A new perspective on reasoning with qualitative spatial knowledge

Jae Hee Lee and **Diedrich Wolter**
SFB/TR 8 Spatial Cognition
Universität Bremen
P.O. Box 330440 28334 Bremen, Germany

Abstract

In this paper we call for considering a paradigm shift in the reasoning methods that underly qualitative spatial representations. As alternatives to conventional methods we propose exploiting methods from linear programming and real algebraic geometry. We argue that using mathematical theories of the spatial domain at hand might be the key to effective reasoning methods, and thus to practical applications.

1 Introduction

Qualitative spatial knowledge is ubiquitous in natural language. Thus, it is essential in human-computer interaction, which is an integral part of our everyday life where interaction with digital equipments is omnipresent. In the field of artificial intelligence, reasoning with qualitative spatial knowledge has been researched under the umbrella term *Qualitative Spatial Reasoning (QSR)* [Cohn and Renz, 2008]. QSR pursues a relation-algebraic approach that provides universal means to deal with any type of qualitative spatial knowledge (e.g., topology, direction, distance). It has been assumed that the relation-algebraic approach will allow for an efficient, effective, universal reasoning method. Despite its promising properties, however, the relation-algebraic approach suffers from its incompleteness for many representations of qualitative spatial knowledge. Furthermore, it is not capable of generating a model for given constraints, which is a desirable feature for many real-world applications.

In this paper we call for considering a paradigm shift in the reasoning methods that underly qualitative spatial representations. As alternatives to conventional methods we propose exploiting methods from linear programming and real algebraic geometry. We argue that using mathematical theories of the spatial domain at hand might be the key to effective reasoning methods, and thus to practical applications.

2 The Relation-Algebraic Approach and Its Limitations

The building blocks of QSR are a spatial domain $\mathcal{D}$, a finite set $\mathcal{R} = \{R_1, R_2, \ldots, R_n\}$ of binary relations on $\mathcal{D}$ which partitions $\mathcal{D}^2$, and a map $\circ : \mathcal{R} \times \mathcal{R} \to 2^{\mathcal{D}}, R_1 \circ R_2 = \{y \in \mathcal{D} \mid xR_1y \text{ and } yR_2z\}$, which is called the *composition*. A prominent, simple example is the one-dimensional space (e.g., a queue) equipped with the relations *before*, *behind*, *equal* and the usual notion of composition, e.g., if Alice is *behind* Bob and Bob is *behind* Charlie than Alice is *behind* Charlie (i.e., *behind* $\circ$ *behind* $=$ *behind*).

For a given domain $\mathcal{D}$ (e.g., a queue), a partition $\mathcal{R}$ (e.g., *before*, *behind*, *equal*), a composition $\circ$, a set of variables (e.g., Alice, Bob, Charlie), and a set of spatial constraints (e.g., Alice is *behind* Bob, Bob is *behind* Charlie, Charlie is *behind* Alice), a common *reasoning task* is figuring out whether there is an instantiation of the variables over the domain $\mathcal{D}$, such that the given spatial constraints are consistent (the example is not consistent, as there is no instantiation for Alice, Bob and Charlie that satisfies the constraints). For this reasoning problem QSR employs the path-consistency method, which is used for solving constraint satisfaction problems over *finite* domains. Since the domain $\mathcal{D}$ of interest in QSR is usually *infinite* as opposed to the domain of a finite CSP, partition $\mathcal{R}$ and composition $\circ$ have to meet certain requirements, such that the path-consistency method is applicable to the constraints (See [Renz and Nebel, 2007] and [Renz and Ligozat, 2005] for more details). A triple $(\mathcal{D}, \mathcal{R}, \circ)$ that meets those requirements forms a non-associative algebra; it forms a relation algebra, if it is additionally closed under composition [Ligozat, 2005].

We will call the reasoning approach that utilizes the path-consistency method the *relation-algebraic* approach. The main deficiency of the relation-algebraic approach is that there is no guarantee for its completeness, i.e., the algorithm can fail to identify all inconsistent scenarios. Accordingly, research has been concentrated on finding out whether the consistency of constraints defined by a triple $(\mathcal{D}, \mathcal{R}, \circ)$ can be decided with the path-consistency method. The recent result showed that spatial representations for directional information *cannot* be decided by the path-consistency method in general [Wolter and Lee, 2010]. Thus, we have to question the idea of keeping the relation-algebraic approach as a universal means, and should be open to search for alternative methods for a sound and complete reasoning.

The relation-algebraic approach is also not capable of providing models for the given input constraints. However, in real application domains (e.g., computer-aided design, geographic information systems) not only deciding the consistency of constraints, but also determining the positions of spatial objects satisfying those constraints is desired.

In the next two sections we introduce a selection of qualitative spatial representations for directional information, and methods for reasoning with those representations, which overcome the deficiencies of the relation-algebraic approach.

3 Representations for Qualitative Spatial Knowledge

If a set of spatial objects are represented by a finite number of points in the Euclidian space—which is generally the case in many applications—then the qualitative spatial relations between those objects can be described by a system of polynomial equations or inequalities. For example, we can model people in a queue as points in $\mathbb{R}$ and represent "Alice is *behind* Bob, Bob is *behind* Charlie, Charlie is *behind* Alice" with the system $x_A - x_B > 0 \wedge x_B - x_C > 0 \wedge x_C - x_A > 0$, where $x_A, x_B, x_C \in \mathbb{R}$.

If we leave the one-dimensional Euclidian space $\mathbb{R}$ and move to the two-dimensional Euclidian space $\mathbb{R}^2$, new constraints emerge which were not existent in the one-dimensional case. An important new constraint in the two-dimensional case is based on the relative positions of three points, i.e., whether the points are oriented in clockwise (CW) order, counterclockwise (CCW) order, or collinear. Formally, such a constraint can be expressed as a polynomial inequality or equation based on three points $p_1 = (x_1, y_1)$, $p_2 = (x_2, y_2)$ and $p_3 = (x_3, y_3)$ in the following way

$$x_2y_3 + x_1y_2 + x_3y_1 - y_2x_3 - y_1x_2 - y_3x_1 < 0 \qquad (\textit{CW})$$

$$x_2y_3 + x_1y_2 + x_3y_1 - y_2x_3 - y_1x_2 - y_3x_1 > 0 \qquad (\textit{CCW})$$

$$x_2y_3 + x_1y_2 + x_3y_1 - y_2x_3 - y_1x_2 - y_3x_1 = 0, \qquad (\textit{collin.})$$

where the polynomials on the lefthand side are obtained from

$$\det \begin{pmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{pmatrix}, \qquad (1)$$

where $\det$ stands for determinant. The importance and ubiquity of this relationship of three points in a plane will be evident in the next subsections, where we introduce a selection of qualitative spatial representations for directional information. In each of the subsections we will show how a relation of each spatial representation can be translated to a polynomial constraint, which is based on the relative position of three points presented above.

3.1 The $\mathcal{LR}$ calculus

The domain of the $\mathcal{LR}$ calculus [Scivos and Nebel, 2005] 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 relative position 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, which are depicted in Figure 1 are: **l**eft, **r**ight, **f**ront, **s**tart, **i**nbetween, **e**nd, **b**ack. In Figure 1 the Euclidian plane is partitioned by points p_1 and p_2, $p_1 \neq p_2$ into seven regions: 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

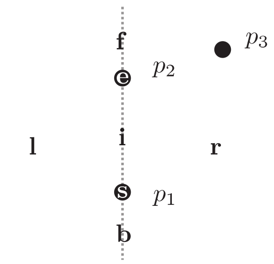

Figure 1: Illustration of $\mathcal{LR}$ relation $p_1\ p_2\ \mathbf{r}\ p_3$

$$\begin{aligned}
p_1\ p_2\ \mathbf{l}\ p_3 &\Leftrightarrow x_2y_3 + x_1y_2 + x_3y_1 - y_2x_3 - y_1x_2 - y_3x_1 > 0 \\
p_1\ p_2\ \mathbf{r}\ p_3 &\Leftrightarrow x_2y_3 + x_1y_2 + x_3y_1 - y_2x_3 - y_1x_2 - y_3x_1 < 0 \\
p_1\ p_2\ \mathbf{b}\ p_3 &\Leftrightarrow x_2y_3 + x_1y_2 + x_3y_1 - y_2x_3 - y_1x_2 - y_3x_1 = 0 \\
&\quad \wedge\ p_1\ p_2\ \mathbf{r}\ p_4\ \wedge\ p_4\ p_1\ \mathbf{l}\ p_3 \\
p_1\ p_2\ \mathbf{s}\ p_3 &\Leftrightarrow x_3 = x_1\ \wedge\ y_3 = y_1\ \wedge\ x_3 \neq x_2\ \wedge\ y_3 \neq y_2 \\
p_1\ p_2\ \mathbf{i}\ p_3 &\Leftrightarrow x_2y_3 + x_1y_2 + x_3y_1 - y_2x_3 - y_1x_2 - y_3x_1 = 0 \\
&\quad \wedge\ p_1\ p_2\ \mathbf{r}\ p_4\ \wedge\ p_4\ p_1\ \mathbf{r}\ p_3\ \wedge\ p_4\ p_2\ \mathbf{l}\ p_3 \\
p_1\ p_2\ \mathbf{e}\ p_3 &\Leftrightarrow x_3 = x_2\ \wedge\ y_3 = y_2\ \wedge\ x_3 \neq x_1\ \wedge\ y_3 \neq y_1 \\
p_1\ p_2\ \mathbf{f}\ p_3 &\Leftrightarrow x_2y_3 + x_1y_2 + x_3y_1 - y_2x_3 - y_1x_2 - y_3x_1 = 0 \\
&\quad \wedge\ p_1\ p_2\ \mathbf{r}\ p_4\ \wedge\ p_4\ p_2\ \mathbf{r}\ p_3 \\
p_1\ p_2\ \mathbf{d}\ p_3 &\Leftrightarrow x_1 = x_2\ \wedge\ y_1 = y_2\ \wedge\ x_1 \neq x_3\ \wedge\ y_1 \neq y_3 \\
p_1\ p_2\ \mathbf{t}\ p_3 &\Leftrightarrow x_1 = x_2 = x_3\ \wedge\ y_1 = y_2 = y_3,
\end{aligned}$$

Table 1: A correspondence table for the $\mathcal{LR}$ calculus.

to p_1 and p_2. The remaining two relations are: **d**ouble $:= \{(p_1, p_2, p_3) \mid p_1, p_2, p_3 \in \mathbb{R}^2, p_1 = p_2, p_1 \neq p_3\}$, **t**riple $:= \{(p_1, p_2, p_3) \mid p_1, p_2, p_3 \in \mathbb{R}^2, p_1 = p_2 = p_3\}$. By describing the relations using polynomial constraints, we obtain the correspondences in Table 1, where we introduce a new point p_4 when required. We note that an inequation "$\neq$" can be written as a disjunction of "$>$" and "$<$".

3.2 The $\mathcal{OPRA}_m$ calculus

The domain of the $\mathcal{OPRA}_m$ calculus [Moratz, 2006] 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 an orientation vector $\vec{o}_p := (v, w) - (x, y)$. 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 $\vec{o}_p$. 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 Figure 2). An $\mathcal{OPRA}_m$ relation is a binary relation which describes for points p_1 and p_2 their positions to each other with respect to the aforementioned partitioning. This is represented by the notation $p_1\ {}_m\angle_i^j\ p_2$, where m is as defined

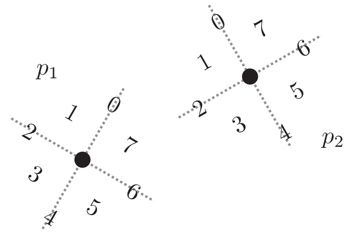

Figure 2: Illustration of $\mathcal{OPRA}_2$ relation $p_1\ {}_2\angle_7^2\ p_2$

before, i is number of the sector (or half-line) of p_1, in which p_2 is located, and j is the number of the sector (or half-line) of p_2, in which p_1 is located. We write $p_1\ {}_m\angle_=\ p_2$ if they share the same position.[1] 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

$$\begin{pmatrix} r_x(v,w,k) \\ r_y(v,w,k) \end{pmatrix} := \begin{pmatrix} \cos(k \cdot \frac{\pi}{m}) & -\sin(k \cdot \frac{\pi}{m}) \\ \sin(k \cdot \frac{\pi}{m}) & \cos(k \cdot \frac{\pi}{m}) \end{pmatrix} \begin{pmatrix} v \\ w \end{pmatrix} \qquad (2)$$

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

$$\begin{aligned} p_1\ {}_m\angle_i^*\ p_2 \ :\Leftrightarrow\ & \det \begin{pmatrix} 1 & x_1 & y_1 \\ 1 & r_x(v_1, w_1, \frac{i}{2}) & r_y(v_1, w_1, \frac{i}{2}) \\ 1 & x_2 & y_2 \end{pmatrix} = 0 \\ & \wedge\ \det \begin{pmatrix} 1 & x_1 & y_1 \\ 1 & r_x(v_1, w_1, \frac{i}{2}+1) & r_y(v_1, w_1, \frac{i}{2}+1) \\ 1 & x_2 & y_2 \end{pmatrix} < 0, \end{aligned}$$

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

$$\begin{aligned} p_1\ {}_m\angle_i^*\ p_2 \ :\Leftrightarrow\ & \det \begin{pmatrix} 1 & x_1 & y_1 \\ 1 & r_x(v_1, w_1, \frac{i-1}{2}) & r_y(v_1, w_1, \frac{i-1}{2}) \\ 1 & x_2 & y_2 \end{pmatrix} > 0 \\ & \wedge\ \det \begin{pmatrix} 1 & x_1 & y_1 \\ 1 & r_x(v_1, w_1, \frac{i+1}{2}) & r_y(v_1, w_1, \frac{i+1}{2}) \\ 1 & x_2 & y_2 \end{pmatrix} < 0, \end{aligned}$$

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

$$p_1\ {}_m\angle_i^j\ p_2 \quad \Leftrightarrow \quad p_1\ {}_m\angle_i^*\ p_2 \quad \wedge \quad p_2\ {}_m\angle_j^*\ p_1$$

and

$$p_1\ {}_m\angle_=\ p_2 \quad \Leftrightarrow \quad (x_1, y_1) = (x_2, y_2),$$

and we obtain the desired polynomial constraints.

The polynomial constraints from $\mathcal{OPRA}_m$ relations consist of quadratic polynomials with real algebraic numbers[2] as their coefficients. Dealing with real algebraic numbers requires more computing effort than with rational numbers. As the real algebraic numbers are resulted from $\cos(k\frac{\pi}{m})$ and $\sin(k\frac{\pi}{m})$ from (2) which are responsible for the positions of the half-lines, we can avoid real algebraic numbers by slightly modifying the definition for the positions of the half-lines so as to have only rational numbers as the coefficients.

3.3 The $\mathcal{STAR}_m$ calculus

The $\mathcal{STAR}_m$ calculus [Renz and Mitra, 2004] is similar to the $\mathcal{OPRA}_m$ calculus except it has a fixed reference direction. Consequently, for all oriented points $p = (x, y, v, w)$ the values for (v, w) are fixed to $v = x$, $w = y + 1$ to allow $\vec{o}_p = (v, w) - (x, y) = (0, 1)$ as the orientation for all points. This restriction on the expressibility of the representation has a computational advantage that the resulting polynomial constraints require less variables and they are linear and not quadratic. Hence, they can be solved more efficiently, for example, by the simplex method in subsection 4.1.

So far, we have seen the correspondences between qualitative spatial constraints and polynomial constraints from several spatial representations. Once we have these correspondences, deciding the consistency or finding a model of a set of constraints amounts to solving a system (i.e., a conjunction) of corresponding polynomial equations or inequalities. The approaches to this very problem is discussed in the following section.

4 Alternative Methods for Reasoning with Qualitative Spatial Knowledge

This section introduces methods for solving constraints coming from qualitative spatial relations. As seen in the preceding section, directional constraints can be translated to a system of polynomial equations and inequalities. If the polynomials in the system have degree at most 1 (i.e., the systems is linear), than the simplex method from linear programming can be applied. Otherwise, the Gröner base method from algebraic geometry, or the cylindrical algebraic decomposition method from real algebraic geometry can be applied to polynomial systems with arbitrary degrees.

4.1 The Simplex Method

Many mathematical optimization problems can be formulated as a *Linear Programming* [Dantzig and Thapa, 1997] problem, i.e., finding a maximum (or minimum) of a linear function subject to a set of constraints which is given by a system of linear inequalities. The *simplex method* is one of the techniques in linear programming that is widely used. The simplex method is divided in two phases. In Phase I, it searches for a feasible solution of the given linear system. If a solution is found, then the solution is used in Phase II to find an optimal solution. As our objective is solving a linear system and not optimization, only the algorithm for Phase I is relevant.

The simplex method is a sound and complete method, and has single exponential time complexity.

4.2 The Gröbner Base Method

Several methods have been developed to solve systems of multivariate polynomial equations over the complex field. Gröbner bases introduced by Buchberger [Buchberger, 1985] offer a computational approach that allows us to rewrite a set of polynomial equations, not altering their common zero set. In spirit, the approach of computing Gröbner bases is related Wu's method [Wu, 1978; 1986] as both methods determine elimination polynomials to rewrite polynomials by means of polynomial division. The rewriting process cancels variables and thus leads to equations that are easier to handle. Both elimination techniques are common foundations of algebraic approaches to geometric theorem proving. When computing the Gröbner basis a normalization step is usually carried out to

[1] The original paper [Moratz, 2006] introduces also the so-called *same relations* that further differentiate $p_1\ {}_m\angle_=\ p_2$ by the orientations of p_1 and p_2.

[2] A real algebraic number is a real number that is a root of a polynomial with integer coefficients (e.g., $\sqrt{2}$ as a root of x^2).

obtain the basis in normal form, called the *reduced Gröbner basis*. This form exhibits a remarkable feature: when the initial set of polynomials does not have a common solution, then the reduced Gröbner basis is equal to $\{1\}$. This property suggests that Gröbner basis enable a straight-forward approach to test the zero set for emptiness, but recall that polynomial equations can also involve complex roots. Henceforth, in cases where the reduced Gröbner basis does not equal $\{1\}$, a common solution is known to exist, but one still needs to check whether the common solution is real-valued. The approach of first computing the Gröbner basis and then further examining existence of real-valued solutions can handle problems arising when analyzing constraint calculi [Wolter, to appear], e.g., automatically computing the composition operation. However, this approach does not provide us with a complete decision procedure and it appears to be very difficult to turn it into a provenly complete one.

4.3 Cylindrical Algebraic Decomposition

The *Cylindrical Algebraic Decomposition* (CAD) [Collins, 1975; Arnon *et al.*, 1984] overcomes the deficiencies of the two previously introduced methods; compared to the simplex method, CAD can handle any polynomial systems and is not limited to linear systems, and where as the Gröber base method is not complete, CAD provides a complete algorithm.

Given a finite set of polynomials $f_1, \ldots, f_m$ in r variables with coefficients from $\mathbb{Q}$, the CAD algorithm computes a finite subset S of $\mathbb{R}^r$, such that

$$\begin{aligned}\{(\operatorname{sgn}(f_1(s)), \ldots, \operatorname{sgn}(f_n(s))) \mid s \in S\} \quad (3)\\ = \{(\operatorname{sgn}(f_1(x)), \ldots, \operatorname{sgn}(f_r(x))) \mid x \in \mathbb{R}^r\},\end{aligned}$$

where sgn is a real-valued function that returns the sign (i.e., -1, 0, or 1) of its argument. Thus, solving a system of polynomial equations and inequalities having $f_1, \ldots, f_m$ on the left-hand side of the system can be accomplished by evaluating $f_1, \ldots, f_m$ over the elements of S and checking their signs. Due to condition (3) this decision procedure is sound and complete. It also terminates as S is finite.

To generate the set of sample points S the CAD algorithm decomposes $\mathbb{R}^r$, the domain of variables $x_1, \ldots, x_r$, into finitely many subsets $C_1, \ldots, C_K$ of $\mathbb{R}^r$, such that each cell C_i is *sign-invariant* with respect to $f_1, \ldots, f_m$, meaning that the signs of $f_1, \ldots, f_m$ are constant when evaluated over C_i. Set S is then obtained by calculating a sample point in each of the cells $C_1, \ldots, C_K$.

The complexity of CAD is doubly exponential in the number r of the variables.

CAD is designed for general polynomial systems. As a consequence, it is not optimized for particular polynomial systems translated from qualitative spatial relations. For instance, the fact that most polynomial constraints coming from directional relations have their origins in the determinant expression in (1) is not deployed. This lack of integration results in the low performance of the CAD algorithm when dealing with qualitative spatial constraints. We observe in the evaluation of the computer algebra system Mathematica[3] in Figure 5 that CAD is not able to deal with more than 5 objects efficiently.

[3] http://www.wolfram.com/mathematica

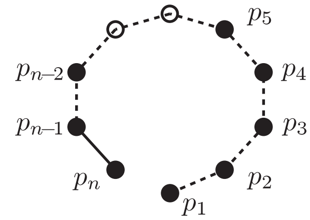

Figure 3: The benchmark problem LR-ALL-LEFT(n) consists of a set of $\mathcal{LR}$ constraints $\{p_i\ p_j\ \mathbf{l}\ p_k \mid 1 \leq i < j < k \leq n\}$ over n varibles, which are consistent by construction.

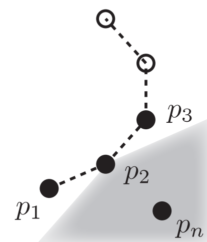

Figure 4: The benchmark problem LR-INDIAN-TENT(n) is a generalization of the Indian Tent Problem for four points (see [Wallgrün *et al.*, 2007]). The problem consists of the same set as LR-ALL-LEFT(n) except two constraints $p_1\ p_2\ \mathbf{l}\ p_n$ and $p_2\ p_3\ \mathbf{l}\ p_n$ are substituted with $p_1\ p_2\ \mathbf{r}\ p_n$ and $p_2\ p_3\ \mathbf{r}\ p_n$. These new two constraints contradict $p_1\ p_3\ \mathbf{l}\ p_n$, because they force p_n to be placed in the shaded region. Hence, LR-INDIAN-TENT(n) is inconsistent for all $n \geq 4$.

Accordingly, future research has to concentrate on the theoretical analysis of the interaction between the CAD algorithm and qualitative spatial constraints, and also on the tight integration thereof to achieve better performance.

5 Conclusions

In this paper we have discussed several approaches that propose themselves as alternatives to the conventional relation-algebraic method. From the three presented approaches the simplex method and CAD provide sound and complete algorithms, which are also constructive and are therefore able to generate models for consistent constraints. The simplex method, which runs faster than CAD, is well suited for qualitative spatial constraints that can be translated to a system of linear equations and inequalities (e.g., constraints from the $\mathcal{STAR}_m$ calculus). On the other hand, CAD is versatile, and can deal with any system of polynomial constraints. However, CAD suffers from its poor performance in solving qualitative spatial constraints, since it is a general solver and is therefore not tailored to these specific constraints. We see this deficiency of CAD as an open research question. To overcome this issue, a thorough analysis of the input polynomials is needed in the future. Analyzing the determinant expression (1) and adapting the result to the CAD algorithm might be a key to the improvement of this approach.

In summary, there is a need to adopt the mentioned new

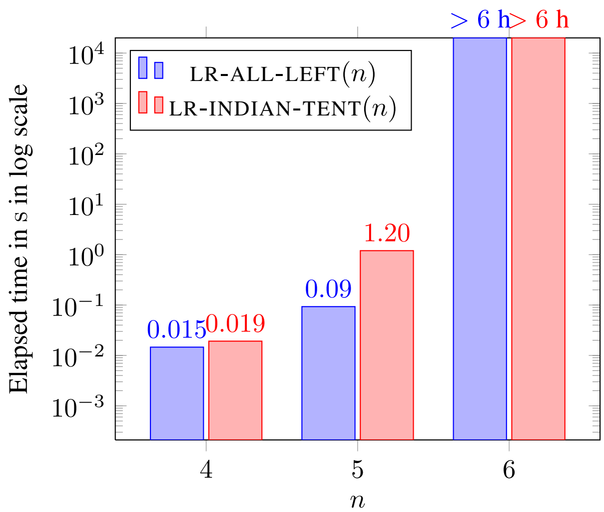

Figure 5: Evaluation of Mathematica™ ver. 8.0.1.0 with benchmark problems LR-ALL-LEFT(n) (see Figure 3) and LR-INDIAN-TENT(n) (see Figure 4) using the function `FindInstance`. Although Mathematica finds consistent instances for LR-ALL-LEFT(4) and LR-ALL-LEFT(5), and inconsistencies of LR-INDIAN-TENT(4) and LR-INDIAN-TENT(5) in less than few seconds, it was not able to decide consistency of LR-ALL-LEFT(6) and inconsistency of LR-INDIAN-TENT(6) within 6 hours. The evaluation was done on an OS X machine with Intel Core 2 Duo 2.66 GHz processor and 4 GB memory.

approaches for reasoning with qualitative spatial information. The future research in qualitative spatial reasoning should therefore consider—besides investigating qualitative spatial representations with regard to their relation-algebraic properties—analyzing and optimizing the introduced new approaches by exploiting the structure of polynomials from qualitative spatial constraints.

Acknowledgements

This work was supported by the DFG Transregional Collaborative Research Center SFB/TR 8 Spatial Cognition. Funding by the German Research Foundation (DFG) is gratefully acknowledged.

References

[Arnon *et al.*, 1984] Dennis S. Arnon, George E. Collins, and Scott McCallum. Cylindrical algebraic decomposition i: the basic algorithm. *SIAM J. Comput.*, 13:865–877, November 1984.

[Buchberger, 1985] Benno Buchberger. Göbner-bases: An algorithmic method in polynomial ideal theory. In N.K. Bose, editor, *Multidimensional Systems Theory - Progress, Directions and Open Problems in Multidimensional Systems*, chapter 6, pages 184–232. Reidel Publishing Company, Dordrecht—Boston—Lancaster, 1985.

[Cohn and Renz, 2008] Anthony G. Cohn and Jochen Renz. Chapter 13 qualitative spatial representation and reasoning. In Vladimir Lifschitz Frank van Harmelen and Bruce Porter, editors, *Handbook of Knowledge Representation*, volume 3 of *Foundations of Artificial Intelligence*, pages 551 – 596. Elsevier, 2008.

[Collins, 1975] George Collins. Quantifier elimination for real closed fields by cylindrical algebraic decompostion. In H. Brakhage, editor, *Automata Theory and Formal Languages 2nd GI Conference Kaiserslautern, May 20–23, 1975*, volume 33 of *Lecture Notes in Computer Science*, pages 134–183. Springer Berlin / Heidelberg, 1975.

[Dantzig and Thapa, 1997] George B. Dantzig and Mukund N. Thapa. *Linear programming 1: introduction*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1997.

[Ligozat, 2005] Gérard Ligozat. Categorical methods in qualitative reasoning: The case for weak representations. In Anthony Cohn and David Mark, editors, *Spatial Information Theory*, volume 3693 of *Lecture Notes in Computer Science*, pages 265–282. Springer Berlin / Heidelberg, 2005.

[Moratz, 2006] Reinhard 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*, pages 407–411, Amsterdam, The Netherlands, The Netherlands, 2006. IOS Press.

[Renz and Ligozat, 2005] Jochen Renz and Gérard Ligozat. Weak composition for qualitative spatial and temporal reasoning. In Peter van Beek, editor, *Principles and Practice of Constraint Programming - CP 2005*, volume 3709 of *Lecture Notes in Computer Science*, pages 534–548. Springer Berlin / Heidelberg, 2005.

[Renz and Mitra, 2004] Jochen Renz and Debasis Mitra. Qualitative direction calculi with arbitrary granularity. In Chengqi Zhang, Hans W. Guesgen, and Wai-Kiang Yeap, editors, *PRICAI 2004: Trends in Artificial Intelligence*, volume 3157 of *Lecture Notes in Computer Science*, pages 65–74. Springer Berlin / Heidelberg, 2004.

[Renz and Nebel, 2007] Jochen Renz and Bernhard Nebel. Qualitative spatial reasoning using constraint calculi. In Marco Aiello, Ian Pratt-Hartmann, and Johan Benthem, editors, *Handbook of Spatial Logics*, pages 161–215. Springer Netherlands, 2007.

[Scivos and Nebel, 2005] Alexander Scivos and Bernhard Nebel. The finest of its class: The natural point-based ternary calculus for qualitative spatial reasoning. In Christian Freksa, Markus Knauff, Bernd Krieg-Brückner, Bernhard Nebel, and Thomas Barkowsky, editors, *Spatial Cognition IV. Reasoning, Action, and Interaction*, volume 3343 of *Lecture Notes in Computer Science*, pages 283–303. Springer Berlin / Heidelberg, 2005.

[Wallgrün *et al.*, 2007] Jan Wallgrün, Lutz Frommberger, Diedrich Wolter, Frank Dylla, and Christian Freksa. Qualitative spatial representation and reasoning in the sparq-toolbox. In Thomas Barkowsky, Markus Knauff, Gérard

Ligozat, and Daniel Montello, editors, *Spatial Cognition V Reasoning, Action, Interaction*, volume 4387 of *Lecture Notes in Computer Science*, pages 39–58. Springer Berlin / Heidelberg, 2007.

[Wolter and Lee, 2010] D. Wolter and J.H. Lee. Qualitative reasoning with directional relations. *Artificial Intelligence*, 174(18):1498 – 1507, 2010.

[Wolter, to appear] Diedrich Wolter. Analyzing qualitative spatio-temporal calculi using algebraic geometry. *Spatial Cognition and Computation*, to appear.

[Wu, 1978] Wen-Tsun Wu. On the decision problem and the mechanization of theorem proving in elementary geometry. *Scientia Sinica*, 21:157–179, 1978.

[Wu, 1986] Wen-Tsun Wu. Basic principles of mechanical theorem proving in elementary geometries. *Journal of Automated Reasoning*, 2:221–252, 1986.