A Decision Procedure for Geometry in Coq

Julien Narboux

A Decision Procedure for Geometry in Coq

Julien Narboux

LIX, École Polytechnique*
Julien.Narboux@inria.fr

Abstract We present in this paper the development of a *decision procedure* for affine plane geometry in the Coq proof assistant. Among the existing decision methods, we have chosen to implement one based on the *area method* developed by Chou, Gao and Zhang, which provides *short* and *"readable"* proofs for geometry theorems. The idea of the method is to express the goal to be proved using three geometric quantities and eliminate points in the reverse order of their construction thanks to some *elimination lemmas*.

1 Introduction

Geometry is one of the most successful areas of automated theorem proving. Many difficult theorems can be proved by computer programs using synthetic and algebraic methods. A decision procedure using quantifier elimination was first introduced by A. Tarski [15]. His method was further improved by Collins' cylindrical decomposition algorithm [4]. Among the efficient methods we can cite also the algebraic method of Wu which succeeded in finding the proofs of hundreds of geometry theorems [3, 18] and later the method of Chou, Gao, Zhang which produces short and readable proofs [2] (they are readable in the sense that one can understand these proofs without difficulty as they manipulate small terms.)
Recently, developments have also been produced towards the formalization of elementary geometry in proof assistants: Hilbert's *Grundlagen* [10] have been formalized in Isabelle/Isar by Laura Meikle and Jacques Fleuriot [14], and by Christophe Dehlinger in the Coq system [5]. Gilles Kahn has formalized Jan von Plato's constructive geometry in the Coq system [12, 17]. Frédérique Guilhot has done a large development in Coq dealing with French high school geometry [7].
We believe that automated theorem proving and interactive proof development are complementary to formal proof generation. Proof assistants can deal with a very large span of theorems, but they need automation to ease the development. The goal of this work is to bring the level of automation provided by the method of Chou, Gao and Zhang to the Coq proof assistant [13]. This is done by implementing the decision procedure as a Coq tactic. A tactic is a program

* Projet LogiCal, Pôle Commun de Recherche en Informatique, plateau de Saclay, École Polytechnique, CNRS, INRIA Futurs, Université Paris Sud.

which expresses the sequence of the basic logical steps needed to formally prove a theorem.

Formalizing a decision procedure within Coq, has not only the advantage of simplifying the tedious task of proving geometry theorems but also allows us to combine the geometrical proofs provided by the tactic with arbitrary complicated proofs developed interactively using the full strength of the underlying logic of the theorem prover. For instance, theorems involving induction over the number of points in the figure can be formalized in Coq. This approach has also the advantage of providing a higher level of reliability than *ad hoc* geometry theorem provers because the proofs generated by our tactic are double checked by the Coq internal proof-checker.

The issues related to the treatment of nondegeneracy conditions are crucial; this is emphasized in our formalization.

This paper is arranged as follows: we will first give an overview of the decision method, and then we will explain how it has been implemented in the Coq proof assistant.

2 The Chou, Gao and Zhang Decision Procedure

Chou, Gao and Zhang's decision procedure is the mechanization of the *area method*. It is a mix of algebraic and synthetic methods. The idea of the method is to express the goal in a constructive way and treat the points in the reverse order of their construction. The treatment of each point consists in eliminating every occurrence of the point in the goal. This can be done thanks to the *elimination lemmas*.

To be in the language of the procedure, the goal to be proved must verify two conditions: first the theorem has to be stated as a sequence of constructions (constructing points as intersections of lines or on the parallel to a line passing through a point, *etc.*)[1]. Second, the goal must be expressed as an arithmetic expression using only three geometric quantities: the ratio of two oriented distances ($\frac{\overline{AB}}{\overline{CD}}$) with AB parallel to CD, the signed area of a triangle ($\mathcal{S}_{ABC}$) and the Pythagoras difference (the difference between the sum of the squares of two sides of a triangle and the square of the other side $\mathcal{P}_{ABC} = \overline{AB}^2 + \overline{BC}^2 - \overline{AC}^2$)[2]. These three geometric quantities are sufficient to deal with a large part of plane geometry as shown in Table 1 on page 4. They verify elementary properties such as $\mathcal{S}_{AAB} = 0$, $\mathcal{S}_{ABC} = -\mathcal{S}_{BAC}$ and $\mathcal{S}_{ABC} = \mathcal{S}_{BCA}$. That will be made explicit in Sect. 3. For the time being only the first two geometric quantities are formalized in our development in Coq, it means that we can only deal with affine plane

[1] Note that it is subject to conditions (that may not be decidable) and that constructive in this context does not mean the same as constructible with ruler and compass (this will be detailed later in Sect. 3.1). Note also that different constructions can lead to slightly different nondegeneracy conditions and so slightly different theorems.

[2] Note that $\mathcal{P}_{ABC} = -2(\overrightarrow{AB}.\overrightarrow{BC})$ and $4\mathcal{S}^2_{ABC} = (\overrightarrow{AB} \wedge \overrightarrow{BC})^2$ where . is dot product and $\wedge$ is vector cross product.

geometry. The formulas treated by our tactic are those of the form:

$$\forall A_1, \ldots A_n : Point,\, C_i(A_o, \ldots, A_p) \rightarrow \ldots \rightarrow C_j(A_q, \ldots, A_r) \rightarrow R = 0$$

where R is an arithmetical expression containing signed areas and ratios and C_i are predicates expressing the sequence of constructions. For each constructed point there is some C_i stating how it has been constructed. Note that the dependency graph of the constructions must be cycle free.
To eliminate a point from the goal we need to apply one of the elimination lemmas shown on Table 2 on page 5. This table can be read as follows: To eliminate a point Y, choose the line corresponding to the way Y has been constructed, and apply the formula given in the column corresponding to the geometric quantity in which Y is used. The lemmas rewrite any geometric quantity containing an occurrence of a point Y ($\mathcal{S}_{ABY}$ or $\frac{\overline{AY}}{\overline{CD}}$ for any A,B,C and D such that $AY \parallel CD$.) into an expression with no occurrence of Y[3]. There is one lemma for each combination of construction and geometric quantity. As far as geometry of incidence is concerned, we have five ways to construct a point and two geometric quantities; this provides ten elimination lemmas. Note that there are more constructions than needed (some constructions can be expressed using others). This is used to simplify the statement of the theorems and shorten the proofs by providing specific elimination lemmas for non primitive constructions. The constructions involving a quantity λ can be used to build a point at some fixed distance (if λ is instantiated) or at any distance (if λ is kept as a variable). These last constructions are used to build what are called "semi-fixed points".
When all the constructed points have disappeared from the goal, the result is an arithmetic expression containing geometric quantities using only free points (free points are those that can be freely moved in the plane, those whose position can be arbitrarily chosen while drawing the figure). At this step these geometric quantities use only free points but are not necessarily independent. In case these geometric quantities are not independent we decompose them using three non collinear points (that can be seen as a base). This will be detailed in Sect. 3. If all the geometric quantities are independent, the goal can be seen as an equation between two polynomials, which can be easily decided.
The steps of the method can be summarized using this informal description:

– express the goal in a *constructive* way (as a sequence of basic constructions) using only the three geometric quantities;
– remove bound points from the goal using the *elimination lemmas*;
– change the goal into an expression containing only *independent* geometric quantities;
– *decide* if the resulting equality is universally true or not.

[3] Note that every occurrence of Y is removed only if the points present in the geometric quantity containing Y (A,B,C and D) are different from Y, this problem is treated in the implementation.

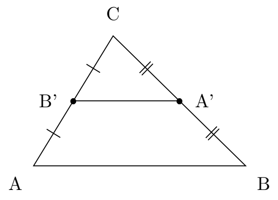
Table 1. Expressing some common geometric notions using $\mathcal{S}$, ratios and $\mathcal{P}$

Geometric notions	**Formalization**
A,B and C are collinear	$\mathcal{S}_{ABC} = 0$
$AB \parallel CD$	$\mathcal{S}_{ABC} = \mathcal{S}_{ABD}$
I is the midpoint of AB	$\frac{\overline{AB}}{\overline{AI}} = 2 \wedge \mathcal{S}_{ABI} = 0$
$AB \perp BC$	$\mathcal{P}_{ABC} = 0$
$AB \perp CD$	$\mathcal{P}_{ACD} = \mathcal{P}_{BCD}$
$A = B$	$\mathcal{P}_{ABA} = 0$

2.1 Example

Let's consider the midpoint theorem as an example:

Example 1 (Midpoint theorem). Let ABC be a triangle, and let A' and B' be the midpoints of BC and AC respectively. Then the line $A'B'$ is parallel to the base AB.



Proof (using the method). We first translate the goal ($A'B' \parallel AB$) into its equivalent using the signed area:

$$\mathcal{S}_{A'B'A} = \mathcal{S}_{A'B'B}$$

Then we eliminate compound points from the goal starting by the last point in the order of their construction. The geometric quantities containing an occurrence of B' are $\mathcal{S}_{A'B'B}$ and $\mathcal{S}_{A'B'A}$, B' has been constructed using the first construction on Table 2 with $\lambda = \frac{1}{2}$:

$$\mathcal{S}_{A'B'A} = \mathcal{S}_{AA'B'} = \frac{1}{2}\mathcal{S}_{AA'A} + \frac{1}{2}\mathcal{S}_{AA'C} = \frac{1}{2}\mathcal{S}_{AA'C}$$

and

$$\mathcal{S}_{A'B'B} = \mathcal{S}_{BA'B'} = \frac{1}{2}\mathcal{S}_{BA'A} + \frac{1}{2}\mathcal{S}_{BA'C}$$

The new goal is

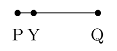
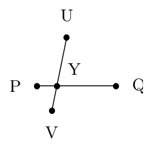
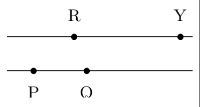
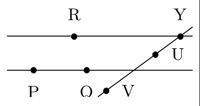
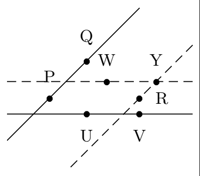
$$\mathcal{S}_{AA'C} = \mathcal{S}_{BA'A} + \mathcal{S}_{BA'C}$$

Now we eliminate A' using:

$$\mathcal{S}_{CAA'} = \frac{1}{2}\mathcal{S}_{CAB} + \frac{1}{2}\mathcal{S}_{CAC} = \frac{1}{2}\mathcal{S}_{CAB}$$

$$\mathcal{S}_{ABA'} = \frac{1}{2}\mathcal{S}_{ABB} + \frac{1}{2}\mathcal{S}_{ABC} = \frac{1}{2}\mathcal{S}_{ABC}$$

Table 2. Elimination lemmas

Construction	Description	Elimination formulas	
	(Nondegeneracy condition)	$\mathcal{S}_{ABY} =$	If $AY \parallel CD$ then $\frac{\overline{AY}}{\overline{CD}} =$
	Take Y on line PQ such that $\frac{\overline{PY}}{\overline{PQ}} = \lambda$. $(P \neq Q)$	$\lambda S_{ABQ} + (1-\lambda)\mathcal{S}_{ABP}$	$\begin{cases} \frac{\frac{\overline{AP}}{\overline{PQ}}+\lambda}{\frac{\overline{CD}}{\overline{PQ}}} & \text{if } A \in PQ \\ \frac{\mathcal{S}_{APQ}}{\mathcal{S}_{CPDQ}} & \text{otherwise}^{a}. \end{cases}$
	Take Y at the intersection of PQ and UV. $(PQ \nparallel UV)$	$\frac{\mathcal{S}_{PUV}\mathcal{S}_{ABQ}+\mathcal{S}_{QVU}\mathcal{S}_{ABP}}{\mathcal{S}_{PUQV}}$	$\begin{cases} \frac{\mathcal{S}_{AUV}}{\mathcal{S}_{CUDV}} & \text{if } A \notin UV \\ \frac{\mathcal{S}_{APQ}}{\mathcal{S}_{CPDQ}} & \text{otherwise.} \end{cases}$
	Take Y on the parallel to PQ passing through R such that $\frac{\overline{RY}}{\overline{PQ}} = \lambda$. $(P \neq Q)$	$\mathcal{S}_{ABR} + \lambda\mathcal{S}_{APBQ}$	$\begin{cases} \frac{\frac{\overline{AR}}{\overline{PQ}}+\lambda}{\frac{\overline{CD}}{\overline{PQ}}} & \text{if } A \in RY \\ \frac{\mathcal{S}_{APRQ}}{\mathcal{S}_{CPDQ}} & \text{otherwise.} \end{cases}$
	Take Y at the intersection of UV and the parallel to PQ passing through R. $(PQ \nparallel UV)$	$\frac{\mathcal{S}_{PUQR}\mathcal{S}_{ABV}-\mathcal{S}_{PVQR}\mathcal{S}_{ABU}}{\mathcal{S}_{PUQV}}$	$\begin{cases} \frac{\mathcal{S}_{AUV}}{\mathcal{S}_{CUDV}} & \text{if } A \notin UV \\ \frac{\mathcal{S}_{APRQ}}{\mathcal{S}_{CPDQ}} & \text{otherwise.} \end{cases}$
	Take Y at the intersection of the parallel to PQ passing through R and the parallel to UV passing through W. $(PQ \nparallel UV)$	$\frac{\mathcal{S}_{PWQR}}{\mathcal{S}_{PUQV}}.\mathcal{S}_{AUBV} + \mathcal{S}_{ABW}$	$\begin{cases} \frac{\mathcal{S}_{APRQ}}{\mathcal{S}_{CPDQ}} & \text{if } AY \nparallel PQ \\ \frac{\mathcal{S}_{AUWV}}{\mathcal{S}_{CUDV}} & \text{otherwise.} \end{cases}$

[a] $\mathcal{S}_{ABCD}$ is a notation for $\mathcal{S}_{ABC} + \mathcal{S}_{ACD}$.

$$\mathcal{S}_{CBA'} = \frac{1}{2}\mathcal{S}_{CBB} + \frac{1}{2}\mathcal{S}_{CBC} = 0$$

The new goal is:

$$\frac{1}{2}\mathcal{S}_{CAB} = \frac{1}{2}\mathcal{S}_{ABC}$$

The proof is completed as $\mathcal{S}_{CAB} = \mathcal{S}_{ABC}$.

3 Implementation in Coq

The formalization of the procedure consists in choosing an axiomatic, proving the propositions needed by the tactic and writing the tactic itself. These three steps are described in this section, but to ease the development, in our implementation we have intermixed the proofs of the propositions and the tactics. Our tactic is decomposed into sub-tactics performing the following tasks (we will give their precise description later):

- initialization;
- simplification;
- unification;
- elimination;
- conclusion.

The simplification and unification tactics are used to prove some propositions needed by the other sub-tactics.
Our tactic is mainly implemented using the Ltac language included in the Coq system. This language provides primitives to describe Coq tactics within Coq itself (without using Ocaml, the implementation language of Coq). But some of our sub-tactics are implemented using the reflection method [11, 9, 1]. This method consists of reflecting a subset of the Coq language (here the arithmetical expressions build on the geometric quantities) into an object of the Coq language itself (in our case an inductive type denoting arithmetical expressions). This means that the computation performed by the traditional tactic in some metalanguage (Ltac or Ocaml) is here done using the internal reduction of Coq. The reflexive tactic is composed of:

- a small piece of Ltac (or Ocaml) to reflect the object language into the metalanguage,
- a Coq term which solves the problem expressed in the metalanguage,
- a Coq term which reflects the metalanguage into the object language,
- and the proof of the validity of the transformation performed by this term.

This method has the advantage of producing more efficient tactics and shorter proofs because the application of the tactic is just one computation step (using the conversion rule of the calculus of inductive constructions).
We have used the reflection method to implement the simplification and unification tactics. We have not chosen to use the reflection method for the whole tactic for two reasons:

1. We believe that the proof process would not be much faster and the generated proof would be comparable in size. Indeed the proofs generated by our tactic are roughly a sequence of the few applications of the elimination lemmas.
2. Expressing the tactic as a Coq term, and proving the validity of this transformation would have been cumbersome. We make heavy use of the high level primitives provided by the Ltac language such as matching the context for terms or sub-terms, clearing hypotheses *etc.* All this machinery and the proof of its validity would have to be developed within Coq to use the reflection method on the whole tactic.

3.1 The axiomatic

There are many axiomatics for elementary geometry. The best known are the axiomatics of Euclid, Tarski and Hilbert [6, 16, 10]. The axiomatic used to formalize this decision method in Coq is inspired by the axiomatic of Chou, Gao and Zhang and is given in Table 3 on the following page. We could define this axiomatic as an undirected, semi-analytic, axiomatic with points as primitive objects. We mean by these adjectives that:

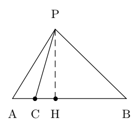
– This axiomatic has the property of being unordered, this simplifies the treatment of a lot of cases but it has the drawback that one cannot express the *Between* predicate which can be found in Tarski[4].
– This axiomatic contains the axioms of a field. This means that there is some notion of numbers, but it is still coordinate free.
– This axiomatic has the characteristic of being based on points: lines are not primitive objects as in Hilbert's axiomatic for example which contains not only *Points* but also *Lines* as primitive objects. This means that we can not quantify over the set of lines, *etc.*

The first axiom is the fact that we have a set of points.
We assume that we have a field of characteristic different from two. The axioms of a field are standard and hence omitted. The fact that the characteristic is different from two is used first to simplify the axiomatic (because $2 \neq 0 \wedge \mathcal{S}_{ABC} = -\mathcal{S}_{BAC} \rightarrow \mathcal{S}_{AAC} = 0$) and second to allow the construction of the midpoint[5] of a segment without explicitly stating that two is different from zero.
We assume that we have one binary function ($\overline{AB}$) and one ternary ($\mathcal{S}_{ABC}$) from points to our field (F). The first depicts the signed distance between A and B, the second represents the signed area of the triangle A,B,C.
The axioms of dimension express that all points are in the same plane and not

[4] This issue can be addressed using an ordered field. In this case we can express the *Between* predicate and we could generalize the procedure to allow the treatment of a goal which is an inequality. But the procedure could still not deal with inequalities in the hypotheses.
[5] The midpoint is given such attention because it is used to prove the validity of constructions involving parallel lines.

all points are collinear.
The axioms of construction express that we can build a point on a line determined by two points A and B at some given distance. The given distance is not necessarily a “constructive distance” so the notion of theorems stated constructively is not the same as the notion of constructible with ruler and compass. The constructed point is unique if A is different from B.
The axiom of proportions is central and gives a relation between oriented distances and signed areas.

Table 3. The axiomatic

Points	Point : Set
Field	F is a field $2 \neq 0$
Signed distance	$\overline{\cdot}$: Point $\rightarrow$ Point $\rightarrow F$ $\overline{AB} = 0 \iff A = B$
Signed area	$\mathcal{S}$: Point $\rightarrow$ Point $\rightarrow$ Point $\rightarrow F$ $\mathcal{S}_{ABC} = \mathcal{S}_{CAB}$ $\mathcal{S}_{ABC} = -\mathcal{S}_{BAC}$
Chasles'axiom	$\mathcal{S}_{ABC} = 0 \rightarrow \overline{AB} + \overline{BC} = \overline{AC}$
Dimension	$\exists A, B, C$: Point, $\mathcal{S}_{ABC} \neq 0$ $\mathcal{S}_{ABC} = \mathcal{S}_{DBC} + \mathcal{S}_{ADC} + \mathcal{S}_{ABD}$
Construction	$\forall r : F \; \exists P : Point,\ \mathcal{S}_{ABP} = 0 \wedge \overline{AP} = r\overline{AB}$ $A \neq B \begin{array}{l} \wedge\ \mathcal{S}_{ABP} = 0 \wedge \overline{AP} = r\overline{AB} \\ \wedge\ \mathcal{S}_{ABP'} = 0 \wedge \overline{AP'} = r\overline{AB} \end{array} \rightarrow P = P'$
Proportions	$A \neq C \rightarrow \mathcal{S}_{PAC} \neq 0 \rightarrow \mathcal{S}_{ABC} = 0 \rightarrow \frac{\overline{AB}}{\overline{AC}} = \frac{\mathcal{S}_{PAB}}{\mathcal{S}_{PAC}}$ 

Our axiomatic differs in some points with the axiomatic of Chou, Gao and Zhang.
First we do not assume that we have a notion of collinearity, this notion is defined using the signed area. In [2] the notion of collinearity of three points A,B and C is used to express some axioms and then proved to be equivalent to $\mathcal{S}_{ABC} = 0$.
Second we can divide arbitrary distances, whereas Chou's axiomatic restricts to ratios of oriented distances $\frac{\overline{AB}}{\overline{CD}}$ where the lines AB and CD are parallel. The coherence is preserved because the oriented distance can be interpreted by

the standard analytic model. The fact that we can divide arbitrary distances means that to give an interpretation to the distance function we have to give an orientation to the lines of our plane.
But the decision procedure requires explicitly that for every ratio of oriented distances $\frac{\overline{AB}}{\overline{CD}}$, AB is parallel to CD. Our lemmas used in the procedure state explicitly that the lines are parallel. This means that in our formalization one can write the ratio of two arbitrary distances but it cannot be dealt with by the decision procedure. This choice of formalization implies that the decision procedure is not complete (the goals which are not in the language of the tactic will be rejected).
This formalization is more convenient because it is more general and it allows to use an "ordinary" field, and use the standard tactic dealing with fields provided by Coq. Otherwise we would have had to give some axioms and prove some properties concerning the link between the ratio function and products, sums, *etc.* It also allows us to manipulate ratios of distances which are supported by parallel lines without *explicitly* stating that these lines are parallel. This is useful sometimes. For example with the same assumptions as in the midpoint theorem, if we want to state that $\frac{\overline{A'B'}}{\overline{AB}} = \frac{1}{2}$ we do not want to add an assumption stating that $A'B' \parallel AB$ because it is a *consequence* of other assumptions. As a result of this choice, two invariants must be kept along the proof:

1. for each denominator of a fraction there is a proof in the context that it is different from zero
2. for each ratio of oriented distances $\frac{\overline{AB}}{\overline{CD}}$ there is a proof in the context that AB is parallel to CD

3.2 Propositions needed by the tactic

Here is a quick overview of the propositions that have been proven using the Coq system for this development:

Basic propositions are used to rewrite geometric quantities, *etc.* these are the very basic propositions used by unification and simplification tactics, they come very early in order to take advantage of the unification and simplification tactics as soon as possible.
Lemmas are used in the whole development.
Construction lemmas are used to prove that each construction shown on Table 2 is a consequence of the axiom of construction.
Constructed points elimination lemmas are used to eliminate fixed points and preserve our invariants.
Free points lemmas are used to express geometric quantities using independent variables.

3.3 The tactic itself

We give in this section a detailed description of the sub-tactics we use.

Initialization tactic

1. The initialization tactic (called `geoinit`) checks that the goal is compatible with the decision procedure. (This includes verification that the invariants are initially true.)
2. It unfolds all the definitions which are not treated directly by the decision procedure. (for example midpoint is expanded as a ratio of distances, and a statement expressing the collinearity)
3. It introduces all the hypotheses in the context.
4. It decomposes the logical part of the goal if needed. (split the conjunctions and decompose the compound constructions)

Example 2. The midpoint theorem is stated using our language [8] in the syntax of Coq V8.0 as follows:

```
Theorem midpoint_A :
 forall A B C A' B' : Point, midpoint A' B C -> midpoint B' A C ->
 parallel A' B' A B.
geoInit.
1 subgoal
  A : Point
  B : Point
  C : Point
  A' : Point
  B' : Point
  H : on_line_d A' B C (1 / 2)
  H0 : on_line_d B' A C (1 / 2)
  ============================
   S A' A B' + S A' B' B = 0
```

`on_line_d A' B C (1/2)` states that A' is on line BC and $\frac{\overline{BA'}}{\overline{BC}} = \frac{1}{2}$.

Simplification tactics. The simplification tactic (`basic_simpl`) performs basic simplifications in the hypotheses *and* the goal. Note that we need to perform exactly the same simplifications in the goal and hypotheses in order to preserve our invariants. For instance if the denominator of a fraction is simplified, the same simplification must be applied to the proof that this denominator is non-zero otherwise we lose the invariant that we have a proof that every term which syntactically occurs in the denominator of a fraction is non-zero.

Basic simplifications consist in:

– removing degenerated directed distances or signed areas (*e.g.* $\frac{\overline{AA}}{\overline{AB}}$, $\mathcal{S}_{AAB}\ldots$)
– rewriting $-(-x)$ into x
– rewriting -0 into 0
– rewriting $0 * x$ and $x * 0$ into 0
– rewriting $1 * x$ and $x * 1$ into x
– rewriting $x + 0$ and $0 + x$ into x

This tactic is necessary to keep the goal as small as possible. Not simplifying the goal at each step would lead to huge terms. Examples show that the computation becomes intractable without simplification.

Unification tactics. (`unify_signed_areas`,`unify_signed_distances`)
There are two unification tactics, one for each geometric quantity. The unification tactics change the goal and hypotheses in order to unify the geometric quantities. For instance if both $\overline{AB}$ and $\overline{BA}$ are used in the context or in the goal, $\overline{AB}$ is changed into $-\overline{BA}$[6].
This has two purposes :

1. It can speed up some steps, because any rewrite of one of these quantities will be done only once.
2. It is necessary that geometric quantities which are equals have the same form. Indeed the last step of the procedure is a call to the Coq standard `field`[7] tactic on an arithmetic expression containing independent geometric quantities, and `field` would consider $\overline{AB}$ and $\overline{BA}$ as different variables (because `field` doesn't know anything about the oriented distance function).

Example 3. In this context:

```
H9 : S C A B <> 0
H8 : S B A C <> 0
H1 : S A B C <> 0
============================
  S P B C / S A B C + S P A C / S B A C + S P A B / S C A B = 1
```

the tactic `unify_signed_areas` changes the goal into:

```
H8 : - S A B C <> 0
H1 : S A B C <> 0
============================
  S P B C / S A B C + S P A C / - S A B C + S P A B / S A B C = 1
```

Elimination tactic. This tactic (called `eliminate_all`) first searches the context for a point which is not used to build another point (a leaf in the dependency graph). Then for each occurrence of the point in the goal, it applies the right lemma from Table 2 by finding in the context how the point has been constructed and which geometric quantity it appears in. Finally it removes the hypotheses stating how the point has been constructed from the context.
Note that some lemmas have a side condition to their application, in this case a recursive call on the whole tactic is done. If the condition is true then the lemma is applied, in the other case we need to do a step of classical reasoning: we reason

[6] The choice to rewrite $\overline{AB}$ or $\overline{BA}$ is arbitrarily made by the tactic.
[7] `field` is a reflexive tactic included in the distribution of Coq. It decides equality on any field defined by the user.

by cases on the side condition. The formalization in Coq emphasizes the use of this classical reasoning step. As noted before, the elimination lemmas given in Table 2 on page 5, do eliminate an occurrence of a point Y only if Y appears only one time in the geometric quantity (A,B,C and D must be different from Y). If Y appears twice in $\mathcal{S}$, this is not a problem because then the geometric quantity is zero, and so already eliminated by the simplification phase. But if Y appears twice in a ratio (for instance in $\frac{\overline{AY}}{\overline{BY}}$) this is a special case which needs to be treated apart. This is done in the implementation.

Example 4. In this context:

```
1 subgoal
A : Point
B : Point
C : Point
A' : Point
B' : Point
H : on_line_d A' B C (1 / 2)
H0 : on_line_d B' A C (1 / 2)
  ============================
S A' A B' + S B A' B' = 0
```

the tactic `eliminate B'` changes the goal into:

```
1 subgoal
A : Point
B : Point
C : Point
A' : Point
B' : Point
H : on_line_d A' B C (1 / 2)
  ============================
1 / 2 * S A' A C + (1 - 1 / 2) * S A' A A +
(1 / 2 * S B A' C + (1 - 1 / 2) * S B A' A) = 0
```

Free point elimination tactic. This tactic supposes that the goal is an expression using geometric quantities involving only free points (every constructed point has already been eliminated by the elimination tactic). The role of this tactic is to change the goal into an expression involving only independent variables. Geometric quantities involving free points are not necessarily independent, they are bound by the following relation:

$$\mathcal{S}_{ABC} = \mathcal{S}_{DBC} + \mathcal{S}_{ADC} + \mathcal{S}_{ABD}$$

But geometric quantities involving free points can be transformed into a bunch of independent variables by expressing them with respect to a base. For that purpose we choose three arbitrary non collinear points O,U and V and we use the following lemma to rewrite geometric quantities containing more than one point which is different from the base points:

$$\mathcal{S}_{OUV} \neq 0 \rightarrow \mathcal{S}_{ABC} = \begin{vmatrix} \mathcal{S}_{OUA} & \mathcal{S}_{OVA} & 1 \\ \mathcal{S}_{OUB} & \mathcal{S}_{OVB} & 1 \\ \mathcal{S}_{OUC} & \mathcal{S}_{OVC} & 1 \end{vmatrix}$$

If there are three points in the context which are known to be not collinear, we use them as the base O, U, V. Otherwise we build three non collinear points thanks to the dimension axiom.

Conclusion tactic. When this tactic is called, the goal is an expression of independent variables. If the rational equality is universally true, the theorem is proved. Otherwise there is some mapping from the variables to the field which make the equality false, and this provides a counter-example to the goal. To check if the equality is universally true, this last tactic applies the standard Coq tactic `field` to solve the goal and then solves the generated subgoals (stating that the denominators are different from zero) using the hypotheses and/or decomposing them using the fact that $x \neq 0 \wedge y \neq 0 \rightarrow x * y \neq 0$. As the `field` tactic does not provide counter-examples, our tactic is not able to give counter-examples either. This is just a technical limitation. This tactic is small enough to be fully explained:

```
Ltac field_and_conclude :=
    abstract(field; repeat (assumption || apply nonzeromult); geometry).
```

This tactic does a call to `field`[8], and tries to apply one of the assumptions to the generated subgoals. If it fails, it decomposes the product in the goal and solves the subgoals using `geometry`. This last tactic is able to solve common goals such as $\overline{AB} \neq 0$ when the fact that A is different from B is one of the hypotheses. The `abstract` tactic is here for technical reasons: this Coq tactic speeds up the typing process by creating a lemma.

3.4 A full example

Example 5. In this section we give a detailed description of how the tactic works on the first example by decomposing the procedure into small steps[9].

```
forall A B C A' B' : Point, midpoint A' B C -> midpoint B' A C ->
parallel A' B' A B.
```

[8] The tactic `field` from Coq version 8.0 is very slow at solving some goals, the reason is that the `field` tactic is based on another simplification tactic called `ring` that is very slow at computing with constants on abstract domains such as our field of measures. We incidentally had to reimplement a version of `ring` that computes using binary numbers in order to be able to compute efficiently the last phase of our decision procedure for geometry.

[9] These steps are not exactly the same steps as those executed by our automatic procedure (the automatic procedure may treat the points in another order, and perform more simplification and unification steps).

At this step it would be enough to type `autogeom` to solve the goal using our decision procedure, but for this presentation we mimic the behavior of the decision procedure using the sub-tactics described in the previous sections. We give the name of the sub-tactics on the left, and Coq output on the right[10]:

```
geoInit.              H : on_line_d A' B C (1 / 2)
                      H0 : on_line_d B' A C (1 / 2)
                      ============================
                       S A' A B' + S A' B' B = 0

eliminate B'.         H : on_line_d A' B C (1 / 2)
                      ============================
                       1 / 2 * S A' A C + (1 - 1 / 2) * S A' A A +
                       (1 / 2 * S B A' C + (1 - 1 / 2) * S B A' A) = 0

basic_simpl.          H : on_line_d A' B C (1 / 2)
                      ============================
                       1 / 2 * S A' A C + (1 / 2 * S B A' C + 1 / 2 * S B A' A) = 0

eliminate A'.         ============================
                       1 / 2 * (1 / 2 * S A C C + (1 - 1 / 2) * S A C B) +
                       (1 / 2 * (1 / 2 * S C B C + (1 - 1 / 2) * S C B B) +
                        1 / 2 * (1 / 2 * S A B C + (1 - 1 / 2) * S A B B)) = 0

basic_simpl.          ============================
                       1 / 2 * (1 / 2 * S A C B) + 1 / 2 * (1 / 2 * S A B C) = 0

unify_signed_areas.   ============================
                       1 / 2 * (1 / 2 * S A C B) + 1 / 2 * (1 / 2 * - S A C B) = 0

field_and_conclude.   Proof completed.
```

4 Future Work

This development can be extended in two directions: treat more geometrical notions and adapt this work to other axiomatic systems or formal developments. The first direction is straightforward and consists in extending the approach presented in this paper to deal with circles, perpendiculars, vectors, complex numbers and spatial geometry as shown in the book of Chou, Gao and Zhang. To achieve this goal, our tactic can easily be adapted, we only need to prove the construction and elimination lemmas corresponding to the new geometric quantities (for example the pythagoras difference) and update the unification and simplification tactics.

[10] For this presentation the fact that A, B, C, D and E are of type `point` has been removed from the context.

The second direction consists in building bridges to other formalizations of geometry (using different axiomatics). Preliminary work has been done towards the integration of our tactic with Frédérique Guilhot's Coq development dealing with high school geometry. This integration would open the door to pedagogical applications. Involving a student in the process of formally proving a basic geometry theorem is not an unreachable goal if he is saved from the burden of solving some technical goals thanks to our automatic tactic (for instance goals dealing with nondegeneracy conditions). We have initiated a discussion in order to define a common language for stating formal geometry theorems [8]. Although the logic used to formalize elementary geometry is very simple, the problem of defining a common language is not trivial. Indeed, different axiomatics can lead to different yet natural definitions for the same informal object. For instance the common notion of collinearity in a vector-based approach (A, B, C are collinear if $\exists k, \overrightarrow{AB} = k.\overrightarrow{AC}$) is different from the notion of collinearity in our development.

5 Conclusion

We have shown in this paper how automatic theorem proving can be combined with interactive proof development in the framework of the Coq proof assistant. Our implementation gives an example of how the tactic language of Coq (Ltac) and the reflection mechanism can be jointly used to build a somewhat short development of a tactic (6500 lines) without sacrificing the efficiency (our implementation within Coq is slower than the original but 20 examples including the well-known theorems of Ceva, Menelaus, Pascal and Desargues are proved in a couple of minutes). This formalization at the same time emphasizes the role of nondegeneracy conditions and provides a way to get rid of them. Our formalization also clarifies the usage of classical reasoning.

Availability. This development is available at:
`http://www.lix.polytechnique.fr/~jnarboux/ChouGaoZhang/index.html`

Acknowledgements. I want to thank Hugo Herbelin for his help during the elaboration of this work.

References

[1] S. Boutin. Using reflection to build efficient and certified decision procedures. In M. Abadi and T. Ito, editors, *Proceedings of TACS'97*, volume 1281 of *LNCS*. Springer-Verlag, 1997.

[2] S.C. Chou, X.S. Gao, and J.Z. Zhang. *Machine Proofs in Geometry.* World Scientific, Singapore, 1994.

[3] Shang-Ching Chou. *Mechanical Geometry Theorem Proving.* D. Reidel Publishing Company, 1988.

[4] G. E. Collins. Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In *Lecture Notes In Computer Science*, volume 33, pages 134–183. Springer-Verlag, 1975.

[5] Christophe Dehlinger, Jean-François Dufourd, and Pascal Schreck. Higher-order intuitionistic formalization and proofs in Hilbert's elementary geometry. In *Automated Deduction in Geometry*, pages 306–324, 2000.

[6] Euclide. *Les éléments*. Presses Universitaires de France, 1998. Traduit par Bernard Vitrac.

[7] Frédérique Guilhot. Formalisation en coq d'un cours de géométrie pour le lycée. In *Journées Francophones des Langages Applicatifs*, Janvier 2004.

[8] Frédérique Guilhot and Julien Narboux. Toward a "common" language for formally stating elementary geometry theorems. Draft.

[9] J. Harrison. Meta theory and reflection in theorem proving:a survey and critique. Technical Report CRC-053, SRI International Cambridge Computer Science Research Center, 1995.

[10] David Hilbert. *Les fondements de la géométrie.* Dunod, Paris, Jacques Gabay edition, 1971. Edition critique avec introduction et compléments préparée par Paul Rossier.

[11] D. Howe. Computation meta theory in nuprl. In E. Lusk and R. Overbeek, editors, *The Proceedings of the Ninth International Conference of Automated Deduction*, volume 310, pages 238–257. Springer-Verlag, 1988.

[12] Gilles Kahn. Constructive geometry according to Jan von Plato. Coq contribution. Coq V5.10.

[13] The Coq development team. *The Coq proof assistant reference manual.* LogiCal Project, 2004. Version 8.0.

[14] Laura I. Meikle and Jacques D. Fleuriot. Formalizing Hilbert's grundlagen in isabelle/isar. In *Theorem Proving in Higher Order Logics*, pages 319–334, 2003.

[15] A. Tarski. *A decision method for elementary algebra and geometry.* University of California Press, 1951.

[16] A. Tarski. What is elementary geometry? In P. Suppes L. Henkin and A. Tarski, editors, *The axiomatic Method, with special reference to Geometry and Physics*, pages 16–29, Amsterdam, 1959. North-Holland.

[17] Jan von Plato. The axioms of constructive geometry. In *Annals of Pure and Applied Logic*, volume 76, pages 169–200, 1995.

[18] Wen-Tsün Wu. On the decision problem and the mechanization of theorem proving in elementary geometry. In *Scientia Sinica*, volume 21, pages 157–179. 1978.