Cliques in Association Schemes

Andrew Nagarajah
Supervised by: Prof. Mike Newman

23 March 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 in particular, 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*. This perspective enables the computation of various parameters of interest, including the eigenvalues of graphs and upper bounds on codes, cliques, and independent sets.

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 is reviewed in the appendix.

Contents

1 Introduction **3**

1.1 Coding Theory . . . 3

1.2 Hamming Graphs . . . 3

1.3 Distance-Regular Graphs . . . 3

2 Association Schemes **4**

2.1 Association Schemes . . . 4

2.1.1 The Bose-Mesner Algebra . . . 4

2.1.2 Duality . . . 4

2.2 P-Polynomial Schemes . . . 4

2.2.1 Q-Polynomial Schemes . . . 4

2.3 Automorphisms and Cayley Graphs . . . 4

2.4 Translation Schemes . . . 7

2.5 The Eigenvalues of the Hamming Scheme . . . 9

3 Delsarte's Linear Programming Bound **10**

3.1 Linear Programming . . . 10

3.1.1 Duality . . . 11

3.2 The LP Bound . . . 12

3.3 The Ratio Bound . . . 14

3.4 The Clique-Coclique Bound . . . 14

4 Schrijver's SDP Bound **15**

4.1 The Terwilliger Algebra of the Hamming Scheme . . . 15

4.2 Semi-Definite Programming . . . 15

5 Computation **16**

A Linear Algebra **17**

A.1 The Spectral Theorem . . . 17

A.2 Adjacency Matrices 17
A.3 Positive Semi-Definite Matrices 17

B Group Theory **18**
B.1 Group Actions 18
B.2 Character Theory 20

C Notation **23**
C.1 Godsil 23
C.2 Delsarte 23
C.3 Schrijver 23

Bibliography **24**

1. Introduction

1.1 Coding Theory

I'm also not sure if I should include a section like this, but seeing as coding theory was the original motivation behind Delsarte's LP bound, and it remains (presumably?) a strong motivator for this theory, I figured it might be interesting to mention this as an application.

- Basics of Coding Theory
- Linear Graphs
- Finite Vector Spaces

1.2 Hamming Graphs

1.3 Distance-Regular Graphs

I'm not sure if I should maybe merge this section with P-polynomial section?

- Definition
- Basic parameters
- Examples?

2. Association Schemes

2.1 Association Schemes

- Definition(s)
- Examples?
- Basic parameters

2.1.1 The Bose-Mesner Algebra

2.1.2 Duality

2.2 P-Polynomial Schemes

- Definitions
- "Equivalence" to DRGs

If we want to focus on the LP bound and translation schemes, I'm not sure that this section is necessary, but it is very interesting and provides an important class of examples.

2.2.1 Q-Polynomial Schemes

2.3 Automorphisms and Cayley Graphs

This section may be merged with the following section.

- Action of a regular group of automorphisms
- Cayley graphs
- Eigenspaces from characters

The material in this section follows mostly from [3, Chapter 9].

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 to compute their eigenvalues significantly more efficiently than otherwise would be the case, as outlined in the previous sections.

Definition 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 vertex set* X *is a map* $X \to X$ *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 (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* $\operatorname{Cay}(G, C)$*, and defined as follows:*

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

Since C *is inverse closed,* $gh^{-1} \in C \iff hg^{-1} \in C$ *so that* $\operatorname{Cay}(G, C)$ *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 (6) on $\operatorname{Cay}(G, C)$ through the group operation, since

the vertex set is also equal to 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 which provides a characterization of Cayley graphs.

(This action is also faithful since for $g \neq h \in G$, the vertex 1 gets mapped to g and h respectively, which are unequal. However this observation irrelevant for this lemma, since it restricts to the action of an automorphism group, which is automatically faithful.)

Lemma 1
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 $C \subseteq G$.

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

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

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 $vg_u = u$.

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.

Lemma 2
Let G be a finite abelian group, let $C \subseteq G \setminus \{1\}$ 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 (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 $G \cong G^*$ (3) implies that the character table is square, the rows form a basis of eigenvectors for A. □

TODO How to get real eigenvectors out of this?

2.4 Translation Schemes

- Equitable partitions of matrices
- Group partitions yielding association schemes
- Dual schemes? (Interesting, but not particularly necessary for the rest of this report)

In a sense, this section generalizes the latter result of the previous section to the setting of association schemes, using the former result. Throughout this section, a transitive, abelian group of automorphisms will replace the regular automorphism group which corresponds to a Cayley graph. As per (5) the transitive, abelian group will act regularly, so that (1) still applies. This motivates the following definition.

Definition 3 (Translation Schemes)
A TRANSLATION SCHEME *is an association scheme whose automorphism group contains a transitive, abelian subgroup.*

Lemma 3
If $\mathcal{A}$ is a translation scheme, and G is a transitive, abelian automorphism group, then there is a partition into inverse-closed sets $C_0, C_1, \ldots, C_d$ of G 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 $\operatorname{Aut}\Gamma_i$ for each $i = 0, 1, \ldots, d$, G acts regularly on Γ_i. Therefore, by (1), 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, $C_0 = \{1\}$ generates the graph.

Otherwise, it suffices to show that $C_0, C_1, \ldots, C_d$ partition G. Recall from the proof of (1) 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 which 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 into inverse-closed subsets of an abelian group which generate a translation scheme from those which do not. [3, Section 12.10]

Definition 4 (Partition Matrix)
If σ is a partition of a set X, then the PARTITION MATRIX *of σ is the* 01 *matrix whose rows are indexed by the elements of X, and whose columns are indexed by the parts of σ, 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.)

Given any matrix H, if σ is a partition of the columns with partition matrix S then the INDUCED ROW PARTITION σ^* 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 HS are equal. In other words, if f is the function which maps each row index i of H to the row vector $(HS)_i$, then σ^* is the partition given by the fibres of f.

Lemma 4
If σ is a column partition of a matrix H with full column-rank, and σ^* is the induced row partition, then $|\sigma^*| \geq |\sigma|$.

Proof. Let S be the partition matrix of σ. If $HSx = 0$ then since the columns of H are linearly independent, $Sx = 0$, and since the columns of S are linearly independent, $x = 0$. Therefore the columns of HS are linearly independent. Since there are $|\sigma|$ columns of HS, $\operatorname{rank} HS = |\sigma|$.

Similarly, if R is the partition matrix of $\sigma*$, then $\operatorname{rank} R = |\sigma^*|$. Therefore, to prove $|\sigma| \leq |\sigma^*|$ it suffices to show that $\operatorname{col} HS \subseteq \operatorname{col} R$.

However, σ^* is defined so that whever two rows of HS are equal, the rows will belong to the same part of σ^*, and thus the rows will be equal in R (this should be clear upon contemplating the definition of R). Therefore, for any vector in the column space of HS, where R has equal rows the vector is guaranteed to have equal components (since those rows of HS will be equal) so that the vector will also belong to the column space of R. □

Note in fact that the same argument will apply to any refinement of $\sigma*$, and will show that $\operatorname{col} HS \not\subseteq \operatorname{col} R$ for any partition which is not a refinement of σ^*. [3, Section 12.7]

TODO

2.5 The Eigenvalues of the Hamming Scheme

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

$$C_i := \left\{ x \in \mathbb{Z}_q^d \;\middle|\; \text{there are exactly } i \text{ 0's in } x \right\} .$$

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

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

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.

3. Delsarte's Linear Programming Bound

3.1 Linear Programming

- Basics of Linear Programming – done
- Duality – done
- Algorithms?

The terminology and results from this section, except for the adjective *principal* for constraints, follows from [4].

A LINEAR PROGRAMMING PROBLEM (or LINEAR PROGRAM) is an optimization problem in which one seeks to maximise 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 contraints 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 [1], 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 constrast 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:

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

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

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

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

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

Re-writing it in standard form,

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

taking the dual

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

and re-writing in standard form

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

the original (called PRIMAL) linear program is recovered.

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

Theorem 1 (Weak Duality)
If x is a feasible solution to a linear program

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

and y is a feasible solution to its dual program,

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

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

Definition 5

The INNER DISTRIBUTION *is TODO. I might also put this in the section on association schemes; I'm not sure if it belongs better there or here.*

Theorem 2 (Delsarte Thm 3.3 [1])

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.)

Proof. TODO. Note that this will require a number of lemmas which I've omitted here for brevity, but will include in the final product. □

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 M-clique with inner distribution y. Then $y_i = 0$ for all $i \notin M$, so $Q^T y \geq 0 \iff Q^T \operatorname{diag}(\chi_M)\, y \geq 0$ since the action of $\operatorname{diag}(\chi_M)$ acting on the left is to zero out the *rows* of y with index not in M. Similarly,

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

since the action of $\operatorname{diag}(\chi_{M^*})$ on the right is to zero out the *columns* of Q^T with index not in M^*, $y_0 = 1$, 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 $Q^T y \geq 0 \iff \operatorname{diag}(\chi_{N^*})\, Q^T y \geq 0$.

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

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

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

Taking the dual yields

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

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

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

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

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

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

3.3 The Ratio Bound

We will use (3.5) frequently.

3.4 The Clique-Coclique Bound

4. Schrijver's SDP Bound

4.1 The Terwilliger Algebra of the Hamming Scheme

4.2 Semi-Definite Programming

5. Computation

I wasn't sure if I ought to mention anything about the code I've written for this project (or even if there's anything worth saying that won't be covered elsewhere in the report).

Also, if there are some specific results that would be interesting to show, but do not fit naturally into other sections of the report, then perhaps they could go here as well.

A. Linear Algebra

A.1 The Spectral Theorem

A.2 Adjacency Matrices

Basic results about the spectra of adjacency matrices, which may be used elsewhere in the report. E.g. the sum of eigenvalues with multiplicity, and consequences.

A.3 Positive Semi-Definite Matrices

Depending on which proof of the clique-coclique bound I use, and how much detail I go into Schrijver's SDP bound, I could make some comments about PSD matrices.

B. Group Theory

B.1 Group Actions

The material of this section comes primarily from [2, Section 1.7, Chapter 4].

Definition 6 (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 [2] 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 7 (Types of Group Actions)
If 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 [3].)*

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 identity $G \hookrightarrow \operatorname{Sym} X$, which is clearly injective.

Lemma 5

If an abelian group G acts faithfully and transitively on a set X, then the action is free, and thus also regular. [2, 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.

B.2 Character Theory

Definition 8 (Characters)
Given a group G, a CHARACTER *of the group G is a homomorphism $G \to \mathbb{C}^\times$, the group of non-zero complex numbers under multiplication. Then G^* will denote the set of characters of G. [3, Chapter 8]*

(For abelian groups, this corresponds to irreducible degree 1 characters over $\mathbb{C}$ in [2, 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 3

For all finite abelian groups G,

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

Proof. TODO □

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

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

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:

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

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 6

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 1

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

I've included these sections mostly as an excuse to add the citations to the bibliography, though it may be somewhat useful to have a sort of guide to the similarities and differences in notation in this report, as well as in the references. (At least, I would find such a thing useful.)

C.1 Godsil

See [3] Ch. 10.

C.2 Delsarte

See [1] Ch. 3.

C.3 Schrijver

See [5].

Bibliography

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

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

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

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

[5] 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.