



Determining automatically compass and straightedge unconstructibility in triangles

Pascal Mathis and Pascal Schreck

Abstract

In many areas, some geometry problems can not be solved using only geometry and are treated by means of algebraic tools. However, geometric properties can be still employed to simplify the system of equations. This allows either to speed up the treatment or, more radically, to make the treatment possible. In this article, we illustrate this approach with a family of problems involving the construction of triangles. In all these problems the goal it is to determine if there is a compass-and-straightedge construction of the three vertices of a triangle knowing only three located points of this triangle such as altitudes, inner center, circumcenter and so on. Algebraic tools, basically Galois theory, are needed to answer the question. In this paper we describe a geometric reasoning phase that is required to provide a polynomial algebraic system that algebraic softwares can address within an acceptable time despite the exponential complexity of the underlying algorithms.

1 Introduction

Geometric constructions are well known for who is interested in the epistemology of mathematics. Indeed, they played a central role in the definition and understanding of numbers in Ancient Greece. Besides, they also have a practical aspect in several technical domains like architecture, mechanical design or topography. Straightedge (or ruler) and compass constructions in short RC-constructions are geometric constructions performed using only straightedge and compass. They are famous since they allowed to precisely define a class of constructions meeting philosophical considerations for the Ancient Greeks. Through the ages, they have also provided to generations of students in mathematics a plethora of problems in geometry where the imaginations and the inventivenesses of the students are exercised. This is still the case today. Moreover, such problems have also a recreational aspect in mathematical circles: they are not difficult to understand, some are easy while other are difficult but the solution, usually, does not invoke big specialized theories. For instance, in the eighties, William Wernick proposed a list of 139 construction problems about triangles [14], this corpus has been extended by Harold Connelly [2] by adding four characteristic points related to the nine points circle (also known as the Euler circle). We shortly present these corpora in the section 2.

By a strange paradox, RC-construction problems are also famous because of problems that are *not* solvable using only straightedge and compass. The most famous of them is perhaps the problem of squaring the circle which passed to the common language as a synonym of problem impossible to solve. It was not before the nineteenth century that this kind of problems was

proved not RC-constructible by going *outside* the RC-construction theory through the algebraic notion of field extensions.

Nevertheless, even today with powerful computer algebra systems, it is not easy to prove that a geometric construction problem cannot be solved by a RC-construction (we will say RC-solvable). Even with small geometric problems involving 5 or 6 points, the polynomials that the user has to handle can grow exponentially in size and the resolution time can become huge. For common desktop computers, it is usual to overflow the memory and to be unable to complete the calculus. In a previous work [10], we designed an automatic tool implemented with one of the most famous CAS (*Computer Algebra System*), namely Maple, to decide if a problem coming from Wernick corpus is RC-solvable or not.

Unfortunately, when tackling Connelly corpus, we find that the techniques that we used for Wernick corpus are not powerful enough to solve all the problems. In order to solve the whole Connelly corpus, we had to pre-process the algebraic systems to make them simple enough to be solved with our Maple program. This article explains how geometrical knowledge is used for this purpose.

The rest of the paper is organized as follows. Section 2 briefly summarizes the basic algebraic notions underlying geometric constructions. Section 3 presents a method to dynamically set up algebraic systems by taking the properties between the given characteristic points into account.

2 Some basics about RC-constructions and algebra

2.1 RC-constructibility and algebra

Let us begin with this classical definition of RC-constructibility.

Definition 1. Given a finite set of points $\mathcal{B} = \{B_0, \dots, B_m\}$ in the Euclidean plane, a point P is *RC-constructible from the set \mathcal{B}* if there is a finite set of points $\{P_0, \dots, P_n\}$ such that $P = P_n$, $P_0 \in \mathcal{B}$ and every point P_i ($1 \leq i \leq n$) is either a point from \mathcal{B} or is obtained as the intersection of two lines, or of a line and a circle, or of two circles, themselves obtained as follows:

- any considered circle has its center in the set $\{P_0, \dots, P_{i-1}\}$ and its radius is equal to the distance $P_j P_k$ for some $j < i$ and $k < i$;
- any considered line passes through two points from the set $\{P_0, \dots, P_{i-1}\}$.

The sequence of these points with their basic construction in terms of intersection between lines and circles is called a RC-construction of point P .

An ambiguity which will be crucial in the following lies in the status of the points in \mathcal{B} : they can be real points in the plane, or variable points also called free points in the dynamic geometry terminology or also parameters in constraint system domain. It should be noticed that the notion of RC construction is in fact a little bit more complicated when all the cases, included the degenerated ones, are taken into account [6]. We will not go any further with this notion, but this will be discussed in the following.

When considering the problems of construction which are expressible by means of polynomial equations involving the coordinates of the points, Wantzel result can be used in order to prove that a problem is not RC-solvable. This theorem states that if F is the field extension of \mathbb{Q} containing the coordinates of point set \mathcal{B} , the sought points are straightedge and compass constructible in F if and only if their coordinates can be expressed by arithmetic expressions

with radicals involving only numbers in F , arithmetic operations and square roots. Such numbers are algebraic in F , and their degrees over F are some powers of two. However, the converse of the later result is false. So if a number is the solution of an irreducible polynomial of degree three, it can be established that it is not constructible. But if the degree of this polynomial is four, we can not decide. This is why a stronger result is generally needed. This result is a consequence of Galois theory: an algebraic number on F is constructible if and only if the splitting field of its minimal polynomial; P , is an extension of degree 2^m for some m over F . The degree of this extension, 2^m , is also the cardinal of the Galois group of P [12].

It is important to notice that, in general, the considered problems are generic, that is, the coordinates of points in \mathcal{B} are parameters. Furthermore, proving that a problem is RC-solvable consists in showing that whatever the values of the parameters, the solutions are RC-constructible. If one wants to prove that a problem is not RC-solvable, it is enough to compute a counterexample, and if one wants to prove RC-constructibility, the generic problem has to be solved. Proving RC-constructibility leads in general to heavier computations. It is also worth to mention that there are some constructive algebraic methods, that is, methods which give formulas expressing the solutions. To our knowledge the first method is due to Lebesgue [5] and has been implemented by our team in 1992 (see the report [1], unfortunately not published). A more recent and more tractable method is described in [3]. Both methods are usable to solve problems with equations of degree at most four and are impracticable for equations of degree 8. Geometric construction can be extracted from the algebraic formula but these constructions are very complicated, and they are not appealing [10].

2.2 Triangle problems

In the folklore of geometric constructions, problems around triangles are often considered. For instance, William Wernick proposed in 1982 to consider all the problems consisting in constructing a triangle ABC given three characteristic points among the points

- A, B, C , themselves and their circumcenter O ;
- M_a, M_b, M_c, G : the side midpoints and gravity center ;
- H_a, H_b, H_c, H : three feet of altitudes and orthocenter ;
- T_a, T_b, T_c, I : three feet of the internal angle bisectors, and incenter.

More recently, Harold Connelly completed this framework by adding the possibility of considering 4 more points :

- E_a, E_b, E_c the midpoints of H and respectively A, B and C ; and N the center of the nine-points circle i.e. the circle passing by $E_a, E_b, E_c, M_a, M_b, M_c, H_a, H_b$ and H_c .

Wernick and Connelly drew up the lists of all not trivial problems up to some symmetries giving 139 distinct problems for Wernick corpus and 140 distinct problems for Connelly corpus. Wernick problems are presented in Table 1 and Connelly's ones in Table 2 where each problem is given with its status and with a bibliographic reference for the most recent results. The status is either **Solvable**, **Unsolvable**, **Redundant** or **Locus-restricted**.

Problems with a **R** or **L** status refer to over-constrained problems but in two different manners. The three points of each problem must be chosen arbitrarily, in other words each point has two degrees of freedom. But for some problem, one point can be constrained by the two others. In problem W3, which is the third problem in Wernick list, point M_c , the middle point of AB , is completely defined by the two other points A and B , so M_c is redundant, M_c can not be freely chosen. In problem W1, O must lie on the perpendicular bisector of segment

AB . If it does not, there is no solution. If it does, the set of solutions is infinite. So, if one point has zero degree of freedom, the problem is redundant, if it has one degree of freedom, the problem is locus restricted.

This list served as a benchmark for automated geometric construction [7]. We developed an automatic method [10] able to prove (by giving counterexamples chosen arbitrary) the RC-unconstructibility of all problems in Wernick corpus with status U . We also proved, but not automatically, that problems W108 and W119 are RC-constructible. But trying to use our method for Connelly corpus, we discover that for eight problems (whose marked with an asterisk on Table 2, six for unconstructibility and two for constructibility) using a naïve algebraic method does not work with a standard 2015 computer (say with a standard Intel i7 processor and 16Gb of memory). But we manage to completely treat the corpus by preprocessing the problems using geometric knowledge as explained in the next section. Notice that among all the problems in that list which were not solved by Connelly, problem C81 is the only one which is RC-constructible. We prove it by considering the generic case and using a special pre-treatment as explained below. Unfortunately, the resulting polynomials are too complicated for hoping a readable geometric construction.

1. A, B, O	L	36. A, M_b, T_c	S	71. O, G, H	R	106. M_a, H_b, T_c	U [8]
2. A, B, M_a	S	37. A, M_b, I	S	72. O, G, T_a	U [8]	107. M_a, H_b, I	U [8]
3. A, B, M_c	R	38. A, G, H_a	L	73. O, G, I	U [8]	108. M_a, H, T_a	S [10]
4. A, B, G	S	39. A, G, H_b	S	74. O, H_a, H_b	U [8]	109. M_a, H, T_b	U [11]
5. A, B, H_a	L	40. A, G, H	S	75. O, H_a, H	S	110. M_a, H, I	U [11]
6. A, B, H_c	L	41. A, G, T_a	S	76. O, H_a, T_a	S	111. M_a, T_a, T_b	U [11]
7. A, B, H	S	42. A, G, T_b	U [8]	77. O, H_a, T_b	U [10]	112. M_a, T_a, I	S
8. A, B, T_a	S	43. A, G, I	S [8]	78. O, H_a, I	U [10]	113. M_a, T_b, T_c	U [10]
9. A, B, T_c	L	44. A, H_a, H_b	S	79. O, H, T_a	U [8]	114. M_a, T_b, I	U [8]
10. A, B, I	S	45. A, H_a, H	L	80. O, H, I	U [8]	115. G, H_a, H_b	U [8]
11. A, O, M_a	S	46. A, H_a, T_a	L	81. O, T_a, T_b	U [10]	116. G, H_a, H	S
12. A, O, M_b	L	47. A, H_a, T_b	S	82. O, T_a, I	S [8]	117. G, H_a, T_a	S
13. A, O, G	S	48. A, H_a, I	S	83. M_a, M_b, M_c	S	118. G, H_a, T_b	U [10]
14. A, O, H_a	S	49. A, H_b, H_c	S	84. M_a, M_b, G	S	119. G, H_a, I	S [10]
15. A, O, H_b	S	50. A, H_b, H	L	85. M_a, M_b, H_a	S	120. G, H, T_a	U [8]
16. A, O, H	S	51. A, H_b, T_a	S	86. M_a, M_b, H_c	S	121. G, H, I	U [8]
17. A, O, T_a	S	52. A, H_b, T_b	L	87. M_a, M_b, H	S [8]	122. G, T_a, T_b	U [10]
18. A, O, T_b	S	53. A, H_b, T_c	S	88. M_a, M_b, T_a	U [8]	123. G, T_a, I	U [10]
19. A, O, I	S	54. A, H_b, I	S	89. M_a, M_b, T_c	U [8]	124. H_a, H_b, H_c	S
20. A, M_a, M_b	S	55. A, H, T_a	S	90. M_a, M_b, I	U [11]	125. H_a, H_b, H	S
21. A, M_a, G	R	56. A, H, T_b	U [8]	91. M_a, G, H_a	L	126. H_a, H_b, T_a	S
22. A, M_a, H_a	L	57. A, H, I	S [8]	92. M_a, G, H_b	S	127. H_a, H_b, T_c	U [10]
23. A, M_a, H_b	S	58. A, T_a, T_b	S [8]	93. M_a, G, H	S	128. H_a, H_b, I	U [10]
24. A, M_a, H	S	59. A, T_a, I	L	94. M_a, G, T_a	S	129. H_a, H, T_a	L
25. A, M_a, T_a	S	60. A, T_b, T_c	S	95. M_a, G, T_b	U [8]	130. H_a, H, T_b	U [8]
26. A, M_a, T_b	U [8]	61. A, T_b, I	S	96. M_a, G, I	S [8]	131. H_a, H, I	S [8]
27. A, M_a, I	S [8]	62. O, M_a, M_b	S	97. M_a, H_a, H_b	S	132. H_a, T_a, T_b	U [10]
28. A, M_b, M_c	S	63. O, M_a, G	S	98. M_a, H_a, H	L	133. H_a, T_a, I	S
29. A, M_b, G	S	64. O, M_a, H_a	L	99. M_a, H_a, T_a	L	134. H_a, T_b, T_c	U [10]
30. A, M_b, H_a	L	65. O, M_a, H_b	S	100. M_a, H_a, T_b	U [8]	135. H_a, T_b, I	U [10]
31. A, M_b, H_b	L	66. O, M_a, H	S	101. M_a, H_a, I	S	136. H, T_a, T_b	U [10]
32. A, M_b, H_c	L	67. O, M_a, T_a	L	102. M_a, H_b, H_c	L	137. H, T_a, I	U [10]
33. A, M_b, H	S	68. O, M_a, T_b	U [8]	103. M_a, H_b, H	S	138. T_a, T_b, T_c	U [13]
34. A, M_b, T_a	S	69. O, M_a, I	S	104. M_a, H_b, T_a	S	139. T_a, T_b, I	S
35. A, M_b, T_b	L	70. O, G, H_a	S	105. M_a, H_b, T_b	S		

Table 1: Wernick problems represented by their three characteristic points, their status by a letter in $\{L, R, S, U\}$, and for the recently solved problems, a bibliographic reference.

1. A, B, E_a	S	36. A, M_a, N	S	71. E_a, H, T_b	U	106. E_a, M_b, T_c	U
2. A, B, E_c	S	37. A, M_b, N	S	72. E_a, H_a, H_b	S	107. E_a, N, O	S
3. A, B, N	S	38. A, N, O	S	73. E_a, H_a, I	S	108. E_a, N, T_a	S
4. A, E_a, E_b	S	39. A, N, T_a	U	74. E_a, H_a, M_a	L	109. E_a, N, T_b	U
5. A, E_a, G	S	40. A, N, T_b	U	75. E_a, H_a, M_b	S	110. E_a, O, T_a	U
6. A, E_a, H	R	41. E_a, E_b, E_c	S	76. E_a, H_a, N	L	111. E_a, O, T_b	U
7. A, E_a, H_a	L	42. E_a, E_b, G	S	77. E_a, H_a, O	S	112. E_a, T_a, T_b	U
8. A, E_a, H_b	L	43. E_a, E_b, H	S	78. E_a, H_a, T_a	L	113. E_a, T_b, T_c	U
9. A, E_a, I	S	44. E_a, E_b, H_a	S	79. E_a, H_a, T_b	U	114. G, H, N	R
10. A, E_a, M_a	S	45. E_a, E_b, H_c	S	80. E_a, H_b, H_c	L	115. G, H_a, N	S
11. A, E_a, M_b	S	46. E_a, E_b, I	U	81. E_a, H_b, I	S *	116. G, I, N	U
12. A, E_a, N	S	47. E_a, E_b, M_a	L	82. E_a, H_b, M_a	L	117. G, M_a, N	S
13. A, E_a, O	S	48. E_a, E_b, M_c	S	83. E_a, H_b, M_b	S	118. G, N, O	R
14. A, E_a, T_a	S	49. E_a, E_b, N	L	84. E_a, H_b, M_c	S	119. G, N, T_a	U
15. A, E_a, T_b	U	50. E_a, E_b, O	S	85. E_a, H_b, N	L	120. H, H_a, N	S
16. A, E_b, E_c	S	51. E_a, E_b, T_a	U	86. E_a, H_b, O	S	121. H, I, N	U *
17. A, E_b, G	S	52. E_a, E_b, T_c	U	87. E_a, H_b, T_a	U	122. H, M_a, N	S
18. A, E_b, H	S	53. E_a, G, H	S	88. E_a, H_b, T_b	U	123. H, N, O	R
19. A, E_b, H_a	S	54. E_a, G, H_a	S	89. E_a, H_b, T_c	U	124. H, N, T_a	U
20. A, E_b, H_b	L	55. E_a, G, H_b	S	90. E_a, I, M_a	S	125. H_a, H_b, N	L
21. A, E_b, H_c	S	56. E_a, G, I	U	91. E_a, I, M_b	U	126. H_a, I, N	S
22. A, E_b, I	U	57. E_a, G, M_a	S	92. E_a, I, N	S *	127. H_a, M_a, N	L
23. A, E_b, M_a	S	58. E_a, G, M_a	S	93. E_a, I, O	U *	128. H_a, M_b, N	L
24. A, E_b, M_b	S	59. E_a, G, N	S	94. E_a, I, T_a	U	129. H_a, N, O	S
25. A, E_b, M_c	S	60. E_a, G, O	S	95. E_a, I, T_b	U	130. H_a, N, T_a	S
26. A, E_b, N	S	61. E_a, G, T_a	U	96. E_a, M_a, M_b	L	131. H_a, N, T_b	U
27. A, E_b, O	S	62. E_a, G, T_b	U	97. E_a, M_a, N	R	132. I, M_a, N	S
28. A, E_b, T_a	U	63. E_a, H, H_a	L	98. E_a, M_a, O	S	133. I, N, O	U *
29. A, E_b, T_b	U	64. E_a, H, H_b	L	99. E_a, M_a, T_a	S	134. I, N, T_a	U *
30. A, E_b, T_c	U	65. E_a, H, I	S	100. E_a, M_a, T_b	U	135. M_a, M_b, N	L
31. A, G, N	S	66. E_a, H, M_a	S	101. E_a, M_b, M_c	S	136. M_a, N, O	S
32. A, H, N	S	67. E_a, H, M_b	S	102. E_a, M_b, N	L	137. M_a, N, T_a	S
33. A, H_a, N	S	68. E_a, H, N	S	103. E_a, M_b, O	S	138. M_a, N, T_b	U
34. A, H_b, N	S	69. E_a, H, O	S	104. E_a, M_b, T_a	U	139. N, O, T_a	U *
35. A, I, N	U	70. E_a, H, T_a	S	105. E_a, M_b, T_b	U	140. N, T_a, T_b	U *

Table 2: Connelly corpus [2]. A status written in boldface indicates that the result was not known. An asterisk means that a geometric preprocessing was needed. Notice that problems C81 and C92 were treated, after the automatic process, by hand considering parameters.

2.3 From geometry to algebra

Translating a problem from geometry to algebra requires to choose coordinates for the involved points and then to translate the geometric constraints into polynomial equations.

Concerning the first issue, it is easy to see that one of the three points can be chosen to be at coordinates $(0, 0)$ and a second one on the x -axis for instance. Actually, since the problems are invariant by similarities, or in other words, the scale can be chosen, this second point can put at coordinates $(v, 0)$ where v has some arbitrarily chosen not zero value (for instance 10). For the third point, if we want to prove the RC-constructibility, we set it at coordinates (a, b) where a and b are some parameters, and, on the contrary, if we want to check the RC-unconstructibility, the coordinates of the third point are integers more or less randomly chosen : we just verify that there is at least one solution with that choice of coordinates. Notice that we only consider integers as coordinates because, first, we want to perform exact computations and, second, in our case, we consider some transcendental field extension of \mathbb{Q} for the computation of the Galois groups of the produced polynomial.

For the second issue, a first idea is to exploit some static algebraic definitions of the char-

Point	Equation	Term for equations
A	$A = A$	
M_a	$x_{M_a} = \frac{x_B + x_C}{2}$ $y_{M_a} = \frac{y_B + y_C}{2}$	Ma=midpoint(B,C)
G	$x_G = \frac{x_A + x_B + x_C}{3}$ $y_G = \frac{y_A + y_B + y_C}{3}$	G=bar(A,B,C)
H_a	$\overrightarrow{AH_a} \cdot \overrightarrow{BC} = 0$ $\det(\overrightarrow{BH_a}, \overrightarrow{BC}) = 0$	perpend(A,Ha,B,C) collinear(B,Ha,C)
H	$\overrightarrow{AH} \cdot \overrightarrow{BC} = 0$ $\overrightarrow{BH} \cdot \overrightarrow{AC} = 0$	perpend(A,H,B,C) perpend(B,H,A,C)
T_a	$\det(\overrightarrow{AB}, \overrightarrow{AT_a}) \cdot \ AC\ = \det(\overrightarrow{AT_a}, \overrightarrow{AC}) \cdot \ AB\ $ $\det(\overrightarrow{BT_a}, \overrightarrow{BC}) = 0$	onAngleBisector(Ia,A,B,C) collinear(B,Ia,C)
I	$\det(\overrightarrow{AB}, \overrightarrow{AI}) \cdot \ AC\ = \det(\overrightarrow{AI}, \overrightarrow{AC}) \cdot \ AB\ $ $\det(\overrightarrow{BC}, \overrightarrow{BI}) \cdot \ BA\ = \det(\overrightarrow{BI}, \overrightarrow{BA}) \cdot \ BC\ $	onAngleBisector(I,A,B,C) onAngleBisector(I,B,C,A)
E_a	$x_{E_a} = \frac{x_A + x_H}{2}$ $y_{E_a} = \frac{y_A + y_H}{2}$	Ea=midpoint(A,H)

Table 3: Usual definitions of some characteristic points. These formulas can be straightforwardly translated into polynomial equations

acteristic points such as depicted on Table 3. This table shows equations for expressing the statement about points but, for the sake of simplicity, we use the term in the third column instead of the equations. Unfortunately, this leads sometimes to unnecessary complicated polynomials. For instance, in problem C81, one of the given characteristic points is the incenter I : it can be defined as the intersection of the two inner bisectors of respective angles $\angle ABC$ and $\angle BAC$, but since point H_b which is on line AC is given, it is better to define point I as the intersection of the two inner bisectors of respective angles $\angle BCH_b$ and $\angle BAH_b$ where lesser unknowns are involved. In this paper, we propose a heuristic to systematize this trick.

3 Geometric reasoning for algebraic systems

In this section, we show how geometry rules are applied in the process of determining the status of a problem. We only mention rules about geometric reasoning. Other rules are also needed for term management, for instance, permutation rules. For a problem p , the pipeline is as follows:

1. Determining a set of problems S from the corpus that are geometrically equivalent to p
2. For each $p_i \in S$
 - (a) set up a *default* system of equations for p_i
 - (b) replace some equations by simpler ones according to some geometric rules
3. Choosing one of these systems among the *easiest* to solve

This pipeline outputs a system that will be triangularized. Then algebraic reasoning will be performed in order to establish the status. This section explains each step of the pipeline.

3.1 Reduction and constructibility classes

In papers presenting Wernick and Connelly corpora, equivalent problems up to some basic symmetries have been removed so that only one representative occurs in the list. For example, the first Wernick problem is A, B, O which is equivalent to B, C, O or A, C, O . However some problems are equivalent in terms of constructibility [9]. For example, problem A, O, G (W13) has the same properties as the problem A, O, H (W16). Indeed, it is known that points O, G, H are located on the Euler line and are linked by the relationship $HO = \frac{3}{2}HG$. There is thus a simple construction to go from W13 to W16 and conversely. So if one of the problems can be constructed the others can be constructed as well. It is the same for unconstructibility.

However, in terms of symbolic processing, these two problems are not equivalent. Recall that the degree of a polynomial is the highest degree of its monomials. Thus the degree of the polynomial $x_A^2y_B + x_Ay_B$ is three. The degree of a system is the product of the degrees of polynomials.

Consider equations on Table 3. The two equations for point H are two polynomials of degree two with four unknowns each (the coordinates of points B and C). While the two equations for point G have degree one and each contain two unknowns.

The first stage in the treatment of a new problem p is to identify any problem p_i such that there is a known construction that gives p_i from p . For this, some replacement rules are used. We give a list of these rules where x denotes any point A, B or C . Of course, these rules are made to fit exactly Connelly corpus and new rules must be added for other corpora. We designed the rules by expressing interesting properties on the specific points used in this corpus. In [9], a method based on a knowledge base is described in order to discover these rules in the case of Wernick corpus and then to gather problems in that list in classes. Rule 1 expresses the symmetry of E_x and M_x from center N , indeed, for any x , segment E_xM_x is a diameter of the Euler circle. Rule 2 comes from the well-known metric relations among points G, H, O, N which all lie on the Euler line. Rule 3 translates the fact that E_x is the midpoint of x and H .

Rule 1: Replace E_x by M_x (and conversely) if N is also in the statement

Rule 2: If there are two points among G, H, O, N then each of these two points can be replaced by one of the two others

Rule 3: If there are two points among x, E_x, H then each of these two points can be replaced by the third

With problem C53 : E_a, H, G , the application of rule 2 leads to the 5 following problems:

E_a, H, O : C69	E_a, H, N : C68
E_a, G, O : C60	E_a, G, N : C59
E_a, N, O : C107	

Then, by applying rule 1, it comes:

$E_a, H, N \rightarrow M_a, H, N$: C122
$E_a, G, N \rightarrow M_a, G, N$: C117
$E_a, N, O \rightarrow M_a, N, O$: C136

Rule 3 then gives:

$$\begin{array}{ll}
E_a, H, O \rightarrow A, H, O : \text{W16} & E_a, H, O \rightarrow A, E_a, O : \text{C13} \\
E_a, H, G \rightarrow A, H, G : \text{W49} & E_a, H, G \rightarrow A, E_a, G : \text{C5} \\
E_a, H, N \rightarrow A, H, N : \text{C32} & E_a, H, N \rightarrow A, E_a, N : \text{C12}
\end{array}$$

From these new triples, rule 2 applies again:

$$\begin{array}{ll}
M_a, O, G : \text{W63} & M_a, O, H : \text{W66} \\
M_a, G, H : \text{W93} & A, O, G : \text{W13} \\
A, O, N : \text{C38} & A, G, N : \text{C31}
\end{array}$$

Thereby, from problem C53, a set of 21 problems are identified as similar. The status of all these problems are the same. Problem A, H, N seems to be one of the easiest to solve since point A is already given. Indeed, a construction can be easily found:

1. set E_a the midpoint of segment AH and M_a the symmetric of E_a from N
2. draw circle \mathcal{C} , the Euler circle, with center N and radius NE_a
3. line AH intersects \mathcal{C} at E_a and H_a
4. circle of diameter AH intersects \mathcal{C} at H_b
5. point B is on line HH_b and line H_aM_a
6. point C is the symmetric of B from M_a

Consequently, all these 21 problems are constructible and their constructions easily derive from the reduction rules.

In Wernick list, problem O, G, T_a (W72) is said to be not constructible. By applying rule 2, this problem is in a set containing:

$$\begin{array}{l}
O, H, T_a : \text{W79} \\
O, N, T_a : \text{C139} \\
G, H, T_a : \text{W120} \\
H, N, T_a : \text{C124} \\
G, N, T_a : \text{C119}
\end{array}$$

If W72 is not constructible, it is the same for the problems of the whole set.

In such a set, the aim is now to find the simplest problem in terms of symbolic expression. To achieve this, a system of equations is set for each problems. One system is selected according to some combinatorial criteria. This latter will be analyzed to establish the status of all the problems of the set. But first, let us examine how to design the system.

3.2 Replacing equations

Each point of the statement gives rise to two equations. For point H for example, we have:

$$\begin{array}{l}
\text{perpend}(H, A, B, C) \\
\text{perpend}(H, B, A, C)
\end{array}$$

Each algebraic equation has degree two and eight monomials. For point H_a , equations are:

$$\begin{array}{l}
\text{perpend}(H_a, A, B, C) \\
\text{collinear}(H_a, B, C)
\end{array}$$

Again, each of these two polynomials are of degree 2 and they have respectively eight and six monomials. We will call these equations for each point the *default* equations.

But we can do better. If point H and H_a are together in a statement, we could express that A, H and H_a are collinear. The equation for `collinear(A,Ha,H)` is of degree one and has three terms. It could replace one equation among the four equations above of degree two.

Rather than considering each point independently, a simpler system can be built by considering several points of the statement in the equations. Each of these equations replaces a default equation. However, the replacement should be done with caution. Consider problem W112 : I, T_a, M_a . The default system could be this one:

For I : (eq1) `onAngleBisector(I,A,B,C)`
 (eq2) `onAngleBisector(I,B,A,C)`
 For T_a : (eq3) `onAngleBisector(Ta,A,B,C)`
 (eq4) `collinear(Ta,B,C)`
 For M_a : (eq5, eq6) `Ma=midpoint(A,B)`

Specific equations for that problem are possible. We know that points A, I and T_a are collinear and that points T_a, M_a and B (or C) are collinear:

(eq7) `collinear(Ta,I,A)`
 (eq8) `collinear(Ta,Ma,B)`

Equations (eq7) and (eq8) could replace two equations in the default system. Since (eq7) involves I and T_a , it has to replace an equation among (eq1), (eq2), (eq3) or (eq4). If (eq4) is selected, the system becomes misconstrained since (eq7) is a consequence of (eq1) and (eq3) but also of (eq2) and (eq3). So, (eq7) has to take the place of (eq3). This operation is relevant because an equation of degree four with more than ten monomials is simplified into a degree one polynomial with three monomials.

3.3 Rules for setting up a system of equations

Each polynomial system is made in two steps. First, a default system is built. Next, a rule-based system applies basic geometric rules in order to replace some equations by simpler ones.

The first step consists in choosing the default equations. For most points, they are clearly defined. But some centers of triangle like I, O, N, H are defined as the meeting point of three lines and two of them could be chosen for expressing the intersection. Centroid G is not concerned since to get simpler equations it is better to express it as the barycenter of A, B, C than as the intersection of the medians.

Wernick and Connelly built their list by putting forward point A first and then point B . For example, problem C, M_c, M_b is not in the list and is represented by the symmetric equivalent A, M_a, M_b . Then, for most centers, the best way to set up the equations is to put forward points A and B . For instance, point O is given by the two equations:

`onPerpBisector(O,A,B)`
`onPerpBisector(O,A,C)`

But for incenter I , equations must be chosen carefully. The point I lies on angle bisectors that are expressed by a polynomial of degree four that computer algebra systems have trouble triangularizing. If point, say, M_b is given, instead of expressing `onAngleBisector(I,A,B,C)` we can use `onAngleBisector(I,A,B,Mb)`. This kind of transformation is performed by the rule-based system used in the second step (see below). But the latter relies on equations of the default system and the change can be done only if `onAngleBisector(I,C,A,B)` appears.

For point I , we apply the following rule to select our two bisectors with x, y and z three different points among A, B and C :

```

if neither Tx nor Hx nor Mx appear in the problem
then insert onAngleBisector(I,y,x,z)

```

In problem O, M_a, I (W69) applying this rule leads to the two following equations for I :

```

onAngleBisector(I,B,A,C)
onAngleBisector(I,C,B,A)

```

So, the equation `onAngleBisector(I,A,B,C)` is avoided. An example below shows how this will allow to introduce point M_a in both angle bisector equations.

In the second step, the default system is simplified according to some rules implemented by a rule-based system (RBS). The inputs are the three given points and the six terms expressing the polynomials. These are the nine facts for the RBS. As usual with RBS, the set of rules can be dynamically extended. The set of rules is given in Table 4. Each rule is preceded by a comment indicating its role in a specific case. The first three sets of rules are trivial. The fourth set corresponds to more sophisticated theorems. For instance, rule IVc allows to translate problem C81 : E_a, I, H_b into a system that can be triangularized within a few seconds. If the default system is used for that problem, the triangularization can not be achieved in a computer with a memory of 16Gb. Each rule applies on more than 10 problems. For instance, rule IVa simplifies 13 problems (C12, C26, ...).

In the rules, a premise such M_a means M_a is a given point. As above, x, y and z stands for any of the points A, B and C but differently instantiated i.e. if x is point A then y is either B or C . P is any point of the statement. When a fact has to be named, we use the notation `f <- fact` meaning that the variable `f` corresponds to `fact`. The instruction `retract` removes a fact from the base while `insert` adds a fact. Symbol `_` is a wildcard that matches all possible values. Comments are introduced by a semicolon. The function `mostPresent(l)` returns a point among A, B and C that is not in the list `l` and which appears the most in the fact. The goal of the function is to keep the system of equations as triangular as possible.

In problem W131: H_a, H, I default equations are replaced in the following way:

```

For  $H_a$ :  perpend(A,Ha,B,C)  $\xrightarrow{\text{Rules I}}$  collinear(A, H,Ha)
           collinear(B,Ha,C)
For  $H$ :    perpend(A,H,B,C)
           perpend(B,H,A,C)
For  $I$ :    onAngleBisector(I,B,A,C)  $\xrightarrow{\text{Rules III}}$  onAngleBisector(I,B,A,Ha)
           onAngleBisector(I,C,A,B)  $\xrightarrow{\text{Rules III}}$  onAngleBisector(I,C,A,Ha)

```

Notice that substitutions for equations of point I can be made because the two angle bisectors were chosen in such a way that H_a, M_a or T_a can be introduced. Since the points of the statement have numerical coordinates, new equations have four unknowns instead of six.

3.4 Selecting a system

Now a set of statements similar to the original one is built and a algebraic system is associated with each of these statements. The best algebraic system has to be chosen, that is the easiest to triangularize by means of regular chains. However, the selection criterion is difficult to determine. We use a heuristic that mainly relies on the degree of the system.

Among the 21 problems of section 3.1 derived from problem E_a, H, G , seven problems are of degree four, 11 of degree 16 and three problems are of degree 64. To select a system among those having a minimal degree, we use two criteria: the average number of unknowns per polynomial

```

; Rules I
; if H and Ha are given then express that A H Ha are collinear
if H, Hx, f <- perpend(x,Hx,y,z)
then retract f
    insert collinear (x,H,Hx)

; similar rules for
; I and Tx : collinear(x,I,Tx) ; Ex and Hx : collinear(x,Ex,Hx)
; Ex and H : collinear(x,Ex,H) ; G and Mx : collinear(x,G,Mx)

; Rules II
; if Ha and Ta are given then express that Ha Ta and either B or C are collinear
if Hx, Tx, f <- onAngleBisector(Tx,x,-,.) or f <- perpend(x,Hx,-,.) , y = mostPresent([x])
then collinear(Hx,y,Tx)

; similar rules with Hx and Mx ; Tx and Mx

; Rules III
; if Ha is given and onAngleBisector(I,C,A,B) then replace this latter equation
; with onAngleBisector(I,C,A,Ha)
if Hx, f <- onAngleBisector(P,y,x,z)
then retract f
    insert onAngleBisector(P,y,x,Hx)

; similar rules with Mx instead of Hx ; Tx instead of Hx

; Rules IV
; A few rules for useful theorems

; Rule IVa. if Ea and N are given then Ma is the symmetric of Ea from N
if Ex, N, f <- perpend(x,Ex,y,z)
then retract perpend(x,Ex,y,z)
    insert midpoint(y,z) - (2*N-Ex) ; meaning that midpoint(y,z)=(2*N-Ex)

; Rule IVb. if Ea and Hb then EaHb = EaA and B H Hb are collinear
if Ex, Hy, f1,f2 <- two equations involving Ex
then retract f1, retract f2
    insert onPerpBisector(Ex,x,Hy)
    insert collinear(2*Ex-x,Hy,y)

; Rule IVc. midpoint of segment MaA is the midpoint of segment EaO
if Ex, O, f1,f2 <- two equations involving Ex
then retract f1, retract f2
    insert midpoint(midpoint(y,z), x) - midpoint(Ex,O) ; two equations

```

Table 4: Substitution rules for equations

and the total number of monomials in the system. Our heuristic estimated that A, O, G is the best with a degree four and an average of 2.66 unknowns per polynomials. With Maple 18 using an Intel®Core i5, it takes 0.04 sec. Besides problem E_a, H, O has a degree 64, an average of 5.33 unknowns and 40 monomials. The status is given in 0.66 s.

For all of the problems of the second class presented in 3.1, triangularization by regular chains shows that the first equation to solve is a polynomial of degree six. Three of them have a degree 32 and theirs status are determined within less than 0.5 sec. The other three have a degree 128 and their computation take between 2 and 2.5 sec.

4 Conclusion

The issue of constructibility by ruler and compass of certain geometrical figures is one of the oldest mathematical questions. Algebra and results of Galois give results about constructibility which is not always easily usable if one wants an effective geometric construction, but they are useful if one wants to know if a problem is RC-unconstructible.

Recent progresses in implementation of regular chains and Galois theory allowed us to successfully check the whole Wernick corpus. Unfortunately, our automatic method was not powerful enough to deal with Connelly corpus. In this paper, we show how to use geometry in order overcomes this issue for RC-unconstructible problems. To this end, we use a rule based system to set up the algebraic system coming from the geometrical one. Some rules are basic, others have been added for specific set of problems that have not been solvable otherwise. New rules can further enrich the database but the ones proposed here are sufficient for making every problem treated within seconds. According to the order of rules application, the system may be different but in any case the resulting system is quicker to triangularize than the initial system.

This work illustrates our approach aiming at marrying geometry with algebra, analysis or graphs (See for instance [4]). Other corpora could be studied: for instance by considering Feuerbach points which are related to Euler circle, incircle and excircles. Besides, some issues remain open: could RC-unconstructible problems be constructed using other tools such, for instance, origami foldings which allow to solve polynomials of degree three and four ? Is it possible to automatically simplify the ruler and compass construction driven by algebraic expression ?

References

- [1] Guoting Chen. Les constructions à la règle et au compas par une méthode algébrique. Technical Report Master thesis, Université Louis Pasteur, 1992.
- [2] Harold Connelly. An extension of triangle constructions from located points. *Forum Geometricorum*, 9:109–112, 2009.
- [3] Xiao-Shan Gao and Shang-Ching Chou. Solving geometric constraint systems. II. A symbolic approach and decision of Rc-constructibility. *Computer Aided Design*, 30(2):115–122, 1998.
- [4] Rémi Imbach, Pascal Schreck, and Pascal Mathis. Leading a continuation method by geometry for solving geometric constraints. *Computer-Aided Design*, 46:138–147, 2014.
- [5] Henri Lebesgue. *Leçons sur les constructions géométriques*. Gauthier-Villars, Paris, 1950. in French, re-edition by Editions Jacques Gabay, France.
- [6] Vesna Marinkovic, Predrag Janicic, and Pascal Schreck. Computer theorem proving for verifiable solving of geometric construction problems. In Francisco Botana and Pedro Quaresma, editors, *Automated Deduction in Geometry - 10th International Workshop, ADG 2014, Coimbra, Portugal, July 9-11, 2014, Revised Selected Papers*, volume 9201 of *Lecture Notes in Computer Science*, pages 72–93. Springer, 2014.
- [7] Vesna Marinković and Predrag Janičić. Towards understanding triangle construction problems. In J. Jeuring et al., editor, *Intelligent Computer Mathematics - CICM 2012*, volume 7362 of *Lecture Notes in Computer Science*. Springer, 2012.
- [8] Leroy F. Meyers. Update on William Wernick’s „triangle constructions with three located points”. *Mathematics Magazine*, 69(1):46–49, 1996.
- [9] Pascal Schreck, Vesna Marinkovic, and Predrag Janicic. Constructibility classes for triangle location problems. *submitted*, 2015.

- [10] Pascal Schreck and Pascal Mathis. Rc-constructibility of problems in wernick's list. In Francisco Botana and Pedro Quaresma, editors, *Proceedings of the 10th Int. Workshop on Automated Deduction in Geometry*, volume TR 2014/01, pages 85–104. University of Coimbra, 2014.
- [11] Eckehard Specht. Wernicks Liste (in Deutsch). *Only available on the web*, 2009. <http://hydra.nat.uni-magdeburg.de/wernick/>.
- [12] Ian Stewart. *Galois Theory (third edition)*. Chapman Hall, 2003.
- [13] Alexey V. Ustinov. On the construction of a triangle from the feet of its angle bisectors. *Forum Geometricorum*, 9:279–280, 2009.
- [14] William Wernick. Triangle constructions with three located points. *Mathematics Magazine*, 55(4):227–230, 1982.