Automated Proving in Geometry using Gröbner Bases in Isabelle/HOL

Danijela Petrović

Faculty of Mathematics, University of Belgrade, Serbia

Abstract. For several decades, algebraic method have been successfully used in automated deduction in geometry. Objects in Euclidean geometry and relations between them are expressed as polynomials, and algebraic methods (e.g., Gröbner bases) are used over that set of polynomials. We describe a formalization of an algorithm in Isabelle/HOL that accepts a term representation of a geometry construction and returns a corresponding set of polynomials. Our further work will be to use the method of Gröbner bases within Isabelle system on the generated polynomials, in order to implement a formally verified algebraic prover for geometry.

1 Introduction

Automated reasoning in geometry. In the past 20 years highly successful methods for geometry theorem proving and discovering have been developed. Generally there are two approaches to proving geometry theorems using computers: the artificial intelligence (AI) approach and the algebraic computation approach. The earliest work in geometry theorem proving by computer programs was done by Gelernter and his collaborators[2].

A breakthrough in automated geometry theorem proving is made by Wu [3]. Wu introduced a method in 1977 which can be used to prove quite difficult geometry theorems efficiently. The basic idea of Wu's method is to transform geometry construction into set of polynomials and then using certain algebraic transformations over these polynomials show that statement holds. Later it was clarified that the algebraic tools needed in Wu's approach can be developed from Ritt's work[1]. The algebraic aspect of this approach is now known as Wu-Ritt's characteristic set (CS) method. It is now the case that hundreds of theorems in Euclidean and non-Euclidean geometries can be proved automatically by computers programs.

The success of Wu's method has revived interest in proving geometry theorems by computers. In particular, the application of Gröbner basis method [4] to the same class of geometry theorems that Wu's method addressed has been successfully investigated. The main disadvantage of these methods is that they can only deal with theorems involving equalities, but not inequalities.

All above methods have the same character that they first transform geometry properties into equations in coordinates of the related points and then deal

with these equations. The search for a vector based method began in the mid-eighties, because it is believed that such a method would produce more elegant proofs. Of these methods, the area method was very successful.

Methods of automated reasoning in geometry have a wide range of applications, including kinetic analysis for robotics, solid modeling, linkage design etc.

Formal theorem proving. Formal mechanized theorem proving assumes formalizing mathematical statements within *proof assistants* — specialized software tools used to find and check proofs semi-automatically, guided by user interaction. Formal theorem proving is used both in classical mathematics and in hardware and software production. Using mechanical theorem prover significantly increases the confidence in mathematical results, because many mathematical problems are usually so complex that there is no strong confidence that pen-and-paper reasoning about them is sound.

One of the leading proof assistants used for interactive theorem proving is Isabelle/HOL[5]. Isabelle is a generic system for implementing logical formalisms, and Isabelle/HOL is the specialization of Isabelle for Higher-Order Logic (HOL). It could be said that Isabelle/HOL merges Functional Programming and Logic. Isabelle/Isar is a high-level language for writing proofs in a declarative manner. Working with Isabelle assumes creating theories. Roughly speaking, a theory is a named collection of types, objects and functions, and theorems.

The central motivation of this work is to connect automated and formal proving in geometry and to construct a formally verified automated prover for geometry. Before applying algebraic methods, geometry constructions and statements are transformed into set of polynomial equations, but there is no unique, nor verified algorithm for this. Usually transformation is done by ad-hoc methods and there is no formal link between obtained polynomials and given geometric objects. With a formally verified translation method this problems would be resolved and this work is a step into that direction.

2 Using algebraic methods for proving geometry theorems

2.1 Ideal membership problem and Gröbner basis method

Before applying algebraic methods to prove geometry theorems, theorems are translated into polynomial equations and usually it suffices to show that some polynomials representing the statement belong to the ideal generated by the polynomials representing the construction. This kind of problem is formalized by the next definition[7].

Definition 1. *Let* $f, f_1, \ldots, f_k \in K[X_1, \ldots, X_n]$ *where* $f, f_1, \ldots, f_k$ *are polynomials and* $K[X_1, \ldots, X_n]$ *is a polynomial ring over* K*, and let* $< f_1, \ldots f_k >$ *be an ideal generated by* $f_1, \ldots f_k$*. The* ideal membership problem *is the problem of checking if* $f \in < f_1, \ldots f_k >$ *satisfied.*

Gröbner bases give the tool that is used to give an answer to ideal membership problem.

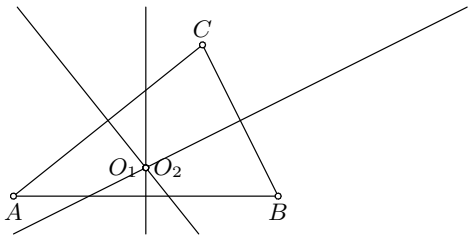
Definition 2. *Let $I =< f_1, \ldots, f_n >$ be an ideal generated by polynomials f_1, $\ldots$, f_n. G is a* Gröbner basis *of an ideal I if and only if multivariate division (denoted by $\rightarrow_G$) of any polynomial in the ideal I by G gives 0.*

Theorem 1. *The ideal membership problem $f \in I$ is equivalent to $f \overset{*}{\rightarrow}_G 0$.*

2.2 Applying the Gröbner basis method for proving a geometry theorem — an example

The next simple example we illustrate how a simple geometric statement can be reduced to the ideal membership problem and solved by using the Gröbner basis method.

Bisectors of the sides of a triangle are intersecting in one point.



The basic idea is to place the figure above in the coordinate plane and then to interpret the hypotheses of the theorem as statements in coordinate, rather than Euclidean, geometry. So we begin by coordinatizing the parallelogram by placing the point A at the origin, so $A = (0, 0)$. Now we can say that the point B corresponds to $(c, 0)$, and that C corresponds to (a, b) (it is obvious that the theorem should be proved for any set of coordinates, but it can be easily shown that we can translate any coordinates into these ones). Now, this geometric construction is going to be translated into set of polynomials.

Assume that bisectors of sides AB and BC are intersecting in point $O_1 = (x_1, y_1)$. Bisectors are completely determined by points A, B, C and O_1 and this yields the following equations:

$$p_1 : \qquad x_1 - \tfrac{c}{2} = 0$$

$$p_2 : \tfrac{c-a}{b} \cdot x_1 - y_1 + \tfrac{b^2-c^2+a^2}{2 \cdot b} = 0$$

Assume that bisectors of sides AB and AC are intersecting in point $O_2 = (x_2, y_2)$. This yields another two equations:

$$p_1' : \qquad x_2 - \tfrac{c}{2} = 0$$

$$p_3 : \tfrac{a}{b} \cdot x_2 + y_2 - \tfrac{b}{2} - \tfrac{a^2}{2 \cdot b} = 0$$

Thus, there is a set:

$$G = \{p_1, p_2, p_1', p_3\}$$

These polynomials describe the construction. Now, it should be proved that $O_1 = O_2$. i.e. $(x_1, y_1) = (x_2, y_2)$. In order to apply the Gröbner basis method, Gröbner basis G' of the set G is going to be calculated and the aim is to prove $x_1 - x_2 \rightarrow_{G'} = 0$ and $y_1 - y_2 \rightarrow_{G'} = 0$. With the polynomials on the left hand side of these equations the statement is described.

In order to compute the Gröbner basis of the set G Buchberger's algorithm is used and finally calculated result is:

$$\begin{aligned} G' = \{g_1, g_2, g_3, g_4, g_5, g_6\} = \{ & x_1 - \frac{c}{2}, \quad x_2 - \frac{c}{2}, \\ & \frac{c-a}{b} \cdot x_1 - y_1 + \frac{b^2 - c^2 + a^2}{2 \cdot b}, \quad \frac{a}{b} \cdot x_2 + y_2 - \frac{b}{2} - \frac{a^2}{2 \cdot b}, \\ & -y_1 + \frac{b}{2} + \frac{a^2}{2 \cdot b} - \frac{a \cdot c}{2 \cdot b}, \quad y_2 - \frac{b}{2} - \frac{a^2}{2 \cdot b} + \frac{a \cdot c}{2 \cdot b} \} \end{aligned}$$

Using this, $x_1 - x_2 \overset{*}{\rightarrow}_{G'} 0$ can be easily shown. Indeed,

$$x_1 - x_2 \quad \rightarrow_{g_1} \quad -x_2 + \frac{c}{2} \quad \rightarrow_{g_3} \quad -x_2 + \frac{c}{2} + x_2 - \frac{c}{2} = 0 \, .$$

Similarly $y_1 - y_2 \overset{*}{\rightarrow}_{G'} 0$ can be shown,

$$\begin{aligned} & y_1 - y_2 \quad \rightarrow_{g_5} \quad -y_2 + \frac{b}{2} + \frac{a^2}{2b} - \frac{ac}{2b} \quad \rightarrow_{g_6} \\ \rightarrow_{g_6} \quad & -y_2 + \frac{b}{2} + \frac{a^2}{2b} - \frac{ac}{2b} + y_2 - \frac{b}{2} - \frac{a^2}{2b} - \frac{ac}{2b} = 0 \, . \end{aligned}$$

3 Formalization in Isabelle/HOL

Being able to show geometry statements using polynomials and Gröbner basis gives the opportunity to automate proving geometric statements. However, the connection between geometry and algebra is usually not formally given. Our main goal is to do this and prove the correctness of the whole Gröbner basis method. The most important part is the verification of the step that translates geometry constructions to polynomial equations and to the ideal membership problem.

3.1 Term representation of geometry constructions

First, it is necessary to represent geometry constructions in a convenient way so that it can be easily processed by computer, i.e. used by our algorithm. Thus, geometry constructions are represented using terms. Currently, two types of objects are supported – points and lines. Also, geometry statements are represented as terms. In Isabelle corresponding data types are defined by:

```
datatype
point_term = MkPoint nat (* Independent points determined by their index *)
          | MkIntersection line_term line_term
          | MkMidpoint point_term point_term

and line_term =  MkLine point_term point_term
          | MkNormal line_term point_term
          | MkParallel line_term point_term
          | MkBisector point_term point_term

datatype statement_term = EqualP point_term point_term
          | EqualL line_term line_term
          | Incident point_term line_term
          | Midpoint point_term point_term point_term
          | Parallel line_term line_term
          | Normal line_term line_term
          | Colinear point_term point_term point_term
```

As can be seen, a point can be specified with its identifier, or it can be constructed as an intersection of two lines, or it can be constructed as a midpoint between two points. Similarly, a line can be constructed as a line determined with two points or it can be constructed as bisector of a segment etc. Also, there are different type of statements. For example, `Incident point_term line_term` denotes that a point belongs to a line, `EqualP point_term point_term` denotes that two points are equal etc.

Example 1: The term `Incident (MkPoint 1) (MkLine (MkPoint 1) (MkPoint 2))` represents the statement that a point belongs to a line determined with that point and another one.

Example 2: The term

```
let c = MkBisector (MkPoint 1) (MkPoint 2);
   b = MkBisector (MkPoint 1) (MkPoint 3);
   a = MkBisector (MkPoint 2) (MkPoint 3);
   O1 = MkIntersection a b;
   O2 = MkIntersection a c in
   EqualP O1 O2
```

represents the example from the previous section — the two points given from the line intersection are equal.

Having defined the term representation of geometry constructions, the next step is to translate the term into a set of polynomials so that the Gröbner basis method can be applied.

3.2 Brief description of the translation algorithm

The algorithm is used to translate the term representation of geometry constructions and statements into their polynomial counterparts. The algorithm is

recursive and actually produces two sets. The first set is the set of polynomials representing geometry construction and thus is called the *construction-set*. The second set is the set of polynomials representing the statement and it will be called the *statement-set*. The Gröbner basis method relies on showing that each polynomial in the statement-set can be reduced to zero by using the Gröbner basis calculated for the construction-set.

The algorithm recursively processes terms and for all unknown objects new coordinates are added. Also, at the same time, new polynomials are added to the corresponding sets regarding identities in analytic geometry.

As an example, let us show the translation step for the statement of the form `Incident point_t line_t`, where `point_t` and `line_t` can be arbitrarily complex terms for a point and a line. The algorithm for this particular example works like this:

- we add new variables x_0 and y_0. These variables are unknown coordinates for the point O given by the term `point_t` — $O(x_0, y_0)$.
- we add variables a_0, b_0, and c_0 and these are unknown coefficients for the line p given by the term `line_t` — $p = a_0 \cdot x + b_0 \cdot y + c_0$.
- we call function $point\_poly(point\_t, x_0, y_0)$ that construct polynomials connecting the variables x_0 and y_0 with the term `point_t`.
- we call function $line\_poly(line\_t, a_0, b_0, c_0)$ that construct polynomials connecting the variables a_0, b_0 and c_0 with the term `line_t`.
- we add the polynomial $a_0 \cdot x_0 + b_0 \cdot y_0 + c_0$ in statement-set.

Having in mind that the idea of this work is to formally prove correctness of the method, it should be clear that everything used in the description of the method must be formally proved and thus is case with polynomials. This means that it is desirable to have such a formally proved theory that we could be able to represent polynomials and preform different calculations with them. Since this is not the main focus of this work and is already developed before, the theory used here is the Theory of Executable Multivariate Polynomials [6]. In this work the authors represent polynomials using lists and formally prove many of their properties.

As an illustration, we show a fragment of Isabelle code implementing this translation step.

```
translate (Incident p l) ==
  let x = point_id_x 0; y = point_id_y 0;
      a = line_id_a 0;  b = line_id_b 0; c = line_id_c 0;
      (s', pp) = point_poly p x y (| maxp = 0, maxl = 0 |);
      (_, lp) = line_poly l a b s'  in
  (sup pp lp,
   Fset.Set[poly_of (PSum [PMult[PVar a, PVar x],
                           PMult[PVar b, PVar y], PVar c])])"
```

Here there are two new functions *point_poly* and *line_poly* that have two arguments — the term and two variables. These functions are mutually recursive and are used to calculate the construction-set As an example, consider

`MkIntersect line1_t line2_t`. As before, `line1_t` and `line2_t` are terms representing lines and this term is representing a point. What should be calculated here are the polynomials describing the point. These polynomials depend on terms and variables. In this example the method will work like this:

- we add variables a_1, b_1 and c_1 and these are unknown coefficients for the line p given by the term `line1_t` — $p = a_1 \cdot x + b_1 \cdot y + c_1$.
- we add variables a_2, b_2 and c_2 and these are unknown coefficients for the line q given by the term `line2_t` — $q = a_2 \cdot x + b_2 \cdot y + c_2$.
- we call function $line\_poly(line1\_t, a_1, b_1, c_1)$.
- we call function $line\_poly(line2\_t, a_2, b_2, c_2)$.
- we add polynomials $x(a_2 \cdot b_1 - a_1 \cdot b_2) + b_1 \cdot c_2 - c_1 \cdot b_2 = 0$ and $y(a_2 \cdot b_1 - a_1 \cdot b_2) + a_2 \cdot c_1 - a_1 \cdot c_2 = 0$ to the construction-set.

These polynomials are gained using geometric identities so that a given geometric property holds.

3.3 Proving correctness

The main part of our work is to prove the correctness of our translation by showing the connection between the geometric statement and the obtained sets of polynomial equations. In order to do so, analytic geometry will be used as a connection between the synthetic geometry and algebra. It should be shown that everything that is proved using the algebraic method also holds in models of the synthetic geometry. On the one hand, it could be shown that analytic geometry is a model of synthetic, and furthermore it could be shown that all models are isomorphic, so everything that holds in one model holds in all others. On the other hand, it should be shown that everything that is proved using algebraic method is also correct in analytic geometry.

Connection between synthetic and analytic geometry. Let us consider the first problem — to show that analytic geometry is a model of synthetic geometry. Basic objects in analytic geometry have to be formally defined. For example, a point can be defined as a pair of two real numbers:

```
type_synonym point = "real * real"
```

Now, defining the line is a bit more complex. Lines can be identified by triples of their coefficients. However, first two components cannot be zero simultaneously. Additionally, lines can have different coefficients and still be the same (if coefficients are proportional). For example $x+2 \cdot y+1 = 0$ and $2 \cdot x+4 \cdot y+2 = 0$ determine the same line. Two lines $a1 \cdot x + b1 \cdot y + c1 = 0$ and $a2 \cdot + b2 \cdot y + c2 = 0$ are equal if and only if $\exists k.a1 = k \cdot a2, b1 = k \cdot b2, c1 = k \cdot c2$. In order to represent a line we define an equivalence relation between triples $(a1, b1, c1)$ and $(a2, b2, c2)$ such that $\exists k.a1 = k \cdot a2, b1 = k \cdot b2, c1 = k \cdot c2$. A line can be defined as an equivalence class over the set $\{(a, b, c) \mid a, b, c \in \mathbb{R}, a \neq 0 \vee b \neq 0\}$. Isabelle code formalizing this definition is:

```
typedef line_coeffs = "{(A::real, B::real, C::real). A ~= 0 | B ~= 0}"
by auto

definition line_coeffs_eq :: "line_coeffs => line_coeffs => bool" where:
 "line_coeffs_eq c c1 =
   (EX A B C A1 B1 C1.
    (Rep_line_coeffs c = (A, B, C) &  Rep_line_coeffs c1 = (A1, B1, C1) &
      (EX k. k ~= 0 & A1 = k*A & B1 = k*B & C1 = k*C)))"

lemma line_coeffs_eq_equivp: "equivp line_coeffs_eq"
(* prove that line_coeffs_eq is an equivalence relation *)

quotient_type  line = line_coeffs / "line_coeffs_eq"
by (rule line_coeffs_eq_equivp)
```

Based on these definitions, additional geometric primitives (e.g., incidency) can be defined and their geometric properties (e.g., Hilbert's axioms) can be proved.

```
lemma ax2:
  assumes "P1 ~= P2"
   "incident P1 l1" "incident P2 l1" "incident P1 l2" "incident P2 l2"
  shows "l1 = l2"
```

Connection between analytic geometry and algebra. The other thing we need to do is to show that our method is correct, i.e. that if we show something using the Gröbner basis method, it really holds in analytic geometry. To do so we need to show this:

$$(\forall(u,x))(\forall g \in G)((\forall f \in F.f(u,x)=0) \Rightarrow g(u,x)=0) \Rightarrow \text{geometric statment}$$

where $F(u,x)$ is a construction-set and $G(u,x)$ is a statement-set. From the geometric construction in analytic geometry it is necessary to show that $\forall f \in F$ and $\forall(u,x)\ f(u,x)=0$ This is the first part. The second part is to show that if it is proved $(\forall g \in G)(\forall f \in F.f(u,x)=0) \Rightarrow g(u,x)=0)$ then the geometric statement in analytic geometry holds. This will be inductively proved in Isabelle/HOL. In the proof objects used in the algebraic proof gain coordinates and then the proof can be connected with analytic geometry. Consider the following example:

```
Incident (MkMidpoint (MkPoint 0) (MkPoint 1)) (MkLine (MkPoint 0) (MkPoint 1))
```

`MkPoint 0` `MkPoint 1` get coordinates (p_0^x, p_0^y) and (p_1^x, p_1^y) and these are fixed. Then `MkMidpoint (MkPoint 0) (MkPoint 1)` gets coordinates (x_1, y_1) and these depend of the coordinates of `MkPoint 0` and `MkPoint 1`. The same holds of `MkLine (MkPoint 0) (MkPoint 1)` where we add coordinates (a_1, b_1, c_1) for line. The following equations should be satisfied:

$$2x_1 = p_0^x + p_1^x \qquad a_1(p_1^x p_0^y - p_1^y p_0^x) - c_1(p_1^y - p_0^y) = 0$$
$$2y_1 = p_0^y + p_1^y \qquad b_1(p_1^x p_0^y - p_1^y p_0^x) + c_1(p_1^x - p_0^x) = 0$$

and as a conclusion there is the equation: $\qquad a_1x_1 + b_1y_1 + c_1 = 0$

Using identities in analytic geometry this is easily shown.

4 Conclusion and future work

Since the work presented here is at its early stages there are many things that should be improved. First of all, the algorithm for translating the term representation of geometry construction to sets of polynomials should be optimized. Next, more geometric objects (circles, ellipses etc.) should be included and more types of geometric statements should be added. Many of the proofs are not yet done and currently this is the main focus of our work.

Further, the translation method should be connected to trusted implementation of the Gröbner basis construction (already available in Isabelle/HOL). In this way, we would have a fully formally verified automated prover for geometry.

Since our translation does not depend on the algebraic method used for the generated sets of polynomials, a significant part of our formalization can be reused for making a fully verified implementation of other algebraic methods — most notably, Wu's method.

References

1. Ritt J., *Differential Algebra*. American Mathematical Society, 1950.
2. H. Gelernter, *Realization of a geometry theorem proving machine*. IFIP Congress', pp. 273 281 1959.
3. Joran Elias, *Automated Geometric Theorem Proving: Wu's Method*. The Montana Mathematics Enthusiast, vol.3, no.1, p.3-50, 1998.
4. Franz Baader, Tobias Nipkow, *Term rewriting and all that*. Cambridge University Press, 1998.
5. Tobias Nipkow, Markus Wenzel, Lawrence C. Paulson, *A Proof Assistant for Higher-Order Logic*. Springer-Verlag, 2009.
6. Christian Sternagel, René Thiemann, Executable Multivariate Polynomials, In G. Klein et al eds., *The Archive of Formal Proofs*. Available http://afp.sourceforge.net/entries/Polynomials.shtml 2010.
7. John Harrison, *Handbook of Practical Logic and Automated Reasoning*. Springer-Verlag, 2009.