# Constructive Geometric Constraint Solving as a General Framework forKR-Based Declarative Spatial Reasoning 

Carl Schultz, ${ }^{1,2}$ Mehul Bhatt ${ }^{1,3}$<br>${ }^{1}$ The Design Space Group, Germany<br>${ }^{2}$ University of Münster, Germany<br>${ }^{3}$ University of Bremen, Germany


#### Abstract

We present a robust and scalable KR-centered foundation for modularly supporting general declarative spatial representation and reasoning within diverse declarative programming AI frameworks. Based on Constructive Geometric Constraint Solving, our approach provides the foundations for mixed qualitative-quantitative reasoning about space -mereotopology, relative orientation, size, proximity - encompassing key applicationdriven capabilities such as qualification, spatial consistency solving, quantification, and dynamic geometry. The paper also demonstrates: (a) the framework with benchmark problems (e.g., contact and orientation problems) and applications in spatial Q/A; (b) integration with constraint logic programming, and (c) empirical results illustrating how the proposed encodings outperform existing methods by orders of magnitude on the selected problems.


A range of application areas within the purview of knowledge representation and reasoning (e.g., questionanswering, computer-aided learning) require the ability to perform mixed qualitative-quantitative reasoning about space: mereotopology, direction, size, proximity. However, what is critically lacking is a general, robust, and scalable KR-centered foundation for modularly supporting spatial reasoning within diverse KR frameworks -e.g., (constraint) logic programming, answer-set programming, description logics, and so on.

Although this gap between KR frameworks and spatial reasoning exists, qualitative spatial representation and reasoning has received considerable attention from the artificial intelligence community, especially from the viewpoint of spatial information theory, and knowledge representation and reasoning. Knowledge representation and reasoning about space may be formally interpreted within diverse frameworks such as: (a) geometric reasoning \& constructive (solid) geometry (Kapur and Mundy, 1988); (b) relational algebraic semantics of 'qualitative spatial calculi' (Ligozat, 2011); and (c) by axiomatically constructed formal systems of mereotopology and mereogeometry (Aiello, PrattHartmann, and Benthem, 2007). Thus, there is a growing need to bring these theoretical advances in spatial reasoning

[^0]into more mainstream use in AI through a seamless integration with wider KR systems to facilitate e.g. rule-based spatial reasoning, spatially valid action planning, combined ontological-spatial reasoning, and so on. Recent initiatives in this direction include constraint logic programming over qualitative spaces, CLP(QS) (Bhatt, Lee, and Schultz, 2011), and Answer Set Programming modulo theories over qualitative spaces, ASPMT(QS) (Walega, Bhatt, and Schultz, 2015), both of which are based on polynomial constraint solving via satisfiability modulo theories (SMT) for spatial reasoning.

We investigate the utilisation of constructive geometric constraint solving as a mechanism for modularly equipping KR-based frameworks with commonsense spatial reasoning over a range of spatial domains. The tasks we focus on specifically include: (1) consistency: determining whether a concrete configuration of objects exists that satisfies the given qualitative relations; (2) instantiation: producing a consistent concrete configuration of objects that satisfies the qualitatively described scenario; (3) dynamic geometry: allowing a user to modify a consistent configuration, and in real-time have the solver update the other objects so that the given qualitative relations are maintained.

Axiomatic methods The state of the art in qualitative spatial reasoning using relational algebraic methods (e.g. the left-right calculus LR (Ligozat, 1993)) (Ligozat, 2011) has resulted in prototypical algorithms and black-box systems that do not integrate with KR languages, such as those dealing with semantics and conceptual knowledge necessary for handling background knowledge, action \& change, relational learning, rule-based systems etc. Moreover, while efficient, these methods are incomplete in general (Ligozat, 2011; Ladkin and Maddux, 1994; Lee, 2014) ${ }^{1}$, cannot deal with problems where partial numerical information is available (although some research exists in this direction, see

[^1](Liu et al., 2011)), and cannot be used for the instantiation task in general nor dynamic geometry.
Geometric Constraint Solving Alternatively, the field of geometric constraint solving adopts an analytic geometry approach where classes of objects are parameterised, and spatial relations are encoded as systems of polynomial equation and inequality constraints (Chou, 1988). For example, we can define a sphere as having a 3D centroid point $(x, y, z)$ and a radius $r$, where $x, y, z, r$ are reals. Two spheres $s_{1}, s_{2}$ externally connect or touch if
\[

$$
\begin{equation*}
\left(x_{s_{1}}-x_{s_{2}}\right)^{2}+\left(y_{s_{1}}-y_{s_{2}}\right)^{2}+\left(z_{s_{1}}-z_{s_{2}}\right)^{2}=\left(r_{s_{1}}+r_{s_{2}}\right)^{2} \tag{1}
\end{equation*}
$$

\]

If the system of polynomial constraints is satisfiable then the spatial constraints are consistent. The system of polynomial (in)equalities over variables $X$ is satisfiable if there exists a real number assignment for each $x \in X$ such that the (in)equalities are true. Partial numerical information is utilised by assigning the given real numerical values to the corresponding object parameters.

A range of algorithms have been developed for geometric constraint solving via solving systems of polynomial constraints, and can be broadly categorised as: numerical optimisation (e.g. (Ge, Chou, and Gao, 1999)), symbolic methods (e.g. (Chou, 1988; Gao and Chou, 1998b)), and constructive methods (e.g. (Owen, 1991; Bouma et al., 1995; Gao and Chou, 1998a)).
Constructive Geometric Constraint Solving In a seminal paper, Owen (1991) presents a method now termed Constructive Geometric Constraint Solving (CGCS) or the graph reduction approach. Owen identifies a particular set of spatial relations that, on one hand, are useful for a wide range of applications, particularly engineering and computer aided manufacturing, and on the other hand, can be reasoned about efficiently enough to address real-world scale problems. The particular set of relations correspond to distance, incidence, and angle constraints that can be encoded as quadratic equations over 2D points, lines, and circles. Geometrically, these correspond to relations that can be constructed using the familiar idealised ruler and compass from Euclid's Elements (Heath, 1956). We refer to this restricted set of spatial relations as the standard geometric constraint language. This set of relations is now standard within the geometric constraint solving community (see (Lee and Kim, 1998; Bouma et al., 1995)), and all prominent, industry-standard constraint solvers adopt precisely this set of relations, particularly within Computer Aided Design and Manufacturing (e.g. Autodesk Inventor, ${ }^{2}$ LEDAS LGS2D, ${ }^{3}$ FreeCAD ${ }^{4}$ ). We emphasise that CGCS is capable of solving our required consistency, instantiation, and dynamic geometry tasks.

## An Integrated Spatial-KR Framework

Figure 1 illustrates a system diagram of the interaction of the spatial solver with KR frameworks. The solver is embedded natively within each KR framework; the basic me-

[^2]

Figure 1: System diagram of the modular spatial solver component integrated into various KR frameworks.
chanics of the spatial solver are independent of the particular KR framework, and are accessed by a standard interface. In the case of constraint logic programming we have built on top of CLP(QS) (Bhatt, Lee, and Schultz, 2011) through Prolog's foreign language interface. In the case of ASPMT we are building on top of ASPMT(QS) (Walega, Bhatt, and Schultz, 2015) so that spatially inconsistent stable models are rejected.

Conceptually, the KR framework handles control of the overall reasoning procedure. When a select set of spatial predicates are encountered, as specified in the solver's spatial language library, they are sent to the spatial solver by the controlling KR framework. The spatial language is readily extensible, in this paper we present a sample of relations that are currently supported. The solver accepts conjunctions of spatial predicates with variables ranging over spatial domains (i.e. the objects in a configuration) and determines consistency. The solver provides consistent configurations, and additionally the solving history of configurations is accessible (particularly to support diagnosing inconsistencies). A configuration can be modified and updated to maintain the qualitative constraints (i.e. dynamic geometry). Objects and relations can be removed to facilitate backtracking.

The Standard Geometric Constraint Language The spatial domains of objects in the standard geometric constraint language are points $\mathbf{P}$, lines $\mathbf{L}$, and circles $\mathbf{C}$ :

- a point $p \in \mathbf{P}$ is a pair of reals, $\left(x_{p}, y_{p}\right) \in \mathbb{R}^{2}$;
- a line $l_{a b} \in \mathbf{L}$ is a pair of distinct points, $a, b \in \mathbf{P}, a \neq b$;
- a circle $C_{i} \in \mathbf{C}$ is a circle with centre point $p_{i} \in \mathbf{P}$ and radius $r_{i} \in \mathbb{R}, 0<r_{i}$.

We use lower case letters to refer to points. We use $l_{p_{1} p_{2}}$ to refer to lines between points in the subscript. We use upper case $C_{i}$ with a subscript number (if needed) to refer to circles. For brevity, if two points $a, b$ have been declared, then we can refer to the line $l_{a b}$ without explicitly quantifying $l$, e.g. let $\varphi$ be a predicate, then:
$\exists a, b \in \mathbf{P}, \exists l_{a b} \in \mathbf{L}\left(\varphi\left(l_{a b}\right)\right) \equiv \exists a, b \in \mathbf{P}\left(\varphi\left(l_{a b}\right)\right)$.
Table 1 presents polynomial encodings for the standard set of geometric constraints between points, lines, circles. They correspond to:

- incidence between points-lines, and points-circles (collinear, coincident);

| Relation | Polynomial Encoding |
| :--- | :--- |
| collinear (COLL) <br> (point $p$, line $l_{a b}$ ) | $\left(x_{b}-x_{a}\right)\left(y_{p}-y_{a}\right)=\left(x_{b}-y_{a}\right)\left(x_{p}-x_{a}\right)$ |
| coincident (COIN) <br> (point $a$, circle $C_{i}$ ) | $\left(x_{p_{i}}-x_{a}\right)^{2}+\left(y_{p_{i}}-y_{a}\right)^{2}=r_{i}^{2}$ |
| perpendicular (PERP) <br> (lines $l_{a b}, l_{c d}$ ) | $\left(y_{b}-y_{a}\right)\left(y_{d}-y_{c}\right)=-\left(x_{b}-x_{a}\right)\left(x_{d}-x_{c}\right)$ |
| parallel (PARA) <br> (lines $\left.l_{a b}, l_{c d}\right)$ | $\left(y_{b}-y_{a}\right)\left(x_{d}-x_{c}\right)=\left(y_{d}-y_{c}\right)\left(x_{b}-x_{a}\right)$ |
| angle (ANG) <br> (points $a, b, p$, constant $\theta)$ | $\theta=\operatorname{atan2((y_{b}-y_{p}),(x_{b}-x_{p}))}$ <br> $-\operatorname{atan2} 2\left(\left(y_{a}-y_{p}\right),\left(x_{a}-x_{p}\right)\right)$ |

Table 1: Polynomial encodings of geometric constraint relations.

- orientation between lines (parallel, perpendicular);
- constant distance and angles for lines and circles (distance between points, radii of circles, angle between points $a, b$ about a reference point $p$ ).


## Expressing Qualitative Spatial Relations using Geometric Constraint Languages

In this section we present a range of novel encodings that enable us to reason about qualitative spatial relations over extended regions (in particular, relative orientation, size, proximity, and mereotopology over regions) using traditional geometric constraint solving methods that are restricted to the standard geometric constraint language.
Point-segment coincidence While the collinear constraint between points and lines is common in geometric constraint systems (i.e. a point lies anywhere on an infinite line), the ability to constrain a point to lie coincident on a line segment (i.e. between two points) is typically not supported. The following encoding realises a coincidence constraint between a point $p$ and a segment $l_{a b}$.

Firstly we define a useful BRACE relation between a line segment and a circle that ensures the diameter of the circle is equal to the length of the segment (Figure 2(a)). That is, the endpoints of the line $l_{a b}$ are constrained to lie on the circle $C_{i}$, and the centroid of $C_{i}$ is constrained to be collinear with $l_{a b}$.

$$
\begin{aligned}
& \mathrm{BRACE}\left(l_{a b}, C\right) \equiv \\
& \quad \operatorname{COIN}\left(a, C_{i}\right) \wedge \operatorname{COIN}\left(b, C_{i}\right) \wedge \operatorname{COLL}\left(p_{i}, l_{a b}\right)
\end{aligned}
$$

As illustrated in Figure 2(b), the point-segment coincidence encoding adds a circle $C_{1}$ and a brace relation with line $l_{a b}$, so that the diameter of $C_{1}$ equals the length of $l_{a b}$. Next, the encoding adds a line $l_{p c}$ perpendicular to $l_{a b}$ with endpoint $c$ coincident to $C_{1}$. The perpendicular constraint ensures that the two lines always intersect within the interior of $C_{1}$. Finally, the given point $p$ is constrained to be collinear to $l_{a b}$, and is thus always constrained to lie on the segment $l_{a b}$.

$$
\begin{aligned}
& \operatorname{COIN}\left(p, l_{a b}\right) \equiv \exists C_{1} \in \mathbf{C}, \exists c \in \mathbf{P}\left(\operatorname{BRACE}\left(l_{a b}, C_{1}\right)\right. \\
& \left.\quad \wedge \operatorname{COIN}\left(c, C_{1}\right) \wedge \operatorname{PERP}\left(l_{a b}, l_{p c}\right) \operatorname{COLL}\left(p, l_{a b}\right)\right)
\end{aligned}
$$

For convenience and brevity we also define the relation that segment $l_{a b}$ is coincident with segment $l_{c d}$ as: the endpoints $a, b$ are coincident with the segment $l_{c d}$.


Figure 2: (a) Brace relation between circle $C_{i}$ and line $l_{a b}$; (b) point $p$ is constrained to lie on the segment between points $a, b$.

$$
\operatorname{COIN}\left(l_{a b}, l_{c d}\right) \equiv \operatorname{COIN}\left(a, l_{c d}\right) \wedge \operatorname{COIN}\left(b, l_{c d}\right)
$$

Point $p$ can not be equal to either endpoint $a, b$ as the line $l_{c d}$ can not have zero length. If we drop this line constraint for $l_{c d}$ so that $c, d$ can also be equal, then $p$ can also equal the endpoints. These are useful predicates for defining topological relations, and thus we refer to them as: $\operatorname{COIN} \subseteq\left(p, l_{a b}\right)$ and $\operatorname{COIN} \subseteq\left(l_{a b}, l_{c d}\right)$.

Relative Orientation As illustrated in Figure 3, the encoding for the left of relation adds a new point $c$ collinear to the given line $l_{a b}$, and adds a line $l_{c p}$, between the given point $p$ and the new point $c$. The encoding then adds the constraint that the angle between $l_{a b}$ and $l_{c p}$ is $90^{\circ}$ counterclockwise. The length of the line $l_{c p}$ is unbounded, and thus $p$ can be moved an arbitrary distance away from $l_{a b}$. The key is that, if $p$ is moved to the right side of $l_{a b}$, then the angle constraint is violated, and thus $p$ is forced to remain on the left side.

```
\(\operatorname{LEFT}\left(p, l_{a b}\right) \equiv \exists c \in \mathbf{P}\left(\operatorname{ANG}\left(b, p, c, \frac{\pi}{2}\right) \wedge \operatorname{COLL}\left(c, l_{a b}\right)\right)\)
\(\operatorname{RIGHT}\left(p, l_{a b}\right) \equiv\)
    \(\exists c \in \mathbf{P}\left(\operatorname{ANG}\left(b, p, c,-\frac{\pi}{2}\right) \wedge \operatorname{COLL}\left(c, l_{a b}\right)\right)\)
```

We extend this definition to relative orientation relations between lines and circles (Figure 3(b)).

$$
\begin{aligned}
& \operatorname{LEFT}\left(C_{1}, l_{a b}\right) \equiv \exists c, d \in \mathbf{P}\left(\operatorname{ANG}\left(b, p_{1}, c, \frac{\pi}{2}\right)\right. \\
& \left.\quad \wedge \operatorname{COLL}\left(c, l_{a b}\right) \wedge \operatorname{COIN}\left(d, C_{1}\right) \wedge \operatorname{COIN}\left(d, l_{c p_{1}}\right)\right) \\
& \operatorname{RIGHT}\left(C_{1}, l_{a b}\right) \equiv \exists c, d \in \mathbf{P}\left(\operatorname{ANG}\left(b, p_{1}, c,-\frac{\pi}{2}\right)\right. \\
& \left.\quad \wedge \operatorname{COLL}\left(c, l_{a b}\right) \wedge \operatorname{COIN}\left(d, C_{1}\right) \wedge \operatorname{COIN}\left(d, l_{c p_{1}}\right)\right)
\end{aligned}
$$

Topological relations between circles In standard geometric constraint solvers there is no way of directly specifying mereotopological constraints between higher-level objects and regions such as circles, squares, triangles, polygons, and so on. In this section we present encodings for topological relations between circles, and then use these encodings as a basis for defining relations between more general regions.

We adopt the terminology of the prominent topological spatial logic, the Region Connection Calculus (RCC) Randell, Cui, and Cohn (1992): disconnects (DC), externally connects (EC), partial overlap (PO), tangential proper part (TPP), non-tangential proper part (NTPP), proper part


Figure 3: (a) Point $p$ is constrained to lie anywhere to the left of line $(a, b)$. The angle from point $b$ to $p$ about $c$ is fixed at $\frac{\pi}{2}$ counter-clockwise. The distance between $c, p$ is not constrained. (b) Circle $C_{1}$ is constrained to lie anywhere to the left of line $(a, b)$.
(PP), part of (P), discrete from (DR), equal (EQ). Note that EQ between two circles is trivially satisfied by constraining the centroids and radii to be equal.

The topological relation encodings are illustrated in Figure 4. To ensure circle intersection (e.g. TPP, NTPP, PO), the encodings constrain one or both endpoints of the brace segments of one circle to be coincident to the brace segment of the other circle; a pair of brace endpoints are made equal for boundary contact (e.g. TPP). EC is encoded with a point of boundary contact $a$ that is coincident to a segment $l_{p_{1} p_{2}}$ between the circle centroids. DC is encoded by introducing a third circle $C_{3}$ so that one endpoint of each of the braces of $C_{1}$ and $C_{2}$ lie on different sides of the centroid of $C_{3}$, along the brace of $C_{3}$.

Observe that the brace segment within a circle can be rotated about the circle's centroid. Thus, considering NTPP for example, $C_{1}$ can occupy any circular region within $C_{2}$ by moving $C_{1}$ along the brace of $C_{2}$, and rotating the brace of $C_{2}$.

$$
\begin{gathered}
\operatorname{TPP}\left(C_{1}, C_{2}\right) \equiv \exists l_{a b}, l_{a c} \in \mathbf{L}\left(\operatorname{BRACE}\left(l_{a b}, C_{2}\right)\right. \\
\left.\wedge \operatorname{BRACE}\left(l_{a c}, C_{1}\right) \wedge \operatorname{COIN}\left(c, l_{a b}\right)\right) \\
\operatorname{NTPP}\left(C_{1}, C_{2}\right) \equiv \exists l_{a b}, l_{c d} \in \mathbf{L}\left(\operatorname{BRACE}\left(l_{a b}, C_{2}\right)\right. \\
\left.\wedge \operatorname{BRACE}\left(l_{c d}, C_{1}\right) \wedge \operatorname{COIN}\left(l_{c d}, l_{a b}\right)\right) \\
\operatorname{PO}\left(C_{1}, C_{2}\right) \equiv \exists l_{a b}, l_{c d} \in \mathbf{L}\left(\operatorname{BRACE}\left(l_{a b}, C_{2}\right)\right. \\
\left.\wedge \operatorname{BRACE}\left(l_{c d}, C_{1}\right) \wedge \operatorname{COIN}\left(a, l_{c d}\right) \wedge \operatorname{COIN}\left(d, l_{a b}\right)\right) \\
\operatorname{EC}\left(C_{1}, C_{2}\right) \equiv \exists a \in \mathbf{P}\left(\operatorname{COIN}\left(a, l_{p_{1} p_{2}}\right)\right. \\
\left.\wedge \operatorname{COIN}\left(a, C_{1}\right) \wedge \operatorname{COIN}\left(a, C_{2}\right)\right) \\
\operatorname{DC}\left(C_{1}, C_{2}\right) \equiv \exists a, b \in \mathbf{P}, \exists C_{3} \in \mathbf{C}\left(\operatorname{BRACE}\left(l_{p_{1} p_{2}}, C_{3}\right)\right. \\
\\
\left.\wedge \operatorname{COIN}\left(a, l_{p_{1} p_{3}}\right) \wedge \operatorname{COIN}\left(a, C_{1}\right)\right) \\
\\
\left.\wedge \operatorname{COIN}\left(b, l_{p_{2} p_{3}}\right) \wedge \operatorname{COIN}\left(b, C_{2}\right)\right)
\end{gathered}
$$

We can drop the distinction between boundaries (i.e. corresponding to RCC5 and other RCC relations) by employing the modified coincident constraint between points and segments $\operatorname{COIN} \subseteq\left(p, l_{a b}\right)$, where a point $p$ can also equal the segment endpoints $l_{a b}$. Thus, we encode the definitions that:

- PP is a disjunction of NTPP and TPP;
- P is a disjunction of PP and EQ ;


Figure 4: Topological relations between circles.

- DR is a disjunction of DC and EC.

$$
\begin{aligned}
& \operatorname{PP}\left(C_{1}, C_{2}\right) \equiv \exists l_{a b}, l_{c d} \in \mathbf{L}\left(\operatorname{BRACE}\left(l_{a b}, C_{2}\right)\right. \\
& \left.\quad \wedge \operatorname{BRACE}\left(l_{c d}, C_{1}\right) \wedge \operatorname{COIN} \subseteq\left(c, l_{a b}\right) \wedge \operatorname{COIN}\left(d, l_{a b}\right)\right)
\end{aligned}
$$

$$
\begin{gathered}
\mathrm{P}\left(C_{1}, C_{2}\right) \equiv \exists l_{a b}, l_{c d} \in \mathbf{L}\left(\operatorname{BRACE}\left(l_{a b}, C_{2}\right)\right. \\
\left.\wedge \operatorname{BRACE}\left(l_{c d}, C_{1}\right) \wedge \operatorname{COIN} \subseteq\left(l_{c d}, l_{a b}\right)\right) \\
\operatorname{DR}\left(C_{1}, C_{2}\right) \equiv \exists a, b \in \mathbf{P}, \exists C_{3} \in \mathbf{C}\left(\operatorname{BRACE}\left(l_{p_{1} p_{2}}, C_{3}\right)\right. \\
\left.\wedge \operatorname{COIN} \subseteq\left(a, l_{p_{1} p_{3}}\right) \wedge \operatorname{COIN}\left(a, C_{1}\right)\right) \\
\wedge \operatorname{COIN} \subseteq^{\left.\left(b, l_{p_{2} p_{3}}\right) \wedge \operatorname{COIN}\left(b, C_{2}\right)\right)}
\end{gathered}
$$

Qualitative Size and Proximity We can make $C_{1}$ strictly larger than $C_{2}$ (Figure 5(a)) by introducing a point $c$ collinear with $l_{p_{1} p_{2}}$, making segment $l_{a c}$ perpendicular to $l_{p_{1} p_{2}}$, and adding a circle $C_{3}$ centred on $p_{1}$ (i.e. $p_{3}=p_{1}$ ) coincident with $c$. As the radius of $C_{3}$ is necessarily greater than 0 , the length of $l_{p_{1} a}$ is greater than $l_{p_{2} b}$ by Pythagoras' theorem. For the equi-sized relation, constraining two radii to be equal is already supported as a primitive relation in the standard geometric constraint language.

$$
\begin{aligned}
& \operatorname{LARGER}\left(C_{1}, C_{2}\right) \equiv \exists a, b, c \in \mathbf{P}, \exists C_{3} \in \mathbf{C}( \\
& \quad \operatorname{COIN}\left(a, C_{1}\right) \wedge \operatorname{COIN}\left(b, C_{2}\right) \wedge \operatorname{COLL}\left(c, l_{p_{1} p_{2}}\right) \\
& \left.\wedge p_{3}=p_{1} \wedge \operatorname{PARA}\left(l_{p_{1} p_{2}}, l_{a b}\right) \wedge \operatorname{PERP}\left(l_{p_{2} b}, l_{a b}\right)\right)
\end{aligned}
$$

We can use this right triangle construction to order circles in terms of proximity (Figure 5(b)). Segment $l_{p_{1} b}$ is at least as long as $l_{p_{1} a}$, and $l_{p_{1} c}$ is at least as long as $l_{p 1 b}$, and thus $C_{1}$ is nearer to $C_{2}$ than $C_{3}$. Additionally we can impose a


Figure 5: (a) $C_{1}$ larger than $C_{2}$ (b) $C_{1}$ nearer $C_{2}$ than $C_{3}$.
part of constraint between $C_{2}$ and $C_{5}$ to express that $C_{1}$ is nearer to all parts of $C_{2}$ than $C_{3}$.

```
\(\operatorname{NEARER}\left(C_{1}, C_{2}, C_{3}\right) \equiv \exists a, b, c \in \mathbf{P}, \exists C_{4}, C_{5} \in \mathbf{C}(\)
    \(p_{4}=p_{1} \wedge p_{5}=p_{1} \wedge \operatorname{COIN}\left(a, C_{1}\right) \wedge \operatorname{COIN}\left(b, C_{4}\right)\)
        \(\wedge \operatorname{COIN}\left(c, C_{5}\right) \wedge \operatorname{PERP}\left(l_{p_{1} a}, l_{a b}\right) \wedge \operatorname{PERP}\left(l_{p_{1} b}, l_{b c}\right)\)
        \(\left.\wedge \mathrm{EC}\left(C_{2}, C_{4}\right) \wedge \mathrm{EC}\left(C_{3}, C_{5}\right)\right)\)
```

Egg-yolk approach for defining relations between regions We employ the egg-yolk method of modelling regions with indeterminante boundaries (Cohn and Gotts, 1996) to characterise a class of regions (including polygons) that satisfies topological, relative direction, qualitative size and proximity relations. Each egg-yolk region is an equivalence class for all regions that are contained within the upper approximation (the egg white), and completely contain the lower approximation (the egg yolk). Let $\mathbf{R}$ be the domain of egg-yolk regions, where egg-yolk region $R \in \mathbf{R}$ consists of a circular upper and a lower approximation $R^{+}, R^{-} \in \mathbf{C}$ such that $\operatorname{NTPP}\left(R^{-}, R^{+}\right)$(see Figure 6(a)).

We can realise these regions through constructive geometric constraint encodings, giving us a method of generating arbitrary regions that satisfy qualitative spatial constraints. We declaratively define a (simple, non-self-intersecting) polygon as a sequence of vertices such that:

1. all vertices are contained within the upper approximation
2. no segment between adjacent vertices intersects the lower approximation
3. the (absolute) winding number about the centroid of the lower approximation is 1
We can generate polygons by placing $n$ vertices on the upper approximate circle, evenly distributed (satisfying Condition 3), such that each vertex and line segment is geometrically constrained to satisfy Conditions 1 and 2 above. The user can explore the space of consistent polygons directly through dynamic geometry Winroth (1999), or polygons can be randomly generated.

Relative orientation between egg-yolk regions and lines (see Figure 3(b)) and qualitative size and proximity can now be defined based on the approximations:

```
\(\operatorname{LEFT}\left(R, l_{a b}\right) \equiv \operatorname{LEFT}\left(R^{+}, l_{a b}\right)\)
\(\operatorname{RIGHT}\left(R, l_{a b}\right) \equiv \operatorname{RIGHT}\left(R^{+}, l_{a b}\right)\)
\(\operatorname{LARGER}\left(R_{1}, R_{2}\right) \equiv \operatorname{LARGER}\left(R_{1}^{-}, R_{2}^{+}\right)\)
\(\operatorname{NEARER}\left(R_{1}, R_{2}, R_{3}\right) \equiv \operatorname{NEARER}\left(R_{1}^{+}, R_{2}^{-}, R_{3}^{+}\right)\)
```



Figure 6: Egg-yolk region $R$ is defined by a lower circular approximation $R^{-}$and an upper circular approximation $R^{+}$.

| Solver | CLP(QS) | z 3 | Redlog | GQR |
| :---: | ---: | ---: | ---: | ---: |
| $n=3$ | 0.111 | 0.020 | 0.626 | 0.001 |
| $n=4$ | 0.903 | 21.294 | 0.629 | 0.001 |
| $n=5$ | 6.979 | time out | 173.852 | fail |

Table 2: Time (sec) to solve the circle contact problem.

The following topological relations between pairs of eggyolk regions are defined based on the relation between their approximations (see Figure 6):

$$
\begin{aligned}
\operatorname{PP}\left(R_{1}, R_{2}\right) \equiv & \mathrm{P}\left(R_{1}^{+}, R_{2}^{-}\right) \\
\operatorname{DC}\left(R_{1}, R_{2}\right) \equiv & \mathrm{DC}\left(R_{1}^{+}, R_{2}^{+}\right) \\
\operatorname{DR}\left(R_{1}, R_{2}\right) \equiv & \operatorname{DR}\left(R_{1}^{+}, R_{2}^{+}\right) \\
\operatorname{PO}\left(R_{1}, R_{2}\right) \equiv & \operatorname{PO}\left(R_{1}^{-}, R_{2}^{-}\right) \wedge \\
& \operatorname{PO}\left(R_{1}^{+}, R_{2}^{-}\right) \wedge \operatorname{PO}\left(R_{1}^{-}, R_{2}^{+}\right)
\end{aligned}
$$

The partial overlap definition requires some explanation: the partial overlap condition between the lower approximations ensures that the regions share a common interior part, but one region might completely contain the other. The partial overlap condition between the upper and lower approximations ensures that the regions each have interior parts not shared by the other, but the regions could still be disconnected. Thus, together the conditions encode the partial overlap relation between egg-yolk regions.

The egg-yolk relation encodings are sound, i.e. they correctly encode the intended relation between the true regions, and are incomplete, i.e. they do not capture all possible ways that the true regions can satisfy the intended relation.

## Empirical Evaluation

In this section we empirically compare our spatial solver (based on FreeCAD CGCS, integrated in CLP(QS)) with other popular spatial reasoning approaches: z3 SMT solver (De Moura and Bjørner, 2008), Cylindrical Algebraic Decomposition in Redlog (Dolzmann, Seidl, and Sturm, 2004), and GQR qualitative spatial calculus solver (Gantner, Westphal, and Wölfl, 2008). Experiments were run on a MacBookPro, OS X 10.8.5, 2.6 GHz, Intel Core i7. In the experiments, time out was issued after a runtime of 10 minutes.

Circle Contact Determine whether $n$ circles can be mutually externally connected; consistent for $2 \leq n \leq 4$ (see Table 2).

| $n$ | 4 | 5 | 6 | 7 | 8 | 9 |
| :---: | ---: | ---: | ---: | ---: | ---: | ---: |
| $\mathbf{C L P}(\mathbf{Q S})$ | 0.233 | 0.431 | 0.847 | 1.577 | 2.629 | 4.522 |
| $\mathbf{z 3}$ | 0.022 | time out | time out | time out | time out | time out |
| Redlog | 3.571 | 4.508 | time out | time out | time out | time out |
| $\mathbf{G Q R}$ | $n / a$ | $n / a$ | $n / a$ | $n / a$ | $n / a$ | $n / a$ |

Table 3: Time (sec) to solve the spiral chain problem, $n=4 \ldots 10$.


Figure 7: A spiral chain for $n=7$.

Spiral Chain Given $n$ circles, make circles $i,(i+1)$ externally connected, and make the centroid of circle $(i+2)$ left of the line between centroids $i,(i+1)$, for $i=1, \ldots, n-1$. This problem combines topology and orientation relations (see Fig 7 and Table 3).
Lamp Design The lamp has a base and three bars connected by three joints; the joints can only turn inwards; the lamp shade connects to the third joint; the bulb must fit completely within the lamp shade. Figure 8 illustrates the constraint graph and corresponding FreeCAD interactive diagram that maintains the specified qualitative relations. As the user manipulates the diagram, the FreeCAD geometric solver maintains the qualitative constraints in real time. The solving time is 0.001 seconds per adjustment.

```
point(Base), point(Joint1),
point(Joint2), point(Joint3),
line(Bar1, point(Base), point(Joint1)),
line(Bar2, point(Joint1), point(Joint2)),
line(Bar3, point(Joint2), point(Joint3)),
circle(Shade), circle(Lamp),
fix(Base, point(0,0)),
length(Bar1, value(31)),
orientation(left_of, Joint2, Bar1),
orientation(left_of, Joint3, Bar2),
topology (proper_part, Lamp, Shade).
```

Next, we can specify Prolog queries that check possible relations between objects that were not directly constrained. Prolog finds potential spatial relations that could apply according to its knowledge base and then determines spatial validity by consulting the spatial reasoning module. When Prolog finds a valid solution, a consistent configuration is provided (as a consequence of the CGCS solving process). Moreover, the configuration is dynamic and can be manipulated within the current set of qualitative constraints:

```
?- orientation(Relation, Shade, Bar2).
Relation = left_of
```

When we backtrack Prolog reports the next qualitative relation in front with the corresponding dynamic configuration. Further backtracks do not yield any more qualitative


Figure 8: (a) Constraint graph of lamp product design; (b) screenshot of corresponding FreeCAD interactive lamp diagram with qualitative constraints.
solutions.

```
...;
Relation = in_front.
```


## Discussion and Future Work

We have presented a framework and geometric encodings that enable the application of constructive geometric constraint solving for spatial reasoning within a KR-based paradigm, specifically CLP(QS). Thus we employ rulebased reasoning for formalising domain knowledge and supporting querying and inference, extended to natively interpret qualitative spatial predicates. Our spatial reasoning approach can be modularly applied to other KR frameworks, e.g. we have experimented with integrating our spatial module with ASPMT(QS) for reasoning about action and spatial change; results are forthcoming.

It is straightforward to show that our definitions (except relative orientation) hold true in the 3D case when we exchange 2D points for 3D points, 2D lines for 3D lines, and circles for spheres. Our relative orientation (left, right) definition is modified to be defined with respect to 3D points and planes. Due to limited space we have omitted 3D spatial domains, although our preliminary experiments with Autodesk Inventor 3D CGCS are promising.

An interesting open question is how to handle inconsistent qualitative spatial constraints in general within a KR framework: methods such as Cylindrical Algebraic Decomposition are both sound and complete, whereas constructive geometric constraint solving is incomplete in general. Thus, a result of inconsistency using constructive approaches is usually annotated with some measure of confidence (i.e. the problem, or sub-problems, are executed a number of times with different initial randomised parameter values until no further progress towards a solution is made). Identifying tractable classes of qualitative problems that have specific properties with respect to completeness (and statistical confidence in the case of reported inconsistency) is an interesting direction for future research.

## References

Aiello, M.; Pratt-Hartmann, I. E.; and Benthem, J. F. v. 2007. Handbook of Spatial Logics. Secaucus, NJ, USA: SpringerVerlag New York, Inc.

Bhatt, M.; Lee, J. H.; and Schultz, C. 2011. CLP(QS): A Declarative Spatial Reasoning Framework. In Proceedings of the 10th international conference on Spatial information theory, COSIT' 11, 210-230. Berlin, Heidelberg: SpringerVerlag.
Bouma, W.; Fudos, I.; Hoffmann, C.; Cai, J.; and Paige, R. 1995. Geometric constraint solver. Computer-Aided Design 27(6):487-501.
Chou, S.-C. 1988. Mechanical geometry theorem proving, volume 41. Springer Science \& Business Media.
Cohn, A. G., and Gotts, N. M. 1996. The 'egg-yolk' representation of regions with indeterminate boundaries. Geographic objects with indeterminate boundaries 2:171-187.
De Moura, L., and Bjørner, N. 2008. Z3: An efficient smt solver. In Tools and Algorithms for the Construction and Analysis of Systems. Springer. 337-340.
Dolzmann, A.; Seidl, A.; and Sturm, T. 2004. REDLOG User Manual. Edition 3.0.

Gantner, Z.; Westphal, M.; and Wölff, S. 2008. GQR-A fast reasoner for binary qualitative constraint calculi. In Proc. of $A A A I$, volume 8.
Gao, X.-S., and Chou, S.-C. 1998a. Solving geometric constraint systems. i. a global propagation approach. ComputerAided Design 30(1):47-54.

Gao, X.-S., and Chou, S.-C. 1998b. Solving geometric constraint systems. ii. a symbolic approach and decision of rcconstructibility. Computer-Aided Design 30(2):115-122.

Ge, J.-X.; Chou, S.-C.; and Gao, X.-S. 1999. Geometric constraint satisfaction using optimization methods. ComputerAided Design 31(14):867-879.
Heath, T. L. 1956. The thirteen books of Euclid's Elements, volume 1. Courier Dover Publications.
Kapur, D., and Mundy, J. L., eds. 1988. Geometric Reasoning. Cambridge, MA, USA: MIT Press.

Ladkin, P. B., and Maddux, R. D. 1994. On binary constraint problems. Journal of the ACM (JACM) 41(3):435-469.
Lee, J. Y., and Kim, K. 1998. A 2-d geometric constraint solver using dof-based graph reduction. Computer-Aided Design 30(11):883-896.
Lee, J. H. 2014. The complexity of reasoning with relative directions. In 21st European Conference on Artificial Intelligence (ECAI 2014).
Ligozat, G. F. 1993. Qualitative triangulation for spatial reasoning. In Spatial Information Theory A Theoretical Basis for GIS. Springer. 54-68.
Ligozat, G. 2011. Qualitative Spatial and Temporal Reasoning. Wiley-ISTE.

Liu, W.; Wang, S.; Li, S.; and Liu, D. 2011. Solving qualitative constraints involving landmarks. In Principles and Practice of Constraint Programming-CP 2011. Springer. 523-537.

Owen, J. C. 1991. Algebraic solution for geometry from dimensional constraints. In Proceedings of the first ACM symposium on Solid modeling foundations and CAD/CAM applications, 397-407. ACM.
Randell, D. A.; Cui, Z.; and Cohn, A. 1992. A spatial logic based on regions and connection. In KR'92. Principles of Knowledge Representation and Reasoning. San Mateo, California: Morgan Kaufmann. 165-176.
Walega, P. A.; Bhatt, M.; and Schultz, C. 2015. ASPMT (QS): Non-Monotonic Spatial Reasoning with Answer Set Programming Modulo Theories. International Conference on Logic Programming and Non-Monotonic Reasoning (LPNMR).
Winroth, H. 1999. Dynamic projective geometry. Tekniska högsk.


[^0]:    Copyright (C) 2015, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

[^1]:    ${ }^{1}$ Incompleteness refers to the inability of a spatial reasoning method to determine, in general, whether a given set of qualitative spatial constraints is consistent or inconsistent. Relation-algebraic spatial reasoning (i.e. using algebraic closure based on weak composition) has been shown to be incomplete for a number of spatial languages and cannot guarantee consistency in general, e.g. relative directions (Lee, 2014) and containment relations between linearly ordered intervals (Ladkin and Maddux, 1994), Theorem 5.9.

[^2]:    ${ }^{2}$ www.autodesk.com/products/inventor/overview
    ${ }^{3}$ ledas.com/products/lgs2d/
    ${ }^{4}$ www.freecadweb.org/

