Formalization and automation of geometric reasoning using Coq.

Julien Narboux
under the supervision of
Hugo Herbelin

LIX, INRIA Futurs, École Polytechnique
↓
Nominal Methods Group, TU München

October 13, 2006, Munich, Germany

Outline

1. Formalization

Outline

1. Formalization
2. Automation

Outline

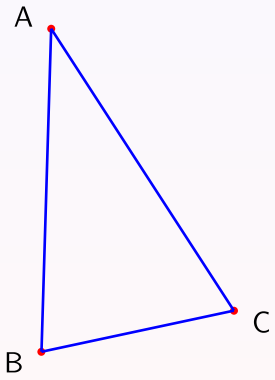
1. Formalization
2. Automation
3. GeoProof: A graphical user interface for proofs in geometry

Outline

1. Formalization
2. Automation
3. GeoProof: A graphical user interface for proofs in geometry
4. Diagrammatic proofs in abstract rewriting

Every triangle is isosceles.

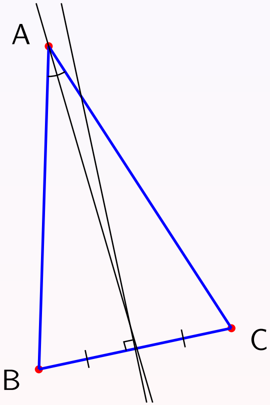
- Let ABC be a triangle.



▸ Solution

Every triangle is isosceles.

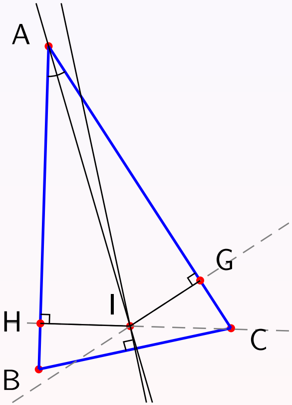
- Let ABC be a triangle.
- Let D be the perpendicular bisector of $[BC]$ and let D' be the bisector of $\angle BAC$.



▸ Solution

Every triangle is isosceles.

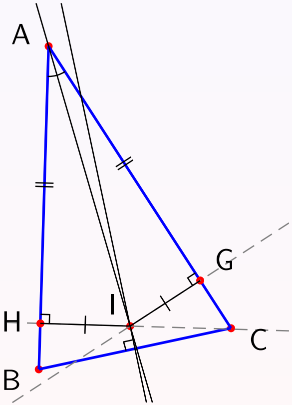
- Let ABC be a triangle.
- Let D be the perpendicular bisector of $[BC]$ and let D' be the bisector of $\angle BAC$.
- Let I be the intersection of D and D'.



Solution

Every triangle is isosceles.

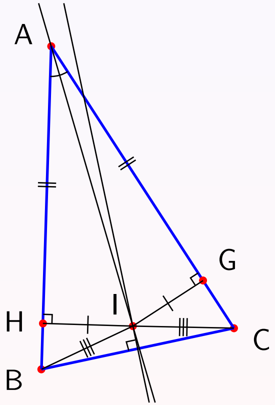
- Let ABC be a triangle.
- Let D be the perpendicular bisector of $[BC]$ and let D' be the bisector of $\angle BAC$.
- Let I be the intersection of D and D'.
- $HI = IG \wedge AH = AG$



▸ Solution

Every triangle is isosceles.

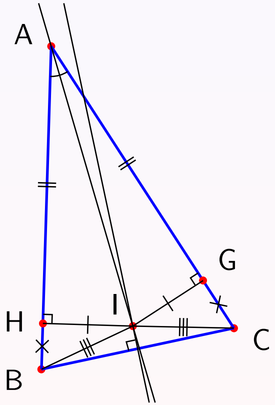
- Let ABC be a triangle.
- Let D be the perpendicular bisector of $[BC]$ and let D' be the bisector of $\angle BAC$.
- Let I be the intersection of D and D'.
- $HI = IG \wedge AH = AG$
- $IB = IC$



Solution

Every triangle is isosceles.

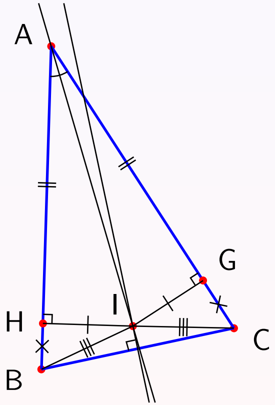
- Let ABC be a triangle.
- Let D be the perpendicular bisector of $[BC]$ and let D' be the bisector of $\angle BAC$.
- Let I be the intersection of D and D'.
- $HI = IG \wedge AH = AG$
- $IB = IC$
- $HB = GC$



Solution

Every triangle is isosceles.

- Let ABC be a triangle.
- Let D be the perpendicular bisector of $[BC]$ and let D' be the bisector of $\angle BAC$.
- Let I be the intersection of D and D'.
- $HI = IG \wedge AH = AG$
- $IB = IC$
- $HB = GC$
- $AB = AC$



Solution

Formalization of geometry.

Related work

- Gilles Kahn (Coq) [Kah95]

Formalization of geometry.

Related work

- Gilles Kahn (Coq) [Kah95]
- Christophe Dehlinger, Jean-François Dufourd and Pascal Schreck (Coq) [DDS00]

Formalization of geometry.

Related work

- Gilles Kahn (Coq) [Kah95]
- Christophe Dehlinger, Jean-François Dufourd and Pascal Schreck (Coq) [DDS00]
- Laura Meikle and Jacques Fleuriot (Isabelle) [MF03]

Formalization of geometry.

Related work

- Gilles Kahn (Coq) [Kah95]
- Christophe Dehlinger, Jean-François Dufourd and Pascal Schreck (Coq) [DDS00]
- Laura Meikle and Jacques Fleuriot (Isabelle) [MF03]
- Frédérique Guilhot (Coq) [Gui05]

Formalization of geometry.

Related work

- Gilles Kahn (Coq) [Kah95]
- Christophe Dehlinger, Jean-François Dufourd and Pascal Schreck (Coq) [DDS00]
- Laura Meikle and Jacques Fleuriot (Isabelle) [MF03]
- Frédérique Guilhot (Coq) [Gui05]
- Julien Narboux (Coq) [Nar04, Nar06c]

Formalization of geometry.

Related work

- Gilles Kahn (Coq) [Kah95]
- Christophe Dehlinger, Jean-François Dufourd and Pascal Schreck (Coq) [DDS00]
- Laura Meikle and Jacques Fleuriot (Isabelle) [MF03]
- Frédérique Guilhot (Coq) [Gui05]
- Julien Narboux (Coq) [Nar04, Nar06c]

Formalization of geometry.

Related work

- Gilles Kahn (Coq) [Kah95]
- Christophe Dehlinger, Jean-François Dufourd and Pascal Schreck (Coq) [DDS00]
- Laura Meikle and Jacques Fleuriot (Isabelle) [MF03]
- Frédérique Guilhot (Coq) [Gui05]
- Julien Narboux (Coq) [Nar04, Nar06c]

Motivations

- We need foundations to combine the different formal developments.

Why Tarski's axioms ?

- They are simple.
  - 11 axioms
  - two predicates ($\beta\ A\,B\,C$, $AB \equiv CD$)

Why Tarski's axioms ?

- They are simple.
  - 11 axioms
  - two predicates ($\beta\ A\ B\ C$, $AB \equiv CD$)
- They have good meta-mathematical properties.
  - coherent
  - complete
  - decidable
  - categorical
  - its axioms are independent (almost)

Why Tarski's axioms ?

- They are simple.
  - 11 axioms
  - two predicates ($\beta\ A\,B\,C$, $AB \equiv CD$)
- They have good meta-mathematical properties.
  - coherent
  - complete
  - decidable
  - categorical
  - its axioms are independent (almost)
- They can be generalized to different dimensions and geometries.

History

1940 [Tar67]	1951 [Tar51]	1959 [Tar59]	1965 [Gup65]	1983 [SST83]
1	1	1	1	1
2	2	2	2	2
3	3	3	3	3
4	4	4	4	4
5_1	5_1	$\rightarrow 5$	5	5
6	6	6		6
7_2	7_2	$\rightarrow 7_1$	7_1	$\rightarrow 7$
8(2)	8(2)	8(2)	8(2)	8(2)
$9_1(2)$	$9_1(2)$	$\rightarrow 9(2)$	9(2)	9(2)
10	10	$\rightarrow 10_1$	10_1	$\rightarrow 10$
11	11	11	11	11
12	12			
13				
14	14			
15	15	15	15	
16	16			
17	17			
18	18	18		
19				
20	$\rightarrow 20_1$			
21	21			
20 + 1 schema	18 + 1 schema	12 + 1 schema	10 + 1 schema	10 + 1 schema

Formalization

W. Schwabhäuser
W. Szmielew
A. Tarski

Metamathematische Methoden in der Geometrie

Springer-Verlag 1983

Overview

About 200 lemmas and 6000 lines of proofs and definitions.

Chapter 1 Axioms

...

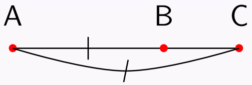
Chapter 5 Transitivity properties for `Col`.

...

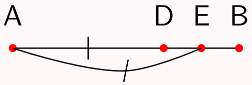
Chapter 8 Existence of the midpoint.

Two crucial lemmas

$$\forall ABC, \beta\ A\ C\ B \wedge AC \equiv AB \Rightarrow C = B$$



$$\forall ABDE, \beta\ A\ D\ B \wedge \beta\ A\ E\ B \wedge AD \equiv AE \Rightarrow D = E.$$



($\beta\ A\ B\ C$ means $B \in [AC]$)

About degenerated cases

Calculus with binders	α-conversion
Geometry	Degenerated cases

About degenerated cases

Calculus with binders	α-conversion
Geometry	Degenerated cases

- We need specialized tactics.

About degenerated cases

Calculus with binders	α-conversion
Geometry	Degenerated cases

- We need specialized tactics.
- It is simple but effective !

About degenerated cases

Calculus with binders	α-conversion
Geometry	Degenerated cases

- We need specialized tactics.
- It is simple but effective !
- Still, the axiom system is important.

Comparison with other formalizations

- ☺ There are fewer degenerated cases than in Hilbert's axiom system.

Comparison with other formalizations

- ☺ There are fewer degenerated cases than in Hilbert's axiom system.
- ☺ The axiom system is simpler.

Comparison with other formalizations

- ☺ There are fewer degenerated cases than in Hilbert's axiom system.
- ☺ The axiom system is simpler.
- ☺ It has good meta-mathematical properties.

Comparison with other formalizations

- ☺ There are fewer degenerated cases than in Hilbert's axiom system.
- ☺ The axiom system is simpler.
- ☺ It has good meta-mathematical properties.
- ☺ Generalization to other dimensions is easy.

Comparison with other formalizations

- ☺ There are fewer degenerated cases than in Hilbert's axiom system.
- ☺ The axiom system is simpler.
- ☺ It has good meta-mathematical properties.
- ☺ Generalization to other dimensions is easy.
- ☹ Lemma scheduling is more complicated.

Comparison with other formalizations

- ☺ There are fewer degenerated cases than in Hilbert's axiom system.
- ☺ The axiom system is simpler.
- ☺ It has good meta-mathematical properties.
- ☺ Generalization to other dimensions is easy.
- ☹ Lemma scheduling is more complicated.
- ☹ It is not well adapted to teaching.

Comparison with ATP

- We can not use a decision procedure specialized in geometry.
- Problems which can be solved by at least one general purpose ATP and appear in my formalization have short proofs.

Examples

Lemma	Coq proof	Otter	Vampire
symmetry of betweeness	6 lines	0s	0s
reflexivity of equidistance	2 lines	0s	0s
transitivity of equidistance	2 lines	0s	0s
existence of the midpoint	6000 lines	timeout	timeout

Automated deduction in geometry

- Algebraic methods (Wu, Gröbner bases, . . .)
- Coordinate free methods (the full-angle method, the area method,. . .)

The area method

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

The elimination method

The elimination method :

1. Find a point which is not used to build any other point.

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.

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.

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.

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.

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.

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),

The goal must be :

- stated constructively (as a sequence of constructions),
- using only two geometric quantities :
  1. the signed area of a triangle ($\mathcal{S}_{ABC} = \mathcal{S}_{BCA} = -\mathcal{S}_{BAC}$)

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 $AB \parallel CD$

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 (+,-,*,/).

The goal must be :

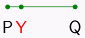
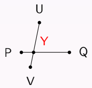
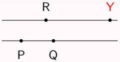
- stated constructively (as a sequence of constructions),
- using only two geometric quantities :
  1. the signed area of a triangle ($\mathcal{S}_{ABC} = \mathcal{S}_{BCA} = -\mathcal{S}_{BAC}$)
  2. the ratio of two oriented distances $\frac{\overline{AB}}{\overline{CD}}$ where $AB \parallel CD$
- 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	$\frac{\overline{AB}}{\overline{AI}} = 2 \wedge \mathcal{S}_{ABI} = 0$

We can deal with affine geometry.

The method can be extended to deal with euclidean geometry.

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

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

It cannot prove automatically:

1. Theorems outside affine geometry.
2. - Theorems involving a quantification over constructions.
     - The pentagon can be constructed with ruler and compass.
     - The heptagon can not be constructed with ruler and compass.
     - ...

...

It cannot prove automatically:

1. Theorems outside affine geometry.
2. - Theorems involving a quantification over constructions.
     - The pentagon can be constructed with ruler and compass.
     - The heptagon can not be constructed with ruler and compass.
     - ...

   Theorems stated non constructively.

     - Let C be a point such that $AC = BC$ ...

   ...

The implementation is done :

- using L_{tac} (the tactic language of Coq),

The implementation is done :

- using L_{tac} (the tactic language of Coq),
- the reflection mechanism (some sub-tactics are written using Coq itself).

The implementation is done :

- using L_{tac} (the tactic language of Coq),
- the reflection mechanism (some sub-tactics are written using Coq itself).

We have to :

1. describe the axiomatic,

The implementation is done :

- using L_{tac} (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,

The implementation is done :

- using L_{tac} (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.

Some tactics:

initialization translates the goal into the language.

Some tactics:

initialization	translates the goal into the language.
simplification	performs trivial simplifications.

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.

Some tactics:

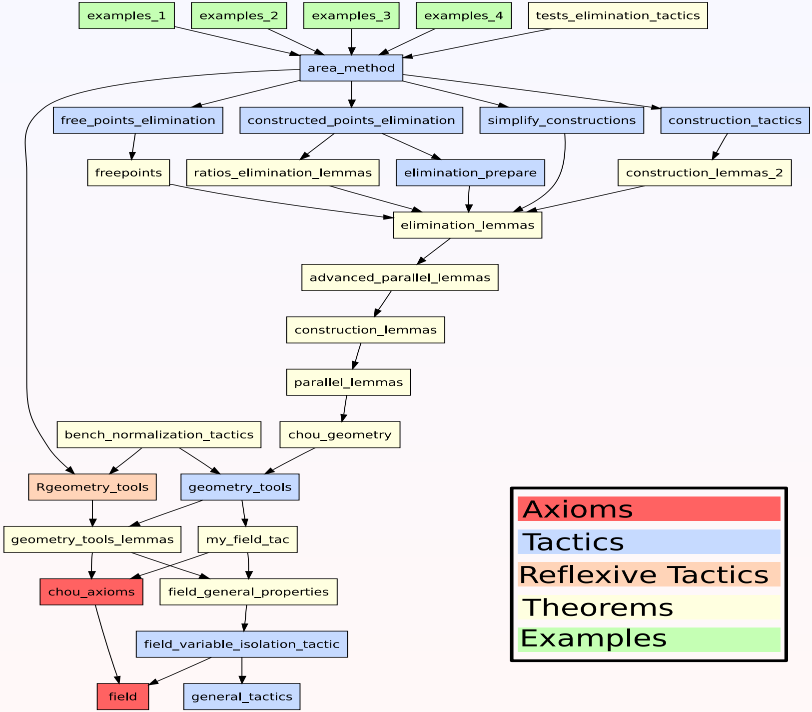
initialization	translates the goal into the language.
simplification	performs trivial simplifications.
unification	rewrites all occurrences of a geometric quantity into the same expression.
elimination	eliminates a point from a goal.

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.

Some tactics:

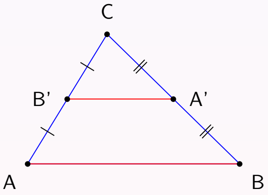
initialization	translates the goal into the language.
simplification	performs trivial simplifications.
unification	rewrites all occurrences of a geometric quantity into the same expression.
elimination	eliminates a point from a goal.
free point elimination	treat the goal in order to keep only independent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


examples_1
examples_2
examples_3
examples_4
tests_elimination_tactics
area_method
free_points_elimination
constructed_points_elimination
simplify_constructions
construction_tactics
freepoints
ratios_elimination_lemmas
elimination_prepare
construction_lemmas_2
elimination_lemmas
advanced_parallel_lemmas
construction_lemmas
parallel_lemmas
bench_normalization_tactics
chou_geometry
Rgeometry_tools
geometry_tools
geometry_tools_lemmas
my_field_tac
chou_axioms
field_general_properties
field_variable_isolation_tactic
field
general_tactics
Axioms
Tactics
Reflexive Tactics
Theorems
Examples

An example

The midpoint theorem

if A' is the midpoint of $[BC]$ and B' is the midpoint of $[AC]$ then $(A'B') \parallel (AB)$.



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.
```

What we learned

- We fixed some details about degenerated conditions.
- We clarified the use of classical logic

Example

Let Y on the line PQ such that $\frac{\overline{PY}}{\overline{PQ}} = \lambda$ $(P \neq Q)$.

$$\frac{\overline{AY}}{\overline{CD}} = \begin{cases} \frac{\frac{\overline{AP}}{\overline{PQ}}+\lambda}{\frac{\overline{CD}}{\overline{PQ}}} & \text{if } A \in PQ \\ \frac{\mathcal{S}_{APQ}}{\mathcal{S}_{CPDQ}} & \text{otherwise.} \end{cases}$$

If $A = Y$ it can happen that $CD \not\parallel PQ$.

We need to perform a case distinction using classical logic.

Examples

Some well-known theorems

Ceva
Menelaus
Pascal
Pappus
Desargues
Centroïd
Gauss-Line

$>$ 40 examples

average time : 9 seconds

GeoProof combines these features:

- dynamic geometry
- automatic theorem proving
- interactive theorem proving (using Coq/CoqIDE)

Motivations

- The use of a proof assistant provides a way to combine geometrical proofs with larger proofs (involving induction for instance).

Motivations

- The use of a proof assistant provides a way to combine geometrical proofs with larger proofs (involving induction for instance).
- There are facts than can not be visualized graphically and there are facts that are difficult to understand without being visualized.

Motivations

- The use of a proof assistant provides a way to combine geometrical proofs with larger proofs (involving induction for instance).
- There are facts than can not be visualized graphically and there are facts that are difficult to understand without being visualized.
- We should have both the ability to make arbitrarily complex proofs and use a base of known lemmas.

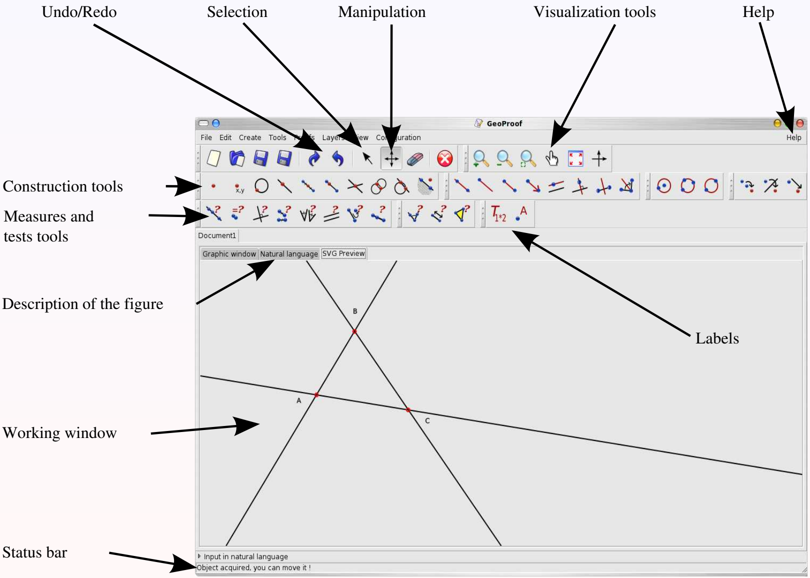
Motivations

- The use of a proof assistant provides a way to combine geometrical proofs with larger proofs (involving induction for instance).
- There are facts than can not be visualized graphically and there are facts that are difficult to understand without being visualized.
- We should have both the ability to make arbitrarily complex proofs and use a base of known lemmas.
- The verification of the proofs by the proof assistant provides a very high level of confidence.

Overview of GeoProof

- Ocaml and LablGTK2 ($\approx$ 20000 lines of code)
- License: GPL2
- Multi-platform

Overview of GeoProof



Dynamic geometry features

- points, lines, circles, vectors, segments, intersections, perpendicular lines, perpendicular bisectors,angle bisectors. . .
- central symmetry, translation and axial symmetry
- traces
- text labels with dynamic parts:
  - measures of angles, distances and areas
  - properties tests (collinearity,orthogonality,. . .)

- **layers**
- Computations use **arbitrary precision**
- Input: XML
- Output: XML, natural language, SVG, PNG, BMP, Eukleides (**latex**), **Coq**

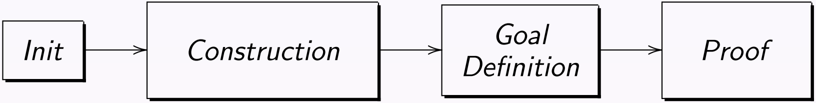
Missing features:

- loci and conics
- macros
- animations

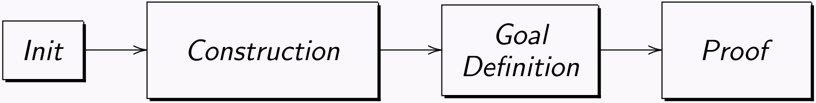
Proof related features

1. Automatic proof using an embedded ATP
2. Automatic proof using Coq
3. Interactive proof using Coq

Interactive proof using Coq

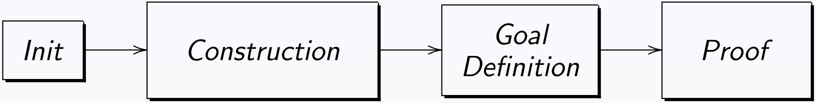


Interactive proof using Coq



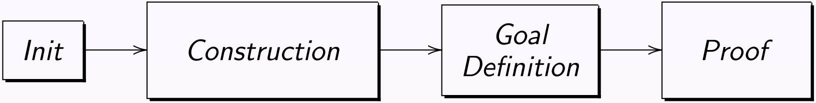
- GeoProof loads the library (Guilhot or Narboux) and updates the interface.

Interactive proof using Coq



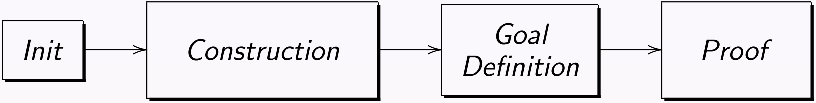
- GeoProof loads the library (Guilhot or Narboux) and updates the interface.
- The user performs the construction.

Interactive proof using Coq



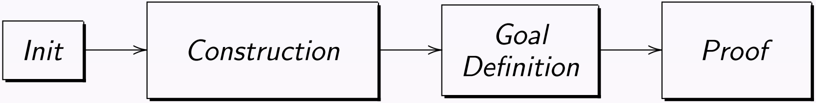
- GeoProof loads the library (Guilhot or Narboux) and updates the interface.
- The user performs the construction.
- It translates each construction as an hypothesis in Coq syntax.

Interactive proof using Coq



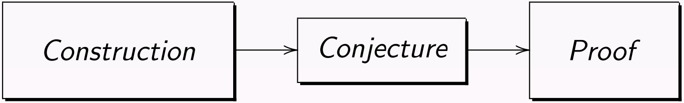
- GeoProof loads the library (Guilhot or Narboux) and updates the interface.
- The user performs the construction.
- It translates each construction as an hypothesis in Coq syntax.
- It translates the conjecture into Coq syntax.

Interactive proof using Coq



- GeoProof loads the library (Guilhot or Narboux) and updates the interface.
- The user performs the construction.
- It translates each construction as an hypothesis in Coq syntax.
- It translates the conjecture into Coq syntax.
- It translates each construction into the application of a tactic to prove the existence of the newly introduced object.

Typical use



Demo !

- We want to extend GeoProof to perform proofs in different domains.

- We want to extend GeoProof to perform proofs in different domains.
- First, we concentrate on abstract rewriting.

Example

Definition

The composition of two relations $\xrightarrow{a}$ and $\xrightarrow{b}$ is defined by:

$$\forall xy, x \xrightarrow{a.b} y \iff \exists z, x \xrightarrow{a} z \xrightarrow{b} y$$

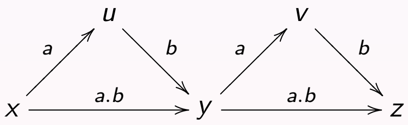
Example

If $\xrightarrow{a}$ and $\xrightarrow{b}$ are transitive and $\xrightarrow{b.a} \subseteq \xrightarrow{a.b}$ then $\xrightarrow{a.b}$ is transitive.

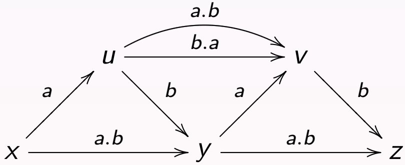
Running example

$$x \xrightarrow{a.b} y \xrightarrow{a.b} z$$

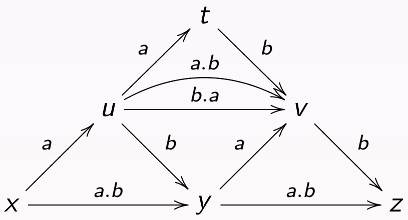
Running example



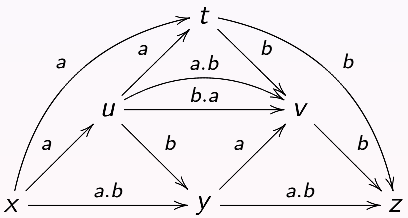
Running example



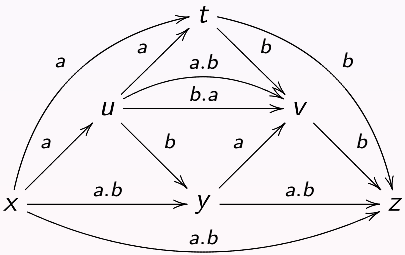
Running example



Running example



Running example



Diagrams as proofs

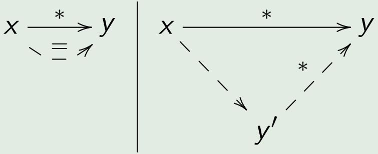
Diagrams can be seen as proof hints.

Diagrams as proofs

Diagrams can be seen as proof ~~hints~~ objects.

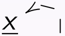
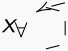
Diagrams can be defined by labeled oriented graphs verifying some properties.

Definition of $\xrightarrow{*}$



$$\forall xy, x \xrightarrow{*} y \Rightarrow (x \xrightarrow{=} y \vee \exists y', x \longrightarrow y' \xrightarrow{*} y)$$

More examples

Formula	Diagram
$x \longrightarrow x$	
$\forall x,\ x \longrightarrow x$	
$\exists x,\ x \longrightarrow x$	
$\exists xy,\ x \longrightarrow y$	$x \dashrightarrow y$
$\forall x \exists y,\ x \longrightarrow y$	$x_\forall \dashrightarrow y$
$\forall xy,\ x \longrightarrow y$	$x_\forall \dashrightarrow y_\forall$
$x \longrightarrow y$	$\underline{x} \dashrightarrow \underline{y}$

Diagrammatic formulas

Formulas which can be represented by a diagram are those of the form:

$$\forall\, \overline{u} \bigwedge_i H_i \Rightarrow \bigvee_i \exists\, \overline{e_i} \bigwedge_j C_{i_j}$$

where H_i and C_{i_j} are predicates of arity two.

Diagrammatic formulas

Formulas which can be represented by a diagram are those of the form:

$$\forall\, \overline{u} \bigwedge_i H_i \Rightarrow \bigvee_i \exists\, \overline{e_i} \bigwedge_j C_{i_j}$$

where H_i and C_{i_j} are predicates of arity two.

This class of formulas is exactly what is called coherent logic by Marc Bezem and Thierry Coquand.

Inference rules

The system contains five inference rules:

`intros` to introduce hypotheses in the context,

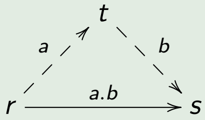
Inference rules

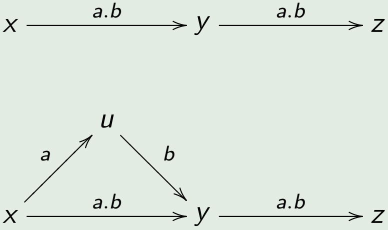
The system contains five inference rules:

`intros` to introduce hypotheses in the context,

`apply` to use the information contained in a universal diagram to enrich the factual diagram,

Example





Inference rules

The system contains five inference rules:

- `intros` to introduce hypotheses in the context,
- `apply` to use the information contained in a universal diagram to enrich the factual diagram,
- `conclusion` to conclude when the factual diagram contains enough information,

Inference rules

The system contains five inference rules:

- `intros` to introduce hypotheses in the context,
- `apply` to use the information contained in a universal diagram to enrich the factual diagram,
- `conclusion` to conclude when the factual diagram contains enough information,
- `substitute` and `reflexivity` deal with equality.

Correctness and completeness

Intuitionist vs classical logic

For the class of formulas considered intuitionist and classical provability coincide.

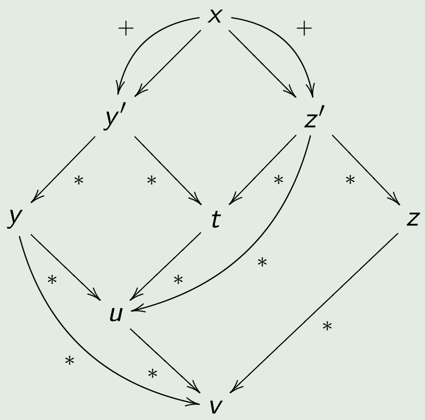
Theorem

The system is correct and complete for the coherent logic (restricted to predicate of arity two).

Induction

The system can be extended to deal with well founded induction.

Newman's lemma

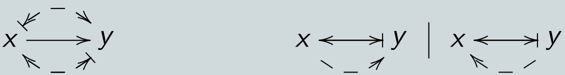


A better understanding of diagrammatic reasoning

To have a diagrammatic proof system we need:

1. Visualization by a syntax that mimics the semantic.

Symmetric closure



2. An inference system which is complete and does not change the conclusion.

intro apply* conclusion

Future work

- Adapt GeoProof to diagrammatic proof in abstract rewriting.

Perspectives

Future work

- Adapt GeoProof to diagrammatic proof in abstract rewriting.
- ...

Perspectives

Future work

- Adapt GeoProof to diagrammatic proof in abstract rewriting.
- ...

Perspectives

- Formalize other ATP methods (Wu...).

Future work

- Adapt GeoProof to diagrammatic proof in abstract rewriting.
- ...

Perspectives

- Formalize other ATP methods (Wu...).
- Adapt GeoProof to the education.

Future work

- Adapt GeoProof to diagrammatic proof in abstract rewriting.
- ...

Perspectives

- Formalize other ATP methods (Wu...).
- Adapt GeoProof to the education.
- Toward a diagrammatic logic (category theory, projective geometry, ...).

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 et visualisation d'un cours de géométrie pour le lycée.
Revue des Sciences et Technologies de l'Information, Technique et Science Informatiques, Langages applicatifs, 24:1113–1138, 2005.
Lavoisier.

Haragauri Narayan Gupta.
Contributions to the axiomatic foundations of geometry.
PhD thesis, University of California, Berkley, 1965.

Gilles Kahn.
Constructive geometry according to Jan von Plato.
Coq contribution, 1995.
Coq V5.10.

Laura Meikle and Jacques Fleuriot.
Formalizing Hilbert's Grundlagen in Isabelle/Isar.
In Theorem Proving in Higher Order Logics, pages 319–334, 2003.

Julien Narboux.
A decision procedure for geometry in Coq.
In Slind Konrad, Bunker Annett, and Gopalakrishnan Ganesh, editors, Proceedings of TPHOLs'2004, volume 3223 of Lecture Notes in Computer Science. Springer-Verlag, 2004.

Julien Narboux.
Toward the use of a proof assistant to teach mathematics.
In Proceedings of the 7th International Conference on Technology in Mathematics Teaching (ICTMT7), 2005.

Julien Narboux.
A formalization of diagrammatic proofs in abstract rewriting.
2006.

Julien Narboux.
A graphical user interface for formal proofs in geometry.
the Journal of Automated Reasoning special issue on User Interface for Theorem Proving, 2006.
to appear.

Julien Narboux.
Mechanical theorem proving in Tarski's geometry.
In Proceedings of Automatic Deduction in Geometry 06, 2006.

Wolfram Schwabhäuser, Wanda Szmielew, and Alfred Tarski.
Metamathematische Methoden in der Geometrie.
Springer-Verlag, Berlin, 1983.

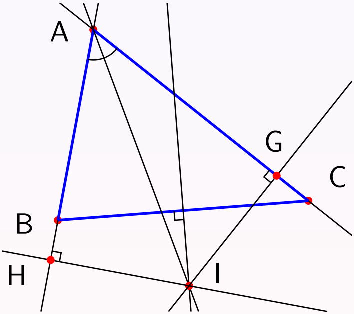
Alfred Tarski.
A decision method for elementary algebra and geometry.
University of California Press, 1951.

Alfred 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.

Alfred Tarski.
The completeness of elementary algebra and geometry, 1967.

Solution

- Let ABC be a triangle.
- Let D be the perpendicular bisector of $[BC]$ and let D' be the bisector of $\angle BAC$.
- Let I be the intersection of D and D'.
- $HI = IG \wedge AH = AG$
- $IB = IC$
- $HB = GC$
- ~~$AB = AC$~~



Back