5-2009

Applications of the Combinatorial Nullstellensatz on bipartite graphs.

Timothy M. Brauch
University of Louisville

APPLICATIONS OF THE COMBINATORIAL NULLSTELLENSATZ ON BIPARTITE GRAPHS

By

Timothy M. Brauch
B.S., Centre College, 2002
M.A., Wake Forest University, 2004
M.A., University of Louisville, 2007

A Dissertation
Submitted to the Faculty of the
Graduate School of the University of Louisville
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy

Department of Mathematics
University of Louisville
Louisville, KY

May 2009

APPLICATIONS OF THE COMBINATORIAL NULLSTELLENSATZ ON BIPARTITE GRAPHS

Submitted by

Timothy M. Brauch

A Dissertation Approved on

April 1, 2009

(Date)

by the Following Reading and Examination Committee:

André Kézdy, Dissertation Director

Dale B. Billingsley

Ewa Kubicka

Grzegorz Kubicki

Shi-Yu Wu

ACKNOWLEDGEMENTS

To my advisor, Professor André Kézdy, I am grateful for his patience and assistance. Even though our work styles differ, he was always available for me and kept pushing me to do my best. Through this, I have become a better mathematician.

To Professor Gregorz Kubicki who inspired me to choose discrete mathematics as my area of study.

To Professors Eva Kubicka, Shi-Yu Wu, and Dale Billingsley who spent time reading and understanding topics that might not be in their own areas of research and provided important guidance.

To the mathematics faculty at the University of Louisville, Wake Forest University, and Centre College, especially Dr. Bill Johnston, who encouraged me and supported me through my education and for believing in me and encouraging me to keep going.

To Lesley, Adam, and Ben who kept me grounded as we discussed mathematics in the office as well as outside of the office.

To my family who, while they might not have understood me most of the time, were always willing to at least humor me and only asked when I would finally graduate a few times.

ABSTRACT

APPLICATIONS OF THE COMBINATORIAL NULLSTELLENSATZ ON BIPARTITE GRAPHS

Timothy M. Brauch

May 9, 2009

The Combinatorial Nullstellensatz can be used to solve certain problems in combinatorics. However, one of the major complications in using the Combinatorial Nullstellensatz is ensuring that there exists a nonzero monomial. This dissertation looks at applying the Combinatorial Nullstellensatz to finding perfect matchings in bipartite graphs.

The first two chapters provide background material covering topics such as linear algebra, group theory, graph theory and even the discrete Fourier transform. New results start in the third chapter, showing that the Combinatorial Nullstellensatz can be used to solve the problem of finding perfect matchings in bipartite graphs. Using the Combinatorial Nullstellensatz also allows for a nice use of matroid intersection to find the nonzero monomial. By also applying the uncertainty principle, the number of perfect matchings in a bipartite graph can be bound.

The fourth chapter examines properties of the polynomials created in the use of the Combinatorial Nullstellensatz to find perfect matchings in bipartite graphs. Many of the properties of the polynomials have analogous properties for the transforms of the polynomials, which are also examined. These properties often relate back to the structure of the graph which gave rise to the polynomial.

The fifth chapter provides an application of the results. Since finding a

nonzero monomial can be difficult and the polynomials created in this dissertation give polynomials with such a nonzero monomial the application shows how certain polynomials can be rewritten in terms of the matching polynomials. Such a rewriting may permit an easy method to find a nonzero monomial so that the Combinatorial Nullstellensatz can be applied to the polynomial. Finally, the fifth chapter concludes with some open problems that may be areas of further research.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
LINEAR ALGEBRA, GRAPH THEORY, AND GROUP THEORY

This chapter provides an introduction to the *linear algebra*, *graph theory*, and *group theory* concepts that are used in this dissertation. This introduction is not meant to be an exhaustive list of these concepts nor as an overview of these areas in general but rather is intended to cover the topics important to this dissertation.

Notations and definitions for the linear algebra topics follow from those used in Lay [15]. The definitions and notation for graphs is based on what is used in Chartrand [5] and in Wilson [24]. For the group theory topics the notations and definitions are based on the Gallian [9] and Artin [2].

1.1 Matrices and Vectors

A *matrix* is a rectangular array and will be written either as a capital letter such as M or between brackets, as seen in Matrix 1.1. The elements in the cells are usually numbers or variables but can be any object which allows itself to be added and multiplied. Matrices are often used to describe systems of linear equations or, as is shown in Section 1.2, the adjacency and incidence of a set of vertices from a graph.

A *linear equation* in the variables $x_1, x_2, \ldots, x_n$ is an equation that can be written in the form $a_1x_1 + a_2x_2 + \ldots + a_nx_n = a_0$ where each a_i is a number, possibly complex. A *system of linear equations* is a collection of one or more linear equations

in the same variables such as

$$\begin{aligned} a_{1,1}x_1 + a_{1,2}x_2 + \ldots + a_{1,n}x_n &= a_{1,0} \\ a_{2,1}x_1 + a_{2,2}x_2 + \ldots + a_{2,n}x_n &= a_{2,0} \end{aligned} \tag{1.1}$$

where some of the $a_{i,j}$ might be zero.

The system of linear equations in (1.1) can be written

$$\left[\begin{array}{cccc|c} a_{1,1} & a_{1,2} & \ldots & a_{1,n} & a_{1,0} \\ a_{2,1} & a_{2,2} & \ldots & a_{2,n} & a_{2,0} \end{array}\right] = [a_{i,j}]$$

Matrix 1.1: Matrix form of (1.1)

with the coefficients of the linear equations making up the entries of the matrix.

The horizontals in a matrix are called *rows* and the verticals are called *columns*. The *dimensions* of a matrix are the number of rows and the number of columns typically written as $m \times n$ where m is the number of rows and n is the number of columns. The dimensions are always given with the number of rows before the number of columns. A matrix with the same number of rows as columns, an $n \times n$ matrix, called a *square* matrix, is of importance in this dissertation.

If a matrix has only one row or only one column it is called a *vector*. A $1 \times n$ matrix is called a *row*-vector and a $m \times 1$ matrix is called a *column*-vector. Each row (or column) of a matrix can also be considered a row (or column) vector. To denote a vector the symbol $\overrightarrow{v}$ is used or when the vector is described explicitly it will be written between angle brackets, as $\langle v_0, v_1, \ldots, v_{n-1} \rangle$.

A matrix can be multiplied by a number c, also called a *scalar* by multiplying each entry of the matrix by c. Two matrices can be added, as long as they have the same dimensions, by adding each entry in the first matrix to the corresponding entry in the second matrix.

An entry of a matrix is referenced by the row and column in which the entries

lies. That is, for a matrix A the value in the ith row and jth column is denoted by $a_{i,j}$.

A matrix is called *symmetric* if $a_{i,j} = a_{j,i}$ for all values of i and j. The *transpose* of a matrix A is the matrix A^T where the columns of A form the rows of A^T and the rows of A form the columns of A^T, as shown, for example, in Matrices 1.2 and 1.3. If a matrix A is symmetric, then $A = A^T$.

$$\left[\begin{array}{cccc} a_1 & a_2 & \dots & a_n \\ \hline b_1 & b_2 & \dots & b_n \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline c_1 & c_2 & \dots & c_n \end{array}\right]$$

Matrix 1.2: The Matrix A

$$\left[\begin{array}{c|c|c|c} a_1 & b_1 & \dots & c_1 \\ a_2 & b_2 & \dots & c_2 \\ \vdots & \vdots & \ddots & \vdots \\ a_n & b_n & \dots & c_n \end{array}\right]$$

Matrix 1.3: The Matrix A^T

A *linear combination* of vectors $\overrightarrow{v_1}, \overrightarrow{v_2}, \dots, \overrightarrow{v_n}$ is a vector $\overrightarrow{v}$ that can be written as a sum of the vectors $\overrightarrow{v_1}$ through $\overrightarrow{v_n}$ where each $\overrightarrow{v_i}$ has been multiplied by an appropriate constant: $\overrightarrow{v} = c_1 \times \overrightarrow{v_1} + \dots c_n \times \overrightarrow{v_n}$. A collection of vectors is called *linearly independent* or said to exhibit *linear independence* if any one vector cannot be written as a linear combination of the other vectors. That is, a set of n vectors is *linearly independent* if the only coefficients that make the equation

$$a_1\overrightarrow{v_1} + a_2\overrightarrow{v_2} + \dots + a_n\overrightarrow{v_n} = \overrightarrow{0} \tag{1.2}$$

true are all equal to zero. It should be noted that the right hand side of (1.2) is a

vector of all zeros, not the number zero. Vectors that are not linearly indenpendent are said to be *linearly dependent.*

The set of all linear combinations of a set of vectors $\overrightarrow{v_1}, \ldots, \overrightarrow{v_p}$ is denoted $\text{Span}\{\overrightarrow{v_1}, \ldots, \overrightarrow{v_p}\}$ and called the *subset spanned by* $\overrightarrow{v_1}, \ldots, \overrightarrow{v_p}$. That is to say that $\text{Span}\{\overrightarrow{v_1}, \ldots, \overrightarrow{v_p}\} = c_1\overrightarrow{v_1}, \ldots, c_p\overrightarrow{v_p}$ for all possible scalars $c_1, \ldots, c_p$.

If the vectors that make up the rows (or columns) of a square matrix are linearly independent the matrix is said to be *nonsingluar*. Otherwise the matrix is called *singular*.

For a permutation σ let $\text{sgn}(\sigma) = (-1)^{(k)}$, where $\sigma = \tau_1 \ldots \tau_k$ for transpositions τ_i. Then, for a square matrix, A, the *determinant* of A is a scalar associated with the matrix given by the equation:

$$\det(A) = \sum_{\sigma \in S_n} \text{sgn}\,(\sigma) \prod_{i=1}^{n} a_{i,\sigma(i)}$$

The *trace* of a square matrix is the sum of the entries along the diagonal, given by the equation $\text{Tr}(A) = \sum_{i=1}^{n} a_{i,i}$ where A is an $n \times n$ matrix.

A complex number $a + bi$ can be thought of as a vector $\langle a, b \rangle$. Using this convention, the absolute value (or *modulus*) of a complex number $|a + bi|$ is the norm of the vector $\langle a, b \rangle$ and thus $|a + bi| = \sqrt{a^2 + b^2}$.

A *vector space* V is a set of vectors that satisfy the following axioms. For all $\overrightarrow{u}$, $\overrightarrow{v}$, and $\overrightarrow{w}$ in V and α and β constants if V

1. contains the zero vector: $\overrightarrow{0} \in V$,
2. is closed under vector addition: $\overrightarrow{u} + \overrightarrow{v} \in V$,
3. is transitive: $\overrightarrow{u} + \overrightarrow{v} = \overrightarrow{v} + \overrightarrow{u}$,
4. is associative: $(\overrightarrow{u} + \overrightarrow{v}) + \overrightarrow{w} = \overrightarrow{u} + (\overrightarrow{v} + \overrightarrow{w})$,

5. contains additive inverses: for each $\overrightarrow{u}$ there exists $\overrightarrow{-u} \in V$ with $\overrightarrow{u} + (\overrightarrow{-u}) = \overrightarrow{0}$,

6. is closed under scalar multiplication: $\alpha\overrightarrow{u} \in V$,

7. has a scalar distributive property: $\alpha(\overrightarrow{u} + \overrightarrow{v}) = \alpha\overrightarrow{u} + \alpha\overrightarrow{v}$,

8. has a vector distributive property: $(\alpha + \beta)\overrightarrow{u} = \alpha\overrightarrow{u} + \beta\overrightarrow{u}$, and

9. has a scalar associative property: $\alpha(\beta\overrightarrow{u}) = (\alpha\beta)\overrightarrow{u}$,

then V is a vector space.

A *subspace* H of a vector space V is a subset of V that satisfies the following properties. For all $\overrightarrow{u}$ and $\overrightarrow{v}$ in H and α constant, if H

1. contains the zero vector: $\overrightarrow{0} \in H$.

2. is closed under vector addition: $\overrightarrow{u} + \overrightarrow{v} \in H$.

3. is closed under scalar multiplication: $\alpha\overrightarrow{u} \in H$.

then H is a subspace.

PROPOSITION 1.1. A subspace H of a vector space V is a vector space.

PROPOSITION 1.2. If $\overrightarrow{v_1}, \ldots, \overrightarrow{v_p}$ are in a vector space V, then $\text{Span}\{\overrightarrow{v_1}, \ldots, \overrightarrow{v_p}\}$ is a vector space and a subspace of V.

The proof of these two propositions follows by checking that the definitions are satisfied.

The *column space* of a matrix is another subspace associated with a matrix. The column space of an $m \times n$ matrix A, written as $\text{Col}(A)$, is the set of all linear combinations of the columns of A. If $A = [\overrightarrow{a_1}, \overrightarrow{a_2}, \ldots, \overrightarrow{a_n}]$, then $\text{Col}(A) = \text{Span}\{\overrightarrow{a_1}, \overrightarrow{a_2}, \ldots, \overrightarrow{a_n}\}$.

COROLLARY 1.1. Col(A) is a subspace.

This follows directly from the definition of Col(A) and Proposition 1.2.

Let H be a subspace of a vector space V and let $\mathcal{B} = \{\overrightarrow{b_1}, \ldots, \overrightarrow{b_p}\}$ be a set of vectors in V. If

1. $\mathcal{B}$ is a linearly independent set, and
2. $H = \text{Span}\{\overrightarrow{b_1}, \ldots, \overrightarrow{b_p}\}$,

then $\mathcal{B}$ is a *basis* for H.

The *dimension* of a vector space, subspace, or basis is the maximum number of linearly independent vectors in the space. The *rank* of a matrix A is the dimension of the column space Col(A).

1.2 Graphs

In mathematics a *graph* is a mathematical structure used to model relationships between objects. A graph $G(V, E)$, often abbreviated simply as G, is collection of two sets. The first set, known as the *vertex set*, is a non-empty, finite set of elements called *vertices* and is denoted by $V(G)$ or, when it is clear what graph is of interest, simply as V. The second set, known as the *edge set*, is a finite, possibly empty, set of unordered pairs of vertices of the graph G called *edges* and is denoted by $E(G)$. If the graph that is being discussed is clear, the edge set is often denoted simply as E. The cardinality of $V(G)$ is the number of vertices in G and is called the *order* of G. This number is denoted by $|V(G)|$. Tbe number of edges in a graph, denoted $|E(G)|$, is called the *size* of G. A graph with n vertices is said to be a graph on n vertices.

An edge, an unordered pair of vertices, $e = \{u, v\}$, often simply denoted $e = uv$, signifies there exists a relationship between the vertices u and v. In this

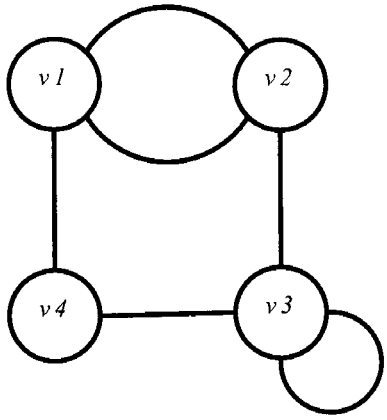

Figure 1.1 – A Multigraph

case u is said to be *adjacent* to v or u and v are adjacent. In this case u and v are also said to be *incident* to the edge e or that u and v lie on the edge e. If there exists another edge $f = vw$ then, since e and f are both incident to the vertex v, e and f are called incident edges.

An edge from a vertex to itself is called a *loop*. Pair of vertices that appear two or more times in $E(G)$ are *multiple edges*. A graph with loops or multiple edges is a *multi*graph. A graph that is not a multigraph is a *simple* graph. Unless otherwise specified, all graphs in this dissertation are simple graphs.

A graph is commonly represented as a diagram with the vertices drawn as small circles and the edges drawn as line segments, or arcs, connecting the circles. As an example, let G be a multigraph on four vertices with $V(G) = \{v_1, v_2, v_3, v_4\}$ and $E(G) = \{v_1v_2, v_1v_2, v_2v_3, v_3v_3, v_3v_4, v_1v_4\}$ (see Figure 1.1).

A *complete graph* is a graph such that for every u and v in $V(G)$ with $u \neq v$ the edge uv exists in G. The complete graph on n vertices is denoted K_n. As an example, let $G = K_4$ be the complete graph on four vertices with $V(G) = \{v_1, v_2, v_3, v_4\}$ then $E(G) = \{v_1v_2, v_1v_3, v_1v_4, v_2v_3, v_2v_4, v_3v_4\}$ (Figure 1.2).

Representing graphs using matrices is useful. Let G be a graph with $V(G) = \{v_1, v_2, \ldots, v_n\}$ and $E(G) = \{e_1, e_2, \ldots, e_m\}$, the *adjacency matrix* $A(G) = [a_{ij}]$ is

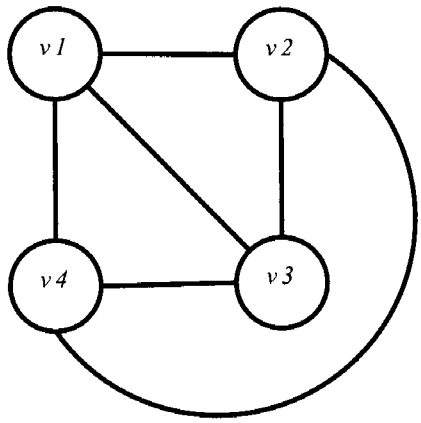

Figure 1.2 – The Complete Graph K_4

an $n \times n$ matrix where

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

An adjacency matrix for a graph G is always symmetric.

Incidence matrix representations are also useful. Let G be a graph with $V(G) = \{v_1, v_2, \ldots, v_n\}$ and $E(G) = \{e_1, e_2, \ldots, e_m\}$ the *incidence matrix* $B(G) = [b_{ij}]$ is an $m \times n$ matrix where

$$b_{ij} = \begin{cases} 1 & \text{if } v_i \text{ is incident to } e_j, \\ 0 & \text{if } v_i \text{ is not incident to } e_j. \end{cases}$$

The graph shown in Figure 1.3 has Matrix 1.4 as its adjacency matrix and Matrix 1.5 as its incidence matrix.

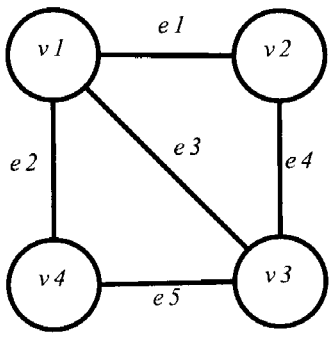

Figure 1.3: A Graph G

$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Matrix 1.4: The Adjacency Matrix of of the graph in Figure 1.3

$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{ccccc} e_1 & e_2 & e_3 & e_4 & e_5 \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

Matrix 1.5: The Incidence Matrix of the graph in Figure 1.3

A graph H is a *subgraph of* G if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. If H is a subgraph of G it is denoted $H \subseteq G$. An *induced* subgraph of a graph G is a graph $H \subseteq G$ such that for every u and v in $V(H)$ if $uv \in E(G)$, then $uv \in E(H)$. If U is some subset of $V(G)$, then the subgraph induced by U is written as $\langle U \rangle$ (Figure 1.4).

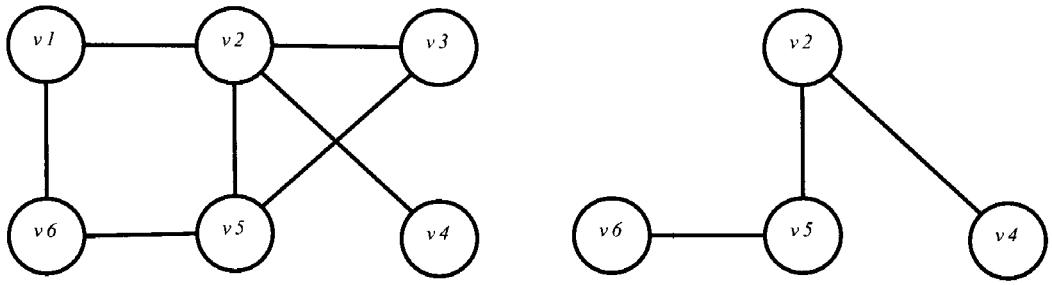

Figure 1.4: A Graph G and an Induced Subgraph H

A *bipartite graph* is a graph in which the vertices can be partitioned into two partite sets $V_1(G)$ and $V_2(G)$ with $V(G) = V_1(G) \cup V_2(G)$ such that every edge

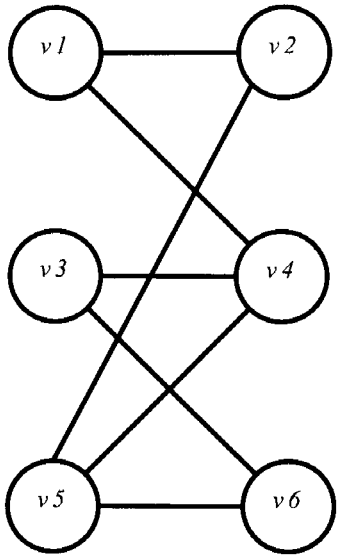

Figure 1.5 – A Bipartite Graph

$e \in E(G)$ contains one vertex from V_1 and one vertex from V_2. A bipartite graph is illustrated in Figure 1.5.

A *complete bipartite graph* is a bipartite graph such that for every $v_1 \in V_1(G)$ and $v_2 \in V_2(G)$ the edge v_1v_2 exists in G. If $|V_1(G)| = s$ and $|V_2(G)| = t$, then the complete bipartite graph on these vertices is denoted $K_{s,t}$. Let $G = K_{2,3}$ be a graph with partite sets $V_1 = \{u_1, u_2\}$ and $V_2 = \{v_1, v_2, v_3\}$. Then $V(G) = V_1(G) \cup V_2(G) = \{u_1, u_2, v_1, v_2, v_3\}$ and $E(G) = \{uv | u \in V_1 \text{ and } v \in V_2\} = \{u_1v_1, u_1v_2, u_1v_3, u_2v_1, u_2v_2, u_2v_3\}$ (Figure 1.6).

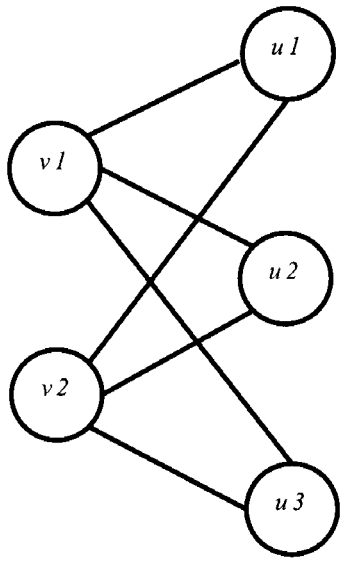

Figure 1.6: The Complete Bipartite Graph $K_{2,3}$

Since elements in the same partite set of a bipartite matrix are not adjacent, the adjacency matrix of a bipartite graph has a large number of zeroes as entries. If G is a bipartite graph with partite sets $V = \{v_1, v_2, \ldots, v_n\}$ and $U = \{u_1, u_2, \ldots, u_m\}$ the adjacency matrix has a *block* matrix form as seen in Matrix 1.6.

$$A = \begin{array}{c} \\ v_1 \\ \vdots \\ v_n \\ u_1 \\ \vdots \\ u_m \end{array} \begin{array}{c} \begin{array}{ccc} v_1 & \ldots & v_n \end{array} \\ \left[\begin{array}{ccc} 0 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & 0 \\ \hline a_{n+1,1} & \ldots & a_{n+1,n} \\ \vdots & \ddots & \vdots \\ a_{n+m,1} & \ldots & a_{n+m,n} \end{array}\right. \end{array} \begin{array}{c} \begin{array}{ccc} u_1 & \ldots & u_m \end{array} \\ \left.\begin{array}{ccc} a_{1,n+1} & \ldots & a_{1,n+m} \\ \vdots & \ddots & \vdots \\ a_{n,n+1} & \ldots & a_{n,n+m} \\ \hline 0 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & 0 \end{array}\right] \end{array}$$

Matrix 1.6: The Adjacency Matrix of a Bipartite Graph

Since the adjacency matrix for a graph is always symmetric, the matrix in Matrix 1.6 can be written in block form as in Matrix 1.7 where $\widehat{A}$ is the matrix as

given in Matrix 1.8. The matrix $\widehat{A}$ is called the *reduced adjacency matrix* of the bipartite graph.

$$A = \begin{array}{c} \\ V \\ U \end{array} \begin{array}{c} \begin{array}{cc} V & U \end{array} \\ \left[\begin{array}{c|c} 0 & \widehat{A} \\ \hline \widehat{A}^T & 0 \end{array}\right] \end{array}$$

Matrix 1.7: The Block Adjacency Matrix of a Bipartite Graph

$$\widehat{A} = \begin{array}{c} \\ v_1 \\ \vdots \\ v_n \end{array} \begin{array}{c} \begin{array}{ccc} u_1 & \cdots & u_m \end{array} \\ \left[\begin{array}{ccc} a_{1,n+1} & \cdots & a_{1,n+m} \\ \vdots & \ddots & \vdots \\ a_{n,n+1} & \cdots & a_{n,n+m} \end{array}\right] \end{array}$$

Matrix 1.8: The Reduced Matrix $\widehat{A}$

A *matching* is a set of edges, $M \subseteq E(G)$, in a graph G such that no two edges in M are incident. An example of a matching is given in Figure 1.7. Such a set is also called an *independent set* of edges. A vertex that is incident to an edge in the matching is said to be *matched* while a vertex not incident to an edge in the matching is called *unmatched.* The collection of matched vertices is also commonly said to be *covered* by the matching.

There are three major types of matchings:

1. maximal matchings
2. maximum matchings
3. perfect matchings.

All three of these matchings are extremal matchings in the sense that any other matching can be extended to one of these three by the addition of edges that are not incident to an edge already in the matching.

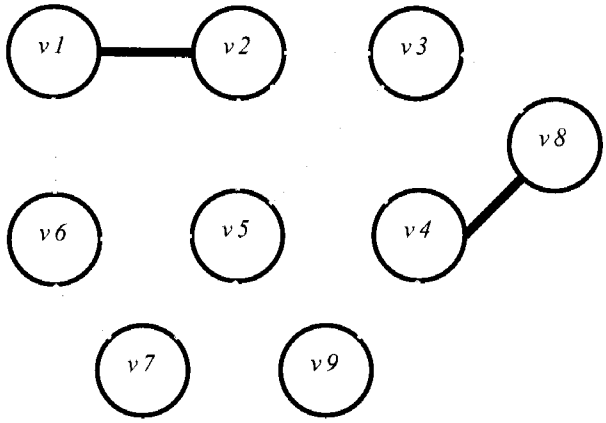

Figure 1.7 – A Matching

A *maximal matching* $M \subseteq E(G)$ is a matching of a graph G such that no other matching properly contains M. In a maximal matching all vertices not covered by the matching are only adjacent to vertices that are covered by the matching. An example of a maximal matching is given in Figure 1.8.

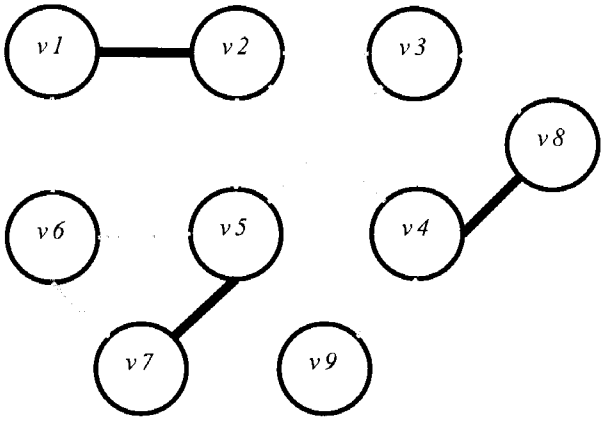

Figure 1.8: A Maximal Matching

A *maximum matching* $M \subseteq E(G)$ is a matching of a graph G such that no other matching has more edges than M. A maximal matching is not necessarily a maximum matching; however, every maximum matching is a maximal matching. Figure 1.9 shows a maximum matching for the graph used in Figure 1.8.

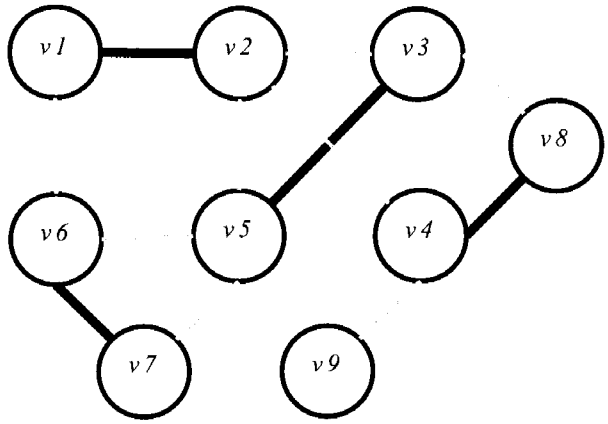

Figure 1.9: A Maximum Matching

A *perfect matching* $M \subseteq E(G)$ is a matching of a graph G such that every vertex in $V(G)$ is covered by the matching. Since each edge is incident to exactly two vertices, it follows that only graphs with an even number of vertices can have a perfect matching. A perfect matching is necessarily maximal since it is also a maximum matching. An example of a perfect matching is given in Figure 1.10.

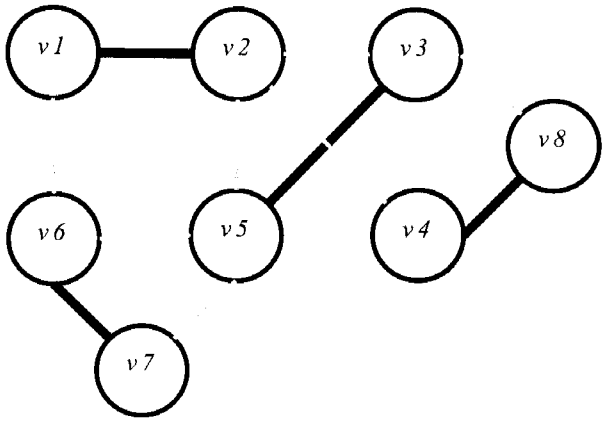

Figure 1.10: A Perfect Matching

Lawler describes in [14] an augmenting path algorithm to find a perfect matching (if it exists) in a bipartite graph. The algorithm given by Lawler is equivalent to finding a maximum flow in a combinatorial optimization problem. The algorithm is performed by introducing two new vertices, a *source* vertex which is adjacent to all vertices in one partite set and a *sink* vertex which is adjacent to

all vertices in the other partite set. Numerous paths are found one at a time from the source vertex to the sink vertex such that no edge is used in more than one path. The result is a maximal matching, found by looking at the edges between the partite sets. The matching is perfect if the number of edges in the matching is exactly one half the number of vertices in the graph. This algorithm can find a perfect mathing in $O(n^2)$ time for a bipartite graph on $2n$ vertices.

The Hopcroft-Karp algorithm, first introduced in [11], greatly improves the speed of the maximum flow algorithm, running in $O(m\sqrt{n})$ time where n is the number of vertices in the graph and m is the number of edges. In very dense graphs, however, this is actually worse, with a time bound near $O(n^{5/2})$ but for random graphs the algorithm is nearly linear. The main improvement to this algorithm comes by finding not just one path from the source to the sink but by finding a maximal set of shortest paths.

1.3 Groups

A *binary operation* is a function that takes exactly two inputs and gives exactly one output. The *associative property* states that the order in which a sequence of certain binary operations is performed does not change the final answer. That is, $a + (b + c) = (a + b) + c$. This property is also called *associativity.* An *identity* is an element such that when a binary operation is applied to the identity and any other element a the resulting answer is a. An *inverse* of an element a is another a^{-1} such that when a binary operation is applied to a and its inverse, the resulting answer is the identity.

Let G be a non-empty set of elements with a binary operation $+$ on those elements such that for any two elements $a, b \in G$ $a + b$ is also an element of G. G is a *group* if the following three properties hold:

1. The binary operation is associative such that for all $a, b, c \in G$, $a + (b + c) = (a + b) + c$.

2. There is an identity element 0 such that for $a \in G$, $a + 0 = a = 0 + a$.

3. Every element $a \in G$ has an inverse $-a \in G$ such that $a+(-a) = 0 = (-a)+a$.

The *order* of a group G, $|G|$ is the number of elements in the group.

A group of special interest in this dissertation is the group $\mathbb{Z}_n^n$. The elements of $\mathbb{Z}_n^n$ are n-tuples with each entry being an integer between 0 and $n - 1$. The binary operation is componentwise addition where addition is carried out modulo n for each component. The order of $\mathbb{Z}_n^n$ is n^n.

A concise way to illustrate a group and the binary operation is through a *Cayley table*. A Cayley table describes the structure of the group by labeling each row and each column with an element of the group so that every element in the group labels exactly one row and exactly one column. Such an arrangment is similar to an elementary multiplication or addition table. For two elements $a, b \in G$ with binary operation $+$ the entry in the ath row and bth column is the element of the group $a + b$. In general, the order of the elements is important. If the order of the elements in the binary operation is not important, that is, if $a + b = b + a$ then the group is *abelian* and the operation is *commutative*.

EXAMPLE 1.1. Using $\mathbb{Z}_2^2 = \{(0,0), (0,1), (1,0), (1,1)\}$ as an example of a group, some of the properties defined above are illustrated. In this group, $|\mathbb{Z}_2^2| = 2^2 = 4$. Let $+$ be the binary operation, then $(0,1) + (1,1) = (0 + 1, 1 + 1) = (1,2) = (1,0)$. The Cayley table for $\mathbb{Z}_2^2$ (Table 1.1) illustrates the group is abelian. The identity of this group is $(0,0)$ which the Cayley table verifies. Inverses of elements can be found by looking in a column of the Cayley table and finding the identity; the inverse is the label of the row in which the identity is found.

$+$	$(0,0)$	$(0,1)$	$(1,0)$	$(1,1)$
$(0,0)$	$(0,0)$	$(0,1)$	$(1,0)$	$(1,1)$
$(0,1)$	$(0,1)$	$(0,0)$	$(1,1)$	$(1,0)$
$(1,0)$	$(1,0)$	$(1,1)$	$(0,0)$	$(0,1)$
$(1,1)$	$(1,1)$	$(1,0)$	$(0,1)$	$(0,0)$

Table 1.1: The Cayley Table of $\mathbb{Z}_2^2$

A group *homomorphism* ϕ is a function from a group G to a group G', where it is possible $G = G'$ that satisfies one property. For $a, b \in G$

$$\phi_G(a + b) = \phi_G(a) + \phi_G(b)$$

That is, to say, ϕ preserves the operation of the group.

A homomorphism $\phi : G \to G'$ that takes every element of G to exactly one element of G' (*one-to-one*) such that every element of G' is mapped to (*onto*) is called a *isomorphism*. If $\phi : G \to G'$ is an isomorphism between G and G', then G is *isomorphic* to G'. If $G' = G$ then an isomorphism maps each element to another element in the same group and is called an *automorphism*. The set of automorphisms of a group G forms a group $\mathrm{Aut}(G)$, the automorphism group of G.

Let F be a non-empty set of elements with two binary operations $+$ and $\times$ on those elements such that for any two elements $a, b \in F$ $a + b$ and $a \times b$ are also elements of F. F is a *field* if the following three properties hold:

1. The $+$ operation makes F into an abelian group F^+ with identity 0.

2. The $\times$ operation makes $F/\{0\} = F^\times$ into an abelian group with identity 1.

3. There is a distributive law that says for all $a, b, c \in F$, $a \times (b+c) = a \times b + a \times c$.

PROPOSITION 1.3. $\mathbb{Z}_n$ is a field.

The proof of Proposition 1.3 can be found in [2] or any beginning algebra text.

For any group G, the *conjugacy class* of an element $g \in G$ is $cl(g) = \{xgx^{-1} | x \in G\}$. These subsets partition G. For an abelian group, since the elements commute, each element is its own conjugacy class.

CHAPTER 2
MATROIDS, CHARACTERS, AND FOURIER TRANSFORMS

This chapter is meant as an introduction to *matroids*, *characters*, *representations*, and *Fourier transforms*. These topics are defined as they relate to material that will be covered in later chapters, providing background material that can be referenced then. Matroid theory generalizes many of the ideas from graph theory and linear algebra. Characters and representations are extensions of group theory. Fourier transforms are discussed only enough to describe results in this dissertation. A very brief final section dealing with computational complexity, based on [10], ends the chapter.

Notations and definitions for matroids are based on Oxley [16] and Wilson [24]. Conventions used in Artin [2] and Serre [19] are used for representations and characters. The information concerning Fourier transforms uses the style established in Terras's text [20]. The section on the Combinatorial Nullstellensatz comes from the original paper by Alon [1].

2.1 Matroids

A matroid is a combinatorial structure that generalize the ideas of linear independence (see Section 1.1). Many of the ideas and terminology used when working with matroids come from linear algebra and graph theory. Whitney was the first to describe matroids in [23] and is credited with giving them the name matroid.

Because matroids generalize many of the ideas of linear algebra as well as graph theory, they provide a rich connection between the two subjects. Matroids have been studied extensively by Whitney [23], Oxley [16], Rota and Crapo [17], and various others. Many important properties of matroids have been discovered with one of the most notable being matroid intersection, defined later in this section. Matroid intersection provides a quick, efficient method for finding perfect matchings in bipartite graphs as described in this dissertation.

Let $\emptyset$ denote the *empty set*, the set with no elements. A *matroid* $M = (E, \mathcal{I})$ is a nonempty, finite set E and a collection subsets of E called $\mathcal{I}$ satisfying three properties:

1. $\emptyset \in \mathcal{I}$
2. If $I \in \mathcal{I}$ and $I' \subseteq I$ then $I' \in \mathcal{I}$
3. If I_1 and I_2 are in $\mathcal{I}$ and $|I_1| < |I_2|$, then there is an element e of $I_2 - I_1$ such that $\{I_1 \cup e\} \in \mathcal{I}$

The set E, occassionally written $M(E)$, is the *ground set* of M and the members of $\mathcal{I}$, occassionally written $\mathcal{I}(M)$, are the *independent sets* of M. A subset of E not in $\mathcal{I}$ is called *dependent.*

There are many equivalent ways to define matroids, often naming types of matroids based on the definition most convenient for the application. Two specific types of matroids are a *vector matroid* and a *partition matroid.*

PROPOSITION 2.1. Let E be the set of columns of an $m \times n$ matrix A over a field. Let $\mathcal{I}$ be the set of subsets S of E such that S is linearly independent (as described in Section 1.1). Then $(E, \mathcal{I})$ is a vector matroid.

Proof. To prove the proposition it is necessary to show it satisfies the properties of the definition of a matroid. The properties are shown to be satisified in the order

they are given in the definition. The empty set is independent by definition, thus property 1 is satisfied. A subset of an independent set is independent satisfying property 2. To show property 3, assume I_1 and I_2 are independent sets with $|I_1| < |I_2|$. Let W be the subspace spanned by $I_1 \cup I_2$. Then $|I_2| \leq \dim(W)$ since $I_2 \subseteq W$. Assume that for all $e \in I_2 - I_1$ that $I_1 \cup e$ is dependent. Then it must be the case that W is contained in the span of I_1 and so $|I_2| \leq \dim(W) \leq |I_1| < |I_2|$ or $|I_2| < |I_2|$, a contradiction. Therefore, there must be some $e \in I_2 - I_1$ such that $I_1 \cup e$ is indepedent. Property 3 is satisfied. □

PROPOSITION 2.2. Let E be a finite set that has been partitioned into m non-empty partitions. Let $\mathcal{I}$ be the set of subsets S of E such that no two elements in S are in the same partition of E. Then, $(E, \mathcal{I})$ is a partition matroid.

Proof. Proving the proposition is accomplished by showing the properties of a matroid given in the definition are satsified by going through the properties in order. The empty set clearly does not have two elements from the same partition of E, thus property 1 is satisfied. A subset of a set without two elements from the same partition of E does not have two elements from the same partitition, satisfying property 2. To show property 3, assume I_1 and I_2 are independent sets (that is, no two elements from the same partition of E) with $|I_1| < |I_2|$. Since there are more elements in I_2 than I_1 and no two elements are from the same partition in I_2 there must be at least one element e in I_2 that is from a partition not represented in I_1. Then $I_1 \cup e$ does not have two elements in from the same partition of E, thus is independent. Property 3 is satisfied. □

When two matroids M_1 and M_2 share on a common ground set E, their intersection, called a *matroid intersection*, can be defined. The set E is the ground set common to M_1 and M_2. The independent sets $\mathcal{I}$ are the subsets of E that are independent in both M_1 and M_2. That is, $\mathcal{I}(M) = \mathcal{I}(M_1) \cap \mathcal{I}(M_2)$. In general,

the result from a matroid intersection is not, itself, a matroid, often failing to satisfy property 3. The idea of matroid intersection is important in combinatorial optimization with one application being to find a maximum size of a matching in a bipartite graph.

2.2 Group Representations and Characters

This dissertation concerns representations of finite abelian groups isomorphic to $\mathbb{Z}_n^n$. Group representations theory studies properties of groups (as defined in Section 1.3) through their representations as linear transformations of vector spaces. This allows group theory problems to be reduced to problems in linear algebra. Linear algebra problems tend to be more tractable.

Frobenius initially developed representations and characters in [8]. Further work was done by Brauer in [4] and [3]. Group representations allow abstract groups to be described using vector spaces. Group elements can be represented as matrices allowing the group operation to be matrix multiplication. By changing the group operation to matrix mutliplication, studying groups is made easier because the operation is consistent over many groups. Characters of a representation provide much of the same information as the representation, but in a condensed form allowing for easier study of the structural properties of groups.

For finite groups, character values are always sums of roots of unity. This property allows an easy connection to using matroid intersection and the Combinatorial Nullstellensatz to find maximum size matchings in bipartite graphs, as explained later in this dissertation.

A *representation* of a group is a homomorphism from the group to the automorphism group of a vector space. Let $GL(V)$ be the group of isomorphisms of a vector space V onto itself. This group is isomorphic to the group of invertible $n \times n$

matrices where n is the dimension of V. A representation of a group G on a vector space V over a field K is a homomorphism from G to $GL(V)$ given by the map:

$$\rho : G \to GL(V)$$

$$\rho(g_1 g_2) = \rho(g_1)\rho(g_2)$$

for all $g_1, g_2 \in G$. In this case V is called the *representation space* and the dimension of V is the *dimension* of the representation. Often, the notation ρ_g is used for $\rho(g)$, meaning the representation at the element g.

Let ρ be a representation of a group G on a vector space V with W a subspace of V. If $gw \in W$ for all $w \in W$ and all $g \in G$ then W is *G-invariant*. That is, the operation ρ acting on V is restricted to an operation on W. If a representation ρ of a group G on a vector space V does not have a proper G-invariant subspace then ρ is *irreducible*.

PROPOSITION 2.3. Let G be a group of order n. There are the same number of irreducible representations as there are conjugacy classes in G. Furthermore if d_i is the dimension of the irreducible representation ρ_i and there are r such representations, $n = d_1^2 + d_2^2 + \ldots + d_r^2$.

The proof of this proposition can be found in [2] or [19].

PROPOSITION 2.4. If G is a finite abelian group of order n, then every irreducible representation of G is one-dimensional.

Proof. Since G is abelian every element is its own conjugacy class, thus the number of conjugacy classes is n. By Proposition 2.3 there are n irreducible representations. Since $n = d_1^2 + d_2^2 + \ldots + d_n^2$ and $d_i \neq 0$ it must be that $d_i = 1$ for all i. □

Let $\rho : G \to GL(V)$ be a representation of a finite group G over the vector space V. For each $g \in G$, let

$$\chi_\rho(g) = \mathrm{Tr}(\rho_g).$$

The function χ_ρ is called the *character* of the representation ρ. The name character emphasizes the fact that χ_ρ characterizes the representation ρ. A character is defined for a representation at an element.

If the representation is being taken of the group G then the set of irreducible characters of G is the set $\widehat{G}$, called the *dual*. For an abelian group, G is isomorphic to $\widehat{G}$ and $\widehat{G}$ is a group. In particular, the irreducible characters are isomorphic to the elements of the group.

EXAMPLE 2.1. Let $C_n = \{1, r^1, r^2, \ldots, r^{n-1}\}$ be the cyclic group of order n, where $r^n = 1$. This is an abelian group under multiplication of the elements. It is isomorphic to the group $\mathbb{Z}_n$ under the map $\phi : C_n \to \mathbb{Z}_n$ given by $\phi(r^n) = n$. According to Proposition 2.4 the irreducible representations C_n are all of degree 1, thus there are n irreducible representations. Let $\rho(r^k) = k$ be the representations and let $\omega = e^{2\pi i/n}$. The irreducible characters $\chi_0, \chi_1, \ldots, \chi_{n-1}$ are given by

$$\chi_h(r^k) = \omega^{hk} = e^{2\pi ihk/n}$$

where $\chi_i = \chi_{\rho(r^i)}$.

EXAMPLE 2.2. For $n = 4$ it is easier to represent the irreducible characters in a *character table* as shown in Table 2.1 where the columns are labeled by the elements and the rows are labeled by the characters at irreducible representations. The entry in the table is the character at the irreducible representation of the row evaluated at the element of the column.

EXAMPLE 2.3. The groups $\mathbb{Z}_n^n$ are used in the results in this dissertation. The character table for the group $\mathbb{Z}_2^2$ is given in Table 2.2.

Even though $|\mathbb{Z}_4| = |\mathbb{Z}_2^2|$, the groups are not isomorphic. This nonisomorphism becomes more clear by looking at the character tables; the tables differ.

	1	r	r^2	r^3
χ_0	1	1	1	1
χ_1	1	ω	ω^2	ω^3
χ_2	1	ω^2	1	ω^2
χ_3	1	ω^3	ω^2	ω

Table 2.1 – The Character Table of $\mathbb{Z}_4$

	1	r	r^2	r^3
χ_0	1	1	1	1
χ_1	1	1	ω	ω
χ_2	1	ω	1	ω
χ_3	1	ω	ω	1

Table 2.2 – The Character Table of $\mathbb{Z}_2^2$.

2.3 Fourier Transforms on Finite Abelian Groups

In a finite group G the Fourier transform uses the matrix entries of irreducible representations, as described in Section 2.2. That is, $\rho : G \to U(n)$ such that $\rho(gh) = \rho(g)\rho(h)$, where $U(n)$ is the group of unitary $n \times n$ matrices. Since the only groups of interest in this dissertation are abelian groups, the representations are one-dimensional thus $\rho : G \to \mathbb{C}$.

The vector space $L^2(G)$ for a finite group G is defined by $L^2(G) = \{f : G \to \mathbb{C}\}$ = the set of all complex-valued functions on G.

The *discrete Fourier transform* (DFT) of $f \in L^2(G)$ where G is abelian is

$$\mathcal{F}f(\chi_\rho) = \widehat{f}(\chi_\rho) = \sum_{g \in G} f(g)\overline{\chi_\rho(g)} = \langle f, \chi_\rho \rangle,$$

for $\chi_\rho \in \widehat{G}$.

The *Fourier matrix* of the discrete Fourier transform of order n, F_n is seen in Matrix 2.1, where $\xi^k = \omega^{-k} = e^{-2\pi i/n}$.

$$F_n = \frac{1}{\sqrt{n}} \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \xi^1 & \xi^2 & \dots & \xi^{n-1} \\ 1 & \xi^2 & \xi^4 & \dots & \xi^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \xi^{n-1} & \xi^{2(n-1)} & \dots & \xi^{(n-1)(n-1)} \end{bmatrix}$$

Matrix 2.1: F_n

F_n is symmetric and invertible. The inverse of F_n is its conjugate transpose, denoted F_n^* and is easily obtained from F_n by replacing ξ with ω.

The *Vandermonde matrix* $V(z) = V(z_0, z_1, \dots, z_{n-1})$ is defined as in Matrix 2.2.

$$V(z_0, z_1, \dots, z_{n-1}) = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ z_0 & z_1 & z_2 & \dots & z_{n-1} \\ z_0^2 & z_1^2 & z_2^2 & \dots & z_{n-1}^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ z_0^{n-1} & z_1^{n-1} & z_2^{n-1} & \dots & z_{n-1}^{n-1} \end{bmatrix}$$

Matrix 2.2: Vandermonde Matrix

The Fourier matrix can be written as $F_n = n^{-1/2} V(\xi^0, \xi^1, \dots, \xi^{n-1})$ by making the substitutions into the Vandermonde matrix as necessary.

The *Vandermonde identity* which gives rise to the Vandermonde matrix is

$$V(x) = V(x_0, x_1, \dots, x_{n-1}) = \prod_{0 \le i < j < n} (x_j - x_i) = \sum_{\pi \in S_n} \operatorname{sign}(\pi) \prod_{i=0}^{n-1} x_i^{\pi(i)}, \qquad (2.1)$$

where S_n is the set of permutations. See [20] for a proof of this identity.

2.4 Combinatorial Nullstellensatz

In [1] Alon proved the following two thereoms, which when taken together, form the *Combinatorial Nullstellensatz*.

THEOREM 2.1. Let F be an arbitrary field, and let $f = f(x_1, \ldots, x_n)$ be a polynomial in $F[x_1, \ldots, x_n]$. Let $S_1, \ldots, S_n$ be nonempty subsets of F and define $g_i(x_i) = \prod_{s \in S_i} (x_i - s)$. If f vanishes over all the common zeros of $g_1, \ldots, g_n$ (that is, if $f(s_1, \ldots, s_n) = 0$ for all $s_i \in S_i$), then there are polynomials $h_1, \ldots, h_n \in F[x_1, \ldots, x_n]$ satisfying $deg(h_i) \leq deg(f) - deg(g_i)$ so that

$$f = \sum_{i=1}^{n} h_i g_i.$$

Moreover, if $f, g_1, \ldots, g_n$ lie in $R[x_1, \ldots, x_n]$ for some subring R of F then there are polynomials $h_i \in R[x_1, \ldots, x_n]$ as above.

THEOREM 2.2. Let F be an arbitrary field, and let $f = f(x_1, \ldots, x_n)$ be a polynomial in $F[x_1, \ldots, x_n]$. Suppose the degree $deg(f)$ of f is $\sum_{i=1}^{n} t_i$, where each t_i is a nonnegative integer, and suppose the coefficient of $\prod_{i=1}^{n} x_i^{t_i}$ in f is nonzero. Then, if $S_1, \ldots, S_n$ are subsets of F with $|S_i| > t_i$, there are $s_1 \in S_1, \ldots s_n \in S_n$ so that

$$f(s_1, \ldots, s_n) \neq 0.$$

In short, these theorems imply that by constructing an appropriate non-zero polynomial the existence of a certain combinitorial structure can be tested by looking at the evaluation of the polynomial. The Combinatorial Nullstellensatz is related to the discrete Fourier transform over a finite group. Exploring this relationship is the bulk of Chapter 3.

The Combinatorial Nullstellensatz has been employed successfully in a variety of circumstances, but there is still no clear understanding of which circumstances are favorable to its application despite the many problems that are apparently prime

candidates. One of the main purposes of this dissertation is to show how it can be applied to the problem of detecting perfect matchings in bipartite graphs. Other problems for which it seems aptly suited include: the problem of showing that every tree has a ρ-valuation (see [12]), showing that every odd order Latin square has a Latin transversal (see [13]), and proving the existence of a Hamiltonian cycle in middle levels of the boolean lattice, just to name a few of the highly symmetric, famous and still open problems.

In each of these problems it is straightforward to construct polynomials that vanish completely on some appropriate domain if and only if the desired combinatorial object does not exist. The main source of our frustration is the realization that the polynomials in question are presented in compact, factored form; determining whether a nonzero coefficient appears in its expansion (modulo an appropriate ideal) is a formidable problem (in general, this problem is NP-hard, see Section 2.5). Most successful applications of the Nullstellensatz technique so far, when applied to problems with more than one instance of each size, have been to problems with the special property that all instances of a given size determine a collection of polynomials that have a common monomial with a nonzero coefficient; thus, proving the monomial is nonzero for one canonical instance shows it is nonzero for others. Many natural problem formulations do not share this property. The natural formulation of the ρ-valuations-for-trees problem, for example, does not have this property. Similarly, the natural formulation of the perfect-matching-in-a-bipartite graph problem also does not, as shown in this dissertation. Because this latter problem is easy from a complexity point of view, one would expect a polynomial-time algorithm to find a nonzero coefficient in the expansion of the corresponding encoding polynomial, if such a coefficient exists. The matroid-intersection algorithm suffices for this purpose.

We hope that further investigation will provide a sharpened form of the Com-

binatorial Nullstellensatz, perhaps incorporating elements of the matroid-intersection algorithm. It seems very likely a nice formulation along these lines awaits discovery. This dissertation demonstrates that such a formulation applies in the perfect-matching-in-a-bipartite graph problem. Formulating and solving this problem in the nullsellensatz fashion has shed some light on the relation between the number of perfect matchings and the number of maximum independent sets in the intersection of certain matroids via the uncertainty principle, as formulated through the Fourier transform on a finite group.

2.5 Computational Complexity

One of the fundamental questions in the theory of computational complexity is whether $P = NP$ or $P \neq NP$. In order to understand this statement it is necessary to know what P and NP mean. A *decision problem* is a problem for which the answer is either "yes" or "no," depending on the inputs. Traditional examples include asking whether a number n is prime.

A problem is classified as being in P, standing for "polynomial" time, if it is a decision problem that can be solved by a deterministic Turing machine in a polynomial amount of time. Problems belonging to P are often said to be "tractable" problems. A problem is classified as being in NP, standing for "nondeterministic polynomial" time, if it is a decision problem and a positive answer can be verified in polynomial time by a deterministic Turing machine. An alternate definition is a problem is classified as being NP if it is solvable in polynomial time by a non-deterministic Turing machine.

If a problem can be solved in polynomial time by a deterministic Turing machine then such a solution can be checked in polynomial time by a deterministic Turing machine. Thus, $P \subseteq NP$. However, NP contains additional problems,

especially problems classified as NP-*complete*. A problem is classified as being NP-complete if the problem is in NP, thus a solution can be verified in polynomial time, and the problem is equivalent to all other problems that are in NP. An important note is that just because a problem is classified as NP does not necessarily mean a polynomial time algorithm for finding the solution cannot exist; it simply says that no such algorithm has been discovered. Discovering one would solve the question of whether $P = NP$ in the affirmative.

Problems considered $\#P$, pronounced "sharp P" or "number P," are different sorts of problems. Problems in this complexity class are not longer decision problems. While P and NP problems ask whether something exists or not, or whether something is true or not, $\#P$ problems ask the question "how many?" A $\#P$ problem is as difficult as a corresponding NP problem; counting the number of positive solutions is at least as difficult as finding whether there is a positive solution. The complexity class $\#P$ was first introduced by Valiant in [21]. Similar to NP-complete, a problem is considered $\#P$-*complete* if the problem is in $\#P$ and the problem is equivalent to all other problems in $\#P$.

CHAPTER 3
PERFECT MATCHINGS IN BIPARTITE GRAPHS

Perfect matchings in bipartite graphs is a classic area of study in discrete mathematics. Hall's Marriage Theorem, one of the most widely recognized applications of perfect matchings in bipartite graphs, is a very common topic for courses and is included in [24], an undergraduate text on graph theory. Algorithms to find such a matching can be found in [14]. The Combinatorial Nullstellensatz is a method to detect the existence of combinatorial structures. As stated in Section 2.4 it has been applied in a variety of circumstances already. However, there are many problems for which the Combinatorial Nullstellensatz seems ideally suited but has not yet been applied. Finding perfect matchings in bipartite graphs seems ideal for applying the method.

There are other known bounds for the number of perfect matchings in certain classes of bipartite graphs, such as the bound proven by Voorhoeve in [22] for cubic bipartite graphs which was improved by Schrijver in [18] for k-regular bipartite graphs on $2n$ vertices. The bound in this dissertation is different because it works for all bipartite graphs that have at least one perfect matching. However, finding a bound is not the main focus of this dissertation. In proving the main result of this section it shows how the Combinatorial Nullstellensatz and the discrete Fourier transform can be used together to solve problems for which the Combinatorial Nullstellesatz seems applicable, such as bounding the number of perfect matchings in bipartite graphs. The hope is that similar methods will work for other problems as presented in the last chapter.

Consider an $n \times n$ *circular lock* consisting of n equal-sized wheels placed one on top of the other. Each wheel has n cells of same size, filled with a complex number. Each wheel rotates independently, both clockwise and counter-clockwise, but only in discrete intervals corresponding to the cell sizes. After rotations are complete, cells align forming columns. A *setting* of the lock is such a rotation of the wheels. Because a setting of the lock means that the cells align, each setting determines (up to a rotation of all wheels by the same number of cells) an $n \times n$ matrix whose i, jth entry is simply the entry of the jth cell of wheel i (Figure 3.1).

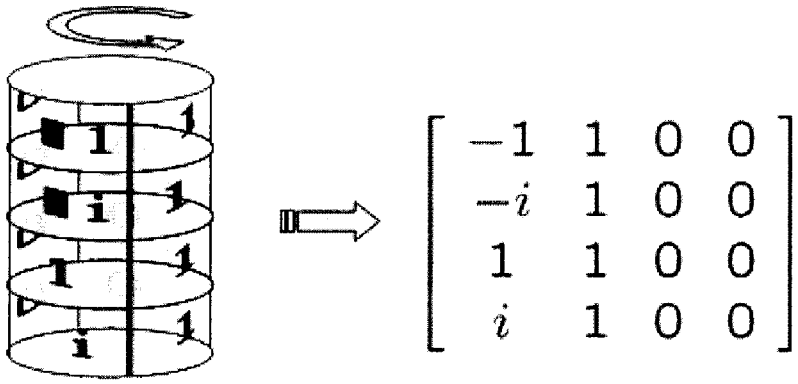

Figure 3.1: A Setting on a Circular Lock with the Corresponding Matrix

A circular lock is unlocked or *opens* if its wheels are placed into a setting in which the corresponding matrix has nonzero determinant; otherwise, the lock remains *closed.* In this chapter, we explore what settings open a lock, if any at all do, by considering a polynomial designed to detect perfect matchings in bipartite graphs and its Fourier transform which detects maximum cardinality independent sets in the intersection of two specific types of matroids. Only circular locks arising from bipartite graphs, whose rows are coefficients of polynomials with zeros that are all nth roots of unity are considered.

3.1 Circular Locks from Bipartite Graphs

This section will provide some background to see how the circular lock idea corresponds to finding perfect matchings in bipartite graphs. The Combinatorial Nullstellensatz, as it applies, is described first, and then two results are proven. The first result shows that the polynomial derived from the graph shows what perfect matchings exist in the graph, and the second (and main) result shows that the settings which open the circular lock correspond to the perfect matchings in the graph that gives rise to the lock. Each result is followed by examples.

Let the symbol $\mathbb{C}$ denote the field of complex numbers and $\omega = e^{2\pi i/n}$ where $i = \sqrt{-1}$. For a positive integer n, let $\Omega_n = \{\omega^0, \ldots, \omega^{n-1}\}$ be the set of nth roots of unity.

Let G be a bipartite graph with vertex set $A \cup B$, where $A = \{0, 1, \ldots, n-1\}$ and $B = \Omega_n$, and edge set $E \subseteq \{\{a, b\} : a \in A, b \in B\}$. From Section 1.2, a perfect matching is a matching in which every vertex of the graph is contained in an edge of the matching. The existence of a perfect matching has been well studied and there are many results. Hall's Theorem, as seen in [24] and others, is an example of a classical characterization of whether a bipartite graph has a perfect matching.

To use the Combinatorial Nullstellensatz it is necessary to have an appropriate polynomial. For $i = 0, 1, \ldots, n-1$, introduce a variable x_i. The variable x_i is used to create a polynomial

$$g_i(x_i) = \begin{cases} 1 & \text{if } x_i \text{ is adjacent to all vertices in } B \\ \prod_{\{i,\omega^j\} \notin E} (x_i - \omega^j) & \text{otherwise.} \end{cases}$$

For notational purposes, $g_i(x_i) = 1$ if x_i is adjacent to all vertices in B. The polynomial

$$g_i(x_i) = \prod_{\{i,\omega^j\} \notin E} (x_i - \omega^j) \tag{3.1}$$

is used throughout the dissertation. Also, as a notational convention, for any $\alpha \in \mathbb{Z}_n^n$, x^α means $\prod_{i=0}^{n-1} x_i^{\alpha_i}$.

The Vandermonde polynomial $V(x)$, along with the g_i's of Equation (3.1), is used to construct the appropriate polynomial.

PROPOSITION 3.1. There exists a perfect matching in G if and only if the polynomial

$$f_G(x) = f_G(x_0, x_1, \ldots, x_{n-1}) = V(x) \prod_{i=0}^{n-1} g_i(x_i) \tag{3.2}$$

is nonzero for some input from Ω_n^n.

Proof. Assume $\alpha = (\alpha_0, \alpha_1, \ldots, \alpha_{n-1}) \in \Omega_n^n$ describes a perfect matching; that is, $i \in A$ is adjacent to vertex $\alpha_i \in B = \Omega_n$. Because α is a perfect matching, the α_i's are distinct. We must show that f_G is nonzero at α. It is important to note that f_G is a product of monomials and that the product of two or more nonzero elements is nonzero.

Because of the the Vandermonde identity $V(\alpha_0, \alpha_1, \ldots \alpha_{n-1}) = \prod_{0 \le i < j < n} (\alpha_j - \alpha_i)$ and the fact that the α_i's are distinct and nonzero, this factor is nonzero. Consider the g_i factors. If vertex $i \in A$ is adjacent to all vertices in B, then $g_i = 1$ and is nonzero. If vertex i is not adjacent to some vertex in B then g_i is as described in Equation (3.1). By assumption, α is a perfect matching, thus there exists an edge from vertex i to vertex $\alpha_i = \omega^k$, for some k. Thus the term $(x_i - \omega^k)$ does not appear in g_i. When substituting α_i for x_i in g_i the terms have the form $(\omega^k - \omega^j)$ with $k \neq j$. These values are nonzero. Thus each g_i is nonzero. Therefore, f_G is a product of nonzero terms when evaluated at a perfect matching. Thus if α is a perfect matching, then $f(\alpha) \neq 0$.

Assume $f_G(\beta) \neq 0$ for some $\beta = (\beta_0, \beta_1, \ldots \beta_{n-1})$. As before, it is important to note that f_G is a product of factors and that if any one factor is zero, then the whole evaluation is zero. Because $f_G(\beta) \neq 0$, it must be the case that no factor in

f_G is zero. In the Vandermonde factor $V(\beta)$, if no term is zero, then the difference between any pair β_i and β_j must be nonzero, so $\beta_i \neq \beta_j$, for $i \neq j$. For each $g_i(\beta)$ to be nonzero means that each term is nonzero. The only way to have a zero term in a g_i is to attempt to evaluate at an edge that does not exist in the neighborhood of vertex i. Therefore, there must be an edge from vertex i to vertex $\beta_i = \omega^k$ for some k. Thus, β describes a set of distinct vertices in B in which every vertex in A is adjacent to exactly one vertex B; β is a perfect matching when $f(\beta) \neq 0$. □

Each polynomial g_i can be expanded into a sums of powers of x_i as

$$g_i(x_i) = \prod_{\{i,\omega^j\} \notin E} (x_i - \omega^j) = \sum_{j=0}^{n-1} l_{ij} x_i^j.$$

Let $L_G = [l_{ij}]$ be the $n \times n$ matrix of the coefficients of the g_i's; that is, L_G is the circular lock derived from the graph G. Likewise, G_L is the bipartite graph derived from a circular lock L. For any $\alpha \in \mathbb{Z}_n^n$, $L[\alpha]$ is the matrix obtained from L by rotating row i to the left α_i units. Rotating the lock gives many different matrices, all describing the same graph. Therefore, the *canonical* circular lock is $L_G = [l_{ij}]$ without any rotations.

EXAMPLE 3.1. Let G be the bipartite graph given in Figure 3.2. The reduced adjacency matrix of G, $\widehat{A_G}$, is Matrix 3.1. The polynomial f_G is computed as

$$\begin{aligned}
g_0(x_0) &= (x_0 - \omega^2) = -\omega^2 + 1x_0^1 + 0x_0^2 \\
g_1(x_1) &= (x_1 - \omega^0) = -1x_1^0 + 1x_1^1 + 0x_1^2 \\
g_2(x_2) &= 1 = 1x_2^0 + 0x_2^1 + 0x_2^2 \\
V(x_0, x_1, x_2) &= (x_1 - x_0)(x_2 - x_0)(x_2 - x_1) \\
f_G(x) &= (x_1 - x_0)(x_2 - x_0)(x_2 - x_1)(x_0 - \omega^2)(x_1 - \omega^0)
\end{aligned}$$

This gives a lock matrix L_G as in Matrix 3.2. A rotation of $[1, 2, 2]$ is Matrix 3.3. The graph G has 3 perfect matchings given by

$$\{(\omega^0, \omega^1, \omega^2), (\omega^0, \omega^2, \omega^1), (\omega^1, \omega^2, \omega^0)\}.$$

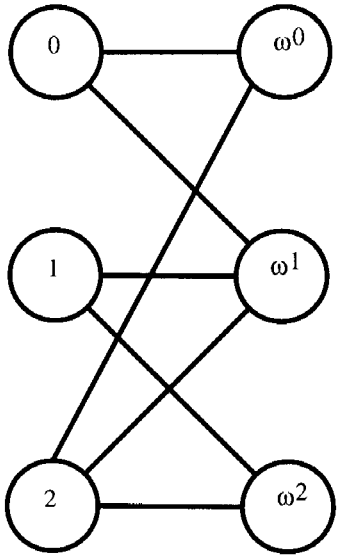

Figure 3.2: The Bipartite Graph for Example 3.1

$$\widehat{A_G} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \end{array} \begin{array}{c} \begin{array}{ccc} \omega^0 & \omega^1 & \omega^2 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \end{array}$$

Matrix 3.1: The Reduced Adjacency Matrix A_G for Example 3.1

$$L_G = \begin{bmatrix} -\omega^2 & 1 & 0 \\ -1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

Matrix 3.2: The Coefficient Matrix L_G for Example 3.1

Recall from Proposition 3.1 that there is a perfect matching in a graph G if and only if f_G is non-zero for some input. Finding the inputs for which f_G is nonzero is equivalent to finding the inputs for which it is zero. Let $\mathcal{I}_n$ be the ideal in $\mathbb{C}[x_0, x_1, \ldots, x_{n-1}]$, the polynomials that vanish on all inputs from the nth roots of unity; that is, $f \in \mathcal{I}_n$ if and only if $f(\alpha) = 0$ for all $\alpha \in \Omega_n^n$. Kézdy and Snevily

$$L_G[1,2,1] = \begin{bmatrix} 1 & 0 & -\omega^2 \\ 0 & -1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Matrix 3.3: The Rotated Coefficient Matrix $L_G[1,2,1]$ for Example 3.1

in [13] showed that $\mathcal{I} = \langle x_i^n - 1 \rangle_{i=0}^{n-1}$ and so $f(\alpha) = 0$ over all roots of unity if and only if $f(\alpha) \in \langle x_i^n - 1 \rangle_{i=0}^{n-1}$.

THEOREM 3.1. A circular lock L_G opens if and only if G_L has a perfect matching.

Proof. Recall that there exists a perfect matching in $G = G_L$ if and only if the polynomial

$$f_G(x) = V(x) \prod_{i=0}^{n-1} g_i(x_i) \tag{3.3}$$

is nonzero for some input from Ω_n^n. Now consider the polynomial

$$f(x) = f_G(x) \text{ modulo } \mathcal{I}_n.$$

We first prove that

$$f(x) = \sum_{\alpha \in \mathbb{Z}_n^n} \det(L[\alpha]) x^\alpha \tag{3.4}$$

To prove Equation (3.4), it suffices to prove that, for all $\alpha \in \mathbb{Z}_n^n$, the constant coefficient of $x^{-\alpha} f(x)$ modulo $\mathcal{I}_n$ is $\det(L[\alpha])$ (all exponents are reduced modulo n). Now the computation

$$x^{-\alpha} f(x) \equiv V(x) \left(\prod_{i=0}^{n-1} x_i^{-\alpha_i} g_i(x_i) \right) \text{ modulo } \mathcal{I}_n \tag{3.5}$$

shows that multiplying $f(x)$ by $x^{-\alpha}$ has the effect (modulo $\mathcal{I}_n$) of shifting, for all i, the coefficients of each $g_i(x_i)$ to the left by α_i units in the matrix L. To obtain the constant coefficient of $x^{-\alpha} f(x)$, observe that the Vandermonde polynomial expands

into $\sum_{\pi \in S_n} \text{sign}(\pi) \prod_{i=0}^{n-1} x_i^{\pi(i)}$ so, in order to obtain a constant coefficient, a monomial must be chosen from each of the factors $x_i^{-\alpha_i} g_i(x_i)$ that appear in Equation (3.5) in such a way that no two monomials have the same exponent; that is, we must select a transversal in the matrix $L[\alpha]$. This, along with the weightings of permutations by signs that appears in the expansion of the Vandermonde polynomial, means that the constant coefficient of $x^{-\alpha} f(x)$ modulo $\mathcal{I}_n$ is $\det(L[\alpha])$.

Because $f(x)$ has the form of Equation (3.3), it is clear that $f(\alpha) \neq 0$, for some $\alpha \in \Omega_n^n$, if and only if G_L has a perfect matching. On the other hand, form 3.4 of $f(x)$ shows that $f \not\equiv 0$ modulo $\mathcal{I}_n$, if and only if $\det(L[\alpha]) \neq 0$, for some $\alpha \in \mathbb{Z}_n^n$; the theorem follows. □

EXAMPLE 3.2. The graph from Example 3.1 has a perfect matching. Thus, L_G from that example has a setting that opens the lock. In fact, the rotation $L[1, 2, 1]$ as shown in Matrix 3.3 is one such setting. There are other settings which open the lock as well.

3.2 Matroid Intersection

The previous section showed that finding a setting to unlock the lock corresponds to finding a perfect matching in the graph. This section shows that a setting that opens the lock can be found efficiently by using matroid intersection.

Assume a graph has a perfect matching, which can be checked efficiently using the Hopcroft-Karp algorithm (in [11]), for example. By Theorem 3.1 a lock L_G can be unlocked; however, the theorem does not give an efficient method to find a setting that opens the lock. Knowing that a setting exists to open the lock is nice but being able to find such a setting efficiently is of greater value. Simply knowing that one exists without being able to find it does not tell you how to open the lock. The Nullstellensatz method relies on finding a non-zero coefficient

which corresponds to the setting of the lock. In Section 2.1 two matroids were discussed, the vector matroid and the partition matroid. The intersection of these two matroids forms the focus of the proof of the next theorem.

THEOREM 3.2. Let $G(A \cup B; E)$ be a bipartite graph and f_G the corresponding polynomial as in Equation (3.2). The settings that open the circular lock L_G correspond to the nonzero coefficients in the polynomial f_G. Furthermore, such a setting, if it exists, can be found in polynomial time via matroid intersection.

Proof. Consider a circular lock $L = L_G$, and its corresponding polynomial $f_G(x)$ as in Equation (3.3). As in the proof of Theorem 3.1 it suffices to show it is true for $f(x)$ modulo an appropriate ideal as in Equation (3.4). The coefficients of $f(x)$ correspond to determinants arising from rotations of the rows of L. Let $r_1, \ldots, r_n$ be the row vectors of L. Define E_j as the set of the n vectors obtained from r_j by cyclically permuting coordinates. Now define two matroids, M_1 and M_2 on the common ground set $E = \cup_{j=1}^{n} E_j$. The matroid M_1 is the vector matroid on E in which a set of elements is independent if and only if they are linearly independent (over $\mathbb{C}$). The matroid M_2 is the partition matroid on E in which a set of elements $S \subseteq E$ is independent if and only if $|S \cap E_j| \leq 1$, for all $j = 1, \ldots, n$. Now there is some $\alpha \in \mathbb{Z}_n^n$ such that $\det(L[\alpha]) \neq 0$ if and only if there exists a common independent set in M_1 and M_2 with cardinality n. Detecting the existence of such an independent set (and constructing such a set, if it exists) can be accomplished in polynomial time by Edmond's Matroid Intersection Algorithm, see [7]. □

This theorem shows the connection between the coefficients of $f(x)$ that correspond to the cardinality of n sets in the intersection of two matroids and the valuations of $f(x)$ over Ω_n^n that correspond to perfect matchings in G_L.

3.3 Fourier Transforms on Bipartite Graphs

A few techinical calculations need to be carried out to further the goal of this chapter. Most of these calculations involve the discrete Fourier transform and thus this section will manipulate the transform and provide a result necessary to bound the number of perfect matchings. Let $p(z) = \sum_{j=0}^{n-1} a_j z^j$ be a polynomial of degree at most $n-1$. Linear algebra shows that $p(z)$ is determined by its value at n distinct values, as we now review. If these n values are the roots of unity, then

$$n^{1/2} F_n^* \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix} = \begin{bmatrix} p(\omega^0) \\ p(\omega^1) \\ \vdots \\ p(\omega^{n-1}) \end{bmatrix}. \tag{3.6}$$

Recall that $F_n \times F_n^* = 1$. Multiplying both sides of Equation (3.6) by $n^{-1/2} F_n$ on the left transforms it into

$$\begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix} = n^{-1/2} F_n \begin{bmatrix} p(\omega^0) \\ p(\omega^1) \\ \vdots \\ p(\omega^{n-1} \end{bmatrix} \tag{3.7}$$

Let $G(A \cup B; E)$ be a bipartite graph where $A = \{0, 1, \ldots, n-1\}$ and $B = \Omega_n$. Recall Equation (3.2)

$$f_G(x_0, x_1, \ldots, x_{n-1}) = V(x) \prod_{i=0}^{n-1} g_i(x_i).$$

Using the Vandermonde identity, f_G can be written as

$$f_G(x_0, x_1, \ldots, x_{n-1}) = \prod_{0 \le i < j < n} (x_j - x_i) \prod_{i=0}^{n-1} g_i(x_i).$$

Recall from Section 2.3 that the Fourier transform of $f_G : \Omega_n^n \to \mathbb{C}$ is

$$\widehat{f}(\chi_\rho) = \sum_{\alpha \in \Omega_n^n} f(\alpha) \overline{\chi_\rho(\alpha)},$$

where

$$\overline{\chi_\rho(\alpha)} = \prod_{i=0}^{n-1} \alpha^{-r(i)}$$

and χ_ρ is an element of the dual $\widehat{\Omega_n^n}$, which is isomorphic to Ω_n^n as Ω_n^n is abelian. Let $\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_{n-1}\}$ and $\chi_\rho = \{\chi_1, \chi_2, \ldots, \chi_n\}$ with $\chi_i = \omega^{r(i)}$.

Consider the expansion of the Fourier transform:

$$\begin{aligned}
\hat{f}(\chi) &= \sum_{\alpha \in \Omega_n^n} f(\alpha)\overline{\chi(\alpha)} \\
&= \sum_{\alpha \in \Omega_n^n} \det(V(\alpha_1, \ldots, \alpha_n)) \left(\prod_{i=0}^{n-1} g_i(\alpha_i) \right) \left(\prod_{i=0}^{n-1} \alpha_i^{-r(i)} \right) \\
&= \sum_{\pi \in \mathcal{S}_n} \det(V(\omega^{\pi_1}, \ldots, \omega^{\pi_n})) \left(\prod_{i=0}^{n-1} (\omega^{\pi_i})^{-r(i)} g_i(\omega^{\pi_i}) \right) \\
&= \sum_{\pi \in \mathcal{S}_n} \det(V(\xi^0, \ldots, \xi^{n-1}) \cdot P_\pi) \left(\prod_{i=0}^{n-1} (\omega^{\pi_i})^{-r(i)} g_i(\omega^{\pi_i}) \right) \\
&= \sum_{\pi \in \mathcal{S}_n} \det(n^{1/2} F) \det(P_\pi) \left(\prod_{i=0}^{n-1} (\omega^{\pi_i})^{-r(i)} g_i(\omega^{\pi_i}) \right) \\
&= n^{n/2} \det(F) \sum_{\pi \in \mathcal{S}_n} \det(P_\pi) \left(\prod_{i=0}^{n-1} (\omega^{\pi_i})^{-r(i)} g_i(\omega^{\pi_i}) \right).
\end{aligned}$$

Because the remaining sum is a determinant,

$$\sum_{\pi \in \mathcal{S}_n} \det(P_\pi) \left(\prod_{i=1}^{n} (\omega^{\pi_i})^{-r(i)} g_i(\omega^{\pi_i}) \right)$$

$$= \left| \det \begin{pmatrix} (\omega^0)^{-r(0)} g_0(\omega^0) & \cdots & (\omega^0)^{-r(n-1)} g_{n-1}(\omega^0) \\ (\omega^1)^{-r(0)} g_0(\omega^1) & \cdots & (\omega^1)^{-r(n-1)} g_{n-1}(\omega^1) \\ (\omega^2)^{-r(0)} g_0(\omega^2) & \cdots & (\omega^2)^{-r(n-1)} g_{n-1}(\omega^2) \\ \vdots & \cdots & \vdots \\ (\omega^{n-1})^{-r(0)} g_0(\omega^{n-1}) & \cdots & (\omega^{n-1})^{-r(n-1)} g_{n-1}(\omega^{n-1}) \end{pmatrix} \right|,$$

and because $\det(AB) = \det(A)\det(B)$, in particular, F can be factored out,

$$\widehat{f}(\chi) = n^{n/2} \left| \det \left(F \cdot \begin{pmatrix} (\omega^0)^{-r(0)} g_0(\omega^0) & \cdots & (\omega^0)^{-r(n-1)} g_{n-1}(\omega^0) \\ (\omega^1)^{-r(0)} g_0(\omega^1) & \cdots & (\omega^1)^{-r(n-1)} g_{n-1}(\omega^1) \\ (\omega^2)^{-r(0)} g_0(\omega^2) & \cdots & (\omega^2)^{-r(n-1)} g_{n-1}(\omega^2) \\ \vdots & \cdots & \vdots \\ (\omega^{n-1})^{-r(0)} g_0(\omega^{n-1}) & \cdots & (\omega^{n-1})^{-r(n-1)} g_{n-1}(\omega^{n-1}) \end{pmatrix} \right) \right|.$$

From (3.7) the main observation follows:

$$\widehat{f}(\chi) = n^n \left| \det(L[r(i) + 1]) \right| \tag{3.8}$$

where L is the $n \times n$ matrix defined in Section 3.1 and $L[\beta]$ is L with the rows rotated by β.

This result answers the question of how many settings open the lock. The values for which the Fourier transform of f is nonzero, the *support* of $\widehat{f}$, correspond to the settings that open the lock L; that is, finding one is equivalent to finding the other. This equivalence is necessary to find a bound on the number of perfect matchings in a bipartite graph using the uncertainity principle, which is the focus of the next section.

3.4 Bounds from the Uncertainity Principle

In this section a bound for the number of perfect matchings is found. The bound found in this method is a good bound in the sense that the bound is acheived for certain classes of bipartite graphs such as the complete bipartite graphs. Bounding the number of perfect matchings in a bipartite graph also has other uses. Note that computing the number of perfect matchings in a bipartite graph is a $\#P$-complete problem(see Section 2.5), as shown by Valiant in [21], and is equivalent to a permanent computation.

The *permanent* of a matrix $A = [a_{i,j}]$ is defined as

$$\text{perm}(A) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} a_{i,\sigma(i)}.$$

Therefore, finding a bound on the number of perfect matchings provides a bound on the permanent. A bound on the number of perfect matchings (and thus the permanent of the matrix) can be found using the uncertainity principle.

The uncertainity principle states that a nonzero polynomial and its transform cannot both be highly concentrated. In the case of Fourier analysis over a finite group, Donoho and Stark in [6] proved that for G a finite, abelian group and f a nonzero polynomial $f : G \to \mathbb{C}$

$$\text{supp}(f)\text{supp}(\widehat{f}) \geq |G| \tag{3.9}$$

where $\text{supp}(f)$ is the support of the polynomial f, or the values for which f has a nonzero evaluation. An example of this principle ends this section. In the previous section the equivalence between the support of the transform of f and the settings that open the lock L was established. The Donoho-Stark version of the uncertainity principle can be applied as in the following theorem:

THEOREM 3.3. Let L_G be a circular lock derived from a bipartite graph G. If G has at least one perfect matching, then the product of the number of perfect matchings in G times the number of rotations that open the lock L_G is at least n^n.

Proof. Let f_G be the matching polynomial of the graph G and $\widehat{f_G}$ be the transform of f_G. From Proposition 3.1

$$\text{supp}(f_G) = \text{number of perfect matchings of } G,$$

while Equation (3.8) shows that

$$\text{supp}(\widehat{f_G}) = \text{number of rotations that open } L.$$

By Equation (3.9) it follows that the number of perfect matchings of G times the number of rotations that open L is greater than or equal to $|\Omega_n^n| = n^n$ and the result follows. □

Let $m(G) = \text{supp}(f_G)$, the number of perfect matchings in the bipartite graph G, and $r(G) = \text{supp}(\widehat{f_G})$, the number of rotations that open the lock L_G. The previous theorem states that $m(G)r(G) \geq n^n$ as long as $m(G) \neq 0$.

EXAMPLE 3.3. Let G be a bipartite graph such that the edge set of G is a perfect matching, then $m(G) = 1$ and $r(G) = n^n$. Theorem 3.3 holds since $m(G)r(G) = 1 \times n^n \geq n^n = |\Omega_n^n|$.

EXAMPLE 3.4. Let G be the graph given in Figure 3.2. In Example 3.1 it was shown that $m(G) = 3$. It can be verified that the rotations that unlock the lock are

$$\{(0,0,2),(0,1,0),(0,1,1),(0,1,2),(0,2,0),(0,2,1),(0,2,2),$$
$$(1,1,0),(1,2,1),(1,2,2),(1,2,0),(1,0,1),(1,0,2),(1,0,0),$$
$$(2,2,1),(2,0,2),(2,0,0),(2,0,1),(2,1,2),(2,1,0),(2,1,1)\}$$

so that $r(G) = 21$. In this case $m(G)r(G) = 3 \times 21 = 63 \geq 27 = 3^3 = |\Omega_3^3|$ and the theorem holds.

The bound found in Theorem 3.3 is a sharp bound for some cases, as proven in the next theorem.

THEOREM 3.4. For a bipartite graph $G(A \cup B, E)$ on $2n$ vertices with edge set exactly a perfect matching, the bound from Theorem 3.3 is sharp.

Proof. In this case the number of perfect matchings, $m(G) = 1$. Thus it must be shown that the support of the transform, the number of rotations for which the determinant is nonzero is n^n. Since the $g_i(x_i)$'s are determined by the edges that do not exist and each vertex in the partite set A is adjacent to one, distinct vertex in the partite set B each $g_i(x_i)$ is distinct. Thus every row in the coefficient matrix is

unique. This allows for any rotation to give a matrix with a nonzero determinant. There are $|\mathbb{Z}_n^n| = n^n$ possible rotations.

Thus $m(G) \times r(G) = 1 \times n^n = n^n$ for bipartite graphs whose edge set is a perfect matching. □

THEOREM 3.5. For a complete bipartite graph $G(A \cup B, E)$ on $2n$ vertices there are $n!$ perfect matchings and $n!$ rotations that give a nonzero determinant for the coefficient matrix.

Proof. Showing the number of perfect matchings is $n!$ is fairly intuitive. The first vertex in the partite set A can be matched with any of the n vertices in partite set B, the second vertex can be matched with any of the $n-1$ vertices left and so on. This gives $n!$ perfect matchings in the graph.

For a complete bipartite graph each of the $g_i(x_i) = 1$ for all $i \in \{0 \ldots n-1\}$. Thus each row in the coefficient matrix has exactly one 1 and the rest of the entries are zero. In order for the matrix to have a nonzero determinant no row can be a multiple of another row. In this case, this is equivalent to saying that each column must have exactly one 1. For the first row, the 1 can be placed in any of the n columns. For the second row the 1 can be placed in any of the $n-1$ columns, avoiding the column of the 1 in the first row and so on for each of the rows. This gives $n!$ matrices, all of which can be found by a rotation of the rows of the canonical matrix. □

Thus, the bound found in Theorem 3.3 is sharp for the bipartite graph with a minimum number of edges for which the theorem applies. However, in the bipartite graph with the maximum number of edges the bound is not sharp, since when $n > 2$, $n^n < (n!)^2$. The case of a complete bipartite graph is the worst case, in regards to the bound.

CHAPTER 4
PROPERTIES OF THE MATCHING POLYNOMIAL AND TRANSFORM

Our matching polynomials and their transforms contain much information about the graphs on which they are based. There are some interesting combinatorial and algebraic properties and interpretations that relate back to the graphs. As was seen in the previous section, our matching polynomials allow the perfect matchings in a graph to be found; however, as this section will show, perfect matchings play a more important role for our matching polynomials and their transforms. In fact, two of the main theorems in this section show that the perfect matchings form a basis for our matching polynomials and their transforms.

The coefficients of our polynomials also reveal some information about the functions. The last section in this chapter deals with the coefficients and how to find them. The result presented in that section mirrors a result given in Equation (3.8).

4.1 A Basis for Our Matching Polynomial

One interesting property of the matching polynomials is that for any bipartite graph on $2n$ vertices, the matching polynomial is a weighted average of the perfect matchings in the graph. The main theorem in this section will make this statement precise. This result is interesting because it shows that the perfect matchings can be used to build our matching polynomials. It also provides some algebraic insight as to why our matching polynomials evaluate to a nonzero result only on perfect

matchings.

Before the main result of this section it is important to present some notational conventions. The polynomial f_G for bipartite graph $G(A \cup B; E)$ with $A = \{0, 1, \ldots, n-1\}$ and $B = \Omega_n$ takes the form

$$f_G = \sum_{\alpha \in \Omega_n^n} d_\alpha x_0^{\alpha_0} x_1^{\alpha_1} \ldots x_{n-1}^{\alpha_{n-1}}$$

when expanded where $\alpha = (\alpha_0, \alpha_1, \ldots, \alpha_{n-1})$. The coefficients of these polynomials give rise to coefficient vectors in which the entries of the vectors are in a lexicographic monomial ordering. The next example shows this ordering.

EXAMPLE 4.1. Let $f(x) = 1x_0 + 2x_1 + 3x_0x_1 + 4 = 1x_0^1x_1^0 + 2x_0^0x_1^1 + 3x_0^1x_1^1 + 4x_0^0x_1^0$. To find the lexicographic monomial ordering look at the exponents on the terms. These can be thought of as the numbers $10, 01, 11$, and 00. The lowest of these numbers is 00 and thus the term that is first in a lexicographic monomial ordering is $4x_0^0x_1^0$. Thus corresponding coefficient vector would be $\langle 4, 2, 1, 3 \rangle$.

EXAMPLE 4.2. Consider the graphs given in Figure 4.1

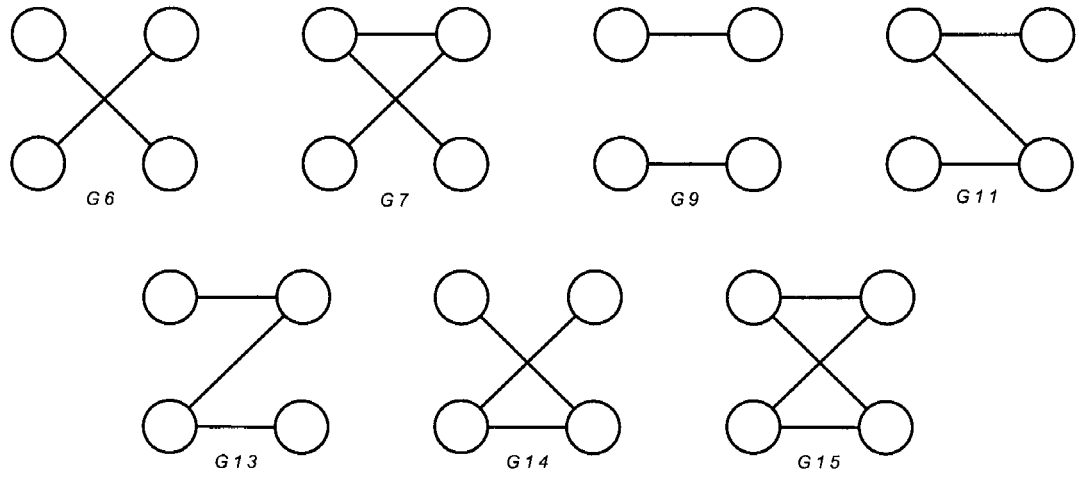

Figure 4.1: All Bipartite Graphs on 4 Vertices with at least 1 Perfect Matching

and their reduced adjacency matrices:

$$G_6 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \; G_7 = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}, \; G_9 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \; G_{11} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix},$$

$$G_{13} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \; G_{14} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}, \; G_{15} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}.$$

These graphs are all of the bipartite graphs on 4 vertices with at least one perfect matching. From Equation (3.2), these graphs have polynomials

$$\begin{aligned} f_6(x) &= 2 + 2x_1 - 2x_0 - 2x_0x_1 \\ f_7(x) &= -1 + -1x_1 + 1x_0 + 1x_0x_1 \\ f_9(x) &= -2 + 2x_1 - 2x_0 + 2x_0x_1 \\ f_{11}(x) &= -1 + 1x_1 - 1x_0 + 1x_0x_1 \\ f_{13}(x) &= 1 - 1x_1 + 1x_0 - 1x_0x_1 \\ f_{14}(x) &= 1 + 1x_1 - 1x_0 - 1x_0x_1 \\ f_{15}(x) &= 0 - 1x_1 + 1x_0 + 0x_0x_1 \end{aligned}$$

where $f_i = f_{G_i}$. The terms of the polynomials are in lexicographic monomial order. The coefficient vectors of these polynomials are

$$\overrightarrow{v_6} = \langle 2, 2, -2, -2 \rangle, \; \overrightarrow{v_7} = \langle -1, -1, 1, 1 \rangle, \; \overrightarrow{v_9} = \langle -2, 2, -2, 2 \rangle, \; \overrightarrow{v_{11}} = \langle -1, 1, -1, 1 \rangle$$
$$\overrightarrow{v_{13}} = \langle 1, -1, 1, -1 \rangle, \; \overrightarrow{v_{14}} = \langle 1, 1, -1, -1 \rangle, \; \overrightarrow{v_{15}} = \langle 0, -1, 1, 0 \rangle$$

In the previous example, the edge sets of G_6 and G_9 are perfect matchings. Call a bipartite graph with an edge set that is a perfect matching a *perfect matching graph*, the matching polynomial associated with the graph a *perfect matching polynomial*, and the vector associated with the perfect matching polynomial a *perfect matching vector*. These perfect matching structures play an important role in both the graphs and in our matching polynomials.

THEOREM 4.1. The n-tuple perfect matching vectors form a basis for the n-dimensional matching vectors corresponding to bipartite graphs on $2n$ vertices.

Proof. Let $\mathcal{B} = \left\{ \overrightarrow{b_\pi} \mid \pi \in S_n \right\}$ be the set of vectors determined by the perfect matching graphs on $2n$ vertices and $\mathcal{V}$ be the set of vectors determined by all bipartite graphs on $2n$ vertices with at least one perfect matching. From the end of Section 1.1, a basis is a set of linearly independent vectors that span a vector space. We must show that the perfect matching vectors are linearly independent and span.

To show $\mathcal{B}$ is linearly independent, assume $\overrightarrow{b_\sigma} = \sum_{\pi \in S_n, \pi \neq \sigma} c_\pi \overrightarrow{b_\pi}$ for some constants c_π, where $\overrightarrow{b_\sigma} \in \mathcal{B}$. This implies

$$f_\sigma(x) = \sum_{\pi \in S_n, \pi \neq \sigma} c_\pi f_\pi(x), \tag{4.1}$$

where $f_\sigma(x)$ is the polynomial determined by a perfect matching graph.

By Proposition 3.1, $f_\sigma(x)$ has a nonzero evaluation on exactly one input, σ. If (4.1) is evaluated at σ, the left hand side is nonzero while the right hand side is the sum of $n! - 1$ terms that all evaluate to zero. Therefore, the assumption that $\overrightarrow{b_\sigma} = \sum_{\pi \in S_n, \pi \neq \sigma} c_\pi \overrightarrow{b_\pi}$ is incorrect and, by contradiction, $\mathcal{B}$ is linearly independent.

To show $\mathcal{B}$ spans, let $\overrightarrow{v} \notin \mathcal{B}$ be some vector determined by a bipartite graph on $2n$ vertices with at least one perfect matching. Let $f(x)$ be our matching polynomial for some graph G. We must show $\overrightarrow{v} = \sum_{i=1}^{n!} c_i \overrightarrow{b_i}$ or, equivalently,

$$f(x) = \sum_{i=1}^{n!} c_i f_i(x),$$

where $f_i = f_{\overrightarrow{b_i}}$ is the polynomial determined by the $\overrightarrow{b_i}$ perfect matching graph.

Let $\alpha = \{\alpha_0, \alpha_1, \ldots, \alpha_{n-1}\}$ describe a perfect matching in the graph G where vertex i is adjacent to vertex $\alpha_i = \omega^k$ for some $k \in \{0 \ldots n-1\}$. When $f(x)$ is evaluated at α, the result is nonzero by Proposition 3.1. Since the f_i's are nonzero

only on a perfect matching, when evaluated at α all are zero except $f_\alpha(\alpha)$. Thus, we can write $f(\alpha) = c_\alpha f_\alpha(\alpha)$ for some constant c_α. If $\beta = \{\beta_0, \beta_1, \ldots, \beta_{n-1}\}$ is not a perfect matching in the graph G then $f(\beta) = 0$ by Proposition 3.1 and likewise $c_\beta = 0$.

Since the only points at which $f(x)$ evaluates to a nonzero value are at perfect matchings, running through all perfect matchings yields the constants of the f_i's. Thus $f(x)$ can be written as a linear combination of the $f_{\overrightarrow{b_i}}$'s and similarly, $\overrightarrow{v}$ can be written as a linear combination of the $\overrightarrow{b_i}$'s. The definition of spanning a vector space is satisfied, showing $\mathcal{B}$ spans $\mathcal{V}$.

Since $\mathcal{B}$ is linearly independent and spans the space of all coefficient vectors, it is a basis for $\mathcal{V}$. □

COROLLARY 4.1. Our matching polynomial $f_G(x) = f(x)$ is a linear combination of the perfect matching polynomials for the perfect matchings in G. Furthermore, $f(\pi) = c_\pi f_\pi(\pi)$ where π is a perfect matching in the graph G.

Proof. This result follows directly from the previous theorem and how the coefficient vectors are defined. □

Through the main result in this section, theorem 4.1 and its corollary, the matching polynomial in a general bipartite graph can be viewed as this weighted average of the perfect matching polynomials. This result is a nice combinitorial interpretation of what is happening with these polynomials and how they interact with the perfect matchings in the graphs. Our matching polynomials are built on the perfect matchings in the graphs.

The proof of Propostion 3.1, showing that there was a perfect matching if and only if $f(x)$ had a nonzero evaluation, relied on the clever construction of our matching polynomials. The previous result further emphasizes this property and gives an intuitive reason for the results of evaluating our matching polynomial.

Since the perfect matching polynomials have exactly one input for which they are nonzero and our matching polynomial is a linear combination of the perfect matching polynomials for the perfect matchings in the graph, it follows that our matching polynomial will have nonzero entries only on the inputs that correspond to perfect matchings in the graph.

The next example demonstrates the theorem using the reults from the previous example.

EXAMPLE 4.3. In Example 4.2 the perfect matching graphs are G_6 and G_9 with vectors $\overrightarrow{v_6} = \langle 2, 2, -2, -2 \rangle$ and $\overrightarrow{v_9} = \langle -2, 2, -2, 2 \rangle$.

$$\begin{aligned}
\overrightarrow{v_7} &= \frac{-1}{2}\overrightarrow{v_6} + 0\overrightarrow{v_9} \\
\overrightarrow{v_{11}} &= 0\overrightarrow{v_6} + \frac{1}{2}\overrightarrow{v_9} \\
\overrightarrow{v_{13}} &= 0\overrightarrow{v_6} + \frac{-1}{2}\overrightarrow{v_9} \\
\overrightarrow{v_{14}} &= \frac{1}{2}\overrightarrow{v_6} + 0\overrightarrow{v_9} \\
\overrightarrow{v_{15}} &= \frac{-1}{4}\overrightarrow{v_6} + \frac{-1}{4}\overrightarrow{v_9}
\end{aligned}$$

This section concerned our matching polynomial. The next section will look at results on the transform of our matching polynomial.

4.2 A Basis for the Transform

Theorem 4.1 shows that the perfect matching polynomials form a basis for all matching polynomials of bipartite graphs. This section will give a similar result for the discrete Fourier transforms of the matching polynomials. By the end of this section we see that our perfect matching polynomials are also building blocks for the transforms of our matching polynomials.

In order to prove the main result of this section, a lemma will be helpful. Recall that the discrete Fourier transform of our matching function is given by

$$\widehat{f_\pi}(\chi_\rho) = \sum_{\alpha \in \Omega_n^n} f_\pi(\alpha)\overline{\chi_\rho(\alpha)}$$

LEMMA 4.1. If f_π is a perfect matching polynomial, then $\widehat{f_\pi}(\chi_\rho) = f_\pi(\pi)\overline{\chi_\rho(\pi)}$.

Proof. Let $\widehat{f_\pi}(\chi_\rho)$ be the transform of a perfect matching polynomial, f_π. By proposition 3.1, f_π is nonzero only when evaluated at π. Thus,

$$\widehat{f_\pi}(\chi_\rho) = \sum_{\alpha \in \Omega_n^n} f_\pi(\alpha)\overline{\chi_\rho(\alpha)} = \sum_{\alpha = \pi} f_\pi(\alpha)\overline{\chi_\rho(\alpha)} = f_\pi(\pi)\overline{\chi_\rho(\pi)} \tag{4.2}$$

□

With this lemma, the main result for this section can be proven.

THEOREM 4.2. The transform of a matching polynomial is a linear combination of the transform of the perfect matching polynomials.

Proof. The transform of the matching polynomial, $\widehat{f}(\chi_\rho)$, is given by $\sum_{\alpha \in \Omega_n^n} f(\alpha)\overline{\chi_\rho(\alpha)}$. By Proposition 3.1, $f(\alpha)$ is nonzero only when α is a perfect matching realized in the graph and perfect matchings are equivalent to permutations. Therefore, $\sum_{\alpha \in \Omega_n^n} f(\alpha)\overline{\chi_\rho(\alpha)}$ can be restricted to $\alpha \in S_n$. That is,

$$\widehat{f}(\chi_\rho) = \sum_{\alpha \in S_n} f(\alpha)\overline{\chi_\rho(\alpha)}.$$

Expanding the above sum yields

$$\widehat{f}(\chi_\rho) = f(\pi_1)\overline{\chi_\rho(\pi_1)} + f(\pi_2)\overline{\chi_\rho(\pi_2)} + \ldots + f(\pi_{n!})\overline{\chi_\rho(\pi_{n!})}$$

where π_i is an element of S_n. Using Corollary 4.1, $f(\pi_i) = c_{\pi_i} f_{\pi_i}(\pi_i)$, the above can be written as

$$\widehat{f}(\chi_\rho) = c_{\pi_1} f_{\pi_1}(\pi_1)\overline{\chi_\rho(\pi_1)} + c_{\pi_2} f_{\pi_2}(\pi_2)\overline{\chi_\rho(\pi_2)} + \ldots + c_{\pi_{n!}} f_{\pi_{n!}}(\pi_{n!})\overline{\chi_\rho(\pi_{n!})}. \tag{4.3}$$

Using Equation (4.2) from the lemma, Equation (4.3) can be rewritten as

$$\widehat{f}(\chi_\rho) = c_{\pi_1}\widehat{f_{\pi_1}}(\chi_\rho) + c_{\pi_2}\widehat{f_{\pi_2}}(\chi_\rho) + \ldots + c_{\pi_{n!}}\widehat{f_{\pi_{n!}}}(\chi_\rho) = \sum_{\pi \in S_n} c_\pi \widehat{f_\pi}(\chi_\rho).$$

□

This theorem has a corollary which reveals a little more about the transform of the matching polynomials.

COROLLARY 4.2. The transform of a matching polynomial is a linear combination of the perfect matching polynomials.

Proof. This result follows from Theorem 4.2 and Lemma 4.1. □

This result shows that the perfect matching polynomials are essential to the construction of both our matching polynomials and the transforms of our matching polynomials. This last result should not be surprising. The discrete Fourier transform of our matching polynomial is built on the matching polynomials and the main result of the previous section showed that our matching polynomials are built from the perfect matching polynomials.

Theorem 3.3 gave a bound on the number of perfect matchings by using the uncertainity principle. The uncertainity principle relied on the support of the matching polynomial and the support of the transform of the matching polynomial. The above corollary shows that it is not necessary to compute the transform of the matching polynomial to find the bound because the transform is a linear combination of the perfect matching polynomials, although the characters of the group $\mathbb{Z}_n^n$ are still necessary.

4.3 Coefficients as Determinants

The previous two sections helped give a combinatorial understanding of the coefficients of both the matching polynomials and the transforms of the matching polynomials. This section gives an algebraic interpretation of the coefficients of the matching polynomials.

Equation (3.8), $\widehat{f}(\chi) = n^n \left|\det(L[r(i)+1])\right|$ shows that the transform of the matching polynomial can be thought of as the determinant of a matrix. The next

theorem will provide a similar result for the matching polynomial.

Let $f(x)$ be a polynomial in the variables $x_0, x_1, \ldots, x_{n-1}$ modulo the ideal $\langle x_i^n - 1\rangle_{i=1}^n$. Thus, the exponents on the variables in $f(x)$ range between 0 and $n-1$. For some term $t = x_0^{r_0} x_1^{r_1} \ldots x_{n-1}^{r_{n-1}}$, define the *term annihilator* of t, to be the term $\tau(t) = \tau(x_0^{r_0} x_1^{r_1} \ldots x_{n-1}^{r_{n-1}}) = x_0^{n-r_0} x_1^{n-r_1} \ldots x_{n-1}^{n-r_{n-1}}$. Then, $t \times \tau(t) = x_0^n x_1^n \ldots x_{n-1}^n \equiv 1$ modulo $\langle x_i^n - 1\rangle_{i=0}^{n-1}$; the variables are annihilated in the term and it becomes a constant.

THEOREM 4.3. If

$$f_G(x) = \sum_{\alpha \in \mathbf{Z}_n^n} d_\alpha x_0^{\alpha_0} x_1^{\alpha_1} \ldots x_{n-1}^{\alpha_{n-1}} = \sum_{\alpha \in \mathbf{Z}_n^n} d_\alpha x^\alpha$$

then

$$d_\alpha = \det L[(n-1-\alpha_0), (n-1-\alpha_1), \ldots, (n-1-\alpha_{n-1})] = \det L[n-1-\alpha]$$

The proof of the theorem is easier to follow after having seen an example first.

EXAMPLE 4.4. For some graph G, one can associate our matching polynomial $f(x)$ and some coefficient matrix

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}.$$

Then the following polynomials make up $f(x)$.

$$\begin{aligned} V(x) &= (x_3 - x_2)(x_3 - x_1)(x_2 - x_1) \\ &= x_2 x_3^2 - x_2^2 x_3 - x_1 x_3^2 + x_1 x_2^2 + x_1^2 x_3 - x_1^2 x_2 \\ g_1(x_1) &= a_{11} + a_{12} x_1 + a_{13} x_1^2 \\ g_2(x_2) &= a_{21} + a_{22} x_2 + a_{23} x_2^2 \\ g_3(x_3) &= a_{31} + a_{32} x_3 + a_{33} x_3^2. \end{aligned}$$

To find the constant term of $f(x)$ the term annihilators of each term in $V(x)$ must be found. Look at each term of $V(x)$ and the corresponding term annihilator in $g_1 \times g_2 \times g_3$:

$$\begin{aligned}
\tau(x_2x_3^2) &= a_{11}a_{23}x_2^2a_{32}x_3 = a_{11}a_{23}a_{32}x_2^2x_3 \\
\tau(x_2^2x_3) &= a_{11}a_{22}x_2a_{33}x_3^2 = a_{11}a_{22}a_{33}x_2x_3^2 \\
\tau(x_1x_3^2) &= a_{13}x_1^2a_{21}a_{32}x_3 = a_{13}a_{21}a_{32}x_1^2x_3 \\
\tau(x_1x_2^2) &= a_{13}x_1^2a_{22}x_2a_{31} = a_{13}a_{22}a_{31}x_1^2x_2 \\
\tau(x_1^2x_3) &= a_{12}x_1a_{21}a_{33}x_3^2 = a_{12}a_{21}a_{33}x_1x_3^2 \\
\tau(x_1^2x_2) &= a_{12}x_1a_{23}x_2^2a_{31} = a_{12}a_{23}a_{31}x_1x_2^2
\end{aligned}$$

Thus, the constant term of $f(x)$ is

$$a_{11}a_{23}a_{32} - a_{11}a_{22}a_{33} - a_{13}a_{21}a_{32} + a_{11}a_{23}a_{32} + a_{12}a_{21}a_{33} - a_{12}a_{23}a_{31}.$$

This corresponds exactly to the determinant of A when each row has been rotated $n-1$ positions.

Proof. Recall $f(x) = \prod_{i=1}^{n} g_i(x_i)V(x)$ and that we are working modulo the ideal $\langle x_i^n - 1\rangle_{i=1}^n$. Let $G(x) = \prod_{i=1}^{n} g_i(x_i)$; then, $f(x) = G(x)V(x)$. Let $A = [a_{i,j}]$ be the coefficient matrix defined by the graph from which the fuction $f(x)$ is defined. The proof is accomplished by showing the constant term is the desired determinant, and then using the term annihilator to show the other terms.

Since $f(x)$ is the product of two functions, the constant term of $f(x)$ is determined by each term of $V(x)$ and the corresponding term annihilator for each term of $V(x)$. One formulation of $V(x)$ is $V(X) = \sum_{\pi \in S_n} sign(\pi) \prod_{i=0}^{n-1} x_i^{\pi(i)}$.

Let $t_\pi = x_0^{\pi(0)}x_1^{\pi(1)} \ldots x_{n-1}^{\pi(n-1)}$ be a term of $V(x)$. To construct $\tau(t)$ from $G(x)$, note that x_i appears only in $g_i(x_i)$. The coefficient of $x_i^{n-\pi(i)}$ is $sign(\pi)a_{i,n-\pi(i)}$. Thus, $\tau(t) = \prod_{i=0}^{n-1} a_{i,n-\pi(i)}x_i^{n-\pi(i)}$. Note that the coefficient of the term annihilator of

t_π is a transversal of the coefficient matrix where each row has been shifted $n-1$ positions. This gives one part of the constant term of $f(x)$.

Since $V(x)$ runs through all permutations in S_n and the corresponding term annihilators give transversals of the shifted coefficient matrix, the constant term of $f(x)$ is a signed sum of all transversals of the shifted coefficient matrix. This matches the definition of a determinant given in Section 1.1. Thus, the constant term is the determinant of the shifted coefficient matrix.

To see that all other coefficients are also determinants, let t be the term under consideration. Multiplying $f(x)$ by $\tau(t) = x_0^{b_0} x_1^{b_1} \dots x_{n-1}^{b_{n-1}}$ makes the coefficient of t in $f(x)$ become the constant term in $\widetilde{f(x)} = \tau(t) f(x)$. It was just shown that the constant term is a determinant. The effect of multiplying $f(x)$ by $\tau(t)$ must be considered.

Since multiplication is commutative and associative the effect of multiplying $f(x)$ by $\tau(t)$ is the same as multiplying each g_i in $f(x)$ by the appropriate $x_i^{b_i}$. Such a multiplication shifts the exponents on each x_i by b_i. This has the same effect as rotating row i of the coefficient matrix by $n-1-b_i$. The previous result on the constant term of $f(x)$ holds for the constant term of $\widetilde{f(x)}$ which is the coefficient of the term t in $f(x)$. □

These functions, both our matching function and its transform, hold much of the information about the graph. This chapter showed some of the properties of our these functions. One common property that appeared many times is that the perfect matchings are building blocks for the graphs and for the polynomials. This property appeared not only in our matching functions but also in the transforms of our matching functions.

Another property that was explored in this section is how the transforms have many analogous properties of our matching functions. It was shown in Section 3.3 that the transform of our matching function can be thought of as a determinant.

In this section it was shown how the coefficients of our matching function are related to a similar determinant.

The next chapter will show how these functions can be helpful with other problems.

CHAPTER 5
APPLICATIONS, CONCLUSIONS, AND FUTURE WORK

This chapter provides an application of the work done in this dissertation as well as a summary of what was accomplished and where this idea might be extended in the future. We have spent the previous two chapters creating and examining properties of our matching polynomials and the transform. Here we see how to use this information in ways to solve other problems.

5.1 Applications

One of the more difficult aspects of applying the Combinatorial Nullstellensatz to a problem is finding a nonzero coefficient in the polynomial used to model the problem. The methods discussed in this dissertation provide a quick way to check for a nonzero coefficient for our matching polynomial by using a matroid intersection algorithm. If such an approach could be applied to other functions, it would be an efficient method to apply the Combinatorial Nullstellensatz.

EXAMPLE 5.1. Consider the following function:

$$h(x) = -5 - 19x_1 + 13x_0 + 5x_0x_1.$$

Does this polynomial vanish on all second roots of unity? For this polynomial, since there are only two variables, a brute force method of checking might make sense since only there would be only 4 possible options to consider: $(x_0 = 1, x_1 = 1), (x_0 = 1, x_1 = -1), (x_0 = -1, x_1 = 1), (x_0 = -1, x_1 = -1)$. These are the only

options to consider because the two variables mean we are working over the second roots of unity.

However, what follows is an alternate approach to check if it vanishes on all second roots of unity. The function $h(x)$ can be rewritten as

$$\begin{aligned} h(x) &= -5 - 19x_1 + 13x_0 + 5x_0x_1 \\ &= (-1 - x_1 + x_0 + x_0x_1) + 4(-1 + x_1 - x_0 + x_0x_1) + 16(-x_1 + x_0) \\ &= f_7(x) + 4f_{11}(x) + 16f_{15}(x) \end{aligned}$$

where $f_7(x)$, $f_{11}(x)$, and $f_{15}(x)$ are our matching polynomials defined in Example 4.2 corresponding to the graphs given in Figure 4.1. It is known that these polynomials do not vanish on all second roots of unity. There is a possibility, however, that upon evaluation the results of our matching polynomials might sum to zero. In this instance that is not a problem as the coefficients on the polynomials are large enough to spread out the evaluations.

The above polynomial was constructed specifically so that it would factor nicely into a linear combination of known matching polynomials. The next example works through an algorithmic approach, namely the division algorithm over $\mathbb{C}[x_0, x_1, \ldots, x_{n-1}]$, when it might not be as clear that it can be written as a linear combination of our matching polynomials.

EXAMPLE 5.2. Suppose we are given the following polynomial:

$$h(x) = 1 + x_3^2x_1^2 - x_1^2x_2^2 + x_1x_3x_2^2 - x_3^2x_1x_2 - x_3 + x_2.$$

For this polynomial attempting a brute force evaluation starts to become less tractable. Since there are three variables the 3rd roots of unity need to be considered, and each needs to be considered for each variable. This gives $3^3 = 27$ cases to consider. While 27 is not an overly large number, as the number of variables

increases to n the number of cases to consider for a brute force evaluation increases to n^n, which grows very quickly, where as few as 8 variables gives rise to nearly 17 million cases to consider.

By choosing appropriate terms to add and subtract, $h(x)$ can be transformed into a linear combination of our matching polynomials with some remainder. Ideally the remainder can handled easily. Our matching polynomial for the complete bipartite graph on 6 vertices (see Figure 5.1) is given by:

$$f_{511}(x) = x_1^2x_2 - x_3x_1^2 + x_1x_3^2 - x_1x_2^2 + x_3x_2^2 - x_2x_3^2.$$

Adding and subtracting the terms of f_{511} invites a division algorithm.

$$\begin{aligned} h(x) &= 1 + x_3^2x_1^2 - x_1^2x2^2 + x_1x_3x_2^2 - x_3^2x_1x_2 - x_3 + x_2 \\ &= 1 + x_3^2x_1^2 - x_1^2x2^2 + x_1x_3x_2^2 - x_3^2x_1x_2 - x_3 + x_2 + f_{511} - f_{511} \\ &= f_{511}(x) + 1 + x_3^2x_1^2 - x_1^2x_2^2 + x_1x_3x_2^2 - x_3^2x_1x_2 - x_3 + x_2 \\ &\quad -x_1^2x_2 + x_3x_1^2 - x_1x_3^2 + x_1x_2^2 - x_3x_2^2 + x_2x_3^2 \end{aligned}$$

At first glance it does not appear that $h(x)$ has been simplified or made easier to work with. In fact, by adding and subtracting $f_{511}(x)$ it appears that more terms have been introduced. Continuing the process of adding and subtracting appropriate terms to find examples of our matching polynomial will provide additional insight.

Our matching polynomial for the complete bipartite graph on 6 vertices missing one edge is given by:

$$f_{510} = x_3^2x_1^2 - x_1^2x2^2 + x_1x_3x_2^2 - x_3^2x_1x_2 - x_3 + x_2 - x_1^2x_2 + x_3x_1^2 - x_1x_3^2 + x_1x_2^2 - x_3x_2^2 + x_2x_3^2.$$

Checking the terms of $h(x)$ against the terms in $f_{510}(x)$ and we see that all of the terms of $f_{510}(x)$ appear. Thus $h(x)$ can be rewritten in terms of $f_{511}(x)$, $f_{510}(x)$ and a remainder. In this case, $h(x) = f_{511}(x) + f_{510}(x) + 1$. Since $f_{511}(x)$ and $f_{510}(x)$ are examples of our matching polynomial we know that they do not vanish on all

third roots of unity. The remainder, 1, never vanishes either. Thus, $h(x)$ will not vanish on all third roots of unity as long as one is careful to check that cancellation of the terms does not happen upon evaluation.

It might appear that $h(x)$ in Example 5.2 was chosen specifically so that when $f_{511}(x)$ and $f_{510}(x)$ were removed the remainder was something nice. The next example shows that while choosing the appropriate matching polynomial makes the problem simplify quicker, there are other possibilities as well.

EXAMPLE 5.3. Consider the following graphs.

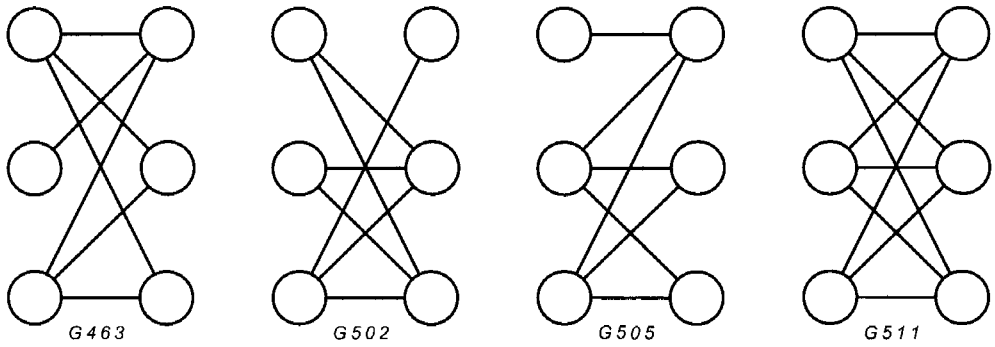

Figure 5.1: Bipartite Graphs on 6 Vertices

The reduced adjacency matrices of the graphs in Figure 5.1 are

$$G_{463} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}, \; G_{502} = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix},$$

$$G_{505} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \; G_{511} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

The coefficient vectors for these graphs are

$$\begin{aligned}
\overrightarrow{v_{463}} &= \langle 0,1,-1,0,1,-1,0,1,-1,-1,0,1,-1,0,1,-1,0,1,1,-1,0,1,-1,0,1,-1,0\rangle \\
\overrightarrow{v_{502}} &= \langle 0,0,0,-1,-1,-1,1,1,1,1,1,1,0,0,0,-1,-1,-1,-1,-1,-1,1,1,1,0,0,0\rangle \\
\overrightarrow{v_{505}} &= \langle 0,-1,1,1,0,-1,-1,1,0,0,-1,1,1,0,-1,-1,1,0,0,-1,1,1,0,-1,-1,1,0\rangle \\
\overrightarrow{v_{511}} &= \langle 0,0,0,0,0,-1,0,1,0,0,0,1,0,0,0,-1,0,0,0,-1,0,1,0,0,0,0,0\rangle.
\end{aligned}$$

Compare

$$\overrightarrow{v_{463}}+\overrightarrow{v_{502}}+\overrightarrow{v_{505}} = \langle 0,0,0,0,0,-3,0,3,0,0,0,3,0,0,0,-3,0,0,0,-3,0,3,0,0,0,0,0\rangle$$

with

$$3 \times \overrightarrow{v_{511}} = \langle 0,0,0,0,0,-3,0,3,0,0,0,3,0,0,0,-3,0,0,0,-3,0,3,0,0,0,0,0\rangle.$$

Thus, $\overrightarrow{v_{511}} = 1/3(\overrightarrow{v_{463}}+\overrightarrow{v_{502}}+\overrightarrow{v_{505}})$. In Example 5.2, $f_{511}(x)$ could have been replaced by $1/3(f_{463}(x) + f_{502}(x) + f_{505}(x))$.

The above example, though, is not the only way to rewrite $f_{511}(x)$ as a linear combination of other matching polynomials. Using Corollary 4.1, we have $f_{511}(x) = 1/27(f_{84}(x) + f_{98}(x) + f_{140}(x) + f_{161}(x) + f_{266}(x) + f_{273}(x))$ where $f_{84}(x)$, $f_{98}(x)$, $f_{140}(x)$, $f_{161}(x)$, $f_{266}(x)$, $f_{273}(x)$ are the perfect matching polynomials corresponding to the perfect matching graphs in Figure 5.2. Thus in Example 5.2, $f_{511}(x)$ could have been replaced by the perfect matching functions, with the appropriate scalar.

Similarly, Corollary 4.1 allows us to write $f_{510}(x)$ as $(-1/18-i\sqrt{3}/54)f_{84}(x)+(-1/18+i\sqrt{3}/54)f_{98}(x)+(-1/18-i\sqrt{3}/54)f_{140}(x)+(-1/18+i\sqrt{3}/54)f_{161}(x)$. Thus, there are many different ways to write $h(x)$ in terms of our matching polynomials. A future goal, explained in more detail in Section 5.3 is to characterize which polynomials can be written as combinations of our matching polynomials.

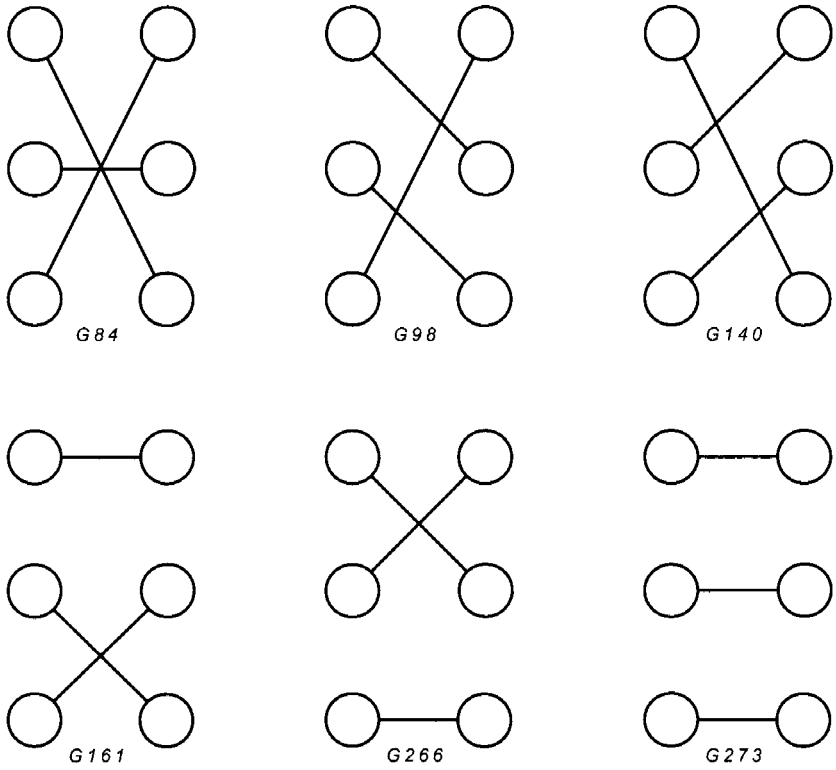

Figure 5.2 – All Bipartite Perfect Matching Graphs on 6 Vertices

5.2 Conclusions

Finding an efficient method by which the Combinatorial Nullstellensatz can be used was the main goal of this dissertation. While finding the exact number of perfect matchings in a bipartite graph is $\#P$-complete (see [21]), it is easy, in a complexity sense, to find a lower bound on the number of perfect matchings in a bipartite graph. The matroid intersection algorithm provides a polynomial time algorithm to find a nonzero coefficient in the expansion of our matching polynomial. The existence of a nonzero coefficient is necessary in order to apply the Combinatorial Nullstellensatz to the problem.

The number of perfect matchings in a bipartite graph and the number of maxium independent sets in the intersection of two matroids are related. This relation is not obvious, however. Using the Nullstellensatz method and the discrete

Fourier transform, the relationship was revealed through the uncertainity principle. This relationship has helped understand the application of the Combinatorial Nullstellensatz. It seems that incorporating the matroid intersection algorithm will allow for a sharper version of the Combinatorial Nullstellensatz, as it did in the problem of bounding the number of perfect matchigns in a bipartite graph.

After using the Nullstellensatz method additional questions arose concerning the polynomials created by the instances that corresponded to the bipartite graphs. These polynomials contain information about the graphs, such as structural information. One thing that became apparent through the results in Chapter 4 was that the perfect matchings in the graph play an important role in the polynomials and their transforms. The perfect matching polynomials serve as building blocks for our matching polynomials and also for the discrete Fourier transforms of our matching polynomials.

Our matching polynomials are well understood, in a Nullstellensatz sense. They are well behaved in that finding a nonzero coefficient can be done in polynomial time via the matroid intersection algorithm. Since finding a nonzero coefficient can be difficult, being able to use polynomials for which there is a known nonzero coefficient is useful if the Combinatorial Nullstellensatz is going to applied. It was shown that some polynomials can be written as a combination of our matching polynomials. In some cases where the polynomial is made up of only matching polynomials, it becomes clear that the polynomial does not vanish on all nth roots of unity.

Unfortunately, not all polynomials can be written as a combination of only our matching polynomials. In some cases there is a remainder term. Since a brute force evaluation to see if the polynomial disappears on all nth roots of unity requires n^n evaluations, it is possible that even with a remainder the evaluations that must be checked can be reduced. This case was seen in Example 5.2 where the remainder

term was a constant, thus it never vanished.

5.3 Future Work

There are several directions for further research.

1). **Extend these results to general matching and f-factor theorems.** A perfect matching is also called a *1-factor*. The edges of a perfect matching form a subgraph containing every vertex of the graph and the degree of each vertex in the subgraph is one. In general, an f-factor of a graph is an subgraph such that every vertex appears in the subgraph and the degree of every vertex in the subgraph is f. Extending these results to 2-factors and beyond would be nice as well as extending to general matchings.

2). **Extend these results to different families of graphs.** The results presented in this dissertation work for bipartite graphs, one family of graphs. There are many other graph families such as complete graphs (see Section 1.2) claw-free graphs, planar graphs, k-regular graphs, or triangle-free graphs, just to name a few. It would be nice if similar results could be found for some of these other families.

3). **Identify properties of these extensions that permit easy application of the Combinatorial Nullstellensatz.** As was stated in Section 2.4, most of the successful applications of the Combinatorial Nullstellsatz have the property that for all instances of the same size have a common monomial with a nonzero coefficient. This allows finding a nonzero coefficient for all instances by looking at a canonical instance.

4). **Characterize which polynomials can be written as combinations of our matching polynomials.** The applications given in Section 5.1 hint at this idea. Two polynomials are written as combinations of our matching polynomials in the examples presented. Both of these example polynomials were selected because

they allowed for an easy rewritting in terms of our matching polynomials. In general, though, it is not known which polynomials can be written as combinations of our matching polynomials. Being able to characterize such polynomials could allow for an efficient method to determine if a certain polynomial vanishes on all nth roots of unity and thus allow for an application of the Combinatorial Nullstellensatz.

5). **Relate $r(G)$, the size of support of the discrete Fourier transform of our matching polynomial, to structural properties of G.** It is clear how $m(G)$, the size of the support of our matching polynomial, relates to the structure of the graph G. It tells exactly how many perfect matchings exist in the graph. As was seen in Chapter 4, many of the properties of our matching polynomials have analogous properties for the transforms. Since $m(G)$ reveals a structural property of the graph there is hope that $r(G)$ has a similar revelation about the structure of the graph.

In writing this dissertation, much was discovered about the properties of our matching functions, their transforms, and how they relate to the structure of the graph. These discoveries have helped to understand perfect matchings in bipartite graphs as well as ways in which the Combinatorial Nullstellensatz can be applied in novel ways. Much of the future work described will continue to provide insight into the topics covered in this dissertation.

REFERENCES

[1] Noga Alon, *Combinatorial nullstellensatz*, Combinatorics, Probability, and Computing **8** (1999), no. 1–2, 7–29.

[2] Michael Artin, *Algebra*, Prentice Hall, Inc., Englewood Cliffs, 1991.

[3] Richard Brauer, *Investigations on group characters*, Annals of Mathematics **42** (1941), no. 2, 936–958.

[4] Richard Brauer and Cecil Nesbitt, *On the modular characters of groups*, Annals of Mathematics **42** (1941), no. 2, 556–590.

[5] Gary Chartrand and Lisa Lesniak, *Graphs and digraphs*, fourth ed., Chapman & Hall/CRC, Boca Raton, 2005.

[6] David L. Donoho and Philip B. Stark, *Uncertainity principles and signal recovery*, SIAM Journal of Applied Mathematics (1989), no. 3, 906–931.

[7] Jack Edmonds, *Discrete optimization (proc. adv. res. inst. discrete optimization and systems appl., banff, alta., 1977), i.*, Annals of Discrete Mathematics **4** (1979), 39–49.

[8] F. Georg Frobenius, *Gesammelte abhandlungen. bnde i, ii, iii (german)*, Springer-Verlag, Berlin, 1969.

[9] Joseph A. Gallian, *Contemporary abstract algebra*, fourth ed., Houghton Mifflin Company, Boston, 1998.

[10] Michael R. Garey and David S. Johnson, *Computers and intractability: A guide to the theory of np-completeness*, W. H. Freeman and Company, New York, 1979.

[11] John E. Hopcroft and Richard M. Karp, *An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs*, SIAM Journal of Computing **2** (1973), no. 4, 225–231.

[12] André E. Kézdy, *ρ-valuations for some stunted trees*, Discrete Mathematics **306** (2006), no. 21, 2786–2789.

[13] André E. Kézdy and Hunter S. Snevily, *Polynomials that vanish on distinct nth roots of unity*, Combinatorics, Probability, and Computing **13** (2004), no. 1, 37–59.

[14] Eugene Lawler, *Combinatorial optimization: Networks and matroids*, Dover Publications, Inc., Mineola, 1976.

[15] David C. Lay, *Linear algebra and its applications*, second ed., Addison Wesley Longman, Inc., Reading, 1996.

[16] James Oxley, *Matroid theory*, Oxford University Press, Oxford, 1992.

[17] Gian-Carlo Rota and Henry H. Crapo, *On the foundations of combinatorial theory: Combinatorial geometries*, M.I.T. Press, Cambridge, 1970.

[18] A. Schrijver, *Counting 1-factors in regular bipartite graphs*, Journal of Combinatorial Theory, Series B **72** (1998), no. 1, 122–135.

[19] Jean-Piere Serre, *Linear representations of finite groups*, second ed., Springer-Verlag, New York, 1977.

[20] Audrey Terras, *Fourier analysis on finite groups and applications*, London Mathematical Society Student Texts, no. 43, Cambridge University Press, Cambridge, fourth ed., 1999.

[21] Leslie G. Valiant, *The complexity of computing the permanent*, Theoretical Computer Science **8** (1979), no. 2, 189–201.

[22] Marc Voorhoeve, *A lower bound for the permanents of certain (0, 1)-matrices*, Koninklijke Nederlandse Akademie van Wetenschappen. Indagationes Mathematicae **41** (1979), no. 1, 83–86.

[23] Hassel Whitney, *On the abstract properties of linear dependence*, American Journal of Mathematics **57** (1935), 509–533.

[24] Robin J. Wilson, *Introduction to graph theory*, fourth ed., Addison Wesley Longmann Limited, Essex, 1996.

INDEX

CURRICULUM VITAE

Timothy M. Brauch

Education

University of Louisville, Louisville, Kentucky
Ph.D., Applied and Industrial Mathematics, expected Spring 2009
M.A., Mathematics, May 2007

Wake Forest University, Winston-Salem, North Carolina
M.A., Mathematics, May 2004

Centre College, Danville, Kentucky
B.S., Mathematics & Spanish, May 2002

Professional experience

University of Louisville, Louisville, Kentucky
Graduate Teaching Assistant, Department of Mathematics, 2005–2009

University of Louisville, Louisville, Kentucky
National Science Foundation GK-12 Fellow, 2004–2005, 2007

Jefferson Community and Technical College, Louisville, Kentucky
Adjunct Instructor of Mathematics, 2005

Wake Forest University, Winston-Salem, North Carolina
Graduate Teaching Assistant, Department of Mathematics, 2002–2004

Forsyth Technical and Community College, Winston-Salem, North Carolina
Adjunct Instructor of Mathematics, 2003–2004

Centre College, Danville, Kentucky
Teaching Assistant, Department of Mathematics, 2000–2002

Invited presentations

The DFT and Perfect Matchings in Bipartite Graphs. May 2008
Twenty-First Cumberland Conference, Vanderbilt University, Nashville, Tennessee

Presentations

Studies of Perfect Matchings in Bipartite Graphs. September 2008
Algebra & Combinatorics Seminar, University of Louisville, Louisville, Kentucky

Perfect Matchings in Bipartite Graphs. March 2008
Mathematical Association of American Kentucky Chapter Annual Meeting, Western Kentucky University, Bowling Green, Kentucky

Creating the Next Generation of Scholars: The Call for Increased Professional Development of Doctoral Students. November 2005
2nd National Conference on Graduate Student Leadership, Washington University in Saint Louis, Missouri

Publications

The DFT and Perfect Matchings in Bipartite Graphs, in publication. (With Kézdy and Snevily)

Professional memberships

Omicron Delta Kappa Honor Society	Since 2007
Golden Key International Honour Society	Since 2006
Mathematical Association of America	Since 2004
Society for Industrial and Applied Mathematics	Since 2004
Pi Mu Epsilon	Since 2004
American Mathematical Society	Since 2002

Awards & fellowships

Ken F. and Sandra S. Hohman Fellowship, Department of Mathematics, University of Louisville, 2008-2009

Student Travel Grant, University of Louisville, 2005, 2007, 2009

James Graham Brown Leadership Scholar, Centre College, 1998 - 2002

Corella & Bertram F. Bonner Scholar, Centre College, 1999 - 2002