bigvee_{e_i^P} \bigvee_{v_j^Q} (e_i^P \ \mathtt{contains}\ v_j^Q) \\ & \vee \bigvee_{e_i^Q} \bigvee_{v_j^P} (e_i^Q \ \mathtt{contains}\ v_j^P) \Big]\end{aligned} \tag{17}$$

$$(P \ \mathtt{ntpp}\ Q) =_{\text{def}} (P \ \mathtt{pp}\ Q) \wedge \bigwedge_{v_j^Q} \neg(v_j^Q \ \mathtt{inside}\ P) \tag{18}$$

Due to space constraints we omit converse relations $\mathtt{ntppi}, \mathtt{tppi}$ and equality $\mathtt{eq}$, as well as $(P \otimes Q)$. □

6 Using Spatial Reasoning to Reduce Formula Size

Key to making reasoning in Q_{basic} efficient is reducing formula size. Aside from rewriting and simplification, we also apply classic QSTR reasoning methods to prune away implicit sub-formulae. The process of simplification can be interwoven with how QCSP instances are translated into Q_{basic} formulae to avoid uneccesarily generating systems of finite disjunctive linear programs.

Removing Redundant Information In case of partially grounded information, we first check whether constraint relations are declared between two grounded entities. Then, we check if the relation holds and replace it accordingly by $\top$ or $\bot$.

Given a set of constraints over a single qualitative calculus, we can apply composition-based constraint propagation to identify redundant constraints, e.g., in the set $\{(A \ \mathtt{dc}\ B), (C \ \mathtt{ntpp}\ B), (A \ \mathtt{dc}\ C)\}$ the constraint $(A \ \mathtt{dc}\ C)$ is redundant since it is implied by the other: A must be disconnected form C since A is already disconnected from a container of C. Unfortunately, determining the minimal set of constraints is NP-complete [5], so we only perform a greedy search.

Avoiding disjunctions There are several ways of encoding a spatial relation in Q_{basic}. To avoid disjunctions, we consider alternative mappings stored in a table and choose the option that introduces the fewest disjunctions. For example, instead of encoding $\neg(A \ \mathtt{dc}\ B)$ at the cost of several disjunctions as explained further below, it can simply we rewritten by saying there exist a common point τ, either truly inside or at their border: $(\tau \ \mathtt{inside}\ A) \wedge (\tau \ \mathtt{inside}\ B)$. Since spatial calculi comprise a jointly exhaustive set of relations, negation can sometimes be rewritten with less disjunctions by considering the mapping of complementary relations.

7 Deciding Q_{basic} and Computing Realizations

In this section we introduce two translations of Q_{basic} to LP frameworks, namely mixed integer linear programming (MILP) and AND/OR graphs of LPs. While existing MILP solvers provide all functionality for deciding consistency of a Q_{basic} formula encoded as a mixed integer linear program, we give an incremental method for solving formulas encoded as AND/OR graphs of LPs.

Definition 4. Given an finite set $\mathcal{O}$ and a system of finite disjunctive linear inequalites $S = \langle \mathcal{O}, G \rangle$ we say for a $o \in \mathcal{O}$

$$[o]_S =_{\text{def}} \{o' \in \mathcal{O} \mid G(o') = G(o)\}$$

Algorithm 1. Translate (normalized) Q_{basic} formula ϕ to Mixed-Integer Linear Program L, with $x, v_* \in \mathbb{R}^n, y_* \in \{0,1\}$

$L \leftarrow$ *empty* mixed integer linear program
for all $S \in \phi$ **do**
  $\mathcal{O}_S \leftarrow \{[o]_S \mid o \in \mathcal{O}\}$
  for all $[o]_S \in \mathcal{O}_S$ **do**
    chose a $o \in [o]_S$
    $\langle A, b\rangle \leftarrow G_S(o)$
    $L \leftarrow L \cup Av_{S,[o]_S} \leq by_{S,[o]_S}$ ▷ Add inequalities
    $L \leftarrow L \cup \left[y_{S,[o]_S} \leq y_S\right]$ ▷ Add *relation* implication
    $L \leftarrow L \cup \left[y_{S,[o]_S} \leq \sum_{o\in[o]_S} y_o\right]$ ▷ oracle implication
  end for
  ▷ Aggregate Disjunction Constrains
  $L \leftarrow L \cup \left[x = \sum_{[o]_S \in \mathcal{O}_S} v_{S,[o]_S}\right]$
  $L \leftarrow L \cup \left[0 \leq v_{S,[o]_S} \leq y_{S,[o]_S} U \text{ for all } [o]_S \in \mathcal{O}_S\right]$
  $L \leftarrow L \cup \left[\sum_{[o]_S \in \mathcal{O}_S} y_{S,[o]_S} = 1\right]$
end for
$\psi \leftarrow$ replace each S in ϕ with y_S
$\psi_{\text{CNF}} \leftarrow$ conjunctive normal form of ψ
for all disjunctive clauses Q in ψ_{CNF} **do**
  $L \leftarrow L \cup \left[\sum_{y_S \in Q} y_S \geq 1\right]$ ▷ All y_S are not negated
end for

is the induced congruent set of o with respect to S. In other words, $[o]_S$ collects all oracle variables that lead to the same linear program.

In order to decide satisfiability of a Q_{basic} formula and to obtain realizations for satisfiable formulae, we first perform normalization. First, we rewrite Boolean operators to only have $\vee, \wedge$ and we remove $\top, \bot$ by absorption, e.g., $\phi \vee \bot \mapsto \phi$. Second, negation is moved inward such that we only have negated atoms $\neg S_k^n$. This negated atom can be replaced by a positive one at the cost of introducing disjunctions which select an inequality from S_k^n that is violated. We can thus assume to be given a Q_{basic} formula without negation.

7.1 Mapping Q_{basic} to MILP

We base our translation from Q_{basic} to mixed-integer linear programming upon the fundamental work of Balas[1] on disjunctive linear programming. Further we draw inspirations from Lee and Grossman[9], who describe a method for approximating non-linear disjunctions, which requires upper bounds u_i on all variables.

The general approach for a given disjunction over k sets of linear inequalities ($A_i x \leq b_i$) is that x is disaggregated into $x = v_1 + ... + v_k$ and for each set of linear inequalities a variable $y_i \in \{0,1\}$ is defined. Then, $A_i v_i \leq b_i y_i$ constitutes the program, replacing the original set of linear inequalities. Choosing $y_i = 0$ effectively disables the inequality and $y_i = 1$ enables it. A further inequality $v_i \leq y_i u_i$ is added, forcing v_i to zero if the inequality is disabled.

In our case, we have a disjunction for each $S \in \phi$ over the oracle values. The only thing left is to ensure that at least one of the disjunctions is active if the corresponding y_S is: $\sum y_i \geq y_S$.

Alg. 1 shows the complete procedure in algorithmic form. If the resulting MILP has a solution, that solution is also a realization of the Q_{basic} formula[4]. Which oracle value was used, can also be read of from the MILP solution. If no solution was found, the Q_{basic} formula is not realizable.

7.2 Incremental Expansion of Linear Programms

Considering the parse tree of a Q_{basic} formula, we can regard the formula as AND/OR graph whose leaves are systems of finite disjunctive linear inequalites. In order to compute a solution we perform a depth-first search with backtracking as shown in Alg. 2. The starting parameters are the original AND/OR tree T, the partial grounding LP encoded in LP[5], and the set of oracle values $\mathcal{O}$. A solution found at a node is propagated upwards, accumulating the (pure) linear programs (line 29). The algorithm either returns a realization, the corresponding LP, and the oracle values or $\emptyset, \emptyset, \emptyset$ to signal unsatisfiability.

Algorithm 2. Incremental Expansion

1: **procedure** REALIZETREE($T, LP, \mathcal{O}$)
2:   **if** T_{root} is conjunction **then** ▷ And-Node
3:     $C \leftarrow$ select one child of T
4:     **while** $\mathcal{O} \neq \emptyset$ **do**
5:       $S, LP', \mathcal{O}' \leftarrow$ REALIZETREE($C, LP, \mathcal{O}$)
6:       $\mathcal{O} \leftarrow \mathcal{O} \setminus \mathcal{O}'$
7:       **if** T has other children **then**
8:         $S, LP', \mathcal{O}'' \leftarrow$ REALIZETREE($T \setminus C, LP', \mathcal{O}'$)
9:       **end if**
10:       **if** $S \neq \emptyset$ **then**
11:         return $S, LP', \mathcal{O}''$
12:       **end if**
13:     **end while**
14:     return $\emptyset, \emptyset, \emptyset$
15:   **else if** T_{root} is disjunction **then** ▷ Or-Node
16:     **for all** children C of T **do**
17:       $S, LP', \mathcal{O}' \leftarrow$ REALIZETREE($C, LP, \mathcal{O}$)
18:       **if** $S \neq \emptyset$ **then**
19:         return $S, LP', \mathcal{O}'$
20:       **end if**
21:     **end for**
22:     return $\emptyset, \emptyset, \emptyset$
23:   **else** ▷ Symbol/Relation
24:     **for** T induced congruent sets $\mathcal{O}' \subset \mathcal{O}$ **do**
25:       $o \leftarrow$ select from $\mathcal{O}'$
26:       $LP' \leftarrow G_p(o)$
27:       $S \leftarrow$ SOLVE($LP \cup LP'$)
28:       **if** $S \neq \emptyset$ **then**
29:         return $S, LP \cup LP', \mathcal{O}'$
30:       **end if**
31:     **end for**
32:     return $\emptyset, \emptyset, \emptyset$
33:   **end if**
34: **end procedure**

calculus	encoding properties
Allen's interval relations	✓
Block Algebra	✓
Cardinal Direction Calculus	✓
Dipole Calculus	discretized 2D directions
INDU	✓
LR calculus	discretized 2D directions
$\mathcal{OPRA}$	discretized 2D directions
Point algebra	✓
Positional point calculi	discretized 2D directions
Qualitative Trajectory Calculi	via encoding to $\mathcal{OPRA}$
Region Cardinal Dir. Calc.	N-vertex polygons or polyhedra only
RCC	N-vertex polygons or polyhedra only
STAR	✓
StarVars	✓

Table 1. Encoding properties of qualitative calculi in Q_{basic}

8 Practical Analysis

We evaluate the performance of the strategies MILP and incremental expansion experimentally. Since our method actually computes a realization for any consistent QCSP instance, comparison with algorithms that merely check for consistency but cannot compute a realization are not adequate. Additionally, both strategies are compared against the results published for StarVars reasoning algorithm[8] that also computes a realization. This comparison is particularly interesting since a

[4] With a lot of extra variables y_i which however can easily be filtered out

[5] If not applicable an empty LP is provided

n	m	StarVars[8]	MILP	IncExpand
4	4	0.64 ±0.39	0.02 ±0.01	0.13 ±0.00
4	8	1.15 ±0.71	0.14 ±0.04	0.20 ±0.01
4	16	2.01 ±1.13	1.08 ±0.30	0.34 ±0.01
4	32	2.63 ±1.61	8.26 ±2.67	0.62 ±0.02
5	4	1.06 ±0.58	0.06 ±0.01	0.19 ±0.01
5	8	1.66 ±1.26	0.44 ±0.10	0.30 ±0.01
5	16	2.56 ±2.14	4.72 ±1.15	0.50 ±0.02
5	32	4.35 ±3.87	34.11 ±9.91	0.92 ±0.02
6	4	2.55 ±0.00	0.14 ±0.03	0.27 ±0.01
6	8	3.16 ±1.64	1.31 ±0.27	0.42 ±0.02
6	16	4.27 ±3.48	12.96 ±2.91	0.69 ±0.03
6	32	6.10 ±5.88	109.46 ±24.15	1.25 ±0.04
7	4	6.83 ±0.10	0.28 ±0.05	0.37 ±0.01
7	8	7.55 ±0.10	3.23 ±0.55	0.55 ±0.02
7	16	8.30 ±1.82	36.21 ±8.03	0.91 ±0.03
7	32	8.76 ±2.68	310.05 ±73.63	1.61 ±0.02

Table 2. Compute time in seconds with standard deviation for 100 random scenarios for n entities with m distinct orientations

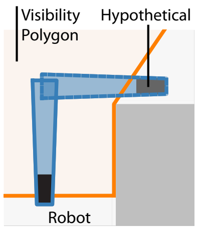

Figure 3. Realization computed by our algorithm when provided with shape of breaking regions, outline of the obstacle (grey box), and $\psi'_{\text{dangerous}}$ from Section 1.1

StarVars requires a large number of oracle values to be introduced and its parameters allow controlling problem size n (number of entities, $O(n^2)$ constraints) and required oracle values ($|\mathcal{O}| = m \cdot n$) independently. For each combination of n and m we randomly generate 100 QCSP instances, using base relations as constraints.

We implemented the translation described in Alg. 1, 2 in Python. For MILP and LP solving, we rely on lp_solve [2]. Tab. 2 gives compute times measured on an Intel Core i7 @3.4GHz with 16 GB RAM. The results in the column StarVars are those as reported in [8] using a different, slower machine, and are thus not comparable as such, but sufficient for a qualitative comparison.

Discussion of the Results

Let us first consider compute times for MILP shown in Tab. 2. The time increases with problem size and, more significantly, with respect to m. This likely results from the translation into MILP since unfolding disjunctions leads to exponential problem size. The steep scaling wrt. m also leads to longer compute times than reported for the handcrafted StarVars algorithm on a slower machine. For problems with few disjunctions (e.g., $m = 4$), MILP can outperform incremental expansion. For most of the configurations tested, incremental expansion shows superior performance though. This is due to the algorithm exploiting the structure of the formula, something that gets lost in the translation to MILP. In comparison to the results obtained for the original StarVars algorithm handcrafted for these constraints, we observe a similar scaling with respect to increasing m.

In summary, incremental expansion provides a practical method for reasoning with Q_{basic} formulae.

9 Summary and Conclusion

This paper outlines a practically relevant answer to two longstanding questions in qualitative spatial and temporal reasoning. By encoding spatial and temporal relations into an LP framework, we are able to represent the important domains of points, lines, and polygons. We show how relations from various qualitative calculi can be expressed in our framework, including directional knowledge. This allows distinct qualitative representations to be combined and jointly to be reasoned about. Doing so, we advance earlier work in temporal reasoning by Jonsson and Bäckström [7]. The algorithm of incremental expansion for solving AND/OR LP problems is however more efficient than using disjunctive linear relations like in their work, since incremental expansion avoids exponential blow up of disjunctions occurring with disjunctive linear relations or MILP. While this paper proposes the unifying language Q_{basic} that can be tackled with LP techniques, identifying the most efficient reasoning algorithms is subject to further investigations.

REFERENCES

[1] E. Balas, 'Disjunctive programming and a hierarchy of relaxations for discrete optimization problems', *SIAM J ALGEBRA DISCR*, **6**(3), 466–486, (1985).
[2] M. Berkelaar, K. Eikland, and P. Notebaert. lpsolve : Open source (mixed-integer) linear programming system. http://lpsolve.sourceforge.net.
[3] M. Bhatt, J. H. Lee, and C. Schultz, 'CLP(QS): a declarative spatial reasoning framework', in *Proc. of Cosit*, pp. 210–230, (2011).
[4] F. Dylla, T. Mossakowski, T. Schneider, and D. Wolter, 'Algebraic properties of qualitative spatio-temporal calculi', in *Proc. of COSIT*, volume 8116 of *LNCS*, pp. 516–536. Springer, (2013).
[5] G. Gottlob, 'On minimal constraint networks', in *Principles and Practice of Constraint Programming CP 2011*, ed., J. Lee, volume 6876 of *LNCS*, 325–339, Springer Berlin Heidelberg, (2011).
[6] J Hue, M Westphal, and S Wölfl, 'An Automatic Decomposition Method for Qualitative Spatial and Temporal Reasoning', in *Proc. ICTAI-2012*, pp. 588–595. University Freiburg, SFB/TR 8, (2012).
[7] P. Jonsson and C. Bäckström, 'A unifying approach to temporal constraint reasoning', *Artif. Intell.*, **102**(1), 143 – 155, (1998).
[8] J. H. Lee, J. Renz, and D. Wolter, 'StarVars—effective reasoning about relative directions', in *IJCAI*, ed., F. Rossi. IJCAI, (2013).
[9] S. Lee and I. E. Grossmann, 'New algorithms for nonlinear generalized disjunctive programming', *COMPUT CHEM ENG*, **24**(9), 2125–2141, (2000).
[10] S. Li, W. Liu, and S. Wang, 'Qualitative constraint satisfaction problems: An extended framework with landmarks', *Artif. Intell.*, **201**, 32–58, (2013).
[11] G. Ligozat, *Qualitative Spatial and Temporal Reasoning*, Wiley, 2011.
[12] T. Mossakowski and R. Moratz, 'Qualitative reasoning about relative direction of oriented points', *Artif. Intell.*, **180–181**, 34–45, (2012).
[13] D. A. Randell, Z. Cui, and A. G. Cohn, 'A spatial logic based on regions and connection', in *Proc. of KR'92*, eds., B. Nebel, C. Rich, and W. Swartout, 165–176, Morgan Kaufmann, San Mateo, CA, (1992).
[14] J. Renz and D. Mitra, 'Qualitative direction calculi with arbitrary granularity', in *Proc. of PRICAI-04*, volume 3157 of *LNAI*, pp. 65–74. Springer, (2004).
[15] J. Renz and B. Nebel, 'Qualitative spatial reasoning using constraint calculi', in *Handbook of Spatial Logics*, eds., M. Aiello, I. E. Pratt-Hartmann, and J. F.A.K. van Benthem, 161–215, Springer, (2007).
[16] R.Kontchakov, I. E. Pratt-Hartmann, F. Wolter, and M. Zakharyaschev, 'Spatial logics with connectedness predicates', *LOG METH COMPUT SCI*, (2010).
[17] S.Wölfl and M.Westphal, 'On combinations of binary qualitative constraint calculi', in *Proc. of IJCAI*, pp. 967–973, (2009).
[18] H. Täubig, U. Frese, C. Hertzberg, C. Lüth, S. Mohr, E. Vorobev, and D. Walter, 'Guaranteeing functional safety: design for provability and computer-aided verification', *AUTON ROBOT*, **32**(3), 303–331, (2012).
[19] D. Wolter, 'Analyzing qualitative spatio-temporal calculi using algebraic geometry', *Spatial Cognition and Comp.*, **12**(1), 23–52, (2012).
[20] D. Wolter and J. H. Lee, 'Qualitative reasoning with directional relations', *Artif. Intell.*, **174**(18), 1498–1507, (2010).