



## Automatic generating of theorems and constructing models for projective geometry

Joanna Meksuła\*

*Institute of Mathematics, Maria Curie-Skłodowska University,  
pl. Marii Curie-Skłodowskiej 1, 20-031 Lublin, Poland*

### Abstract

The paper discusses the use of computers to construct models and generate theorems of projective geometry. After signalling the history of the issue, the axiomatics as well as basic properties of projective geometry have been introduced. The main body of the paper constitutes a proposed and discussed idea of building a plane by implementing axioms. As an essential extension, the theorems are pointed out, that, together with proofs, appear in the course of the program work. The limits and possible modifications of the proposed application are given in Conclusions.

### 1. History and significance of the issue

The source of projective geometry is a notice that our intuition about the notion of space is perceived mainly by means of sight. That yields a defined vision of space, noticeably different from the Euclidean one. The question about this kind of vision was known as the problem of perspective in painting: how to display a three-dimensional object in the two-dimensional space. The first examples of projective geometry distinctiveness were demonstrated by Girard Desargues, an architect of gardens (that is a form of space – managing). In 1639 he edited his work ‘Rough Draft for an Essay on the Results of Taking Plane Section of a Cone’. In 1648 he published the theorem named after him.

Victor Poncelet is a creator of projective geometry as a separate branch of mathematics. In 1822 his ‘Traité des propriétés des figures’ appeared, which is a monograph of projective geometry. There are projections, collineations, cone correlations, cross-ratios, harmonics as well as axiomatics of projective geometry. General theorems of projective geometry examine the properties of figures that remain unchanged during projections. Neither the length nor the length ratio are of importance here. There are the relations, like situating a point

---

\*E-mail address: [joannap@hektor.umcs.lublin.pl](mailto:joannap@hektor.umcs.lublin.pl)

on a straight line or crossing the given points by a straight line, that are crucial here [1].

One of the trends of projective geometry research is looking for structures called projective planes of various orders. Mathematics has not answered the question comprehensively, the problem is solved only partially. Therefore certain scientists took up checking this fact by means of the computer. However, the discussion whether to trust the computer was held much earlier. It was initiated when in 1976 the proof of 'The hypothesis of four colours' was published by Wolfgang Haken and Kenneth Appel. The major part of the proof consisted of computer computation. So far the computer as the most advanced tool, has only been assistance in works. It has made possible to obtain an approximate answer or has been used to generate data. Many mathematicians have been disappointed, as they have expected a smart solution. They are dissatisfied with dispersing the proof into thousands of cases and checking all of them by the computer. Besides, using computers does not eliminate a human error: the computer itself is man-made.

However, the same method – dispersing the proof into particular cases and checking them by the computer – was used by C. W. H. Lam while looking for the finite plane of order 10 [3].

## 2. Projective geometry

Projective geometry is an axiom theory.

**Definition 2.1** *Axiomatics* is any set of propositions.

**Definition 2.2** The text  $D$ ,  $Ln(D)$  long is a **proof** on the basis of axiomatics  $X$  iff:

$$\forall_{i:(i < Ln(D))} \left( D_i \text{ is tautology} \vee D_i \in X \vee \exists_{j, k < i} (D_j = (D_k \rightarrow D_i)) \right)$$

**Definition 2.3**  $D$  is a **proof of proposition  $B$**  on the basis of axiomatics  $X$  iff  $D$  is a proof and the last proposition of the text  $D$  is identical to proposition  $B$ :

$$D_{Ln(D)-1} = B.$$

**Definition 2.4** [3] Formula  $A$  is a **consequence** of a set  $X$  (marked:  $A \in Cn(X)$ ) iff there exists a proof  $D$  of the proposition  $D$  based on axiomatics  $X$

$$Cn(X) = \{x : x \text{ is a consequence of } X\}.$$

**Definition 2.5** [3] A set  $X$  is a **theory** iff it is identical to its consequences, that is when it fulfils the formula:

$$X = Cn(X).$$

Let us apply a two-sort formalisation of projective geometry which stresses the property of duality characteristic of this branch of geometry. The formalisation is modelled after [3]. In this approach straight lines and points are treated independently.

We will use an elementary two-sort language with one two-argument predicate:

- *variables of the first type* are marked with small letters of the Latin alphabet,
- *variables of the second type* are marked with big letters of the Latin alphabet,
- *predicate* | which arguments are: the variable of the first type, the variable of the second type accordingly.

The following structures are the realisations of the assumed language:

$$\langle U_1, U_2; | \rangle,$$

where  $| \subset U_1 \times U_2$ .

The elements of the first universum are called *points*, and the second – *straight lines*<sup>1</sup>.

To read the relation  $p | P$  we can use one of the equivalent statements:

- point  $p$  is incident to straight line  $P$ ,
- point  $p$  lies on straight line  $P$ ,
- straight line  $P$  crosses point  $p$ .

The following abbreviations will be used in the paper:

$$p_1 \dots p_n | P_1 \dots P_m \equiv \forall_{i:(1 \leq i \leq n)} \forall_{j:(1 \leq j \leq m)} (p_i | P_j)$$

and

$$(p_1 \dots p_n \sim | P_1 \dots P_m) \equiv \forall_{i:(1 \leq i \leq n)} \forall_{j:(1 \leq j \leq m)} (p_i \sim | P_j) .$$

**Definition 2.6** [3] *A pencil* is a set of the form:

$$a^* = \{P \in U_2 : a | P\}$$

and a *chain* is a set of the form:

$$A^* = \{p \in U_1 : p | A\} .$$

The straight lines belonging to one pencil are called *co-pencil* and the points belonging to one chain – *colinear*.

The *power of chain* (that is the number of its points) will be designated as  $\overline{A^*}$ , and the *power of pencil* (quantity of straight lines in a pencil) as  $\overline{a^*}$ .

**Definition 2.7** [3] *An operation of taking a dual formula* we define as follows:

<sup>1</sup>In projective geometry points and straight lines are objects of different types; straight lines should not be identified with a set of points.

- $(p_i | P_j)^\circ = (p_j | P_i)$
- $(p_i | p_j)^\circ = (P_i | P_j)$
- $(P_i | P_j)^\circ = (p_i | p_j)$ .

For complex formulas the definition undergoes the induction in a usual way.

The dual formula to a given formula  $\sigma$  will be denoted  $\sigma^\circ$ .

In projective geometry the rule exists that if in any theorem (concerning plane) we change the symbols of straight lines and points as well as ‘crosses’ and ‘lies on’, we obtain a new statement which is also a theorem: every theorem has an equivalent dual statement.

Let us consider the sentences:

$$\mathbf{A1} \quad ab | AB \rightarrow (a = b \vee A = B)$$

$$\mathbf{A2} \quad \forall ab \exists C (ab | C)$$

$$\mathbf{A3} \quad \forall AB \exists C (c | AB)$$

$$\mathbf{A4} \quad \exists abcd ABCD$$

$$(ab | A \wedge bc | D \wedge cd | C \wedge da | B \wedge ab \sim | C \wedge bc \sim | B \wedge cd \sim | A \wedge da \sim | D).$$

**Definition 2.8** [3] *Projective geometry is an elementary theory based on axioms A1..A4 that is*

$$Cn(\{A1, A2, A3, A4\}),$$

*and its models are projective planes.*

**Definition 2.9** [3] *The order of point  $a$  is a number  $\overline{\overline{a^*}} - 1$ , and the order of straight line  $A$  is a number  $\overline{\overline{A^*}} - 1$ .*

**Definition 2.10** [3] *The order of plane is the order of its any point or its any straight line.*

**Theorem 2.1** *For any plane of the order  $n$  the power of any pencil  $a^*$  is  $\overline{\overline{a^*}} = n + 1$ .*

**Theorem 2.2** *For any plane of the order  $n$  the power of any chain  $A^*$  is  $\overline{\overline{A^*}} = n + 1$ .*

**Theorem 2.3** In any model  $\langle U_1, U_2; I \rangle$  of projective geometry, if for a given  $A \in U_2$ ,  $\overline{A^*} = m$ , then

$$\overline{\overline{U_1}} = \overline{\overline{U_2}} = m(m-1)+1.$$

Until now the problem: which natural numbers can be an order of projective plane, has not been completely solved. Only a partial answer to this question exists.

**Theorem 2.4** There are projective planes of order  $p^k$  where  $p$  is a prime number and  $k$  is a natural number.

On the basis of this fact we can state that there are projective planes of the order, for example: 2 ( $p=2, k=1$ ), 3 ( $p=3, k=1$ ), 4 ( $p=2, k=2$ ), 5 ( $p=5, k=1$ ), 7 ( $p=7, k=1$ ), 8 ( $p=2, k=3$ ), 9 ( $p=3, k=2$ ), 11 ( $p=11, k=1$ ), 13 ( $p=13, k=1$ ), 16 ( $p=2, k=4$ ), 17 ( $p=17, k=1$ ), 19 ( $p=19, k=1$ ).

It is unknown whether the other numbers can be orders of projective planes. We only know that:

**Theorem 2.5** (Bruck, Ryser) [4]  $n$  cannot be the order of any projective plane when:

$$(n+1 = 2 \pmod{4} \vee n+1 = 3 \pmod{4}) \wedge \exists_p \exists_k (p^{2k-1} | n \wedge \sim (p^{2k} | n) \wedge p = 3 \pmod{4})$$

where  $p$  and  $n$  are natural numbers.

This enables us to claim that the projective planes of the orders: 6, 14, 21 (for example) do not exist.

As we notice, there are the numbers for which the problem has been unsolved yet. We do not know, for example, if the projective planes of orders: 12, 15, 18 or 20 exist at all. These are open questions of projective geometry.

However, the problem of existence of projective planes of order 10 was unsolved. But in 1991 C. W. H. Lam from Concordia University, Canada, published the results of his long-term studies of this question [2]. With the use of computers and after a huge number of laborious and time – consuming calculations he stated that the projective plane of order 10 does not exist. In his research he used the axiomatics of projective geometry and, partially, already available theorems. The Lam algorithm was based on the analysis of dependencies among the lines of matrix incidences of the projective plane (Bruck – Ryser – Chowl theorem, the generalised Bruck – Ryser theorem).

### 3. Information Technology approach

The present attempt of solving the problem of projective planes existence is based only on axiom A1-A4. Any additional knowledge about the generated configuration is not considered.

A table is an object representing the model where rows will display straight lines and columns – points. At the beginning the table is empty. The program will fill it in gradually, using the axioms A1-A4. Therefore appropriate assumptions will be made. Filling in the system means marking a correlation between a straight line and a point in the corresponding place of crossing a line and a column. At some time it may occur that the approved suppositions lead to a contradiction. This discrepancy appears when in an already filled (on the basis of one axiom) field, as the consecutive step, we want to insert an opposite value. Then the program will generate a list of assumptions which are the reason for the contradiction. That results in a proper modification of the program taking into account the obtained suppositions; and another attempt of generating the model takes place.

At present the program is prepared for generating the model of projective plane of order 2.

*Three-valued logic* is used in the work

**Definition 3.1** [5] *Three-valued logic is the logic taking one of the three values:*

- TRUTH
- FALSE
- NOT KNOWN YET

The following notions will be used:

- TRUTH: 1
- FALSE: - 1
- NOT KNOWN YET: 0.

The values of logic operations are described:

- 1)  $\sim p = -p$
- 2)  $p \wedge q = \min(p, q)$
- 3)  $p \vee q = \max(p, q)$ .

The program arose with the compiler Delphi 5 from Borland. An objective approach to programming is used. The types to store information are defined: about the model and about the theorems. One of the major functions is an attempt of generating the model. It uses minor procedures that indicate possible incidences as well as procedures testing consistency of the produced model and axioms. The functions are also crucial that generate theorems on the basis of the program procedure (which is also automatically created at any step of the program). Implementing the obtained theorems is a human task.

#### 4. The results of the program

The program was equipped with a possibility of checking an agreement of a generated model and axioms. However, checking axiom A4, we encounter the problem of time. It results from the fact that A4 is an existential axiom defined on four points and four straight lines (for the plane of order 2, four points can be

chosen by  $\binom{7}{4} = 35$  possibilities, the same for the fourth straight lines.

Altogether it is  $\binom{7}{4}^2 * 4! = 29400$  possibilities of choice). This is the

existentiality of the axiom that allows us to finish checking after finding the first four points and four straight lines that fulfil the axiom. Additionally, we know that arbitrary permutations of straight lines and points yield an equivalent model. The above considerations suggest the conclusion that the model can be initiated at the beginning to fulfil axiom A4. The table representing the model after initiating is as follows:

Table 1. Axiom A4 application

1	1	0	0	0	0	0
1	0	1	0	0	0	0
0	1	0	1	0	0	0
0	1	0	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Moreover, on the basis of axiom A1, we can supplement additionally our model with certain non – incidences:

Table 2. Initiated scheme

1	1	-1	-1	0	0	0
1	-1	1	-1	0	0	0
-1	1	-1	1	0	0	0
-1	1	-1	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Owing to such a method we have 16 (out of 49) fields inserted properly.

From this point the process of generating the model will proceed as a sequence of using axioms A2 and A3 together with axiom A1.

Axiom A2 says that every two points belong to the same chain, that is for every two points  $x_1$  and  $x_2$ , the straight line  $X$  exists that is incident. The algorithm is implemented in such a way that it chooses the first possible straight line meeting this condition. Then we apply axiom A1.

Axiom A3 and its meaning is dual for A2, that is why the procedures in this case are very similar.

The procedures are continued up to the moment when we obtain a filled-in table. Next we check an agreement of the model produced with axioms. In the case of any discrepancy we have a look at the files generated during the operation of the program. In these files the lemmas and their proofs automatically generated by the computer are recorded in a human-comprehensive form.

A distinctive feature of the model generating method is an automatic production of lemmas and their proofs.

The first application of the program gives the following scheme:

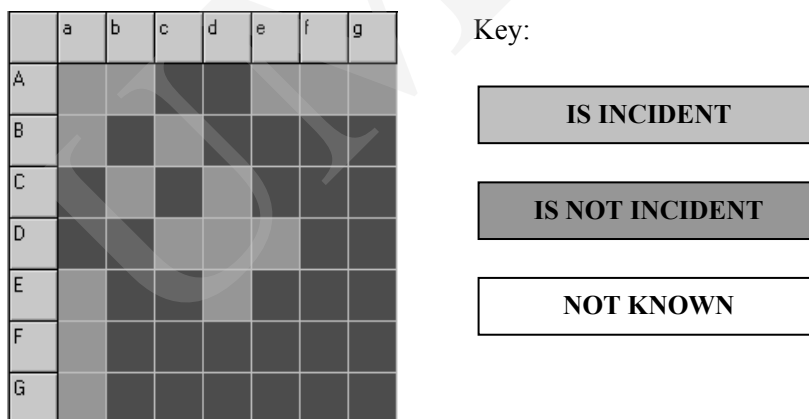


Fig. 1. The first application

Contradictions appearing during its generation lead to the following lemma **L1**:

$$\begin{aligned}
 \forall abcdefABCD \quad & a \sim | AB \vee b \sim | AC \vee c \sim | BD \vee d \sim | CD \vee e \sim | A \vee f \sim | A \vee \\
 & a = b \vee a = c \vee a = d \vee a = e \vee a = f \vee b = c \vee b = d \vee b = e \vee \\
 & b = f \vee c = d \vee c = e \vee c = f \vee d = e \vee d = f \vee e = f \vee \\
 & A = B \vee A = C \vee A = D \vee B = C \vee B = D \vee C = D
 \end{aligned}$$

It means that the incidences applied during the initialisation of the model and incidence  $ef | A$  cannot be simultaneous.

This results in the fact that the computer is not able to generate a proper model merely on the basis of axioms.

Graphically lemma L1 denotes that the following scheme cannot take place:

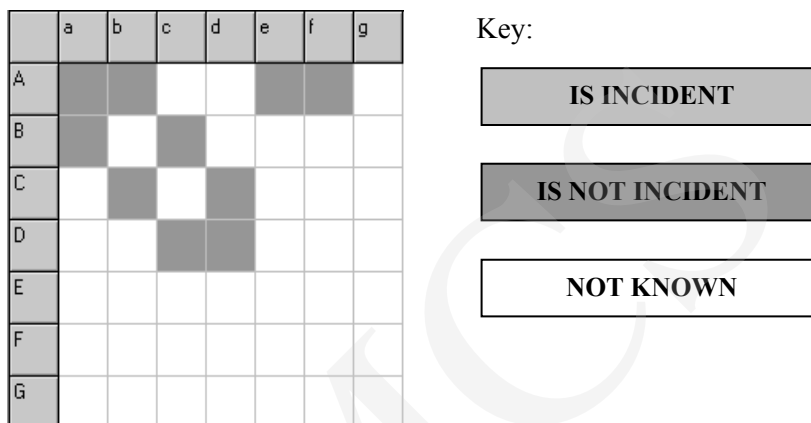


Fig. 2. Graphical meaning of L1

Implementing lemma L1 means modifying the program in such a way that it prevents from appearing the 10 incidences that can be brought to the above scheme by permutations.

The procedure including lemma L1 yields the configuration that is not a model because it still does not fulfil axioms A2 and A3:

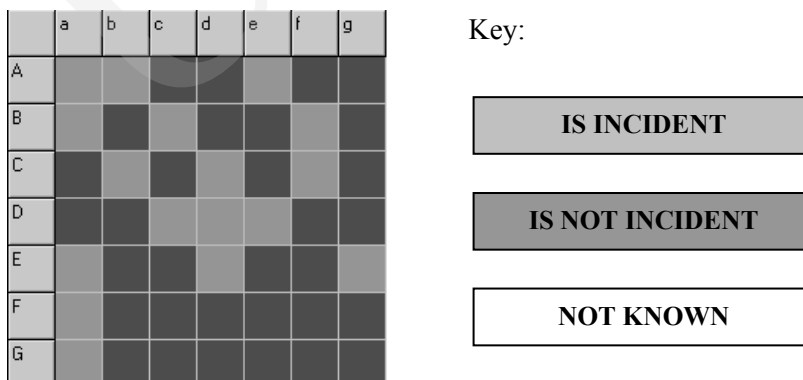


Fig. 3. Results of lemma L1 activity

As well as the next lemma **L2**:

$$\begin{aligned}
 \forall abcdefg \quad & a \sim | AB \vee b \sim | AC \vee c \sim | BD \vee d \sim | CD \vee a \sim | E \vee a \sim | F \vee \\
 & A = B \vee A = C \vee A = D \vee A = E \vee A = F \vee B = C \vee B = D \vee B = E \vee \\
 & B = F \vee C = D \vee C = E \vee C = F \vee D = E \vee D = F \vee E = F \vee \\
 & a = b \vee a = c \vee a = d \vee a = e \vee b = c \vee b = d \vee b = e \vee c = d
 \end{aligned}$$

It can be seen that the lemma is dual to L1. The procedures are, thus, analogous to the previous ones.

Next, covering both lemmas, we obtained the configuration that fulfils the axioms A1-A4:

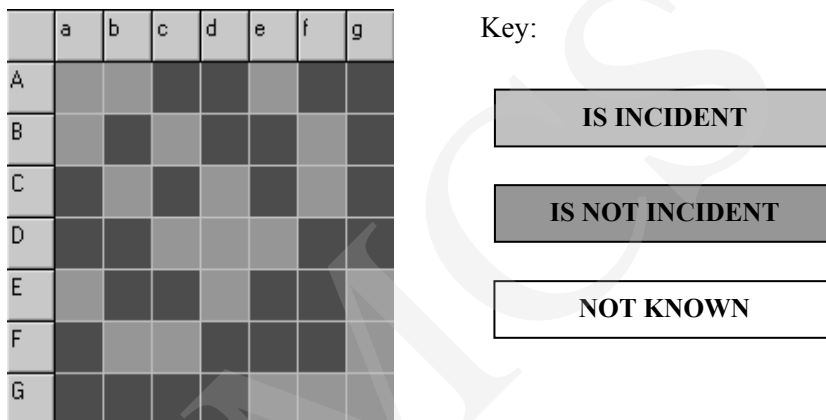


Fig. 4. Model of projective plane of order 2

### 5. Conclusions

It occurred that the dual lemmas L1 and L2 are a complement of axiomatics A1-A4 that is sufficient to construct the model of projective plane of order 2. That brings associations to dual theorems about the power of pencil [2.1] and chain [2.2].

After implementing the above theorems it occurred that they are equal to the generated pair of lemmas, assuming the initiation of the model. Due to L1 and L2 activity we obtain the same model as the one produced by L1 and [2.2], [2.1] and L2, [2.1] and [2.2]. It is interesting that lemmas L1 and L2 are more natural for the computer. It is also important that we reached the proofs of these lemmas. These are not, however, brilliant mathematical proofs but a laborious analysis of cases, yet they point out not only the possibilities of constructing plates but also of generating theorems.

A significant difference between lemmas L1, L2 and the theorems mentioned lies in generality: the lemmas can be applied only while generating the model of order 2. For the rest of orders, the automatically coined theorems would be different.

The program could be completely automatic if we apply a turn-back after any encountered contradiction. However, this would lead to an exponential complexity of the program. The present approach gives an interesting outcome of interactions between the computer which generates partial results and a mathematician – programmer who optimises the program operations without any

delay. In the future, using one of the symbolic languages (Lisp, Prolog), modifying of the program content would be automatic. Such a solution decreases the order complexity of an algorithm. Another worth examining approach deals with using parallelism or concurrency in the issue of projecting plates, especially those of higher orders.

### References

- [1] Kordos M., *Wykłady z historii matematyki*, Wydawnictwa Szkolne i Pedagogiczne, Warszawa, (1994), in Polish.
- [2] Clement W.H. Lam, *The search for a finite projective plane of order 10*, American Mathematical Monthly 98, (1991).
- [3] Kordos M., *Podstawy geometrii rzutowej i rzutowo – metrycznej*, Państwowe Wydawnictwo Naukowe, Warszawa, (1984), in Polish.
- [4] Bruck R.H., Ryser H.J., *The non-existence of certain finite projective planes*, Can. J. Math., (1949).
- [5] Grzegorzcyk A., *Zarys logiki matematycznej*, Państwowe Wydawnictwo Naukowe, Warszawa, (1961), in Polish.