

A much better polynomial time approximation of consistency in the $\mathcal{LR}$ calculus

Dominik Lücke¹ and Till Mossakowski²

Abstract. In the area of qualitative spatial reasoning, the $\mathcal{LR}$ calculus is a quite simple constraint calculus that forms the core of several orientation calculi like the Dipole calculi and the $\mathcal{OPRA}_1$ calculus by introducing the left-right-dichotomy.

For many qualitative spatial calculi, algebraic closure is applied as the standard polynomial time “decision” procedure. For a long time it was believed that this can decide the consistency of scenarios of the quite simple and basic $\mathcal{LR}$ calculus (a refinement of Ligozat’s flip-flop calculus). However, in [8] it was shown that algebraic closure is a quite bad approximation of consistency of $\mathcal{LR}$ scenarios: scenarios in the base relations “Left” and “Right” are always algebraically closed, no matter if those scenarios are consistent or not. So algebraic closure is completely useless here. Furthermore, in [15] it was proved that the consistency problem for any calculus with relative orientation containing the relations “Left” and “Right” is NP-hard.

In this paper we propose a new and better polynomial time approximation procedure for this NP-hard problem. It is based on the angles of triangles in the Euclidean plane. $\mathcal{LR}$ scenarios are translated to sets of linear inequalities over the real numbers. We evaluate the quality of this procedure by comparing it both to the old approximation using algebraic closure and to the (exact but exponential time) Buchberger algorithm for Gröbner bases (used as a decision method).

1 Introduction

Since the work of Allen [1] on temporal intervals, constraint calculi have been used to model a variety of aspects of space and time in a way that is both qualitative (and thus closer to natural language than quantitative representations) and computationally efficient (by appropriately restricting the vocabulary of rich mathematical theories about space and time). For example, the well-known region connection calculus [9] allows for reasoning about regions in space. Applications include geographic information systems, human-machine interaction, and robot navigation.

Directions in the (Euclidean) plane are described by *orientation* calculi. These directions can be given with respect to a global reference frame like for the Cardinal directions algebra. The Cardinal directions calculus features the (absolute) orientations North, South, East, and West. On the other hand a

(relative) reference frame can be given locally at any object of the calculus, like e.g. for the Flip-Flop calculus [7], where the reference frame is given by an oriented line from a start to an end point. The $\mathcal{LR}$ -calculus [13] is a refinement of Flip-Flop curing its shortcomings.

These *Relative orientation* calculi are quite important examples of qualitative calculi, because relative orientation is very natural in many real world applications. Just consider the situation that you want to describe the way to some point of interest in your home-town. You will tell your addressee to turn left or right at some crossing with respect to their current orientation. You will neither tell them to change the direction to e.g. North, nor will you tell them to turn by a certain amount of degrees. If you describe the layout of some point of interest, you will most certainly use the same qualitative relations either. But how can we make a machine, e.g. a robot, decide efficiently that a qualitative description is consistent?

This is where the reasoning facilities of the qualitative calculi are needed. Efficient reasoning in them mainly relies on the *algebraic closure* algorithm. Using relational composition and converse, it refines (basic) constraint networks in polynomial time. If algebraic closure detects an inconsistency, the original network is surely inconsistent. If no inconsistency is detected, for some calculi, this implies consistency of the original network — but not for all calculi. For the cardinal direction calculus, it can be easily shown that algebraic closure indeed decides consistency for scenarios. However, in [8] we have shown that the consistency of $\mathcal{LR}$ -scenarios cannot be decided by applying the algebraic closure method. This result naturally carries over to the Flip-Flop calculus. To cure the lack of a decision procedure, in [8] we have tried a factorization method along globally consistent 7-point scenarios. However, there we admit that global consistency is not the same as consistency; indeed, it turns out to be a notion that is too strong.

Wolter and Lee [15] give a new explanation of this phenomenon: through a reduction to oriented matroids, they prove that deciding consistency for scenarios in the $\mathcal{LR}$ calculus is NP-hard. Assuming the generally believed hypothesis that $P \neq NP$, this explains why algebraic closure (a polynomial procedure) does not decide consistency of scenarios: all what we can hope for are *polynomial approximations* of consistency. Here, we develop a polynomial approximation that improves on algebraic closure. Of course, we cannot do this by staying at the level of abstract relation algebras (where algebraic closure operates). Rather, we have to take properties of the “natural” domain of the calculus at hand into account. Our

¹ SFB/TR 8 Spatial Cognition, Bremen, Germany, email: luecke@informatik.uni-bremen.de

² DFKI GmbH Bremen, Germany, email: Till.Mossakowski@dfki.de

new approach to approximate the consistency of $\mathcal{LR}$ -scenarios is based on the properties of triangles in the Euclidean plane, which are formalized algebraically as a set of linear inequalities. Our notion of *triangle consistency* can be decided in polynomial time; this essentially follows from the solvability of the corresponding inequalities in polynomial time [2]. First experiments showed that triangle consistency approximates consistency of $\mathcal{LR}$ scenarios much better than algebraic closure does. We focus on comparing our approach to algebraic closure, since tool-support is available for that method, which is still lacking for e.g. neighborhood-based reasoning. Further experiments gave some evidence that the approach is quite well suited for approximating the consistency of $\mathcal{LR}$ scenarios. This approximation procedure also naturally applies to the $\mathcal{DRA}$ -calculi, another family of calculi dealing with relative orientation, as well as to $\mathcal{OPRA}_1$ ³.

2 Qualitative Calculi

Qualitative calculi are employed for representing knowledge about a domain using a finite set of labels, so-called base relations. Base relations partition the domain into discrete parts. One example is distinguishing points on the time line by binary relations such as “before”, “after” and “same”. A qualitative representation only captures membership of domain objects in these parts. For example, it can be represented that time point A occurs before B , but not how much earlier nor at which absolute time. Thus, a qualitative representation abstracts, which is particularly helpful when dealing with infinite domains like time and space that possess an internal structure like for example $\mathbb{R}^n$.

In order to ensure that any constellation of domain objects is captured by exactly one qualitative relation, a special property is commonly required:

Definition 1. Let $\mathcal{B} = \{B_1, \dots, B_k\}$ be a set of n -ary relations over a domain $\mathcal{D}$. These relations are said to be *jointly exhaustive and pairwise disjoint (JEPD)*, if they satisfy the properties

1. $\forall i, j \in \{1, \dots, k\}$ with $i \neq j$: $B_i \cap B_j = \emptyset$
2. $\mathcal{D}^n = \bigcup_{i \in \{1, \dots, k\}} B_i$

For representing uncertain knowledge within a qualitative calculus, e.g., to represent that objects $x_1, x_2, \dots, x_n$ are either related by relation B_i or by relation B_j , *general relations* are introduced.

Definition 2. Let $\mathcal{B} = \{B_1, \dots, B_k\}$ be a set of n -ary relations over a domain $\mathcal{D}$. The set of *general relations* $\mathcal{R}_{\mathcal{B}}$ (or simply $\mathcal{R}$) is the powerset $\mathcal{P}(\mathcal{B})$. The semantics of a relation $R \in \mathcal{R}_{\mathcal{B}}$ is defined as follows:

$$R(x_1, \dots, x_n) :\Leftrightarrow \exists B_i \in R. B_i(x_1, \dots, x_n)$$

In a set of base relations that is JEPD, the empty relation $\emptyset \in \mathcal{R}_{\mathcal{B}}$ is called the *impossible relation*. Reasoning with qualitative information takes place on the symbolic level of relations $\mathcal{R}$, so we need special operators that allow us to manipulate qualitative knowledge. These operators constitute the algebraic structure of a qualitative calculus.

³ The $\mathcal{OPRA}_n$ calculi with $n > 1$ encode more information than just the left-right-dichotomy and some information about the position of points on a line and therefore need more refined decision procedures.

2.1 Algebraic Structure of Qualitative Calculi

The most fundamental operators in a qualitative calculus are those for relating qualitative relations in accordance to their set-theoretic disjunctive semantics. So, for $R, S \in \mathcal{R}$, intersection ($\cap$) and union ($\cup$) are defined canonically. The set of general relations is closed under these operators. Set-theoretic operators are independent of the calculus at hand, further operators are defined using the calculus semantics.

Qualitative calculi need to provide operators for interrelating relations that are declared to hold for the same set of objects but differ in the order of arguments. Put differently, we need operators which allow us to change perspective. For binary relations this operation is just forming the converse relation.

Definition 3. Permutation operators for ternary calculi:

$$\begin{aligned} INV(R) &:= \{ (y, x, z) \mid (x, y, z) \in R \} && \text{(inverse)} \\ SC(R) &:= \{ (x, z, y) \mid (x, y, z) \in R \} && \text{(shortcut)} \\ SCI(R) &:= \{ (z, x, y) \mid (x, y, z) \in R \} && \text{(inverse shortcut)} \\ HM(R) &:= \{ (y, z, x) \mid (x, y, z) \in R \} && \text{(homing)} \\ HMI(R) &:= \{ (z, y, x) \mid (x, y, z) \in R \} && \text{(inverse homing)} \end{aligned}$$

If the set of relations is closed under INV , SC and HM , then SCI and HMI can be defined as compositions of the respective operations (ref. to [3]). A restriction to few operations particularly eases definition of higher arity calculi.

Definition 4 ([3]). Let $R_1, R_2, \dots, R_n \in \mathcal{R}_{\mathcal{B}}$ be a sequence of n general relations in an n -ary qualitative calculus over the domain $\mathcal{D}$. Then the operation

$$\begin{aligned} \circ(R_1, \dots, R_n) &:= \{ (x_1, \dots, x_n) \in \mathcal{D}^n \mid \exists u \in \mathcal{D}, \\ &\quad (x_1, \dots, x_{n-1}, u) \in R_1, (x_1, \dots, x_{n-2}, u, x_n) \in R_2, \\ &\quad \dots, (u, x_2, \dots, x_n) \in R_n \} \end{aligned}$$

is called *n -ary composition*.

Note that for $n = 2$ one obtains the classical composition operation for binary calculi (cp. [11]) which is usually noted as infix operator. Nevertheless different kinds of binary compositions have been used for ternary calculi, too, as e.g. for the $\mathcal{LR}$ -calculus.

2.2 Strong and Weak Operations

Permutation and composition operators define relations. Per se it is unclear whether the relations obtained by application of an operation are expressible in the calculus, i.e. whether the set of general relations $\mathcal{R}_{\mathcal{B}}$ is closed under an operation. Indeed, for some calculi the set of relations is not closed, there even exist calculi for which no closed set of finite size can exist, e.g. the composition operation in Freksa’s double cross calculus [12].

Definition 5. Let an n -ary qualitative calculus with relations $\mathcal{R}_{\mathcal{B}}$ over domain $\mathcal{D}$ and an m -ary operation $\phi : \mathcal{B}^m \rightarrow \mathcal{P}(\mathcal{D}^n)$ be given. If the set of relations is closed under ϕ , i.e. for $\forall \vec{B} \in \mathcal{B}^m \exists R' \in \mathcal{R}_{\mathcal{B}} : \phi(\vec{B}) = \bigcup_{B \in R'} B$, then the operation ϕ is called *strong*.

In qualitative reasoning we must restrict ourselves to a finite set of relations. Therefore, if some operation is not strong in the sense of Def. 5, an upper approximation of the true operation is used instead.

Definition 6. Given a qualitative calculus with n -ary relations $\mathcal{R}_{\mathcal{B}}$ over domain $\mathcal{D}$ and an operation $\phi : \mathcal{B}^m \rightarrow \mathcal{P}(\mathcal{D}^n)$, then the operator

$$\begin{aligned} \phi^* : \mathcal{B}^m &\rightarrow \mathcal{R}_{\mathcal{B}} \\ \phi^*(B_1, \dots, B_k) &:= \{R \in \mathcal{B} \mid R \cap \phi(B_1, \dots, B_k) \neq \emptyset\} \end{aligned}$$

is called a *weak* operation, namely the weak approximation of ϕ .

Note that the weak approximation of an operation is identical to the original operation if and only if the original operation is strong. Further note that any calculus is closed under weak operations. Applying weak operations can lead to a loss of information which may be critical in certain reasoning processes. In the literature the weak composition operation is usually denoted by $\diamond$.

Definition 7. We call an m -ary relation R over $\mathbb{R}^n$ convex, if $\{y \mid R(x_1, \dots, x_{m-1}, y), (x_1, \dots, x_{m-1}, y) \in \mathbb{R}^n\}$ is a convex subset of $\mathbb{R}^n$.

3 Constraint Based Qualitative Reasoning

Qualitative reasoning is concerned with solving constraint satisfaction problems (CSPs) in which constraints are expressed using relations of the calculus. Definitions from the field of CSP are carried over to qualitative reasoning (cp. [4]).

Definition 8. Let $\mathcal{R}$ be the general relations of a qualitative calculus over the domain $\mathcal{D}$. A *qualitative constraint* is a formula $R(X_1, \dots, X_n)$ (also written $X_1 \dots X_{n-1} R X_n$) with variables X_i taking values from the domain and $R \in \mathcal{R}$. A *constraint network* is a set of constraints. A constraint network is said to be a *scenario* if it gives base relations for all relations $R(X_1, \dots, X_n)$ and the base relations obtained for different permutations of variables $X_1, \dots, X_n$ must be agreeable wrt. the permutation operations.

One key problem is to decide whether a given CSP has a solution or not. This can be a very hard problem. Infinity of the domain underlying qualitative CSPs inhibits searching for an agreeable valuation of the variables.

Definition 9. A constraint network is called *consistent* if a valuation of all variables exists, such that all constraints are fulfilled. A constraint network is called *n -consistent* ($n \in \mathbb{N}$) if every solution for $n-1$ variables can be extended to a solution with n variables involving any further variable. A constraint network is called *strongly n -consistent*, if it is m -consistent for all $m \leq n$. A CSP in n -variables is *globally consistent*, if it is strongly n -consistent.

A fundamental technique for deciding consistency in a classical CSP is to enforce k -consistency by restricting the domain of variables in the CSP to mutually agreeable values. Backtracking search can then identify a consistent variable assignment. If the domain of some variable gets restricted to down to zero size while enforcing k -consistency, the CSP is

not consistent. This procedure except for backtracking search (which is not applicable in infinite domains) is also applied to qualitative CSPs [11]. For a JEPD calculus with n -ary relations any qualitative CSP is strongly n -consistent unless it contains a constraint with the empty relation. So the first step in checking consistency would be to test $n+1$ -consistency. In the case of a calculus with binary relations this would mean analyzing 3-consistency, also called *path-consistency*. This is the aim of the algebraic closure algorithm which exploits that composition lists all 3-consistent scenarios.

Definition 10. A CSP over binary relations is called *algebraically closed* if for all variables X_1, X_2, X_3 and all relations R_1, R_2, R_3 the constraint relations

$$R_1(X_1, X_2), \quad R_2(X_2, X_3), \quad R_3(X_1, X_3)$$

imply $R_3 \subseteq R_1 \diamond R_2$. To enforce algebraic closure, the operation $R_3 := R_3 \cap R_1 \diamond R_2$ (as well as a similar operation for converses) is applied for all variables until a fixed-point is reached.

Enforcing algebraic closure preserves consistency, i.e., if the empty relation is obtained during refinement, then the qualitative CSP is inconsistent. However, algebraic closure does not mandatorily decide consistency: a CSP may be algebraically closed but inconsistent — even if composition is strong [10].

Algebraic closure has also been adapted to ternary calculi using binary composition [6]. Binary composition of ternary relations involves 4 variables, it may not be able to represent all 4-consistent scenarios though. Scenarios with 4 variables are specified by 4 ternary relations. However, binary composition $R_1 \diamond R_2 = R_3$ only involves 3 ternary relations. Therefore, using n -ary composition in reasoning with n -ary relations is more natural (cp. [3]).

4 The $\mathcal{LR}$ -calculus

In this section we introduce the $\mathcal{LR}$ -calculus [13], a coarse relative orientation calculus. We use it as our starting point to develop new decision procedures for relative orientation calculi. This calculus defines nine base relations which are depicted in Fig. 1. The $\mathcal{LR}$ -calculus deals with the relative position of a point C with respect to the oriented line from point A to point B , if $A \neq B$. The point C can be to the left of (l), to the right of (r) the line, or it can be on a line collinear to the given one and in front of (f) B , between A and B with the relation (i) or behind (b) A ; furthermore, it can be on the start-point A (s) or on the end-point B (e). If $A = B$, then we can distinguish between the relations *Tri*, expressing that $A = C$ and *Dou*, meaning $A \neq C$. Freksa's double cross calculus *DCC* is a refinement of the $\mathcal{LR}$ -calculus and, henceforth, our findings for the $\mathcal{LR}$ -calculus can be directly applied to the *DCC*-calculus as well.

5 Algebraic Closure is no Decision Procedure for $\mathcal{LR}$

In [8] we have shown that algebraic closure does not decide consistency for the $\mathcal{LR}$ -calculus, i.e., not every algebraically closed scenario is consistent. We recall the most crucial results, for the proofs refer to the particular paper. The most staggering result is:

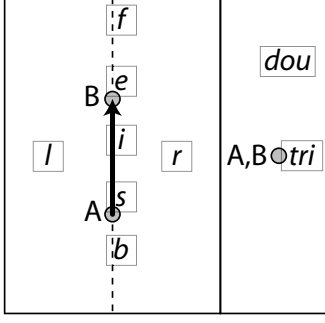


Figure 1. The nine base relations of the $\mathcal{LR}$ -calculus; tri designates the case of $A = B = C$, whereas dou stands for $A = B \neq C$.

Proposition 11. *All scenarios only containing the relations l and r are algebraically closed wrt. the $\mathcal{LR}$ -calculus with binary composition, as soon as they agree on the permutation operations.*

This result almost disqualifies algebraic closure as a decision procedure for $\mathcal{LR}$. Moreover, neither classical binary nor for ternary algebraic closure can decide consistency for $\mathcal{LR}$.

Theorem 12. *Classical algebraic closure does not enforce scenario consistency for the $\mathcal{LR}$ -calculus.*

Proof. In [8] the scenario

$$\begin{aligned} \text{SCEN} := & \{(AB \ r \ C), (AE \ r \ D), (DB \ r \ A), \\ & (DC \ r \ A), (DC \ r \ B), (DE \ r \ B), \\ & (DE \ l \ C), (EB \ r \ A), (EC \ r \ A), \\ & (EC \ r \ B)\} \end{aligned}$$

was used, which is algebraically closed wrt. the $\mathcal{LR}$ -composition (binary and ternary) but not realizable in the domain, to show this theorem. $\square$

Theorem 13. *Algebraic closure wrt. ternary composition does not enforce scenario consistency for the $\mathcal{LR}$ -calculus.*

A next approach to get a decision procedure was to stick to global consistency, as e.g. used in [4], and to factorize scenarios along certain globally consistent sub-scenarios. This approach turned out to be fruitless. Prop. 15 shows that that approach is a mere approximation, too.

Proposition 14. *For CSPs over convex $\{\mathcal{LR}, \text{DCC}\}$ -relations strong 7-consistency decides global consistency.*

Proposition 15. *For the $\mathcal{LR}$ -calculus not every consistent scenario is globally consistent.*

6 Triangle Consistency: A New Approximation of Consistency for $\mathcal{LR}$

Our approach is inspired by the simple observation that if given n points in the Euclidean plane and all $\frac{n \cdot (n-1)}{2}$ undirected lines connecting each point with every other point in such a configuration, the connecting lines between arbitrary 3 points form a (possibly degenerated, if all points lie on the

same line) triangle. Because of that for any such triple of points and their connecting lines, all well-known properties of triangles need to be fulfilled. “Left” and “Right” can be distinguished by the orientation (read sectors of a circle) of the involved angles (ref. to. Fig. 2). We currently restrict ourselves to the interesting cases of the relations l and r for which algebraic closure performs very badly, as we have seen in section 5. This means that we will not have to deal with any degenerate triangles. Our approach works for all base-relations of the $\mathcal{LR}$ -calculus, but the restriction to the “interesting” cases is necessary due to space limitations. The extension to all base-relations is performed by just adding inequalities for the properties of degenerate triangles in a straightforward way.

In our new approach, we translate any $\mathcal{LR}$ -scenario (which has to contain base relations between all permutation of all triples of distinct points) into a set of inequalities over triangles in the Euclidean plane. We normalize all angles to the interval $(-\pi, \pi)$ to simplify later calculations. Let 3 points A , B and C in relation $(AB \ r \ C)$ be given. For such a relation, we get a scenario in the plane as in Fig 2. We can derive that

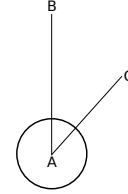


Figure 2. $(AB \ r \ C)$ in the plane

the angle from C to B at A , which we call CAB , is in the open interval $(0, \pi)$. The derivation for $(AB \ l \ C)$ is similar and yields angles in the interval $(-\pi, 0)$. Further, by a simple geometrical argument, we can show, that:

Lemma 16. *For a non-degenerate triangle in the Euclidean plane with points A , B , C , if any of the angles BAC , ACB and CBA is in the interval $(0, \pi)$, so are all of the others. The same is true for angles from the interval $(-\pi, 0)$.*

With this and the well-known properties of triangles in the plane, we can derive our system of inequalities $\text{INEQNB}(BAC)$ for any arbitrary angle BAC which is depicted in Fig. 3.

To the inequalities $\text{INEQNB}(BAC)$, we add the ones derived from lemma 16

$$0 < BAC < \pi \Leftrightarrow 0 < ACB < \pi \Leftrightarrow 0 < CAB < \pi.$$

With them, we obtain the set $\text{INEQN}(BAC)$ for any angle BAC . Such sets of inequalities are generated for each triple of points in an $\mathcal{LR}$ -scenario. By INEQN we denote the set of all inequalities for such a scenario.

Definition 17. We call an $\mathcal{LR}$ -scenario *triangle consistent*, if there is at least one solution for all of its inequalities INEQN .

The “compression” of knowledge is done at this point by not considering any lengths of lines. Considering them would yield non-linear inequalities/equations that cannot be solved efficiently. We want to use as little knowledge in this approach as possible to make it computationally efficient:

Distinction l/r	
$0 < BAC < \pi$	if $(AB \mid C)$
$-\pi < BAC < 0$	if $(AB \mid r C)$
Opposite angles	
$BAC = -CAB$	
Sum of angles	
$BAC + CBA + ACB = \pi$	if $(AB \mid C)$
$BAC + CBA + ACB = -\pi$	if $(AB \mid r C)$
$BAC + CAD = BAD + 2 \cdot \pi$	if $\begin{cases} (AB \mid C) \\ (AC \mid D) \\ (AB \mid r D) \end{cases}$
$BAC + CAD = BAD - 2 \cdot \pi$	if $\begin{cases} (AB \mid r C) \\ (AC \mid r D) \\ (AB \mid D) \end{cases}$
$BAC + CAD = BAD$	otherwise

Figure 3. $\text{INEQNB}(BAC)$

Theorem 18. *Systems of linear inequalities can be decided in polynomial time.*

Proof. This follows from [2]. $\square$

Triangle consistency yields a decision procedure consisting of just two steps:

1. Translate the $\mathcal{LR}$ -scenario to a system of linear inequalities (this is just a substitution in the number of relations contained in the scenario and can clearly be performed in linear time),
2. Check the solvability of the system with a standard polynomial algorithm. (In fact, also the **simplex** can be applied in this step, it has exponential worst-case running time, but often performs very well.)

With Thm. 18 we obtain:

Proposition 19. *Triangle consistency has polynomial running time.*

Each geometric realization of an $\mathcal{LR}$ scenario obviously leads to a system of angles for the involved point triples; it is easy to show that such a system of angles is a solution for the inequalities INEQN . We thus arrive at:

Proposition 20. *Consistency implies triangle consistency.*

By Prop. 19 and the fact that deciding consistency is NP-hard, we obtain that under the assumption $P \neq NP$, the converse implication (triangle consistency implies consistency) does not hold. However, we have not found a counterexample yet.

7 Experiments

We have conducted intensive experiments evaluating the quality of our approach of triangle consistency. A big issue about doing experiments with relative orientation calculi is the infinite size of their domain. It is basically impossible to list all possible quantitative positions of points in the Euclidean plane. But for restricted numbers of points an upper bound for the number of qualitatively different configurations can be derived. This still leaves the problem that for a certain number of points (6 in our case) the memory consumption

of a program listing those different configurations becomes excessive, as well as the running time.

We have implemented a prototype solver in Haskell for deciding whether a given $\mathcal{LR}$ -scenario is triangle consistent. This tool can generate all complete $\mathcal{LR}$ -scenarios in n -points and calculate the corresponding set of inequalities INEQN . As the reasoning engine this tool currently uses the Yices SMT-solver [5]. We decided to use an external possibly not completely optimal reasoning engine for the prototype to overcome the issue of programming bugs and intensive debugging. However, the performance of the actual equation solver dominates the running time. Further, we have written a Haskell program that enumerates n -point $\mathcal{LR}$ -scenarios using a grid of $m \times m$ points. This program starts with a specified value of m_0 and calculates all possible $\mathcal{LR}$ -scenarios, then it increases the current bound and calculates again. This is continued until the user requests to terminate it. If the list of scenarios from run l and $l + 1$ differ, the new list of scenarios is displayed, otherwise a message, that the scenarios are the same. To list all 5 point scenarios completely, it turned out, that a grid of 8×8 points already is sufficient, which can be shown by an involved geometric argument. Unfortunately, for scenarios in 6 points we neither had the computation power nor the memory to enumerate all possible configurations. With 5 points, we can define one point being the intersection point by the lines given by the 4 other points as a worst-case scenario for our grid-method. But with 6 points, we can already define 2 points as being such intersection points. We conjecture that this fact has a bad impact on the required grid-size.

We also have used the Gröbner reasoner that is available in **SparQ** [14]. It led to the same set of consistent scenarios. However, the exponential runtime of both the grid method and the Gröbner reasoner prevented a computation of all consistent 6 point scenarios. For algebraic closure, we used the tool **SparQ** [14]. To examine scenarios with more than 5 points, we implemented a Haskell program that constructs algebraically closed scenarios in up to n points (where n is a given parameter). We let the program run for several days with the parameter n set up to 9 (examining thousands of scenarios). Unfortunately in many cases the Gröbner reasoner was not able to decide consistency, but if it was it gave the same answer as triangle consistency. Candidates for failure of the triangle consistency remained problems where the Gröbner reasoner could not decide consistency, but triangle consistency was fulfilled. We double checked dozens of such scenarios by hand, and could deem all of them being consistent.

Since we were able to enumerate all consistent all 5-point scenarios in the relations l and r^4 , we can quantify how well our method performs on them. We could identify 1955 consistent $\mathcal{LR}$ -scenarios (s_{con}) of this kind, algebraic closure with binary composition yields 3095 scenarios (s_{bin}) while ternary algebraic closure leads to 2355 scenarios (s_{ter}). We found exactly 1955 triangle consistent scenarios (s_{Δ}), and when inspecting them, we found out that $s_{con} = s_{\Delta}$. These numbers are depicted in Fig. 4. Please note that the numbers of the scenarios are given modulo swapping of the names of the

⁴ Smaller scenarios are not really interesting, since we can already detect inconsistencies with algebraic closure with ternary composition for 4 point scenarios, all smaller ones do not need additional consideration, since they are at most base relations. The 14 scenarios in 4 points are detected by our method.

points. As noted above, we could not identify all consistent 6

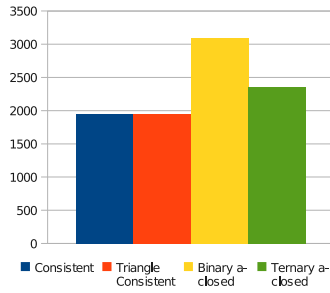


Figure 4. Comparison of algebraic closure with triangle consistency

point scenarios, mainly because of hardware restrictions. But we were still able to identify a subset of consistent $\mathcal{LR}$ -scenarios. So far we were able to classify 429449 scenarios with our approach, of which 4698 are deemed triangle consistent. Not all of them are in the list of our pre-calculated scenarios, but we have to remember that that list is incomplete. We took samples from the list of triangle consistent scenarios that were not in the pre-calculated list, and it turned out that we could find a realization for all of them. Many of those samples had a cloud of points lying close together, with at least one point lying very far away from the others. This is a case that could not be found with our grid method in limits that yield a feasible computation time.

Algebraic closure (binary as well as ternary) failed badly when we had scenarios with one more point than the number of points captured by the composition table; this is fortunately not true for triangle consistency.

All of these experimental results imply that triangle consistency approximates better than binary as well as ternary algebraic closure. Compared to Tarski's quantifier elimination and Gröbner Reasoning, it has a much better running time, since it is a polynomial time algorithm. Since we are only using a prototype implementation of our algorithm and since the used Gröbner reasoner can detect patterns in the equations that imply that it cannot decide consistency, quantifying the running time of the programs is in vain. Indeed, the Gröbner reasoner integrated into *SparQ* often fails to determine consistency for scenarios if the number of points grows. Sometimes it even gives up for quite small scenarios. For scenarios in more than nine or ten points the Gröbner reasoner gives up more often than it can decide. By contrast, triangle consistency works well with scenarios consisting of dozens of points, which is a size that is quite realistic for e.g. robot navigation applications.

8 Conclusion/Outlook

We have approximated the NP-hard problem of deciding consistency of scenarios in the $\mathcal{LR}$ calculus. Algebraic closure is a tool that has severe limits: for the $\mathcal{LR}$ -calculus it approximates consistency in its intended domain (the Euclidean plane) only quite badly. Information provided by the relations of the relative orientation calculi can be described algebraically by non-linear inequalities. Such systems of non-linear inequalities over $\mathbb{R}$ can of course be decided by Tarski's

quantifier elimination. Gröbner reasoning is another applicable procedure, but the running time of both of them is exponential and far too slow for many real world applications, e.g. in mobile robots. In fact, we need a reasoning procedure that is computationally feasible and as accurate as possible. In designing such a procedure, the incorporation of properties of the domain of the calculus at hand for simplifying the problem seems to be a sane way to go. Currently, our method of triangle consistency outperforms algebraic closure by far on $\mathcal{LR}$ -scenarios, at least for scenarios of reasonable size. It would be interesting to study the performance of our method on really big scenarios, but a lot of computing power is needed for that.

Future work will extend triangle consistency to the *DRA*-calculus, which feels very natural, since *DRA* is defined using $\mathcal{LR}$ -relations.

References

- [1] J.F. Allen, 'Maintaining knowledge about temporal intervals', *Communications of the ACM*, 832–843, (1983).
- [2] A. R. Bradley and Z. Manna, *The Calculus of Computation*, Springer, 2007.
- [3] J.-F. Condotta, M. Saade, and G.F. Ligozat, 'A Generic Toolkit for n-ary Qualitative Temporal and Spatial Calculi', in *TIME '06: Proceedings of the Thirteenth International Symposium on Temporal Representation and Reasoning*, pp. 78–86. IEEE Computer Society, (2006).
- [4] R. Dechter, 'From Local to Global Consistency', *Artificial Intelligence*, **55**, 87–108, (1992).
- [5] B. Dutertre and L. de Moura. The Yices SMT solver. Tool paper at <http://yices.csl.sri.com/tool-paper.pdf>, August 2006.
- [6] F. Dylla and R. Moratz, 'Empirical complexity issues of practical qualitative spatial reasoning about relative position', in *Proceedings of the Workshop on Spatial and Temporal Reasoning at ECAI 2004*, (2004).
- [7] G.F. Ligozat, 'Qualitative triangulation for spatial reasoning', in *Proc. International Conference on Spatial Information Theory. A Theoretical Basis for GIS*, pp. 54–68, (1993).
- [8] D. Lücke, T. Mossakowski, and D. Wolter, 'Qualitative Reasoning About Convex Relations', in *Spatial Cognition VI*, ed., Christian Freksa et al., (2008).
- [9] D. A. Randell, Zhan Cui, and A. Cohn, 'A Spatial Logic Based on Regions and Connection', in *KR'92. Principles of Knowledge Representation and Reasoning*, eds., Bernhard Nebel, Charles Rich, and William Swartout, 165–176, Morgan Kaufmann, (1992).
- [10] J. Renz and G.F. Ligozat, 'Weak Composition for Qualitative Spatial and Temporal Reasoning', in *Principles and Practice of Constraint Programming - CP 2005*, ed., Peter van Beek, volume 3709 of *LNCS*, pp. 534–548. Springer, (2005).
- [11] J. Renz and B. Nebel, 'Qualitative spatial reasoning using constraint calculi', in *Handbook of Spatial Logics*, eds., M. Aiello, I. E. Pratt-Hartmann, and J. F. A. K. van Benthem, Springer, (2007).
- [12] A. Scivos and B. Nebel, 'Double-crossing: Decidability and computational complexity of a qualitative calculus for navigation', in *Spatial Information Theory: Foundations of GIS*, ed., Dan R. Montello, volume 2205 of *LNCS*, pp. 431–446. Springer, (2001).
- [13] A. Scivos and B. Nebel, 'The Finest of its Class: The Natural, Point-Based Ternary Calculus $\mathcal{LR}$ for Qualitative Spatial Reasoning', in *Spatial Cognition*, pp. 283–303, (2004).
- [14] J. O. Wallgrün, L. Frommberger, D. Wolter, F. Dylla, and C. Freksa, 'A toolbox for qualitative spatial representation and reasoning', in *Spatial Cognition V: Reasoning, Action*, eds., T. Barkowsky, M. Knauff, G. Ligozat, and D. Montello, number 4387 in *Lecture Notes in Computer Science*, pp. 39–58, (2007).
- [15] D. Wolter and J. H. Lee, 'Qualitative reasoning about relative point position'. Submitted for publication, 2009.