Formalizing Some " Small " Finite Models of Projective Geometry in Coq

David Braun, Nicolas Magaud, Pascal Schreck

Formalizing Some "Small" Finite Models of Projective Geometry in Coq

David Braun, Nicolas Magaud, and Pascal Schreck

Icube UMR 7357 CNRS - Université de Strasbourg (IGG)
{david.braun,magaud,schreck}@unistra.fr

Abstract. We study two different descriptions of incidence projective geometry: a synthetic, mathematics-oriented one and a more practical, computation-oriented one, based on the combinatorial concept of rank of a set of points. Using both axiom systems, we prove that some specific finite planes (resp. spaces) verify the axioms of projective plane (resp. space) geometry and Desargues' property. It requires using repeated case analysis on all variables of some finite inductive data-types and leads to numerous (sub-)goals in the Coq proof assistant. We thus investigate to what extend Coq can deal with such a combinatorial explosion in the number of cases to handle. We propose some easy-to-implement but relevant proof optimizations which, combined together, lead to an efficient way to deal with such large proofs.

Keywords: Coq, proof automation, combinatorial explosion, finite inductive types, projective geometry, finite geometry, Desargues' property

1 Introduction

Incidence projective geometry is one of the simplest and most expressive frameworks used to describe some aspects of geometry. It is a good candidate for formalization: few axioms are needed and some key geometric properties such as Desargues' one can be formally stated and proved correct under some specific assumptions (see [11, 12]).

The notion of incidence projective plane is mainly defined by two axioms: two distinct points define a single line and two lines concur in a single point. A third axiom is usually used to catch precisely the dimension of geometry. For higher dimensions, the second axiom is a bit more complicated and defined as the two following statements: (1) two lines concur in at most one point and (2) *Pasch*'s axiom: given four different points A, B, C and D, if lines AB and CD concur, so do lines AC and BD. Moreover, other axioms can be added to avoid degenerate cases.

Proving properties in projective geometry or proving that some planes or spaces are actual models of projective geometry is usually based on analyzing a few general configurations as well as numerous degenerate cases. Using a proof assistant such as Coq [3, 6] makes it easier for the user to write a correct and comprehensive proof. Indeed, Coq forces the programmer to handle each possible

case in the proof. In addition, all details of the proof must be provided, which improves the confidence in it and allows the system to verify the proofs (by type-checking). The drawback is that it represents a tremendous amount of work for the proof developer. Thankfully, the Coq proof assistant and its tactic language Ltac allow to build *ad-hoc* tactics to automate large parts of the proofs efficiently.

We use two equivalent formal descriptions of projective geometry: a synthetic one and an alternative one using a matroid structure operating on points [4]. We check to what extent each of them allows to perform tractable, readable, easy-to-write and easy-to-process proofs. To achieve this goal, we work with some finite models of projective geometry: $pg(2,2)$, also known as Fano plane, $pg(2,3)$ and $pg(2,5)$; as well as the smallest finite projective space $pg(3,2)$ (see subsection 2.3). As models grow bigger, we need smarter proof techniques to cope with the inherent complexity and to keep memory usage, proof search and compile time under control.

Related Work Finite geometry has been studied since the late 19th century and is intrinsically linked to the development of algebraic structures like division rings, near fields or ternary rings. There has been a renewed interest with its application to computational domains like cryptography or planning (see [2, 5] for a comprehensive state of the art). The theoretical aspects are out of the scope of this paper. Rather we are interested in efficiently automating proofs with numerous cases within the Coq proof assistant. The use of ranks to carry out proofs in projective geometry was first introduced by Michelucci and Schreck [14]. Our work reuses some ideas of the mathematical components library about finite types [13] but we choose to refactor parts of it to suit our own needs.

Outline This article is organized as follows. In Section 2, we present two different ways of specifying projective geometry, directly or by using rank theory. We also introduce some common properties (e.g. Desargues' property) and describe some finite models of projective geometry. In Section 3, we study the inherent complexity of the finite models and describe some techniques to handle these complexity issues properly in Coq. In Section 4, we present some more practical tools to help the user to write formal proofs easily via proof structuring and automation. Finally, in Section 5, we summarize our contributions and present some suitable perspectives.

Notation We name axioms *AXYN*. *A* stands for axiom, X is the axiom number, Y may take two values (P = projective, R = rank) and N denotes the dimension.

2 Formal Specification of Projective Geometry, Rank Theory and Finite fields

We define two equivalent axiom systems for incidence projective geometry: one based on the usual synthetic description, and another one based on the combi-

natorial notion of rank provided by the matroid structure of incidence projective geometry. Then, we prove, using these two specifications, that some finite planes/spaces are models of incidence projective geometry and we study Desargues' theorem.

2.1 Axiom Systems for Incidence Projective Geometry

Incidence Geometry is a simple view of geometry, where only points and lines, together with the incidence relation linking them are kept. Projective geometry is obtained by assuming that two coplanar lines always meet. Incidence projective geometry can be easily described as a small set of axioms, as shown in Coxeter's book [7].

Plane The axiom system for projective plane geometry consists of five axioms presented in Fig. 1. Axioms (A1P2) and (A2P2) deal with construction of points and lines. Axiom (A3P2) concerns uniqueness of points and lines. Finally, axiom (A4P2) states that each line contains at least three points; and axiom (A5P2) expresses that there always exists two distinct lines.

(A1P2) Line-Existence : $\forall$ A B : Point, $\exists$ l : Line, A $\in$ l $\wedge$ B $\in$ l

(A2P2) Point-Existence : $\forall$ l m : Line, $\exists$ A : Point, A $\in$ l $\wedge$ A $\in$ m

(A3P2) Uniqueness : $\forall$ A B : Point, $\forall$ l m : Line, A $\in$ l $\wedge$ B $\in$ l $\wedge$ A $\in$ m $\wedge$ B $\in$ m $\Rightarrow$ A = B $\vee$ l = m

(A4P2) Three-Points : $\forall$ l : Line, $\exists$ A B C : Point, A $\neq$ B $\wedge$ B $\neq$ C $\wedge$ A $\neq$ C $\wedge$ A $\in$ l $\wedge$ B $\in$ l $\wedge$ C $\in$ l

(A5P2) Lower-Dimension : $\exists$ l m: Line, $l \neq m$

Fig. 1: Axiom system for projective plane geometry

Space and higher dimensions Similarly, we define an axiom system to capture projective space geometry in Fig. 2 by extending the previous one. The system still contains five axioms with three of them remaining unchanged (A1P3, A3P3, A4P3). *Pasch*'s axiom replaces (A2P2) and assumes that two coplanar lines always meet. Furthermore, we modify the axiom *Lower-Dimension* to capture projective geometry for spaces of dimension greater or equal than 3. It is possible to limit this to spatial geometry by adding the optional axiom (A6P3) to constrain the dimension to be exactly 3.

(A1P3) Line-Existence : $\forall$ A B : Point, $\exists$ l : Line, A $\in$ l $\wedge$ B $\in$ l

(A2P3) Pasch : $\forall$ A B C D : Point, $\forall$ l_{AB} l_{CD} l_{AC} l_{BD} : Line,
A $\neq$ B $\wedge$ A $\neq$ C $\wedge$ A $\neq$ D $\wedge$ B $\neq$ C $\wedge$ B $\neq$ D $\wedge$ C $\neq$ D $\wedge$
A $\in$ l_{AB} $\wedge$ B $\in$ l_{AB} $\wedge$ C $\in$ l_{CD} $\wedge$ D $\in$ l_{CD} $\wedge$
A $\in$ l_{AC} $\wedge$ C $\in$ l_{AC} $\wedge$ B $\in$ l_{BD} $\wedge$ D $\in$ l_{BD} $\wedge$
($\exists$ I : Point, I $\in$ l_{AB} $\wedge$ I $\in$ l_{CD}) $\Rightarrow$
($\exists$ J : Point, J $\in$ l_{AC} $\wedge$ J $\in$ l_{BD})

(A3P3) Uniqueness : $\forall$ A B : Point, $\forall$ l m : Line,
A $\in$ l $\wedge$ B $\in$ l $\wedge$ A $\in$ m $\wedge$ B $\in$ m $\Rightarrow$ A = B $\vee$ l = m

(A4P3) Three-Points : $\forall$ l : Line, $\exists$ A B C : Point,
A $\neq$ B $\wedge$ B $\neq$ C $\wedge$ A $\neq$ C $\wedge$ A $\in$ l $\wedge$ B $\in$ l $\wedge$ C $\in$ l

(A5P3) Lower-Dimension : $\exists$ l m : Line, $\forall$ p : Point, p $\notin$ l $\vee$ p $\notin$ m

(A6P3) Upper-Dimension : $\forall$ l1 l2 l3 : Line, l1 $\neq$ l2 $\wedge$ l1 $\neq$ l3 $\wedge$ l2 $\neq$ l3 $\Rightarrow$
$\exists$ l4 : Line, $\exists$ P1 P2 P3 : Point, P1 $\in$ l1 $\wedge$
P1 $\in$ l4 $\wedge$ P2 $\in$ l2 $\wedge$ P2 $\in$ l4 $\wedge$ P3 $\in$ l3 $\wedge$ P3 $\in$ l4

Fig. 2: Axiom system for projective space geometry

2.2 A Rank-based Axiom Systems

Ranks are based on matroids [16] and they allow a combinatorial approach to theorem proving in projective geometry. Matroid theory allows us to capture and generalize the main set of properties of linear dependence in vector spaces. When combined with a finite set of points, it captures incidence (collinearity, coplanarity, ...) between these points without handling directly lines or planes. It makes the computational content of projective geometry more accessible, the price to pay being less readable statements and proofs. It is quite similar to analytic geometry which also favors computability at the expense of readability.

A rank function is an integer-valued function on a finite set of objects E that can be associated to a matroid if and only if the following conditions of Fig. 3 are satisfied. To illustrate rank function, we give an intuitive interpretation of how the synthetic and rank-based descriptions correspond (see Tab. 1).

(A1R2-R3) nonnegative and subcardinal : $\forall$ X $\subseteq$ E, 0 $\leq$ rk(X) $\leq$ |X|

(A2R2-R3) nondecreasing : $\forall$ X $\subseteq$ Y, rk(X) $\leq$ rk(Y)

(A3R2-R3) submodular : $\forall$ X,Y $\subseteq$ E, rk(X$\cup$Y) + rk(X$\cap$Y) $\leq$ rk(X) + rk(Y)

Fig. 3: Matroid properties for the rank function

$rk\{A,B\} = 1$	$A = B$
$rk\{A,B\} = 2$	$A \neq B$
$rk\{A,B,C\} = 2$	A,B,C are collinear with at least two of them distinct
$rk\{A,B,C\} \leq 2$	A,B,C are collinear
$rk\{A,B,C\} = 3$	A,B,C are not collinear
$rk\{A,B,C,D\} = 3$	A,B,C,D are coplanar, not all collinear
$rk\{A,B,C,D\} = 4$	A,B,C,D are not coplanar

Tab. 1: Some rank statements and their geometric interpretations

Plane To capture projective geometry entirely, we need to add some more *geometry-oriented* axioms. These five additional axioms are presented in Fig. 4. The first two ones establish the non-degeneracy of the rank function. The other ones are more or less direct translations of the axioms of projective geometry.

(A4R2) Rk-Singleton : $\forall$ P : Point, $rk\{P\} \geq 1$

(A5R2) Rk-Couple : $\forall$ P Q: Point, $P \neq Q \Rightarrow rk\{P, Q\} \geq 2$

(A6R2) Rk-Inter : $\forall$ A B C D, $\exists$ J, $rk\{A, B, J\} = rk\{C, D, J\} = 2$

(A7R2) Rk-Three-Points : $\forall$ A B, $\exists$ C, $rk\{A, B, C\} = rk\{B, C\} = rk\{A, C\} = 2$

(A8R2) Rk-Lower-Dimension : $\exists$ A B C, $rk\{A, B, C\} \geq 3$

Fig. 4: Rank-based axiom system for projective plane geometry

Space Finally, we define a rank-based axiom system to describe projective space in Fig 5. Again, only the axioms *Pasch* and *Lower-Dimension* are modified. To restrict the dimension to 3, we add the optional axiom (A9R3).

Equivalence proof We recently proved [4] that the two descriptions of incidence geometry presented above are equivalent:

Theorem. *The axiom system based on incidence projective geometry and the rank-based axiom system are equivalent respectively in 2D , $\geq$3D and 3D.*

This equivalence gives us the possibility to choose the most adequate theory to prove a lemma. Indeed, statements can be bilaterally translated. This important fact allows us both to compare proofs carried out with two different approaches but also to complete some demonstrations when one of the two theories is not conducive to a tractable proof.

(A4R3) Rk-Singleton : $\forall$ P : Point, rk{P} ≥ 1

(A5R3) Rk-Couple : $\forall$ P Q: Point, P $\neq$ Q $\Rightarrow$ rk{P, Q} ≥ 2

(A6R3) Rk-Pasch : $\forall$ A B C D, rk{A, B, C, D} $\leq 3 \Rightarrow \exists$ J, rk{A, B, J} = rk{C, D, J} = 2

(A7R3) Rk-Three-Points : $\forall$ A B, $\exists$ C, rk{A, B, C} = rk{B, C} = rk{A, C} = 2

(A8R3) Rk-Lower-Dimension : $\exists$ A B C D, rk{A, B, C, D} ≥ 4

(A9R3) Rk-Upper-Dimension : $\forall$ A B C D E, rk{A, B, C, D, E} ≤ 4

Fig. 5: Rank-based axiom system for projective space geometry

2.3 Finite models

The first examples of incidence geometries are built with fields. For instance, affine planes often arise from F^2, where F is a field, via a coordinate system and projective planes from F^3 via a homogeneous coordinate system. Considering finite fields leads to classical examples of finite geometries. For instance, Fano spaces come from field $\mathbb{Z}/2\mathbb{Z}$.

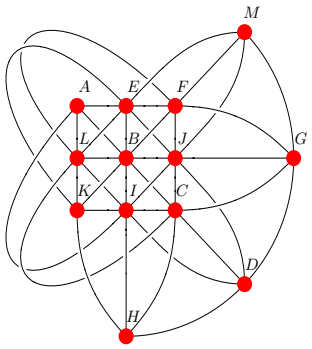

Fig. 6: A configuration of $pg(2, 3)$: 13 points and 13 lines (e.g. AEFG, CELM, DILF).

Finite fields of cardinality n denoted by $GF(n)$ are called Galois fields as they are isomorphic to the field $\mathbb{Z}_p[X]/f(X)$ where p is a prime number, $\mathbb{Z}_p$ stands for $\mathbb{Z}/p\mathbb{Z}$ and f is an irreducible polynomial over $\mathbb{Z}_p[X]$. It follows that, k being the degree of f, such a finite field has cardinality $n = p^k$ and each line of a corresponding affine space (resp. projective space) has cardinality n (resp. $n+1$). Finite projective spaces arising from $GF(n)$ are then denoted by $pg(d, n)$

	point(s)	line(s)	plane(s)
`pg(2,2)`	7	7	1
`pg(2,3)`	13	13	1
`pg(2,4)`	21	21	1
`pg(2,5)`	31	31	1
`pg(3,2)`	15	35	15

Tab. 2: Description of several finite projective plane/space.

where d is the dimension of the space and n the order of the underlying field. Tab. 2 summarizes cardinalities and Fig. 6 represents $pg(2,3)$.

Forgetting the way that such spaces are built, $pg(d,n)$ spaces offer a convenient benchmark to test our strategies for mechanizing proofs in Coq. Although we work in the context of $pg(d,n)$, we only take into account its geometric characteristics. This means that while keeping in mind the theoretical results, we do not use coordinates in our formalization.

2.4 Desargues' property

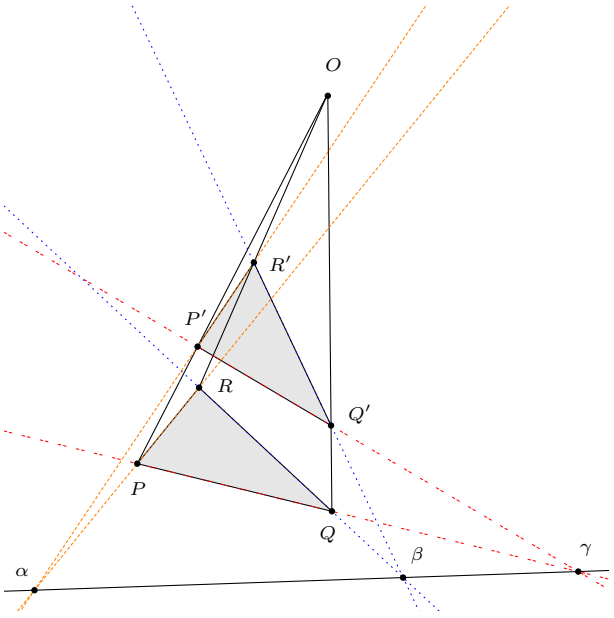

Fig. 7: A configuration of Desargues' property.

It is well known that Desargues' property (see Fig. 7) holds in any projective space of dimension higher or equal to 3. This was formally proven in [12]. How-

ever, when considering projective planes, Desargues' property is independent of the axiom system of Fig. 1. This means that there exists Desarguesian and non-Desarguesian planes. For instance, Moulton's plane (see [11, 15] for details) or Hall's [9] planes of order 9 are non-Desarguesian planes. Desargues' theorem states:

Theorem. *If the three lines joining the corresponding vertices of two triangles PQR and $P'Q'R'$ all meet in a point O called the perspector*[1]*, then the three intersections of pairs of corresponding triangle sides lie on a line $\alpha\beta\gamma$. Equivalently, if two triangles are perspective from a point, then they are perspective from a line.*

Now that the geometric framework is depicted, we shall investigate possibilities of automation within proofs. Throughout this paper, we aim at proving that some finite structures described using only points, lines and an incidence relation are models of these axiom systems. When dealing with plane projective geometry, we also analyze whether Desargues' property actually holds.

3 Dealing with complexity in building some finite models of incidence projective geometry

3.1 Plane

We use finite projective models to study the large-scale automation of proofs of geometric properties. One can prove fairly easily that the axioms of projective plane geometry hold for $pg(2, 2)$, $pg(2, 3)$ and $pg(2, 5)$. In the same way, we show that the axioms of rank theory hold for $pg(2, 2)$ and $pg(2, 3)$. We use these examples to show how to manage the proof complexity in Coq.

We identify several criteria (e.g. the geometric context, the formulation of the statements) which can strongly influence the complexity of the proofs. As an example, we compare some proofs which have been mechanized in Coq using both incidence projective geometry and rank theory.

Finite model First of all, we work on a finite domain. In this context, to carry out geometric reasoning, it is necessary to know all the points and lines (and planes) that describe our finite projective plane (resp. projective space). For instance the description of $pg(2, 3)$ contains 13 points and 13 lines (see Fig. 6) in incidence projective geometry and looks like:

```
Inductive ind_Point : Set := A | B | C | ... | K | L | M.

Inductive ind_line : Set := ABCD | AEFG | AIJM | AHKL | BEHI | BGJL
| BFKM | CELM | CFHJ | CGIK | DEJK | DGHM | DFIL.
```

[1] The perspector is the point at which the three lines connecting the vertices of two perspective triangles concur.

```
Definition Incid_bool (P:Point) (l:Line) : bool := match P with
  | A => match l with
           | ABCD | AEFG | AIJM | AHKL => true
           | _ => false
         end
  [...]
end.
```

The description of finite models can be easily generated algorithmically by only specifying all points and lines. In this way, the relation of incidence linking these two objects is thus automatically created. The size of the specification of $pg(2,n)$ increases quickly as n grows bigger, indeed $pg(2,n)$ has n^2+n+1 points and as many lines.

Case analysis In such a finite model, to prove a geometric statement requires to check all the possible configurations of this theorem, i.e. to perform case analysis on both points and lines. Most often a brute-force approach leads to too many cases, which makes the proof not tractable in Coq. Let us illustrate this case analysis issue on one of the axioms of the incidence projective geometry in the finite projective plane $pg(2,3)$. For instance, the (A3P2) *Uniqueness* axiom:

```
Lemma 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.
```

As $pg(2,3)$ contains 13 points and 13 lines, basic case analysis leads to $13^4 = 28\ 051$ cases to be dealt with. This situation is not yet critical, such a proof is still easily performed. It becomes more tedious when dealing with $pg(2,5)$ and its 31 points and 31 lines, where 923 521 cases must be studied. For a given n, the projective plane $pg(2,n)$ has $(n^2+n+1)^4$ possible combinations to be investigated, so such proofs are tractable only for some small n.

Formulation and choice of theory A second factor strongly influencing complexity is the formulation of statements. This question is well known and studied in the theory of complexity especially in the problem SAT[1, 17]. Criteria such as the size of the clauses, number of propositions and the order of propositions have a significant impact on the resolution time of a proof. For example, let us consider the two definitions of intersection existence in $pg(2,3)$ first using incidence geometry, and second using ranks.

```
Lemma point_existence : forall (l1 l2 :Line),
   exists A : Point, Incid A l1 /\ Incid A l2.


Lemma rk_inter : forall A B C D : Point,
   exists J, rk(triple A B J) = 2 /\ rk(triple C D J) = 2.
```

Case analysis in the first description generates $13^2 = 169$ cases before providing a witness to the existential quantifier whereas in the second statement we again face $13^4 = 28\ 051$ cases. It would be necessary to create a method of resolution of the existential formula one hundred times faster in rank theory to obtain the same execution time as in incidence projective geometry. So choosing an appropriate description of a formula is utterly relevant to make the proofs doable in practise. The best way to deal properly with the combinatorial explosion caused by successive case analysis is to manage the pruning of the proof tree as early as possible.

Proof tree pruning Let us consider again the axiom of *Uniqueness* (A3P2) and its proof in the finite projective plane $pg(2, 3)$:

```
Lemma 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.
Proof.
induction A;induction B;induction l;induction m;
try discriminate;try (left;reflexivity);try (right;reflexivity).
Qed.
```

Basic case analysis without pruning and quantifier management gives rise to 28 051 cases. A brute-force execution takes on a standard machine[2] about 40 seconds in this situation. More clever strategies are required to ensure that the proofs remain tractable. The variable A is linked to l in the hypothesis `Incid A l`. It is thus possible to prune the proof tree after the induction on line l when the point A is not incident to the line l. Another improvement consists in solving directly the left hand side of the goals right after the induction on B when the equality $A = B$ (i.e. the left side of the disjonction) holds. It is not necessary to carry on and perform case analysis on the next variable m if the goal can be discarded or is already verified. These two adjustments allow the proof to be built in less than 1 second.

```
Lemma 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.
Proof.
induction A;induction l;try discriminate;
induction B;try discriminate;try (left;reflexivity);
induction m;try discriminate;try (right;reflexivity).
Qed.
```

Constraining hypothesis Scheduling quantifiers based on assumptions can have a strong impact on proof tree pruning. In other words, the order in which the case analysis is performed is important. Furthermore, it is important to consider the pruning power of each hypothesis. The idea is to use first the most

[2] Computer setup : Intel(R) Core(TM) i5-4460 CPU @ 3.20GHz with 16G of memory

restrictive assumptions to prune as much as possible and as soon as possible to limit the width of the proof tree. Let us consider the assumptions $A \neq B$ and `Incid A l` in $pg(2,3)$. After performing induction on all variables, the first assumption allows to eliminate 7 cases out of 49 while the second one removes 28 cases out of 49. It is therefore more interesting to take the incidence hypothesis into account to quickly eliminate goals.

Pseudo depth-first search In highly-branching proofs, when the previous optimizations are not sufficient (because memory consumption is too big), we adapt the classical breadth-first search of Coq (`tac1;tac2;tac3`). By taking advantage of the right assiociativity, we carry out pseudo depth-first search in order to limit number of cases at each level of the demonstration (`tac1;(tac2;tac3)`). Finally, we work with the `abstract` [6] tactic to prove a sub-goal as a separate lemma to structure huge proof terms and to facilitate type checking.

These optimizations are independent of each others and allow to prove more lemmas, even when the combinatorial is huge.

3.2 Space

The above-mentioned techniques are even more relevant when dealing with the smallest projective space $pg(3,2)$[3]. It features 15 points and 35 lines. In the same way as in the plane, we can prove that the axioms of projective space geometry hold for $pg(3,2)$. However, it is a little more challenging to prove this. Indeed, while writing and feeding Coq with the proofs, we face strong limitations related to memory usage. Tactics have to be carefully designed and decomposition should be smart enough to avoid facing thousands of millions of sub-goals at the same level. Consider for instance the statement of *Pasch*'s axiom in $pg(3,2)$:

```
Lemma pasch : forall A B C D : Point, forall lAB lCD lAC lBD : Line,
              all_distinct A B C D ->
              Incid A lAB /\ Incid B lAB ->
              Incid C lCD /\ Incid D lCD ->
              Incid A lAC /\ Incid C lAC ->
              Incid B lBD /\ Incid D lBD ->
              (exists I : Point, (Incid I lAB /\ Incid I lCD)) ->
              exists J : Point, (Incid J lAC /\ Incid J lBD).
```

As finite space $pg(3,2)$ contains 15 points and 35 lines, case analysis leads to $15^4 \times 35^4 = 75\ 969\ 140\ 625$ cases to be dealt with. It is thus essential to limit the size of proof tree by eliminating as many cases as soon as possible. The order in which we perform inductions is no longer sufficient to maintain a tractable proof.

[3] An interactive representation of $pg(3,2)$ can be viewed on wolfram web site: http://demonstrations.wolfram.com/15PointProjectiveSpace/.

Proof parts usually proved using Ltac sophisticated tactics without user interaction need to be factorized into relevant lemmas and a careful decomposition into several intermediate lemmas is mandatory to complete the proof. In the proof of *Pasch*'s property, we state the following intermediate lemma which provides the actual line which carries two given (distinct) points T and Z. The function `l_from_points` computes a line which goes throught the two points T and Z (this line is unique when we have $T \neq Z$).

Here, the proofs-as-programs paradigm is fully exploited. Indeed, this function can be written as a simplified (non-dependent) version of the property (A1P3) *Line-existence* which can be directly used as a program[4]. It allows us to perform case analysis on lines without adding further cases (only one case is correct at each step).

Similarly, a program which retrieves the points which belongs to a given line l can easily be extracted from theorem (A4P3) *Three-Points*.

```
Lemma points_line : forall T Z : Point, forall x : Line,
Incid T x -> Incid Z x -> T<>Z -> x=(l_from_points(T,Z)).
```

In this way, we reduce the overall number of cases to check to $15^4 = 50625$ cases, before performing the elimination of the existential hypothesis in *Pasch*'s axiom: `exists I :Point, (Incid I lAB /\ Incid I lCD)`.

So far we made proofs manageable by the system, but we still need to help the user to write proofs. That is what we shall study in the next section.

4 Automating proofs of Desargues's property

All the techniques presented above in order to prove that some small planes or projective spaces are models of the projective incidence geometry can be reused to carry out the proof of Desargues' theorem in each of these models.

```
Lemma Desargues : forall O P Q R P' Q' R' X Y Z X' Y' Z'
X'' Y'' Z'' alpha beta gamma,
all_distinct O X Y Z X' Y' Z' X'' Y'' Z'' ->
rk(O,X,Y,Z)=2 -> rk(O,X',Y',Z')=2 -> rk(O,X'',Y'',Z'')=2 ->
rk(P,Q,gamma)=2 -> rk(P',Q',gamma)=2 -> rk(P,R,beta)=2 ->
rk(P',R',beta)=2 -> rk(Q,R,alpha)=2 -> rk(Q',R',alpha)=2 ->
rk(P,O,X,Y,Z)=2 -> rk(P',O,X,Y,Z)=2 ->
rk(Q,O,X',Y',Z')=2 -> rk(Q',O,X',Y',Z')=2 ->
rk(R,O,X'',Y'',Z'')=2 -> rk(R',O,X'',Y'',Z'')=2 ->
rk(O,P,P')=2 -> rk(O,Q,Q')=2 -> rk(O,R,R')=2 -> rk(O,P,Q)=3 ->
rk(O,P,R)=3 -> rk(O,Q,R)=3 -> rk(P,Q,R)=3 -> rk(P',Q',R')=3 ->
( rk(P,P')=2 \/ rk(Q,Q')=2 \/ rk(R,R')=2 ) ->
rk(alpha,beta,gamma)=2.
```

[4] Fully-specified functions can be automatically defined using the proof search capabilities of Coq.

It is well-known that the projective planes $pg(2, n)$ are Desarguesian. We formaly prove these results in Coq for $pg(2, 2)$ and $pg(2, 3)$. As in the previous proofs, using a naive approach leads to intractable proofs. The property of Desargues is expressed using 10 points. The last three ones can be automatically calculated from the first seven ones. In $pg(2, 3)$, induction on the first 7 points yields several billion cases to be treated without pruning.

In this case, ranks provide a much more efficient approach to handle the numerous configurations that we need to check. It is tractable if we prune the proof tree as much as possible during inductions on the ten points of the property. Automating this proof relies on some geometric aspects of Desargues' property and on the data structure of ranks.

4.1 Automation through geometry

First of all, we take advantage of some symmetries in Desargues' property. In the first place, we use the symmetry of the problem w.r.t. the center of perspective. By fixing this center as one of the points of the model $pg(2, x)$ and proving that the permutation of the points in a finite model remains a finite model, it is possible to prove that the property of Desargues holds whatever the center of perspective selected. Intuitively, this symmetry allows us to avoid induction on the perspector point.

The second symmetry that we use to decompose the problem follows from the permutation of the concurrent lines at the center of perspective. Let A be the perspector, it is possible to fix the straight lines containing A to form the two triangles. Subsequently, we show that every permutation of these lines always satisfies the property.

Finally, we take advantage of the conditions of non-degeneracy to quickly eliminate the degenerate cases of Desargues' theorem and thus limit the combinatorial explosion. For example, it is possible to consider a more general theorem where the two triangles can share at most two points in common. This theorem leads to a contradiction in the specification of the line $\alpha\beta\gamma$ (some lines are confused). By restricting the theorem to the case where triangles can have only one point in common, we eliminate approximately 33% of the goals at all levels of the demonstration.

4.2 Automation thanks to proof engineering

Thanks to the rank structure, we can represent homogeneously all incidences of our geometric context by dealing only with points. Intuitively this means that we can avoid performing case analysis on lines without increasing the number of cases on the points. For instance considering Desargues' theorem, six case analyses on lines can be removed. It becomes even more meaningful in the higher dimensions when manipulating planes, etc.

In addition, when writing tactics with Ltac to perform simplifications (e.g. rewriting, elimination of contradiction, attempt to solve), there is no need to consider objects of several types or multiple predicates. We simply match the result

returned by the function of rank, as all propositions are of the form $rk(E) = n$ where E is a set of points and n is a natural number representing intuitively the dimension of the set.

Finally, it is better to avoid generic tactics such as `auto`, `intuition` or `omega`, and to use specific lemmas which solve the goal instead. Proofs of statements of the form $rk(E) = n$ usually proceed by first proving separately that $rk(E) \leq n$ and $rk(E) \geq n$, and then use `omega` to deduce the equality from the two inequalities. Of course, if such a proof scheme is heavily used, running `omega` becomes a bottleneck. We can instead write a simple lemma $(\forall n : \texttt{nat}, rk(E) \leq n \to rk(E) \geq n \to rk(E) = n)$ and apply it to conclude the proof. An single application of `apply` is always significantly cheaper than calling `omega`. However, the drawback is that we have a more specific proof, which may be less robust to changes in the specification. Finding such bottlenecks can be easily achieved using the Ltac profiler [18].

5 Conclusion

We verify that some finite planes (resp. spaces) are actually models of projective plane (resp.space) geometry. We achieve that by using two distinct approaches, a mathematics-oriented one and a computer-science-oriented one featuring ranks. Overall it represents 5000 lines of specification and 2500 lines of proofs. All the results are summarized in Tab. 3. For each formalization, it presents three key figures: the number of lines of specification, the number of lines of proof as well as the time required to compile it.

	Formalization of Projective Geometry					
	using the synthetic description			using ranks		
	spec.	proof	compile time	spec.	proof	compile time
$pg(2,2)$ is a model	216	71	2s	127	42	16s
`Desargues` holds in $pg(2,2)$	188	205	37s	297	162	26s
$pg(2,3)$	149	46	7s	309	77	2055s
`Desargues` *in* $pg(2,3)$	191	225	CE	2089	386	10700s
$pg(2,5)$	74	28	90s	CE		
`Desargues` *in* $pg(2,5)$	CE			CE		
$pg(3,2)$	267	67	4309s	CE		
`Desargues` *in* $pg(3,2)$	Overall proof in 3D thanks to [4, 12]					

Tab. 3: Benchmarks for various proofs using Coq on an Intel(R) Core(TM) i5-4460 CPU @3.20GHz with 16G of memory. CE means Combinatorial Explosion.

This provides a good stress test for Coq. Indeed, it is a small theory, but proving that the axioms hold requires performing huge proofs with numerous cases. Our experiments shed light on some regression in the efficiency of Coq to perform proofs and type-check them, starting from version 8.5. This issue is currently being addressed by the coqdev team.

The optimizations that we propose allow to go further in the order of magnitude of the planes/spaces that we can handle. Eventually, an interesting goal would be to tackle some of Hall's planes which feature 91 points and 91 lines.

Currently, we are working on a more comprehensive benchmark featuring more projective planes/spaces and using various provers using the TPTP framework [17]. Using brute-force, only 3 provers find a proof of Desargues' theorem in a suitable time of 300 seconds for $pg(2,2)$ (iprover, Vampire [10] and Z3 [8]). The Vampire SAT seems very promising with solutions 10 times faster. However, provers do not provide a formal checkable proof.

The Coq development is available at https://github.com/ProjectiveGeometry/.

References

1. M. Armand, G. Grégoire, B. Keller, L. Théry, and B. Werner. Verifying SAT and SMT in Coq for a Fully Automated Decision Procedure. In *International Workshop on Proof-Search in Axiomatic Theories and Type Theories (PSATTT'11)*, 2011.
2. L. M. Batten. *Combinatorics of Finite Geometries*. Cambridge Univ. Press, 1997.
3. Y. Bertot and P. Castéran. *Interactive Theorem Proving and Program Development, Coq'Art: The Calculus of Inductive Constructions*. Springer, 2004.
4. D. Braun, N. Magaud, and P. Schreck. An Equivalence Proof Between Rank Theory and Incidence Projective Geometry. In *Automated Deduction in Geometry (ADG'2016)*, pages 62–77, 2016.
5. F. Buekenhout, editor. *Handbook of Incidence Geometry*. North Holland, 1995.
6. Coq development team. *The Coq Proof Assistant Reference Manual, Version 8.6*. LogiCal Project, 2017.
7. H. S. M. Coxeter. *Projective Geometry*. Springer Science & Business Media, 2003.
8. L. M. de Moura and N. Bjørner. Z3: An Efficient SMT Solver. In *Proceedings of TACAS 2008*, volume 4963 of *LNCS*, pages 337–340. Springer, 2008.
9. M. Hall. Projective planes. *Transactions of the American Mathematical Society*, 54(2):229–277, 1943.
10. L. Kovács and A. Voronkov. First-order theorem proving and Vampire. In *International Conference on Computer Aided Verification*, pages 1–35. Springer, 2013.
11. N. Magaud, J. Narboux, and P. Schreck. Formalizing Projective Plane Geometry in Coq. In *Automated Deduction in Geometry (ADG'2008)*, LNAI 6301, pages 141–162. Springer, 2008.
12. N. Magaud, J. Narboux, and P. Schreck. A Case Study in Formalizing Projective Geometry in Coq: Desargues Theorem. *Computational Geometry: Theory and Applications*, 45(8):406–424, 2012.
13. A. Mahboubi and E. Tassi. *Mathematical Components*. Draft, 2016.
14. D. Michelucci and P. Schreck. Incidence Constraints: a Combinatorial Approach. *Int. Journal of Computational Geometry and Applications*, 16(5-6):443–460, 2006.
15. F. R. Moulton. A Simple Non-Desarguesian Plane Geometry. *Transactions of the American Mathematical Society*, 3(2):192–195, 1902.
16. J. G. Oxley. *Matroid Theory*, volume 3. Oxford University Press, USA, 2006.
17. G. Sutcliffe. The TPTP Problem Library and Associated Infrastructure. *Journal of Automated Reasoning*, 43(4):337, 2009.
18. T. Tebbi and J. Gross. A Profiler for Ltac. In *Coq PL Workshop 2015*, 2015.