Solving Geometric Construction Problems Supported by Theorem Proving

Vesna Marinković[1], Predrag Janičić[1], and Pascal Schreck[2]

[1] Faculty of Mathematics, University of Belgrade, Serbia
[2] ICube, UMR CNRS 7357, Université de Strasbourg, France

Abstract. Over the last sixty years, a number of methods for automated theorem proving in geometry, especially Euclidean geometry, have been developed. Almost all of them focus on universally quantified theorems. On the other hand, there are only few studies about logical approaches of geometric constructions. Consequently, automated proving of $\forall\exists$ theorems, that correspond to geometric construction problems, have seldom been studied. In this paper, we present a formal logic framework describing the traditional four phases process of geometric construction solving. It leads to automated production of constructions with corresponding human readable correctness proofs. To our knowledge, this is the first study in that direction. In this paper we also discuss algebraic approaches for solving ruler-and-compass construction problems. There are famous problems showing that it is often difficult to prove non-existence of constructible solutions for some tasks. We show how to put into practice known methods based on algebra and, in particular, field theory, to prove RC-constructibility in the case of problems from Wernick's list.

1 Introduction

In spite of a long tradition of straightedge and compass constructions [3] , automation and mechanization of *solving of geometric construction problems* has been barely touched in computer science. As far as we know, except for works described in [23, 5, 25, 20, 14], for a geometric treatment, and in [6, 29], for an algebraic point of view, results the closest to this subject are about geometric constraint solving in CAD [1, 22, 4, 8] (see [16] for a recent survey), but there, the challenges are quite different [25, 24]. Moreover, most of the existing methods for solving construction problems do not consider proving correctness of solutions. Consider, for instance, the method designed by Gulwani [14]: this method is derived from general methods for testing and synthesizing pieces of software. It finds a formal construction by using a probabilistic approach of having a particular solution which serves to guide a search in a big space of formal functional terms. For such a method, a proof of correctness is really needed.

[3] The English word *ruler* designates a tool with measurement in opposition to *straightedge*. In this paper, however, we will conform to the habits and use the terms *ruler-and-compass constructibility* or resolvability, in short RC-constructibility or RC-resolvability, for straightedge and compass constructibility or resolvability [29].

Some studies about the foundations of geometry also consider geometric constructions in order to define constructive geometry through the elimination of quantifiers and the use of functional symbols (see [30, 31] for instance). But the contexts are very different: we consider here RC-constructions within the classical Euclidean plane, while in constructive geometry the aim is to define a logical framework where all considered objects correspond to ground functional terms (or in other words, the ground set of its initial model is the set of points RC-constructible from $\{O, I\}$).

This limited interest is even more surprising given that solving geometric construction problems links two important fields of computer science applied to geometry:

- automated theorem proving, with geometry as one of the most successful domains since the seminal work by Gelerntner [11];
- dynamic geometry software, which provoked huge changes in educational practices in geometry by effective visualizations and rewarding experimentations.

Moreover, both these fields have deep connections with construction problems. A lot of methods for automated theorem proving in geometry rely on basic geometric constructions, and constructions performed within dynamic geometry tools typically correspond to RC-constructions. Still, links between automated solving of construction problems with automated theorem proving or dynamic geometry software have been hardly explored.

From the logical point of view, solving a geometric construction problem requires proving a theorem of the form $\forall X \exists Y. \Psi(X, Y)$ in an intuitionist way. The witness for Y that is found during such proof represents a *construction of a solution* and must involve only points that are RC-constructible from the set X. In other words, the task is, given a declarative specification of the required figure $\Psi(X, Y)$, to provide a corresponding— possibly equivalent—procedural specification based on available construction steps. Both directions of this equivalence have to be proved as we will discuss in more details in this paper.

This transformation of a declarative statement into a procedural specification within a formal framework relies on formalization of the tools allowed to perform the construction. Usually, ruler and compass are considered and operations like intersection of lines and circles can be used. However, the folklore of geometric constructions also consider many variations, for instance, by forbidding either ruler or compass, by restricting operations by some tools (for example, collapsible compass or blocked compass), or by allowing new tools like tool for tracing the MacLaurin trisectrix or Origamis. Some of these sets of tools are equivalent to ruler and compass, while MacLaurin trisectrix and Origamis are more powerful. In this paper, we restrict ourselves to RC-constructibility which definition is recalled here:

Definition 1. *Given a finite set of points $\mathcal{B} = \{B_0, \ldots, B_m\}$ in the Euclidean plane, a point P is* RC-constructible from the set $\mathcal{B}$ *if there is a finite set of points $\{P_0, \ldots, P_n\}$ such that $P = P_n$ and every point P_i ($0 \leq i \leq n$) 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 $\{P_0, \ldots, P_{i-1}\}$ and its radius is equal to the distance P_jP_k for some $j < i$ and $k < i$;*
- *any considered line passes through two points from the set $\{P_0, \ldots, P_{i-1}\}$*

For problems involving parameters, the *solution* is in fact a way to construct all solutions. Already in the early 40s, Lebesgue was calling this a *program of construction* [18].

It is important to note that sometimes there is no solution for a given construction problem. The absence of solutions does not necessarily mean that there is no solutions in the Euclidean plane, but no solution of the problem can be constructed using only ruler and compass. Examples are famous classical problems like the circle quadrature problem. It is often difficult to prove such impossibility theorems. Let us note that it suffices to find a counter-example to prove that a problem is RC-unconstructible.

In this paper, we will focus on triangle construction problems and one particular corpus of such problems – Wernick's list [28]. This corpus consists of triangle construction problems with located points, and for each, the task is, given some points X to construct a triangle ABC such that the points X meet the condition $\Psi(X, A, B, C)$. We first discuss a geometrical approach for solving construction problems, with classical „four-phases solutions" and then algebraic approach, both for proving constructibility and unconstructibility. Within the geometry part, we will comment on our wider project: automation of the solving process, accompanied by machine verifiable proofs.

2 Geometrical Approach

In this section we give a rigorous, first-order logic description of classical form of solutions of construction problems. To our knowledge, surprisingly, this is the first such description, despite the fact that construction problems have been around for two and a half millennia. Our rigorous description serves as a basis for a semi-automatic methodology for solving construction problems and generating their solutions, supported by automated theorem provers and formal proofs within proof assistants. To our knowledge, automated and formal proving in the context of automated solving of construction problems were never treated so far.

2.1 Goal

As said above, for a construction problem, roughly said, the task is to prove constructively a theorem of the following form (where X and Y denote vectors of objects—points, lines, rays, etc):

$$\forall X \exists Y . \Psi(X, Y) \tag{1}$$

The above subsumes two claims: that the problem is solvable and that a particular construction (that witnesses $\exists Y.\Psi(X,Y)$) is correct.

Within the problem description, there could be given some constraints imposed on the given objects X. Not all construction problems have solutions: some problems do not have solutions and some problems have solutions only under some additional conditions, not known in advance. So, instead of (1), one typically has to discover $\Phi(X)$ (for the given $\Psi(X,Y)$) and to prove:

$$\forall X.(\Phi(X) \Rightarrow \exists Y.\Psi(X,Y)) \tag{2}$$

The above claims that solution exists under some conditions. But one may claim even more:

$$\forall X.(\Phi(X) \Rightarrow \exists Y.\Psi(X,Y) \ \wedge \ \neg\Phi(X) \Rightarrow \neg\exists Y.\Psi(X,Y)) \tag{3}$$

The above gives a complete characterization of resolvability: it states that solution exists under some conditions Φ and solution does not exist otherwise. The problem is that often conditions for resolvability cannot be expressed only in terms of the given objects X, but have to involve some auxiliary objects (used within the construction).

In solving specific classes of construction problems, some goal conditions may be assumed. For instance, in solving triangle construction problems, an implicit goal condition is that the constructed points A, B, and C are not collinear.

2.2 Constructible Cases and Four-Phases Solutions

Before going to theory, let us bring our esteemed readers back to school and remind them that traditionally, finding a solution of a construction problem passes through the following four phases [7]:

Analysis: One starts from the assumption that certain geometrical objects satisfy the conditions of the problem $\Psi(X,Y)$ (see Section 2.1) and proves properties $Plans(X,Y)$ that enable construction (geometric loci and theorems are used for producing candidates for solutions).

Construction: A construction is based on the analysis, that is, on the procedural (ruler-and-compass) counterpart to the specification $\Psi(X,Y)$.

Proof: It has to be proved that, the constructed figure meets the conditions $\Psi(X,Y)$ (possibly under some preconditions).

Discussion: The discussion should state sufficient and necessary conditions for solutions to exist, and should also consider how many possible solutions to the problem there exist. Ideally, the number of solutions should be expressed effectively in the function of mutual relations of the given elements, but sometimes it is sufficient to express it implicitly in the function of the relation of the figures obtained during the construction.

In previous works on geometric construction or geometric constraint solving the first two phases are often set in the foreground, while the last two are hardly mentioned (while they are seldom easy to achieve).

In the following text, we will give an account of all solution phases while, for illustrating them, we will use one running example (the problem 4 from Wernick's list): *Given points A, B, and G, construct a triangle ABC, such that G is the centroid of ABC.* For this problem, $\Psi(X,Y)$ is $\neg collinear(A,B,C) \wedge centroid(G,A,B,C)$, i.e., the task is to prove:

$$\forall A,B,G.(? \Leftrightarrow \exists C.(\neg collinear(A,B,C) \wedge centroid(G,A,B,C)))$$

where $centroid(G,A,B,C)$ states that G is the centroid of the triangle ABC and ? is a condition, not known in advance that characterizes resolvability of the problem.

Analysis

The purpose of analysis is to detect knowledge sufficient for a procedural specification $Plans(X,Y')$ for a given declarative specification $\Psi(X,Y)$. More precisely, analysis consists of a sequence of proofs of statements of the following form, for $k = 1,\ldots,n$:[4]

$$\forall X,Y'.(\Phi_a(X) \wedge \Psi(X,Y) \wedge Def(X,Y') \wedge \bigwedge_{i=1}^{k-1} Rel_i(X,Y_i') \Rightarrow Rel_k(X,Y_k')) \quad (4)$$

where:

- Y' is a sequence of variables $y_1,\ldots,y_n$ such that $Y \subseteq Y'$ (informally, $Y' \setminus Y$ are auxiliary points to be constructed and used in the construction, along the objects from Y);
- Y_k' is a sequence of variables $y_1,\ldots,y_k$;
- y_n belongs to Y;
- $\Phi_a(X)$ represents some constraints on the given objects (possibly $\top$, if there are no constraints);
- $Def(X,Y')$ introduces properties of $Y' \setminus Y$;
- $Rel_k(X,Y_k')$ is a formula that corresponds to an effective way for constructing y_k by ruler and compass using X and Y_{k-1}'.[5]

Let us denote $\bigwedge_{i=1}^{n} Rel_i(X,Y_i')$ by $Cons(X,Y')$. From the above sequence of theorems, the following theorem follows:

$$\forall X,Y'.(\Phi_a(X) \wedge \Psi(X,Y) \wedge Def(X,Y') \Rightarrow Cons(X,Y')) \quad (5)$$

In order to enable a construction as an effective procedure, it is needed to turn implicit relationship $Rel_i(X,Y_i')$ (for $i = 1$ to n) into the form $\bigvee_{k=1}^{K_i} y_i = RC_{i,k}(X,Y_i')$ [24, 9], which expresses the way(s) in which y_i can be obtained

[4] In later stages of the solution, the given condition $\Phi_a(X)$ may be extended to some condition Φ for which (3) holds.

[5] This formula may involve disjunctions corresponding to different ,,cases" for X and Y. For instance, $(A \neq B \wedge midpoint(C,A,B)) \vee (A = B \wedge C = A)$

from X and Y_i' using ruler and compass.[6] Here, K_i denotes a number of different *ways* in which y_i can be constructed. This number has to be finite, although some *ways* may alow infinite choices. For example, it may be the case that y_i is the intersection point of two lines p and q or an arbitrary point on the line r. It must hold:

$$\forall X, Y'.(Rel_i(X, Y_i') \Leftrightarrow \bigvee_{k=1}^{K_i} y_i = RC_{i,k}(X, Y_i')) \tag{6}$$

Since $Cons(X, Y')$ denotes $\bigwedge_{i=1}^{n} Rel_i(X, Y_i')$, it also holds:

$$\forall X, Y'.(Cons(X, Y') \Leftrightarrow \bigwedge_{i=1}^{n} \bigvee_{k=1}^{K_i} y_i = RC_{i,k}(X, Y_i')) \tag{7}$$

If we denote by $Plan_j(X, Y')$ the conjunction $\bigwedge_{i=1}^{n} y_i = RC_{i,k_i}(X, Y_i')$, for some $k \in \{1, \ldots, K_i\}$ for each $i = 1, \ldots, n$ then, by distributivity, we obtain some J disjuncts as individual construction plans:

$$\forall X, Y'.(Cons(X, Y') \Leftrightarrow \bigvee_{j=1}^{J} Plan_j(X, Y')) \tag{8}$$

If we denote $\bigvee_{j=1}^{J} Plan_j(X, Y')$ by $Plans(X, Y')$, from the above formula and (5), we have:

$$\forall X, Y'.(\Phi_a(X) \wedge \Psi(X, Y) \wedge Def(X, Y') \Rightarrow Plans(X, Y')) \tag{9}$$

Since we are interested in effective constructions of solutions expressed by $Plans(X, Y')$, and if we are careful to introduce only needed auxiliary objects in $Y' \setminus Y$, then it is necessary that they can be defined for every solution. Expressing this obligation for *Def*, we have the following requirement:

$$\forall X, Y.(\Phi_a(X) \wedge \Psi(X, Y) \Rightarrow \exists Y' \setminus Y.Def(X, Y')) \tag{10}$$

There is a subtle issue with $Plans(X, Y')$ — it has to be precise enough to enable the construction, but also it has to be strong enough to prove that the constructed figure indeed meets the specification.

Because of the specific goal, the analysis is more a search process than a proving process. It can be implemented as a search process, while at the end, it can produce a required formula.

[6] Strictly speaking, functions $RC_{i,k}$ may involve more than only ruler and compass. For instance, it may be the case that only one intersection point of two circles can be picked (e.g. ,,that is different from the point. . . “, ,,that is not on the same side. . . “, etc). Also, some of $RC_{i,k}$ may be non-deterministic, for instance ,,pick a point on the line . . . “.

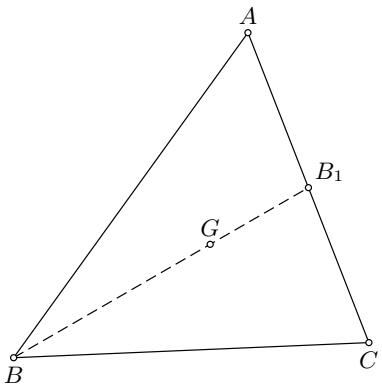

Fig. 1. Illustration for the solution for the running example

Example 1. Let $sratio(P, Q, R, S, m, n)$ mean that $\overrightarrow{PQ}/\overrightarrow{RS} = m/n$. Let $midpoint(B_1, A, C)$ denote that B_1 is the midpoint of the segment AC. The first derivation step for our running example is (Figure 1):

$$\begin{aligned}&\forall A, B, G, B_1, C.\\&(\neg collinear(A, B, C) \wedge centroid(G, A, B, C) \wedge midpoint(B_1, A, C)\\&\Rightarrow sratio(B, B_1, B, G, 3, 2))\end{aligned}$$

and the second derivation step is:

$$\begin{aligned}&\forall A, B, G, B_1, C.\\&(\neg collinear(A, B, C) \wedge centroid(G, A, B, C) \wedge midpoint(B_1, A, C)\\&\wedge sratio(B, B_1, B, G, 3, 2)\\&\Rightarrow sratio(A, C, A, B_1, 2, 1))\end{aligned}$$

These two steps combined give:

$$\begin{aligned}&\forall A, B, G, B_1, C.\\&(\neg collinear(A, B, C) \wedge centroid(G, A, B, C) \wedge midpoint(B_1, A, C)\\&\Rightarrow sratio(B, B_1, B, G, 3, 2) \wedge sratio(A, C, A, B_1, 2, 1))\end{aligned}$$

Here, $\Phi_a(A, B, G)$ is (there may be some preconditions added within the proof phase) $\top$ (meaning that there are no constraints on A, B, G), $Def(A, B, C, G, B_1)$ is $midpoint(B_1, A, C)$, and $Cons(A, B, C, G, B_1)$ is $sratio(B, B_1, B, G, 3, 2) \wedge sratio(A, C, A, B_1, 2, 1)$.

Let $sratioF$ be a partial function such that $Q = sratioF(P, R, S, m, n)$ if $\overrightarrow{PQ}/\overrightarrow{RS} = m/n$. In our running example, there are no different ways for con-

struction, so each K_i equals 1. Then the following holds:

$$\forall A, B, G, B_1, C.$$
$$(sratio(B, B_1, B, G, 3, 2) \wedge sratio(A, C, A, B_1, 2, 1))$$
$$\Rightarrow (B_1 = sratioF(B, B, G, 3, 2) \wedge C = sratioF(A, A, B_1, 2, 1))$$

and the direct consequence of the previous two formulae is:

$$\forall A, B, G, B_1, C.$$
$$(\neg collinear(A, B, C) \wedge centroid(G, A, B, C) \wedge midpoint(B_1, A, C)$$
$$\Rightarrow B_1 = sratioF(B, B, G, 3, 2) \wedge C = sratioF(A, A, B_1, 2, 1))$$

From the formula (10) we get:

$$\forall A, B, C, G.$$
$$(\neg collinear(A, B, C) \wedge centroid(G, A, B, C)) \Rightarrow \exists B_1.(midpoint(B_1, A, C))$$

Construction

The analysis yields the formula enabling effective constructions. For each $j \in \{1, \ldots, J\}$, $Plan_j(X, Y')$ yields one *construction plan* of the form:

- Objects X are given (as *free* objects);
- For $i = 1$ to n
  construct y_i as $y_i = RC_{i,k}(X, Y_i')$ (for some $k \in \{1, \ldots, K_i\}$)

Compound construction steps can also be used (say, construction of the midpoint) so it should be proved that each of $RC_{i,k}$ is expressible using ruler and compass.

Example 2. The construction, derived from the formula $sratio(B, B_1, B, G, 3, 2)$ $\wedge$ $sratio(A, C, A, B_1, 2, 1)$ is as follows:

- The points A, B, G are given (as *free* points);
- $B_1 = sratioF(B, B, G, 3, 2)$;
- $C = sratioF(A, A, B_1, 2, 1)$.

Proof

Within the proof phase, we have to prove correctness for each construction plan $Plan_j(X, Y')$. We have to prove:

$$\forall X.Y'.(\Phi_a(X) \wedge ? \wedge Plan_j(X, Y') \Rightarrow \Psi(X, Y)) \tag{11}$$

where ? is some condition still to be discovered.

Automated theorem provers for geometry typically handle procedural representations of a geometric construction and can deal with conjectures of the above form. We first try to prove the conjecture:

$$\forall X.Y'.(\Phi_a(X) \wedge Plan_j(X, Y') \Rightarrow \Psi(X, Y)) \tag{12}$$

If the conjecture is proved by the prover, the prover can return also some NDG conditions (note that in general case this NDG is not necessarily the weakest condition under which the conjecture holds) so only a weaker statement is proved:

$$\forall X, Y'.(\Phi_a(X) \wedge NDG(X, Y') \wedge Plan_j(X, Y') \Rightarrow \Psi(X, Y)) \tag{13}$$

Example 3. We want to prove:

$$\begin{aligned}&\forall A, B, G, B_1, C.\\&(? \wedge B_1 = sratioF(B, B, G, 3, 2) \wedge C = sratioF(A, A, B_1, 2, 1)\\&\Rightarrow centroid(G, A, B, C) \wedge \neg collinear(A, B, C))\end{aligned}$$

where ? denotes some set of conditions expressed only in terms of points A, B and G, still to be discovered.

Let us first focus on the $centroid(G, A, B, C)$ part. Let us suppose that our theorem provers support $sratioF$, but does not support *centroid* and that we have the following definition of centroid:

$$\begin{aligned}&\forall A, B, C, A_1, B_1.\\&(A_1 = sratioF(B, B, C, 1, 2) \wedge B_1 = sratioF(A, A, C, 1, 2) \wedge\\&G = intersec(AA_1, BB_1) \Rightarrow centroid(G, A, B, C))\end{aligned}$$

We can pass the following conjecture to an automatic (e.g., algebraic) prover:

$$\begin{aligned}&\forall A, B, G, B_1, C, A'_1, B'_1, G'.\\&(B_1 = sratioF(B, B, G, 3, 2) \wedge C = sratioF(A, A, B_1, 2, 1) \wedge\\&A'_1 = sratioF(B, B, C, 1, 2) \wedge B'_1 = sratioF(A, A, C, 1, 2) \wedge\\&G' = intersec(AA'_1, BB'_1) \Rightarrow G = G')\end{aligned}$$

For instance, the above theorem is proved by the prover based on Wu's method, implemented within OpenGeoProver [19]. The prover proves the above conjecture, but returns non-degeneracy conditions „line AA'_1 is not parallel with line BB'_1, and points A and A'_1 are not identical“ ($\neg parallel(AA'_1, BB'_1) \wedge A \neq A'_1$), so we actually proved:

$$\begin{aligned}&\forall A, B, G, B_1, A'_1, B'_1, C.\\&\neg parallel(AA'_1, BB'_1) \wedge A \neq A'_1 \wedge\\&B_1 = sratioF(B, B, G, 3, 2) \wedge C = sratioF(A, A, B_1, 2, 1) \wedge\\&A'_1 = sratioF(B, B, C, 1, 2) \wedge B'_1 = sratioF(A, A, C, 1, 2)\\&\Rightarrow centroid(G, A, B, C)\end{aligned}$$

We also need to prove that points A, B and C are not collinear (under some conditions). It is easily proved (by the ArgoCLP prover [27]) that:

$$\begin{aligned}&\forall A, B, C, G, B_1.\\&(collinear(A, B, C) \wedge B_1 = sratioF(B, B, G, 3, 2) \wedge C = sratioF(A, A, B_1, 2, 1)\\&\Rightarrow collinear(A, B, G))\end{aligned}$$

and its contraposition gives:

$\forall A, B, C, G, B_1.$
$(\neg collinear(A, B, G) \wedge B_1 = sratioF(B, B, G, 3, 2) \wedge C = sratioF(A, A, B_1, 2, 1)$
$\Rightarrow \neg collinear(A, B, C))$

From the previous theorems, we have:

$\forall A, B, G, B_1, A'_1, B'_1, C.$
$\neg parallel(AA'_1, BB'_1) \wedge A \neq A'_1 \wedge \neg collinear(A, B, G) \wedge$
$(B_1 = sratioF(B, B, G, 3, 2) \wedge C = sratioF(A, A, B_1, 2, 1) \wedge$
$A'_1 = sratioF(B, B, C, 1, 2) \wedge B'_1 = sratioF(A, A, C, 1, 2)$
$\Rightarrow centroid(G, A, B, C) \wedge \neg collinear(A, B, C))$

Discussion

Recall that, in general, we want to state sufficient and necessary conditions for a solution to exist (and to counter the number of solutions). Ideally, within discussion we should prove a statement of the form (3). For simplicity, we will assume that $Plans(X, Y')$ has only one disjunct (i.e., only one construction plan), but the following consideration can be easily generalized for cases with more than one disjunct.

From the analysis we have:

$$\forall X, Y'.(\Phi_a(X) \wedge \Psi(X, Y) \wedge Def(X, Y') \Rightarrow Plans(X, Y')) \tag{14}$$

but also:

$$\forall X.(\exists Y'.(\Phi_a(X) \wedge \Psi(X, Y) \wedge Def(X, Y')) \Rightarrow \exists Y'.Plans(X, Y'))) \tag{15}$$

From the above formula and from (10) we get:

$$\forall X.(\exists Y.(\Phi_a(X) \wedge \Psi(X, Y)) \Rightarrow \exists Y'.Plans(X, Y')) \tag{16}$$

and

$$\forall X.(\Phi_a(X) \Rightarrow (\exists Y.\Psi(X, Y) \Rightarrow \exists Y'.Plans(X, Y'))) \tag{17}$$

On the other hand, from the proof we have:

$$\forall X, Y'.(\Phi_a(X) \wedge NDG(X, Y') \wedge Plans(X, Y') \Rightarrow \Psi(X, Y)) \tag{18}$$

and hence:

$$\forall X.(\exists Y'.(\Phi_a(X) \wedge NDG(X, Y') \wedge Plans(X, Y')) \Rightarrow \exists Y.\Psi(X, Y)) \tag{19}$$

and also:

$$\forall X.(\Phi_a(X) \Rightarrow (\exists Y'.(NDG(X, Y') \wedge Plans(X, Y') \Rightarrow \exists Y.\Psi(X, Y))) \tag{20}$$

Therefore, we have necessary (17) and sufficient (20) conditions for $\exists Y.\Psi(X,Y)$ (under the preconditions $\Phi_a(X)$). However, they are not equal, so we still don't have a complete characterization of solvability. We can try do discover[7] a formula $\Phi_d(X)$ such that

$$\forall X.(\Phi_a(X) \Rightarrow (\Phi_d(X) \Rightarrow \exists Y'(NDG(X,Y') \wedge Plans(X,Y'))) \tag{21}$$

If it holds that

$$\forall X.(\Phi_a(X) \Rightarrow (\exists Y.\Psi(X,Y) \Rightarrow \Phi_d(X))) \tag{22}$$

then $\Phi_a(X) \wedge \Phi_d(X)$ is the required formula $\Phi(X)$, and we finally have the theorem (3).

Note that in some cases we can discharge some of conjuncts $ndg(X,Y')$ of $NDG(X,Y')$. For instance, one conjunct may imply an other one, so the latter can be omitted. Also, if:

$$\forall X,Y'.(Plans(X,Y') \Rightarrow ndg(X,Y')) \tag{23}$$

then we can eliminate $ndg(X,Y')$ from $NDG(X,Y')$ in (18). In some cases, this way we can eliminate all of $NDG(X)$, and then $\Phi(X)$ can be equal $\top$.

In some cases, Φ_d involves also some Y', but here we consider only a simple case. In addition, as history teaches us, in some cases this cannot be done using only means of synthetic geometry. (For some unsolvable problems, synthetic approach can be used using reduction, as discussed in Section 2.3).

The above gives a characterization of solvability. Concerning the number of solutions, in solvable case the number of solutions is the product of n numbers of possible choices for each of y_i (see Analysis).

Example 4. From the analysis we have:

$$\begin{aligned}&\forall A,B,G,B_1,C.\\&(centroid(G,A,B,C) \wedge midpoint(B_1,A,C) \wedge \neg collinear(A,B,C)\\&\Rightarrow B_1 = sratioF(B,B,G,3,2) \wedge C = sratioF(A,A,B_1,2,1))\end{aligned}$$

and

$$\begin{aligned}&\forall A,B,G.\\&\exists B_1,C.(centroid(G,A,B,C) \wedge midpoint(B_1,A,C) \wedge \neg collinear(A,B,C))\\&\Rightarrow \exists B_1,C.(B_1 = sratioF(B,B,G,3,2) \wedge C = sratioF(A,A,B_1,2,1))\end{aligned}$$

and, thanks to (11):

$$\begin{aligned}&\forall A,B,G.\\&\exists C.(centroid(G,A,B,C) \wedge \neg collinear(A,B,C))\\&\Rightarrow \exists B_1,C.(B_1 = sratioF(B,B,G,3,2) \wedge C = sratioF(A,A,B_1,2,1))\end{aligned}$$

[7] One can try a finite number of predicates over X.

From the proof, using the following lemma:

$$\forall A, B, C, G, A'_1, B'_1.$$
$$(\neg collinear(A, B, C) \wedge A'_1 = sratioF(B, B, C, 1, 2) \wedge B'_1 = sratioF(C, C, A, 1, 2)$$
$$\Rightarrow \neg parallel(AA'_1, BB'_1) \wedge A \neq A'_1)$$

we obtain:

$$\forall A, B, G, B_1, C.$$
$$(B_1 = sratioF(B, B, G, 3, 2) \wedge C = sratioF(A, A, B_1, 2, 1) \wedge \neg collinear(A, B, G)$$
$$\Rightarrow centroid(G, A, B, C) \wedge \neg collinear(A, B, C))$$

and also:

$$\forall A, B, G.$$
$$\exists B_1, C.(B_1 = sratioF(B, B, G, 3, 2) \wedge C = sratioF(A, A, B_1, 2, 1) \wedge$$
$$\neg collinear(A, B, G))$$
$$\Rightarrow \exists C.(centroid(G, A, B, C) \wedge \neg collinear(A, B, C))$$

From the above theorem, and the theorem:

$$\forall A, B, G, B_1, C.$$
$$\neg collinear(A, B, G)$$
$$\Rightarrow \exists B_1, C.(B_1 = sratioF(B, B, G, 3, 2) \wedge C = sratioF(A, A, B_1, 2, 1))$$

we obtain:

$$\forall A, B, G.$$
$$\neg collinear(A, B, G) \Rightarrow \exists C.(centroid(G, A, B, C) \wedge \neg collinear(A, B, C))$$

which is one direction of the statement we want to prove. For the choice $\Phi_d(A, B, G) = \neg collinear(A, B, G)$, we can prove the formula (22):

$$\forall A, B, G.$$
$$\exists C.(centroid(G, A, B, C) \wedge \neg collinear(A, B, C)) \Rightarrow \neg collinear(A, B, G))$$

Therefore, we have proved:

$$\forall A, B, G.$$
$$\neg collinear(A, B, G) \Leftrightarrow \exists C.(centroid(G, A, B, C) \wedge \neg collinear(A, B, C))$$

2.3 Unconstructible Cases and Reduction

Using geometrical means, it can be proved that a figure is RC-unconstructible (*i*) if the specification is inconsistent (so there is no required figure in Euclidean plane, no matter how it can be produced); (*ii*) if the problem can be reduced to another problem (typically, some *canonical* RC-unconstructible problem), known to be unsolvable using ruler and compass. Here is a simple example of the latter approach, based on one Archimedes's construction.

Example 5. Given three non-collinear points A, B and O, construct points X and Y such that (see Figure 2):

- $OX \cong OB$,
- $XY \cong OB$,
- points B, X and Y are collinear, and
- points A, O and Y are collinear

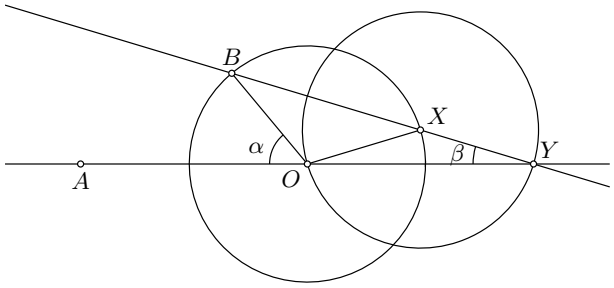

Fig. 2. Example of a RC-unconstructible problem

Using elementary geometry it can be proved that the angle $\alpha = \angle AOB$ is three times angle $\beta = \angle AYB$. Thus, if this problem is RC-solvable, so is the trisection of an angle. But it is well known that in general one cannot divide an angle in three using only straightedge an compass.

2.4 Mechanization

Mechanization of the solving process described in Section 2.2 is the subject of our current work. Our ultimate goal is producing machine verifiable (within the proof assistant Isabelle [21]) solutions for construction problems from Wernick's list. This complex task requires synergy of a tool for solving construction problems, algebraic automated theorem provers, synthetic automated theorem provers, and proof assistants, aided by some human's guidance (and dynamic geometry tools for visualization). In this section we report on the current state.

Analysis. The analysis is performed by our ArgoTriCS tool for solving construction problems [20]. Based on a small number of definitions, lemmas and construction steps, it can solve almost all solvable problems from Wernick's list (66 out of 72). In addition, it can detect if the problem is redundant or locus dependent, and in these cases a point belonging to the geometric loci of points is chosen arbitrarily and construction continues. Also, the problem is tested for symmetry to some of the previous problems according to the available definitions and lemmas. A solution trace from ArgoTriCS (which also contains a subset of

definitions, lemmas and construction steps needed) is translated into a sequence of theorems (4) and the theorem (5). These conjectures (along with relevant axioms/lemmas) are fed to our coherent-logic based automated theorem prover ArgoCLP that is capable of producing machine verifiable proofs [27]. At the moment, ArgoTriCS can automatically produce input files for ArgoCLP (consisting of the set of relevant axioms, the set of relevant lemmas and the theorem to be proved) only for a subset of all considered construction problems and this is the subject of the current work. Since ArgoCLP does not support functional symbols, the formula (6) and subsequent formulae have to be proved manually in Isabelle, and this is also the subject of current work. Also, it should be proved that each of used construction steps is expressible using ruler and compass but at the moment we use them all as primitive steps.

Construction. The construction is automatically exported from ArgoTriCS to our dynamic geometry tool GCLC where it can be visualized and stored in a number of formats, inluding LaTeX, EPS, SVG. This part of our framework is completed.

Proof. ArgoTriCS automatically exports proof specification which can be passed to two different automated theorem provers – OpenGeoProver [19] and the provers integrated into GCLC tool [15]. These tools return as an output a proof object, and the set of non-degenerate (NDG) conditions. The central theorem (of the form (12)) is proved by one of these provers. Since there is no trusted link between these provers and Isabelle, we use these theorems as axioms within Isabelle.

Discussion. NDG conditions obtained from the provers may involve some auxiliary objects. Statements needed for translating these conditions into ones that involve only given objects are proved using ArgoCLP. Simple consequences that do not belong to coherent logic are proved within Isabelle manually (as they cannot be proved by ArgoCLP). For attempts at discovering a sufficient and necessary conditions for solution to exist ($\neg colin(A, B, G)$ in the running example), we test a finite number of predicates over the set of given points. Also, we check if come conjunct of NDG implies some other one. Finally, the final proof is glued together within Isabelle by simple steps (still to be automated, currently they are performed manually).

Overall, in our formalized solutions of construction problems, there are two important gaps. One of them is the link to external algebraic provers. Conjectures proved within the Proof phase are proved using algebraic provers, but there is no trusted link between them and Isabelle. Therefore, the conjectures proved by external algebraic provers we use as axioms. Currently, there are some limited formalizations of algebraic provers for geometry within proof assistants [13, 12], but not for Isabelle. For theorems proved by ArgoCLP we have formal Isabelle proofs. The second gap is between our proofs and the typical geometrical axioms

(e.g., Tarski's or Hilbert's axioms). In our proofs we use high-level geometry lemmas as axioms. They can be proved from basic axioms (e.g., Tarski's or Hilbert's axiom) but it is extremely complex task and beyond the scope of this paper. Only recent advances provide (in Coq) formally proved high-level lemmas from the basic axioms [3, 2].

3 Algebraic Approach

Mathematical progress in algebra in the beginnings of the nineteenth century enabled solving of many geometric construction problems that were open since the ancient Greeks. When considering construction problems, two aspects have to be distinguished: constructibility and construction. In both cases algebraic methods can have significant role but algebraic tools are famous for their success in proving RC-unconstructibility. Actually, it is theoretically possible to extract a construction from a proof of RC-constructibility but it is often impracticable and when it is effectively possible, it is often pedagogically useless.

3.1 Classical results

In the introduction, we recall the definition of the constructibility of points, lines and circle from a set of points B also called base points which correspond more or less to the notion of free points in dynamic geometry. We define now RC-constructibility of numbers: a number is said RC-constructible from set B if it is a coordinate of a RC-constructible point. As a convention, it is said that a point or a number is RC-constructible when $B = \{(0,0),(1,0)\}$. For instance, any rational number is RC-constructible; given $\cos(\alpha)$ for some number α, $\cos(\alpha/3)$ is RC-unconstructible in general (this fact is known as the impossibility of the trisection of an angle using only straightedge and compass).

A fundamental example is given by the classical operations —addition, subtraction, multiplication, division and square radical extraction— which can all be translated in terms of RC-construction. The converse is true: a number is RC-constructible from points of set B if and only if it is expressible with the five operations operating on the coordinates of points in B. This fact is closely related to field theory which in turn gave a theoretical decision procedure for RC-constructibility problems [18].

Field theory allows to link numbers and polynomials: an algebraic number over a field, usually $\mathbb{Q}$ is a solution of a polynomial equation. A fundamental result is that to any algebraic number over K α is associated a monic irreducible polynomial $P \in K[X]$, called the minimal polynomial of P and $K(\alpha) \sim K[X]/P$. The degree of P is the degree of the extension $[K(\alpha) : K]$ which is also called the degree of α (over K). Then, the main tool for proving RC-unconstructibility lies in Wantzel's result:

Theorem 1 (Wantzel 1837). *Each RC-constructible number is algebraic over $\mathbb{Q}$ and its degree is equal to 2^k for some $k \in \mathbb{N}$.*

This theorem can be used to prove that $\sqrt[3]{2}$ is not RC-constructible since polynomial $X^3 - 2$ is irreducible over $\mathbb{Q}$. But also to prove that problem #90 of Wernick's list is not RC-constructible as, for some choice for the coordinates, it is equivalent to solve the irreducible polynomial equation (obtained by using resultants and factorization within Maxima):

$$2x_A^5 + 45x_A^4 + 372x_A^3 + 1368x_A^2 + 2160x_A + 972 = 0$$

Note that the reciprocal of Wantzel's theorem is false: for instance the roots of the irreducible polynomial $X^4 + 2X - 2$ are not RC-constructible. There are several theorems of algebra which fully determine RC-constructibility:

Theorem 2 (Galois). *Let α be an algebraic number over $\mathbb{Q}$ or an extension of $\mathbb{Q}$ and P be its minimal polynomial; α is RC-constructible if and only if the degree of splitting field of P is a power of 2.*

This theorem has a more practical formulation:

Theorem 3. *A number α is RC-constructible if and only if it exists some algebraic number $r_1, \ldots, r_n = \alpha$ such that $[\mathbb{Q}(r1) : \mathbb{Q}] = 2$, $[\mathbb{Q}(r_{i+1}) : \mathbb{Q}(r_i)] = 2$ for every $i = 1, \ldots, n-1$.*

The method proposed by Gao & Chou in [10] exploit this latter one.

As far as we know, there are two automatic implemented methods for deciding RC-constructibility (but, in the mathematic literature there are several papers dealing with resolution by radicals of polynomial equations, see for instance [17]). The first one comes from a book of H. Lebesgue about geometric construction problems [18] and it has been implemented by G. Chen in 1992 [6]. The second one is described in papers presented in the second ADG workshop and published in Journal of CAD ([29]).[8] For the sake of completeness, we show in the next section how algebraic method can be used to prove RC-unconstructibility of a problem.

We present two examples coming from famous Wernick's list [28], and we solve them by a classical method: thanks to a Computer Algebra System (CAS in short), we obtain one or more triangular systems, if possible irreducible. There are various methods to triangularize a polynomial system: one can either using successive resultants, or computing Gröbner bases with a lexicographic order, or computing the Ritt's characteristic sets. Here, we use the Maple package *RegularChains* and particularly the *Triangularize* function. Then, we study the polynomial equations as polynomials with a single variable using here Wantzel's theorem or Gao & Chou's method.

3.2 Unconstructible Case

Wernick's problem #122. In this problem, points G, T_a and T_b are given, and the task is to construct a triangle $T = (A, B, C)$ such that points G,

[8] The technical report can be found here:
`http://www.mmrc.iss.ac.cn/pub/mm15.pdf/gao.pdf`.

T_a and T_b are respectively the centroid, the foot of the inner-bisector from A and the inner-bisector from B of T. To our knowledge, the status (constructible/unconstructible) or this problem is still unknown (one of 15 unsolved Wernick's problems).

Without loss of generality, we choose a reference system in order to fix the coordinates: let T_b have coordinates $(0, 0)$ and let T_a have coordinates $(4, 0)$. On one hand, if we want to prove RC-constructibility, or even to produce a construction, the coordinates of G must be parameters. On the other hand, if we only want to check RC-unconstructibility, we can choose arbitrary coordinates for G. Here, we choose the coordinates $(2, 1)$ for G. If we are unlucky, it could happen that even if the problem is not RC-constructible in general, in this particular case it is. Let us see what happen here.

First, we classically translate the geometric problem in an algebraic formulation consisting in a polynomial system S. There is of course some issues like, for instance, the fact that we represent both internal and external bisectors when using algebraic formulation, but we do not discuss these points here (see [26]). Then, we have to triangularize S: to this end, we use regular chains method that is implemented in Maple. For the triangulation of the polynomial system corresponding to the statement, we choose the ordering x_C, y_C, x_B, y_B, x_A, y_A for the variables. We find two irreducible systems which corresponds to non degenerate cases, say S_1 and S_2: this means that under the non degeneracy conditions $S \Leftrightarrow S_1 \vee S_2$ and it is enough to show that none of these systems corresponds to a RC-constructible problem (if one of these systems was RC-constructible, then we could construct some solutions of the problem)

In the first one, we have the irreducible polynomial equation:

$$4y_A^4 - 12y_A^3 - 51y_A^2 + 192y_A - 144 = 0$$

and for the second one:
$7295401y_A^6 - 30894038y_A^5 + 107596129y_A^4 - 127795968y_A^3 - 3722832y_A^2$
$+ 24966144y_A + 4064256 = 0$
The two polynomials are irreducible and Wantzel theorem ensures that the second one is not RC-solvable. But there is more work to do for the first equation as it is of degree 2^2. Using the formula of Gao & Chou [29], we have to see if the polynomial:

$$8g^3 4h_2 g^2 + (2h_1 h_3 8h_0)g h_0 h_3^2 + 4h_0 h_2 h_1^2 = 0$$

has rational solutions, where $x^4 + h_3x^3 + h_2x^2 + h_1x + h_0$ is the minimal monic irreducible polynomial we want to test: here, we have $h_3 = -12/4$, $h_2 = -51/4$, $h_1 = 192/4$ and $h_0 = 144/4$ and we get the polynomial:

$$8g^3 + 51g^2 - 144$$

Then, using the `factor` command, we prove that this polynomial is irreducible and thus that the problem is RC-unconstructible. Notice that this method can also be used to find a construction when the problem is RC-constructible, but usually the construction is impracticable and not in the spirit of classical RC-constructions.

3.3 Constructible Case

Wernick's problem #116. In this problem, points G, H_a and H are given and the task is to construct a triangle $T = (A, B, C)$ such that points G, H_a and H are respectively the centroid, the feet of the altitude from point A and the orthocenter of triangle T,

It is easy to construct the line BC, M_a and A and O using the fact that G is the center of the homothety with ratio $-1/2$ transforming O into H. We left the construction to the reader.

Let us consider the algebraic version. Let the given points have the following coordinates: $H(0,0)$, $H_a(1,0)$, $G(a,b)$, where a and b are some real numbers. We have then to solve the system:

$$\begin{cases} x_A(x_B - x_C) + y_A(y_B - y_C) = 0 \\ x_B(x_C - x_A) + y_B(y_C - y_A) = 0 \\ (x_A - 1)(x_C - 1) + y_A y_C = 0 \\ (1 - x_B)(y_C - y_B) - y_B(x_B - x_C) = 0 \\ 3a - x_A - x_B - x_C = 0 \\ 3b - y_A - y_B - y_C = 0 \end{cases}$$

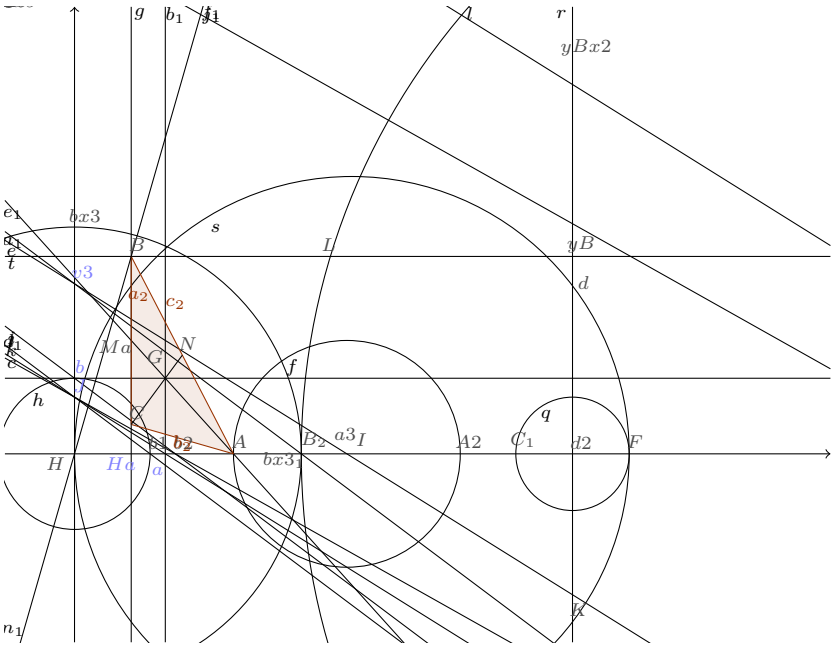

Fig. 3. Performing algebraic computations with geometry (details where the attentive reader can see, for instance, the extraction of a square root $d2 \to d$)

After triangularization, we have only one non degenerate system:

$$\begin{cases} x_C - 1 = 0 \\ y_C + y_B - 3b = 0 \\ -1 + x_B = 0 \\ y_B^2 - 3b.y_B + 3a - 3 = 0 \\ 2 - 3a + x_A = 0 \\ y_A = 0 \end{cases}$$

which yields two, one or zero solutions depending on the discriminant of the fourth equations. The two solutions correspond to each other by permuting B and C. This problem is then obviously constructible since we are able to perform all the operations with straightedge and compass. Such a construction is depicted on Fig.3 made with GeoGebra, where parameters a and b correspond to free points constrained to be on the x-axis and y axis respectively. But that solution usually is not the solution that the teacher wanted. However, it is not difficult to algebraically verify that the solutions provided by the triangular system are solutions to the problem. Of course, you have to be confident in your CAS.

4 Conclusions and Future Work

We presented a geometrical and an algebraic perspective on solving construction problems using ruler and compass. We showed that many steps of this process can be automated and supported by proofs which can be formalized within proof assistants.

For our future work, we are planning to complete, as much as possible, automation of solving for problems from Wernick's corpus, but also for other classes of construction problems. We are planning to integrate this automated process into dynamic geometry systems, having in mind applications in education. We are also planning to implement proving unconstructibility by reduction.

References

1. B. Aldefeld. Variations of geometries based on a geometric-reasoning method. *Computer-Aided Design*, 20(3):117–126, 1988.
2. Pierre Boutry, Julien Narboux, Pascal Schreck, and Gabriel Braun. Using small scale automation to improve both accessibility and readability of formal proofs in geometry. In *submitted to International Workshop on Automated Deduction in Geometry ADG14*, page submitted. Universidade de Coimbra, 2014.
3. Gabriel Braun and Julien Narboux. From tarski to hilbert. In Tetsuo Ida and Jacques D. Fleuriot, editors, *Automated Deduction in Geometry*, volume 7993 of *Lecture Notes in Computer Science*, pages 89–109. Springer, 2012.
4. William Buoma, Ioannis Fudos, Christoph Hoffman, Jiazhen Cai, and Robert Paige. A Geometric Constraint Solver. In *CAD 27*, pages 487–501, 1995.
5. Michel Buthion. Un programme qui résoud formellement des problèmes de constructions géométriques. *RAIRO Informatique*, (3), 1979.

6. Guoting Chen. Les constructions à la règle et au compas par une méthode algébrique. Technical Report Rapport de DEA, Université Louis Pasteur, 1992.
7. Mirjana Djorić and Predrag Janičić. Constructions, instructions, interactions . *Teaching Mathematics and its Applications*, 23(2):69–88, 2004.
8. Jean-François Dufourd, Pascal Mathis, and Pascal Schreck. Geometric construction by assembling solved subfigures. *Artificial Intelligence Journal*, 99(1):73–119, 1998.
9. Caroline Essert-Villard, Pascal Schreck, and Jean-François Dufourd. Sketch-based pruning of a solution space within a formal geometric constraint solver. *Artif. Intell.*, 124(1):139–159, 2000.
10. 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.
11. Herbert Gelernter. Realization of a geometry theorem proving machine. In *Proceedings of the International Conference Information Processing*, pages 273–282, Paris, June 15-20 1959.
12. Jean-David Génevaux, Julien Narboux, and Pascal Schreck. Formalization of wu's simple method in coq. In *Certified Programs and Proofs (CPP 2011)*, volume 7086 of *Lecture Notes in Computer Science*, pages 71–86. Springer, 2011.
13. Benjamin Grégoire, Loïc Pottier, and Laurent Théry. Proof certificates for algebra and their application to automatic geometry theorem proving. In Thomas Sturm and Christoph Zengler, editors, *Automated Deduction in Geometry*, volume 6301 of *Lecture Notes in Computer Science*, pages 42–59. Springer, 2008.
14. Sumit Gulwani, Vijay Anand Korthikanti, and Ashish Tiwari. Synthesizing geometry constructions. In *Programming Language Design and Implementation, PLDI 2011*, pages 50–61. ACM, 2011.
15. Predrag Janičić. Geometry Constructions Language. *Journal of Automated Reasoning*, 44(1-2):3–24, 2010.
16. Christophe Jermann, Gilles Trombettoni, Bertrand Neveu, and Pascal Mathis. Decomposition of Geometric Constraint Systems: a Survey. *International Journal of Computational Geometry and Applications*, 16(5-6):379–414, 2006. CNRS Math-STIC.
17. Susan Landau and Gary L. Miller. Solvability by radicals is in polynomial time. *Journal of Computer and System Sciences*, 30(2):179–208, April 1985. invited publication.
18. Henri Lebesgue. *Leçons sur les constructions géométriques*. Gauthier-Villars, Paris, 1950. in French, re-edition by Editions Jacques Gabay, France.
19. Filip Marić, Ivan Petrović, Danijela Petrović, and Predrag Janičić. Formalization and implementation of algebraic methods in geometry. In Pedro Quaresma and Ralph-Johan Back, editors, Proceedings First Workshop on *CTP Components for Educational Software,* Wrocław, Poland, 31th July 2011, volume 79 of *Electronic Proceedings in Theoretical Computer Science*, pages 63–81. Open Publishing Association, 2012.
20. Vesna Marinković and Predrag Janičić. Towards understanding triangle construction problems. In J. et.al. Jeuring, editor, *Intelligent Computer Mathematics - CICM 2012*, volume 7362 of *Lecture Notes in Computer Science*. Springer, 2012.
21. Tobias Nipkow, Lawrence C. Paulson, and Markus Wenzel. *Isabelle/HOL: A Proof Assistant for Higher-Order Logic*. Springer, 2002. Lecture Notes in Computer Science Tutorial 2283.
22. J. Owen. Algebraic solution for geometry from dimensional constraints. In *Proceedings of the 1th ACM Symposium of Solid Modeling and CAD/CAM Applications*, pages 397–407. ACM Press, 1991.

23. Joseph M. Scandura, John H. Durnin, and Wallace H. Wulfeck II. Higher order rule characterization of heuristics for compass and straight edge constructions in geometry. *Artificial Intelligence*, 5(2):149–183, 1974.
24. Pascal Schreck. Robustness in CAD Geometric Constructions. In *IV 2001*, 111-116.
25. Pascal Schreck. Modélisation et implantation d'un système à base de connaissances pour les constructions géométriques. *Revue d'intelligence artificielle*, 8(3):223–247, 1994.
26. Pascal Schreck and Pascal Mathis. Rc-constructibility of problems in wernick's and connelly's lists. In *submitted to International Workshop on Automated Deduction in Geometry ADG14*, page submitted. Universidade de Coimbra, 2014.
27. Sana Stojanović, Vesna Pavlović, and Predrag Janičić. A coherent logic based geometry theorem prover capable of producing formal and readable proofs. In Pascal Schreck, Julien Narboux, and Jürgen Richter-Gebert, editors, *Automated Deduction in Geometry*, volume 6877 of *Lecture Notes in Computer Science*. Springer, 2011.
28. William Wernick. Triangle constructions vith three located points. *Mathematics Magazine*, 55(4):227–230, 1982.
29. Xiao-Shan Gao X.S. and Shang-Ching Chou. Solving geometric constraint systems ii. a symbolic approach and decision of rc-constructibility. *Computer-Aided Design*, 30:115–122, 1998.
30. Victor Pambuccian. Axiomatizing geometric constructions. *Journal of Applied Logic*, 6,1:24–46, 2008.
31. Michael Beeson. Logic of Ruler and Compass Constructions, In S. Barry Cooper and Anuj Dawar and Benedikt Löwe, editors, *Proceedings of the 8th Conference on Computability in Europe*, volume 7318 of *Lecture Notes in Computer Science*. Springer, 2012.

5 Appendix

Here we present a list of theorems proved within Isabelle and the list of axioms used:

5.1 The list of axioms

```
ax_GL04a: "sratio_2_3 A B C D ==> sratio_3_2 C D A B"

ax_GL04b: "sratio_1_2 A B C D ==> sratio_2_1 C D A B"

ax_GL05: "sratio_1_2 A B A C ==> sratio_1_2 C B C A"

ax_L56: "centroid G A B C /\ midpoint B1 A C ==> sratio_2_3 B G B B1"

ax_D4a: "inc A MA /\ inc A1 MA /\ inc B MB /\ inc B1 MB /\ sratio_1_2 B A1 B C
         /\ sratio_1_2 C B1 C A /\ centroid G A B C ==> (inc G MA /\ inc G MB)"

ax_D4b: "inc A MA /\ inc A1 MA /\ inc B MB /\ inc B1 MB /\ sratio_1_2 B A1 B C
         /\ sratio_1_2 C B1 C A /\ inc G MA /\ inc G MB ==> centroid G A B C"

ax_D22a: "sratio_1_2 C B1 C A ==> midpoint B1 A C"

ax_D22b: "midpoint B1 A C ==> sratio_1_2 C B1 C A "

ax_sratio_1_2_false: "sratio_1_2 PO1 PO2 PO3 PO4 /\
                      not_sratio_1_2 PO1 PO2 PO3 PO4 ==> False"

ax_sratio_3_2_false: "sratio_3_2 PO1 PO2 PO3 PO4 /\
                      not_sratio_3_2 PO1 PO2 PO3 PO4 ==> False"

ax_sratio_2_3_false: "sratio_2_3 PO1 PO2 PO3 PO4 /\
                      not_sratio_2_3 PO1 PO2 PO3 PO4 ==> False"

ax_eq_point_false: "eq_point PO1 PO2 /\
                    not_eq_point PO1 PO2 ==> False"

ax_eq_point_false1: "(eq_point PO1 PO2 ==> False) ==>
                      not_eq_point PO1 PO2"

ax_collinear_false: "collinear PO1 PO2 PO3 /\
                     not_collinear PO1 PO2 PO3 ==> False"

ax_collinear_false1: "(collinear PO1 PO2 PO3 ==> False) ==>
                       not_collinear PO1 PO2 PO3"
```

```
ax_parallel_false1: "(parallel PO1 PO2 PO3 PO4 ==> False) ==>
                     not_parallel PO1 PO2 PO3 PO4"

ax_ogp_proof: "sratio_3_2 B B1 B G /\ sratio_2_1 A C A B1 /\
               sratio_1_2 B A1' B C /\ sratio_1_2 C B1' C A /\
               not_eq_point A A1' /\ not_parallel A A1' B B1'
               ==> centroid G A B C"

ax_constr_sratio_3_2: "not_eq_point C D ==>
                       ( \<exists> (B::point). sratio_3_2 A B C D)"

ax_constr_sratio_2_1: "not_eq_point C D ==>
                       ( \<exists> (B::point). sratio_2_1 A B C D)"

ax_constr_sratio_1_2: "not_eq_point C D ==>
                       ( \<exists> (B::point). sratio_1_2 A B C D)"

ax_eq_points_collinear: "eq_point C A ==> collinear A B C"
```

5.2 The list of theorems

Analysis

```
lemma  Thm_analysis_step_1:
assumes "not_collinear A B C" and "centroid G A B C"
and "midpoint B1 A C"
shows "(sratio_3_2 B B1 B G)"

lemma  Thm_analysis_step_2:
assumes "not_collinear A B C" and "centroid G A B C"
and "midpoint B1 A C" and "sratio_3_2 B B1 B G"
shows "(sratio_2_1 A C A B1)"

lemma  Thm_analysis_all_steps:
assumes "not_collinear A B C" and "centroid G A B C"
and "midpoint B1 A C"
shows "(sratio_3_2 B B1 B G & sratio_2_1 A C A B1)"

lemma Thm_analysis_distinct_points:
assumes "not_collinear A B C"
shows "not_eq_point C A"

lemma Thm_analysis_existence_of_definitions:
assumes "not_collinear A B C" and "centroid G A B C"
shows "(\<exists> (B1::point). midpoint B1 A C)"
```

Proof

```
lemma Thm_proof_1:
assumes "sratio_3_2 B B1 B G" and "sratio_2_1 A C A B1"
and "collinear A B C"
shows "collinear A B G"

lemma Thm_proof_1c:
assumes "sratio_3_2 B B1 B G" and "sratio_2_1 A C A B1"
and "not_collinear A B G"
shows "not_collinear A B C"

lemma Thm_DistinctPoints1:
assumes "sratio_3_2 B B1 B G"
and "sratio_2_1 A C A B1"
and "not_collinear A B G"
shows "not_eq_point B C"

lemma Thm_DistinctPoints2:
assumes "sratio_3_2 B B1 B G"
and "sratio_2_1 A C A B1"
and "not_collinear A B G"
shows "not_eq_point C A"

lemma Thm_proof_construction:
assumes "sratio_3_2 B B1 B G" and "sratio_2_1 A C A B1"
and "sratio_1_2 B A1' B C" and "sratio_1_2 A B1' A C"
and "not_parallel A A1' B B1'" and "not_eq_point A A1'"
and "not_collinear A B G"
shows "(centroid G A B C \<and> not_collinear A B C)"
```

Discussion

```
lemma  Thm_discussion_exist1:
assumes " ( \<exists> (B1::point) (C::point).
              centroid G A B C & midpoint B1 A C & not_collinear A B C)"
shows " ( \<exists> (B1::point) (C::point).
          sratio_3_2 B B1 B G & sratio_2_1 A C A B1)"

lemma  Thm_discussion_exist2:
assumes " ( \<exists> (C::point).
              centroid G A B C & not_collinear A B C)"
shows " ( \<exists> (B1::point) (C::point).
          sratio_3_2 B B1 B G & sratio_2_1 A C A B1)"

lemma Thm_discussion_2a:
```

```
assumes "eq_point A A1'" and "sratio_1_2 B A1' B C"
and "sratio_1_2 C B1' C A"
shows "(collinear A B C)"

lemma Thm_discussion_2ac:
assumes "not_collinear A B C" and "sratio_1_2 B A1' B C"
and "sratio_1_2 C B1' C A"
shows "(not_eq_point A A1')"

lemma Thm_discussion_2b:
assumes "parallel A A1' B B1'"
and "sratio_1_2 B A1' B C"
and "sratio_1_2 C B1' C A"
shows "(collinear A B C )"

lemma Thm_discussion_2bc:
assumes "not_collinear A B C"
and "sratio_1_2 B A1' B C"
and "sratio_1_2 C B1' C A"
shows "(not_parallel A A1' B B1' )"

lemma Thm_discussion_construction:
assumes "sratio_3_2 B B1 B G"
and "sratio_2_1 A C A B1"
and "not_collinear A B G"
shows "(centroid G A B C \<and> not_collinear A B C)"

lemma Thm_discussion_final1:
assumes "( \<exists> (C::point).
           centroid G A B C \<and> not_collinear A B C)"
shows "not_collinear A B G"

lemma Thm_discussion_final2:
assumes "not_collinear A B G"
shows "( \<exists> (C::point).
         centroid G A B C \<and> not_collinear A B C)"
```