Cliques, Codes, and Association Schemes

Andrew Nagarajah
Supervised by: Prof. Mike Newman

15 May 2021

Abstract

This report is an exploration of some of the topics within *Delsarte theory*, which uses ideas from graph theory, algebra, and optimization to address questions in coding theory and combinatorics more broadly. The centre of study is the *association scheme*, which provides a setting in which to view various objects, especially *distance-regular graphs*. While computing the size of the largest clique or coclique in a graph is famously NP-hard, in this setting, upper bounds for these quantities can be computed much more efficiently using linear or semi-definite programming. This approach extends to the computation of upper bounds on the size of codes of a certain minimum distance; this was one of Delsarte's primary motivations.

The first chapter is contains an elementary introduction to the theory of error-correcting codes, which motivates this study. Instead of using the setting of finite vector spaces, this chapter discusses how coding theory is done in the setting of graphs, especially the Hamming graphs. Background on some important properties of the Hamming graphs is also given.

Motivated by the distance-regularity property of the Hamming graphs, the second chapter introduces the *association scheme* and its *Bose-Mesner algebra*. This chapter also discusses the P-polynomial and translation schemes, and uses the results obtained to calculate the eigenvalues of the Hamming graphs.

The third chapter introduces the problem of linear programming. Using many of the properties of association schemes from the second chapter, a linear program is constructed whose maximum is an upper bound on the size of codes. Using the weak duality property of linear programs, some general bounds are derived.

The fourth chapter introduces the problem of semi-definite programming. It also introduces the Terwilliger algebra of the binary Hamming scheme, as well as a semi-definite program in this algebra whose maximum is an upper bound on the size of codes. The relatively low dimension of this algebra makes this computationally feasible.

This report aims to be mostly self-contained, though background knowledge of basic linear algebra and group theory is required. The important aspects of this theory are reviewed in the appendix. The language of graph theory is used throughout.

Contents

1. Introduction

1.1 Coding Theory

In most communication, a transmitter in one location must send a message to a receiver in another location, possibly by means of a faulty (or *noisy*) channel which may introduce errors into the message.

For example, a spacecraft might send scientific data back to earth from another planet by means of a radio signal. Along the way, other sources of electromagnetic radiation might interfere with the signal, such that when it arrives at earth, the signal received is slightly different from the one sent. The scientific data, if successfully received, might be needed to be saved in some storage medium so that it may be accessed in the future. Over time however, the storage medium may degrade, so that when the data is retrieved, it may be slightly different from when it was saved.

In human speech, if two friends are speaking in a loud environment and one says "I love you", if the ambient noise is loud enough, the friend might hear "I lave you" instead. Not all possible sounds (or combinations of letters) are valid messages in English though, so the friend could reasonably guess at what was meant. In this sense, English contains *redundancy*. However, if the environment is so loud that the friend hears "I late you", then they might not know what the message what intended to mean, or incorrectly guess that the true message was "I hate you". On the other hand, if the friend hears "l shgbot gtin", then they will know for sure that the message was corrupted; they will know not to misinterpret the message, and may ask that it be re-transmitted.

These are the two fundamental goals of coding theory: to encode data in such a way that it can be recovered if some errors are introduced; or if it cannot be recovered, then at least to detect that the message was corrupted. As the above example illustrates, and encoding scheme may be able to correct a limited number of errors, or detect some number of errors. For this report, we will focus on the former goal.

In general, a coding scheme will consist of a set of signals V, among which a subset Y (called the CODE) might consist of valid messages. Then, the receiver will need a function

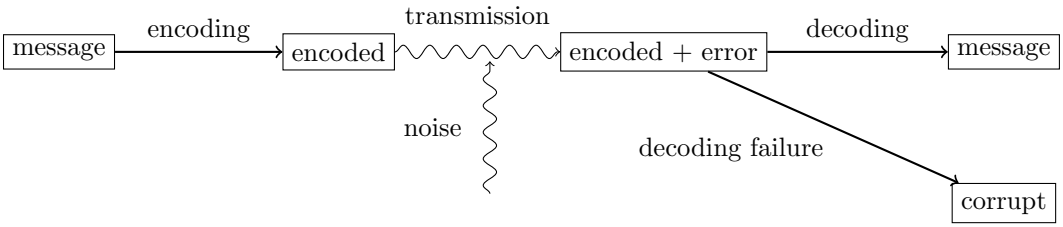

Figure 1.1: The process of encoding messages against transmission errors and decoding the resulting message.

$V \times V \to \mathbb{N}$ to tell how many errors would be required to change one signal into another. We assume that if s errors convert signal u into signal v, then s "opposite" errors can convert v into u. Furthermore, if v can be converted into w with t errors, then u can be converted into w with no more that $s+t$ errors (this is called the TRIANGLE INEQUALITY). Finally, u will require 0 errors to be converted into itself; conversely for any pair of distinct signals u and v, a nonzero number of errors should be required to convert one into the other. Such a set of signals, paired with such an error-counting function, is called a (DISCRETE) METRIC SPACE, and its function is typically called a DISTANCE. When a signal u arrives at the receiver, they can try and find a signal v in the code Y which minimizes the distance between u and v. If there is a unique such minimizer v, then we decode u into v.

For example, if we wish to encode the messages 0 (or 'no') and 1 (or 'yes'), the set of signals might consist of all binary strings of length three, with 000 and 111 as the codewords. If errors consist of individual bit flips, then the message 101 can be decoded into 111, as only one error is required to convert the latter into the former. However if two errors occur and 100 is received, then this will be decoded into 000, since fewer errors are required to convert between the two. Therefore, this code (called a *repetition code*) can correct at most 1 error.

More generally, if δ is the MINIMUM DISTANCE between any two codewords, and w is a signal at distance no more than $\left\lfloor \frac{\delta-1}{2} \right\rfloor$ from a codeword u, then its distance to any other codeword v must be greater. Otherwise, u could be converted to v with at most $\delta - 1$ errors – a contradiction!

On the other hand, in the example repetition code given above, there were $2^1 = 2$ codewords, and $2^3 = 8$ possible signals. So, for every message we wish to send, three times as many bits need to be transmitted. This is called the RATE of the code.

These two parameters, the minimum distance of the code, and the proportion of signals which are codewords, are the two primary measures of the effectiveness and efficiency of a code. A large minimum distance in a code will mean that many errors can be corrected, while a large proportion will mean that messages are being transmitted efficiently. The challenge, then, is to design codes which have simultaneously a large minimum distance and a large

proportion of codewords. One of the aims of this report is to put bounds on how efficient a code can be given a target effectiveness; plainly, if we want a certain minimum distance code in a set of signals, at most how many codewords can there be? The approach to this question is ultimately due to Delsarte [Del73], and has been expanded upon by Schrijver [Sch05].

This introduction was inspired by the introduction to [Ple98] (though it does not use the language of metrics).

1.2 Linear Codes and Distance-Regular Graphs

Let V be denote a vector space of dimension d over a finite field of order q. Then $Y \subseteq V$ is a LINEAR CODE if Y is a linear subspace of V. One such vector space is the set of length d bitstrings: this is a vector space of dimension d over the finite field of two elements. (Here, the addition of vectors corresponds to the bitwise exclusive-or of bitstrings.) A code in such a vector space would be immediately applicable to digital data. This motivates the study of linear codes. One way in which binary data can be corrupted is through bit flips. Analogously, if $v = v_1 e_1 + \cdots + v_d e_d$ is a vector written in the fixed basis $e_1, \ldots, e_d$, then an error may occur by picking an index i, and changing the value of v_i. In order to count such errors, we define the following distance function on V.

Definition 1.2.1
If V is the vector space of d-tuples over a finite field, then the (HAMMING) WEIGHT *of a vector $v = (v_1, \ldots, v_d)$ is the number of nonzero components v_i; this is denoted $|v|$. Then the* (HAMMING) DISTANCE *between vectors u, v is defined*

$$\operatorname{dist}(u, v) := |u - v| \ .$$

In other words, the distance between two vectors is the number of coordinates on which they differ. Since a linear subspace must be closed under subtraction, the minimum distance of a linear code is the same as its minimum (nonzero) weight. Meanwhile, the rate of the code is the ratio of the dimension of the code to the dimension of its ambient space.

While much of coding theory is done in this setting, this report will look at codes through a combinatorial lens. A DIRECTED GRAPH on a vertex set V is a binary relation $\Gamma \subseteq V \times V$. If the relation is symmetric, then the graph is said to be UNDIRECTED. (Note that this definition will prohibit parallel edges, but not loops.) The relation Γ is typically denoted $\sim$ and written infix when the graph is clear from the context. For each vector space as described above, we can define a corresponding graph and investigate codes in this graph instead.

Definition 1.2.2
The HAMMING GRAPH $H(d, q)$ *is defined so that its vertex set* V *is the set of all* d*-tuples, each entry of which belongs to a fixed set of* q *elements. Two vertices are said to be adjacent if they have Hamming distance* 1*.*

The graph $H_i(d, q)$ *is defined on the same vertex set; however, vertices in this graph are adjacent if they have Hamming distance* i*. With this definition,* $H(d, q) = H_1(d, q)$*.*

Note that, unlike vector spaces of the form $\mathrm{GF}(q)^d$, the Hamming graph $H(d, q)$ is not restricted to prime powers for q. Therefore, we will frequently take $\mathbb{Z}_q^d$ to be the vertex set of the Hamming graph. In the case that q is a prime power, we may take $\mathrm{GF}(q)^d$ as the vertex set instead. While these algebraic structures are not isomorphic, their Hamming graphs are.

By generalizing finite vector spaces to Hamming graphs in this way, the explicit algebraic structure of the vector space is lost. However, the Hamming graphs have many combinatorial properties which will prove useful in this report. It will also turn out that the Hamming graphs have interesting algebraic properties; for example, they have large automorphism groups. More importantly however, they are *distance-regular graphs.*

Definition 1.2.3
Let Γ *be a graph, and* u, v *be vertices in the graph. Then the distance between* u *and* v *in* Γ*, denoted* $\mathrm{dist}\,(u, v)$*, is the length of a shortest path between* u *and* v*. Let* $N_i(u)$ *denote the set of vertices at distance* i *from* u *(if* u *is not incident to a loop, then* $N_1(u)$ *will be the neighbourhood of* u*).*

A graph is said to be DISTANCE-REGULAR *if for all distances* i*,* j*, and* k*, and any two vertices* u, v *at distance* k*, there is a constant* p_{ij}^k *such that*

$$|N_i(u) \cap N_j(v)| = p_{ij}^k \ .$$

In particular, at $u = v$ *and* $i = j = 1$*, if the graph is loopless, the above constraint implies that the graph is* REGULAR *– that the degree of each vertex is constant. This degree is also called the* VALENCY *of the graph.*

If Γ *is any graph, then the* DISTANCE GRAPHS *of* Γ *are the graphs* $\Gamma_0, \Gamma_1, \Gamma_2, \ldots$ *where each shares the same vertex set as* Γ*, and vertices are adjacent in* Γ_i *if and only if they are at distance* i *in* Γ*. Note that if* Γ *is loopless, then* $\Gamma = \Gamma_1$*. If* Γ_d *is the last non-empty graph in this sequence (i.e.* d *is the maximum distance in the graph), then* d *is called the* DIAMETER*.*

The Hamming graphs are distance-regular, where the graphs $H_i(d, q)$ are the distance graphs of $H(d, q)$, and d is the diameter. Other examples of distance-regular graphs include the Johnson graphs and the Grassmann graphs.

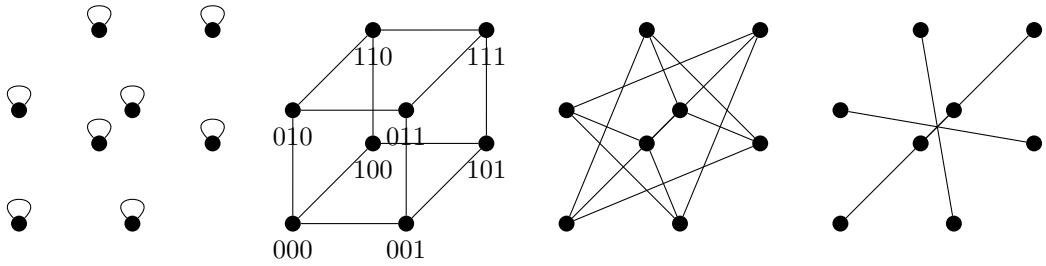

Figure 1.2: The distance graphs of $H(3, 2)$.

For any graph with distance graphs $\Gamma_0, \Gamma_1, \ldots, \Gamma_d$, we can define a *code* to be a subset of the vertex set, where the distance between vertices is their distance in the graph; that is, $\text{dist}(u, v) = i$ if and only if u and v are adjacent in Γ_i. Then the code has minimum distance δ if and only if δ is the least positive distance i in $\{1, \ldots, d\}$ for which there exists a pair of codewords which are adjacent in Γ_i. In part because the Hamming graphs belong to this class, we will study coding theory in the context of distance-regular graphs for the remainder of this report.

Note that the study of codes in graphs is also connected to the study of *cocliques* (or *independent sets*). A COCLIQUE in a graph is a set of vertices, none of which are adjacent. Equivalently, a coclique is a code in a graph with minimum distance 2. On the other hand, given distance graphs $\Gamma_0, \Gamma_1, \ldots, \Gamma_d$, an independent set in the graph whose edge set is the union of edge sets from $\Gamma_1, \ldots, \Gamma_\delta$ is a code of minimum distance δ.

More discussion on the use of finite fields (and vector spaces) in codes can be found in [Ple98]. The discussion of graphs comes from [God93, Chapter 11].

1.3 Hamming Graphs

This section will discuss some of the algebraic properties of the Hamming graphs. The fact that the Hamming graphs are distance-regular will be used without comment throughout the remainder of the report. The automorphism group of a Hamming graph is helpful in demonstrating this fact. The automorphism group also plays an important role in Schrijver's SDP bound (Section 4.3). Readers who are familiar with the automorphism groups of the Hamming graphs, or are willing to accept that these graphs are distance-regular, may omit this section.

The simplest way to show that the Hamming graphs are distance-regular is to show that they satisfy a stronger property, called *distance-transitivity*. To do this, it will be helpful to examine the automorphism group of the Hamming graph $H(d, q)$.

Throughout this section, the vertex set of $H(d,q)$ will be taken to be $\mathbb{Z}_q^d$, and e_i will denote the tuple of all zeroes, and a 1 in the i^{th} entry. Then, the neighbours of any vertex v are the vertices of the form $v + \alpha e_i$, where $\alpha \in \mathbb{Z}_q \setminus \{0\}$. We will denote by $[d]$ the set $\{1, \ldots, d\}$. Note also that group actions are written on the *right*, so that the result of applying action g to element u is denoted ug.

The most important property of the automorphism group of a Hamming graph is that the action of any automorphism is completely determined by its action on a small subset of the graph. In particular, given any vertex v, any automorphism is determined by its action on v and the neighbourhood of v.

Lemma 1.3.1
Fix any vertex v in $H(d,q)$. Then for each vertex u, the relation

$$D_v(u) := \{(v + \alpha e_i,\ \text{dist}\,(v + \alpha e_i, u)) \mid \alpha \in \mathbb{Z}_q,\ i \in [d]\}$$

completely characterizes the vertex u. (We will call this relation the distance profile *of u with respect to v.)*

Proof. Note that for any two vertices w and u, their distance is given by the Hamming weight of their difference: $\text{dist}\,(w,u) = |w - u|$.

Then, for any $\alpha \neq 0$, we know that

$$\text{dist}\,(u, v + \alpha e_i) = |u - (v + \alpha e_i)| = |(u - v) - \alpha e_i| \begin{cases} > |u - v| & \text{if} \quad u_i = v_i \\ < |u - v| & \text{if} \quad u_i - v_i = \alpha \\ = |u - v| & \text{else} \end{cases}$$

precisely; the only case when subtracting αe_i from $u - v$ lowers the weight of the vector is in the case that the i^{th} component of $u - v$ is α. Therefore, if we are given all the weights $\text{dist}\,(u, v + \alpha e - i)$, we can reconstruct u as follows:

$$u = v + \sum_{i \in [d]} e_i \sum_{\alpha \in \mathbb{Z}_q \setminus \{0\}} \alpha \cdot \mathbb{1}\left[\text{dist}\,(u, v + \alpha e_i) < \text{dist}\,(u, v)\right]$$

where $\mathbb{1}$ is the indicator function. It is defined on propositions such that

$$\mathbb{1}[\text{condition}] := \begin{cases} 1 & \text{if the condition is true} \\ 0 & \text{if the condition is false} \end{cases}.$$

Note that for each i, there will be *at most one value* of α for which $\text{dist}\,(u, v + \alpha e_i) <$

dist (u, v).

■

Corollary 1.3.2

For any vertex v, the action of any automorphism of a Hamming graph is completely determined by its action on $\{v + \alpha e_i \mid \alpha \in \mathbb{Z}_q,\ i \in [d]\}$.

Proof. Note that every graph automorphism preserves the distances in that graph.

Fix a vertex v and suppose that g and g' are automorphisms of $H(d, q)$ such that $(v + \alpha e_i)g = (v + \alpha e_i)g'$ for all α and i. Then for any vertex u, let $D_v(u) = \{(v + \alpha e_i,\ d_{\alpha,i})\}_{\alpha,i}$ be the distance profile of u with respect to v. Then since automorphisms preserve distances, for all α and i,

$$d_{\alpha,i} = \text{dist}\,((v + \alpha e_i)g, ug)$$
$$d_{\alpha,i} = \text{dist}\,((v + \alpha e_i)g', ug')$$

and $(v + \alpha e_i)g = (v + \alpha e_i)g'$. Therefore, the distance profiles of ug and ug' with respect to vg are identical, since $vg = vg'$, so that $ug = ug'$. ■

Now, to compute the automorphism group of $H(d, q)$, we will employ a strategy following the orbit-stabilizer theorem: we will show that there is a single orbit, so that the size of the group is the size of the vertex set times the size of stabilizer of a single vertex, say $\mathbf{0}$. Since the group is finite, by finding a set of automorphisms which map $\mathbf{0}$ to each other vertex, and the stabilizer of $\mathbf{0}$, taking the compositions of all pairs of automorphisms from these two subgroups will generate all automorphisms.

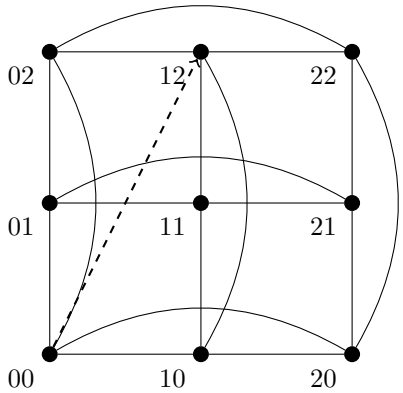

Figure 1.3: An shift-type automorphism of $H(2, 3)$ (see Lemma 1.3.3).

Lemma 1.3.3
For any vertex v, the map $T_v : u \mapsto u + v$ is an automorphism of $H(d, q)$. On the other hand, every automorphism of $H(d, q)$ is the composition of an automorphism of the form T_v for some vertex v, and an automorphism fixing $\mathbf{0}$.

Proof. Let v be a vertex of $H(d, q)$. Then T_v is invertible, since $T_v^{-1} = T_{-v}$. Furthermore, for any vertices u and w,

$$|uT_v - wT_v| = |(u + v) - (w + v)| = |u - w|$$

so T_v preserves adjacency.

Let g be any automorphism of $H(d, q)$, and let $\mathbf{0}g = v$. Then $\mathbf{0}gT_{-v} = \mathbf{0}$, so $g = g'T_v$, where g' is an automorphism fixing $\mathbf{0}$. ■

Note that this result tells us that the Hamming graph $H(d, q)$ is vertex transitive, since for all vertices u and u', $uT_{u'-u} = u + (u' - u) = u'$.

Now let us consider the stabilizer of $\mathbf{0}$. Since automorphisms preserve distances, if g fixes $\mathbf{0}$, then it will permute the neighbours αe_i of $\mathbf{0}$.

Lemma 1.3.4
Let g be an automorphism of $H(d, q)$ which fixes $\mathbf{0}$. Then for each α and e_i, $g : \alpha e_i \mapsto \beta e_j$ where α is zero if and only if β is. Moreover, if $g : \alpha e_i \mapsto \beta e_j$ for some α, then for all α' there exists a β' such that $g : \alpha' e_i \mapsto \beta' e_j$.

Proof. Note that if $g : \alpha e_i \mapsto \beta e_j$, then $\alpha = 0 \iff \beta = 0$ follows directly from the fact that g fixes $\mathbf{0}$. Then, if α is nonzero, αe_i will be a neighbour of $\mathbf{0}$, so its image under an automorphism must also be a neighbour of $\mathbf{0}$. Since every neighbour of $\mathbf{0}$ is of the form βe_j, the result follows.

Suppose then that $g : \alpha e_i \mapsto \beta e_j$ with $\alpha \neq 0$. If $\alpha' \neq 0$ and $\alpha' \neq \alpha$, then αe_i and $\alpha' e_i$ are neighbours, so $\alpha' e_i g$ must be adjacent to βe_j. By the previous paragraph, $\beta \neq 0$, so all neighbours of βe_j are of the form $\beta' e_j$. ■

Now let us consider some examples of automorphisms of $H(d, q)$ that fix $\mathbf{0}$. We will then show how all such automorphisms can be constructed from the examples we find.

Lemma 1.3.5
Let $\sigma \in \mathrm{Sym}[d]$, and let σ act on $H(d, q)$ by mapping $(x_1, \ldots, x_d)$ to $(x_{\sigma(1)}, \ldots, x_{\sigma(d)})$. Then σ is an automorphism of $H(d, q)$ which fixes $\mathbf{0}$.

Proof. Note that the action of σ on $H(d, q)$ is invertible, with inverse given by the action of σ^{-1} (as a permutation on $[d]$). Furthermore, note that σ does not alter the weight the vertices it acts on. Thus, $\mathbf{0}\sigma = \mathbf{0}$, and since, $v\sigma - u\sigma = (v - u)\sigma$, σ is an automorphism. ■

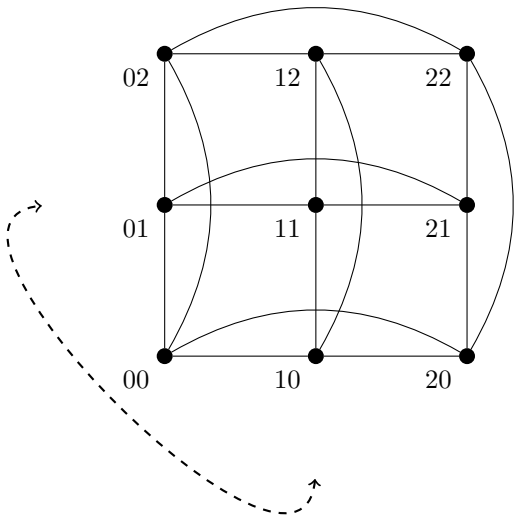

Figure 1.4: A component permutation of $H(2,3)$ (see Lemma 1.3.5).

Lemma 1.3.6
Let $\tau \in \mathrm{Sym}\,(\mathbb{Z}_q \setminus \{0\})$; then τ extends naturally to a permutation on $\mathbb{Z}_q$ by fixing 0. Fixing any $i \in [d]$, τ acts on $H(d,q)$ by mapping $(x_1,\ldots,x_i,\ldots,x_d)$ to $(x_1,\ldots,\tau(x_i),\ldots,x_d)$. For each τ and i, this action is an automorphism on $H(d,q)$ fixing $\mathbf{0}$.

Proof. Note that since the action of τ on $\mathbb{Z}_q$ fixes 0, the action of τ on $H(d,q)$ fixes $\mathbf{0}$. Furthermore, if τ acts on the i^{th} component of vertices in the graph, then this action of τ is invertible by applying the inverse of τ in $\mathrm{Sym}\,\mathbb{Z}_q$ to the i^{th} component.

Suppose that $u \sim v$ are vertices: then $u - v = \alpha e_j$ for some $\alpha \neq 0$ and $j \in [d]$. For $j' \neq j$, $u_{j'} = v_{j'}$ which implies that $\tau(u_{j'}) = \tau(v_{j'}) \implies (u\tau - v\tau)_{j'} = 0$. On the other hand, since τ is a permutation, $u_j \neq v_j \implies \tau(u_j) \neq \tau(v_j)$ so that $(u\tau - v\tau)_j \neq 0$. ■

Theorem 1.3.7
Every automorphism of $H(d,q)$ is of the form $\tau\sigma T_v$ where $\tau = \tau_1 \cdots \tau_d$, and each $\tau_i \in \mathrm{Sym}\,(\mathbb{Z}_q \setminus \{0\})$; where $\sigma \in \mathrm{Sym}[d]$; and where v is any vertex.

Proof. Let g be any automorphism of $H(d,q)$, and suppose $\mathbf{0}g = v$. Then by Lemma 1.3.3 $g' := gT_{-v}$ is an automorphism which fixes $\mathbf{0}$. Then by Lemma 1.3.4 for all i there exists a j such that for all nonzero α, $g' : \alpha e_i \mapsto \beta e_j$ for some nonzero β. Therefore, let σ^{-1} be the permutation mapping each j back to i, so that $g'' := g'\sigma^{-1}$ maps αe_i to βe_i (Lemma 1.3.5). Finally, for each i and nonzero α, by Lemma 1.3.4 β is nonzero, so define τ_i to be the

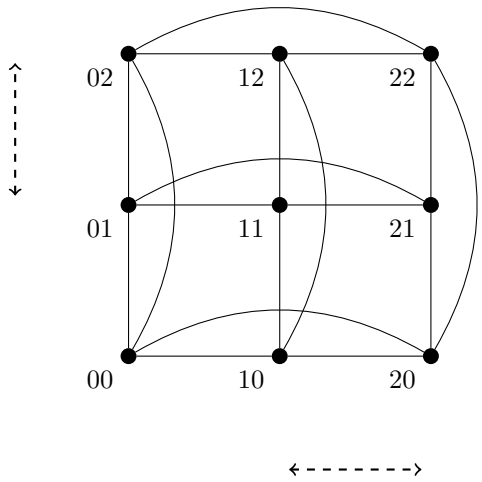

Figure 1.5: An automorphism permuting the nonzero elements of each component of $H(2,3)$ (see Lemma 1.3.6).

permutation on $\mathbb{Z}_q$ mapping α to β, and fixing 0 (Lemma 1.3.6). Setting $\tau := \tau_1 \cdots \tau_d$, we see that $g''\tau^{-1} : \alpha e_i \mapsto \alpha e_i$ for all i and all α. Since automorphisms of $H(d,q)$ preserve distances in the graph and are completely determined by their action on $\{\alpha e_i \mid \alpha \in \mathbb{Z}_q,\ i \in [d]\}$ (see Corollary 1.3.2), $gT_v^{-1}\sigma^{-1}\tau^{-1}$ is the identity on $H(d,q)$. ■

Corollary 1.3.8
There are exactly $q!^d d!$ automorphisms of $H(d,q)$.

Proof. It is straightforward to see that for each distinct tuple of vertex v and permutations $\sigma \in \mathrm{Sym}[d]$ and $\tau_1, \ldots, \tau_d \in \mathrm{Sym}\,(\mathbb{Z}_q \setminus \{0\})$ there is a distinct automorphism $\tau_1 \cdots \tau_d \sigma T_v$. There are q^d vertices v, $d!$ permutations σ, and $(q-1)!^d$ permutations $\tau = \tau_1 \cdots \tau_d$. ■

Definition 1.3.9 ([God93, Section 11.1])
A graph Γ is DISTANCE-TRANSITIVE *if for all pairs of vertices u, v and u', v' such that $\mathrm{dist}\,(u,v) = \mathrm{dist}\,(u',v')$, there is an automorphism of Γ mapping u to u' and v to v'.*

Recall that $N_k(v)$ denotes the set of vertices at distance k from v. The following result follows almost directly from the characterization of the automorphism group of $H(d,q)$.

Lemma 1.3.10
The stabilizer of $\mathbf{0}$ acts transitively on $N_k(\mathbf{0})$. That is, for any two vertices $u, u' \in N_k(\mathbf{0})$, there is an automorphism fixing $\mathbf{0}$ which maps u to u'.

Proof. If $u \in N_k(\mathbf{0})$, then there exist coefficients $\alpha_i \in \mathbb{Z}_q$ – exactly k of which are nonzero

– such that $u = \sum_i \alpha_i e_i$. Likewise, since $u' \in N_k(\mathbf{0})$, there exists exactly k nonzero coefficients $\alpha'_i \in \mathbb{Z}_q$ such that $u' = \sum_i \alpha'_i e_i$. Note that each permutation $\sigma \in \mathrm{Sym}[d]$ gives an automorphism fixing $\mathbf{0}$; so we may choose two permutations to map u and u' such that the first k values of α_i and α'_i are nonzero. Therefore, assume without loss of generality that

$$u = \sum_{i=0}^{k} \alpha_i e_i \qquad \text{and} \qquad u' = \sum_{i=0}^{k} \alpha'_i e_i \ .$$

Then, for all i in $[k]$, $\alpha_i \neq 0$ and $\alpha'_i \neq 0$, so there exist permutations τ_i in $\mathrm{Sym}\left(\mathbb{Z}_q \setminus \{0\}\right)$ such that $\tau := \tau_1 \cdots \tau_k$ maps each α_i to α'_i. ■

Corollary 1.3.11
The Hamming graph $H(d, q)$ is distance-transitive.

Proof. Let u, u', v, v' be vertices such that $\mathrm{dist}\,(u, v) = \mathrm{dist}\,(u', v')$. Since automorphisms preserve distance, and the shift automorphisms $T_{-u'}$ and T_{-u} map u' and u to $\mathbf{0}$, we may assume without loss of generality that $u = u' = \mathbf{0}$. Then, for $k := \mathrm{dist}\,(\mathbf{0}, v) = \mathrm{dist}\,(\mathbf{0}, v')$, we find that v and v' are in $N_k(\mathbf{0})$. Since this set is transitive under the action of the stabilizer of $\mathbf{0}$, there must exist an automorphism fixing $\mathbf{0}$ and mapping v to v'. ■

Lemma 1.3.12
Every distance-transitive graph is distance-regular.

Proof. Let i, j, and k be non-negative integers, and let u, u', v, v' be vertices such that $k = \mathrm{dist}\,(u, v) = \mathrm{dist}\,(u', v')$. It suffices to show that the number of vertices w satisfying $\mathrm{dist}\,(u, w) = i$ and $\mathrm{dist}\,(w, v) = j$ equals the number of vertices w' satisfying $\mathrm{dist}\,(u', w') = i$ and $\mathrm{dist}\,(w', v') = j$. Since the graph in question is distance-transitive, there exists an automorphism mapping u to u' and v to v'. Then, since automorphisms preserve distances, each w in $N_i(u) \cap N_j(v)$ will get mapped to an element w' in $N_i(u') \cap N_j(v')$. Since automorphisms are invertible, this provides the required bijection. ■

Corollary 1.3.13
The Hamming graphs are distance-regular.

By considering the automorphisms of the Hamming graphs $H(d, q)$, we noticed that they satisfying the property of *distance-transitivity*. In turn, all distance-transitive graphs satisfy a regularity condition in the numbers of vertices at certain distances from other vertices. It is interesting to note that not all distance-regular graphs have large automorphism groups like the Hamming graphs. While this numerical regularity may not seem particularly important, by replacing the notion of *distance graphs* with any list of graphs, we arrive (roughly) at the definition of an association scheme. It will turn out that these objects have many

incredible properties, some of which will help us answer our original question: what can we say about the size of a code in a graph? While this numerical regularity will be sufficient for our answer in general, it will be computationally easier when the association schemes have "nice" automorphism groups. In particular, the automorphism group of the Hamming graph will help us answer our original question with greater accuracy.

The fact that the Hamming graphs are distance-transitive (and thus distance-regular) is a well-known fact (seen in [God93, Section 11.1] for example). The automorphism group of the Hamming graphs is also well-known, and given for example in [MZ19]. To the best of the author's knowledge, the proofs given here are unique. The construction of the automorphism group of $H(d, q)$ follows the outline of [Win17].

2. Association Schemes

2.1 Association Schemes

Definition 2.1.1 (Commutative Association Scheme – Combinatorial [Del73, Section 2.1]) *Let $D = \{0, 1, \ldots, d\}$ for some $d \geq 1$. A* COMMUTATIVE ASSOCIATION SCHEME *$\mathcal{A}$ is a set V, called the* VERTEX SET*, together with a set of relations $\{\Gamma_i\}_{i=0}^{d}$ satisfying the following axioms:*

1. *The set of relations $\{\Gamma_i\}_{i=0}^{d}$ partitions $V \times V$;*
2. *Γ_0 is the diagonal relation $\{(u, u) \mid u \in V\}$;*
3. *For each $i \in D$, there is an $i' \in D$ such that $\Gamma_{i'}$ is the opposite relation $\{(v, u) \mid (u, v) \in \Gamma_i\}$ of Γ_i;*
4. *For every triple $i, j, k \in D$, there exists a constant $p_{i,j}^k$ such that for all $(u, v) \in \Gamma_k$, there are exactly $p_{i,j}^k$ vertices w such that $(u, w) \in \Gamma_i$ and $(w, v) \in \Gamma_j$; furthermore, $p_{i,j}^k = p_{j,i}^k$.*

As used above, the elements of V are called VERTICES*, and vertices $(u, v) \in \Gamma_i$ are called* i^{TH} ASSOCIATES.

Each relation Γ_i is called a CLASS *of $\mathcal{A}$, which has* DIAMETER *d (this will be explained in connection with distance-regular graphs in the next section (2.2)). Sometimes, $\mathcal{A}$ is said to be a d-class association scheme (as the diagonal relation is discounted).*

To every relation $\Gamma \subseteq V \times V$ there exists a $V \times V$ 01-matrix A, where

$$A_{uv} = \begin{cases} 1 & \text{if} \quad (u, v) \in \Gamma \\ 0 & \text{if} \quad (u, v) \notin \Gamma \end{cases}; \tag{2.1}$$

A is then called the ADJACENCY MATRIX of Γ. This is a bijective correspondence between relations on $V \times V$, and $V \times V$ matrices with each entry either 0 or 1. (Note that binary relations are precisely directed graphs without parallel edges, and this usage of the term *adjacency matrix* agrees with its usage in graph theory.)

By exchanging the relations Γ_i for their adjacency matrices A_i, the combinatorial definition of an association scheme can be reformulated as follows:

Definition 2.1.2 (Commutative Association Scheme – Algebraic [God93, Chapter 12]) *Let $\circ$ denote the* SCHUR *product of matrices of the same shape:*

$$(A \circ B)_{uv} = A_{uv} B_{uv} \ . \tag{2.2}$$

(This is also called the Hadamard, or entrywise product.)

A COMMUTATIVE ASSOCIATION SCHEME *is a vertex set V along with $V \times V$ matrices $A_0, A_1, \ldots A_d$ such that*

1. $\sum_{i=0}^{d} A_i = J$, *the all-ones matrix, and* $A_i \circ A_j = \delta_{i,j} A_i$;
2. $A_0 = I$, *the identity matrix;*
3. *For every $i \in D$ there is an $i' \in D$ such that $A_i^T = A_{i'}$;*
4. *For every i, j, k, there exists a constant $p_{i,j}^k$ such that*

$$A_i A_j = A_j A_i = \sum_{k=0}^{d} p_{i,j}^k A_k \ .$$

It is clear from the first axiom that the A_i are 01-matrices.

The first three equivalences are straightforward translations. To see the last, observe that

$$(A_i A_j)_{uv} = \sum_{z \in V} (A_i)_{uw} (A_j)_{wv}$$

which counts the number of vertices $z \in V$ such that

$$\begin{cases} (A_i)_{uw} = 1 & \iff (u,w) \in \Gamma_i \\ (A_j)_{wv} = 1 & \iff (w,v) \in \Gamma_j \end{cases} .$$

Then, $(A_i A_j)_{uv} = p_{ij}^k$ exactly when $(A_k)_{uv} = 1 \iff (u,v) \in \Gamma_k$.

The requirement in (Combinatorial 4) that $p_{i,j}^k = p_{j,i}^k$ corresponds to the requirement that $A_i A_j = A_j A_i$ (Algebraic 4), which is why such association schemes are called *commutative*.

If the requirement of *commutativity* is dropped, then every finite group G is an association scheme in the following way. Cayley's theorem says that every group is isomorphic to

the group of permutations on G given by left multiplication (the same construction will work if right multiplication is used everywhere instead of left). By identifying each element of G with the $G \times G$ *permutation matrix*, we obtain an association scheme with G as the vertex set, and permutation matrices as the classes. Since the identity element of G will map to the identity matrix, the product of two such permutations is another such permutation, and the transpose of a permutation matrix is its inverse, axioms 2, 3, 4 are satisfied. Since for every $g, h \in G$, only the element hg^{-1} maps g to h, exactly one of the permutation matrices will have a 1 in the (g, h)-entry; all the others will be 0. This association scheme will be commutative if and only if the group G is.

Not every association scheme (commutative or not) arises in this way, but many schemes of interest are closely related to a particular group. In any case, much of the utility of association schemes, and the elegance of their theory, derives from the connection between the combinatorial view of schemes as relations or (di)graphs (Definition 2.1.1), and the algebraic view of schemes as matrices (Definition 2.1.2).

A particularly important class of association scheme, called SYMMETRIC, is one in which every relation is symmetric (i.e. $i = i'$ in Combinatorial 3), or equivalently, each adjacency matrix is symmetric ($A_i^T = A_i$ in Algebraic 3). In this case, the relations Γ_i form *undirected* graphs Γ_i with vertex set V, and edge set given by

$$u \sim v \iff (u, v) \in \Gamma_i \iff (v, u) \in \Gamma_i \ .$$

The most important class of symmetric association scheme (for the purposes of this report) arise as the distance graphs of a distance-regular graph (see Section 2.2).

2.1.1 The Bose-Mesner Algebra

For any association scheme $\mathcal{A}$, there are the adjacency matrices $A_0, A_1, \ldots, A_d$ (Definition 2.1.2). Since any product of these matrices belongs to their span (axiom 4), their span

$$\mathbb{A} := \operatorname{span}\{A_0, A_1, \ldots, A_d\} \tag{2.3}$$

is closed under matrix multiplication. This structure is called the BOSE-MESNER ALGEBRA of $\mathcal{A}$.

Note also that from the first axiom of an association scheme (1), $\mathbb{A}$ is also closed under the *Schur product*, so that $\mathbb{A}$ is actually an algebra with respect to two *different* products. This duality will form an important aspect of the theory of association schemes, and will be discussed in detail in this section and the next (2.1.2).

Since the adjoint $A_i^* = A_i^T$ belongs to $A_0, A_1, \ldots, A_d$ for every i, and these matrices all commute, each A_i is a normal operator (see Theorem A.1.4). From the spectral theorem,

we can decompose each matrix

$$A_i = \sum_j \theta_{ij} \widetilde{F_{ij}}$$

into a linear combination of orthogonal idempotents $\widetilde{F_{ij}}$ (with i fixed), where

$$I = \sum_j \widetilde{F_{ij}} \ .$$

Since each $\widetilde{F_{ij}}$ is a polynomial in A_i, and the A_i all commute, so too do the $\widetilde{F_{ij}}$ (see Proposition A.1.5). Therefore, the products $\prod_i \widetilde{F_{ik_i}}$ (for any choices of k_i) are also orthogonal idempotents, though some may be zero, so that the nonzero products are linearly independent. Furthermore, their sum is the identity matrix

$$I = I^{d+1} = \prod_i \left(\sum_j \widetilde{F_{ij}} \right) = \sum \left(\prod_i \widetilde{F_{ik_i}} \right)$$

where the latter sum is taken over all choices of k_i. Therefore, we can express each A_i as linear combinations of these products, as

$$\begin{aligned} A_i \prod_{i'} \widetilde{F_{ik_i}} &= A_i \widetilde{F_{ik_i}} \prod_{i' \neq i} \widetilde{F_{ik_i}} \\ &= \theta_{ik_i} \widetilde{F_{ik_i}} \prod_{i' \neq i} \widetilde{F_{ik_i}} \\ &= \theta_{ik_i} \prod_{i'} \widetilde{F_{ik_i}} \\ \implies A_i = A_i I &= \sum \theta_{ik_i} \prod_{i'} \widetilde{F_{ik_i}} \ . \end{aligned}$$

Therefore, the nonzero products $\prod_i \widetilde{F_{ik_i}}$ span $\mathbb{A}$ and are linearly independent, so there are precisely $d+1 = \dim \mathbb{A}$ of them: $F_0, F_1, \ldots, F_d$. They are orthogonal idempotents, and each is self-adjoint since the $\widetilde{F_{ij}}$ are self-adjoint and commute. These matrices are called the PRINCIPAL IDEMPOTENTS of $\mathcal{A}$, and form an alternative basis to $A_0, A_1, \ldots, A_d$. (Because the A_i are orthogonal idempotents under the Schur product, they are sometimes called SCHUR IDEMPOTENTS of $\mathcal{A}$.) Then, one can write

$$A_i = \sum_{j=0}^{d} P_i(j) F_j$$

and form the $(d+1) \times (d+1)$ matrix P by $P_{ji} = P_i(j)$ (note the reversed indices). This is called the MATRIX OF EIGENVALUES of $\mathcal{A}$. [God93, Theorem 12.2.1]

In the case that $\mathcal{A}$ is symmetric, each A_i is a real, symmetric operator, so its eigenvalues θ_{ij} are real, as are its idempotents $\widetilde{F_{ij}}$. Therefore, the idempotents F_j are real, as is the matrix of eigenvalues P.

This subsection follows [God93, Section 12.2] closely.

2.1.2 Duality and Characterization

Already a comparison between the two bases $A_0, A_1, \ldots, A_d$ and $F_0, F_1, \ldots, F_d$ of $\mathbb{A}$ suggests a duality. For example, the A_i are orthogonal idempotents with respect to the Schur product, while the F_j are orthogonal idempotents with respect to the usual product; $\sum_i A_i = J$, the identity of the Schur product, while $\sum_j F_j = I$, the identity of the usual product. In this section, additional "dual" properties will be discovered by examining relations with respect to one basis, and searching for analogous ones in the opposite basis.

Many of these properties will involve two matrices to be introduced, which describe how each basis is transformed into the other. Some will be vital to the later analysis of cliques in association schemes. Others will be used to show that either of these matrices, taken in isolation, completely characterize the scheme!

First, the axioms of an association scheme (Definition 2.1.2) require that $A_0 = I$, the identity of the usual product. As already illustrated, J the all-ones matrix acts as the "dual" of I, so let us examine:

$$JF_j = \left(\sum_i A_i\right) F_j = \left(\sum_i P_i(j)\right) F_j \ .$$

Letting $\gamma_j := \sum_i P_i(j)$, $JF_j = \gamma_j F_j$ demonstrates that γ_j is an eigenvalue of J, with eigenvectors in the column space of F_j. Since J is orthogonally diagonalizable, with eigenvalues $n = |V|$ of multiplicity 1 and 0 of multiplicity $n-1$, and the F_j are orthogonal idempotents whose columns are eigenvectors of J, there is exactly one j with $\gamma_j = n$. Since $JF_j = nF_j$, each row of F_j is equal; but since F_j is self-adjoint the columns are all equal as well. Therefore, $F_j = \alpha J$, and F_j is idempotent, so that

$$F_j = \alpha J = \alpha^2 J^2 = \alpha^2 nJ \implies \alpha = \frac{1}{n} \ .$$

Since the ordering $F_0, F_1, \ldots, F_d$ is arbitrary, by convention an order is taken such that $F_0 = \frac{1}{n}J$.

Since the F_j are orthogonal, $\gamma_j = 0$ for all $j > 0$, so that the row sums of P are known:

$$\sum_{i=0}^{d} P_i(j) = \begin{cases} n & \text{if} \quad j = 0 \\ 0 & \text{if} \quad j > 0 \end{cases} . \tag{2.4}$$

Moreover, since $F_0 = \frac{1}{n}J$, the all-ones vector $\mathbf{1}$ is an eigenvector of each A_i: let ν_i be the eigenvalue, so that $A_i\mathbf{1} = \nu_i\mathbf{1}$. Note that ν_i is the (constant) row sum of A_i. Since $A_i^T = A_{i'}$, the column sums of A_i are constant as well, and if the scheme is symmetric, $A_i^T = A_i$, so the column sums are equal to ν_i as well. The quantity ν_i is called the VALENCY of A_i. (This usage agrees with its usage for regular graphs, where it equals the (constant) degree of vertices. This is also equal to the row sum of the adjacency matrix.)

Next, since $F_0, F_1, \ldots, F_d$ is a basis for $\mathbb{A}$, we can write

$$A_i = \sum_j P_i(j) F_j \tag{2.5}$$

uniquely, and since the F_j are orthogonal idempotents, $A_i F_j = P_i(j) F_j$. Dually, $A_0, A_1, \ldots, A_d$ is a basis for $\mathbb{A}$, so we can write

$$F_j = \frac{1}{n} \sum_i Q_j(i) A_i \tag{2.6}$$

uniquely, and since the A_i are Schur orthogonal idempotents, $F_j \circ A_i = \frac{1}{n} Q_i(j) A_i$. These coefficients form the MATRIX OF DUAL EIGENVALUES Q, defined such that $Q_{ij} = Q_j(i)$ (note the reversed indices).

By substituting Equation 2.6 into Equation 2.5, we obtain

$$nA_i = \sum_j P_i(j) \sum_{i'} Q_j(i') A_{i'} = \sum_{i'} \left(\sum_j P_i(j) Q_j(i') \right) A_{i'} .$$

Because the A_i form a basis, nA_i is written uniquely, so that

$$\sum_j P_i(j) Q_j(i') = n\delta_{ii'} \implies PQ = nI .$$

Since the first row sum of P is n (by Equation 2.4), if we are given P and *no other information*, then by computing its inverse we can derive Q.

If we were given only Q instead, we could calculate P by taking the inverse and multiplying by n. All that remains, then, is to derive n from Q. In the other direction, this was done by examining the row sums of P, so we do the same with Q. The rows of Q correspond

to the A_i, the columns to the F_j, and $F_j \circ A_i = \frac{1}{n} Q_j(i) A_i$, so we fix a row and sum over the columns:

$$\left(\sum_j F_j \right) \circ A_i = \frac{1}{n} \left(\sum_j Q_j(i) \right) A_i \ .$$

But $\sum_j F_j = I = A_0$ and $A_0 A_i = \delta_{0i} A_i$ so

$$\sum_{j=0}^{d} Q_j(i) = \begin{cases} n & \text{if} \quad i = 0 \\ 0 & \text{if} \quad i > 0 \end{cases} \ . \tag{2.7}$$

Therefore, given either P or Q, and no other information, the other can be derived. However, this process involves matrix inversion, which is both expensive to compute and awkward to contemplate. In fact, if a little extra information is known, then there exists an extraordinarily simple formula relating the two matrices. This extra information consists of the valencies and their dual concept, the multiplicities, of the scheme.

We have already encountered the valency ν_i of A_i. Since by definition $\nu_i = P_i(0)$, consider the first row of Q and denote $Q_j(0)$ by μ_j. From the definition of Q we have $F_j \circ A_0 = \frac{1}{n} \mu_j A_0$, but A_0 is the identity matrix, so that the diagonal entries of F_j are identically μ_j / n. Moreover, F_j is an $n \times n$ matrix, so $\operatorname{trace} F_j = \mu_j$, and since F_j is idempotent, $\operatorname{rank} F_j = \operatorname{trace} F_j$ (see Proposition A.1.2). If we fix i and suppose momentarily that the $P_i(j)$ are all distinct, then the column space of F_j is the $P_i(j)$-eigenspace of A_i, in which case $\mu_j = \dim \operatorname{col} F_j$, which is the algebraic multiplicity. For this reason (even if the $P_i(j)$ are not distinct), μ_j will be called the MULTIPLICITY of F_j.

Consider the different products between matrices in one basis and matrices in the other: $A_i F_j$ and $A_i \circ F_j$. Since F_j is self-adjoint, and commutes with A_i, $A_i F_j = F_j^* A_i$. Writing the product in this way, we might notice that the trace of the regular product is actually the (standard) inner product of A_i and F_j, as is the sum of all entries in the Schur product.

$$\langle A_i \ , \ F_j \rangle = \operatorname{trace} F_j^* A_i = \operatorname{sum} A_i \circ F_j$$

On one hand,

$$\operatorname{trace} A_i F_j = P_i(j) \operatorname{trace} F_j = P_i(j) \mu_j \ ,$$

while on the other hand, the n rows of A_i all sum to ν_i, so

$$\operatorname{sum} A_i \circ F_j = \frac{1}{n} Q_j(i) \operatorname{sum} A_i = Q_j(i) \nu_i \ .$$

Of course these two expressions must be equal, so letting ν denote the vector of valencies,

and μ the vector of multiplicities, we see that

$$P^T \operatorname{diag}(\mu) = \operatorname{diag}(\nu) Q \ . \tag{2.8}$$

Therefore, given the valencies and multiplicities, Q can be computed inexpensively from P and vice versa. Furthermore, this explicit expression will be useful in later chapters when contemplating expressions involving Q and vice versa.

Briefly, note that in the definition of an association scheme, it was required that $A_0 = I$, but it is only by a conventional order that $F_0 = \frac{1}{n}J$. If instead we only required that $A_i = I$ for some i, then ν can still be identified within the permuted matrix of eigenvalues as the only row with nonzero sum. Similarly, μ can be identified within Q as the only row with nonzero sum, even if we do not adhere to our convention.

For the remainder of this report, the above properties of the Bose-Mesner algebra and its duality will suffice. However, with just a little extra work, we show that $\mathbb{A}$ is actually characterized by its matrix of eigenvalues (or its matrix of dual eigenvalues).

The Bose-Mesner algebra has three kinds of structures on it: it is a vector space over $\mathbb{C}$, it is a ring with respect to the usual matrix product, and it is a ring with respect to the Schur product. In order to characterize $\mathbb{A}$, we will need to define all three structures from P alone.

First, P is a $(d+1) \times (d+1)$ matrix when $\dim \mathbb{A} = d+1$. Since vector spaces are characterized complete by their field and their dimension, the shape of P suffices here.

Next, since we have a basis $F_0, F_1, \ldots, F_d$ in which the usual matrix product is "known" (that is, they are orthogonal idempotents), this structure also will be characterized (vacuously) by P.

Finally, we must characterize the Schur product in $\mathbb{A}$. We also have a basis of orthogonal idempotents with respect to this product – namely $A_0, A_1, \ldots, A_d$ – but since this is a different basis from the one used previously, for the two products to be characterized simultaneously, we must be able to perform a change of basis. However, $A_i = \sum_j P_i(j) F_j$ and $F_j = \sum_i Q_j(i) A_i$, where Q can be derived from P.

In other words, all the information needed to compute the isomorphism between $\mathbb{A}$ and $\mathbb{C}^{d+1}$ is encapsulated in P. By picking a basis $f_0, f_1, \ldots, f_d$ in $\mathbb{C}^{d+1}$, the regular product can be defined by $f_i f_j := \delta_{ij} f_i$, and extended bilinearly. Then, the Schur product can be defined by rewriting any element of $\mathbb{C}^{d+1}$ in the basis $a_i = \sum_j P_i(j) f_j$ computing $a_i \circ a_j = \delta_{ij} a_i$ and extending bilinearly.

Because P can be derived from Q, the Bose-Mesner algebra of $\mathcal{A}$ is also characterized by Q. Moreover, if $\mathcal{A}$ is symmetric, then $\mathbb{C}$ can be replaced by $\mathbb{R}$ throughout the previous discussion.

However, more can be said about the characterization of the two products on $\mathbb{A}$. In particular, the usual product can be defined in terms of the basis of Schur idempotents where

$$A_i A_j = \sum_{k=0}^{d} p_{ij}^k A_k \ .$$

These p_{ij}^k are called the INTERSECTION NUMBERS of $\mathcal{A}$. Similarly, we can define numbers q_{ij}^k, called the KREIN PARAMETERS, such that

$$F_i \circ F_j = \frac{1}{n} \sum_{k=0}^{d} q_{ij}^k F_k \ .$$

These dual families of parameters can also be derived from P or Q in the following manner.

Since the A_i are Schur orthogonal idempotents, we can pick out the p_{ij}^k by multiplying $(A_i A_j) \circ A_k = p_{ij}^k A_k$. Summing over all entries of this product, $\operatorname{sum} p_{ij}^k A_k = p_{ij}^k n\nu_k$. However, since the sum of entries in a Schur product is the inner product of the factors,

$$\operatorname{trace} A_k^* A_i A_j = \operatorname{trace} \left(\sum_{r=0}^{d} P_i(r) P_j(r) \overline{P_k(r)} F_r \right) = \sum_{r=0}^{d} P_i(r) P_j(r) \overline{P_k(r)} \mu_r \ ,$$

where $\overline{P_k(r)}$ is the complex conjugate of $P_k(r)$. (This is done by rewriting each of A_i, A_j, A_k in terms of the F_r and expanding the product.) In the case that $\mathcal{A}$ is symmetric P is real, so that $\overline{P_k(r)} = P_k(r)$. Then, it is clear that p_{ij}^k remains unchanged when the indices i, j, k are permuted.

Similarly, $(F_i \circ F_j) F_k = q_{ij}^k F_k$, and F_k is self-adjoint, so taking the trace, $\operatorname{trace} q_{ij}^k F_k = q_{ij}^k \mu_k$. However, since the trace of a product is the inner product of the factors,

$$\operatorname{sum} F_k \circ F_i \circ F_j = \operatorname{sum} \left(\frac{1}{n^3} \sum_{r=0}^{d} Q_i(r) Q_j(r) Q_k(r) A_r \right) = \frac{1}{n^2} \sum_{r=0}^{d} Q_i(r) Q_j(r) Q_k(r) \nu_r \ .$$

As with the intersection numbers, q_{ij}^k remains unchanged when the indices i, j, k are permuted.

Conversely, the eigenmatrices can be derived from the intersection numbers. For more on this, see [Del73, Section 2.3].

Proposition 2.1.3
The first columns of P and Q are the all-ones vector, $\mathbf{1}$.

Proof. The first column of P holds the eigenvalues of A_0, which by definition is equal to I. Therefore, $A_0 F_j = P_0(j) F_j = F_j$, so $P_0(j) = 1$.

The first column of Q corresponds to $F_0 = \frac{1}{n} J$. Therefore, $F_0 \circ A_i = \frac{1}{n} Q_0(i) A_i$, so $Q_0(j) = 1$. ■

Note that because $P^T \operatorname{diag}(\mu) = \operatorname{diag}(\nu) Q$, the each of the two results above could be derived from the other.

Thus far, almost purely algebraic techniques have been used to derive results. However, the following results come from the algebraic theory of graphs. They are immediate consequences of the Perron-Frobenius theorem for the adjacency matrices of regular graphs.

Proposition 2.1.4
For all $i, j \in D$, we have $|P_i(j)| \leq \nu_i$.

Proof. We know that for each i, we have $A_i \mathbf{1} = \nu_i \mathbf{1}$. Let x be a θ-eigenvector of A_i, scaled so that its largest component (say, x_u) has modulus 1; let θ have largest modulus among the eigenvalues of A_i. Then $|(A_i x)_u| = |(\theta x)_u| = |\theta|$, and

$$|\theta| = |(A_i x)_u| \leq \sum_v (A_i)_{vu} |x_v| \leq \sum_v (A_i)_{vu} |x_u| = (A_i \mathbf{1})_u = \nu_i \ .$$

The proposition follows, as each $P_i(j)$ is an eigenvalue of A_i. ■

Note that $A_i \mathbf{1}$ is the vector of row sums of A_i; the sum of each row is precisely the out-degrees of the relation (directed graph) of A_i. Therefore, this proposition shows that the eigenvalues of a directed graph all have modulus less than the maximum out-degree. In the case of a regular graph, this is simply the valency.

Corollary 2.1.5
For all $i, j \in D$, we have $|Q_j(i)| \leq \mu_j$.

Proof. This follows from the previous proposition, as $Q_j(i) = \frac{\mu_j}{\nu_i} P_i(j)$, and both the valencies and multiplicities are all positive integers. ■

This completes the list of standard properties of the Bose-Mesner algebra and its duality. The results are summarized in the table on the next page.

This subsection follows [God93, Section 12.2] closely.

Table 2.1: Dual Properties of the Bose-Mesner Algebra

2.1.1	$I = A_0$	$\frac{1}{n}J = F_0$
2.1.2	$J = \sum_{i=0}^{d} A_i$	$I = \sum_{j=0}^{d} F_j$
2.1.3	$A_i \circ A_j = \delta_{ij} A_i$	$F_i F_j = \delta_{ij} F_j$
2.1.4	$A_i = \sum_{j=0}^{d} P_i(j) F_j$	$F_j = \frac{1}{n} \sum_{i=0}^{d} Q_j(i) A_i$
2.1.5	$A_i F_j = P_i(j) F_j$	$F_j \circ A_i = \frac{1}{n} Q_j(i) A_i$
2.1.6	$P(0) = \nu^T$	$Q(0) = \mu^T$
2.1.7	$P_0 = \mathbf{1}$	$Q_0 = \mathbf{1}$
2.1.8	$\sum_{i=0}^{d} P_i(j) = n\delta_{0j}$	$\sum_{j=0}^{d} Q_j(i) = n\delta_{0i}$
2.1.9	$\lvert P_i(j) \rvert \leq \nu_i$	$\lvert Q_j(i) \rvert \leq \mu_j$
2.1.10	$P \operatorname{diag}(\nu)^{-1} P^T = n \operatorname{diag}(\mu)^{-1}$	$Q \operatorname{diag}(\mu)^{-1} Q^T = n \operatorname{diag}(\nu)^{-1}$
2.1.11	$PQ = nI$	
2.1.12	$P^T \operatorname{diag}(\mu) = \operatorname{diag}(\nu) Q$	
2.1.13	$A_i A_j = \sum_{k=0}^{d} p_{ij}^k A_k$	$F_i \circ F_j = \frac{1}{n} \sum_{k=0}^{d} q_{ij}^k F_k$
2.1.14	$p_{ij}^k = \frac{1}{n\nu_k} \sum_{r=0}^{d} P_i(r) P_j(r) \overline{P_k(r)} \mu_r$	$q_{ij}^k = \frac{1}{n^2 \mu_k} \sum_{r=0}^{d} Q_i(r) Q_j(r) Q_k(r) \nu_r$

2.2 P-Polynomial Schemes

Our motivation in this report for the definition of an association scheme was the class of distance-regular graphs. However, the class of association schemes – even symmetric ones – is much more general. In this section, we will investigate a restricted class of symmetric association schemes: the P-polynomial schemes.

Note that in this section, and for the remainder of this report, all association schemes will be symmetric unless otherwise specified.

Definition 2.2.1 ([God93, Section 12.3])
An association scheme of adjacency matrices $A_0, A_1, \ldots, A_d$ is P-POLYNOMIAL *if they can be ordered such that A_i is a polynomial in A_1 of degree i.*

It happens that every distance-regular graph forms a P-polynomial scheme, and that every P-polynomial scheme arises in this way.

To demonstrate that distance-regular graphs generate P-polynomial schemes, we show that each distance matrix A_i is a degree-i polynomial in A_1 by induction. In order to do this, we need to be able to write A_{i+1} in terms of the matrices at distance $i' \leq i$.

Lemma 2.2.2
Let Γ be a distance-regular graph with diameter d, vertex set V, and adjacency matrices $A_0, A_1, \ldots, A_d$. Then for all i, j,

$$A_i A_j = \sum_{k=0}^{d} p_{ij}^k A_k$$

where $p_{ij}^k = |\{w \in V \mid \text{dist}\,(u,w) = i \quad \text{and} \quad \text{dist}\,(w,v) = j\}|$ for fixed vertices u and v such that $\text{dist}\,(u,v) = k$.

Proof. Let Γ be a distance-regular graph with vertex set V, diameter d, and distance graphs $\Gamma_0, \Gamma_1, \ldots, \Gamma_d$ with adjacency matrices $A_0, A_1, \ldots, A_d$. Then for vertices u, v at distance k,

$$\begin{aligned} p_{ij}^k &= |N_i(u) \cap N_j(v)| \\ &= \{w \in V \mid \text{dist}\,(u,w) = i \quad \text{and} \quad \text{dist}\,(w,v) = j\} \\ &= \{w \in V \mid (u,w) \in \Gamma_i \quad \text{and} \quad (w,v) \in \Gamma_j\} \ . \end{aligned}$$

Then $A_i A_j = \sum_k p_{ij}^k A_k$ as shown in the equivalence between the two definitions of an association scheme (Definitions 2.1.2 and 2.1.1). ■

Lemma 2.2.3
Let Γ be a distance-regular graph with diameter d and adjacency matrices $A_0, A_1, \ldots, A_d$.

Then for all i,

$$A_i A_1 = \sum_{k=0}^{d} p_{i1}^k A_k$$

where $p_{i1}^{i+1} \neq 0$ (for $i < d$), and $p_{i1}^k = 0$ for all $k > i+1$.

Proof. If Γ has diameter d, then there is a shortest path of length d. Note that every subpath of a shortest path is also a shortest path, so that there exists a shortest path of every length $k \leq d$.

Let u, v be vertices at distance $i+1$, and consider a shortest path between them. On this path from u to v, consider the first vertex w after u. Then $\operatorname{dist}(u,w) = 1$, and since the remaining portion of the path, from w to v is also a shortest path, $\operatorname{dist}(w,v) = i$. Therefore,

$$|\{w \in V \mid \operatorname{dist}(u,w) = 1 \quad \text{and} \quad \operatorname{dist}(w,v) = i\}| = p_{1i}^{i+1} = p_{i1}^{i+1} > 0 \ .$$

Now consider u, v at distance any k, and $w \in N_1(u) \cap N_i(v)$. By the triangle inequality,

$$k = \operatorname{dist}(u,v) \leq \operatorname{dist}(u,w) + \operatorname{dist}(w,v) = 1 + i \ ,$$

so that if $k > i+1$, then $p_{i1}^k = 0$. ■

Theorem 2.2.4

Every distance-regular graph Γ forms a P-polynomial association scheme $\mathcal{A}$, with the distance graphs of Γ for classes. Moreover, the diameter of Γ is the diameter of $\mathcal{A}$.

Proof. Let Γ be a distance-regular graph with diameter d, distance graphs $\Gamma_0, \Gamma_1, \ldots, \Gamma_d$, and distance matrices $A_0, A_1, \ldots, A_d$. From Lemma 2.2.2, the product of distance matrices of a distance-regular graph belongs to the span of distance matrices. Since the distance matrices are symmetric, the rest of the axioms follow. Therefore, Γ is an association scheme.

Note that $A_0 = I$ is a polynomial of degree 0 in A_1, and A_1 is a polynomial of degree 1 in itself. Then, supposing A_i is a polynomial of degree i in A_1, we have that $A_i A_1$ is a polynomial of degree $i+1$. Then by Lemma 2.2.3,

$$A_i A_1 = \sum_{k=0}^{i+1} p_{i1}^k A_k \implies A_{i+1} = \frac{1}{p_{i1}^{i+1}} \left(A_i A_1 - \sum_{k=0}^{i} p_{i1}^k A_k \right) \ ,$$

which demonstrates that A_{i+1} is a polynomial of degree $i+1$ in A_1. ■

In fact, the P-polynomial schemes correspond exactly with the association schemes arising from distance-regular graphs. In particular, every P-polynomial scheme contains at least

one class which is a distance-regular graph, and whose distance graphs form the classes of the scheme. A P-polynomial scheme can be generated by more than one class, but no more than two classes.

There is also an analogous definition of a *Q-polynomial scheme*. If there exists an idempotent, declared F_1 such that each F_j is a degree-j polynomial in F_1 (using the Schur product in place of the regular product), then the scheme is Q-polynomial. There is an additional theory of duality with P- and Q-polynomial translation schemes (see Section 2.4), but these will not be discussed in this report.

Most of this section comes from [God93, Section 11.2]. See [God93, Section 12.3] for more.

2.3 Automorphisms and Cayley Graphs

This section and the next describes a special class of symmetric graphs (respectively, association schemes). As with many other mathematical structures, the *symmetry* of graphs (or schemes) is made precise by examining its automorphisms – those transformations of the object in question which leave its structure unchanged. Graphs (or schemes) with certain automorphisms may be classified in this way.

More importantly for the purposes of this report, the structure revealed by the automorphisms of graphs (or schemes) allows one to compute their eigenvalues significantly more efficiently than otherwise would be the case, as outlined in the previous sections.

Definition 2.3.1 (Automorphism)
For graphs Γ, Γ', a map $\varphi : V(\Gamma) \to V(\Gamma')$ is a HOMOMORPHISM *if*

$$\forall u, v \in V(\Gamma)\ \ u \sim_\Gamma v \implies \varphi(u) \sim_{\Gamma'} \varphi(v)\ .$$

An ISOMORPHISM *is an invertible homomorphism whose inverse is also a homomorphism; an* AUTOMORPHISM *is an isomorphism from a graph to itself.* $\operatorname{Aut}\Gamma$ *denotes the set of all automorphisms on Γ; it is the subgroup of* $\operatorname{Sym}(V(\Gamma))$*, consisting of those permutations which preserve the (edge) structure of Γ.*

An automorphism of an association scheme $\mathcal{A}$ on a vertex set V is a map $V \to V$ which is simultaneously an automorphism of every class in the scheme. In other words, $\operatorname{Aut}\mathcal{A}$ *is the intersection of the automorphism groups of each class.*

Definition 2.3.2 (Cayley Graphs)
Given any group G and a subset $C \subseteq G$, then C is INVERSE-CLOSED *if for each $g \in C$,*

$g^{-1} \in C$ *as well.*

If $C \subseteq G$ *is an inverse-closed subset of a group* G*, then the* CAYLEY GRAPH *of* G *with respect to* C *is denoted* $\mathrm{Cay}(G, C)$*, and defined as follows:*

- *Its vertex set is* G
- $g \sim h$ *in* $\mathrm{Cay}(G, C)$ *if and only if* $gh^{-1} \in C$

Since C *is inverse closed,* $gh^{-1} \in C \iff hg^{-1} \in C$ *so that* $\mathrm{Cay}(G, C)$ *is undirected.*

Furthermore, if $1_G \notin C$*, then* $g \not\sim g$ *so that the graph is loopless. (By definition it already lacks parallel edges.)*

Because Cayley graphs are defined from groups using only the group structure, it is intuitive that these graphs should be highly symmetric. For example, every Cayley graph is VERTEX TRANSITIVE: for every pair u, v in the vertex set, there is a automorphism taking $u \mapsto v$. To see this, note that G acts (see Definition B.1.1) on $\mathrm{Cay}(G, C)$ through the group operation, since the vertex set is also the group. To verify that this is a homomorphism, if $u \sim v$ in $\mathrm{Cay}(G, C)$, then

$$uv^{-1} \in C \implies (ug)(vg)^{-1} = ugg^{-1}v^{-1} = u1v^{-1} = uv^{-1} \in C$$

so that $ug \sim vg$. Then, $\mathrm{Cay}(G, C)$ is clearly vertex transitive if G acts transitively, and for any vertices u, v, the group element $u^{-1}v$ maps u to v. Moreover, this action is free, since if $ug = u$, then the group cancellation law implies that g is trivial. Together, this implies that the action of G is regular, which suggests the following lemma characterizing Cayley graphs.

(This action is also faithful since if g and h in G are distinct, then they map the vertex 1 to different images – namely g and h respectively. However this observation irrelevant for this lemma, since it restricts to the action of an automorphism group, which is automatically faithful.)

Lemma 2.3.3
For a graph Γ*, there exists a subgroup* $G \leq \mathrm{Aut}\,\Gamma$ *which acts regularly on* Γ *if and only if* $\Gamma \cong \mathrm{Cay}(G, C)$ *for some inverse-closed* $C \subseteq G$*.*

Since the above argument demonstrates the reverse implication, only the forward direction will be shown here.

Before beginning the proof, it will be worthwhile to note the neighbours of 1_G in $\mathrm{Cay}(G, C)$: $g \sim 1_G$ precisely when $g1^{-1} = g \in C$.

Proof. Choose a vertex $v \in V(\Gamma)$ to identify with 1_G. (This choice will not matter in the end, as Cayley graphs are vertex transitive.) Since the action is regular, for each $u \in V(\Gamma)$ there exists a unique $g_u \in G$ such that $vg_u = u$ (Lemma B.1.4).

Then define

$$C := \{g \in G \mid vg \sim v\}$$

and observe that for $u, w \in V(\Gamma)$, $ug_u^{-1} = v$, and $w = vg_w \implies wg_u^{-1} = vg_wg_u^{-1}$. So, since g_u^{-1} is an automorphism of Γ,

$$u \sim w \iff ug_u^{-1} \sim wg_u^{-1} \iff v \sim vg_wg_u^{-1} \iff g_wg_u^{-1} \in C \ .$$

Therefore, the map $u \mapsto g_u$ is the desired isomorphism $\Gamma \to \mathrm{Cay}(G, C)$. ■

As promised at the beginning of the section, the next lemma demonstrates (for graphs) how automorphisms may be used to derive eigenvalues, and moreover, their eigenvectors. Naively, computing the eigenvalues of a matrix A involves solving its characteristic polynomial, which is generically difficult. Then for an eigenvalue θ, finding a θ-eigenvector involves computing the kernel of $A - \theta I$, which can be computed in polynomial time (though not in linear time), and fast numeric algorithms are typically inexact.

However, given the right information about a group, the following result finds the eigenvectors and eigenvalues almost instantaneously.

Lemma 2.3.4
Let G be a finite abelian group, let $C \subseteq G \backslash \{1\}$ be inverse-closed, and define $\Gamma := \mathrm{Cay}(G, C)$. Then the rows of the character table of G provide a complete set of eigenvectors for the adjacency matrix A of Γ. Specifically, if ψ is a character of G (equivalently, a row of its character table), then $\psi(C)$ is the eigenvalue of ψ.

Proof. Note first that the neighbours $h \sim g$ of a vertex $g \in G$ consist of precisely the set $\{cg \mid c \in C\} = Cg$ since $h \sim g \iff hg^{-1} \in C$, and multiplication by g is invertible.

As in Equation B.1, characters are identified with row vectors such that $\psi(g) \rightsquigarrow \psi_g$.

Then

$$(A\psi)_g = \sum_{h \in G} A_{g,h}\psi(h) = \sum_{h \sim g} \psi(h) = \sum_{c \in C} \psi(cg) = \psi(g) \sum_{c \in C} \psi(c) = \psi_g\psi(C) \ .$$

Furthermore, since the rows of the character table are orthogonal, the eigenvectors ψ are linearly independent, and since $G \cong G^*$ (Theorem B.3.2), the character table is square, so that the rows form a basis of eigenvectors for A. ■

The material in this section follows mostly from [God93, Chapter 9].

2.4 Partitions and Translation Schemes

In a sense, this section generalizes the characterization of Cayley graphs from the previous section to the setting of association schemes. Throughout this section, a transitive, abelian group of automorphisms will replace the regular automorphism group which corresponds to a Cayley graph. As per Lemma B.1.3 the transitive, abelian group will act regularly, so that Lemma 2.3.3 still applies. This motivates the following definition.

Definition 2.4.1 (Translation Schemes [God93, Section 12.10])
A TRANSLATION SCHEME *is an association scheme whose automorphism group contains a transitive, abelian subgroup.*

Lemma 2.4.2 ([God93, Section 12.10])
If $\mathcal{A}$ is a translation scheme, and G is a transitive, abelian automorphism group, then there is a partition of G into inverse-closed sets $C_0, C_1, \ldots, C_d$ where $C_0 = \{1\}$, and each graph Γ_i in $\mathcal{A}$ is isomorphic to $\mathrm{Cay}(G, C_i)$.

Proof. Since G is abelian and is a transitive subgroup of $\mathrm{Aut}\,\Gamma_i$ for each $i = 0, 1, \ldots, d$, G acts regularly on Γ_i. Therefore, by Lemma 2.3.3, there exists an inverse-closed set $C_i \subseteq G$ such that $\Gamma_i \cong \mathrm{Cay}(G, C_i)$.

In particular, since the edges of Γ_0 are the diagonal relation, the Cayley graph of G with respect to $C_0 = \{1\}$ generates Γ_0.

Otherwise, it suffices to show that $C_0, C_1, \ldots, C_d$ partition G. Recall from the proof of Lemma 2.3.3 that any vertex may be chosen to identify with 1_G, so that the same vertex (say, v) may be chosen for each graph Γ_i without loss of generality; in this case C_i consists of the neighbours of v. By the definition of an association scheme, for each vertex u there is exactly one graph Γ_i in which $u \sim v$, so that for each vertex, there is exactly one C_i containing it. ■

In order to characterize the translation schemes in a similar manner to the Cayley graphs, an examination of partitions of matrices and groups will be required. This will lead to a simple criterion that distinguishes those partitions which generate a translation scheme from those which do not. [God93, Section 12.10]

Definition 2.4.3 (Partition Matrix [God93, Section 12.7])
If ρ is a partition of a set X, then the PARTITION MATRIX *of ρ is the* 01 *matrix whose rows are indexed by the elements of X, and whose columns are indexed by the parts of ρ, in which each row – corresponding to $x \in X$ – has exactly one* 1*, in the column corresponding to the part that contains x.*

Any partition matrix may be obtained from an $X \times X$ identity matrix by merging the

columns which correspond to elements in the same part. Note that this implies that the columns are linearly independent. (The rows will **not** be linearly independent unless the partition is induced by the diagonal relation.)

Definition 2.4.4 (Induced Row Partition [God93, Section 12.7])
Given a matrix H, if ρ is a partition of the columns with partition matrix $\chi(\rho)$ then the INDUCED ROW PARTITION *ρ^* is the partition of the rows of H such that two rows are in the same part if and only if the corresponding rows in $H\chi(\rho)$ are equal.*

In other words, if f is the function which maps each row index i of H to the row vector $(H\chi(\rho))_i$, then ρ^* is the partition given by the fibres of f. [God93, Section 12.7]

Theorem 2.4.5 (Bridges and Mena [God93, Theorem 12.10.1])
Let G be a finite abelian group, let $\rho = \{C_0, C_1, \ldots, C_d\}$ be a partition of G into inverse-closed parts where $C_0 = \{1\}$, and let ρ^* be the induced row partition of the character table H of G.

Then $|\rho^*| \geq |\rho|$, and the graphs $\Gamma_i := \mathrm{Cay}(G, C_i)$ form the classes of an association scheme if and only if $|\rho^*| = |\rho|$.

Proof. Let A_i be the adjacency matrix of Γ_i, and observe that the set $\{A_0, A_1, \ldots, A_d\}$ is linearly independent. This is because the sets C_i partition G, and in each Γ_i the set C_i consists of the neighbours of 1. The fact that the C_i partition G also implies that $\sum_i A_i = J$, and since $C_0 = \{1\}$, $A_0 = I$.

By Lemma 2.3.4, each character ψ of G (i.e. row of H) is a common eigenvector of $A_0, A_1, \ldots, A_d$, with eigenvalue $\psi(C_i)$ at A_i. Define $\mathbb{A} := \mathrm{span}\{A_0, A_1, \ldots, A_d\}$. Let χ_{C_i} be the characteristic vector of C_i in G, and let

$$\chi(\rho) = \begin{bmatrix} | & | & & | \\ \chi_{C_0} & \chi_{C_1} & \cdots & \chi_{C_d} \\ | & | & & | \end{bmatrix} \tag{2.9}$$

be the partition matrix of ρ.

Let $D_0, D_1, \ldots, D_e$ be the parts of ρ^*; then i, k (or, their characters ψ^i, ψ^k) belong to the same part D_j precisely when the rows $\psi^i\chi(\rho), \psi^k\chi(\rho)$ in

$$H\chi(\rho) = \begin{bmatrix} — & \psi^1 & — \\ & \vdots & \\ — & \psi^n & — \end{bmatrix} \begin{bmatrix} | & | & & | \\ \chi_{C_0} & \chi_{C_1} & \cdots & \chi_{C_d} \\ | & | & & | \end{bmatrix} \tag{2.10}$$

are equal. Together, the characters of each D_j span a common eigenspace of the A_i: let F_j be the orthogonal projection matrix onto this subspace.

Define $\mathbb{F} := \operatorname{span}\{F_0, F_1, \ldots, F_e\}$. Since the $\operatorname{col} F_j$ are spanned by disjoint sets of characters, the subspaces are orthogonal and the F_j are linearly independent; since together the characters span $\mathbb{C}^n$ (where n is the order of G), the (direct) sum of the subspaces is $\mathbb{C}^n$ as well. Therefore,

$$I = F_0 + F_1 + \cdots + F_e \ .$$

Furthermore, since $\operatorname{col} F_j$ is a common eigenspace for the A_i, for $i = 0, 1, \ldots, d$ and $j = 0, 1, \ldots, e$ there exist constants $P_i(j)$ such that

$$A_i F_j = P_i(j) F_j \implies A_i = \sum_{j=0}^{e} P_i(j) F_j \implies \mathbb{A} \leq \mathbb{F} \ . \tag{2.11}$$

This implies that

$$|\rho| = d + 1 = \dim \mathbb{A} \leq \dim \mathbb{F} = e + 1 = |\rho^*| \ .$$

Note that the $F_0, F_1, \ldots, F_e$ are orthogonal idempotents, so they are closed under the regular matrix product. This implies that the algebra they generate is simply $\mathbb{F}$. On the other hand, while $A_0, A_1, \ldots, A_d$ are orthogonal idempotents with respect to the *Schur product*, they may generate an algebra with the usual product that is strictly larger than $\mathbb{A}$ – it must, however, be contained in $\mathbb{F}$. We will show that these two algebras are actually equal. In this case, $e = d$ if and only if $\mathbb{A}$ is closed under regular matrix multiplication; given the results above, this will then be true if and only if $A_0, A_1, \ldots, A_d$ forms an association scheme.

From Equation 2.11, if $g(x)$ is any polynomial, then $g(A_i) = \sum_j g(P_i(j)) F_j$. In particular, if $x_0^{s_0} x_1^{s_1} \cdots x_d^{s_d}$ is any monomial, $A_i^{s_i} = \sum_j P_i(j)^{s_i} F_j$ as above, so that evaluating the monomial at $(A_0, A_1, \ldots, A_d)$ yields

$$A_0^{s_0} A_1^{s_1} \cdots A_d^{s_d} = \prod_i \sum_j P_i(j)^{s_i} F_j = \sum_j \left(\prod_i P_i(j)^{s_i} \right) F_j$$

since the F_j are orthogonal idempotents. Since any polynomial g in $d+1$ variables is a linear combination of such monomials, it follows that

$$\begin{aligned} g(A_0, A_1, \ldots, A_d) &= \sum_j g(P_0(j), P_1(j), \ldots, P_d(j)) F_j \\ \implies g(A_0, A_1, \ldots, A_d) F_j &= g(P_0(j), P_1(j), \ldots, P_d(j)) F_j \end{aligned}$$

for all $j = 0, 1, \ldots, e$.

Now let P be the $(e+1) \times (d+1)$ matrix such that $P_{ji} = P_i(j)$. Note that the rows of P are precisely the distinct rows of $H\chi(\rho)$, so that for any two rows $j \neq j'$ of P, there

exists a column $i(j,j')$ such that $P_{i(j,j')}(j) \neq P_{i(j,j')}(j')$. This allows for the definition of the polynomials

$$g_j(x_0, x_1, \ldots, x_d) := \prod_{j' \neq j} \left(x_{i(j,j')} - P_{i(j,j')}(j')\right)$$

so that when applied at $A_0, A_1, \ldots, A_d$,

$$\begin{aligned} g_j(A_0, A_1, \ldots, A_d) F_{j''} &:= \prod_{j' \neq j} \left(P_{i(j,j')}(j'') - P_{i(j,j')}(j')\right) F_{j''} \\ &= g_j\left(P_0(j''), P_1(j''), \ldots, P_d(j'')\right) F_{j''} \ . \end{aligned}$$

By construction, if $j'' = j$, then $f_j := g_j\left(P_0(j''), P_1(j''), \ldots, P_d(j'')\right)$ will be nonzero, but if $j'' \neq j$, then there will be some $j' = j''$ at which $P_{i(j,j')}(j'') - P_{i(j,j')}(j') = 0$ so that $g_j\left(P_0(j''), P_1(j''), \ldots, P_d(j'')\right) = 0$.

This construction demonstrates that for each j, there exists a polynomial g_j such that

$$g_j(A_0, A_1, \ldots, A_d) = \sum_{j'} f_j \delta_{jj'} F_{j'} = f_j F_j$$

where $f_j \neq 0$, so that F_j can be written as a polynomial in $A_0, A_1, \ldots, A_d$. This proves that each F_j is contained in the algebra generated by $\mathbb{A}$, and so all of $\mathbb{F}$ is contained in this algebra. Since the reverse inclusion was already shown, this proves that the two algebras are equal, as desired. ■

It is interesting to note that, while the character table H may in general be complex, each of the orthogonal projection matrices F_j is real. To see this, note that the A_i are real, symmetric matrices, so that their eigenvalues $P_i(j)$ are real as well. Then, each of the polynomials g_j (used to express F_j in the algebra generated by the A_i) must also be real, since they were defined in terms of the $P_i(j)$. Therefore, not only are the $\mathbb{C}$-algebras of $\mathbb{A}$ and $\mathbb{F}$ equal, but so are their $\mathbb{R}$-algebras.

With this result, translation schemes are characterized by a finite abelian group G and partition ρ satisfying the condition given. Moreover, a finite abelian group G is isomorphic to its groups of characters, G^* (Theorem B.3.2), and the group of characters is completely described by the character table H. Likewise, the group partition ρ is completely described by its partition matrix $\chi(\rho)$. Since the condition in Theorem 2.4.5 depends only on these two matrices (both with respect to the same ordering on G), if it is satisfied, then the matrices completely describe the translation scheme they generate.

2.5 The Eigenvalues of the Hamming Scheme

The theory just developed in the previous section can be applied immediately to the Hamming scheme. This scheme is of great utility in the setting of coding theory, in part because it is a P-polynomial scheme, generated by the distance-regular Hamming graph. Moreover, it is also a translation scheme, with respect to a particularly nice group, and a simple partition, which will allow us to deduce a formula for the eigenvalues of the scheme. In the particular case of the Hamming graph, an explicit expression for its eigenvalues can be given.

To this end, let $\mathbb{Z}_q$ denote the cyclic group of order q (written additively), and consider the direct product $\mathbb{Z}_q^d$ with subsets

$$C_i := \left\{ x \in \mathbb{Z}_q^d \;\middle|\; |x| = i \right\} \ .$$

In particular, $C_0 = \{0\}$ and

$$C_1 = \mathbb{Z}_q \setminus \{0\} \times \{0\}^{d-1} \bigsqcup \{0\} \times \mathbb{Z}_q \setminus \{0\} \times \{0\}^{d-2} \bigsqcup \cdots \bigsqcup \{0\}^{d-1} \times \mathbb{Z}_q \setminus \{0\} \ .$$

Then x, y are i^{th} associates if and only if $x - y \in C_i$; that is, the Hamming distance between x and y is i, so that this partition yields the Hamming scheme. In particular, $H(d, q) \cong \mathrm{Cay}(\mathbb{Z}_q^d, C_1)$.

From the proof that $\mathbb{Z}_q^d \cong \left(\mathbb{Z}_q^d\right)^*$ (Theorem B.3.2), if ω is a q^{th} complex root of unity, we see that we can identify characters ψ with row vectors of the form

$$\psi \rightsquigarrow \begin{bmatrix} \omega^{\psi_1} & \cdots & \omega^{\psi_d} \end{bmatrix}$$

where $\psi_i \in \{0, 1, \ldots, q-1\}$, and

$$\psi(x) = \prod_{i=1}^{d} \omega^{\psi_i x_i} \ .$$

(Note that this notation deviates from Equation B.1, but will be more convenient for this purpose.) As in the proof of Theorem B.3.2, let e_i denote the tuple of all zeroes, with a 1 in the i^{th} position, so that each group element $x = (x_1, \ldots, x_d)$ can be expressed as $\sum_{i=1}^{d} x_i e_i$.

Then, in the i^{th}-distance Hamming graph, ψ is a $\psi(C_i)$-eigenvalue (Lemma 2.3.4), and

for the Hamming graph, it can be computed directly.

$$\begin{aligned}
\psi(C_1) &= \sum_{c \in C_1} \psi(c) \\
&= \sum_{i=1}^{d} \sum_{j=1}^{q-1} \psi(je_i) \qquad \text{here, the outer sum picks which entry of } c \text{ will be nonzero} \\
&\qquad\qquad\qquad\qquad\qquad\qquad \text{and the inner sum picks the value} \\
&= \sum_{i=1}^{d} \sum_{j=1}^{q-1} \omega^{\psi_i j} = \sum_{i=1}^{d} \sum_{j=1}^{q-1} \left(\omega^{\psi_i}\right)^j \\
&= \sum_{i=1}^{d} \left(\frac{1 - \left(\omega^{\psi_i}\right)^q}{1 - \omega^{\psi_i}} - 1 \right) \qquad \text{using the usual formula for geometric sums} \\
&= \sum_{i=1}^{d} q\delta_{0,\psi_i} - d \\
&= q\,(\text{the number of } i \text{ with } \psi_i = 0) - d
\end{aligned}$$

Since the number k of indices i for which $\psi_i = 0$ can vary from $0, 1, \ldots, d$, and there are $\binom{d}{k}$ places i at which $\psi_i = 0$ and $(q-1)^{d-k}$ choices for the other ψ_j, the eigenvalues of the Hamming graph are given by

$$\begin{cases} qk - d & k = 0, 1, \ldots, d \\ \binom{d}{k}(q-1)^{d-k} & \text{(multiplicity)} \end{cases} . \tag{2.12}$$

For the other graphs in the Hamming scheme, their eigenvalues can be computed as $\psi(C_i)$, although there may not be a particularly nice expression for this sum.

3. Delsarte's Linear Programming Bound

3.1 Linear Programming

The terminology and results from this section, except for the adjective *principal* for constraints, follows [Mat07].

A LINEAR PROGRAMMING PROBLEM (or LINEAR PROGRAM) is an optimization problem in which one seeks to maximize or minimize a linear function of one or more variables, subject to linear constraints. That is, fixing a vector c, one tries to maximize or minimize the linear combinations of the components of c:

$$c_1x_1 + \cdots + c_nx_n = c^Tx$$

for some x. Note that maximizing c^Tx is equivalent to minimizing $(-c)^Tx$, so that for the theory of linear programming, it suffices to consider maximization problems without loss of generality. As in other optimization problems, the function to be maximized (c^Tx in this case) is called the OBJECTIVE (FUNCTION).

In most cases, there will be constraints on the inputs to the objective function, and for the purposes of linear programming these will also have to be linear. That is, there will be a matrix A and vector b such that only inputs x satisfying $Ax \leq b$ will be allowed. (Note that for *vectors* a and b, $a \leq b$ will mean that each component a_i is less than or equal to the corresponding component b_i.) These are called the (PRINCIPAL) CONSTRAINTS, and vectors x which satisfy the constraints will be called FEASIBLE (SOLUTIONS). (Note that in [Del73], the term *program* is used to refer to a feasible solution.)

If there are no constraints on the problem (and even in some cases where there are) through appropriate choices of feasible solution x, the objective c^Tx may be made arbitrarily large, and such problems are called UNBOUNDED. Conversely, if no feasible solutions exist, then the problem is called INFEASIBLE.

Finally, in most applications of linear programming – in particular to the cliques of association schemes – the feasible solutions will be further constrained to those with all non-negative components (i.e. $x \geq 0$). These are called the NON-NEGATIVITY CONSTRAINTS,

in contrast with the *principal constraints*. The non-negativity constraints will be required throughout the remainder of this report.

Therefore, for an objective $c^T x$ and constraints $Ax \leq b$, the associated linear program will be written in STANDARD FORM:

$$\max \left\{ c^T x \mid Ax \leq b,\ x \geq 0 \right\} .$$

3.1.1 Duality

The most important observation about linear programs (for the purposes of this report, at least) is that they come in dual pairs.

Given a linear program $\mathcal{P}$ written in standard form

$$\max \left\{ c^T x \mid Ax \leq b,\ x \geq 0 \right\}$$

its DUAL program is $\mathcal{P}^*$:

$$\min \left\{ b^T y \mid A^T y \geq c,\ y \geq 0 \right\} .$$

Re-writing it in standard form,

$$\max \left\{ (-b)^T y \mid -A^T y \leq -c,\ y \geq 0 \right\}$$

taking the dual

$$\min \left\{ -c^T x \mid -Ax \geq -b,\ x \geq 0 \right\}$$

and re-writing in standard form

$$\max \left\{ c^T x \mid Ax \leq b,\ x \geq 0 \right\}$$

the original (called PRIMAL) linear program is recovered.

This demonstrates that $(\mathcal{P}^*)^* = \mathcal{P}$, so that linear programs come in dual pairs.

Theorem 3.1.1 (Weak Duality)
If x is a feasible solution to a linear program

$$\max \left\{ c^T x \mid Ax \leq b,\ x \geq 0 \right\},$$

and y is a feasible solution to its dual program,

$$\min \left\{ b^T y \mid A^T y \geq c,\ y \geq 0 \right\},$$

then $c^T x \leq b^T y$.

Proof. Let u, v, w be vectors with $u \geq 0$, and $v \leq w$. Then for all components i, $u_i \geq 0$ and $v_i \leq w_i$ implies that $u_i v_i \leq u_i w_i$ so that

$$u^T v = \sum_i u_i v_i \leq \sum_i u_i w_i = u^T w \ .$$

In particular, since y is a feasible solution to the dual program, and $x \geq 0$,

$$c \leq A^T y \implies x^T c \leq x^T A^T y = y^T A x \ .$$

(Here one may take the transpose of the whole expression, since the result is a scalar.) Similarly, since x is a feasible solution to the primal program, and $y \geq 0$,

$$b \geq Ax \implies y^T b \geq y^T A x \ .$$

By combining the two inequalities,

$$b^T y = y^T b \geq y^T A x \geq x^T c = c^T x$$

which is the desired result. ■

As a result of the weak duality of linear programs, every feasible solution to the dual program provides an upper bound on the maximum of the primal, and every feasible solution to the primal program provides a lower bound on the minimum of the dual.

In fact the extremal values of dual programs (the maximum of the primal, and the minimum of the dual) coincide, although this will not be needed for the purposes of this report. This is referred to as *Strong Duality* of linear programs.

3.2 The LP Bound

In this section, we will consider a symmetric association scheme $\mathcal{A}$ with vertex set V, relations $\Gamma_0, \Gamma_1, \ldots, \Gamma_d$, adjacency matrices $A_0, A_1, \ldots, A_d$, and eigenmatrices P and Q. (Since we will assume that $\mathcal{A}$ is symmetric, this will guarantee that P and Q are real.) We will denote by D the set of distances $\{0, 1, \ldots, d\}$, and consider a subset Y of the vertex set whose distances (to one another) belong to a subset of D. Intuitively, we think of Y as a *code* in V; if the minimum distance of the code is greater than 1, then this restricts the possible distances between codewords.

Definition 3.2.1 (Cliques and Cocliques [Del73, Section 3.3])
*Let $Y \subseteq V$, and $C \subseteq D$, where $0 \in C$. Then Y is called a C-*CLIQUE *if*

$$\Gamma_i \cap Y^2 = \emptyset \quad \forall i \in D \setminus C \ .$$

Equivalently, Y is a C-clique if for all $x, y \in Y$,

$$(x, y) \in \Gamma_i \implies i \in C \ .$$

*Let $C^* := C \setminus \{0\}$, and $\overline{C} = D \setminus C^*$. Then Y is a C-clique if and only if it is a $\overline{C}$-*COCLIQUE.
(Note that such a set Y is called a C-code, or $\overline{C}$-anticode in [God93].)

Then a code of minimum distance δ corresponds to a $\{0, \delta, \ldots, d\}$-clique. The terminology derives from the fact that, if the association scheme is generated by a distance-regular graph, then cliques in the graph correspond to $\{0, 1\}$-cliques; cocliques correspond to $\{0, 1\}$-cocliques. (Recall that a *clique* in a graph is a set of vertices such that every pair is adjacent; a *coclique*, or *independent set*, is a set of vertices such that no pair is adjacent.)

However, the notion of C-cliques is a very strict one: that no pair of vertices is i-associated for $i \notin C$. In order to measure arbitrary subsets of the vertex set, we will need to consider a more general indicator of the distribution of distances within the subset.

Definition 3.2.2 (Inner Distribution [Del73, Section 3.1])
The INNER DISTRIBUTION *of a subset Y of vertices is the vector y defined by*

$$y_i := \frac{\Gamma_i \cap Y^2}{|Y|}$$

for $i = 0, 1, \ldots, d$.

Note that if χ_Y is the characteristic vector of Y in V, then for all i,

$$y_i = \frac{\chi_Y^T A_i \chi_Y}{|Y|} \ .$$

Furthermore, $y_0 = 1$ for all subsets, Y, and $\sum_{i=0}^{d} y_i = |Y|$.

By restating the above observation, that $\mathbf{1}^T y = \sum_i y_i = |Y|$, we recognize a linear function which gives the size of a code. For a given minimum distance, we generally try to maximize the size of a code with that minimum distance. This setup should be reminiscent of a linear programming problem. All that remains is to find a linear inequality to constrain the possible inner distributions y. However, if y is truly the inner distribution of a code of minimum distance δ, then we know that $y_i = 0$ for $i = 1, \ldots, \delta - 1$. This will lead to an inequality we can use.

Theorem 3.2.3 (Delsarte Theorem 3.3 [Del73])
For any inner distribution y,

$$Q^T y \geq 0$$

where Q is the matrix of dual eigenvalues. (Here, $x \geq 0$ means that each component of the vector x is not less than 0.)

We will prove this theorem after discussing how it may be used to bound the size of a code in a graph.

This theorem provides the key inequality that will allow the application of linear programming to cliques in association schemes. However, because the constraint vector in a primal linear program becomes the objective in the dual program, this inequality will require some transformation to make it suitable for use in linear programming.

Let Y be an C-clique with inner distribution y. Recall that if $D = \{0, 1, \ldots, d\}$, then for any $C \subseteq D$, that $C^* := C \setminus \{0\}$, and that $Q(0)$ is the first row of Q. Let χ_C denote the characteristic vector of C in D. Then $y_i = 0$ for all $i \notin C$, so $Q^T y \geq 0 \iff Q^T \operatorname{diag}(\chi_C)\, y \geq 0$ since the action of $\operatorname{diag}(\chi_C)$ acting on the left is to zero out the *rows* of y with index not in C. Similarly,

$$Q^T \operatorname{diag}(\chi_C)\, y = Q(0)^T y_0 + Q^T \operatorname{diag}(\chi_{C^*})\, y = \mu + Q^T \operatorname{diag}(\chi_{C^*})\, y$$

since the action of $\operatorname{diag}(\chi_{C^*})$ on the right is to zero out the *columns* of Q^T with index not in C^*, $y_0 = 1$, and

$$Q^T = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \mu_1 & & & \\ \vdots & & * & \\ \mu_d & & & \end{bmatrix} .$$

Finally, since $y \geq 0$, $Q_0^T y \geq 0$ adds no new constraint, so that under the non-negativity constraint $Q^T y \geq 0 \iff \operatorname{diag}(\chi_{D^*})\, Q^T y \geq 0$.

Putting all this together, $Q^T y \geq 0 \iff \operatorname{diag}(\chi_{D^*})\, Q^T \operatorname{diag}(\chi_{C^*})\, y \geq -\mu$ so that Delsarte's LP can be written in standard form:

$$\max \left\{ \mathbf{1}^T \operatorname{diag}(\chi_C)\, y \;\middle|\; Q^T \operatorname{diag}(\chi_C)\, y \geq 0,\ y \geq 0,\ y_0 = 1 \right\} \tag{3.1}$$

$$= \max \left\{ \chi_{C^*}^T y \;\middle|\; -\operatorname{diag}(\chi_{D^*})\, Q^T \operatorname{diag}(\chi_{C^*})\, y \leq \operatorname{diag}(\chi_{D^*})\, \mu,\ y \geq 0 \right\} + 1 \ . \tag{3.2}$$

Taking the dual yields

$$\min \left\{ \mu^T \operatorname{diag}(\chi_{D^*})\, z \;\middle|\; -\operatorname{diag}(\chi_{C^*})\, Q \operatorname{diag}(\chi_{D^*})\, z \geq \chi_{C^*},\ z \geq 0 \right\} + 1 \tag{3.3}$$

$$= \min \left\{ \mu^T z \;\middle|\; -\operatorname{diag}(\chi_{C^*})\, Q \operatorname{diag}(\chi_{D^*})\, z \geq \chi_{C^*},\ z \geq 0,\ z_0 = 1 \right\} \ . \tag{3.4}$$

Therefore, if $z_0 = 1$ is required, recalling that $Q_0 = \mathbf{1}$ and $\operatorname{diag}(\chi_{C^*})\,\mathbf{1} = \chi_{C^*}$, then

$$\begin{aligned}
&\operatorname{diag}(\chi_{C^*})\, Q \operatorname{diag}(\chi_D)\, z \\
=&\operatorname{diag}(\chi_{C^*})\,(Q_0 z_0 + Q \operatorname{diag}(\chi_{D^*})\, z) \\
=&\chi_{C^*} + \operatorname{diag}(\chi_{C^*})\, Q \operatorname{diag}(\chi_{D^*})\, z \\
\leq&0\ .
\end{aligned}$$

This equivalence recovers Delsarte's formulation of the dual linear program:

$$\min\left\{\mu^T z \;\middle|\; \operatorname{diag}(\chi_{C^*})\, Qz \leq 0,\ z \geq 0,\ z_0 = 1\right\}\ . \tag{3.5}$$

Therefore, given an association scheme with dual matrix of eigenvalues Q, if we wish to find an upper bound on the size of a code with minimum distance δ, then we set $C := \{0, \delta, \ldots, d\}$, and solve the linear program in Equation 3.1. Alternatively, any feasible solution to the dual problem (Equation 3.5) will provide a (possibly worse) upper bound.

All that remains is to prove Theorem 3.2.3. To do this, we will begin by introducing a different measure of the distribution of distances in a subset of vertices.

Definition 3.2.4 (Outer Distribution [Del73, Section 3.1])
The OUTER DISTRIBUTION *of Y is the $V \times D$ matrix B whose (v,i)-entry is defined by*

$$B(v,i) := |\Gamma_i \cap (\{v\} \times Y)|\ .$$

The outer distribution can also be given in block-column form:

$$B = \begin{bmatrix} A_0\chi_Y & A_1\chi_Y & \cdots & A_d\chi_Y \end{bmatrix}\ .$$

Lemma 3.2.5 (Delsarte Theorem 3.1 [Del73])
Let Y be a subset of vertices with inner distribution y and outer distribution B. Let P and Q be the eigenmatrices of the association scheme. Then

$$B^T B = \frac{|Y|}{|V|} P^T \operatorname{diag}\left(Q^T y\right) P\ .$$

Proof. Note that the i^{th} column of B is given by $A_i\chi_Y$, so we can compute the (i,j)-entry

of $B^T B$ in the following manner:

$$\begin{aligned}(B^T B)_{i,j} &= \left(\chi_Y^T A_j^T\right)\left(A_i \chi_Y\right) \\ &= \chi_Y^T \left(A_j^T A_i\right) \chi_Y \\ &= \chi_Y^T \left(\sum_{k=0}^{d} p_{ij}^k A_k\right) \chi_Y \\ &= \sum_{k=0}^{d} p_{ij}^k \chi_Y^T A_k \chi_Y \\ &= |Y| \sum_{k=0}^{d} p_{ij}^k y_k\end{aligned}$$

since by definition, $|Y|\, y_k = \chi_Y^T A_k \chi_Y$.

Next, we define $\widetilde{y}^T := y^T Q$ so that

$$\widetilde{y}^T P = y^T Q P = y^T |V| \implies y_k = \frac{1}{|V|} \sum_{r=0}^{d} \widetilde{y}_r P_k(r) \ .$$

Substituting this sum for y_k in the previous expression, we obtain

$$(B^T B)_{i,j} = \frac{|Y|}{|V|} \sum_{k=0}^{d} p_{ij}^k \sum_{r=0}^{d} \widetilde{y}_r P_k(r) = \frac{|Y|}{|V|} \sum_{r=0}^{d} \widetilde{y}_r \sum_{k=0}^{d} p_{ij}^k P_k(r) \ .$$

Then, by noticing that combining Equations 2.1.13 and 2.1.5 yields the inner sum above:

$$\left.\begin{aligned} A_i F_j &= P_i(j) F_j \\ A_i A_j &= \textstyle\sum_{k=0}^{d} p_{ij}^k A_k \end{aligned}\right\} \implies P_i(r) P_j(r) = \sum_{k=0}^{d} p_{ij}^k P_k(r) \ .$$

Therefore,

$$(B^T B)_{i,j} = \frac{|Y|}{|V|} \sum_{r=0}^{d} \widetilde{y}_r P_i(r) P_j(r) \ .$$

Finally, we obtain the desired result by noticing that

$$\sum_{r=0}^{d} \widetilde{y}_r P_i(r) P_j(r) = P^T \operatorname{diag}\left(Q^T y\right) P \ .$$

■

At first, this result may seem to be unrelated to the theorem which we wish to prove.

However, by realizing that $B^T B$ is a positive semi-definite matrix, Theorem 3.2.3 follows as a quick corollary.

Proof of Theorem 3.2.3. From the previous Lemma 3.2.5, we know that

$$B^T B = \frac{|Y|}{|V|} P^T \operatorname{diag}\left(Q^T y\right) P \ .$$

Since $PQ = |V|\, I$ (Property 2.1.11), we can rewrite the above equation as

$$Q^T B^T B Q = |Y|\, |V| \operatorname{diag}\left(Q^T y\right) \ .$$

Then, the (k, k) entry of this matrix is

$$(Q^T y)_k = Q_k^T (B^T B) Q_k \geq 0$$

since $B^T B$ is positive semi-definite. Therefore,

$$Q^T y \geq 0 \ .$$

■

3.3 The Ratio Bound

From the LP bound, we can quickly derive an eigenvalue bound on the size of a clique in a distance-regular graph. We do this by finding a feasible solution to the dual problem (Equation 3.5).

Theorem 3.3.1 (Ratio Bound for Cliques [Del73, Section 3.3.2])
Let Y be a clique in a distance-regular graph. Let λ_1 be the greatest and λ_n the least eigenvalue of the adjacency matrix of the graph. Then

$$|Y| \leq 1 - \frac{\lambda_1}{\lambda_n} \ .$$

Proof. Let P and Q be the eigenmatrices of the association scheme generated by the graph, and let ν and μ be the vectors of valencies and multiplicities respectively. Note that the 1^{th} column of P, P_1 contains the eigenvalues of the adjacency matrix. We will sort the rows of P such that P_1 is sorted in descending order. Note also that the valency ν_1 is the largest

eigenvalue (Property 2.1.9), so $\nu_1 = \lambda_1$. Therefore, we can write

$$P^T = \begin{bmatrix} 1 & \cdots & 1 \\ \nu_1 & \cdots & \lambda_n \\ \vdots & * & \\ \nu_d & & \end{bmatrix} .$$

Define the vector

$$z := \begin{bmatrix} 1 & 0 & \cdots & \frac{-\lambda_1}{\mu_d \lambda_n} \end{bmatrix}^T .$$

Note that since the trace of an adjacency matrix is zero, and that the trace of any diagonalizable matrix is the sum of its eigenvalues, since λ_1 is positive, λ_n must be negative. Therefore, $z \geq 0$ and $z_0 = 1$. So if z satisfies $\operatorname{diag}(\chi_{C^*}) Qz \leq 0$, then it will be a feasible dual solution; furthermore, $\mu^T z = 1 - \lambda_1/\lambda_n$, which is the desired bound. Since we are looking for an upper bound on the size of a clique, we set $C := \{0, 1\}$.

Using Equation 2.1.12, we compute

$$\begin{aligned} \operatorname{diag}(\chi_{C^*}) Qz &= \operatorname{diag}(\chi_{C^*}) \operatorname{diag}(\nu)^{-1} P^T \operatorname{diag}(\mu) z \\ &= \begin{bmatrix} \frac{\nu_1}{\nu_1} & \cdots & \frac{\lambda_n}{\lambda_1} \end{bmatrix} \begin{bmatrix} 1 \\ \vdots \\ \frac{-\lambda_1}{\mu_d \lambda_n} \end{bmatrix} \\ &= 1 - 1 = 0 \leq 0 \end{aligned}$$

as desired. ■

3.4 The Clique-Coclique Bound

Similar to the bound in the previous section, by finding a feasible solution to the dual linear program, we will obtain an upper bound on the size of a C-clique.

Theorem 3.4.1 (Clique-Coclique Bound [Del73, Theorem 3.9])
Let Y be a C-clique and Z be a C-coclique (i.e. a $\overline{C}$-clique) in a symmetric association scheme with vertex set V. Then

$$|Y| \cdot |Z| \leq |V| .$$

Proof. Let y and z be the inner distributions of Y and Z respectively, let μ be the vector of multiplicities, and for each $k \in D = \{0, 1, \ldots, d\}$ define

$$\zeta_k := \frac{1}{\mu_k |Z|} \sum_j z_j Q_k(j) .$$

We will look for an upper bound on $|Y|$ by demonstrating that ζ is a feasible solution to the dual linear program *for* Y.

Then $\zeta = \frac{1}{|Z|} \operatorname{diag}(\mu)^{-1} Q^T z$, and $Q^T z \geq 0$ so $\zeta \geq 0$ as well. At $k = 0$, $\mu_0 = 1$ and $Q_0 = \mathbf{1}$, so $\zeta_0 = \frac{1}{|Z|} \mathbf{1}^T z = 1$.

To prove that ζ is a feasible dual solution, it remains to show that $\operatorname{diag}(\chi_{C^*}) Q\zeta \leq 0$. However,

$$Q\zeta = \frac{1}{|Z|} Q \operatorname{diag}(\mu)^{-1} Q^T z = \frac{|V|}{|Z|} \operatorname{diag}(\nu)^{-1} z$$

using Equation 2.1.10. Since $z_i = 0$ for all $i \in C$ (since Z is a C-coclique), it follows that $\operatorname{diag}(\chi_{C^*}) Q\zeta = 0 \leq 0$ as desired.

Then to compute the upper bound, we compute

$$\mu^T \zeta = Q(0)\zeta = \left(\frac{|V|}{|Z|} \operatorname{diag}(\nu)^{-1} z \right)_0 = \frac{|V|}{|Z|}$$

which implies that $|Y| \leq |V| / |Z|$ as desired. ■

4. Schrijver's Semi-Definite Programming Bound

Delsarte's linear programming bound results from the discovery of an object which depends on a given code (the inner distribution) in such a way that the size of the code can be computed from the object. On the other hand, all of these objects resulting from such codes satisfied a certain property (an inequality, derived from the fact that a certain matrix is positive semi-definite). Therefore, we may put an upper bound on the size of certain codes by maximizing the object in question, subject to its constraints.

In this chapter, we will investigate an improved bound on the size of a code in a binary Hamming graph $H(d, 2)$ by following the outline above. Where previously the objects were non-negative vectors satisfying a linear inequality, now the objects will be positive-definite matrices satisfying linear equalities. In both cases, the objective will be to maximize a linear function of the objects in question.

4.1 Semi-Definite Programming

The material in this section comes from [Ber09].

For a given distance-regular graph, and a minimum distance, Delsarte's LP bound is the result of a maximization problem based on the input parameters. There is no known general solution to this problem, and so there is no explicit formula for the bound for all distance-regular graphs and minimum distances. While some bounds can be found be specifying a feasible dual solution, as done in Sections 3.3 and 3.4, these are not guaranteed to be optimal.

Similarly, Schrijver's SDP bound is computed by maximizing a semi-definite program built from the input graph and minimum distance. To this end, this section will discuss some aspects of semi-definite programming.

We will use the notation $S \succeq 0$ to assert that the matrix S is positive semi-definite, and

$S \succeq T$ to assert that $S - T \succeq 0$. Given an objective matrix C, constraint matrices K_i, and constraint values b_i for $i = 1, \ldots, m$, the semi-definite programming problem is to maximize the following constrained problem

$$\max \{\langle C \ , \ X\rangle \mid \langle K_i \ , \ X\rangle = b_i \ , X \succeq 0\} \ .$$

In fact, semi-definite programs are a strict generalization of linear programs. Given a linear program,

$$\max \left\{c^T x \ \middle| \ kx \leq b, \ x \geq 0\right\}$$

the inequality constraint can be re-expressed as an equality constraint $kx' = b$ by adding an extra variable to x for each row of k (these variables are called *slack* variables). If each new variable s is constrained to be non-negative, then for each row k_i in k,

$$k_i x + s = b_i \iff kx \leq b_i$$

so that the constraints are equivalent. By extending the objective c with zeroes to match the length of x', the optimal value will also be equal. Then, an equivalent semi-definite program can be constructed by associating each vector with the diagonal matrix formed by that vector (including the rows of k):

$$\begin{aligned} c &\leftrightsquigarrow \operatorname{diag}(c) \\ x \geq 0 &\leftrightsquigarrow \operatorname{diag}(x) \succeq 0 \\ c^T x &\leftrightsquigarrow \langle \operatorname{diag}(c) \ , \ \operatorname{diag}(x)\rangle \\ k_i &\leftrightsquigarrow \operatorname{diag}(k_i) \\ k_i x = b_i &\leftrightsquigarrow \langle \operatorname{diag}(k_i) \ , \ \operatorname{diag}(x)\rangle = b_i \ . \end{aligned}$$

As with linear programming, each semi-definite program is equipped with a dual program:

$$\min \left\{ -\sum_{i=1}^{m} y_i b_i \ \middle| \ -C - \sum_{i=1}^{m} y_i K_i \succeq 0 \right\} \ .$$

Furthermore, the weak duality theorem also holds true for semi-definite programs.

Theorem 4.1.1 (Weak Duality [Ber09, Proposition 5.1])
If feasible and bounded, the maximum value of the primal semi-definite program is at most the minimum value of the dual program.

Proof. Let X be a feasible solution to the primal program, and y a feasible solution to the

dual. Then define

$$S := -C - \sum_{i=1}^{m} y_i K_i \succeq 0 \ ,$$

and compute

$$\langle S \ , \ X \rangle = - \langle C \ , \ X \rangle - \sum_i y_i \langle K_i \ , \ X \rangle = - \langle C \ , \ X \rangle - \sum_i y_i b_i \ .$$

Then, to prove $\langle S \ , \ X \rangle \geq 0$ is equivalent to proving that $-\sum_i y_i b_i \geq \langle C \ , \ X \rangle$ as desired.

Since both S and X are symmetric, positive semi-definite matrices, each can be orthogonally diagonalized, and their eigenvalues must be non-negative. Write

$$S = U_S \Delta_S U_S^T$$
$$X = U_X \Delta_X U_X^T$$

where U_S, U_X are orthogonal matrices, and Δ_S, Δ_X are diagonal and non-negative.

Using the cyclicity of trace, we can compute

$$\begin{aligned} \langle S \ , \ X \rangle &= \operatorname{trace} SX \\ &= \operatorname{trace} U_S \Delta_S U_S^T U_X \Delta_X U_X^T \\ &= \operatorname{trace} \Delta_S U_S^T U_X \Delta_X U_X^T U_S \\ &= \sum_i (\Delta_S)_i (U_S^T U_X \Delta_X U_X^T U_S)_i \ . \end{aligned}$$

Since Δ_S and Δ_X are positive semi-definite, and $U_X^T U_S$ is invertible, the conjugated matrix $U_S^T U_X \Delta_X U_X^T U_S$ is also positive semi-definite, so the sum is too. This proves the claim. ■

Unlike with linear programs, the difference between the minimum of the dual program and the maximum of the primal may be zero, finite, or infinite.

4.2 The Terwilliger Algebra of the Hamming Scheme

This section follows [Sch05, Subsection I. A] closely.

Where Delsarte's LP bound used properties of the Bose-Mesner algebra of an association scheme, Schrijver's SDP bound will use a *Terwilliger algebra*, which is a non-commutative superset of the Bose-Mesner algebra.

Recall that for any Hamming graph $H(d, 2)$, we typically take the vertex set to be the group $\mathbb{Z}_2^d$. However, this group also forms a ring, with entrywise multiplication: in

fact, this is an example of a *boolean ring*. One way to view the set $\mathbb{Z}_2^d$ is as the set of functions $[d] \to \{0,1\}$, which has a natural interpretation as the powerset of $[d]$. In this interpretation, each function is identified with its support (the set of elements in $[d]$ which map to 1). Then, the addition in the ring corresponds to taking the symmetric difference of sets; the multiplication in the ring corresponds to the intersection of sets; the additive identity corresponds to the empty set; and the multiplicative identity corresponds to the entire set $[d]$. The union of sets u and v can also be computed as

$$u \cup v = u + v + uv \ .$$

Furthermore, the Hamming weight of a vertex is precisely its cardinality as a set. Both of these interpretations of the vertex set of $H(d,2)$ – as a ring, and as the powerset of $[d]$ – will be used throughout the following two sections.

Definition 4.2.1 (Terwilliger Algebra)
Let $H(d,2)$ be a binary Hamming graph with vertex set V. For all integers $i,j,t \in D$, define the $V \times V$ matrix $M_{i,j}^t$ such that for all vertices u and v,

$$(M_{i,j}^t)_{u,v} := \begin{cases} 1 & \text{if} \quad |u| = i, \ |v| = j, \quad \text{and} \quad |uv| = t \\ 0 & \text{else} \end{cases} .$$

The algebra generated by the $M_{i,j}^t$ is called the Terwilliger algebra *of $H(d,2)$. (In fact, it is the Terwilliger algebra with respect to $\mathbf{0}$, but this is has no bearing on this report.)*

From the definition of the Bose-Mesner algebra, the adjacency matrix A_k is 1 wherever $|u+v| = k$, and 0 elsewhere. For all u and v, such that $|u| = i$, $|v| = j$, and $|uv| = t$, then $|u+v| = |u| + |v| - 2|uv| = i + j - 2t$, so the set of entries at which $M_{i,j}^t$ is 1 is a subset of the entries at which A_{i+j-2t} is 1. In particular, each A_k is a sum of some $M_{i,j}^t$, so that the Terwilliger algebra does indeed contain the Bose-Mesner algebra.

Like the Bose-Mesner algebra, the Terwilliger algebra is closed under transposition, since for all i,j,t, $(M_{i,j}^t)^T = M_{j,i}^t$. Like the adjacency matrices A_k, the $M_{i,j}^t$ are closed under matrix multiplication. To see this, note that when $j \neq j'$, the product $M_{i,j}^t M_{j',k}^s$ is zero, and when $j = j'$, and u and w are vertices satisfying $|u| = i$, $|w| = k$, and $|uw| = r$,

$$\begin{aligned} (M_{i,j}^t M_{j',k}^s)_{u,w} &= |\{v \in V \mid |v| = j, \ |uv| = t, \ |vw| = s\}| \\ &= \sum_{l=0}^{r} \binom{r}{l} \binom{i-r}{t-l} \binom{k-r}{s-l} \binom{d-i-k+r}{j-t-s+l} . \end{aligned}$$

Since this depends only on the values of $|u|$, $|w|$, and $|uw|$, it must be a linear combination of the $M_{i,k}^p$. However, unlike the Bose-Mesner algebra, the Terwilliger algebra is not generally

commutative.

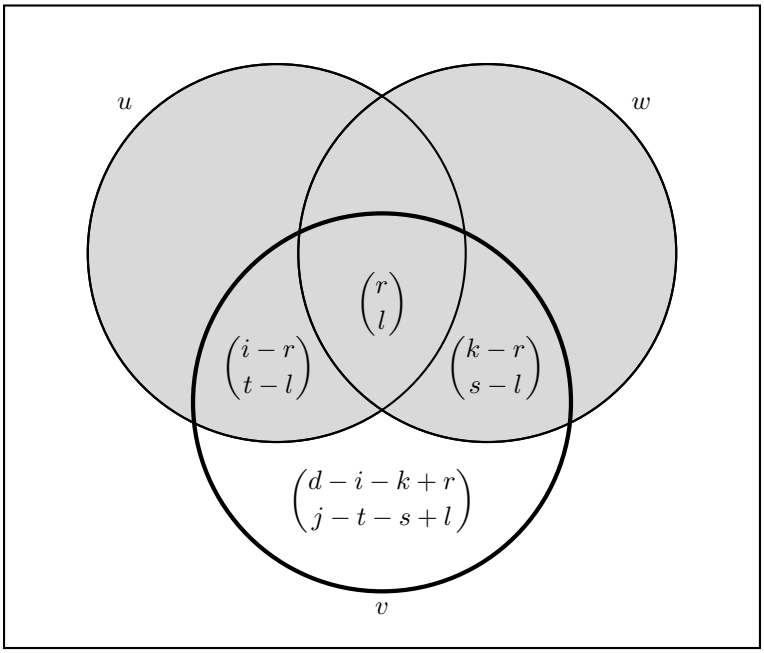

Figure 4.1: A term in the (u, w) entry of $M^t_{i,j} M^s_{j,k}$.

Like the adjacency matrices, the nonzero $M^t_{i,j}$ of the Terwilliger algebra are linearly independent, since for any pair of vertices u, v, and weights $i' := |u|$, $j' := |v|$, and $t' := |uv|$, there will be exactly one of the $M^t_{i,j}$ which is nonzero at the (u, v)-entry: when $i = i'$, $j = j'$, and $t = t'$. On the other hand, the number of triples (i, j, t) such that $M^t_{i,j}$ is nonzero is equal to the number of triples (a, b, t) such that there exist disjoint subsets u', v', w' of $[d]$ with cardinalities a, b, and t. This is straightforward to see, as every pair of vertices u and v with $|u| = i$, $|v| = j$, and $|uv| = t$ can be decomposed into three disjoint sets $u \setminus v$, $v \setminus u$, and uv, and vice versa. There exist disjoint subsets u', v' and w' for every a, b, and t as long as $a + b + t \leq d$, and the number of solutions to this in non-negative integers is $\binom{d+3}{3}$. Therefore, this is the dimension of the Terwilliger algebra.

In fact, there exists an explicit diagonalization of the Terwilliger algebra, such that computations in the algebra may be done with objects of dimension $\binom{d+3}{3}$.

4.3 The SDP Bound

This section follows [Sch05, Subsection I. B] closely.

In order to place an upper bound on the sizes of codes in a binary Hamming graph following the recipe outlined at the beginning of this chapter, for each code we require a

positive semi-definite matrix from which we can compute the size of the code. Furthermore, these objects must satisfy some linear (in)equality constraints.

Therefore, let Y be a code, and let $\lambda^t_{i,j}$ denote the number of triples (u, v, w) in Y^3 such that $|u+v| = i$, $|u+w| = j$, and $|(u+v)(u+w)| = t$. Using the multinomial notation

$$\binom{d}{i,j,t} := \frac{d!}{i!j!t!(d-i-j-t)!},$$

we define

$$x^t_{i,j} := \frac{\lambda^t_{i,j}}{|Y| \binom{d}{i-t,j-t,t}} \tag{4.1}$$

when $\binom{d}{i-t,j-t,t}$ is nonzero, and $x^t_{i,j} := 0$ when it is zero. (Here, i, j, t run over $D = \{0, 1, \ldots, d\}$, where $i + j - t \leq d$.)

With this notation, we can define the "objects" of our code Y:

$$\begin{aligned} R &:= \sum_{i,j,t} x^t_{i,j} M^t_{i,j} \\ R' &:= \frac{|Y|}{2^d - |Y|} \sum_{i,j,t} \left(x^0_{i+j-2t,0} - x^t_{i,j}\right) M^t_{i,j} \ . \end{aligned} \tag{4.2}$$

It will turn out that these matrices are both positive semi-definite. However, it will be easier to compute $|Y|$, as well as to place constraints on the numbers $x^t_{i,j}$ directly. For any valid triple i', j', t' we can find a pair of vertices u, v satisfying $|u| = i'$, $|v| = j'$, and $|uv| = t'$ and create the matrix M which is 1 in the (u, v) entry, and 0 elsewhere. Then,

$$\langle M \ , \ R \rangle = \sum_{i,j,t} x^t_{i,j} \langle M \ , \ M^t_{i,j} \rangle = x^{t'}_{i',j'}$$

So any linear (in)equality constraint on the $x^t_{i,j}$ can be translated into a linear (in)equality constraint on R (or R'). Similarly, any linear combination of the $x^t_{i,j}$ can be expressed as a linear function of R.

Below, we discuss the objective function on the "objects" whose maximum is an upper bound on the size of the code; the positive semi-definiteness of the "objects"; and the constraints on the "objects". We do each of these in terms of the values $x^t_{i,j}$ instead of R or R' for simplicity.

Theorem 4.3.1 (Objective)

The size of a code Y can be computed from its matrix R. In particular,

$$|Y| = \sum_{i=0}^{d} \binom{d}{i} x_{i,0}^0 \ .$$

Proof. To begin,

$$\sum_{i=0}^{d} \binom{d}{i} x_{i,0}^0 = \sum_{i=0}^{d} \binom{d}{i} \frac{\lambda_{i,0}^0}{|Y| \binom{d}{i}} = \frac{1}{|Y|} \sum_{i=0}^{d} \lambda_{i,0}^0 \ ,$$

so it is sufficient to show that $\sum_{i=0}^{d} \lambda_{i,0}^0 = |Y|^2$.

However, $\lambda_{i,0}^0$ counts the number of triples $(u, v, w) \in Y^3$ such that $|u + v| = i$ and $|u + w| = |(u + v)(u + w)| = 0$. However, $|u + w| = 0$ implies that $u = w$ and $|(u + v)(u + w)| = 0$, so $\lambda_{i,0}^0$ counts the number of pairs $(u, v) \in Y^2$ such that $|u + v| = i$. Summing over all i, there are $|Y|^2$ such pairs. ■

Theorem 4.3.2 (Positive Semi-Definiteness)
For any code Y, its matrices R and R' are positive semi-definite.

Proof. Recall from Section 1.3 the automorphism group of the binary Hamming graph, $H(d, 2)$. Let Π be the set of automorphisms π such that $\mathbf{0} \in \pi(Y)$; let Π' be the set of automorphisms π such that $\mathbf{0} \notin \pi(Y)$. For each automorphism π, let $\chi_{\pi(Y)}$ be the characteristic vector of the image of Y under π. Then we define

$$\begin{aligned} R &:= \frac{1}{|\Pi|} \sum_{\pi \in \Pi} \chi_{\pi(Y)} \chi_{\pi(Y)}^T \\ R' &:= \frac{1}{|\Pi'|} \sum_{\pi \in \Pi'} \chi_{\pi(Y)} \chi_{\pi(Y)}^T \ . \end{aligned} \tag{4.3}$$

It suffices to show that these definitions equal those given in Equation 4.2 since every matrix of the form BB^T is positive semi-definite, and a non-negative linear combination of positive semi-definite matrices is also positive semi-definite.

Fix a vertex u, and let Π_u be the set of automorphisms such that $\pi(u) = \mathbf{0}$. Then as seen in Corollary 1.3.8, there are $d!(2-1)!^d = d!$ automorphisms in Π_u. Let $\lambda_{i,j}^{t,u}$ denote the number of pairs $(v, w) \in Y^2$ such that $|u + v| = i$, $|u + w| = j$, $|(u + v)(u + w)| = t$, define

$$R_u := \frac{1}{|\Pi_u|} \sum_{\pi \in \Pi_u} \chi_{\pi(Y)} \chi_{\pi(Y)}^T$$

and claim that

$$R_u = \sum_{i,j,t} \binom{d}{i - t, j - t, t}^{-1} \lambda_{i,j}^{t,u} M_{i,j}^t \ . \tag{4.4}$$

Since $|\Pi_u| = d!$, the above claim is equivalent to the statement that

$$\sum_{\pi \in \Pi_u} \chi_{\pi(Y)} \chi^T_{\pi(Y)} = \sum_{i,j,t} (i-t)!\ (j-t)!\ t!\ (n-i-j+t)!\ \lambda^{t,u}_{i,j} M^t_{i,j}\ ,$$

which is demonstrated in the following Lemma 4.3.3.

Now, note that Π is the disjoint union of Π_u, as u runs over Y, while Π' is the disjoint union of Π_u where u runs over the complement of Y. Since there are 2^d total vertices, and $|\Pi| = |Y|\, d! = |Y|\, |\Pi_u|$ for any u, $|\Pi'| = (2^d - |Y|)\, |\Pi_u|$. Therefore,

$$R = \sum_{u \in Y} \sum_{\pi \in \Pi_u} \frac{1}{|Y|\, |\Pi_u|} \chi_{\pi(Y)} \chi^T_{\pi(Y)} = \frac{1}{|Y|} \sum_{u \in Y} R_u\ ,$$

and similarly,

$$R' = \frac{1}{2^d - |Y|} \sum_{u \notin Y} R_u\ .$$

Substituting Equation 4.4 into the equations above and simplifying using the following facts proves the claim.

$$\sum_{u \in Y} \lambda^{t,u}_{i,j} = \lambda^t_{i,j} \tag{4.5}$$

$$\sum_{u \notin Y} \lambda^{t,u}_{i,j} = \binom{i+j-2t}{i-t} \binom{d-i-j+2t}{t} \lambda^0_{i+j-2t,0} - \lambda^t_{i,j} \tag{4.6}$$

$$\binom{d}{i+j-2t}^{-1} = \binom{d}{i-t, j-t, t}^{-1} \binom{i+j-2t}{i-t} \binom{d-i-j+2t}{t} \tag{4.7}$$

$$\tag{4.8}$$

The first fact above follows directly from the definition of $\lambda^t_{i,j}$, while the third fact can be shown by expanding each binomial and multinomial coefficient, and cancelling the appropriate terms. The second fact above is shown in Lemma 4.3.4.

Following the proofs of Lemma 4.3.3 and Lemma 4.3.4, this completes the proof of the theorem.

■

Lemma 4.3.3

For any code Y, and any vertex u (not necessarily in the code), the following equality holds:

$$\sum_{\pi \in \Pi_u} \chi_{\pi(Y)} \chi^T_{\pi(Y)} = \sum_{i,j,t} (i-t)!\ (j-t)!\ t!\ (n-i-j+t)!\ \lambda^{t,u}_{i,j} M^t_{i,j}\ .$$

(Here, $\lambda^{t,u}_{i,j}$ is used as in the proof of Theorem 4.3.2.)

Proof. Note that the (v, w)-entry of $\chi_{\pi(Y)}\chi_{\pi(Y)}^T$ is 1 if v and w both belong to $\pi(Y)$, and is 0 otherwise. Therefore, the (v, w)-entry of the sum $\sum_{\pi\in\Pi_u} \chi_{\pi(Y)}\chi_{\pi(Y)}^T$ counts the number of automorphisms π such that $\pi(u) = \mathbf{0}$ and $v, w \in \pi(Y)$. By taking the inverse of each such automorphism, we see this is equal to the number of automorphisms π such that $\pi(\mathbf{0}) = u$, and $\pi(v), \pi(w) \in Y$. Using the automorphism decomposition from Theorem 1.3.7, we see that such an automorphism $\pi = \sigma T_u$ (since $\tau = \mathrm{id}$ for all automorphisms of $H(d, 2)$). Therefore, the number of automorphisms π satisfying the required property is precisely the number of permutations $\sigma \in \mathrm{Sym}[d]$ such that $v\sigma + u$ and $w\sigma + u$ belong to Y.

Recall that the vertex set $\mathbb{Z}_2^d$ can be viewed as the powerset of $[d]$, where each vertex is identified with its support. Then, each permutation $\sigma \in \mathrm{Sym}[d]$ acts on subsets of $[d]$ by mapping them to their image under σ.

Suppose that $|v| = i$, $|w| = j$, and $|vw| = t$. Then $\lambda_{i,j}^{t,u}$ gives the number of pairs $(v', w') \in C^2$ such that $|u + v'| = i$, $|u + w'| = j$, and $|(u + v')(u + w')| = t$. For each such pair (v', w'), there are $(i-t)!\ (j-t)!\ t!\ (n-i-j+t)!$ permutations σ mapping v to $u+v'$, and w to $u+w'$. To see this, note that each such permutation σ can be decomposed into a permutation fixing the sets $v \setminus w$, $w \setminus v$, vw, and $\overline{v \cup w}$, followed by a permutation mapping

$$\begin{cases} v \setminus w & \mapsto (u + v') \setminus (u + w') \\ w \setminus v & \mapsto (u + w') \setminus (u + v') \\ vw & \mapsto (u + v')(u + w') \\ \overline{v \cup w} & \mapsto \overline{(u + v') \cup (u + w')} \end{cases}.$$

This is possible since each of the sets above is the same size as its image, and together the sets partition the powerset of $[d]$. (Here, $\overline{u}$ denotes the complement of u as a subset of $[d]$.) ■

Lemma 4.3.4

For all i, j, and t,

$$\sum_{u\notin Y} \lambda_{i,j}^{t,u} = \binom{i+j-2t}{i-t}\binom{d-i-j+2t}{t}\lambda_{i+j-2t,0}^0 - \lambda_{i,j}^t\ .$$

(Here, $\lambda_{i,j}^{t,u}$ is used as in the proof of Theorem 4.3.2.)

Proof. Note that it suffices to show that

$$\sum_{u} \lambda_{i,j}^{t,u} = \binom{i+j-2t}{i-t}\binom{d-i-j+2t}{t}\lambda_{i+j-2t,0}^0\ .$$

Furthermore, $\lambda_{i+j-2t,0}^0$ counts the number of pairs $(u', v', w') \in C^3$ such that $|v' + w'| =$

$i + j - 2t$ and $|u' + w'| = |(v' + w')(u' + w')| = 0$. (Note that here, the roles of u' and w' have been swapped with respect to their usual roles.) For each such pair (v', w'), define $v := v'$ and $w := w'$. Then, there will be $\binom{i+j-2t}{i-t}\binom{d-i-j+2t}{t}$ vertices u such that $|u + v| = i$, $|u + w| = j$, and $|(u + v)(u + w)| = t$.

To see this, we instead count $\widetilde{u} := u + v$, since v is fixed, and u can be recovered as $u = \widetilde{u} + v$. We can choose $\widetilde{u}$ directly (as a subset of [d]) so that $|u + v| = i$ by selecting a set a of $i - t$ elements from $\binom{i+j-2t}{i-t}$ choices in $v + w$, and selecting a set b of t elements from $\binom{d-i-j+2t}{t}$ choices in the complement of $v + w$. In this way we can write $\widetilde{u} = a \sqcup b$, where $a \subseteq v + w$ and $b \cap (v + w) = \emptyset$.

Then, we can write $u + w = v + z \sqcup b \setminus a$, from which we can compute

$$|u + w| = (i + j - 2t) + (t) - (i - t) = j$$

as desired. Furthermore, we can compute

$$\begin{aligned} |v + w| &= |u + v| + |u + w| - 2\,|(u + v)(u + w)| \\ \implies |(u + v)(u + w)| &= -\frac{1}{2}\left((i + j - 2t) - i - j\right) = t\ , \end{aligned}$$

which proves the claim.

■

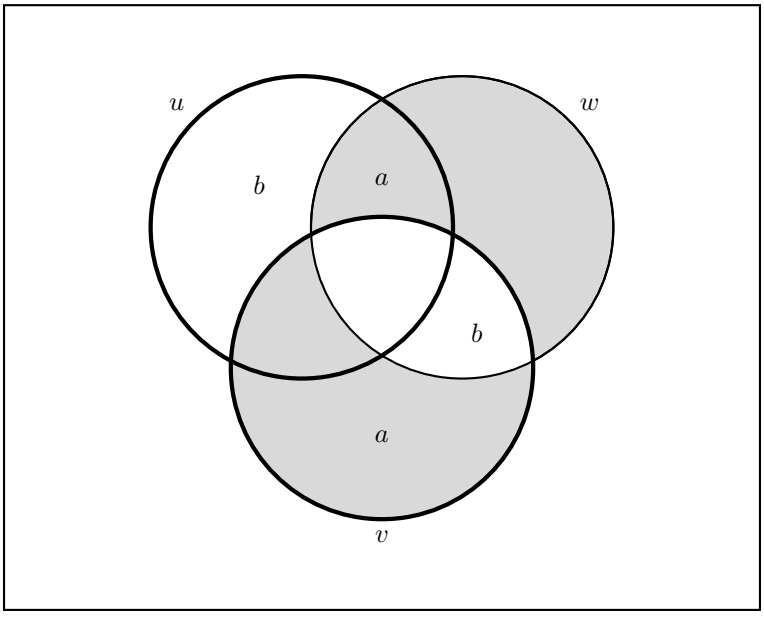

Figure 4.2: In the proof of Lemma 4.3.4, we construct a set $u + v$ given sets v and w as the disjoint union of $a = u(v + w)$ and $b = u \setminus (v + w)$.

Theorem 4.3.5 (Constraints)
Let Y be a code of minimum distance δ. Then the following constraints hold.

(i) $x^0_{0,0} = 1$.

(ii) For all i, j, t,

$$0 \leq x^t_{i,j} \leq x^0_{i,0} \leq 1 \qquad \textit{and} \qquad x^0_{i,0} + x^0_{j,0} \leq 1 + x^t_{i,j} \ .$$

(iii) For all i, j, t, if the tuple $(i, j, i + j - 2t)$ is a permutation of $(i', j', i' + j' - 2t')$, then $x^t_{i,j} = x^{t'}_{i',j'}$.

(iv) For all i, j, t,

$$\{i, j, i + j - 2t\} \cap \{1, \ldots, \delta - 1\} \neq \emptyset \implies x^t_{i,j} = 0 \ .$$

Proof. (i) Note that $\lambda^0_{0,0}$ is the number of triples $(u, v, w) \in Y^3$ such that $|u + v| = |u + w| = |(u + v)(u + w)| = 0$. Since the weight of any vertex is 0 if and only if that vertex is itself the zero vertex, we conclude that $u = v = w$. (This follows from the fact that in $\mathbb{Z}_2^d$, each element is its own additive inverse.) Since there are $|Y|$ choices of vertex in Y, and $\binom{d}{0,0,0} = 1$, we see that $x^0_{0,0} = |Y| / |Y| = 1$.

(ii) Consider the two equivalent definitions of R (Equations 4.2 and 4.3):

$$\frac{1}{|\Pi|} \sum_{\pi \in \Pi} \chi_{\pi(Y)} \chi^T_{\pi(Y)} = \sum_{i,j,t} x^t_{i,j} M^t_{i,j} \ .$$

For any vertex u, if $|u| = i$, then the diagonal element of the u^{th} row of R is $x^0_{i,0}$, since $(M^t_{i,j})_{u,v}$ is nonzero when $|u| = i$, $|v| = j$, and $|u + v| = t$. Therefore, it will suffice to prove that for each row of R, its diagonal element is at least as large as any other.

However, the (u, v)-entry in $\chi_{\pi(Y)} \chi^T_{\pi(Y)}$ is 1 whenever u and v both belong to $\pi(Y)$, and is 0 elsewhere. However, if u is in $\pi(Y)$, then the (u, u) entry will certainly be nonzero, whereas the (u, v)-entry may or may not be zero. Therefore, when summing over $\pi \in \Pi$, the diagonal entry is counted at least as many times as any other entry in its row. Therefore, $x^t_{i,j} \leq x^0_{i,0}$.

Further, let $\Pi(u)$ be the set of permutations π such that $u \in \pi(Y)$. Since $\Pi(u) \subseteq \Pi$, it follows that $x^0_{i,0} = |\Pi(u)| / |\Pi| \leq 1$. So, for each triple (i, j, t), let u and v be vertices such that $|u| = i$, $|v| = j$, and $|u + v| = t$, so that the (u, v)-entry of R is $x^t_{i,j}$. Then, $x^t_{i,j}$ counts the number of permutations π for which $u \in \pi(Y)$ and $v \in \pi(Y)$, and since

$$|\Pi(u)| + |\Pi(v)| = |\Pi(u) \cap \Pi(v)| + |\Pi(u) \cup \Pi(v)| \leq |\Pi(u) \cap \Pi(v)| + |\Pi|$$

by dividing the above relation by $|\Pi|$, we see that $x^0_{i,0} + x^t_{j,0} \leq 1 + x^t_{i,j}$.

(iii) Recall that the vertex set $\mathbb{Z}_2^d$ of a binary Hamming graph may also be viewed as the powerset of $[d]$, by identifying each vertex with its support. Furthermore, recall that $\lambda^t_{i,j}$ counts the number of triples $(u, v, w) \in Y^3$ such that $|u + v| = i$, $|u + w| = j$, and $|(u + v)(u + w)| = t$. Since $(u + v) + (u + w) = v + w$ we can see that

$$|v + w| = |u + v| + |u + w| - 2\,|(u + v)(u + w)|$$

so that $|v + w| = i + j - 2t$ if and only if $|(u + v)(u + w)| = t$. Therefore, $\lambda^t_{i,j}$ also counts the number of triples $(u, v, w) \in Y^3$ such that $|u + v| = i$, $|u + w| = j$, and $|v + w| = i + j - 2t$. Since the roles of u, v, and w are symmetric in this definition, a permutation of the values i, j, and $i + j - 2t$ must leave $\lambda^t_{i,j}$ unchanged.

(iv) Recall from Definition 3.2.1 that Y is a C-clique if the distance between any pair of vertices is in C. By this definition, Y is a code of minimum distance δ if and only if it is a C-clique with $C = \{0, \delta, \ldots, d\}$. Following the previous proof, then for all $(u, v, w) \in Y^3$, their distances $i := |u + v|$, $j := |u + w|$, and $i + j - 2t := |v + w|$ must belong to C. If any of i, j, or $i + j - 2t$ belong to $D \setminus C$, then $\lambda^t_{i,j} = 0$, as no such triple of vertices (u, v, w) may all belong to Y.

■

5. Conclusion

Motivated by the desire to transmit messages in such a way that they are robust to errors which may occur during transmission, algebraic coding theory typically seeks a subspace of a finite vector space in which to encode messages. To correct as many errors as possible, it is beneficial for the subspace to have a large minimum distance. However, for the purposes of efficiency, it is desirable that the subspace should be large. Due to this tradeoff, given a fixed minimum distance, we seek upper bounds on the size of a code.

This question transfers naturally from the subspaces of a finite vector space, to the subsets of a Hamming graph – or more generally, a distance-regular graph. In this setting, a code of a certain minimum distance is equivalent to a coclique (or independent set) in a related graph. Then, to find upper bounds on the codes, cocliques, and cliques as well, we notice that distance-regular graphs form a structure called an association scheme.

Association schemes have a number of useful algebraic properties that allow us to treat codes, cocliques, and cliques uniformly, and to compute upper bounds on their size, given a specific graph and minimum distance. This computation relies specifically on the change of basis matrices P and Q in the association scheme. While for arbitrary association schemes computing these matrices involves diagonalizing all the classes in the scheme, for the P-polynomial schemes (those arising from a distance-regular graph) there exists an adjacency matrix A_1 such that every other adjacency matrix A_i is a polynomial in A_1. Therefore, any eigenvector of A_1 is automatically an eigenvector of each A_i, which is significantly more efficient. Even more efficient than this, when an association scheme contains a transitive abelian subgroup of its automorphism group, the eigenvectors of the entire scheme can be read directly from the character table of the group.

For the Hamming scheme in particular, in large part because it has such a large automorphism group, it forms a translation scheme. Furthermore, there exists a stronger bound for the codes of the Hamming scheme, although this requires working in the larger Terwilliger algebra. In both cases, the bounds are formulated as convex optimization problems: the first is expressed as a linear programming problem with constraint matrix Q; the other is posed as a semi-definite programming problem.

The LP bound is generally more efficient, and linear programs are easier to solve than

semi-definite ones, and moreover, the objects in question in the LP bound (the matrix of dual eigenvalues Q) are of order $(d+1) \times (d+1)$. On the other hand, the dimension of the Terwilliger algebra used in the SDP bound is $\binom{d+3}{3}$, which is cubic in d, the diameter of the scheme. For the Hamming scheme in particular, the Q matrix can be directly computed, and the Terwilliger algebra has an explicit diagonalization, but more generally, the orders of objects in the Terwilliger algebra is exponential in d, as are objects such as the character table of the transitive, abelian automorphism group. Therefore, when computing such bounds, the explicit diagonalization of the Terwilliger algebra is necessary; the explicit formula for Q helps, but as character tables may be generated on-the-fly the LP bound may be formulated efficiently for translation schemes.

However, despite the fact that the LP bound is generally more efficient to compute, and its simplicity allows for explicit upper bounds given by feasible dual solutions, the SDP bound is more powerful. An example of the LP and SDP upper bounds on a Hamming $H(d, 2)$ scheme for some diameters d and minimum distances δ is shown in Figure 5.1.

d	δ	SDP	LP
19	6	1280	1289
23	6	13766	13775
25	6	47998	48138
19	8	142	145
20	8	274	290
25	8	5477	5557
27	8	17768	18189
28	8	32151	32206
22	10	87	95
25	10	503	551
26	10	886	1040

Figure 5.1: Upper bounds for binary Hamming codes of diameter d and minimum distance δ, from [Sch05, Table I]

In the process of deriving these bounds, much is learned about the automorphisms and eigenvalues of the Hamming scheme, as well as about association schemes in general. This topic uses a unique blend of ideas from graph theory, group theory, linear algebra, and optimization. There are also a number of additional uses of association schemes to combinatorial design, representation theory, and statistics that were not covered in this report. Furthermore, not all aspects of the theory of association schemes were covered. More information may be found in the sources used to compile this report; see the bibliography for more.

A. Linear Algebra

The material in this chapter may be drawn from most standard texts on linear algebra; for example, see [Fri03]. Most of the material mentioned here is also mentioned in [God93, Sections 2.5, 2.6, and the Appendix].

The reader of this report is expected to be familiar with the standard notions of finite-dimensional vector spaces over $\mathbb{R}$ or $\mathbb{C}$. Some topics which are more important to this report, and less commonly emphasized elsewhere, are included here for reference.

Throughout the report, the vector spaces in question will typically be $\mathbb{C}^n$ or $\mathbb{R}^n$, for various values of n, with the standard inner product. Given a matrix (or vector) M, its adjoint will be written M^*. If M is real, then this is simply the transpose; if M is complex, then it is the conjugate transpose.

A.1 The Spectral Theorem

A.1.1 Orthogonal Projection

Let X be an inner product space, let W be a subspace of X, and let the columns of a matrix B be a basis for W. Then the matrix BB^*, which maps all vectors into W, is the *orthogonal projection* onto W. It is a *projection* in that it is idempotent ($(BB^*)^2 = BB^*$), and it is *orthogonal* because it maps vectors x to the closest vector w in W to x (then $x - w$ is orthogonal to W).

Recall that the *trace* of a matrix is the sum of its diagonal entries. If the matrix is diagonalizable, then it is also the sum of its eigenvalues (with multiplicity). The following facts will be importing in the course of the report.

Lemma A.1.1
The eigenvalues of an idempotent matrix are all in $\{0, 1\}$.

Proof. Let F be an idempotent matrix, so that $F^2 - F = 0$. If v is an eigenvector with

eigenvalue λ, then

$$0 = (F^2 - F)v = F^2v - Fv = \lambda^2 v - \lambda v = (\lambda^2 - \lambda)v \ .$$

Since v cannot be zero, this implies that $\lambda(\lambda - 1) = 0$. ■

Proposition A.1.2
The trace of diagonalizable idempotent matrix is its rank.

Proof. The trace of a matrix is the sum of its eigenvalues, and the eigenvalues of an idempotent matrix are all in $\{0, 1\}$. The trace is the number of 1s in the diagonal representation of the matrix, which is equal to the rank. ■

A.1.2 Spectral Decomposition

Definition A.1.3
A linear operator is called NORMAL *if it commutes with its adjoint.*

The following result regarding normal operators of a finite-dimensional inner product space is central to this report.

Theorem A.1.4
If M is a normal operator, then M is unitarily diagonalizable. That is, there exists a unitary matrix U and a diagonal matrix Δ such that $M = U\Delta U^*$.

While this is the way that the spectral theorem is normally presented, it will be useful to look closer at this result.

Let M be a normal matrix with distinct eigenvalues θ, and eigenspaces E_θ. Let F_θ be the orthogonal projection onto E_θ. Then,

$$M = \sum_\theta \theta F_\theta \qquad \text{and} \qquad I = \sum_\theta F_\theta \ .$$

This *spectral decomposition* will be the form used throughout this report.

There are a number of important facts about this decomposition which will also be useful. First, the F_θ are orthogonal idempotents, in that $F_\theta F_{\theta'} = \delta_{\theta\theta'} F_\theta$. This implies the following facts.

Proposition A.1.5
Let M, θ, and F_θ be as above.

1. *If f is any polynomial, then $f(M) = \sum_\theta f(\theta) F_\theta$.*
2. *Each F_θ is a polynomial in M.*

Proof (sketch). If f has degree 1, then the first fact is clearly true. If instead $f(M) = M^2$, then

$$\left(\sum_\theta \theta F_\theta\right)\left(\sum_{\theta'} \theta' F_{\theta'}\right) = \sum_\theta \sum_{\theta'} \theta\theta' F_\theta F'_\theta = \sum_\theta \sum_{\theta'} \theta\theta' \delta_{\theta\theta'} F_\theta = \sum_\theta \theta^2 F_\theta = \sum_\theta f(\theta) F_\theta \ .$$

These facts extend in the same way to polynomials of any degree.

Then, since the eigenvalues θ are distinct, for each eigenvalue θ, there exists a polynomial – for example, $\prod_{\theta' \neq \theta}(x - \theta')$ – which is nonzero on exactly θ and no other eigenvalues. This allows one to express each F_θ as a polynomial in M. ■

A.2 Positive Semi-Definite Matrices

In this section, we will restrict the discussion to real, symmetric matrices.

Definition A.2.1
A real, symmetric matrix F is POSITIVE SEMI-DEFINITE *if for all vectors x,*

$$x^T F x \geq 0 \ .$$

This is the most common definition of a positive semi-definite matrix, but in fact there are a number of other equivalent conditions.

Theorem A.2.2
The following are equivalent:

- *The matrix F is positive semi-definite.*
- *There exists a (possibly non-square) matrix B such that $F = B^T B$.*
- *All the eigenvalues of F are non-negative and F is real and symmetric (or more generally, complex and hermitian).*

These conditions will be used interchangeably in this report.

It also follows directly from the definition, that any non-negative linear combination of positive semi-definite matrices is also positive semi-definite. That is, if $\alpha_0, \alpha_1, \ldots, \alpha_d$ are non-negative real numbers, and $F_0, F_1, \ldots, F_d$ are positive semi-definite matrices, then $\alpha_0 F_0 + \alpha_1 F_1 + \cdots + \alpha_d F_d$ is also positive semi-definite.

Furthermore, it is important to note that if U is any invertible operator, then F is positive semi-definite if and only if $U^T F U$ is also positive semi-definite. This is because for any vector x, there exists a unique pre-image y of x, such that $x^T F x = (Uy)^T F (Uy) \geq 0$.

B. Group Theory

B.1 Group Actions

The material of this section comes primarily from [Dum04, Section 1.7, Chapter 4].

Definition B.1.1 (Group Action)
Given a group G and a set X, GROUP ACTION *is a homomorphism $G \to \operatorname{Sym} X$, where $\operatorname{Sym} X$ is the symmetric group on X.*

A group action $\varphi : G \to \operatorname{Sym} X$ induces a product $X \times G \to X$ by mapping $(x, g) \mapsto \varphi(g)(x)$. When the action is clear from context, this will be denoted $x \cdot g$, or simply xg. This is called a RIGHT ACTION, as *g acts on the right of x* (the corresponding notion of a LEFT ACTION can also be defined.)

Conversely, given a product $X \times G \to X$, the same expression defines a map $G \to \operatorname{Sym} X$. If such a product satisfies

$$\forall x \in X \; x1_G = x$$
$$\text{and } \forall x \in X \; \forall g, h \in G \; (xg)h = x(gh)$$

then the induced map $G \to \operatorname{Sym} X$ will be a homomorphism, so that these definitions are equivalent.

(In [Dum04] this is taken as the definition of a group action, and the homomorphism $G \to \operatorname{Sym} X$ is called its PERMUTATION REPRESENTATION. It will be occasionally convenient to adopt each perspective.)

Definition B.1.2 (Types of Group Actions)
If a homomorphism $G \to \operatorname{Sym} X$ is injective, then the action is called FAITHFUL. *Note that a group homomorphism is injective if and only if it has a trivial kernel.*

Given a group action $G \to \operatorname{Sym} X$, $g \in G$ is called FIXED POINT-FREE *if $\forall x \in X \; xg \neq x$. The group action itself is called* FIXED POINT-FREE *(or just* FREE*) if all its nontrivial elements are fixed point-free.*

A group action $G \to \operatorname{Sym} X$ is called TRANSITIVE *if $\forall x, y \in X$ there exists some $g \in G$ such that $xg = y$.*

A group action is called REGULAR *if it is simultaneously transitive and free. (This terminology follows [God93].)*

Note that if X is a structure with automorphisms (such as a graph or group), G is a subgroup of $\operatorname{Aut} X$, and G acts in the natural way on X (i.e. $xg = g(x)$), then this action is faithful. That is, $\operatorname{Aut} X \leq \operatorname{Sym} X$, so that this action is induced by the inclusion $G \hookrightarrow \operatorname{Sym} X$, which is clearly injective.

Lemma B.1.3

If an abelian group G acts faithfully and transitively on a set X, then the action is free, and thus also regular. [Dum04, Section 4.1, Exercise 3]

Proof. Let $g \in G$ be nontrivial, and $x \in X$. The goal is to prove that $xg \neq x$.

Since g is not the identity, there exists some $y \in X$ such that $z := yg \neq y$. Furthermore, since G acts transitively on X, there exists some $h \in G$ such that $yh = x \iff y = xh^{-1}$. Then,

$$\begin{aligned} xg &= (yh)g \\ &= y(hg) \\ &= y(gh) \quad \text{since } G \text{ is abelian} \\ &= (yg)h \\ &= zh\ . \end{aligned}$$

If $zh = x$ then, $z = xh^{-1} = y$, but by definition, $z = yg \neq y$, so $xg = zh \neq x$. ∎

An alternate characterization of regular actions will be useful in this report. To see this, note that for a pair $x, y \in X$, there exists a $g \in G$ such that $xg = y$ by transitivity; for any $g' \in G$ satisfying $xg' = y$,

$$xg = xg' \implies x = xg'g^{-1}$$

so that $g'g^{-1} = 1$ since the action is free, and so $g' = g$. Conversely, if for each $x, y \in X$ there existed a unique $g \in G$ satisfying $xg = y$, then the action would clearly be transitive; since $x1 = x$, 1 is the unique group element fixing any point, so the action must be free.

Lemma B.1.4

A group action G on X is regular if and only if for all $x, y \in X$, there exists a unique $g \in G$ such that $xg = y$.

B.2 The Structure of Finite Abelian Groups

Theorem B.2.1 (Structure Theorem)
If G is a finitely generated abelian group, then G is isomorphic to a direct product of cyclic groups. Specifically,

$$G \cong \mathbb{Z}_{q_1}^{d_1} \times \cdots \times \mathbb{Z}_{q_t}^{d_t} \times \mathbb{Z}^f$$

where the q_i are distinct prime powers. Moreover, this decomposition is unique up to the ordering of its factors.

This is a well-known and well-used result (see for example [Dum04, Section 5.2]). It comes as a direct result of the Structure Theorem for Finitely Generated Modules over PIDs, using the language of rings and modules, but this is out of the scope of this report.

B.3 Character Theory

Most of the material in this section comes from [God93, Section 12.8].

Definition B.3.1 (Characters)
Given a group G, a CHARACTER *of the group G is a homomorphism $G \to \mathbb{C}^\times$, the group of nonzero complex numbers under multiplication. Then G^* will denote the set of characters of G. [God93, Chapter 8]*

(For abelian groups, this corresponds to irreducible degree 1 characters over $\mathbb{C}$ in [Dum04, Section 18.3].)

On G^* a product of characters can be defined by setting

$$\varphi\psi : g \mapsto \varphi(g)\psi(g)$$

under which the character taking each $g \in G$ identically to 1 acts as identity.

Furthermore, for any character $\psi \in G^*$, the map $g \mapsto \psi\left(g^{-1}\right)$ is a homomorphism since

$$gh \longmapsto \psi\left((gh)^{-1}\right) = \psi\left(h^{-1}g^{-1}\right) = \psi\left(h^{-1}\right)\psi\left(g^{-1}\right) = \psi\left(g^{-1}\right)\psi\left(h^{-1}\right)$$

and for any $g \in G$,

$$\psi(g)\psi\left(g^{-1}\right) = \psi\left(g^{-1}\right)\psi(g) = \psi(1) = 1$$

so that this homomorphism is an inverse for ψ.

Therefore, G^* forms a group under the above product of characters.

For the remainder of this section (and the rest of this report), discussion of characters will be restricted to the case of finite abelian groups. Throughout this section, G will denote a finite abelian group of order n.

In this case, by Lagrange's theorem, $g^n = 1_G$ for every $g \in G$, and so for any character $\psi \in G^*$

$$1 = \psi(1) = \psi\left(g^n\right) = \psi(g)^n$$

– that is, the image of each character is contained in the set of n^{th} roots of unity.

Since the inverse of a complex number with modulus 1 is also its complex conjugate, looking at the inversion in G^*,

$$\psi\left(g^{-1}\right) = \psi(g)^{-1} = \overline{\psi(g)}$$

so that the inverse of $\psi \in G^*$ is $\overline{\psi} : g \mapsto \overline{\psi(g)}$.

Theorem B.3.2
For all finite abelian groups G,

$$G \cong G^* \ .$$

Proof. Let

$$G = \mathbb{Z}_{q_1} \times \cdots \times \mathbb{Z}_{q_d}$$

for $q_i \in \mathbb{N}_{>0}$ (Theorem B.2.1). Since all finite abelian groups are isomorphic to a group of this form, it suffices to prove that each group of this form is isomorphic to its group of characters.

For each i, let e_i be the tuple of all zeroes, with a 1 in the i^{th} spot. Then, every element x of G is of the form $x = (x_1, \ldots, x_d) = \sum_{i=1}^{d} x_i e_i$, where each x_i is some element of $\mathbb{Z}_{q_i}$. Furthermore, for all i, let ω_i be *any* primitive q_i^{th} complex root of unity, so that $\omega_i^0, \omega_i^1, \ldots, \omega_i^{q_i - 1}$ are all distinct.

First, we claim that every character ψ is completely determined by its values on $e_1, \ldots, e_d$. To see this, take any group element z, and compute

$$\psi(z) = \prod_{i=1}^{d} \psi(z_i e_i) = \prod_{i=1}^{d} \psi\left(\sum_{j=1}^{z_i} e_i\right) = \prod_{i=1}^{d} \prod_{j=1}^{z_i} \psi(e_i) = \prod_{i=1}^{d} \psi(e_i)^{z_i} \ .$$

Furthermore, note that $\psi(e_i)$ is a q_i^{th} complex root of unity, since $q_i \cdot e_i = 0$ and $\psi(q_i \cdot e_i) = \psi(e_i)^{q_i}$, while $\psi(0) = 1$.

So, define the map $^* : G \to G^*$ by

$$x^* : z \mapsto \prod_{i=1}^{d} \omega_i^{x_i z_i} \ .$$

Then x^* is a character of G since $x^*(0) = \prod_{i=1}^{d} \omega_i^0 = 1$, and for group elements z and $\widetilde{z}$,

$$x^*(z+\widetilde{z}) = \prod_{i=1}^{d} \omega_i^{x_i(z_i+\widetilde{z}_i)} = \prod_{i=1}^{d} \omega_i^{x_i z_i} \omega_i^{x_i \widetilde{z}_i} = \prod_{i=1}^{d} \omega_i^{x_i z_i} \prod_{i=1}^{d} \omega_i^{x_i \widetilde{z}_i} = x^*(z) x^*(\widetilde{z}) \ .$$

Therefore, this map is well-defined.

Next, we claim that this map is a homomorphism. To see this, note that for any group element z, $0^*(x) = \prod_{i=1}^{d} \omega_i^0 = 1$. Then, for any three group elements x, y, and z,

$$(x+y)^*(z) = \prod_{i=1}^{d} \omega_i^{(x_i+y_i)z_i} = \prod_{i=1}^{d} \omega_i^{x_i z_i} \omega_i^{y_i z_i} = \prod_{i=1}^{d} \omega_i^{x_i z_i} \prod_{i=1}^{d} \omega_i^{y_i z_i} = x^*(z) y^*(z)$$

which demonstrates that $(x+y)^* = x^* y^*$.

Finally, we show that this map is bijective. First, note that if a character maps each e_i to 1, then that character is the identity. Since, $x^*(e_i) = \omega_i^{x_i}$, which is 1 if and only if $x_i = 0$, x^* is the identity if and only if $x = 0$. Furthermore, if ψ is any character, then ψ is completely determined by the values $\psi(e_1), \ldots, \psi(e_d)$. Since each $\psi(e_i)$ is a q_i^{th} root of unity, $\psi(e_i) = \omega_i^{x_i}$ for some x_i in $\mathbb{Z}_{q_i}$. Therefore, $\psi = (x_1, \ldots, x_d)^*$. ■

While by transitivity this shows that $G^{**} \cong G$, this can be seen more directly via the isomorphism

$$g \mapsto (\psi \mapsto \psi(g)) \ .$$

Given an ordering $G = \{g_1, \ldots, g_n\}$, the character $\psi \in G^*$ can be identified with the row vector

$$\psi \rightsquigarrow \begin{bmatrix} \psi(g_1) & \cdots & \psi(g_n) \end{bmatrix} \ . \tag{B.1}$$

With this identification, the product of characters becomes the entrywise product of vectors (the *Schur product* of $n \times 1$ matrices), and the inverse of a character in G^* is the entrywise inversion of the vector.

Furthermore, given an ordering $G^* = \{\psi^1, \ldots, \psi^n\}$, the matrix whose rows consist of the characters of G^* is called the CHARACTER TABLE of G:

$$H = \begin{bmatrix} — & \psi^1 & — \\ & \vdots & \\ — & \psi^n & — \end{bmatrix} \ . \tag{B.2}$$

Remarkably, this matrix turns out to be (almost) unitary.

For any subset $C \subseteq G$, define

$$\psi(C) := \sum_{g \in C} \psi(g) = \psi \chi_C$$

where χ_C is the characteristic vector of C in G (with the same ordering). So, given characters ψ, φ, their inner product can be written

$$\psi \varphi^* = \sum_{g \in G} \psi(g) \overline{\varphi}(g) = \sum_{g \in G} (\psi \overline{\varphi})(g) = (\psi \overline{\varphi}) \chi_G = (\psi \overline{\varphi})(G) \ .$$

(Note here that φ^* denotes the conjugate transpose of φ, and χ_G is also the all-ones column vector $\mathbf{1}$.)

Lemma B.3.3
For any $\psi \in G^*$

$$\psi(G) = \begin{cases} |G| & \text{if } \psi \text{ is the identity of } G^* \\ 0 & \text{else.} \end{cases}$$

Proof. For any $h \in G$, since $g \mapsto hg$ is an automorphism of G, $hG = G$, so that

$$\psi(G) = \sum_{g \in G} \psi(g) = \sum_{g \in G} \psi(hg) = \sum_{g \in G} \psi(h)\psi(g) = \psi(h) \sum_{g \in G} \psi(g) = \psi(h)\psi(G)$$

which implies that either $\psi(h) = 1$ or $\psi(G) = 0$.

But this holds for arbitrary $h \in G$, so that either

$$\forall h \in G \ \psi(h) = 1 \implies \psi = 1_{G^*} \quad \text{and} \quad \psi(G) = \sum_{g \in G} 1 = |G|$$

or else

$$\exists h \in G \ \psi(h) \neq 1 \implies \psi(G) = 0 \ .$$

■

Corollary B.3.4
If H is the character table of a finite abelian group G of order n, then $HH^* = nI$, where H^* is the conjugate transpose of H.

Proof. If ψ, φ are characters of G, and $\psi \neq \varphi$, then letting $\theta = \psi \overline{\varphi}$, the (ψ, φ)-entry of HH^* is given by $\psi \varphi^* = \theta(G) = 0$, since θ is not the identity of G^*.

However, for the diagonal, (ψ, ψ)-entries of HH^*, $\psi \psi^* = 1_{G^*}(G) = n$, which proves the claim. ■

C. Notation

$\lvert\cdot\rvert$	the Hamming weight of a vector;
	the modulus of a complex number;
	the cardinality of a set
$\bar{\cdot}$	the conjugate of a complex number;
	the complement of a set;
	in the context of C-cliques, where C is a subset of D containing 0, $\overline{C} = D \setminus C^*$
$\langle \cdot\ ,\ \cdot \rangle$	an inner product
$\circ$	the Schur (entrywise) product of matrices; the composition of functions
$\mathbf{0}, \mathbf{1}$	the vector (or tuple) of all 0s, or 1s
$\mathbb{1}$	the indicator function
$\mathcal{A}$	an association scheme
A^*	the adjoint of A
A_i	the adjacency matrices (Schur idempotents) of an association scheme
A^T	the transpose of a matrix
$\mathbb{A}$	the Bose-Mesner algebra of an association scheme
$\operatorname{Aut}\Gamma$	the automorphism group of Γ
C	a subset of D containing 0;
	an inverse-closed subset of a group
C^*	in the context of C-cliques, where C is a subset of D containing 0, $C^* = C \setminus \{0\}$
$\operatorname{Cay}(G, C)$	the Cayley graph of the group G with inverse-closed subset C
χ	a characteristic vector or matrix
D	the set $\{0, 1, \ldots, d\}$
$D_v(u)$	in a Hamming graph, the distances from u to the neighbourhood of v (including v)
d	the diameter of a graph or association scheme
$[d]$	the set $\{1, \ldots, d\}$
δ	the minimum distance of a code
δ_{ij}	the Kronecker delta: 1 if $i = j$, 0 if $i \neq j$

$\operatorname{diag}(x)$	the diagonal matrix with diagonal x
$\dim(\cdot)$	the dimension of a vector space
$\operatorname{dist}(u, v)$	the distance between vertices u and v in a graph
e_i	the i^{th} standard basis element; a tuple of all 0s, and a 1 in the i^{th} position
F_j	the principal idempotents of an association scheme
G, G^*	a group, its group of characters
Γ	a graph
$\mathrm{GF}(q)$	the finite field with q elements
$H(d, q)$	the Hamming graph of d-tuples over a q-set
H	a character table
I	the identity matrix
J	the all-ones matrix
$M_{i,j}^t$	the generators of a Terwilliger algebra
μ	the multiplicities of an association scheme
$N_i(u)$	the set of vertices at distance i from u
n	the size of a vertex set
ν	the valencies of an association scheme
ω	a complex root of unity
P	the matrix of eigenvalues of an association scheme with $P_i(j)$ in row j, column i
p_{ij}^k	the intersection numbers of an association scheme
ψ	a character of a group
Q	the matrix of dual eigenvalues of an association scheme with $Q_j(i)$ in row i, column j
q_{ij}^k	the Krein parameters of an association scheme
$\rho, \rho*$	a partition, its induced row partition
$S \succeq T$	the matrix $S - T$ is positive semi-definite
σ	a permutation; an automorphism of $H(d, q)$ permuting the coordinates of vertices
$\operatorname{sum} A$	the sum of all entries in the matrix A
$\operatorname{Sym} X$	the symmetric group on X (the set of permutations $X \to X$)
T_v	the automorphism of $H(d, q)$ obtained by adding v to each vertex
τ	a permutation of $\operatorname{Sym}(\mathbb{Z}_q \setminus \{0\})$; an automorphism of $H(d, q)$ induced by this permutation
$\operatorname{trace} A$	the trace of a matrix A: the sum of its diagonal entries
u, v, w	vertices
$V, V(\Gamma)$	a vertex set (of Γ)
$x_{i,j}^t$	parameters in the SDP bound
Y	a code; a subset of V
y, z	the inner distributions of C-cliques Y, Z
$\mathbb{Z}_q$	the integers modulo q

Bibliography

[Ber09] Dimitris Bertsimas. "6.251J Introduction to Mathematical Programming". Fall 2009. URL: https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-251j-introduction-to-mathematical-programming-fall-2009/.

[Del73] P. Delsarte. "An algebraic approach to the association schemes of coding theory". PhD thesis. 1973.

[Dum04] David Steven. Dummit. *Abstract algebra*. eng. 3rd ed. New York: Wiley, 2004. ISBN: 0471433349.

[Fri03] Stephen H. Friedberg. *Linear algebra*. eng. 4th ed. Upper Saddle River, N.J: Pearson Education, 2003. ISBN: 0130084514.

[God93] C. Godsil. *Algebraic Combinatorics*. Chapman Hall/CRC Mathematics Series. Taylor & Francis, 1993. ISBN: 9780412041310.

[Mat07] Jiří Matoušek. *Understanding and using linear programming*. eng. Universitext. Berlin ; Springer, 2007.

[MZ19] S. Morteza Mirafzal and Meysam Ziaee. *A note on the automorphism group of the Hamming graph*. 2019. arXiv: 1901.07784 [math.GR].

[Ple98] Vera Pless. *Introduction to the Theory of Error-Correcting Codes*. eng. New York: John Wiley & Sons, Incorporated, 1998. ISBN: 9780471190479.

[Sch05] Alexander Schrijver. "New Code Upper Bounds From the Terwilliger Algebra and Semidefinite Programming". In: *Information Theory, IEEE Transactions on* 51 (Sept. 2005), pp. 2859–2866. DOI: 10.1109/TIT.2005.851748.

[Win17] Martin Winter. *Number of automorphisms of n-dimensional hypercube graph*. https://math.stackexchange.com/a/2481352. Accessed: 13 May 2021. 2017.