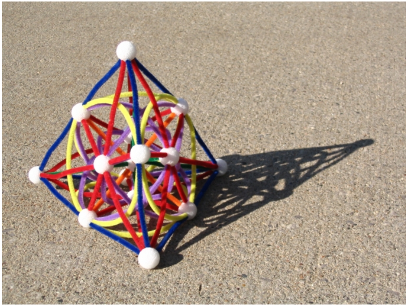
Building some Finite Models of Projective Space Geometry in Coq



Nicolas Magaud

groupe LTP (GDR GPL) - 6 dec. 2018

Outline

1. Motivations and Context
2. Examples pg(3,2) and pg(3,3)
3. Coq specifications
4. Proof Optimizations
5. Results and Future Work

Projective Space Geometry

- Incidence Geometry
  - points, lines and an incidence relation
- Projective Incidence Geometry
  - in 2D : 2 lines always intersect
  - in 3D : Pasch's axiom
- Simple description : only 6 axioms
- Finite Models in Coq
  - focusing on 3D models : pg(3,2), . . .
  - taking Coq to its limits (w.r.t. specification and w.r.t. proof)

Objects and Operations

- Objects : Point, Line

```
Parameter Point, Line : Type.
```

- Incidence relation : incid_lp

```
Parameter incid_lp : Point -> Line -> bool.
```

- Boolean equalities on points and lines : eqP, eqL

```
Parameter eqP : Point -> Point -> bool.
Parameter eqL : Line -> Line -> bool.
```

- All distinct for points and lines : dist_3p, dist_4p, dist_3l

```
Definition dist_3p (A B C :Point) : bool :=
(negb (eqP A B)) && (negb (eqP A C)) && (negb (eqP B C)).

Definition dist_4p (A B C D:Point) : bool := ...
Definition dist_3l (A B C :Line) : bool := ...
```

- Intersection of 2 lines : Intersect_In

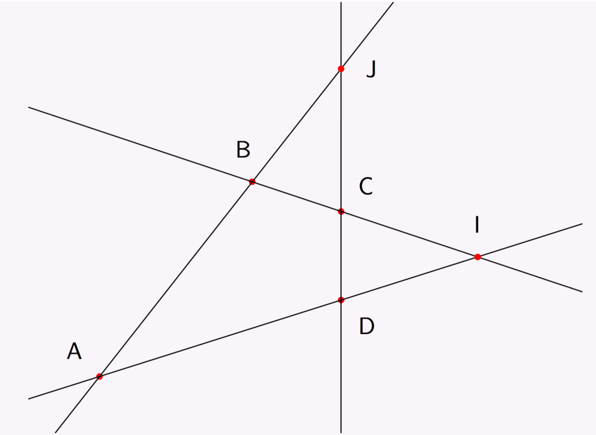
```
Definition Intersect_In (l1 l2 :Line) (P:Point) :=
incid_lp P l1 && incid_lp P l2.
```

Axioms for Projective Space Geometry : from a geometry point of view

- a1 : throught 2 points, there is one line.
- uniqueness : Given 2 points and 2 lines, if the 2 points are both on both lines, either the points are equal, or the lines.
- a2 : Pasch's axiom (if 2 lines intersect. . .).
- a3_1 : Each line has at least 3 points.
- a3_2 : There exists 2 lines which do not intersect (dim>2).
- a3_3 : Given 3 distinct lines, there exists a fourth one which intersects with all three (dim<=3).

Axioms for Projective Space Geometry : from a geometry point of view

- Pasch's axiom



Axioms for Projective Space Geometry : from a geometry point of view

```
Axiom a1_exists : forall A B : Point, { l : Line| incid_lp A l && incid_lp B l}.

Axiom uniqueness : forall (A B :Point)(l1 l2:Line),
incid_lp A l1 -> incid_lp B l1 -> incid_lp A l2 -> incid_lp B l2 -> A = B \/ l1 = l2.

Axiom a2 : forall A B C D:Point, forall lAB lCD lAC lBD :Line, dist_4p A B C D ->
incid_lp A lAB && incid_lp B lAB -> incid_lp C lCD && incid_lp D lCD ->
incid_lp A lAC && incid_lp C lAC -> incid_lp B lBD && incid_lp D lBD ->
(exists I:Point, incid_lp I lAB && incid_lp I lCD) ->
exists J:Point, incid_lp J lAC && incid_lp J lBD.

Axiom a3_1 : forall l:Line,
{A:Point & {B:Point & {C:Point |
(dist_3p A B C) && (incid_lp A l && incid_lp B l && incid_lp C l)}}}.

Axiom a3_2 : exists l1:Line, exists l2:Line,
forall p:Point,  (incid_lp p l1 && incid_lp p l2).

Axiom a3_3 : forall l1 l2 l3:Line, dist_3l l1 l2 l3 ->
exists l4 :Line, exists J1:Point, exists J2:Point, exists J3:Point,
Intersect_In l1 l4 J1 && Intersect_In l2 l4 J2 && Intersect_In l3 l4 J3.
```

Axioms for Projective Space Geometry : from a logic point of view

```
Axiom a1_exists : forall A B : Point, { l : Line| incid_lp A l && incid_lp B l}.

Axiom uniqueness : forall (A B :Point)(l1 l2:Line),
incid_lp A l1 -> incid_lp B l1 -> incid_lp A l2 -> incid_lp B l2 -> A = B \/ l1 = l2.

Axiom a2 : forall A B C D:Point, forall lAB lCD lAC lBD :Line, dist_4p A B C D ->
incid_lp A lAB && incid_lp B lAB -> incid_lp C lCD && incid_lp D lCD ->
incid_lp A lAC && incid_lp C lAC -> incid_lp B lBD && incid_lp D lBD ->
(exists I:Point, incid_lp I lAB && incid_lp I lCD) ->
exists J:Point, incid_lp J lAC && incid_lp J lBD.

Axiom a3_1 : forall l:Line,
{A:Point & {B:Point & {C:Point |
(dist_3p A B C) && (incid_lp A l && incid_lp B l && incid_lp C l)}}}.

Axiom a3_2 : exists l1:Line, exists l2:Line,
forall p:Point,  (incid_lp p l1 && incid_lp p l2).

Axiom a3_3 : forall l1 l2 l3:Line, dist_3l l1 l2 l3 ->
exists l4 :Line, exists J1:Point, exists J2:Point, exists J3:Point,
Intersect_In l1 l4 J1 && Intersect_In l2 l4 J2 && Intersect_In l3 l4 J3.
```

Outline

1. Motivations and Context
2. Examples pg(3,2) and pg(3,3)
3. Coq specifications
4. Proof Optimizations
5. Results and Future Work

Examples : pg(3,q)

	# points	# lines	# points per line
$pg(3,2)$	15	35	3
$pg(3,3)$	40	130	4
$pg(3,4)$	85	357	5
$pg(3,q)$	$(q^2+1)(q+1)$	$(q^2+q+1)(q^2+1)$	$q+1$

- By duality : # planes = # points.
- Describing the incidence relation of pg(3, q) :
  for each line, we provide the q+1 points which belong to it.
- e.g. pg(3,3) [1]

1. Alan R. Prince. Projective planes of order 12 and PG(3,3). Discrete Mathematics, 208-209 :477-483, 1999.

pg(3,3) - description of the incidence relation

*L*1	0	1	4	13	*L*14	1	2	5	14	*L*27	1	7	12	33	*L*40	1	8	24	26	*L*53	1	10	37	38
*L*2	0	2	24	17	*L*15	2	3	6	15	*L*28	2	19	26	4	*L*41	2	22	12	32	*L*54	2	33	29	13
*L*3	0	3	12	39	*L*16	3	5	27	20	*L*29	3	38	32	24	*L*42	3	10	26	28	*L*55	3	4	7	16
*L*4	0	5	26	34	*L*17	5	6	9	18	*L*30	5	30	28	12	*L*43	5	33	32	36	*L*56	5	24	19	13
*L*5	0	6	32	11	*L*18	6	27	35	1	*L*31	6	16	36	26	*L*44	6	4	28	21	*L*57	6	12	38	17
*L*6	0	27	28	31	*L*19	27	9	25	2	*L*32	27	13	21	32	*L*45	27	24	36	23	*L*58	27	26	30	39
*L*7	0	9	36	37	*L*20	9	35	14	3	*L*33	9	17	23	28	*L*46	9	12	21	8	*L*59	9	32	16	34
*L*8	0	35	21	29	*L*21	35	25	15	5	*L*34	35	39	8	36	*L*47	35	26	23	22	*L*60	35	28	13	11
*L*9	0	25	23	7	*L*22	25	14	20	6	*L*35	25	34	22	21	*L*48	25	32	8	10	*L*61	25	36	17	31
*L*10	0	14	8	19	*L*23	14	15	20	6	*L*36	14	11	10	23	*L*49	14	28	22	33	*L*62	14	21	39	37
*L*11	0	15	22	38	*L*24	15	20	1	9	*L*37	15	31	33	8	*L*50	15	36	10	4	*L*63	15	23	34	29
*L*12	0	20	10	30	*L*25	20	18	2	35	*L*38	20	37	4	22	*L*51	20	21	33	24	*L*64	20	8	11	7
*L*13	0	18	33	16	*L*26	18	1	3	25	*L*39	18	29	24	10	*L*52	18	23	4	12	*L*65	18	22	31	19

*L*66	1	11	21	31	*L*79	1	16	23	39	*L*92	1	17	19	34	*L*105	1	22	30	36	*L*118	1	28	29	32
*L*67	2	31	23	37	*L*80	2	13	8	34	*L*93	2	39	38	11	*L*106	2	10	16	21	*L*119	2	36	7	28
*L*68	3	37	8	29	*L*81	3	17	22	11	*L*94	3	34	30	31	*L*107	3	33	13	23	*L*120	3	21	19	36
*L*69	5	29	22	7	*L*82	5	39	10	31	*L*95	5	11	16	37	*L*108	5	4	17	8	*L*121	5	23	38	21
*L*70	6	7	10	19	*L*83	6	34	33	37	*L*96	6	31	13	29	*L*109	6	24	39	22	*L*122	6	8	30	23
*L*71	27	19	33	38	*L*84	27	11	4	29	*L*97	27	37	17	7	*L*110	27	12	34	10	*L*123	27	22	16	8
*L*72	9	38	4	30	*L*85	9	31	24	7	*L*98	9	29	39	19	*L*111	9	26	11	33	*L*124	9	10	13	22
*L*73	35	30	24	16	*L*86	35	37	12	19	*L*99	35	7	34	38	*L*112	35	32	31	4	*L*125	35	33	17	10
*L*74	25	16	12	13	*L*87	25	29	26	38	*L*100	25	19	11	30	*L*113	25	28	37	24	*L*126	25	4	39	33
*L*75	14	13	26	17	*L*88	14	7	32	30	*L*101	14	38	31	16	*L*114	14	36	29	12	*L*127	14	24	34	4
*L*76	15	17	32	39	*L*89	15	19	28	16	*L*102	15	30	37	13	*L*115	15	21	7	26	*L*128	15	12	11	24
*L*77	20	39	28	34	*L*90	20	38	36	13	*L*103	20	16	29	17	*L*116	20	23	19	32	*L*129	20	26	31	12
*L*78	18	34	36	11	*L*91	18	30	21	17	*L*104	18	13	7	39	*L*117	18	8	38	28	*L*130	18	32	37	26

pg(3,3) - comments on the description

- The formal proof for the axioms fails. Why ?
- Checking the formal statement is correct.
- Checking the proof method works properly
  (yes, it works for pg(3,2)).
- Checking the description is correct.
  The incidence relation is incorrect.
- Fixing the incidence relation : using the number of points per line property, to locate the errors.

pg(3,3) - description of the incidence relation

*L*1	0	1	4	13	*L*14	1	2	5	14	*L*27	1	7	12	33	*L*40	1	8	24	26	*L*53	1	10	37	38
*L*2	0	2	24	17	*L*15	2	3	6	15	*L*28	2	19	26	4	*L*41	2	22	12	32	*L*54	2	33	29	13
*L*3	0	3	12	39	*L*16	3	5	27	20	*L*29	3	38	32	24	*L*42	3	10	26	28	*L*55	3	4	7	16
*L*4	0	5	26	34	*L*17	5	6	9	18	*L*30	5	30	28	12	*L*43	5	33	32	36	*L*56	5	24	19	13
*L*5	0	6	32	11	*L*18	6	27	35	1	*L*31	6	16	36	26	*L*44	6	4	28	21	*L*57	6	12	38	17
*L*6	0	27	28	31	*L*19	27	9	25	2	*L*32	27	13	21	32	*L*45	27	24	36	23	*L*58	27	26	30	39
*L*7	0	9	36	37	*L*20	9	35	14	3	*L*33	9	17	23	28	*L*46	9	12	21	8	*L*59	9	32	16	34
*L*8	0	35	21	29	*L*21	35	25	15	5	*L*34	35	39	8	36	*L*47	35	26	23	22	*L*60	35	28	13	11
*L*9	0	25	23	7	*L*22	25	14	20	6	*L*35	25	34	22	21	*L*48	25	32	8	10	*L*61	25	36	17	31
*L*10	0	14	8	19	*L*23	14	15	20	6	*L*36	14	11	10	23	*L*49	14	28	22	33	*L*62	14	21	39	37
*L*11	0	15	22	38	*L*24	15	20	1	9	*L*37	15	31	33	8	*L*50	15	36	10	4	*L*63	15	23	34	29
*L*12	0	20	10	30	*L*25	20	18	2	35	*L*38	20	37	4	22	*L*51	20	21	33	24	*L*64	20	8	11	7
*L*13	0	18	33	16	*L*26	18	1	3	25	*L*39	18	29	24	10	*L*52	18	23	4	12	*L*65	18	22	31	19

18 27

30

*L*66	1	11	21	31	*L*79	1	16	23	39	*L*92	1	17	19	34	*L*105	1	22	30	36	*L*118	1	28	29	32
*L*67	2	31	23	37	*L*80	2	13	8	34	*L*93	2	39	38	11	*L*106	2	10	16	21	*L*119	2	36	7	28
*L*68	3	37	8	29	*L*81	3	17	22	11	*L*94	3	34	30	31	*L*107	3	33	13	23	*L*120	3	21	19	36
*L*69	5	29	22	7	*L*82	5	39	10	31	*L*95	5	11	16	37	*L*108	5	4	17	8	*L*121	5	23	38	21
*L*70	6	7	10	19	*L*83	6	34	33	37	*L*96	6	31	13	29	*L*109	6	24	39	22	*L*122	6	8	30	23
*L*71	27	19	33	38	*L*84	27	11	4	29	*L*97	27	37	17	7	*L*110	27	12	34	10	*L*123	27	22	16	8
*L*72	9	38	4	30	*L*85	9	31	24	7	*L*98	9	29	39	19	*L*111	9	26	11	33	*L*124	9	10	13	22
*L*73	35	30	24	16	*L*86	35	37	12	19	*L*99	35	7	34	38	*L*112	35	32	31	4	*L*125	35	33	17	10
*L*74	25	16	12	13	*L*87	25	29	26	38	*L*100	25	19	11	30	*L*113	25	28	37	24	*L*126	25	4	39	33
*L*75	14	13	26	17	*L*88	14	7	32	30	*L*101	14	38	31	16	*L*114	14	36	29	12	*L*127	14	24	34	4
*L*76	15	17	32	39	*L*89	15	19	28	16	*L*102	15	30	37	13	*L*115	15	21	7	26	*L*128	15	12	11	24
*L*77	20	39	28	34	*L*90	20	38	36	13	*L*103	20	16	29	17	*L*116	20	23	19	32	*L*129	20	26	31	12
*L*78	18	34	36	11	*L*91	18	30	21	17	*L*104	18	13	7	39	*L*117	18	8	38	28	*L*130	18	32	37	26

Outline

1. Motivations and Context
2. Examples pg(3,2) and pg(3,3)
3. Coq specifications
4. Proof Optimizations
5. Results and Future Work

Coq specifications

- Point and Line as simple inductive types.
  - Case analysis is easy.
  - Finding a witness can be challenging.
    = trying each possible value and running the tactics.
  - Writing the specification is a bit boring.

    ```
    Inductive Point := P0 | P1 | P2 | ... | P40.
    ```

- Solutions
  - Using finite types (ssreflect/mathcomp)
  - Using plain inductive data-types and an external program to generate the specification

Our choice : an external program

- We choose to have an external program generating the specification (actually outputs a gallina specification)
- Indeed, we need a specification generation process anyway (for witnesses).
- Our implementation
  - plain data-types combined with boolean reflection
  - generating data-types such as Line (130 constructors)
  - incidence relation as a boolean predicate
  - equality (decidable)
  - order relation (decidable and total)
  - The witness for existential quantification are computed beforehand.

Outline

1. Motivations and Context
2. Examples pg(3,2) and pg(3,3)
3. Coq specifications
4. Proof Optimizations
5. Results and Future Work

Proof Optimizations

- Witness finding reduced to function computation

```
Definition f_a3_3 (l1:Line) (l2:Line) (l3:Line) := ...
```

  computes a line which intersects the 3 lines l1, l2 and l3.

- Factorizing proofs as lemmas.

$$\forall T\ Z\ x, \texttt{incid\_lp}\ T\ x\ \rightarrow\ \texttt{incid\_lp}\ Z\ x \rightarrow T <> Z\ \rightarrow\ x = (\texttt{l\_from\_points}\ T\ Z)$$

- Proof-engineering : sequences of tactics, abstract, par

```
par :abstract (time (case v2; intros hp1p2; first [exact (degen_bool _ hp1p2) | (case
v3; intros hp1p3 hdist x; solve [ (exact (degen_bool _ hp1p3)) | (exact (degen_bool
_ hdist)) | exists_lppp (fst x) (fst (fst (snd x))) (snd (fst (snd x))) (snd (snd
x)) ])])).
```

  no try, each goal is solved the first time it is encountered.

Symmetries

- Using appropriate symmetries to reduce the number of cases to check.

```
Axiom a2 : forall A B C D:Point, forall lAB lCD lAC lBD :Line, dist_4p A B C D ->
incid_lp A lAB && incid_lp B lAB -> incid_lp C lCD && incid_lp D lCD ->
incid_lp A lAC && incid_lp C lAC -> incid_lp B lBD && incid_lp D lBD ->
(exists I:Point, incid_lp I lAB && incid_lp I lCD) ->
exists J:Point, incid_lp J lAC && incid_lp J lBD.
```

for pg(3,3) : at least 40*40*40*40 = 2 560 000 cases to go

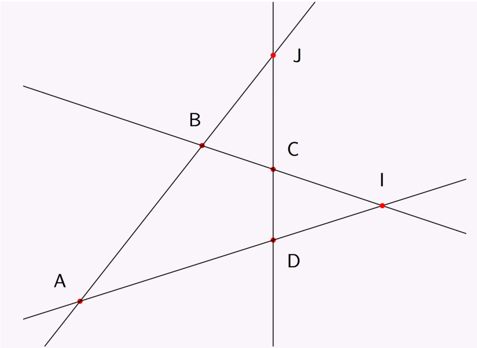
- Adding an order relation on points and lines.

```
Axiom a2 : forall A B C D:Point, forall lAB lCD lAC lBD :Line,

leP A B -> leP C D ->

dist_4p A B C D ->
incid_lp A lAB && incid_lp B lAB -> incid_lp C lCD && incid_lp D lCD ->
incid_lp A lAC && incid_lp C lAC -> incid_lp B lBD && incid_lp D lBD ->
(exists I:Point, incid_lp I lAB && incid_lp I lCD) ->
exists J:Point, incid_lp J lAC && incid_lp J lBD.
```

Without loss of generality

- Implementing a without loss of generality principle
  - Re-ordering points in a specific order
  - Re-using the previous statement (in a tactic)
  - it requires adapting the statement as follows :

```
... (exists I:Point, incid_lp I lAB && incid_lp I lCD) ->
(exists J:Point, (incid_lp J lAC && incid_lp J lBD)) /\
(exists K:Point, (incid_lp K lAD && incid_lp K lBC)).
```



Outline

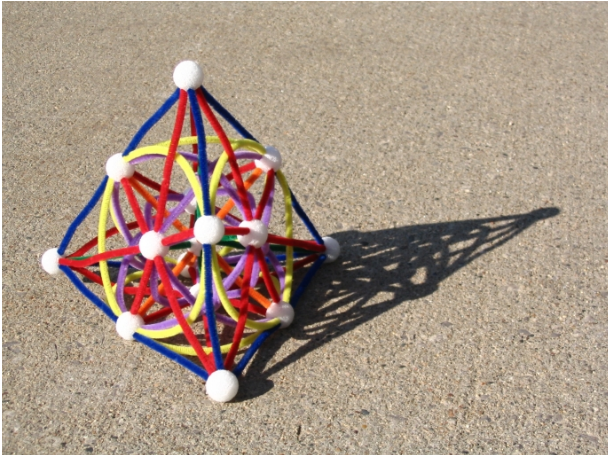
1. Motivations and Context
2. Examples pg(3,2) and pg(3,3)
3. Coq specifications
4. Proof Optimizations
5. Results and Future Work

Results and Future Work

- Results
  - Using Intel(R) Core(TM) i5-4460 CPU @ 3.20GHz, 32 GB
  - 24 min to verify the axioms of projective geom. for pg(3,2)
  - 2 h to verify the axioms of projective geom. for pg(3,3)
  - some experiments with Z3 and lean
- Related and future work
  - Ranks (of sets of points) are an interesting alternative approach (PhD work of David Braun)
  - Next step : spreads and packings in pg(3,2)
  - Example of state-of-the-art results :
    Svetlana Topalova and Stela Zhelezova. *On transitive parallelisms of PG(3,4)*. 2017

Questions ?

- Thank you for your attention !



©David Richter - Western Michigan University