Projective Geometry

Anthony Bordg

February 6, 2026

Abstract

We formalize the basics of projective geometry.
In particular, we give a proof of the so-called Hessenberg's theorem in projective plane geometry (see [1] for an alternative proof using a Coherent Logic prover in Prolog which generates Coq proof scripts).
We also provide a proof of the so-called Desargues's theorem based on an axiomatization [2] of (higher) projective space geometry using the notion of rank of a matroid. This last approach allows to handle incidence relations in an homogeneous way dealing only with points and without the need of talking explicitly about lines, planes or any higher entity.

Contents

theory *Projective-Plane-Axioms*
imports *Main*
begin

Contents:

- We introduce the types of points and lines and an incidence relation between them.
- A set of axioms for the projective plane (the models of these axioms are n-dimensional with $n \geq 2$).

1 The Axioms of the Projective Plane

locale *projective-plane* =

fixes *incid* :: *'point* ⇒ *'line* ⇒ *bool*

assumes *ax1*: $\exists\, l.\ incid\ P\ l \wedge incid\ Q\ l$

assumes *ax2*: $\exists\, P.\ incid\ P\ l \wedge incid\ P\ m$

assumes *ax-uniqueness*: $[\![incid\ P\ l;\ incid\ Q\ l;\ incid\ P\ m;\ incid\ Q\ m]\!] \Longrightarrow P = Q \vee l = m$

assumes *ax3*: $\exists\, A\ B\ C\ D.\ distinct\ [A,B,C,D] \wedge (\forall\, l.$
$(incid\ A\ l \wedge incid\ B\ l \longrightarrow \neg(incid\ C\ l) \wedge \neg(incid\ D\ l)) \wedge$
$(incid\ A\ l \wedge incid\ C\ l \longrightarrow \neg(incid\ B\ l) \wedge \neg(incid\ D\ l)) \wedge$
$(incid\ A\ l \wedge incid\ D\ l \longrightarrow \neg(incid\ B\ l) \wedge \neg(incid\ C\ l)) \wedge$
$(incid\ B\ l \wedge incid\ C\ l \longrightarrow \neg(incid\ A\ l) \wedge \neg(incid\ D\ l)) \wedge$
$(incid\ B\ l \wedge incid\ D\ l \longrightarrow \neg(incid\ A\ l) \wedge \neg(incid\ C\ l)) \wedge$
$(incid\ C\ l \wedge incid\ D\ l \longrightarrow \neg(incid\ A\ l) \wedge \neg(incid\ B\ l)))$

end

theory *Pappus-Property*
imports *Main Projective-Plane-Axioms*
begin

Contents:

- We give two formulations of Pappus's property for a configuration of nine points *is-pappus1* *is-pappus2*.
- We prove the equivalence of these two formulations *pappus-equiv*.
- We state Pappus property for a plane *is-pappus*.

2 Pappus's Property

context *projective-plane*
begin

definition *col* :: $['point, 'point, 'point] \Rightarrow bool$ **where**
$col\ A\ B\ C \equiv \exists l.\ incid\ A\ l \wedge incid\ B\ l \wedge incid\ C\ l$

lemma *distinct6-def*:
$distinct\ [A,B,C,D,E,F] \equiv (A \neq B) \wedge (A \neq C) \wedge (A \neq D) \wedge (A \neq E) \wedge (A \neq F)$
$\wedge$
$(B \neq C) \wedge (B \neq D) \wedge (B \neq E) \wedge (B \neq F) \wedge$
$(C \neq D) \wedge (C \neq E) \wedge (C \neq F) \wedge$
$(D \neq E) \wedge (D \neq F) \wedge$
$(E \neq F)$
⟨*proof*⟩

definition *lines* :: $'point \Rightarrow 'point \Rightarrow 'line\ set$ **where**
$lines\ P\ Q \equiv \{l.\ incid\ P\ l \wedge incid\ Q\ l\}$

lemma *uniq-line*:
assumes $P \neq Q$ **and** $l \in lines\ P\ Q$ **and** $m \in lines\ P\ Q$
shows $l = m$
⟨*proof*⟩

definition *line* :: $'point \Rightarrow 'point \Rightarrow 'line$ **where**
$line\ P\ Q \equiv @l.\ incid\ P\ l \wedge incid\ Q\ l$

definition *is-a-proper-intersec* :: $['point, 'point, 'point, 'point, 'point] \Rightarrow bool$ **where**
is-a-proper-intersec $P\ A\ B\ C\ D \equiv (A \neq B) \wedge (C \neq D) \wedge (line\ A\ B \neq line\ C\ D)$
$\wedge\ col\ P\ A\ B \wedge col\ P\ C\ D$

definition *is-pappus1* ::
$['point, 'point, 'point, 'point, 'point, 'point, 'point, 'point, 'point] => bool$ **where**
is-pappus1 $A\ B\ C\ A'\ B'\ C'\ P\ Q\ R \equiv$
$distinct[A,B,C,A',B',C'] \longrightarrow col\ A\ B\ C \longrightarrow col\ A'\ B'\ C'$
$\longrightarrow$ *is-a-proper-intersec* $P\ A\ B'\ A'\ B \longrightarrow$ *is-a-proper-intersec* $Q\ B\ C'\ B'\ C$
$\longrightarrow$ *is-a-proper-intersec* $R\ A\ C'\ A'\ C$

$\longrightarrow$ *col P Q R*

definition *is-a-intersec* :: $[{}'point, {}'point, {}'point, {}'point, {}'point] \Rightarrow bool$ **where**
is-a-intersec P A B C D $\equiv$ *col P A B* $\wedge$ *col P C D*

definition *is-pappus2* ::
$[{}'point, {}'point, {}'point, {}'point, {}'point, {}'point, {}'point, {}'point, {}'point] \Rightarrow bool$ **where**
is-pappus2 $A\ B\ C\ A'\ B'\ C'\ P\ Q\ R \equiv$
$(distinct\ [A,B,C,A',B',C'] \vee (A \neq B' \wedge A' \neq B \wedge line\ A\ B' \neq line\ A'\ B \wedge$
$B \neq C' \wedge B' \neq C \wedge line\ B\ C' \neq line\ B'\ C \wedge$
$A \neq C' \wedge A' \neq C \wedge line\ A\ C' \neq line\ A'\ C))$
$\longrightarrow col\ A\ B\ C \longrightarrow col\ A'\ B'\ C' \longrightarrow$ *is-a-intersec* $P\ A\ B'\ A'\ B$
$\longrightarrow$ *is-a-intersec* $Q\ B\ C'\ B'\ C \longrightarrow$ *is-a-intersec* $R\ A\ C'\ A'\ C$
$\longrightarrow col\ P\ Q\ R$

lemma *is-a-proper-intersec-is-a-intersec*:
assumes *is-a-proper-intersec P A B C D*
shows *is-a-intersec P A B C D*
⟨*proof*⟩

lemma *pappus21*:
assumes *is-pappus2* $A\ B\ C\ A'\ B'\ C'\ P\ Q\ R$
shows *is-pappus1* $A\ B\ C\ A'\ B'\ C'\ P\ Q\ R$
⟨*proof*⟩

lemma *col-AAB*: *col A A B*
⟨*proof*⟩

lemma *col-ABA*: *col A B A*
⟨*proof*⟩

lemma *col-ABB*: *col A B B*
⟨*proof*⟩

lemma *incidA-lAB*: *incid A* (*line A B*)
⟨*proof*⟩

lemma *incidB-lAB*: *incid B* (*line A B*)
⟨*proof*⟩

lemma *degenerate-hexagon-is-pappus*:
assumes $distinct\ [A,B,C,A',B',C']$ **and** *col A B C* **and** $col\ A'\ B'\ C'$ **and**
is-a-intersec $P\ A\ B'\ A'\ B$ **and** *is-a-intersec* $Q\ B\ C'\ B'\ C$ **and** *is-a-intersec* $R\ A\ C'\ A'\ C$
and $line\ A\ B' = line\ A'\ B \vee line\ B\ C' = line\ B'\ C \vee line\ A\ C' = line\ A'\ C$
shows *col P Q R*
⟨*proof*⟩

lemma *pappus12*:
  assumes *is-pappus1* $A\ B\ C\ A'\ B'\ C'\ P\ Q\ R$
  shows *is-pappus2* $A\ B\ C\ A'\ B'\ C'\ P\ Q\ R$
⟨*proof*⟩

lemma *pappus-equiv*: *is-pappus1* $A\ B\ C\ A'\ B'\ C'\ P\ Q\ R$ = *is-pappus2* $A\ B\ C\ A'\ B'\ C'\ P\ Q\ R$
  ⟨*proof*⟩

definition *is-pappus* :: *bool* **where**
is-pappus ≡ ∀ $A\ B\ C\ D\ E\ F\ P\ Q\ R$. *is-pappus2* $A\ B\ C\ D\ E\ F\ P\ Q\ R$

end

end
theory *Pascal-Property*
  imports *Main Projective-Plane-Axioms Pappus-Property*
begin

Contents:

- A hexagon is pascal if its three opposite sides meet in collinear points *is-pascal*.
- A plane is pascal, or has Pascal's property, if for every hexagon of that plane Pascal property is stable under any permutation of that hexagon.

3 Pascal's Property

context *projective-plane*
begin

definition *inters* :: *'line* ⇒ *'line* ⇒ *'point set* **where**
inters l m ≡ {*P*. *incid P l* ∧ *incid P m*}

lemma *inters-is-singleton*:
  assumes $l \neq m$ **and** $P \in$ *inters l m* **and** $Q \in$ *inters l m*
  shows $P = Q$
  ⟨*proof*⟩

definition *inter* :: *'line* ⇒ *'line* ⇒ *'point* **where**
inter l m ≡ @*P*. $P \in$ *inters l m*

lemma *uniq-inter*:

assumes $l \neq m$ **and** *incid P l* **and** *incid P m*
shows *inter l m* $= P$
⟨*proof*⟩

definition *is-pascal* :: $['point, 'point, 'point, 'point, 'point, 'point] \Rightarrow bool$ **where**
is-pascal A B C D E F $\equiv$ *distinct* $[A,B,C,D,E,F] \longrightarrow$ *line B C* $\neq$ *line E F* $\longrightarrow$
line C D $\neq$ *line A F*
$\longrightarrow$ *line A B* $\neq$ *line D E* $\longrightarrow$
(*let P* = *inter* (*line B C*) (*line E F*) *in*
let Q = *inter* (*line C D*) (*line A F*) *in*
let R = *inter* (*line A B*) (*line D E*) *in*
col P Q R)

lemma *col-rot-CW*:
assumes *col P Q R*
shows *col R P Q*
⟨*proof*⟩

lemma *col-2cycle*:
assumes *col P Q R*
shows *col P R Q*
⟨*proof*⟩

lemma *distinct6-rot-CW*:
assumes *distinct* $[A,B,C,D,E,F]$
shows *distinct* $[F,A,B,C,D,E]$
⟨*proof*⟩

lemma *lines-comm*: *lines P Q* = *lines Q P*
⟨*proof*⟩

lemma *line-comm*:
assumes $P \neq Q$
shows *line P Q* = *line Q P*
⟨*proof*⟩

lemma *inters-comm*: *inters l m* = *inters m l*
⟨*proof*⟩

lemma *inter-comm*: *inter l m* = *inter m l*
⟨*proof*⟩

lemma *inter-line-line-comm*:
assumes $C \neq D$
shows *inter* (*line A B*) (*line C D*) = *inter* (*line A B*) (*line D C*)
⟨*proof*⟩

lemma *inter-line-comm-line*:

assumes $A \neq B$
shows *inter* (*line A B*) (*line C D*) = *inter* (*line B A*) (*line C D*)
⟨*proof*⟩

lemma *inter-comm-line-line-comm*:
assumes $C \neq D$ **and** *line A B* $\neq$ *line C D*
shows *inter* (*line A B*) (*line C D*) = *inter* (*line D C*) (*line A B*)
⟨*proof*⟩

lemma *is-pascal-rot-CW*:
assumes *is-pascal A B C D E F*
shows *is-pascal F A B C D E*
⟨*proof*⟩

lemma *incid-C-AB*:
assumes $A \neq B$ **and** *incid A l* **and** *incid B l* **and** *incid C l*
shows *incid C* (*line A B*)
⟨*proof*⟩

lemma *incid-inters-left*:
assumes $P \in$ *inters l m*
shows *incid P l*
⟨*proof*⟩

lemma *incid-inters-right*:
assumes $P \in$ *inters l m*
shows *incid P m*
⟨*proof*⟩

lemma *inter-in-inters*: *inter l m* $\in$ *inters l m*
⟨*proof*⟩

lemma *incid-inter-left*: *incid* (*inter l m*) *l*
⟨*proof*⟩

lemma *incid-inter-right*: *incid* (*inter l m*) *m*
⟨*proof*⟩

lemma *col-A-B-ABl*: *col A B* (*inter* (*line A B*) *l*)
⟨*proof*⟩

lemma *col-A-B-lAB*: *col A B* (*inter l* (*line A B*))
⟨*proof*⟩

lemma *inter-is-a-intersec*: *is-a-intersec* (*inter* (*line A B*) (*line C D*)) *A B C D*
⟨*proof*⟩

definition *line-ext* :: *'line* ⇒ *'point set* **where**
line-ext l ≡ {*P*. *incid P l*}

lemma *line-left-inter-1*:
assumes $P \in$ *line-ext l* **and** $P \notin$ *line-ext m*
shows *line* (*inter l m*) $P = l$
⟨*proof*⟩

lemma *line-left-inter-2*:
assumes $P \in$ *line-ext m* **and** $P \notin$ *line-ext l*
shows *line* (*inter l m*) $P = m$
⟨*proof*⟩

lemma *line-right-inter-1*:
assumes $P \in$ *line-ext l* **and** $P \notin$ *line-ext m*
shows *line P* (*inter l m*) $= l$
⟨*proof*⟩

lemma *line-right-inter-2*:
assumes $P \in$ *line-ext m* **and** $P \notin$ *line-ext l*
shows *line P* (*inter l m*) $= m$
⟨*proof*⟩

lemma *inter-ABC-1*:
assumes *line A B* ≠ *line C A*
shows *inter* (*line A B*) (*line C A*) $= A$
⟨*proof*⟩

lemma *line-inter-2*:
assumes *inter l m* ≠ *inter l' m*
shows *line* (*inter l m*) (*inter l' m*) $= m$
⟨*proof*⟩

lemma *col-line-ext-1*:
assumes *col A B C* **and** $A \neq C$
shows $B \in$ *line-ext* (*line A C*)
⟨*proof*⟩

lemma *inter-line-ext-1*:
assumes *inter l m* ∈ *line-ext n* **and** $l \neq m$ **and** $l \neq n$
shows *inter l m* = *inter l n*
⟨*proof*⟩

lemma *inter-line-ext-2*:
assumes *inter l m* ∈ *line-ext n* **and** $l \neq m$ **and** $m \neq n$
shows *inter l m* = *inter m n*

⟨*proof*⟩

definition *pascal-prop* :: *bool* **where**
pascal-prop ≡ ∀ *A B C D E F*. *is-pascal A B C D E F* ⟶ *is-pascal B A C D E F*

lemma *pappus-pascal*:
assumes *is-pappus*
shows *pascal-prop*
⟨*proof*⟩

lemma *is-pascal-under-alternate-vertices*:
assumes *pascal-prop* **and** *is-pascal A B C A′ B′ C′*
shows *is-pascal A B′ C A′ B C′*
⟨*proof*⟩

lemma *col-inter*:
assumes *distinct* [*A*,*B*,*C*,*D*,*E*,*F*] **and** *col A B C* **and** *col D E F*
shows *inter* (*line B C*) (*line E F*) = *inter* (*line A B*) (*line D E*)
⟨*proof*⟩

lemma *pascal-pappus1*:
assumes *pascal-prop*
shows *is-pappus1 A B C A′ B′ C′ P Q R*
⟨*proof*⟩

lemma *pascal-pappus*:
assumes *pascal-prop*
shows *is-pappus*
⟨*proof*⟩

theorem *pappus-iff-pascal*: *is-pappus* = *pascal-prop*
⟨*proof*⟩

end

end
theory *Desargues-Property*
imports *Main Projective-Plane-Axioms Pappus-Property Pascal-Property*
begin

Contents:

- We formalize Desargues's property, *desargues-prop*, that states that if two triangles are perspective from a point, then they are perspective from a line. Note that some planes satisfy that property and some others don't, hence Desargues's property is not a theorem though it is a theorem in projective space geometry.

4 Desargues's Property

context *projective-plane*
begin

lemma *distinct3-def*:
distinct $[A, B, C] = (A \neq B \wedge A \neq C \wedge B \neq C)$
⟨*proof*⟩

definition *triangle* :: [′*point*, ′*point*, ′*point*] ⇒ *bool* **where**
triangle A B C ≡ *distinct* [*A*,*B*,*C*] ∧ (*line A B* ≠ *line A C*)

definition *meet-in* :: ′*line* ⇒ ′*line* => ′*point* => *bool* **where**
meet-in l m P ≡ *incid P l* ∧ *incid P m*

lemma *meet-col-1*:
assumes *meet-in* (*line A B*) (*line C D*) *P*
shows *col A B P*
⟨*proof*⟩

lemma *meet-col-2*:
assumes *meet-in* (*line A B*) (*line C D*) *P*
shows *col C D P*
⟨*proof*⟩

definition *meet-3-in* :: [′*line*, ′*line*, ′*line*, ′*point*] ⇒ *bool* **where**
meet-3-in l m n P ≡ *meet-in l m P* ∧ *meet-in l n P*

lemma *meet-all-3*:
assumes *meet-3-in l m n P*
shows *meet-in m n P*
⟨*proof*⟩

lemma *meet-comm*:
assumes *meet-in l m P*
shows *meet-in m l P*
⟨*proof*⟩

lemma *meet-3-col-1*:
assumes *meet-3-in* (*line A B*) *m n P*
shows *col A B P*
⟨*proof*⟩

lemma *meet-3-col-2*:
assumes *meet-3-in l* (*line A B*) *n P*
shows *col A B P*
⟨*proof*⟩

lemma *meet-3-col-3*:

assumes *meet-3-in l m* (*line A B*) *P*
shows *col A B P*
⟨*proof*⟩

lemma *distinct7-def*: *distinct* $[A,B,C,D,E,F,G] = ((A \neq B) \wedge (A \neq C) \wedge (A \neq D) \wedge (A \neq E) \wedge (A \neq F) \wedge (A \neq G) \wedge$
$(B \neq C) \wedge (B \neq D) \wedge (B \neq E) \wedge (B \neq F) \wedge (B \neq G) \wedge$
$(C \neq D) \wedge (C \neq E) \wedge (C \neq F) \wedge (C \neq G) \wedge$
$(D \neq E) \wedge (D \neq F) \wedge (D \neq G) \wedge$
$(E \neq F) \wedge (E \neq G) \wedge$
$(F \neq G))$
⟨*proof*⟩

definition *desargues-config* ::
[*'point*, *'point*, *'point*, *'point*, *'point*, *'point*, *'point*, *'point*, *'point*, *'point*] => *bool*
where
desargues-config $A\ B\ C\ A'\ B'\ C'\ M\ N\ P\ R \equiv$ *distinct* $[A,B,C,A',B',C',R] \wedge \neg$ *col* $A\ B\ C$
$\wedge \neg$ *col* $A'\ B'\ C' \wedge$ *distinct* [(*line* $A\ A'$),(*line* $B\ B'$),(*line* $C\ C'$)] $\wedge$
meet-3-in (*line* $A\ A'$) (*line* $B\ B'$) (*line* $C\ C'$) $R \wedge$ (*line* $A\ B$) $\neq$ (*line* $A'\ B'$) $\wedge$
(*line* $B\ C$) $\neq$ (*line* $B'\ C'$) $\wedge$ (*line* $A\ C$) $\neq$ (*line* $A'\ C'$) $\wedge$ *meet-in* (*line* $B\ C$) (*line* $B'\ C'$) $M \wedge$
meet-in (*line* $A\ C$) (*line* $A'\ C'$) $N \wedge$ *meet-in* (*line* $A\ B$) (*line* $A'\ B'$) P

lemma *distinct7-rot-CW*:
assumes *distinct* $[A,B,C,D,E,F,G]$
shows *distinct* $[C,A,B,F,D,E,G]$
⟨*proof*⟩

lemma *desargues-config-rot-CW*:
assumes *desargues-config* $A\ B\ C\ A'\ B'\ C'\ M\ N\ P\ R$
shows *desargues-config* $C\ A\ B\ C'\ A'\ B'\ P\ M\ N\ R$
⟨*proof*⟩

lemma *desargues-config-rot-CCW*:
assumes *desargues-config* $A\ B\ C\ A'\ B'\ C'\ M\ N\ P\ R$
shows *desargues-config* $B\ C\ A\ B'\ C'\ A'\ N\ P\ M\ R$
⟨*proof*⟩

definition *are-perspective-from-point* ::
[*'point*, *'point*, *'point*, *'point*, *'point*, *'point*, *'point*] ⇒ *bool* **where**
are-perspective-from-point $A\ B\ C\ A'\ B'\ C'\ R \equiv$ *distinct* $[A,B,C,A',B',C',R] \wedge$
triangle $A\ B\ C \wedge$

triangle A' B' C' ∧ *distinct* [(*line* A A'),(*line* B B'),(*line* C C')] ∧
meet-3-in (*line* A A') (*line* B B') (*line* C C') R

definition *are-perspective-from-line* ::
['*point*, '*point*, '*point*, '*point*, '*point*, '*point*] ⇒ *bool* **where**
are-perspective-from-line A B C A' B' C' ≡ *distinct* [A,B,C,A',B',C'] ⟶ *triangle* A B C ⟶
triangle A' B' C' ⟶ *line* A B ≠ *line* A' B' ⟶ *line* A C ≠ *line* A' C' ⟶ *line* B C ≠ *line* B' C' ⟶
col (*inter* (*line* A B) (*line* A' B')) (*inter* (*line* A C) (*line* A' C')) (*inter* (*line* B C) (*line* B' C'))

lemma *meet-in-inter*:
assumes $l \neq m$
shows *meet-in* l m (*inter* l m)
⟨*proof*⟩

lemma *perspective-from-point-desargues-config*:
assumes *are-perspective-from-point* A B C A' B' C' R **and** *line* A B ≠ *line* A' B' **and**
line A C ≠ *line* A' C' **and** *line* B C ≠ *line* B' C'
shows *desargues-config* A B C A' B' C' (*inter* (*line* B C) (*line* B' C')) (*inter* (*line* A C) (*line* A' C'))
(*inter* (*line* A B) (*line* A' B')) R
⟨*proof*⟩

definition *desargues-prop* :: *bool* **where**
desargues-prop ≡
∀ A B C A' B' C' P.
are-perspective-from-point A B C A' B' C' P ⟶ *are-perspective-from-line* A B C A' B' C'

end

end
theory *Pappus-Desargues*
imports *Main Projective-Plane-Axioms Pappus-Property Pascal-Property Desargues-Property*
begin

Contents:

- We prove Hessenberg's theorem *hessenberg-theorem*: Pappus's property implies Desargues's property in a projective plane.

5 Hessenberg's Theorem

context *projective-plane*

begin

lemma *col-ABC-ABD-1*:
assumes $A \neq B$ **and** *col* $A\ B\ C$ **and** *col* $A\ B\ D$
shows *col* $B\ C\ D$
⟨*proof*⟩

lemma *col-ABC-ABD-2*:
assumes $A \neq B$ **and** *col* $A\ B\ C$ **and** *col* $A\ B\ D$
shows *col* $A\ C\ D$
⟨*proof*⟩

lemma *col-line-eq-1*:
assumes $A \neq B$ **and** $B \neq C$**and** *col* $A\ B\ C$
shows *line* $A\ B$ = *line* $B\ C$
⟨*proof*⟩

lemma *col-line-eq-2*:
assumes $A \neq B$ **and** $A \neq C$ **and** *col* $A\ B\ C$
shows *line* $A\ B$ = *line* $A\ C$
⟨*proof*⟩

lemma *desargues-config-not-col-1*:
assumes *desargues-config* $A\ B\ C\ A'\ B'\ C'\ M\ N\ P\ R$
shows ¬ *col* $A\ A'\ B'$
⟨*proof*⟩

lemma *desargues-config-not-col-2*:
assumes *desargues-config* $A\ B\ C\ A'\ B'\ C'\ M\ N\ P\ R$
shows ¬ *col* $B\ B'\ C'$
⟨*proof*⟩

lemma *desargues-config-not-col-3*:
assumes *desargues-config* $A\ B\ C\ A'\ B'\ C'\ M\ N\ P\ R$
shows ¬ *col* $C\ C'\ B'$
⟨*proof*⟩

lemma *desargues-config-not-col-4*:
assumes *desargues-config* $A\ B\ C\ A'\ B'\ C'\ M\ N\ P\ R$
shows ¬ *col* $A\ A'\ C'$
⟨*proof*⟩

lemma *desargues-config-not-col-5*:
assumes *desargues-config* $A\ B\ C\ A'\ B'\ C'\ M\ N\ P\ R$
shows ¬ *col* $B\ B'\ A'$
⟨*proof*⟩

lemma *desargues-config-not-col-6*:
assumes *desargues-config* $A\ B\ C\ A'\ B'\ C'\ M\ N\ P\ R$

shows ¬ *col* C C' A'
⟨*proof*⟩

lemma *desargues-config-not-col-7*:
assumes *desargues-config* A B C A' B' C' M N P R
shows ¬ *col* A B B'
⟨*proof*⟩

lemma *desargues-config-not-col-8*:
assumes *desargues-config* A B C A' B' C' M N P R
shows ¬ *col* A C C'

⟨*proof*⟩

lemma *desargues-config-not-col-9*:
assumes *desargues-config* A B C A' B' C' M N P R
shows ¬ *col* B A A'
⟨*proof*⟩

lemma *desargues-config-not-col-10*:
assumes *desargues-config* A B C A' B' C' M N P R
shows ¬ *col* B C C'
⟨*proof*⟩

lemma *desargues-config-not-col-11*:
assumes *desargues-config* A B C A' B' C' M N P R
shows ¬ *col* C A A'
⟨*proof*⟩

lemma *desargues-config-not-col-12*:
assumes *desargues-config* A B C A' B' C' M N P R
shows ¬ *col* C B B'
⟨*proof*⟩

lemma *col-inter*:
assumes $A \neq C$ **and** $B \neq C$ **and** *col* A B C
shows *inter* l (*line* B C) = *inter* l (*line* A C)
⟨*proof*⟩

lemma *lemma-1*:
assumes *desargues-config* A B C A' B' C' M N P R **and** *is-pappus*
shows *col* M N P ∨ *incid* A (*line* B' C') ∨ *incid* C' (*line* A B)
⟨*proof*⟩

corollary *corollary-1*:
assumes *desargues-config* A B C A' B' C' M N P R **and** *is-pappus*
shows *col* M N P ∨ ((*incid* A (*line* B' C') ∨ *incid* C' (*line* A B)) ∧
(*incid* C (*line* A' B') ∨ *incid* B' (*line* A C)) ∧ (*incid* B (*line* A' C') ∨ *incid* A'

(*line B C*)))
  ⟨*proof*⟩

definition *triangle-circumscribes-triangle* ::
  [*'point*, *'point*, *'point*, *'point*, *'point*, *'point*] ⇒ *bool* **where**
triangle-circumscribes-triangle A' B' C' A B C ≡ *incid* A (*line* B' C') ∧ *incid* C (*line* A' B') ∧
incid B (*line* A' C')

lemma *lemma-2*:
  assumes *desargues-config* A B C A' B' C' M N P R **and** *incid* A (*line* B' C') ∨ *incid* C' (*line* A B)
    and *incid* C (*line* A' B') ∨ *incid* B' (*line* A C) **and** *incid* B (*line* A' C') ∨ *incid* A' (*line* B C)
  shows *col* M N P ∨ *triangle-circumscribes-triangle* A B C A' B' C' ∨ *triangle-circumscribes-triangle* A' B' C' A B C
  ⟨*proof*⟩

lemma *lemma-3*:
  assumes *is-pappus* **and** *desargues-config* A B C A' B' C' M N P R **and**
    triangle-circumscribes-triangle A' B' C' A B C
  shows *col* M N P
⟨*proof*⟩

theorem *pappus-desargues*:
  assumes *is-pappus* **and** *desargues-config* A B C A' B' C' M N P R
  shows *col* M N P
⟨*proof*⟩

theorem *hessenberg-thereom*:
  assumes *is-pappus*
  shows *desargues-prop*
  ⟨*proof*⟩

corollary *pascal-desargues*:
  assumes *pascal-prop*
  shows *desargues-prop*
  ⟨*proof*⟩

end

end
theory *Higher-Projective-Space-Rank-Axioms*
  imports *Main*
begin

Contents:

- Following [2] we introduce a set of axioms for projective space geometry based on the notions of matroid and rank.

6 A Based-rank Set of Axioms for Projective Space Geometry

locale *higher-projective-space-rank* =

fixes *rk* :: *'point set* ⇒ *nat*

assumes
matroid-ax-1a: $rk\ X \geq 0$ **and**
matroid-ax-1b: $rk\ X \leq card\ X$ **and**
matroid-ax-2: $X \subseteq Y \longrightarrow rk\ X \leq rk\ Y$ **and**
matroid-ax-3: $rk\ (X \cup Y) + rk\ (X \cap Y) \leq rk\ X + rk\ Y$

assumes
rk-ax-singleton: $rk\ \{P\} \geq 1$ **and**
rk-ax-couple: $P \neq Q \longrightarrow rk\ \{P,Q\} \geq 2$ **and**
rk-ax-pasch: $rk\ \{A,B,C,D\} \leq 3 \longrightarrow (\exists\, J.\ rk\ \{A,B,J\} = 2 \wedge rk\ \{C,D,J\} = 2)$ **and**
rk-ax-3-pts: $\exists\, C.\ rk\ \{A,B,C\} = 2 \wedge rk\ \{B,C\} = 2 \wedge rk\ \{A,C\} = 2$ **and**
rk-ax-dim: $\exists\, A\ B\ C\ D.\ rk\ \{A,B,C,D\} \geq 4$

end
theory *Matroid-Rank-Properties*
  imports *Main Higher-Projective-Space-Rank-Axioms*
begin

Contents:

- In this file we introduce the basic lemmas and properties derived from our based-rank axioms that will allow us to simplify our future proofs.

7 Proof Techniques Using Ranks

context *higher-projective-space-rank*
begin

lemma *matroid-ax-3-alt*:
  assumes $I \subseteq X \cap Y$
  shows $rk\ (X \cup Y) + rk\ I \leq rk\ X + rk\ Y$
  ⟨*proof*⟩

lemma *rk-uniqueness*:
assumes $rk\ \{A,B\} = 2$ **and** $rk\ \{C,D\} = 2$ **and** $rk\ \{A,B,M\} \leq 2$ **and** $rk\ \{C,D,M\} \leq 2$ **and**
$rk\ \{A,B,P\} \leq 2$ **and** $rk\ \{C,D,P\} \leq 2$ **and** $rk\ \{A,B,C,D\} \geq 3$
shows $rk\ \{M,P\} = 1$
⟨*proof*⟩

lemma *rk-ax-dim-alt*: $\exists A\ B\ C\ D.\ \forall M.\ rk\ \{A,B,M\} \neq 2 \vee rk\ \{C,D,M\} \neq 2$
⟨*proof*⟩

lemma *rk-empty*: $rk\ \{\} = 0$
⟨*proof*⟩

lemma *matroid-ax-2-alt*: $rk\ X \leq rk\ (X \cup \{x\}) \wedge rk\ (X \cup \{x\}) \leq rk\ X + 1$
⟨*proof*⟩

lemma *matroid-ax-3-alt'*: $rk\ (X \cup \{y\}) = rk\ (X \cup \{z\}) \longrightarrow rk\ (X \cup \{z\}) = rk\ X \longrightarrow rk\ X = rk\ (X \cup \{y,z\})$
⟨*proof*⟩

lemma *rk-ext*:
assumes $rk\ X \leq 3$
shows $\exists P.\ rk(X \cup \{P\}) = rk\ X + 1$
⟨*proof*⟩

lemma *rk-singleton* : $\forall P.\ rk\ \{P\} = 1$
⟨*proof*⟩

lemma *rk-singleton-bis* :
assumes $A = B$
shows $rk\ \{A,\ B\} = 1$
⟨*proof*⟩

lemma *rk-couple* :
assumes $A \neq B$
shows $rk\ \{A,\ B\} = 2$
⟨*proof*⟩

lemma *rk-triple-le* : $rk\ \{A,\ B,\ C\} \leq 3$
⟨*proof*⟩

lemma *rk-couple-to-singleton* :
assumes $rk\ \{A,\ B\} = 1$
shows $A = B$
⟨*proof*⟩

lemma *rk-triple-to-rk-couple* :
assumes $rk\ \{A,\ B,\ C\} = 3$

shows rk $\{A,\ B\} = 2$
$\langle proof \rangle$

end

end
theory *Desargues-2D*
imports *Main Higher-Projective-Space-Rank-Axioms Matroid-Rank-Properties*
begin

Contents:

- We prove Desargues's theorem: if two triangles ABC and A'B'C' are perspective from a point P (ie. the lines AA', BB' and CC' are concurrent in P), then they are perspective from a line (ie. the points $\alpha = BC \cap B'C'$, $\beta = AC \cap A'C'$ and $\gamma = AB \cap A'B'$ are collinear). In this file we restrict ourself to the case where the two triangles ABC and A'B'C' are coplanar.

8 Desargues's Theorem: The Coplanar Case

context *higher-projective-space-rank*
begin

definition *desargues-config-2D* ::
$['point,\ 'point,\ 'point,\ 'point,\ 'point,\ 'point,\ 'point,\ 'point,\ 'point,\ 'point] \Rightarrow bool$
where $desargues\text{-}config\text{-}2D\ A\ B\ C\ A'\ B'\ C'\ P\ \alpha\ \beta\ \gamma \equiv rk\ \{A,\ B,\ C\} = 3 \wedge rk\ \{A',\ B',\ C'\} = 3 \wedge$
$rk\ \{A,\ A',\ P\} = 2 \wedge rk\ \{B,\ B',\ P\} = 2 \wedge rk\ \{C,\ C',\ P\} = 2 \wedge rk\ \{A,\ B,\ \gamma\} = 2 \wedge rk\ \{A',\ B',\ \gamma\} = 2 \wedge$
$rk\ \{A,\ C,\ \beta\} = 2 \wedge rk\ \{A',\ C',\ \beta\} = 2 \wedge rk\ \{B,\ C,\ \alpha\} = 2 \wedge rk\ \{B',\ C',\ \alpha\} = 2 \wedge$
$rk\ \{A,\ B,\ C,\ A',\ B',\ C'\} = 3 \wedge$
— We add the following non-degeneracy conditions
$rk\ \{A,\ B,\ P\} = 3 \wedge rk\ \{A,\ C,\ P\} = 3 \wedge rk\ \{B,\ C,\ P\} = 3 \wedge$
$rk\ \{A,\ A'\} = 2 \wedge rk\ \{B,\ B'\} = 2 \wedge rk\ \{C,\ C'\} = 2$

lemma *coplanar-ABCA'B'C'P* :
assumes $rk\ \{A,\ A'\} = 2$ **and** $rk\ \{A,\ B,\ C,\ A',\ B',\ C'\} = 3$ **and** $rk\ \{A,\ A',\ P\} = 2$
shows $rk\ \{A,\ B,\ C,\ A',\ B',\ C',\ P\} = 3$
$\langle proof \rangle$

lemma *non-colinear-A'B'P* :
assumes $rk\ \{A,\ B,\ P\} = 3$ **and** $rk\ \{A,\ A',\ P\} = 2$ **and** $rk\ \{B,\ B',\ P\} = 2$ **and** $rk\ \{A',\ P\} = 2$

and rk $\{B', P\} = 2$
shows rk $\{A', B', P\} = 3$
⟨*proof*⟩

lemma *desargues-config-2D-non-collinear-P* :
assumes *desargues-config-2D* $A\ B\ C\ A'\ B'\ C'\ P\ \alpha\ \beta\ \gamma$ **and** rk $\{A', P\} = 2$ **and** rk $\{B', P\} = 2$
and rk $\{C', P\} = 2$
shows rk $\{A', B', P\} = 3$ **and** rk $\{A', C', P\} = 3$ **and** rk $\{B', C', P\} = 3$
⟨*proof*⟩

lemma *rk-A′B′PQ* :
assumes rk $\{A, A'\} = 2$ **and** rk $\{A, B, C, A', B', C'\} = 3$ **and** rk $\{A, A', P\} = 2$ **and**
rk $\{A, B, P\} = 3$ **and** rk $\{B, B', P\} = 2$ **and** rk $\{A', P\} = 2$ **and** rk $\{B', P\} = 2$ **and**
rk $\{A, B, C, A', B', C', P, Q\} \geq 4$
shows rk $\{A', B', P, Q\} = 4$
⟨*proof*⟩

lemma *desargues-config-2D-rkA′B′PQ-rkA′C′PQ-rkB′C′PQ* :
assumes *desargues-config-2D* $A\ B\ C\ A'\ B'\ C'\ P\ \alpha\ \beta\ \gamma$ **and** rk $\{A', P\} = 2$ **and** rk $\{B', P\} = 2$
and rk $\{C', P\} = 2$ **and** rk $\{A, B, C, A', B', C', P, Q\} \geq 4$
shows rk $\{A', B', P, Q\} = 4$ **and** rk $\{A', C', P, Q\} = 4$ **and** rk $\{B', C', P, Q\} = 4$
⟨*proof*⟩

lemma *rk-A′B′PR* :
assumes rk $\{P, Q, R\} = 2$ **and** rk $\{P, R\} = 2$ **and** rk $\{A', B', P, Q\} = 4$
shows rk $\{A', B', P, R\} = 4$
⟨*proof*⟩

lemma *rk-A′C′PR* :
assumes rk $\{P, Q, R\} = 2$ **and** rk $\{P, R\} = 2$ **and** rk $\{A', C', P, Q\} = 4$
shows rk $\{A', C', P, R\} = 4$
⟨*proof*⟩

lemma *rk-B′C′PR* :
assumes rk $\{P, Q, R\} = 2$ **and** rk $\{P, R\} = 2$ **and** rk $\{B', C', P, Q\} = 4$
shows rk $\{B', C', P, R\} = 4$
⟨*proof*⟩

lemma *rk-ABA′* :
assumes rk $\{A, B, P\} = 3$ **and** rk $\{A, A'\} = 2$ **and** rk $\{A, A', P\} = 2$
shows rk $\{A, B, A'\} = 3$
⟨*proof*⟩

lemma *desargues-config-2D-non-collinear* :

assumes *desargues-config-2D A B C A′ B′ C′ P α β γ*
shows *rk* {*A*, *B*, *A′*} = *3* **and** *rk* {*A*, *B*, *B′*} = *3* **and** *rk* {*A*, *C*, *C′*} = *3*
⟨*proof*⟩

lemma *rk-Aa* :
assumes *rk* {*A*, *B*, *P*} = *3* **and** *rk* {*A*, *A′*} = *2* **and** *rk* {*A*, *A′*, *P*} = *2* **and** *rk* {*Q*, *A′*, *a*} = *2*
and *rk* {*A*, *B*, *C*, *A′*, *B′*, *C′*, *P*, *Q*} ≥ *4* **and** *rk* {*A*, *B*, *C*, *A′*, *B′*, *C′*} ≤ *3*
shows *rk* {*A*, *a*} = *2*
⟨*proof*⟩

lemma *desargues-config-2D-rkAa-rkBb-rkCc* :
assumes *desargues-config-2D A B C A′ B′ C′ P α β γ* **and** *rk* {*A*, *B*, *C*, *A′*, *B′*, *C′*, *P*, *Q*} ≥ *4*
and *rk* {*Q*, *A′*, *a*} = *2* **and** *rk* {*Q*, *B′*, *b*} = *2* **and** *rk* {*Q*, *C′*, *c*} = *2*
shows *rk* {*A*, *a*} = *2* **and** *rk* {*B*, *b*} = *2* **and** *rk* {*C*, *c*} = *2*
⟨*proof*⟩

lemma *rk-ABPRa* :
assumes *rk* {*A*, *B*, *P*} = *3* **and** *rk* {*A*, *B*, *C*, *A′*, *B′*, *C′*, *P*} = *3* **and** *rk* {*P*, *Q*, *R*} = *2*
and *rk* {*P*, *R*} = *2* **and** *rk* {*A′*, *B′*, *P*, *Q*} = *4*
shows *rk* {*A*, *B*, *P*, *R*, *a*} ≥ *4*
⟨*proof*⟩

lemma *rk-ABPa* :
assumes *rk* {*A*, *B*, *P*} = *3* **and** *rk* {*A*, *A′*} = *2* **and** *rk* {*A*, *A′*, *P*} = *2* **and** *rk* {*Q*, *A′*, *a*} = *2*
and *rk* {*A*, *B*, *C*, *A′*, *B′*, *C′*, *P*, *Q*} ≥ *4* **and** *rk* {*A*, *B*, *C*, *A′*, *B′*, *C′*, *P*} = *3*
and *rk* {*P*, *Q*, *R*} = *2*
and *rk* {*P*, *R*} = *2* **and** *rk* {*A′*, *B′*, *P*, *Q*} = *4* **and** *rk* {*R*, *A*, *a*} = *2*
shows *rk* {*A*, *B*, *P*, *a*} ≥ *4*
⟨*proof*⟩

lemma *desargues-config-2D-rkABPa-rkABPb-rkABPc* :
assumes *desargues-config-2D A B C A′ B′ C′ P α β γ* **and** *rk* {*A*, *B*, *C*, *A′*, *B′*, *C′*, *P*, *Q*} ≥ *4*
and *rk* {*P*, *Q*, *R*} = *2* **and** *rk* {*P*, *R*} = *2* **and** *rk* {*A′*, *P*} = *2* **and** *rk* {*B′*, *P*} = *2* **and**
rk {*Q*, *A′*, *a*} = *2* **and** *rk* {*R*, *A*, *a*} = *2* **and** *rk* {*Q*, *B′*, *b*} = *2* **and** *rk* {*R*, *B*, *b*} = *2* **and**
rk {*Q*, *C′*, *c*} = *2* **and** *rk* {*R*, *C*, *c*} = *2*
shows *rk* {*A*, *B*, *P*, *a*} ≥ *4* **and** *rk* {*A*, *B*, *P*, *b*} ≥ *4* **and** *rk* {*A*, *B*, *P*, *c*} ≥ *4*
⟨*proof*⟩

lemma *rk-AA′C* :
assumes *rk* {*A*, *C*, *P*} = *3* **and** *rk* {*A*, *A′*} = *2* **and** *rk* {*A*, *A′*, *P*} = *2*
shows *rk* {*A*, *A′*, *C*} ≥ *3*
⟨*proof*⟩

lemma *rk-AA′C′* :
assumes $rk\ \{A', C', P\} = 3$ **and** $rk\ \{A, A'\} = 2$ **and** $rk\ \{A, A', P\} = 2$
shows $rk\ \{A, A', C'\} \geq 3$
⟨*proof*⟩

lemma *rk-AA′Ca* :
assumes $rk\ \{A, A', C\} \geq 3$ **and** $rk\ \{A, B, P, a\} \geq 4$ **and** $rk\ \{A, B, C, A', B', C', P\} = 3$
shows $rk\ \{A, A', C, a\} \geq 4$
⟨*proof*⟩

lemma *rk-AA′C′a* :
assumes $rk\ \{A, A', C'\} \geq 3$ **and** $rk\ \{A, B, P, a\} \geq 4$ **and** $rk\ \{A, B, C, A', B', C', P\} = 3$
shows $rk\ \{A, A', C', a\} \geq 4$
⟨*proof*⟩

lemma *rk-Ra* :
assumes $rk\ \{Q, A', a\} = 2$ **and** $rk\ \{P, Q, R\} = 2$ **and** $rk\ \{R, Q\} = 2$ **and** $rk\ \{A, A', P\} = 2$
and $rk\ \{A', P\} = 2$ **and** $rk\ \{A, B, C, A', B', C', P\} = 3$ **and** $rk\ \{A, B, A'\} = 3$ **and**
$rk\ \{A, B, C, A', B', C', P, Q\} \geq 4$
shows $rk\ \{R, a\} = 2$
⟨*proof*⟩

lemma *desargues-config-2D-rkRa-rkRb-rkRc* :
assumes *desargues-config-2D* $A\ B\ C\ A'\ B'\ C'\ P\ \alpha\ \beta\ \gamma$ **and** $rk\ \{A, B, C, A', B', C', P, Q\} \geq 4$
and $rk\ \{P, Q, R\} = 2$ **and** $rk\ \{Q, R\} = 2$ **and** $rk\ \{Q, A', a\} = 2$ **and** $rk\ \{Q, B', b\} = 2$ **and**
$rk\ \{Q, C', c\} = 2$ **and** $rk\ \{A', P\} = 2$ **and** $rk\ \{B', P\} = 2$ **and** $rk\ \{C', P\} = 2$
shows $rk\ \{R, a\} = 2$ **and** $rk\ \{R, b\} = 2$ **and** $rk\ \{R, c\} = 2$
⟨*proof*⟩

lemma *rk-acACβ* :
assumes $rk\ \{R, A, a\} = 2$ **and** $rk\ \{R, C, c\} = 2$ **and** $rk\ \{A, C\} = 2$ **and** $rk\ \{A, C, \beta\} = 2$
and $rk\ \{Q, A', a\} = 2$ **and** $rk\ \{A, A', C, a\} \geq 4$
shows $rk\ \{a, c, A, C, \beta\} = 3$
⟨*proof*⟩

lemma *rk-acA′C′β* :
assumes $rk\ \{Q, A', a\} = 2$ **and** $rk\ \{Q, C', c\} = 2$ **and** $rk\ \{A', C'\} = 2$ **and** $rk\ \{A', C', \beta\} = 2$
and $rk\ \{R, A, a\} = 2$ **and** $rk\ \{A', A, C', a\} \geq 4$
shows $rk\ \{a, c, A', C', \beta\} = 3$
⟨*proof*⟩

lemma *plane-representation-change* :
assumes $rk\ \{A, B, C, P\} = 3$ **and** $rk\ \{B, C, P\} = 3$ **and** $rk\ \{A, B, C, Q\} = 4$
shows $rk\ \{P, B, C, Q\} = 4$
⟨*proof*⟩

lemma *desargues-config-2D-rkABCP* :
assumes *desargues-config-2D* $A\ B\ C\ A'\ B'\ C'\ P\ \alpha\ \beta\ \gamma$
shows $rk\ \{A, B, C, P\} = 3$
⟨*proof*⟩

lemma *desargues-config-2D-rkABCabc* :
assumes *desargues-config-2D* $A\ B\ C\ A'\ B'\ C'\ P\ \alpha\ \beta\ \gamma$ **and** $rk\ \{A, B, C, A', B', C', P, Q\} \geq 4$
and $rk\ \{Q, A', a\} = 2$ **and** $rk\ \{P, Q, R\} = 2$ **and** $rk\ \{P, R\} = 2$ **and** $rk\ \{R, A, a\} = 2$ **and**
$rk\ \{A', P\} = 2$ **and** $rk\ \{B', P\} = 2$
shows $rk\ \{A, B, C, a, b, c\} \geq 4$
⟨*proof*⟩

lemma *rk-abc* :
assumes *desargues-config-2D* $A\ B\ C\ A'\ B'\ C'\ P\ \alpha\ \beta\ \gamma$ **and** $rk\ \{A, B, C, A', B', C', P, Q\} \geq 4$
and $rk\ \{Q, A', a\} = 2$ **and** $rk\ \{Q, B', b\} = 2$ **and** $rk\ \{Q, C', c\} = 2$ **and** $rk\ \{P, Q, R\} = 2$ **and**
$rk\ \{P, R\} = 2$ **and** $rk\ \{Q, R\} = 2$ **and** $rk\ \{R, A, a\} = 2$ **and** $rk\ \{R, B, b\} = 2$ **and**
$rk\ \{R, C, c\} = 2$ **and** $rk\ \{A', P\} = 2$ **and** $rk\ \{B', P\} = 2$ **and** $rk\ \{C', P\} = 2$
shows $rk\ \{a, b, c\} = 3$
⟨*proof*⟩

lemma *rk-acβ* :
assumes *desargues-config-2D* $A\ B\ C\ A'\ B'\ C'\ P\ \alpha\ \beta\ \gamma$ **and** $rk\ \{A, B, C, A', B', C', P, Q\} \geq 4$
and $rk\ \{Q, A', a\} = 2$ **and** $rk\ \{Q, B', b\} = 2$ **and** $rk\ \{Q, C', c\} = 2$ **and** $rk\ \{P, Q, R\} = 2$ **and**
$rk\ \{P, R\} = 2$ **and** $rk\ \{Q, R\} = 2$ **and** $rk\ \{R, A, a\} = 2$ **and** $rk\ \{R, B, b\} = 2$ **and**
$rk\ \{R, C, c\} = 2$ **and** $rk\ \{A', P\} = 2$ **and** $rk\ \{B', P\} = 2$ **and** $rk\ \{C', P\} = 2$
shows $rk\ \{a, c, \beta\} = 2$
⟨*proof*⟩

lemma *rk-abγ* :
assumes *desargues-config-2D* $A\ B\ C\ A'\ B'\ C'\ P\ \alpha\ \beta\ \gamma$ **and** $rk\ \{A, B, C, A', B', C', P, Q\} \geq 4$
and $rk\ \{Q, A', a\} = 2$ **and** $rk\ \{Q, B', b\} = 2$ **and** $rk\ \{Q, C', c\} = 2$ **and** $rk\ \{P, Q, R\} = 2$ **and**
$rk\ \{P, R\} = 2$ **and** $rk\ \{Q, R\} = 2$ **and** $rk\ \{R, A, a\} = 2$ **and** $rk\ \{R, B, b\} = 2$

and
rk {*R*, *C*, *c*} = *2* **and** *rk* {*A'*, *P*} = *2* **and** *rk* {*B'*, *P*} = *2* **and** *rk* {*C'*, *P*} = *2*
shows *rk* {*a*, *b*, *γ*} = *2*
⟨*proof*⟩

lemma *rk-bcα* :
assumes *desargues-config-2D A B C A' B' C' P α β γ* **and** *rk* {*A*, *B*, *C*, *A'*, *B'*, *C'*, *P*, *Q*} ≥ *4*
and *rk* {*Q*, *A'*, *a*} = *2* **and** *rk* {*Q*, *B'*, *b*} = *2* **and** *rk* {*Q*, *C'*, *c*} = *2* **and** *rk* {*P*, *Q*, *R*} = *2* **and**
rk {*P*, *R*} = *2* **and** *rk* {*Q*, *R*} = *2* **and** *rk* {*R*, *A*, *a*} = *2* **and** *rk* {*R*, *B*, *b*} = *2* **and**
rk {*R*, *C*, *c*} = *2* **and** *rk* {*A'*, *P*} = *2* **and** *rk* {*B'*, *P*} = *2* **and** *rk* {*C'*, *P*} = *2*
shows *rk* {*b*, *c*, *α*} = *2*
⟨*proof*⟩

lemma *rk-abcαβγ* :
assumes *desargues-config-2D A B C A' B' C' P α β γ* **and** *rk* {*A*, *B*, *C*, *A'*, *B'*, *C'*, *P*, *Q*} ≥ *4*
and *rk* {*Q*, *A'*, *a*} = *2* **and** *rk* {*Q*, *B'*, *b*} = *2* **and** *rk* {*Q*, *C'*, *c*} = *2* **and** *rk* {*P*, *Q*, *R*} = *2* **and**
rk {*P*, *R*} = *2* **and** *rk* {*Q*, *R*} = *2* **and** *rk* {*R*, *A*, *a*} = *2* **and** *rk* {*R*, *B*, *b*} = *2* **and**
rk {*R*, *C*, *c*} = *2* **and** *rk* {*A'*, *P*} = *2* **and** *rk* {*B'*, *P*} = *2* **and** *rk* {*C'*, *P*} = *2*
shows *rk* {*a*, *b*, *c*, *α*, *β*, *γ*} = *3*
⟨*proof*⟩

lemma *rk-ABCαβγ* :
assumes *desargues-config-2D A B C A' B' C' P α β γ* **and** *rk* {*A*, *B*, *C*, *A'*, *B'*, *C'*, *P*, *Q*} ≥ *4*
and *rk* {*Q*, *A'*, *a*} = *2* **and** *rk* {*Q*, *B'*, *b*} = *2* **and** *rk* {*Q*, *C'*, *c*} = *2* **and** *rk* {*P*, *Q*, *R*} = *2* **and**
rk {*P*, *R*} = *2* **and** *rk* {*Q*, *R*} = *2* **and** *rk* {*R*, *A*, *a*} = *2* **and** *rk* {*R*, *B*, *b*} = *2* **and**
rk {*R*, *C*, *c*} = *2* **and** *rk* {*A'*, *P*} = *2* **and** *rk* {*B'*, *P*} = *2* **and** *rk* {*C'*, *P*} = *2*
shows *rk* {*A*, *B*, *C*, *α*, *β*, *γ*} = *3*
⟨*proof*⟩

lemma *rk-αβγ* :
assumes *desargues-config-2D A B C A' B' C' P α β γ* **and** *rk* {*A*, *B*, *C*, *A'*, *B'*, *C'*, *P*, *Q*} ≥ *4*
and *rk* {*Q*, *A'*, *a*} = *2* **and** *rk* {*Q*, *B'*, *b*} = *2* **and** *rk* {*Q*, *C'*, *c*} = *2* **and** *rk* {*P*, *Q*, *R*} = *2* **and**
rk {*P*, *R*} = *2* **and** *rk* {*Q*, *R*} = *2* **and** *rk* {*R*, *A*, *a*} = *2* **and** *rk* {*R*, *B*, *b*} = *2* **and**
rk {*R*, *C*, *c*} = *2* **and** *rk* {*A'*, *P*} = *2* **and** *rk* {*B'*, *P*} = *2* **and** *rk* {*C'*, *P*} = *2*
shows *rk* {*α*, *β*, *γ*} ≤ *2*
⟨*proof*⟩

lemma *rk-αβγ-special-case-1* :
assumes *desargues-config-2D* $A\ B\ C\ A'\ B'\ C'\ P\ \alpha\ \beta\ \gamma$ **and** $rk\ \{A', P\} = 1$
shows $rk\ \{\alpha, \beta, \gamma\} \leq 2$
⟨*proof*⟩

lemma *rk-αβγ-special-case-2* :
assumes *desargues-config-2D* $A\ B\ C\ A'\ B'\ C'\ P\ \alpha\ \beta\ \gamma$ **and** $rk\ \{B', P\} = 1$
shows $rk\ \{\alpha, \beta, \gamma\} \leq 2$
⟨*proof*⟩

lemma *rk-αβγ-special-case-3* :
assumes *desargues-config-2D* $A\ B\ C\ A'\ B'\ C'\ P\ \alpha\ \beta\ \gamma$ **and** $rk\ \{C', P\} = 1$
shows $rk\ \{\alpha, \beta, \gamma\} \leq 2$
⟨*proof*⟩

theorem *desargues-2D* :
assumes *desargues-config-2D* $A\ B\ C\ A'\ B'\ C'\ P\ \alpha\ \beta\ \gamma$
shows $rk\ \{\alpha, \beta, \gamma\} \leq 2$
⟨*proof*⟩
end

end
theory *Desargues-3D*
imports *Main Higher-Projective-Space-Rank-Axioms Matroid-Rank-Properties*
begin

Contents:

- We prove Desargues's theorem: if two triangles ABC and A'B'C' are perspective from a point P (ie. the lines AA', BB' and CC' are concurrent in P), then they are perspective from a line (ie. the points $\alpha = BC \cap B'C'$, $\beta = AC \cap A'C'$ and $\gamma = AB \cap A'B'$ are collinear). In this file we restrict ourself to the case where the two triangles ABC and A'B'C' are not coplanar.

9 Desargues's Theorem: The Non-coplanar Case

context *higher-projective-space-rank*
begin

definition *desargues-config-3D* ::
['*point*, '*point*, '*point*, '*point*, '*point*, '*point*, '*point*, '*point*, '*point*, '*point*] => *bool*
where *desargues-config-3D* $A\ B\ C\ A'\ B'\ C'\ P\ \alpha\ \beta\ \gamma \equiv rk\ \{A, B, C\} = 3 \wedge rk\ \{A', B', C'\} = 3 \wedge$
$rk\ \{A, A', P\} = 2 \wedge rk\ \{B, B', P\} = 2 \wedge rk\ \{C, C', P\} = 2 \wedge rk\ \{A, B, C, A', B', C'\} \geq 4 \wedge$
$rk\ \{B, C, \alpha\} = 2 \wedge rk\ \{B', C', \alpha\} = 2 \wedge rk\ \{A, C, \beta\} = 2 \wedge rk\ \{A', C', \beta\} = 2 \wedge rk\ \{A, B, \gamma\} = 2 \wedge$

rk $\{A', B', \gamma\} = 2$

lemma *coplanar-4* :
assumes *rk* $\{A, B, C\} = 3$ **and** *rk* $\{B, C, \alpha\} = 2$
shows *rk* $\{A, B, C, \alpha\} = 3$
⟨*proof*⟩

lemma *desargues-config-3D-coplanar-4* :
assumes *desargues-config-3D* $A\ B\ C\ A'\ B'\ C'\ P\ \alpha\ \beta\ \gamma$
shows *rk* $\{A, B, C, \alpha\} = 3$ **and** *rk* $\{A', B', C', \alpha\} = 3$
⟨*proof*⟩

lemma *coplanar-4-bis* :
assumes *rk* $\{A, B, C\} = 3$ **and** *rk* $\{A, C, \beta\} = 2$
shows *rk* $\{A, B, C, \beta\} = 3$
⟨*proof*⟩

lemma *desargues-config-3D-coplanar-4-bis* :
assumes *desargues-config-3D* $A\ B\ C\ A'\ B'\ C'\ P\ \alpha\ \beta\ \gamma$
shows *rk* $\{A, B, C, \beta\} = 3$ **and** *rk* $\{A', B', C', \beta\} = 3$
⟨*proof*⟩

lemma *coplanar-4-ter* :
assumes *rk* $\{A, B, C\} = 3$ **and** *rk* $\{A, B, \gamma\} = 2$
shows *rk* $\{A, B, C, \gamma\} = 3$
⟨*proof*⟩

lemma *desargues-config-3D-coplanar-4-ter* :
assumes *desargues-config-3D* $A\ B\ C\ A'\ B'\ C'\ P\ \alpha\ \beta\ \gamma$
shows *rk* $\{A, B, C, \gamma\} = 3$ **and** *rk* $\{A', B', C', \gamma\} = 3$
⟨*proof*⟩

lemma *coplanar-5* :
assumes *rk* $\{A, B, C\} = 3$ **and** *rk* $\{B, C, \alpha\} = 2$ **and** *rk* $\{A, C, \beta\} = 2$
shows *rk* $\{A, B, C, \alpha, \beta\} = 3$
⟨*proof*⟩

lemma *desargues-config-3D-coplanar-5* :
assumes *desargues-config-3D* $A\ B\ C\ A'\ B'\ C'\ P\ \alpha\ \beta\ \gamma$
shows *rk* $\{A, B, C, \alpha, \beta\} = 3$ **and** *rk* $\{A', B', C', \alpha, \beta\} = 3$
⟨*proof*⟩

lemma *coplanar-5-bis* :
assumes *rk* $\{A, B, C\} = 3$ **and** *rk* $\{B, C, \alpha\} = 2$ **and** *rk* $\{A, B, \gamma\} = 2$
shows *rk* $\{A, B, C, \alpha, \gamma\} = 3$
⟨*proof*⟩

lemma *desargues-config-3D-coplanar-5-bis* :

assumes *desargues-config-3D A B C A′ B′ C′ P α β γ*
shows rk {A, B, C, α, γ} = *3* **and** rk {A', B', C', α, γ} = *3*
⟨*proof*⟩

lemma *coplanar-6* :
assumes rk {A, B, C} = *3* **and** rk {B, C, α} = *2* **and** rk {A, B, γ} = *2* **and** rk {A, C, β} = *2*
shows rk {A, B, C, α, β, γ} = *3*
⟨*proof*⟩

lemma *desargues-config-3D-coplanar-6* :
assumes *desargues-config-3D A B C A′ B′ C′ P α β γ*
shows rk {A, B, C, α, β, γ} = *3* **and** rk {A', B', C', α, β, γ} = *3*
⟨*proof*⟩

lemma *desargues-config-3D-non-coplanar* :
assumes *desargues-config-3D A B C A′ B′ C′ P α β γ*
shows rk {A, B, C, A', B', C', α, β, γ} $\geq$ *4*
⟨*proof*⟩

theorem *desargues-3D* :
assumes *desargues-config-3D A B C A′ B′ C′ P α β γ*
shows rk {α, β, γ} $\leq$ *2*
⟨*proof*⟩

end

end
theory *Projective-Space-Axioms*
imports *Main*
begin

Contents:

- We introduce the types *′point* of points and *′line* of lines and an incidence relation between them.
- A set of axioms for the (3-dimensional) projective space. An alternative set of axioms could use planes as basic objects in addition to points and lines

10 The axioms of the Projective Space

lemma *distinct4-def*:
distinct [A,B,C,D] = (($A \neq B$) ∧ ($A \neq C$) ∧ ($A \neq D$)∧ ($B \neq C$) ∧ ($B \neq D$) ∧ ($C \neq D$))
⟨*proof*⟩

lemma *distinct3-def*:
distinct $[A,\ B,\ C] = (A \neq B \wedge A \neq C \wedge B \neq C)$
⟨*proof*⟩

locale *projective-space* =

fixes *incid* :: *'point* ⇒ *'line* ⇒ *bool*
fixes *meet* :: *'line* ⇒ *'line* ⇒ *'point*
assumes *meet-def*: (*incid* (*meet l m*) *l* ∧ *incid* (*meet l m*) *m*)

assumes *incid-dec*: (*incid P l*) ∨ ¬(*incid P l*)

assumes *ax1-existence*: ∃ *l*. (*incid P l*) ∧ (*incid M l*)
assumes *ax1-uniqueness*: (*incid P k*) ⟶ (*incid M k*) ⟶ (*incid P l*) ⟶ (*incid M l*) ⟶ (*P* = *M*) ∨ (*k* = *l*)

assumes *ax2*: *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*) ⟶ (∃ *I*.(*incid I lAB* ∧ *incid I lCD*)) ⟶ (∃ *J*.(*incid J lAC* ∧ *incid J lBD*))

assumes *ax3*: ∃ *A B C*. *distinct3 A B C* ∧ (*incid A l*) ∧ (*incid B l*) ∧ (*incid C l*)

assumes *ax4*: ∃ *l m*.∀ *P*. ¬(*incid P l* ∧ *incid P m*)

assumes *ax5*: *distinct* [*l1*,*l2*,*l3*] ⟶ (∃ *l4 J1 J2 J3*. *distinct* [*J1*,*J2*,*J3*] ∧ *meet l1 l4* = *J1* ∧ *meet l2 l4* = *J2* ∧ *meet l3 l4* = *J3*)

end
theory *Higher-Projective-Space-Axioms*
imports *Main*
begin

Contents:

- We introduce the types of 'point and 'line and an incidence relation between them.
- A set of axioms for higher projective spaces, i.e. we allow models of dimension > 3.

11 The axioms for Higher Projective Geometry

lemma *distinct4-def*:
distinct $[A,B,C,D] = ((A \neq B) \wedge (A \neq C) \wedge (A \neq D) \wedge (B \neq C) \wedge (B \neq D) \wedge (C \neq D))$
⟨*proof*⟩

lemma *distinct3-def*:
distinct $[A,B,C] = ((A \neq B) \wedge (A \neq C) \wedge (B \neq C))$
⟨*proof*⟩

locale *higher-projective-space* =

fixes *incid* :: *'point* ⇒ *'line* ⇒ *bool*

assumes *ax1-existence*: $\exists\, l.\ (incid\ P\ l) \wedge (incid\ M\ l)$
assumes *ax1-uniqueness*: $(incid\ P\ k) \longrightarrow (incid\ M\ k) \longrightarrow (incid\ P\ l) \longrightarrow (incid\ M\ l) \longrightarrow (P = M) \vee (k = l)$

assumes *ax2*: *distinct* $[A,B,C,D] \longrightarrow (incid\ A\ lAB \wedge incid\ B\ lAB) \longrightarrow (incid\ C\ lCD \wedge incid\ D\ lCD) \longrightarrow (incid\ A\ lAC \wedge incid\ C\ lAC) \longrightarrow (incid\ B\ lBD \wedge incid\ D\ lBD) \longrightarrow (\exists\, I.(incid\ I\ lAB \wedge incid\ I\ lCD)) \longrightarrow (\exists\, J.(incid\ J\ lAC \wedge incid\ J\ lBD))$

assumes *ax3*: $\exists\, A\ B\ C.$ *distinct* $[A,B,C] \wedge (incid\ A\ l) \wedge (incid\ B\ l) \wedge (incid\ C\ l)$

assumes *ax4*: $\exists\, l\ m. \forall\, P.\ \neg(incid\ P\ l \wedge incid\ P\ m)$

end

12 Acknowledgements

The author was supported by the ERC Advanced Grant ALEXANDRIA (Project 742178) funded by the European Research Council and led by Professor Lawrence Paulson at the University of Cambridge, UK.

References

[1] M. Bezem and D. Hendriks. On the mechanization of the proof of hessenberg's theorem in coherent logic. *Journal of Automated Reasoning*, 40(1):61–85, Jan 2008.

[2] N. Magaud, J. Narboux, and P. Schreck. A case study in formalizing projective geometry in coq: Desargues theorem. *Computational Geometry*, 45(8):406–424, Oct 2012.