

GeoGebra and the *realgeom* Reasoning Tool

Róbert Vajda^a, Zoltán Kovács^b

^aBolyai Institute, University of Szeged, H-6720 Szeged, Aradi vértanúk tere 1., Hungary

^bThe Private University College of Education of the Diocese of Linz, A-4020 Linz, Salesianumweg 3, Austria

Abstract

Reasoning tools for geometric equalities based on elimination methods are already integrated in some dynamic geometry systems (DGS), in particular in GeoGebra. However, no reasoning tools in any popular DGS environments for elementary inequalities exist yet. This paper investigates if the generic real quantifier elimination (RQE) methods and their existing implementations are mature enough from the practical point of view to explore simple elementary Euclidean triangle inequalities and the best possible geometric constants occurring in the inequalities. It is known, that from a theoretical point, real geometry inequality problems can be cast as RQE problems and can be solved by RQE algorithms. Unfortunately, the computational complexity of the RQE algorithms is rather high. The practical question is, if the RQE problem formulations for the investigated geometric inequality are tractable in a reasonable time limit. If this is so, an RQE based geometry inequality prover may be also an integral part of DGS. As a first experimental step, we also implemented a real reasoning tool for GeoGebra called *realgeom* which uses Wolfram's Mathematica poly-algorithmic RQE tool as an external service for solving the semi-algebraic problem associated to the geometric inequality exploration problem. Finally, we show by an example that parametric real root counting algorithms may be sufficient to attack certain inequality exploration problems and they can be later substitutes/alternatives for the general RQE tool in our framework.

Keywords

automated geometry proving, benchmark problem, database, dynamic geometry software, Euclidean planar geometry, GeoGebra, geometric inequality, Mathematica, mathematical software, QEPCAD, real quantifier elimination, Redlog, RegularChains, SyNRAC

1. Introduction

Complex algebraic geometry and real quantifier elimination (RQE) approaches can solve non-trivial elementary geometry problems in the field of automated theorem proving, see [11, 39, 15]. Dynamic geometry systems (DGS) are computer programs that allow the users to construct geometric figures by creating some *free* points and using them to define new objects such as lines, circles or other *dependent* points—later one can move the free points and see how the construction dynamically changes.

Automated reasoning for equational constraints based on, for example, Gröbner basis techniques is already available in some DGS, such as GeoGebra, see [23, 21]. That is, the system

SC² '20: Fifth International Workshop on Satisfiability Checking and Symbolic Computation, July 05, 2020, Paris, France (Virtual)

✉ vajda@math.u-szeged.hu (R. Vajda); zoltan@geogebra.org (Z. Kovács)

🌐 <http://www.math.u-szeged.hu/~vajda/> (R. Vajda); <http://matek.hu/zoltan> (Z. Kovács)

🆔 0000-0002-2439-6949 (R. Vajda); 0000-0003-2512-5793 (Z. Kovács)

© 2020 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).



CEUR Workshop Proceedings (CEUR-WS.org)

can prove/disprove in a geometric construction that certain quantities are equal or lines are orthogonal or parallel to each other, independently of the choice of the free points. A summary on the capabilities of the reasoning engine *Relation Tool* in GeoGebra 5 is reported in [25]. On the other hand, it is known that some other elementary geometry problems cannot be solved by equational reasoning—that is, the problem formulation and/or its solution needs polynomial inequalities [40]. One powerful general tool to deal with real geometry is real quantifier elimination (RQE) [12]. In the literature there are remarkable examples which treat some geometry problems via RQE, see [39, 19] and there are some computer algebra systems (CAS) or dedicated software packages which offer at least one method for RQE:

1. Maple’s RegularChains based package [9]
2. Maple’s SyNRAC package [24]
3. the QEPCAD B package [13, 8]
4. the Redlog package for Reduce [18]
5. Wolfram Mathematica’s combined built-in RQE engine [42]

Still, the high complexity of the algorithms for quantifier elimination from the theoretical side (see [17]) and the state of the art of the implementations from the practical side delay the occurrence of these as part of a powerful reasoning tool in DGS. In particular, there is no reasoning tool for geometric inequalities yet available in GeoGebra or in any other popular DGS. For the problems and the didactical challenges in the application of reasoning tools in DGS we refer to [4].

The current paper focuses only on geometric inequality exploration problems, which will be defined in Section 3 and leave aside the easier inequality-verifying problems. On a classical set of geometric inequalities for triangles mainly taken from [5], it explores how these problems can be cast as RQE problems, what sort of inequalities can be discovered by RQE using the already existing RQE implementations, which instances can be solved in a reasonable time.

As an experimental and practical first step of a larger work, our vision is the automated proving of simple inequalities and the discovery of (linear) inequalities between two geometric invariants. This translates to RQE problems with 2-7 variables.

The translated exploration problems are conjunctions of polynomial equations and inequalities where all variables are existentially quantified except for one, which is free. That is, we do not have *quantifier alternations* in the translated problems. Therefore, a direct application of some specialized RQE process (see Section 6 and [43, Chap. 6], [40, Chap. 2.5]) may still enlarge the real-time tractable problem class (see Section 4 for our problem-timing categories) with the proposed method and is under investigation by the authors. We note that although a translated problem is not a (quantifier-free) SMT instance, in fact it is very close to it and a huge part of the parameter space may be excluded from the exploration by proving unsatisfiability of the initial formula extended with inequalities for the free variable as a pre-processing step.

We implemented a real reasoning tool for GeoGebra called *realgeom* which uses Wolfram’s Mathematica RQE tool for solving the semi-algebraic problem associated to the geometric inequality exploration problem. The test implementation is freely available for various platforms [22, 1]¹. We illustrate our implementation with variants of seven inequality exploration problems

¹We suggest using a Raspberry Pi 4 machine with the Raspbian operating system that provides a free educational version of Wolfram’s Mathematica.

for triangles². All of them are well-known inequalities that were classified and proven in [5]:

- Bottema 1.1 (sides of triangles),
- Bottema 1.19 (sides of triangles),
- Bottema 4.2 (perimeter and area),
- Bottema 5.3 (perimeter and circumradius),
- Bottema 6.1 (perimeter and sum of heights),
- Bottema 8.1 (perimeter and sum of medians),
- Bottema 11.20 (sides of (right) triangles).

For example, Bottema 1.1 stands for the inequality

$$3 \cdot (ab + bc + ca) \leq (a + b + c)^2 < 4 \cdot (ab + bc + ca)$$

where a , b and c denote the lengths of the sides of an arbitrary non-degenerate triangle in the plane.

We do not aim for a complete method, neither for human readable proofs, nor for the improvement of the known underlying theoretical results.

The paper is structured as follows. Section 2 overviews the related works, benchmarks and repositories. Section 3 gives the definition of the geometric inequality exploration scenario in the GeoGebra/DGS framework. Section 4 gives the hierarchy of inequality exploration problems and their formulation as coordinate free (CF) RQE problems. Section 5 gives the formulation and the test results for the coordinate based (CB) GeoGebra-generated examples. Section 6 gives a detailed example of a problem formulation as real parametric root finding problem (PRF). Section 7 concludes the paper.

2. Related Work, and CAD, SMT Repositories

In the field of effective real quantifier elimination quite interesting and new theories and implementations arose over the last decade. Not only were Collins' Cylindrical Algebraic Decomposition (CAD) based RQE algorithm and Weispfenning's Virtual Substitution (VS) RQE method refined and improved and their powers were combined, but a new, regular chains based CAD approach also has emerged [9]. To our knowledge the five RQE tools investigated in the current work (RegularChains version 2548, QEPCAD version B 1.72, Redlog version 3258, SyNRAC 2016, Wolfram Mathematica 12.1) cover almost all the practical implementations which are available now for a user. The various tools were separately applied to solve many scientific problems and cited in many works (for references see [40]), but only very recently were their performance compared in practice, too, see [6]. This paper is also among the first ones which compares them for automated geometry inequality proving.

Many scientific papers report about success in automated reasoning in geometrical inequalities, see e.g. the BOTTEMA prover in [43]. However, in certain sub-problem classes, in particular for inequalities that contain nonlinear real arithmetic and (arbitrarily quantified) Tarski formulas, there are only a few public repositories available with clearly formatted problems. It

²In addition, a benchmark suite of more than 100 test cases is available at [14].

is worth to mention the Redlog Remis database [36] which contains RQE problems, about 15 of them being elementary geometry problems. Another repository [7] is maintained by Wilson at the University of Bath—it contains 80 input formulas in Maple and QEPCAD format, but these are typically not classical planar Euclidean geometry problems. It is also important to mention the web repository GeoThms [33], where a rich list of related geometry proving tools, DGS are also given—this database contains, however, only equational theorems. The importance of standardized, structured publicly available benchmark problems were noted in the SAT/SMT community and created SMT-LIB standard format and a public library for various logics, among them for QF_NRA (quantifier free nonlinear real arithmetic) and NRA (nonlinear real arithmetic), see [38]. We acknowledge that the support for the SMT-LIB format [20], [41] and the integration of computer algebra systems and SMT-solvers [20] reached a mature level now.

Our recent work contributes also to the publicly available test problems in this field by providing the investigated problems in various formats.

3. Exploration of Inequalities in DGS

In this section the geometry inequality exploration scenario and the functionalities of an integrated reasoning tool of a DGS is described. The section is written from a computer user's perspective, presumably from a mathematics teacher's or student's point of view.

That is, we assume that the user is familiar with the usual DGS construction steps in planar geometry. For the sake of simplicity, we consider triangle constructions. It is important that the user can label and refer to some invariants of the constructed triangles. That is, if the lengths of the triangle sides (a, b, c), the circumradius (R), the heights (h_a, h_b, h_c) or medians (m_a, m_b, m_c) are needed, the user is expected to be able to construct the according segments by using the standard tools of the DGS. We also allow to form a polynomial expressions of these quantities.

In this way we may refer to the perimeter of triangle as the sum of the triangle sides ($p = a + b + c$) or we may also form the positive integer powers of the circumradius (R^k , for a fixed integer k). Now by using a dedicated tool, the user may specify two referenced quantities \mathcal{A}, \mathcal{B} occurring in the construction and ask by a command of the form $\text{Compare}(\mathcal{A}, \mathcal{B})$, if a nontrivial infimum and/or supremum of the ratio $m = \frac{\mathcal{A}}{\mathcal{B}}$ exists. Then the reasoning tool returns sharp algebraic constants k, K and user is informed that $k \cdot \mathcal{B} \underset{(\text{=})}{<} \mathcal{A} \underset{(\text{=})}{<} K \cdot \mathcal{B}$ holds and possibly

no sharper results for the range of the ratio exists. We note that in the case of trivial lower or upper bound such as $k = 0$ or $K = +\infty$, the output formula may be simpler.

As future work we may consider utilizing a knowledge base of formulas for substituting the input of geometric invariants of a triangle by expressions of the sides or vertex coordinates of the triangle [43, p. 240].

Our prototype implementation *realgeom* [35] is based on the vertex coordinates. It can be seen as an extension of the Relation tool [25] in GeoGebra. The Relation tool can provide basic metric information about quantities, but its capabilities are mostly constrained to decide if two quantities are equal or not.

Now, with the new tool *realgeom*, interesting pieces of further information in Euclidean planar geometry may be studied. For example, a student starts an experiment by drawing a

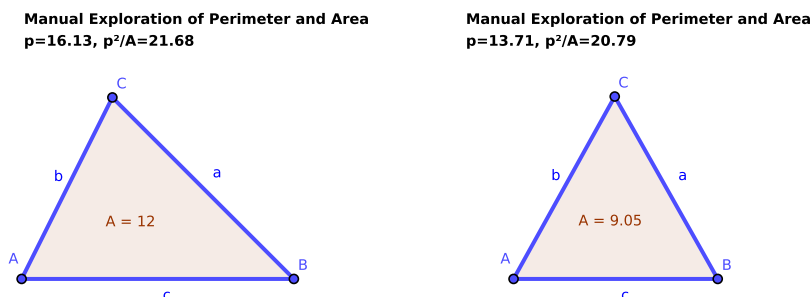


Figure 1: Two moments of manual exploration of the ratio p^2/A in GeoGebra

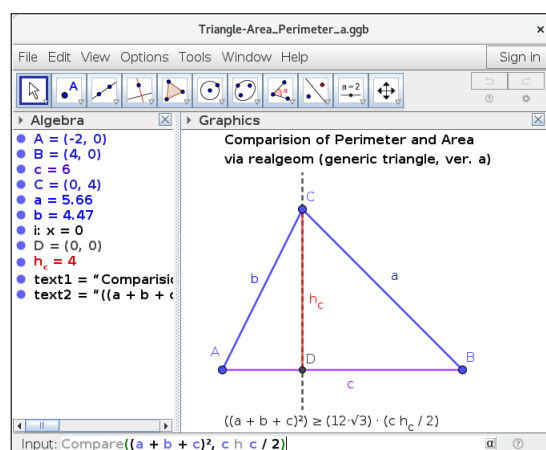


Figure 2: Automatic result on computing the ratio p^2/A in GeoGebra/*realgeom*

triangle ABC and collecting experimental data on the quantities $p = a + b + c$, A , and the ratio p^2/A . Here we recall that p stands for the perimeter of the triangle and A for the area. The experiment shows that p^2/A is minimal if the triangle is close to an equilateral triangle and the minimal ratio is about 20.79 (see Fig. 1 for two screenshots of the experiment). By continuing the experiment the student enters the command `Compare((a + b + c)^2, c * h_c / 2)`. Here we recall that h_c stands for the height that belongs to side c . Then the tool informs (see the formula in Fig. 2) the user on the fact that $(a + b + c)^2 \geq 12\sqrt{3} \cdot c \cdot h_c / 2$, for all non-degenerate triangles. Indeed, $12\sqrt{3} \approx 20.78$, so the DGS exploration is confirmed by CAS/RQE support.

The first implementation of *realgeom* on the top of GeoGebra uses Mathematica's general quantifier elimination tool [42] as an external service to get the solution (see [3] for a similar concept). The *realgeom* system preprocesses the input formula that was generated by GeoGebra coordinatization by eliminating as many variables as possible from linear equations.

We note that in our first implementation no deep explicit structural analysis of the problem is present—the number of variables/equations/inequalities, the degree of the polynomials, or other statistics do not influence the internal algorithm—we rely just on the poly-algorithmic

heuristic of the Mathematica tool to pick up good variable ordering or divide the problem into sub-problems.

Later on, this analysis needs to be added to our implementation. Moreover, an internal algorithm should be utilized to avoid closed source work. Some possible ways to go is to compile QEPCAD or Redlog into a portable module (as it was performed for the Giac CAS for various platforms, see [27]), or use a library like [2].

Free availability of the presented system can be crucial in an educational context. In fact, teaching basic geometric inequalities is a part of the secondary mathematics curriculum in many countries. Among others, the triangle inequality, or simple consequences of the law of cosines, or discussion of constructability are typical possible uses. On the other hand, geometric inequalities often appear at mathematics contests [29] as problems to solve. Therefore, a computer software tool that can help in conjecturing or even in proving geometric inequalities, seems to be a helpful step forward.

4. Hierarchy of the Inequality Exploration Problems

In this section we give an overview on a possible workflow on solving a given exploration problem.

An important part of our research was started with looking up standard collections of elementary geometric inequalities. As already mentioned above, the collection [5] was a very supporting database in our work.

First of all, we observed that the simplest items in this database are the inequalities for triangles which involve only the sides of the triangles. On the other hand, it is important to note that the typical inequalities are given in terms of geometric invariants of the triangle, rather than coordinates of the vertices of the triangles. Almost all inequalities enumerated in [5, Chap 1.] are of the type above, and more than half of them are about the ratio of two trivariate symmetric homogeneous polynomials of degree at most three, that is p_1/p_2 , where $p_1, p_2 \in \mathbb{Z}[a, b, c]$. In later chapters of the database more invariants such as radii, altitudes, medians of the triangle come into play.

We built up our own test database in an own fork of the GeoGebra DGS at GitHub at <https://github.com/kovzol/geogebra>, in folder `test/scripts/benchmark/compare/tests`, based mostly on the Bottema collection. For this paper we selected 7 benchmark problems, namely the ones given in Section 1. That is, the examples 1.1, 1.19, 4.2, 5.3, 6.1, 8.1 and 11.20 were chosen for testing and we decided to investigate their *variants*. As we will see, several variants of a geometry-inequality exploration problem (GIEP) can exist. On the one hand we may need to test the inequality problems not only for *general (non-degenerate) triangle (GT)*, but also for

- isosceles triangles (IT),
- right triangles (RT),
- isosceles right triangles (IRT), and
- equilateral triangles (ET).

On the other hand, variants of the given problem can exist also in terms of the algebraic formulation of the problem: the formulation can be coordinate free (CF) as it appears in [5] or

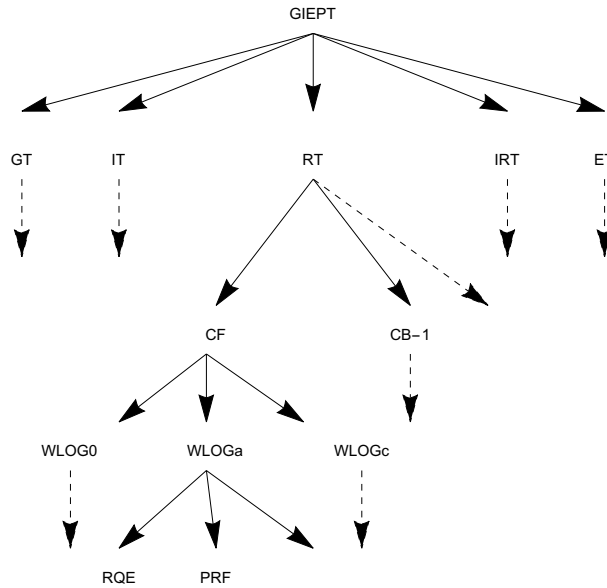


Figure 3: Structure of the analysis of the exploration problem

coordinate based (CB), as in our case, and the latter is also dependent of the particular DGS construction which were used to construct certain quantities or the triangle itself.

Finally, in fact, as usual, some variables of the semi-algebraic formula which encodes the inequality problem may be specialized to a constant to reduce the complexity of the algebraic problem (for instance, by putting one vertex of a triangle to $(0, 0)$ or assuming that the side of the triangles satisfy that $a \leq b \leq c = 1$). Depending on the decision of the particular specializations, again, several problem-variants may be derived for the same GIEP.

Fig. 3 shows an illustration of types and hierarchy of the exploration of a geometry-inequality problem (GIEPT). On the top level, a triangle class is selected: a (non-degenerate) general triangle (GT) or the same problem for a specialized triangle class (IT, RT, IRT, ET). Instead of investigating the problem for general triangle (GT), the analysis of the specialized right triangle (RT) variant will be given in detail.

On the next level, a decision must be made if the computation will be a coordinate-free (CF) or a certain coordinate-based one (CB-1, CB-2, ...)—note that there are several ways to describe the same GIEP problem with coordinates. As a next decision, the algorithm must decide if the variables remain as general as possible (WLOG0, that is, no “without loss of generality” assumption will be made), or, eventually the cathetus a (WLOGa) or the hypotenuse c (WLOGc) will be substituted by a particular value—“without loss of generality”. (See [16] for a short summary on the topic.) Finally, the underlying computation system may have multiple choices on selecting the best fitting “real” algorithm, namely an RQE or a PRF based calculation. Since Fig. 3 is just a partial illustration, the full structure in the background is usually more complicated.

Table 1

Performance of various RQE tools for the general CF benchmarks

B-T-CFC7	Solved (ST+S2+S3)	Solved in Time (ST)
Mathematica	7/7	6/7
QEPCAD	5/7	5/7
RegularChains	5/7	5/7
Redlog	5/7	5/7
SyNRAC	5/7	5/7

4.1. Testing 7 CF variants with 5 RQE tools

In this section we report on testing the general CF variants of the seven problems above with 5 different RQE tools (see Table 1, the abbreviation B-T-CFC7 stands for “Bottema’s triangles, coordinate-free cases, 7 tests”). This preliminary research founded our further work to select the most suitable system for the additional steps in our study.

To illustrate this variant let us consider the problem Bottema 5.3, where inequalities for the ratio of $\mathcal{A} = a + b + c$ and $\mathcal{B} = R$ (perimeter and circumradius) are given. Without loss of generality we can assume that $a = 1$, to reduce the number of variables. We formulate the RQE problem as follows:

$$\begin{aligned}
 \exists_{b,c,R} \quad & m > 0 \wedge 1 + b > c \wedge 1 + c > b \wedge b + c > 1 \wedge R > 0 \\
 & \wedge (b + c + 1) \cdot (b + c - 1) \cdot (1 + b - c) \cdot (1 + c - b) \cdot R^2 = b^2 \cdot c^2 \\
 & \wedge (1 + b + c) = m \cdot R.
 \end{aligned} \tag{1}$$

All five tools solve the problem and return an output formula equivalent to

$$0 < m \leq 3\sqrt{3}. \tag{2}$$

Not all the investigated RQE tools return syntactically exactly the formula (2). It could be considered a canonical form of the description of the one-dimensional semi-algebraic solution set. Moreover, some other output formula may contain real algebraic numbers as bounds either directly in the output formula or implicitly encoded in the Tarskian language. Still, the extraction of the needed sharp geometrical constants after some post-processing from the obtained one variable quantifier free RQE output formula can be considered standard, the computation is cheap, and is not described in more detail here.

For the further classification we set up four categories. In this way we can benchmark if the investigated problem is tractable in a reasonable time limit.

- Solved in Time (ST): Problem solved in 10 sec³. This could be considered in a real time exploration via a DGS as reasonable time limit.
- Solved (S2): Problem solved in 300 sec. This is still promising, because some progress on the practical implementation may bring the problem to the practically tractable class.

³All computations were performed on a PC with Intel Core i7 3.2 Ghz, 9 GB RAM.

- Solved (S3): Problem solved less than 3600 sec, but more than 300 sec, it is interesting only for research purposes, not for practical real time applications in the context of proving.
- Not Solved, TimeOut (TO): Not solved in one hour.

We see from the Table 1, and also further test results for problems with symmetric homogeneous polynomials and for [5, Chap. 1]) confirmed, that Mathematica’s poly-algorithmic tool performed the best among the available—and for us known—RQE tools. So for further testing and for proto-implementation, we decided to use Mathematica’s RQE.

5. Test Results of the *realgeom* Tool

In this section we report about the CB variants of the 7 inequality problems, which were enumerated in Section 1. The tests and the results are available at [14] and classified into the following 3 groups.

5.1. Isosceles and Right Triangles

Group 1 (abbreviated with B-IR-CBC7 which stands for Bottema’s database, the isosceles/right triangles, that cover 7 cases) consists of 28 problems,

- 2×7 CB problems for isosceles triangles and
- 2×7 CB problems for right triangles.

Because of two different construction paths, for each triangle type from above and for each inequality problem there are two variants. For example, for isosceles triangles, Version B (Ver. B) refers to a DGS construction, where from a perpendicular bisector g of a segment AB , a point C is chosen to form an isosceles triangle, while version A (Ver. A) indicates another construction path: From a circle with center A and radius b , two different points B, C are chosen to form an isosceles triangle. The different construction paths lead to different (although mathematically equivalent) semialgebraic problems.

We also note that these problems also have proven theoretically important, because the corresponding semi-algebraic problem, after a suitable specialization, can be considered as a 1-parameter problem, see Section 6.

The results generated by the *realgeom* tool are presented in Table 2.

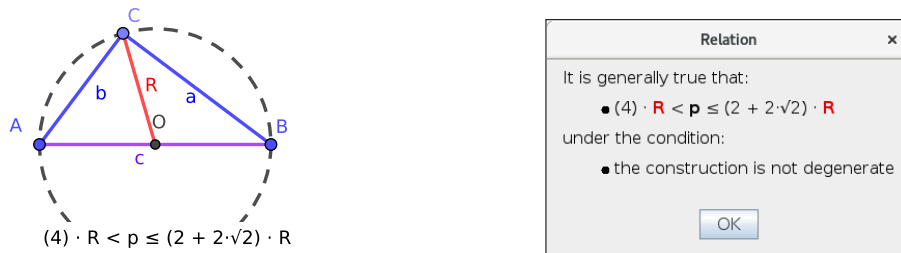
For illustration, we give the exploration DGS window (construction with the usage of the tools) and the derived semi-algebraic formula for the test case RightTriangle-Bottema5.3a (referenced as R 5.3 in Table 2).

$$\begin{aligned}
 \exists_{v_8, v_{11}, v_{13}, v_{14}} \quad & m > 0 \wedge v_{11} > 0 \wedge v_{13} > 0 \wedge v_{14} > 0 \wedge v_8^2 - v_{13}^2 = 0 \\
 & \wedge v_8^2 - v_{14}^2 + 1 = 0 \wedge \frac{1}{4}v_8^2 - v_{11}^2 + \frac{1}{4} = 0 \wedge \\
 & \wedge 1 + v_{13} + v_{14} = m \cdot v_{11}.
 \end{aligned} \tag{3}$$

Table 2Performance of the RQE-based *realgeom* tool for isosceles and right triangles

B-IR-CBC7	Result	Timing Ver. A (ms)	Timing Ver. B (ms)
I 1.1	$3 \leq m < 4$	60	73
R 1.1	$\frac{10+8\sqrt{2}}{7} \leq m < 4$	61	60
I 1.19	$\frac{1}{3} \leq m < \frac{1}{2}$	38	44
R 1.19	$6 - 4\sqrt{2} \leq m < \frac{1}{2}$	62	60
I 4.2	$m \geq 12\sqrt{3}$	66	48
R 4.2	$m \geq (12 + 8\sqrt{2})$	48	52
I 5.3	$m \leq 3\sqrt{3}$	1150	84
R 5.3	$4 < m \leq (2 + 2\sqrt{2})$	192	931
I 6.1	$m \leq \frac{\sqrt{3}}{2}$	216	374
R 6.1	$\frac{1}{2} < m \leq \frac{3-\sqrt{2}}{2}$	196	9156
I 8.1	$\frac{3}{4} < m < 1$	237	97
R 8.1	$\sqrt{5} - \sqrt{3 - \sqrt{5}} - \frac{1}{2} \leq m < 1$	4120	179494
I 11.20	$m \geq 1$	35	35
R 11.20	$1 < m \leq \sqrt{2}$	42	42

**Comparison of Circumradius and Perimeter
via *realgeom*, Bottema 5.3
(right triangle, ver. a)**

**Figure 4:** Exploration of the ratio p/R in GeoGebra with the *realgeom* tool

The output formula is equivalent to

$$4 < m \leq 2 + 2\sqrt{2}. \quad (4)$$

Fig. 4 shows how our implementation communicates this theorem via the Compare command, or, in a more user-friendly way, via the Relation tool—here the user can obtain the result by clicking the icon of the tool, then p , R , and finally on “More. . .” to compare the two lengths symbolically.

5.2. Equilateral and Isosceles Right Triangles

Group 2 (abbreviated as B-EIR-CBC7) consists again 2×7 problems for equilateral, and for isosceles right triangles. This is the easiest problem class with corresponding zero dimensional

Table 3Performance of the RQE-based *realgeom* tool for equilateral and isosceles right triangles

B-EIR-CBC7	Result	Timing Ver. A (ms)
E 1.1	$m = 3$	60
IR 1.1	$m = \frac{10+8\sqrt{2}}{7}$	61
E 1.19	$m = \frac{1}{3}$	38
IR 1.19	$m = 6 - 4\sqrt{2}$	62
E 4.2	$m = 12\sqrt{3}$	66
IR 4.2	$m = (12 + 8\sqrt{2})$	48
E 5.3	$m = 3\sqrt{3}$	1150
R 5.3	$m = (2 + 2\sqrt{2})$	192
E 6.1	$m = \frac{\sqrt{3}}{2}$	216
IR 6.1	$m = \frac{3-\sqrt{2}}{2}$	196
E 8.1	$m = \frac{\sqrt{3}}{2}$	237
IR 8.1	$m = \sqrt{5} - \sqrt{3} - \sqrt{5} - \frac{1}{2}$	4120
E 11.20	$m = 1$	35
IR 11.20	$m = \sqrt{2}$	42

ideals (same amount of variables and equations in the algebraic problem) so we give here only one variant per geometry inequality problem.

In fact, these test cases can be studied without real reasoning in a more or less satisfactory way. After certain specialization only elimination of all variables but m is required, and this can be achieved via Gröbner basis computation. After obtaining an elimination ideal we learn that it is zero dimensional and after factorization a set of possible algebraic numbers can be obtained. Finally, excluding those that are negative we get one or more algebraic numbers.

That is, in some cases we get more than one algebraic number—for instance, the test case `IsoRightTriangle-Bottema1.19a` (abbreviated as IR 1.19 in Table 3) produces the possible set $\{6 - 4\sqrt{2}, 6 + 4\sqrt{2}\}$. Actually, the correct solution $6 - 4\sqrt{2}$ has the minimal polynomial $x^2 - 12x + 4$ that has two positive roots, and the other root cannot be excluded without real reasoning. Here we refer to [34, 15, 26] for further details on this approach.

Comparison of speed of the two different approaches should be addressed by future work.

5.3. General Triangles

Finally, Group 3 (B-T-CBC7) are coordinate based counterparts of the 7 problems for general triangles (B-T-CFC7) investigated in Section 4.

We see that those coordinate based versions of the inequality problems for the general triangles, which consist of the sides only or the sides and an extra geometric invariant, could be handled again quite well with RQE. But two problems remained unsolved (TB 6.1, TB 8.1). The result for the case TB 6.1 (sum of heights and perimeter) was a bit surprising, because a CF version formulation could be solved via Mathematica’s RQE easily (cf. Table 1). The case TB 6.1 was expected (sum of medians and perimeter), because the easier CF version is proven to be also difficult and needed more than an hour for its RQE solution. This problem is known to be challenging geometry problem ([32, p. 86]) and we do not see at the moment a chance to solve

Table 4

Performance of RQE-based Realgeom tool for general triangles.

B-T-CBC7	Result	Timing Ver. A (ms)
TB 1.1	$3 \leq m < 4$	333
TB 1.19	$\frac{1}{3} \leq m < \frac{1}{2}$	284
TB 4.2	$m \geq 12\sqrt{3}$	283
TB 5.3	$m \geq 3\sqrt{3}$	450
TB 6.1	$m = ?$	TO
TB 8.1	$m = ?$	TO
TB 11.20	$m \geq 1$	66

it with our tool or with an improved direct RQE call in a few seconds, because it contains many variables.

Therefore in the next section, we suggest an alternative method (PRF) for attacking problems with many variables. We note that the all details of the alternative method is not yet elaborated or implemented by the author.

6. Real Parametric Root Finding (PRF)

In this section we reconsider the test case Bottema 5.3 for right triangles. With this example we demonstrate that certain geometry inequality problems can be solved without using full RQE, that is, without building up a CAD for the space R^n , where n is defined by all the occurring variables in the system. Instead, we need only a smaller dimensional generic CAD decomposition (see e.g. [30, Section 2]) and some Gröbner basis computations. This method is referenced as a variant of parametric root finding (PRF).

To utilize PRF, we assume that the input formula of the geometry exploration problem consist of conjunctions of equations and strict inequalities and we can consider part of the occurring variables as *parameters* of the system, keeping the other variables as indeterminates. If the system is well-behaved [30] with respect to the parameters, that is, we have as many equations as indeterminates and some auxiliary technical conditions, then generically, the system will have only finitely many solutions with respect to a fixed parameter set. Moreover, except for a small dimensional subset of the parameter space, which is described by the minimal discriminant variety (MDV), PRF is able to characterize those parameter values for which the system has exactly k (constantly many) solutions by cylindrically decomposing the parameter space into open cells.

We analyse the semi-algebraic problem related to one of the coordinate based versions of RB 5.3. with PRF. Here is the derived polynomial constraint system with equations and strict inequalities (cf. (3)):

$$\begin{aligned}
 m > 0 \wedge v_{11} > 0 \wedge v_{13} > 0 \wedge v_{14} > 0 \wedge v_8^2 - v_{14}^2 + 1 = 0 \\
 \wedge v_8^2 - v_{13}^2 = 0 \wedge \frac{1}{4}v_8^2 - v_{11}^2 + \frac{1}{4} = 0 \wedge (1 + v_{13} + v_{14}) = m \cdot v_{11}.
 \end{aligned} \tag{5}$$

The problem has 5 variables and 4 equations, that is, one less equations than variables. Here $m, v_{13}, v_{14}, v_{11}$ refer to the parameter, the sides a and b and the circumradius R , in an

automatically translated way in GeoGebra’s internals. Side c , the length of the hypotenuse is not present, because due to the suitable coordinatization, it has been derived to be 1.

If we fix m , the system has only finitely many solutions, and therefore it can be also analyzed by parametric root finding algorithms [28, 31]. It is known that for these problems RQE is an overkill and in order to decide if the system has a real solution for a parameter value m , CAD is needed only for 1-D space.

We expect from this a speed-up for problems with more than 5 variables. Our benchmark problem `IsoTriangle-Perimeter_CircumRadius_c` from our collection—this is a variant of I 5.3, but uses different construction steps—results in a problem with 9 variables. It times out by RQE, but can be smoothly solved via PRF within 5 seconds.

Here, for problem (5), it is clear that the minimal discriminant variety (MDV) is given by the polynomial

$$p = m(m - 4)(m + 4)(m^2 - 4m + 4)(m^2 + 4m - 4), \quad (6)$$

and assuming that $p \neq 0$, the generic CAD decomposition consisting of 4 cells (!) reveals that the system has finitely many (namely: four) solutions if $4 < m < 2 + 2\sqrt{2}$. Testing extra the finitely many positive real roots of p , and by combining the results, we can extract again necessary and sufficient conditions for m . Thus, we obtain the same quantifier free formula that we have via RQE in (4).

We note that we have the same problem class for all CF and CB based formulation for isosceles and right triangles (cf. Table 2). The treatment of the general triangle case is also possible with the same method, but it is more involved, because we have to analyze the solution of a system with 2 parameters.

To our knowledge, two Maple packages are available that treat general real root classification problems without computing a full CAD for the full space [30, 10]. Their specific performance for the triangle problems we are interested in, is yet under investigation by the authors.

7. Conclusion and Future Work

We investigated the possibility of the automatic exploration of geometric inequalities based on RQE methods. We tested five available RQE tools which offer various different algorithms for performing effective quantifier elimination. Our coordinate free and coordinate based problem formulations for triangle inequality exploration problems and the first test runs indicated that problems involving only two geometric invariants may be translated into a first order input formula with at most seven variables (one being free among them). At least one of the tools, namely Mathematica’s combined tool is able to solve most of the problems in a reasonable time limit, which in our opinion is essential for exploration in a DGS in a didactical context.

We implemented a *realgeom* reasoning tool on top of GeoGebra, which uses Mathematica RQE algorithm as an external service to solve the derived RQE problems. Inequalities involving more variables or radical expressions may need an improved approach like parametric real root finding, see also [37], [43].

Although specialized real SMT-solvers such as SMT-RAT, Z3, etc., directly cannot explore the exact sharp constants in a geometry exploration problem, they can approve (unvalidated)

numerically obtained bounds and thus, as a pre-processing step, by narrowing the search space, they may contribute to the efficiency of the geometric reasoning tool.

In our opinion the first steps on investigating if some RQE algorithms are already mature enough to support learning experiments on inequalities, are done. The next steps would be to use refined and specialized methods to explore elementary inequalities in a DGS framework.

Acknowledgments

The first author was supported by the EU-funded Hungarian grant EFOP-3.6.1-16-2016-00008. The second author was partially supported by a grant MTM2017-88796-P from the Spanish MINECO (Ministerio de Economía y Competitividad) and the ERDF (European Regional Development Fund).

References

- [1] Abar, C.A.A.P., Kovács, Z., Vajda, R., Recio, T.: Conectando Mathematica e GeoGebra para explorar construções geométricas planas. Brazilian Wolfram Technology Conference, São Paulo, Brazil (2019)
- [2] Basic Polynomial Algebra Subprograms (BPAS) Library, bpaslib.org. Last accessed 1 May 2020
- [3] Botana, F., Kovács, Z.: A Singular web service for geometric computations. *Ann. Math. Artif. Intelligence* **74**(3-4), 359–370 (2015)
- [4] Botana F., Recio T., Velez M.P.: The role of automated reasoning of geometry statements in mathematics instruction, CERME 10 position poster (2017)
- [5] Bottema, O., Djordjevic, R.Z., Janic, R.R., Mitrovic, D.S., Vasic, P.M.: *Geometric Inequalities*. Wolters-Noordhoff Publishing, Groningen (1969)
- [6] Bradford R., Davenport J., England M., McCallum S., Wilson D.: Truth Table Invariant Cylindrical Algebraic Decomposition. In: Gerdt, V.P., Koepf, W., Seiler, W.M., Vorozhtsov, E.V. (eds.) *Computer Algebra in Scientific Computing*. CASC 2014. LNCS, vol. 8660, pp. 44–58. Springer, Cham (2014)
- [7] Bradford, R., Davenport, J., Wilson, D.: A repository for CAD Examples. *ACM Comm. in Comput. Alg.* **46**(3), 67–69 (2013)
- [8] Brown, C.W.: An Overview of QEPCAD B: a Tool for Real Quantifier Elimination and Formula Simplification. *J. Japan Soc. Symb. Alg. Comput.* **10**(1), 13–22 (2003)
- [9] Chen, C., Maza, M.M.: Quantifier Elimination by Cylindrical Algebraic Decomposition based on Regular Chains. *J. Symb. Comput.* **75**, 74–93 (2016)
- [10] Chen, C., Maza, M.M.: Solving Parametric Polynomial Systems by RealComprehensive-Triangularize. In: *Proceedings of ICMS 2014 – 4th International Congress*, Seoul, South Korea, LNCS, vol. 8592, pp. 504–511. (2014)
- [11] Chou, S.C.: *Mechanical Geometry Theorem Proving*. D. Reidel Publishing Company, Dordrecht, Netherlands (1988)
- [12] Collins, G.E.: *Quantifier Elimination for Real Closed Fields by Cylindrical Algebraic De-*

- composition. In: Caviness, B.F., Johnson, J.R. (eds.): Quantifier elimination and cylindrical algebraic decomposition, 85–121, Springer (1998)
- [13] Collins, G.E., Hong, H.: Partial Cylindrical Algebraic Decomposition for Quantifier Elimination, *J. Symb. Comput.* **12**, 299–328 (1991)
- [14] Comparison benchmark for GeoGebra 46, <https://tinyurl.com/realgeom46>. Last accessed 14 May 2020
- [15] Dalzotto, G., Recio, T.: On protocols for the automated discovery of theorems in elementary geometry. *J. Autom. Reasoning* **43**(2), 203–236 (2009)
- [16] Davenport, J.: What Does “Without Loss of Generality” Mean, and How Do We Detect It. *Math. in Comp. Sci.* **11**, 297–303 (2017)
- [17] Davenport, J., Heintz J.: Real quantifier elimination is doubly exponential. *J. Symbolic Comput.* **5**, 29–35 (1988)
- [18] Dolzmann, A., Sturm, T.: Redlog: Computer Algebra Meets Computer Logic. *ACM SIGSAM Bulletin* **31**(2), 2–9 (1997)
- [19] Dolzmann A., Sturm, T., Weispfenning V.: A new approach for automatic theorem proving in real geometry. *J. Autom. Reasoning* **21**, 357–380 (1998)
- [20] Forrest, S.A.: Integration of SMT-LIB Support into Maple. Second Annual *SC²* Workshop, ISSAC 2017, Kaiserslautern, Germany (2017)
- [21] GeoGebra, <https://www.geogebra.org/>. Last accessed 1 May 2020
- [22] GeoGebra Discovery, <https://github.com/kovzol/geogebra-discovery/>. Last accessed 14 May 2020
- [23] Hohenwarter, M.: Ein Softwaresystem für dynamische Geometrie und Algebra der Ebene. Master’s thesis. Paris Lodron University, Salzburg (2002)
- [24] Iwane, H., Yanami, H., Anai, H.: SyNRAC: a toolbox for solving real algebraic constraints. In: Hong, H., Yap, C. (eds.) *ICMS 2014*. LNCS, vol. 8592, pp. 518–522. Springer, Heidelberg (2014)
- [25] Kovács, Z.: The Relation Tool in GeoGebra 5. In: Botana, F., Quaresma, P. (eds.), *Post-conference Proceedings of the 10th International Workshop on Automated Deduction in Geometry (ADG 2014)*, 9–11 July 2014, LNAI, vol. 9201, pp. 53–71. Springer, Heidelberg (2015)
- [26] Kovács, Z.: Discovering Geometry Theorems in Regular Polygons. In: Fleuriot, J., Wang, D., Calmet, J. (eds.) *Artificial Intelligence and Symbolic Computation. AISC 2018*. LNCS, vol. 11110, pp. 155–169. Springer, Cham (2018)
- [27] Kovács Z., Parisse B., Giac and GeoGebra – Improved Gröbner Basis computations, LNCS, vol. 8942, 126–138 (2015)
- [28] Lazard, D., Rouillier, F.: Solving parametric polynomial systems. *J. Symb. Comput.* **42**(6), 636–667 (2007)
- [29] Leng, G.: *Geometric Inequalities*. Mathematical Olympiad Series, Vol. 12. East China Normal University Press, World Science Publishing (2016)
- [30] Liang, S., Gerhard, J., Jeffrey, D.J., Moroz, G.: A package for solving parametric polynomial systems. *ACM Comm. Comp. Algebra* **169**(43), 61–72 (2009)
- [31] Moroz, G.: Complexity of the resolution of parametric systems of polynomial equations and inequations. In: *ISSAC ’06: Proceedings of the 2006 international symposium on Symbolic and algebraic computation*, pp. 246–253 (2006)

- [32] Posamentier, A.S., Salkind, T.S.: Challenging Problems in Geometry. Dover Publications, New York (1996)
- [33] Quaresma, P., Janičić, P., GeoThms: a Web System for the Euclidean Geometry. *Electronic Notes in Theoretical Computer Science* **174**(2), 35–48 (2007)
- [34] Recio, T., Vélez, M.P.: Automatic discovery of theorems in elementary geometry. *J. Autom. Reasoning* **23**, 63–82 (1999)
- [35] realgeom, <http://github.com/kovzol/realgeom>. Last accessed 15 May 2020
- [36] Remis, <http://www.redlog.eu/remis/>. Last accessed 1 May 2020
- [37] Robu, J.: Automated proof of geometry theorems involving order relation in the frame of the Theorema project. KEPT (2007)
- [38] The Satisfiability Modulo Theories Library (SMT-LIB), www.SMT-LIB.org. Last accessed 1 May 2020
- [39] Sturm, T., Weispfenning, V.: Computational geometry problems in REDLOG. *International Workshop on Automated Deduction in Geometry. LNCS*, vol. 1360, pp. 58–86. Springer, Berlin, Heidelberg (1996)
- [40] Sturm, T.: A survey of some methods for real quantifier elimination, decision, and satisfiability and their applications. *Math. Comput. Sci.* **11**, 483–502 (2017)
- [41] Vale-Enriquez, F., Brown, C.W.: Polynomial Constraints and Unsat Cores in TARSKI. In: Davenport, J., Kauers, M., Labahn, G., Urban, J. (eds.) *Mathematical Software – ICMS 2018. LNCS*, vol. 10931, pp. 466–474. Springer, Cham (2018)
- [42] Wolfram Research, Inc.: Mathematica, Version 12.1, Champaign, IL (2020)
- [43] Yang L., Xia B.: *Automated Inequality Proving and Discovering*, World Scientific (2016)