DOI : 10.1007/978-3-642-21898-9_32



# A Coq-based Library for Interactive and Automated Theorem Proving in Plane Geometry*

Tuan-Minh Pham[1], Yves Bertot[1], and Julien Narboux[2]

[1] INRIA Sophia Antipolis
[2] LSIIT, Université de Strasbourg, CNRS

**Abstract.** In this article, we present the development of a library of formal proofs for theorem proving in plane geometry in a pedagogical context. We use the Coq proof assistant [4]. This library includes the basic geometric notions to state theorems and provides a database of theorems to construct interactive proofs more easily. It is an extension of the library of F. Guilhot for interactive theorem proving at the level of high-school geometry [7], where we eliminate redundant axioms and give formalizations for the geometric concepts using a vector approach. We also enrich this library by offering an automated deduction method which can be used as a complement to interactive proof. For that purpose, we integrate the formalization of the area method [3] which was developed by J. Narboux in Coq [12, 10].

**Keywords:** formalization, automation, geometry, Coq, teaching

## 1 Introduction

Technological tools are widely used to teach mathematics in schools. Dynamic Geometry Software (DGS) and Computer Algebra Software (CAS) are the two families of software that are well represented. These tools are widely used to explore, experiment, visualize, calculate, measure, find counter examples, conjectures... but most of them can not be used directly to build or check a proof. Proof is a crucial aspect of mathematics and therefore should be integrated more into the mathematical tools. The exploration and proof activities are interlaced. We think that these two activities could be better interlaced if they were both conducted using the computer.

Dynamic geometry systems (DGS) are used more and more to teach geometry in school. They not only allow students to understand construction steps that lead to final drawings, but also provide access to geometric objects, allow students to move free points and see the influence on the rest. Then, students can find out new properties from the drawings. To justify conjectures, some among the numerous DGS provide proving features. These systems are almost

* This work is partially supported by the french ANR project Galapagos

all based on algebraic (coordinate-based) methods or semi-algebraic (coordinate-free) methods. Most DGS do not allow users to interactively build traditional proofs. The few DGS which provide interactive theorem proving features are either based on databases of proofs known in advance or on adhoc theorem provers which can not be easily extended. In [13], we give a more precise description of the different systems. We believe that deductive proofs should continue to be an essential part of the high-school curriculum for a geometry course. We also think that the use of a proof assistant to interactively build proofs could help students understand the concepts of deduction. This use has the following advantages:

– It gives clear logical view about the geometric problems. The system makes clear what are the hypotheses and the conclusions. Student can understand the logical inferences used in each reasoning step by observing the change of the proof environment.
– Reasoning steps are verified by the proof assistant, thus constructed proofs have very high level of confidence.
– It allows to combine purely geometric arguments with other kinds of proofs (using complex numbers for instance).

This leads to the necessity of developing a geometry proving tool for high school students which they can use to interactively construct traditional geometry proofs with the support of a proof assistant.

The first step in this direction was done by F. Guilhot, a high school teacher who developed a library in the Coq proof assistant for interactive theorem proving in geometry at the level of high-school [7]. This development is based on a specific axiom system which is adapted to the knowledge of high school students. It covers a large portion of basic notions of plane geometry, properties and theorems. Its proofs and its geometry reasoning are close to what students learn in high school. Some classical theorems are proved in this library, such as Menelaus, Ceva, Desargues, Pythagoras, Simsons' line etc. Then, a tool called GeoView was developped in order to visualize graphically geometry statements using a DGS [2].

Another tool using Coq that was developed by J. Narboux is GeoProof. This tool works in the opposite direction: the DGS is used to generate statements and tactics for Coq. Its proving feature is based on the automated deduction method - the area method. It is not an interactive proving tool [14].

Recently, the first author developed a tool for interactive proof using GeoGebra [6] and Coq [16]. A proof window is added in GeoGebra, it allows to communicate with Coq. The user can draw figures and state conjectures, geometric constructions and conjectures are translated into relations between geometric objects and sent to Coq. A geometric problem is expressed by a set of hypotheses and conclusions. The reasoning steps are interactively built by mouse clicks over geometric objects and hypotheses using appropriate proving methods. The reasoning steps are translated into commands and executed by Coq. The new state is sent back to the user. The changes of hypotheses and conclusions allows the user to continue the proof. We refer readers to [16] for more information about the graphical user interface for this tool.

A crucial part of this tool lies in its library for synthetic geometry in Coq. The library contains formalizations of geometric notions, and of geometric theorems that allow the user to manipulate geometric objects and to produce proofs. Developing a library in the style of the synthetic geometry system is appropriate. In the synthetic geometry, we postulate the existence of geometric objects (e.g. points, lines,. . . ) and geometric relations as primitive notions and build up the geometry from an axiom system that deals with these primitive notions. The geometry proofs are constructed by making use of axioms, theorems and logical arguments. This approach is pedagogically suitable and satisfactory from a logical point of view. In this paper, we focus on this synthetic library and we present why and how we extended [7].

The development by F.Guilhot has three main drawbacks: first it is based on many axioms which are not really needed, second it lacks constructive definitions for the existence of geometric objects, third it lacks automation. Indeed, in this library, each geometric notion is defined by axioms that assert its properties. This leads to an explosion of the number of axioms. Moreover, the library does not provides functions to construct points and lines and compound geometric constructions, it only provides axioms which states the existence of the compound objects. We will detail these drawbacks in section 2. Finally, the library does not include any automatic theorem proving methods. We think that having an automatic theorem proving method is important because:

- During an interactive proof, students can meet minor proof obligations. It they need to solve these proof obligations interactively this can lead to very technical proofs. This is not adapted to their level of abstraction.
- It may be useful to be able to check if a statement is correct in order to help the student or to give better error messages when the student make a mistake.

This motivates us to improve this library. We reduce the number of axioms by providing a more compact axiom system and constructively building up the geometric notions from this system. We also enrich the library with an automated deduction method - the area method [3]. This method gives readable proofs, based on the notions: signed area, ratio of directed segment and Pythagoras difference which are also very close to high school knowledge. For that purpose, we integrate the formalization of the area method in Coq by J.Narboux [12, 10].

This article is organized as follows. After giving some discussion about the formalization of F. Guilhot in Section 2, we present in Section 3 our developments to eliminate its redundant axioms and to redefine geometric notions. Section 4 deals with the integration of the area method. The last section is for the conclusion and our perspectives.

**Related works** Several axiom systems for synthetic geometry were produced, Hilbert proposed his axiom system [8] with 20 axioms relating two primitive notions: point and line. Another was proposed by Tarski [18] which is simpler, relies on first order logic, and contains only eleven axioms, two predicates, and one primitive notion: point. The axiom system of Hilbert was formalized in Coq by C. Dehlinger [5] and in Isabelle/Isar by L. Meikle [11]. The one of Tarski was formalized in Coq by J. Narboux [15]. However, systems such as Hilbert's system or Tarski's system have starting points that are too low, hence they can not be used in a pedagogical context. The work closest to ours is the one by P. Scott and J. Fleuriot [17]. This work present the integration of an automatic theorem proving tool in the user interface of a proof assistant. The difference with our work is that the automatic theorem proving tool they integrate is not specialized in geometry. Hence, it can be used in many contexts as for example a low level development about Hilbert axiom system but it may be less efficient for complex geometry theorems.

## 2 Formalization of High-school Geometry

In this section, we present the development of F.Guilhot and what can be improved. As mentioned, F. Guilhot does not try to provide a system with a minimal number of axioms, nor to provide an automated tool for theorem proving. She only tries to provide a system in which definitions of geometric notions, theorems and geometry reasonings are described as they are taught in high school. She does not build up the whole of Euclidean geometry from a fundamental axiomatic system (such as the systems mentioned above). She uses an alternative approach to the same geometrical results. She first constructs a vector space attached to the affine geometry, then she construct an Euclidean space by adding the notion of scalar product to the affine geometry.

We use small letters $a$, $b$, $c\ldots$ to denote real numbers; capital letters $A$, $B$, $C\ldots$ to denote points; pairs of a real number and a point in the form $aA$ to denote mass points.

The key technique in her formalization lies in using the universal space proposed by M. Berger [1]. The vector space is extended to the universal space which is a union of points with a given non-zero mass and the vectors in the vector space. The rule $aB - aA = a\overrightarrow{AB}$ is used to convert vectors to the representation in mass points. The vector space property of the universal space is preserved and defined by the following axioms:

- Axiom 1 (Definition of addition) : With $(m+n \neq 0)$, there is a unique point R (called barycenter) on $PQ$ such that $nP + mQ = (m+n)R$.
- Axiom 2 (Definition of vector): With $(m+n = 0)$, the sum of $(nP + mQ)$ is a vector $m\overrightarrow{PQ}$.
- Axiom 3 (Idempotent): $nP + mP = (m+n)P$.
- Axiom 4 (Commutative): $nP + mQ = mQ + nP$.
- Axiom 5 (Associative): $nP + (mQ + kR) = (nP + mQ) + kR$.
- Axiom 6 (Definition of scalar multiplication): $k(nP) = (k * n)P$.
- Axiom 7 (Distributivity) $k(nP + mQ) = knP + kmQ$.

These properties form an extension of the theory of mass points (where mass is not limited to a positive real number). So we can completely perform calculations in mass points as we can do with problem-solving technique of mass point theory. Furthermore, these calculations are taught in high school courses and straightforward.

In Coq, the type of mass points is declared as a record composed of a real number and a point. A variable *add_MP* which takes two arguments of type mass point and gives a mass point is declared for addition operator and a variable *mult_MP* which takes a real number and a mass point as arguments and gives a mass point is declared for scalar multiplication operator.

```
Record MassPoint:Type := cons{number: R; point:Point}.
Variable add_MP  : MassPoint -> MassPoint -> MassPoint.
Variable mult_MP : Real      -> MassPoint -> MassPoint.
```

The system of axioms which defines the universal space is introduced. It allows us to manipulate mass points. A good idea for calculating mass points in Coq is mapping this space to an abstract field structure. This enables us to simplify equations of mass points using automated tactics in Coq library for field (such as `ring_simplify`, `field_simplify`...). It makes calculations easier.

Plane Euclidean geometry can be obtained by equipping affine geometry with the notion of scalar product. The other geometric notions are added step by step using existing formalized notions and their relations in the library.

However, the geometric objects are almost all declared as abstract functions that take points as input, and axioms allow to manipulate these notions. For example, the orthogonal projection of a point $C$ onto the line $AB$ is defined as follows.

```
Variable orthogonalProjection : Point->Point->Point->Point.
Axiom def_orthogonalProjection :
 forall A B C H : Point,
 A <> B ->
 collinear A B H ->
 orthogonal (vect A B) (vect H C) ->
 H = orthogonalProjection A B C.
Axiom def_orthogonalProjection2 :
 forall A B C H : Point,
 A <> B ->
 H = orthogonalProjection A B C ->
 collinear A B H /\ orthogonal (vect A B) (vect H C).
```

Where a variable expresses the orthogonal projection function. The axioms say that if we have collinearity of $A$, $B$ and $H$ and orthogonality of $\overrightarrow{AB}$ and $\overrightarrow{CH}$ then $H$ is the orthogonal projection of $C$ onto the line $AB$ and vice versa.

Using axioms asserting properties of objects to define them is a usual manner in high school. However, this leads to an explosion of the number of axiom and makes the axiomatic system redundant. Moreover, for compound objects, this kind of definitions are not constructive. The existence of the compound objects is stated by axioms and constructing these objects from existing simpler objects is not clear.

In the next section, we introduce a more compact axiom system, we redefine geometric notions and build up a new geometry system. Redundant axioms in the older system are put in the form of theorems to be proved. Theorems and properties that are proved in the older system are maintained. New properties corresponding to new definition of notions are introduced.

# 3 From Affine to 2D-Euclidean Geometry : a Vector Approach

## 3.1 Axiomatic system

The use of mass points is a good approach for computation, similar to what students know. However, students may feel bored in these computations since they can not visualize objects that they are manipulating. So, we use the notion of vector as the key notion in our development. We will define other geometric notions by relations of vectors.

In the last section, the vector is defined as a sum of two mass points in the special case where the sum of mass equals zero. We use a sub-type of mass point to define it in Coq.

```
Definition isVector (v:MassPoint):= exists A, B :Point,
                                 v = add_MP ((−1) A) (1 B).
Record Vector : Type :=
          vecCons { mpOf :MassPoint; proof:isVector mpOf}.
```

The vector data type is constructed from an element *mpOf* having the mass point type with a proof showing that this element can be expressed by the sum of mass point of two certain points $A$ and $B$ with mass values (-1) and 1 respectively. After proving the fact that a linear combination of two mass points satisfying *isVector* predicate also satisfies this predicate, we can easily define the addition operator and the scalar multiplication operator for the vector type. It is clear that we can perform computation over vectors by reusing computations over mass points.

To build up the Euclidean space, we introduce a system of axioms for the scalar product (also called the dot product or the inner product).

Axiom 8_positivity: $\forall\ \overrightarrow{v}$ :Vector, $\overrightarrow{v}\cdot\overrightarrow{v} \geqslant 0$
Axiom 9_positivity2: $\forall\ \overrightarrow{v}$ :Vector, $\overrightarrow{v}\cdot\overrightarrow{v} = 0$ −> $\overrightarrow{v} = \overrightarrow{0}$
Axiom 10_symmetry: $\forall\ \overrightarrow{u}\ \overrightarrow{v}$ :Vector, $\overrightarrow{u}\cdot\overrightarrow{v} = \overrightarrow{v}\cdot\overrightarrow{u}$
Axiom 11_distributivity: $\forall\ \overrightarrow{v_1}\ \overrightarrow{v_2}\ \overrightarrow{v_3}$ :Vector, $(\overrightarrow{v_1}+\overrightarrow{v_2})\cdot\overrightarrow{v_3} = \overrightarrow{v_1}\cdot\overrightarrow{v_3}+\overrightarrow{v_2}\cdot\overrightarrow{v_3}$
Axiom 12_homogeneity: $\forall$ (k : R)($\overrightarrow{u}\ \overrightarrow{v}$ :Vector), $(k\times\overrightarrow{u})\cdot\overrightarrow{v} = k\times(\overrightarrow{u}\cdot\overrightarrow{v})$

To define the Euclidean plane, we introduce an axiom about the existence of three distinct and non-collinear points and another about co-planarity any fourth point with them. These axioms not only ensure that all given points lie on the same plane, but also allow us to define the orientation for planes. The concept of orientation is important because of its use in the definition of trigonometric functions.

```
Variable O O1 O2 :Point.
Axiom 13_exist_3_distinct_notcol_points:
      O ≠ O1 ∧ O ≠ O2 ∧ O ≠ O3 ∧ ¬collinear OO1O2.
Axiom 14_coplanarity: ∀ (M :Point), ∃ k1 k2 :R,
      OM→ = (k1 × OO1→) + (k2 × OO2→).
```

### 3.2 Overview of the structure of the formalization

We construct geometric notions from our primitive notions and prove their properties using our axiom system. Properties and theorems in the library of F. Guilhot are either preserved or reformalized with new definitions of geometric notions. Let's take a look at the Fig. 1 and 2 to have a overview of our formalization.

Figure 1 shows the dependency between concepts in our formalization for affine geometry. Notions in affine geometry are easily formalized from vector. Alignment of three points $A$, $B$ and $C$ is defined by collinearity of $\overrightarrow{AC}$ and $\overrightarrow{AB}$, parallelism of two vectors is defined by collinearity of them. The other ones are also covered such as midpoint, center of gravity of triangle, parallelogram, ...

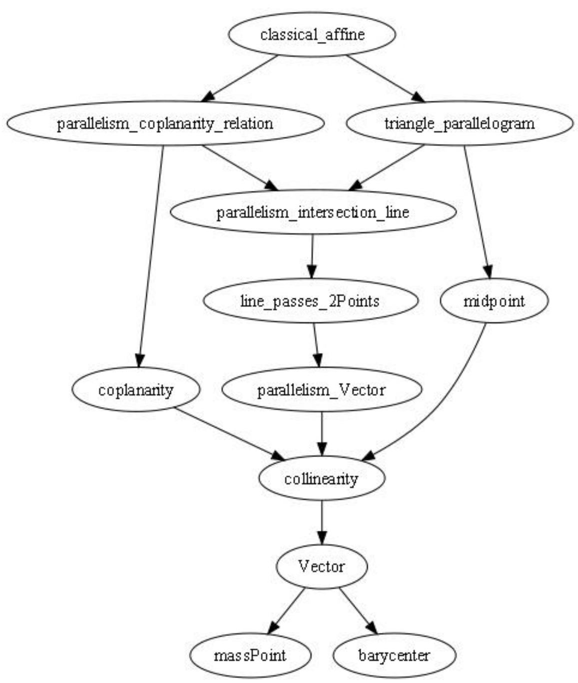



Fig. 1: Formalization of notions in affine geometry

The formalization of plane geometry is presented in Fig. 2, we start with definition of orthogonality of vector and Euclidean distance. Two vectors are orthogonal if their scalar product equals zero. Euclidean distance of two points $A$ and $B$ is expressed by root square of the scalar product of $\overrightarrow{AB}$ with itself.

```
Definition orthogonal (vec1 vec2 :Vector) :=
  scalarProduct vec1 vec2 = 0.
Definition distance (A B :Point) :=
  sqrt (scalarProduct (vector A B) (vector A B)).
```

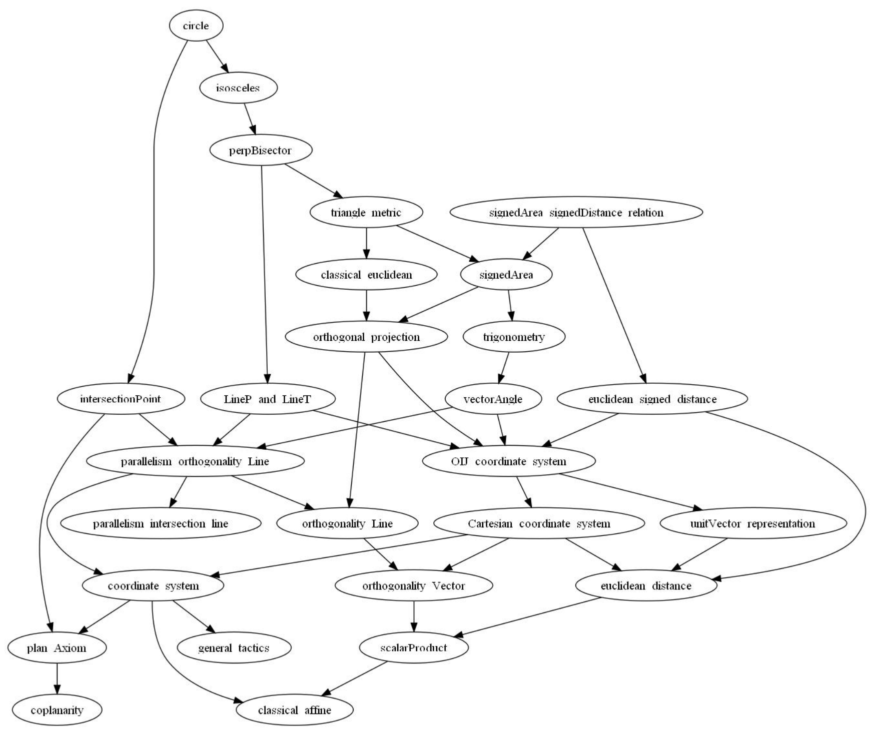



Fig. 2: Formalization of plane geometry

With notions of orthogonality and parallelism of vectors, we can define complete straight line including: line passing through two points, parallel and perpendicular lines. Orthogonal projection and the orthocenter of triangle are also formalized. The Euclidean distance allows us to define unit representation of vector. With the support of orthogonality, we can construct the Cartesian coordinate system. Trigonometric functions and oriented angle are formalized. The signed area and determinant of vectors are also expressed thanks to the trigonometric functions and the angle. The equality of distances allows to define perpendicular bisector, isosceles triangle, circle . . .

Our formalization covers a large part of the geometry curriculum in French high school. Many notions are formalized, their properties are verified. They are considered as basic concepts to produce more complex geometric proofs.

### 3.3 A constructive geometry library

In this section and the following one, we use these notations:

- $\overrightarrow{v_1} \cdot \overrightarrow{v_2}$ for the scalar product of $\overrightarrow{v_1}\overrightarrow{v_2}$
- $|AB|$ for the Euclidean distance from $A$ to $B$
- $\overrightarrow{v}^{\perp}$ for the orthogonal vector of $\overrightarrow{v}$
- $\overline{AB}$ for the signed distance from $A$ to $B$
- $\mathcal{S}_{ABC}$ for the signed area of $\triangle ABC$
- $|\overrightarrow{v}|$ for the measure of $\overrightarrow{v}$

In GeoGebra or other DGS, the user can create and manipulate geometric constructions. The user starts with points and simple constructions, then constructs new geometric object from existing ones. So, to match our library with constructions in DGS, we try to provide primitive constructions which are elementary constructions by rules and compass and mentioned in [9]. They include: the point lying on a given line, the midpoint of two given points, the line passing through two given points, the line passing through a given point and perpendicular to a given line; the line passing through a given point and parallel to a given line, the intersection point of two lines, the circle with a given center passing through a given point, the circle with a given diameter,...

These constructions one by one are defined by vector or by relations over existing notions. The line $AB$ is formalized by $A$ and $\overrightarrow{AB}$. The midpoint $I$ of two points $A$ and $B$ is formalized by constructing $\overrightarrow{AI}$ such that $\overrightarrow{AI} = \frac{1}{2} \times \overrightarrow{AB}$. The circle with center $O$ that passes through $A$ is formalized as the set of points $M$ which distance $OM$ equals distance $OA$...

One interesting formalization here is the one of line. We look for a data structure to express all sorts of line. In Coq, we use a record with a root point and a direction vector to define the line. To ensure existence of line, an element is added to record, this is a proof to show that this direction vector is not the null vector.

```
Record Line:Type := lineCons {rootOf: Point; vecOf: Vector;
 proof: isNotZeroVec vecOf }.
```

The line passing through two points $A$ and $B$ is expressed by a null Line in the case $A = B$, and by the construction of $A$ and $\overrightarrow{AB}$ in the other case. The line which goes through a given point $A$ and which is parallel to another line $a$ (denoted *lineP* $A$ $a$) is expressed by the construction of $A$ and $\overrightarrow{AB}$ where $B$ is constructed such that $\overrightarrow{AB}$ equals the direction vector of line $a$. The perpendicular line (denoted *lineT*) is formalized in the same way.

We define also equality of lines by collinearity of their direction vectors for the case that two lines have the same root point, and by collinearity of their direction vector with the vector constructed by two root points for the case where these points are distinct. Properties concerning equality of lines are verified with this definition, such as:

```
Lemma align_line:
  forall A B C : Point,
 A <> B -> A <> C ->
  collinear A B C -> line A B = line A C.
Lemma liesOnLine_eqLine:
  forall A B C D: Point,
 A <> B -> C <> D ->
  liesOnLine C (line A B) -> liesOnLine D (line A B) ->
  line A B = line C D.
```

For example, to prove the first lemma, we start with the hypotheses $A \neq B$ and $A \neq C$ we have that $\overrightarrow{AB} \neq \overrightarrow{0}$ and $\overrightarrow{AC} \neq \overrightarrow{0}$. By the definition of line $\overrightarrow{AB}$ and $\overrightarrow{AC}$ are direction vectors of line $AB$ and line $AC$ respectively. By the definition of collinearity we have that $\overrightarrow{AB}$ and $\overrightarrow{AC}$ are collinear. So $line\ AB = line\ AC$ by the definition of equality of lines for the case that lines have the same root point.

For compound constructions, we formalize them by the way that they are built from primitive constructions. For example, instead of defining the orthogonal projection with three given points by axioms as mentioned in Section 3, we define an orthogonal projection of a given point $C$ onto a given line $a$. We construct the line that passes through $C$ and which is perpendicular with $a$, we prove that this line and $a$ are not parallel, then we get the intersection point of them.

```
Definition orthogonalProjection(C :Point) (a :Line) :=
  intersectionPoint (lineT C a) a.
```

As we said above, students are familiar with definition of geometric objects by axioms about their properties. So, to avoid losing the pedagogical meaning of the library and to verify if objects are well formalized, we keep the former axioms in the form of theorems, and prove them. Each object is accompanied by a pair of theorems. One allows us to get properties from its definition, the other is to get a definition from properties.

```
Lemma orthogonalProjection_Properties :
  forall A B C H :Point,
 A <> B->
 H = orthogonalProjection C (line A B) ->
  collinear A B H /\ orthogonal (vect A B) (vect H C).
Lemma properties_orthogonalProjection :
  forall A B C H : Point,
 A <> B ->
  collinear A B H ->
  orthogonal (vect A B) (vect H C) ->
 H = orthogonalProjection C (line A B).
```

Proving theorems of the first kind is simpler than proving theorems of the second kind. The latter usually lead to the proof of uniqueness property of objects. Many compound constructions are introduced: the perpendicular bisector of two given points, orthogonal projection of a point onto a line, the circumcircle of three given points, the center of circumcircle, the orthocenter, the center of gravity,...

Reformalizing the geometric objects allows us to eliminate a lot of redundant axioms. Remaining axioms are used to introduce other geometric notions such as: the parallelism of lines, the perpendicularity of lines, the signed area of vectors or of triangle, the vector angles, the trigonometric functions,... These notions are essential for the library manipulating geometric objects and stating geometric problems. Except for vector angles and the trigonometric functions, the rest are formalized without much effort, and many axioms are eliminated.

## 4 Integration with the Area Method

Our library contains a large number of geometric notions and propositions. This allows us to prove many geometry theorems. Constructing a traditional proof consists in finding a sequence in logical steps to the conclusion. But in the context of education, a drawback of constructing fully traditional proofs is that, as mentioned in the introduction section, there are minor goals in interactive proving which are necessary to complete the formal proof but which lead to tedious steps and are not adapted to the level of abstraction at which we usually work with students. Hence we integrate a coordinate-free automatic deduction method with a library for interactive proof. This improves the power and does not decrease pedagogical meaning of the library. The coordinate-free automatic deduction method we chose is the area method of Chou, Gao and Zhang. This method consists in expressing the goal to be proved using three geometric quantities (the signed area of a triangle, the signed distance and the Pythagoras difference), and eliminating the points from the goal in the reverse order of their construction using some elimination lemmas.

To integrate the area method in our development, we need first to ensure the correctness of all the elimination lemmas of the area method and second we need to create a mapping between the definitions of the geometric constructions of the two systems.

### 4.1 Correctness of the area method

The first work in the process of integration is to ensures the correctness of this method in our library. Because the area method is constructed based on its own axiom system mentioned in Table 1, we need only to verify these axioms. Before proving them, we have to make a mapping of primitive notions of this method into our library. In fact, this method only has two primitive notions, being the signed area and the signed distance. Others notions such as the Pythagoras difference, the parallelism, the perpendicularity, the collinearity..., as well as geometric constructions, are defined from these two primitive notions.

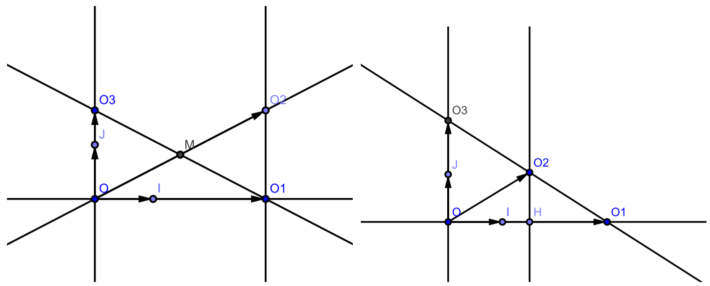
Table 1: The axiom system for the area method

1. $\overline{AB} = 0$ if and only if the points $A$ and $B$ are identical
2. $\mathcal{S}_{ABC} = \mathcal{S}_{CAB}$
3. $\mathcal{S}_{ABC} = -\mathcal{S}_{BAC}$
4. If $\mathcal{S}_{ABC} = 0$ then $\overline{AB} + \overline{BC} = \overline{AC}$ (Chasles' axiom)
5. There are points $A$, $B$, $C$ such that $\mathcal{S}_{ABC} \neq 0$ (dimension; not all points are collinear)
6. $\mathcal{S}_{ABC} = \mathcal{S}_{DBC} + \mathcal{S}_{ADC} + \mathcal{S}_{ABD}$ (dimension; all points are in the same plane)
7. For each element $r$ of $F$, there exists a point $P$, such that $\mathcal{S}_{ABP} = 0$ and $\overline{AP} = r\overline{AB}$ (construction of a point on the line)
8. If $A \neq B, \mathcal{S}_{ABP} = 0, \overline{AP} = r\overline{AB}, \mathcal{S}_{ABP'} = 0$ and $\overline{AP'} = r\overline{AB}$, then $P = P'$ (uniqueness)
9. If $PQ \parallel CD$ and $\frac{\overline{PQ}}{\overline{CD}} = 1$ then $DQ \parallel PC$ (parallelogram)
10. If $\mathcal{S}_{PAC} \neq 0$ and $\mathcal{S}_{ABC} = 0$ then $\frac{\overline{AB}}{\overline{AC}} = \frac{\mathcal{S}_{PAB}}{\mathcal{S}_{PAC}}$ (proportions)
11. If $C \neq D$ and $AB \perp CD$ and $EF \perp CD$ then $AB \parallel EF$
12. If $A \neq B$ and $AB \perp CD$ and $AB \parallel EF$ then $EF \perp CD$
13. If $FA \perp BC$ and $\mathcal{S}_{FBC} = 0$ then $4\mathcal{S}^2_{ABC} = \overline{AF}^2\overline{BC}^2$ (area of a triangle)

The crucial thing in the formalization of these two basic notions is constructing three points $O$, $I$ and $J$ that form a Cartesian coordinate system. With the support of these points, we can define the two primitive notions of the area method in our system.

$O$, $I$ and $J$ form a Cartesian coordinate system, in other words, they satisfy $\overrightarrow{OI} \cdot \overrightarrow{OJ} = 0$ (or $\overrightarrow{OI} \perp \overrightarrow{OJ}$), $|OI| = 1$ and $|OJ| = 1$. The axiom 13 gives us the existence of three non-collinear points $O$, $O_1$ and $O_2$. The axiom system for mass point (the key here is axiom 1 for the barycenter) allows us to produce a new point $C$ from two given points $A$, $B$ and a real number $k$ such that $\overrightarrow{AC} = k \times \overrightarrow{AB}$. We explain how to construct $O$, $I$, $J$ from $O$, $O_1$, $O_2$ using this rule for producing new points.

We first construct a point $O_3$ such that $\overrightarrow{OO_1} \perp \overrightarrow{OO_3}$ and $O \neq O_3$. Let's consider two configurations of $O$, $O_1$, $O_2$ in Fig. 3. For the first case $\overrightarrow{OO_1} \perp \overrightarrow{O_1O_2}$, we first construct M such that $\overrightarrow{OM} = \frac{1}{2} \times \overrightarrow{OO_2}$, we then construct $O_3$ such that $\overrightarrow{O_1O_3} = 2 \times \overrightarrow{O_1M}$. $M$ is the midpoint of $OO_2$ and $O_1O_3$, so $\overrightarrow{OO_3} \parallel \overrightarrow{O_1O_2}$. From the hypothesis $\overrightarrow{OO_1} \perp \overrightarrow{O_1O_2}$ we have $\overrightarrow{OO_3} \perp \overrightarrow{OO_1}$. For the second case $\overrightarrow{OO_1} \not\perp \overrightarrow{O_1O_2}$, we construct $H$ as the orthogonal projection of $O_2$ on $OO_1$. $O_3$ is constructed from $O_1$, $O_2$ by $\overrightarrow{O_1O_3} = \frac{|O_1O|}{|O_1H|} \times \overrightarrow{O_1O_2}$. Thanks to an extension of the Thales theorem for parallel lines in our library, we can prove $\overrightarrow{OO_3} \parallel \overrightarrow{HO_2}$ from $\frac{|O_1H|}{|O_1O|} = \frac{|O_1O_2|}{|O_1O_3|}$. From the definition of H, we get $\overrightarrow{O_2H} \perp \overrightarrow{OO_1}$, therefore $\overrightarrow{OO_3} \perp \overrightarrow{OO_1}$. $O \neq O_3$ in the two cases is easily verified thanks to distinctions of $O$, $O_1$, $O_2$ in the axiom 13.




(a) $OO_1 \perp O_1O_2$ (b) $OO_1 \not\perp O_1O_2$

Fig. 3

Now we construct $I$ from $O$, $O_1$ and $J$ from $O$, $O_3$ by $\overrightarrow{OI} = \frac{1}{|OO_1|} \times \overrightarrow{OO_1}$ and $\overrightarrow{OJ} = \frac{1}{|OO_3|} \times \overrightarrow{OO_3}$ respectively. We have $O \neq O_1$ (axiom 13), hence $\overrightarrow{OI}$ is a unit representation of $\overrightarrow{OO_1}$, and we have $|OI| = 1$. By the same way, we have $\overrightarrow{OJ}$ is a unit representation of $\overrightarrow{OO_3}$, and we have $|OJ| = 1$. By the construction of $O_3$) explained above, we have $\overrightarrow{OO_1} \perp \overrightarrow{OO_3}$, so we can prove $\overrightarrow{OI} \perp \overrightarrow{OJ}$ $\square$.

Three points $O$, $I$, $J$ offer an orientation for the Euclidean plane. This is used to define sign of the signed distance $\overline{AB}$. The magnitude of $\overline{AB}$ is defined by their Euclidean distance $|AB|$. The definition is as follows:

$$\begin{cases} \text{if } \overrightarrow{AB} \cdot \overrightarrow{OI} > 0 & \overline{AB} = |AB| \\ \text{if } \overrightarrow{AB} \cdot \overrightarrow{OI} < 0 & \overline{AB} = -|AB| \\ \text{if } \overrightarrow{AB} \cdot \overrightarrow{OI} = 0 & \begin{cases} \text{if } \overrightarrow{AB} \cdot \overrightarrow{OJ} > 0 & \overline{AB} = |AB| \\ \text{if } \overrightarrow{AB} \cdot \overrightarrow{OJ} < 0 & \overline{AB} = -|AB| \\ \text{if } \overrightarrow{AB} \cdot \overrightarrow{OJ} = 0 & \overline{AB} = 0 \end{cases} \end{cases}$$

On the other hand, the Cartesian coordinate system with the new points $O$, $I$, $J$ enables us to construct the orthogonal vector for the given vector.

$\overrightarrow{v}^{\perp} := (-\overrightarrow{v} \cdot \overrightarrow{OJ}) \times \overrightarrow{OI} + (\overrightarrow{v} \cdot \overrightarrow{OI}) \times \overrightarrow{OJ}$.

We can then formalize the trigonometric functions of two vectors and the signed area of a triangle as follows:

$\cos \overrightarrow{v_1}\overrightarrow{v_2} = \frac{\overrightarrow{v_1} \cdot \overrightarrow{v_2}}{|\overrightarrow{v_1}| \times |\overrightarrow{v_2}|}$ and $\sin \overrightarrow{v_1}\overrightarrow{v_2} = \frac{\overrightarrow{v_1}^{\perp} \cdot \overrightarrow{v_2}}{|\overrightarrow{v_1}^{\perp}| \times |\overrightarrow{v_2}|}$

$\mathcal{S}_{ABC} = \frac{1}{2} \times |AB| \times |AC| \times \sin \overrightarrow{AB}\overrightarrow{AC}$

Two primitive notions of the area method are formalized, many properties are also proved. They are properties of the signed distance $\overrightarrow{AB} = k \times \overrightarrow{CD} \leftrightarrow \overline{AB} = k \times \overline{CD} \wedge \overrightarrow{AB} \parallel \overrightarrow{CD}$, $col\ A\ B\ C \rightarrow \overline{AB} + \overline{BC} = \overline{AC}$; properties of the orthogonal vector $(\overrightarrow{v_1} + \overrightarrow{v_2})^{\perp} = \overrightarrow{v_1}^{\perp} + \overrightarrow{v_2}^{\perp}$, $(k \times \overrightarrow{v})^{\perp} = k \times \overrightarrow{v}^{\perp}$, $|\overrightarrow{v}^{\perp}| = |\overrightarrow{v}|$, $\overrightarrow{v_1} \cdot \overrightarrow{v_2}^{\perp} = -\overrightarrow{v_1}^{\perp} \cdot \overrightarrow{v_2}$; as well as properties of signed area $\mathcal{S}_{ABC} = \frac{1}{2} \times \overrightarrow{AB}^{\perp} \cdot \overrightarrow{AC}$, $\mathcal{S}_{ABC} = \mathcal{S}_{BCA}$, $\mathcal{S}_{ABC} = -\mathcal{S}_{ACB}$... They serve as basic properties to prove the axiom system in Table 1. We do not intend to detail long proofs, we give only the following proof of the property 6: $\mathcal{S}_{DAB} + \mathcal{S}_{DBC} + \mathcal{S}_{DCA} = \mathcal{S}_{ABC}$ presented with two column method as an example. The full Coq proofs consists of approximately 1500 lines of Coq tactics.

| Property | Reasoning |
|---|---|
| 1. $\mathcal{S}_{DAB} = \mathcal{S}_{BDA} = \frac{1}{2} \times \overrightarrow{BD}^{\perp} \cdot \ \overrightarrow{BA}$ | signed area props |
| 2. $\mathcal{S}_{DBC} = \mathcal{S}_{BCD} = \frac{1}{2} \times \overrightarrow{BC}^{\perp} \cdot \ \overrightarrow{BD}$ | signed area props |
| 3. $\frac{1}{2} \times \overrightarrow{BC}^{\perp} \cdot \ \overrightarrow{BD} = -\frac{1}{2} \times \overrightarrow{BC} \cdot \ \overrightarrow{BD}^{\perp}$ | orthor vect props |
| 4. $-\frac{1}{2} \times \overrightarrow{BC} \cdot \ \overrightarrow{BD}^{\perp} = \frac{1}{2} \times \overrightarrow{CB} \cdot \ \overrightarrow{BD}^{\perp}$ | orthor vect props |
| 5. $\mathcal{S}_{DBC} = \frac{1}{2} \times \overrightarrow{CB} \cdot \ \overrightarrow{BD}^{\perp}$ | from (2) (3) (4) |
| 6. $\mathcal{S}_{DAB} + \mathcal{S}_{DBC} = \frac{1}{2} \times \overrightarrow{BD}^{\perp} \cdot \ \overrightarrow{BA} + \frac{1}{2} \times \overrightarrow{CB} \cdot \ \overrightarrow{BD}^{\perp}$ | from (1) (5) |
| 6. $\mathcal{S}_{DAB} + \mathcal{S}_{DBC} = \frac{1}{2} \times \overrightarrow{BD}^{\perp} \cdot \ (\overrightarrow{BA} + \overrightarrow{CB})$ | distrib prop |
| 7. $\mathcal{S}_{DAB} + \mathcal{S}_{DBC} = \frac{1}{2} \times \overrightarrow{BD}^{\perp} \cdot \ \overrightarrow{CA}$ | $(\overrightarrow{BA} + \overrightarrow{CB}) = \overrightarrow{CA}$ |
| 8. $\mathcal{S}_{DAB} + \mathcal{S}_{DBC} = \frac{1}{2} \times (\overrightarrow{BC} + \overrightarrow{CD})^{\perp} \cdot \overrightarrow{CA}$ | $\overrightarrow{BD}^{\perp} = \overrightarrow{BC} + \overrightarrow{CD}$ |
| 9. $\mathcal{S}_{DAB} + \mathcal{S}_{DBC} = \frac{1}{2} \times (\overrightarrow{BC}^{\perp} \cdot \overrightarrow{CA} + \overrightarrow{CD}^{\perp} \cdot \overrightarrow{CA})$ | $\overrightarrow{BD}^{\perp} = \overrightarrow{BC} + \overrightarrow{CD}$ |
| 10. $\mathcal{S}_{DAB} + \mathcal{S}_{DBC} = \frac{1}{2} \times (-\overrightarrow{CB}^{\perp} \cdot \overrightarrow{CA} + \overrightarrow{CD}^{\perp} \cdot \overrightarrow{CA})$ | $\overrightarrow{BC}^{\perp} = -\overrightarrow{CB}^{\perp}$ |
| 11. $\mathcal{S}_{DAB} + \mathcal{S}_{DBC} = -\mathcal{S}_{CBA} + \mathcal{S}_{CDA}$ | signed area props |
| 12. $\mathcal{S}_{DAB} + \mathcal{S}_{DBC} = \mathcal{S}_{CAB} - \mathcal{S}_{CAD}$ | signed area props |
| 13. $\mathcal{S}_{DAB} + \mathcal{S}_{DBC} + \mathcal{S}_{CAD} = \mathcal{S}_{CAB}$ | signed area props |
| 14. $\mathcal{S}_{DAB} + \mathcal{S}_{DBC} + \mathcal{S}_{DCA} = \mathcal{S}_{ABC}$ | signed area props |
| Qed. | |

### 4.2 Usability of the area method

This method consists in expressing the goal to be proved using three geometric quantities (the signed area of a triangle, the signed distance and the Pythagoras difference), and eliminating the points from the goal in the reverse order of their construction using some elimination lemmas. It deals with problems stated in terms of sequences of specific geometric construction steps. So to make it runnable in our library, we have to convert geometry statements in our library to its statements. Precisely, from sequences of geometric constructions of our library, we have to construct sequences in the area method.

The first step in this process is to normalize the constructions in our library. Compound constructions are unfolded and replaced by sequences of primitive constructions. These sequences are also reduced without loosing their semantics. For example, a line passing through $A$ that is parallel with perpendicular bisector of $BC$ is simplified to a line passing through $A$ that is perpendicular to line $BC$.

In the second step, we try to extract constructions of the area method from the sequence of constructions we get in the first step. Let's consider constructions of the area method. In fact, each construction aims to create a geometric object with a precise semantics. For example, *on_inter_line_perp* $Y\ R\ U\ V\ P\ Q$ is a construction in the area method and defined by

```
Definition on_inter_line_perp (Y R U V P Q : Point):=
  Col Y U V /\ perp Y R P Q /\ ~ perp P Q U V.
```

It means that $Y$ is at the intersection of $UV$ and the perpendicular to $PQ$ going through $R$. With the semantics of construction, we can construct a sequence of primitive constructions of our library that produces this object. For each constructions, a lemma is introduced to ensure that sequences of primitive constructions give us exactly the object defined by this construction. In our example, the corresponding sequence of the construction *on_inter_line_perp Y R U V P Q* is *Y = intersectionPoint (lineT R (line P Q)) (line U V)*. The lemma that ensures exactness is as follows:

```
Lemma constr_on_inter_line_perp:
  forall Y R U V P Q : Point,
  P <> Q -> U <> V ->
  ~ perpendicular (line P Q) (line U V) ->
  Y = intersectionPoint (lineT R (line P Q)) (line U V) ->
  on_inter_line_perp Y R U V P Q .
```

To complete this step, we write tactics to automatically introduce constructions of the area method when their corresponding sequence of primitive constructions appears in hypotheses. The following tactic is for our example:

```
Ltac convert_to_AMConstructions_12 :=
  repeat match goal with
   | H: ?Y =
  intersectionPoint (lineT ?R (line ?P ?Q)) (line ?U ?V)|-_
  => try (assert (on_inter_line_perp Y R U V P Q) by
     (apply (@constr_on_inter_line_perp Y R U V P Q);
     auto with geo);revert H)
  end;
  intros.
```

## 5 Conclusion

Our development provides a library in Coq for interactive and automated geometry theorem proving. As far as we know, this is the first system which integrate both *interactive* and *automatic* theorem proving in geometry in a formal setting. We give an axiom system, from which we build up Euclidean plane geometry. We then verify the axiom system of the area method and integrate this method into our library. In this library, geometric objects are formalized in a constructive manner. Proof of properties, lemmas, and theorems are formal and traditional proofs, they are adapted to student knowledge. The combination with the automated deduction method offers flexibility and power in proving geometry theorem. The integration of this library in a dynamic geometry tool offers advantages in education. It allows student to build proofs themselves. In addition, it highlights the structure of the proofs. Student have a logical view of geometry problems and understands what are the hypotheses and the goal. This work could also be used to study the combination of different automatic theorem proving methods in geometry.

In the future, we will study the integration of algebraic approaches to automatic theorem proving in geometry such as Gröbner bases and Wu's method. To be useful in high-school, we will also need to extend our proof system to allow proofs which look less formal. For that purpose we will study how to deal automatically with degenerated cases.

## References


1. Berger, M., Pansu, P., Berry, J.P., Saint-Raymond, X.: Problems in Geometry. Springer (1984)
2. Bertot, Y., Guilhot, F., Pottier, L.: Visualizing geometrical statements with GeoView. Electronic Notes in Theoretical Computer Science 103, 49–65 (September 2003)
3. Chou, S.C., Gao, X.S., Zhang, J.Z.: Machine Proofs in Geometry. World Scientific, Singapore (1994)
4. Coq development team: The Coq Proof Assistant Reference Manual, Version 8.3. TypiCal Project (2010), http://coq.inria.fr
5. Dehlinger, C., Dufourd, J.F., Schreck, P.: Higher-Order Intuitionistic Formalization and Proofs in Hilbert's Elementary Geometry. In: ADG'00. LNAI, vol. 2061, pp. 306–324. Springer-Verlag (2000)
6. Geogebra development team: Introduction to GeoGebra, http://www.geogebra.org/book/intro-en/
7. Guilhot, F.: Formalisation en Coq et visualisation d'un cours de géométrie pour le lycée. TSI 24, 1113–1138 (2005), in french
8. Hilbert, D.: Les fondements de la géométrie. Dunod, Paris, Jacques Gabay edn. (1971), edition critique avec introduction et compléments préparée par Paul Rossier
9. Janičić, P.: Geometry construction language. Journal of Automated Reasoning 44, 3–24 (2010)
10. Janicic, P., Narboux, J., Quaresma, P.: The Area Method : a Recapitulation. Journal of Automated Reasoning (2010)
11. Meikle, L., Fleuriot, J.: Formalizing Hilbert's Grundlagen in Isabelle/Isar. In: Theorem Proving in Higher Order Logics. pp. 319–334 (2003)
12. Narboux, J.: A decision procedure for geometry in Coq. In: TPHOLs'04. LNCS, vol. 3223, pp. 225–240. Springer-Verlag (2004)
13. Narboux, J.: Toward the use of a proof assistant to teach mathematics. In: Proceedings of the 7th International Conference on Technology in Mathematics Teaching (ICTMT7) (2005)
14. Narboux, J.: A graphical user interface for formal proofs in geometry. J. Autom. Reasoning 39(2), 161–180 (2007)
15. Narboux, J.: Mechanical theorem proving in Tarski's geometry. In: ADG'06. LNAI, vol. 4869, pp. 139–156. Springer-Verlag (2007)
16. Pham, T.M., Bertot, Y.: A combination of a dynamic geometry software with a proof assistant for interactive formal proofs. In: UITP Workshops '10 (2010)
17. Scott, P., Fleuriot, J.: Idle time discovery in geometry theorem proving. In: Proceedings of ADG'10 (2010)
18. Tarski, A.: What is Elementary Geometry? In: Henkin, L., Suppes, P., Tarski, A. (eds.) The axiomatic Method, with special reference to Geometry and Physics. pp. 16–29. North-Holland, Amsterdam (1959)