Automation of geometry using Coq

Julien Narboux
under the supervision of
Hugo Herbelin

LIX, École Polytechnique

TPHOLs 2004, Park City, Utah

Outline

Motivations

- Geometry is one of the most successful areas of automated theorem proving.
- Proof assistants need *automation*, formalizing geometry is a tedious task.

- The use of a proof assistant provides a way to combine geometrical proofs with larger proofs (involving induction for instance).
- The verification of the proofs by the Coq kernel provides a high level of confidence.

Motivations

- Geometry is one of the most successful areas of automated theorem proving.
- Proof assistants need *automation*, formalizing geometry is a tedious task.

- The use of a proof assistant provides a way to combine geometrical proofs with larger proofs (involving induction for instance).
- The verification of the proofs by the Coq kernel provides a high level of confidence.

Motivations

- Geometry is one of the most successful areas of automated theorem proving.
- Proof assistants need *automation*, formalizing geometry is a tedious task.

- The use of a proof assistant provides a way to combine geometrical proofs with larger proofs (involving induction for instance).
- The verification of the proofs by the Coq kernel provides a high level of confidence.

Motivations

- Geometry is one of the most successful areas of automated theorem proving.
- Proof assistants need *automation*, formalizing geometry is a tedious task.

- The use of a proof assistant provides a way to combine geometrical proofs with larger proofs (involving induction for instance).
- The verification of the proofs by the Coq kernel provides a high level of confidence.

The Chou-Gao-Zhang method.

- S.C. Chou, X.S. Gao, and J.Z. Zhang.
  Machine Proofs in Geometry.
  World Scientific, Singapore, 1994.

Why *this* method ?

- Coordinate free (but not number free).
- Produces *readable* proofs.

The Chou-Gao-Zhang method.

- S.C. Chou, X.S. Gao, and J.Z. Zhang.
  Machine Proofs in Geometry.
  World Scientific, Singapore, 1994.

Why *this* method ?

- Coordinate free (but not number free).
- Produces *readable* proofs.

The elimination method

The elimination method :

1. Find a point which is not used to build any other point.
   - The theorem must be stated *constructively*.
2. *Eliminate* every occurrence of this point from the goal.
   - We need some theorem to *eliminate* the point.
3. Repeat until the goal contains only *free* points.
4. Deal with the *free* points.
5. Check if the remaining goal (an equation on a field) is true.

The elimination method

The elimination method :

1. Find a point which is not used to build any other point.
   - The theorem must be stated *constructively*.
2. *Eliminate* every occurrence of this point from the goal.
   - We need some theorem to *eliminate* the point.
3. Repeat until the goal contains only *free* points.
4. Deal with the *free* points.
5. Check if the remaining goal (an equation on a field) is true.

The elimination method

The elimination method :

1. Find a point which is not used to build any other point.
   - The theorem must be stated *constructively*.
2. *Eliminate* every occurrence of this point from the goal.
   - We need some theorem to *eliminate* the point.
3. Repeat until the goal contains only *free* points.
4. Deal with the *free* points.
5. Check if the remaining goal (an equation on a field) is true.

The elimination method

The elimination method :

1. Find a point which is not used to build any other point.
   - The theorem must be stated *constructively*.
2. *Eliminate* every occurrence of this point from the goal.
   - We need some theorem to *eliminate* the point.
3. Repeat until the goal contains only *free* points.
4. Deal with the *free* points.
5. Check if the remaining goal (an equation on a field) is true.

The elimination method

The elimination method :

1. Find a point which is not used to build any other point.
   - The theorem must be stated *constructively*.
2. *Eliminate* every occurrence of this point from the goal.
   - We need some theorem to *eliminate* the point.
3. Repeat until the goal contains only *free* points.
4. Deal with the *free* points.
5. Check if the remaining goal (an equation on a field) is true.

The elimination method

The elimination method :

1. Find a point which is not used to build any other point.
   - The theorem must be stated *constructively*.
2. *Eliminate* every occurrence of this point from the goal.
   - We need some theorem to *eliminate* the point.
3. Repeat until the goal contains only *free* points.
4. Deal with the *free* points.
5. Check if the remaining goal (an equation on a field) is true.

The elimination method

The elimination method :

1. Find a point which is not used to build any other point.
   - The theorem must be stated *constructively*.
2. *Eliminate* every occurrence of this point from the goal.
   - We need some theorem to *eliminate* the point.
3. Repeat until the goal contains only *free* points.
4. Deal with the *free* points.
5. Check if the remaining goal (an equation on a field) is true.

The goal must be :

- stated constructively (as a sequence of constructions),
- using only two geometric quantities :
  1. the signed area of a triangle ($S_{ABC} = S_{BCA} = -S_{BAC}$)
  2. the ratio of two oriented distances $\frac{\overline{AB}}{\overline{CD}}$ where $AB \parallel CD$
- combined using arithmetic expressions (+,-,*,/).

Using these two quantities :

Geometric notions	Formalization
A,B and C are collinear	$S_{ABC} = 0$
$AB \parallel CD$	$S_{ABC} = S_{ABD}$
I is the midpoint of AB	$\frac{\overline{AB}}{\overline{AI}} = 2 \wedge S_{ABI} = 0$

The goal must be :

- stated constructively (as a sequence of constructions),
- using only two geometric quantities :
  1. the signed area of a triangle ($\mathcal{S}_{ABC} = \mathcal{S}_{BCA} = -\mathcal{S}_{BAC}$)
  2. the ratio of two oriented distances $\frac{\overline{AB}}{\overline{CD}}$ where $AB \parallel CD$
- combined using arithmetic expressions (+,-,*,/).

Using these two quantities :

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	$\frac{\overline{AB}}{\overline{AI}} = 2 \wedge \mathcal{S}_{ABI} = 0$

The goal must be :

- stated constructively (as a sequence of constructions),
- using only two geometric quantities :
  1. the signed area of a triangle ($\mathcal{S}_{ABC} = \mathcal{S}_{BCA} = -\mathcal{S}_{BAC}$)
  2. the ratio of two oriented distances $\frac{\overline{AB}}{\overline{CD}}$ where $AB \parallel CD$
- combined using arithmetic expressions (+,-,*,/).

Using these two quantities :

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	$\frac{\overline{AB}}{\overline{AI}} = 2 \wedge \mathcal{S}_{ABI} = 0$

The goal must be :

- stated constructively (as a sequence of constructions),
- using only two geometric quantities :
  1. the signed area of a triangle ($\mathcal{S}_{ABC} = \mathcal{S}_{BCA} = -\mathcal{S}_{BAC}$)
  2. the ratio of two oriented distances $\frac{\overline{AB}}{\overline{CD}}$ where $AB \parallel CD$
- combined using arithmetic expressions (+,-,*,/).

Using these two quantities :

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	$\frac{\overline{AB}}{\overline{AI}} = 2 \wedge \mathcal{S}_{ABI} = 0$

The goal must be :

- stated constructively (as a sequence of constructions),
- using only two geometric quantities :
  1. the signed area of a triangle ($\mathcal{S}_{ABC} = \mathcal{S}_{BCA} = -\mathcal{S}_{BAC}$)
  2. the ratio of two oriented distances $\frac{\overline{AB}}{\overline{CD}}$ where $AB \parallel CD$
- combined using arithmetic expressions (+,-,*,/).

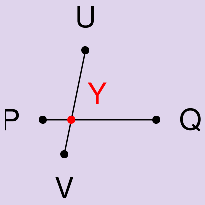
Using these two quantities :

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	$\frac{\overline{AB}}{\overline{AI}} = 2 \land \mathcal{S}_{ABI} = 0$

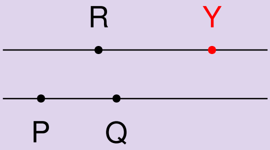
The basic constructions

Point on line.

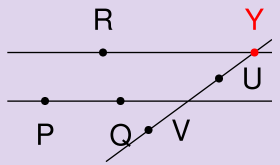
Point on intersection.

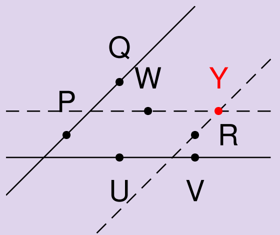


Point on parallel line.



The complex constructions





Elimination lemmas

We have to get rid of :

- ratios of oriented distances,
- signed areas

One example :

If Y is the intersection of (PQ) and (UV) then :

$$\text{For every } A \text{ and } B,\ S_{ABY} = \frac{S_{PUV} * S_{ABQ} + S_{QVU} * S_{ABP}}{(S_{PUQV})}$$

Elimination lemmas

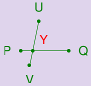
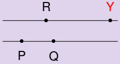
We have to get rid of :

- ratios of oriented distances,
- signed areas

One example :

If Y is the intersection of (PQ) and (UV) then :

$$\text{For every } A \text{ and } B,\ \mathcal{S}_{ABY} = \frac{\mathcal{S}_{PUV} * \mathcal{S}_{ABQ} + \mathcal{S}_{QVU} * \mathcal{S}_{ABP}}{(\mathcal{S}_{PUQV})}$$

Construction	Elimination formulas	
	$\mathcal{S}_{ABY} =$	If $AY \parallel CD$ then $\frac{\overline{AY}}{\overline{CD}} =$
	$\lambda \mathcal{S}_{ABQ} + (1-\lambda)\mathcal{S}_{ABP}$	$\begin{cases} \frac{\frac{\overline{AP}}{\overline{PQ}}+\lambda}{\frac{\overline{CD}}{\overline{PQ}}} & \text{if } A \in PQ \\ \frac{\mathcal{S}_{APQ}}{\mathcal{S}_{CPDQ}} & \text{otherwise}^a. \end{cases}$
	$\frac{\mathcal{S}_{PUV}\mathcal{S}_{ABQ}+\mathcal{S}_{QVU}\mathcal{S}_{ABP}}{\mathcal{S}_{PUQV}}$	$\begin{cases} \frac{\mathcal{S}_{AUV}}{\mathcal{S}_{CUDV}} & \text{if } A \notin UV \\ \frac{\mathcal{S}_{APQ}}{\mathcal{S}_{CPDQ}} & \text{otherwise.} \end{cases}$
	$\mathcal{S}_{ABR} + \lambda \mathcal{S}_{APBQ}$	$\begin{cases} \frac{\frac{\overline{AR}}{\overline{PQ}}+\lambda}{\frac{\overline{CD}}{\overline{PQ}}} & \text{if } A \in RY \\ \frac{\mathcal{S}_{APRQ}}{\mathcal{S}_{CPDQ}} & \text{otherwise.} \end{cases}$

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

Eliminating free points

Choose three non collinear points O,U and V

$$S_{ABY} = \begin{vmatrix} S_{OUA} & S_{OVA} & S_{UVA} \\ S_{OUB} & S_{OVB} & S_{UVB} \\ S_{OUY} & S_{OVY} & S_{UVY} \end{vmatrix}$$

The implementation is done :

- using LTac (the tactic language of Coq),
- the reflection mechanism (some sub-tactics are written using Coq itself).

We have to :

1. describe the axiomatic,
2. prove the elimination lemmas,
3. automate the elimination process thanks to some tactics.

The implementation is done :

- using LTac (the tactic language of Coq),
- the reflection mechanism (some sub-tactics are written using Coq itself).

We have to :

1. describe the axiomatic,
2. prove the elimination lemmas,
3. automate the elimination process thanks to some tactics.

The implementation is done :

- using LTac (the tactic language of Coq),
- the reflection mechanism (some sub-tactics are written using Coq itself).

We have to :

1. describe the axiomatic,
2. prove the elimination lemmas,
3. automate the elimination process thanks to some tactics.

The implementation is done :

- using LTac (the tactic language of Coq),
- the reflection mechanism (some sub-tactics are written using Coq itself).

We have to :

1. describe the axiomatic,
2. prove the elimination lemmas,
3. automate the elimination process thanks to some tactics.

The implementation is done :

- using LTac (the tactic language of Coq),
- the reflection mechanism (some sub-tactics are written using Coq itself).

We have to :

1. describe the axiomatic,
2. prove the elimination lemmas,
3. automate the elimination process thanks to some tactics.

The axiomatic

Some field

whose characteristic is different from two.

An oriented distance

- $\overline{AB} = -\overline{BA}$
- $\overline{AB} = 0 \iff A = B$

The axiomatic

Some field

whose characteristic is different from two.

An oriented distance

- $\overline{AB} = -\overline{BA}$
- $\overline{AB} = 0 \iff A = B$

A signed area

- $\mathcal{S}_{ABC} = \mathcal{S}_{CAB}$
- $\mathcal{S}_{ABC} = -\mathcal{S}_{BAC}$

Chasles' relation

$(\mathcal{S}_{ABC} = 0) \rightarrow \overline{AB} + \overline{BC} = \overline{AC}$

A signed area

- $\mathcal{S}_{ABC} = \mathcal{S}_{CAB}$
- $\mathcal{S}_{ABC} = -\mathcal{S}_{BAC}$

Chasles' relation

$(\mathcal{S}_{ABC} = 0) \rightarrow \overline{AB} + \overline{BC} = \overline{AC}$

Dimension axioms

lower bound $\exists A, B, C | \mathcal{S}_{ABC} \neq 0$

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

Construction axioms

existence $(\forall A, B : Point, r : F), \exists P : Point \mid (\mathcal{S}_{ABP} = 0) \wedge \overline{AP} = r\overline{AB}$

unicity $\forall A, B, P, P' : Point, r : F\ A \neq B \rightarrow$
$(\mathcal{S}_{ABP} = 0) \rightarrow \overline{AP} = r\overline{AB} \rightarrow$
$(\mathcal{S}_{ABP'} = 0) \rightarrow \overline{AP'} = r\overline{AB} \rightarrow P = P'$

Dimension axioms

lower bound $\exists A, B, C | \mathcal{S}_{ABC} \neq 0$

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

Construction axioms

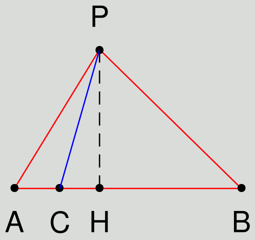
existence $(\forall A, B : Point,\ r : F), \exists P : Point\ | \ (\mathcal{S}_{ABP} = 0) \wedge \overline{AP} = r\overline{AB}$

unicity $\forall A, B, P, P' : Point,\ r : F\ A \neq B \rightarrow$

$$\begin{array}{l} (\mathcal{S}_{ABP} = 0) \rightarrow \overline{AP} = r\overline{AB} \rightarrow \\ (\mathcal{S}_{ABP'} = 0) \rightarrow \overline{AP'} = r\overline{AB} \rightarrow \end{array} P = P'$$

Proportions axiom

$$A \neq C \rightarrow \neg(\mathcal{S}_{PAC} = 0) \rightarrow (\mathcal{S}_{ABC} = 0) \rightarrow \frac{\overline{AB}}{\overline{AC}} = \frac{\mathcal{S}_{PAB}}{\mathcal{S}_{PAC}}$$



We need to prove :

- some simplification lemmas,
- the construction lemmas,
- the elimination lemmas,
- ...

Approximately 6000 lines of Coq.

Some tactics:

initialization translates the goal into the language.

simplification performs trivial simplifications.

unification rewrites all occurrences of a geometric quantity into the same expression.

elimination eliminates a point from a goal.

free point elimination treat the goal in order to keep only *independent* variables.

conclusion mainly apply the standard Coq tactic `Field` (dealing with equalities on fields).

Some tactics:

initialization	translates the goal into the language.
simplification	performs trivial simplifications.
unification	rewrites all occurrences of a geometric quantity into the same expression.
elimination	eliminates a point from a goal.
free point elimination	treat the goal in order to keep only *independent* variables.
conclusion	mainly apply the standard Coq tactic `Field` (dealing with equalities on fields).

Some tactics:

initialization	translates the goal into the language.
simplification	performs trivial simplifications.
unification	rewrites all occurrences of a geometric quantity into the same expression.
elimination	eliminates a point from a goal.
free point elimination	treat the goal in order to keep only *independent* variables.
conclusion	mainly apply the standard Coq tactic `Field` (dealing with equalities on fields).

Some tactics:

initialization	translates the goal into the language.
simplification	performs trivial simplifications.
unification	rewrites all occurrences of a geometric quantity into the same expression.
elimination	eliminates a point from a goal.
free point elimination	treat the goal in order to keep only *independent* variables.
conclusion	mainly apply the standard Coq tactic `Field` (dealing with equalities on fields).

Some tactics:

initialization	translates the goal into the language.
simplification	performs trivial simplifications.
unification	rewrites all occurrences of a geometric quantity into the same expression.
elimination	eliminates a point from a goal.
free point elimination	treat the goal in order to keep only *independent* variables.
conclusion	mainly apply the standard Coq tactic `Field` (dealing with equalities on fields).

Some tactics:

initialization	translates the goal into the language.
simplification	performs trivial simplifications.
unification	rewrites all occurrences of a geometric quantity into the same expression.
elimination	eliminates a point from a goal.
free point elimination	treat the goal in order to keep only *independent* variables.
conclusion	mainly apply the standard Coq tactic `Field` (dealing with equalities on fields).

Invariants

We maintain proofs that :

- denominators are different from zero,
- $AB \parallel CD$ for every $\frac{\overline{AB}}{\overline{CD}}$.

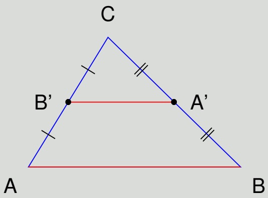
Invariants

We maintain proofs that :

- denominators are different from zero,
- $AB \parallel CD$ for every $\frac{\overline{AB}}{\overline{CD}}$.

Midpoint theorem.

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



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.
```

Some examples :

- Ceva
- Menelaus
- Pascal
- Desargues
- Centroid
- Midpoint
- Gauss-Line
- . . .

Some problems :

- For example :
  - Nerhing
  - Pappus
  - . . .
- The `Field` tactic is not very efficient.
- We need to perform more simplifications.
- No counter example is provided.

23 examples are proved within 160 seconds.

This formalization :

- shows that non degeneracy conditions are crucial,
- emphasizes classical reasoning steps,
- provides trustable proofs.

This formalization :

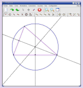
- shows that non degeneracy conditions are crucial,
- emphasizes classical reasoning steps,
- provides trustable proofs.

This formalization :

- shows that non degeneracy conditions are crucial,
- emphasizes classical reasoning steps,
- provides trustable proofs.

Work in progress

- Implementation of a graphical user interface (GeoCaml[a]).



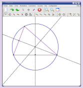
- Integration of this development with Frédérique Guilhot's work on high school geometry.

[a] `https://gna.org/projects/geocaml`

Perspectives

- Extension of the method to deal with Pythagoras differences...
- Implementation using other theorem provers ?
- Pedagogical applications.

Work in progress



- Implementation of a graphical user interface (GeoCaml[a]).
- Integration of this development with Frédérique Guilhot's work on high school geometry.

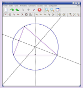
[a] `https://gna.org/projects/geocaml`

Perspectives

- Extension of the method to deal with Pythagoras differences…
- Implementation using other theorem provers ?
- Pedagogical applications.

Work in progress

- Implementation of a graphical user interface (GeoCaml[a]).



- Integration of this development with Frédérique Guilhot's work on high school geometry.

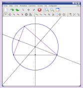
[a] https://gna.org/projects/geocaml

Perspectives

- Extension of the method to deal with Pythagoras differences…
- Implementation using other theorem provers ?
- Pedagogical applications.

Work in progress

- Implementation of a graphical user interface (GeoCaml[a]).



- Integration of this development with Frédérique Guilhot's work on high school geometry.

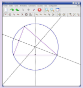
[a] https://gna.org/projects/geocaml

Perspectives

- Extension of the method to deal with Pythagoras differences...
- Implementation using other theorem provers ?
- Pedagogical applications.

Work in progress

- Implementation of a graphical user interface (GeoCaml[a]).



- Integration of this development with Frédérique Guilhot's work on high school geometry.

[a] `https://gna.org/projects/geocaml`

Perspectives

- Extension of the method to deal with Pythagoras differences…
- Implementation using other theorem provers ?
- Pedagogical applications.

Bibliography

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

- Gilles Kahn.
  Constructive geometry according to Jan von Plato.
  Coq contribution.
  Coq V5.10.
- 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.