DISSERTATION

CLASSIFICATION ALGORITHMS FOR GRAPHS, DIGRAPHS, AND LINEAR SPACES

Submitted by

Abdullah Al-Azemi

Department of Mathematics

In partial fulfillment of the requirements
For the Degree of Doctor of Philosophy
Colorado State University
Fort Collins, Colorado
Summer 2007

COLORADO STATE UNIVERSITY

July 6, 2007

WE HEREBY RECOMMEND THAT THE DISSERTATION PREPARED UNDER OUR SUPERVISION BY **ABDULLAH AL-AZEMI** ENTITLED "**CLASSIFICATION ALGORITHMS FOR GRAPHS, DIGRAPHS, AND LINEAR SPACES**" BE ACCEPTED AS FULFILLING IN PART REQUIREMENTS FOR THE DEGREE OF **DOCTOR OF PHILOSOPHY**.

Committee on Graduate Work

Alexander Hulpke

Dr. Alexander Hulpke

Robert A Liebler

Dr. Robert Liebler

Ross McConnell

Dr. Ross McConnell

Anton Betten

Adviser: Dr. Anton Betten

Simon Tavener

Department Head: Dr. Simon Tavener

ABSTRACT OF DISSERTATION

Classification Algorithms For Graphs, Digraphs, and Linear Spaces

Combinatorial incidence structures like graphs, digraphs, and linear spaces are defined modulo an isomorphism relation. Typically we are interested in determining complete systems of representatives of the isomorphism classes, in order to test conjectures or to prove existence or non-existence of examples for new theorems.

In this thesis, we present classification algorithms for graphs, digraphs and incidence structures. We discuss both the use of invariants and the use of partition backtracking for solving the isomorphism problems of $\{0, 1\}$-matrices.

After that, we consider the inverse problem of finding all structures for a given invariant. This leads to the composition principle for incidence structures and eventually to the computation of all $8,592,194,823$ linear spaces on 13 points.

Abdullah Saad Al-Azemi
Mathematics Department
Colorado State University
Fort Collins, CO 80523
Summer 2007

ACKNOWLEDGMENTS

This thesis has been contributed by a number of people, either directly or indirectly. First, I would like to express my deepest gratitude to Dr. Anton Betten, my thesis supervisor, for guiding me to the areas of combinatorial classification and the world of science in general. His encouragements and extensive collaboration were one of the main reasons that this work is done. Also, I wish to thank each of Dr. Alexander Hulpke, Prof. Robert Liebler, and Dr. Ross McConnell for all the advice and care they extended to me, or to the production of the presented thesis.

Finally, I would like to thank all of my family, especially my Mother and Wife, and friends, especially Basil Zailaie and his Wife, for their support and encouragements which make this work much easier to accomplish.

Table of Contents

Chapter 1

Introduction

In graph theory, given a collection of properties, decide whether there exist a graph satisfying those properties. This is called a *classification problem for graphs* in which one is asked to list, up to some criterion of equivalence relation, all the graphs that have the desired properties. In such a list, no graph can be obtained from the other in any way. In otherwords, the classification problem is a counting problem in which the task is to count, upto equivalence (*isomorphism*), the number of distinct graphs satisfying the properties. However, we expand our considerations to larger set of objects called *incidence structures* which might be described as an arrangement of finite sets of points and blocks so that some prescribed properties are satisfied.

Solving such a problem is of interest for practical and theoretical reasons. According to Ronald C. Read [69], examination of a classified lists may suggest conjectures, find a counter example to conjectures, or indicate possible lines of investigation.

The *isomorphism problem* consists in deciding whether two given structures are isomorphic, i.e. whether there is a bijective mapping (*permutation*) such that one structure is produced from the other.

Algorithmic methods are widely used for solving existence and classification problems [11, 38, 55, 66, 71, 81]. One of the used methods is the *exhaustive* *search* which considers all candidates solutions and is guaranteed to find a solution if one exists. This method is also used for demonstrating the nonexistence of particular structure. The nonexistence of a finite projective planes of order 10 is may be the most notable such result to date [56].

In Particular, a classification of structures of interest is not an easy task to achieve. In such a problem, one starts by listing all possible structures that satisfy some given set of constriants, and then eliminate identical (isomorphic) objects among that list, keeping only one object among every isomorphism class. Despite many differences, this is what is called a "*classical method*".

Assuming that we want to enumerate all graphs on n vertices that posses some given properties. Then, the classical method starts to produce larger graphs from an existing list of smaller graphs. For instance, we create a list of graphs having $i+1$ vertices which satisfy the set of properties from smaller graphs with i vertices. Clearly, we shall have every graph having $i+1$ vertices in this way. However, we would get a quite large number of duplicates for large n. It is therefore advisable to eliminate duplicates in the new list once the structure is constructed. It is then necessary to have a method for such elimination at each step of the construction procedure.

In fact, there are two more efficient algorithms which have been developed for that manner. Namely, *generation by canonical representative* or *orderely generation* due to C. Read [69] and I. A. Faradžev [23] independently in (1978), and the other method, which is the one used in the presented thesis, is *generation by canonical augmentation* due to Brendan D. McKay [64] in (1998). For applications of those method see [11, 12, 25, 45, 47, 49, 51, 64, 72].

Moreover, Laue and Kerber together with collaborators have used extensively the method of *homomorphisms of group actions* [7, 53, 57] in classification problems and that this does not apply in many cases. In other words, this method only applies if the object has a well defined subobject.

In this thesis, we used the framework invented in the method by McKay

[64] in classifying some structures of interests. We use the general theory in McKay's paper to construct a specific theory which is then applicable to construct incidence structures in a row by row strategy. Moreover, our concern in this thesis is to work out the details that are hard to find in other works on generation algorithms and as a result a classification result is obtained.

In what follows, we give a brief summary on the main interests of the thesis and what each chapter of the thesis is about.

In algebraic setting, the isomorphism classes of structures correspond to the orbits of a group action. The idea of isomorph-free exhaustive generation is to generate, without isomorph, all structures satisfying a collection of properties.

Consequently, this idea corresponds to produce a set of structures satisfying a collection of constraints that contains exactly one element out of each isomorphism class. This set is then what we call an orbit transversal.

Algebraically, the fundamental problem to be considered in this thesis is that we have a group G that acts on a finite set X of incidence structures, and we are asked to produce an orbit transversal for the action of G on X, denoted by $\mathcal{T}(G, R)$.

The principle of homomorphisms of group actions have extensively considered, for instance by authors like Kerber and Laue [7, 53, 57], in solving problems related to the construction of some combinatorial structures. The idea is to solve an orbit transversal for a secondary action, and then use the produced transversal to solve the primary transversal problem where both actions are connected in the sense of homomorphisms of group actions.

In particular, we consider projection maps as homomorphisms of group actions. Let G act on two finite sets X and Y, and let $R \subseteq X \times Y$ be a G-invariant relation with G acting on R coordinatewise. Assuming that a transversal for G-orbits on X is given (or trivially can be constructed), we consider the problem of constructing a transversal for the G-orbits on Y, $\mathcal{T}(G, Y)$, step by step. First, we consider the step of constructing a transversal

$\mathcal{T}(G,R)$ by using a given transversal $\mathcal{T}(G,X)$. This step is called *lifting orbits step.* The second step is called the *projecting orbits step* which depends on the ideas of isomorph-rejection techniques. Those techniques are used in eliminating duplicate structures, and the name of such techniques was used first by SWIFT [78].

The organization of the thesis follows. First, Chapter 2 on page 6 gives the needed background, notations, and some results that are used throughout the thesis.

Second, in Chapter 3 on page 52, we discuss the algebraic concepts of finite group actions. Moreover, we construct an algorithm which enables us to construct an orbit transversal $\mathcal{T}(G,Y)$ if given an orbit transversal $\mathcal{T}(G,X)$. Also, we present the isomorph-rejection theory of what so called orderly generation due to Faradžev [23] and Read [69] (in the 1970's, independently).

Third, in Chapter 4 on page 74, we define the class of incidence structures X in a way that we can apply the considered theory of Chapter 3 on page 52. Moreover, the idea of generation by induction is developed in that chapter as well. Also, we consider some examples of constructing incidence structures like regular graphs with a collection of properties.

Fourth, in Chapter 5 on page 103, we survey isomorphism invariants, and the partition refinements. Moreover, we use a method called *TDO-method*, developed by D. Betten and M. Braun [9], in the partition refinement procedure.

In Chapter 6 on page 118, we explain the ideas of partition backtrack using the TDO language and use the so called *derived TDO* to approximate the orbits of the automorphism group of a given incidence structure. Also, we concentrate more on methods for computing a canonical labeling map along with the automorphism group of a given incidence structure. We then reconsider the concepts of isomorph-rejection techniques by considering the ideas of canonical augmentations, due to McKay [64].

Finally, in Chapters 7 on page 150 and 8 on page 176, we consider the generation algorithm, which was constructed in the thesis by using isomorph-rejection techniques related to canonical augmentation, of two families of incidence structures which are linear spaces on 13 points and normally regular digraphs, and thereby new classification results were achieved.

Chapter 2

Preliminaries

This chapter gives a brief introduction to graphs and incidence structures. Moreover, we give some definitions and some notations that we will use throughout the thesis.

For a general introduction to discrete mathematics, see [34, 35]. For an introduction to **graph theory**, we refer the reader to [14, 33, 41, 59, 71, 77]. More on incidence structures like **designs**, **configurations**, and **linear spaces** can be found [42, 52].

2.1 Graphs

A (finite) **graph** $\mathcal{G}$ is a pair (V, E), where $V(\mathcal{G}) = \{a, b, c, \dots\}$ is a finite set of elements called **vertices** or **nodes**, and $E(\mathcal{G})$ is a set of unordered pairs of distinct vertices in $V(\mathcal{G})$ called **edges**. Notice that we simply write V and E if $\mathcal{G}$ is clear from the context.

The number of vertices in a graph is called its **order**, denoted by $|V|$, and the number of edges, its **size** denoted by $|E|$. If $\alpha = \{x, y\} = \{y, x\}$ is an edge of $\mathcal{G}$, then we say that α **joins** x and y. Also, we say that x and y are **adjacent** or **neighboring**. Moreover, we say that the vertex x and the edge α are **incident**. In that case, we write $x \sim y$ to denote that x and y are adjacent.

The **neighborhood** of a vertex x, denoted by $N(x)$, is the set of all vertices adjacent to x, i.e. $N(x) = \{y \in V(\mathcal{G}) \mid x \sim y\}$. The **degree** of a vertex x, denoted by $deg(x)$, is the number of vertices adjacent to x. Thus, $deg(x) = |N(x)|$ where "$|\,.\,|$" denotes the **cardinality**.

A graph in which all vertices have the same degree k is said to be **regular** or **k-regular**.

Example 2.1.1. A graph $\mathcal{G} = (V, E)$ with

$$V = \{1, 2, 3, 4, 5, 6\},$$

and

$$E = \{\{1,2\}, \{1,3\}, \{1,4\}, \{2,3\}, \{2,5\}, \{3,6\}, \{4,5\}, \{4,6\}, \{5,6\}\} \quad (2.1)$$

is a graph of order 6 and size 9. It can be drawn as in Figure 2.1.1.

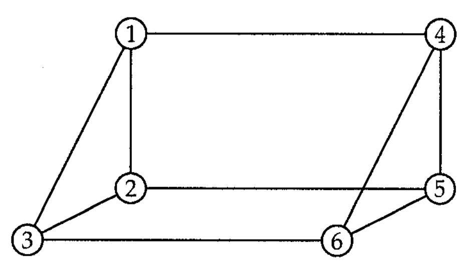


Figure 2.1.1: A drawing of Example 2.1.1, with 6 nodes and 9 edges.

■

As each vertex in the graph in Figure 2.1.1 has precisely 3 neighbors, the graph is 3-**regular** or **cubic**. Thus, the graph of Example 2.1.1 is a cubic graph of order 6.

For a finite set A we denote by $\mathcal{P}(A)$ (the **power set**) the set of all subsets of A. Moreover, if i is an integer, we denote by $\mathcal{P}_i(A)$ the set of all i-subsets of A. For sets A and B, we let $A \setminus B$ be the set of elements of A which are not in B.

If $\mathcal{G} = (V, E)$ is a graph, then the set of edges E can be considered a subset of $\mathcal{P}_2(V)$. If $E = \mathcal{P}_2(V)$, then the graph $\mathcal{G}$ is called **complete**. If $E = \emptyset$, then the graph $\mathcal{G}$ is said to be **empty**. The complete graph on n vertices is denoted K_n (from the German word for complete "komplett").

If $\mathcal{G}$ is a graph, then the **complement** of $\mathcal{G}$ is the graph $\overline{\mathcal{G}} = (V, \mathcal{P}_2(V) \setminus E)$. That is, $\overline{\mathcal{G}}$ has the same vertex set as $\mathcal{G}$ and $\alpha \in E(\overline{\mathcal{G}})$ if and only if $\alpha \notin E(\overline{\mathcal{G}})$ for any pair $\alpha = \{x, y\} \subseteq V$. For instance, the complement of a complete graph is the empty graph.

A sequence that represents the degree of every vertex in a given graph in a decreasing order is called a **degree sequence**.

Definition 2.1.1. *A* ***path*** *is a sequence of vertices such that consecutive pairs are connected. We require that the vertices are pairwise distinct except possibly for the first and the last vertices, which may be the same. The first vertex is called the* ***start vertex****, and the last vertex is called the* ***end vertex****. A* ***cycle*** *is a path such that the start vertex and the end vertex are the same. The length of a path (a cycle) in a graph is the number of edges in that path (cycle). We do not consider the path of length zero or two to be a cycle, and thus the length of a cycle is always greater than or equal to* 3*. For instance, the smallest cycle in our consideration is of length* 3*, namely a triangle (a cycle of three vertices).*

Definition 2.1.2. *Two graphs* $\mathcal{G}_1$ *and* $\mathcal{G}_2$ *are said to be* ***isomorphic*** *if there exists a bijection* $f : V(\mathcal{G}_1) \to V(\mathcal{G}_2)$ *such that*

$$\{x, y\} \in E(\mathcal{G}_1) \Longleftrightarrow \{f(x), f(y)\} \in E(\mathcal{G}_2).$$

for all $x, y \in V(\mathcal{G}_1)$*. Such a bijection then is called an* ***isomorphism*** *of* $\mathcal{G}_1$ *onto* $\mathcal{G}_2$*. An isomorphism* f *of* $\mathcal{G}_1$ *onto itself is called an* ***automorphism****. All such automorphisms form a group called the* ***automorphism group****, denoted by* $Aut(\mathcal{G}_1)$*.*

If $\mathcal{G}_1$ is isomorphic to $\mathcal{G}_2$, then also $\mathcal{G}_2$ is isomorphic to $\mathcal{G}_1$. It is therefore common practice to say that two graphs are isomorphic, and write $\mathcal{G}_1 \cong_f \mathcal{G}_2$ to denote that the two graphs are isomorphic via f. However, we simply write $\mathcal{G}_1 \cong \mathcal{G}_2$ if f is clear from the context.

In simple words, we say that two graphs are isomorphic if and only if apart from the labeling of their vertices they are the same.

A necessary condition for two graphs to be isomorphic is that they have the same number of vertices and the same number of edges. Moreover, they must have the same degree sequence. Example 2.1.2 shows two graphs which look different but in fact are the same.

Given two graphs on n vertices, the naive algorithmic approach to solve a graph isomorphism is to generate all $n!$ permutations of the vertices and test if they induce an isomorphism between the graphs. Of course, this is not efficient as we will see later on.

Example 2.1.2. Figure 2.1.2 shows two isomorphic graphs along with an isomorphism f that maps one onto the other.

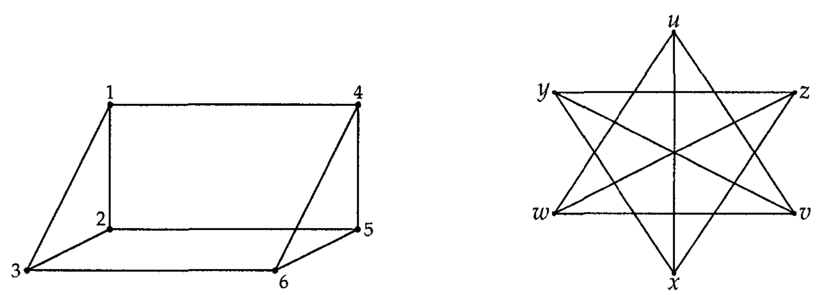


Figure 2.1.2: Two isomorphic graphs.

The isomorphism is given by the following mapping f:

$$f(1) = u, \qquad f(2) = v, \qquad f(3) = w,$$

$$f(4) = x, \qquad f(5) = y, \qquad f(6) = z.$$

■

Definition 2.1.3. *Let $\mathcal{G} = (V, E)$ be a graph of order n with $|V| = \{v_1, v_2, \ldots, v_n\}$. Subject to this labeling, the adjacency matrix of $\mathcal{G}$ is the $n \times n$ $\{0, 1\}$-matrix $A = (\, a_{ij} \,)$ defined for all $1 \leq i, j \leq n$ by*

$$a_{ij} = \begin{cases} 0, & \text{if } \{v_i, v_j\} \notin E; \\ 1, & \text{if } \{v_i, v_j\} \in E. \end{cases}$$

Note that an adjacency matrix for a graph is always symmetric, i.e. $a_{ij} = a_{ji}$, and that the diagonal entries a_{ii} are always equal to zero.

Example 2.1.3. The **adjacency matrix** of the graph of Example 2.1.1 is

$A =$

v	1	2	3	4	5	6
1		x	x	x		
2	x		x		x	
3	x	x				x
4	x				x	x
5		x		x		x
6			x	x	x	

For our convenience, we leave the zero entries in the matrix "empty" and replace any 1 entry with an "x" so that vertices i and j are adjacent if and only if $a_{ij} = 1$ in matrix A.

■

Notice that the adjacency matrix A has (exactly) three entries 1 in each row and each column. This correspond to the fact that $\mathcal{G}$ is cubic.

A second way to represent a graph is by means of its **incidence matrix**.

Definition 2.1.4. *Let $\mathcal{G} = (V, E)$ be a graph of order n with $|V| = \{v_1, v_2, \ldots, v_n\}$. Assume that $\mathcal{G}$ is of size t with $E = \{e_1, e_2, \ldots, e_t\}$. Then, an incidence matrix* *of $\mathcal{G}$ is an $n \times t$ $\{0,1\}$-matrix $A = (a_{ij})$ defined for all $1 \le i \le n$ and $1 \le j \le t$ by*

$$a_{ij} = \begin{cases} 1, & \textit{if } v_i \textit{ and } e_j \textit{ are incident;} \\ 0 & \textit{otherwise.} \end{cases}$$

We remark that some authors define the transpose of A to be the incidence matrix of $\mathcal{G}$.

Example 2.1.4. The incidence matrix A for the graph of Example 2.1.1 is

$A =$

ve	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9
v_1	x	x	x						
v_2	x			x	x				
v_3		x		x		x			
v_4			x				x	x	
v_5					x		x		x
v_6						x		x	x

where rows correspond to vertices and columns correspond to edges (in the order assign by Equation 2.1).

■

A more general class of graphs is the following:

Definition 2.1.5. *A* ***directed graph*** *D is a pair (V, E) where $V = \{a, b, c, \dots\}$ is a finite set of elements called vertices and E is a set of* ***ordered pairs*** *of distinct vertices in V called* ***directed edges****. We require that $E \cap E^T = \emptyset$, where $E^T = \{(y, x) \mid (x, y) \in E\}$. Two vertices $x, y \in V(D)$ are adjacent if there is a directed edge from x to y (or from y to x), and we say that x dominates y (or y dominates x) and denoted by $x \to y$ (or $y \to x$), respectively.*

Two directed graphs D_1 and D_2 are isomorphic if there exists a bijection $f : V(D_1) \to V(D_2)$ such that

$$(x, y) \in E(D_1) \iff (f(x), f(y)) \in E(D_2).$$

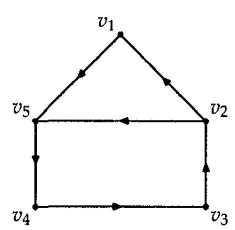
for all $x, y \in V(D_1)$. Thus f is an isomorphism.

Then, the adjacency matrix $A = (a_{ij})$ of a digraph $D = (V, E)$ with n vertices is an $n \times n$ $\{0, 1\}$-matrix where

$$a_{ij} = \begin{cases} 1, & \text{if } v_i \rightarrow v_j; \\ 0 & \text{otherwise.} \end{cases}$$

If a vertex v dominates another vertex u in a digraph D, then we say that u is an **out-neighbor** for v. The **out-degree** of the vertex v is the number of all vertices in D that v dominates. If v dominates u, v is an **in-neighbor** of u and the number of all vertices in D dominating u is called **in-degree**.

Example 2.1.5. Figure 2.1.3 shows a digraph D on 5 vertices together with its adjacency matrix A.



$A =$

	v_1	v_2	v_3	v_4	v_5
v_1					x
v_2	x				x
v_3		x			
v_4			x		
v_5				x	

Figure 2.1.3: A digraph D with its adjacency matrix

■

Definition 2.1.6. *A graph $\mathcal{H}$ is a* ***subgraph*** *of a graph $\mathcal{G}$ if $V(\mathcal{H}) \subseteq V(\mathcal{G})$ and $E(\mathcal{H}) \subseteq E(\mathcal{G})$ such that for all $\{x, y\} \in E(\mathcal{H})$ both x and y are in $V(\mathcal{H})$. A subgraph is* ***spanning*** *if $V(\mathcal{G}) = V(\mathcal{H})$.*

A **spanning cycle** in a graph is called **Hamiltonian cycle**. The graph in Figure 2.1.1, has a Hamiltonian cycle as it can be seen by the cycle $1-2-3-6-5-4-1$, where we start from vertex 1 and stop at the same vertex such that no vertex is visited more than once. On the other hand, the graph in Figure 2.1.4 has no such cycle. It is called the **Petersen graph**.

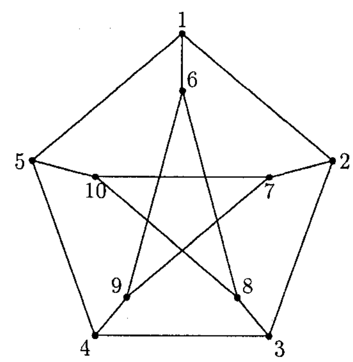


Figure 2.1.4: The Petersen Graph.

Recall from Definition 2.1.1 that a cycle consists of at least 3 vertices (and edges).

Definition 2.1.7. *The* ***girth*** *of a graph $\mathcal{G}$ is the length of the shortest cycle contained in $\mathcal{G}$. If $\mathcal{G}$ does not contain any cycle, its girth is defined to be infinity.*

The girth of the Petersen graph of Figure 2.1.4 is 5, whereas the girth of the graph of Figure 2.1.1 is 3.

Definition 2.1.8. *For a graph $\mathcal{G}$, the* ***subgraph induced*** *by $W \subseteq V(\mathcal{G})$ has W as its vertices set and its edge set consists of all edges $\{x, y\} \in E(\mathcal{G})$ for which $x, y \in W$ holds.*

For more details about cliques, we refer the reader to [52, 67]. Algorithmic techniques that are related to cliques can be found in [50].

Two vertices in a graph $\mathcal{G}$ are said to be **connected** if there exists a path in $\mathcal{G}$ connecting the two vertices. A graph $\mathcal{G}$ is **connected** if any two vertices are connected. The subgraphs of $\mathcal{G}$ largest in size which are connected are the connected components of $\mathcal{G}$. Every graph $\mathcal{G}$ is the union of its components.

Definition 2.1.9. *A* ***tree*** *is a connected graph without cycles. A* ***spanning tree*** *for a connected graph $\mathcal{G}$ is a subgraph T that is spanning and a tree.*

Definition 2.1.10. *A* ***rooted tree*** *is a pair (T, r) where $T = (V, E)$ is a tree with vertex set V and edge set E, and $r \in V$ is a root of T.*

If x and y are two vertices of a rooted tree (T, r), then we say that x is a **descendant** of y if y occurs on the path connecting x to the root r. We say that x is an **ancestor** of y if y is a descendant of x.

The **parent** of a non-root vertex y denoted by $p(y)$ is the vertex that is adjacent to y in the path connecting y to r. In addition, if $p(y) = x$, then y is said to be a child for x, and we write $C(x)$ for the set of all children of x. Note that the root r has no parent.

Two vertices having the same parent are called **siblings**. A vertex with no children is called a **leaf**. The **depth** or **level** of a vertex x is the length of the path connecting x to the root. The **height** of a rooted tree is the maximum depth of a vertex. The **subtree** rooted at a vertex x is the rooted tree with root x induced by all descendants of x.

Definition 2.1.11. *An* ***unordered partition*** *Π of a finite set S is a collection $S_1, S_2, \ldots, S_m$ of pairwise disjoint subsets of S, called* ***cells or classes****, such that each element of S is in exactly one of these subsets:*

$$S = \bigcup_{l=1}^{m} S_l, \quad \text{and} \quad S_i \cap S_j \;=\; \emptyset, \; (i \neq j). \tag{2.2}$$

On the other hand, an ***ordered partition*** *Π' of S is a sequence $S_1, S_2, \ldots, S_m$ of pairwise disjoint subsets of S that satisfies 2.2.*

Note that, we write $\{S_1, S_2, \ldots, S_m\}$ for unordered partition Π, while we write $\{S_1 \mid S_2 \mid \ldots \mid S_m\}$ for ordered partition Π'. We remark that for finite sets A and B, partitions $\{A, B\}$ and $\{B, A\}$ are the same while partitions $\{A \mid B\}$ and $\{B \mid A\}$ are different.

Let $\Pi_1 = \{C_0, C_1, \ldots, C_r\}$ be a partition of a finite set S. The subsets C_i's are called **cells** or **classes** of the partition Π_1 for $i = 0, 1, \ldots, r$. Let $\Pi_2 = \{D_0, D_1, \ldots, D_s\}$ be another partition of S. If every cell of Π_1 is a subset of some cell of Π_2, then we say that Π_1 is **finer** than Π_2, and that Π_2 is **coarser** than Π_1. Also, we write $\Pi_1 < \Pi_2$. A cell with only one element is called a **singleton**. A partition with only singleton cells is called a **discrete partition**, while a partition with only one cell is a **unit partition**. We write $|\Pi|$ for the number of cells in Π, and $|V_i|$ for number of vertices in cell V_i. For more readings about partitions see [52, 62, 63, 75, 76].

If Π is a discrete partition of S, then Π defines a permutation on the points in S. This permutation is denoted by $\sigma(\Pi)$.

Example 2.1.6. If $V = \{a, b, c, d\}$ is a finite set, then the following partitions of V

$$\begin{array}{ll} \{a, b, c, d\} & (coarsest) \\ \{a, b \mid c, d\} & \\ \{a, b \mid c \mid d\} & \\ \{a \mid b \mid c \mid d\} & (finest) \end{array}$$

are successive refinements from the coarsest to finest.

■

Definition 2.1.12. *A graph $BP = (V,\ E)$ is called* ***bipartite*** *if there exists a partition of the vertex set $V = \{V_1 \mid V_2\}$ so that both V_1 and V_2 are independent sets, i.e. there is no edge in the graph $\mathcal{G}$ that has both end-points in the same set. One often writes $BP = (\{V_1 \mid V_2\},\ E)$ to denote a bipartite graph whose partition has the parts V_1 and V_2.*

Bipartite graphs can be used to represent other incidence structures. For instance, if $\mathcal{G} = (V, E)$ is the cubic graph given in Figure 2.1.1, then its corresponding bipartite graph can be defined as follows: Let $BP(\mathcal{G}) = (\{V \mid E\}, \mathcal{E})$ with $|V| = 6$, $|E| = 9$, and $\{v, e\} \in \mathcal{E}$ if and only if the vertex v and edge e are incident in $\mathcal{G}$. Thus, $BP(\mathcal{G})$ can be drawn as in Figure 2.1.5. Note that, we denote $v_i \in V$ and $e'_j \in E$, by i and j' for all $1 \leq i \leq |V|$ and $1 \leq j \leq |E|$, respectively:

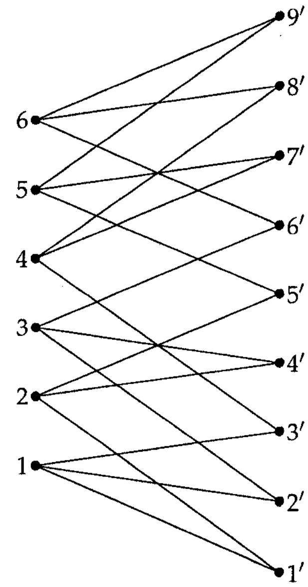


Figure 2.1.5: The corresponding bipartite graph of the graph $\mathcal{G}$ drawn in Figure 2.1.1.

2.2 Cages

In this section, we consider a proof of the theorem which says that corresponding to any two integers $k \geq 2$ and $g \geq 2$, there exists a k-regular graph of girth g. The proof is paraphrased from [73]. See [22, 74] for further details. In Section 4.7 on page 93, we consider construction procedures for cubic graphs with a given girth. More information about constructing graphs with a given girth can be found in [13, 60, 65, 66].

We allow, in this section, graphs to have double edges where this is not allowed in any other sections. A (k, g)-**cage** is a k-regular graph of girth g with the fewest possible number of vertices.

Definition 2.2.1. *A k-factor F of a graph G is a panning subgraph such that every vertex of F is of degree k.*

If a and b are integers with $b > 0$, then the remainder upon the division of

a by b is denoted by $(a \mod b)$.

Theorem 2.2.2. *For any two integers $k \geq 2$ and $g \geq 2$, there exists a k-regular graph of girth g that has a Hamiltonian cycle.*

Proof. The proof is by double induction on k and g. We first show the base step. Note that, if $k = 2$ then there exists a connected graph which is a cycle of length g.

Assume that $g = 2$. Then there exists a graph G consisting of 2 vertices joined by two edges, i.e. $g = k = 2$.

Let $k \geq 3$. Then a graph G satisfying the theorem can be constructed as follows. Let $H_{k-1} = (\{V, U\}, E)$ where $V = \{v_1, v_2, \ldots, v_{k-1}\}$ and $U = \{u_1, u_2, \ldots, u_{k-1}\}$ so that every vertex $v \in V$ is adjacent to every vertex $u \in U$ by exactly one edge. Thus, H_{k-1} is a $(k-1)$-regular graph of girth 4. Also, H_{k-1} contains a Hamiltonian cycle whose edges

$$\{v_1, u_1\}, \{u_1, v_2\}, \{v_2, u_2\}, \ldots, \{v_{k-1}, u_{k-1}\}, \{u_{k-1}, v_1\}.$$

We write vertices $v_1, u_1, v_2, u_2, \ldots, u_{k-1}$ in order, for such a cycle.

Let $S = V \cup U$. Then for each $i = 1, 2, \ldots, k-1$, $F_j = (S, E_j)$ is a 1-factor of H_{k-1} where

$$E_j = \big\{\{v_i, u_{(i+j-1 \mod k-1)}\} \,|\, v_i \in V, u_{(i+j-1 \mod k-1)} \in U, \text{ and } j = 1, \ldots, k-1\big\}.$$

Choose one j for $1 \leq j \leq k-1$, and replace each edge in F_j by a double edge. Then, the resulting graph is of girth $g = 2$ and degree $k \geq 2$.

We already have showed that there exists a k-regular graph of girth g for $g = 2$ and for all $k \geq 2$.

Induction on g:

Assume that the statement is true for all $g = 2, 3, \ldots, g_0 - 1$ (with $g_0 \geq 3$), and all $k \geq 2$, then we prove the statement for $g = g_0$ and for all $k \geq 2$. Again, if $k = 2$, then we are done (a cycle of length g_0 exists).

Induction on k:

Assuming that the statement has been proved for $g = g_0$ and $k = 2, 3, \ldots, k_0-1$ (with $k_0 \geq 3$), we prove that the statement is also true for $g = g_0$ and $k = k_0$.

By the induction on k hypothesis, there exists a $(k_0 - 1)$-regular graph X of girth g_0 with a Hamiltonian cycle $\mathcal{H}(X)$. Let $x_1, x_2, \ldots, x_m$ in order denote the vertices of $\mathcal{H}(X)$.

By the induction of g hypothesis, there exists a m-regular graph Y of girth $g_0 - 1$ with a Hamiltonian cycle $\mathcal{H}(Y)$. Let $y_1, y_2, \ldots, y_n$ in order denote the vertices of $\mathcal{H}(Y)$.

Let $X_1, X_2, \ldots, X_n$ be n disjoint graphs that are isomorphic to X. For $i = 1, 2, \ldots, n$, let $V(X_i) := \{x_{i,1}, x_{i,2}, \ldots, x_{i,m}\}$ such that $x_j \longleftrightarrow x_{i,j}$ for $j = 1, 2, \ldots, m$ is an isomorphism between X and X_i. Therefore, $x_{i,1}, x_{i,2}, \ldots, x_{i,m}$ in order are the vertices of a Hamiltonian cycle $\mathcal{H}(X_i)$ of X_i for $i = 1, 2, \ldots, n$.

We construct a graph G from Y and $X_1, X_2, \ldots, X_n$ satisfying the statement as follows. Replace each vertex y_i of Y by X_i for $i = 1, 2, \ldots, n$ such that each edge $\{y_i, y_j\}$ in Y is replaced by an edge connecting a vertex in X_i to a vertex X_j so that these edges are distributed as follows.

First, edges that are in $\mathcal{H}(Y)$. Replace edge $\{y_1, y_2\}$ of $\mathcal{H}(Y)$ by an edge connecting $x_{1,m}$ with $x_{2,1}$, replace edge $\{y_2, y_3\}$ by an edge connecting $x_{2,m}$ with $x_{3,1}$, and so on until we replace the edge $\{y_n, y_1\}$ by an edge connecting $x_{n,m}$ with $x_{1,1}$. Then, G contains a Hamiltonian cycle whose vertices are in order $x_{1,1}, x_{1,2} \ldots, x_{1,m}, x_{2,1}, \ldots, x_{n,m}$.

Second, edges that are not in $\mathcal{H}(Y)$. Let $X_i' = (V', E')$ such that $V(X_i') = V(X_i) \setminus \{\{x_{i,1}\}, \{x_{i,m}\}\}$ for $i = 1, 2, \ldots, n$. Each edge $\{y_i, y_j\}$ of Y that are not in $\mathcal{H}(Y)$ is replaced by an edge connecting a vertex X_i' with a vertex X_j'. These new edges are distributed so that each vertex of $X_1', X_2', \ldots, X_n'$ is incident with exactly one of the edges. This is can be done because of the fact that vertices in Y are incident with exactly $m - 2$ edges (that are not in $\mathcal{H}(Y)$), and each X_i' contains exactly $m - 2$ vertices for all $i = 1, 2, \ldots, n$.

Therefore, $X_1, X_2, \ldots, X_n$ are (k_0-1)-regular graphs and each vertex of X_i is joined to exactly one vertex not in X_i for all $i = 1, 2, \ldots, n$. Moreover, each X_i is isomorphic to X which is of girth g_0, and thus the constructed graph G is a k_0-regular graph of girth g_0 and has a Hamiltonian cycle.

□

Different examples of (k, g)-cages (no double edges allowed) will be considered throughout the thesis. The Petersen graph is one example that has been already presented in Figure 2.1.4 which is $(3, 5)$-cage. See [74] for some drawings of small (k, g)-cages.

The graph $A_{6,1}$ of Figure 3.3.6 on page 71 is a $(3, 3)$-cage, where the graph of Figure 4.7.1 on page 95 is a $(3, 4)$-cage.

2.3 Incidence Structures

Definition 2.3.1. *A finite* ***incidence structure*** $\mathcal{X}$ *is a triple* $(P, \mathcal{B}, I)$*, where* P *and* $\mathcal{B}$ *are finite sets and* $I \subseteq P \times \mathcal{B}$ *is a relation. In particular* $P = \{p_1, p_2, \ldots, p_v\}$ *is a set of* v ***points*** *and* $\mathcal{B} = \{B_1, B_2, \ldots, B_b\}$ *is a set of* b ***blocks****, sometimes called* ***lines****, such that* $B_i \subseteq P$ *for* $i = 1, 2, \ldots, b$ *and* $I \subseteq P \times \mathcal{B}$ *is the incidence relation. The elements of* I *are called flags.*

Given two incidence structures $\mathcal{X}_1 = (P_1, \mathcal{B}_1, I_1)$ *and* $X_2 = (P_2, \mathcal{B}_2, I2)$*, we say that* $\mathcal{X}_1$ *is isomorphic to* $\mathcal{X}_2$ *if there is a bijective map* $f : P_1 \longrightarrow P_2$ *which maps* $\mathcal{B}_1$ *onto* $\mathcal{B}_2$*. Here, a block* $B \in \mathcal{B}_1$ *with* $B = \{p_1, p_2, \ldots, p_k\}$ *is mapped onto* $B^f = \{{p_1}^f, {p_2}^f, \ldots, {p_k}^f\}$*. Thus isomorphisms are incidence preserving maps, so that* $p^f \in B^f$ *if and only if* $p \in B$*. Moreover, if* f *maps* $\mathcal{X}_1$ *to itself then* f *is called an automorphism, and the group formed by all such automorphism is called the* ***automorphism group****, denoted by* $Aut(\mathcal{X}_1)$*.*

It is possible that in an incidence structure, two blocks are incidence with the same set of points. In that case, we speak of **repeated blocks**. If this is not the case, we can identify each block with the set of points it is incident with, and replace I by the relation "$\in$". In this case, we usually omit the incidence relation altogether.

The number of blocks containing a point $p \in P$ is called the **degree**, denoted by $[p]$. Similarly, the number of points that are contained in a block B is called the length of B denoted by $|B|$. A pair (p, B) with $p \in B \in \mathcal{B}$ is called a **flag**. In this case, we say that p lies on B, B passes through p, or that p and B are incident.

For instance, one can define a graph $\mathcal{G}$ as an incidence structure (V, E) where V is a finite set of points, and E is a finite set of blocks (edges) with E a set of subsets of V of size 2. In this case we have $|B| = 2$ for any block in E. An example of an incidence structure (which is a graph) is given below.

Example 2.3.1. Consider the incidence structure $\mathcal{G} = (P, \mathcal{B})$ where

$$P = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$$

$$\begin{aligned} \mathcal{B} = \Big\{ &\{1,2\}, \{1,5\}, \{1,6\}, \{2,3\}, \{2,7\}, \{3,4\}, \{3,8\}, \\ &\{4,5\}, \{4,9\}, \{5,10\}, \{6,8\}, \{6,9\}, \{7,9\}, \{7,10\}, \{8,10\} \Big\} \end{aligned}$$

Then the corresponding incidence structure can be seen as the Petersen graph of Figure 2.1.4 on page 13.

Such an incidence structure $\mathcal{G}$ (Petersen graph) can be represented in different ways. One particular way is an incidence matrix. The following incidence matrix A is a representation for the Petersen graph where each row and column in A correspond to a point and a block in $\mathcal{G}$, respectively.

$A =$

	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	B_{10}	B_{11}	B_{12}	B_{13}	B_{14}	B_{15}
p_1	x	x	x												
p_2	x			x	x										
p_3				x		x	x								
p_4						x		x	x						
p_5		x						x		x					
p_6			x								x	x			
p_7					x								x	x	
p_8							x				x				x
p_9									x			x	x		
p_{10}										x				x	x

■

If $\mathcal{X} = (P, \mathcal{B})$ is an incidence structure with $|P| = v$ and $|\mathcal{B}| = b$ associated with an incidence matrix A of dimension $v \times b$ with $A = (a_{i,j})$ for $1 \leq i \leq v$ and $1 \leq j \leq b$, then we define

$$\text{row-sum}_i(A) = \sum_{j=1}^{b} a_{ij},$$

and

$$\text{row}_i(A) = \{ j \mid a_{ij} = 1 \ \text{ for } 1 \leq j \leq b \ \}.$$

And for $1 \leq j \leq b$, we define

$$\text{col-sum}_j(A) = \sum_{i=1}^{v} a_{ij}.$$

and

$$\text{col}_j(A) = \{ i \mid a_{ij} = 1 \ \text{ for } 1 \leq i \leq v \ \}.$$

and

The incidence matrix A of Example 2.3.1 has $\text{row-sum}_i(A) = 3$, $\text{col-sum}_j(A) = 2$, and $|\text{row}_i(A) \ \cap \ \text{row}_h(A) \ | \leq 1$, for $1 \leq i, h \leq 10$ with $i \neq h$, and for $1 \leq j \leq 15$.

Definition 2.3.2. *Let t, v, k, and λ be positive integers with $v \geq k \geq t$. Then, a t-(v, k, λ) design is an incidence structure over v points such that the following holds:*

- *each block contains exactly k points;*
- *each t-subset of points is contained in exactly λ block.*

A t-(v, k, λ) with $t = 2$ is called a (**balanced incomplete**) **block design**. A double counting argument shows that every point of a block design is incident to r blocks and that the number of blocks is b, where

$$\lambda(v-1) = r(k-1), \qquad vr = bk. \tag{2.3}$$

Definition 2.3.3. *Let $(P, \mathcal{B})$ be an incidence structure without repeated blocks, such that*

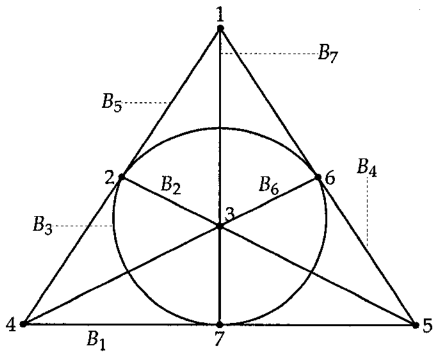
- *each block contains exactly k points;*
- *each set of t points is contained in a unique block.*

Then $(P, \mathcal{B})$ is called a ***Steiner system*** *$S(t, k, v)$, where $v = |P|$.*

If $t = 2$, $k = 3$, we speak of a **Steiner triple system** denoted by $STS(v)$. A class of Steiner systems are the finite **projective planes**. A finite **projective plane** is a $S(2, n+1, n^2+n+1)$ for an integer $n \geq 2$, called **order**.

Example 2.3.2. The smallest example is the projective plane of order 2 which is shown in Figure 2.3.1, also known as the **Fano plane**, denoted by $F = (P, \mathcal{B})$ where $P = \{p_1, p_2, \ldots, p_7\}$ and $\mathcal{B} = \{B_1, B_2, \ldots, B_7\}$.

Moreover, we construct a bipartite graph $BP(F) = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{P \mid \mathcal{B}\}$ and for $v, u \in \mathcal{V}$ we have $\{u, v\} \in \mathcal{E}$ only if $u \in P$ and $v \in \mathcal{B}$ with $u \in v$ in F. Figure 2.3.2 shows an incidence matrix of $BP(F)$.



$A =$

	B_1	B_2	B_3	B_4	B_5	B_6	B_7
p_1				x	x		x
p_2		x	x		x		
p_3		x				x	x
p_4	x				x	x	
p_5	x	x		x			
p_6			x	x		x	
p_7	x		x				x

Figure 2.3.1: The Fano plane F with its associated incidence matrix A.

	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9	e_{10}	e_{11}	e_{12}	e_{13}	e_{14}	e_{15}	e_{16}	e_{17}	e_{18}	e_{19}	e_{20}	e_{21}
p_1	x	x	x																		
p_2				x	x	x															
p_3							x	x	x												
p_4										x	x	x									
p_5													x	x	x						
p_6																x	x	x			
p_7																			x	x	x
B_1										x			x						x		
B_2				x			x							x							
B_3					x											x				x	
B_4	x														x		x				
B_5		x				x					x										
B_6								x				x						x			
B_7			x						x												x

Figure 2.3.2: The incidence matrix of the bipartite graph $BP(F)$.

Note that the automorphism group of both F and $BP(F)$ is of order 168. If we relabel the rows and columns of the incidence matrix of Figure 2.3.2 to be $\{0, 1, \ldots, 6 \mid 7, \ldots, 13\}$ and $\{14, 15, \ldots, 34\}$, respectively, then the $Aut(BP(F))$ is generated by a set S, where S is given by

$$\Big\{(2\ 5)(4\ 6)(8\ 9)(10\ 13)(14\ 16)(17\ 18)(20\ 29)(21\ 31)(22\ 30)(26\ 32)(27\ 33)(28\ 34),$$
$$(2\ 6)(4\ 5)(7\ 12)(8\ 9)(17\ 18)(20\ 33)(21\ 32)(22\ 34)(23\ 25)(26\ 31)(27\ 29)(28\ 30),$$
$$(1\ 2)(3\ 6)(9\ 12)(11\ 13)(15\ 16)(17\ 20)(18\ 21)(19\ 22)(23\ 32)(24\ 34)(25\ 33)(29\ 31),$$
$$(0\ 1)(2\ 5)(8\ 10)(9\ 13)(14\ 17)(15\ 19)(16\ 18)(20\ 30)(21\ 31)(22\ 29)(27\ 28)(33\ 34)\Big\}.$$

Moreover, the automorphism group orbits are partition as follows

$$\Big\{\{0, 1, \ldots, 6\}, \{7, 8, \ldots, 13\}, \{14, 15, \ldots, 34\}\Big\}.$$

Note that, if the partition of points and blocks of the Fano plane is left out, then the resulting graph has the points and the blocks of F as vertices where two vertices p and B are adjacent only if $p \in B$ in F. In this case one more automorphism is added to the automorphism group given above which is the mapping $p_i \longleftrightarrow B_i$ for all $i = 1, 2, \ldots, 7$. This resulting graph is in fact called the **Heawood graph** which is a $(3,6)$-cage, see Example 6.1.2 on page 125.

■

Definition 2.3.4. *An incidence geometry is an incidence structure* $(P, \mathcal{B}, I)$ *with*

1. $|B| \geq 2$ *for each* $B \in \mathcal{B}$, *and*

2. *each pair of points is on at most one block.*

If $|P| \geq 2$, *then Condition 2 excludes repeated blocks. By the above, an incidence geometry does not have repeated blocks, and we can identify each block* $B \in \mathcal{B}$ *with a distinct subset of* P. *That is, we may write* $(P, \mathcal{B})$ *instead of* $(P, \mathcal{B}, I)$.

Definition 2.3.5. *An incidence geometry* $(P, \mathcal{B})$ *is called a configuration of type* (v_r, b_k) *if:*

1. *Every line is incident with* k *points, i.e.* $|B_j| = k$ *for* $j = 1, 2, \ldots, b$,

2. *Every point is incident with* r *lines, i.e.* $[p_i] = r$ *for* $i = 1, 2, \ldots, v$.

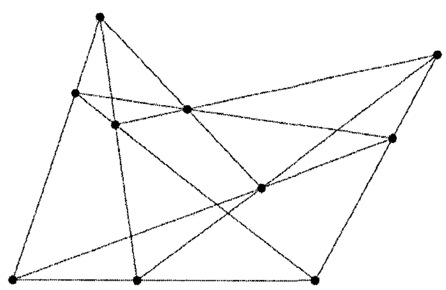
3. *Any two distinct points are incident with at most one line.*

A configuration $(P, \mathcal{B})$ *with* $v = b$ *is called* ***symmetric*** *(see, for instance [5]) and thus is denoted by* v_r.

In a projective plane of order n, we have

$$b = \lambda \frac{v(v-1)}{k(k-1)} = n^2 + n + 1 = v,$$
$$r = \lambda \frac{v-1}{k-1} = n + 1 = k,$$

by (2.3). This shows that every finite projective plane of order n is a symmetric configuration $(n^2 + n + 1)_{(n+1)}$ and vice versa. An example of a symmetric configuration that is not a projective plane is the following 10_3 configuration, which is called the **Desargues configuration** and shown in Figure 2.3.3.



$$A = \begin{array}{|c|c|c|c|c|c|c|c|c|c|}
\hline
x & x & x & & & & & & & \\ \hline
 & & & x & x & x & & & & \\ \hline
x & & & & & & x & x & & \\ \hline
 & x & & x & & & & & x & \\ \hline
 & & x & & x & & x & & & \\ \hline
 & & & & & x & & x & x & \\ \hline
x & & & & & & & & x & x \\ \hline
 & & x & x & & & & & & x \\ \hline
 & x & & & x & & & x & & \\ \hline
 & & & & & x & x & & & x \\ \hline
\end{array}$$

Figure 2.3.3: The Desargues' Configuration, a 10_3 configuration, with its associated incidence matrix A.

Definition 2.3.6. *A* ***linear space on*** $|P|$ ***points*** *is an incidence geometry* $(P, \mathcal{B})$ *in which each pair of points is on at least one block, and hence on exactly one block.*

An incidence geometry is a configuration if every point is on r lines and every line has k points for some numbers r and k.

A **projective plane** of order n is an $(n^2 + n + 1)_{n+1}$ configuration and at the same time it is a linear space on $(n^2 + n + 1)$ points.

Definition 2.3.7. *Let* $(P, \mathcal{B})$ *be a linear space on* v *points. Define*

$$a_i := \#\textit{lines of length } i \textit{ in } (P, \mathcal{B}),$$

The vector $a := (a_2, a_3, \ldots, a_v)$ *is called the* ***line type*** *of the space* $(P, \mathcal{B})$*. Line types are also called* ***parameters of the first type****. Often, it is convenient to denote line types in exponential notation, that is* $(2^{a_2}, 3^{a_3}, \ldots, v^{a_v})$*. Exponents* 1 *may not be explicitly written.*

Then, by the previous definition, a projective plane is also an $(n+1)^{(n^2+n+1)}$ linear space. For extensive details about linear spaces, the reader is referred to [4, 5, 6].

Another example where linear spaces coincide with configurations is a **Steiner triple system**. A Steiner triple system of order v is a 2-$(v, 3, 1)$ design and also is a (v_r, b_3) configuration, where one can conclude from 2.3 the following with $k = 3$ and $\lambda = 1$.

$$r = \frac{v-1}{2},$$

and thus

$$b = \frac{vr}{3}$$

Clearly, $STS(v)$ is a (3^b) linear space on v points. For instance, an $STS(13)$ is a linear space 3^{26} as there are 26 blocks.

For more readings about linear spaces, see [4, 5, 6]. For more about configurations with existence and non-existence results and classification of configuration with at most 21 points, see [6, 8, 29, 30, 31, 32].

In order to treat the isomorphism problem for graphs and incidence structures more thoroughly, we need to discuss the algebraic concept of a group action. The theory of group actions is the theoretical framework for isomorphism of discrete structures. A very broad account for finite group actions and applications is the book by Kerber [53]. Here, we summarize the most fundamental concepts. For further facts on permutation groups, see [1, 7, 16, 17, 20, 58, 79].

2.4 Finite Group Actions

In general, we say that G is a group instead of writing $(G, \cdot)$. A subset H of G is said to be a **subgroup** if it is closed under the operation "$\cdot$" associated with

G, and it is a group itself. We write $H \leq G$ to indicate that H is a subgroup of G.

Let $X = \{x_1, x_2, \ldots, x_t\}$ be a finite non-empty set. A **permutation** of X is a bijection of X to itself. For a permutation g of X, we write x^g for the image of $x \in X$ under g. The set of all permutations of X forms a group with respect to functional composition. It is called the **symmetric group** on X and is denoted by $\mathrm{Sym}_{(X)}$. A **permutation group** on X is a subgroup of $\mathrm{Sym}_{(X)}$.

Definition 2.4.1. *Let G be a group and let X be a finite non-empty set. An **action** of G on X is a mapping*

$$X \times G \rightarrow X, \quad (x, g) \mapsto x^g$$

such that

- $x^{1_G} = x$, *for all* $x \in X$, *and*
- $x^{gh} = (x^g)^h$, *for all* $x \in X$ *and for all* $g, h \in G$.

Let G be a group that acts on the finite set X. Then, for $x \in X$, the **orbit of x in X under the action of G** is

$$G(x) = x^G = \{x^g \in X \mid g \in G\}.$$

The **stabilizer of x in G** is the subgroup

$$G_x = \mathrm{Stab}_G(x) = \{g \in G \mid x^g = x\},$$

which is the automorphism group $Aut(x)$, in the sense of Definition 2.1.2 for graphs (or Definition 2.3.1 for incidence structures).

Moreover, if $Y = \{y_1, y_2, \ldots, y_s\} \subseteq X$ then we define the **setwise stabilizer** as the subgroup

$$G_Y = \{g \in G \mid Y^g = Y\} = \{g \in G \mid \forall i,\ y_i^g \in Y\}.$$

The **pointwise stabilizer** is the subgroup

$$G_{y_1,y_2,\ldots,y_s} = \{g \in G \mid y_i^g = y_i \quad \text{for} \quad 1 \le i \le s\}.$$

If $X = \{x_1, x_2, \ldots, x_t\}$ and $G_{x_1,x_2,\ldots,x_t} = 1_G$, i.e. only the identity element 1_G stabilizes every element in X, then G is said to act **faithfully** on X or it is said that G is a **group of permutations** of X.

Given two elements x and y in X, we say that x is **isomorphic** to y under the action of a group G if there is a permutation $g \in G$ such that $x^g = y$, denoted by $x \cong_G y$ to emphasize the group action of G on X. Simply we write $x \cong y$ if the group action is clear from the context. In this case, $g \in G$ is called an **isomorphism** from x to y. If $x^g = x$, then g is called an **automorphism** of x. The **automorphism group of** x, denoted by $Aut(x)$, consists of all automorphism of x. In particular,

$$Aut(x) = G_x = \{g \in G \mid x^g = x\}.$$

Clearly then x and y are contained in the same G-orbit on X if and only if they are isomorphic by an element $g \in G$.

If G is a group and $\gamma \in G$, then the **conjugate** of γ by α is $\alpha^{-1}\gamma\alpha$, which is denoted by γ^α.

Lemma 2.4.2. *For any $x, y \in X$ and $g \in G$ with $x^g = y$, we have*

$$G_y = g^{-1} G_x g.$$

Lemma 2.4.3. *Let a group G act on the finite set X. Then the mapping*

$$\delta : G \to Sym_{(X)}, \quad g \mapsto \overline{g} \quad \textit{where} \quad \overline{g} : x \mapsto x^g \quad \textit{for } x \in X$$

is a homomorphism. G is faithful on X precisely if δ is a one-to-one map.

Definition 2.4.4. *A subset Y of X is said to be G-invariant if $G_Y = G$, i.e. all elements in G stabilize Δ setwise.*

Lemma 2.4.5. *Let G be a group acting on a finite set X. Let Y be a G-invariant subset of X. Then, Y can be partitioned into orbits of G on X.*

If G has only one orbit on X, namely X, then G is said to be **transitive** (or act **transitively** on X).

Definition 2.4.6. *If G is a permutation group acting on a finite set X. Then, we call a non-empty subset Y of X a **block** of G if for each $g \in G$, the image set Y^g either coincides with Y or has no points in common with Y. Obviously, the whole set X, and the set of one element $\{y\} \subseteq X$ are blocks of every G on X. Such blocks are called **trivial blocks**.*

Definition 2.4.7. *A transitive group G acting on a finite set X is called **imprimitive** if there is at least one non-trivial block Y, i.e. $|X| \neq |Y| \neq 1$. Such a block is also called **set of imprimitivity**, see [58].*

Definition 2.4.8. *Let G be a group, and $H \leq G$ is a subgroup. For $g \in G$, the set $Hg = \{hg \,:\, h \in H\}$ is called the **right coset of** H **in** G which contains g. Similarly, $gH = \{gh \,:\, h \in H\}$ is the **left coset of** H **in** G.*

Lemma 2.4.9 (Orbit-Stabilizer Lemma). *Let G be a group acting on a finite set X. Then for any $x, y \in X$, if $y \in G(x)$, then $\{g \in G \,|\, x^g = y\}$ is a coset of G_x in G. In particular, $|G(x)| = |G|/|G_x|$, where $|\cdot|$ denotes the group order.*

Definition 2.4.10. *If G is a finite group and $S = \{s_1, s_2, \ldots, s_r\}$ is a set of elements of G which together generate G, i.e.*

$$\langle S \rangle = G,$$

*then any element $g \in G$ can be written as a word of finite length over the alphabet S. In this case, S is said to be a **set of generators** for G.*

A set $B = \{b_0, b_1, \ldots, b_{k-1}\} \subseteq X$ is called a **base** for G on X if the **pointwise** stabilizer $G_{b_0, b_1, \ldots, b_{k-1}} = 1$, i.e. if only the identity of G fixes all the points of B. An **ordered base** B for G on X is a sequence $(b_0, b_1, \ldots, b_{k-1})$ such that the corresponding set $\{b_0, b_1, \ldots, b_{k-1}\}$ is a base for G on X. An ordered base B gives rise to a chain of subgroups. Write $G^{(0)} = Aut(X)$. Let $G^{(1)}$ fix b_0 in $G^{(0)}$ and so on. In general, let $G^{(i+1)}$ denote the subgroup of $G^{(i)}$ obtained by stabilizing b_i in $G^{(i)}$, for $i = 0, 1, \ldots, k-1$. Thus

$$G = G^{(0)} \geq G^{(1)} \geq G^{(2)} \geq \cdots \geq G^{(k)} = \langle 1 \rangle, \tag{2.4}$$

where $G^{(i+1)} = {G^{(i)}}_{b_i}$.

Definition 2.4.11. *A* ***strong generating set*** *for a finite group G relative to the associated base B with $|B| = k$ is a set S of elements of G with the property that*

$$\langle S \cap G^{(i)} \rangle = G^{(i)}, \qquad \textit{for all } i = 0, 1, \ldots, k.$$

Lemma 2.4.12 (Order-Lemma). *Suppose that G is a permutation group and $B = (b_1, b_2, \ldots, b_k)$ is a base for G on $X = \{x_1, \ldots, x_n\}$ for $n \geq k$. Then,*

$$|G| = \prod_{i=1}^{k} |\, b_i^{G^{(i-1)}} \,|$$

If a group G acts on a finite set X, then for $x, y \in X$, two orbits $G(x)$ and $G(y)$ are either the same or disjoint, i.e.

$$G(x) \;=\; G(y) \;\text{ or }\; G(x) \cap G(y) \;=\; \emptyset.$$

Thus, if X/G denotes the **set of all orbits of the action of** G **on** X, then X/G is a partition of X.

Definition 2.4.13. *An* ***orbit transversal*** *for G on X is a set of elements in X denoted by $\mathcal{T}(G, X)$ such that:*

- *for any $x \in X$, there exists $g \in G$ with $x^g \in \mathcal{T}(G, X)$, and*
- *for any $x, y \in \mathcal{T}(G, X)$, we have $G(x) \cap G(y) = \emptyset$.*

Definition 2.4.14. *Let G act on a finite set X. If a map $\varphi : X \to G$ such that for any $x, y \in X$, φ satisfies the following two conditions:*

1. $x^{\varphi(x)} \cong x$, *and*

2. $x \cong y$ *implies* $x^{\varphi(x)} = y^{\varphi(y)}$,

then φ is called the ***canonical labeling map***. *Moreover, $x^{\varphi(x)}$ is called the* ***canonical form of*** *x, and denoted by $\rho(x)$.*

In particular, if $G(x) = \{x, x_1, x_2, \ldots, x_r\}$ is the orbit of x under the action of G on X, then $\rho(y) = \rho(x)$ for all $y \in G(x)$. Thus, such an element is also called a ***canonical orbit representative***.

Lemma 2.4.15. *If G is a group acting on a finite set X, then two objects x and y are isomorphic in X under the action of G if and only if $\rho(x) = \rho(y)$.*

Now we turn our attention to the problem of computing the orbit of an element x in a finite set X under the action of the group G given by a set of generators $S = \{s_1, \ldots, s_r\}$. See [1, 7, 15, 17, 18] for more reading.

We define the **action graph** of G on X with respect to the set $S \subseteq G$ to be the directed graph $\mathcal{G} = (X, \mathcal{E})$ whose vertices are the elements of X with edges

$$(x, y) \in X \times X : x^{s_j} = y \text{ for some } s_j \in S.$$

The orbit of x under G is then the connected component of x in the action-graph $\mathcal{G}$. We remark that the connected component does not depend on the choice of the generating set S (when thought of as a subset of vertices).

Definition 2.4.16. *Let G be a group acting on a finite set X. Let G be given by generators S. Let x be an element of X. A* ***Schreier tree*** *for the orbit $G(x)$ is any directed spanning tree $T_x = (G(x), \mathcal{E})$ for the connected component containing x in the action graph for G on X with respect to S.*

Examples of Schreier trees will be presented in what follows and in Chapter 3.

Example 2.4.1. Let G be the automorphism group of the cubic graph $\mathcal{C}$ of Figure 2.4.1.

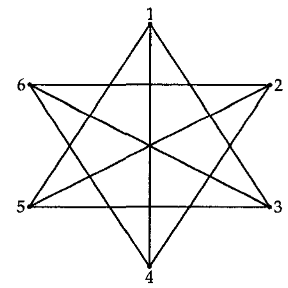


Figure 2.4.1: A cubic graph $\mathcal{C}$ of order 6.

Then, using two generating sets $S = \{s_1, s_2\}$ and $R = \{r_1, r_2\}$ where

$$s_1 = (\ 2\ \ 6\ )\ (\ 3\ \ 5\ ), \qquad r_1 = (\ 2\ \ 6\ )\ (\ 3\ \ 5\ ),$$

$$s_2 = (\ 1\ \ 2\ )\ (\ 3\ \ 6\ )\ (\ 4\ \ 5\ ), \qquad r_2 = (\ 1\ \ 2\ \ 3\ \ 4\ \ 5\ \ 6\ ),$$

results in two action-graphs by using S (left), and R (right) as in Figure 2.4.2.

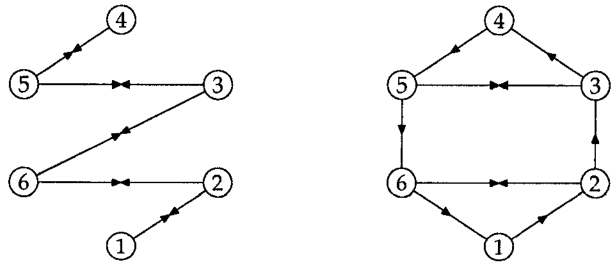


Figure 2.4.2: Two action-graphs of G on $\mathcal{C}$.

■

Assume that for every element $y \in G(x)$, we choose $f(y) \in G$ such that $x^{f(y)} = y$. By the Orbit-Lemma, all such $f(y)$ form a set of coset representatives for G_x in G. However, it is not efficient to store all such elements, so instead we construct a Schreier tree rooted at x, denoted by T_x.

Algorithm 2.4.1, which was taken from [7], is a description of an orbit algorithm which computes the orbit of a given point $x \in X$ under the action of a permutation group G generated by $S = \{s_1, s_2, \dots, s_r\}$. Moreover, a Schreier tree rooted at x, denoted by $T_x = (\mathcal{O}, \mathcal{E})$, is constructed.

Algorithm 2.4.1 orbit computation

Input: A permutation group G acting on a finite set $X = \{x_1, \dots, x_n\}$, a generating set $S = \{s_1, \dots, s_r\}$ of G, a point $x \in X$.

Output: Schreier tree $T_x = (\mathcal{O}, \mathcal{E})$ for the orbit $\mathcal{O} = G(x)$

let $\mathcal{Q}$ be a queue holding the element x
let $T_x = (\{x\}, \emptyset)$ be the tree with only one node x
while $\mathcal{Q} \neq \emptyset$ do
    let y be the first element of $\mathcal{Q}$ (remove y from $\mathcal{Q}$)
    for $i = 1, \dots, r$ do
        $z := y^{s_i}$
        if $z \notin \mathcal{O}$ then
            append z to $\mathcal{Q}$, add z to $\mathcal{O}$
            add the edge (y, z) labeled by s_i to $\mathcal{E}$
        end if
    end for
end while

Example 2.4.2. Consider again Example 2.4.1. Algorithm 2.4.1 with a generating set R and the point 1 computes a Schreier tree drawn in Figure 2.4.3.

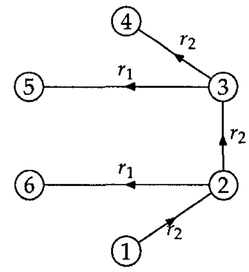


Figure 2.4.3: A Schreier tree corresponding to the (right) action-graph of Figure 2.4.2.

■

Once the orbit of an element x in X is computed, then we say that y is isomorphic to x if and only if $y \in G(x)$. However, it is clear that orbit computations require difficult work. In some cases we might have some problem involving large orbits and thus this method will not be our choice.

2.5 Some Specific Group Actions

In the following, we will discuss some examples of group actions. Let $M_{v,b}$ denote the class of all $v \times b$ $\{0,1\}$-matrices. For $A, B \in M_{v,b}$, we say that A is **equivalent** to B, denoted by $A \sim B$, if one matrix can be obtained from the other by row and column permutations. If $G = Sym_{(v)} \times Sym_{(b)}$, then G acts on $M_{v,b}$ by row and column permutations defined by $\big(A, (\alpha, \beta)\big) \mapsto A^{(\alpha,\beta)}$, where $(\alpha, \beta) \in G$, and if $A = (a_{i,j})$, then

$$A^{(\alpha,\beta)} = \left(a_{i^{\alpha^{-1}}, j^{\beta^{-1}}}\right)$$

for all $1 \leq i \leq v$ and $1 \leq j \leq b$. That is, the entry at row i, and column j of $A^{(\alpha,\beta)}$ is the entry at row $i^{\alpha^{-1}}$, and column $j^{\beta^{-1}}$ of A. Then, A is equivalent to B in $M_{v,b}$ if there exists $(\alpha, \beta) \in G$ such that

$$A^{(\alpha,\beta)} = B$$

where α and β are row and column permutations.

Let $\mathcal{S}_v$ denote the class of all incidence structures with v points, and let $Sym_{(v)}$ act on $\mathcal{S}_v$.

Example 2.5.1. Let $S_1 = (P, \mathcal{B}_1), S_2 = (P, \mathcal{B}_2) \in \mathcal{S}_5$ given in Figure 2.5.1 where

$$
\begin{aligned}
P &= \{1,2,3,4,5\},\\
\mathcal{B}_1 &= \{\{1,2\},\{1,5\},\{2,3\},\{3,4\},\{4,5\}\}, \quad \text{and}\\
\mathcal{B}_2 &= \{\{1,3\},\{1,4\},\{2,4\},\{2,5\},\{3,5\}\}.
\end{aligned}
$$

Then, S_1 and S_2 can be drawn as in Figure 2.5.1.

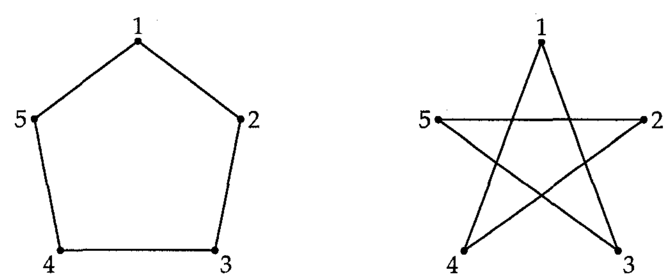


Figure 2.5.1: Incidence structures S_1 (left) and S_2 (right).

Moreover, if $\alpha = (\ 2\ \ 3\ \ 5\ \ 4\ ) \in Sym_{(5)}$, then α induces an isomorphism from S_1 to S_2 in the sense of Definition 2.3.1.

■

Let $\mathcal{S}_v^b$ denote the class of all $v \times b$ $\{0,1\}$-incidence matrices correspond to incidence structures in $\mathcal{S}_v$. Then, the **isomorphism equivalence classes** of incidence structures in $\mathcal{S}_v$ are the orbits of the directed product group $Sym_{(v)} \times Sym_{(b)}$ on $\mathcal{S}_v^b$.

First, we observe that permuting the rows of $A \in \mathcal{S}_v^b$ is equivalent to permuting the points in the blocks of the corresponding incidence structure $S \in \mathcal{S}_v$. Second, permuting the columns of A corresponds to relabeling the blocks in S. Because of that, the following theorem holds.

Theorem 2.5.1. *Any two incidence structures $S, S' \in \mathcal{S}_v$ are contained in the same $Sym_{(v)}$-orbit on $\mathcal{S}_v$ if and only if any two incidence matrices $A, A' \in \mathcal{S}_v^b$ corresponding to S and S', respectively, are contained in the same $Sym_{(v)} \times Sym_{(b)}$-orbit on $\mathcal{S}_v^b$.*

Clearly, Theorem 2.5.1 holds for graphs as well.

Example 2.5.2. Let A_1 and A_2 given below correspond to the two incidence structures of Example 2.5.1 S_1 and S_2, respectively.

$A_1 =$

	B_1	B_2	B_3	B_4	B_5
p_1	x	x			
p_2	x		x		
p_3			x	x	
p_4				x	x
p_5		x			x

$A_2 =$

	B_1	B_2	B_3	B_4	B_5
p_1	x	x			
p_2			x	x	
p_3	x				x
p_4		x	x		
p_5				x	x

If $\alpha = (\ 2\ \ 3\ \ 5\ \ 4\ ) \in Sym_{(5)}$ and $\beta = (\ 3\ \ 5\ ) \in Sym_{(5)}$, then

$$A_1^{(\alpha,\beta)} = A_2.$$

This corresponds to

$$\Big\{\{1,2\},\{1,5\},\{2,3\},\{3,4\},\{4,5\}\Big\}^{\alpha} = \Big\{\{1,3\},\{1,4\},\{3,5\},\{2,5\},\{2,4\}\Big\}.$$

■

Definition 2.5.2. *Let $D = (V, E)$ be a directed graph of order v with $V = \{p_1, \ldots, p_v\}$. Assume that D is of size b with $E = \{e_1, \ldots, e_b\}$. Then an* ***incidence matrix*** *for the digraph D is an $m \times n$ $\{0,1\}$-matrix A where $m = 2v$ and $n = v + b$. This matrix is partitioned in four smaller blocks which are denoted by $I_v, I_v, B,$ and B' as follows:*

$$A = \left[\begin{array}{c|c} I_v & B \\ \hline I_v & B' \end{array}\right],$$

where I_v is the $v \times v$ identity matrix. Block B is an $v \times b$ $\{0,1\}$-matrix with $B = (b_{ij})$ defined for all $1 \leq i \leq v$ and $1 \leq j \leq b$ by

$$b_{ij} = \begin{cases} 1, & \textit{if } e_j = (p_i, p_k) \in E \textit{ for some } p_k \in V; \\ 0, & \textit{otherwise} \end{cases}$$

while block B' is an $v \times b$ $\{0,1\}$-matrix with $B' = (b'_{ij})$ defined for all $1 \leq i \leq v$ and $1 \leq j \leq b$ by

$$b'_{ij} = \begin{cases} 1, & \textit{if } e_j = (p_k, p_i) \in E \textit{ for some } p_k \in V; \\ 0, & \textit{otherwise} \end{cases}$$

In particular, blocks B and B' indicate the out- and in-neighbors for vertices in V, respectively. Moreover, the two identity blocks indicate which vertex dominates which.

Example 2.5.3. The incidence matrix A for the directed graph of Example 2.1.5 on page 12 is is given in Figure 2.5.2.

	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9	e_{10}	e_{11}
1	X					X					
2		X					X	X			
3			X						X		
4				X						X	
5					X						X
6	X						X				
7		X							X		
8			X							X	
9				X							X
10					X	X		X			

Figure 2.5.2: Incidence matrix corresponding to the digraph of Figure 2.1.3.

Two rows in A (one in B and one in B') represent the out- and in-neighbors for a vertex in V. For instance, one can get the out neighbors for vertex 2 as follows: Start from row 2 in the matrix and look at incidences in the block

B then look at incidences in B' that are in the same column as those in B contained in row 2 to see that row 6 and row 10 are. By the identity matrix I_2 we conclude that vertex 2 dominates vertices 1 and 5.

■

In what follows we prove an analogous version of Theorem 2.5.1 for digraphs. First, we introduce some notations.

Let $D_1 = (V, E_1)$ and $D_2 = (V, E_2)$ be two digraphs of order v and size b with $V = \{P_1, P_2, \ldots, P_v\}$ and

$$E_i = \{e_1^{(i)}, e_2^{(i)}, \ldots, e_b^{(i)}\},$$

for $i = 1, 2$ where

$$e_j^{(1)} = (P_{x_j}, P_{y_j}), \text{ and } e_j^{(2)} = (P_{s_j}, P_{t_j}),$$

where P_{x_j}, P_{y_j}, P_{s_j}, and P_{t_j} are in V, $1 \le x_j, y_j \le v$ and $x_j \ne y_j$, $1 \le s_j, t_j \le v$ and $s_j \ne t_j$, and $j = 1, 2, \ldots, b$. Let

$$N_k = \left[\begin{array}{c|c} I_v & A_k \\ \hline I_v & B_k \end{array}\right],$$

for $k = 1, 2$, corresponding to D_1 and D_2, respectively, where A_1 and B_1 are $(v \times b)$ $\{0,1\}$-matrices defined for $A_1 = (a_{i,j}^{(1)})$ and $B_1 = (b_{i,j}^{(1)})$ for $1 \le i \le v$ and $1 \le j \le b$ by

$$a_{i,j}^{(1)} = \begin{cases} 1, & \text{if } P_{x_j} = P_i; \\ 0, & \text{otherwise}. \end{cases} \text{, and} \qquad b_{i,j}^{(1)} = \begin{cases} 1, & \text{if } P_{y_j} = P_i; \\ 0, & \text{otherwise}. \end{cases}$$

Similarly, A_2 and B_2 are $(v \times b)$ $\{0,1\}$-matrices defined for $A_2 = (a_{i,j}^{(2)})$ and $B_2 = (b_{i,j}^{(2)})$ for $1 \le i \le v$ and $1 \le j \le b$ by

$$a_{i,j}^{(2)} = \begin{cases} 1, & \text{if } P_{s_j} = P_i; \\ 0, & \text{otherwise}. \end{cases} \text{, and} \qquad b_{i,j}^{(2)} = \begin{cases} 1, & \text{if } P_{t_j} = P_i; \\ 0, & \text{otherwise}. \end{cases}$$

Then, consider the projection maps π_1 and π_2 defined by

$$\pi_1 : E_l \to V \quad , \ (P_{x_j}, P_{y_j}) \mapsto P_{x_j}, \quad \text{and}$$
$$\pi_2 : E_l \to V \quad , \ (P_{x_j}, P_{y_j}) \mapsto P_{y_j},$$

for $l = 1, 2$. If Δ_1 and Δ_2 are two finite disjoint sets of cardinalities v and b, respectively, then we write $G_{v;b}$ for the group acting on Δ_1 and Δ_2 with respecting the partitions, i.e. there is no element g in $G_{v;b}$ maps $\delta_1 \in \Delta_1$ to $\delta_2 \in \Delta_2$.

If A is a $(v \times b)$ matrix with $A = (a_{i,j})$ for $1 \le i \le v$ and $1 \le j \le b$, then for $(\alpha, \beta) \in Sym_v \times Sym_b$

$$A^{(\alpha,\beta)} = (\ a_{i^{\alpha^{-1}}, j^{\beta^{-1}}}\ ).$$

More general, for a block matrix $N = \left[\ N_{i,j}\ \right]$ with $1 \le i \le l$ and $1 \le j \le k$, we have

$$N^{(\alpha_1,\alpha_2,\ldots,\alpha_l;\beta_1,\beta_2,\ldots,\beta_k)} = \left[N_{i,j}^{(\alpha_i,\beta_j)}\right].$$

Note that, if D_1 is isomorphic to D_2 via the isomorphism $(\alpha, \beta) \in Sym_v \times Sym_b$, i.e. $D_1^{(\alpha,\beta)} = D_2$, then

$$e^{(2)}_{j^{\beta^{-1}}} = (P_{s_{j^{\beta^{-1}}}}, P_{t_{j^{\beta^{-1}}}}) = (P_{x_j^{\alpha^{-1}}}, P_{y_j^{\alpha^{-1}}}),$$
$$\Leftrightarrow e^{(2)}_j = (P_{s_j}, P_{t_j}) = (P_{x_{j^\beta}^{\alpha^{-1}}}, P_{y_{j^\beta}^{\alpha^{-1}}}).$$

Theorem 2.5.3. *Let D_1 and D_2 be two digraphs of order v and size b. Let N_1 and N_2 be two $(m \times n)$ $\{0,1\}$-incidence matrices corresponding to D_1 and D_2, respectively, where $m = 2v$ and $n = v + b$. Then, D_1 is isomorphic to D_2 under the action of $Sym_v \times Sym_b$ if and only if N_1 is equivalent to N_2 under the action of $Sym_{v,v} \times Sym_{v,b}$.*

Proof. First, assume that D_1 is isomorphic to D_2 via an isomorphism $(\alpha, \beta) \in Sym_v \times Sym_b$, i.e. $D_1^{(\alpha,\beta)} = D_2$. Then, we want to show that N_1 is equivalent to N_2 under the action of $Sym_{v,v} \times Sym_{v,b}$, i.e. $N_1^{(\alpha,\alpha;\alpha,\beta)} = N_2$.

Claim 2.5.4. *Clearly,* $I_v^{(\alpha;\alpha)} = I_v$.

Claim 2.5.5. $A_1^{(\alpha;\beta)} = A_2$.

Proof.

$$\begin{aligned}
& a^{(2)}_{i^{\alpha^{-1}}, j^{\beta^{-1}}} = 1, \\
\Leftrightarrow\ & \pi_1(e^{(2)}_{j^{\beta^{-1}}}) = \pi_1(P_{s_{j^{\beta^{-1}}}}, P_{t_{j^{\beta^{-1}}}}) = P_{s_{j^{\beta^{-1}}}} = P_{i^{\alpha^{-1}}}, \\
\Leftrightarrow\ & P_{x_j^{\alpha^{-1}}} = P_{i^{\alpha^{-1}}}, \\
\Leftrightarrow\ & P_{x_j} = P_i, \\
\Leftrightarrow\ & a^{(1)}_{i,j} = 1, \\
\Leftrightarrow\ & A_2 = A_1^{(\alpha;\beta)}.
\end{aligned}$$

□

Claim 2.5.6. $B_1^{(\alpha;\beta)} = B_2$.

Proof.

$$\begin{aligned}
& b^{(2)}_{i^{\alpha^{-1}}, j^{\beta^{-1}}} = 1, \\
\Leftrightarrow\ & \pi_2(e^{(2)}_{j^{\beta^{-1}}}) = \pi_2(P_{s_{j^{\beta^{-1}}}}, P_{t_{j^{\beta^{-1}}}}) = P_{t_{j^{\beta^{-1}}}} = P_{i^{\alpha^{-1}}}, \\
\Leftrightarrow\ & P_{y_j^{\alpha^{-1}}} = P_{i^{\alpha^{-1}}}, \\
\Leftrightarrow\ & P_{y_j} = P_i, \\
\Leftrightarrow\ & b^{(1)}_{i,j} = 1, \\
\Leftrightarrow\ & B_2 = B_1^{(\alpha;\beta)}.
\end{aligned}$$

□

By Claims 2.5.4, 2.5.5, and 2.5.6, N_1 is equivalent to N_2.

Next, assume that $N_1^{(\alpha,\beta;\gamma,\delta)} = N_2$ for $\alpha, \beta, \gamma \in Sym_v$ and $\delta \in Sym_b$. Then,

$$\left[\begin{array}{c|c} I_v & A_1 \\ \hline I_v & B_1 \end{array}\right]^{(\alpha,\beta;\gamma,\delta)} = \left[\begin{array}{c|c} I_v & A_2 \\ \hline I_v & B_2 \end{array}\right]$$

if and only if $I_v^{(\alpha;\gamma)} = I_v$, and $I_v^{(\beta;\gamma)} = I_v$ if and only if $\alpha = \gamma$ and $\beta = \gamma$. Thus, $\alpha = \beta = \gamma$. Therefore,

$$\left[\begin{array}{c|c} I_v & A_1 \\ \hline I_v & B_1 \end{array}\right]^{(\alpha,\alpha;\alpha,\delta)} = \left[\begin{array}{c|c} I_v & A_2 \\ \hline I_v & B_2 \end{array}\right]$$

Let $\tau = \alpha \in Sym_v$ and $\sigma = \delta \in Sym_b$. Then,

$$e^{(2)}_{j^{\sigma^{-1}}} = (P_{s_{j^{\sigma^{-1}}}}, P_{t_{j^{\sigma^{-1}}}}) \in E_2.$$

Then, by projections π_1 and π_2, we get

$$\begin{aligned} & P_{s_{j^{\sigma^{-1}}}} = P_{i^{\tau^{-1}}} \text{ and } P_{t_{j^{\sigma^{-1}}}} = P_{i'^{\tau^{-1}}}, \\ \Leftrightarrow\ & a^{(2)}_{i^{\tau^{-1}}, j^{\sigma^{-1}}} = 1 \text{ and } b^{(2)}_{i'^{\tau^{-1}}, j^{\sigma^{-1}}} = 1, \\ \Leftrightarrow\ & a^{(1)}_{i,j} = 1 \text{ and } b^{(1)}_{i',j} = 1, \\ \Leftrightarrow\ & P_{x_j} = P_i \text{ and } P_{y_j} = P_{i'}, \\ \Leftrightarrow\ & e^{(1)}_j = (P_{x_j}, P_{y_j}) \in E_1. \end{aligned}$$

Therefore, $(\tau, \sigma) \in Sym_v \times Sym_b$ is an isomorphism between D_1 and D_2, and the proof is finished. □

2.6 Posets and Lattices

Definition 2.6.1. *A **partial ordering** on a finite set P is a relation "$\preceq$" on P which is:*

1. ***reflexive:*** *$x \preceq x$,*

2. ***antisymmetric:*** *$x \preceq y$ and $y \preceq x$ implies that $x = y$,*

3. ***transitive:*** *$x \preceq y$ and $y \preceq z$ implies that $x \preceq z$.*

for all $x, y, z \in P$. Then, the pair $(P, \preceq)$ is called a ***partially ordered set*** *or simply a* ***poset****.*

If $(P, \preceq)$ is a finite poset, then it can be represented by a **Hasse diagram**, which is a graph whose vertices are elements of P and the edges correspond to the covering relations. More precisely, an edge from x to y in P is present if

- $x \prec y$, and
- there is no $z \in P$ such that $x \prec z$ and $z \prec y$.

In general, if $x \prec y$, then we draw y lower than x. Because of that, the direction of the edges is never indicated in a Hasse diagram.

Example 2.6.1. If $P = \mathcal{P}(\{1,2,3\})$, and $\preceq$ is the subset relation $\subseteq$, then Figure 2.6.1 displays the Hasse diagram of the power set poset $(P, \preceq)$.

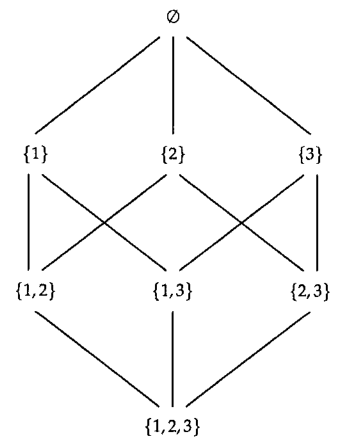


Figure 2.6.1: A Hasse diagram representing $(P, \preceq)$.

Even though $\{3\} \prec \{1,2,3\}$ for instance, there is no edge directly between them because there are in between elements in P, namely $\{2,3\}$ and $\{1,3\}$. However, there still remains an indirect path from $\{3\}$ to $\{1,2,3\}$. The reason that we do not draw such direct edge is that the graph then will get too busy, especially for large poset.

■

We say that a group G acts on the poset $(P, \preceq)$, if

$$x \preceq y \implies x^g \preceq y^g,$$

for all $x, y \in P$ and all $g \in G$.

Now, let $(P, \preceq)$ be a poset and let Q be a subset of P. An **upper (lower) bound** of Q is an element w with $q \preceq w$ ($w \preceq q$) for all $q \in Q$. **The least upper bound of Q (the supremum of Q)** is an upper bound w such that $w \preceq w'$ for any other upper bound w'. On the other hand, **the greatest lower bound of Q (the infimum of Q)** is a lower bound w such that $w' \preceq w$ for any other lower bound w'.

If the supremum (or the infimum) of a set exists, then it is unique. We write $x \vee y$ for the supremum of $x, y \in P$, and $x \wedge y$ for the infimum.

Definition 2.6.2. *A finite set $\mathcal{L}$ is called a **lattice**, if*

1. *$(\mathcal{L}, \preceq)$ is a poset,*

2. *Any two elements x and y of $\mathcal{L}$ have an infimum and a supremum.*

Let $(\mathcal{L}, \preceq)$ be a lattice. A **rank function** for $\mathcal{L}$ is a mapping $rk : \mathcal{L} \to \mathbb{N}$, $x \mapsto rk(x)$ satisfying

$$x \prec y \implies rk(x) < rk(y),$$

for all $x, y \in \mathcal{L}$. In this case, the lattice $(\mathcal{L}, \preceq)$ is called a **ranked lattice**.

A rank function induces layers on the lattice $\mathcal{L}$. The i^{th} layer consists of the elements of rank i:

$$L^{(i)}(\mathcal{L}) = \{x \in \mathcal{L} \mid rk(x) = i\}.$$

Example 2.6.2. The power set poset $(P, \preceq)$ of Example 2.6.1 is a ranked lattice if we consider $rk : P \to \mathbb{N}$, $x \mapsto |x|$ (cardinality). Then the layers of this ranked lattice are as follows:

layer	elements of layers
0	$\emptyset$
1	$\{1\}, \{2\}, \{3\}$
2	$\{1,2\}, \{1,3\}, \{2,3\}$
3	$\{1,2,3\}$

■

If $\mathcal{L}$ is a ranked lattice, then

$$\mathcal{L} = \bigcup_{i \in \mathbb{N}} L^{(i)}(\mathcal{L}).$$

2.7 The Lexicographical Order

If x and y are two elements of a poset $(P, \preceq)$, with $x \preceq y$ or $y \preceq x$, then we say that x and y are **comparable**, and we say that x and y are **incomparable** otherwise.

If $(P, \preceq)$ is a poset with every pair of distinct elements in P are comparable, then the order is called a **total order**, and P is called a **totally ordered set**. That is, given a poset $(P, \prec)$, we embed into a total order $(P, <)$ such that

$$x \prec y \Longrightarrow x < y,$$

for all $x, y \in P$.

Let $(X, \leq)$ (or simply $X_{\leq}$) be a totally ordered set. The total order on X induces a total order on $\mathcal{P}(X)$. This is the **Lexicographical order**, denoted by $\preceq$. See [7] for more discussion on such ordering.

Definition 2.7.1 (The Lexicographic Order). *For subsets* $A = \{a_1, a_2, \ldots, a_m\}$ *and* $B = \{b_1, b_2, \ldots, b_n\}$ *of the totally ordered set* X *with* $a_1 < a_2 < \cdots < a_m$ *and* $b_1 < b_2 < \cdots < b_n$*, we have*

$$A \preceq B \leftrightarrow \begin{cases} \exists r \leq min(m,n) \ : \ a_i = b_i \, \forall \, 1 \leq i < r \ and \ a_r < b_r \quad , \ or \\ m \leq n \ and \ a_i = b_i \, \forall \, 1 \leq i \leq m, \end{cases}$$

and we say that A *is lexicographically less than* B*, denoted by "*$A \preceq B$*".*

Let $(X, \leq)$ be a totally ordered finite set. The lexicographical order on $\mathcal{P}(X)$ can be represented by a tree, the **order tree** $T_{(X,\preceq)}$ or simply $T_{\preceq}$. The nodes of $T_{\preceq}$ are the subsets of X. Two subsets A and B of X are connected by an edge if $A \subseteq B$ and $B = A \cup \{\max \ B\}$, i.e. $|B| = |A| + 1$.

Example 2.7.1. Consider the four elements set $X = \{1, 2, 3, 4\}_{\leq}$. Figure 2.7.1 displays the order tree $T_{\preceq}$. Here, we label the nodes by the largest element of the set which they represent.

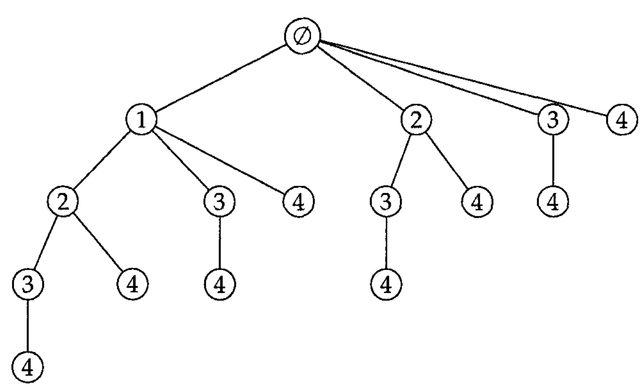


Figure 2.7.1: Order tree of $\mathcal{P}(X)$ with respect to the lexicographical ordering.

■

There are two different ways of traversing the nodes of a tree.

Definition 2.7.2 (Depth-first search). *The* ***depth-first search (DFS)*** *visits all the vertices of a graph as follows. Initially, all vertices are marked "new".*

When a vertex is visited, it is marked "old". DFS works by selecting a new vertex v, marking it old, and then calling itself recursively on each of the vertices adjacent to v.

Definition 2.7.3 (Breadth-first search). *The* ***breadth-first search (BFS)*** *starts from a specified source vertex s from which it visits all nodes adjacent to s. In this strategy, every level in the tree is visited from left to right, where we start from the top.*

So the BFS proceeds as follows:

$$\begin{aligned} \phi &\prec 1 \prec 2 \prec 3 \prec 4 \\ &\prec 12 \prec 13 \prec 14 \prec 23 \prec 24 \prec 34 \\ &\prec 123 \prec 124 \prec 134 \prec 234 \\ &\prec 1234 \end{aligned}$$

while DFS goes as follows:

$$\begin{aligned} \phi &\prec 1 \prec 12 \prec 123 \prec 1234 \prec 124 \\ &\prec 13 \prec 134 \prec 14 \\ &\prec 2 \prec 23 \prec 234 \prec 24 \\ &\prec 3 \prec 34 \prec 4. \end{aligned}$$

More discussion on such techniques can be found in [1, 33, 38, 41, 55].

Now, if $X_\prec$ and $Y_\prec$ are two finite totally ordered sets by the meaning of lexicographical order, then for $(x_1, y_1), (x_2, y_2) \in X \times Y$, we have

$$(x_1, y_1) \prec (x_2, y_2) \leftrightarrow \begin{cases} x_1 \prec x_2 \text{ , or} \\ x_1 = x_2 \text{ and } y_1 \prec y_2. \end{cases}$$

2.8 Tactical Decompositions

Let $\mathcal{X} = (P, \mathcal{B})$ be an incidence structure with $P = \{p_1, p_2, \ldots, p_m\}$ and $\mathcal{B} = \{B_1, B_2, \ldots, B_n\}$. A **decomposition** of $\mathcal{X}$ is a pair of partitions of points and blocks. Let $\Pi(P)$ and $\Pi(\mathcal{B})$ be the set of all partitions of P and $\mathcal{B}$, respectively. Let $(\mathcal{R}, \mathcal{C}) \in \Pi(P) \times \Pi(\mathcal{B})$ be a decomposition of $\mathcal{X}$ where $\mathcal{R} = \{R_1 \mid R_2 \mid \ldots \mid R_M\}$ and $\mathcal{C} = \{C_1 \mid C_2 \mid \ldots \mid C_N\}$.

For $1 \leq i \leq M$ and $1 \leq j \leq N$, define

$$r_{i,j} = \left|\{B \in C_j \mid p \in B\}\right|$$

with $p \in R_i$ fixed. In addition, define

$$c_{i,j} = \left|\{p \in R_i \mid p \in B\}\right|$$

for fixed $B \in C_j$. The decomposition $(\mathcal{R}, \mathcal{C})$ is called **row tactical** if for any $i \leq M$ and $j \leq N$, the number $r_{i,j}$ is independent of the choice $p \in R_i$. It is called **column tactical** if for any $i \leq M$ and $j \leq N$, the number $c_{i,j}$ is independent of the choice $B \in C_j$. The decomposition $(\mathcal{R}, \mathcal{C})$ is called **tactical** if it is both row and column tactical. See [4, 5] for more reading.

If A is an incidence matrix corresponding to $\mathcal{X}$, then the decomposition concepts defined above can be applied to rows and columns of A. Any decomposition allows us to reorder rows and columns of the incidence matrix in order to group together rows and columns according to the classes of the decompositions. Thus, any decomposition gives rise to a block decomposition of the incidence matrix A. The submatrices of size $|R_i| \times |C_j|$ are the **decomposition matrices**.

The matrices containing $r_{i,j}$ and $c_{i,j}$ extended by one row and column indicating the order of the point and block classes are the **row** and **column decomposition schemes**. Let $(\mathcal{R}, \mathcal{C})$ be a decomposition and put $|\mathcal{R}| = M$ and $|\mathcal{C}| = N$. Then the row and column decomposition schemes have the form as in Figure 2.8.1.

$$
\begin{array}{c|cccc}
 & |C_1| & |C_2| & \cdots & |C_N| \\
\hline
|R_1| & r_{1,1} & r_{1,2} & \cdots & r_{1,N} \\
|R_2| & r_{2,1} & r_{2,2} & \cdots & r_{2,N} \\
\vdots & \vdots & & & \vdots \\
|R_M| & r_{M,1} & r_{M,2} & \cdots & r_{M,N}
\end{array}
\quad , \text{ and } \quad
\begin{array}{c|cccc}
 & |C_1| & |C_2| & \cdots & |C_N| \\
\hline
|R_1| & c_{1,1} & c_{1,2} & \cdots & c_{1,N} \\
|R_2| & c_{2,1} & c_{2,2} & \cdots & c_{2,N} \\
\vdots & \vdots & & & \vdots \\
|R_M| & c_{M,1} & c_{M,2} & \cdots & c_{M,N}
\end{array}
$$

Figure 2.8.1: The row and column decomposition schemes forms.

Consider a graph of order and size 5 with the incidence matrix of Figure 2.8.2 where v_i and e_i represents vertices and edges of the graph for $1 \leq i \leq 5$. The **bolded lines** in the matrix indicate the row and column classes. The associated row "($\rightarrow$)" and column "($\downarrow$)" decomposition schemes are given below:

	e_1	e_2	e_3	e_4	e_5
v_1	X	X	X		
v_2	X			X	
v_3		X			X
v_4			X		
v_5				X	X

Figure 2.8.2: Partitioned incidence matrix of a given graph.

$$
\begin{array}{c|ccc}
\rightarrow & 2 & 1 & 2 \\
\hline
1 & 2 & 1 & 0 \\
2 & 1 & 0 & 1 \\
1 & 0 & 1 & 0 \\
1 & 0 & 0 & 2
\end{array}
\quad , \quad
\begin{array}{c|ccc}
\downarrow & 2 & 1 & 2 \\
\hline
1 & 1 & 1 & 0 \\
2 & 1 & 0 & 1 \\
1 & 0 & 1 & 0 \\
1 & 0 & 0 & 1
\end{array}
$$

Such decompositions can be useful in the problem of isomorphism as we will see in Chapter 5 on page 103.

2.9 Hashing Functions

A **hash function** or a **hash algorithm** is a function for summarizing data. Such a summary is known as a hash value or simply a hash, and the process of computing such a value is known as hashing.

In some cases it is useful to use a suitable hashing function so that it can store some combined data in one number in $\mathbb{N}$, for instance.

Let $\mathbb{N}^{\mathbb{N}}$ denote sequences of numbers in $\mathbb{N}$ of arbitrary length with the convention that

$$(x_0, x_1, \ldots, x_s) = (x_0, x_1, \ldots, x_s, 0, 0, \ldots).$$

Then $h : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ is a hash function that stores information about $(x_0, x_1, \ldots, x_s)$ into one number $h((x_0, x_1, \ldots, x_s))$.

2.10 $\{0,1\}$-Matrices

In this section, we introduce some of the notations on $\{0,1\}$-matrices. Recall that $M_{m,n}$ denotes the class of all $m \times n$ $\{0,1\}$-matrices.

Definition 2.10.1. *The **Kronecker product** is a binary operator on matrices. Given a $m \times n$ matrix $A_{m\times n}$ and the $p \times q$ matrix $B_{p\times q}$*

$$A = \begin{bmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & \ddots & \vdots \\ a_{m,1} & \cdots & a_{m,n} \end{bmatrix}_{\times n} \qquad B = \begin{bmatrix} a_{1,1} & \cdots & a_{1,m} \\ \vdots & \ddots & \vdots \\ a_{n,1} & \cdots & a_{n,m} \end{bmatrix}_{p\times q}$$

then their Kronecker product, denoted $A \otimes B$, is the $mp \times nq$ block matrix

$$A \otimes B = \begin{bmatrix} a_{1,1}B & \dots & a_{1,n}B \\ \vdots & \ddots & \vdots \\ a_{m,1}B & \dots & a_{m,n}B \end{bmatrix}_{mp \times nq}$$

For $A \in M_{m,n}$, let $\text{Rowsupp}(A) = \{1 \leq i \leq m \mid \text{row-sum}_i(A) \geq 1\ \}$. Also, for $0 \leq l \leq m$, let

$$M_{m,n}^{(l)} = \left\{A \in M_{m,n} \mid |\text{Rowsupp}(A)| = l\right\} \tag{2.5}$$

Therefore,

$$M_{m,n} = \bigcup_{l=0}^{m} M_{m,n}^{(l)}.$$

For instance,

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \in M_{4,4}^{(2)}.$$

Let $A = (a_{i,j})$ where $0 \leq i \leq m-1$ and $0 \leq j \leq n-1$ be in $M_{m,n}$. Then, we write

$$\text{Row}_i(A) := [a_{i,0}, a_{i,1}, \dots, a_{i,n-1}],$$

for the entries of A that are in row i. Let $E_i \in M_{m,1}$ for $0 \leq i \leq m-1$ such that

$$\text{Row}_j(E_i) = \begin{cases} 1, & \text{if } i = j; \\ 0, & \text{otherwise.} \end{cases}$$

We write $v = [v_0, v_1, \dots, v_{n-1}]$ for a vector of length n, i.e. $v \in M_{1,n}$.

Definition 2.10.2. *If $v \in M_{1,n}$ and $E_i \in M_{m,1}$ for some $0 \leq i \leq m-1$, then their Kronecker product $B = E_i \otimes v$ is in $M_{m,n}^{(1)}$ with $Row_i(B) = v$.*

$$E_i \otimes v = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \otimes \begin{bmatrix} v_0, & v_1, & \dots, & v_{n-1} \end{bmatrix} = \begin{bmatrix} 0 & 0 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & 0 \\ v_0 & v_1 & \dots & v_{n-1} \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix}.$$

Therefore, if $A \in M^{(l)}_{m,n}$, then $A + (E_i \otimes v) =: B$, where

$$\mathrm{Row}_j(B) = \begin{cases} [a_{j,0} + v_0, \dots, a_{j,n-1} + v_{n-1}], & \text{if } i = j; \\ [a_{j,0}, \dots, a_{j,n-1}], & \text{otherwise.} \end{cases}$$

Note that $A - (E_i \otimes v)$ is defined in a similar way.

Chapter 3

The Theory of Isomorph Rejection

Let G be a group that acts on two finite sets X and Y. Let R be a G-invariant relation on the product $X \times Y$.

The problem to solve in this chapter is that we are given an orbit transversal $\mathcal{T}(G, X)$ and we are asked to construct an orbit transversal $\mathcal{T}(G, Y)$. This can be done by considering a construction of a transversal for a secondary action on the relation R, and then use that transversal in solving our primary transversal problem.

This problem is fundamental for the construction and classification of discrete structures. One can see Higman's work [36, 37] on **coherent configurations** with the assumption of finite group actions, and the fundamental book of finite group actions and applications by Kerber [53]. We also refer to Laue [57] and McKay [64] for further details.

In the following, we will summarize the known facts regarding the solution of this problem in our own language. The results are not new. McKay's paper [64] contains most of these ideas. However, we find that his paper is very condensed and therefore hard to read. For this reason, we find it necessary to present the theory here once again and in some more detail.

We follow McKay's ideas by first providing the framework for a classification algorithm. This framework involves a μ-function, which is used to eliminate isomorphic copies. Finding and realizing such a μ-function seems to be the hardest issue. It is possible to rephrase orderly generation in the language of this μ-function, and we will do so in Theorem 3.2.1. Probably McKay's major contribution is the idea of using partition backtrack to provide another realization of a μ-function. This will be our Theorem 6.3.1 on page 143. In order to get to this result, it is necessary to present a range of algorithms which facilitate partition backtrack for incidence structures (or $\{0,1\}$-matrices in general). This material is covered in Chapters 5 on page 103 and 6 on page 118. The reason why McKay's μ-function is interesting is that it can be computed faster than the μ-function based on the lexicographical ordering from Theorem 3.2.1. The ideas for partition backtrack and for the invariants we use are drawn from a variety of different sources. Leon [75, 76] has a series of papers on partition backtrack. The invariant for $\{0,1\}$-matrices we are going to discuss in Chapter 5 on page 103 is from D. Betten and M. Braun [9]. Last but not least, Chapter 4 on page 74 is about the search space. In there, we present some more specific material which addresses the construction of regular graphs with given girth.

3.1 Orbits on Ordered Pairs

Let G be a group that acts on two finite sets X and Y. Recall that a subset $\Delta \subseteq X$ is called G-invariant if $G_{\Delta} = \{g \in G \mid \Delta^g = \Delta\} = G$. Moreover, let G act **coordinate-wise** on the product $X \times Y$, i.e. for $(x,y) \in X \times Y$ and $g \in G$, we have

$$(x,y)^g = (x^g, y^g).$$

From Lemma 2.4.5 it follows that R is a union of G-orbits on pairs from $X \times Y$. Any G-orbit on pairs from $X \times Y$ which is contained in R is called **flag orbit**.

For $x \in X$ and $y \in Y$, define the following **projection maps**:

$$\pi_1 : R \to X, \quad \text{defined by} \quad (x, y) \mapsto x, \tag{3.1}$$

and,

$$\pi_2 : R \to Y, \quad \text{defined by} \quad (x, y) \mapsto y, \tag{3.2}$$

Note that for any $g \in G$, we have

$$\pi_1\big((x,y)^g\big) = \pi_1\big((x^g, y^g)\big) = x^g = \pi_1\big((x,y)\big)^g,$$

and thus

$$\pi_1((x,y)^g) = \pi_1((x,y))^g. \tag{3.3}$$

Similarly, we have $\pi_2((x,y)^g) = \pi_2((x,y))^g$.

Definition 3.1.1. *Let G be a group act on two finite sets X and Y, and let $R \subseteq X \times Y$ be a G-invariant relation. Let π_1 and π_2 be defined as in (3.1) and (3.2), respectively. Then, for any $(x, y) \in R$, the* ***shadow orbits***

$$\pi_1\big((x,y)^G\big) = x^G, \qquad \text{and} \quad \pi_2\big((x,y)^G\big) = y^G.$$

We also consider the following two sets

$$\pi_1^{-1}(x) = \{\ (x, z) \in R\ \}, \tag{3.4}$$

and,

$$\pi_2^{-1}(y) = \{\ (z, y) \in R\ \}, \tag{3.5}$$

which we call the "**extension set**" and the "**pre-image set**", respectively.

Suppose that G acts transitively on a finite set X, and let $R \subseteq X \times X$ be a G-invariant relation. The **diagonal** $I = \{(x,x) \in R \mid x \in X\}$ is a G-invariant relation.

A flag orbit determines two shadow orbits, but the converse is not true as the following example shows.

Example 3.1.1. Consider a transitive action of a group G on a finite set X, and let $R \subseteq X \times X$ be a G-invariant relation.

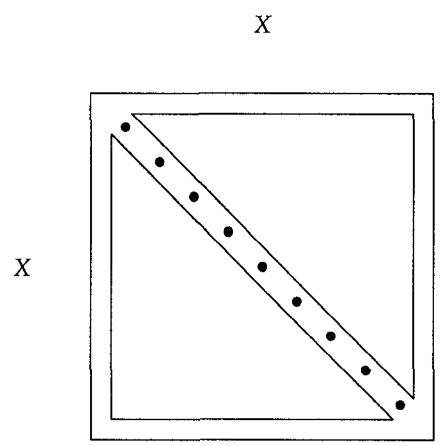


Figure 3.1.1: Orbits of a transitive group G on $X \times X$.

Since G is a transitive, G has one orbit on X. However, for all $(x, x) \in R$ and for all $g \in G$, we have $(x, x)^g = (y, y)$ for some $y \in X$, and thus $(x, x)^G = I$, where I is represented in Figure 3.1.1 by •'s. Therefore, G has at least 2 flag orbits on $X \times X$ with the same shadow orbits.

■

Theorem 3.1.2. *Let G be a group acting on two finite sets X and Y as above and let R be a G-invariant relation between X and Y. Moreover, let π_1 and π_2 be projection maps as defined in (3.1) and (3.2), respectively. Then, the following three sets are in canonical one-to-one correspondence.*

1. *The set of flag orbits R/G,*

2. $$\bigcup_{x \in \mathcal{T}(G,X)} \pi_1^{-1}(x)/G_x,$$
   which is called "G_x-orbits of the x-extensions for all $x \in \mathcal{T}(G, X)$", and

3.

$$\bigcup_{y \in \mathcal{T}(G,Y)} \pi_2^{-1}(y)/G_y,$$

which is called "G_y-orbits of the y-preimages for all $y \in \mathcal{T}(G,Y)$",

The canonical correspondence between the objects in 1 and in 2 and between the objects in 1 and in 3 is characterized by the fact that objects correspond whenever they intersect nontrivially.

Proof. Let $\mathcal{T}(G,X)$ be any given orbit transversal for G on X. Then, define a function

$$f : R \to \bigcup_{x \in \mathcal{T}(G,X)} \pi_1^{-1}(x)/G_x$$

defined by $(a,b) \mapsto (\ (a^g, b^g)\ )^{G_x}$ where $g \in G$ is such that $a^g = x \in \mathcal{T}(G,X)$ (such an element $g \in G$ exists). We first show that f is well defined. Assume that

$$(a,b) \mapsto (\ (a^{g'}, b^{g'})\ )^{G_x}$$

for some other element $g' \in G$. Then,

$$\begin{aligned}
&a^g = x = a^{g'} \\
&a = x^{g^{-1}} = x^{g'^{-1}} \implies x = x^{g'^{-1}g} \\
&h = g'^{-1}g \in G_x.
\end{aligned}$$

Therefore,

$$\begin{aligned}
(\ (a^{g'}, b^{g'})\ )^{G_x} &= (\ (x, b^{g'})\ )^{G_x} \\
&= (\ (x, b^{g'})^h\ )^{G_x} \\
&= (\ (x, b^g)\ )^{G_x} = (\ (a^g, b^g)\ )^{G_x}.
\end{aligned}$$

Hence, f is a well defined mapping.

Next, we show that f is a G-invariant, i.e. we want to show that

$$f(\ (a,b)\ ) = f(\ (a,b)^s\ ) \ \text{ for all } \ s \in G.$$

Then, it suffices to show that

$$(a,b) \mapsto (\ (a^g, b^g)\ )^{G_x} \quad \text{and} \quad (a,b) \mapsto (\ (a^{sg'}, b^{sg'})\ )^{G_x}$$

for $a^g = a^{sg'} = x \in \mathcal{T}(G,X)$, are the same.

Note that for any $a \in X$, we have $a^G = (a^s)^G = x^G$ since $a^g = x \in \mathcal{T}(G,X)$. Then, there exist $g, g' \in G$ such that $a^g = a^{sg'} = x$, then

$$\begin{aligned}
&a^g = x = a^{sg'}\\
&a = x^{g^{-1}} = x^{g'^{-1}s^{-1}} \implies x = x^{g'^{-1}s^{-1}g}\\
&h = g'^{-1}s^{-1}g \in G_x.
\end{aligned}$$

Now,

$$\begin{aligned}
f(\ (a^s, b^s)\ ) &= (\ (a^{sg'}, b^{sg'})\ )^{G_x}\\
&= (\ (a^{sg'}, b^{sg'})^h\ )^{G_x}\\
&= (\ (a^{sg'g'^{-1}s^{-1}g}, b^{sg'g'^{-1}s^{-1}g})\ )^{G_x}\\
&= (\ (a^g, b^g)\ ) = f(\ (a,b)\ )^{G_x}.
\end{aligned}$$

Therefore, f is well a defined and G-invariant mapping. Therefore, f descends to a map

$$\tilde{f} : R/G \longrightarrow \bigcup_{x \in \mathcal{T}(G,X)} \pi_1^{-1}(x)/G_x$$

Then, what we want to show is that $\tilde{f}$ is onto and a one-to-one map.

First, we show that $\tilde{f}$ is onto. Choose any $\mathcal{O} = ((x,y))^{G_x} \in \pi_1^{-1}(x)/G_x$ with $x \in \mathcal{T}(G,X)$.

$$f((x,y)) = (x^{id}, y^{id})^{G_x} = \mathcal{O},$$

by definition because $x^{id} = x$. Since in addition f is G-invariant, then $\tilde{f}$ is onto.

Next, we show that $\tilde{f}$ is a one-to-one function. Assume that

$$f((x_1,y_1)) = f((x_2,y_2)) = \mathcal{O} = ((x,y))^{G_x},$$

then $x_1 \cong_G x \cong_G x_2$. Let $g_1, g_2 \in G$ such that

$$x_1^{g_1} = x = x_2^{g_2}.$$

Then,

$$f((x_1, y_1)) = ((x, y_1^{g_1}))^{G_x} = \mathcal{O},$$
$$f((x_2, y_2)) = ((x, y_2^{g_2}))^{G_x} = \mathcal{O},$$

thus, there exists $h \in G_x$ such that $y_1^{g_1 h} = y_2^{g_2}$, and hence $y_1^{g_1 h g_2^{-1}} = y_2$ and $g_1 h g_2^{-1} \in G$.

$$x_1^{g_1 h g_2^{-1}} = x^{h g_2^{-1}} = x^{g_2^{-1}} = x_2.$$

Thus, $(x_1, y_1)^{g_1 h g_2^{-1}} = (x_2, y_2)$, and therefore $(x_1, y_1) \cong_G (x_2, y_2)$. Since f is G-invariant, then $\tilde{f}$ is one to one.

Thus, there is a one-to-one correspondence between (1) and (2). Similarly, we can show by analogous arguments that there is a one-to-one correspondence between (1) and (3), and thus the proof is complete. □

Note that, if G was transitive on both X and Y, then the previous theorem is called **Mackey's Theorem**, see Theorem 1.2.16 of Kerber [53].

Definition 3.1.3. *Let G act on sets X and Y and let R be a G-invariant relation between X and Y with $\pi_2(R) = Y$. A μ-function is a function from Y to $\mathcal{P}(R)$ such that the following properties hold:*

1. *$\mu(y)$ is an orbit of G_y on $\pi_2^{-1}(y)$, and*

2. *$\mu(y^g) = \mu(y)^g$ for all $g \in G$.*

In other words, μ associates to every $y \in Y$ a non-empty (single) G_y-orbit $\mu(y) \subseteq \pi_2^{-1}(y)$ such that $\mu(y^g) = \mu(y)^g$ for all $g \in G$.

These requirement on μ are saying that μ must identify (or select) one of the orbits of G_y on $\pi_2^{-1}(y)$ in a way that is independent of y in such a way that it depends only on the G-orbit of y, but not on y itself. That is, if y were to be replaced by $z \in G(y)$, then the orbit selected for z would be the image of the orbit selected for y under any isomorphism from y to z. Formally, if $z = y^g$ for $g \in G$, then the orbit selected for z is the g-image of the orbit selected for y. Such a G_y-orbit is called the **canonical orbit**. Such μ function can be realized by considering a canonical labeling map φ, see Definition 2.4.14.

Theorem 3.1.4. *Let G act on two finite sets X and Y, and let R be a G-invariant relation between X and Y with $\pi_2(R) = Y$. Assume that $\mathcal{T}(G,X)$, a transversal for the G-orbits on X, is known. Suppose that a μ-function as defined in Definition 3.1.3 is known. Then,*

$$\mathcal{T}(G,Y) = \bigcup_{x \in \mathcal{T}(G,X)} \pi_2\Big(\Big\{ (x,y) \in \mathcal{T}(G_x, \pi_1^{-1}(x)) \mid (x,y) \in \mu(y) \Big\}\Big) \tag{3.6}$$

is a transversal for the G-orbits on Y.

Proof. We need to show that $\mathcal{T}(G,Y)$ is a transversal for G-orbits on Y. This can be done by showing that each G-orbit on Y is represented at least once and at most once, and hence each G-orbit on Y is represented exactly once.

First, we show that each G-orbit on Y is represented at least once. Given $y \in Y$, $\pi_2^{-1}(y) \neq \emptyset$, by assumption. Let $(x_1,y),(x_2,y),\ldots,(x_r,y)$ be representatives for the G_y-orbits on $\pi_2^{-1}(y)$. We may assume that $(x_1,y)^{G_y} = \mu(y)$. Note that, $(x_1,y)^{G_y}$ is the G_y-orbit of (x_1,y).

By Theorem 3.1.2, there exists $x \in \mathcal{T}(G,X)$, and there exists $(x,y') \in \pi_1^{-1}(x) \subseteq R$ such that $(x_1,y)^{G_y}$ corresponds to $(x,y')^{G_x}$. This means that $(x_1,y)^G = (x,y')^G$ (as flag orbits). Therefore, there exists a $g \in G$ such that

$$(x_1^g, y^g) = (x,y').$$

Thus, $y^g = y'$. We claim that $(x, y') \in \mu(y')$.

$$\begin{aligned}\mu(y') = \mu(y^g) = \mu(y)^g &= \left[(x_1, y)^{G_y}\right]^g = \left[(x_1, y)\right]^{G_y g} \\ &= \left[(x_1, y)\right]^{gg^{-1}G_y g} = \left[(x_1, y)^g\right]^{G_{y^g}} = \left(x, y'\right)^{G_{y^g}}, \\ &= \left[(x, y')\right]^{G_{y'}} \in \pi_2^{-1}(y')/G_{y'},\end{aligned}$$

by Definition 3.1.3 and Lemma 2.4.2.

Let $(x, y'') \in \mathcal{T}(G_x, \pi_1^{-1}(x))$ with $(x, y'')^{G_x} = (x, y')^{G_x}$ (such a (x, y'') exists). Then there exists a $h \in G_x$ such that $y'^h = y''$, and

$$\begin{aligned}\mu(y'') = \mu(y'^h) = \mu(y')^h &= \left[(x, y')^{G_{y^g}}\right]^h = \left[(x, y')\right]^{G_{y^g} h} \\ &= \left[(x, y')\right]^{hh^{-1}G_{y^g} h} = \left[(x, y')^h\right]^{G_{y^{gh}}} = \left(x, y''\right)^{G_{y^{gh}}},\end{aligned}$$

Therefore, we have found an element $x \in \mathcal{T}(G, X)$ and a pair $(x, y'') \in \mathcal{T}(G_x, \pi_1^{-1}(x))$ such that $(x, y'') \in \mu(y'')$. This means that the union in (3.6) contains the element y'', and since $y'' = y^{gh}$, this means that every G-orbit on Y is represented at least once in (3.6).

Second, assume that $x_1, x_2 \in \mathcal{T}(G, X)$, $(x_i, y_i) \in \mathcal{T}(G_{x_i}, \pi_1^{-1}(x_i))$, and $(x_i, y_i) \in \mu(y_i)$, for $i = 1, 2$. Assume further that $y_1^G = y_2^G$. We have to show that $x_1 = x_2$ and $y_1 = y_2$.

Let $g \in G$ with $y_1^g = y_2$, then

$$(x_1, y_1)^g \in \mu(y_1)^g = \mu(y_1^g) = \mu(y_2).$$

Therefore, there exists $h \in G_{y_2}$ such that $(x_1, y_1)^{gh} = (x_2, y_2)$. Thus, $(x_1^{gh}, y_2) = (x_2, y_2)$, and $x_1^{gh} = x_2$. But both x_1 and x_2 are contained in $\mathcal{T}(G, X)$. Therefore, $x_1 = x_2$.

Note that $h \in G_{y_2}$ and $x_1^{gh} = x_2 = x_1$, and thus $gh \in G_{x_1}$. Then,

$$\left[(x_1, y_1)\right]^{G_{x_1}} = \left[(x_1, y_1)^{gh}\right]^{G_{x_1}} = \left[(x_1, y_2)\right]^{G_{x_1}}.$$

But both (x_1, y_1) and (x_1, y_2) are in $\mathcal{T}(G_{x_1}, \pi_1^{-1}(x_1))$. Therefore, $(x_1, y_1) = (x_1, y_2)$, and hence $y_1 = y_2$.

Therefore, the union in (3.6) contains each G-orbit on Y at most once, and hence exactly once. □

By the previous Theorem, we can solve the problem of constructing a transversal for G-orbits on Y from a transversal for G-orbits on X by considering the following two steps. First, consider the action of G_x for all $x \in \mathcal{T}(G, X)$ on the set $\pi_1^{-1}(x)$ which results in a transversal $\mathcal{T}(G, R)$. Second, we project by π_2 on elements $(x, y) \in \mathcal{T}(G, R)$ that are contained in the G_y-orbit on $\pi_2^{-1}(y)$. These two steps are called **lifting orbits** and **projecting orbits** steps, respectively. In practice, these two steps can be described by Algorithm 3.1.1. Moreover, Figure 3.1.2 shows the steps from an element $x \in \mathcal{T}(G, X)$ to an element $y \in \mathcal{T}(G, Y)$ passing through a pair $(x, y) \in \mathcal{T}(G_x, \pi_1^{-1}(x))$ such that $(x, y) \in \mu(y)$.

Algorithm 3.1.1 Two-Steps($\mathcal{T}(G, X)$: an orbit transversal)

1: let $\mathcal{T}(G, Y) = \emptyset$
2: **for** each $x \in \mathcal{T}(G, X)$ **do**
3:    compute $\pi_1^{-1}(x)$
4:    compute $\mathcal{T}(G_x, \pi_1^{-1}(x))$
5:    **for** each $(x, y) \in \mathcal{T}(G_x, \pi_1^{-1}(x))$ **(lifting) do**
6:       **if** $(x, y) \in \mu(y)$ **(projecting) then**
7:          Add y to $\mathcal{T}(G, Y)$.
8:       **end if**
9:    **end for**
10: **end for**

For our convenience, given an element $(x, y) \in R \subseteq X \times Y$, we say that the node (x, y) is **accepted** if it passes the test in line (6) in Algorithm 3.1.1, and is **rejected** otherwise.

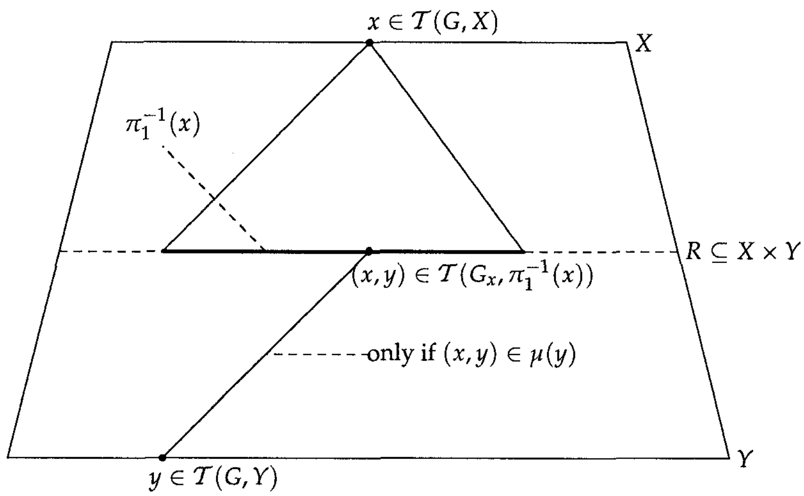


Figure 3.1.2: The steps of Algorithm 3.1.1.

3.2 Orderly Generation

Isomorph-rejection techniques are widely used in classification algorithms. One goal of such techniques is to produce a list of objects with no isomorphs. Another is to avoid redundant work in the search for objects of interests. See [78].

In this section, we construct a μ-function satisfying the properties of Definition 3.1.3 by employing the ideas of Faradžev [23] and Read [69] in the 1970's, independently. This isomorph-rejection technique is called **orderly generation.**

Let a group G act on two finite totally ordered sets X and Y, and let $R \subseteq X \times Y$ be a G-invariant relation. Recall from Definition 2.4.14 that a canonical labeling map $\varphi : Y \to G$ maps an element $y \in Y$ to its orbit representative under the action of G, denoted by $y^{\varphi(y)}$. According to Faradžev [23] and Read [69], an element $y \in Y$ is called a canonical orbit representative if y satisfies the following condition:

$$y \preceq y^g, \quad \text{for all } g \in G. \tag{3.7}$$

In other words, the canonical orbit representative $y^{\varphi(y)}$ is the least element in the orbit $G(y)$ under the given total ordering of Y.

Theorem 3.2.1. *Let a group G act on two finite totally ordered sets X and Y, and let R be a G-invariant relation between X and Y with $\pi_2(R) = Y$. Then, for any given $y \in Y$, let*

$$\mu(y) = \left[\left(\min \pi_2^{-1}(\ y^{\varphi(y)}\ )\right)^{\varphi(y)^{-1}}\right]^{G_y}$$

where the minimum is taken with respect to order on Y. Then, μ satisfies the conditions of Definition 3.1.3. In particular, given $\mathcal{T}(G,X)$, $\mathcal{T}(G,Y)$ can be constructed by using (3.6).

Proof. All that we need to show here is that property 1 and 2 are satisfied with this definition of μ. Then the proof is completed by Theorem 3.1.4.

Let us first show that $\mu(y)$ is a G_y-orbits on $\pi_2^{-1}(y)$. Clearly R is a G-invariant relation and $G_y \leq G$, then G_y acts on R and R is a G_y-invariant relation. Because $\pi_2^{-1}(y) \subseteq R$ and G_y acts on R, we obtain that G_y acts also on $\pi_2^{-1}(y)$. Now, let min $\pi_2^{-1}(y^{\varphi(y)}) = (x_0, y^{\varphi(y)})$. Then by applying $\varphi(y)^{-1}$ on $(x_0, y^{\varphi(y)})$ we would get $(x_0^{\varphi(y)^{-1}}, y) \in \pi_2^{-1}(y)$. Hence the first property of μ holds.

Next we show that $\mu(y^g) = \mu(y)^g$. Note that since $y^{\varphi(y)} = \rho(y) = y^{g\varphi(y^g)}$, we have $\varphi(y)\varphi(y^g)^{-1}$ maps y to y^g. Moreover, $G_{y^g} = g^{-1}G_y g$. Then

$$\begin{aligned}
\mu(y^g) &= \left[\left(\ \min\ \pi_2^{-1}(\ y^{g\varphi(y^g)}\ )\ \right)^{\varphi(y^g)^{-1}}\right]^{G_{y^g}} \\
&= \left[\left(\ \min\ \pi_2^{-1}(\ y^{\varphi(y)}\ )\ \right)^{\varphi(y^g)^{-1}}\right]^{g^{-1}G_y g} \\
&= \left[\left(\ \min\ \pi_2^{-1}(\ y^{\varphi(y)}\ )\ \right)^{(\varphi(y)^{-1}\varphi(y))\cdot\varphi(y^g)^{-1}g^{-1}}\right]^{G_y g} \\
&= \left(\left[\left(\ \min\ \pi_2^{-1}(\ y^{\varphi(y)}\ )\ \right)^{\varphi(y)^{-1}}\right]^{G_y}\right)^{g} \ = \ \mu(y)^g.
\end{aligned}$$

Note that $\varphi(y)^{-1}\varphi(y) = id$ and that $\varphi(y)\varphi(y^g)^{-1}g^{-1} \in G_y$. Thus μ satisfies the second property as well. ∎

Therefore, Algorithm 3.1.1 constructs a transversal for the G-orbits on Y, given a transversal for the G-orbits on X.

3.3 Example: Orderly Generation of Graphs

This section presents a first example for the techniques of Orderly Generation. Namely, we will discuss a very basic algorithm to generate graphs on a given number of vertices. The methods presented are very general, and therefore not optimized.

In practical applications, one would want to consider refinements of this algorithms which are then faster. The purpose of this section however is to show the basic framework of an orderly algorithm.

Example 3.3.1. Let X denote the class of all finite graphs of order 4 with the vertex set $V = \{1, 2, 3, 4\}$. Let $H_1 = (V, E_1) \in X$ be a graph where

$$E_1 \subseteq \mathcal{P}_2(V) = \Big\{\{1,2\}, \{1,3\}, \{1,4\}, \{2,3\}, \{2,4\}, \{3,4\}\Big\},$$

thus $|\mathcal{P}_2(V) = 6|$.

If $H_2(V, E_2)$ is another graph in $\mathcal{G}_4$, then H_1 is isomorphic to H_2 if and only if there exists $f \in Sym_{(4)}$ such that

$$\{i,j\} \in E_1 \Longleftrightarrow \{f(i), f(j)\} \in E_2 \;\; \forall\, i,j \in \{1,2,3,4\} \;\text{ and }\; i \neq j.$$

For instance, if $f = (\ 1 \ \ 3 \ \ 4\ ) \in Sym_{(4)}$, then f permutes the set of all edges as follows:

$$\begin{pmatrix} \{1,2\} & \{1,3\} & \{1,4\} & \{2,3\} & \{2,4\} & \{3,4\} \\ \{2,3\} & \{3,4\} & \{1,3\} & \{2,4\} & \{1,2\} & \{1,4\} \end{pmatrix}$$

Let $Sym^{[2]}_{(4)}$ be the group of permutation of $\mathcal{P}_2(V)$ obtained in this way from $Sym_{(4)}$. Since $Sym_{(4)}$ is a group of permutations, it follows that $Sym^{[2]}_{(4)}$ is also a group of permutations.

In particular, $Sym_{(4)}^{[2]}$ acts on $\mathcal{P}_2(V)$ by the meaning of unordered pairs action, i.e.

$$\mathcal{P}_2(V) \times Sym_{(4)}^{[2]} \rightarrow \mathcal{P}_2(V), \quad (\{i,j\},g) \mapsto \{i^g, j^g\},$$

for all $\{i,j\} \in \mathcal{P}_2(V)$ and for all $g \in Sym_{(4)}^{[2]}$. Therefore, the orbits of $Sym_{(4)}^{[2]}$ on $\mathcal{P}_i(\mathcal{P}_2(V))$ are in one-to-one correspondence with the orbits of $Sym_{(4)}$ on graphs on 4 vertices with i edges.

For $i = 0, 1, \ldots, 6$ let $X_i = \mathcal{P}_i(\mathcal{P}_2(V)) \in X$ be of size i (i.e. with i edges). Then, we always consider the action of $Sym_{(4)}^{[2]}$ on X_i and X_{i+1}, and the elementwise action of $Sym_{(4)}^{[2]}$ on the $Sym_{(4)}^{[2]}$-invariant relation $R_i \subseteq X_i \times X_{i+1}$ such that

$$(A,B) \in R_i \iff \{a,b\} \in A \text{ implies } \{a,b\} \in B.$$

In other words, $(A,B) \in R_i$ if and only if $A \subset B$. Therefore, starting from X_0, these group actions give rise to a sequence of the following form:

$$X_0 \overset{R_0}{\text{——}} X_1 \overset{R_1}{\text{——}} X_2 \overset{R_2}{\text{——}} \cdots\cdots \overset{R_5}{\text{——}} X_6. \tag{3.8}$$

In this way, the relation R_i can be represented by a $(|X_i| \times |X_{i+1}|)$ $\{0,1\}$-matrix M with $M = (m_{xy})$ (for $1 \leq x \leq |X_i|$ and $1 \leq y \leq |X_{i+1}|$) such that

$$m_{xy} = \begin{cases} 1 & \text{if } (x,y) \in R_i, \\ 0 & \text{otherwise.} \end{cases}$$

Figures 3.3.1, 3.3.2, and 3.3.3 show all graphs in X_2, all graphs in X_3, and a matrix M corresponding to R_2, respectively.

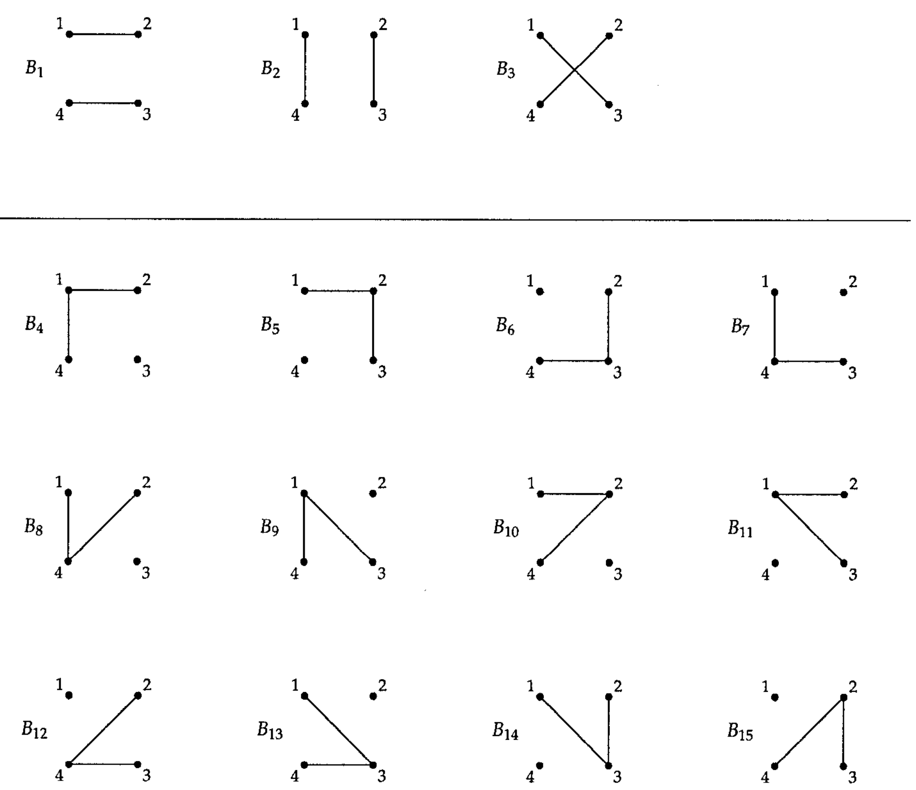


Figure 3.3.1: All elements in X_2

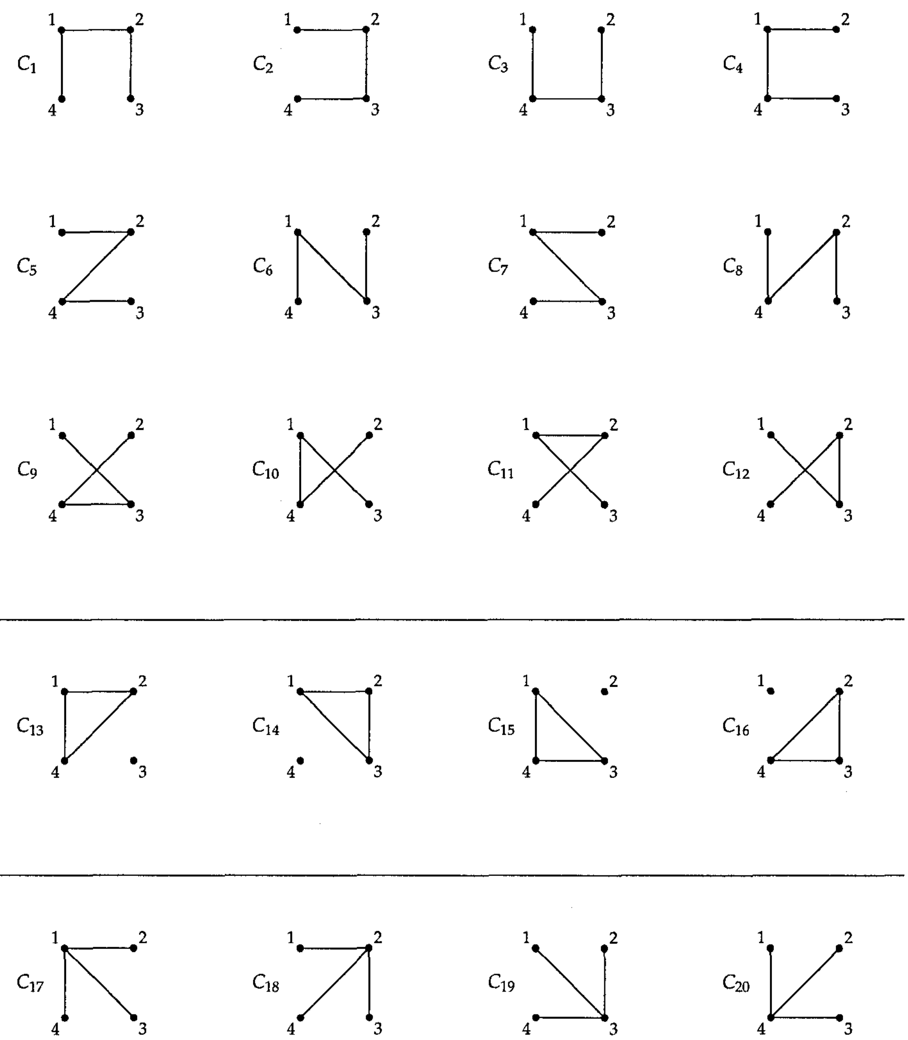


Figure 3.3.2: All elements in X_3

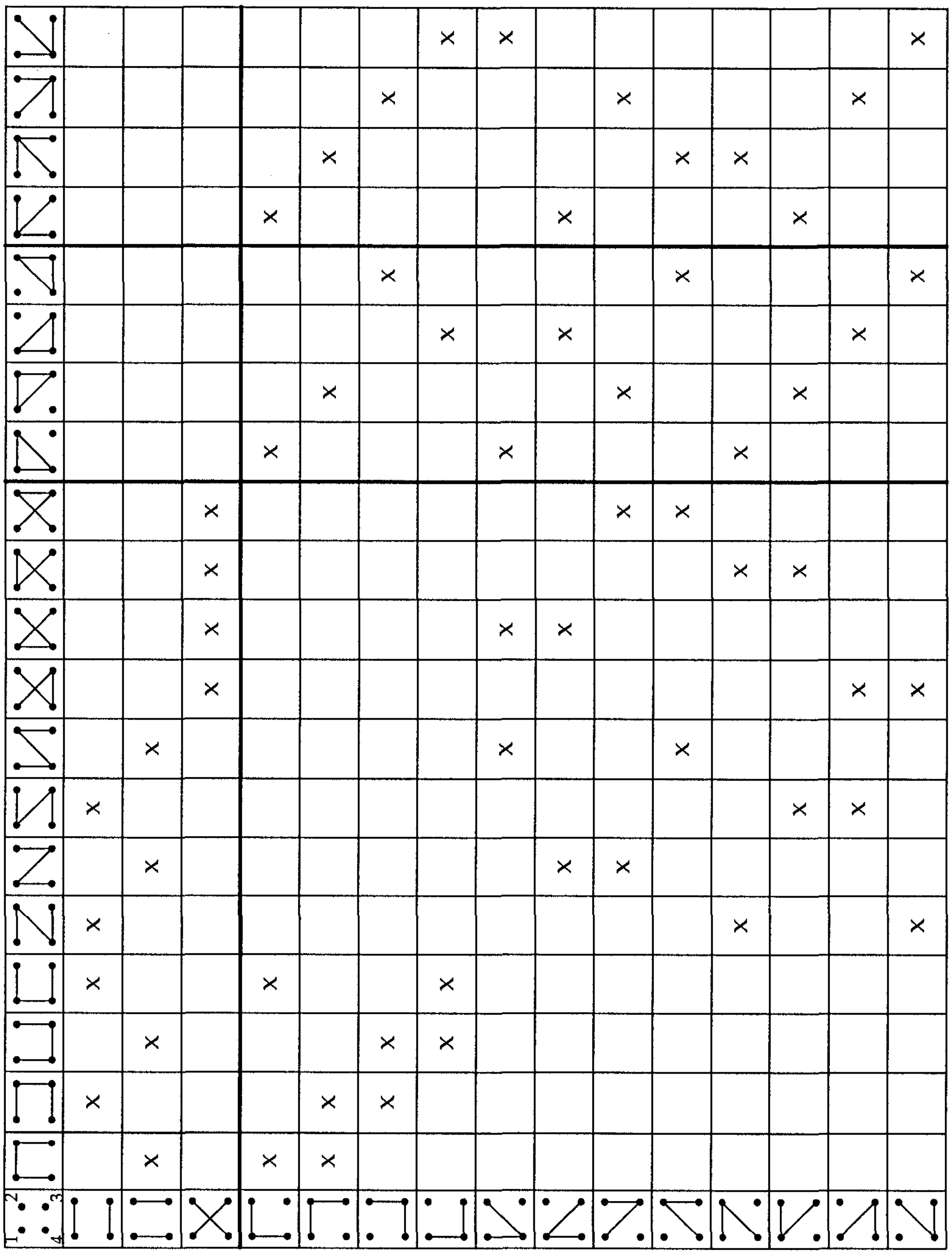


Figure 3.3.3: The matrix M which represents the relation $R_2 \subseteq X_2 \times X_3$.

For the sake of simplicity, we write G for the group $Sym^{[2]}_{(4)}$.

<ins>1^{st} problem</ins>: Starting from X_0, we only have the empty graph, which is the graph of order 4 and has no edges. Then, computing the extension set of this graph, we get 6 different graphs all of which are contained in the same G-orbit, see Figure 3.3.4.

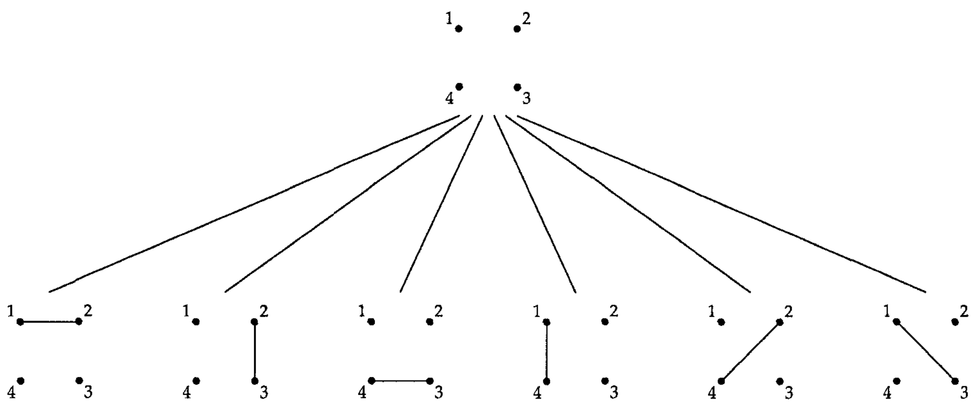


Figure 3.3.4: Extensions of the empty graph.

For that we consider the lifting step in our algorithm as in lines $(2-5)$ in Algorithm 3.1.1. Thus when we start from node $A_{0,1}$ given in Figure 3.3.6, we compute first the stabilizer of $A_{0,1}$ which is $G_{A_{0,1}} = G$ with order $|G_{A_{0,1}}| = 24$ and by the Orbit-Stabilizer Theorem we have six graphs contained in the same orbit, but we only consider one graph as in Figure 3.3.6.

Another non-trivial example is by considering the lifting orbits step for the node $A_{2,1}$ of Figure 3.3.6. Figure 3.3.5 displays the extension set of $A_{2,1}$.

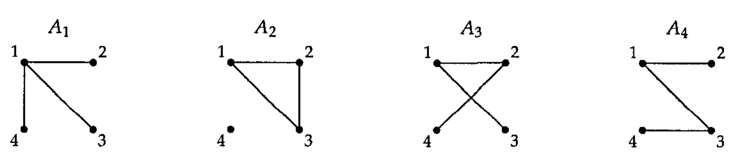


Figure 3.3.5: Extensions of node $A_{2,1}$ of Figure 3.3.6.

Clearly, $Aut(A_{2,1}) = \langle(2 \;\; 3)\rangle$. Thus, we skip A_4 because of the fact that

A_3 and A_4 are contained in the same $Aut(A_{2,1})$-orbit.

2nd problem: In level 2 in the poset of Figure 3.3.6, we have two graphs, namely $A_{2,1}$ and $A_{2,2}$, which both can be extended to the same G-orbit in level 3 of the poset, represented by the node $A_{3,3}$. To make sure that every orbit is visited exactly once in each level of the search tree, we use the μ function given in Lemma 3.2.1 so that we associate with every orbit in level 3 one orbit which it must be constructed from. Thus, we need to compute $\mu(A_{3,3})$ first and then check if we can accept either node $A_{2,1}$ or node $A_{2,2}$ from level 2.

To make things easier, we assume that the canonical labeling map maps a node to its lexicographic least form in its orbit. That is, given node $A_{3,3}$ of Figure 3.3.6, we compute its orbit under the action of G as in table 3.1. Then, the orbit representative is the lexicographical least element in the G-orbit of $A_{3,3}$. Table 3.1 has 4 columns. First, ordering represents the lexicographical ordering of the corresponding graph in the orbit. Second, an element $g \in G$ which maps $A_{3,3}$ to the corresponding graph. Third, $A_{3,3}^g$ is the g-image of $A_{3,3}$. Fourth, we write the edges set of the corresponding graph so that it is easy to see the lexicographical ordering on the graphs in the G-orbit of $A_{3,3}$. Note that, the entries in the table are ordered from least to greatest graphs in the orbit with respect to the lexicographical ordering.

Thus, $\varphi(A_{3,3}) = (id)$ is the canonical labeling map, and $\rho(A_{3,3}) := A_{3,3}$ is the canonical orbit representative.

Next we compute

$$\pi_2^{-1}(A_{3,3}) = \{(B_3, A_{3,3}) \ , \ (B_{10}, A_{3,3}) \ , \ (B_{11}, A_{3,3})\}$$

where B_3, B_{10}, and B_{11} are drawn in Figure 3.3.1. Clearly, then

$$\min \ \pi_2^{-1}(A_{3,3}) = (B_{11}, A_{3,3}).$$

Now we compute $\left(\min \ \pi_2^{-1}(A_{3,3})\right)^{\varphi(A_{3,3})^{-1}}$ which is going to be $(B_{11}, A_{3,3})$. Therefore,

$$\mu(A_{3,3}) = (B_{11}, A_{3,3})^{G_{A_{3,3}}} \ = \ \left\{(B_{11}, A_{3,3}), (B_{10}, A_{3,3})\right\},$$

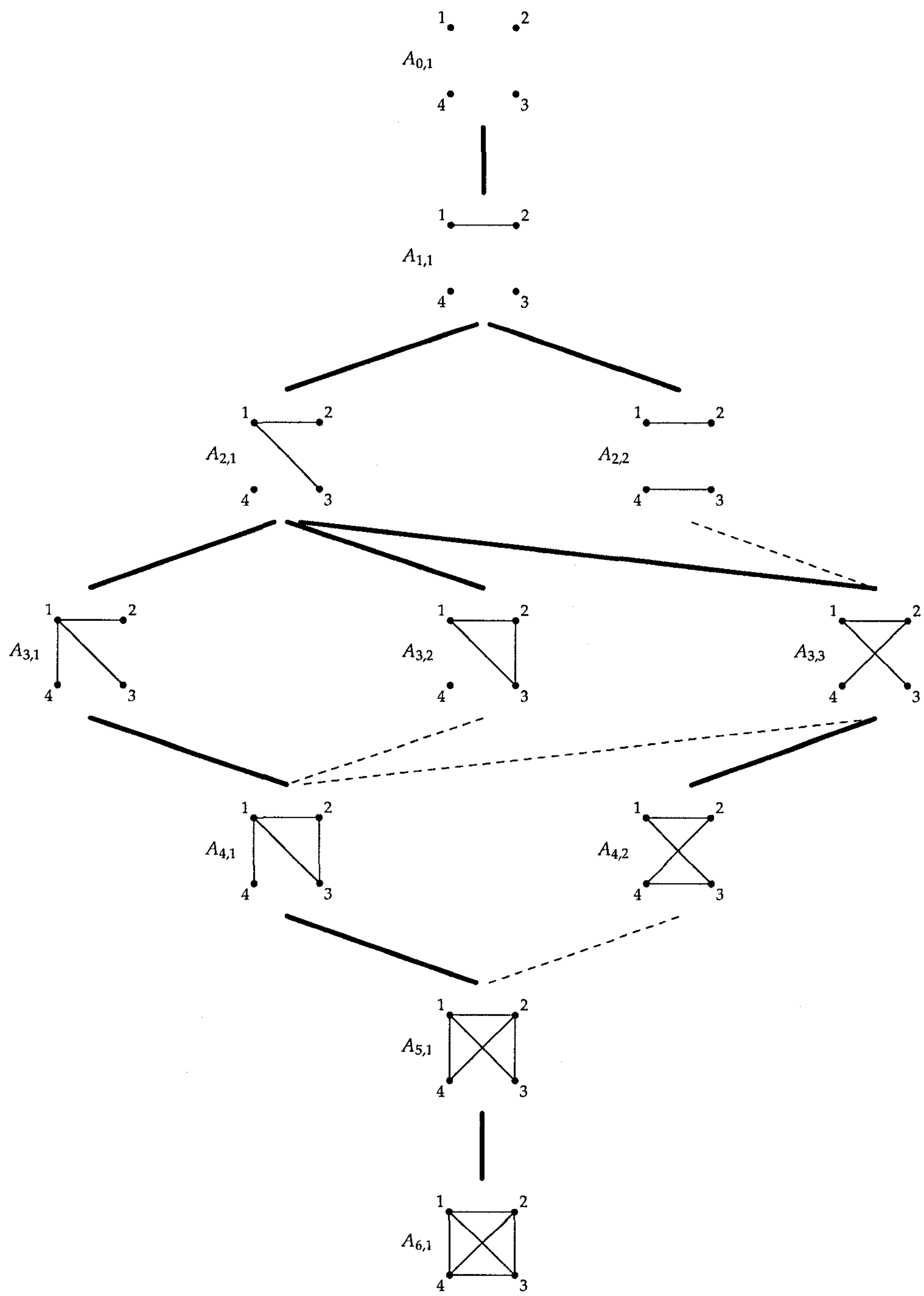
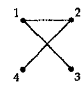
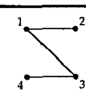
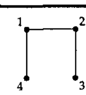
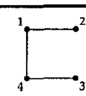
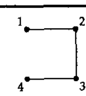
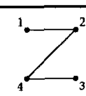
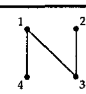
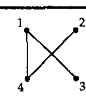
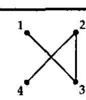
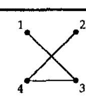
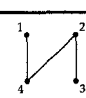
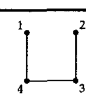


Figure 3.3.6: Search poset for cubic 4. Note that bolded lines mean that $(x, y) \in X_i \times X_{i+1}$, with $(x, y) \in \mu(y)$, for $i = 0, \ldots, 6$.

Table 3.1: The G-orbit of node $A_{3,3}$

ordering	$g \in G$	$A^g_{3,3}$	the edges set
1 ($A_{3,3}$ canonical)	(id)		$\{\{1,2\},\{1,3\},\{2,4\}\}$
2	(2 3)		$\{\{1,2\},\{1,3\},\{3,4\}\}$
3	(3 4)		$\{\{1,2\},\{1,4\},\{2,3\}\}$
4	(2 4 3)		$\{\{1,2\},\{1,4\},\{3,4\}\}$
5	(1 2 3)		$\{\{1,2\},\{2,3\},\{3,4\}\}$
6	(1 2 4 3)		$\{\{1,2\},\{2,4\},\{3,4\}\}$
7	(1 3 2)		$\{\{1,3\},\{1,4\},\{2,3\}\}$
8	(2 4)		$\{\{1,3\},\{1,4\},\{2,4\}\}$
9	(1 3)		$\{\{1,3\},\{2,3\},\{2,4\}\}$
10	(1 4) (2 3)		$\{\{1,3\},\{2,4\},\{3,4\}\}$
11	(1 4 3)		$\{\{1,4\},\{2,3\},\{2,4\}\}$
12	(1 4 2 3)		$\{\{1,4\},\{2,3\},\{3,4\}\}$

where $G_{A_{3,3}} = \langle (1\ 2)\ (3\ 4) \rangle$. Clearly then we accept the extension $(A_{2,1}, A_{3,3}) \in \mu(A_{3,3})$ and reject the other extension, which is $(A_{2,2}, A_{3,3})$.

Let G be generated by a generating set $S = \{s_1, s_2\}$, where

$$\begin{aligned} s_1 &= (\ 1\ \ 2\ ), \\ s_2 &= (\ 1\ \ 2\ \ 3\ \ 4\ ). \end{aligned}$$

A Schreier tree for the orbit $A_{3,3}^G$, see Table 3.1, is displayed in Figure 3.3.7.

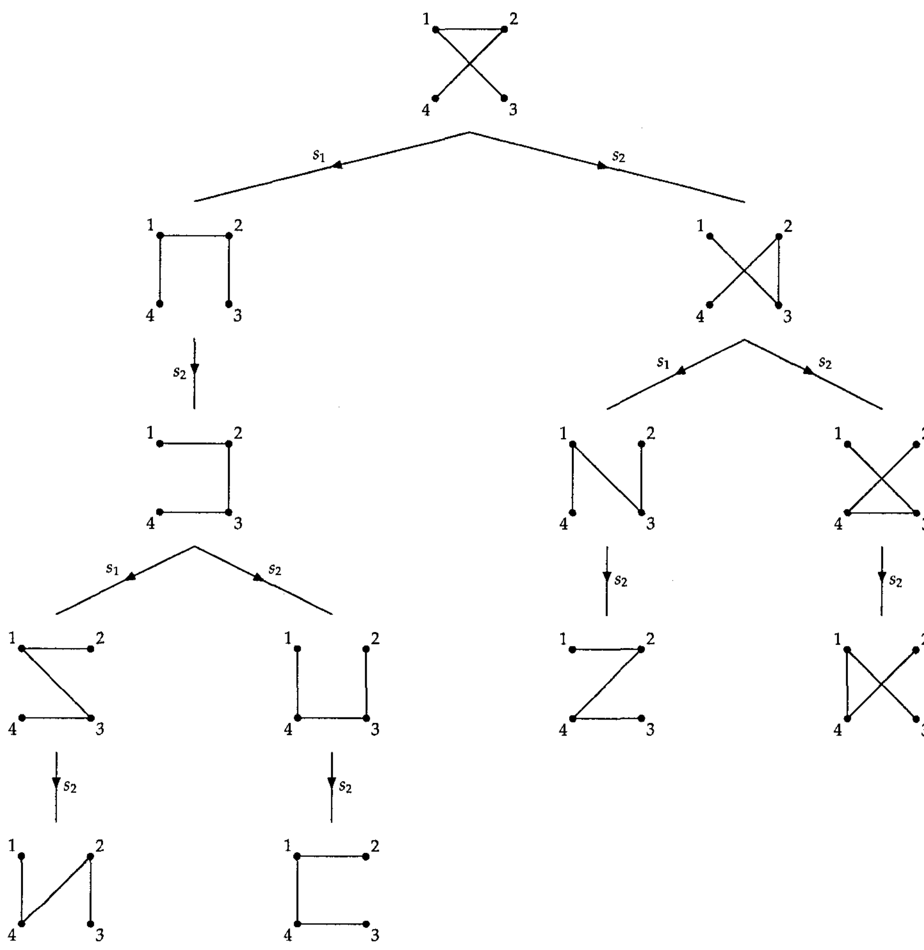


Figure 3.3.7: A Schreier tree for the G-orbit on $A_{3,1}$.

■

Chapter 4

Generation of Incidence Structures

In this chapter, we consider the problem of finding suitable group actions and relations between the respective sets so that the theory introduced in Chapter 3 can be applied.

Thus our goal is to apply the techniques discussed in Chapter 3 to any class of incidence structures for which an inductive construction process exists. Suppose that one is interested in constructing k-regular graphs on m vertices. Simply, it is not easy to do so since the class of k-regular graphs has no good inductive properties. Instead, the same goal can be achieved by considering an induction on larger classes that will be discussed in this chapter.

The construction procedure then gives rise to a backtrack search in a larger class. A **backtrack search** or simply **backtracking** is an algorithmic principle that emphasizes **"step-by-step try out all the possibilities"** in order to find all possible solutions to a given finite problem. For more reading about such a strategy, read [11, 24, 52, 55, 78].

4.1 The Relation, and Search Lattice, Search Poset, and Search Tree

Let X be a finite set and let $X_0, X_1, \ldots, X_m$ be disjoint subsets of X such that

$$X = \bigcup_{i=0}^{m} X_i.$$

Let G be a group that acts on X and on X_i for all $i = 0, 1, \ldots, m$. Let R_i be a G-invariant relation with $R_i \subseteq X_i \times X_{i+1}$. In particular, we say that $(A, B) \in R_i$ if $A \in X_i$ and $B \in X_{i+1}$ and that there exists a relation between A and B in some sense, for $i = 0, 1, \ldots, m-1$. We always require that $\pi_2(R_i) = X_{i+1}$, where $\pi_2 : R_i \to X_{i+1}$, defined by $(A, B) \mapsto B$.

In general, we consider the construction of the relation $R_0 \times R_1 \times \cdots \times R_{m-1}$ step by step over $X_0 \times X_1 \times \ldots X_m$.

The idea is to apply the techniques of Chapter 3 to the sequence of group actions of G on X_i for $i = 0, 1, \ldots, m$, and compute orbit transversal $\mathcal{T}_i = (G, X_i)$ for $= 0, 1, \ldots, m$.

Recall that if $(A, B) \in R_i \subseteq X_i \times X_{i+1}$, then we say that B is in the extension set of A, $\pi_1^{-1}(A)$, and that A is in the pre-image set of B, $\pi_2^{-1}(B)$.

For our convenience, we write $A \prec B$ for $A \in X_i$ and $B \in X_{i+1}$ instead of $(A, B) \in R_i$.

Consider a ranked lattice $(X, \prec)$ and a group G which acts on a finite set X, i.e.

$$(a \vee b)^g = a^g \vee b^g, \quad \text{and}$$
$$(a \wedge b)^g = a^g \wedge b^g,$$

for all $a, b \in X$.

The set X/G of G-orbits on X is a poset with respect to the relation "$\prec$", i.e.

$$G(A) \prec G(B) \iff \exists g \in G \text{ such that } A^g \prec B,$$

where $A, B \in X$, and $G(A), G(B) \in X/G$.

A spanning tree for the search poset $(X, \prec)$ can be established by using the μ-function described in Chapter 3. That is, each node in the tree has a unique ancestor. As a result some of the paths in Figure 3.3.6 would be accepted and some would be rejected depending on passing the μ test.

Let X_i be the i^{th} layer of X for some suitable i. The search tree T has as its nodes the G-orbits on X. The orbit $\{\emptyset\} = X_0$ is the root of T. There is an edge between two nodes $G(A)$ and $G(B)$ if $A \in X_i$, and $B \in X_{i+1}$ and

$$\text{there exists a } \ g \in G \ \text{ such that } \ (A^g, B) \in \mu(B) \in \pi_2^{-1}(B)/G_B. \tag{4.1}$$

Theorem 4.1.1. *Let G be a group acting on a ranked lattice $(X, \prec)$. The tree T whose nodes are the elements of the set X/G of G-orbits on X with root $\{\emptyset\}$ such that two elements $G(A)$ and $G(B)$ are connected by an edge only if Condition 4.1 satisfied is a spanning tree.*

Proof. To show that T is a spanning tree, we need to show that every node in T has an ancestor, and that every node in T has a unique ancestor.

First, consider $G(B)$ for $B \in X$. Assume that $B \in X_{i+1}$. Choose $A \in X_i$ such that $(A, B) \in \mu(B) \in \pi_2^{-1}(B)/G_B$. Then $G(A)$ is connected to $G(B)$ in the tree. By induction on i, we see that there is a path for the root of T to B. Thus, every node in T has an ancestor.

Next, assume that $A_1, A_2 \in X_i$ such that $(A_1^g, B) \in \mu(B)$ and $(A_2^h, B) \in \mu(B)$. Since $\mu(B)$ is a G_B-orbit on R_i (where $R_i \subseteq X_i \times X_{i+1}$ is a G-invariant relation as usual), there is an element $u \in G_B$ such that

$$(A_1^{gu}, B^u) \ = \ (A_1^{gu}, B) \ = (A_2^h, B).$$

Thus, $A_1^{gu} = A_2^h$, and A_1 is in the same G-orbit as A_2. Therefore, there is a unique ancestor for every node in T, and the proof is completed. □

4.2 A Class of k-Regular Graphs $\mathcal{C}^{m,k}$

In this section, we consider the problem of constructing k-regular graphs on m vertices. Note that this class has no good inductive properties and thus a larger class is considered in the following section.

Let V be a finite set of vertices. Denote by $\mathcal{C}^{m,k}$ the class of all k-regular graphs on m vertices.

Definition 4.2.1. *Let C be a k-regular graph. Then,*

$$supp(C) = \{v \in V \mid deg(v) > 0\}.$$

Each vertex in C has exactly k neighbors, i.e. every $v \in V$ is adjacent to k other distinct vertices in V. Thus, the number of edges is

$$\mid E \mid = n = \frac{m \cdot k}{2}.$$

A k-regular graph $C \in \mathcal{C}^{m,k}$ is a pair (V, E) with vertex set

$$V = \{v_0, v_1, \ldots, v_{m-1}\},$$

and an edge set

$$E = \{E_0, E_1, \ldots, E_{n-1}\} \subseteq \mathcal{P}_2(V).$$

The class of graphs $\mathcal{C}^{m,k}$ does not has good inductive properties. For instance, it is not easy to create $\mathcal{C}^{8,3}$ from $\mathcal{C}^{6,3}$. This is because the induced subgraph on 6 vertices of a graph in $\mathcal{C}^{8,3}$ is never in $\mathcal{C}^{6,3}$, for instance. See Figures 4.2.1 and 4.2.2.

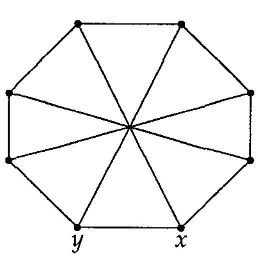


Figure 4.2.1: A graph in $\mathcal{C}^{8,3}$.

$\xrightarrow{\text{removing } x \text{ and } y}$

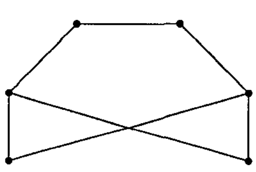


Figure 4.2.2: An induced subgraph on 6 vertices.

Let us consider the following procedure. Let C be a graph in $\mathcal{C}^{m,k}$. Pick a vertex $x_1 \in V(C)$ and delete the incident edges $x_1 \sim y$ for $y \in N(x_1)$. For each vertex $y \in V(C) \backslash \{x_1\}$, keep a counter that remembers how every edges incident to y have been removed. Repeat the procedure by picking another vertex $x_2 \neq x_1$ and remove all incident edges $x_2 \sim y$ for $y \in N(x_2)$. For each vertex $y \in V(C) \backslash \{x_1, x_2\}$, have the counter reflect how many edges have been removed altogether. We represent such a counter between parentheses "( )" as in the following example.

Example 4.2.1. Consider the (right) cubic (3-regular) graph of Figure 2.1.2 on page 9. Picking the vertices y, w, x, v, z, u in order, we arrive at the sequence of graphs of Figure 4.2.3.

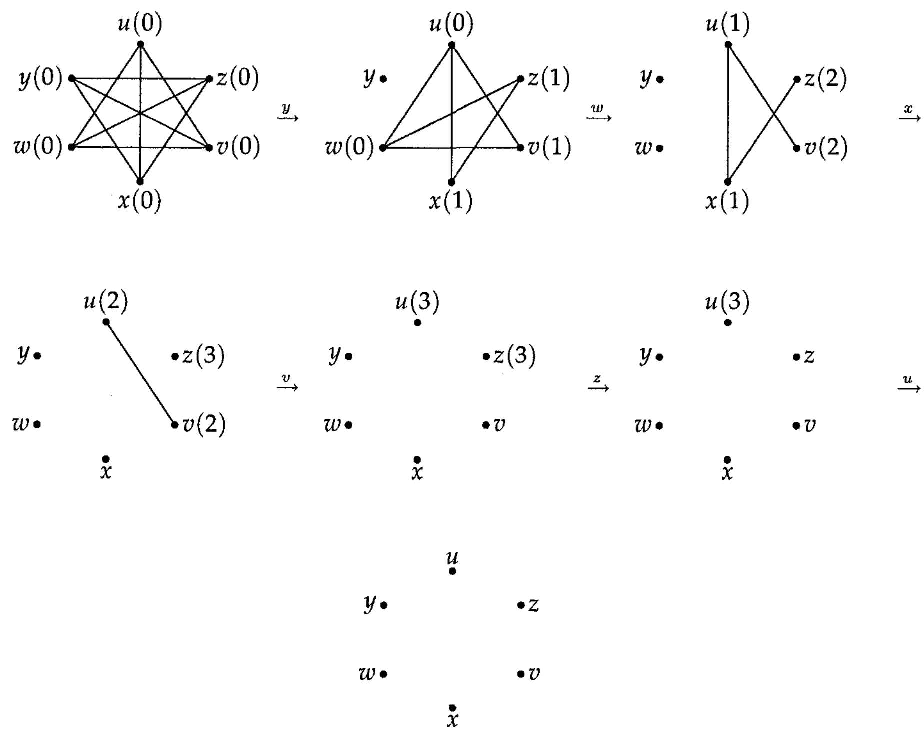


Figure 4.2.3: A sequence of resulting graphs by removing edges as in the above procedure.

■

Note that the idea of the prevoius procedure gives rise to a vertex by vertex generation strategy which corresponds to row by row generation strategy with respect to incidence matrices.

For our convenience, we distinguish the counter **values** (the number of removed edges) by drawing flags instead as in Figure 4.2.4

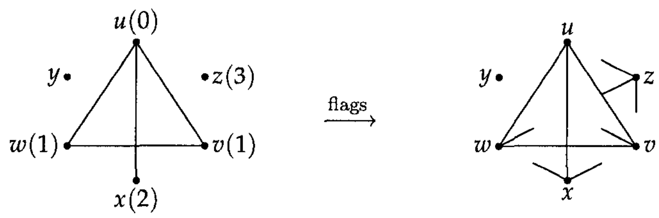


Figure 4.2.4: Representing the number of removed edges in different ways.

4.3 A Class of Flag Graphs $\mathcal{F}^m$

As mentioned before, the class $\mathcal{C}^{m,k}$ has no good inductive properties. Therefore, there is no an easy way to construct regular graphs by considering the class $\mathcal{C}^{m,k}$.

Consider the following more general class of graphs on m vertices, which we call **flag-graphs**, denoted by $\mathcal{F}^m$.

Definition 4.3.1. *A* ***flag-graph*** *C is a triple (V, E, F) where $V = \{v_0, v_1, \dots, v_{m-1}\}$ with m vertices, $E = \{E_0, E_1, \dots, E_{n-1}\} \subseteq \mathcal{P}_2(V)$ with n edges, and a function $F : V \longrightarrow \mathbb{N}$. The function F indicates the number of flags (the counter of removed edges) for every vertex $v \in V$.*

Definition 4.3.2. *Let $C = (V, E, F) \in \mathcal{F}^m$. Then, for $v \in V$, we have*

$$deg(v) = deg_E(v) \ + \ F(v), \ and$$
$$supp(C) = \{v \in V \mid deg(v) > 0\}.$$

where $deg_E(v) = \big|\{E_j \in E \mid v \in E_j \text{ for all } 0 \leq j \leq n-1\}\big|$.

Example 4.3.1. Let $C \in \mathcal{F}^6$ such that $C = (\,\{0,1,\ldots,5\}, \{\{0,1\}\}, (2,2,3,0,0,0)\,)$. Then the flag graph C is displayed in Figure 4.3.1.

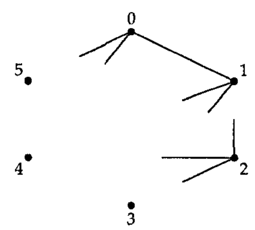


Figure 4.3.1: A flag-graph C.

■

In particular, we are interested in the following subclass of flag graphs on m vertices. If

$$\mathcal{F}_l^{m,k} = \{C \in \mathcal{F}^m \;\big|\; |\text{supp}(C)| = l,\ \deg(v) = k \text{ for all } v \in \text{supp}(C)\},$$

then the class

$$\mathcal{F}^{m,k} = \bigcup_{l=0}^{m} \mathcal{F}_l^{m,k}$$

is the class (with good inductive properties) that we are interested in. For instance, the flag graph C of Example 4.3.1 is in $\mathcal{F}_3^{6,3} \in \mathcal{F}^{6,3}$.

Note that, the class $\mathcal{F}_m^{m,k}$ is the class of k-regular graphs $\mathcal{C}^{m,k}$. Therefore, a construction of graphs in $\mathcal{C}^{m,k}$ can be done via an induction on l in the class $\mathcal{F}^{m,k} = \bigcup_{l=0}^{m} \mathcal{F}_l^{m,k}$ which is clearly has good inductive properties. In particular, we consider $\mathcal{F}^{m,k}$ the **search space** in our backtrack search where $\mathcal{F}_m^{m,k}$ is the **target space.**

Definition 4.3.3. *Let $Sym_{(m)}$ act on $\mathcal{F}^{m,k}$. If $C = (V, E, F) \in \mathcal{F}^{m,k}$ and for any $g \in Sym_{(m)}$, then we have*

$$C^g = (V, E, F)^g = (V, E^g, F^g),$$

with $E^g = \{E_0^g, E_1^g, \ldots, E_{n-1}^g\}$, where $\{v_i, v_j\}^g = \{v_i^g, v_j^g\}$ for $v_i, v_j \in V$, $0 \le i, j \le m-1$, and $i \neq j$. Moreover, $F^g = F \circ g^{-1}$, i.e. for $v, u \in V$ if $v^g = u$, then we have $F^g(u) = F^g(v^g) = F((v^g)^{g^{-1}}) = F(v)$.

Example 4.3.2. Following Example 4.3.1, consider the flag graph $C^g = (V, E^g, F^g)$ of Figure 4.3.2 with $g = (\ 1\ \ 2\ )\ (\ 3\ \ 4\ ) \in Sym_{(6)}$, then we have

$$(\{0, 1, \ldots, 5\}, \{\{0, 1\}\}, (2, 2, 3, 0, 0, 0)\ )^g =$$
$$(\{0, 1, \ldots, 5\}, \{\{0, 2\}\}, (2, 3, 2, 0, 0, 0)\ )$$

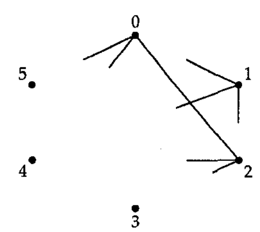


Figure 4.3.2: A drawing for the flag-graph C^g.

■

4.4 The Ordering on $\mathcal{F}^{m,k}$

In this section, we discuss the ordering in the class $\mathcal{F}^{m,k}$. Note that this ordering is different from the totally ordering discussed in Chapter 3.

Definition 4.4.1. *Let $C_1 = (V, E_1, F_1)$ and $C_2 = (V, E_2, F_2)$ be two flag graphs in $\mathcal{F}^{m,k}$. Then, we say that $C_1 \prec C_2$ if and only if*

1. *$supp(C_1) \subseteq supp(C_2)$, and*
2. *the induced subgraph of (V, E_2) on $supp(C_1)$ is equal to (V, E_1).*

If $C_1 = (V, E_1, F_1)$ and $C_2 = (V, E_2, F_2)$ are in $\mathcal{F}^{m,k}$, then clearly an edge $\{x, y\} \in E_1$ implies that $\{x, y\} \in \text{supp}(C_1)$, and hence the edge $\{x, y\} \in (V, E_1)$. But, (V, E_1) is the induced subgraph of (V, E_2) on $\text{supp}(C_1)$. Therefore, the edge $\{x, y\} \in$ induced subgraph of (V, E_2) on $\text{supp}(C_1)$, and hence $\{x, y\} \in E_2$. Therefore, Condition 2 of the above definition implies that $E_1 \subseteq E_2$.

Example 4.4.1. Figure 4.4.1 displays two flag graphs A and B that are in $\mathcal{F}_4^{6,3}$ and $\mathcal{F}_5^{6,3}$, respectively, with $A \prec B$, where Figure 4.4.2 displays two flag graphs C and D that are in $\mathcal{F}_2^{6,3}$ and $\mathcal{F}_3^{6,3}$, respectively, such that $C \not\prec D$. Clearly, the induced subgraph of $(V(D), E(D))$ on $\text{supp}(C) = \{0, 1\}$ is not equal to $(V(C), E(C))$.

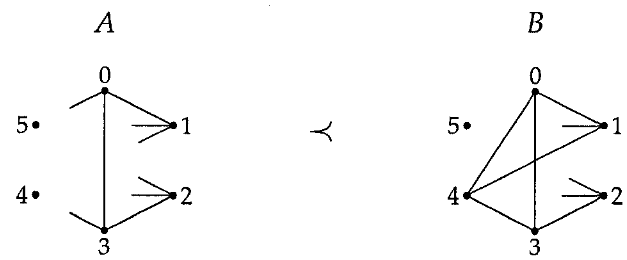


Figure 4.4.1: Graphs $A \in \mathcal{F}_4^{6,3}$ and $B \in \mathcal{F}_5^{6,3}$ such that $A \prec B$.

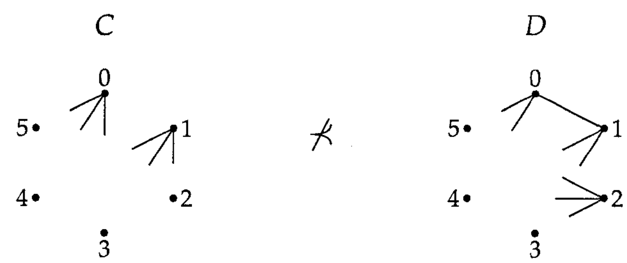


Figure 4.4.2: Graphs $C \in \mathcal{F}_2^{6,3}$ and $D \in \mathcal{F}_3^{6,3}$ such that $C \not\prec D$.

■

A more general concepts of extensions with respect to G action over the class $\mathcal{F}_l^{m,k}$ for $l = 0, 1, \ldots, m$ follows.

Definition 4.4.2. *Let G be a group acting on $\mathcal{F}_l^{m,k}$ and $\mathcal{F}_{l+1}^{m,k}$, for $l = 0, 1, \ldots, m-1$. Then, $G(A) \prec G(B)$ for $G(A) \in \mathcal{F}_l^{m,k}/G$, and $G(B) \in \mathcal{F}_{l+1}^{m,k}/G$, if there exists a $g \in G$ such that $A^g \prec B$.*

In general, we consider the construction of the relation $R_0 \times R_1 \times \cdots \times R_{m-1}$ step by step over $\mathcal{F}_0^{m,k} \times \mathcal{F}_1^{m,k} \times \ldots \mathcal{F}_m^{m,k}$. This can be done by applying the techniques of Chapter 3 to the sequence of group actions of $G = Sym_m$ on

$\mathcal{F}_l^{m,k}$ for $l = 0, 1, \ldots, m$, and compute orbit transversal $\mathcal{T}_l = (G, \mathcal{F}_l^{m,k})$ for $l = 0, 1, \ldots, m$.

Example 4.4.2. Consider the problem of constructing $\mathcal{F}_6^{6,3}$ (cubic graphs of order 6) with the group action of $G = Sym_6$. We start with $\mathcal{T}_0 = \mathcal{T}(G, \mathcal{F}_0^{6,3}) = \{A_{0,1}\}$ as given in Figure 4.4.3.

Because, G is transitive on vertices, $\mathcal{T}_1 = \{A_{1,1}\}$. Now, the point stabilizer of G is still transitive on the remaining points, therefore there are two orbits of graphs in $\mathcal{F}_2^{6,3}$, namely $\mathcal{T}_2 = \{A_{2,1}, A_{2,2}\}$.

The automorphism group of $A_{2,1}$ and $A_{2,2}$, denoted by $Aut(A_{2,1})$ and $Aut(A_{2,2})$, respectively, is $Sym_2 \times Sym_4$. Then, for $\mathcal{F}_3^{6,3}$, we distinguish the cases of adding one more vertex to $A \in \mathcal{T}_2$ as follows.

The third vertex might be joined to both, only one, or none of the vertices in the graphs in $\mathcal{T}_2$. Therefore, we get 4 orbits of G on $\mathcal{F}_3^{6,3}$, namely

$$\mathcal{T}_3 = \{A_{3,1}, A_{3,2}, A_{3,3}, A_{3,4}\}.$$

Here, we note that $A_{3,2}$ is an extension of both $A_{2,1}$ and $A_{2,2}$. Namely, we can think of $A_{3,2}$ as an extension of $A_{2,1}$ on vertices $\{0, 2\}$ by vertex $\{1\}$, and at the same time $A_{3,2}$ is an extension of $A_{2,2}$ on $\{0, 1\}$ by vertex $\{2\}$. Similarly, $A_{3,3}$ is an extension of both elements in $\mathcal{T}_2$.

In order to reduce the isomorphic duplicates, Theorem 3.1.4 on page 59 is applied. However, Theorem 3.1.4 relies on the existence of a μ function which can be realized using the algorithms described in Chapter 6. Nevertheless, we show the resulting search poset in Figure 4.4.3, where the layers correspond to the G-orbits on $\mathcal{F}_l^{6,3}$ for all $l = 0, \ldots, 6$.

■

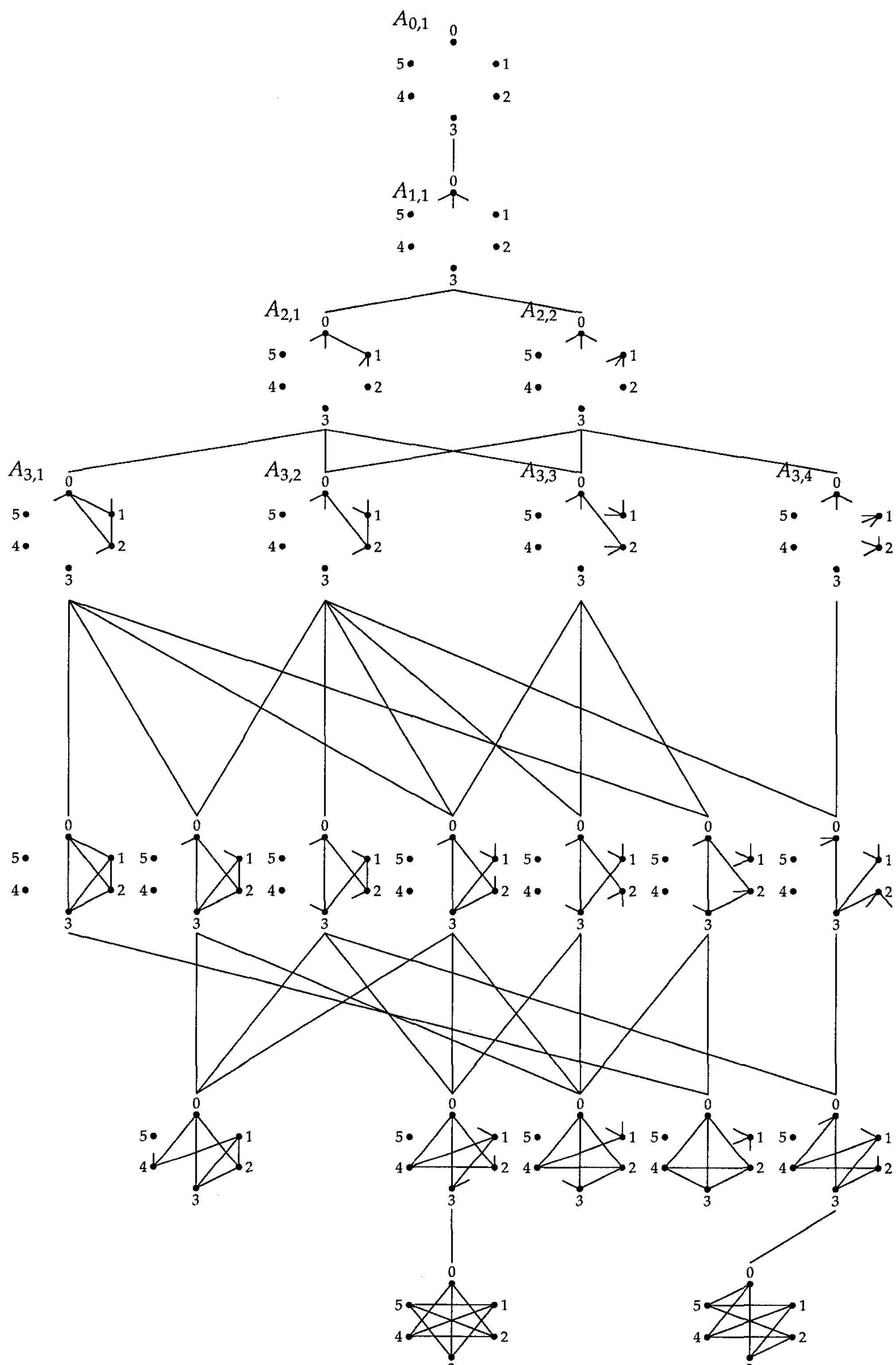


Figure 4.4.3: The search poset for $\mathcal{F}^{6,3}$.

4.5 A Class of $\{0,1\}$-matrices $\mathcal{D}^{m,k}$

This section considers a calss of $\{0,1\}$-matrices that corresponds to the class of flag graphs. Recall that $M_{m,n}$ denote the class of $m \times n$ $\{0,1\}$ matrices. In this section, we consider a class of incidence matrices denoted by $\mathcal{D}_l^{m,k}$ that correspond to the class of flag graphs $\mathcal{F}^{m,k} = \bigcup_{l=0}^{m} \mathcal{F}_l^{m,k}$. For that we consider the following three classes of $\{0,1\}$ matrices in $M_{m,n}$ where $n = \frac{mk}{2}$. For $l = 0, 1, \ldots, m$, let

$$\mathcal{A}_l^{m,k} = \left\{ A \in M_{m,n}^{(l)} \middle| \text{ row-sum}_i(A) = k \text{ for } i \in \text{Rowsupp}(A) \right\},$$
$$\mathcal{B}^{m,k} = \left\{ A \in M_{m,n} \middle| \text{ col-sum}_j(A) \le 2 \text{ for } 0 \le j \le n-1 \right\},$$
$$\mathcal{E}^{m,k} = \left\{ A \in M_{m,n} \middle| \ |\text{row}_i(A) \cap \text{row}_j(A)| \le 1 \text{ for all } 0 \le i, j \le m-1 \text{ and } i \ne j \right\},$$

where $A \in M_{m,n}^{(l)} = \{A \in M_{m,n} \mid |\text{Rowsupp}(A)| = l\}$, as defined in 2.5 on page 50.

Therefore, we consider the following class of $\{0,1\}$-matrices contained in all of the previous three classes. For $l = 0, 1, \ldots, m$, let

$$\mathcal{D}_l^{m,k} = \left\{ A \in M_{m,n} \middle| A \in \mathcal{A}_l^{m,k} \cap \mathcal{B}^{m,k} \cap \mathcal{E}^{m,k} \right\}.$$

In particular we consider the search space $\mathcal{D}^{m,k} = \bigcup_{l=0}^{m} \mathcal{D}_l^{m,k}$, where the search target is $\mathcal{D}_m^{m,k}$. We say that $A \prec B$ for $A = (a_{i,j}) \in \mathcal{D}_l^{m,k}$ and $B = (b_{i,j}) \in \mathcal{D}_{l+1}^{m,k}$, for $0 \le i \le m-1$ and $0 \le j \le n$, if $a_{i,j} = 1$ implies $b_{i,j} = 1$.

Let $G = Sym_{(m)} \times Sym_{(n)}$ act on $\mathcal{D}^{m,k}$, and $H = Sym_{(m)}$ act on $\mathcal{F}^{m,k}$. If $A \in \mathcal{D}^{m,k}$ is any incidence matrix corresponding to $C \in \mathcal{F}^{m,k}$, then permuting rows of A is equivalent to permuting vertices of C, and permuting columns of A is equivalent to reordering edges and flags in C. Thus, C_1 and C_2 in $\mathcal{F}^{m,k}$ are contained in the same H-orbit if and only if A_1 is equivalent to A_2 under the action of G, for any two incidence matrices A_1 and A_2 corresponding to C_1 and C_2, respectively. Therefore, the following theorem arises. See [48, 52, 61].

Theorem 4.5.1. *Let $G = Sym_{(m)} \times Sym_{(n)}$ be a group that acts on $\mathcal{D}^{m,k}$, and let $H = Sym_{(m)}$ act on $\mathcal{F}^{m,k}$. Then, there is a one-to-one correspondence between the following sets of orbits:*

- $\mathcal{F}^{m,k}/H$, *the set of H-orbits on* $\mathcal{F}^{m,k}$,
- $\mathcal{D}^{m,k}/G$, *the set of G-orbits on* $\mathcal{D}^{m,k}$.

Example 4.5.1. Given an incidence matrix $A_1 \in \mathcal{D}_4^{6,3}$ as below

$A_1 =$

	e_0	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8
v_0	x	x	x						
v_1	x			x	x				
v_2		x		x		x			
v_3			x				x	x	
v_4									
v_5									

One can describe A_1 in terms of the flag-graph $C_1 \in \mathcal{F}_4^{6,3}$ of Figure 4.5.1.

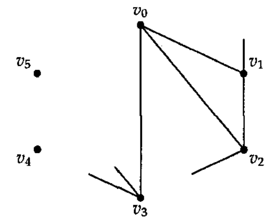


Figure 4.5.1: A flag-graph corresponds to $A_1 \in \mathcal{D}_4^{6,3}$.

Another example could be the incidence matrix $A_2 \in \mathcal{D}_6^{6,3}$, where

$A_2 =$

	e_0	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8
v_0	x	x	x						
v_1	x			x	x				
v_2		x		x		x			
v_3			x				x	x	
v_4					x		x		x
v_5						x		x	x

which can be represented by the flag-graph $C_2 \in \mathcal{F}_6^{6,3}$ of Figure 4.5.2.

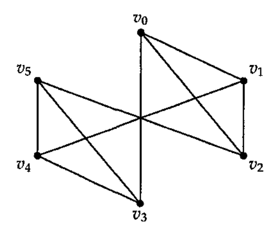


Figure 4.5.2: A flag-graph corresponds to $A_2 \in \mathcal{D}_6^{(6,3)}$.

In this case we see that A_2 is an incidence matrix describing a cubic graph of order 6, and thus C_2 is a cubic graph.

■

Incidence matrices that are in $\mathcal{D}_m^{m,k}$ are called **feasible solution**, while incidence matrices in $\mathcal{D}_l^{m,k}$ are called **partial solution** for $l = 0, 1, \ldots, m-1$.

A search poset in terms of incidence matrices in $\mathcal{D}^{m,k}$ that corresponds to the search poset of Figure 4.4.3 is displayed in Figure 4.5.3.

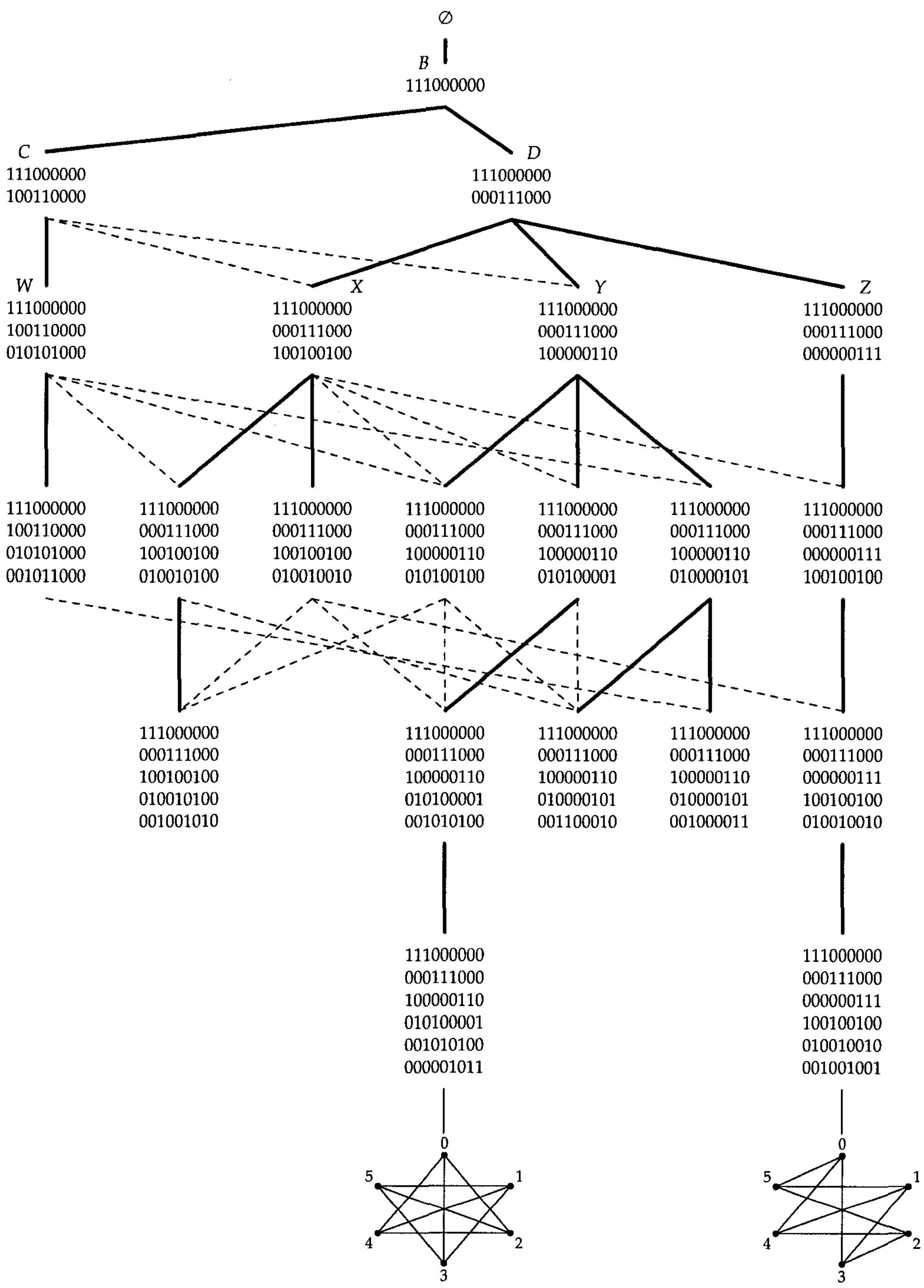


Figure 4.5.3: The search poset of Figure 4.4.3 in terms of incidence matrices.

4.6 Example: Generation of k-Regular Graphs

In this section, we discuss an example of the generation strategy applied to the class of k-regular graphs $\mathcal{F}_m^{m,k}$ by considering an induction on l in the class of incidence matrices $\mathcal{D}_l^{m,k}$ for $l = 0, 1 \ldots, m$. Then with the help of the techniques discussed in Chapter 3, we can construct orbits transversal $\mathcal{T}(G, \mathcal{D}_l^{m,k})$ where G is a group acting on $\mathcal{D}^{m,k}$ and for all $l = 0, 1, \ldots, m$. Note that our target is $\mathcal{T}(G, \mathcal{D}_m^{m,k})$.

Let $n := \frac{mk}{2}$. Let $G = Sym_{(m)} \times Sym_{(n)}$ act on $\mathcal{D}_l^{m,k}$ and $\mathcal{D}_{l+1}^{m,k}$ and let G act coordinatewise on $\mathcal{D}_l^{m,k} \times \mathcal{D}_{l+1}^{m,k}$. Let $R_l \subseteq \mathcal{D}_l^{m,k} \times \mathcal{D}_{l+1}^{m,k}$ be a G-invariant relation such that $R_l = \{(A, B) \in \mathcal{D}_l^{m,k} \times \mathcal{D}_{l+1}^{m,k} \mid A \prec B\}$ for $l = 0, 1, \ldots, m-1$.

The lifting orbits step says that for $A \in \mathcal{T}(G, \mathcal{D}_l^{m,k})$, for $l = 0, 1, \ldots, m-1$, we compute the correspondence extension set $\pi_1^{-1}(A)$ and an automorphism group $Aut(A)$ which is needed to compute a transversal $\mathcal{T}(Aut(A), \pi_1^{-1}(A))$ for every $A \in \mathcal{T}(G, \mathcal{D}_l^{m,k})$. Thus a transversal for the G on R_l is constructed for $l = 0, 1, \ldots, m-1$.

Then, by the projecting orbits step, we add B to $\mathcal{T}(G, \mathcal{D}_{l+1}^{m,k})$ only if $(A, B) \in \mathcal{T}(Aut(A), \pi_1^{-1}(A)) \cap \mu(B)$, where $A \in \mathcal{D}_l^{m,k}$ and the μ function must satisfies the conditions in Definition 3.1.3.

The following example illustrates the two steps between two adjacent layers (levels) in the search poset $(P, \preceq)$ given above in Figure 4.5.3. That is, $m = 6, k = 3$, and $\frac{mk}{2} = 9$.

Example 4.6.1. Consider the poset $(P, \preceq)$ of Figure 4.5.3. Given a group $G = Sym_{(6)} \times Sym_{(9)}$ acting on $\mathcal{D}^{6,3}$, assume that we have an orbit transversal in level 2 with 2 nodes in the transversal. We write $\mathcal{T}(G, \mathcal{D}_2^{6,3}) = \{C, D\}$ with C and D are two matrices given in the search poset P.

The first step in the algorithm is to extend each node in the transversal in every possible way. So we start with the matrix $C \in \mathcal{D}_2^{6,3}$ and first compute

$$\pi_1^{-1}(C) = \big\{(C, C+(E_i \otimes v)) \in R_2 \big| v \in M_{1,n} \text{ and } i \in S \subseteq \{0, \ldots, 5\} \text{ with } |S| = 3\big\}.$$

Then using G_C-rejection in $\pi_1^{-1}(C)$, we can compute

$$\mathcal{T}(G_C, \pi_1^{-1}(C)) = \{(C, C+(E_{i_1} \otimes v_1)), (C, C+(E_{i_2} \otimes v_2)), (C, C+(E_{i_3} \otimes v_3))\},$$

where $i_1, i_2, i_3 \in S$, and

$$v_1 = 010101000$$
$$v_2 = 010001100$$
$$v_3 = 000001110$$

Doing the same thing for $D \in \mathcal{D}_2^{6,3}$, we get

$$\mathcal{T}(G_D, \pi_1^{-1}(D)) = \{(D, D+(E_{j_1} \otimes w_1)), (D, D+(E_{j_2} \otimes w_2)), (D, D+(E_{j_3} \otimes w_3))\},$$

where $j_1, j_2, j_3 \in S$, and

$$w_1 = 100100100$$
$$w_2 = 100000110$$
$$w_3 = 000000111$$

Thus, we can form an orbit transversal for the action of G on R_2 as described above by computing the following set:

$$\mathcal{T}(G, R_l) = \mathcal{T}(G_C, \pi_1^{-1}(C)) \cup \mathcal{T}(G_D, \pi_1^{-1}(D))$$

Thus, the following set

$$\Big\{(C, C + (E_{i_1} \otimes v_1)),\ (C, C + (E_{i_2} \otimes v_2)),\ (C, C + (E_{i_3} \otimes v_3)),$$
$$(D, D + (E_{j_1} \otimes w_1)),\ (D, D + (E_{j_2} \otimes w_2)), (D, D + (E_{j_3} \otimes w_3))\Big\}$$

is an orbit transversal for the action of G on R_2, denoted by $\mathcal{T}(G, R_2)$.

Next, with the help of a given μ function, we construct a transversal

$$\mathcal{T}(G,\mathcal{D}_3^{6,3}) = \{C+(E_{i_1}\otimes v_1), D+(E_{j_1}\otimes w_1), D+(E_{j_2}\otimes w_2), D+(E_{j_3}\otimes w_3)\}$$

Note that if $(\alpha_1,\alpha_2),(\beta_1,\beta_2)\in G$ where

$$(\alpha_1,\alpha_2) = \Big((\ 0\ \ 2\ )\ ,\ (\ 0\ \ 3\ \ 5\ \ 1\ )\ (\ 2\ \ 6\ ) \Big)$$
$$(\beta_1,\beta_2) = \Big((\ 1\ \ 2\ )\ ,\ (\ 3\ \ 6\ )\ (\ 4\ \ 7\ ) \Big)$$

where α_1 and β_1 are row permutations and α_2 and β_2 are column permutations, then

$$(C+(E_{i_2}\otimes v_2))^{(\alpha_1,\alpha_2)} = D+(E_{j_1}\otimes w_1), \quad \text{and}$$
$$(C+(E_{i_3}\otimes v_3))^{(\beta_1,\beta_2)} = D+(E_{j_2}\otimes w_2).$$

Thus, $C+(E_{i_2}\otimes v_2)$ and $C+(E_{i_3}\otimes v_3)$ will not be added to $\mathcal{T}(G,\mathcal{D}_3^{6,3})$. It can be seen in Figure 4.5.3, that the poset P has four different orbits in level 3 represented by four nodes, namely nodes W, X, Y, and Z.

By considering only accepted nodes in the μ test, a spanning tree is resulted and displayed in Figure4.6.1.

■

Note that even though the defined μ function in Lemma 3.2.1 works, it is not efficient since then one need to do expensive computations involving orbit computations and testing minimality. Chapter 6.2 discuss in detail the implementation techniques for a suitable defined μ function so that we can avoid such expensive computations.

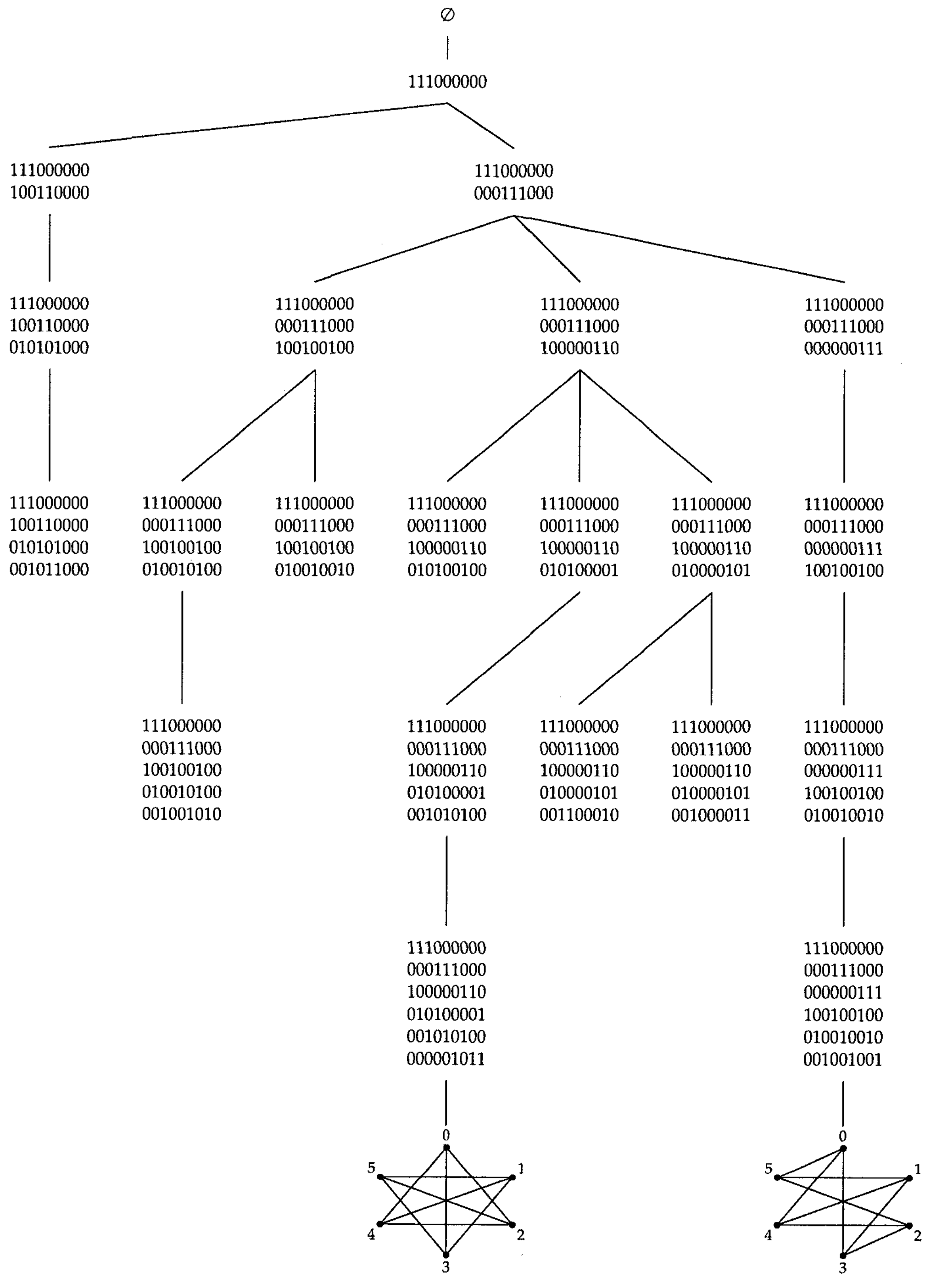


Figure 4.6.1: The spanning tree for the poset of Figure 4.5.3.

4.7 Regular Graphs With Given Girth

In this section, we discuss the generation of a given k-regular graph with a given girth g. Let $\mathcal{X} = (V, E) \in \mathcal{C}^{m,k}$ be a k-regular graph of order m and size n such that $V = \{v_0, v_1, \ldots, v_{m-1}\}$ and $E = \{e_0, e_1, \ldots, e_{n-1}\}$. Let $A \in \mathcal{D}_m^{m,k}$ be an incidence matrix that corresponds to $\mathcal{X}$.

Simply, the girth of any graph is computed by counting edges in the shortest cycle in that graph. On the other hand, we consider the computation of girth g in the graph $\mathcal{X}$ with respect to the incidence matrix A.

Let $D = (d_{i,j})$ be an $n \times n$ $\{0,1\}$-adjacency matrix whose rows and columns correspond to edges in E in a way that $d_{i,j} = 1$ only if there exists a vertex in V that is incident with both e_i and e_j for all $0 \leq i, j \leq n-1$ with $i \neq j$. In other words, if $A = (a_{i,j})$ for all $0 \leq i \leq m-1$ and $0 \leq j \leq n-1$ and $D = (d_{i,j})$ for all $0 \leq i, j \leq n-1$, then

$$d_{ij} = \begin{cases} 1 & \text{if } i \neq j \text{ and } a_{l,i} = 1 \text{ and } a_{l,j} = 1 \text{ where } 0 \leq l \leq m-1, \\ 0 & \text{otherwise.} \end{cases}$$

Such a matrix is called **the edge relation matrix**. Let $\mathcal{X}_D = (V_D, E_D)$ denotes the graph which corresponds to D with $V_D = \{p_0, p_1, \ldots, p_{n-1}\}$ where $p_i = e_i$ for all $i = 0, 1, \ldots, n-1$, and we have $\{p_i, p_j\} \in E_D$ only if there exists a vertex $v \in V$ such that v is incident to both p_i and p_j, for all $0 \leq i, j \leq n-1$ and $i \neq j$. Similarly, $\{p_i, p_j\} \in E_D$ only if there exists $l \in \{0, 1, \ldots, m-1\}$ such that $a_{l,i} = 1$ and $a_{l,j} = 1$, for all $0 \leq i, j \leq n-1$ and $i \neq j$.

Also, we define an $n \times n$ matrix $S = (s_{i,j})$ for all $0 \leq i, j \leq n-1$ whose rows and columns correspond to edges in E. The entry (i, j) in S represents the **shortest path** from edge e_i to edge e_j in $\mathcal{X}$. If there is no path from e_i to e_j, then the length of the shortest path from e_i to e_j is defined to be ∞.

Therefore, S is the all pairs (edges) shortest paths, and can be computed by different methods. One particular method is by using Floyd's algorithm with inputs the matrix D and the size of $\mathcal{X}$, namely n, see Algorithm 4.7.1.

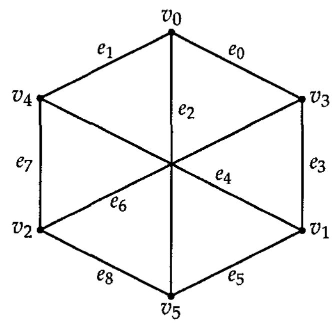
Algorithm 4.7.1 All-Shortest-Paths(D, n)

```
for i = 0 → n − 1 do
  for j = 0 → n − 1 do
    if d_{i,j} = 0 then
      if i = j then
        s_{i,j} = 0
      else
        s_{i,j} = ∞
      end if
    else
      s_{i,j} = 1
    end if
  end for
end for
for k = 0 → n − 1 do
  for i = 0 → n − 1 do
    for j = 0 → n − 1 do
      if s_{i,k} + s_{k,j} < s_{i,j} then
        s_{i,j} = s_{i,k} + s_{k,j}
      end if
    end for
  end for
end for
```

In particular, if $D = (d_{i,j})$ and $S = (s_{i,j})$ for all $0 \leq i, j \leq n-1$, then

$$s_{ij} = \begin{cases} 0 & \text{if } i = j, \\ l & \text{if there exists a path of length } l \text{ from } e_i \text{ to } e_j, \\ \infty & \text{otherwise,} \end{cases}$$

Example 4.7.1. Let $\mathcal{X} \in \mathcal{C}^{6,3}$ be the cubic graph of Figure 4.7.1 along with an incidence matrix A corresponding to $\mathcal{X}$.



$A =$

	e_0	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8
v_0	x	x	x						
v_1				x	x	x			
v_2							x	x	x
v_3	x			x			x		
v_4		x			x			x	
v_5			x			x			x

Figure 4.7.1: A cubic graph $\mathcal{X}$ with an incidence matrix A that corresponds to $\mathcal{X}$.

Then, the corresponding edge relations matrix D and all pairs shortest paths matrix S are displayed in Figure 4.7.2.

$D =$

	e_0	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8
e_0		x	x	x			x		
e_1	x		x		x			x	
e_2	x	x				x			x
e_3	x				x	x	x		
e_4		x		x		x		x	
e_5			x	x	x				x
e_6	x			x				x	x
e_7		x			x		x		x
e_8			x			x	x	x	

$S =$

	e_0	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8
e_0	0	1	1	1	2	2	1	2	2
e_1	1	0	1	2	1	2	2	1	2
e_2	1	1	0	2	2	3	2	2	3
e_3	1	2	2	0	1	1	1	2	2
e_4	2	1	2	1	0	1	2	1	2
e_5	2	2	3	1	1	0	2	2	3
e_6	1	2	2	1	2	2	0	1	1
e_7	2	1	2	2	1	2	1	0	1
e_8	2	2	3	2	2	3	1	1	0

Figure 4.7.2: Matrices D and S which are corresponding to graph $\mathcal{X}$ and incidence matrix A of Figure 4.7.1.

■

The idea of testing the girth property follows. During the row by row generation of A, assume that $r-1$ rows have been already constructed without violating the girth property, and that we are constructing the r^{th}-row for some $r = 0, 1, \ldots, m-1$. If vertex $v_r \in V$ is to be incident to both e_i and e_j for all $0 \le i, j \le n-1$, and $i \neq j$, then the girth condition to be satisfied is:

$$s_{i,j} + 1 \geq \text{girth } (g), \tag{4.2}$$

In fact, this test can be applied to every 1-entry we are trying to add in A during the generation. Consider the following example where we test every 1-entry in A during the construction of a new row.

Example 4.7.2. Let $\mathcal{X}$ be a cubic graph of order 6 that we desire to construct of girth 4. Let $A \in \mathcal{D}^{6,3}$ be an incidence matrix that corresponds to $\mathcal{X}$. Assume that 2 rows in A have been constructed and that we are trying to construct a third row as follows:

$A =$

	e_0	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8
v_0	x	x	x						
v_1	x			x	x				
v_2		x		?					
v_3									
v_4									
v_5									

where "?" says that we are trying to add a "1" in entry $a_{2,3}$ in A. However, the shortest path from e_1 to e_3, denoted by $s_{1,3}$, is 2, namely $e_2 - e_1 - e_4$, see Figure 4.7.3.

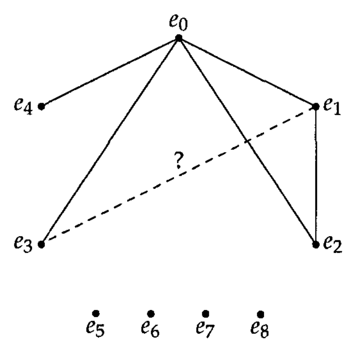


Figure 4.7.3: A graph $\mathcal{X}_D$ which can be constructed as described above.

So testing girth condition, Condition (4.2), we have $2 + 1 \not\geq 4$. Therefore, entry $(2, 3)$ in A can not be 1.

■

The girth condition testing can be added to Algorithm 3.1.1 to produce a new version where we take into account the girth condition, see Algorithm 4.7.2.

Algorithm 4.7.2 Generate(X_l: a transversal for $Sym_{(m)} \times Sym_{(n)}$-orbits on $\mathcal{D}_l^{m,k}$)

for $A \in X_l$ **do**
  compute $\pi_1^{-1}(A)$,
  compute a transversal $\Gamma(A) := \mathcal{T}(Aut(A), \pi_1^{-1}(A))$,
  for $(A, B) \in \Gamma(A)$ **do**
    if B satisfies the girth condition and $(A, B) \in \mu(B)$ **then**
      add B to X_{l+1}
    end if
  end for
end for

For example, Figure 4.7.4 displays a spanning tree with nodes corresponding to transversal for $Sym_{(6)} \times Sym_{(9)}$-orbits on $\mathcal{D}^{6,3}$ with girth 4.

Moreover, Algorithm 4.7.2 can be applied in the construction of flag graphs in $\mathcal{F}^{10,3} = \bigcup_{l=0}^{10} \mathcal{F}_l^{10,3}$ with girth 5 to produce the spanning tree T of Figure 4.7.5. The nodes (flag graphs) of the tree T are not shown because of the space limitation. Instead, we only show the flag graphs that are in the path from $\emptyset$ to node 34 of Figure 4.7.5. Namely, the nodes $\emptyset, 1, 3, 9, 14, 22, 25, 31, 32, 33, 34$ are shown in order in Figure 4.7.6.

Table 4.1 displays the number of orbits in each level of the search tree resulting of the generation procedure of the Petersen graph with employing the girth condition with girth 5.

In Table 4.2 we present some results of small cases of regular graphs (not necessarily connected) with a given girth. Note that, these results are not new, see [66]. The entries in the table gives the exact girth. For instance,

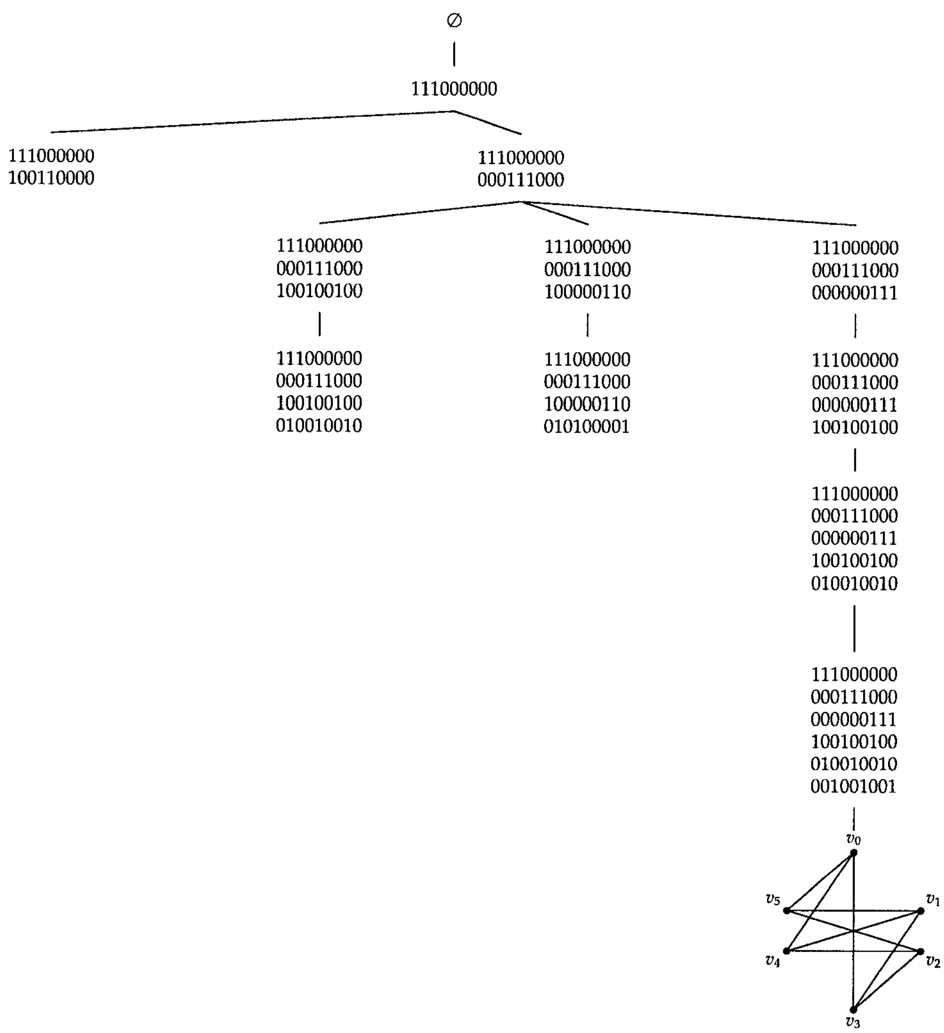


Figure 4.7.4: A spanning tree of cubic graph of order 6 with girth 4.

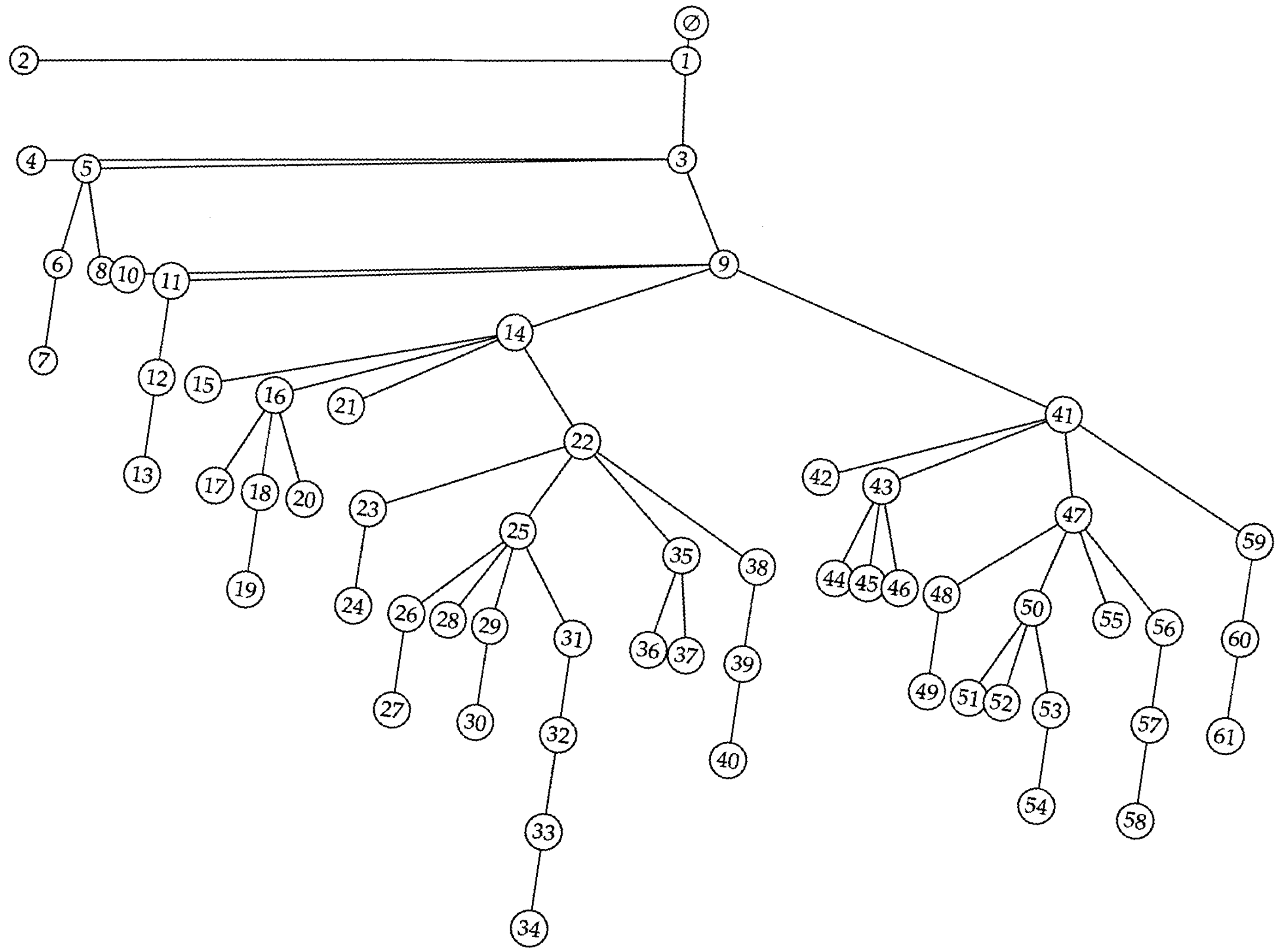


Figure 4.7.5: A search tree of the Petersen graph P with employment of girth condition.

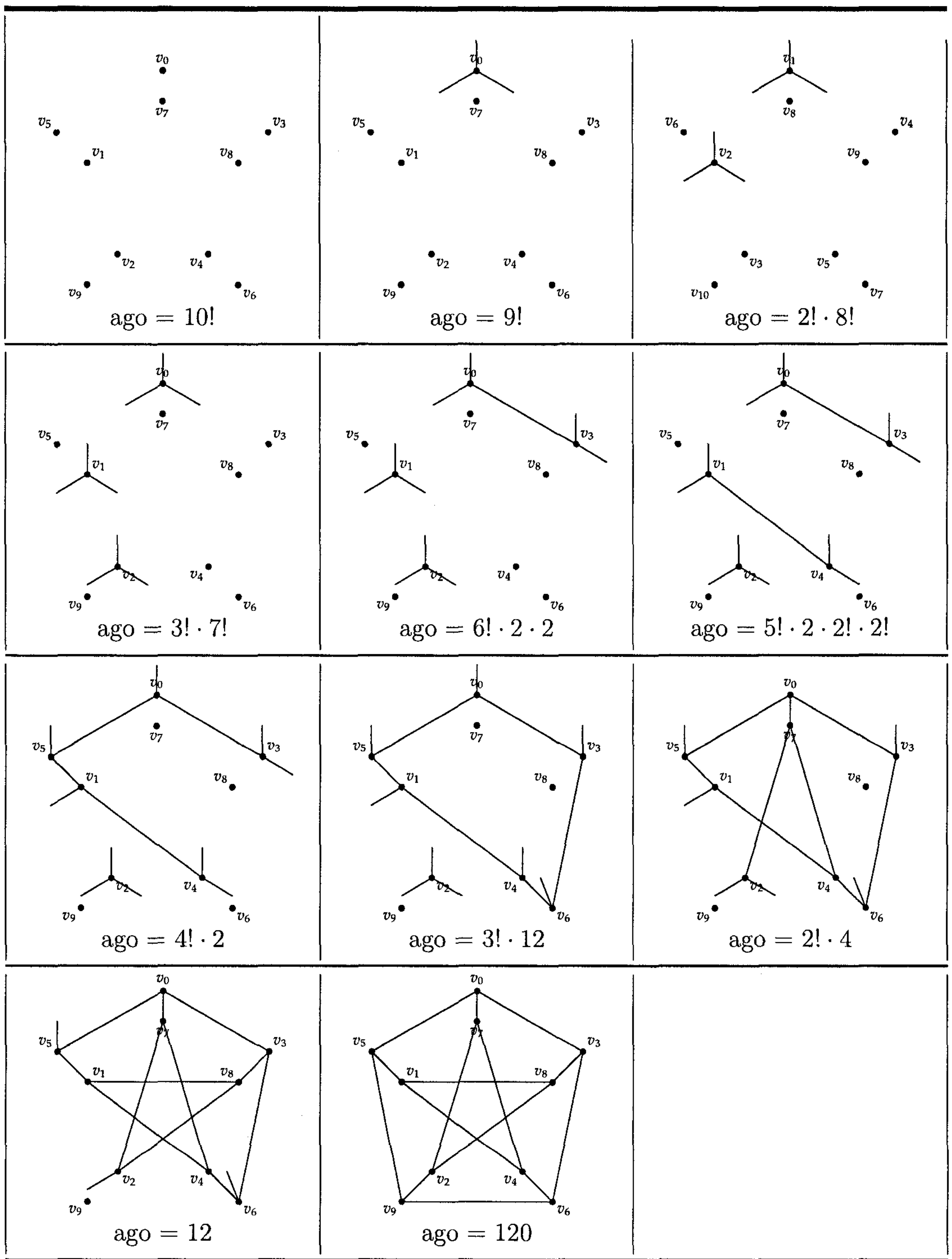


Figure 4.7.6: Flag graphs which correspond to constructing the Petersen graph vertex by vertex, starting from top left and ending at bottom right. Note that, we write **ago** for the automorphism group order of the corresponding graph.

Table 4.1: Number of orbits per level.

level	number of orbits
0	1
1	1
2	2
3	3
4	6
5	10
6	16
7	15
8	6
9	1
10	1

there are exactly 220 4-regular graphs on 14 vertices with girth 4, written as 4^{220} in Table 4.2.

Table 4.2 contains some entries with a "$\star$". Those graphs are in order, the Petersen graph which is a $(3,5)$-cage with automorphism group 120 presented in Figure 2.1.4 on page 13. Then, the **Heawood** graph of order 14 which is a $(3,6)$-cage and automorphism group of order 336, presented in Example 6.1.2 on page 125. This graph is also the incidence graph whose vertices are the points and the blocks of the Fano plane.

The **McGee** graph [60] is the graph of order 24 which is a $(3,7)$-cage presented by its incidence matrix in Example 6.1.3 on page 126. The automorphism group of this graph is of order 32. Finally, The **Tutte**'s graph [80] is a $(3,8)$-cage with automorphism group order 1440. See [74, 80] for more about the Tutte's graph.

Table 4.2: Regular graphs with given girth.

$n \backslash k$	3					4			5		
10			6^0	$\star\ 5^1$	4^5		5^0	4^2	5^0	4^1	3^{59}
11							5^0	4^2			
12			6^0	5^2	4^{21}		5^0	4^{12}	5^0	4^1	$3^{7,848}$
13							5^0	4^{31}			
14		7^0	$\star\ 6^1$	5^8	4^{103}		5^0	4^{220}	5^0	4^7	
15							5^0	$4^{1,606}$			
16		7^0	6^1	5^{48}	4^{752}		5^0	$4^{16,829}$	5^0	4^{388}	
17							5^0	$4^{193,900}$			
18		7^0	6^5	5^{450}	$4^{7,385}$						
19						6^0	5^1				
20		7^0	6^{32}	$5^{5,752}$	$4^{91,939}$						
21						6^0	5^8				
22		7^0	6^{385}								
24	8^0	$\star\ 7^1$	$6^{7,573}$								
26	8^0	7^3									
28	8^0	7^{21}									
30	$\star\ 8^1$										

Chapter 5

Isomorphism Invariants

In this and the following chapter, we will address the problem of isomorphism testing algorithmically. We are primely interested in solving the isomorphism problem for incidence structures. In this chapter, our focus is on computing invariants, which may allow a pre-classification.

The remaining case is when the invariants are not sufficient to tell different objects apart. This problem will be addressed in the following chapter.

The invariant for incidence structures that we have in mind is based on the idea that every incidence structure admits a tactical decomposition in the sense of Section 2.8 on page 46. Amongst all those decompositions we will single out one particular decomposition, which is canonical in the sense that there is an algorithm that computes the same decomposition for isomorphic incidence structures. This decomposition is called **the tactical decomposition obtained by ordering**, or **TDO** for short. It has been described by D. Betten and M. Braun [9].

5.1 Isomorphism Invariants

Isomorphism computations often require long and hard work for incidence structures because of their regular structures. Suitable **(isomorphism) invariants** can be used in this manner to expedite the computations in many different cases. Simply, a property associated with all structures of interest is an isomorphism invariant if it has the same value (or is the same) for any two isomorphic structures.

Definition 5.1.1. *Let G be a group acting on a finite set X. An **isomorphism invariant** is a function I such that for all elements x and y in X, it holds that $x \cong_G y$ implies $I(x) = I(y)$.*

For instance, if G is a group acting on a finite set X of incidence structures with m points, then the number of blocks, and the automorphism group order are isomorphism invariants in X.

An obvious immediate application for an invariant is to show that two incidence structures having different values of I are non-isomorphic, i.e. $I(x) \neq I(y)$ implies $x \not\cong y$.

Example 5.1.1. Consider the following two cubic graphs of order 6.

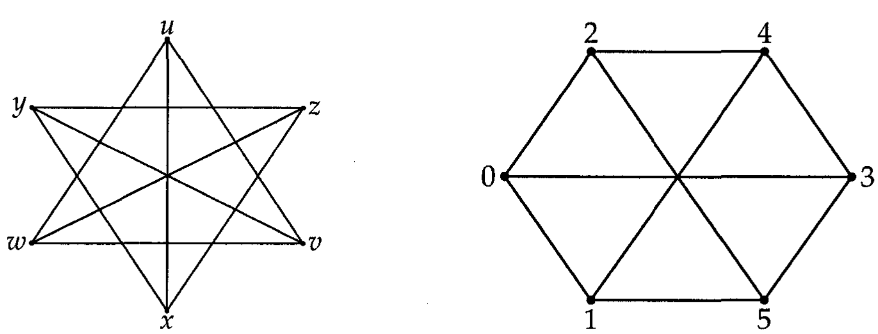


Figure 5.1.1: Two non-isomorphic cubic graphs with 6 vertices.

Both graphs have the same number of points (vertices) and the same number of blocks (edges) and yet they are not isomorphic. It can be seen that the graph on the left-hand side has two triangles, namely $\{x, y, z\}$ and $\{u, v, w\}$, where the other has none. Thus, the number of triangles in graphs is considered as an isomorphism invariant.

■

One particular kind of isomorphism invariant which has the power to perfectly distinguish between different isomorphism classes in a given class of incidence structures is called a **certificate**, see [33].

Definition 5.1.2. *Let G be a group acting on a finite set of incidence structures X. A **certificate** for an isomorphism is an invariant I such that for any two objects x and y in X,*

$$I(x) = I(y) \quad \textit{if and only if} \quad x \cong_G y. \tag{5.1}$$

A certificate is particularly desirable in the context of isomorphism computations because it suffices to test certificate values for equality to test for isomorphism.

Example 5.1.2. Let A be a $(m \times n)$ incidence matrix of an incidence structure $\mathcal{X} = (P, \mathcal{B})$ with m points and n blocks. Define

$$I(\mathcal{X}) = \min\{A^g \,:\, g \in Sym_{(m)} \times Sym_{(n)}\},$$

where the minimum here is taken with respect to lexicographical ordering on the set of all $m \times n$ incidence matrices. In other words, $I(\mathcal{G})$ is the (lexicographical) least incidence matrix associated with $\mathcal{X}$. In this case, $I(\mathcal{X})$ is a certificate by the meaning that $I(\mathcal{X}) \neq I(\mathcal{Y})$ if and only if $\mathcal{X} \ncong \mathcal{Y}$ for some other incidence structure $\mathcal{Y}$.

■

In general, a good invariant is both fast to compute and can distinguish well between different isomorphism classes in a given class of incidence structures. A general strategy for increasing the distinguishing power of invariants is to compound multiple invariants into one, see [19].

Lemma 5.1.3. *Let $I_1, I_2, \ldots, I_m$ be invariants for a class of incidence structures X. The following function $\mathcal{I}$, defined for all elements $x \in X$ by*

$$\mathcal{I}(x) = (I_1(x), I_2(x), \ldots, I_m(x)), \tag{5.2}$$

is an invariant for X.

There are different method of storing such isomorphism invariants for isomorphism testings. One particular method is be defining a **hash algorithm** as follows. Note that this algorithm is given in a C/C++ code.

Algorithm 5.1.1 Hashing Function

```
define HASH PRIME 174962718

int hashing(int old-hash, int add-to-old-hash)

{

    int h = old-hash, a = add-to-old-hash;

    do {

        h <<= 1;

        if (ODD(a)) h++;

        h %= HASH PRIME;

        a >>= 1;

    } while (a);

    return h;

}
```

Lemma 5.1.3 says that isomorphism invariants can be combined in an array. Using Algorithm 5.1.1, we can store all of the information in an integer.

5.2 Partition Refinement And The TDO-Method

In this section, we describe a method to compute an isomorphism invariant for incidence structures or $\{0, 1\}$-matrices. This is the above-mentioned TDO-method of D. Betten and M. Braun [9].

This method is based on the idea that every incidence structure can be decomposed tactically. That is, for every incidence structure one can find a partition of the point and block sets such that the resulting decomposition becomes tactical in the sense of Section 2.8 on page 46. That is, for any pair (C, D), where C is a class of the point partition and D is a class of the block partition, the following holds.

a) For $p \in C$, the number of incident pairs (p, B), where $B \in D$, is independent of the choice of $p \in C$.

b) For $B \in D$, the number of incident pairs (p, B), where $p \in C$, is independent of the choice of $B \in D$.

Once a tactical decomposition has been obtained, we wish to consider the resulting decomposition schemes (as introduced in Section 2.8 on page 46) as an isomorphism invariant, in the sense of Definition 5.1.1 on page 104, of the incidence structure. We remark that this can only be done for certain decompositions. For instance, the discrete partition of points and blocks is always tactical, but it is unsuitable for an invariant since there are many ways in which the points and blocks can be arranged (recall that we are concerned with ordered partitions, that is, the ordering of point classes and block classes matters), see Example 5.2.1.

Example 5.2.1. Consider the two incidence structures (graphs) of Figure 5.2.1, say S_1 (left) and S_2 (right).

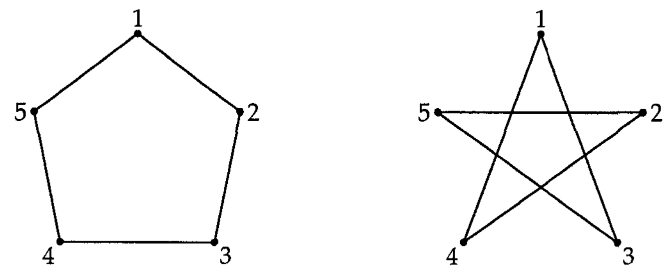


Figure 5.2.1: Incidence structures S_1 (left) and S_2 (right).

Applying discrete partition of the points and blocks, the decompositions schemes are different, whereas the incidence structures (which are graphs) are clearly isomorphic. Figure 5.2.2 displays the corresponding incidence matrices for S_1 and S_2, respectively, after applying a discrete partitioning of the points and blocks.

	e_5	e_6	e_7	e_8	e_9
1	X				X
2	X	X			
3		X	X		
4			X	X	
5				X	X

	e_5	e_6	e_7	e_8	e_9
1	X				X
2			X	X	
3	X	X			
4				X	X
5		X	X		

Figure 5.2.2: Incidence matrices with applying discrete partitions for the points and blocks of S_1 (left) and S_2 (right).

■

The algorithm we are going to describe computes a tactical decomposition whose decomposition scheme is an isomorphism invariant. It is based on an alternating sequence of refinements of the partitions of points and blocks, respectively.

Let $\mathcal{X} = (P, \mathcal{B})$ be a finite incidence structure with $P = \{p_0, p_1, \ldots, p_{m-1}\}$ and $\mathcal{B} = \{B_0, \ldots, B_{n-1}\}$. Let A be an $m \times n$ incidence matrix that corresponds to $\mathcal{X}$. That is, if $A = (a_{i,j})$ for all $0 \leq i \leq m-1$ and for all $0 \leq j \leq n-1$, then $a_{i,j} = 1$ if $p_i \in B_j$ and 0 otherwise.

Let $\mathcal{R} = \{0, 1, \ldots, m-1\}$ and $\mathcal{C} = \{m, \ldots, m+n-1\}$ be the sets of row and column indices, respectively. Let $S = \mathcal{R} \cup \mathcal{C} = \{0, 1, \ldots, m+n-1\}$ be the set of all indices. Define a relation $\mathcal{I}$ such that

$$(i, j) \in \mathcal{I} \text{ for all } i, j \in S \text{ only if } a_{i,j-m} = 1. \tag{5.3}$$

In the following, we will not distinguish between a row and its index (likewise, between a column and its index). That is, we may as well assume that the incidence structure is $(P, \mathcal{B})$, with $P = \{0, 1, \ldots, m-1\}$ and $\mathcal{B} = \{m, \ldots, m+n-1\}$.

Let $\Pi_{\mathcal{R},\mathcal{C}} = \{\mathcal{R} \mid \mathcal{C}\}$ be the partition of S which distinguishes rows and columns. Any refinement Σ of $\Pi_{\mathcal{R},\mathcal{C}}$ has the form

$$\{R_0 \mid R_1 \mid \ldots \mid R_{M-1} \mid C_0 \mid C_1 \mid \ldots \mid C_{N-1}\},$$

where $\{R_0 \mid \ldots \mid R_{M-1}\}$ is a refinement of the row partition $\mathcal{R}$ and $\{C_0 \mid \ldots \mid C_{N-1}\}$ is a refinement of the column partition $\mathcal{C}$. We also say that Σ **is a partition of the incidence structure** $\mathcal{X}$ with M row parts, and N column parts. We also say that Σ is **row tactical** (column tactical, respectively) if the decomposition of $\mathcal{X}$ induced by Σ is row tactical (column tactical, respectively).

In the following, we will present a refinement procedure for partitions of incidence structures. Before we do this, we need to introduce the following notation.

If $G \leq Aut(\mathcal{X})$, and if $\Pi = \{R_0 \mid \ldots \mid R_{M-1} \mid C_0 \mid \ldots \mid C_{N-1}\}$ is a partition of the incidence structure, then

$$G_\Pi = \{g \in G \mid R_i^g = R_i,\ C_j^g = C_j, \text{ for all } i = 0, \ldots, M-1,\ j = 0, \ldots N-1\},$$

is the **stabilizer of the partition in** G. Clearly, $G_{\Pi_{\mathcal{R},\mathcal{C}}} = G$.

Given a partition Π of the incidence structure $\mathcal{X}$, we are looking for a refined partition Σ of Π such that Σ is row tactical or column tactical (or both). In addition, we require that $G_\Sigma = G_\Pi$, that is, the refinement from Π

to Σ preserves all automorphisms that Π admitted. It is desirable to compute a refinement Σ which is as fine as possible, since this gives more information about the orbits of the original stabilizer G_{Π}. There are many different ways in which such a partition Σ can be defined.

The procedure that we apply to refine the partition computes the coarsest tactical decomposition which is a refinement of the given partition Π_0. Since the refinements are obtained using an ordering of certain invariants, this coarsest tactical refinement partition is unique (with respect to the algorithm by which it is computed). The process is known as **tactical decomposition by ordering**, or TDO for short. It has been developed by D. Betten and M. Braun, see [9].

Definition 5.2.1. *Let S be a finite set, and let $\Pi = \{C_0 \mid C_1 \mid \ldots \mid C_{m_1}\}$ be an ordered partition of S. If $f : S \to \mathbb{N}^{\mathbb{N}}$ is a function, then refine$(\Pi, f) = \{D_0 \mid D_1 \mid \ldots \mid D_{m_2}\}$ is the unique partition Σ with:*

- *$\Sigma < \Pi$, and*
- *If $x \in D_i$ and $y \in D_j$ with $D_i, D_j \subseteq C_k$ for $0 \leq i, j \leq m_2$ and $0 \leq k \leq m_1$, then*
  - *$f(x) = f(y) \iff i = j$, and*
  - *$f(y) \prec f(x) \iff i < j$.*

Definition 5.2.2. *Let Π_q be a given partition of an incidence structure $\mathcal{X}$, and let A be an $m \times n$ incidence matrix associated with $\mathcal{X}$. For our convenience, we write*

$$\Pi_q = \{R_{q,0} \mid \ldots \mid R_{q,M-1} \mid C_{q,0} \mid \ldots \mid C_{q,N-1}\}. \tag{5.4}$$

For $k = 0, 1, \ldots, N-1$, let

$$r_{k,i} = \begin{cases} \#(i,j) \in \mathcal{I} & \text{if } i \in \mathcal{R},\ j \in \mathcal{C} \cap C_{q,k}, \\ 0 & \text{if } i \in \mathcal{C}. \end{cases} \tag{5.5}$$

be the **row-sum**$_i(A)$ ***restricted to the column part*** k*, and for* $k = 0, 1, \ldots, M-1$*, let*

$$c_{k,j} = \begin{cases} \#(i,j) \in \mathcal{I} & \text{if } j \in \mathcal{C},\ i \in \mathcal{R} \cap R_{q,k}, \\ 0 & \text{if } j \in \mathcal{R}. \end{cases} \tag{5.6}$$

be the **col-sum**$_j(A)$ ***restricted to the row part*** k.

Assume that $\Pi_0 = \{R_{0,0} \mid C_{0,0}\}$ is a partition (of the Form 5.4) of an incidence structure $\mathcal{X} = (P, \mathcal{B})$ with $P = \{0, 1, \ldots, m-1\} =: R_{0,0}$ and $\mathcal{B} = \{m, \ldots, m+n-1\} =: C_{0,0}$. Moreover, let $S = \{0, 1, \ldots, m+n-1\}$. Let A be an $(m \times n)$ incidence matrix associated with $\mathcal{X}$. We refine Π_0 as follows:

We compute $r_{k,i}$ for $i = 0, \ldots, m-1$ and set $r_{k,i} = 0$ for $i = m, \ldots, m+n-1$. Here, $k = 0$ since we only have one column part, namely $C_{0,0}$. We define a function $f : S \to \mathbb{N}^1$ such that $f(i) = r_{0,i}$ for $i \in S$. Since $f(i) = 0$ for all $i \in C_{0,0}$, the column part will not be refined in this step. Then,

$$\Pi_1 := \text{refine}(\Pi_0, f) := \{R_{1,0} \mid \ldots \mid R_{1,M-1} \mid C_{1,0}\},$$

is the refinement for the partition Π_0.

Note that $C_{1,0} = C_{0,0}$ and that $R_{1,0}, R_{1,1}, \ldots, R_{1,M-1} \subseteq R_{0,0}$. In this case, we say that cell $R_{0,0}$ is split into subcells $R_{1,0}, \ldots, R_{1,M-1}$. Such split is occurred if the points in $R_{0,0}$ are distinguishable by using the corresponding row sums. In particular, $\{R_{1,0} \mid \ldots \mid R_{1,M-1}\}$ is a refinement of the row partition $R_{0,0}$.

Also, we can refine Π_1 as follows: We compute $c_{k,j}$ for row parts $k = 0, 1, \ldots, M-1$, for $j = m, m+1, \ldots, m+n-1$, and we set $c_{k,j} = 0$ for all $j \in R_{0,0}$. Again, we define a function $f : S \to \mathbb{N}^M$ such that $f(j) = (c_{0,j}, c_{1,j}, \ldots, c_{M-1,j})$ for $j \in S$. Note that we omit $f(j)$ for all $j \in R_{0,0}$ in practice. Then, $\Pi_2 := \text{refine}(\Pi_1, f)$ is the refinement for the partition Π_1.

In general, if Π_q is a partition of $\mathcal{X}$, then we have two cases for Π_q. Namely, if Π_q is row tactical, then we would have Π_{q+1} column tactical. While if Π_q is column tactical, then we get Π_{q+1} row tactical. Then we consider the following two procedures for each case.

Case 1: Π_q is column tactical with N column parts.

step 1: Compute $r_{k,i}$ for $i = 0, \ldots, m-1$ and for $k = 0, 1, \ldots, N-1$, and set $r_{k,i} = 0$ for all $i = m, \ldots, m+n-1$.

step 2: Define a function $f : S \to \mathbb{N}^N$ such that $f(i) = (r_{0,i}, r_{1,i}, \ldots, r_{N-1,i})$ for $i \in S$.

step 3: $\Pi_{q+1} := \text{refine}(\Pi_q, f)$ (Π_{q+1} is row tactical now).

Case 2: Π_q is row tactical with M row parts.

step 1: Compute $c_{k,j}$ for $j = m, \ldots, m+n-1$ and for $k = 0, 1, \ldots, M-1$, and set $c_{k,j} = 0$ for all $j = 0, \ldots, m-1$.

step 2: Define a function $f : S \to \mathbb{N}^M$ such that $f(i) = (c_{0,j}, c_{1,j}, \ldots, c_{M-1,j})$ for $j \in S$.

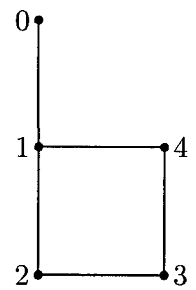
step 3: $\Pi_{q+1} := \text{refine}(\Pi_q, f)$ (Π_{q+1} is column tactical now).

Clearly, if Π_q is a discrete partition, then the previous procedure will make no refinements to Π_q. Assume that $\Pi_{q+1} = \Pi_q$. Then, we say that Π_q is an **equitable (or TDO) partition** with respect to Π_0. Also, we say that Π_q does not split, and we write $\text{TDO}(\Pi_0, \mathcal{X}) = \Pi_q$. We write $\textbf{TDOS}(\Pi_q, \mathcal{X})$ to denote the decomposition schemes of $\mathcal{X}$ with respect to the partition Π_q.

It can be shown that the TDO partition $\Sigma = \text{TDO}(\Pi, \mathcal{X})$ is the coarsest equitable partition refining Π of the graph $\mathcal{G} = (S, \mathcal{I})$, where $\mathcal{I}$ is as in 5.3, in the sense of Godsil [27].

Example 5.2.2. Let $\mathcal{G}$ be the graph of Figure 5.2.3 of order 5 and size 5.

Let $\mathcal{R} = \{0, 1, 2, 3, 4\}$ and $\mathcal{C} = \{5, 6, 7, 8, 9\}$ be the set of vertices and edges (alternatively, rows and columns), respectively. Assume that $\Pi_0 = \{\mathcal{R} \mid \mathcal{C}\}$ is



$$A = \begin{array}{|c|c|c|c|c|c|}\hline & 5 & 6 & 7 & 8 & 9 \\ \hline 0 & \times & & & & \\ \hline 1 & \times & \times & \times & & \\ \hline 2 & & \times & & \times & \\ \hline 3 & & & & \times & \times \\ \hline 4 & & & \times & & \times \\ \hline \end{array}$$

Figure 5.2.3: A graph $\mathcal{G}$ with its incidence matrix A

a partition of $\mathcal{G}$, and that we want to compute $\text{TDO}(\Pi_0, \mathcal{G})$. The **partitioned incidence matrices** corresponds to $\mathcal{G}$ with partition Π_0 is

$$\begin{array}{|c|c|c|c|c|c|}\hline & 5 & 6 & 7 & 8 & 9 \\ \hline 0 & \times & & & & \\ \hline 1 & \times & \times & \times & & \\ \hline 2 & & \times & & \times & \\ \hline 3 & & & & \times & \times \\ \hline 4 & & & \times & & \times \\ \hline \end{array}$$

$$\Pi_0 = \{0,1,2,3,4 \mid 5,6,7,8,9\}$$

We first compute $r_{0,i}$ for rows $i = 0, \ldots, 4$, and set $r_{0,i} = 0$ for all $5 \leq i \leq 9$. Therefore, we get

$$r_{0,0} = 1,$$
$$r_{0,1} = 3,$$
$$r_{0,2} = r_{0,3} = r_{0,4} = 2.$$

We then define a function $f : \mathcal{R} \cup \mathcal{C} \to \mathbb{N}$ such that $f(i) = r_{0,i}$ for all $0 \leq i \leq 4$ and we can ignore f for $i \geq 5$ since $f(i)$ is always zero for $i \geq 5$. Therefore,

$$\Pi_1 := \text{refine}(\Pi_0, f) = \{1 \mid 2,3,4 \mid 0 \mid 5,6,7,8,9\},$$

is a refinement for Π_0 and that Π_1 is row tactical decomposition. Note that no refinements have been made in the column part since we only focusing on row indices and ignoring column indices in $\mathcal{C}$.

	5	6	7	8	9
1	X	X	X		
2		X		X	
3				X	X
4			X		X
0	X				

$\Pi_1 = \{1 \mid 2,3,4 \mid 0 \mid 5,6,7,8,9\}$

$\rightarrow$	5
1	3
3	2
1	1

Now that we have 3 row parts in $\Pi_1 = \{1 \mid 2,3,4 \mid 0 \mid 5,6,7,8,9\}$, we refine Π_1 with ignoring row indices in this step as follows. We compute $c_{k,j}$ for $k = 0,1,2$ as follows.

$$c_{k,5} = (1,0,1),$$
$$c_{k,6} = (1,1,0),$$
$$c_{k,7} = (1,1,0),$$
$$c_{k,8} = (0,2,0),$$
$$c_{k,9} = (0,2,0).$$

Then, we let $f(j) = (c_{0,j}, c_{1,j}, c_{2,j})$ for all $5 \leq j \leq 9$. Therefore,

$$\Pi_2 := \text{refine}(\Pi_1, f) = \{1 \mid 2,3,4 \mid 0 \mid 6,7 \mid 5 \mid 8,9\},$$

is a refinement for Π_1 and that Π_2 is column tactical decomposition.

	6	7	5	8	9
1	X	X	X		
2	X			X	
3				X	X
4		X			X
0			X		

$\Pi_2 = \{1 \mid 2,3,4 \mid 0 \mid 6,7 \mid 5 \mid 8,9\}$

$\downarrow$	2	1	2
1	1	1	0
3	1	0	2
1	0	1	0

Now, we refine $\Pi_2 = \{1 \mid 2,3,4 \mid 0 \mid 6,7 \mid 5 \mid 8,9\}$ to get a refined partition Π_3 which is row tactical. Computing $r_{k,i}$ for $k = 0,1,2$, we get

$$r_{k,0} = (0,1,0),$$
$$r_{k,1} = (2,1,0),$$
$$r_{k,2} = (1,0,1),$$
$$r_{k,3} = (0,0,2),$$
$$r_{k,4} = (1,0,1).$$

Let $f(i) = (r_{0,i}, r_{1,i}, r_{2,i})$ for all $0 \le i \le 4$, and therefore

$$\Pi_3 := \text{refine}(\Pi_2, f) = \{1 \mid 2,4 \mid 3 \mid 0 \mid 6,7 \mid 5 \mid 8,9\}.$$

	6	7	5	8	9
1	X	X	X		
2	X			X	
4		X			X
3			X		
0				X	X

$\Pi_3 = \{1 \mid 2,4 \mid 3 \mid 0 \mid 6,7 \mid 5 \mid 8,9\}$

→	2	1	2
1	2	1	0
2	1	0	1
1	0	1	0
1	0	0	2

↓	2	1	2
1	1	1	0
2	1	0	1
1	0	1	0
1	0	0	1

To refine Π_3 we compute $c_{k,j}$ for $k = 0,1,2,3$, as follows.

$$c_{k,5} = (1,0,1,0),$$
$$c_{k,6} = (1,1,0,0),$$
$$c_{k,7} = (1,1,0,0),$$
$$c_{k,8} = (0,1,0,1),$$
$$c_{k,9} = (0,1,0,1).$$

Let $f(j) = (c_{0,j}, c_{1,j}, c_{2,j}, c_{3,j})$ for all $5 \le j \le 9$. Therefore,

$$\Pi_4 := \text{refine}(\Pi_3, f) = \{1 \mid 2,4 \mid 3 \mid 0 \mid 6,7 \mid 5 \mid 8,9\} = \Pi_3.$$

Thus, we stop here and we say that $\mathrm{TDO}(\Pi_0, \mathcal{G}) = \Pi_3$, and that Π_3 is tactical (equitable partition).

$\blacksquare$

Lemma 5.2.3. *Let $\mathcal{X} = (P, \mathcal{B})$ be an incidence structure, and let $G \leq Aut(\mathcal{X})$ be a subgroup of the automorphism group of $\mathcal{X}$. Let Π be a partition of $\mathcal{X}$, and $\Sigma := TDO(\Pi, \mathcal{X})$. Then, $G_\Pi = G_\Sigma$ (up to conjugacy).*

Proof. It is clear that Σ is obtained from Π by a sequence of splittings. That is, there is a chain of partitions

$$\Pi = \Sigma_0 > \Sigma_1 > \ldots > \Sigma_n = \Sigma,$$

and Σ_{i+1} is obtained from Σ_i by splitting one non-singleton class of Σ_i into two classes of Σ_{i+1}.

We prove the statement by induction on i, the index of Σ_i, for $i = 0, 1, \ldots, n$.

For $i = 0$, the statement is trivial. Assume that $G_{\Sigma_i} = G_{\Sigma_0}$ has already be proven. Then, Σ_{i+1} is obtained by splitting a class C of Σ_i with $|C| > 1$ into classes $A, B \in \Sigma_{i+1}$.

Let $p_{i_1} \in A$ and $p_{i_2} \in B$. Consider first the case that C is a row class. Then, there is a column class D_k such that $r_{k,i_1} \neq r_{k,i_2}$ where $r_{k,i}$'s are as in 5.5. But since the $r_{k,i}$'s are isomorphism invariants with respect to Σ_i, no element $g \in G_{\Sigma_i}$ can map p_{i_1} to p_{i_2}. That is, every element in G_{Σ_i} preserves the partition Σ_{i+1}, and hence $G_{\Sigma_i} \leq G_{\Sigma_{i+1}}$. Moreover, it is clear that $G_{\Sigma_{i+1}} \leq G_{\Sigma_i}$.

The other case where the class C is a column class can be proved by the same arguments with using $c_{k,j}$'s where $c_{k,j}$'s are as in 5.6. Therefore, $G_{\Sigma_i} = G_{\Sigma_{i+1}}$. $\square$

Corollary 5.2.4. *Let $\mathcal{X} = (P, \mathcal{B})$ be an incidence structure, and let $G \leq Aut(\mathcal{X})$ be a subgroup of the automorphism group of $\mathcal{X}$. Let Π be a partition of $\mathcal{X}$. If $TDO(\Pi, \mathcal{X})$ is discrete, then $G_\Pi = 1$.*

Let $\mathcal{G}$ and Π_0 be the graph and the partition of Example 5.2.2. If $G \leq Aut(\mathcal{G})$, then $G_{\Pi_0} = G_{\Pi_3}$ by Lemma 5.2.3.

Chapter 6

Isomorph Rejection with Partition Backtrack

Let G be a group acting on a finite set X of incidence structures. Let $\mathcal{X} = (P, \mathcal{B}) \in X$ be an incidence structure on m points and n columns. This chapter discuss problems related to computing the automorphism group of $\mathcal{X}$, namely $Aut(\mathcal{X})$, by considering an induction of a chain of group stabilizers. This procedure is called **partition backtrack** algorithm.

Moreover, we present the McKay's μ-function [64] which relies on a canonical labeling function φ which can be realized by a partition backtrack algorithm, see Leon [75, 76] for more readings.

Let $\mathcal{X} = (P, \mathcal{B})$ be an incidence structure on m points and n blocks associated with an initial partition $\Pi_0 = \{\mathcal{R} \mid \mathcal{C}\}$, where $\mathcal{R} = \{R_0 \mid R_1 \mid \ldots \mid R_{M-1}\}$ is a partition of P, and $\mathcal{C} = \{C_0 \mid \ldots \mid C_{N-1}\}$ is a partition of $\mathcal{B}$.

6.1 The Derived TDO

In this section, we describe a method, called the **derived TDO**, to approximate the orbits of the automorphism group by an ordered partition Π. That is, if Σ is the (unordered) partition of orbits of $Aut(\mathcal{X})$, then Σ is a refinement of Π. We remark that the derived TDO method was first used by D. Betten and U. Schumacher [10] in 1990 to study the configurations 10_3.

The partition refinement procedure in Chapter 5 simply refines a given partition to some tactical partition. However, in many interesting cases, for instance Steiner systems, cubic graphs, etc, the incidence matrix is already tactical, i.e. $\mathrm{TDO}(\Pi_0, \mathcal{X}) := \Pi_0$. In other words, we are already stuck. In order to gain some information even in these cases, a refinement of the TDO-method can be used. This method is called the refined TDO and we will present it in this section.

Definition 6.1.1. *Let S be a finite set. If $x, y \in S$, then define a function $e_x : S \longrightarrow \mathbb{N}^S$*

$$e_x(y) = \begin{cases} 1 & \textit{if } x = y, \\ 0 & \textit{otherwise.} \end{cases}$$

If $S = P \cup \mathcal{B}$ and $G \leq Aut(\mathcal{X})$ fixes Π and $x \in S$, then G_x stabilizes $\Pi_x := \mathrm{TDO}(\mathrm{refine}(\Pi, e_x), \mathcal{X})$ by Lemma 5.2.3.

Given an incidence structure $\mathcal{X} = (P, \mathcal{B})$, we employ the e_x function for $x \in P$, and consider the set of TDOs, $\mathrm{TDO}(\mathrm{refine}(\Pi_0, e_x), \mathcal{X})$ where $x \in P$. The resulting TDO-scheme of the point x is an invariant. The refinement with respect to this invariant is called the **derived TDO**. Note that the TDO-schemes are an invariant, while the partitioned incidence matrices are not. The reason is that the partitioned incidence matrix may still allow isomorphisms in case it is not discrete.

Let $\Pi_x := \mathrm{TDO}(\mathrm{refine}(\Pi_0, e_x), \mathcal{X}) := \{R_{x,0} \mid \ldots \mid R_{x,s-1} \mid C_{x,0} \mid \ldots \mid C_{x,t-1}\}$. Let $r_{i,j}$'s be defined as in Figure 2.8.1 on page 48 for all $0 \leq i \leq s-1$ and for all $0 \leq j \leq t-1$. Let $\mathrm{TDOS}(\Pi_x, \mathcal{X})$ be the scheme

$$\begin{array}{c|cccc}
 & |C_{x,0}| & |C_{x,1}| & \cdots & |C_{x,t-1}| \\
\hline
|R_{x,0}| & r_{0,0} & r_{0,1} & \cdots & r_{0,t-1} \\
|R_{x,1}| & r_{1,0} & r_{1,1} & \cdots & r_{1,t-1} \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
|R_{x,s-1}| & r_{s-1,0} & r_{s-1,1} & \cdots & r_{s-1,t-1}
\end{array}$$

We define the invariant of $x \in P$ to be $\mathcal{I}(x) := \mathrm{TDOS}(\Pi_x, \mathcal{X})$, and we think of $\mathcal{I}(x)$ as a sequence of integers

$$\big(s, t, |R_{x,0}|, \ldots, |R_{x,s-1}|, |C_{x,0}|, \ldots, |C_{x,t-1}|,$$
$$r_{0,0}, r_{0,1}, \ldots, r_{0,t-1}, r_{1,0}, \ldots r_{1,t-1}, r_{s-1,0}, \ldots, r_{s-1,t-1}\big).$$

Then, $\mathrm{der}(\Pi_0, \mathcal{X}) := \mathrm{TDO}(\mathrm{refine}(\Pi_0, \mathcal{I}), \mathcal{X})$ is the derived TDO. We also define

$$\mathrm{der}_{\mathcal{P}}(\Pi_0, \mathcal{X}) := \mathrm{TDO}(\mathrm{refine}(\Pi_0, \mathcal{I}), \mathcal{X}),$$

where $\mathcal{I}(x) = 0$ if $x \in \mathcal{C}$, and

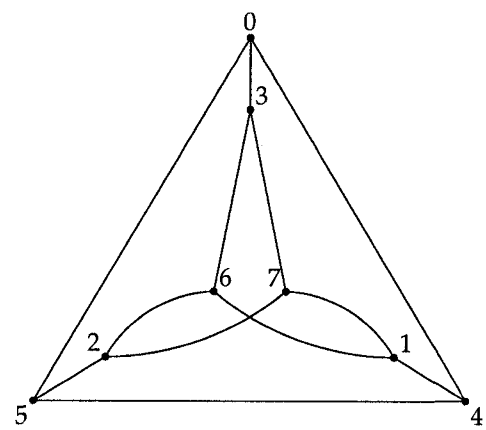
$$\mathrm{der}_{\mathcal{B}}(\Pi_0, \mathcal{X}) := \mathrm{TDO}(\mathrm{refine}(\Pi_0, \mathcal{I}), \mathcal{X}),$$

where $\mathcal{I}(x) = 0$ if $x \in \mathcal{R}$. Also, we write $\mathrm{der}(\mathcal{X}) := \mathrm{der}(\Pi_0, \mathcal{X})$ with $\Pi_0 = \{\mathcal{R} \mid \mathcal{C}\}$. Moreover, $\mathrm{der}_{\mathcal{P}}(\mathcal{X})$ and $\mathrm{der}_{\mathcal{B}}(\mathcal{X})$ are defined similarly.

Example 6.1.1. Let $\mathcal{G} = (V, E)$ be the cubic graph on 8 vertices of Figure 6.1.1, with $V = \{0, 1, \ldots, 7\}$.

Let $\mathcal{R} = \{0, 1, \ldots, 7\}$ and $\mathcal{C} = \{8, \ldots, 19\}$ be the set of row and column indices. Let $\Pi_0 = \{\mathcal{R} \mid \mathcal{C}\}$ be the initial partition of points and blocks of $\mathcal{G}$. Clearly, $\mathrm{TDO}(\Pi_0, \mathcal{G}) := \Pi_0$. Therefore, we consider the employment of the e_x function for $x \in \mathcal{R}$ as described above. Figure 6.1.2 displays the resulting TDOS after computing $\mathrm{TDO}(\mathrm{refine}(\Pi_0, e_x), \mathcal{G})$ where $x \in \mathcal{R}$.

Moreover, Figure 6.1.3 represents the multiset of the resulting TDO-schemes of Figure 6.1.2. In the first column of Figure 6.1.3, we show the set of points



$A =$

	8	9	10	11	12	13	14	15	16	17	18	19
0	x	x	x									
1				x	x	x						
2							x	x	x			
3	x									x	x	
4		x		x								x
5			x				x					x
6					x			x		x		
7						x			x		x	

Figure 6.1.1: A cubic graph $\mathcal{G}$ of order 8 with its incidence matrix A.

$\mathcal{I}(0) \rightarrow$	2	1	1	2	2	4
1	2	1	0	0	0	0
2	1	0	1	1	0	0
1	0	1	0	0	2	0
2	0	0	0	1	0	2
2	0	0	0	0	1	2

$\mathcal{I}(1) \rightarrow$	2	1	4	2	2	1
1	2	1	0	0	0	0
2	1	0	2	0	0	0
1	0	1	0	2	0	0
2	0	0	2	0	1	0
2	0	0	0	1	1	1

$\mathcal{I}(2) \rightarrow$	2	1	4	2	2	1
1	2	1	0	0	0	0
2	1	0	2	0	0	0
1	0	1	0	2	0	0
2	0	0	2	0	1	0
2	0	0	0	1	1	1

$\mathcal{I}(3) \rightarrow$	2	1	4	2	2	1
1	2	1	0	0	0	0
2	1	0	2	0	0	0
1	0	1	0	2	0	0
2	0	0	2	0	1	0
2	0	0	0	1	1	1

$\mathcal{I}(4) \rightarrow$	2	1	1	2	2	4
1	2	1	0	0	0	0
2	1	0	1	1	0	0
1	0	1	0	0	2	0
2	0	0	0	1	0	2
2	0	0	0	0	1	2

$\mathcal{I}(5) \rightarrow$	2	1	1	2	2	4
1	2	1	0	0	0	0
2	1	0	1	1	0	0
1	0	1	0	0	2	0
2	0	0	0	1	0	2
2	0	0	0	0	1	2

$\mathcal{I}(6) \rightarrow$	3	3	3	3
1	3	0	0	0
3	1	1	1	0
1	0	3	0	0
3	0	0	1	2

$\mathcal{I}(7) \rightarrow$	3	3	3	3
1	3	0	0	0
3	1	1	1	0
1	0	3	0	0
3	0	0	1	2

Figure 6.1.2: The invariants $\mathcal{I}(x)$ for $x = 0, 1, \ldots, 7$.

points		TDO-schemes
$\{0,4,5\}$	$3\times$	$\begin{array}{c@{\;\;}c} \to & 2\ 1\ 1\ 2\ 2\ 4 \\ \hline 1 & 2\ 1\ 0\ 0\ 0\ 0 \\ 2 & 1\ 0\ 1\ 1\ 0\ 0 \\ 1 & 0\ 1\ 0\ 0\ 2\ 0 \\ 2 & 0\ 0\ 0\ 1\ 0\ 2 \\ 2 & 0\ 0\ 0\ 0\ 1\ 2 \end{array}$
$\{1,2,3\}$	$3\times$	$\begin{array}{c@{\;\;}c} \to & 2\ 1\ 4\ 2\ 2\ 1 \\ \hline 1 & 2\ 1\ 0\ 0\ 0\ 0 \\ 2 & 1\ 0\ 2\ 0\ 0\ 0 \\ 1 & 0\ 1\ 0\ 2\ 0\ 0 \\ 2 & 0\ 0\ 2\ 0\ 1\ 0 \\ 2 & 0\ 0\ 0\ 1\ 1\ 1 \end{array}$
$\{6,7\}$	$2\times$	$\begin{array}{c@{\;\;}c} \to & 3\ 3\ 3\ 3 \\ \hline 1 & 3\ 0\ 0\ 0 \\ 3 & 1\ 1\ 1\ 0 \\ 1 & 0\ 3\ 0\ 0 \\ 3 & 0\ 0\ 1\ 2 \end{array}$

Figure 6.1.3: The multiset of invariants from Figure 6.1.2.

in $\mathcal{R}$ that have the same such TDO-schemes, where in the second column we present the corresponding TDO-schemes.

Therefore, if $\mathcal{R}_1 := \{1,2,3 \mid 0,4,5 \mid 6,7\}$, and $\Pi_e := \{\mathcal{R}_1 \mid \mathcal{C}\}$, then

$$\begin{aligned} \Pi_d &:= \mathrm{der}_P(\Pi_0, \mathcal{G}) := \mathrm{TDO}(\mathrm{refine}(\Pi_0, \mathcal{I}), \mathcal{G}), \\ &:= \{1,2,3 \mid 0,4,5 \mid 6,7 \mid 8,11,14 \mid 12,13,15,16,17,18 \mid 9,10,19\}. \end{aligned}$$

The partitioned incidence matrix A $(\mathrm{der}_P(\mathcal{G}))$ and the TDO-scheme with respect to Π_d are displayed in Figure 6.1.4.

■

The partition Π_d of Figure 6.1.4 is preserved under $G = Aut(\mathcal{G})$ in the

	8	11	14	12	13	15	16	17	18	9	10	19
1		X		X	X							
2			X			X	X					
3	X							X	X			
0	X									X	X	
4		X								X		X
5			X								X	X
6				X		X		X				
7					X		X		X			

$\rightarrow$	3	6	3
3	1	2	0
3	1	0	2
2	0	3	0

Figure 6.1.4: The $\mathrm{der}_P(\mathcal{G})$ and its TDO-scheme.

sense that $G_{\Pi_d} = G$. In fact, the automorphism group of $\mathcal{G}$ of Example 6.1.1 is as follows.

$$\begin{aligned} Aut(\mathcal{G}) = \langle\ & (6\ \ 7)\ (12\ \ 13)\ (15\ \ 16)\ (17\ \ 18), \\ & (0\ \ 5)\ (2\ \ 3)\ (8\ \ 14)\ (9\ \ 19)\ (15\ \ 17)\ (16\ \ 18), \\ & (1\ \ 2)\ (4\ \ 5)\ (9\ \ 10)\ (11\ \ 14)\ (12\ \ 15)\ (13\ \ 16)\ \rangle. \end{aligned}$$

Moreover, the partition of orbits of G on $\mathcal{G}$ is

$$\Big\{\{1,2,3\},\{0,4,5\},\{6,7\},$$
$$\{8,11,14\},\{12,13,15,16,17,18\},\{9,10,19\}\Big\},$$

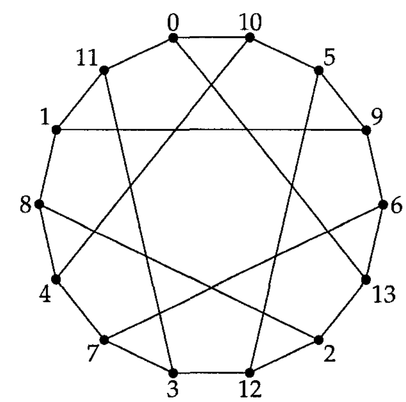
as they appear in $\mathrm{der}_P(\mathcal{G})$ of Figure 6.1.4.

Some of the following examples are taken from Table 4.2 on page 102. We show the graph $\mathcal{G} = (P, \mathcal{B})$ and its incidence matrix A along with its $\mathrm{der}_P(\mathcal{G})$ and its TDO-scheme.

It is not always true that the derived TDO gain more information about the orbits of the automorphism group. Example 6.1.2 shows the Heawood graph whose automorphism group is transitive on the vertices and thus no approximations to the automorphism group orbits are done.

On the other hand, there are some example where the derived TDO is different from the automorphism group orbits partition. Example 6.1.4 shows such an example.

Example 6.1.2. Let $\mathcal{H}$ be the Heawood graph (a $(3,6)$-cage) shown along with its incidence matrix, denoted by A, in Figure 6.1.5.



	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
0	x	x	x																		
1				x	x	x															
2							x	x	x												
3										x	x	x									
4													x	x	x						
5																x	x	x			
6																			x	x	x
7										x			x						x		
8				x			x							x							
9					x											x				x	
10	x														x		x				
11		x				x					x										
12								x				x						x			
13			x						x												x

Figure 6.1.5: The Heawood Graph $\mathcal{H}$ and its incidence matrix A.

Then, the derived TDO is going to be identical to the matrix A of Figure 6.1.5, where the TDO-scheme is

$$\begin{array}{c|c} \rightarrow & 21 \\ \hline 14 & 3 \end{array}$$

Note that, the bipartite graph $BP(F) = \big\{\{P \mid \mathcal{B}\}, \mathcal{E}\big\}$ of Example 2.3.2 whose vertices are the points and the blocks of the Fano plane shown in Figure 2.3.1 is the Heawood graph if we replace the ordered partition $\{P \mid \mathcal{B}\}$ by the unordered partition $\{P, \mathcal{B}\}$. In otherwords, the incidence matrix of Figure

2.3.1 with removing the partition in the rows is identical to the incidence matrix of Figure 6.1.5. Therefore, the Heawood graph $\mathcal{H}$ is $(\{P, \mathcal{B}\}, \mathcal{E})$ where $\{p, B\} \in \mathcal{E}$ only if $p \in B$ in the Fano plane of Figure 2.3.1.

As a result, the automorphism group of $\mathcal{H}$ is generated by the set $\{S \cup \{\alpha\}\}$ where S is the generating set of Example 2.3.2, and

$$\begin{aligned}\alpha =&(0,7)(1,8)(2,9)(3,10)(4,11)(5,12)(6,13)(14,23)(15,26)\\&(16,32)(18,20)(19,27)(21,29)(22,33)(24,28)(25,30).\end{aligned}$$

Therefore, $Aut(\mathcal{H})$ is of order $2 \times 168 = 336$. This is because of the fact that the points and blocks of the Fano plane can be mapped to each other via α.

The partition of orbits of G on $\mathcal{H}$ is

$$\Big\{\{0, 1, \ldots, 13\}, \{14, 15, \ldots, 35\}\Big\}.$$

■

Example 6.1.3. Let $\mathcal{G} = (P, \mathcal{B})$ be the cubic graph of order 24 with girth 7 whose incidence matrix given in Figure 6.1.6. Note that, this graph is called the McGee graph [60] which is a $(3, 7)$-cage.

Let $\Pi_0 = \{\mathcal{R} \mid \mathcal{C}\}$ where $\mathcal{R}$ and $\mathcal{C}$ are partitions of the vertices and edges of the graph $\mathcal{G}$, respectively.

Therefore, applying the same idea of the derived TDO as in Example 6.1.1 on graph $\mathcal{G}$, we compute TDO(refine(Π_0, e_x), $\mathcal{G}$) for all $x \in \mathcal{R}$. Therefore, we get the following TDO-scheme

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

Figure 6.1.6: An incidence matrix of the graph $\mathcal{G}$ of Example 6.1.3.

→	1	2	1	2	2	1	1	1	2	2	2	2	2	2	2	2	2	1	2	2	2
1	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	1	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0	0	0	0	2	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0
1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	2	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	0
2	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

for points $\{0,1,2,3,5,7,9,10,12,13,14,16,17,19,21,22\}$. While we get the following TDO-scheme

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

Figure 6.1.7: The $\mathrm{der}_P(\mathcal{G})$ of the graph $\mathcal{G}$ of Example 6.1.3.

$\rightarrow$	1	2	2	4	4	4	4	2	4	4	4	1
1	1	2	0	0	0	0	0	0	0	0	0	0
1	1	0	2	0	0	0	0	0	0	0	0	0
2	0	1	0	2	0	0	0	0	0	0	0	0
2	0	0	1	0	2	0	0	0	0	0	0	0
4	0	0	0	1	0	1	1	0	0	0	0	0
4	0	0	0	0	0	1	0	1	1	0	0	0
4	0	0	0	0	1	0	0	0	1	1	0	0
4	0	0	0	0	0	0	1	0	0	1	1	0
2	0	0	0	0	0	0	0	0	0	0	2	1

for points $\{4, 6, 8, 11, 15, 18, 20, 23\}$. Therefore, we get the $\mathrm{der}_P(\mathcal{G})$ shown in Figure 6.1.7. Moreover, it has the following TDO-scheme.

$\rightarrow$	16	16	4
16	2	1	0
8	0	2	1

Note that the automorphism group $Aut(\mathcal{G})$ of the graph $\mathcal{G}$ has order 32 and is generated by

$$\{(0\ 14)(1\ 6)(2\ 13)(3\ 12)(4\ 15)(5\ 9)(7\ 11)(8\ 10)(17\ 22)(18\ 23)$$
$$(24\ 29)(25\ 47)(26\ 48)(27\ 52)(28\ 51)(30\ 56)(31\ 53)(32\ 40)(33\ 59)$$
$$(34\ 45)(35\ 43)(36\ 37)(39\ 57)(41\ 49)(42\ 54)(46\ 58),$$
$$(0\ 11)(2\ 9)(3\ 15)(4\ 12)(5\ 13)(7\ 14)(17\ 23)(18\ 22)(19\ 21)$$
$$(24\ 52)(25\ 31)(26\ 42)(27\ 29)(30\ 58)(32\ 34)(33\ 57)(35\ 49)(39\ 59)$$
$$(40\ 45)(41\ 43)(44\ 50)(46\ 56)(47\ 53)(48\ 54),$$
$$(0\ 9)(1\ 7)(2\ 15)(3\ 6)(4\ 5)(8\ 11)(10\ 12)(13\ 14)(16\ 17)(18\ 19)(20\ 22)(21\ 23)$$
$$(24\ 34)(26\ 58)(28\ 54)(29\ 53)(30\ 49)(31\ 57)(33\ 52)(35\ 51)(36\ 42)$$
$$(37\ 43)(38\ 44)(39\ 45)(40\ 47)(41\ 46)(48\ 56)(50\ 55)\}.$$

where the partition of orbits of $Aut(\mathcal{G})$ on $\mathcal{G}$ is

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

■

Example 6.1.4. Let $C = (P, \mathcal{B})$ be the Steiner triple system $STS(13)$ of order 13 and of automorphism group order 6. Let $\Pi_0 = \{\mathcal{R} \mid \mathcal{C}\}$ where $\mathcal{R} = \{0, 1, \ldots, 12\}$ and $\mathcal{C} = \{13, \ldots, 38\}$ are partitions of the points and blocks of C, respectively.

As in the previous examples, here We split the points x in P that have identical TDO-schemes in TDO(refine(Π_0, e_x), C). After that, we compute $\mathrm{der}_P(\Pi_0, C)$. However, in this example we always get the same TDO-scheme as follows

$\rightarrow$	6	20
1	6	0
12	1	5

	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38
0	X	X	X	X	X	X																				
1	X						X	X	X	X	X															
2		X					X					X	X	X	X											
3			X					X				X				X	X	X								
4				X					X				X			X			X	X						
5	X													X			X		X		X	X				
6					X					X		X								X	X		X			
7		X							X									X				X	X	X		
8			X								X		X								X			X	X	
9						X				X					X	X						X			X	
10				X				X						X									X		X	X
11						X	X										X			X				X		X
12					X						X				X			X	X							X

Figure 6.1.8: An incidence matrix corresponding to the Steiner triple system C.

for each point $x \in \mathcal{R}$ induced by the partition $\mathrm{TDO}(\mathrm{refine}(\Pi_0, e_x), C)$ for each $x \in \mathcal{R}$. Therefore,

$$\mathrm{der}_P(\Pi_0, C) = \Pi_0.$$

Note that the automorphism group is of order 6 and is generated by

$$\{(0\ 12)(2\ 11)(3\ 4)(5\ 8)(7\ 10)(13\ 23)(14\ 38)(15\ 31)$$
$$(16\ 30)(18\ 27)(20\ 21)(24\ 32)(25\ 29)(26\ 36)(34\ 37),$$
$$(0\ 6\ 12)(1\ 2\ 11)(3\ 10\ 5)(4\ 8\ 7)(13\ 24\ 38)(14\ 32\ 23)(15\ 35\ 31)$$
$$(16\ 33\ 30)(18\ 22\ 27)(20\ 26\ 29)(21\ 25\ 36)(28\ 37\ 34)\}.$$

The partitions of orbits of the automorphism group on (the points of) C is

$$\{\{0,6,12\},\{1,2,11\},\{3,4,5,7,8,10\},\{9\}\}$$

$\blacksquare$

6.2 Computing Canonical Forms and Automorphism Groups

In this section, we consider some techniques that might be used in order to compute a canonical labeling map φ as defined in Definition 2.4.14 and an automorphism group of a given incidence structure $\mathcal{X}$.

If $\mathcal{X} = (P, \mathcal{B})$ is an incidence structure with an initial partition $\Pi_0 = \{\mathcal{R} \mid \mathcal{C}\}$, then in the case of Example 6.1.1 there are a number of techniques that can be employed to get a discrete partition with respect to Π_d. All these techniques "destroy" automorphisms in the sense that the stabilizer of the refined incidence structure might be smaller than $Aut(\mathcal{X})$.

One particular technique is called **point stabilizing procedure**. The idea is that there exists a class $C \in \Pi_d$ with $|C| > 1$, say $C = \{x_0, x_1, \ldots, x_s\}$. Then, there is some i such that x_i is supposed to be fixed in the first place of the class C, i.e. C is splitted to $\{x_i \mid C\backslash\{x_i\}\}$. We do this for every $x_i \in C$ in turn, and then do the same idea recursively. This idea is explained in more details for graphs in Kocay [54].

Let $b_0, b_1, \ldots, b_r$ be the set of points that have been stabilized in the procedure described above to reach a discrete partition. Then, the idea of constructing the automorphism group $G = Aut(\mathcal{X})$ can be done by induction on a chain of point stabilizing subgroups. That is,

$$G \geq G_{b_0} \geq G_{b_0,b_1} \geq G_{b_0,b_1,b_2} \geq \cdots \geq G_{b_0,b_1,\ldots,b_r} = 1,$$

where $G_{b_0,b_1,\ldots,b_{k-1}} = G^{(k)} = G_{\Pi_{b_0,b_1,\ldots,b_{k-1}}}$ for some $k \leq r$, by Lemma 5.2.3. Note that one can construct $G^{(k)}$ from $G^{(k+1)}$ by finding representatives for the coset $G^{(k+1)}$ in $G^{(k)}$. See [38, 28] for further details.

Lemma 6.2.1. *Let $\mathcal{X} = (P, \mathcal{B})$ be an incidence structure with a partition Π and $G = Aut(\mathcal{X})$. Let $S = P \cup \mathcal{B}$ and let $x_0, x_1, \ldots, x_s$ and $y_0, y_1, \ldots, y_t$ be in S. If $TDOS(\Pi_{x_0,\ldots,x_s}, \mathcal{X}) = TDOS(\Pi_{y_0,\ldots,y_s}, \mathcal{X})$, and $\Pi_{x_0,\ldots,x_s}$ and $\Pi_{x_0,\ldots,x_s}$ are* *discrete partitions, then* $\sigma(\Pi_{x_0,\ldots,x_s})\sigma(\Pi_{y_0,\ldots,y_s})^{-1} \in G_\Pi$ *where* $\sigma(\Pi_{x_0,\ldots,x_s})$ *and* $\sigma(\Pi_{y_0,\ldots,y_s})$ *are two permutations induced by the defined ordering of* $\Pi_{x_0,\ldots,x_s}$ *and* $\Pi_{y_0,\ldots,y_s}$*, respectively.*

Consider Algorithm 6.2.1 denoted by PARTITION BACKTRACK[1] with inputs $\mathcal{X} = P \cup \mathcal{B}$ an incidence structure on p points and b blocks, $\Pi = \{\mathcal{R} \mid \mathcal{C}\}$ is an initial partition of points and blocks of $\mathcal{X}$, and A is an incidence matrix corresponding to $\mathcal{X}$. We say that $\tau \in G$ is best ordering if A^τ is in greatest form of A with respect to lexicographical ordering. In the algorithm, we store such element in τ so that when the algorithm terminates, $\varphi(A) := \tau$ is the canonical labeling map and $\rho(A) := A^{\varphi(A)}$ is the canonical form of A.

Note that, if g and h are two elements of a group G acting on $\mathcal{X}$ such that $A^g = A^h$, then $gh^{-1} \in Aut(\mathcal{X})$.

Algorithm 6.2.1 PARTITIONBACKTRACK[1]$(\Pi_i, \mathcal{X})$

Let $S = P \cup \mathcal{B}$ and $\Pi_{i+1} := \mathrm{TDO}(\Pi_i, \mathcal{X})$.

1. if Π_{i+1} is discrete, then
   - Π_{i+1} defines an ordering, say $g = \sigma(\Pi_{i+1})$, on the points in S.
   - compare A^g with A^τ where τ is the best ordering found so far (if any).
   - If $A^g \prec A^\tau$, $A^g = A^\tau$, or $A^\tau \prec A^g$, then ignore g, an automorphism $g\tau^{-1}$ has been found, or replace τ by g, respectively.
2. else Π_{i+1} is not discrete. Let C be the first non-singleton class (or any other desirable class) of Π_{i+1}.
3. for every $y \in C$ in turn
   - compute $\Pi_{i+1,y} := \mathrm{TDO}(\mathrm{refine}(\Pi_{i+1}, e_y), \mathcal{X})$.
   - call PARTITIONBACKTRACK[1]$(\Pi_{i+1,y}, \mathcal{X})$

Lemma 6.2.2. *Let $\mathcal{X} = (P, \mathcal{B})$ be an incidence structure with a partition Π (which may be tactical). Let $G \leq Aut(\mathcal{X})$ and let $S = P \cup \mathcal{B}$ and $x, y \in S$. If $x, y \in C$ where C is a non-singleton class of Π, and x and y are in the same orbit of G_Π, then*

$$TDOS(\Pi_x, \mathcal{X}) = TDOS(\Pi_y, \mathcal{X}).$$

Proof. Let $g \in G_\Pi$ such that $x^g = y$. If z is incident to k points of cell $D \in \Pi$, then z^g will be also incident to k points in $D^g = D \in \Pi^g$, and so on. Thus, cells of Π_x can be mapped by g to those of Π_y at every refinement step. Thus, $\Pi_y = \Pi_x^g$. Therefore, $\text{TDOS}(\Pi_x, \mathcal{X}) = \text{TDOS}(\Pi_y, \mathcal{X})$. $\square$

By Lemmas 5.2.3, 6.2.2, and 6.2.1, Algorithm 6.2.1 can be modified so that discovered automorphisms can be employed during the algorithm.

We first introduce some notations. Let $\mathcal{X} = (P, \mathcal{B})$ be an incidence structure with a partition Π_i and a group acting on $\mathcal{X}$. Let **found first leaf** be a boolean variable with possible values "TRUE" or "FALSE" which indicates that we already found an ordering g so that we compare it with every other ordering we found later on in the algorithm. Let $\tau \in G$ denote the **best ordering** found so far. That is, $A^\alpha \preceq A^\tau$ for all previously orderings α found in the algorithm. If $\tau = \sigma(\Pi_q)$, then Π_q is called the **canonical node** in the search. Moreover, B is a tuple that stores the stabilized points in the search when we reached the canonical node.

Let H be the group generated by all automorphisms discovered during the search. Initially, $H = \langle\ 1\ \rangle$.

If $x \in \Pi_i$, we say that x is in **level** i and we store a hash value for each x at that level by using Algorithm 5.1.1 which computes $h_i(x) := h(\mathcal{I}(x))$ (see Section 2.9 on page 49), for instance. If $\Pi_{i,x}$ and $\Pi_{i,y}$ are two partitions resulting by stabilizing x and y in the same level i, then we say that $\Pi_{i,x}$ is better than $\Pi_{i,y}$, if $h_i(y) < h_i(x)$.

Algorithm 6.2.2 PARTITIONBACKTRACK$^2(\Pi_i, \mathcal{X})$

Let $S = (P \cup \mathcal{B})$.

1. $\Pi_{i+1} := \mathrm{TDO}(\Pi_i, \mathcal{X})$.

2. If Π_{i+1} is discrete, go to step (3). Otherwise, go to step (4).

3. If found first leaf = FALSE, then go to step (6). Otherwise, let $g := \sigma(\Pi_{i+1})$ and compare A^g with A^τ where τ is the best ordering found so far, and go to step (7).

4. Π_{i+1} is not discrete. Let C be the first non-singleton class of Π_{i+1}. For every $x \in C$ in turn, compute $\Pi_{i+1,x} := \mathrm{TDO}(\mathrm{refine}(\Pi_{i+1}, e_x), \mathcal{X})$. If $h_{i+1}(x) \geq h_{i+1}(y)$ for previously fixed y in Π_{i+1}, then go to step (5). Otherwise, dispose (delete) $\Pi_{i+1,x}$ and continue the for loop to consider another point $z \in C \backslash \{x\}$.

5. If x can be mapped by a $g \in H$ to a previously chosen y in Π_{i+1}, then dispose $\Pi_{i+1,x}$ and return to step (4). Otherwise, set $\Pi_{i+2} := \Pi_{i+1,x}$ and recurse by calling PARTITIONBACKTRACK$^2(\Pi_{i+2}, \mathcal{X})$.

6. Set found first leaf = TRUE. Let $\tau := \sigma(\Pi_{i+1})$. That is τ is the best ordering of A. Store the stabilized points in the base B.

7. Cases of comparing the matrices A^τ and A^g:

   - (equal:) an automorphism $g^{-1}\tau$ has been discovered.
   - (better:) then we have found a better ordering. Set found first leaf = FALSE, and go to step (6).
   - (worse:) ignore it.

Algorithm 6.2.2 clearly terminates since the number of iterations is bounded by the number of points and blocks which is finite. Once the algorithm terminates, we have the canonical labeling map stored in τ, and an automorphism group is stored in H (which is generated by all discovered automorphisms during the search). Moreover, the base is stored in B. That is, if $\tau := \sigma(\Pi_{b_0,b_1,\ldots,b_{r-1}})$, where $\Pi_{b_0,b_1,\ldots,b_{r-1}}$ is a discrete partition, then we have $B = (b_0, \ldots, b_{r-1})$ for $b_0, b_1, \ldots, b_{r-1} \in S = P \cup \mathcal{B}$.

We apply Algorithm 6.2.2 on the graph $\mathcal{G}$ of Example 6.1.1 to demonstrate some of the main steps in Algorithm 6.2.2.

Example 6.2.1. Following what we had in Example 6.1.1, let

$$\Pi_0 := \{1,2,3 \mid 0,4,5 \mid 6,7 \mid 8,11,14 \mid 12,13,15,16,17,18 \mid 9,10,19\},$$

be the initial partition of $\mathcal{G}$ and that A an incidence matrix of $\mathcal{G}$ is given in Figure 6.1.4. Clearly, Π_0 is already tactical, and hence we follow the steps in Algorithm 6.2.2 and show some of the steps in what follows.

begin

Initial set of orbits $= \Big\{\{0\},\{1\},\ldots,\{7\},\{8\},\ldots,\{19\}\Big\}$

$\Pi_0 := \{1,2,3|0,4,5|6,7|8,11,14|12,13,15,16,17,18|9,10,19\}$, is equitable.

level 0: fix 1 : compute TDO(refine(Π_0, e_1), $\mathcal{G}$). See Figure 6.2.1.

$\Pi_1 := \{1\,|\,2,3\,|\,4\,|\,0,5\,|\,6,7\,|\,11\,|\,8,14\,|\,12,13\,|\,15,16,17,18\,|\,9,19\,|\,10\}$.

level 1: fix 2 : compute TDO(refine(Π_1, e_2), $\mathcal{G}$). See Figure 6.2.2.

$\Pi_2 := \{1\,|\,2\,|\,3\,|\,4\,|\,5\,|\,0\,|\,6,7\,|\,11\,|\,14\,|\,8\,|\,12,13\,|\,15,16\,|\,17,18\,|\,19\,|\,9\,|\,10\}$.

level 2: fix 6 : compute TDO(refine(Π_2, e_6), $\mathcal{G}$). See Figure 6.2.3.

$\Pi_3 := \{1\,|\,2\,|\,3\,|\,4\,|\,5\,|\,0\,|\,6\,|\,7\,|\,11\,|\,14\,|\,8\,|\,12\,|\,13\,|\,15\,|\,16\,|\,17\,|\,18\,|\,19\,|\,9\,|\,10\}$.

Here we have found the first leaf node (i.e. found first leaf = TRUE). So a base $B = (1,2,6)$ would be stored and $\tau := \sigma(\Pi_3)$ is also stored as the best ordering. Here we show τ as a list of elements, as they appear in Π_3, as follows

$$\tau := [1,2,3,4,5,0,6,7,11,14,8,12,13,15,16,17,18,19,9,10].$$

	11	8	14	12	13	15	16	17	18	9	19	10
1	X			X	X							
2			X			X	X					
3		X						X	X			
4	X									X	X	
0		X								X		X
5			X								X	X
6				X		X		X				
7					X		X		X			

Figure 6.2.1: $\Pi_1 := \mathrm{TDO}(\mathrm{refine}(\Pi_0, e_1), \mathcal{G})$.

	11	14	8	12	13	15	16	17	18	19	9	10
1	X			X	X							
2		X				X	X					
3			X					X	X			
4	X									X	X	
5		X								X		X
0			X								X	X
6				X		X		X				
7					X		X		X			

Figure 6.2.2: $\Pi_2 := \mathrm{TDO}(\mathrm{refine}(\Pi_1, e_2), \mathcal{G})$.

	11	14	8	12	13	15	16	17	18	19	9	10
1	X			X	X							
2		X				X	X					
3			X					X	X			
4	X									X	X	
5		X								X		X
0			X								X	X
6				X		X		X				
7					X		X		X			

Figure 6.2.3: $\Pi_3 := \mathrm{TDO}(\mathrm{refine}(\Pi_2, e_6), \mathcal{G})$.

	11	14	8	13	12	16	15	18	17	19	9	10
1	X			X	X							
2		X				X	X					
3			X					X	X			
4	X									X	X	
5		X								X		X
0			X								X	X
7				X		X		X				
6					X		X		X			

Figure 6.2.4: $\Pi_3 := \text{TDO}(\text{refine}(\Pi_2, e_7), \mathcal{G})$.

Now, we backtrack Π_2 to fix 7 instead of 6 in Π_2 at level 2. Therefore, we compute $\Pi_3 := \text{TDO}(\text{refine}(\Pi_2, e_7), \mathcal{G})$. See Figure 6.2.4

$$\Pi_3 := \{1\,|\,2\,|\,3\,|\,4\,|\,5\,|\,0\,|\,7\,|\,6\,|\,11\,|\,14\,|\,8\,|\,13\,|\,12\,|\,16\,|\,15\,|\,18\,|\,17\,|\,19\,|\,9\,|\,10\}.$$

Another discrete partition results. If

$$g := [1, 2, 3, 4, 5, 0, 7, 6, 11, 14, 8, 13, 12, 16, 15, 18, 17, 19, 9, 10],$$

then we compare A^g and A^τ to get identical matrices, and thus an automorphism $\alpha_0 := g^{-1}\tau$ has been discovered.

$$\alpha_0 := (\ 6\ \ 7\ )\ (\ 12\ \ 13\ )\ (\ 15\ \ 16\ )\ (\ 17\ \ 18\ ).$$

One also can compute α_0 by comparing the orderings of the points $0, 1, \ldots, 19$ in g and τ.

All choices in cell $\{6, 7\} \in \Pi_2$ have been made, and thus we backtrack in $\{2, 3\} \in \Pi_1$. This time we fix 3 instead of 2.

$\Pi_1 := \{1\,|\,2, 3\,|\,4\,|\,0, 5\,|\,6, 7\,|\,11\,|\,8, 14\,|\,12, 13\,|\,15, 16, 17, 18\,|\,9, 19\,|\,10\}$.

level 1: fix 3 : compute $\text{TDO}(\text{refine}(\Pi_1, e_3), \mathcal{G})$. See Figure 6.2.5.

$\Pi_2 := \{1\,|\,3\,|\,2\,|\,4\,|\,0\,|\,5\,|\,6, 7\,|\,11\,|\,8\,|\,14\,|\,12, 13\,|\,17, 18\,|\,15, 16\,|\,9\,|\,19\,|\,10\}$.

level 2: fix 6 : compute $\text{TDO}(\text{refine}(\Pi_2, e_6), \mathcal{G})$. See Figure 6.2.6.

	11	8	14	12	13	17	18	15	16	9	19	10
1	X			X	X							
3		X				X	X					
2			X					X	X			
4	X									X	X	
0		X								X		X
5			X								X	X
6				X		X		X				
7					X		X		X			

Figure 6.2.5: $\Pi_2 := \mathrm{TDO}(\mathrm{refine}(\Pi_1, e_3), \mathcal{G})$.

	11	8	14	12	13	17	18	15	16	9	19	10
1	X			X	X							
3		X				X	X					
2			X					X	X			
4	X									X	X	
0		X								X		X
5			X								X	X
6				X		X		X				
7					X		X		X			

Figure 6.2.6: $\Pi_3 := \mathrm{TDO}(\mathrm{refine}(\Pi_2, e_6), \mathcal{G})$.

$\Pi_3 := \{1\,|\,3\,|\,2\,|\,4\,|\,0\,|\,5\,|\,6\,|\,7\,|\,11\,|\,8\,|\,14\,|\,12\,|\,13\,|\,17\,|\,18\,|\,15\,|\,16\,|\,9\,|\,19\,|\,10\}$.

Let $g := [1, 3, 2, 4, 0, 5, 6, 7, 11, 8, 14, 12, 13, 17, 18, 15, 16, 9, 19, 10]$. We compare again A^g and A^τ to get the same matrices. Thus, an automorphism $\alpha_1 := g^{-1}\tau$ has been found.

$$\alpha_1 := (\ 0 \ \ 5\ )\ (\ 2 \ \ 3\ )\ (\ 8 \ \ 14\ )\ (\ 9 \ \ 19\ )\ (\ 15 \ \ 17\ )\ (\ 16 \ \ 18\ ).$$

Note that we do not fix 7 in Π_2 again. This is because 6 can be mapped to 7 by α_0 and by Lemma 6.2.2, fixing 7 is equivalent to fixing 6 in the same partition. So, we backtrack in Π_0 to fix 2 instead of 1 as follows.

$\Pi_0 := \{1, 2, 3|0, 4, 5|6, 7|8, 11, 14|12, 13, 15, 16, 17, 18|9, 10, 19\}$.

	14	8	11	15	16	12	13	17	18	10	19	9
2	X			X	X							
1			X			X	X					
3		X						X	X			
5	X									X	X	
0		X								X		X
4			X								X	X
6				X		X		X				
7					X		X		X			

Figure 6.2.7: $\Pi_1 := \mathrm{TDO}(\mathrm{refine}(\Pi_0, e_2), \mathcal{G})$.

	14	11	8	15	16	12	13	17	18	19	10	9
2	X			X	X							
1		X				X	X					
3			X					X	X			
5	X									X	X	
4		X								X		X
0			X								X	X
6				X		X		X				
7					X		X		X			

Figure 6.2.8: $\Pi_2 := \mathrm{TDO}(\mathrm{refine}(\Pi_1, e_1), \mathcal{G})$.

level 0: fix 2 : compute $\mathrm{TDO}(\mathrm{refine}(\Pi_0, e_2), \mathcal{G})$. See Figure 6.2.7.

$\Pi_1 := \{2 \,|\, 1,3 \,|\, 5 \,|\, 0,4 \,|\, 6,7 \,|\, 14 \,|\, 8,11 \,|\, 15,16 \,|\, 12,13,17,18 \,|\, 10,19 \,|\, 9\}$.

level 1: fix 1 : compute $\mathrm{TDO}(\mathrm{refine}(\Pi_1, e_1), \mathcal{G})$. See Figure 6.2.8.

$\Pi_2 := \{2 \,|\, 1 \,|\, 3 \,|\, 5 \,|\, 4 \,|\, 0 \,|\, 6,7 \,|\, 14 \,|\, 11 \,|\, 8 \,|\, 15,16 \,|\, 12,13 \,|\, 17,18 \,|\, 19 \,|\, 10 \,|\, 9\}$.

level 2: fix 6 : compute $\mathrm{TDO}(\mathrm{refine}(\Pi_2, e_6), \mathcal{G})$. See Figure 6.2.9.

$\Pi_3 := \{2 \,|\, 1 \,|\, 3 \,|\, 5 \,|\, 4 \,|\, 0 \,|\, 6 \,|\, 7 \,|\, 14 \,|\, 11 \,|\, 8 \,|\, 15 \,|\, 16 \,|\, 12 \,|\, 13 \,|\, 17 \,|\, 18 \,|\, 19 \,|\, 10 \,|\, 9\}$.

Let $g := [2,1,3,5,4,0,6,7,14,11,8,15,16,12,13,17,18,19,10,9]$. Then, $A^g = A^\tau$, and thus an automorphism $\alpha_2 := g^{-1}\tau$ has been found.

$$\alpha_2 := (\ 1\ \ 2\ )\ (\ 4\ \ 5\ )\ (\ 9\ \ 10\ )\ (\ 11\ \ 14\ )\ (\ 12\ \ 15\ )\ (\ 13\ \ 16\ ).$$

Note that 3 and 2 can be mapped to each other by α_1. Therefore, the algorithm

	14	11	8	15	16	12	13	17	18	19	10	9
2	X			X	X							
1		X				X	X					
3			X					X	X			
5	X									X	X	
4		X								X		X
0			X								X	X
6				X		X		X				
7					X		X		X			

Figure 6.2.9: $\Pi_3 := \text{TDO}(\text{refine}(\Pi_2, e_6), \mathcal{G})$.

terminates here. The best ordering τ is the canonical labeling map such that $\mathcal{G}^\tau$ is the canonical form of $\mathcal{G}$. The base is stored in B, namely $(1, 2, 6)$. That is, $G^{(3)} = G_{1,2,6} = 1$. Moreover, $6^{G_{1,2}} = \{6, 7\}$, where $G^{(2)} = G_{1,2} = \langle\ \alpha_0\ \rangle$. After all, we have

$$\begin{aligned}
G^{(3)} &= G_{1,2,6} = 1, \\
G^{(2)} &= G_{1,2} = \langle\ \alpha_0\ \rangle, \\
G^{(1)} &= G_1 = \langle\ \alpha_0, \alpha_1\ \rangle, \\
G^{(0)} &= G = \langle\ \alpha_0, \alpha_1, \alpha_2\ \rangle.
\end{aligned}$$

Moreover, $6^{G^{(2)}} = \{6, 7\}$, $2^{G^{(1)}} = \{2, 3\}$, and $1^{G^{(0)}} = \{1, 2, 3\}$. Therefore, By the Order-Lemma 2.4.12 on page 30, we have automorphism group order $|G| = 2 \cdot 2 \cdot 3 = 12$.

■

The search in Algorithm 6.2.2 gives rise a backtrack search as seen in Example 6.2.1 equipped with a rooted tree whose root is the initial partition Π_0, and its nodes are all equitable (TDO) partitions in the search. Two nodes Π and Σ are related by a labeled edge if $\Sigma = \text{TDO}(\text{refine}(\Pi, e_x), \mathcal{G})$ with label x for some $x \in \Pi$. Figure 6.2.10 displays a rooted tree corresponding to the search of Example 6.2.1.

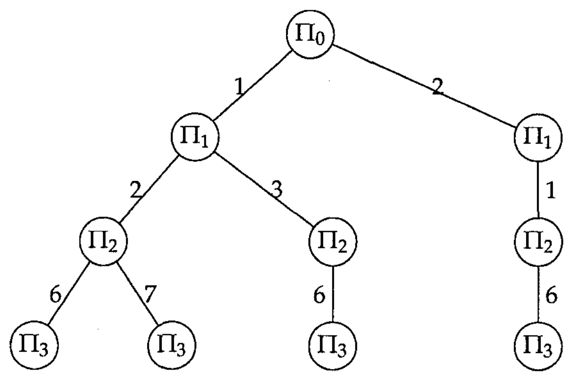


Figure 6.2.10: A rooted tree of the backtrack search of Example 6.2.1.

6.3 McKay's μ-Function

In this section, we consider a definition of the μ-function introduced by B. McKay [64]. Note that the ideas introduced in this section is paraphrasing of those in [64] in different language. The μ-function developed in this section is concerning our interests which are incidence structures. Also that this function relies on the concepts of partition backtracking described in the previous section.

Recall that a μ-function has been defined earlier in the sense of orderly generation techniques due to Faradžev [23] and Read [69], see Section 3.2.

Let G be a group acting on a finite set X of incidence structures on m points and n blocks. Then, the μ-function presented in this section does not depends on the lexicographical ordering as in orderly generation does. It relies on a function $\varphi : X \to G$, which is the canonical labeling map as in Definition 2.4.14, such that for all $x, y \in X$ we have $\rho(x) = \rho(y)$ if and only if x and y are contained in the same G-orbit on X, where $\rho(x) = x^{\varphi(x)}$. Such a function can be realized by the techniques of the partition backtrack discussed in Section 6.2.

In what follows, we may use x and y for incidence structures and at the same time to denote the corresponding incidence matrices. Let $X_0, X_1, \ldots, X_m$

be disjoint subsets of X such that $X = \bigcup_{l=0}^{m} X_l$. Let $R_l \subseteq X_l \times X_{l+1}$ be a G-invariant relation. If x is an incidence matrix with m rows and n columns, then we write $\mathcal{R} = \{0, 1, \ldots, m-1\}$ and $\mathcal{C} = \{m, m+1, \ldots, m+n-1\}$ for the set of row and column indices. For $x = (x_{i,j}) \in X_l$ and $y = (y_{i,j}) \in X_{l+1}$ for all $i \in \mathcal{R}$ and $j \in \mathcal{C}$, we say that $(x, y) \in R_l$ if $x_{i,j} = 1$ implies $y_{i,j} = 1$, and we write $x \prec y$. In particular, the set of pre-images of $y \in X_{l+1}$ follows

$$\pi_2^{-1}(y) := \{(y - (E_i \otimes \mathrm{Row}_i(y)), y) \in R_l \mid i \in \mathrm{Rowsupp}(y)\},$$

where $E_i \in M_{m,1}$ and $y - E_i \otimes \mathrm{Row}_i(y) \in X_l$.

Let

$$t := \min(\mathrm{Rowsupp}(y^{\varphi(y)})), \tag{6.1}$$

and let

$$s := t^{\varphi(y)^{-1}}. \tag{6.2}$$

Figure 6.3.1 describes finding t and s. However, in the figure we assume that t is $\min(\mathrm{Rowsupp}(\rho(y)))$ which is not in general true. This depends on how the refinements in the partition backtrack are done. We call the row index s the **canonical row**.

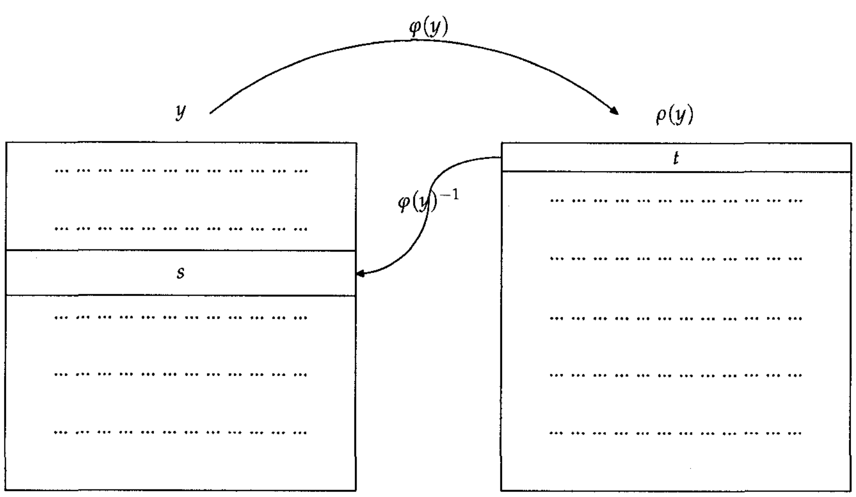


Figure 6.3.1: Finding t and s.

Then, the μ-function is

$$\mu(y) := \Big\{(y - (E_i \otimes \mathrm{Row}_i(y)), y) \in R_l \;\Big|\; E_i \in M_{m,1}, \text{ and } i \in s^{G_y}\Big\}. \qquad (6.3)$$

We call the G_y-orbit of s the **canonical orbit**. Thus $(x, y) \in \mu(y)$ if y is an extension of x and that the last added row in y must contained in the canonical orbit.

Theorem 6.3.1. *Let a group G act on two finite sets of incidence structures X_l and X_{l+1} for $l = 0, 1, \ldots, m$ and $X_l, X_{l+1} \in X = \bigcup_{l=0}^{m} X_l$, and let R_l be a G-invariant relation between X_l and X_{l+1} with $\pi_2(R_l) = X_{l+1}$. Then, for any given $y \in X_{l+1}$, let*

$$\mu(y) = \Big[(y - (E_i \otimes Row_i(y)), y)\Big]^{G_y}$$

where $i \in s^{G_y}$ where s is as defined in 6.2, and $E_i \in M_{m,1}$. Then, μ satisfies the conditions of Definition 3.1.3. In particular, given $\mathcal{T}(G, X_l)$, $\mathcal{T}(G, X_{l+1})$ can be constructed by using (3.6).

Proof. It is clear that $\mu(y)$ is a G_y-orbit on $\pi_2^{-1}(y)$. So, we only show that $\mu(y^g) = \mu(y)^g$ for all $g \in G$. Let $(E_i \otimes \mathrm{Row}_i(y))^g = E_j \otimes \mathrm{Row}_j(y^g)$. Then,

$$\begin{aligned}
\mu(y^g) &= \Big[\ (y^g - (E_j \otimes \mathrm{Row}_j(y^g)), y^g)\ \Big]^{G_{y^g}} \\
&= \Big[\ (y^g - (E_j \otimes \mathrm{Row}_j(y^g)), y^g)\ \Big]^{g^{-1}G_y g} \\
&= \Big[\ (y - (E_j \otimes \mathrm{Row}_j(y^g))^{g^{-1}}, y)\ \Big]^{G_y g} \\
&= \Big(\big[\ (y - (E_i \otimes \mathrm{Row}_i(y)), y)\ \big]^{G_y}\Big)^g = \mu(y)^g.
\end{aligned}$$

Thus 3.6 on page 59 can employed in terms of the μ-function (6.3) to construct a transversal $\mathcal{T}(G, X_{l+1})$, given a transversal $\mathcal{T}(G, X_l)$. □

Applications of McKay's μ-function have been discussed earlier in Example 4.4.2 on page 83 in terms of flag graphs, and its corresponding search poset in terms of incidence matrices as in Figure 4.5.3 on page 88.

In Example 6.3.1, we reconsider Example 3.3.1 on page 64 again with applying the McKay's μ-function given in 6.3 instead of the orderly generation's μ-function.

Example 6.3.1. Following Example 3.3.1, recall that X denotes the class of all finite graphs of order 4 with $V = \{1,2,3,4\}$. Let $G = Sym_{(4)}^{[2]}$ act on X as described in Example 3.3.1. Also, let X_i denote the subclass of X which contains all graphs of order 4 and size i for $i = 0, 1, \ldots, 6$. Thus $X = \bigcup_{i=0}^{6} X_i$. Let $R_i \subseteq X_i \times X_{i+1}$ be a G-invariant relation such that

$$(x, y) \in R_i \iff \text{edge } e \text{ in } x \text{ implies } e \text{ in } y.$$

The construction procedure is applied on an induction on the number of edges and thus we start from the empty graph $A_{0,1}$ of Figure 3.3.6 on page 71 of order 4 and size 0. Then, we consider the extension set $\pi_1^{-1}(A_{0,1})$ as in Figure 3.3.4 on page 69 by choosing only one representative out of each G orbit on $\pi_1^{-1}(A_{0,1})$. This is can be done by considering $G_{A_{0,1}}$-rejection in $\pi_1^{-1}(A_{0,1})$ as in the **lifting orbits step** discussed in Section 3.1 on page 53.

For the **projecting orbits step**, we use the McKay's μ-function defined in this section. The search poset 6.3.2 has two nodes in level 2, namely nodes A and B, which both can be extended to node C. Also, nodes C, D, and E in level 3 can all be extended to node F in level 4. Note that a node in the search poset is representing a G-orbit on X.

In the sense of incidence matrix, a construction procedure can consider the incidence matrix transposed whose rows and columns correspond to edges and vertices of a graph, respectively. To make things easier, we consider the McKay's μ-function as follows. If $(x, y) \in R_i$ for $i = 0, 1, \ldots, 5$, then

$$\mu(y) := \big\{(y \setminus e, y) \in \pi_2^{-1}(y) \mid e \text{ is in } G_y\text{-orbit of } e^*\big\},$$

where e is any edge in y and e^* is the **canonical edge** in y. To compute the canonical edge, we first compute t and s as described in 6.1 and 6.2,

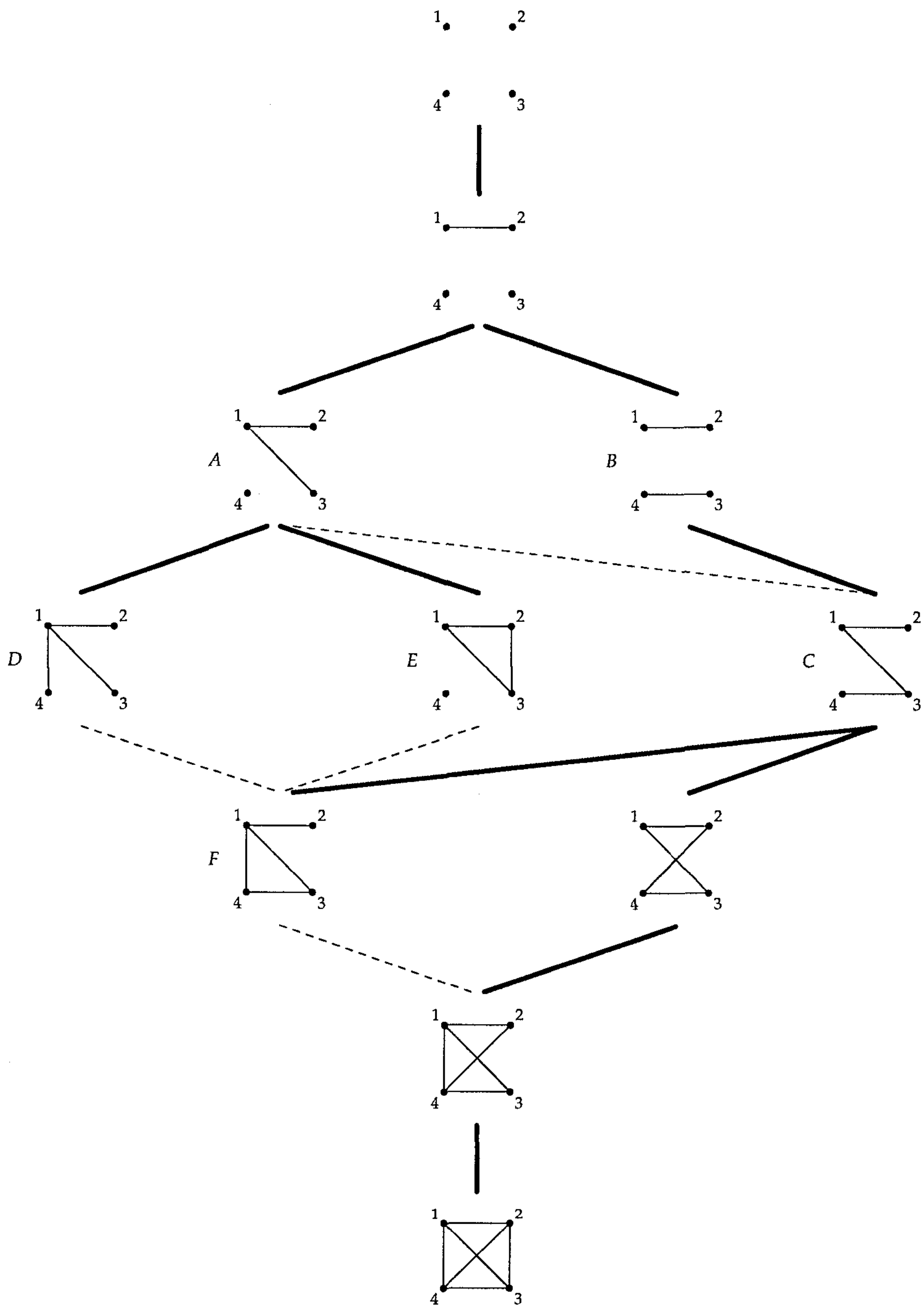


Figure 6.3.2: Search poset for cubic 4 with applying McKay's μ-function.

respectively, and then e^* is the edge corresponds to row s in the transposed incidence matrix of y.

We start now computing $\mu(C)$ to see whether we accept node A or node B in the search poset. First, we compute the preimage set of C as follows

$$\pi_2^{-1}(C) = \Big\{(C \setminus \{3,4\}, C), (C \setminus \{1,3\}, C), (C \setminus \{1,2\}, C)\Big\},$$

where $A = C \setminus \{3,4\}$ and $B = C \setminus \{1,3\}$. Applying a partition backtrack algorithm on C as discussed in Section 6.2, we get $\varphi(C) := (2\ \ 3)$ and $G_C := \langle\ (1\ \ 3)\ (2\ \ 4)\ \rangle$ of order 2. Therefore, $\rho(C) := \big\{\{1,2\},\{1,3\},\{2,4\}\big\}$, and the transposed incidence matrix of $\rho(C)$ is shown in Figure 6.3.3.

	1	2	3	4	
	X	X			$\longleftarrow t$
	X		X		
$\rho(C) =$		X		X	

Figure 6.3.3: The canonical form of C and finding t.

Therefore, row index t corresponds to the edge $\{1,2\}$. So the canonical edge is $\{1,3\} := \{1,2\}^{\varphi(C)^{-1}}$. Therefore,

$$\mu(C) := (C \setminus \{1,3\}, C)^{G_C} = \big\{(C \setminus \{1,3\}, C)\big\}.$$

Thus, we accept the extension of node B and reject the extension of node A.

For level 3, we do the same thing. Then,

$$\pi_2^{-1}(F) = \big\{(F \setminus \{3,4\}, F), (F \setminus \{1,4\}, F), (F \setminus \{1,3\}, F), (F \setminus \{1,2\}, F)\big\}.$$

Computing $\varphi(F)$, we get $\varphi(F) := (2\ \ 4)$ and $\rho(F) = \{\{1,2\},\{1,3\},\{1,4\},\{2,3\}\}$, with $\{1,2\}$ is the minimum element in the canonical orbit. Therefore, the canonical edge is $\{1,4\} := \{1,2\}^{\varphi(F)^{-1}}$. Moreover, $G_F = \langle\ (3\ \ 4)\ \rangle$. Thus,

$$\mu(F) := \big\{(F \setminus \{1,4\}, F)\big\} = \big\{(F \setminus \{1,4\}, F), (F \setminus \{1,3\}, F)\big\}.$$

Therefore, nodes D and E are rejected where node C is accepted.

■

6.4 McKay's Algorithm

In this section, we present McKay's algorithm in the language of [64]. We do this so that it becomes clear that our presentation is roughly a paraphrase of McKay's method with a little change of notation. Mainly, the difference is that we prefer to emphasize the relation between two levels and think of the elements as pairs (x,y). McKay introduces two objects which he calls lower and upper object, respectively. In more detail, the abstract model for McKay's algorithm is as follows.

Let G be a group that acts on a nonempty set X. Elements that are in X are called **labelled objects**, where elements contained in X/G are called **unlabeled objects**. For our convenience, we write $\mathcal{U}$ instead of X/G.

For each labelled object $x \in X$, associate a finite set $L(x)$ of **lower objects** and a finite set $U(x)$ of **upper objects**. We require that for all distinct $x_1, x_2 \in X$ the following six sets are pairwise disjoint.

$$\{x_1\}, \{L(x_1)\}, \{U(x_1)\},$$
$$\{x_2\}, \{L(x_2)\}, \{U(x_2)\}.$$

Let

$$\check{X} = \bigcup_{x \in X} L(x), \quad \text{and} \quad \hat{X} = \bigcup_{x \in X} U(x),$$

for the sets of all lower and upper objects, respectively. Here, the lower and upper objects are connected in the sense of a binary relation $R \subseteq \check{X} \times \hat{X}$, which is accessed using functions $f_1 : \check{X} \to \mathcal{P}(\hat{X})$ and $f_2 : \hat{X} \to \mathcal{P}(\check{X})$ defined by

$$f_1 : \check{y} \mapsto \{\hat{x} \in \hat{X} \,:\, (\check{y}, \hat{x}) \in R\}, \quad f_2 : \hat{x} \mapsto \{\check{y} \in \check{X} \,:\, (\check{y}, \hat{x}) \in R\}.$$

We assume that the group G acts on $X \cup \check{X} \cup \hat{X}$ such that the following axioms satisfied.

C1 G fixes each of X, $\check{X}$, and $\hat{X}$ setwise,

C2 For each $x \in X$ and $g \in G$, $L(x^g) = L(x)^g$ and $U(x^g) = U(x)^g$,

C3 For each $\check{y} \in \check{X}$, $f_1(\check{y}) \neq \emptyset$,

C4 For any $\check{y} \in \check{X}$, $g \in G$, $\hat{x}_1 \in f_1(\check{y})$, and $\hat{x}_2 \in f_1(\check{y}^g)$, $\exists h \in G$ such that $\hat{x}_1^h = \hat{x}_2$,

C5 For any $\hat{x} \in \hat{X}$, $g \in G$, $\check{y}_1 \in f_2(\hat{x})$, and $\check{y}_2 \in f_2(\hat{x}^g)$, $\exists h \in G$ such that $\check{y}_1^h = \check{y}_2$.

Moreover, every $x \in X$ is associated with an **order** $o(x) \in \mathbb{N}$ shared by the elements of $L(x)$ and $U(x)$ so that the following conditions are satisfied.

O1 For each $x \in X$ and $g \in G$, we have $o(x^g) = o(x)$,

O2 For each $\check{x} \in \check{X}$ and $\hat{y} \in f_1(\check{x})$, we have $o(\hat{y}) < o(\check{x})$.

An orbit $S \in \mathcal{U}$ is called **irreducible** if $L(x) = \emptyset$ for each $x \in X$. Otherwise, S is called **reducible**. Note that the consition $L(x) = \emptyset$ is G-invariant, by condition C2. Let $\mathcal{U}_0$ and $\mathcal{U}_1$ denote the set of all irreducible and reducible objects, respectively. Thus, $\mathcal{U} = \mathcal{U}_0 \cup \mathcal{U}_1$.

The final requirement is a function $m : X \to \mathcal{P}(\check{X})$ satisfying the following conditions.

M1 If $L(x) = \emptyset$, then $m(x) = \emptyset$,

M2 If $L(x) \neq \emptyset$, then $m(x) \subseteq L(x)/G_x$,

M3 For each $x \in X$ and $g \in G$, we have $m(x^g) = m(x)^g$.

Lemma 6.4.1. *There is a unique mapping $p : \mathcal{U}_1 \to \mathcal{U}$ satisfying:*

P1 For each $S \in \mathcal{U}_1$, $x \in S$, and $\check{x} \in m(x)$, we have $f_1(\check{x}) \subseteq U(y)$, for some $y \in p(S)$.

Theorem 6.4.2. *Suppose that $x_0 \in S_0 \in \mathcal{U}$ with $o(x_0) \leq n$. Then, calling* **scan**(x_0, n) *outputs exactly one labelled object belonging to each unlabeled object of order at most n, which descendant from S_0.*

Algorithm 6.4.1 **scan**(x: labelled object, n: integer)

if $o(x) \leq n$ **then**
  output x
  for each orbit $A \in U(x)/G_x$ **do**
    select any $\hat{x} \in A$
    if $f_2(\hat{x}) \neq \emptyset$ **then**
      select any $\check{y} \in f_2(\hat{x})$, and suppose that $\check{y} \in L(y)$
      if $o(y) \leq n$ and $\check{y} \in m(y)$ **then**
        scan(y, n)
      end if
    end if
  end for
end if

A modified version of McKay's algorithm which replaces an orbit computations by explicit isomorph testing can be found in [64] along with some examples illustrating the application of **scan**.

In practical applications, providing a suitable μ-function is often the hardest part. Two possible definitions have been already introduced in Lemma 3.2.1 on page 63 and Equation 6.3 on page 143. We also refer to McKay [64] for another definition of the function m.

Chapter 7

The Composition Principle for Linear Spaces

In this chapter we simply consider a class of incidence structures, namely **linear spaces on** 13 **points**, that have been considered by many authors [2, 4, 9, 21, 68].

The enumeration of linear spaces has been started by Doyen [21] when he constructed all of the linear spaces up to 9 points in 1967. More than twenty years later, D. Betten and D. Glynn [26] have constructed (independently) the $5,250$ linear spaces on 10 points. The computation of linear spaces on 11 points was considered by D. Betten and M. Braun [9] when they have developed the TDO method. They found a lower bound for the number of such geometries, without the use of isomorphism tests. In fact, there were only six spaces more. The exact number of linear spaces on 11 points was computed by Ch. Pietsch [68] and (independently) D. Betten together with C. Kuhse: There are $232,929$ linear spaces on 11 points. Finally, the $28,872,973$ linear spaces on 12 points have been constructed by A. Betten and D. Betten [4] in 1999. In this thesis, we consider the construction of linear spaces on 13 points, and thereby a new classification results obtained.

In our algorithm for construction of finite geometries, we make extensive use of **tactical decomposition** (TD) of an incidence matrix representing a given linear space. We use tactical decomposition which is defined by a successive ordering process called **TDO**, introduced by D. Betten and Braun [9], which is essentially an algorithm for computing a good invariant useful for **pre-classification** of the geometries, see [2] for further details.

In order to construct geometries on m points, we go in the opposite direction. That is, given an initial parameter set, we refine these parameters (distribution of lines in the geometry, for instance) step by step. Such refinements eventually stop when a TDO-scheme (or a set of TDO-schemes) is reached. This is because of the fact that TDO-schemes are tactical.

Once a set of TDO-schemes is available, we start trying to construct the related geometries. Note that, a TDO-scheme may be realized, be not realized, or produce several geometries. This method is what we call the **composition method**. Such a method has been considered earlier in constructing linear spaces on 12 points by A. Betten and D. Betten [4]. In [5] the same authors construct the **proper** linear spaces (linear spaces with no lines of length 2) on 17 points.

We remark that the composition priciple was first considered by D. Betten and his student Kaempfer in around 1990. However, Kaempfer never finished his Ph.D. and therefore it was never published. Also, the same method was considered by Volker Widor [82] in his Master's thesis in 2003.

The implementation that was used for our search is due to A. Betten and is different from that used in [4].

7.1 Parameters Refinements

In our construction procedure, we always start with parameters of type 1 (or at depth 1) which describe the **distribution of lines of different lengths**. The generation process is done by taking into account several constraints. Usually the row and column sums for linear space matrices are prescribed. Sometimes, there may be a finer partitioning of rows and columns and the number of incidences is known within the areas of this decomposition. This will allow us to reduce the number of possible matrices considered in the search in many cases. Moreover, incidence matrices with finer partitioning of rows (points) and columns (blocks) are much easier to handle in the process of computing automorphisms and in the isomorphism tests.

Recall Definition 2.3.6 for linear spaces. For a linear space $S = (P, \mathcal{B})$, we write $\{p_0, \ldots, p_{m-1}\}$ for the point set P and $\{B_0, \ldots, B_{n-1}\}$ for the blocks (or lines) set $\mathcal{B}$ where n is a positive integer.

If a point $p \in P$ is in a line $B \in \mathcal{B}$, then we say that p lies on B, that B passes through p, or that p and B are incident, and we write $p \in B$.

The number of lines incident with a fixed point p is called the degree, denoted by $[p]$, where the number of lines of length j passing through p is called the j-degree, denoted by $[p]_j$. Also, we write $|B|$ for the number of points in P lie on B. We say that B is a j-line if $|B| = j$.

In what follows, we consider the refinement procedure for a linear space $S = (P, \mathcal{B})$ with an initial partition $\Pi_0 = \{\mathcal{R} \mid \mathcal{C}\}$, where $\mathcal{R}$ and $\mathcal{C}$ are partitions of the point set P and the block set $\mathcal{B}$, respectively.

Let

$$l_i := \# \text{ of } i\text{-lines in } \mathcal{B}.$$

Then, the vector $\mathbf{l} := (m^{l_m}, \ldots, 3^{l_3}, 2^{l_2})$ is called the **line type** or **parameter of type** $\mathbf{l}$.

Each pair of points in P determines one unique line in $\mathcal{B}$. Thus, counting pairs of points in P, we get

$$\sum_{i=2}^{m} l_i \cdot \binom{i}{2} = \binom{m}{2}. \tag{7.1}$$

Let $a_0, a_1, \ldots, a_{L_1 - 1}$ denote the L_1 lines distributions of non-zero size.

Thus, $\Pi_0 = \{\mathcal{R}, \mathcal{C}\}$ is refined to $\Pi_1 := \{\mathcal{R} \mid C_0 \mid C_1 \mid \ldots \mid C_{L_1-1}\}$. Let β_j denote the length of a line that are contained in C_j for all $j = 0, 1, \ldots, L_1 - 1$. Therefore, $C_0, \ldots, C_{L_1-1} \subseteq \mathcal{C}$ and

$$|C_0| + |C_1| + \ldots + |C_{L_1-1}| = a_0 + a_1 + \ldots + a_{L_1-1} = |\mathcal{C}|.$$

The partition Π_1 can be described by

	C_0	C_1	$\ldots$	C_{L_1-1}
$\mathcal{R}$	$\uparrow$ β_0 $\downarrow$	$\uparrow$ β_1 $\downarrow$	$\ldots$ $\ldots$ $\ldots$	$\uparrow$ β_{L_1-1} $\downarrow$

Let α_j denote the number of β_j-lines in C_j that are incident with a fixed point $p \in P$, for $j = 0, \ldots, L_1 - 1$. Then, clearly

$$\alpha_j \leq a_j. \tag{7.2}$$

As each point in P is joined to each other point, and as each β_j-line joins a fixed point to $\beta_j - 1$ points, we get

$$\sum_{j=0}^{L_1-1} (\beta_j - 1) \cdot \alpha_j = m - 1. \tag{7.3}$$

Let S be a linear space of line type (or line case) $\beta_0^{a_0}, \beta_1^{a_1}, \ldots, \beta_{L_1-1}^{a_{L_1-1}}$. Then, the **point types** $\alpha^{(0)}, \alpha^{(1)}, \ldots, \alpha^{(L_2-1)}$ are all solutions to Equations 7.2 and 7.3, where for any point $p \in P$ of type $\alpha^{(i)}$, we have

$$\alpha^{(i)} = (\alpha_0^{(i)}, \ldots, \alpha_{L_1-1}^{(i)}),$$

for all $0 \leq i \leq L_2 - 1$. Moreover, let **the point type distribution** denoted by b_i be the number of points of type $\alpha^{(i)}$ for all $i = 0, \ldots, L_2 - 1$. The line case together with the point type and point type distribution form the **parameters of the type** 2 of the geometry.

Therefore, Π is refined to

$$\Pi_2 := \{R_0 \mid R_1 \mid \ldots \mid R_{L_2-1} \mid C_0 \mid \ldots \mid C_{L_1-1}\},$$

with $|R_i| = b_i$ for all $i = 0, 1, \ldots, L_2 - 1$. Clearly,

$$\sum_{i=0}^{L_2-1} b_i = m. \tag{7.4}$$

Let $\alpha_{i,j}$ denote the number of β_j lines in C_j that are incident with a point of type $\alpha^{(i)}$, i.e. a point in R_i, for all $0 \le i \le L_2 - 1$ and for all $0 \le j \le L_1 - 1$. Then, Π_2 can be described by

	C_0	C_1	$\ldots$	C_{L_1-1}
R_0	$\leftarrow \alpha_{0,0} \rightarrow$	$\leftarrow \alpha_{0,1} \rightarrow$	$\ldots$	$\leftarrow \alpha_{0,L_1-1} \rightarrow$
R_1	$\leftarrow \alpha_{1,0} \rightarrow$	$\leftarrow \alpha_{1,1} \rightarrow$	$\ldots$	$\leftarrow \alpha_{1,L_1-1} \rightarrow$
$\vdots$	$\vdots$	$\vdots$	$\ldots$	$\vdots$
R_{L_2-1}	$\leftarrow \alpha_{L_2-1,0} \rightarrow$	$\leftarrow \alpha_{L_2-1,1} \rightarrow$	$\ldots$	$\leftarrow \alpha_{L_2-1,L_1-1} \rightarrow$
	$\sum = \beta_0 \cdot a_0$	$\sum = \beta_1 \cdot a_1$	$\ldots$	$\sum = \beta_{L_1-1} \cdot a_{L_1-1}$

Counting incidences in the class C_j of β_j-lines in two different ways, the following equation hold for each $0 \le j \le L_1 - 1$

$$\sum_{i=0}^{L_2-1} \alpha_{i,j} \cdot b_i = \beta_j \cdot a_j. \tag{7.5}$$

We next refine Π_2 into a finer partition Π_3 which is called the **refined line type**.

For each class $C_J \in \Pi_2$, where $J = 0, 1, \ldots, L_1 - 1$, let

$$\gamma_{J,i} = \# \text{ of points of type } \alpha^{(i)} \text{ on a } \beta_J\text{-line},$$

for all $i = 0, 1, \ldots, L_2 - 1$. Then, clearly

$$\gamma_{J,i} \le b_i \text{ for all } i = 0, \ldots, L_2 - 1. \tag{7.6}$$

Also, for each class C_J where $J = 0, 1, \ldots, L_1 - 1$ we have

$$\sum_{i=0}^{L_2-1} \gamma_{J,i} = \beta_J. \tag{7.7}$$

We want to solve Equation 7.7 restricted to Equation 7.6. Assume that $\gamma_J^{(0)}, \gamma_J^{(1)}, \ldots, \gamma_J^{(l_J-1)}$ are all the solutions of class C_J, for $J = 0, 1, \ldots, L_1 - 1$. This is called the **refined line type** of block J. Also, let $a_{J,j}$ denote the nonzero number of β_j-lines that are in class $C_J \in \Pi_2$. Thus $a_{J,0}, \ldots, a_{J,l_J-1}$ is called the **refined line distribution**.

For $i = 0, 1, \ldots, L_2 - 1$, let $\gamma_{J,i}^{(j)}$ denote the number of points of type $\alpha^{(i)}$ that are incident with β_j-lines in C_J.

Therefore, the partition Π_2 is refined to the following partition

$$\Pi_3 := \{R_0 \mid \ldots \mid R_{L_2-1} \mid D_{0,0} \mid \ldots \mid D_{0,l_0-1} \mid D_{1,0} \mid \ldots \mid D_{L_1-1,0} \mid \ldots \mid D_{L_1-1,l_{L_1-1}-1}\},$$

where $D_{J,0}, \ldots, D_{J,l_J-1} \subseteq C_J$ for all $J = 0, 1, \ldots, L_1 - 1$. Π_3 can be described by

	$\ldots$	$\Longleftarrow C_J \Longrightarrow$				$\ldots$
	$\ldots$	$D_{J,0}$	$D_{J,1}$	$\ldots$	D_{J,l_J-1}	$\ldots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
R_i	$\vdots$	$\uparrow$ $\gamma_{J,i}^{(0)}$ $\downarrow$	$\uparrow$ $\gamma_{J,i}^{(1)}$ $\downarrow$	$\ldots$	$\uparrow$ $\gamma_{J,i}^{(l_J-1)}$ $\downarrow$	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Given a class C_J for $J = 0, 1, \ldots, L_1 - 1$, it is clear that

$$\sum_{j=0}^{l_J-1} a_{J,j} = a_J. \tag{7.8}$$

In other words,

$$| \, C_J \, | \; = \; \sum_{j=0}^{l_J-1} | \, D_{J,j} \, |.$$

Moreover, for $i = 0, 1, \ldots, L_2 - 1$, we count incidences in two different ways to get

$$\sum_{j=0}^{l_J-1} \gamma_{J,i}^{(j)} \cdot a_{J,j} = \alpha_{i,J} \cdot b_i. \tag{7.9}$$

Now, we consider the connections between two fixed points in P in two cases. Namely, for $x \in R_{i_1}$ and $y \in R_{i_2}$ there are two cases for when $i_1 = i_2$ and $i_1 \neq i_2$ for all $0 \leq i_1, i_2 \leq L_2 - 1$. First, assume that $x, y \in R_{i_1}$, i.e. both points x and y are of type $\alpha^{(i)}$. Then, for each $i = 0, 1, \ldots, L_2 - 1$ we have

$$\sum_{J=0}^{L_1-1} \sum_{j=0}^{l_J-1} (\ \gamma_{J,i}^{(j)} - 1\ ) a_{J,j} = b_i - 1, \tag{7.10}$$

while if x and y are in different classes R_{i_1} and R_{i_2}, we have

$$\sum_{J=0}^{L_1-1} \sum_{j=0}^{l_J-1} \gamma_{J,i_1}^{(j)} \cdot \gamma_{J,i_2}^{(j)} \cdot a_{J,j} = b_{i_1} \cdot b_{i_2}. \tag{7.11}$$

7.2 Generation of Linear Spaces

In this section, we aim our attention to two tasks. First, we give an example of linear spaces generation. We assume that a TDO-scheme has been constructed using the techniques discussed in Section 7.1 and that our goal is to construct all possible incidence matrices which have such scheme. Second, we apply the algorithm presented in this thesis on the construction of designs, namely we try to construct 2-designs given parameters 2-(v, k, λ). Note that this algorithm can also applied to the construction of t-(v, k, λ) designs in general.

For the first part, assume that we want to construct linear spaces on 13 points with a given line type $(5^3\ \ 3^{16})$. Assume furthermore that we used the techniques for parameters refinement presented in Section 7.1 to produce the following row and column tactical decompositions.

$$\begin{array}{c|cc} \rightarrow & 3 & 16 \\ \hline 1 & 3 & 0 \\ 12 & 1 & 4 \end{array} \qquad \begin{array}{c|cc} \downarrow & 3 & 16 \\ \hline 1 & 1 & 0 \\ 12 & 4 & 3 \end{array} \tag{7.12}$$

Note that in the parameters refinement, we use some techniques which are able to reduce the number of candidates of TDO-schemes that might be not realizable. We refer to A.Betten and D.Betten [4] for more details. The initial partition which corresponds to 7.12 is given by $\Pi := \{R_0 \mid R_1 \mid C_0 \mid C_1\}$, where $R_0 = \{0\}$, $R_1 = \{1, \ldots, 12\}$, $C_0 = \{13, 14, 15\}$, and $C_1 = \{16, \ldots, 31\}$. Also, let $\mathcal{R} = R_0 \cup R_1$ and $\mathcal{C} = C_0 \cup C_1$. Here, if $A \in M_{13,19}$ is an incidence matrix corresponds to an incidence structure which satisfies the decomposition given in 7.12, then for $i \in \mathcal{R}$ and $j \in \mathcal{C}$, A must satisfies the following conditions.

1. $|\text{row}_i(A) \cap C_0|$ equals 3 if $i \in R_0$ and equals 1 if $i \in R_1$.
2. $|\text{row}_i(A) \cap C_1|$ equals 0 if $i \in R_0$ and equals 4 if $i \in R_1$.
3. $|\text{row}_i(A) \cap \text{row}_j(A)| = 1$ for all $i \neq j$ and $i, j \in \mathcal{R}$.
4. $|\text{col}_j(A) \cap R_0|$ equals 1 if $j \in C_0$ and equals 0 if $j \in C_1$.
5. $|\text{col}_j(A) \cap R_1|$ equals 4 if $j \in C_0$ and equals 3 if $j \in C_1$.

Let X_l denote the set of all incidence matrices in $M^{(l)}_{13,19}$ corresponding to the class of linear spaces on 13 points, and let $X = \bigcup_{l=0}^{13} X_l$. Let $G = Sym_{(13)} \times Sym_{(19)}$ act on X and let $R_l \subseteq X_l \times X_{l+1}$ be a G-invariant relation for all $0 \leq l \leq 12$ such that $(A, B) \in R_l$ only if $A = (a_{i,j}) \in X_l$ and $B = (b_{i,j}) \in X_{l+1}$ for all $i \in \mathcal{R}$ and for all $j \in \mathcal{C}$ such that $a_{i,j} = 1$ implies that $b_{i,j} = 1$.

Starting the construction of incidence matrices satisfying 7.12, we start with $A_0 \in X_0$ where we write A_i to denote that $|\text{Rowsupp}(A_i)| = i$, namely A_0 is the incidence matrix whose entries are all zeros. Here, we follow Algorithm 3.1.1 on page 61 in a depth-first search strategy. Thus, we first compute

$$\pi_1^{-1}(A_0) := \big\{(A_0, A_0 + (E_{i_0} \otimes v_{i_0})) \in R_0 | E_{i_0} \in M_{13,1}, v_{i_0} \in M_{1,19}, \text{ and } i_0 \in \mathcal{R}\big\}.$$

Also, we compute G_{A_0} to construct a transversal $\Gamma(A_0) := \mathcal{T}(G_{A_0}, \pi_1^{-1}(A_0))$. Then, we test each element in $\Gamma(A_0)$ for the following test.

$$\text{Is } (A_0, A_0 + (E_{i_0} \otimes v_{i_0})) \in \mu(A_0 + E_{i_0} \otimes v_{i_0})? \tag{7.13}$$

where μ is defined in 6.3 on page 143. If yes then we accept such extension, and we reject it otherwise. Assume that rows $i_0, i_1, \ldots, i_r$ have been constructed to get incidence matrix A_r. Then, A_r must satisfies the following conditions.

1. $|\mathrm{row}_{i_r}(A_r) \cap C_0|$ equals 3 if $i_r \in R_0$ and equals 1 if $i_r \in R_1$.
2. $|\mathrm{row}_{i_r}(A_r) \cap C_1|$ equals 0 if $i_r \in R_0$ and equals 4 if $i_r \in R_1$.
3. $|\mathrm{row}_{i_j}(A_r) \cap \mathrm{row}_{i_r}(A_r)| = 1$ for all $0 \le j < r$ and $i_j, i_r \in \mathrm{Rowsupp}(A_r)$.
4. $|\mathrm{col}_j(A_r) \cap R_0| \le 1$ if $j \in C_0$ and ≤ 0 if $j \in C_1$.
5. $|\mathrm{col}_j(A_r) \cap R_1| \le 4$ if $j \in C_0$ and ≤ 3 if $j \in C_1$.

Then, we extend A_r as follows. We compute

$$\pi_1^{-1}(A_r) := \{(A_r, A_r + (E_{i_r} \otimes v_{i_r})) \in R_r | E_{i_r} \in M_{13,1}, v_{i_r} \in M_{1,19}, \text{ and } i_r \in \mathcal{R}\}.$$

Again we compute the automorphism group G_{A_r} to construct a transversal $\Gamma(A_r) := \mathcal{T}(G_{A_r}, \pi_1^{-1}(A_r))$, and do the Test 7.13 for every element in $\Gamma(A_r)$. We then consider elements accepted in the test, and ignore them otherwise.

When the algorithm is terminated, two incidence matrices are constructed corresponding to two nonisomorphic incidence structures. Figure 7.2.1 displays the constructed linear spaces, say S_1 and S_2, along with their automorphism group generators (permutations are shown only for rows and not for columns).

In fact those two linear spaces correspond to the **Latin square** of order $n = 4$ (with the three 4-lines intersecting in an additional point namely, point 0 in the matrix) in a way described as follows: Skipping the first point, we take the n rows of the Latin square as the first n rows of the incidence matrix of S_1, say. Next, we take the n columns of the Latin square as the next n rows in the incidence matrix and the n digits as the last n rows in the incidence

	⋆	⋆	⋆	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
0	X	X	X																
1	X			X	X	X	X												
2	X							X	X	X	X								
3	X											X	X	X	X				
4	X															X	X	X	X
5		X		X				X				X				X			
6		X			X				X				X				X		
7		X				X				X				X				X	
8		X					X				X				X				X
9			X	X					X					X					X
10			X		X			X							X			X	
11			X			X					X	X					X		
12			X				X			X			X			X			

$Aut(S_1) = \{(5,\ 7)(6,\ 8)(11,\ 12)$

$,(3,\ 5)(4,\ 6)(10,\ 11)$

$,(2,\ 4)(6,\ 8)(9,\ 10)(11,\ 12)$

$,(2,\ 9)(4,\ 10)(6,\ 11)(8,\ 12)$

$,(1,\ 2)(3,\ 4)(5,\ 6)(7,\ 8)\}$

$|Aut(S_1)| = 576$

	⋆	⋆	⋆	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
0	X	X	X																
1	X			X	X	X	X												
2	X							X	X	X	X								
3	X											X	X	X	X				
4	X															X	X	X	X
5		X		X				X				X				X			
6		X			X				X				X				X		
7		X				X				X				X				X	
8		X					X				X				X				X
9			X	X					X					X					X
10			X		X			X							X			X	
11			X			X					X		X			X			
12			X				X			X		X					X		

$Aut(S_2) = \{(3,\ 5)(4,\ 6)(10,\ 11)$

$,(2,\ 4,\ 8,\ 6)(9,\ 10,\ 12,\ 11)$

$,(2,\ 9)(4,\ 11)(6,\ 10)(8,\ 12)$

$,(1,\ 2)(3,\ 4)(5,\ 6)(7,\ 8)(10,\ 11)\}$

$|Aut(S_2)| = 192$

Figure 7.2.1: Two incidence matrices corresponding to linear spaces S_1 (top) and S_2 (bottom) along with their automorphism group generators.

matrix. Now, each of the n^2 elements of the Latin square defines three points, namely the number of its row, the number of its column, and the digit of this element. Hence, we get n^2 3-blocks. The following figure illustrates the two Latin squares corresponding to the linear spaces above, namely S_1 (top) and S_2 (bottom), respectively. Note that we skip the first point. Moreover, the row and column indices correspond to the point indices in the above two incidence matrices, see [3] for more explanation.

	5	6	7	8
1	1	2	3	4
2	2	1	4	3
3	3	4	1	2
4	4	3	2	1

	5	6	7	8
1	1	2	3	4
2	2	1	4	3
3	4	3	1	2
4	3	4	2	1

Figure 7.2.2: The two Latin squares correspond to the above linear spaces S_1 and S_2 in Figure 7.2.1 with the same automorphism group orders, respectively.

Now, we turn our attention to the construction of 2-(v, k, λ) designs. The procedure is similar in some sense. Assume that we want to construct the unique 2-$(13, 4, 1)$ design, then by using Equation 2.3 on page 22, one can conclude that $b = 13$ and $r = 4$. Therefore, we do the same as we have done previously with a slight differences. First, if A is an incidence matrix corresponding to some 2-(v, k, λ) design in general, then A must satisfies the following conditions.

1. every row contains exactly r 1's.
2. every column contains exactly k 1's.
3. $|\text{row}_i(A) \cap \text{row}_j(A)| = \lambda$ for all $0 \leq i, j \leq v-1$ and $i \neq j$.

In the row by row generation, we start with a matrix A_0 whose all entries are zeros, and step by step we add one row at a time. Assuming that rows $i_0, i_1, \ldots, i_r$ have been constructed in A_r, A_r must satisfy the following conditions.

1. every row in $\text{Rowsupp}(A_r)$ contains exactly r 1's.
2. every column contains at most k 1's.
3. $|\text{row}_{i_j}(A_r) \cap \text{row}_{i_r}(A_r)| = \lambda$ for all $0 \leq j < r$ and $i_r, i_j \in \text{Rowsupp}(A_r)$.

Here, also we compute an extension set $\pi_1^{-1}(A_r)$ and construct a transversal $\Gamma(A_r) := \mathcal{T}(G_{A_r}, \pi_1^{-1}(A_r))$ to test which elements pass the test 7.13 as discussed in he generation of linear spaces above. Once the algorithm is terminated, only one design is constructed. Figure 7.2.3 shows the constructed 2-$(13, 4, 1)$ design.

	13	14	15	16	17	18	19	20	21	22	23	24	25
0	X	X	X	X									
1	X				X	X	X						
2		X			X			X	X				
3			X			X		X		X			
4	X								X	X	X		
5				X			X	X			X		
6		X				X					X	X	
7			X				X		X			X	
8				X	X					X		X	
9	X							X				X	X
10		X					X			X			X
11			X		X						X		X
12				X		X			X				X

Figure 7.2.3: The 2-$(13, 4, 1)$ design.

The automorphism group of the design of Figure 7.2.3 has order 5616, and is generated by

$$\begin{aligned}\{&(3\ 5)(6\ 10)(7\ 12)(8\ 11)(15\ 16)(18\ 19)(22\ 23)(24\ 25),\\ &(3\ 7)(4\ 9)(5\ 12)(6\ 10)(18\ 19)(20\ 21)(22\ 24)(23\ 25),\\ &(2\ 3)(6\ 11)(7\ 10)(8\ 12)(14\ 15)(17\ 18)(21\ 22)(24\ 25),\\ &(1\ 2)(4\ 10)(5\ 12)(6\ 9)(13\ 14)(18\ 20)(19\ 21)(23\ 25),\\ &(0\ 1)(6\ 11)(7\ 12)(8\ 10)(14\ 17)(15\ 18)(16\ 19)(24\ 25)\}.\end{aligned}$$

7.3 Results of Linear Spaces with 13 points

The results displayed in this section are the classification of linear spaces on 13 points. First of all we would like to note that linear spaces with up to 11 points were tested by the algorithm described in this thesis and the same results were found in [4] where one can view the results on linear spaces on up to 12 points. Linear spaces with 12 points were also partially tested with the same results found in [4].

The search for linear spaces with 13 points took around 45 to 60 days yielding $8,592,194,823$ linear spaces up to isomorphism. The work was divided among 4 machines with two 3.0 GHZ machines, one 2.6 GHZ, and one 2.0 GHZ processor. Such a division in the search is valid using the techniques described in Chapter 3. The amount of storage needed for some line cases with over $100,000,000$ spaces was between 5 and 19 **GB** (!!). We note that the usage of the TDO-algorithm described in Section 7.1 was really of great help in terms of speed. Also, omitting of the 2-lines in the line type during the construction was very useful. In particular, parameters of different kinds were used in the search mostly with depth 2 and 3 and sometimes at depths 4, 5, and 6. The most desirable depth being sought is the depth where the decomposition coincides with its TDO. However, in some cases it was not possible to reach that depth because of the memory problems mentioned in Section 7. In the table below, we indicate some of the cases which were constructed either in depth 4 and 5 or with tactical decomposition with an "$*$" in the "time" column. Later on, we present another table which shows a comparison between the search time used to construct the same line case with parameters of different depths.

In the tables below, we do not show the number of 2-lines in the line case because of the space and since their numbers can be easily recomputed by using 7.1. Empty parentheses represent the complete graph with v vertices, i.e. K_v.

In the following table we display some comparisons between the times needed for constructing the same line or point cases at different depths. The results presented in the following table may lead to future work on parameters refinement as we believe that using such a technique (parameters refinement) helps to reduce the time needed for constructing incidence structures.

The first column in the table below represents the case at a given depth between parentheses, the second column indicates the number of refined cases, the third column indicates the time needed for the refinement, the fourth column presents the number of constructed spaces, and the time needed for construction is displayed in the fifth column. Finally, we add up all the time needed for the construction starting from the line case up to the final result. We always start from a line case and refine to get the point case and so on. The total time presented in the sixth column is the summation of the construction time (fifth column) and the refinement time (third column). In the following table, we use an "$\star$" to repeat the same entry above the current entry. The last column on the right shows the number of spaces constructed in one second, i.e. the number of solutions divided by the total time.

The following table shows some comparisons between point types in depth 2, 4, and some point types in depth 6. We also show the line types in depth 1 which were used to obtain those point types. For instance, the first row entry in the table may viewed as follows: We start from the line case $(3^{15}, 2^{33})$ in depth 1 and in the first refinement several point cases will be produced in depth 2. One of these point cases is the following point case, which is at depth 2:

$$\begin{cases} 1 \times (3^5, 2^2) \\ 4 \times (3^4, 2^4) \\ 8 \times (3^3, 2^6) \end{cases}$$

Then, we refine it twice to get to depth 4. The needed time for refinements

Table 7.1: Linear Spaces on 13 points (Part I)

line case	#sol	time	line case	#sol	time	line case	#sol	time
(13)	1		$(8,3^2)$	6		$(7,4,3^{12})$	65	
(12)	1		$(8,3)$	2		$(7,4,3^{11})$	459	
$(11,3)$	1		(8)	1		$(7,4,3^{10})$	1567	
(11)	1		(7^2)	1		$(7,4,3^9)$	2805	
$(10,4)$	1		$(7,6,3^5)$	1		$(7,4,3^8)$	3157	
$(10,3^3)$	1		$(7,6,3^4)$	1		$(7,4,3^7)$	2433	
$(10,3^2)$	1		$(7,6,3^3)$	1		$(7,4,3^6)$	1428	
$(10,3)$	2		$(7,6,3^2)$	1		$(7,4,3^5)$	658	
(10)	1		$(7,6,3)$	1		$(7,4,3^4)$	252	
$(9,5)$	1		$(7,6)$	2		$(7,4,3^3)$	84	
$(9,4,3^3)$	1		$(7,5,4,3^6)$	4		$(7,4,3^2)$	24	
$(9,4,3^2)$	1		$(7,5,4,3^5)$	5		$(7,4,3)$	8	
$(9,4,3)$	1		$(7,5,4,3^4)$	9		$(7,4)$	2	
$(9,4)$	2		$(7,5,4,3^3)$	5		$(7,3^{15})$	34	
$(9,3^6)$	4		$(7,5,4,3^2)$	4		$(7,3^{14})$	171	
$(9,3^5)$	3		$(7,5,4,3)$	1		$(7,3^{13})$	803	
$(9,3^4)$	6		$(7,5,4)$	1		$(7,3^{12})$	2197	
$(9,3^3)$	5		$(7,5,3^9)$	13		$(7,3^{11})$	3911	
$(9,3^2)$	4		$(7,5,3^8)$	24		$(7,3^{10})$	4470	
$(9,3)$	2		$(7,5,3^7)$	55		$(7,3^9)$	3645	
(9)	1		$(7,5,3^6)$	53		$(7,3^8)$	2296	
$(8,6)$	1		$(7,5,3^5)$	53		$(7,3^7)$	1202	
$(8,5,3^4)$	1		$(7,5,3^4)$	31		$(7,3^6)$	523	
$(8,5,3^3)$	1		$(7,5,3^3)$	18		$(7,3^5)$	195	
$(8,5,3^2)$	1		$(7,5,3^2)$	8		$(7,3^4)$	68	
$(8,5,3)$	1		$(7,5,3)$	5		$(7,3^3)$	22	
$(8,5)$	2		$(7,5)$	2		$(7,3^2)$	7	
$(8,4^2,3^4)$	4		$(7,4^4,3^3)$	3		$(7,3)$	2	
$(8,4^2,3^3)$	2		$(7,4^4,3^2)$	2		(7)	1	
$(8,4^2,3^2)$	3		$(7,4^4,3)$	1		$(6^2,4,3^8)$	2	
$(8,4^2,3)$	1		$(7,4^4)$	1		$(6^2,4,3^7)$	3	
$(8,4^2)$	1		$(7,4^3,3^6)$	19		$(6^2,4,3^6)$	10	
$(8,4,3^7)$	10		$(7,4^3,3^5)$	43		$(6^2,4,3^5)$	8	
$(8,4,3^6)$	15		$(7,4^3,3^4)$	41		$(6^2,4,3^4)$	11	
$(8,4,3^5)$	26		$(7,4^3,3^3)$	27		$(6^2,4,3^3)$	4	
$(8,4,3^4)$	22		$(7,4^3,3^2)$	11		$(6^2,4,3^2)$	3	
$(8,4,3^3)$	18		$(7,4^3,3)$	4		$(6^2,4,3)$	1	
$(8,4,3^2)$	8		$(7,4^3)$	1		$(6^2,4)$	1	
$(8,4,3)$	5		$(7,4^2,3^9)$	90		$(6^2,3^{11})$	2	
$(8,4)$	2		$(7,4^2,3^8)$	318		$(6^2,3^{10})$	7	
$(8,3^{10})$	9		$(7,4^2,3^7)$	619		$(6^2,3^9)$	26	
$(8,3^9)$	18		$(7,4^2,3^6)$	669		$(6^2,3^8)$	40	
$(8,3^8)$	40		$(7,4^2,3^5)$	486		$(6^2,3^7)$	58	
$(8,3^7)$	49		$(7,4^2,3^4)$	257		$(6^2,3^6)$	50	
$(8,3^6)$	54		$(7,4^2,3^3)$	100		$(6^2,3^5)$	41	
$(8,3^5)$	38		$(7,4^2,3^2)$	33		$(6^2,3^4)$	21	
$(8,3^4)$	26		$(7,4^2,3)$	8		$(6^2,3^3)$	12	
$(8,3^3)$	12		$(7,4^2)$	4		$(6^2,3^2)$	6	

Table 7.2: Linear Spaces on 13 points (Part II)

line case	#sol	time	line case	#sol	time	line case	#sol	time
$(6^2,3)$	4		$(6,5,3^5)$	945		$(6,4^2,3^6)$	62316	
(6^2)	2		$(6,5,3^4)$	282		$(6,4^2,3^5)$	17990	
$(6,5^2,3^9)$	4		$(6,5,3^3)$	84		$(6,4^2,3^4)$	4058	
$(6,5^2,3^8)$	9		$(6,5,3^2)$	24		$(6,4^2,3^3)$	784	
$(6,5^2,3^7)$	23		$(6,5,3)$	8		$(6,4^2,3^2)$	135	
$(6,5^2,3^6)$	34		$(6,5$	2		$(6,4^2,3)$	24	
$(6,5^2,3^5)$	28		$(6,4^6,3^3)$	2		$(6,4^2)$	6	
$(6,5^2,3^4)$	18		$(6,4^6,3^2)$	1		$(6,4,3^{17})$	13	
$(6,5^2,3^3)$	8		$(6,4^6,3)$	2		$(6,4,3^{16})$	361	
$(6,5^2,3^2)$	3		$(6,4^6)$	1		$(6,4,3^{15})$	5042	
$(6,5^2,3)$	1		$(6,4^5,3^5)$	68		$(6,4,3^{14})$	38881	
$(6,5^2)$	1		$(6,4^5,3^4)$	115		$(6,4,3^{13})$	168123	
$(6,5,4^3,3^6)$	6		$(6,4^5,3^3)$	81		$(6,4,3^{12})$	434617	
$(6,5,4^3,3^5)$	18		$(6,4^5,3^2)$	41		$(6,4,3^{11})$	695055	
$(6,5,4^3,3^4)$	39		$(6,4^5,3)$	9		$(6,4,3^{10})$	715749	
$(6,5,4^3,3^3)$	0		$(6,4^5)$	3		$(6,4,3^9)$	489801	
$(6,5,4^3,3^2)$	10		$(6,4^4,3^9)$	80		$(6,4,3^8)$	232819	
$(6,5,4^3,3)$	3		$(6,4^4,3^8)$	635		$(6,4,3^7)$	81085	
$(6,5,4^3)$	1		$(6,4^4,3^7)$	2261		$(6,4,3^6)$	22330	
$(6,5,4^2,3^9)$	25		$(6,4^4,3^6)$	3812		$(6,4,3^5)$	5140	
$(6,5,4^2,3^8)$	180		$(6,4^4,3^5)$	3716		$(6,4,3^4)$	1079	
$(6,5,4^2,3^7)$	576		$(6,4^4,3^4)$	2069		$(6,4,3^3)$	215	
$(6,5,4^2,3^6)$	925		$(6,4^4,3^3)$	749		$(6,4,3^2)$	44	
$(6,5,4^2,3^4)$	450		$(6,4^4,3^2)$	164		$(6,4)$	2	
$(6,5,4^2,3^3)$	156		$(6,4^4,3)$	29		$(6,3^{19})$	2	
$(6,5,4^2,3^2)$	38		$(6,4^4)$	5		$(6,3^{18})$	128	
$(6,5,4^2,3)$	7		$(6,4^3,3^{12})$	56		$(6,3^{17})$	2167	
$(6,5,4^2)$	1		$(6,4^3,3^{11})$	832		$(6,3^{16})$	19335	
$(6,5,4,3^{12})$	21		$(6,4^3,3^{10})$	6457		$(6,3^{15})$	94896	
$(6,5,4,3^{11})$	228		$(6,4^3,3^9)$	23669		$(6,3^{14})$	275878	
$(6,5,4,3^{10})$	1383		$(6,4^3,3^8)$	47894		$(6,3^{13})$	501678	
$(6,5,4,3^9)$	4285		$(6,4^3,3^7)$	56020		$(6,3^{12})$	592632	
$(6,5,4,3^8)$	7563		$(6,4^3,3^6)$	40455		$(6,3^{11})$	471723	
$(6,5,4,3^6)$	5255		$(6,4^3,3^5)$	18496		$(6,3^{10})$	262376	
$(6,5,4,3^5)$	2331		$(6,4^3,3^4)$	5750		$(6,3^9)$	107077	
$(6,5,4,3^4)$	786		$(6,4^3,3^3)$	1260		$(6,3^8)$	34124	
$(6,5,4,3^3)$	209		$(6,4^3,3^2)$	218		$(6,3^7)$	9114	
$(6,5,4,3^2)$	53		$(6,4^3,3)$	30		$(6,3^6)$	2169	
$(6,5,4,3)$	11		$(6,4^3)$	5		$(6,3^5)$	487	
$(6,5,4)$	5		$(6,4^2,3^{15})$	9		$(6,3^4)$	112	
$(6,5,3^{14})$	16		$(6,4^2,3^{14})$	224		$(6,3^3)$	28	
$(6,5,3^{13})$	239		$(6,4^2,3^{13})$	3385		$(6,3^2)$	7	
$(6,5,3^{12})$	1726		$(6,4^2,3^{12})$	24627		$(6,3)$	2	
$(6,5,3^{11})$	6293		$(6,4^2,3^{11})$	98713		$(6$	1	
$(6,5,3^{10})$	12876		$(6,4^2,3^{10})$	226880		$(5^3,4^3,3^6)$	2	
$(6,5,3^8)$	12569		$(6,4^2,3^9)$	316203		$(5^3,4^3,3^5)$	1	
$(6,5,3^7)$	6843		$(6,4^2,3^8)$	276681		$(5^3,4^3,3^4)$	2	
$(6,5,3^6)$	2796		$(6,4^2,3^7)$	158766		$(5^3,4^3,3^3)$	2	

Table 7.3: Linear Spaces on 13 points (Part III)

line case	#sol	time	line case	#sol	time	line case	#sol	time
$(5^3,4^3,3^2)$	2		$(5^2,4^5,3^2)$	15		$(5^2,4,3^{10})$	776876	
$(5^3,4^3,3)$	1		$(5^2,4^5,3)$	5		$(5^2,4,3^9)$	916348	
$(5^3,4^3)$	1		$(5^2,4^5)$	1		$(5^2,4,3^8)$	690691	
$(5^3,4^2,3^{10})$	1		$(5^2,4^4,3^{10})$	3		$(5^2,4,3^7)$	341918	
$(5^3,4^2,3^9)$	3		$(5^2,4^4,3^9)$	10		$(5^2,4,3^6)$	114700	
$(5^3,4^2,3^8)$	12		$(5^2,4^4,3^8)$	88		$(5^2,4,3^5)$	27476	
$(5^3,4^2,3^7)$	35		$(5^2,4^4,3^7)$	440		$(5^2,4,3^4)$	5141	
$(5^3,4^2,3^6)$	76		$(5^2,4^4,3^6)$	1209		$(5^2,4,3^3)$	839	
$(5^3,4^2,3^5)$	104		$(5^2,4^4,3^5)$	1810		$(5^2,4,3^2)$	135	
$(5^3,4^2,3^4)$	93		$(5^2,4^4,3^4)$	1587		$(5^2,4,3)$	24	
$(5^3,4^2,3^3)$	54		$(5^2,4^4,3^3)$	795		$(5^2,4)$	6	
$(5^3,4^2,3^2)$	18		$(5^2,4^4,3^2)$	242		$(5^2,3^{18})$	8	
$(5^3,4^2,3)$	5		$(5^2,4^4,3)$	39		$(5^2,3^{17})$	51	
$(5^3,4^2)$	1		$(5^2,4^4)$	8		$(5^2,3^{16})$	953	
$(5^3,4,3^{11})$	5		$(5^2,4^3,3^{12})$	1		$(5^2,3^{15})$	11211	
$(5^3,4,3^{10})$	45		$(5^2,4^3,3^{11})$	34		$(5^2,3^{14})$	77119	
$(5^3,4,3^9)$	265		$(5^2,4^3,3^{10})$	539		$(5^2,3^{13})$	295371	
$(5^3,4,3^8)$	782		$(5^2,4^3,3^9)$	3892		$(5^2,3^{12})$	670948	
$(5^3,4,3^7)$	1393		$(5^2,4^3,3^8)$	14612		$(5^2,3^{11})$	944874	
$(5^3,4,3^6)$	1486		$(5^2,4^3,3^7)$	30264		$(5^2,3^{10})$	860918	
$(5^3,4,3^5)$	983		$(5^2,4^3,3^6)$	36504		$(5^2,3^9)$	522868	
$(5^3,4,3^4)$	414		$(5^2,4^3,3^5)$	26428		$(5^2,3^8)$	219024	
$(5^3,4,3^3)$	113		$(5^2,4^3,3^4)$	11649		$(5^2,3^7)$	66226	
$(5^3,4,3^2)$	23		$(5^2,4^3,3^3)$	3147		$(5^2,3^6)$	15609	
$(5^3,4,3)$	4		$(5^2,4^3,3^2)$	528		$(5^2,3^5)$	3148	
$(5^3,4)$	1		$(5^2,4^3,3)$	63		$(5^2,3^4)$	617	
$(5^3,3^{16})$	2		$(5^2,4^3)$	7		$(5^2,3^3)$	124	
$(5^3,3^{15})$	2		$(5^2,4^2,3^{14})$	3		$(5^2,3^2)$	29	
$(5^3,3^{14})$	8		$(5^2,4^2,3^{13})$	75		$(5^2,3)$	7	
$(5^3,3^{13})$	27		$(5^2,4^2,3^{12})$	1498		(5^2)	2	
$(5^3,3^{12})$	141		$(5^2,4^2,3^{11})$	14128		$(5,4^8,3^4)$	6	
$(5^3,3^{11})$	568		$(5^2,4^2,3^{10})$	68219		$(5,4^8,3^3)$	6	
$(5^3,3^{10})$	1944		$(5^2,4^2,3^9)$	180822		$(5,4^8,3^2)$	5	
$(5^3,3^9)$	3980		$(5^2,4^2,3^8)$	281095		$(5,4^8,3)$	4	
$(5^3,3^8)$	5194		$(5^2,4^2,3^7)$	266804		$(5,4^8)$	2	
$(5^3,3^7)$	4256		$(5^2,4^2,3^6)$	159171		$(5,4^7,3^7)$	1	
$(5^3,3^6)$	2304		$(5^2,4^2,3^5)$	60729		$(5,4^7,3^6)$	12	
$(5^3,3^5)$	846		$(5^2,4^2,3^4)$	15264		$(5,4^7,3^5)$	77	
$(5^3,3^4)$	251		$(5^2,4^2,3^3)$	2649		$(5,4^7,3^4)$	147	
$(5^3,3^3)$	63		$(5^2,4^2,3^2)$	362		$(5,4^7,3^3)$	138	
$(5^3,3^2)$	16		$(5^2,4^2,3)$	41		$(5,4^7,3^2)$	76	
$(5^3,3)$	5		$(5^2,4^2)$	6		$(5,4^7,3)$	21	
(5^3)	3		$(5^2,4,3^{16})$	4		$(5,4^6,3^{10})$	1	
$(5^2,4^5,3^7)$	1		$(5^2,4,3^{15})$	74		$(5,4^6,3^9)$	6	
$(5^2,4^5,3^6)$	4		$(5^2,4,3^{14})$	1754		$(5,4^6,3^8)$	125	
$(5^2,4^5,3^5)$	15		$(5^2,4,3^{13})$	20869		$(5,4^6,3^7)$	1023	
$(5^2,4^5,3^4)$	26		$(5^2,4,3^{12})$	124429		$(5,4^6,3^6)$	3770	
$(5^2,4^5,3^3)$	28		$(5^2,4,3^{11})$	406144		$(5,4^6,3^5)$	6567	

Table 7.4: Linear Spaces on 13 points (Part IV)

line case	#sol	time	line case	#sol	time	line case	#sol	time
$(5,4^6,3^4)$	6139		$(5,4^2,3^{13})$	6230354	20:47:33	$(5,3^4)$	148	
$(5,4^6,3^3)$	3250		$(5,4^2,3^{12})$	18440779	1d 3:57:00	$(5,3^3)$	30	
$(5,4^6,3)$	151		$(5,4^2,3^{11})$	32581541	22:4:29	$(5,3^2)$	7	
$(5,4^6)$	14		$(5,4^2,3^9)$	26052596	8:2:27	$(5,3)$	2	
$(5,4^5,3^{11})$	20		$(5,4^2,3^8)$	12536329	2:45:25	$(5$	1	
$(5,4^5,3^{10})$	622		$(5,4^2,3^7)$	4119894	1:12:20	(4^{13})	1	
$(5,4^5,3^9)$	7608		$(5,4^2,3^6)$	952882	15:48	$(4^{12},3)$	1	
$(5,4^5,3^8)$	40590		$(5,4^2,3^5)$	163150	3:41	(4^{12})	1	
$(5,4^5,3^7)$	108087		$(5,4^2,3^4)$	22369		$(4^{11},3^2)$	3	
$(5,4^5,3^6)$	155977		$(5,4^2,3^3)$	2726		$(4^{11},3)$	2	
$(5,4^5,3^5)$	129455		$(5,4^2,3^2)$	325		(4^{11})	1	
$(5,4^5,3^3)$	18568		$(5,4^2,3)$	44		$(4^{10},3^4)$	3	
$(5,4^5,3^2)$	3193		$(5,4^2)$	7		$(4^{10},3^3)$	21	
$(5,4^5,3)$	312		$(5,4,3^{19})$	45		$(4^{10},3^2)$	14	
$(5,4^5)$	28		$(5,4,3^{18})$	3051		$(4^{10},3)$	8	
$(5,4^4,3^{14})$	1		$(5,4,3^{17})$	81820		(4^{10})	4	
$(5,4^4,3^{13})$	42		$(5,4,3^{16})$	988429		$(4^9,3^7)$	1	
$(5,4^4,3^{12})$	2077		$(5,4,3^{15})$	6079254	12:8:11	$(4^9,3^6)$	6	
$(5,4^4,3^{11})$	32598		$(5,4,3^{14})$	20912406	19:44:50	$(4^9,3^5)$	46	
$(5,4^4,3^{10})$	229293		$(5,4,3^{13})$	43243192	1d 5:02:47	$(4^9,3^4)$	210	
$(5,4^4,3^9)$	812612		$(5,4,3^{11})$	48781730	10:14:44	$(4^9,3^3)$	209	
$(5,4^4,3^8)$	1590488		$(5,4,3^{10})$	28462372	4:55:35	$(4^9,3^2)$	119	
$(5,4^4,3^7)$	1823744		$(5,4,3^9)$	11527765	2:10:05	$(4^9,3)$	43	
$(5,4^4,3^5)$	554163		$(5,4,3^8)$	3331705	39:59	(4^9)	10	
$(5,4^4,3^4)$	151674		$(5,4,3^7)$	715712	11:11	$(4^8,3^9)$	1	
$(5,4^4,3^3)$	26324		$(5,4,3^6)$	121723		$(4^8,3^8)$	13	
$(5,4^4,3^2)$	2983		$(5,4,3^5)$	17818		$(4^8,3^7)$	248	
$(5,4^4,3)$	247		$(5,4,3^4)$	2489		$(4^8,3^6)$	1698	
$(5,4^4)$	18		$(5,4,3^3)$	354		$(4^8,3^5)$	4665	
$(5,4^3,3^{15})$	58		$(5,4,3^2)$	56		$(4^8,3^4)$	5421	
$(5,4^3,3^{14})$	3584		$(5,4,3)$	9		$(4^8,3^3)$	3344	
$(5,4^3,3^{13})$	78065		$(5,4)$	2		$(4^8,3^2)$	1128	
$(5,4^3,3^{12})$	700220		$(5,3^{21})$	36		$(4^8,3)$	200	
$(5,4^3,3^{11})$	3118079	1:11:24	$(5,3^{20})$	1358		(4^8)	22	
$(5,4^3,3^{10})$	7681795	10:57:34	$(5,3^{19})$	28267		$(4^7,3^{12})$	1	
$(5,4^3,3^9)$	11215598	14:17:19	$(5,3^{18})$	342618		$(4^7,3^{11})$	2	
$(5,4^3,3^7)$	5876320		$(5,3^{17})$	2294103		$(4^7,3^{10})$	58	
$(5,4^3,3^6)$	2238339	2:28:28	$(5,3^{16})$	8878833	14:47:45	$(4^7,3^9)$	1346	
$(5,4^3,3^5)$	557764		$(5,3^{15})$	20922319	23:15:55	$(4^7,3^8)$	14056	
$(5,4^3,3^4)$	96119		$(5,3^{13})$	31532897	10:29:26	$(4^7,3^7)$	59820	28:41
$(5,4^3,3^3)$	12008		$(5,3^{12})$	21602330	6:14:25	$(4^7,3^6)$	120206	28:35
$(5,4^3,3^2)$	1218		$(5,3^{11})$	10395553	1:57:10	$(4^7,3^5)$	125392	22:49
$(5,4^3,3)$	106		$(5,3^{10})$	3604843	31:46	$(4^7,3^4)$	73803	
$(5,4^3)$	12		$(5,3^9)$	931053		$(4^7,3^3)$	24917	
$(5,4^2,3^{17})$	64		$(5,3^8)$	188444		$(4^7,3^2)$	4782	
$(5,4^2,3^{16})$	4352		$(5,3^7)$	32142		$(4^7,3)$	501	
$(5,4^2,3^{15})$	108396		$(5,3^6)$	5078		(4^7)	35	
$(5,4^2,3^{14})$	1167237		$(5,3^5)$	814		$(4^6,3^{14})$	1	

Table 7.5: Linear Spaces on 13 points (Part V)

line case	#sol	time	line case	#sol	time
$(4^6, 3^{13})$	9		$(4^3, 3^{19})$	3	
$(4^6, 3^{12})$	248	5:36	$(4^3, 3^{18})$	1148	
$(4^6, 3^{11})$	6837		$(4^3, 3^{17})$	74707	
$(4^6, 3^{10})$	86035	1:14:9	$(4^3, 3^{16})$	1583018	7:7:12
$(4^6, 3^9)$	487381	2:49:28	$(4^3, 3^{15})$	14754548	16:38:20
$(4^6, 3^8)$	1356861	4:7:9	$(4^3, 3^{14})$	72988666	1d 4:54:24
$(4^6, 3^7)$	2040477	1:54:52	$(4^3, 3^{13})$	196113022	*21:34:19
$(4^6, 3^6)$	1767684	50:53	$(4^3, 3^{12})$	328821551	*1d 14:34:19
$(4^6, 3^5)$	919914	16:24	$(4^3, 3^{11})$	351828103	*1d 10:34:19
$(4^6, 3^4)$	292231	4:8	$(4^3, 3^{10})$	248473404	2d 2:1:30
$(4^6, 3^3)$	56576		$(4^3, 3^9)$	118799099	20:15:52
$(4^6, 3^2)$	6602		$(4^3, 3^8)$	39203496	3:38:8
$(4^6, 3)$	482		$(4^3, 3^7)$	9094590	1:20:51
(4^6)	30		$(4^3, 3^6)$	1524680	8:51
$(4^5, 3^{15})$	4		$(4^3, 3^5)$	193998	1:21
$(4^5, 3^{14})$	654	26:22	$(4^3, 3^4)$	20461	
$(4^5, 3^{13})$	24084	2:23:10	$(4^3, 3^3)$	2007	
$(4^5, 3^{12})$	360474	5:17:57	$(4^3, 3^2)$	213	
$(4^5, 3^{11})$	2475063	22:47:31	$(4^3, 3)$	28	
$(4^5, 3^{10})$	8672320	1d 10:8:41	(4^3)	5	
$(4^5, 3^9)$	16829846	22:24:57	$(4^2, 3^{21})$	4	
$(4^5, 3^8)$	19277783	9:20:30	$(4^2, 3^{20})$	987	
$(4^5, 3^7)$	13614375	4:38:30	$(4^2, 3^{19})$	65021	
$(4^5, 3^6)$	6098046	2:2:42	$(4^2, 3^{18})$	1538893	30:25
$(4^5, 3^5)$	1757334	28:7	$(4^2, 3^{17})$	16225498	2:26:10
$(4^5, 3^4)$	328430	3:42	$(4^2, 3^{16})$	88852151	8:07:22
$(4^5, 3^3)$	40215		$(4^2, 3^{15})$	278442891	*1d 8:34:45
$(4^5, 3^2)$	3421		$(4^2, 3^{14})$	540120259	*2d 9:42:19
$(4^5, 3)$	230		$(4^2, 3^{13})$	673351261	*2d 14:54:23
(4^5)	18		$(4^2, 3^{12})$	561113925	*2d 6:23:17
$(4^4, 3^{17})$	9		$(4^2, 3^{11})$	321129676	3d 2:19:00
$(4^4, 3^{16})$	1080		$(4^2, 3^{10})$	128917244	22:41:41
$(4^4, 3^{15})$	53865	17:30	$(4^2, 3^9)$	36987720	5:12:14
$(4^4, 3^{14})$	964011	14:25:31	$(4^2, 3^8)$	7757524	1:3:26
$(4^4, 3^{13})$	7777790	3d 3:17:16	$(4^2, 3^7)$	1234019	10:34
$(4^4, 3^{12})$	32448444	3d 7:28:20	$(4^2, 3^6)$	158556	1:32
$(4^4, 3^{11})$	76436861	3d 12:37:0	$(4^2, 3^5)$	18084	
$(4^4, 3^{10})$	108183543	*16:34:19	$(4^2, 3^4)$	2073	
$(4^4, 3^9)$	96273769	2d 5:19:06	$(4^2, 3^3)$	263	
$(4^4, 3^8)$	55621336	18:56:28	$(4^2, 3^2)$	40	
$(4^4, 3^7)$	21326125	4:38:18	$(4^2, 3)$	7	
$(4^4, 3^6)$	5511065	1:21:14	(4^2)	2	
$(4^4, 3^5)$	976346	16:35	$(4, 3^{22})$	590	
$(4^4, 3^4)$	122628	1:29	$(4, 3^{21})$	34302	10:23
$(4^4, 3^3)$	11710		$(4, 3^{20})$	824735	3:59:23
$(4^4, 3^2)$	970		$(4, 3^{19})$	9474152	9:54:30
$(4^4, 3)$	82		$(4, 3^{18})$	57722151	1d 15:23:11
(4^4)	11		$(4, 3^{17})$	194993453	*22:55:13

Table 7.6: Linear Spaces on 13 points (Part VI)

line case	#sol	time	line case	#sol	time
$(4,3^{16})$	445479140	*1d 23:14:44	(3^{21})	2306220	
$(4,3^{15})$	633464694	*2d 9:46:27	(3^{20})	15142370	5:55:34
$(4,3^{14})$	579870663	*2d 2:15:39	(3^{19})	58893945	7:56:14
$(4,3^{13})$	405064372	3d 2:16:55	(3^{18})	143386618	*10:45:33
$(4,3^{12})$	191602452	1d 6:27:58	(3^{17})	228896539	*21:23:10
$(4,3^{11})$	65563256	11:07:05	(3^{16})	248583304	*1d 2:45:28
$(4,3^{10})$	16546837	3:20:5	(3^{15})	189057254	1d 12:05:38
$(4,3^{9})$	3165219	41:33	(3^{14})	103043009	1d 8:7:40
$(4,3^{8})$	480802	6:24	(3^{13})	41023224	10:17:50
$(4,3^{7})$	62568		(3^{12})	12142983	4:6:25
$(4,3^{6})$	7786		(3^{11})	2729981	45:13
$(4,3^{5})$	1033		(3^{10})	482568	7:47
$(4,3^{4})$	167		(3^{9})	71311	1:43
$(4,3^{3})$	31		(3^{8})	9768	
$(4,3^{2})$	7		(3^{7})	1419	
$(4,3)$	2		(3^{6})	250	
(4)	1		(3^{5})	54	
(3^{26})	2		(3^{4})	16	
(3^{25})	10		(3^{3})	5	
(3^{24})	267		(3^{2})	2	
(3^{23})	9348		(3)	1	
(3^{22})	197746		$()$	1	

Total Solutions $= 8,592,194,823$

Table 7.7: Parameters refinement comparison

case	#cases	time(R)	#sol	time(sol)	total time	avg
line $(7, 4, 3^8, 2^{27})$	1					
point (2)	46,083	0:05	3,157	4:05	4:10	13
line (3)	1,886	0:32	⋆	3:01	3:38	15
point (4)	2,897	0:07	⋆	0:04	0:48	66
line $(7, 3^9, 2^{30})$	1					
point (2)	2,188	0:00	3,686	0:22	0:22	168
line (3)	1,202	0:02	⋆	0:17	0:19	194
point (4)	2,975	0:04	⋆	0:03	0:09	410
line $(6, 4^2, 3^7, 2^{30})$	1					
point (2)	341,578	0:49	158,766	39:13	40:01	66
line (3)	181,203	5:59	⋆	7:20	14:08	187
point (4)	151,030	4:56	⋆	1:34	13:18	199
line $(6, 4, 3^{13}, 2^{18})$	1					
point (2)	16,099	0:02	168,123	46:22	46:24	60
line (3)	30,150	1:21	⋆	24:47	26:08	107
point (4)	119,953	2:05	⋆	3:05	6:33	428
line $(4, 3^{21}, 2^9)$	1					
point (2)	18	0:00	34,302	10:45	10:45	53
line (3)	256	0:00	⋆	9:30	9:30	60
point (4)	5,856	0:02	⋆	2:29	2:31	227
line (5)	25,444	0:25	⋆	1:46	2:13	258
(**TDO**)point (10)	33,880	0:31	⋆	0:07	1:12	476
line $(3^{22}, 2^{12})$	1					
point (2)	17	0:00	197,746	3:34:17	3:34:17	15
line (3)	164	0:00	⋆	2:55:49	2:55:49	19
point (4)	3,346	0:01	⋆	1:06:54	1:06:55	49
line (5)	63,312	0:30	⋆	1:00:13	1:00:44	54
point (6)	137,928	0:42	⋆	55:23	56:36	58
line (7)	176,320	0:50	⋆	43:56	45:59	72

Table 7.8: Parameters refinement comparison

line type	point type	depth	#cases	time(R)	#sol	time(sol)	avg
	$1 \times (3^5, 2^2)$	2	1	0:00	> 5,000,000	1d 2:31:45	52
$(3^{15}, 2^{33})$	$4 \times (3^4, 2^4)$	4	9,490	0:04	61,341,764	50:12	20,366
	$8 \times (3^3, 2^6)$						
$(3^{15}, 2^{33})$	$6 \times (3^4, 2^4)$	2	1	0:00	> 1,359,542	7:36:24	50
	$7 \times (3^3, 2^6)$	4	2,241	0:00	5,359,458	49:10	1,817
	$2 \times (3^6, 2^0)$	2	1	0:00	59,131	17:28:49	1
$(3^{19}, 2^{21})$	$1 \times (3^5, 2^2)$	4	3	0:00	59,131	2:20	422
	$10 \times (3^4, 2^4)$	6	12,018	0:10	59,131	1:10	845
	$1 \times (4^0, 3^6, 2^0)$						
	$3 \times (4^1, 3^3, 2^3)$	2	1	0:00	1,266,585	23:43:18	15
$(4, 3^{17}, 2^{21})$	$2 \times (4^0, 3^5, 2^2)$	4	144,044	3:23	1,266,585	17:35	1,201
	$1 \times (4^1, 3^2, 2^5)$						
	$6 \times (4^0, 3^4, 2^4)$						
	$1 \times (4^2, 3^2, 2^2)$						
	$1 \times (4^1, 3^4, 2^1)$						
$(4^2, 3^{15}, 2^{21})$	$4 \times (4^1, 3^3, 2^3)$	2	1	0:01	3,641,685	16:33:48	61
	$1 \times (4^0, 3^5, 2^2)$	4	1,151,360	20:05	3,641,685	43:25	1,398
	$1 \times (4^1, 3^2, 2^5)$						
	$5 \times (4^0, 3^4, 2^4)$						

and the generations is displayed below indicated by time(R) and time(sol), respectively.

Note that $> 5,000,000$ at the first line case in the table above means that the search was not completed. These results show that using the TDO algorithm for constructing linear spaces is an improvement in the sense of the time needed to construct such structures. This might be considered as evidence that this algorithm helps to avoid some of the expensive computations needed by the isomorph-rejection in the canonical augmentation. Thus, it may lead to a new study line where one might consider such an algorithm for constructing other incidence structures.

However, in some cases namely with line cases of small classes, this TDO algorithm might not lead to an improvement. The following tables show some information about the generation and the refinement for different line cases in different depths. The time in the third column between parenthesis "( )" represents the time was used in the refinement of the cases in the previous depth.

Table 7.9: Parameters refinement comparison

line case	depth	#cases(time)	time(sol)	#sol	total time
line (6 4 3^{13} 2^{18})	1	1			
	2	16,099 (0:02)	33:09	168,123	
	3	30,150 (1:44)	25:27	⋆	
	4	119,953 (2:51)	3:08	⋆	
	5	159,174 (6:08)	3:52	⋆	
	6	166,266 (9:41)	4:28	⋆	
	7	167,381 (7:35)	4:43	⋆	
line (6 3^{15} 2^{18})	1	1			
	2	296	11:30 (0:00)	94,896	
	3	3,185	5:46:52 (0:02)	⋆	
	4	34,046	3:08 (0:48)	⋆	
	5	77,329	5:40 (7:01)	⋆	
	6	93,262	3:07 (7:06)	⋆	
	7	93,760	4:03 (9:08)	⋆	
	8	93,927	4:03 (7:40)	⋆	
line (5 4^2 3^{16} 2^8)	1	1			
	2	1,401 (0:02)	29:34	4,352	
	3	4,088 (0:10)	28:05	⋆	
	4	4,244 (0:12)	0:16	⋆	
	5	4,543 (0:11)	0:19	⋆	
	6	4,196 (0:10)	0:11	⋆	
	7	4,231 (0:10)	0:11	⋆	
	8	4,219 (0:11)	0:11	⋆	
line (4^3 3^{17} 2^9)	1	1			
	2	260 (0:00)	38:29	74,707	
	3	35,747 (1:01)	> 10:00:00	> 10	
	4	87,491 (2:34)	4:20	74,707	
	5	81,343 (3:04)	37:34	⋆	
	6	74,138 (4:07)	1:57	⋆	
	7	74,310 (3:31)	2:18	⋆	
	8	74,268 (3:40)	2:23	⋆	

Table 7.10: Parameters refinement comparison

line case	depth	#cases(time)	time(sol)	#sol	total time
line ($5\ 4^3\ 3^{13}\ 2^{11}$)	1	1			
	2	46,062 (0:30)	5:03:35	78,065	
	3	247,538 (2:35)	1d 15:38:13	⋆	
	4	85,561 (10:15)	3:08	⋆	
	5	79,346 (5:20)	3:07	⋆	
	6	78,404 (6:56)	3:14	⋆	
	7	78,447 (5:05)	3:17	⋆	
line ($5\ 3^{19}\ 2^{11}$)	1	1			
	2	123 (0:00)	1:44:07	28,267	
	3	669 (0:00)	4:30:41	⋆	
	4	8,030 (0:05)	1:53	⋆	
	5	21,693 (0:45)	9:04	⋆	
	6	24,719 (0:59)	1:00	⋆	
	7	26,908 (1:23)	1:20	⋆	
line ($5\ 4\ 3^{17}\ 2^{11}$)	1	1			
	2	1,742 (0:00)	1:40:20	81,820	
	3	10,885 (0:25)	2:07:53	⋆	
	4	60,010 (2:06)	3:41	⋆	
	5	77,300 (4:09)	6:03	⋆	
	6	80,349 (5:50)	3:05	⋆	
	7	81,337 (5:03)	3:13	⋆	
line ($5\ 4^4\ 3^3\ 2^{35}$)	1	1			
	2	149,793 (0:15)	2:46	26,324	
	3	69:495 (8:10)	1:42	⋆	
	4	26,871 (4:02)	0:25	⋆	
	5	27,088 (1:25)	0:38	⋆	
	6	27,098 (2:25)	0:50	⋆	
	7	27,098 (1:43)	0:46	⋆	
line ($5^2\ 4^2\ 3^4\ 2^{34}$)	1	1			
	2	104,959 (0:32)	2:35	15,264	
	3	357,532 (14:25)	11:17	⋆	
	4	20,299 (28:03)	0:25	⋆	
	5	20,587 (1:17)	0:33	⋆	
	6	20,514 (1:51)	0:40	⋆	
	7	20,515 (1:27)	0:40	⋆	

Table 7.11: Parameters refinement comparison

line case	depth	#cases(time)	time(sol)	#sol	total time
$4^7\ 3^3\ 2^{27}$	1	1			
	2	2,612 (0:00)	1:13	24,917	
	3	262,664 (45:54)	1:21:07	⋆	
	4	23,648 (9:57)	0:31	⋆	
	5	25,643 (1:36)	0:41	⋆	
	6	24,943 (2:21)	0:48	⋆	
	7	25,036 (1:45)	0:52	⋆	
$4^2\ 3^5\ 2^{51}$	1	1			
	2	1,836 (0:00)	0:08	18,084	
	3	22,523 (0:10)	0:32	⋆	
	4	17,480 (2:26)	0:15	⋆	
	5	18,250 (0:49)	0:22	⋆	
	6	18,115 (1:46)	0:28	⋆	
	7	18,122 (1:00)	0:30	⋆	

Chapter 8

Normally Regular Digraph

In this chapter, we discuss a class of digraphs called **normally regular digraphs**, abbreviated by **NRDs**, which were introduced by Jørgensen [44]. In the same time we present some of the theoretical results found in [44], and add some observations on the automorphism group of such structures.

Our goal in this chapter is to expand the results were found in [44] in the sense of constructed digraphs. For more details about stated results in this chapter, one can see [39, 40, 43, 44, 45, 46].

8.1 Definitions and Examples

Definition 8.1.1. *Let $G = (V, E)$ be a finite directed graph, a digraph, and let x and y be any two vertices of G such that $x \neq y$. Then:*

1. *We say that x **dominates** y if there is an edge directed from x to y and we write it as $x \longrightarrow y$.*

2. *Define x^+ to be the set of out-neighbors of x, i.e. the set of vertices in G dominated by x. Sometimes we consider this set as a subgraph of G, $x^+ \subset G$.*

3. *Let x^- be the set of in-neighbors of x, i.e. the set of vertices in G that dominates x. Sometimes we consider this set as a subgraph of G, $x^- \subset G$.*

4. *Denote the cardinalities of x^+ and $x^+ \cap y^+$ by $d^+(x)$ and $d^+(x,y)$, respectively. Similarly, $d^-(x)$ and $d^-(x,y)$ denote the size of the sets x^- and $x^- \cap y^-$, respectively.*

5. *G is **regular** if there is a number k so that $d^+(x) = d^-(x) = k$ for every vertex x in G.*

6. *G is a **tournament** if there is an edge between every distinct vertices in G.*

Definition 8.1.2. *Enumerate the vertices of G from 1 to n, so that $V(G) = \{v_1, v_2, \ldots, v_n\}$. Define **the adjacency matrix** A of G to be the $n \times n$ matrix whose $(i,j)^{th}$ entry equals 1 if $v_i \rightarrow v_j$ and 0 otherwise.*

Claim 8.1.3. *If A^* is the transpose of A, then the $(i,j)^{th}$ entry of AA^* is $d^+(v_i, v_j)$ and the $(i,j)^{th}$ entry of A^*A is $d^-(v_i, v_j)$.*

Thus A is **normal** if and only if $d^+(x,y) = d^-(x,y)$ for every $x, y \in V(G)$.

Definition 8.1.4. *A digraph G is **normal** if its adjacency matrix is normal.*

Lemma 8.1.5. *A tournament is normal if and only if it is regular.*

Proof. First, if tournament T is normal, then $d^+(x) = d^-(x)$ for any vertex x in T. Therefore, T is regular.

For the other direction, assume that T is a regular tournament. If x and y are two vertices in T, then we want to show that $d^+(x,y) = d^-(x,y)$. Now, if $x = y$ we are finished. Assume that $x \neq y$, and without loss of generality assume that $x \rightarrow y$. If the number of out-neighbors of y is $d^+(y)$, then some of those vertices are also out-neighbors of x, which are equal to $d^+(x,y)$. The remaining vertices which are out-neighbor of y, but not out-neighbors of x are equal to $d^+(y) - d^+(x,y)$ and dominate x. Besides these vertices which dominate x, there are additionally $d^-(x) - (d^+(x) - d^+(x,y)) = d^+(x,y)$ vertices that dominate x. And these vertices also dominate y because they are not in y^+. Therefore, $d^+(x,y) = d^-(x,y)$ and then T is normal. □

Definition 8.1.6. *A* ***normally regular digraph****, NRD, with parameters (v,k,λ,μ) is a directed graph, G, on v vertices without* ***2-cycles*** *or* ***multiple edges*** *such that the following properties hold:*

- *Every vertex has* ***out-degree*** *k, i.e. $d^+(x) = k$ where $x \in G$.*
- *Every pair of adjacent vertices has λ common* ***out-neighbors****, i.e. $d^+(x,y) = \lambda$ where $x \sim y$ and $x,y \in G$.*
- *Every pair of non-adjacent vertices has μ common* ***out-neighbors****, i.e. $d^+(x,y) = \mu$ where $x \not\sim y$ and $x,y \in G$.*

Remark 8.1.1.

$$\eta = k - \mu + (\mu - \lambda)^2 \tag{8.1}$$

$$\rho = k + \mu - \lambda \tag{8.2}$$

Claim 8.1.7. *In NRDs, we have the following conditions:*

$$v \geq 2k+1 \quad for \quad k \geq 1 \tag{8.3}$$

$$k \geq 2\lambda+1 \quad for \quad k \geq 1 \tag{8.4}$$

Claim 8.1.8. *If A is an adjacency matrix of an $NRD(v,k,\lambda,\mu)$, G, then A satisfies the following equation*

$$AA^* = kI + \lambda(A+A^*) + \mu(J - I - A - A^*), \tag{8.5}$$

where I is the $v \times v$ identity matrix and J is all 1 $v \times v$ matrix.

Note that the (i, j) entry of the matrix AA^* is $d^+(v_i, v_j)$, which is k, λ, or μ, depending on whether $v_i = v_j$, $v_i \to v_j$ and $v_i \neq v_j$, or $v_i \not\to v_j$ and $v_i \neq v_j$, respectively.

Sometimes it is more convenient to write Equation 8.5 in the following form:

$$(A + (\mu - \lambda)I)(A + (\mu - \lambda)I)^* = \eta I + \mu J \tag{8.6}$$

where $\eta = k - \mu + (\mu - \lambda)^2$ and $\rho = k + \mu - \lambda$.

Theorem 8.1.9. *A normally regular digraph is normal. In particular,* $AA^* = A^*A$.

Proof. Let $B = A + (\mu - \lambda)I$. Then with simple calculation, we have $BB^* = \eta I + \mu J$. If we assume that B is singular, then one of the eigenvalues of $\eta I + \mu J$ is zero. Then, we solve for the determinant in order to find the eigenvalues of $\eta I + \mu J$:

$$det((\eta I + \mu J) - \gamma I) = \begin{vmatrix} \eta + \mu - \gamma & \mu & \dots & \mu \\ \mu & \eta + \mu - \gamma & \dots & \mu \\ \vdots & & \ddots & \vdots \\ \mu & \dots & \mu & \eta + \mu - \gamma \end{vmatrix}$$

and then we add every row to the 1^{st} row:

$$\begin{vmatrix} \eta + v\mu - \gamma & \eta + v\mu - \gamma & \dots & \eta + v\mu - \gamma \\ \mu & \eta + \mu - \gamma & \dots & \mu \\ \vdots & & \ddots & \vdots \\ \mu & \dots & \mu & \eta + \mu - \gamma \end{vmatrix} = \eta + v\mu - \gamma \begin{vmatrix} 1 & 1 & \dots & 1 \\ 0 & \eta - \gamma & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & \eta - \gamma \end{vmatrix}$$

Therefore, we have two eigenvalues, $\eta + v\mu$ and η with multiplicities 1 and $v - 1$, respectively. Then, $\eta + v\mu = 0$ or $\eta = 0$. But because $\mu, v \geq 0$ this is possible only if $\eta = k - \mu + (\mu - \lambda)^2 = 0$. Then, $k + (\mu - \lambda)^2 \geq k \geq \mu$, but also $\mu = k + (\mu - \lambda)^2$. This implies $k = \mu = \lambda$. By 8.4, $k = 0$.

Therefore, we may assume that B is nonsingular. Now, since every vertex has out-degree k, and J is symmetric, then

$$BJ = (A + (\mu - \lambda)I)J = (k + \mu - \lambda)J = \rho J$$

$$\Rightarrow \quad BJ = \rho J \quad \Rightarrow \quad \rho^{-1}J = B^{-1}J$$

$$B^* = B^{-1}(BB^*) = B^{-1}(\eta I + \mu J) = \eta B^{-1} + \mu\rho^{-1}J \tag{8.7}$$

Using that J is symmetric, we get the following:

$$\rho J = (BJ)^* = JB^* = \eta JB^{-1} + \mu\rho^{-1}J^2 = \eta JB^{-1} + \mu\rho^{-1}vJ$$

This implies that

$$JB^{-1} = \frac{\rho - \mu\rho^{-1}v}{\eta}J$$

and so

$$vJ = J^2 = (JB^{-1})(BJ) = \frac{\rho - \mu\rho^{-1}v}{\eta}\rho vJ$$

Thus

$$\frac{\rho - \mu\rho^{-1}v}{\eta} = \rho^{-1} \tag{8.8}$$

and $JB^{-1} = \rho^{-1}J$ or $\rho J = JB$. Now Equation 8.7 implies

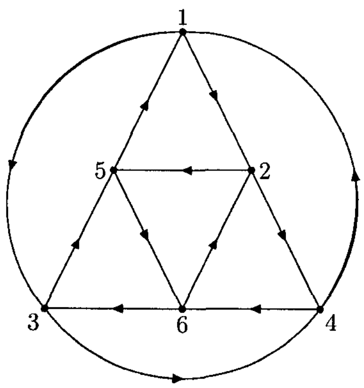
$$B^*B = \eta I + \mu\rho^{-1}JB = \eta I + \mu J = BB^*$$

Thus by using the definition of B, we have $A^*A = AA^*$. Rewriting Equation 8.8, we get

$$\mu v = \rho^2 - \eta \tag{8.9}$$

□

To give a better idea on those digraphs and to see the normality in NRDs, we present the following NRD with parameter set $(6, 2, 0, 2)$, where we have $v = 6$ vertices and each vertex has a degree $k = 2$, and if two vertices are adjacent then they have $\lambda = 0$ common out- and in-neighbors and $\mu = 2$ otherwise.



	v_1	v_2	v_3	v_4	v_5	v_6
v_1		x	x			
v_2				x	x	
v_3				x	x	
v_4	x					x
v_5	x					x
v_6		x	x			

Figure 8.1.1: $NRD(6, 2, 0, 2)$ digraph with its adjacency matrix

From Figure 8.1.1 of the NRD or from its adjacency matrix, we can see that any two adjacent vertices, e.g. vertices 1 and 2, have $\lambda = 0$ common out- and in-neighbors. If we look at vertices 1 and 6 which are not adjacent, then they have $\mu = 2$ common out-neighbors, namely vertices 2 and 3, and in-neighbors, namely vertices 4 and 5.

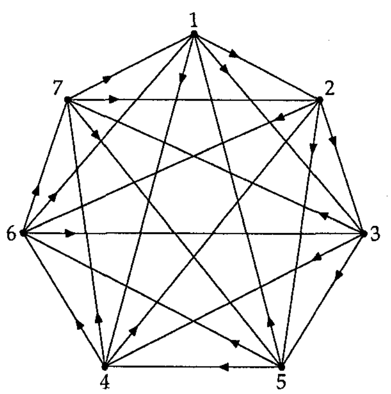
NRDs in general have some connections to some other combinatorial structures including **symmetric 2-design** for some values of $\mu = \lambda$ or $\lambda + 1$, see [39, 40, 44, 45, 46]. Moreover, NRDs which are tournaments have appeared in many applications and were given different names. Reid and Brown [70] called them **doubly regular tournaments**, where Ito [40] used the term **Hadamard tournament**, as these tournaments are equivalent to **skew Hadamard matrices**, see Reid and Brown [70].

For instance, if A is the adjacency matrix of an $NRD(v, k, \lambda, \mu)$ with $\lambda = \mu$, then by Equation 8.5, A satisfies:

$$AA^* = (k - \lambda)I + \lambda J$$

so that A is the incidence matrix of a symmetric $2-(v, k, \lambda)$-design, or a symmetric t-design with $t = 2$. Assume that G is an $NRD(v, k, \lambda, \mu)$ with $\lambda = \mu$ and G has the vertex set $\{x_1, x_2, \ldots, x_v\}$. Then, by Ito's observation [40], we can construct a symmetric $2-(v, k, \lambda)$-design with points $\mathcal{P} = \{p_1, p_2, \ldots, p_v\}$ and blocks $\mathcal{B} = \{B_1, B_2, \ldots, B_v\}$ such that $p_i \in B_j$ if and only if x_i dominates x_j, i.e. $x_i \rightarrow x_j$ in G.

For example, consider the $NRD(7, 3, 1, 1)$ in the following figure.



$A =$

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_1		x	x	x			
v_2			x		x	x	
v_3				x	x		x
v_4		x				x	x
v_5	x			x		x	
v_6	x		x				x
v_7	x	x			x		

Figure 8.1.2: $NRD(7, 3, 1, 1)$ with its adjacency matrix A

In the previous figure, A can be considered also as an incidence matrix of a symmetric $2 - (7, 3, 1)$-design where the rows and columns represent the points $\mathcal{P}$ and the blocks $\mathcal{B}$, respectively. The resulting design (permuted) can be seen as in Figure 2.3.1 which is the Fano plane.

Conversely, suppose that for some symmetric design, there is an enumeration of points and blocks such that $p_i \notin B_i$ so that the diagonal entries are 0s, and if $p_i \in B_j$, then $p_j \notin B_i$. Then, the incidence matrix of that design with respect to this enumeration is an adjacency matrix of an NRD with $\lambda = \mu$.

Jørgensen [44] explained very well the relation between NRDs and some other combinatorial structures with some special values of $\mu = \lambda, \lambda + 1, 0, k$. Here, we consider the last two cases of μ (namely, $\mu = 0$ and $\mu = k$) which

	x	x	x										
		x		x	x								
			x	x		x							
	x				x	x							
x			x		x								
x		x				x							
x	x			x									
								x	x	x			
									x		x	x	
										x	x		x
								x				x	x
							x			x		x	
							x		x				x
							x	x			x		

Figure 8.2.1: $NRD(14, 3, 1, 0)$ with $\mu = 0$.

leads us to some observations about the size of the automorphism groups for such NRDs.

8.2 NRDs with $\mu = 0$

In this section, we investigate three types of NRD in which some properties hold. In the case where $\mu = 0$, an NRD does not need to be connected. However each component will be an NRD with the same value of k and λ. Consider, for example, the unique $NRD(14, 3, 1, 0)$ with $\mu = 0$ whose adjacency matrix is giving below:

It can be seen that this digraph is not connected. In fact, it consists of two connected components, both of which is identical to the $NRD(7, 3, 1, 0)$ which was given in Figure 8.1. Thus we will only consider connected NRDs whose underlying (undirected) graphs are connected. Each vertex has equal in- and out-degree, which implies that the digraph is strongly connected, i.e. for each pair of distinct vertices x and y, there is a directed path from x to y and a directed path from y to x. Therefore, an NRD, G, is one of the following three cases:

<u>case 1</u>:

An NRD with $\mu = 0$ may be a tournament since in a tournament each two distinct vertices are adjacent and thus μ can be equal to any non-negative number.

<u>case 2</u>:

G can be a directed cycle and hence $k = 1$ and $\lambda = 0$.

<u>case 3</u>:

In this case, we assume that $k \geq 2$ so that G is not a directed cycle, and G is not a tournament.

Since in case (3) $\mu = 0$ then by 8.1 and 8.2 we have $\eta = k - \mu + (\mu - \lambda)^2 = k + \lambda^2$ and $\rho = k + \mu - \lambda = k - \lambda$. By inserting ρ and η in Equation 8.9, we get that $\lambda = \frac{k-1}{2}$. Let $x \in G$ be any vertex. Then we claim that x^+ and x^- are regular tournaments.

Claim 8.2.1. *x^+ and x^- are regular tournaments.*

Proof. For a vertex $x \in G$, every vertex in x^+ has out-degree λ in this subgraph since for any vertex $z \in x^+$, we have $d^+(x, z) = \lambda$. Moreover, x^+ is a tournament, because if it is not, then there exist two vertices $v_1, v_2 \in x^+$ such that v_1 and v_2 are not adjacent and each of them has distinct out-neighbors equal to λ in x^+ because $\mu = 0$. Altogether, we have $\{v_1, v_2\}$ and 2λ which will add up to $2\lambda + 2$, but $|x^+| = k = 2\lambda + 1$ is a contradiction. Therefore, x^+ is a regular tournament and by Lemma 8.1.5 it is normal, which means that the size of out-neighbors is the same as the size of in-neighbors. $\square$

Since G is assumed to be connected and because $\mu = 0$, it is clear that G is strongly connected, because if it is not then we have two vertices in G, say x and y, such that there is no directed path from x to y, and this is impossible because of the condition of μ being 0. Without loss of generality, assume that we have a directed path x to v_i and another directed path from v_{i+1} to y, and we have $v_i \leftarrow v_{i+1}$, as follows:

$$x \rightarrow v_1 \rightarrow v_2 \rightarrow \dots v_{i-1} \rightarrow v_i \leftarrow v_{i+1} \rightarrow v_{i+2} \rightarrow \dots \rightarrow v_{n-1} \rightarrow y.$$

But then v_{i-1} and v_{i+1} would have a common out-neighbor, which can not be because $\mu = 0$, so they have to be adjacent. Recursing with this idea, we will get a directed path from x to y. Therefore, G is strongly connected.

Claim 8.2.2. *Every vertex in x^+ dominates λ vertices in x^-, and vice versa.*

Proof. Since G is strongly connected and it is not a tournament, there is a vertex $y \in G$ such that $x \not\sim y$, and there is a directed path from x to y in G. We may choose y so that the directed distance from x to y is minimal, such that y is dominated by a vertex in x^+ or in x^-. But y can not be dominated by any vertex in x^- because $\mu = 0$ and $x \not\sim y$. Similarly, y can not dominate any vertex in x^+.

Therefore, y is dominated by a vertex, say v, in x^+. Suppose there is a vertex w in x^+ that does not dominate y. Since x^+ is a regular tournament and is strongly connected, there is a directed path from v to w in x^+. On this path, there are vertices u and u' so that $u \rightarrow u'$ and $u \rightarrow y$, and $u' \not\rightarrow y$. This contradicts $\mu = 0$. Hence, every vertex in x^+ dominates y. Similarly, y dominates every vertex in x^-. Assume now, if there is another vertex y' with $x \not\sim y'$, and that y' is dominated by a vertex in x^+, it will be dominated by every vertex in x^+ and so $d^-(y, y') = k$ is a contradiction. Therefore, every vertex in x^+ dominates λ vertices in x^-.

With the same argument, we can show that every vertex in x^- dominates λ vertices in x^+, because every vertex in x^- is adjacent to x and thus they have λ common out-neighbors which are contained in x^+. $\square$

Therefore, $V(G) = \{x, y\} \cup x^+ \cup x^-$ and thus $|V(G)| = 2k + 2$. Furthermore, there is an enumeration of vertices $x^+ = \{v_1, v_2, \dots, v_n\}$ and $x^- = \{v'_1, v'_2, \dots, v'_n\}$ such that v'_i is the unique vertex non-adjacent to v_i and vice

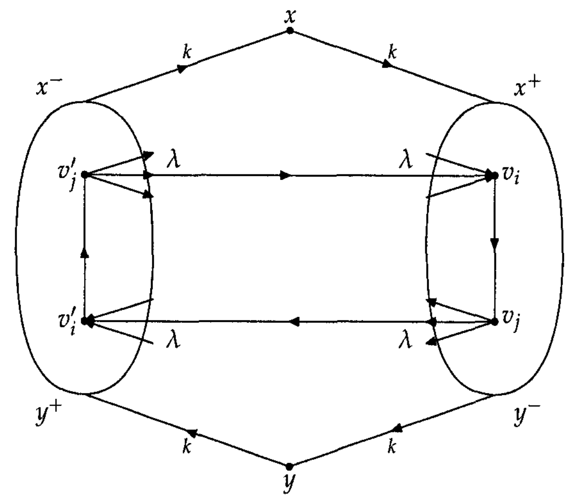


Figure 8.2.2: A graph G explaining Claim 8.2.2.

versa. That means, if $v_i \to v_j$ then since no vertex dominates both v_j and v_j', we conclude that $v_j' \to v_i$. Similarly, $v_j \to v_i'$ and $v_i' \to v_j'$. This is because $v_j \not\to v_j'$ and $\mu = 0$. Thus the mapping $v_i \mapsto v_i'$ is an isomorphism.

The following figure gives more explanation of the situation above. We also see that v_k' is a common out-neighbor of v_i and v_j if and only if v_k is a common in-neighbor of v_i and v_j. Thus the number of vertices in x^+ dominating v_i and v_j plus the number of vertices in x^+ dominated by v_i and v_j is equal $\lambda - 1$. That is because v_i and v_j are adjacent, and then $d^+(v_i, v_j) = \lambda - 1$ in G with out vertex y, and $d^-(v_i, v_j) = \lambda - 1$ in G without vertex x, and thus v_i and v_j would have $\lambda - 1$ common out- or in-neighbors in x^+. That means if $v_i \to v_k \leftarrow v_j$ then $v_i \leftarrow v_k' \to v_j$, but $d^+(v_i, v_j) = d^-(v_i, v_j) = \lambda$ and since $y \notin x^+$ and x^+ is a regular tournament hence it is normal by Lemma 8.1.5. Thus in x^+, $d^+(v_i, v_j) = d^-(v_i, v_j) = \frac{\lambda-1}{2}$, and so λ is odd. Therefore, x^+ is an $NRD(k, \lambda, \frac{\lambda-1}{2}, \cdot)$.

If on the other hand G is a doubly regular tournament (i.e. a regular tournament satisfying λ- and μ-conditions) with degree $k = 2\lambda + 1$ and with vertex-set $= \{x_1, x_2, \ldots, x_n\}$, $n = 2k + 1$, then we can construct a graph with vertex-set $= \{v_0, v_1, \ldots, v_n, v_0', v_1', \ldots, v_n'\}$, and edges :

$$v_0 \to v_i \to v_0' \to v_i' \to v_0 \qquad \text{for} \quad 1 \le i \le n.$$

and

$$v_i \to v_j \to v_i' \to v_j' \to v_i \quad \text{if} \quad x_i \to x_j \quad \text{in} \quad G \ \forall \ 1 \le i, j \le n$$

It is easy to check that this new graph is an $NRD(2n+2, n, k, 0)$. As a result of this section with the case of $\mu = 0$, we summarize the cases for G in the following theorem, as in [44]:

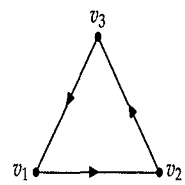
Theorem 8.2.3. *A connected digraph is an NRD with $\mu = 0$ if and only if either:*

1. *It is a directed cycle $(k = 1)$,*
2. *it is a doubly regular tournament, or*
3. *it has an adjacency matrix of the following form*

$$\begin{pmatrix} 0 & 1 & \ldots & 1 & 0 & 0 & \ldots & 0 \\ 0 & & & & 1 & & & \\ \vdots & & A & & \vdots & & A^* & \\ 0 & & & & 1 & & & \\ 0 & 0 & \ldots & 0 & 0 & 1 & \ldots & 1 \\ 1 & & & & 0 & & & \\ \vdots & & A^* & & \vdots & & A & \\ 1 & & & & 0 & & & \end{pmatrix} \tag{8.10}$$

where A is an adjacency matrix of a doubly regular tournament.

As an application for possibility 3, consider the following example: Consider the doubly regular tournament $NRD(3,1,0,0)$ which is a directed cycle with three vertices and has the adjacency matrix A as follows:

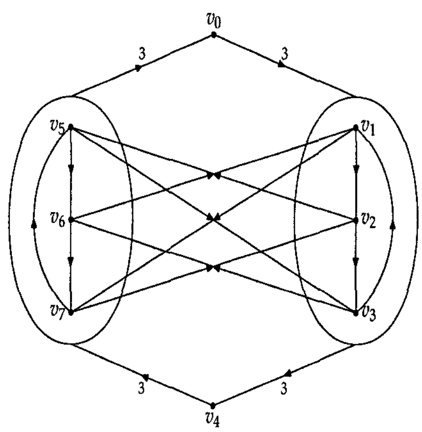


$A =$

	v_1	v_2	v_3
v_1		x	
v_2			x
v_3	x		

Figure 8.2.3: $NRD(3,1,0,0)$ with its adjacency matrix A

and then we can construct an $NRD(8,3,1,0)$ with the following adjacency matrix $\mathcal{A}$ as in the above theorem, as follows:



$\mathcal{A} =$

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
v_1		x	x	x				
v_2			x		x			x
v_3				x	x	x		
v_4		x			x		x	
v_5						x	x	x
v_6	x			x			x	
v_7	x	x						x
v_8	x		x			x		

Figure 8.2.4: The constructed $NRD(8,3,1,0)$ and its adjacency matrix $\mathcal{A}$.

As can be seen, the previous graph G is not a tournament because, for example, vertices v_0 and v_4 are not adjacent, and clearly it has the form as in possibility (3) in Equation 8.10. So it is an $NRD(8,3,1,0)$.

If an $NRD(v,k,\lambda,0)$ with $v = 2k+1$ is known, then we can construct an $NRD(2v+2,v,k,0)$ as described above. For example, we have seen how we constructed the $NRD(8,3,1,0)$ from the existing $NRD(3,1,0,0)$. Another example is to construct an $NRD(16,7,3,0)$ from the $NRD(7,3,1,0)$ in the same way.

We notice that the constructed digraphs with parameters $(2v + 2, v, k, 0)$ have edges such that each four vertices form a cycle, and if we know that $v_i \to v_j$ for example, then we can complete the cycle directly to be $v_i \to v_j \to v'_i \to v'_j \to v_i$ and so on.

Moreover, we notice that the automorphism group order of the constructed digraph is related to the automorphism group order of the tournament. If we denote the tournament by T, and the constructed digraph by G. Then, we have the following relation:

$$|Aut(G)| = |Aut(T)| \cdot |V(G)|$$

where $V(G)$ is the vertex set of the constructed digraph. That is because we can map any vertex from the vertex set of the constructed digraph, say v_i, to the vertex v_0 and then we have to map v'_i to v'_0 and everything else will be settled down to preserve adjacencies, that is to get the same digraph. Next, we multiply the order of the automorphism group of T by $|V(G)|$ since we have $|V(G)|$ vertices that can be mapped to vertex v_0.

A small example of this would be our previous example, which is the construction of $NRD(8, 3, 1, 0) = G$ from the doubly regular tournament $NRD(3, 1, 0, 0) = T$. In that example, we have $|Aut(T)| = 3$ and hence $|Aut(G)| = 3{\cdot}8 = 24$, which is true. Another example is the $NRD(11, 5, 2, 0) = T$ which was used to construct the $NRD(24, 11, 5, 0) = G$. And we have that $|Aut(G)| = |Aut(T)| \cdot 24 = 55 \cdot 24 = 1320$. The same idea works on the construction of the $NRD(32, 15, 7, 0)$ from the $NRD(15, 7, 3, 0)$.

As a result of these connections for the case when $\mu = 0$, we will not be interested in searching for those digraphs in the (3) as in the theorem. However, in the case where the NRD is a doubly regular tournament, μ can be equal to 0, but still it is an interesting digraph since it can not be constructed from a smaller graph.

8.3 NRDs with $\mu = k$

In this section, we consider the case of $\mu = k$.

Theorem 8.3.1. *A graph G is an NRD with $\mu = k$ if and only if there is a number s such that G is obtained from a doubly-regular tournament by replacing each vertex x by a set V_x of s new vertices such that if $x \to y$ in the tournament then $u \to w$ for every $u \in V_x$ and $w \in V_y$. Then, $s = k - 2\lambda = v - 2k$.*

In other words, a graph is an NRD with $\mu = k$ if and only if it has an adjacency matrix which is the Kronecker product of an adjacency matrix of a doubly regular tournament and the all $(s \times s)$ 1's matrix.

Proof. ($\Leftarrow$) It is easy to check that every graph obtained from an $NRD(v, k, \lambda, \bullet)$-tournament as described is an $NRD(sv, sk, , s\lambda, sk)$.

($\Rightarrow$) Suppose now that G is an NRD with $\mu = k$. Suppose $x, y \in G$ and $x \not\to y$. Then, as $\mu = k$, every vertex which dominates x also dominates y. If z is another vertex non-adjacent to x, then every vertex which dominates x also dominates z, then $d^-(y, z) = k$ since, $\lambda \leq \frac{k-1}{2}$, y and z are non-adjacent. Therefore, non-adjacency is a transitive relation on the vertex-set, which is therefore partitioned into classes, say $V_1, V_2, \ldots, V_r$, such that any two vertices are adjacent if and only if they belong to distinct classes.

Suppose that $x \in V_i$ dominates $y \in V_j$, for some $1 \leq i, j \leq r$. If $x \neq x' \in V_i$, then $x \not\to x'$ and thus $x' \to y$. Moreover, if $y \neq y' \in V_j$, then $x \to y'$ and $x' \to y'$. This is because $\mu = k$. Thus, every vertex in V_i dominates every vertex in V_j. Therefore,

$$2k = d^-(x) + d^-(y) = v - |V_j| + \lambda - \lambda = v - |V_j|$$

since every vertex of G dominates either x or y, except for the vertices in V_j and the λ vertices that are dominated by both x and y, and since there are also λ vertices which dominate both x and y. Similarly:

$$2k = d^+(x) + d^+(y) = v - |V_i|$$

It follows that $|V_i| = |V_j| = v - 2k$. Denote this number by s. Then, s divides k and λ, since the out-degree now is sk and the common out-neighbors is $s\lambda$. Therefore, a digraph with vertex-set $\{V_1, V_2, \ldots, V_r\}$ and edges $V_i \to V_j$ if the vertices of V_i dominate the vertices of V_j in G, is obviously a doubly-regular tournament with degree $\frac{k}{s}$ and double degree $\frac{\lambda}{s}$, so that $\frac{k}{s} = 2\frac{\lambda}{s} + 1$ and $k = 2\lambda + s$. □

As an example, consider the $NRD(3, 1, 0, 1)$, see Figure 8.2 and apply the Kronecker product to the all one (2×2) matrix to construct the $NRD(6, 2, 0, 2)$, see Figure 8.1.1, as follows:

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}_{3\times 3} \otimes \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}_{2\times 2} = \left[\begin{array}{cc|cc|cc} 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ \hline 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \end{array}\right]_{6\times 6}$$

As in the theorem above, $s = 2$, and thus we construct an $NRD(2 \cdot 3, 2 \cdot 1, 2 \cdot 0, 2 \cdot 1)$, with $v = 6, k = 2$, and $\lambda = 0$. Therefore, $s = v - 2k$ and $s = k - 2\lambda$ is satisfied.

As a consequence of the previous theorem, the following corollary gives a necessary condition for the parameters in order for an NRD with $\mu = k$ to exist.

Corollary 8.3.2. *An $NRD(v, k, \lambda, k)$ with a prime v does not exist if it is not a tournament.*

Clearly if such a NRD exists then there exists a tournament with v' vertices which was used to obtain that NRD with vertices v such that $s \cdot v' = v$ which is not possible for $s > 1$. For instance, we know that $NRD(29, 11, 2, 11)$ and $NRD(29, 13, 5, 13)$ do not exist because of the previous corollary.

Moreover, as in the previous section, the constructed graph is related somehow to the original graph (tournament) with respect to the automorphism group orders. The automorphism group order of the constructed graph, say G, is related to the automorphism group order of the tournament T in the following relation:

$$|Aut(G)| = (s!)^{|V(T)|} \cdot |Aut(T)|$$

where $s = v - 2k$ in the constructed graph. This is because we are replacing each vertex in the tournament by s new vertices. So we can map any vertex of those s vertices to any other vertex in the same class, and because we have $|V(T)|$ vertices or s new sets, we get our relation.

For example, let $G = NRD(9, 3, 0, 3)$, which was constructed using $T = NRD(3, 1, 0, 1)$, and we have that $|Aut(G)| = (3!)^3 \cdot 3 = 648$ which is true. Also, the construction of $G = NRD(30, 14, 6, 14)$ using $T = NRD(15, 7, 3, 7)$ where we have $s = 30 - 28 = 2$ and $|Aut(T)| = 21$ and by using the previous relation we have

$$|Aut(G)| = (2)^{15} \cdot 21 = 688128$$

8.4 Constructing NRDs

In this section, we describe a procedure which allows us to generate exhaustively NRDs with given parameters v, k, λ, and μ. Note that for those NRDs, it is strongly advised to generate such structures using adjacency matrices rather than incidence matrices. However, for the isomorphism rejection techniques, it is always possible to build the corresponding incidence matrix to apply the isomorph-rejection tests described in the previous chapter. If A is an adjacency matrix of an NRD G, then one can construct the corresponding incidence matrix B by Definition 2.5.2. It is clear that it is much easier to check the λ- and μ-conditions on A rather than on B.

Assuming that we generate NRDs with respect to their adjacency matrices, we proceed to generating an NRD, say G, as follows: Suppose that we started from the zeros adjacency matrix (i.e. empty digraph), then we want to fill in each row with k 1's, because each vertex in G has out-degree equal to k, such that we fill the 1's as far left as possible in the matrix. This is because we are using the first-breadth strategy for constructing rows of the matrix. Later on we come back to shift the last one on the right one step to the right so that we cover all possibilities for any given row.

Thus the first row of A has 0 in the first (diagonal) entry because no loop is allowed. The following k entries are 1, and the remaining entries are 0. The second row would have 0 on the first entry (since $A + A^T$ is a $\{0,1\}$-matrix), 0 on the second (diagonal) entry, and 1 on the following λ entries (since vertex 1 and vertex 2 are adjacent), 0 on the next $k - \lambda - 1$ entries and $k - \lambda$ entries with 1, with the remaining entries filled in with 0.

Suppose that we have filled in rows $0, 1, \ldots, r$. Then row r must satisfy the following:

- It has exactly k entries 1 and $v - k$ entries 0,
- Entry r (diagonal) must be 0,
- For $i < r$, entry i is 0 if entry r in row i is 1,
- If the matrix has 0 in entry (r, i) and in entry (i, r), then the dotproduct of row r and row i equals to μ, otherwise equals to λ for $i = 0, 1, \ldots, r-1$,

For each possible way to fill in row r we repeat this procedure with r replaced by $r+1$ and so on, until either we find some r for which no row satisfies the conditions or else all v rows are completed and the desired adjacency matrix has been reached. In the first case, when we reach some row r which does not satisfy the condition, we go back to row $r-1$ and change it trying to find another row which satisfies the conditions. If we succeed, then we go on to row r and try one more time. Otherwise, we go back again to row $r-2$ and so on until we find a suitable row r and continue or until we go back to row 0 and then stop, concluding that no such NRD exists.

This construction strategy assumes that we construct an adjacency matrix of an NRD row by row starting from row 0 and then row 1 and so on until we construct every possible NRD if any.

8.5 Results of Normally Regular Digraphs

In this section we present some of the results found in the search for normally regular digraphs. Note that we do not show NRDs results, because they can be obtained from what is in the table according to Sections 8.2 and 8.3 for $\mu = 0$ or k. In particular, we show only NRDs with $\mu \neq 0, k$ and those which are doubly regular tournaments.

Note that we use the adjacency matrix of an NRD to check for the λ- and μ-conditions, see Section 8.4. In the table below, we display the results of our search in the column "new" and the results that were found in [44, 45] by Jørgensen in the column "old". On the left side we have the results related to NRDs whose $\mu = \lambda$ or $\lambda + 1$. The part on the right side is related to NRDs whose $\mu \notin \{0, \lambda, \lambda+1, k\}$. The "$\star$" in the table means that we obtained the same results as Jørgensen, so we do not repeat the entry in the old column in the new column.

In the right hand side table, the case $(36, 7, 0, 2)$ was the easiest case in Jørgensen table in [45], where in our table (right hand side) the case $(19, 6, 1, 3)$ was the easiest since a complete search was done in 1 second. Moreover, the most difficult case for our search was the case $(31, 10, 2, 5)$ which was the most difficult case for Jørgensen also, but this case in our search was completed in 3.5 hours while in Jørgensen's it was completed in 30 hours. We note that Jørgensen's search was based on the generation by canonical representatives [45], while ours was based on canonical augmentation, by the meaning of McKay's μ-function.

Table 8.1: Normally Regular Digraphs

v	k	λ	μ	**old**	**new**	v	k	λ	μ	**old**	**new**
7	2	0	1	1	$\star$	7	3	1	1	1	$\star$
7	3	1	1	1	$\star$	19	6	1	3	1	$\star$
11	5	2	2	1	$\star$	21	8	3	2	1	$\star$
11	4	1	2	0	$\star$	23	8	2	4	0	$\star$
13	3	0	1	5	$\star$	25	8	3	1	0	$\star$
13	4	1	1	4	$\star$	27	8	1	4	0	$\star$
15	6	2	3	0	$\star$	27	10	3	5	≥ 1	$\star$
15	6	3	3	2	$\star$	29	7	2	1	4	$\star$
16	5	1	2	16	$\star$	31	10	2	5	0	$\star$
16	6	2	2	4	$\star$	31	10	4	1	0	$\star$
19	8	3	4	0	$\star$	31	12	4	6	≥ 1	$\star$
19	9	4	4	2	$\star$	35	10	1	5	0	$\star$
21	4	0	1	187	$\star$	36	7	0	2	2	$\star$
21	5	1	1	> 1000	$\geq 200,000$						
23	10	4	5	0	$\star$						
23	11	5	5	37	$\star$						
25	8	2	3	≥ 1	$\geq 4,070$						
25	9	3	3	≥ 1	≥ 36						
27	12	5	6								
27	13	6	6	722	$\star$						
31	15	7	7		$\geq 13,330$						
31	5	0	1		$\geq 65,000$						
31	6	1	1		$\geq 50,000$						

Bibliography

[1] Ákos Seress. *Permutation Group Algorithms.* Cambridge University Press, 2003.

[2] L. M. Batten and A. Beutelspacher. *The Theory of Finite Linear Spaces. Combinatorics of Points and Lines.* Cambridge University Press, sixth printing edition, 1993.

[3] Anton Betten and D. Betten. Regular linear spaces. *Beiträge zur Algebra und Geometrie*, 38:111–124, 1997.

[4] Anton Betten and D. Betten. Linear spaces with at most 12 points. *Journal of Combinatorial Designs*, 7:119 – 145, 1999.

[5] Anton Betten and D. Betten. The proper linear spaces on 17 points. *Discrete Applied Mathematics*, 95:83 – 108, 1999.

[6] Anton Betten and D. Betten. Tactical decompositions and some configurations v_4. *Journal of Geometry*, 66:27 – 41, 1999.

[7] Anton Betten, Michael Braun, Harald Fripertinger, Adalbert Kerber, and Axel Kohnert. *Error-correcting Linear Codes : Classification by Isometry And Applications.* Springer, October 2006.

[8] Anton Betten, G. Brinkmann, and T. Pisanski. Counting symmetric configurations v_3. *Discrete Applied Mathematics*, 99:331 – 338, 2000.

[9] D. Betten and M. Braun. A tactical decomposition for incidence structures, combinatorics '90. *Ann. Discrete Mathematics*, 52:37 – 43, 1992.

[10] Dieter Betten and Ulrike Schumacher. The ten configurations 10_3. *Rostock. Math. Kolloq.*, 46:3–10, 1993.

[11] G. Brinkmann. Isomorphism rejection in structure generation programs. *Discrete Mathematics Chemistry*, pages 25 – 38, 2000. (P. Hansen and P. Fowler and M. Zheng , Eds.), Amer. Math. Soc.

[12] G. Brinkmann and B. D. McKay. Posets on up to 16 points. *Order 19*, 2:147 – 179, 2002.

[13] G. Brinkmann and M. Meringer. The smallest 4-regular 4-chromatic graphs with girth 5. *Graph Theory Notes of New York XXXII*, pages 40 – 41, 1997.

[14] A. Richard Brualdi. *Introductory Combinatorics.* PRENTICE HALL, second edition edition, 1992.

[15] G. Butler. *Fundamental Algorithms for Permutation Groups.* Springer Verlag, Berlin, 1991.

[16] P. J. Cameron. *Combinatorics: Topics, Techniques, Algorithms.* Cambridge University Press, Cambridge 1994.

[17] P. J. Cameron. *Permutation Groups.* Cambridge University Press, Cambridge 1999.

[18] Arjeh M. Cohen and Scott H. Murray. An automated proof theory approach to computation with permutation groups. 2002.

[19] D. G. Corneil and C. C. Gotlieb. An efficient algorithm for graph isomorphism. *Journal of the Association for Computing Machinery*, 17 No. 1:51 – 64, January 1970.

[20] D. John Dixon and Brian Mortimer. *Permutation Groups.* Springer, New York, 1996, 1996.

[21] J. Doyen. Sur le nombre d'espaces linéaires non isomorphes de n points. *Bull. Soc. Math. Belg.*, 19:421 – 437, 1967.

[22] P. Erdös and H. Sachs. Reguläre graphen gegebener taillenweite mit minimaler knotenzahl. *Wiss. Z. Univ. Halle, Math.-Nat. Reihe*, XII/3:251–258, 1963.

[23] I. A. Faradžev. Constructive enumeration of combinatorial objects. *Problèmes Combinatoires et Théorie des Graphes*, pages 131 – 135, 1978. (Université d'Orsay, July 9-13 1977), CNRS, Paris.

[24] Jay P. Fillmore and S. G. Williamson. On backtracking: A combinatorial description of the algorithm. *SIAM J. Comput.*, 3 No. 1:41 – 55, 1974.

[25] Noah Giansiracusa. Determining graph isomorphism via canonical labeling. *http://sage.math.washington.edu/home/rlmill/graph-isom/Isomorphism.pdf*, Octobor 2003.

[26] D. G. Glynn. Rings of geometries ii. *J. Combin. Theory*, A 49:26–66, 1988.

[27] C. D. Godsil. *Algebraic Combinatorics.* Chapman And Hall Mathematics, first edition, 1993.

[28] Leslie Ann Goldberg. Efficient algorithms for listing unlabeled graphs. *Journal Of Algorithms*, 13:128 – 143, 1992.

[29] Harald Gropp. On the existence and non-existence of configurations n_k. *Journal Of Combinatorics*, 15:34 – 48, 1990.

[30] Harald Gropp. Non-symmetric configurations with deficiencies 1 and 2. *Annals Of Discrete Mathematics*, 52:227 – 239, 1992.

[31] Harald Gropp. Blocking set free configurations and their relations to digraphs and hypergraphs. *Discrete Mathematics*, 165,166:359 – 370, 1997.

[32] Harald Gropp. Configurations and graphs - ii. *Discrete Mathematics*, 164:155 – 163, 1997.

[33] Jonathan L. Gross and Jay Yellen. *Handbook of Graph Theory.* CRC Press LLC, 2004.

[34] M. Hall. *Combinatorial Theory.* Wiley, New York, 1980, Blaisdell, Waltham, MA, second edition edition, 1967.

[35] James L. Hein. *Discrete Mathematics.* JONES AND BARTLETT, Blaisdell, Waltham, MA, second edition edition, 2003.

[36] D. G. Higman. Invariant relations, coherent configurations, and generalized polygons. *Mathematical Centre Tracts*, 57:27–43, 1974.

[37] D. G. Higman. Coherent configuration, part i. ordinary representation theory. *Geom. Dedicata*, 4:1–32, 1975.

[38] Christoph M. Hoffmann. *Lecture Notes in Computer Science: Group-Theoretic Algorithms And Graph Isomorphism.* Vol. 136, Springer-Verlag, New York-Berlin, Department of Mathematical Sciences, Aalborg University, 1982.

[39] N. Ito. On hadamard tournaments. *J. Algebra*, 131:432 – 443, 1990.

[40] N. Ito. Doubly regular asymmetric digraphs. *Discrete Mathematics*, 72:181 – 185, 1998.

[41] J. Köbler, U. Schöning, and J. Torán. *Graph Isomorphism Problem.* Birkhäuser Boston, 1993.

[42] Charles J.Colbourn and Jeffrey H. Dinitz. *The CRC Handbook of Combinatorial Designs.* Chapman and CRC, second edition edition, 2006.

[43] Leif K. Jørgensen. Bipartite graphs with small diameter and defect 2. *Institute for Electronic Systems, Aalborg University.*

[44] Leif K. Jørgensen. On normally regular digraphs. *Department of Mathematical Sciences, Aalborg University*, 1999.

[45] Leif K. Jørgensen. Search for directed strongly regular graphs. *Department of Mathematical Sciences, Aalborg University*, 1999.

[46] Leif K. Jørgensen. Non-symmetric 3-class association schemes. *Department of Mathematical Sciences, Aalborg University*, 2005.

[47] Petteri Kaski. Isomorph-free exhaustive generation of designs with prescribed groups of automorphisms. *SIAM Journal on Discrete Mathematics.* to be appear.

[48] Petteri Kaski. Isomorph-free exhaustive generation of combinatorial designs. *Research Report*, A70, 2001. Laboratory For Theoretical Computer Science, Helsinki University of Technology.

[49] Petteri Kaski. Algorithms for classification of combinatorial objects. 2005. Research Report.

[50] Petteri Kaski and P. R. J. Óstergård. There exists no symmetric configuration with 33 points and line size 6. *Australasian Journal of Combinatorics, to appear.*

[51] Petteri Kaski and P. R. J. Óstergård. The steiner triple systems of order 19. *Mathematics of Computation*, 73:2075 – 2092, 2004.

[52] Petteri Kaski and Patric R. J. Óstergård. *Classification Algorithms For Codes And Designs.* Springer, 2006.

[53] Adalbert Kerber. *Algebraic Combinatorics Via Finite Group Actions.* Springer, second edition, 1999.

[54] W. Kocay. On writing isomorphism programs. *Computational And Constructive Design Theory*, 72:135 – 175, 1996.

[55] D. L. Kreher and D.R. Stinson. *Combinatorial Algorithms: Generating, Enumeration, and Search.* CRC press, Boca Raton, Fla., 1999.

[56] C. Lam, L. Thiel, and S. Swiercz. The nonexistence of finite projective planes of order 10. *Canadian Journal of Mathematics*, 41:1117 – 1123, 1989.

[57] R. Laue. Construction of combinatorial objects, a tutorial. *Bayreuther Mathematische Schriften*, 43:53 – 96, 1993.

[58] W. Ledermann. *Introduction To Group Theory.* Cambridge University Press, second edition, 1973.

[59] J. H. Van Lint and R. M. Wilson. *A Course in Combinatorics.* Cambridge University Press, second edition, 2001.

[60] W. F. McGee. A minimal cubic graph of girth seven. *Canad. Math. Bull*, 3:149–152, 1960.

[61] Brendan D. McKay. nauty user's guide (version 2.2). Manual.

[62] Brendan D. McKay. Computing automorphisms and canonical labelling of graphs. *Lecture Notes in Mathematics*, 686:223 – 232, 1977. International Conference On Combinatorial Mathematics.

[63] Brendan D. McKay. Practical graph isomorphism. *Congressus Numerantium.*, 30:45 – 87, 1981.

[64] Brendan D. McKay. Isomorph-free exhaustive generation. *J. Algorithms*, 26:306 – 324, 1998.

[65] Brendan D. McKay, W. Myrvold, and J. Nadon. Fast backtracking principles applied to find a new cages. *in Proc. 9th ACM-SIAM Symposium on Discrete Algorithms*, pages 188 – 191, 1998. San Francisco, Jan. 25-27, 1998.

[66] M. Meringer. Fast generation of regular graphs and construction of cages. *J. Graph Theory*, 30:137 – 146, 1999.

[67] Patric R. J. Óstergård. A fast algorithm for the maximum clique problem. *Discrete Applied Mathematics*, 120:197 – 207, 2002.

[68] Ch. Pietsch. On the classification of linear spaces of order 11. *J. of Comb. Designs*, 3:185–193, 1995.

[69] R. C. Read. Every one a winner; or, how to avoid isomorphism search when cataloguing combinatorial configurations. *Annals Discrete Mathematics*, 2:107 – 120, 1978.

[70] K. B. Reid and E. Brown. Doubly regular tournaments are equivalent to skew hadamard matrices. *J. Combinatorial Theory*, (A) 12:332–338, 1972.

[71] Fred S. Roberts and B. Tesman. *Applied Combinatorics.* Prentice Hall, Upper Saddle River, second edition, 2005.

[72] Gordon F. Royle. An orderly algorithm and some applications in finite geometry. *Discrete Mathematics*, pages 105 – 115, 1998.

[73] Horst Sachs. Regular graphs with given girth and restricted circuits. *J. London Math. Soc.*, 38:423–429, 1963.

[74] Horst Sachs. *Einführung In Die Theorie Der Endlichen Graphen.* BSB B.G. Teubner Verlagsgesellschaft, first edition, 1970.

[75] J. S.Leon. Permutation group algorithms based on partitions, i: Theory and algorithms. *J. Symbolic Computations*, 12:533 – 583, 1991.

[76] J. S.Leon. Partitions, refinements, and permutation group computation, groups and computation, ii. *American Mathematics Society*, pages 123 – 158, 1997.

[77] L. H. Soicher. *Computing with graphs and groups, in Topics in Algebraic Graph Theory.* Cambridge University Press, 2004.

[78] J. D. Swift. Isomorph rejection in exhaustive search techniques. *Proceedings Of Symposia In Applied Mathematics*, 10:195 – 200, 1960.

[79] Donald E. Taylor. *The Geometry of The Classical Groups.* Heldermann Verlag, Berlin, sigma series in pure mathematics, volume 9 edition, 1992.

[80] W. T. Tutte. A family of cubical graphs. *Proc. Cambridge Phil. Soc.*, 43:459–575, 1947.

[81] G. Valiente. *Algorithms On Trees And Graphs.* Springer, 2002.

[82] Volker Widor. Ein algorithmus zur oarameterverfeinerung im zusammenhang mit der erzeugung endlicher geometrien. *Master Thesis, Mathematisch-Naturwissenschaftlichen Fakultät der Christian-Albrechts-Universität zu Kiel*, November 2003.

Index