E U C L I D
The Geometry Theorems Prover

Predrag Janičić, Stevan Kordić

Faculty of Mathematics
University of Belgrade

1 Introduction

Geometry is one of the mathematical disciplines demanding a big deal of the human intuition. That's why it is chalenging task to make a program solving a geometry problems. Program EUCLID proves theorems of geometry in a intuitive, geometrical way (*more geometrico*), and presents proves in a natural language form. Besides, the mechanism and the basic principles of the prover EUCLID led us to the new form of the foundation of geometry and the new classification of geometrical axioms.

Program EUCLID was written in Arity PROLOG, but essential machanism of the prover does not rely on PROLOG mechanisms. Despite the limited resources of Arity PROLOG, the program was written in PROLOG because of its suitable characteristics: flecsibility, mechanism of unification etc.

2 Prover EUCLID - The Basic Principles

There are three modules in program EUCLID: the module of axioms, the knowledge-pool and the proving mechanism. Although these modules are independent they are built as a coherent system. Besides, these modules are related by internal language in which all knowledge and conclusions are expressed. The final output - proof of the certain theorem is written in a natural language form. Because of its importance, first of all, let us focus our attention at internal language.

3 Internal Language

The internal language L of the prover covers all objects and relations accuring in geometrical axioms. Also, theorem that is to be proved has to be expressed in the internal language, so the internal language is important for user, also. All relations of the internal language L (including unary relations defining objects) are shown in table 1.

predicate	meaning
$t(a)$	a is a point
$l(b)$	b is a line
$p(c)$	c is a plane
$identical(a, b)$	a and b are identical
$non\_identical(a, b)$	a and b are not identical
$i(a, b)$	a and b are incident
$non\_i(a, b)$	a and b are not incident
$b(a, b, c)$	b lies between a and c
$non\_b(a, b, c)$	b does not lie between a and c
$c(a, b, c, d)$	pair (a, b) is congruent to pair (c, d)
$collinear(a, b, c)$	a, b and c are collinear
$non\_collinear(a, b, c)$	a, b and c are not collinear
$coplanar(a, b, c, d)$	a, b, c and d are coplanar
$non_coplanar(a, b, c, d)$	a, b, c and d are not coplanar
$intersect(a, b)$	a intersects b
$non\_intersect(a, b)$	a does not intersect b

Table 1. Relations (primitive and defined) in system EUCLID

Internal representation of relations (except for unary relations) has one argument more then in a table, and a value of that argument is an index of relation in a knowledge-pool.

We denoted by $\mathcal{E}_L$ the class of all geometry theorems that can be expressed either as:

$$\forall x_1, \forall x_2, ...\forall x_n, \exists Y_1, \exists Y_2, ...\exists Y_m(\phi(x_1, x_2, ...x_n \Rightarrow \psi(x_1, x_2, ...x_n, Y_1, Y_2, ...Y_m))$$

or as:

$$\forall x_1, \forall x_2, ...\forall x_n, \exists Y_1, \exists Y_2, ...\exists Y_m(\phi(x_1, x_2, ...x_n \Rightarrow$$

$$\psi_1(x_1, x_2, ...x_n, Y_1, Y_2, ...Y_m) \vee \psi_2(x_1, x_2, ...x_n, Y_1, Y_2, ...Y_m) \vee ...$$
$$... \vee \psi_k(x_1, x_2, ...x_n, Y_1, Y_2, ...Y_m))$$

or as:

$$\forall x_1, \forall x_2, ...\forall x_n(\phi(x_1, x_2, ...x_n) \Rightarrow \psi(x_1, x_2, ...x_n))$$

or as:

$$\exists Y_1, \exists Y_2, ...\exists Y_m(\psi(Y_1, Y_2, ...Y_m))$$

where ϕ, ψ and ψ_i are conjunctions of L relations ranging over some of the arguments $x_1, x_2, ...x_n$ and $Y_1, Y_2, \ldots Y_m$.

The first mentioned form we shall call *universal-existential form* ($\forall - \exists$), the second *universal-existential-disjunctive form* ($\forall - \exists - \vee$), the third *universal form* ($\forall$) and the fourth *existential form* ($\exists$). All of them we shall denote by $\mathcal{F}$.

We interpret the $\mathcal{E}_L$ theorems in program EUCLID in a appropriate PROLOG way. The class $\mathcal{E}_L$ will be the subject of the following text.

4 The Module of Axioms

Module of axioms consists of all geometrical axioms whithout continuity axioms and so called ADT module. ADT module consists of the following axiom, definitions and trivial theorems:

- The additional geometry axiom:

$$\forall a, \forall b, \forall c(i(a, b) \wedge i(b, c)) \Rightarrow i(a, c))$$

- The identity axioms:

$$\forall a\, identical(a, a)$$
$$\forall a, \forall b\, (identical(a, b) \Rightarrow identical(b, a))$$
$$\forall a, \forall b,\, \forall c\, (identical(a, b) \wedge identical(b, c) \Rightarrow identical(a, c))$$

- The substitution axioms:

$$\forall a_1, \ldots \forall a_i, \ldots \forall a_n, \forall b(\phi(a_1, \ldots a_i, \ldots a_n) \wedge identical(a_i, b) \Rightarrow \phi(a_1, \ldots b, \ldots a_n)).$$

  where ϕ is a L relation.

- The definitions:

  If points a, b and c and line l are such that $i(a, l), i(b, l), i(c, l)$, then we shall say that the points a, b and c are *collinear*.

  If points a, b, c and d and plane p are such that $i(a, p), i(b, p), i(c, p), i(d, p)$, then we shall say that the points a, b, c and d are *coplanar*.

  If a is a point and $i(a, b), i(a, c)$, then we say that b intersects c.

- The trivial theorems:

  If points a, b and c and line l are such that $i(a, l), i(b, l), non\_i(c, l)$, then we shall say that the points a, b and c are non collinear.

  If points a, b, c and d and plane p are such that $i(a, p), i(b, p), i(c, p), non\_i(d, p)$, then we shall say that the points a, b, c and d are not coplanar.

Let us note that all axioms, definitions and theorems just listed above are of one of the $\mathcal{F}$ forms. Also, each geometry axiom (excluding continuity axiom) can be put in one of the $\mathcal{F}$ forms [1] .

We denoted by $\mathcal{E}$ the class of all $\mathcal{E}_L$ theorems which can be established by use of the elements from the module of axioms. It can be shown that program EUCLID can prove all $\mathcal{E}$ theorems.

Although prover EUCLID does not use any set-theory segment, class $\mathcal{E}$ is wide enough to cover great part of usual elementary geometry courses. Also, any geometry theorem could be included into the module of axioms for the sake of more efficient and simplier proving process.

According to the mechanism of the prover, order of the axioms is very important and determine the way of establishing of a theorem. Efficience of the prover is related to the order of axioms and this inspired us for the new classification of geometrical axioms. There are divided into five groups:

-"identity" axioms;

-"unproductive" axioms;

-"branching" axioms;

-"productive" axioms;

-"strongly productive" axioms.

Each group of axioms has a different status, and order of the axioms in a each group is also of the great importance. This order determines efficience of the prover.

[1]Tarski wrote about these forms of geometry axioms in a quite different context!

Proving mechanism is the essential part of the program EUCLID and it is based upon so called "sentinel-principle". That principle enables proving of all E theorems in a finite number of steps.

5 The Knowledge-pool

All objects and knowledge which are used in prover EUCLID are expressed in the knowledge-pool. In the begining of the proving process for a certain theorem, knowledge-pool contains only datas about objects (denoted by letters) and relations given by theorem itself. During the proving process, all objects and relations inferenced upon module of axioms are being added to knowledge-pool with their unique (natural number) index. For unary predicates (defining objects) this index is their only argument and it is their identifier.

The state of the knowledge-pool is determined by value of the so-called sentinel. The sentinel determines the set of objects from the knowledge-pool that are accesable for certain axioms in a process of proving.

The current state of the knowledge-pool is determined by the proving mechanism. In case of branching (in process of theorem proving) parts of knowledge-pool related to disjunctive branches are independent and this saves integrity of the knowledge-pool as a knowledge base.

6 The Proving Mechanism

For sake of illustration, let us see the key PROLOG predikate of the proving mechanism:

```
proof :- contradiction.
proof :- proved.
proof :- adt(M),proof.
proof :- ax_u(M),proof.
proof :- assumption(P,NotP),
     index(B),
     retract(comments(true)),assert(comments(fail)),
     push(B,0),justified(P,IP),pop(B,0),
     push(B,0),justified(NotP,INotP),pop(B,0),
     retract(comments(fail)),assert(comments(true)),
     ((IP=true,INeP=true,proofp([P,NeP]));
     (IP=true,INeP=false,proofp([P]));
```

```
      (IP=false,INeP=true,proofp([NeP]))).
proof :- ax_b(M).
proof :- ax_p(M),proof.
proof :- sentinel(G),
      first_object(G,N),N1 is N+1,
      retract(sentinel(G)),
      assert(sentinel(N1)),
      proof.
proof :- ax_sp(M),proof.
```

In the very beginning of the program's work, there are to be given assumptions of a certain geometry theorem and its conclusion. Before activating the key PROLOG predicate in the proving mechanism - predicate "proof", knowledge-pool contains only datas about objects (denoted by letters) and relations given by theorem assumptions, and all that objects are accesable for the module of axioms.

The key part of the algorithm can be (unpresicely) defined as follows:

(1) Check if there is a contradiction in the knowledge-pool; if there is, report it and finish proving process in the current branch of a proof; if there is not, go to step (2);

(2) Check if there are enough knowledge in the knowledge-pool to conclude that the theorem is proved; If there are, report it, define objects and relations making conclusion of the given theorem and finish proving process in the current branch of a proof; if there are not, go to step (3);

(3) If possible, apply one of the ADT element and go to step (1); if not, go to step (4);

(4) If possible (according to current state of knowledge-pool and sentinel value), apply one of the unproductive axioms and go to step (1); if not, go to step (5);

(5) If possible, assume that some relation over some objects from current knowledge-pool holds and add this assumption to the knowledge-pool as a fact; similary, assume negation of that relation; make proves for both cases; if it is not possible to assume any ralation over objects from current knowledge-pool, go to step (6);

(6) If possible (according to current state of knowledge-pool and sentinel value), apply one of the branching axioms and make proves for all its branches; if not, go to step (8);

(7) If possible (according to current state of knowledge-pool and sentinel value), apply one of the productive axioms and go to step (1); if not, go to

step (8);

(8) If possible (according to current state of knowledge-pool and sentinel value), apply one of the strongly productive axioms and go to step (1); if not, go to step (9);

(9) Select the object with the least index greater then current value of the sentinel; give the value of this index to the sentinel; go to step (1).

This algorithm can be modified in such a way to prove many theorems more efficiently, but that version of the prover can not prove all $\mathcal{E}$ theorems.

The sentinel has a key role in determining which objects from the knowledge-pool are accesable in certain moment of the proving process. It is a sentinel principle which guaranatee ability of proving all $\mathcal{E}$ theorems.

7 The Sentinel Principle

The sentinel value in each moment of the proving process is determining a set of accesable objects for geometry axioms. In the proving process all elements of ADT module could be applied no matter to the sentinel value (i.e. all objects from the knowledge-pool are accesable for them). Immidiate after entering the assumtions of the theorem which is to be proved, all objects occuring in these assumtions are accesable. During the proving process, the knowldge-pool is spreading (according to foregiven algorithm and by application of the geometry axioms and ADT elements). If step (9) of the algorithm is reached in the proving process, none axiom or ADT element could be applied according to current state of the knowledge-pool and the sentinel value. Then the set of accesable objects, i.e. the sentinel value is to be changed. The sentinel is getting a least index value of all objects which have indexes greater then current sentinel value. It means that the first object whose existency was established since the last change of the sentinel value will be added to the set of accesable objects. Then the proving process is continuing with application of axioms. That is how there is ensured inferencing of all possible conclusions for given set of relations and accesable objects. Also, that is how there is ensured inferencing of all concluslions relevant for the given theorem, and enabled occuring of any "infinite" branch in the proving process.

Let's point out (once again) that forementioned mechanism ensures proving of all $\mathcal{E}$ theorems in a finite number of steps (but, many theorems could not be proved because of the limited recources of the Arity PROLOG). Also, let's point out that it is not a difficult task to extend the program EUCLID in such a way to optimize its finished proves (i.e. to eliminate all unneccesary steps).

8 The EUCLID Axiomatic System of Elementary Geometry

In the axiomatic system of elementary geometry (geometry without axiom of continuity) inspired by the program EUCLID, as the primitive notions we take one fixed set $\mathcal{G}$ (*geometry objects set*) and seven primitive relations over geometry objects: three unary relations *is a point*, (denoted by t), *is a line* (l) and *is a plane* (p), two binary relations *identical* (denoted by *identical*) and *incidental* (i), one ternary relation *between* (denoted by b) and one quaternary relation *congruent* (denoted by c). (Instead of writing *For a such that it holds t(a) ...* we shall write *For point a ....*) We also use negations of these relations (except for congruence - its negation does not occur in any axiom). As a defined relations, we use relations *colliner* and *coplanar* (with their negations) and relation *intersect* (we don't use definition for relation non-intersect, but we use it as a assumtion of a theorem or as a assumption during the proving process). We don't use any set-theory segment. We use "classical" geometry axioms, additional incidence axiom and "identity" axioms (see section 4) and use them according to rules of Gentzen's NK calculus. All axioms are divided into five groups:

-"identity" axioms (see section 4);

-"unproductive" axioms (axioms of the form $\forall$);

-"branching" axioms (axioms of the form $\forall - \exists - \vee$);

-"productive" axioms (axioms of the form $\forall - \exists$);

-"strongly productive" axioms (axioms of the form $\exists$).

(according to foregiven classification additional incidence axiom belongs to group of unproductive axioms)

In the forementioned modified version of the program EUCLID some of the productive axioms was put into the group of stronly productive axioms. That version of the program makes some proves more efficiently, but can't prove all $\mathcal{E}$ theorems.

During the proving proces prover EUCLID is denoting all new geometry objects by natural numbers and practicaly is making a model (or models) of elementary geometry in a set of natural number, or more precisely, part of that model sufficient to prove the given theorem. It means that, indepent of any concrete theorem, algorithm EUCLID could generate a model of elementary geometry in a set of natural numbers in an infinite (but recursive!) process.

Appendix

For the sake of traditional approach, a prover EUCLID gives proves according to traditional sense of relation incidence as a set relation.

Predicates occuring in the program's proves (except unary ones) have one additional argument and that is their unique index (see section 3).

Example 1.

Theorem: Just one plane passes through two intersecting and distinct lines.

***** EUCLID v4.00 - Geometry Theorems Prover *****

Enter list of assumtions:

[l(a),l(b),non_identical(a,b),intersect(a,b)].

There exists a point 3 such that i(3,a) and i(3,b).

Enter conclusion of the theorem:

p(X),e(a,X,_),e(b,X,_),unique(X).

By axiom 1.1, there exists a point 6, distinct from 3, such that 6 i(6,a).
Let us assume i(6,b).
By axiom 1.3, lines a and b are identical.
Conradiction: non_identical(a,b) and identical(a,b)!
Let us assume non_i(6,b).
By axiom 1.1, there exists a point 14, distinct from 3, such that i(6,b).
There must be non_identical(6,14) (oposite assumtion is obviously in contradiction to other assumtions).
We have i(3,b), i(14,b) and non_i(6,b), so points 3, 6 and 14 are non_collinear,
Points 3, 6 and 14 are non_collinear so we have non_i(14,a).
By axiom 1.2, there exists a line 26, such that i(6,26) and i(14,26).
Points 6, 14 and 3 are non_collinear so we have non_i(3,26).
By axiom 1.5, there exists a plane 30, such that i(3,30), i(6,30) and i(14,30).
By axiom 1.7, since i(3,30) and i(6,30), we have i(a,30).
By axiom 1.7, since i(3,30) and i(14,30), we have i(b,30).
Therefore there exists the plane we are seeking.
Let us prove that this is only one such plane.
Let us assume the opposite - there exists one more such plane (36), planes 30 and 36 are non identical and let us prove contradiction.
Since i(3,a) and (a,36), we have i(3,30).

Since i(6,a) and (a,36), we have i(6,30).

Since i(14,b) and (b,36), we have i(14,30).

By axiom 1.7, since i(6,30) and i(14,30), we have i(26,30).

There must be non_identical(a,26) (opposite assumtion is obviously in contradiction to other assumtions).

There must be non_identical(b,26) (opposite assumtion is obviously in contradiction to other assumtions).

* By axiom 1.6, since points 3, 6 and 14 are common for planes 30 and 36 there are identical.

Conradiction: non_identical(30,36) and identical(30,36)!

Therefore p(30),e(a,30,34),e(b,30,35),unique(30), QED.