Enumerating Projective Planes of Order 9 with Proof Verification

Daniel Dallaire
Department of Computer Science, University of Windsor
COMP-4960 - Research Project Report

***Abstract*—In this report, I detail the steps necessary to enumerate the projective planes of order 9, up to a certain symmetry. This is a computation which was previously done by Lam et al. which we repeat utilizing an SAT solver with the intent of also doing proof checking.**

I. Background

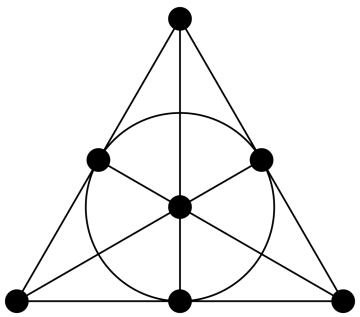
The objects we will be interested in this report are primarily finite projective planes, which we introduce here. These projective planes have a few different representations, the first of which is perhaps the easiest to understand. After giving this as the definition, we discuss another representation which will be more useful for the purposes of doing an exhaustive computer search for these objects.
Also playing a role in this work is the notion of a latin square and a group action, both of which we also introduce in this section.
Lastly, we describe the notion of a Boolean satisfiability problem or SAT problem as well.

To start, we define a projective plane of a given order n:

Definition: A *projective plane of order* n is a collection of n^2+n+1 lines and n^2+n+1 points such that:

1) every line contains $n+1$ points,
2) every point is on $n+1$ lines,
3) any two distinct lines intersect at exactly one point, and
4) any two distinct points lie on exactly one line.

One of the simplest examples of this is the *Fano plane*, which is a projective plane of order 2 (i.e. it has $2^2+2+1=7$ points and lines) as seen below.



From this definition, we can see that projective planes are objects which have a natural interpretation as an incidence structure—that is, two disjoint sets equipped with a relation between them which we call the incidence relation (Godsil & Royle).

With such structures we may associate a bipartite graph, whose parts in our case are the lines and points of the projective plane respectively, and whose edge set is determined by the incidence relation. Specifically, in this bipartite graph, we make an edge between a point and a line if and only if that point lies on the line.

Suppose now that we order the points and lines of the projective plane as: $p_1, p_2, \cdots, p_{n^2+n+1}$ and $\ell_1, \ell_2, \cdots, \ell_{n^2+n+1}$. Then we may represent the associated bipartite graph as an $(n^2+n+1)\times(n^2+n+1)$ 0-1 *incidence matrix*, in which each row represents a point and each column represents a line. The $(i,j)^{\text{th}}$ of this matrix will be 1 if the point p_i lies on the line ℓ_j, and it will be 0 otherwise. This definition is more workable for the purposes of using an SAT solver as we can encode the incidence matrix as $(n^2+n+1)^2$ Boolean variables.
To be able to utilize this representation, we must also translate the axioms of the projective plane (1) - (4) given above, in terms of the incidence matrix.

One can check that an $(n^2+n+1)\times(n^2+n+1)$ incidence matrix is one which arises from a projective plane of order n if the following properties are satisfied:

1) each column sum of the matrix is $n+1$,
2) each row sum of the matrix is $n+1$,
3) two distinct columns of the matrix intersect exactly once, and
4) two distinct rows of the matrix intersect exactly once.

Next, as mentioned, we also introduce latin squares here:

Definition: A $k\times k$ *latin square* is a $k\times k$ array consisting of the integers $1, 2, \cdots, k$ such that:

1) each row contains each of the numbers $1, 2, \cdots, k$ exactly once, and
2) each column contains each of the numbers $1, 2, \cdots, k$ exactly once.

For example, the following is a latin square of order 4

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \\ 3 & 4 & 1 & 2 \\ 4 & 1 & 2 & 3 \end{bmatrix}$$

The role that latin squares play in our search for projective planes will become more clear after our discussion in section III.

Next, we introduce the concept of a group action for which we should also first introduce what a group is.

Definition[Dummit & Foote]: A *group* is an ordered pair $(G, *)$ where G is a set and $*$ is a binary operation on G satisfying the following axioms:

1) $(a * b) * c = a * (b * c)$, for all $a, b, c \in G$, i.e. $*$ is associative,
2) there exists an element e in G, called an *identity* of G, such that for all $a \in G$ we have $a * e = e * a = a$,
3) for each $a \in G$ there is an element a^{-1} of G, called an *inverse* of a, such that $a * a^{-1} = e$.

One important example of a group is the symmetric group S_n, which is the set of permutations on the set $\{1, 2, \cdots, n\}$ where the group operation is function composition.

Groups are algebraic structures which are intended to model symmetry, particularly symmetry which arises from geometry. In our case we will be interested in certain groups which act on a set of projective planes, and others which act on a set of latin squares. This leads us to the notion of a group action below.

Definition[Dummit & Foote]: A *group action* of a group G on a set A is a map from $G \times A$ to A (written as $g \cdot a$, for all $g \in G$ and $a \in A$) satisfying the following properties:

1) $g_1 \cdot (g_2 \cdot a) = (g_1 g_2) \cdot a$, for all $g_1, g_2 \in G, a \in A$, and
2) $1 \cdot a = a$, for all $a \in A$.

Furthermore, given $a \in A$, we call $G \cdot a$ the *orbit* of a with respect to the action of G on A, which is given by:

$$G \cdot a = \{g \cdot a : g \in G\}$$

What is useful regarding this definition is the fact that the orbit of the element a is actually the equivalence classe of a with respect to the following equivalence relation:

$$a \sim b \iff \exists g \in G, g \cdot a = b$$

With this we will be able to describe several of the equivalence relations we use in the other sections.

Lastly, we need to describe SAT problems—which have fairly efficient heuristic algorithms to solve them.

Given a list of Boolean variables, a *literal* is one of these variables or one of these variables but negated. Then, a *clause* is a disjunction of literals. Given a list of clauses, we say the list is *satisfiable* if we can assign true and false values to variables such that every clause evaluates to be true.

An *SAT problem* is the problem of determining if a given list of clauses is satisfiable or not. This problem is known to be NP-complete, and thus there is no known polynomial time algorithm which solves it. However, there are good heuristic algorithms which solve the problem like MapleSAT.

II. Problem Overview

With this background established, we now give more details about the problem we're interested in. As we discussed, we can represent a hypothetical projective plane of order n as an $(n^2 + n + 1) \times (n^2 + n + 1)$ incidence matrix satisfying certain properties. Thus for the order 9 case, we're interested in enumerating the 91×91 incidence matrices (note: $91 = 9^2 + 9 + 1$) satisfying certain properties, *up to a certain symmetry*. We now comment on what this symmetry is.

Notice that given a fixed projective plane of order 9, one can obtain a distinct, but similar, projective plane by relabelling some of the points, and relabelling some of the lines. Realistically however, these "new" planes are the same as the original one, thus we may wish to ignore these extra planes in our search if possible. These relabellings correspond to column and row permutations of the incidence matrix.
Consequently, we can give a group theoretic interpretation of this: Let G be the group $S_{91} \times S_{91}$ (where S_k is the symmetric group on k letters), then we may view the first component of this group as the row permutations, and the second as the column permutations. In this way, we have a group action of G on the set of incidence matrices corresponding to projective planes. Then we can say that two planes are equivalent if they are in the same *orbit* under this action. Two planes are in the same orbit, if one can be obtain from the other via action of the group G, or rather, by applying column and row permutations.

Searching for only a representative of each orbit makes our search drastically more feasible. As mentioned before, this computational search was first done by Lam. Before this search was done, there were four distinct projective planes of order 9 which were known to exist. Lam's search showed that these were the only four planes. We repeat this search using an SAT solver and by exploiting the symmetry group mentioned above to reduce the search space. By applying the elements of the symmetry group to a hypothetical projective plane, we can fix certain structure for the plane. We outline this approach in our next section.

III. Structure of Planes of Order 9

Here we detail the structure we impose on the incidence matrix of our hypothetical projective plane for the purposes

of reducing the search space of our problem.
The first important thing to note is that a hypothetical projective plane must contain a *triangle*—that is, a set of three non-colinear points. We may suppose that the first three points p_1, p_2, and p_3 form this triangle, and furthermore that ℓ_1 is the unique line joining p_2 and p_3, ℓ_2 is the unique line joining p_1 and p_2, and lastly that ℓ_3 is the unique line joining p_1 and p_3. We may impose this order simply by applying the needed row and column permutations to the incidence matrix. With this ordering of the points and lines, the upper left 3×3 sub-matrix of the incidence matrix will look like:

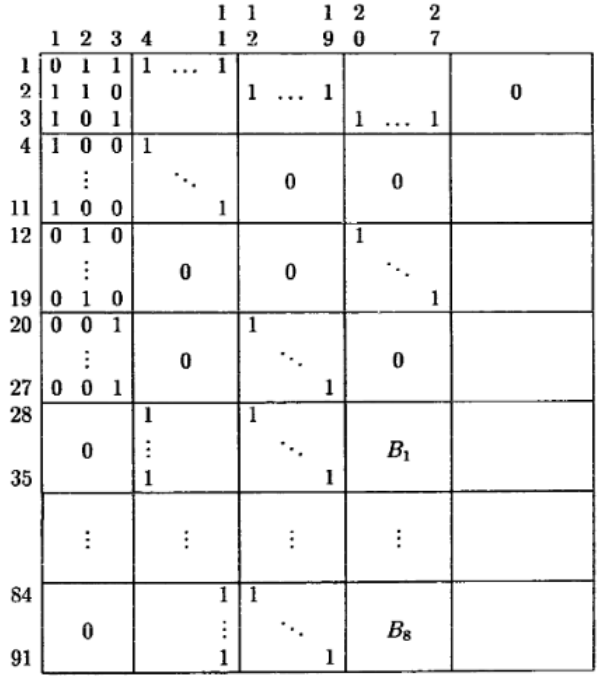
$$\begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

Once this is done, we know by the first two axioms of a projective plane, that there should be 8 more 1's in each of the first three rows and columns. By applying further row and column permutations, we may impose a staircase like structure on these entries. For example, we can make the first three rows in columns 4 to 27 will look like:

$$\left[\begin{array}{cccc|cccc|cccc} 1 & 1 & \cdots & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 1 & 1 & \cdots & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 1 & 1 & \cdots & 1 \end{array}\right]$$

with the entries in columns 28 and above being 0 in those rows.

Then, using further column and row permutations, we may fix many of the entries in the first 27 columns of the incidence matrix. Those entries in columns 20-27 and rows 28-91 will not be fixed, however they have a nice structure.



Form of the normalized incidence matrix from Lam's 1991 paper

One can check that the 8×8 blocks $B_1, B_2, \cdots, B_8$ given in the above figure will be 8×8 permutation matrices. Consequently, these blocks correspond to permutations $\pi_1, \pi_2, \cdots, \pi_8 \in S_8$. Furthermore, we know that we can't have $\pi_i(k) = \pi_j(k)$ where $i \neq j$ since otherwise we would have two distinct rows intersecting twice.
This property ensures that we may encode this set of 8 permutations as an 8×8 latin square.

Thus, For a given 8×8 latin square, we may obtain the 27 columns of a partial projective plane, and in this way, all such partial planes may be obtained. Thus our approach for generating the projective planes of order 9 will begin with generating the latin squares of order 8. However you may note that two distinct latin squares can sometimes give us equivalent partial planes. Thus we will be interested in a certain equivalence classes of latin squares, which we detail in the next section.

IV. Step 1: Latin Squares

As the structure we identified in the last section suggests, a good first step to generate the projective planes of order 9 will be to first generate all 8×8 latin squares up to a certain equivalence. More specifically, we will try to generate a representative of each *main class* of latin squares. Two 8×8 latin squares are said to belong to the same main class, if they differ only by a permutation of the rows, columns, or symbols, or also possibly an exchange of the roles of these three things. Thus we can view the main class of a latin square as its orbit under the natural action of the group $(S_8 \times S_8 \times S_8) \rtimes S_3$ on the set of latin squares.

To generate all such representatives, we reduce the problem to SAT, and then utilize a SAT solver to solve it. To reduce the computation, we also provide a mechanism for doing isomorphism checking along the way; that is, if two partial squares belong to the same main class, we can throw away the duplicates to reduce computation time. We first detail our reduction to SAT.

We will encode an arbitrary 8×8 latin square $(A_{i,j})$ as a Boolean satisfiability problem as follows: We will have 8^3 Boolean variables $\ell_{i,j,k}$ $(1 \leq i, j, k \leq 8)$ where $\ell_{i,j,k}$ will be 1 if and only if $A_{i,j} = k$.

Now we must translate the requirement that rows and columns contain the numbers $1, 2, \cdots, 8$ exactly once into many Boolean clauses.

First, for each $1 \leq k \leq n$, we need k to appear at least once in each row and column. From this we get the following clauses for $1 \leq i, j \leq n$:

$$\ell_{i,1,k} \vee \ell_{i,2,k} \vee \cdots \vee \ell_{i,n,k} \text{ and } \ell_{1,j,k} \vee \ell_{2,j,k} \vee \cdots \vee \ell_{n,j,k}$$

Additionally, we need there to be at most one occurrence of k in each row and column. That is, for any $1 \leq i, j \leq n$ and $a_1 \neq a_2$ we have the clauses

$$\ell_{i,a_1,k} \Rightarrow \neg\ell_{i,a_2,k} \equiv \neg\ell_{i,a_1,k} \vee \neg\ell_{i,a_2,k}$$

and

$$\ell_{a_1,j,k} \Rightarrow \neg\ell_{a_2,j,k} \equiv \neg\ell_{a_1,j,k} \vee \neg\ell_{a_2,j,k}$$

This gives us our reduction to SAT. With this, we use a SAT solver to generate the latin squares in several steps.

We first normalize the latin squares by insisting that the numbers $1, 2, \cdots, 8$ appear in order in the first column and row. We can do this by simply permuting the columns and rows needed. Next, we extend from row 1, to row 2, 3, 4, and then 8 each in their own steps, while also doing isomorphism removal after each step.

Then we perform isomorphism removal by translating our latin squares to graphs and then checking if the corresponding graphs are isomorphic to each other. We do this in such a way that the corresponding graphs will be isomorphic if and only if the original latin squares belonged to the same main class.

Our translation of a latin square $(A_{i,j})$ to a graph is as follows:
The vertex set is:

$$\{V_{i,j} | 1 \leq i, j \leq n\}$$

Then we draw an edge between $V_{i,j}$ and $V_{i',j'}$ if:

$$i = i',\ j = j',\ \text{or } A_{i,j} = A_{i',j'}$$

One can check that this translation gives us the desired property.

This step of generating the latin squares is done using a SAT solver called MapleSAT. Once completed, there were 283657 representatives of the main classes of latin squares, which serve as the starting point for the next step in our computation.

V. Step 2: Column 40 Extension

The next major step in our enumeration of the projective planes to extend each of the 283657 partial planes (each with 27 columns to start) to partial planes with 40 columns. We accomplish this by once again using an SAT solver. However, we take a slightly different approach than we did with the latin squares since the axioms of a projective plane don't translate so nicely into SAT.

A partial plane $(A_{i,j})$ has a natural encoding into a set Boolean variables by making a variable $a_{i,j}$ for each entry.

Our partial encoding of the axioms of the projective plane into SAT will involve only the axioms (3) and (4): That two distinct rows / columns intersect exactly once.

First, the requirement that they intersect at most one is given by the *quad free* clauses:

$$\neg a_{i,j} \vee \neg a_{i',j} \vee \neg a_{i,j'} \vee \neg a_{i',j'}$$

for $i < i'$ and $j < j'$.

They are called such since if one of these clauses is false, we will have a rectangle in the incidence matrix whose corners are 1's, which is exactly what happens if two rows / columns intersect more than once.

Then, we only check that a given column intersects the first 19 columns of the partial plane each at least once. This is because these first 19 columns are the only ones which are fixed among all partial planes. If column $j \leq 19$ has 1's in the entries $(i_1, j), (i_2, j), \cdots, (i_{10}, j)$, then for $j' > 27$, we include the clause:

$$a_{i_1,j'} \vee a_{i_2,j'} \vee \cdots \vee a_{i_{10},j'}$$

which ensures that column j' intersects column j at least once.

In addition to these clauses, we can also impose some structure on the first 19 rows (similar to the first 19 columns) to remove symmetry. Once this is done as well, we can use MapleSAT on the problem to generate all possible extensions to partial planes with 40 columns, of which there will actually be very few.

This step needed to be done utilizing Compute Canada's servers to split the problem over 100's of processors. For the few partial planes that it generates, these can then simply be extended to the full 91 columns, and then they are verified to indeed be projective planes.

VI. Proof Verification

The SAT solver MapleSAT can return proofs which can be checked using a proof verifier (such as DRAT-trim) as part of the output for their exhaustive search.

Since in step 2 we split the list of 27 column partial planes over many processors, we actually end up with a separate proof for each one, which all need to be verified independently. This step will also need to be done using Compute Canada.

This is done for the column 40 extensions, and then also for the single extension from column 40 to the full 91 columns, which can be done on a local machine.

VII. Future Work

The proof checking aspect of the project is still on-going and will be the focus going forward. Another enhancement

which can made to the project is using a more restrictive encoding into SAT for the projective plane problem. We only partially used axioms (3) and (4) in the encoding, however we can create more clauses to fully encode the requirement that two columns intersect at least once by having a different set of clauses for each column 40 extension. This makes things slightly more complicated compared to the current setup, since we currently have one set of clauses which describes the problem for any given column extension.

VIII. References

Dummit, D.; Foote, R. Abstract Algebra.

Godsil, C.; Royle, G. Algebraic Graph Theory.

Lam, C.; Kolesova, G.; and Thiel, L. 1991. A computer search for the projective planes of order 9. In Discrete Mathematics 92 (1991) 187-195.