Formalizing Projective Plane Geometry in Coq

Nicolas Magaud, Julien Narboux, and Pascal Schreck

LSIIT UMR 7005 CNRS-ULP, Université de Strasbourg*

Abstract We investigate how projective plane geometry can be formalized in a proof assistant such as Coq. Such a formalization increases the reliability of textbook proofs whose details and particular cases are often overlooked and left to the reader as exercises. Projective plane geometry is described through two different axiom systems which are formally proved equivalent. Usual properties such as decidability of equality of points (and lines) are then proved in a constructive way. The duality principle as well as formal models of projective plane geometry are then studied and implemented in Coq. Finally, notions of flats and ranks are introduced and their basic properties are proved. This would allow to develop a more algebraic approach to proofs of alignment properties such as Desargues' theorem.

1 Introduction

This paper deals with formalizing projective plane geometry. Projective plane geometry can be described by a fairly simple set of axioms. However it captures the main aspects of plane geometry, especially perspective. It is a good candidate to be formalized in a proof assistant. Most of the description and proofs are available in textbooks such as [8, 4]. However, in most books, many lemmas are considered trivial and many proofs are left to the reader. Building a formal development in a proof assistant allows for more flexibility. If required, axioms can be changed easily and proofs can be rechecked automatically by the system. Such changes may only require minor rewriting of the proofs by the user. In all cases, the proofs are computer-verified, which dramatically increases their reliability compared to paper-and-pencil proofs.

This formalization is not only interesting in itself. It also allows to evaluate the adequacy of a proof assistant such as Coq to develop a formal theory and to build some models of this theory. More significantly, we formalize projective plane geometry because we are interested in building reliable and robust constraint solving programs (see [18, 17]). Indeed, in geometric constraint solving, handling the numerous particular cases is crucial to ensure robustness. Detecting whether a configuration is degenerated or not [28] requires theorem proving: which theorems are required and how to prove them is among the issues we want to address. As shown in [21], point-line incidences in the projective plane are sufficient to express usual geometric contraints.

* This work is partially supported by the ANR project Galapagos.

Finally, as computer scientists, we are interested in the effectiveness of proofs in order to extract programs from these proofs. The Coq proof assistant [7, 1] implements a constructive logic and allows program extraction from constructive proofs. Therefore, it is the perfect tool to carry out a constructive formalization.

Related work Proof assistants have already been used in the context of geometry. The task consisting in mechanizing Hilbert's *Grundlagen der Geometrie* has been partially achieved. A first formalization using the Coq proof assistant was proposed by Christophe Dehlinger, Jean-François Dufourd and Pascal Schreck [10]. This first approach was realized in an intuitionist setting, and concluded that the decidability of point equality and collinearity is necessary to check Hilbert's proofs. Another formalization using the Isabelle/Isar proof assistant [26] was performed by Jacques Fleuriot and Laura Meikle [19]. Both formalizations have concluded that, even if Hilbert has done some pioneering work about formal systems, his proofs are in fact not fully formal, in particular degenerated cases are often implicit in the presentation of Hilbert. The proofs can be made more rigorous by machine assistance. Frédérique Guilhot realized a large Coq development about Euclidean geometry following a presentation suitable for use in french high-school [15]. In [24, 25], Julien Narboux presented the formalization and implementation in the Coq proof assistant of the area decision procedure of Chou, Gao and Zhang [5] and a formalization of foundations of Euclidean geometry based on Tarski's axiom system [32, 29]. In [12], Jean Duprat proposes the formalization in Coq of an axiom system for compass and ruler geometry.

Regarding formal proofs of algorithms in the field of computational geometry, we can cite David Pichardie and Yves Bertot [27] for their formalization of convex hulls algorithms in Coq as well as Laura Meikle and Jacques Fleuriot [20] for theirs in a Hoare-like framework in Isabelle. In addition, Jean-François Dufourd recently formalized an image segmentation algorithm [11] in Coq.

Notations Naming/Writing conventions follow the guidelines edited in a recent document proposed by Duprat, Guilhot and Narboux [13]. Most Coq notations, which are really close to mathematical ones, will be explained along the course of the paper. The negation is noted `~`. The most awkward notation for the reader not accustomed to Coq is the curly-brackets notation for constructive existential quantification over the sort `Type`. For instance, the formula `forall l:Line, {X:Point | ~Incid X l}` expresses that $\forall l : Line, \exists X : Point, \neg\ Incid\ X\ l$.

Outline The paper is organized as follows. In section 2, we present the axioms for projective plane geometry and their description in the Coq proof assistant. In section 3, we study the duality between points and lines. Section 4 deals with finite and infinite models for projective plane geometry. Section 5 introduces the notion of flats and a characterization of flats with respect to the usual notions of point and line.

Axiom Line Existence

$$\forall A\ B : Point, (\exists l : Line, A \in l \wedge B \in l)$$

Axiom Point Existence

$$\forall l\ m : Line, (\exists A : Point, A \in l \wedge A \in m)$$

Axiom Line Uniqueness

$$\forall A\ B : Point, A \neq B \Rightarrow \forall l\ m : Line, A \in l \wedge B \in l \wedge A \in m \wedge B \in m \Rightarrow l = m$$

Axiom Point Uniqueness

$$\forall l\ m : Line, l \neq m \Rightarrow \forall A\ B : Point, A \in l \wedge A \in m \wedge B \in l \wedge B \in m \Rightarrow A = B$$

Definition (distinct4)

$$distinct4\ A\ B\ C\ D \equiv A \neq B\ \wedge\ A \neq C\ \wedge\ A \neq D\ \wedge\ B \neq C\ \wedge\ B \neq D\ \wedge\ C \neq D$$

Axiom Four Points

$$\begin{array}{l}
\exists A : Point, \exists B : Point, \exists C : Point, \exists D : Point, \\
distinct4\ A\ B\ C\ D\ \wedge \\
(\forall l : Line, (A \in l \wedge B \in l \Rightarrow C \notin l \wedge D \notin l) \wedge \\
(A \in l \wedge C \in l \Rightarrow B \notin l \wedge D \notin l) \wedge \\
(A \in l \wedge D \in l \Rightarrow B \notin l \wedge C \notin l) \wedge \\
(B \in l \wedge C \in l \Rightarrow A \notin l \wedge D \notin l) \wedge \\
(B \in l \wedge D \in l \Rightarrow A \notin l \wedge C \notin l) \wedge \\
(C \in l \wedge D \in l \Rightarrow A \in l \wedge B \in l))
\end{array}$$

Figure 1. A first axiomatization of projective plane geometry.

2 Axioms

2.1 A First Set of Axioms

We assume that we have two kinds of objects which we call points and lines. We also assume that we have a relation ($\in$) between elements of these two sets. We can describe projective plane geometry using the axioms presented in figure 1. The first two axioms deal with existence of points and lines. We choose not to require points (resp. lines) to be distinct in axiom 'Line Existence' (resp. 'Point Existence'). If the points (resp. lines) are equal, the line (resp. the point) still exists: actually there exists an infinity of lines (resp. points). This design choice follows a general rule in formal geometry: it is crucial to consider statements which are as general as possible.

The next two axioms deal with uniqueness of the above defined line and point. These axioms hold only if the two points (resp. lines) are distinct. As suggested in [2], axioms 'Point Uniqueness' and 'Line Uniqueness' can be merged into a more convenient axiom with no negation. This axiom is classically equivalent to the conjunction of the two others:

Axiom Uniqueness

$$\begin{array}{l}\forall A\ B : Point, \forall l\ m : Line, \\ A \in l \Rightarrow B \in l \Rightarrow A \in m \Rightarrow B \in m \Rightarrow A = B \vee l = m\end{array}$$

Finally, axiom 'Four points' states that there exists at least four distinct points, no three of them being collinear. This means dimension is at least 2. Together with axiom 'Point Existence' which expresses that the dimension is at most 2 (two lines always intersect), it imposes that the dimension of this projective space is exactly 2. The formalization of this axiom system is straightforward, but from a practical point of view, proofs in most textbooks often use some variants of this system. To ease mechanization of proofs, we formalize the equivalence between these systems.

2.2 Another Axiom System for Projective Plane Geometry

Another non-degeneracy Axiom Axiom 'Four Points' states a non-degeneracy condition, namely that the projective space we consider is not reduced to a single line. This can be expressed in another way through two new axioms:

Axiom Three Points

$$\forall l : Line, \exists ABC : Point, (A \neq B \wedge B \neq C \wedge A \neq C) \wedge A \in l \wedge B \in l \wedge C \in l$$

Axiom Lower Dimension

$$\exists l_1 : Line, \exists l_2 : Line, l_1 \neq l_2$$

The first axiom expresses that each line contains at least three points; the second one states that there exist two distinct lines.

We prove that axiom 'Four points' can be replaced by axiom 'Three points' and axiom 'Lower dimension' in the system defined in the previous section and vice-versa. Both settings share the following axioms: Line Existence, Point Existence, Uniqueness. In mathematics textbooks, the equivalence of these two sets of axioms is usually presented as a remark. For instance in [4], the proof is left to the reader. In a proof assistant such as Coq, these proofs have to be made explicit and proving them formally requires some technical work mostly related to handling the numerous configurations of points. The basic principles of the proof are presented in appendix A.

2.3 Implementation in Coq

We formalize the previous definitions in the Coq proof assistant [1, 7]. To do so, we take advantage of the modules and functors of Coq. Modules [6] allow to define parameterized theory and to put together types and definitions into a module structure. It enhances the reusability of developments, by providing a formal interface for such a structure. In addition, functors can be used to connect module types to one another.

Modules and Projective Plane Our first module `PreProjectivePlane` contains axioms dealing with point (resp. line) existence and uniqueness. From that we derive some basic properties, including uniqueness of a line (resp. of a point), from the general uniqueness axiom. Then, on top of `PreProjectivePlane`, we build two modules `ProjectivePlane` which contains axiom 'Four points' and `ProjectivePlane'` which contains axiom 'Three points' and axiom 'Lower dimension'. A theory is of type `ProjectivePlane` if it contains all the notions presented in figure 2 on the following page. The two module types `ProjectivePlane` and `ProjectivePlane'` are connected through two functors `Back` and `Forth` which prove the equivalence of these two axiom systems when the axioms 'Line Existence' and 'Point Existence' as well as Uniqueness are available. Figure 2 sums up the module type for projective plane geometry and figure 3 on page 7 presents the global organization of the development.

Deciding Equality From the assumption that incidence is decidable,

$$\forall A : Point, \forall l : Line, (A \in l \vee \neg A \in l)$$

we prove that point (resp. line) equality is decidable. The proofs of decidability for point (resp. line) equality are intuitionist, in the sense that they do not use the excluded middle property. Details of these proofs are available in appendix B.

From these basic axioms, we can consider proving some theorems about projective plane geometry. For instance, we prove that if we consider lines as set of points, there always exists a bijection between two lines (see appendix E for details). In order to improve genericity, we show that the well-known principle of duality between point and line can be derived in Coq. It allows us to prove automatically half of the theorems of interest from the proofs of their dual counterparts.

3 Duality

3.1 Principle of duality

It is well known that projective geometry enjoys a principle of duality, namely that every definition remains significant and every theorem remains true, when we interchange the words *point* and *line*. To formalize this principle in Coq, we make use of the module system of Coq [1, 6]. In practice, we consider the module type (`ProjectivePlane'`) defined in the previous section and we build a functor from `ProjectivePlane'` to itself in which we map points to lines and lines to points:

```
Module swap (M: ProjectivePlane') : ProjectivePlane'.
Definition Point := M.Line.
Definition Line  := M.Point.
Definition Incid := fun (x:Point) (y:Line) => M.Incid y x.
...
```

```
Module Type ProjectivePlane.

Parameter Point: Set.
Parameter Line: Set.
Parameter Incid : Point -> Line -> Prop.

Axiom incid_dec : forall (A:Point)(l:Line), {Incid A l} + {~Incid A l}.

(* Line Existence : any two points lie on a unique Line *)

Axiom a1_exist : forall (A B :Point), {l:Line | Incid A l /\ Incid B l}.

(* Point Existence : any two lines meet in a unique point *)
Axiom a2_exist : forall (l1 l2:Line), {A:Point | Incid A l1 /\ Incid A l2}.

Axiom uniqueness : forall A B :Point, forall l m : Line,
  Incid A l -> Incid B l  -> Incid A m -> Incid B m -> A=B \/ l=m.


(* Four points : there exist four points with no three collinear *)
Axiom a3: {A:Point & {B :Point & {C:Point & {D :Point |
  (forall l :Line, distinct4 A B C D/\
       (Incid A l /\ Incid B l -> ~Incid C l /\ ~Incid D l)
    /\ (Incid A l /\ Incid C l -> ~Incid B l /\ ~Incid D l)
    /\ (Incid A l /\ Incid D l -> ~Incid C l /\ ~Incid B l)
    /\ (Incid C l /\ Incid B l -> ~Incid A l /\ ~Incid D l)
    /\ (Incid D l /\ Incid B l -> ~Incid C l /\ ~Incid A l)
    /\ (Incid C l /\ Incid D l -> ~Incid B l /\ ~Incid A l))}}}}.

End ProjectivePlane.
```

Figure 2. The module type with axioms required to describe a projective plane. The incidence relation ($\in$) is noted `Incid` in our Coq development. `{Incid A l} + {~Incid A l}` expresses that we know *constructively* that $A \in l \vee \neg A \in l$.

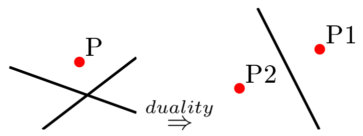
To build this functor we need to show that the dual of each axiom holds. It is clear that the axioms of existence and uniqueness of lines are the dual of the axioms for existence and uniqueness of points:

```
Definition a1_exist  := M.a2_exist.
Definition a1_unique := M.a2_unique.
Definition a2_exist  := M.a1_exist.
Definition a2_unique := M.a1_unique.
```

To prove the dual version of axiom 'Three points' and axiom 'Lower dimension' it is necessary to use the other axioms. Appendix C provides the detailed proof of the fact that incidence geometry is a dual of itself and Figure 3 a summary of the organization of the development.

3.2 Applications

Formalizing this principle of duality leads to an interesting theoretical result. In addition, this principle is also useful in practice. For every theorem we prove, we can easily derive its dual using our functor `swap`. For instance, from the lemma `outsider` stating that for every couple of lines, there is a point which is not on these lines, we can derive its dual automatically: for every couple of points, there is a line not going through these points.




```
Module Example (M': ProjectivePlane').

Module Swaped := swap M'.
Export M'.

Module Back := back.back Swaped.
Module ProjectivePlaneFacts_m := decidability.decidability Back.

Lemma dual_example :
forall P1 P2 : Point,{l : Line | ~ Incid P1 l /\ ~ Incid P2 l}.
Proof.
apply ProjectivePlaneFacts_m.outsider.
Qed.

End Example.
```

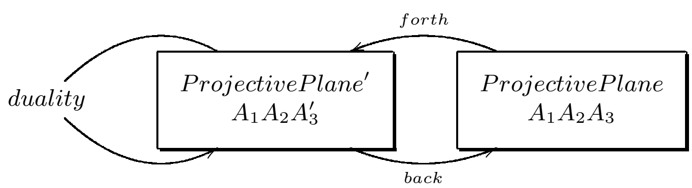


Figure 3. A modular organization. Arrows represent functors and boxes represent modules types.

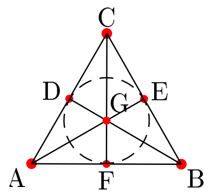
So far, we focused on axiom systems and formal proofs. The next step is to check whether well-known models verify our axioms for projective plane geometry.

4 Models

In order to prove formally that our sets of axioms are consistent, we build some models. We build both finite and infinite models: among them the smallest projective plane and an infinite model based on homogeneous coordinates.

4.1 Finite Models

Following Coxeter's notation [8], a finite projective geometry is written $PG(a, b)$ where a is the number of dimensions, and given a point on a line, b is the number of other lines through the point. We build two finite models: $PG(2, 2)$ and $PG(2, 5)$. $PG(2, 2)$ is the smallest projective plane and is also known as Fano's plane.



Fano's plane In two dimensions, we can easily build the model with the least number of points and lines: 7 each. This model is called Fano's plane. On the figure, points are simply represented by points, whereas lines are represented by six segments and a circle (DEF).

We define a module `FanoPlane` of type `ProjectivePlane`. The typing system of Coq will ensure that our definitions are really instances of the abstract definition of a projective plane.

The set of points is defined by an inductive[1] type with 7 constructors and the set of lines as well:

```
Inductive ind_point : Set := A | B | C | D | E | F | G.
Inductive ind_line  : Set := ABF | BCD | CAE | ADG | BEG | CFG | DEF.

Definition point : Set := ind_point.
Definition line  : Set := ind_line.
```

The incidence relation is given explicitly by its graph:

```
Definition incid_bool : Point -> Line -> bool := fun P L =>
 match (P,L) with
 (A,ABF) | (A,CAE) | (A,ADG) | (B,BCD) | (B,BEG) | (B,ABF)
|(C,BCD) | (C,CAE) | (C,CFG) | (D,BCD) | (D,ADG) | (D,DEF)
|(E,CAE) | (E,BEG) | (E,DEF) | (F,ABF) | (F,DEF) | (F,CFG)
|(G,ADG) | (G,CFG) | (G,BEG)  => true
| _ => false
end.
```

The proofs of existence and uniqueness are performed by case analysis. Note that in order to prove the axioms of uniqueness, we must prove that for every couple of points (resp. lines) there is a unique line (resp. point). This creates $7^2 = 49$ cases. For each of these cases, we have to perform a case analysis on the lines, this produces again 49 cases, leading to a total of 2401 cases. The proof is computed easily by Coq.

PG(2,5) We follow [8] and build another model of the projective plane which is still finite but larger than Fano's plane. This model is called $PG(2, 5)$. It contains 31 points and as many lines. The incidence relation is given on table 1

[1] Note that this type is not really inductive, but sum types are defined in Coq using a special case of the general concept of inductive types.

in appendix D. From the technical view of the formalization, this model is harder to build than Fano's plane because the proof produces 923 521 cases[2]. However, the proofs of these cases can be automated. The total size of the proof object generated by Coq (a term of the calculus of inductive constructions) is 7 Mo.

4.2 Infinite Model: Homogeneous Coordinates

To build an infinite model of projective geometry we use homogeneous coordinates introduced by August Ferdinand Möbius. We present our formalization in the context of the projective plane, but it can be easily generalized to any other dimension. The homogeneous coordinates of a point (resp. of a line) of a projective plane is a triple of numbers which are not all zero. These numbers are elements of any commutative field of characteristic different from two. Two triples which are proportional are considered as equal: for any $\lambda \neq 0$, $(x_1, x_2, x_3) = (\lambda x_1, \lambda x_2, \lambda x_3)$.

To formalize this notion in Coq it would be natural to define pseudo-points as triple of elements of a field and then define points (resp. lines) as the equivalence classes of proportional non-zero triple in this field. Unfortunately, defining a type by quotient is something difficult to do in the calculus of inductive constructions used by Coq [30]. Therefore, we choose to define the quotient type directly by representing the classes of points and lines by a normal form. Points and lines are represented by their triple of coordinates such that the last non zero coordinate is 1. Consider a point (x_1, x_2, x_3). If $x_3 \neq 0$ we can represent it by $(x_1/x_3, x_2/x_3, 1)$. If $x_3 = 0$, we perform case distinction on x_2. If $x_2 \neq 0$ we can represent it by $(x_1/x_2, 1, 0)$, else we represent it by $(1, 0, 0)$. This definition can be formalized in Coq using the following inductive type where `F` is the type of the elements of our field and `P0`, `P1` and `P2` are the constructors for the three different cases:

```
Inductive Point : Set :=
 | P2 : F -> F -> Point (* (x1,x2,1) *)
 | P1 : F -> Point      (* (x1,1 ,0) *)
 | P0 : Point.          (* (1 ,0 ,0) *)
```

The second and third constructors correspond to ideal points (points at infinity).

The incidence relation (noted `Incid` in Coq and $\in$ in this paper) can then be defined as the inner product of a point and line. The definition of the inner product can be made more generic by using triples, instead of giving a definition distinguishing each of the $3 * 3$ cases.

[2] In [8], the proof given is the following: "we observe that any two residues are found together in just one column of the table (see on page 18), and that any two columns contain just one common number". This amounts to checking, for more than 400 different configurations, whether two sets of six elements have only one common element. In such a case, mechanized theorem proving is the best way to ensure correctness.

To do this, we define two functions, one to transform a point into a triple of coordinates, and another one to normalize a triple of coordinates to obtain a point. We can then prove two lemmas which state that our definitions are consistent:

```
Lemma triple_point :
  forall P : Point, point_of_triple (triple_of_point P) = P.

Lemma point_triple :
  forall a b c : F, (a,b,c) <> (0,0,0) ->
  exists l, triple_of_point (point_of_triple (a,b,c)) = (a*l,b*l,c*l).

Lemma point_of_triple_functionnal:
  forall a b c l : F, (a,b,c) <> (0,0,0) -> l <> 0 ->
  point_of_triple(a,b,c) = point_of_triple(a*l,b*l,c*l).
```

The inner product and incidence relations can then be defined as:

```
Definition inner_product_triple A B :=
match (A,B) with
  ((a,b,c),(d,e,f)) => a*d+b*e+c*f
end.

Definition Incid : Point -> Line -> Prop := fun P L =>
 inner_product_triple (triple_of_point P) (triple_of_line L) = 0.
```

Now, we need to prove that the axioms of a projective plane hold in this setting. The proof of the decidability of `Incid` and of axioms (Three Points) and (Lower Dimension) are straightforward. For the uniqueness axiom, after unfolding of definitions, the problem reduces to a goal involving equations such as the following ones:

$$\begin{array}{r} r * r_5 + r_0 * r_6 + 1 = 0 \\ r * r_3 + r_0 * r_4 + 1 = 0 \\ r_1 * r_5 + r_2 * r_6 + 1 = 0 \\ r_1 * r_3 + r_2 * r_4 + 1 = 0 \end{array} \Rightarrow (r = r_1 \wedge r_0 = r_2) \vee (r_3 = r_5 \wedge r_4 = r_6)$$

Using the following equivalences considered as rewrite rules, we can convert this goal into an ideal-membership problem which can be solved using the Gröbner basis tactic developed by Loïc Pottier [9].

$$\begin{array}{lcl} \forall ab, \quad a = b & \Leftrightarrow & a - b = 0 \\ \forall ab, \ (a = 0 \vee b = 0) & \Leftrightarrow & ab = 0 \\ \forall ab, \ (a = 0 \wedge b = 0) & \Leftrightarrow & a^2 + b^2 = 0 \end{array}$$

For the existence axioms, we need to define the line passing through two points (resp. the point at the intersection of two lines).

Proofs of these lemmas illustrate how combining *automated* and *interactive* theorem proving can be successful. However, most proofs of theorems in projective geometry still heavily require user-interaction and lack automation. The amount of case distinctions required in formal proofs make them difficult to handle. In the next section, we formalize flats and ranks which allow to express in a simpler way non-degeneracy assumptions.

5 Flats

Degenerated cases are a central issue in the formalization and automation of geometry. In this section, we will introduce the concepts of flats and rank which provide a nice abstraction to express generically degenerated cases.

There are two different approaches to prove incidence properties in projective plane geometry. One may simply consider our axioms, state that a given point A belongs to a given line l and use all the available theorems to prove that A actually belongs to l. Another (more algebraic) approach relies on the notions of flats and rank as presented for example in the review by Bonin [3].

In a word, the notion of rank allows to distinguish between equal/non-equal points as well as collinear/non-collinear points. As an example, $rk(A, B) = 1$ means A and B are equal. If $rk(A, B) = 2$, this means, A and B are distinct. This generalizes to collinear/non-collinear points: if $rk(A, B, C) = 2$ this means A, B and C are collinear (with at least two points distinct from one another); if $rk(A, B, C) = 3$, this means A, B and C are not collinear and all distinct. In this section, we define flats and their characterization in a two-dimensional setting. This represents the first step towards defining ranks and using them to prove some collinearity properties in a projective geometry.

5.1 Definitions

Definition (Set of points) A set of points E is represented by a predicate $E : Point \rightarrow Prop$ which, given a point P, will return `True` if P belongs to the set, `False` otherwise.

This corresponds to the well-known concept of characteristic function. This view of sets has two main advantages: it remains close to the mathematical intuition and it can be conveniently implemented in Coq. Basic operations on sets (union, intersection, etc.) are easily defined as higher-order functions.

In this setting, the empty set is denoted by `fun (x:Point) => False`, the whole plane by `fun (x:Point) => True`. The singleton {x} is denoted by

```
fun (y:Point) => if eq_point_dec x y then True else False.
```

Finally, the line l is denoted by

```
fun (x:Point) => if incid_dec x l then True else False.
```

Definition (Flat) A flat is a set of points such that the entire line defined by two points A and B lies in the flat whenever A and B belong to it.

Formally, it can be defined as follows:

```
Definition flat (v:pset) : Prop :=
  forall A B:Point, v A -> v B -> A<>B ->
    forall l:Line, Incid A l -> Incid B l ->
      forall C:Point, Incid C l -> v C.
```

5.2 Characterization

In order to define the notion of rank, we need to characterize flats. We first show that the empty set, singletons such as $\{x\}$, lines and the whole plane are actually flats. Then we prove that in a two-dimensional setting, they are the only sets of points which are flats.

```
Lemma characterization : forall v:pset, flat v ->
(forall x:Point, (v x)<->(pempty x))
\/ (exists y:Point, forall x:Point, ((v x) <-> ((psingleton y) x)))
\/ (exists l:Line,  forall x:Point, ((v x) <-> ((pline l) x)))
\/ (forall x:Point, (v x) <-> (pplane x)).
```

The proof proceeds as follows:

Let us consider a set E which verifies the flat property: there are two cases to consider: the set E has either 0 or at least 1 element. If it has 0 element, it is the empty set. If not, either it has exactly 1 element or at least 2 elements. If it has 1 element, it has to be a singleton set. Then, once again, if it has at least 2 elements, then it either has exactly 2 or at least 3 elements. If it has 2 elements depending on whether they are distinct or not, we find that E is either a singleton or a line. If it has 2 or more, it depends whether they are all on the same line (then it is a line) or if at least one of them is outside the line, in this case this is the whole plane.

Discussion about the implementation of sets The above-mentioned proof simply relies on the following proposition:

$$\forall V\, W,\, V \subseteq W \vee (\exists p, p \in V \wedge p \notin W)$$

It is immediate to prove such a lemma in a classic setting. In addition, it can be proved correct in an intuitionistic way if we consider a concrete representation of finite sets such as the one used in ssreflect [14].

6 Conclusion and future work

In this paper, we have shown how projective plane geometry can be formalized in Coq using two different axiom systems. We proved them equivalent. We then managed to mechanize the duality principle and to build finite and infinite models. Using Coq helped us produce more precise proofs which handle all cases whereas in textbooks some very particular cases can sometimes be overlooked. Finally, we proposed a concise way to represent potentially degenerated incidence relations thanks to the concept of flats and ranks. Overall our Coq development of projective plane geometry amounts to 5K lines with about 200 definitions and lemmas.

Our development makes use of a rather strong axiom, namely decidability of the incidence predicate. All subsequent properties are derived in an intuitionnistic logic from this axiom and those of projective plane geometry. It would be also interesting to perform our formalization using a purely constructive system of axioms as the ones proposed by Heyting and von Plato [16, 33]. These systems of axioms are based on the apartness predicate which is the negation of the incidence predicate. It is easy to prove using classical logic that the axioms of a projective plane implies the axioms of Heyting. It would be also interesting to derive more theorems in a purely constructive framework.

In the future, we plan to carry on our investigations in two main directions. On the one hand, we expect to write a reliable algorithm for constraint solving in incidence geometry. It requires to specify projective plane geometry, which is what we achieve in this paper. The next step will be to certify that whenever the prover says three points are collinear (resp. non-collinear), we can build a proof at the specification level that these points are actually collinear (resp. non-collinear).
On the other hand, we want to consider arbitrary projective spaces by relaxing the dimension constraint. This would require changing our axiom system. We will study how to use ranks and the matroid structure over ranks as presented in [22] to carry out a formal proof of Desargues' theorem. We expect these notions, which are related to the concept of flats we presented in the previous section, to make the proof more generic and more concise through a better handling of degenerated cases.

On the technical side, we also plan to study how our development can make use of first-order type classes [31] instead of modules and functors[3]. We expect this new feature to improve the readability of the formal description by making implicit some technical details.

Availability The whole development is freely available on the web site from the Galapagos research project `http://galapagos.gforge.inria.fr/`.

Acknowledgments We would like to thank Loïc Pottier for providing the Coq tactic for solving systems of equations using Gröbner basis. We also thank Simon E.B. Thierry for proof-reading early drafts of this paper.

[3] The next version of Coq (to be released in the next weeks) will include this feature.

A Equivalence of axiom systems

A.1 From axiom 'Four points' to axiom 'Three points' and axiom 'Lower dimension'

We first prove that each line contains at least three points.

$$(\forall l : Line, \exists ABC : Point, distinct3\ A\ B\ C \wedge A \in l \wedge B \in l \wedge C \in l)$$

We have as an assumption that there exists four points A, B, C and D with no three collinear. We have three cases to study depending on how many points are on line l: either two, one or zero points of these four points are on l.

- Two points are on l (say P and Q), two are not on l (say R and S).
  We build m which goes through R and S, it intersects l on a point (say X) which is different from P and Q. X has to be distinct from P (resp. Q), otherwise we would have P, R and S collinear (resp. Q, R and S collinear).
- One point is on l (say A) , the three other ones are not on l (say B, C and D).
  We have to build two more points on l. We proceed by creating lines going through points outside of l. We have to distinguish cases in order to avoid alignment issues.
- No point is on l, all four points (say A, B, C and D) are outside of l .
  We have to build three distinct points. We do the same reasoning steps, building lines from A, B, C and D.

All possible configuration for the 4 points can be captured by these three cases, sometimes via renaming of points. Details can be found in the formal Coq development.

Axiom 'Lower dimension' can be proved very easily:

$$\exists l_1 : Line, \exists l_2 : Line, l_1 \neq l_2$$

We simply consider 2 lines l (which goes through A and B) and m (which goes through C and D). It is straightforward to show they are different: if they were not, then A, B, C and D would be collinear and this would contradict axiom 'Four points'.

A.2 From axioms 'Three points' and 'Lower dimension' to axiom 'Four points'

We prove some preliminary lemmas: for any two distinct lines l_1 and l_2, each of them carrying at least three points (say M, N and O for l_1 and P, Q and R for l_2), we make a case distinction depending on where these points lie with respect to the intersection of l_1 and l_2. There are four cases to consider:

- One of the three points of l_1 (say M) and one of those of l_2 (say P, which is actually equal to M) are at the intersection of l_1 and l_2. Then the remaining points (M, O, Q and R) verify axiom 'Four points'. No three of them can be collinear otherwise we would have $l_1 = l_2$.

– One point of l_1 (say M) is at the intersection of l_1 and l_2. Then points M, O, Q and R verify axiom 'Four points'.
– One point of l_2 (say P) is at the intersection of l_1 and l_2. Exchanging l_1 and l_2 in the previous lemma solves the case.
– No point of l_1 and l_2 is at the intersection. Then points M, O, Q and R also verify axiom 'Four points'.

Axiom 'Four points' is then proven by first making two lines l_1 and l_2 explicit (through axiom 'Lower dimension'), then considering three distinct points on each line (through axiom 'Three points'). The four lemmas allow to prove the existence of four points in the various possible configurations depending on which points (if any) lie at the intersection of l_1 and l_2.

B Decidability proofs

From the axiom system ProjectivePlane (see figure 2 on page 6) and a decidability axiom about incidence, namely

$$\forall A : Point, l : Line, Incid\ A\ l \vee \neg Incid\ A\ l$$

we can derive proofs of decidability of point equality as well as line equality. Both theorems can be proven independently, in a intuitionistic way (none of them require the use of classic logic).

B.1 Line Equality

Given any two lines l_1 and l_2, they either are equal or not.

$$\forall l_1\ l_2 : Line, l_1 = l_2 \vee l_1 \neq l_2$$

From axiom 'Three points', we know there exists three distinct points M, N and P on l_1. We then proceed by case analysis depending on whether M and N are on l_2.

$M \in l_2$:
  $N \in l_2$: $l_1 = l_2$ because of the Uniqueness Axiom and the fact that $M \neq N$.
  $N \notin l_2$: l_1 and l_2 are different because N is on l_1 and not on l_2.
$M \notin l_2$: l_1 and l_2 are different because M is on l_1 and not on l_2.

B.2 Point Equality

Given any two points A and B, they either are equal or not.

$$\forall A\ B : Point, A = B \vee A \neq B$$

We first prove an auxiliary lemma

$$\forall A\ B : Point, \forall d : Line, A \notin d \Rightarrow B \notin d \Rightarrow A = B \vee A \neq B$$

From axiom 'Three points', we know there exists three distinct points M, N and P incident to d. We build two lines $l_1 = (AM)$ and $l_2 = (AN)$. These two lines are different because N and M are distinct and A is not incident to d.

$B \in l_1$:

  $B \in l_2$: $A = B$ from the Uniqueness Axiom and the fact that $l_1 \neq l_2$.

  $B \notin l_2$: $A \neq B$ because A is incident to l_2 and B is not.

$B \notin l_1$: $A \neq B$ because A is incident to l_1 and B is not.

The main theorem can now be proved: from axiom 'Lower dimension', there exists two distinct lines Δ_0 and Δ_1. We proceed by case analysis on whether A and B belong to Δ_0 and Δ_1.

$A \in \Delta_0$:

  $B \in \Delta_0$:

    $A \in \Delta_1$:

      $B \in \Delta_1$: $A = B$ from the Uniqueness Axiom and the fact that $\Delta_0 \neq \Delta_1$.

      $B \notin \Delta_1$: $A \neq B$, because A is incident to Δ_1 and B is not.

    $A \notin \Delta_1$:

      $B \in \Delta_1$: $A \neq B$, because A is not incident to Δ_1 and B is.

      $B \notin \Delta_1$: We apply the previous lemma with $d = \Delta_1$.

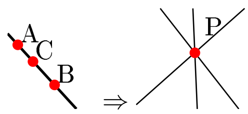
  $B \notin \Delta_0$: $A \neq B$, because A is incident to Δ_0 and B is not.

$A \notin \Delta_0$:

  $B \in \Delta_0$: $A \neq B$, because A is not incident to Δ_0 and B is.

  $B \notin \Delta_0$: We apply the previous lemma with $d = \Delta_0$.

C Duality



As stated before the proof of most axioms is straightforward, hence we only prove the dual of axiom 'Three points'. We need to prove that:

$$\forall P, \exists l_1 l_2 l_3, P \in l_1 \wedge P \in l_2 \wedge P \in l_3$$

First, we prove the following two lemmas:

$$\forall P l, P \notin l \Rightarrow \exists l_1 l_2 l_3, P \in l_1 \wedge P \in l_2 \wedge P \in l_3$$

and

$$\forall l_1 l_2, \exists P, P \notin l_1 \wedge P \notin l_2$$

Proof of the first lemma: let's take three distinct points A, B and C on l using axiom 'Three points'. Then we can build the lines (PA), (PB) and (PC). Those lines are distinct because otherwise using the uniqueness axiom we could prove that A,B and C are not distinct.

Proof of the second lemma: If $l_1 = l_2$ we need to build a point not on l_1. From axiom 'Lower dimension', we know there are two lines. From axiom 'Three points' we can conclude because we know there are at least three points on each line.

Otherwise $l_1 \neq l_2$. Let's call C the intersection of l_1 and l_2. Then, we can build two points P_1 and P_2 on l_1 and l_2 respectively which are different from C. We know that $P_1 \neq P_2$ because otherwise $l_1 = l_2$. Let d be the line through P_1 and P_2. We can build a third point Q on d. Q is neither on l_1 nor on l_2. This concludes the lemma.

Finally, we can prove the dual of axiom 'Three points'. We build two lines l_1 and l_2 using axiom 'Lower dimension'. Then we perform case distinction on $P \in l_1$ and $P \in l_2$. If $P \in l_1 \wedge P \in l_2$ we use the second lemma. Otherwise $P \notin l_1 \vee P \notin l_2$. In both cases, we can use the first lemma.

E Lines as set of points

In our development, we consider two basic notions: points and lines. Lines can actually be viewed as sets of points. With this representation, for any lines l_1 and l_2 we can build a bijection from l_1 to l_2.

We first define the set of points corresponding to a given line l, it consists of all the points of the plane which are incident to l.

```
Definition line_as_set_of_points (l:Line):= {X:Point | Incid X l}.
```

From this definition, we want to prove the following theorem:

```
Theorem line_set_of_points : forall l1 l2:line,
exists f:(line_as_set_of_points l1) -> (line_as_set_of_points l2),
          bijective f.
```

It states there exists a bijective function f from l_1 to l_2 when these lines are viewed as sets of points. We build a constructive proof of this existential formula, which requires to make explicit the function f and then check whether it is actually a bijection, i.e. verifies the one-to-one and onto properties.
The proof proceeds as follows:
First of all, one can safely assume that l_1 and l_2 are different. If not, then the identity function works just fine. The first step of the proof is to write a function which, given two lines l_1 and l_2 computes a point P which belongs neither to l_1 nor to l_2.

```
Lemma outsider : forall l1 l2: Line,
                 {P:Point | ~Incid P l1/\~Incid P l2}.
```

We now explicitly construct the function f as shown on figure 4. Given a point A_1 of l_1, we can build a line (say Δ) going through A_1 and P. Lines Δ and l_2 intersect in a point A_2. We define f such that $f(A_1) = A_2$. It remains to prove that this function is actually bijective. Proving that this function is one-to-one requires to assume proof irrelevance [23]. Proof irrelevance expresses that proofs of the same formula are equal. It allows us to show existential propositions with the same type are equal regardless of the proof terms proving the formulas. Proving the onto property requires to apply the construction process of f in the reverse order going from line l_2 to line l_1.

D PG(2,5)

P_{30}	P_{29}	P_{28}	P_{27}	P_{26}	P_{25}	P_{24}	P_{23}	P_{22}	P_{21}	P_{20}	P_{19}	P_{18}	P_{17}	P_{16}	P_{15}	P_{14}	P_{13}	P_{12}	P_{11}	P_{10}	P_9	P_8	P_7	P_6	P_5	P_4	P_3	P_2	P_1	P_0
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	0
2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	0	1
4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	0	1	2	3
9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	0	1	2	3	4	5	6	7	8
13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	0	1	2	3	4	5	6	7	8	9	10	11	12
19	20	21	22	23	24	25	26	27	28	29	30	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18

Table 1. The incidence relation of PG(2,5). Each column lists the lines incident to the given point.

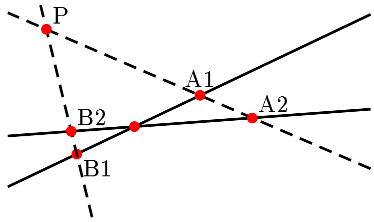


Figure 4. Building a bijection between l_1 and l_2.

References

[1] Yves Bertot and Pierre Castéran. *Interactive Theorem Proving and Program Development, Coq'Art: The Calculus of Inductive Constructions*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2004.

[2] Marc Bezem and Dimitri Hendriks. On the Mechanization of the Proof of Hessenberg's Theorem in Coherent Logic. *Journal of Automated Reasoning*, 40(1):61–85, 2008.

[3] Joseph E. Bonin. A Brief Introduction to Matroid Theory, 2001. available from http://home.gwu.edu/˜jbonin/survey.ps.

[4] Francis Buekenhout, editor. *Handbook of Incidence Geometry*. North Holland, 1995.

[5] Shang-Ching Chou, Xiao-Shan Gao, and Jing-Zhong Zhang. *Machine Proofs in Geometry*. Series on Applied Mathematics. World Scientific, 1994.

[6] Jacek Chrzaszcz. Implementing Modules in the Coq System. In *Theorem Proving in Higher-Order Logics*, volume 2758 of *Lecture Notes in Computer Science*, pages 270–286. Springer-Verlag, 2003.

[7] Coq development team, The. *The Coq Proof Assistant Reference Manual, Version 8.0*. LogiCal Project, 2004.

[8] Harold Scott Macdonald Coxeter. *Projective Geometry*. Springer, 1987.

[9] Jérôme Créci and Loïc Pottier. Gb: une procédure de décision pour Coq. In *Actes JFLA 2004*, 2004. In french.

[10] Christophe Dehlinger, Jean-François Dufourd, and Pascal Schreck. Higher-Order Intuitionistic Formalization and Proofs in Hilbert's Elementary Geometry. In *Automated Deduction in Geometry 2000*, volume 2061 of *Lecture Notes in Computer Science*, pages 306–324. Springer-Verlag, 2000.

[11] Jean-François Dufourd. Design and Formal Proof of a New Optimal Image Segmentation Program with Hypermaps. *Pattern Recognition*, 40(11):2974–2993, 2007.

[12] Jean Duprat. Une axiomatique de la géométrie plane en coq. In *Actes des JFLA 2008*, pages 123–136. INRIA, 2008. In french.

[13] Jean Duprat, Frédérique Guilhot, and Julien Narboux. Toward a "common" language for formally stating elementary geometry theorems, 2008. Draft.

[14] Georges Gonthier and Assia Mahboubi. *A Small Scale Reflection Extension for the Coq System*. INRIA and Microsoft Research, 2007.

[15] 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. In french.

[16] Arend Heyting. Axioms for intuitionistic plane affine geometry. In P. Suppes L. Henkin and A. Tarski, editors, *The axiomatic Method, with special reference to Geometry and Physics*, pages 160–173, Amsterdam, 1959. North-Holland.

[17] Christoph M. Hoffmann and Robert Joan-Arinyo. *Handbook of Computer Aided Geometric Design*, chapter Parametric Modeling, pages 519–541. Elsevier, 2002.

[18] Christophe Jermann, Gilles Trombettoni, Bertrand Neveu, and Pascal Mathis. Decomposition of geometric constraint systems: a survey. *International Journal of Computational Geometry and Application*, 16(5-6):379–414, november 2006.

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

[20] Laura Meikle and Jacques Fleuriot. Mechanical theorem proving in computation geometry. In Hoon Hong and Dongming Wang, editors, *Automated Deduction in Geometry 2004*, volume 3763 of *Lecture Notes in Computer Science*, pages 1–18. Springer-Verlag, November 2005.

[21] Dominique Michelucci, Sebti Foufou, and Pascal Schreck Loïc Lamarque. Geometric constraints solving: some tracks. In *SPM '06: Proceedings of the 2006 ACM symposium on Solid and physical modeling*, pages 185–196. ACM, ACM Press, New York, USA, 2006.

[22] Dominique Michelucci and Pascal Schreck. Incidence Constraints: a Combinatorial Approach. *International Journal of Computational Geometry and Application*, 16(5-6):443–460, 2006.

[23] Alexandre Miquel and Benjamin Werner. The Not So Simple Proof-Irrelevant Model of CC. In Herman Geuvers and Freek Wiedijk, editors, *TYPES 2002*, volume 2646 of *LNCS*, pages 240–258. Springer, 2002.

[24] 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.

[25] Julien Narboux. Mechanical theorem proving in Tarski's geometry. In *Post-proceedings of Automatic Deduction in Geometry 06*, volume 4869 of *LNCS*, pages 139–156. Springer-Verlag, 2007.

[26] Lawrence C. Paulson. The Isabelle reference manual, 2006.

[27] David Pichardie and Yves Bertot. Formalizing convex hulls algorithms. In *Proc. of 14th International Conference on Theorem Proving in Higher Order Logics (TPHOLs'01)*, volume 2152 of *Lecture Notes in Computer Science*, pages 346–361. Springer-Verlag, 2001.

[28] Pascal Schreck. Robustness in CAD Geometric Construction. In *5th International Conference on Information Visualisation IV2001*, pages 111–116, London, july 2001.

[29] Wolfram Schwabhauser, Wanda Szmielew, and Alfred Tarski. *Metamathematische Methoden in der Geometrie*. Springer-Verlag, 1983. In german.

[30] Carlos Simpson. Computer Theorem Proving in Math, April 2004. http://hal.archives-ouvertes.fr/hal-00000842/en/.

[31] Matthieu Sozeau and Nicolas Oury. First-Class Type Classes. In *Proceedings of TPHOLs 08*, LNCS. Springer, 2008. to appear.

[32] Alfred Tarski. What is Elementary Geometry? In L. Henkin, P. Suppes, and A. Tarski, editors, *The axiomatic Method, with special reference to Geometry and Physics*, pages 16–29, Amsterdam, 1959. North-Holland.

[33] Jan von Plato. The axioms of constructive geometry. In *Annals of Pure and Applied Logic*, volume 76, pages 169–200, 1995.