Formalization of Wu's Simple Method in Coq

Jean-David Génevaux   Julien Narboux   Pascal Schreck

University of Strasbourg - LSIIT - CNRS

Fifth Workshop on Formal and Automated Theorem Proving and Applications, Belgrade, Serbia, 2012

Automatic Deduction in Geometry

Algebraic methods

- Gröbner bases [Kap86]
- Wu's method [Wu78, Cho85, Cho88, Wan01, Wan04]
- Geometric Algebra [LW00]

Synthetic

- Gelernter [Gel59]
- Deductive database [cCsGzZ00]
- The area method [CGZ94]
- Full angle method [CGZ96]

Automatic Deduction in Geometry

Algebraic methods

- Gröbner bases [Kap86][Pot08, GPT10][CW07]
- Wu's method [Wu78, Cho85, Cho88, Wan01, Wan04]
- Geometric Algebra [LW00][FT11]

Synthetic

- Gelernter [Gel59]
- Deductive database [cCsGzZ00]
- The area method [CGZ94][Nar04, JNQ10]
- Full angle method [CGZ96]

Automatic Deduction in Geometry

Algebraic methods

- Gröbner bases [Kap86][Pot08, GPT10][CW07]
- Wu's method [Wu78, Cho85, Cho88, Wan01, Wan04]
- Geometric Algebra [LW00][FT11]

Synthetic

- Gelernter [Gel59]
- Deductive database [cCsGzZ00]
- The area method [CGZ94][Nar04, JNQ10]
- Full angle method [CGZ96]

Outline

An algebraic method

The initial goal is to show that:

$$\forall AB\ldots, h_1 \wedge \ldots \wedge h_k \Rightarrow g$$

This goal is translated into:

$$\forall \overrightarrow{x}, \qquad \bigwedge_i (h_i(\overrightarrow{x}) = 0) \Rightarrow (g(\overrightarrow{x}) = 0)$$

where h_i and g are multivariate polynomials in $\mathbb{F}(x_1, \ldots x_m)$.
We need to show that:

$$\bigcap_i Zeros(h_i) \subseteq Zero(g)$$

Noted:

$$\boxed{Zero(h_i) \subseteq Zero(g)}$$

An algebraic method

The initial goal is to show that:

$$\forall AB \ldots, h_1 \wedge \ldots \wedge h_k \Rightarrow g$$

This goal is translated into:

$$\forall \overrightarrow{x}, \qquad \bigwedge_i (h_i(\overrightarrow{x}) = 0) \Rightarrow (g(\overrightarrow{x}) = 0)$$

where h_i and g are multivariate polynomials in $\mathbb{F}(x_1, \ldots x_m)$.
We need to show that:

$$\bigcap_i Zeros(h_i) \subseteq Zero(g)$$

Noted:

$$\boxed{Zero(h_i) \subseteq Zero(g)}$$

Only equalities $\rightarrow$ unordered geometry.

Which zeros ?

- For elementary geometry, we are interested in ensuring the set of real zeros of the hypothesis polynomials is contained in the set of real zeros of the conclusion polynomial.
- In practice, it often suffices to consider complex zeros instead of the real zeros, but not always.

(First) Incompleteness

Wu's method is incomplete as it considers only complex zeros.
For instance, $\forall x, y. x^2 + y^2 = 0 \implies x = 0 \wedge y = 0$ is true in $\mathbb{R}$ but not in $\mathbb{C}$.

Nullstellensatz

- If $\exists r,\; q_1, \ldots, q_k \;\; g^r = \sum_i q_i h_i$ then $Zero(h_i) \subseteq Zero(g)$.

Nullstellensatz

- If $\exists r,\ q_1, \ldots, q_k\ \ g^r = \sum_i q_i h_i$ then $Zero(h_i) \subseteq Zero(g)$.
- Hilbert's *Nullstellensatz* theorem states that if $\mathbb{F}$ is algebraically closed, then the converse is also true:

$$\exists r,\ q_1, \ldots, q_k\ \ g^r = \sum_i q_i h_i \ \Leftrightarrow\ Zero(h_i) \subseteq Zero(g)$$

That is, we can *always* find such polynomials.

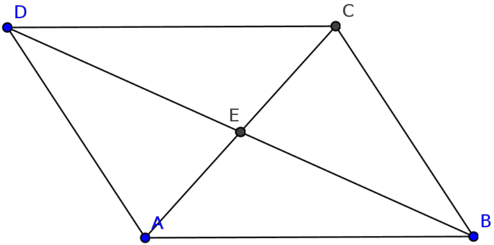
Example

Parallelogram

If $AB \parallel DC$ and $AD \parallel BC$ and $Col\ EAC$ and $Col\ EBD$

then

$$AE \equiv EC$$



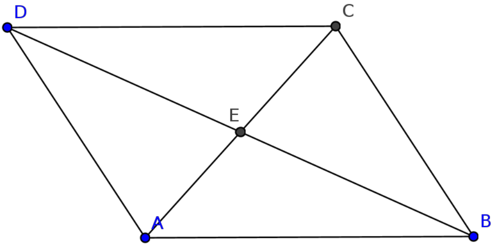
Example

Parallelogram

If $AB \parallel DC$ and $AD \parallel BC$ and $Col\ EAC$ and $Col\ EBD$ and $\neg ColABC$

then

$$AE \equiv EC$$



Non degeneracy conditions

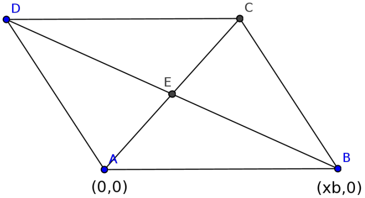
- Non degeneracy conditions (ndgs): $p(x) \neq 0$:
    - $A \neq B$
    - $\neg Col\ ABC$
    - $\neg Parallel\ ABCD$
    - ...
- Non degeneracy conditions is a central issue in formal geometry (see [DDS00, Nar08] for instance)
    - hard to find
    - proofs of degenerated cases are often difficult
- Wu's method generates non degeneracy conditions

$$\boxed{Zero(h_i) - \bigcup Zero(ndgs) \ \subseteq \ Zero(g)}$$

The main ideas:

1. Algebraization
   In practice the choice of a coordinate system is crucial.

$$
\begin{aligned}
h_1 &= -x_B * (y_D - y_C) \\
h_2 &= x_D * y_C + y_D * (x_B - x_C) \\
h_3 &= x_E * -y_C + y_E * x_C \\
h_4 &= (x_E - x_B) * y_D + \\
&\quad y_E * (x_B - x_D) \\
g &= x_E^2 + y_E^2 - \\
&\quad (x_E - x_C)^2 + (y_E - y_C)^2
\end{aligned}
$$



The main ideas:

1. Algebraization
   In practice the choice of a coordinate system is crucial.
2. Triangulation

   In general triangulation is slow, but constructive geometry statements are almost in triangular form.

x_A	y_A	x_B	y_B	x_C	y_C	x_D	y_D	x_E	y_E
x	x	x	x	x	x	x	x		
x	x	x	x	x	x	x	x		
x	x	x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x	x	x

The main ideas:

1. Algebraization
   In practice the choice of a coordinate system is crucial.
2. Triangulation

   In general triangulation is slow, but constructive geometry statements are almost in triangular form.
3. Successive pseudo-division: divide the goal by the hypotheses

The main ideas:

1. Algebraization
   In practice the choice of a coordinate system is crucial.
2. Triangulation
   (using pseudo-division)
   In general triangulation is slow, but constructive geometry statements are almost in triangular form.
3. Successive pseudo-division: divide the goal by the hypotheses

The main character: pseudo-division

Pseudo-division of g by h in the variable v

$$a^k g = qh + r$$

where a is the leading coefficient of h and $deg(r, v) < deg(h, v)$.

Remark 1

If we know that $r = 0$ then :

$$\forall \overrightarrow{x}\ h(\overrightarrow{x}) = 0 \wedge a(\overrightarrow{x}) \neq 0 \implies g(\overrightarrow{x}) = 0$$

Remark 2

$r = prem(g, h)$ belongs to the ideal generated by g and h:

$$r = a^k \times g + (-q) \times h$$

Successive pseudo-division

$$sprem(g, [h_1, \ldots, h_n]) = sprem((prem_g(g, h_n), [h_1, \ldots, h_{n-1}])$$

$$a_1^{k_1} a_n^{k_n} g = q_1 h_1 + q_2 h_2 + \ldots + q_n h_n + r$$

Remark 1'

If we know that $r = 0$ then :

$$\forall \overrightarrow{x} \begin{array}{c} h_1(\overrightarrow{x}) = 0 \wedge \ldots \wedge h_n(\overrightarrow{x}) = 0 \\ \wedge \\ a_1(\overrightarrow{x}) \neq 0 \wedge \ldots \wedge a_n(\overrightarrow{x}) \neq 0 \end{array} \Longrightarrow g(\overrightarrow{x}) = 0$$

Completeness of the method

What does that mean if $r \neq 0$?

For some triangulation process (Ritt/Wu's characteristic sets) a theorem of Wu states that:

- Either the system is not irreducible
- or the theorem is generally false in metric geometry.

Available approaches

- Ltac
  - ☺ High-level language
  - ☹ Hard to debug
  - ☹ Generate large proofs terms
- Ocaml
  - ☺ Full programming language
  - ☺ Easier to debug
  - ☹ Need to generate proof term
- Coq itself (a reflexive approach)
  - ☺ Generate small proofs terms
  - Termination should be proved
- Using a certificate/validator approach
  - ☹ It is not possible to prove completeness
  - ☺ Certificate generator can be written using a different programming language, use heuristics, ...
  - ☺ Same validator can be used for several provers

Our choices:

- Ltac : algebraization, choice of a reference, simplification
- Triangulation and successive pseudo-division:
  - Ocaml: certificate generation
  - Coq: validator using reflection

Limitation

- We can only extract a (self certifying) prover for the *core* of Wu's method

Link between algebraic definition and synthetic geometry

In his Phd (12/2011), Tuan Minh Pham proves that:

```
Lemma transcol :
forall (A B C : Point), col A B C ->
(X A-X B)*(Y C-Y B)-(Y A-Y B)*(X C-X B)=0.
Lemma transparallel :
forall (A B C D : Point),
parallelLine (line A B ) (line C D) ->
(X B-X A)*(Y D-Y C)=(Y B-Y A)*(X D-X C).
Lemma transliesOn :
forall (A B C : Point),
liesOnLine A (line B C) ->
(X B-X A)*(Y C-Y B)-(Y B-Y A)*(X C-X B)=0.
```

```
Lemma transperpendicular :
forall (A B C D : Point),
 perpencicular (line A B) (line C D) ->
(X B-X A)*(X D-X C)+(Y B-Y A)*(Y D-Y C)=0 .
Lemma transsamedistance :=
 forall (A B C D : Point),
 distance A B = distance C D ->
(X B-X A)2 + (Y B-Y A)2=(X D-X C)2 + (Y D-Y C)2.
```

Choice of a coordinate system: the lemmas

In practice the choice of a convenient coordinate system is crucial. Following John Harrison's "Without loss of generality" [Har09], we show that the predicates are invariant under translation and rotation.

Example: collinear

```
Lemma collinear_inv_translation: forall A B C V,
 collinear A B C <->
 collinear (trans A V) (trans B V) (trans C V).

Lemma collinear_inv_rotation: forall A B C cos sin,
 cos*cos + sin*sin = 1 ->
 (collinear A B C <->
  collinear (rot A cos sin) (rot B cos sin) (rot C cos sin)).
```

Proofs can be done using ring/Gröbner basis.

Choice of a coordinate system: the tactic

The tactic

Algebraization O I H

The following predicates/functions are available:

collinear, parallel, orthogonal, midpoint, intersection of lines, square of length, equality of points, angles or lengths.

Limitations

- The tactic can not deal with user defined predicates automatically. Adding a new predicate requires to add the lemmas for invariance under translation and rotation and to update the tactic.

Design of the certificate

Main idea:

Provide r, l and $q_1, \ldots, q_k$ such that:

$$l \times g^r = \sum_i q_i \times h_i$$

Grégoire, Pottier, Théry's idea:

Use `let ... in` to compress this certificate using sharing (straight line programs).

$$\text{let } p_1 = q_1 * h_1 + q_2 * h_2 \text{ in}$$
$$\text{let } p_2 = q_3 * p_1 + q_4 * h_2 \text{ in}$$
$$\ldots$$

Generation of the certificate

We need a prover based on Wu's method which generate a *certificate*.

- Existing implementations either not open source or relying on proprietary CAS (Maple)[1].
- We aim at integration into Coq.

Hence we developed our own implementation of Wu's method in Ocaml based on a slightly optimized version of Loïc Pottier library for multivariate polynomials.

Second incompleteness

Our implementation is incomplete because we do not check polynomials for irreducibility (this requires factorization).

[1] OpenGeoProver was not available when we started this work.

Certificates generation

We generate certificates for:

1. The pseudo-division
2. The successive pseudo-division
3. Triangulation

Certificate generation I

1. We just need to keep the quotient and the lead coefficient:

$$r = a^k \times g + (-q) \times h$$

```
let pseudo_div_num g h x certif =
    let (r,a,k,q) = pseudo_div (g.p) (h.p) x in
    let new_n = new_num () in
      certif := (new_n, r, [(a^^k, g.n);(p_zero -- q, h.n)])
                ::(!certif);
      {p=r ; n= new_n}
```

Certificate generation II

2. We know that :

$$l \times g = \sum_i (t_i \times s_i) + r_{final}$$

where

$$s_i = q_i \times \prod_{j=1}^{i-1} c_j^{d_j}$$

3. The triangulation phase is based on pseudo-division.
Invariant: the polynomials are in the ideal generated by the hypotheses.

$$[h_1, \ldots, h_i, \ldots, h_j, \ldots, h_n] \rightarrow [h_1, \ldots, h_i, \ldots, prem(h_i, h_j, v), \ldots, h_n]$$

Certificate generation III

Remark

- To prove correctness we *do not need* to prove that the triangulation phase really triangulates.
- This shows that proving the method in Coq itself would not be so difficult.

Checking the certificate

Algorithm

- Computing $l \times g^r$ and $\sum_i q_i \times h_i$
- Checking equality using ring tactic normalization function

Reuse:

We reuse the validator proven correct by Grégoire, Pottier and Théry.

Technical limitation:

All shared polynomials must be in the ideal.

- Well-adapted to Gröbner basis
- Using Wu's method some other polynomials could be shared.

Benchmark I

Theorem	Wu / Caml	Wu / Coq	GB / Coq	Wu / GB
Pascal_2	0.013	21	-	-
Pascal_1	0.024	22	1652	$\times$75
Ptolemy95	0.010	10	30	$\times$3
Pappus	0.043	3	8	$\times$2.6
Altitudes	0.002	3	7	$\times$2.3
Simson	0.002	5	8	$\times$1.6
Perp-bisect	0.001	2	3	$\times$1.5
Pythagore	0.001	1	1	$\times$1
Feuerbach	0.038	15	15	$\times$1
Isoceles	0.001	1	1	$\times$1
Euler Line	0.063	9	6	$\times$0.6
Medians	0.001	3	2	$\times$0.6
Chords	0.015	4	2	$\times$0.5
Thales	0.003	6	3	$\times$0.5
Bissectors	0.001	6	3	$\times$0.5
Desargues	0.027	99	10	$\times$0.1
Ceva	0.025	98	6	$\times$0.06

Intel(R) Core(TM) i5 CPU 750 @ 2.67GHz with 4Gb RAM.

Benchmark II

Checking certificate using Ocaml:

- Between 1% and 80%
- On average: 50%

Conclusion

- Certificate based approaches are flexible: we could reuse Pottier et al. checker.
- But certificate checking time is significant.
- Wu's method and Gröbner basis seems to be complementary.

Perspectives

- Implement the full method of Wu.
- Add automatic geometrization.
- Add automatic choice of a reference.
- Other data-structure for certificates (pseudo-division ?).

Questions ?

Bibliography I

Shang ching Chou, Xiao shan Gao, and Jing zhong Zhang.
A deductive database approach to automated geometry theorem proving and discovering.
Journal of Automated Reasoning, 25:219–246, 2000.

Shang-Ching Chou, Xiao-Shan Gao, and Jing-Zhong Zhang.
Machine Proofs in Geometry.
World Scientific, Singapore, 1994.

Shang-Ching Chou, Xiao-Shan Gao, and Jing-Zhong Zhang.
Automated generation of readable proofs with geometric invariants, theorem proving with full angle.
Journal of Automated Reasoning, 17:325–347, 1996.

Shang-Ching Chou.
Proving and discovering geometry theorems using Wu's method.
PhD thesis, The University of Texas, Austin, December 1985.

Bibliography II

Shang-Ching Chou.
Mechanical Geometry Theorem Proving.
D. Reidel Publishing Company, 1988.

Amine Chaieb and Makarius Wenzel.
Context aware Calculation and Deduction — Ring Equalities via Gröbner Bases in Isabelle.
In M. Kauers, M. Kerber, R. Miner, and W. Windsteiger, editors, *CALCULEMUS 2007*, volume 4573 of *Lecture Notes in Computer Science*, pages 27–39. Springer, 2007.

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.

Bibliography III

Laurent Fuchs and Laurent Théry.
A Formalisation of Grassmann-Cayley Algebra in Coq.
In *Post-proceedings of Automated Deduction in Geometry (ADG 2010)*, 2011.

H. Gelernter.
Realization of a geometry theorem machine.
In *Proc. Int. Conf. in Info. Process*, pages 273–282, Paris, 1959.

Benjamin Grégoire, Loïc Pottier, and Laurent Théry.
Proof certificates for algebra and their application to automatic geometry theorem proving.
In *Post-Proceedings of ADG 2008*, volume 6301 of *LNAI*, 2010.

Bibliography IV

John Harrison.
Without loss of generality.
In Stefan Berghofer, Tobias Nipkow, Christian Urban, and Makarius Wenzel, editors, *TPHOLs*, volume 5674 of *Lecture Notes in Computer Science*, pages 43–59. Springer, 2009.

Predrag Janičić, Julien Narboux, and Pedro Quaresma.
The Area Method: a Recapitulation.
Journal of Automated Reasoning, 2010.
online first.

Deepak Kapur.
Geometry Theorem Proving using Hilbert's Nullstellensatz.
In *SYMSAC '86: Proceedings of the fifth ACM symposium on Symbolic and algebraic computation*, pages 202–208, New York, NY, USA, 1986. ACM Press.

Bibliography V

Hongbo Li and Yihong Wu.
Mechanical theorem proving in projective geometry with bracket algebra.
Computer Mathematics, pages 120–129, Singapore, 2000. World Scientific.

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.
Mechanical theorem proving in Tarski's geometry.
In *Post-proceedings of Automatic Deduction in Geometry 06*, volume 4869 of *Lecture Notes in Computer Science*, pages 139–156. Springer, 2008.

Bibliography VI

Loïc Pottier.
Connecting Gröbner Bases Programs with Coq to do Proofs in Algebra, Geometry and Arithmetics.
In G. Sutcliffe, P. Rudnicki, R. Schmidt, B. Konev, and S. Schulz, editors, *Knowledge Exchange: Automated Provers and Proof Assistants*, CEUR Workshop Proceedings, page 418, Doha, Qatar, 2008.

Dongming Wang.
Elimination Method.
Springer-Verlag, 2001.

Dongming Wang.
Elimination Practice.
Springer-Verlag, 2004.

Bibliography VII

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.