CLP(QS): A Declarative Spatial Reasoning Framework

Mehul Bhatt, Jae Hee Lee, Carl Schultz

{bhatt, jay, cschultz}@informatik.uni-bremen.de
Spatial Cognition Research Center (SFB/TR 8)
University of Bremen, Germany

Abstract. We propose CLP(QS), a *declarative spatial reasoning* framework capable of representing and reasoning about high-level, qualitative spatial knowledge about the world. We systematically formalize and implement the semantics of a range of *qualitative spatial calculi* using a system of non-linear polynomial equations in the context of a classical *constraint logic programming* framework. Whereas CLP(QS) is a general framework, we demonstrate its applicability for the domain of Computer Aided Architecture Design. With CLP(QS) serving as a prototype, we position declarative spatial reasoning as a general paradigm open to other formalizations, reinterpretations, and extensions. We argue that the accessibility of qualitative spatial representation and reasoning mechanisms via the medium of high-level, logic-based formalizations is crucial for their utility toward solving real-world problems.

Keywords: geometric and qualitative spatial reasoning, constraint logic programming, declarative programming, spatial computing, architecture design

1 Introduction

Declarative programming is a paradigm concerned with the development of computational models that can solve problems directly from high-level domain specifications consisting of the core logic of computation, without a complete specification of the precise flow of control [37]. It is a model of computation aiming to solve a problem by specifying its '*what*', as opposed to its '*how*'. As an example, consider the definition of a square root: $\{\sqrt{x} = y, \{ y \geq 0, \; y^2 = x\}\}$. This definition of a square root does not say anything about the actual computation of square roots; it is a declarative specification of the concept of a square root constituting a rather extreme form of declarativeness: the holy grail of research in declarative programming. Within computer science, or specifically Artificial Intelligence (AI), several declarative computation paradigms, programming languages, and frameworks exist, primarily among them based on the theoretical foundations of Logic Programming (LP) [17, 33], Constraint Logic Programming (CLP) [29] and derivatives such as Abductive Constraint Logic Programming (ACLP) [31], Answer-Set Programming (ASP) [36], and Functional Programming [1]. Knowledge Representation and Reasoning (KR) research in AI is concerned with the development of formalisms and systems that can deal with high-level knowledge, ranging from the abstract mathematical, to the ontological, spatial, temporal, action and change driven etc [2, 50]. The main focus of this paper is the integration of one such specialization in the field of Artificial Intelligence, i.e., qualitative spatial representation and reasoning [14], with a declarative approach to problem solving, i.e., constraint logic programming.

▷ **Qualitative Spatial Representation and Reasoning** Qualitative Spatial Representation and Reasoning (QSR) provides a commonsensical interface to abstract and reason about spatial information. Formal methods in the field of QSR consist of *qualitative spatial calculi*, which are relational-algebraic systems pertaining to one or more aspects of space such as *topology, orientation, direction, size* [14].[1] The basic tenets in QSR consist of constraint-based reasoning algorithms over an infinite (spatial) domain to solve *consistency* problems in the context of qualitative spatial calculi. The key idea here is to partition an infinite quantity space into finite disjoint categories, and utilize the special relational properties of such a partitioned space for reasoning purposes.

The application of QSR mechanisms is a topic that is gaining significant momentum in the community [10]. One objective, from the viewpoint of these application goals, is the use of qualitative spatial abstraction and reasoning mechanisms that have been developed in QSR in domains involving representing and reasoning with static and dynamic spatial information., e.g., *spatial design*, *geographic information systems*, *cognitive robotics* etc. From a methodological viewpoint, the integration of formal qualitative spatial and temporal techniques within general commonsense reasoning frameworks in KR is a crucial next-step for their applicability in real-world domains [6].

In this paper, we are concerned with representing and reasoning with qualitative abstractions of spatial information within a declarative programming framework. A declarative programming interface to qualitative spatial representation and reasoning techniques serves as a natural way for systems and programmers to seamlessly access specialized spatial reasoning capabilities toward the development of intelligent spatial systems. With this hypothesis, we propose a Declarative Spatial Reasoning framework and demonstrate its applicability for real-world problem solving.

▷ **Declarative Spatial Reasoning**. We propose declarative spatial reasoning as a general paradigm for the integration of specialized qualitative spatial representation and reasoning techniques with declarative programming languages and frameworks such that qualitative spatial, and geometric reasoning capabilities may be directly exploited within state-of-the-art declarative knowledge representation and reasoning frameworks in Artificial Intelligence. A crucial motivation is to be able to declaratively specify and solve real-world problems related to spatial representation and reasoning. From the viewpoint of the main proposal presented in this paper:

---

[1] The mereotopological RCC calculus, which was first proposed within a first-order logical framework, is an exception.

> *Declarative spatial reasoning denotes the ability of declarative programming frameworks to handle* spatial objects *and the* spatial relationships *amongst them as* native *entities, e.g., as is possible with concrete domains of Integers, Reals and Inequality relationships. The objective is to enable points, oriented points, directed line-segments, regions, and topological and orientation relationships amongst them as* first-class *entities within declarative frameworks in AI.*

The principal advantage of this mode of computation is that high-level specifications and constraints about the spatial domain of interest may be expressed in a manner that is similar or close to their conceptualization and modeling, e.g., as high-level clauses (i.e., *facts*, *rules*), of the declarative semantics of a logic program. Although this paper is categorically focused on spatial abstractions defined in QSR, our general approach lends itself to re-interpretations and extensions with other perspectives such as Visual and Diagrammatic Representations, as well as other cognitively-driven modalities or mental models of space.

Basic Approach and Contribution

Spatial representation and reasoning problems may be approached in a multitude of representational forms, ranging from qualitative, visual, diagrammatic, purely geometric etc. The underlying thread in any approach is that the spatial domain is special, and similar to other specialized sub-disciplines in AI (e.g., ontological, action and change driven), the development of an intelligent spatial representation and reasoning capability, regardless of its methodological underpinnings, requires its own set of specialized techniques and algorithms. Keeping in mind such specializations, the fundamental question that arises is:

> *By what means may specialized qualitative spatial representation and reasoning techniques be integrated within general knowledge representation and reasoning techniques in Artificial Intelligence*?

This question is rather open-ended, and acquires several different interpretations depending on the precise KR technique, and aspects of space, actions, events, and change being considered [6]. In this paper, we focus on one concrete interpretation:

> *How may specialized qualitative spatial representation and reasoning techniques be embedded within the state-of-the-art declarative programming approach of CLP*?

An embedding of this nature (i.e., with methods such as CLP) would imply that spatial representation and reasoning problems may be directly modeled within state-of-the-art KR techniques, and thus be directly usable by the direct benefactors of AI tools and frameworks, in this case, users of CLP. This particular interpretation, and its prototypical implementation in this paper, is guided by the hypothesis that:

> *The accessibility of generic qualitative spatial abstraction and reasoning techniques, e.g., qualitative spatial calculi, qualification and consistency algorithms,*

via the interface of declarative programming frameworks such as constraint logic programming is crucial, and presents one model for their applicability toward real-world problem solving.

This paper illustrates its key propositions by developing the declarative spatial reasoning framework CLP(QS), which is an integration of a polynomial-based CLP framework with a qualitative spatial domain $\mathcal{QS}$ consisting of point and region based spatial calculi. To achieve the integration, we characterize the spatial domain $\mathcal{QS}$ using a system of non-linear polynomial equations in a manner that is consistent with the chosen CLP framework. The applicability of CLP(QS) is demonstrated by way of qualitatively abstracted geometric reasoning problems from the domain of Computer-Aided Architecture Design (CAAD). Since our focus is on real-world problem solving, all exemplified data-sets, i.e., the CAAD models, are sourced and generated from professional design tools, and they conform to industry scales and standards.

Organization. The paper is organized as follows: Section 2 provides an overview of declarative spatial reasoning and application-guided motivations thereof. Section 3 presents CLP(QS): included are the qualitative spatial domain $\mathcal{QS}$ consisting of positional and topological spatial calculi, a polynomial characterization for $\mathcal{QS}$, and its implementation, and (methodological and empirical) evaluation. Section 4 demonstrates the application potential for CLP(QS) for real-world problem solving in the domain of computer-aided architecture design. In Section 5, we discuss the relationship of our work and contributions with respect to existing research. In Section 6 we conclude and provide research perspectives.

2 What is Declarative Spatial Reasoning?

Declarative spatial reasoning, in so far as its broad conception as a paradigm is concerned, is intuitively best understood with respect to the kinds of computational (and by implication, representational) challenges that it aims to address. The kinds of fundamental reasoning tasks that may be identified within the purview of declarative spatial reasoning span a wide spectrum, e.g., including reasoning patterns such as spatial property projection, spatial simulation, spatial planning (e.g., for configuration problems), hypothetical reasoning (e.g., for abductive explanation) with spatial information to name a few. Both within and beyond the range of domains identified in this paper, these are reasoning problems that involve an inherent interaction between space, actions, events, and spatial change in the backdrop of domain-specific knowledge and commonsense knowledge about the world [6]. For this paper, we restrict ourselves to the domain of *space* alone.

2.1 Need for a Declarative Interface

One of the principal reasons behind the success of declarative (logic and constraint logic) programming frameworks has been their ability to provide a range, and combination thereof, of constraint and logic-based reasoning abilities in the context of a high-level first-order language that may be used to directly encode a domain of interest.[2] For instance, consider the following fragment of Prolog code that recursively computes the transitive closure of a relationship $\mathcal{R}$ (e.g., one may imagine this to be a traversal problem):

```
t_closure_R(X, Y) :- R(X, Y).
t_closure_R(X, Y) :- R(X, Z), t_closure_R(Z, Y).
```

Here, the search for the transitive closure `t_closure_R`, and term unification (based on syntactic equality) for the variables involved (i.e., `X, Y, Z`) is built into a logic programming language such as Prolog. The precise semantics of the terms being unified does not acquire any special significance since *equality* is the only relation that is available for term unification. Constraint logic based extensions make it possible to utilize inequalities and the existence of inequality constraints over Integers, Reals etc. For instance, now consider a rather specialized fragment that computes the set of points that are *inside* of a 3D solid object (e.g., a sphere). This could be imagined to be a reasoning task in the context of Constructive Solid Geometry (CSG) [27], with the following fragment requiring a constraint solver for quadratic and linear constraints:

```
entity(ball, sphere(center(1,1,1), radius(5)).
inSphere(point(X, Y, Z), sphere(center(Cx, Cy, Cz), radius(R))) :-
   (X - Cx)*(X - Cx)+(Y - Cy)*(Y - Cy)+(Z - Cz)*(Z - Cz) <= R*R, R>=0.
```

What we are aiming at is the general ability to declaratively refer to high-level statements about real and hypothetical spatial worlds, without the need to specify problems with the level of formalization in the above. Declarative programming frameworks such as CLP are indeed not pre-equipped to deal with spatial reasoning, or general spatial computation capabilities. For instance, a CLP engine cannot understand the semantics of space, and spatial relations such as inside, front-of, or in general, the semantics of relational spatial systems constituted by formal qualitative spatial calculi. By analogy, this is similar to the case where a general logic programming language such as Prolog is not intended to understand the meaning of the predicate 'likes' in the statement 'likes(john, films)', or complex taxonomic structures formalized therefrom without giving an explicit formalization of the descriptive semantics; such specialized ontological reasoning would instead fall within the purview of Description Logic reasoners. Consider the domain of spatial computing for design [8]: automated computer-assisted architecture design (CAAD) systems require the capability to solve structural and functional *design requirement consistency* problems. Such problems are expressible as spatial constraints —topological, orientational, size— among the domain entities (i.e., regions, line-segments and points) that constitute a CAAD model. The following scenario constitutes a design requirement:

[2] This is applicable to other declarative frameworks such as functional programming. However, this paper will specifically deal with logic and constraint-logic based programming approaches.

Security / Privacy. A typical design requirement may entail that certain parts of the environment may or may not be visible or readily accessible. For instance, it may be desired that the WashRoom is as isolated as possible from other work-areas, or that the main entrance area be within the reach of sensing apparatuses such as an in-house Camera.

This constraint may, for instance, be directly encoded at a higher-level of abstraction within a rule-based programming mode[3]; in the following example the *operational* space denotes the region of space that an object requires to perform its intrinsic function, and the *range space* denotes the region of space that lies within the scope of a sensory device [9]:

```
secure_by(Door, Sensor) :-
        physical_geometry(Door, PGeom),
        operational_space(PGeom, OpSpace),
        range_space(Sensor, RgSpace),
        topology(OpSpace, RgSpace, inside).
```

The ability to declaratively handle spatial entities, and the topological and orientation relationships amongst them is only part of the story; in Section 2.2, which is to follow, we present a general class of application requirements.

2.2 Key Application Requirements

Application domains that involve spatial information processing typically require the following fundamental capabilities:

▷ **domain constraints**. express (spatial) constraints between domain entities by way of high-level *rules*, e.g., of the kind typically expressible within a logic programming framework

▷ **consistency**. check for (in)consistency of the *rules*, involving checking for spatial consistency, by considering the special properties that a domain such as *space* merits. For instance, this involves using the specialized spatial representation and reasoning mechanisms developed within the QSR community

▷ **hypothetical reasoning**. perform hypothetical reasoning at the qualitative spatial level, involving reasoning about *what could be*, on the basis of *what is*. Here, the key requirement is to use the special properties of the spatial relationship space and commonsense knowledge about space to derive those spatial configurations that are physically realizable. In a dynamic context, which is not addressed in this paper, this also translates to the task of *scenario and narrative completion*, e.g., by spatio-temporal abduction. In conjunction with *quantification* (see below), this may be used to support a recommendation function in a spatial design context [8].

▷ **quantification**. to compute *quantifications* that provide a metric grounding for the hypothesized spatial scenarios. This capability could, for instance, serve

[3] The scenario is further built-up and illustrated in Fig. 4; Section 4.

as the low-level computational foundation for high-level reasoners capable of abductive reasoning with spatial scenarios, and narratives of spatio-temporal information [7].

In principle, by broadening the interpretation of spatial reasoning, many more specialized spatial reasoning and computing requirements may be identified, e.g., spatio-temporal projection, simulation, planning, abductive explanation, inductive generalization, spatial similarity and matching, spatial data merging and integration [6]. Regardless, it is imperative that such facilities have to be provided in a domain neutral manner.

3 CLP($\mathcal{QS}$): A Declarative Spatial Reasoning Framework

Constraint Logic Programming (CLP) is a form of constraint programming in which logic programming is extended to include concepts from constraint satisfaction [29]. The CLP framework combines methods in *Constraint Programming* (CP) with *Logic Programming* (LP) techniques, thereby providing for a seamless integration of logical methods with algebraic techniques. A constraint logic program is essentially a logic program containing constraints (e.g., over the domain $\mathbb{R}$) in the body of clauses. The difference is that, whereas within a LP problem solving is reduced to syntactic unification and theorem-proving, within CLP the task is to interpret term *unification* as more than *equality* by regarding it as a constraint system. Several CLP solvers exist: CLP($\mathcal{R}$) [30] and Prolog(III) for solving constraints over real numbers, the RISC-CLP(Real) for non-linear real constraints [28], CLP(RL) [46] for first-order formulas over various numeric domains, Abductive CLP [31] and so on.[4]

3.1 The Qualitative Spatial Domain $\mathcal{QS}$

Qualitative spatial calculi can be classified into two groups: topological and positional calculi. With topological calculi such as the Region Connection Calculus (RCC) [40] and the 9-Intersection Model [22], the primitive entities are spatially extended regions of space, and in the case of the mereotopological RCC system, could possibly even be 4D spatio-temporal histories, e.g., for *motion-pattern* analysis. Alternately, within a dynamic domain involving translational motion, point-based abstractions with orientation calculi suffice. Fig 2(a) is a 2D illustration of the RCC-8 relations. Other spatial calculi include the $\mathcal{LR}$ calculus [45], the Oriented-Point Relation Algebra ($\mathcal{OPRA}_m$) [39], the Double-Cross Calculus [24], and the line-segment based *Dipole Calculus* [43].

In this paper, we focus on two-dimensional point-based spatial calculi for representing intrinsically and extrinsically referenced orientation information. Spatial calculi such as the Oriented-Point Relation Algebra ($\mathcal{OPRA}_m$), the $\mathcal{LR}$ calculus, $\mathcal{STAR}$ calculus, Dipole calculus ($\mathcal{DP}$), Double Cross Calculus ($\mathcal{DCC}$) are applicable in this context.

[4] As further described in Section 3.2, a solver in the class of RISC-CLP(Real) is relevant from the viewpoint of this paper.

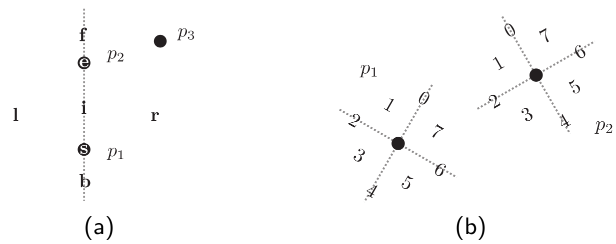


Fig. 1: Illustrations of $\mathcal{LR}$ relation $p_1\ p_2\ \mathbf{r}\ p_3$ and $\mathcal{OPRA}_2$ relation $p_1\ {}_2\angle^2_7\ p_2$

The spatial domain $\mathcal{QS}$ consists of: $\mathcal{LR}$, $\mathcal{OPRA}_m$, $\mathcal{STAR}$, $\mathcal{DP}$, $\mathcal{DCC}$, and topological calculi such RCC and the 9-Intersection model restricted to a specific class of 2D polygonal regions.[5] Our formalization of the qualitative spatial domain $\mathcal{QS}$ relies on the support for non-linear real constraints within CLP [28].

3.2 A CLP Based Polynomial Characterization for $\mathcal{QS}$

The decidability of the problem of solving first-order polynomial constraints over reals (also known as the Quantifier Elimination (QE) problem) was found by Tarski [47]. After the seminal contribution by Collins [15], QE over reals has been extensively investigated, where the main focus was to cope with the intrinsic doubly exponential complexity of the problem [18]. There are several approaches to this problem, the cylindrical algebraic decomposition algorithm by Collins [15, 16] being the most prominent approach ([38] provides a concise overview on this topic).
In what follows we present qualitative spatial relations from $\mathcal{LR}$, $\mathcal{OPRA}_m$ as polynomial constraints over reals, of which the decision problem is in PSPACE [11] in the number of objects, and RCC as first-order constraints over the $\mathcal{LR}$ calculus, of which the worst case complexity for the decision problem is known to be doubly exponential [18]. Thereby, we are not only able to decide the problem, but we are also capable of providing a model of the solution by means of *quantification*.

Throughout the following descriptions all points are two-dimensional points from the Euclidian plane.

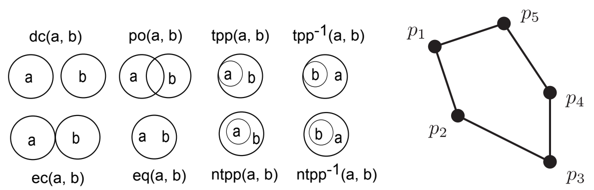
$\mathcal{LR}$ calculus The domain of the $\mathcal{LR}$ calculus [45] is the set of all points in the Euclidian plane. A $\mathcal{LR}$ relation describes for three points $p_1 = (x_1, y_1)$, $p_2 = (x_2, y_2)$, $p_3 = (x_3, y_3)$ the direction of p_3 with respect to p_1, where the orientation of p_1 is determined by p_2. There are altogether nine $\mathcal{LR}$ relations; seven relations for points are depicted in Fig 1(a) : **l**eft, **r**ight, **f**ront, **s**tart, **i**nbetween, **e**nd, **b**ack. In Fig 1(a) the Euclidian plane is partitioned by points p_1 and p_2, $p_1 \neq p_2$ into seven regions:

[5] In a number of application domains (e.g., architecture, construction IT, urban planning), the input data for describing regions can be adequately represented by polygons.

two half-planes (**l**, **r**), two half-lines (**f**, **b**), two points (**s**, **e**), and a line segment (**i**). These regions determine the relation of the third point to p_1 and p_2. The remaining two relations are: **d**ouble $:= \left\{ (p_1, p_2, p_3) \;\middle|\; p_1, p_2, p_3 \in \mathbb{R}^2, p_1 = p_2, p_1 \neq p_3 \right\}$, **t**riple $:= \left\{ (p_1, p_2, p_3) \;\middle|\; p_1, p_2, p_3 \in \mathbb{R}^2, p_1 = p_2 = p_3 \right\}$. By describing the relations using polynomial constraints, we obtain the correspondences (1)–(12), where we introduce a new point p_4 for the equivalences (9), (6) and (3), if there is no point p_4, such that $p_1\ p_2\ r\ p_4$. The polynomial constraints in (1) and (2) come from the determinant of the matrix $\left(\begin{smallmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{smallmatrix}\right)$, whose sign determines the relative position of points p_1, p_2, p_3, where $-, 0, +$ mean clockwise orientation, collinear, counterclockwise orientation, respectively.

$$
\begin{aligned}
p_1\ p_2\ \mathbf{l}\ p_3 \ &\equiv_{\text{def}}\ x_2y_3 + x_1y_2 + x_3y_1 - y_2x_3 - y_1x_2 - y_3x_1 > 0 && (1)\\
p_1\ p_2\ \mathbf{r}\ p_3 \ &\equiv_{\text{def}}\ x_2y_3 + x_1y_2 + x_3y_1 - y_2x_3 - y_1x_2 - y_3x_1 < 0 && (2)\\
p_1\ p_2\ \mathbf{b}\ p_3 \ &\equiv_{\text{def}}\ x_2y_3 + x_1y_2 + x_3y_1 - y_2x_3 - y_1x_2 - y_3x_1 = 0 && (3)\\
&\qquad \wedge\ p_1\ p_2\ \mathbf{r}\ p_4\ \wedge\ p_4\ p_1\ \mathbf{l}\ p_3 && (4)\\
p_1\ p_2\ \mathbf{s}\ p_3 \ &\equiv_{\text{def}}\ x_3 = x_1\ \wedge\ y_3 = y_1\ \wedge\ x_3 \neq x_2\ \wedge\ y_3 \neq y_2 && (5)\\
p_1\ p_2\ \mathbf{i}\ p_3 \ &\equiv_{\text{def}}\ x_2y_3 + x_1y_2 + x_3y_1 - y_2x_3 - y_1x_2 - y_3x_1 = 0 && (6)\\
&\qquad \wedge\ p_1\ p_2\ \mathbf{r}\ p_4\ \wedge\ p_4\ p_1\ \mathbf{r}\ p_3\ \wedge\ p_4\ p_2\ \mathbf{l}\ p_3 && (7)\\
p_1\ p_2\ \mathbf{e}\ p_3 \ &\equiv_{\text{def}}\ x_3 = x_2\ \wedge\ y_3 = y_2\ \wedge\ x_3 \neq x_1\ \wedge\ y_3 \neq y_1 && (8)\\
p_1\ p_2\ \mathbf{f}\ p_3 \ &\equiv_{\text{def}}\ x_2y_3 + x_1y_2 + x_3y_1 - y_2x_3 - y_1x_2 - y_3x_1 = 0 && (9)\\
&\qquad \wedge\ p_1\ p_2\ \mathbf{r}\ p_4\ \wedge\ p_4\ p_2\ \mathbf{r}\ p_3 && (10)\\
p_1\ p_2\ \mathbf{d}\ p_3 \ &\equiv_{\text{def}}\ x_1 = x_2\ \wedge\ y_1 = y_2\ \wedge\ x_1 \neq x_3\ \wedge\ y_1 \neq y_3 && (11)\\
p_1\ p_2\ \mathbf{t}\ p_3 \ &\equiv_{\text{def}}\ x_1 = x_2 = x_3\ \wedge\ y_1 = y_2 = y_3, && (12)
\end{aligned}
$$

$\mathcal{OPRA}_m$ calculus The domain of the $\mathcal{OPRA}_m$ calculus is the set of all oriented points. An oriented point p is a quadruple (x, y, v, w), $x, y, v, w \in \mathbb{R}$, where (x, y) is the location of p, and (v, w) defines the orientation of p by means of the orientation vector $\boldsymbol{o}_p := (v, w) - (x, y)$. Two orientated points p_1 and p_2 are equal if their positions and orientations are equal. With m lines passing through p, we can partition the whole plane (without the point itself) equally into $2m$ open sectors and $2m$ half-lines, where exactly one distinguished half-line has the same orientation as $\boldsymbol{o}_p$. Starting with the distinguished half-line, and going through the sectors and half-lines alternately in the counterclockwise order, we can assign numbers 0 to $4m - 1$ to the open sectors and half-lines (See Fig 1(b)). An $\mathcal{OPRA}_m$ relation is a binary relation which describes for points p_1 and p_2 their positions relative to each other with respect to the aforementioned partitioning. This is represented by $p_1\ {}_m\angle_i^j\ p_2$, where m is as defined before, i is the number of the sector (or half-line) of p_1, in which p_2 is located, and j is



(a) RCC-8 Relations

(b) A convex, polygonal region $(p_1, p_2, p_3, p_4, p_5)$

Fig. 2: Topological Relations

the number of the sector (or half-line) of p_2, in which p_1 is located.[6] Then for $p_1 = (x_1, y_1, v_1, w_1)$, $p_2 = (x_2, y_2, v_2, w_2)$, and the rotation map

$$\begin{pmatrix} r_x(v, w, \theta) \\ r_y(v, w, \theta) \end{pmatrix} := \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} v \\ w \end{pmatrix}$$

we can define for $i = 0, 2, \ldots, m-4, m-2$:

$$\begin{aligned} p_1 \;{}_m\angle_i^* \; p_2 \quad \equiv_{\text{def}} \quad & \det \begin{pmatrix} 1 & x_1 & y_1 \\ 1 & r_x\left(v_1, w_1, i\frac{\pi}{m}\right) & r_y\left(v_1, w_1, i\frac{\pi}{m}\right) \\ 1 & x_2 & y_2 \end{pmatrix} = 0 \\ \wedge \quad & \det \begin{pmatrix} 1 & x_1 & y_1 \\ 1 & r_x\left(v_1, w_1, (i+\frac{m}{2})\frac{\pi}{m}\right) & r_y\left(v_1, w_1, (i+\frac{m}{2})\frac{\pi}{m}\right) \\ 1 & x_2 & y_2 \end{pmatrix} < 0, \end{aligned}$$

which describe that p_2 is in half-line i of p_1, and for $i = 1, 3, \ldots, m-3, m-1$:

$$\begin{aligned} p_1 \;{}_m\angle_i^* \; p_2 \quad \equiv_{\text{def}} \quad & \det \begin{pmatrix} 1 & x_1 & y_1 \\ 1 & r_x\left(v_1, w_1, (i-1)\frac{\pi}{m}\right) & r_y\left(v_1, w_1, (i-1)\frac{\pi}{m}\right) \\ 1 & x_2 & y_2 \end{pmatrix} > 0 \\ \wedge \quad & \det \begin{pmatrix} 1 & x_1 & y_1 \\ 1 & r_x\left(v_1, w_1, (i+1)\frac{\pi}{m}\right) & r_y\left(v_1, w_1, (i+1)\frac{\pi}{m}\right) \\ 1 & x_2 & y_2 \end{pmatrix} < 0, \end{aligned}$$

which describe that p_2 is in sector i of p_1. Then

$$p_1 \;{}_m\angle_i^j \; p_2 \quad \equiv \quad p_1 \;{}_m\angle_i^* \; p_2 \quad \wedge \quad p_2 \;{}_m\angle_j^* \; p_1,$$

and we obtain the desired polynomial constraints.

Other point-based calculi Other point-based calculi like the $\mathcal{STAR}$ calculus, dipole calculus, single-cross, and double-cross calculus can be modeled in a similar way as described previously. Indeed, all of the point-based calculi mentioned in this paper except for $\mathcal{STAR}$ can be reduced to $\mathcal{OPRA}_m$ [21], which has been formally characterized in this paper.

---

[6] The original paper [39] also introduces the so-called *same relations* for two coinciding oriented points, which are differentiated by their orientations. Since these cases are primarily relevant for relation-algebraic reasoning, they are omitted in this paper.

$\mathfrak{p} \in R^\circ$	$\forall i\,(p_i\ p_{i+1}\ \mathbf{l}\ \mathfrak{p})$
$\mathfrak{p} \in \partial R$	$\exists i\,(p_i\ p_{i+1}\ \mathbf{s}\ \mathfrak{p}\ \vee\ p_i\ p_{i+1}\ \mathbf{i}\ \mathfrak{p}\ \vee\ p_i\ p_{i+1}\ \mathbf{e}\ \mathfrak{p})$
$\mathfrak{p} \in R^C$	$\exists i : p_i\ p_{i+1}\ \mathbf{r}\ \mathfrak{p}$
$\mathfrak{p} \in R$	$\mathfrak{p} \in R^\circ$ or $\mathfrak{p} \in \partial R$
$\mathfrak{p} \in R \cap S$	$\mathfrak{p} \in R$ and $\mathfrak{p} \in S$
$\mathfrak{p} \in R \cup S$	$\mathfrak{p} \in R$ or $\mathfrak{p} \in S$
$\mathfrak{p} \in R \backslash S$	$\mathfrak{p} \in R \cap S^C$

Table 1: The correspondence table

R dc S	$\forall \mathfrak{p} \left(\mathfrak{p} \in S \Rightarrow \mathfrak{p} \in R^C\right)$
R ec S	$\exists \mathfrak{p}\,(\mathfrak{p} \in \partial R \cap \partial S) \wedge \forall \mathfrak{p} \left(\mathfrak{p} \in R \Rightarrow \mathfrak{p} \in \partial S \cup S^C\right)$
R po S	$\exists \mathfrak{p}, \mathfrak{p}', \mathfrak{p}''\,(\mathfrak{p} \in R \cap S \wedge \mathfrak{p}' \in R \backslash S \wedge \mathfrak{p}'' \in S \backslash R)$
R tpp S	$\exists \mathfrak{p}\,(\mathfrak{p} \in \partial R \cap \partial S) \wedge \forall \mathfrak{p}\,(\mathfrak{p} \in R \Rightarrow \mathfrak{p} \in S)$
R tppi S	$\exists \mathfrak{p}\,(\mathfrak{p} \in \partial R \cap \partial S) \wedge \forall \mathfrak{p}\,(\mathfrak{p} \in S \Rightarrow \mathfrak{p} \in R)$
R ntpp S	$\forall \mathfrak{p}\,(\mathfrak{p} \in R \Rightarrow \mathfrak{p} \in S^\circ)$
R ntppi S	$\forall \mathfrak{p}\,(\mathfrak{p} \in S \Rightarrow \mathfrak{p} \in R^\circ)$
R eq S	$\forall \mathfrak{p}\,(\mathfrak{p} \in R \Leftrightarrow \mathfrak{p} \in S)$

Table 2: The correspondence table for complex regions R and S

Topological calculi We model topological relations (Fig. 2(a)) [22, 40] between regions of a restricted class, namely, polygons that can consist of disconnect pieces and can contain holes.

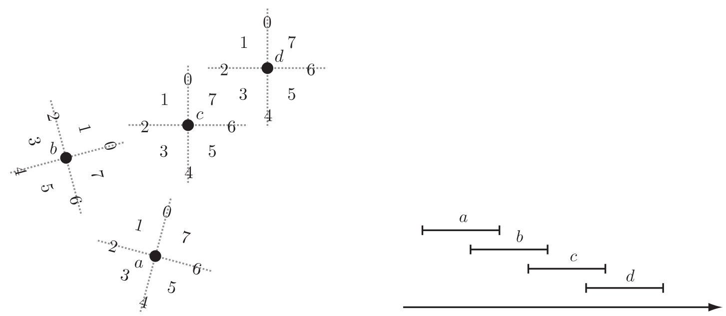
Firstly, we define a region R as a simple polygon given by a sequence of 2D points, i.e., $R := (p_1, p_2, \ldots, p_m)$, $m \geq 3$. Region R is convex, i.e., $p_i\ p_{i+1}\ \mathbf{l}\ p_{i+2}$, for all $i = 1, 2, \ldots, m-1$, where we set $p_{m+1} := p_1$ (See Fig 2(b)). With this setting we can decide the relative position of a point q with respect to a convex region R (i.e., whether the point is inside, in the interior, in the boundary, or outside of R) by using the constraints from the $\mathcal{LR}$ calculus. Table 1 shows the correspondences.

As any simple polygon can be partitioned into convex polygons in polynomial-time [12], and is therefore a disjunction of convex polygons, our definitions are naturally extended to concave polygons.

On the basis of this information we can further decide the RCC relations between regions R and S. This is shown in Table 2. Our encoding of the semantics of topological relations, and the restricted class of regions that we admit, suffices for a characterization of topological relations for both the 9-Intersection Model [22] as well as the mereotopological Region Connection Calculus [40].

3.3 Relational Algebraic and Polynomial Characterizations

The algebraic-closure method, which utilizes the relational algebraic structure of qualitative calculi, provides a sound and complete algorithm for calculi like Allen's Interval Algebra or RCC8. However, the method is not sound for point-based 2D calculi, including $\mathcal{LR}$ or $\mathcal{OPRA}_m$ as shown in [25, 52], on which our applications



(a) Algebraic closure cannot detect the $\mathcal{OPRA}_2$ inconsistency that b,c,d must be colinear, a,c,d must be colinear, but a,b,c,d must not be colinear.

(b) To be consistent with the $\mathcal{CA}$ constraints, d must also overlap a while being disjoint from b. Algebraic closure cannot detect that this is impossible on a 1D acyclic domain of intervals.

Fig. 3

are based. By contrast, the quantifier elimination problem as addressed at the beginning of Section 3.2 is decidable and has effective decision methods like the cylindrical decomposition algorithm [15]. In what follows, we address the weaknesses of the relation algebraic approach. These weaknesses call for the quantifier elimination approach underlying our formulation of $\mathcal{QS}$.

▷ **Failing on atomic networks**. For calculi that are not *closed under constraints* [41], algebraic closure is not able to determine the consistency of *atomic* networks. For example, using $\mathcal{OPRA}_m$ [25] let a,b be distinct oriented points directly facing points c,d. We make this inconsistent by placing a,b to the left-rear of c, and d to the front-right of c (see Figure 3(a)),[7]

```
opra₂(a,b,1,7),        opra₂(c,d,7,3),
opra₂(a,c,0,3),        opra₂(a,d,0,3),
opra₂(b,c,0,3),        opra₂(b,d,0,3).
```

Algebraic closure cannot detect this inconsistency. Other examples include $\mathcal{LR}$ [52] and $\mathcal{INDU}$ [4].

▷ **Failing on interpretations**. Changing the domain of interpretation in a seemingly trivial manner can dramatically change the effectiveness of algebraic closure. Moreover, an interpretation that is suited to one calculus can be precisely the interpretation that causes algebraic closure to fail in another calculus, making it tricky to freely mix and swap calculi. An example is the containment algebra ($\mathcal{CA}$) [35].[8] Algebraic closure cannot infer that a $\mathcal{CA}$ cycle of at least four intervals requires at least two spatial dimensions or a cyclic domain (see Figure 3(b)),

---

[7] The predicate `opra`$_m$`(p`$_1$`,p`$_2$`,i,j)` represents the $\mathcal{OPRA}_m$ relation $p_1 \,{}_m\angle_i^j\, p_2$.

[8] In $\mathcal{CA}$, algebraic closure cannot decide consistency when interpreted in the domain of linearly ordered intervals [35]. For example, this could be 1D axis-aligned blocks motivated by the block-algebra for $p = 1$, or spatial intervals along an acyclic path motivated by Allen's interval algebra. The trap for developers is that it *can* decide

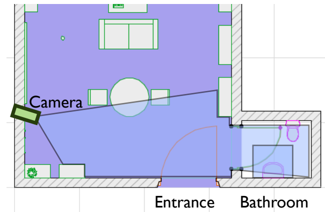


Fig. 4: Work-in-progress floorplan of an office (range space is illustration only)

```
partialOverlap(a,b),   partialOverlap(b,c),
partialOverlap(c,d),   partialOverlap(d,a),
disjoint(a,c),         disjoint(b,d).
```

3.4 CLP(QS): Implementation Overview

In order to test the basic principles of CLP(QS), we have implemented a prototype through a loose integration (in C++) between SWI-Prolog [51] for high-level reasoning and REDUCE [26] for solving polynomial equations. The REDLOG package [19] of the computer algebra system REDUCE allows quantifier elimination over Reals. Prolog manages the control of query-answering by building REDLOG expressions for $\mathcal{LR}$ relations as described in Section 3.2 using specialized predicates such as `i_LR(P1,P2,P3)`.

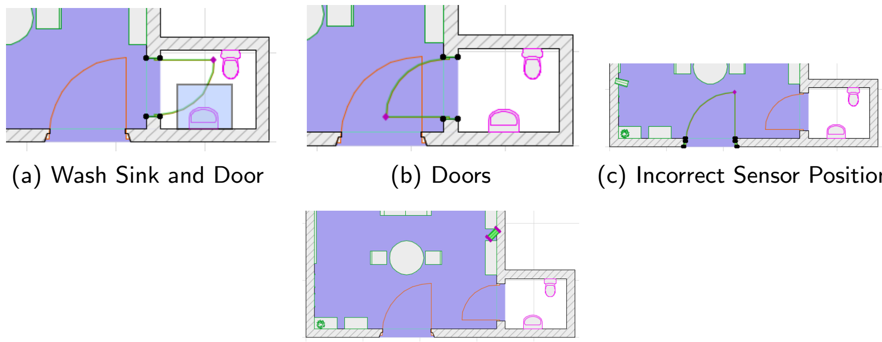
Although optimization and benchmarking are not the aims of this paper, it must be noted that the complexity of general polynomial systems is doubly-exponential [18].[9] As future outlook of this aspect of the work, we are interested in applying constraint optimizations, investigating more computationally efficient relation encodings, and developing a dedicated solver for $\mathcal{QS}$ consistency and quantification problems (Section 6).

4 CLP(QS): An Application in Architecture Design

In this section we describe the role of CLP(QS) in the domain of Computer Aided Architectural Design (CAAD). During the process of designing a building, architects routinely analyze an enormous amount of detailed information in order to determine whether certain requirements have been met. These can range from meeting strictly objective safety codes, to eliciting subjective, emotional responses, and typically employ geometric and high-level spatial features. Imagine that an architect is designing the floorplan of an office. Figure 4 illustrates a work-in-progress design. We will now demonstrate how CLP(QS) can be used for checking the consistency of high-level constraints and for quantification.

consistency of atomic networks for less structured domains (i.e. RCC-5 [42]) such as closed disks [20] and the block-algebra for $p > 1$.

[9] Most poor run-times have been experienced with our sample data for topological queries where a number of polygon vertices are ungrounded.



(a) Wash Sink and Door (b) Doors (c) Incorrect Sensor Position

(d) Sensor Repositioning

Fig. 5: Design (In)Consistency

4.1 Consistency

The structural form corresponding to high-level design constraints (e.g., coming from design guidelines, designer expertise, client requirements) may be conceived and directly translated into a high-level declarative specification. Some examples follow:

▷ *Safety.* Doors should not open onto areas where a person might be located while occupied with some activity. Formally, the operational space of doors should not overlap with the functional space of activity objects such as the washbasin,

```
safety(Door, Object) :-
        operational_space(Door, Op),
        functional_space(Object, Fs),
        not(topology(Op, Fs, overlaps)).
```

CLP(QS) detects that the current design does not meet this constraint for the washbasin (see Figure 5(a)). The architect changes the opening direction of the door which causes a new problem as detected by CLP(QS) (see Figure 5(b)),

```
noCollidingDoors(Door1, Door2) :-
        operational_space(Door1, Op1),
        operational_space(Door2, Op2),
        not(topology(Op1, Op2, overlaps)).
```

The architect resolves the problem by sliding the door down the wall (see Figure 5(c)).

▷ *Security.* The architect must position a camera so that people entering the office can be identified. This is formalized using the range space of the camera and the operational space of the entrance door,

```
secure_by(Door, Camera) :-
        physical_geometry(Door, G),
        operational_space(G, O),
        range_space(Camera, R),
        topology(O, R, inside).
```

CLP(QS) confirms that the design currently satisfies this constraint by translating it into the equivalent $\mathcal{LR}$ constraint [10]

$$\texttt{ntpp(O,R)} \equiv \forall \texttt{p} \cdot \texttt{inside(p,O)} \rightarrow \texttt{interior(p,R)}$$
$$\equiv \forall \texttt{p} \; (\; \neg\exists \; \texttt{i} \cdot \texttt{r}(\texttt{p}_i^O, \texttt{p}_{i+1}^O, \texttt{p}) \rightarrow \forall \texttt{j} \cdot \texttt{l}(\texttt{p}_j^R, \texttt{p}_{j+1}^R, \texttt{p}) \;) \; .$$

▷ *Privacy.* Security cameras must not be able to record inside the adjoining bathroom. That is, cameras must be directed away from the bathroom,

```
privacy(Bathroom, Camera) :-
        oriented_point(Camera, C),
        physical_geometry(Bathroom, G),
        not orientation(C, G, facing).
```

Facing can be interpreted as the constraint:

$$\texttt{facing(C,G)} \equiv \exists \texttt{p} \; (\; \texttt{inside(p,G)}, \; \exists \texttt{i} \in \{15, 0, 1\} \; \texttt{opra}_4\texttt{(C,p,i,*)}).$$

On the *privacy* constraint, CLP(QS) detects that the current design is inadequate. The architect attempts to solve the problem by rotating the camera to face away from the bathroom, however, CLP(QS) detects that this causes the *security* constraint to fail.

4.2 Quantification

We will now get CLP(QS) to suggest a location and orientation of the camera that satisfies the *security* and *privacy* constraints. To do this we will specify the location and orientation of the camera as an ungrounded variable, and further constrain its domain of acceptable configurations.

▷ *Position.* The camera must be mounted on one of the perimeter walls of the office,

```
onPerimeter(Camera, Office) :-
         ∃W ( isWall(w), in(W, Office),
        physical_geometry(W, R),
        oriented_point(Camera, P),
        topology(P, R, boundary)).
```

[10] For clarity we have assumed in the example that the regions are convex and do not have holes. As we described in Section 3.2, regions with holes can also be handled.

This is equivalent to the $\mathcal{LR}$ constraint

```
∃j ( s(w_1^j,w_2^j,p) or i(w_1^j,w_2^j,p) or e(w_1^j,w_2^j,p)) .
```

where a wall W_j is the line segment (w_1^j, w_2^j).
▷ *Visibility.* A region R is visible from an observation point p (in a room with objects $Objs$) if there is at least one unobstructed line between the observer and some point within the region,

```
visible(R,p,Objs) :-
      ∃p2 ( ∀S ∈ Objs/{R} (
      inside(p2,R),
      not( lineIntersect(p,p2,S)) )).
```

If region R is a sequence of points, then the predicate for determining whether a line intersects a region is

```
lineIntersect(p1,p2,R) :-
   ∃i (p_i,p_{i+1} ∈ R, lineIntersect(p1,p2,p_i,p_{i+1})).
```

where line intersection is defined using $\mathcal{LR}$ relations,

```
lineIntersect(a1,a2,b1,b2) :-
   ∃p ( (i(a1,a2,p) or s(a1,a2,p) or f(a1,a2,p) )
    and (i(b1,b2,p) or s(b1,b2,p) or f(b1,b2,p) )).
```

CLP(QS) suggests the location and orientation illustrated in Figure 5(d).

5 Discussion and Related Work

Researchers have investigated high-level modeling and reasoning with spatial knowledge; most direct connections of our work emerge with the works by Almendros-Jiménez [3], Banerjee and Chandrasekaran [5], Bhatt et al. [9], Escrig and Toledo [23], Kurup and Cassimatis [34], Uribe et al. [49]. Related work also exists in the constraint databases community, although there exist fundamental differences between constraint query languages and CLP [13, 32]. Focussing on the knowledge representation approach, we differentiate our approach and contributions with respect to three main aspects:

1. Use of qualitative spatial calculi, as construed within QSR, whilst preserving their relational semantics (and therefore, where applicable, their cognitive and psycholinguistic underpinnings). This is crucial in domains such as design where constraints are tightly connected to their high-level conceptualization.
2. Providing an underlying polynomial characterization for a range of topological and positional spatial calculi, thereby:
   (a) overcoming several limitations, in a relational algebraic sense, of conventional compositional reasoning with precomputed composition tables

   (b) providing inherent support for quantification of relational spatial information
3. Enabling spatial entities and qualitative relations as first-class, native objects within the declarative framework of classical constraint logic programming systems, and providing a prototypical implementation and demonstration of its application for real-world problem solving.

The declarative approach adopted by Bhatt et al. [9] uses a description logic based reasoning technique for ensuring spatio-terminological consistency. In this work, only topological consistency may be checked for using a terminological system, and it is not possible to solve constraints, quantify, or specify positional constraints. The work by Uribe et al. [49] aims to construct a module that can provide spatial reasoning services in the context of a larger system that is not necessarily restricted to spatial knowledge. Their view of spatial knowledge is grounded in QSR, and: (a) aims at providing question-answering support within a first-order theorem prover, (b) solely relies on spatial reasoning by incorporating composition tables within (the prover). This is different from our approach in two major respects: (1) we utilize constraint logic programming, where problem-solving is based on constraint solving foundations, as opposed to a theorem-proving approach; and (2) our polynomial characterization of the spatial domain $\mathcal{QS}$ does not utilize compositional reasoning thereby overcoming the limitations of reasoning with composition tables (as elaborated on in Section 3.3). The work by Escrig and Toledo [23] may also be situated in this category, since here too spatial reasoning (with positional calculi) is performed using composition tables. It is worth noting here that spatial representation is performed using encoding composition tables using constraint handling rules[11] framework. Almendros-Jiménez [3] approach the problem of constraint solving over sets of spatial objects by considering spatial constraints as an instance of CLP. Notwithstanding the fact that Almendros-Jiménez neither addresses QSR methods, nor utilizes any form of linear or non-linear formalization, there exist interesting potentials when considering the synergy afforded by the contributions of [3] and this paper: their in-depth study on the operational semantics of the CLP solver, i.e., the interaction of the spatial constraint solver with the underlying resolution mechanism in CLP, whilst considering constraints over sets of spatial objects, provides useful insights for a future extensions of our prototypical CLP(QS) framework in a manner such that it may be tightly integrated within state-of-the-art CLP engines (e.g., as a specialized spatial reasoning library for ECLIPSE).
Banerjee and Chandrasekaran [5] deal with a range of spatial perception and action problems from a diagrammatic reasoning perspective. Whereas they employ similar underlying methods (namely, quantified constraint satisfaction) for solving spatial problems, their approach does not seek a direct integration with CLP or other declarative programming approaches. Furthermore, the equivalent of

[11] Constraint Handling Rules (CHR) are a special purpose language designed to write and combine constraint systems [44]. CHR have been used to encode a range of constraint handlers (solvers), including domains such as terminological and temporal reasoning.

the spatial domain $\mathcal{QS}$ in their work consists of a diagrammatic representation, where in our case, $\mathcal{QS}$ is founded on qualitative spatial calculi in QSR. As emphasised in this paper, our perspectives on the spatial domain $\mathcal{QS}$ within declarative spatial reasoning are subject to reinterpretations and extensions: from our perspective, we see interesting synergies and possibilities to compare different formalizations for $\mathcal{QS}$ possibly encompassing visual and diagrammatic models of space. Methodologically similar to the work of Banerjee and Chandrasekaran [5] in its use of a diagrammatic representations is the work by Kurup and Cassimatis [34]. Within a propositional logic framework, Kurup and Cassimatis [34] approach the spatial reasoning problem by integrating a diagrammatic representation with a DPLL-based backtracking algorithm that is specialized for spatial relations of objects in a *grid*. This approach is efficient compared to other approaches using SAT solvers or SMT solvers. However, it will not find appropriate solutions in the real world, as spatial reasoning in a grid leads to information loss. Similar to the work by Uribe et al. [49], the approach of Kurup and Cassimatis [34] is also not grounded to the formal semantics of qualitative spatial calculi, but instead, utilizes a diagrammatic representation. The principal motivation of their approach is to overcome the limitations of (diagrammatic) spatial reasoning with propositional satisfiability solvers. In doing so, they show the manner in which the DPLL algorithm augmented with diagrammatic reasoning can be used to make SAT more efficient when reasoning about spatial relations in a grid. Indeed, none of the related works discussed so far consider *quantification* of relational spatial information, with only the approach of Banerjee and Chandrasekaran [5] being quantification capable. Quantification, in our case, has been identified, implemented, and demonstrated to be a crucial computational requirement within applications.

6 Conclusion and Outlook

We have put forward a case for the accessibility of specialized spatial representation and reasoning mechanisms via the medium of high-level, logic-based formalizations in KR. This entails basic scientific challenges, and is motivated by need to solve specialized real-world problems in spatial representation and reasoning, and applying QSR in such application scenarios.

The core contributions of this paper lie in the integration of qualitative spatial representation and reasoning with constraint logic programming. Given the support for polynomial systems within constraint logic programming, our method is directly realizable within state-of-art CLP solvers with support for polynomials. We have demonstrated a prototypical implementation of this approach, and also illustrated its applicability toward practical spatial computing in the domain of computer-aided architecture design.

From a theoretical perspective, the general motivating principle of our ongoing research is that any form of high-level spatial reasoning (e.g., spatial projection, explanation etc) will rely on some form of spatial constraint solving capability, in addition to other forms of non-classical inference patterns such as non-monotonic inference, spatial belief revision capabilities etc. Hence, integration, along the lines of CLP(QS), with other declarative frameworks such as Answer-Set Programming, Event Calculus is one line of work presenting interesting challenges. The general motivating principle here is that any form of high-level spatial reasoning (e.g., spatial projection, explanation etc) will rely on some form of spatial constraints solving capability, in addition to other forms of non-classical inference patterns such as non-monotonic inference, spatial belief revision capabilities etc.

From an application perspective, we have so far considered a rather limited range of problems solely within the context of architecture design. Given the generality of CLP(QS), there exist several possibilities for studies with other application domains; here, geographic information systems, and cognitive robotics are a prime candidates in our ongoing projects. With this backdrop, we are also extending CLP(QS) in order to include support for reasoning with a high-level, qualitative model for 3D visibility [48].

Acknowledgements. This work has benefitted from past collaborations with Gregory Flanagan, Christian Freksa, Frank Dylla, Jan Oliver Wallgün, and Diedrich Wolter. We gratefully acknowledge the funding and support of the German Research Foundation (DFG), `www.dfg.de/`.

References

[1] H. Abelson, G. J. Sussman, and J. Sussman. *Structure and Interpretation of Computer Programs*. MIT Press, Cambridge, Massachusetts, 1985.

[2] M. Aiello, I. Pratt-Hartmann, and J. van Benthem, editors. *Handbook of Spatial Logics*. Springer, 2007.

[3] J. M. Almendros-Jiménez. Constraint logic programming over sets of spatial objects. In *Proceedings of the 2005 ACM SIGPLAN workshop on Curry and functional logic programming*. ACM, 2005.

[4] P. Balbiani, J.-F. Condotta, and G. Ligozat. On the consistency problem for the indu calculus. *Journal of Applied Logic*, 2006.

[5] B. Banerjee and B. Chandrasekaran. A Constraint Satisfaction Framework for Executing Perceptions and Actions in Diagrammatic Reasoning. *Journal of Artificial Intelligence Research*, 2010.

[6] M. Bhatt. Reasoning about space, actions and change: A paradigm for applications of spatial reasoning. In *Qualitative Spatial Representation and Reasoning: Trends and Future Directions*. IGI Global, USA, 2010.

[7] M. Bhatt and G. Flanagan. Spatio-temporal abduction for scenario and narrative completion. In M. Bhatt, H. Guesgen, and S. Hazarika, editors, *Proceedings of the International Workshop on Spatio-Temporal Dynamics, co-located with the European Conference on Artificial Intelligence (ECAI-10)*. ECAI Workshop Proceedings., August 2010.

[8] M. Bhatt and C. Freksa. Spatial computing for design: An artificial intelligence perspective. In *Visual and Spatial Reasoning for Design Creativity (SDC'10)*, 2011.

[9] M. Bhatt, F. Dylla, and J. Hois. Spatio-terminological inference for the design of ambient environments. In *COSIT*, 2009.

[10] M. Bhatt, H. Guesgen, S. Woelfl, and S. Hazarika. Qualitative Spatial and Temporal Reasoning: Emerging Applications, Trends and Directions. *Journal of Spatial Cognition and Computation*, 2011.

[11] J. Canny. Some algebraic and geometric computations in pspace. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*. ACM, 1988.

[12] B. Chazelle and D. Dobkin. Optimal convex decompositions. In *Computational Geometry*. North-Holland, 1985.

[13] J. Chomicki and P. Z. Revesz. Constraint-based interoperability of spatiotemporal databases*. *GeoInformatica*, 1999-09-01.

[14] A. G. Cohn and J. Renz. Qualitative spatial reasoning. In *Handbook of Knowledge Representation*. Elsevier, 2007.

[15] G. Collins. Quantifier elimination for real closed fields by cylindrical algebraic decompostion. In *Automata Theory and Formal Languages 2nd GI Conference Kaiserslautern, May 20–23, 1975*. Springer Berlin / Heidelberg, 1975.

[16] G. E. Collins and H. Hong. Partial cylindrical algebraic decomposition for quantifier elimination. *Journal of Symbolic Computation*, 1991.

[17] A. Colmerauer and P. Roussel. The birth of prolog. In *History of programming languages—II*. ACM, 1996.

[18] J. H. Davenport and J. Heintz. Real quantifier elimination is doubly exponential. *Journal of Symbolic Computation*, 1988.

[19] A. Dolzmann, A. Seidl, and T. Sturm. *Redlog User Manual*, 3.1 edition, November 2006.

[20] I. Duntsch, H. Wang, and S. Mccloskey. Relation algebras in qualitative spatial reasoning. *Fundamenta Informaticae*, 1999.

[21] F. Dylla and J. Wallgrün. On generalizing orientation information in $\mathcal{OPRA}_m$. In *KI 2006: Advances in Artificial Intelligence*. Springer Berlin / Heidelberg, 2007.

[22] M. J. Egenhofer and R. D. Franzosa. Point set topological relations. *International Journal of Geographical Information Systems*, 1991.

[23] M. T. Escrig and F. Toledo. Qualitative spatial orientation with constraint handling rules. In *ECAI*, 1996.

[24] C. Freksa. Using orientation information for qualitative spatial reasoning. In *Proceedings of the Intl. Conf. GIS, From Space to Territory: Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*. Springer-Verlag, 1992.

[25] L. Frommberger, J. H. Lee, J. O. Wallgrün, and F. Dylla. Composition in $\mathcal{OPRA}_m$. Technical report, SFB/TR 8 Spatial Cognition, 2007.

[26] A. C. Hearn. *REDUCE User's Manual*. Santa Monica, CA, USA, 3.8 edition, February 2004.

[27] C. M. Hoffmann. *Geometric and solid modeling: an introduction*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1989.

[28] H. Hong. *RISC-CLP (Real): logic programming with non-linear constraints over the reals*. MIT Press, 1993.

[29] J. Jaffar and M. J. Maher. Constraint logic programming: A survey. *J. Log. Program.*, 1994.

[30] J. Jaffar, S. Michaylov, P. J. Stuckey, and R. H. C. Yap. The clp(r) language and system. *ACM Trans. Program. Lang. Syst.*, 1992.

[31] A. C. Kakas, A. Michael, and C. Mourlas. ACLP: Abductive constraint logic programming. *J. Log. Program.*, 2000.

[32] P. C. Kanellakis, G. M. Kuper, and P. Z. Revesz. Constraint query languages. In *Proceedings of the ninth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*. ACM, 1990. ISBN 0-89791-352-3.

[33] R. A. Kowalski. The early years of logic programming. *Commun. ACM*, 1988.

[34] U. Kurup and N. Cassimatis. Integrating constraint satisfaction and spatial reasoning. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, 2010.

[35] P. B. Ladkin and R. D. Maddux. On binary constraint networks. Technical report, Kestrel Institute, 1988.

[36] V. Lifschitz. What is answer set programming? In *Proceedings of the 23rd national conference on Artificial intelligence - Volume 3*. AAAI Press, 2008.

[37] J. W. Lloyd. Practical advtanages of declarative programming. In M. Alpuente, R. Barbuti, and I. Ramos, editors, *GULP-PRODE (1)*, 1994.

[38] B. Mishra. Computational real algebraic geometry. In *Handbook of discrete and computational geometry*. CRC Press, Inc., 1997.

[39] R. Moratz. Representing relative direction as a binary relation of oriented points. In *Proceeding of the 2006 conference on ECAI 2006: 17th European Conference on Artificial Intelligence August 29 – September 1, 2006, Riva del Garda, Italy*. IOS Press, 2006.

[40] D. A. Randell, Z. Cui, and A. Cohn. A spatial logic based on regions and connection. In *KR'92. Principles of Knowledge Representation and Reasoning*. Morgan Kaufmann, 1992.

[41] J. Renz and G. Ligozat. Weak composition for qualitative spatial and temporal reasoning. In P. van Beek, editor, *CP*, volume 3709 of *Lecture Notes in Computer Science*. Springer, 2005.

[42] J. Renz and B. Nebel. On the complexity of qualitative spatial reasoning: A maximal tractable fragment of the region connection calculus. In *IJCAI (1)*, 1997.

[43] C. Schlieder. Reasoning about ordering. In *Proc. of COSIT'95*. Springer, 1995.

[44] T. Schrijvers and T. W. Frühwirth, editors. *Constraint Handling Rules, Current Research Topics*, volume 5388 of *Lecture Notes in Computer Science*. Springer, 2008.

[45] A. Scivos and B. Nebel. The finest of its class: The natural point-based ternary calculus for qualitative spatial reasoning. In *Spatial Cognition IV. Reasoning, Action, and Interaction*. Springer Berlin / Heidelberg, 2005.

[46] T. Sturm. Quantifier elimination for constraint logic programming. In *Computer Algebra in Scientific Computing*. Springer Berlin / Heidelberg, 2005.

[47] A. Tarski. A decision method for elementary algebra and geometry. Technical report, Santa Monica, CA: RAND Corporation, 1951.

[48] S. Tassoni, P. Foliaroni, M. Bhatt, and G. D. Felice. Toward a Qualitative Model of 3D Visibility. In *25th International Workshop on Qualitative Reasoning (IJCAI 2011)*, 2011. (position paper).

[49] T. E. Uribe, V. Chaudhri, P. J. Hayes, and M. E. Stickel. Qualitative spatial reasoning for question-answering: Axiom reuse and algebraic methods. In *AAAI Spring Symposium on Mining Answers from Texts and Knowledge Bases*, 2002.

[50] F. van Harmelen, V. Lifschitz, and B. Porter, editors. *Handbook of Knowledge Representation (Foundations of Artificial Intelligence)*. Elsevier Science, 2007.

[51] J. Wielemaker, T. Schrijvers, M. Triska, and T. Lager. Swi-prolog. *CoRR*, 2010.

[52] D. Wolter and J. H. Lee. Qualitative reasoning with directional relations. *Artificial Intelligence*, 2010.